Martin Davis
Edmond Schonberg *Editors*

From Linear Operators to Computational Biology

Essays in Memory of Jacob T. Schwartz

Springer

From Linear Operators to Computational Biology

Martin Davis • Edmond Schonberg
Editors

From Linear Operators to Computational Biology

Essays in Memory of Jacob T. Schwartz

Springer

Editors
Martin Davis
Professor (emeritus) Mathematics and Computer Science
Courant Institute of Mathematical Science
New York University
New York, USA

Edmond Schonberg
Professor (emeritus) Computer Science Department
Courant Institute of Mathematical Science
New York University
New York, USA

ISBN 978-1-4471-4281-2 ISBN 978-1-4471-4282-9 (eBook)
DOI 10.1007/978-1-4471-4282-9
Springer London Heidelberg New York Dordrecht

Library of Congress Control Number: 2012944721

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Jacob T. Schwartz, 1930–2009

Recollecting Jack Schwartz

Jack had the most powerful mind of anyone I knew, except von Neumann. These two had in common a fantastic ability to learn a new subject extremely rapidly; both had an incredibly wide range of interest.

When Jack left the Department of Mathematics to lead the Department of Computer Science newly created by him, he related to me that when he came to the Courant Institute he had decided to teach all courses listed in our bulletin, and that he had carried out his plan. When I pointed out that there must have been some subjects about which he knew little or nothing, he replied that in those cases he took from the library the Summer before the leading text on the subject and learned it. When I asked him if there was a subject he had trouble learning, he admitted that he did, fluid dynamics. "It is not a subject that can be expressed in terms of theorems and their proofs".

Jack first came to the attention of the mathematical community as the coauthor of Dunford-Schwartz, an impressive exposition of functional analysis. It was much more than a compilation and organization of known material; there is much original work in it. A striking example is the theorem that the trace of a trace class operator in Hibert space is the sum of its eigenvalues. This theorem and its proof are presented in Vol. II, but no attribution is given to Lidskii, to whom it is due. Since D-S is compulsive about giving references, this was mystifying, The explanation is that D-S vol. II came out before the publication of the English translation of Lidskii's paper; Jack had proved the result independently.

Jack has made valuable contributions to operator algebras, a subject founded by von Neumann and Murray. Jack described von Neumann's work as "coming repeatedly to a stone wall and crashing through it."

Jack had a delicious sense of humor. In a paper titled "The pernicious influence of mathematics on science", he starts with the observation that "computer intelligence" has three major shortcomings: single-mindedness, literal-mindedness, and simple-mimdedness. He then makes the point that mathematics also has these shortcomings, although to a lesser extent. As an example he points to the claim that the Birkhoff ergodic theorem is the basis of the foundation of statistical mechanics, and then neatly demolishes it.

As another example Jack quotes Keynes' criticism of some mathematical economics as "...a mere concoction, as imprecise as the initial assumptions they rest on, which allow the author to lose sight of the complexities and interdependencies of the real world in a maze of pretensions and unhelpful symbols."

Hans Bethe once remarked,only half in jest,that von Neumann's brain was an upward mutation of the human brain; the same could have been said about Jack.

We shall not see the like of him for a long time.

Courant Institute of Mathematical Sciences Peter D. Lax
NYC, NY, USA

Contents

Contributors

Domenico Cantone Dipartimento di Matematica e Informatica, Università di Catania, Catania, Italy

Martin Davis Courant Institute of Mathem. Science, New York University, New York, NY, USA

Robert Dewar Courant Institute, New York University, New York, USA; AdaCore Technologies, New York, USA

Alfredo Ferro Dept. of Mathematics and Computer Sciences, University of Catania, Catania, Italy; Dept. of Biomedical Sciences, University of Catania, Catania, Italy

David Ritz Finkelstein Georgia Institute of Technology, Atlanta, GA, USA

Rosalba Giugno Dept. of Mathematics and Computer Sciences, University of Catania, Catania, Italy

Alessandro Laganà Dept. of Mathematics and Computer Sciences, University of Catania, Catania, Italy; Dept. of Biomedical Sciences, University of Catania, Catania, Italy

B. Mishra Courant Institute, New York, NY, USA

Eugenio G. Omodeo Dipartimento di Matematica e Informatica, Università di Trieste, Trieste, Italy

Alfredo Pulvirenti Dept. of Mathematics and Computer Sciences, University of Catania, Catania, Italy

Edmond Schonberg Courant Institute of Mathem. Science, New York University, New York, NY, USA

Micha Sharir School of Computer Science, Tel Aviv University, Tel Aviv, Israel; Courant Institute of Mathematical Sciences, New York University, New York, NY, USA

Michael Wigler Cold Spring Harbor Laboratory, Cold Spring Harbor, NY, USA

From Linear Operators to Computational Biology: Essays in Honor of Jacob T. Schwartz

Martin Davis and Edmond Schonberg

Abstract We present a thumbnail sketch of the scientific career of Jack Schwartz, in order to place the essays in this volume in the proper chronological context.

1 Introduction

This volume of essays honors the memory of Jacob T. Schwartz ("Jack") a distinguished mathematician and computer scientist who was our friend, colleague, mentor and inspiration in the course of an amazingly prolific trajectory as a scientist and teacher.

In his long and distinguished career as a mathematician and computer scientist, Jacob T. Schwartz made numerous important contributions to a remarkable variety of different subject areas. His style was to enter a new field, to master quickly the existing research literature, to add the stamp of his own forceful vision in a series of research contributions, and finally to leave behind an active research group that would continue fruitful research for many years along the lines he laid down. A brief list of some of the areas to which Schwartz made major contributions will give some notion of the breath of his interests: spectral theory of linear operators, von Neumann algebras, macro economics, the mathematics of quantum field theory, parallel computation, computer time-sharing, high-level programming languages, compiler optimization, transformational programming, computational logic, motion planning in robotics, multimedia and educational software, and finally computational biology. He had an enormous talent for conveying his enthusiasms, and the essays that follow reflect the impact that he has had on the scientific interests of his students and collaborators.

M. Davis (✉) · E. Schonberg
Courant Institute of Mathem. Science, New York University, Mercer Street 251, New York, 10012 NY, USA
e-mail: martin@eipye.com

E. Schonberg
e-mail: schonberg@adacore.com

M. Davis, E. Schonberg (eds.), *From Linear Operators to Computational Biology*,
DOI 10.1007/978-1-4471-4282-9_1,

This is not the place for a full biography, and the brief sketch that follows only aims to place the essays in this volume in the perspective of Jack's interests over the decades. A fuller biographical essay can be found in [1].

Jack Schwartz began his career in mathematical analysis, specifically in the theory of linear operators. The monumental three volume treatise *Linear Operators* written with Nelson Dunford, for which the authors were awarded the Leroy P. Steele prize by the American Mathematical Society, is not only the definitive work in the area, but is also a wonderful compendium bringing together results from various branches of analysis and much that was new, all presented in an exciting manner. What is probably Schwartz's best known work in analysis settled an important problem regarding von Neumann algebras that had been posed by von Neumann himself.

Inevitably, his work on Hilbert Spaces led him to theoretical Physics and the foundations of quantum theory. The essay by David Finkelstein *Nature as Quantum Computation* responds to scientific discussions with Jack over the past half-century.

In the nineteen fifties, While participating in a study of Karl Marx's classic *Capital* with a like-minded group in New York, Jack came to the conclusion that Marx had failed to confront adequately a contradiction between his economic analysis and empirical reality. Jack developed some simple economic models in an unavailing effort to convince the others in the group. Later, examining both Marx and Keynes, through the eyes of a supremely gifted mathematician, he produced a sharp critique of their work, and synthesized his ideas on economic phenomena. A selected bibliography of Jack's works can be found in the appendix. The genesis of this work is described by Martin Davis in *Jack Schwartz meets Karl Marx*.

The pioneering *Compilers and Computer Languages* by John Cocke and Jack Schwartz was the first systematic treatise on the problems of translation of programming languages. It provided a general framework for understanding complex iterative and elimination methods for solving global optimization problems and initiated extensive research seeking to implement these methods efficiently. These methods are the heart of modern compilers and static analysis techniques for programming languages.

Jack was a prolific software writer. His experiences with the programming languages available in the 1960s led him to reflect on the enormous difficulties involved in going from a mathematical description of an algorithm (what a mathematician would consider the completion of the task) to its embodiment in a running program. These reflections led to *On Programming* a treatise on software construction and an encyclopedic survey of Algorithms in different field of Computer science. The centerpiece of the work is a new programming language: SETL (for Set Language). SETL, though never widely adopted in its original form, proved highly influential and useful. Drawing on the experience of mainstream mathematics that has made set theory its lingua franca, SETL makes the task of the programmer easier by drawing on a fixed but powerful collection of set-theoretic primitives in terms of which data structures can be modelled and algorithms can be specified clearly and succinctly. Apart from its pervasive influence in the presentation of algorithms in the literature (where the term "SETL-like notation" is commonly used) SETL proved to be well-suited for what came to be known a decade later as Software Prototyping. The development of the first validated Ada compiler, written in SETL in the form of an operational definition of the language, showed the usefulness of very-high level languages in the construction of executable specifications. Robert Dewar discusses the impact of SETL and its influence on modern programming languages in *SETL and the Evolution of Programming*.

Developing efficient implementations, that is to say optimizing compilers for SETL and similar languages has been the focus of considerable effort over the decades. Two major goals for compiling SETL programs into efficient code, recognized from the outset, were (1) the automatic selection of appropriate storage structures for abstract sets and maps, and (2) the implementation of mathematical value semantics (as opposed to reference semantics) while avoiding expensive copy operations in imperative languages. To these ends, Jack extended the methods of global analysis to deal with three new optimization problems: type inference, analysis for set theoretic inclusion and membership, and alias analysis. Schwartz's pioneering work in type inference has been particularly influential. Schwartz also examined more general optimizing transformations that could affect the asymptotic behavior of algorithms by generalizing finite difference techniques to apply to iterative programs.

Schwartz's hope was that the set theoretic locutions that form the basis of SETL as well as of contemporary mathematics could make possible a seamless link between the two, in which computer generated proofs in set theory could serve to insure the correctness of programs. Towards this end, he developed correctness preserving transformational methods, to allow the step-wise refinements of programs originally written in high-level concise fashion, into efficient programs written at the semantic level of C. In addition he initiated a research program investigating decidable fragments of set theory. This program has been actively pursued his students and collaborators. The papers by Cantone *Decision procedures for elementary sublanguages of set theory XVII: commonly occurring extensions of multi-level syllogistic* of which Jack is a posthumous author, and by and Omodeo *The Ref proofchecks and its common shared scenario* present recent results in this fruitful area, which lies at the intersection of logic, theorem proving, and program-proof technology.

Although parallel processing is now recognized as a major discipline in computer science, this was hardly the case almost four decades ago when Jack Schwartz began his work in this area. A paper published in 1966 introduced a class of architectures he called *Athene*, nowadays called shared-memory MIMD computers. In this early work, Schwartz already anticipated the major challenges in obtaining an effective implementation of this class: memory latency and contention. Like its modern counterparts, the Athene architecture featured a uniform address space, "public" and "private" variables, and special coordination primitives. Although the technology of the sixties was clearly inadequate for the actual construction of an Athene computer, interesting simulation studies were carried out. By the beginning of the eighties, Schwartz felt that technology had caught up. His seminal paper *Ultracomputers* presented a collection of parallel algorithms for a computer using a shuffle-exchange interconnection network complete with complexity analyses. This work served to initiate a substantial research effort in both hardware and software design, that continued at NYU for the following 15 years.

After parallel computing, Jack turned his interests towards Robotics, both in its theoretical aspects (kinematics, motion planning, the Physics of grasping) as well as its industrial applications. During his tenure as head of the Robotics laboratory that he created, he wrote seminal papers on motion planning (*The piano Movers problem*). The paper by Sharir: *Jack Schwartz and Robotics*: *the Roaring eighties* describes some of the pioneering work of the Robotics Lab. The paper by Mishra: *Mathematics' Mortua Manus*: *Discovering Dexterity* discusses recent work in the theory of prehension.

From Robotics Schwartz turned his interests to the then burgeoning field of multimedia, and created the Multimedia Laboratory at NYU. From there it was computational biology, a topic in which he created some of the first courses on the topic as NYU. His late involvement in the field is described by Mike Wigler in *The last ten yards*.

References

1. Davis, M., Schonberg, E.: Jacob Theodore Schwartz (1930–2009). A Biographical Memoir. National Academy of Sciences, Washington (2011)

Nature as Quantum Computer

David Ritz Finkelstein

Abstract Set theory reduces all processes to assembly and disassembly. A similar architecture is proposed for nature as quantum computer. It resolves the classical space-time underlying Feynman diagrams into a quantum network of creation and annihilation processes, reducing kinematics to quantum statistics, and regularizing the Lie algebra of the Einstein diffeomorphism group. The usually separate and singular Lie algebras of kinematics, statistics, and conserved currents merge into one regular statistics Lie algebra.

1 Quantum Theories

I once asked Jack Schwartz what the difference was between mathematics and physics. At the time both were just equation-juggling to me. He was strap-hanging homeward from Stuyvesant High School, where we had just met, and he answered by drawing a hat in the subway air with his free hand:

(1)

He explained that the bottom line is the real world and the top line is a mathematical theory. At its left-hand edge we take data from the real world and put them into a mathematical computation, and at the right-hand side we compare the output of the computation with nature. The loop closes if the theory is right.

This diagram also applies to quantum systems, if the statistical nature of quantum theory is taken into account. Then the bottom line is not one experiment on the system but a statistical population of them.

The question remains of what the symbols of mathematics mean to a mathematician. Some decades later I asked Jack Schwartz what "1" means, and he replied that

D.R. Finkelstein (✉)
Georgia Institute of Technology, Atlanta, GA 30342, USA
e-mail: finkelstein@gatech.edu

M. Davis, E. Schonberg (eds.), *From Linear Operators to Computational Biology*,
DOI 10.1007/978-1-4471-4282-9_2, © Springer-Verlag London 2013

it means itself. This took me aback. I had not considered that possibility. Symbols generally mean something not themselves. Memorandum:

$$1 = \text{“1”}. \tag{2}$$

After mathematizing aspects of economics, robotics, computing, relativity, and even knots, with a theory that predicted when ropes would slip on capstans, Jack Schwartz turned to the problem of putting quantum theory on firm foundations. His concern was not with mathematical rigor but physical. In a paper on “the pernicious influence of mathematics on science”, obviously conversing with Wigner’s famous lecture at the Courant Institute on “the unreasonable effectiveness of mathematics in the natural sciences” [14], Schwartz wrote:

> The mathematical structure of operators in Hilbert space and unitary transformations is clear enough, as are certain features of the interpretation of this mathematics to give physical assertions, particularly assertions about general scattering experiments. But the larger question here, a systematic elaboration of the world-picture which quantum theory provides, is still unanswered. Philosophical questions of the deepest significance may well be involved. Here also, the mathematical formalism may be hiding as much as it reveals. [10]

I respond to this call here, though my answer might not be acceptable to him. He takes it for granted that quantum theory provides a mathematical world-picture, faithful or not, as classical physics did. It is not clear how literally he intended this. Some writings on physics assume that there are complete world pictures; in Gödel’s sense of deciding all well-formed questions, not Bohr’s weaker one, of answering all physical questions that can be answered. Bohr’s “complete” is von Neumann’s “maximal”. One of the critical differences between quantum and classical physics is that quantum physics denies the existence of complete world pictures, yet asserts the existence of merely maximal ones. Perhaps mathematics is the most “pernicious influence” when it has been the most “unreasonably effective”.

Classically, a mathematical model of a physical system is an isomorphism between physical predicates about the system and mathematical predicates about the model. Physical predicates are defined by physical processes of filtration or categorization. The stock example is a polarizing filter.

Boole already defined predicates by “acts of election”. Mental predicates were mental acts in his theory; physical predicates are physical acts in quantum theory. There are then both input and outtake predicate logics, corresponding to input and outtake filtrations. These are dual lattices. They may be operationally defined as the Galois lattices of the relation “Inputs through filter A do not trigger counters following filter B”.

Mathematical predicates, however, are defined axiomatically and form Boolean predicate lattices by fiat. Since models have Boolean predicate lattices and quantum systems have projective predicate lattices, the two cannot be isomorphic. Quantum systems do not have exact mathematical models because they have different logics than models.

Here the pernicious influence of mathematics is the highly infectious conviction that there must be an objective public reality, despite empirical evidence to the contrary. The mathematics of the quantum theory was invented shortly before the quantum theory was born, demonstrating the unreasonable effectiveness of mathematics. The quantum evolution has made mathematics more important for physics, not less. Mathematics still provides classical models for classical theories and now it also provides statistical models for quantum systems, in the following sense.

In classical mechanics a "state" completely describes the system, answering all experimental questions about it. Plato said clearly that what is real must have an objective state.

Quantum thought relinquishes this idea. This is no great loss, since we never came close to having the state of any physical system. When theorists speak of the state of Mars, they often mean its position and momentum, or possibly its orientation and angular velocity as well, ignoring trillions of coordinates of Martian atoms.

Some give the word "state" a new statistical meaning that can serve quantum theories, but the result has been continuing widespread confusion. We should keep the old meaning of complete information for "state" until the dust settles, so we can tell our students something important: quantum systems have none.

A *pure* population is one that cannot be imitated by mixing statistically distinguishable ones. It goes with an atom of the lattice of predicates. Classically, a pure state, when such exist, is described by a probability distribution supported by one point: pure implies complete. A classical point particle with continuous coordinates, however, already has no pure state, because a point in a homogeneous continuum has probability 0.

In quantum theory, all systems admit pure probability distributions, represented by points in a projective geometry, of probability measure 1, not 0; and in analytic projective geometry, by a ray $\{\lambda\psi\} \subset \mathcal{V}$ in a linear space $\mathcal{V}$ that we associate with the system.

Von Neumann was drawn to projective geometries that have a continuous range of dimensionalities. They can have no singlets. This ignores the problem of the infinities; singlets are the solution. Classical space-time already has no singlets.

As Malus found for linear polarization, the transition probability from one predicate ray to another is the squared cosine of the angle between them, almost never 0 or 1. Heisenberg therefore called a vector with this probabilistic interpretation a "probability vector". Its components are probability amplitudes, their squares are relative probabilities. Quantum logic is a square root of classical logic.

But a lone classical system has no probability distribution, nor has the individual quantum system. Schrödinger and many others believed, at least at first, that a vector ψ, up to a factor, exists and evolves in a single quantum experiment, and that it was the state of a real physical object, a wave running around the atom. For brevity, call holders of this ontological interpretation "wavers", and holders of Heisenberg's more pragmatic one "chancers". Since the state of a real wave is indeed a wave-function. the term "wave-function" is a Trojan horse, smuggling waves into the camp of the chancers.

On the other hand, a population of similar experiments has at least two probability vectors or distributions, one for the input operation and a dual one for the outtake operation. To cope with this ancient duality wavers say that "the state vector collapses" from one to the other during a measurement. This locution is not part of Heisenberg's or Kolomogorov's probability theory, for a probability distribution or vector, classical or quantum, is unaffected by what happens to one member of its statistical population. It is not necessarily part of von Neumann's system either, since it was contributed to his book on quantum mechanics by a friend, and does not seem to occur in his later writings. When we interact with one quantum in a statistical population, we may wish to transfer the quantum to another statistical population but neither probability distribution changes; the quantum does.

Quantum theory eliminated what had been taken for granted: the possibility of a mathematical model of the physical system. Some classical conceptions assumes a complete picture of nature, as though taken from a preferred viewpoint outside nature. The corresponding quantum conception would be a plurality of partial representations of interactions with small parts of nature.

It is not obviously possible to visualize an atom completely, since on the atomic scale photons are not immaterial messengers but massive projectiles. Receiving them changes us, so emitting them must change their sources as much. On the atomic scale this reaction by the atom is greater than the action upon us is in the human scale. The smaller the system, the more its perception exhibits hysteresis, memory, non-commutativity. Moreover, we see one atomic transition by absorbing one photon that we cannot share. Quantum perception is ultimately private as well as non-commutative.

Yet all these impediments to depiction could have classical models. They make room for quantum theory, but do not determine its specific features. If it is obvious that complete descriptions are impossible, it is amazing that maximal descriptions are possible. Partially order classes (predicates) by proper inclusion, and call classes that are n steps away from the empty class, n-plets. The points of the projective geometry are the singlets the maximal descriptions of the quantum theory.

A classical doublet includes exactly two singlets. It cannot have a continuous symmetry group, only S_2.

A quantum doublet has a theoretical infinity of statistically distinguishable singlets and an SO(2) symmetry.

Before studies of polarization, no physicist came close to a singlet of any system; Newton prepared polarization singlets with his crystals of Iceland spar. It is likely that when we reach the bottom of the world, we will find polarizations and spins on the beach; not strings, which have no singlets.

Heisenberg called his probabilistic brand of physics non-objective; it does not represent objects but laboratory actions and their probabilities. In the language of categories, a classical system can be presented as a category, whose objects are its states. There is a category of quantum systems too, but one quantum system is not a category, precisely because it has not enough objects (states, identities) in the categorical sense. Instead it is represented by an operator algebra, and only statistically.

"Philosophical questions of the deepest significance" are indeed involved. Jack Schwartz agreed with Kolmogorov that probability was not objective, that physics ought to be objective, and that therefore probability had no role in a fundamental physical theory. Some physicists who use quantum theory to great effect declare that they do not understand it, and expect it to devolve into a more objective theory. This likely results from a deep philosophical preference for objects, which are supposed to be knowable "as they are". This may be a pernicious influence of the unreasonable success of classical mathematical models in astronomy. But it might be innate, as if we are hard-wired to see objects.

Exercises in physics often give a complete mathematical model of the system. This does not prepare the student for quantum physics. No one has ever encountered anything near a complete description of any physical system, classical or quantum. If quantum theory is right, there are none. Those who study physical systems only through such mathematical models may find it absurd and incomprehensible to say that physical systems have none.

Again, some note that quantum theory is "merely" instrumental, and find it unsatisfactory on that ground. This conveys more about the critic's philosophy than about the quantum theory. A Beethoven score too is "merely" instrumental. Physics is a performing art, and physicists are the performers, not the spectators; experimenters, not observers. The relation of a physical theory to physics is that of a menu to a meal. It is natural but naive to think that anything in nature has a complete objective description; in the sense that stoning the villain in a movie is naive.

Again, Heisenberg has been criticized by wavists for providing no mathematical description of the measurement process in his theory; although his main point is that none exists.

We can view any regular quantum system and its co-system as a quantum digital computer and its user. The quantum universe as computer system I sometimes call Qunivac for short. Qunivac differs crucially from artificial computers, however: It has no fixed hardware; it is all quantum jumping. It includes both computer and user. The interface between them is relatively fixed in artificial computers, but highly movable in Qunivac. Science is a Promethean attempt to hack into Qunivac.

1.1 Canonical Quantum Theories

Classical momentum and position coordinates commute: $pq - qp = 0$. Classical physicists were never aware of this as a physical hypothesis; it seems to have been an unconscious assumption. Canonical quantization corrected this commutation relation to $\check{p}\check{q} - \check{q}\check{p} = i\hbar$. This has led physicists to search out other unconscious assumptions, make them conscious, and test them.

One operational meaning of this non-commutativity is that filters defining predicates about p and q do not commute. Such quantum or non-commutative logic was found in the laboratory by Newton for polarizations of light corpuscles, and was described in a quantum-theoretical way by Malus, who unwittingly used a two-dimensional real Hilbert space of linear polarizations. When Boole first axiomatized what eventually became Boolean algebra, he noted excitedly that such a non-commutative logic was possible, without bringing up Newton's polarizers.

1.2 Regular Quantum Theories

In kinematics we represent our physical operations on the system by operators on a space of probability vectors. A *regular* quantum system is one with a finite-dimensional probability vector space, like a spin or a system of spins [1]. Its Lie algebra is simple. Its observable or normal operators have finite spectra. A regular theory still mentions infinities, such as the real number system $\mathbb{R}$, but these result from regarding the co-system as infinite and are harmless if we abstain from questions about the co-system.

2 Yang Space-Times

Quantizing space-time to avoid infinities was proposed in about 1930 by Heisenberg, Ivanenko, and others. The first example was provided by Snyder in 1947 and its commutation relations were immediately simplified by Yang to those of $\mathfrak{y} := \mathrm{so}(5; 1)$, precisely for the sake of simplicity [15]. A *Yang space-time* (in the general sense) is one whose orbital variables span a semisimple Lie algebra, called the Yang Lie algebra.

Yang so(5, 1) is not conformal so(5, 1). It defines a quantum space, while conformal so(5, 1) acts on a classical one. Snyder and Yang did not complete their regularizations but continued to represent their Lie algebras in the singular su(∞) of Hilbert space.

Earlier, R.P. Feynman had quantized space-time by replacing continuous space-time coordinates by sums of Dirac spin operators (apparently unpublished), which also leaves Hilbert space; though he broke off this work in an early phase to study the Lamb shift for Bethe. Feynman quantized space-time but not momenta. The Yang model quantizes space-time and momentum-energy, but is still singular. The Penrose position vector **x** is a finite sum of Pauli spin operators; the momentum vector is not represented.

Here are the quantum space-time variables of Feynman, Yang, and Penrose, in quantum units:

$$\begin{array}{lll} \text{Feynman [3]} & \delta\check{x}^{\mu} = \gamma^{\mu}, & \\ \text{Yang [15]} & \check{x}^{\mu} = L^{5\mu} = i\eta^{[5}\partial^{\mu]}, & \check{p}_{\mu} = L_{6\mu} = i\eta_{[6}\partial_{\mu]}, \\ \text{Penrose [8]} & \delta\check{x}^{k} = \sigma^{k}. & \end{array} \tag{3}$$

In such theories, particles do not define irreducible unitary representations of the Poincaré group as Wigner proposed, but irreducible normal representations of a slightly different simple Lie algebra. The physical constants of the Feynman or Yang groups are the speed and action units $c, \bar{h}$ of earlier group de-contractions, with additional elementary time and energy units X and E with $\mathsf{XE} = \bar{h}$, and a huge integer N. The quantum units X of time and E of energy are here called the chrone and the erge; it is not yet clear whether these are the Planck units. Under the Yang relativity group, time in chrones in one frame is just energy in erges in another; time converts into energy, mass. The conversion factor is a huge quantum of power, one erge per chrone, perhaps the Planck power. The conversion is not easy but requires melting the vacuum organization that distinguishes energy and time.

The synthesis described here is finite-dimensional, quantizes all the orbital and field variables, and includes spin. The other internal coordinates of the Standard Model are easily tacked on, but it would be disappointing if the quantization of space-time did not lead us to a deeper synthesis of the internal variables.

Einstein's local equivalence principle again suggests that space-time must be quantized. Following Galileo, Einstein equates a gravitational field and an acceleration. Since the field is quantized, so is the acceleration. This is a second time derivative of spacial coordinates with respect to time; if the space and time coordinates were commutative, the acceleration would be too.

Since i and $-i$ are interchanged by Wigner time reversal, they can be regarded as two values of a discrete dynamical variable i that happens to be central. This centrality makes the Heisenberg Lie algebra singular and leads to infinities. The Yang simplification suspended the centrality of i. It is therefore a real quantum theory of the Stückelberg kind [13]. For correspondence with the standard complex theory, Yang provides a quantized imaginary $\check{i} \in \mathbb{S}$ whose classical correspondent is i, with $\check{i} \multimap i$, but a self-organization must be invoked to single one i out of many possibilities.

The Standard Model spinlike groups can all be defined by their actions on a probability vector space of about 16 dimensions. The orbital group seems to act on a much larger number of dimensions $\mathsf{N} \gg 16$. Therefore events of history are not randomly scattered but highly organized locally into something like a thin truss dome in four dimensions. This dome must support the particle spectrum, sharp bands of highly coherent transmission, and so is presumably crystalline, as Newton inferred from transverse photon polarization.

Regular space-times call for regular Lie algebras. Here is one suggested by those of (3), based on the contraction spin(3, 3) $\multimap$ $\mathfrak{hp}(3,3)$:

$$\begin{aligned}
\check{x}^{\mu} &= L^{5\mu} = \overline{\psi}\gamma^{[5\mu]}\psi, \\
\check{p}_{\mu} &= L_{6\mu} = \overline{\psi}\gamma_{[6\mu]}\psi, \\
\check{i} &= \mathsf{N}^{-1}L_{65} = \overline{\psi}\gamma^{\top}\psi.
\end{aligned} \tag{4}$$

Here ψ is a chronon IO operator and $\overline{\psi} = \beta\psi$ is its Pauli adjoint. The combination $\overline{\psi}\dots\psi$ is the usual covariant accumulator, summing many replicas of its argument with attention to polarity. The eight γ^{ν} generate the Clifford algebra of spin(4, 4), with top (volume) element $\gamma^{\top} = \gamma^{8}\dots\gamma^{1}$.

3 Whither Physics?

First the ancient axioms of space and time failed us in physical experiments and then the axioms of Boolean logic. There are now well-known physical geometries and lower-order physical logic side by side with the older mathematical ones. Higher-order logic is evidently next in line to join the empire of the empirical.

Define an operational theory of a system as a semigroup whose elements are the feasible operations on the system by the co-system (the rest of the cosmos, including us), provided with their probabilities. The kinematics gives the possibilities, the dynamics attaches probabilities.

In quantum theories, each operation is represented by a projective transformation of a specified projective geometry whose points are singlet quantum input or outtake operations. A special conic section in the projective geometry defines transition probability amplitudes.

The first-order logic of the system is the sub-semigroup of filtration operations. The set theory of the system deals with operations of system assembly and disassembly.

Our operations ordinarily rely on the organization of the co-system, natural or artificial, as do operating manuals or cookbooks. Highly organized complex elements of the co-system may enter the system theory only through several of their many parameters.

Thus the idea of a Universal Theory is absurd. Measurements do not bring us ever closer to Truth, but invalidate earlier facts as fast as they validate new ones, and omit much about the co-system. A dynamical law cannot be universal if it is overridden whenever we measure anything.

Then what shapes our course? Two processes by which physical theories evolve correspond to biological evolutionary processes studied by Charles Darwin and Lynn Margulis. The Darwinian one is implicit in Yang space-time and explicit in work of Segal: *Physical theory evolves towards semisimple Lie algebras* [11].

Almost all Lie groups have regular Killing forms. A singular Killing form is a very rare fish; the least change in its structure tensor can regularize it. Singularity is structurally unstable. As measurements of structure constants improve, a singular Lie algebra has survival probability 0 relative to its regular neighbors, which outnumber it ∞-to-1.

Gerstenhaber, influenced by Segal, described homologically a rich terrain of Lie groups connected by contractions that carry groups out of stable valleys of simplicity, along ridges between the valleys, and up to singular peaks [5]. According to the simplicity principle, physics today is a glacier flowing down the simplicity-gradient to valleys in Gerstenhaber-land.

Thus the Galileo Lie algebra lies on a singular ridge between the valleys of Lorentz so(3, 1) and Euclidean so(4). Poincaré iso(3; 1) perches on a higher singular ridge between the valleys of deSitter so(3; 2) and so(4; 1).

The special relativity Lie algebra iso(3, 1) makes the observer a rigid body with 10 degrees of freedom, like a speck of diamond dust. The general relativity group Diff, however, makes the observer an infinite squid, unaffected by gravity, crossing all horizons freely, continuously deformable without limit. This surely overcompensates. It still works at the level of astronomy, and it greatly influenced the Weyl, Yang-Mills, Schwinger, and Standard Model theories of gauge. Yet general relativity is much more singular than special relativity, and so disproves the theory of evolution as regularization.

The Margulis evolutionary process is expressed in biology by symbiogenesis [6] and in technology by modular architecture [12]. In modular evolution, two modules of some complexity unite with each other to form a more complex system that can use survival strategies of both.

To cope with complexity, physics has united singular modules rather than wait for regular ones. Stability and complexity are both vital for theories, and they pull in opposite directions at present. They must be harnessed to pull together. General covariance cannot be right in its present form. Its group Diff(4) too must be de-contracted.

The de-contraction of Diff should be compatible with the Yang de-contraction of the Heisenberg-Poincaré Lie algebra $\mathfrak{hp}(3,1) \leftarrow\!\circ\, Y$. Here Y is a high-dimensional orthogonal representation $R : \mathfrak{y} \to \mathrm{so}(N)$, $Y = R(\mathfrak{y})$, of the simple Yang Lie algebra $\mathfrak{y}$. The most obvious candidate for the de-contracted diff is $\mathrm{so}(N)$, assuming a commutative diagram of Lie algebra transformations

$$\begin{array}{ccc} Y & \circ\!\!\rightarrow & \mathfrak{hp}(3,1) \\ \downarrow & & \downarrow \\ \mathrm{so}(N) & \circ\!\!\rightarrow & \mathrm{diff}(4) \end{array} \tag{5}$$

Gauge theories like gravity theory combine identical gauge modules at every point of space-time. This is a quantification too. It brings together three Lie algebras: a kinematic one for orbital variables, a statistical one for the gauge vector quanta, and a gauge Lie algebra for conserved currents. All need regularization. In a quantum set theory, all operations are reduced to assembly and disassembly. Kinematics is all statistics. The three singular Lie algebras must then become aspects of one regular one.

4 Below Hilbert Space

Canonical quantization uses an infinite-dimensional Hilbert space $\mathcal{H}$ of probability vectors. This space is still too weak and already too large for quantum theory.

Too large, in that there are infinitely more orthogonal rays in $\mathcal{H}$ than there can be disjoint pure populations in any physical laboratory. This makes its unitary Lie algebra singular and leads to divergent sums.

Too weak, in that $\mathcal{H}$ lacks fundamental concepts of modular structure and interaction. True, it can represent general actions; but under close inspection all actions resolve into interactions. Hilbert space has to become smarter to express these.

I no longer believe that a quantum set theory will work for quantum physics as classical set theory works for classical physics [4]. The main problem is that set theory formulates "laws of thought", not of physics. A set is a collection "thought of as one". A proton's position and spin are usually united by bracing, as one unites sets, but presumably not by our thinking of them as one. Higher-order logic cannot be given operational meaning like the first-order logic.

One may regard Cantor and Peano as proposing basic vertices $x \in y$ and $y = \{x\}$, respectively, for the graph (category) of all mathematical objects. Quantum theory must replace these ideational vertices by operational ones adequate for the non-category of physical processes (Sect. 9). The result does not resemble set theory enough to warrant the name.

5 Quantification

Quantification is a logical process that turns a theory of an individual into a theory of a multiplicity of like individuals. Set theory iterates it. The logician William Hamilton introduced the term in 1850. It has two famous quantum correspondents: Fermi quantification is regular, and Bose quantification is singular but is regularized by Palev (Sect. 6).

Quantization and quantification traverse the same road in opposite directions. Quantification assembles individuals into an individual of higher rank. Quantization resolves an individual into individuals of lower rank. Quantization introduces a quantum constant when it begins from a classical limit in which that constant has approached 0. In quantification a quantum constant provided by the lower-rank individual vanishes in a singular limit. A "second quantization" is well-known to be a quantification, but it is also a "second quantification" since it follows a quantization, which implies a first quantification. In this project we express all quantizations, including Yang space-time quantization (Sect. 2), as inverse quantifications, to arrive at the modular architecture of the quantum universe. *All kinematics is statistics*.

The Standard Model uses bracing or uniting operation $\{a, b, \ldots\}$, at least tacitly, to assemble quanta from their various conceptual parts: orbital, spin, isospin, color, and so forth. The lowest-order predicate algebrahas been quantized. The higher-order set theory rests on the lower-order; it must be quantized or dropped out of physics. Here we quantize it.

Cantor intended to represent the workings of the infinite Mind of God, while physicists seek merely to represent the workings of finite quantum systems. For finite algebraic purposes, Peano's one-to-one uniting operation $y = \iota x = \{x\}$, sometimes written $\overline{x}$ here, will do as the key construct of a truncated set theory. ι turns what it touches into a monad, a unit set. Polyads are built from monads by disjoint union $a \vee b$. For example $\{a, b\} := \iota a \vee \iota b$ is a dyad; not to be confused with a doublet. An n-ad is a product of n factors; an m-tuplet is a sum of m terms.

Uniting (bracing) occurs in many important constructs of the Standard Model and gravity theory. It is used to suspend associativity of the tensor product. For example it associates spin variables with their proper orbital variables. Again, a hadron is a triad of quarks, and so the triads must be united to associate their quarks properly when a pair of hadrons forms a deuteron. This hierarchy of unitings is commonly tacit.

Let

$$\mathbb{S} = 2^{\mathbb{S}} = \exp_2 \mathbb{S} \qquad (6)$$

designate the group of *perfinite* sets (sets that are ancestrally finite, hereditarily finite, finite all the way down). $\mathbb{S}$ is an infinite $\mathbb{N}$-graded group generated recursively from the empty set 1 by

- the monadic uniting operation ιx; and
- the dyadic product operation $x \sqcup y = x$ XOR y, for the symmetric union or XOR.

All $x \in \mathbb{S}$ obey the Clifford-like rule

$$x \sqcup x = 1. \tag{7}$$

$\mathbb{S}$ is $\mathbb{N}$-graded by cardinality (“adicity”). $\mathbb{S}$ is also $\mathbb{N}$-graded by rank, the number of nested ι operations. $\mathbb{S}$ also has a product $x \vee y = x$ POR y, the Peircian or partial OR, determined by $\sqcup$. POR formalizes Boole’s original partial-addition operation $\dot{+}$ and obeys the Grassmann-like rule

$$x \vee x = 0. \tag{8}$$

This 0 is the OM (for Ω) of Jack Schwartz’s programming language SETL: a space-filler indicating the intentional omission of any meaningful symbol. It is the semantic 0.

A plausible probability space for quantum sets is a linearized $\mathbb{S}$,

$$\check{\mathbb{S}} = \check{2}^{\check{\mathbb{S}}} = \exp_{\check{2}} 1\check{S}, \tag{9}$$

the least linear space that is its own Clifford algebra. It is generated recursively from the linear space $\mathbb{R} \subset \check{\mathbb{S}}$ representing the empty set by three operations:

- a monadic uniting operation $\iota : \check{\mathbb{S}} \to \check{\mathbb{S}}$;
- a dyadic Clifford product $x \sqcup y$, sometimes written xy, for the symmetric union; and
- the dyadic addition operation $x + y$ for quantum superposition

$\check{\mathbb{S}}$ also has a Grassmann product $x \vee y$ determined by $\sqcup$. For all x in a certain basis called classical, $x \sqcup x = \pm 1$ and $x \vee x = 0$, as in the classical theory and with the classical meanings.

$\check{\mathbb{S}}$ is $\mathbb{N}$-graded by a cardinality operator Grade $\check{\mathbb{S}}$ is also graded by the operator Rank, the number of nested ι operations.

A physical theory needs only a finite-dimensional probability tensor space, but it is convenient to keep $\check{\mathbb{S}}$ infinite-dimensional so that it also contains the singular limits presently in common use.

6 Palev Statistics

Palev regularized quantum statistics [7] as Yang regularized quantum kinematics [15]: by de-contracting a singular Lie algebra into a nearby regular one.

In an even statistics of *Palev type* $\mathfrak{p}$ [7],

1. $\mathfrak{p}$ is a classical (simple) Lie algebra.

2. The probability vector space of the individual quantum is $\mathfrak{p}$.
3. The probability algebra of the aggregate is $\mathcal{P} = \text{poly}\,\mathfrak{p}$, an algebra of non-commutative polynomials over $\mathfrak{p}$ identified modulo the commutation relations of $\mathfrak{p}$, defining a representation of $\mathfrak{p}$.

Palev also considers mixed even and odd statistics, where $\mathfrak{p}$ is a Lie superalgebra. In the present instance $\mathfrak{p} = \text{so}(N, N)$ with $N \gg 1$. Bose statistics is merely a useful singular limit of an even Palev statistics, ultimately unphysical [4].

There is no physical boundary between statistics and kinematics (Sect. 5), only a historical one. It is natural to regularize both at once. Di-fermions obey a Bose statistics only approximately, a Palev statistics exactly. The Palev Lie algebra can even be the Yang Lie algebra.

A brief review: The three-dimensional Heisenberg Lie algebra $\mathfrak{h}(1)$, with the singular canonical commutation relation

$$\mathfrak{h}(1): \quad [q, p] = i\bar{h}, \tag{10}$$

underlies both quantum oscillator kinematics and Bose statistics. $\mathfrak{h}(1)$ lies on the ridge between spin(2, 1) and spin(3). Quantum relativity needs an indefinite metric, so choose the indefinite case in this toy example:

$$\text{spin}(2, 1): \quad [q, p] = r, \qquad [p, r] = q, \qquad [q, r] = p \tag{11}$$

To contract spin(2, 1) to $\mathfrak{h}(1)$, the variable r must freeze to a central imaginary as the dimension D of the representation goes to infinity: $r \approx Ni$, where $N \to \infty$ with D. Call such a process an "organized singular limit" and write, for example,

$$\text{spin}(2, 1) \;\circ\!\!\rightarrow\; \mathfrak{h}(1), \qquad \check{q} \;\circ\!\!\rightarrow\; q, \qquad \check{p} \;\circ\!\!\rightarrow\; p, \qquad \check{r} \;\circ\!\!\rightarrow\; \mathsf{N}i. \tag{12}$$

The circle in "$\circ\!\!\rightarrow$" represents the regular algebra, the tip of the arrow the singular one, and the connecting line the self-organization, if any, and the homotopy that connect the regular to the singular.

7 Neutral Metrics

The metric in the Clifford algebra must be specified. To represent in the one space $\check{\mathbb{S}}$ the duality between source and sink that each experimenter sees, the probability form should be neutral, like that of a quantum space in the sense of [9], and like the Pauli-Cartan metric β of spinor space. To fix the sign convention: source vectors have positive norm, sink vectors negative.

Every finite-dimensional Grassmann algebra, and therefore every subspace $\check{\mathbb{S}}[r] \subset \check{\mathbb{S}}$ of finite rank r, has a natural neutral norm, the Berezin $L^{(2)}$ norm

$$\|w\| := \int d\gamma^{\top} w \vee w = \frac{d}{d\gamma^{\top}} w \vee w = \beta w w \tag{13}$$

where $\gamma^{\top} \in \check{\mathbf{S}}[r]$ is a top Grassmann element, and $\beta w w$ is the polarization of $\|w\|$. The Berezin norm is identical to the Cartan norm for spinors.

There is a frame-dependent scalar factor in this top element, and therefore in the norm (13); but all physical quantities, which are of degree 0 in the norm, so this factor drops out.

The previous rank $\check{\mathbf{S}}[r-1]$ is an isotropic space of this norm, representing linear combinations of sinks and sources with equal probability.

The norm β defines the Clifford product $x \sqcup y$ on $\check{\mathbb{S}}[r]$ by the Clifford rule

$$x \sqcup x = \|x\| \tag{14}$$

for vectors (of grade 1) and dual vectors (of grade $\operatorname{Dim} \check{\mathbb{S}}[r] - 1$). Here this is the exclusion principle.

In this quantum set theory as in classical set theory, any set can be in any set only 0 or 1 times. Multiple occupancy is forbidden by fiat. Sets with even statistics have to be pairs of monads. This set theory does not acknowledge elementary bosons, which violate Leibniz's principle that indistinguishable objects are one. Indistinguishable laser photons can be 10^{10} and more. The *grade parity* operator $g(x) \doteq 0, 1$ of a set x is the parity of the grade (= cardinality) of x. It defines the "statistics" of x; called even (or bosonic) if $g = 0$, and odd (fermionic) if $g = 1$.

The quantum set theory of $\check{\mathbb{S}}$ immediately conflicts with the Standard Model on the conservation of exchange parity (statistics). For any $x, y \in \check{\mathbb{S}}$, ιx and ιy are monadic (of grade 1) and so is their uniting $\{\{x, y\}\} = \iota(\iota x \vee \iota y)$. But in nature so far, and in the Standard Model, a composite of two odd quanta is always even. In nature, composition conserves grade parity but not in $\check{\mathbb{S}}$. The question was not considered explicitly in $\mathbb{S}$, which opted to deal in sets alone, although other modes of aggregation exist in classical thought. Sets are all of odd parity in that they obey the exclusion principle $x \vee x = 0$. The functor Grass works well on the probability space of fermions. But its iteration Grass^2 violates the conservation of statistics (exchange parity $X = 0, 1$).

Another problem with $\check{\mathbb{S}}$ as a paradigm is that its ι is infinitely reducible. This has classical roots. The classical ι is a formal sum of its restrictions $\iota^{(m)}$ to sets of cardinality m. Correspondingly, the quantum ι reduces to a sum

$$\iota = \sum_m \iota^{(m)} \tag{15}$$

of its projections on m-adics. In the classical basis $\{1_s\} \subset \check{\mathbb{S}}$, the operator $\iota^{(m)}$ has the matrix elements

$$\iota^{\{s_1 \ldots s_m\}}{}_{s_1 \ldots s_m} \tag{16}$$

in which for any $s_1, \ldots, s_m$, $\{s_1 \ldots s_m\}$ is a single collective index of one higher rank than any of $s_1, \ldots, s_m$. The repeated unsummed indices break the linear group, but the tensor

$$\iota^{\{s_3 s_4\}}{}_{s_1 s_2} := \delta^{s_3 s_4}{}_{s_1 s_2} \tag{17}$$

is invariant under the linear group and reduces to $\iota^{(2)}$ in the classical basis. Such $\iota^{(m)}$ are the irreducible vertices of this quantum set theory. If they occur in nature, their occurrences supply their operational definitions. If they do not occur in nature, we have to remove them from the physical theory, by burying them in its infrastructure if necessary.

There is also a problem with relativity. Set theory takes the Eternal view and has a distinguished frame, while a quantum theory admits only the secular perspectives of many limited experimenters. $\mathbb{S}$ and $\check{\mathbb{S}}$ both have preferred frames.

Here $\check{\mathbb{S}}$ will be truncated to two ranks, the points and links required to build a network of interactions. The rest of the usual hierarchy of ranks is replaced by a hierarchy of clusterings of clusters in the network. The points have Fermi statistics, the links Palev.

8 Quantum Events

Einstein understood an event to be a smallest possible occurrence, and took the collision of two small hard balls as an approximation to an event. Collisions of much smaller things have been studied since then, and they have more internal variables than Einstein's idealized buckshot, such as spins and Standard Model charges. It is remarkable that the Standard Model still uses the same mathematical representation of space-time events that Einstein did. Is this unreasonable effectiveness or pernicious influence?

To infer new classical space-time dimensions from the internal quantum numbers is archaic today. Any classical description is a low-resolution many-quantum description. Quanta are not born out of continua; continua are assembled quanta. To be sure, the first quanta were explained by quantization. Similarly, Swift tells us, the first roast pig was discovered when a barn burnt down, and for some time, a barn was burnt down for every roast pig. The continua of string theory and of modern followers of Kaluza are our barns. Eventually it will be recognized as Dirac did, that one can have a quantum without quantizing some continuum.

The charges of the elementary particles have small discrete spectra, in stark contrast to the quasi-continuous spectra of position coordinates. This tells us that the charge degrees of freedom do *not* organize themselves into quasi-continua as spins do in the models of Feynman and Penrose. Conceivably this is the salient difference between the charges and spin.

According to present physical theory, we never perceive space-time but only quanta. Quantizing space-time is best understood as extending the familiar quantization of orbital angular momentum to the other orbital variables x^{μ}, p_{μ}, like Feynman, Snyder, Penrose, and Yang. It resolves a fine-structure of quanta that canonical quantum theories smear out. Since our measurements always concern quanta, we have no need for both quanta and space-time. Regard space-time as another classical reification, a mental extension of the solid laboratory floor beam to the distant stars.

Particle collisions take the place of Einstein's buckshot collisions. In the relativistic canonical quantum theory the 15 orbital operators of en event,

$$x^{\mu}, p_{\mu}, L_{\mu'\mu}, i \in \mathfrak{hp}(3,1). \tag{18}$$

span the singular Heisenberg-Poincare Lie algebra. In addition quanta have the internal variables of the Standard Model. To describe one quantum in a canonical theory it suffices to tensor-multiply certain of these spaces and unite the product with braces or ι. In the canonical theory, however, collisions break up into several IO processes for quanta. One space-time quantization resolves these in turn into chronons two ranks of aggregation lower, the fundamental actions of the theory.

Since fermions exist, we cannot begin the inductive construction of the quantum universe as computer with the empty set alone, which is even. Begin with a foundation of N primal odd chronons represented by basis spinors $\chi_a \in \mathcal{X}$, $a \in \mathsf{N} \gg 1$. A set-theoretic fermion field hierarchically unites an odd number of chronons first into events $\varepsilon = \{\chi_e \vee \cdots \vee \chi_e\}$, and then into fields $\phi = \varepsilon_a \vee \cdots \vee \varepsilon_b$. To form a gauge boson, unite chronons into di-chronons, into gaugeon events, into a gaugeon field.

Set theory envisages the construction of its universe from the empty set in an infinity of ranks, by a Mathematician outside the theory. A similar construction within superset theory $\check{\mathbb{S}}$ requires on the order of 10 ranks or less to accommodate physics. It does not represent the entire universe, since mundane experimenters, like the extramundane Mathematician, are there from start to finish, controlling the system but largely undescribed in the theory.

Nevertheless it is advantageous for $\check{\mathbb{S}}$ to be infinite-dimensional. so that the singular limit of classical space-time can be carried out within $\check{\mathbb{S}}$.

Absolute Space and Absolute Time have left the theater but Absolute Space-Time remains in the Standard Model and general relativity. How did this myth begin?

Etymologically, a "line" is a linen string, a "point" is a puncture or stick, and "geometry" is earth-measurement. Sticks connected by linen strings used to puncture the Nile flood-plain each spring. This is supposed to be the origin of geometry. These sticks, however, have position, momentum, angular momentum, time, and energy. Presumably these were first neglected and then lost on the way to Greece, giving rise ultimately to the constructs of space-time. In quantum field theory space-time is merely an index on some variables, part of the infrastructure. Space-time coordinates are actually carried by physical quanta, not by mythical space-time points.

The fundamental events in nature are then IO operations for quanta. In the Standard Model every fermion carries many variables: one hypercharge y, one generation index g, three isospin τ^k, four Dirac γ^μ, eight color charges χ^c, four space-time x^μ, four energy-momentum p_μ, and six angular momenta $L_{\mu'\mu}$. Like atomic number and atomic weight, these variables tell us something about the structure and composition of the fermion. We are to fit them all into a semisimple operator algebra $\mathcal{E}$ of event variables.

To recover the canonical quantum theory from a Yang spin$(3,3)$ theory, one freezes one rotational degree of freedom $L_{65} \multimap Ni$, as a step in the organized singular limit in which $\bar{h}, \mathsf{X} \to 0$ and $N \to \infty$. Like the Higgs and gravitational fields, i is the non-zero vacuum value of a non-commutative field operator $\check{\imath} = Ł_{65}/N$. Presumably all vacuum values result from freezing and self-organization. The possibility that the Yang $\check{\imath}$ is the Higgs field has not been excluded.

In the Standard Model, odd probability vectors form a Clifford algebra, even ones a Heisenberg Lie algebra, and orbital coordinates x, p another Heisenberg algebra of lower rank. This singular structure is presumably a jury-rig. A nearby finite-dimensional algebra arises from the construction out of fermion pairs; not a Bose Lie algebra but a Palev one based on $\mathrm{so}(N, N)$.

Since the noncompact Lorentz group will not fit into a compact unitary group, a finite-dimensional relativistic quantum theory has to renounce definiteness of the probability norm as well as the space-time norm. Relativistic spinor theory provides one resolution [2]. The space R of real Majorana Dirac spinors has no invariant definite metric form. Instead it has one Pauli form β that is invariant but not definite, and a plethora of Hilbert forms h that are definite but not invariant, associated with different time axes; and nevertheless it works.

Take spinor spaces as elementary building blocks, so that their aggregates inherit this multi-metric structure. Correspondence with present physics requires that a unique invariant global hermitian form $\mathsf{H} \leftarrow\!\circ\, \mathsf{h}$ exist in the singular limit of classical space-time.

9 Quantum Gauges

Experience severely breaks the kinematic symmetry between position and momentum in quantum mechanics. So does the locality principle, which requires field variables coupled in the Lagrangian or action to share a space-time point, not a point in the Fourier transform momentum-energy space. So does gauge field theory, including general relativity. We must resolve this discord between the diffeomorphism and canonical groups into a harmony; they work too well to be merely discarded.

Gauge theories today rest on the following relation, with readings depending on context; "energy" stands for "energy-momentum" here:

$$\begin{gathered} D_\mu = \partial_\mu - \Delta_\mu, \\ \text{covariant derivative} = \text{Lie derivative} - \text{connection}, \\ \text{kinetic energy} = \text{total energy} - \text{potential energy}. \end{gathered} \tag{19}$$

In a Yang-Feynman quantum event space, the invariant concept is the Yang-Feynman quantized coordinate $\check{p}_{y'y}$, This is a cumulation

$$\check{p}_{y'y} = \overline{\psi}\gamma_{y'y}\psi \tag{20}$$

of many spin-matrix terms $\gamma_{y'y}$, generating a regular Lie algebra $\mathfrak{y}$. In turn, $\check{p}_{y'y}$ has a singular semi-classical contraction $p_{y'y} \in \mathfrak{hp}(3,1) \leftarrow\!\circ\, \mathfrak{y}$, the usual 15 canonical quantum event coordinates $p_{y'y} = (x^\mu, p_\mu, L_{\mu'\mu}, i)$. Then underlying the usual gauge field kinematics (19) is the single-event relation

$$\begin{gathered} \check{p}_{y'y} = p_{y'y} - P_{y'y}, \\ \text{Yang coordinate} = \text{canonical coordinate} - \text{quantum correction}. \end{gathered} \tag{21}$$

Both terms on the right-hand side are singular; only the left-hand side is physically meaningful. Accumulate this relation over all events in a field by a higher-rank cumulation $\check{\phi}\ldots\phi$ and we arrive at the regular correspondent of (19).

In older terms: classical gravitational curvature and the electromagnetic field are higher order corrections that remain when we contract quantum event space to a flat, field-free, classical space-time. They are classical effective descriptions of quantum non-commutativity at the chronon level.

C.S. Peirce noted that while only line graphs can be built up from a 2-vertex alone, the most general graph can be simulated with a triadic vertex alone. The natural physical candidate for a universal vertex is indeed triadic, that of gauge physics, with three limbs: a chiral spinor ψ, its dual $\overline{\psi}$, and a gauge vector boson (actually, palevon) ϕ:

$$\overline{\psi}\gamma^m A_m \psi =: \overline{\psi}\phi\psi = \gamma^{y''y'y}\overline{\psi}_{y''}\phi_{y'}\psi_y. \tag{22}$$

Here spin(8) triality cries out for physical interpretation, so far in vain.

The proposed probability space consists of all tensors constructed inductively from a finite number of triadic vertices γ of (22), by tensor multiplication, linear combination, contraction (connecting two compatible lines), and identification modulo commutation relations of the Yang-Palev kind. Odd lines obey Fermi statistics. Even lines obey the unique Palev statistics induced by this Fermi statistics.

The outstretched arms of γ represent chiral spinors, and the leg represents a vector in the first grade of a Clifford algebra associated with the spinor space. The spinors have exchange parity 1 and the vector exchange parity 0. All the terminals of γ are polarized. Spinors plug only into dual spinors, and dyads only into dyads.

Call the quantum structure whose history probability tensors are so constructed a quantum *interaction network*. Its probability tensors derive from Feynman diagrams and Penrose spin networks more than Cantor sets.

This diagram algebra will be developed further.

But it begins to seem likely that the three simple physical Lie algebras introduced by Yang into kinematics, Palev into statistics, and Yang-Mills into differential geometry are actually different representations of one. Whether this shows the pernicious influence or the unreasonable effectiveness of mathematics remains to be seen.

Acknowledgements Jack Schwartz and Martin Davis taught me some of the logical works of Post and Gödel. Roger Penrose and Feynman showed me their spin atomizations of space or space-time. The Lindisfarne lectures of Lynn Margulis introduced me to symbiogenesis. Tchavdar Palev discussed his statistics with me. David Bohm convinced me that the mathematics in mathematical physics is best regarded as an extension of natural language. Frank (Tony) Smith shared his knowledge of Lie and Clifford algebras with me. James Baugh, David Edwards, Shlomit Ritz Finkelstein, Andry Galiautdinov, Dennis Marks, Zbigniew Oziewicz, Heinrich Saller, Sarang Shah, and Mohsen Shiri-Garakani provided many helpful discussions and corrections. I thank them all deeply.

This paper is dedicated to Jack Schwartz.

References

1. Bopp, F., Haag, R.: Über die Möglichkeit von Spinmodellen. Z. Naturforsch. **5a**, 644 (1950)
2. Dirac, P.A.M.: Spinors in Hilbert Space. Plenum, New York (1974)

3. Feynman, R.P.: Personal communication of 1961. Feynman wrote much this formula ca. (1941). It may have appeared in a footnote, but we were unable to find it
4. Finkelstein, D.R.: Palev statistics. In: Dobrev, V.K. (ed.) Lie Theory and its Applications in Physics: IX International Workshop. Springer, Berlin (2012)
5. Gerstenhaber, M.: On the deformation of rings and algebras. Ann. Math. **79**, 59–103 (1964)
6. Margulis, L., Sagan, D.: Acquiring Genomes: A Theory of the Origin of Species. Basic Books, New York (2002)
7. Palev, T.D.: Lie algebraical aspects of the quantum statistics. Unitary quantization (A-quantization). Joint Institute for Nuclear Research Preprint JINR E17-10550, Dubna (1977). hep-th/9705032
8. Penrose, R.: Angular momentum: an approach to combinatorial space-time. In: Bastin, T. (ed.) Quantum Theory and Beyond, Cambridge, pp. 151–180 (1971). Penrose kindly shared much of this seminal work with me ca. 1960
9. Saller, H.: Operational Quantum Theory I. Nonrelativistic Structures. Springer, New York (2006)
10. Schwartz, J.: The pernicious influence of mathematics on science. In: Logic, Methodology and Philosophy of Science, Proceeding of the 1960 International Congress, vol. 44, pp. 356–360 (1966)
11. Segal, I.E.: A class of operator algebras which are determined by groups. Duke Math. J. **18**, 221–265 (1951). Especially §6A
12. Simon, H.: The architecture of complexity. Proc. Am. Philos. Soc. **106**, 6 (1962)
13. Stückelberg, E.C.G.: Quantum theory in real Hilbert space. Helv. Phys. Acta **33**, 727–752 (1960)
14. Wigner, E.P.: The unreasonable effectiveness of mathematics in the natural sciences. Commun. Pure Appl. Math. **13**, 1–14 (1960)
15. Yang, C.N.: On quantized space-time. Phys. Rev. **72**, 874 (1947)

Jack Schwartz Meets Karl Marx

Martin Davis

Abstract Participating in a group study of Karl Marx's classic Capital led Jack Schwartz in unexpected directions. He was surprised to find, not only that Marx's celebrated labor theory of value is in blatant contradiction to empirical reality, but also that Marx was quite aware of this. To get around this roadblock, Marx had returned to the Hegelian dialectic that he and Friedrich Engels had previously abandoned and even ridiculed.

More than one-third of the way through the first volume of Marx's acknowledged masterpiece *Capital*, the author had to confront the fact that, reasoning from his basic assumptions, he had been led to the to the conclusion that lines of industry that employ much human labor and little machinery would be more profitable than those that are more mechanized. Marx notes that this is in "apparent contradiction" with "all experience based on appearance". What Marx says next is bound to arouse the curiosity of any mathematician who reads it: "For the solution of this apparent contradiction, many intermediate terms are as yet wanted, as from the standpoint of elementary algebra many intermediate terms are wanted to understand that $\frac{0}{0}$ may represent an actual quantity."[1]

During the late 1950s, Jack Schwartz and I were part of a radical group that, over a period of several years, and in addition to other activities, had embarked on a study of *Capital*. One byproduct of Jack's study of Marx's economic system was his own work in mathematical economics.[2] Another was a novel understanding of the role of Hegel's philosophy in Marx's system.

[1] [9], p. 335.

[2] [13, 14].

I am grateful to Judith Dunford who read a draft of this article and made a number of very helpful suggestions.

M. Davis (✉)
Courant Institute of Mathem. Science, New York University, Mercer Street 251, New York, 10012 NY, USA
e-mail: martin@eipye.com

M. Davis, E. Schonberg (eds.), *From Linear Operators to Computational Biology*,
DOI 10.1007/978-1-4471-4282-9_3, © Springer-Verlag London 2013

1 Prologue: *Contemporary Issues*

I was an advanced undergraduate mathematics student at City College in New York when Jack Schwartz arrived there from Stuyvesant High School. He was in the process of changing his main interest from chemistry to mathematics, and we became friends right away. I was astonished at the way he absorbed difficult treatises one after another. He did his graduate work at Yale, while I did mine at Princeton, but we remained in close touch. In the summer of 1949 Jack and I, along with my Princeton roommate Mel Hausner, were living together in Jack's parents' apartment, while his parents and sister were escaping the summer heat in a bungalow at Rockaway Beach. During that summer I became engaged to Beatrice Appelstein, and I believe it was then that Jack met her. It was also during that summer that Beatrice introduced me to Murray Bookchin.[3]

Although Murray and I had both grown up in the Bronx, our lives had been utterly different. Seven years older than me, Murray had been an adolescent during the Great Depression while I had been a small child. Self-educated, first a Communist and then a Trotskyist, by the time I met him he had become a follower of the program of the German radical Joseph Weber. If I were forced to summarize Weber's program in a succinct formula, the phrase "Marxism with full democracy but without the proletariat" might come close. In a programmatic article [15] Weber wrote:

> To have elaborated the simple fact that the dream of humanity was not realizable without definite material prerequisites (while at the same time proving that these prerequisites were maturing) remains an historical fact of sweeping significance—the great merit of ... Marx and Engels.... The conception that the organized workers would overcome the capitalist system ... is at least historically obsolete.... the task falls directly on the overwhelming majority of mankind ... A new consciousness of the practicality of the old 'Utopias of reason' is necessary ...

Weber asserted that the environment would continue to be seriously damaged as long as "the profit motive determines [the] economy":

> ... the contamination, dangerous to life, of rivers and coasts through their excessive discharge of industrial refuse (including many chemicals), urban dirt and human excrement ...

Weber proposed a "party" that in effect was to be the new society in embryo. As such it was to be free of institutional apparatus: it was to have no officials and to own no property.[4]

I don't remember how much of this came across during my first meeting with Murray, but I do remember being immediately struck by his knowledge and intelligence. I also remember finding some of what I heard very hard to accept. In any case I did not see Murray again until the following summer. By then I was married

[3] See the obituary http://www.nytimes.com/2006/08/07/us/07bookchin.html.

[4] Of course this brief summary does not give an adequate account of Weber's ideas. See also: http://www.bopsecrets.org/images/weber.pdf and http://en.wikipedia.org/wiki/The_Movement_For_a_Democracy_of_Content.

to Beatrice, had completed my doctorate, and had accepted a faculty position at the University of Illinois in Champaign-Urbana to begin in the fall. The marriage did not last very long. When I moved to Urbana, Beatrice remained in New York. Later she and Murray married.

Back in New York in December 1950 for the Christmas vacation, I was very surprised to learn that Jack had been having extensive conversations with Murray and had become part of the movement based on Weber's ideas. I spent two academic years in Champaign-Urbana, and then was a Visiting Member at the Institute for Advanced Study in Princeton until the fall of 1954. Being near New York, I began attending some of the Saturday evening CI meetings, "CI" being the initials of the name of our quarterly magazine *Contemporary Issues*. Typically present at these meetings in a New York apartment were Joseph Weber (we called him "Joe"), Jack and his first wife Sandra, Murray and Beatrice, my wife Virginia (whom I had met and married in Urbana), and perhaps half a dozen others. There was a sister German publication called *Dinge der Zeit*. In New York there was also a Friday evening CI group with little overlap between the two New York groups. Part of each meeting would be taken up with reading correspondence from affiliates. There was Philip MacDougal in Carmel Heights, California who had a monthly commentary program on the Pacifica radio station KPFA. (When Phil retired from this, Virginia took over the program and continued for about two years.) There were also affiliates in London and there was a small group in South Africa vigorously fighting apartheid.

In addition to the magazine, we published occasional leaflets that we distributed to the public. Virginia and I put Murray's leaflet against nuclear testing into all the mailboxes at the Institute for Advanced Study. Jack wrote a powerful leaflet *America on the Road of Hitler and Stalin* on the erosion of civil liberty that was being justified by the Cold War. We published an article on the plight of the Palestinian refugees from the 1948 war long before the issue had any prominence. We were very conscious of the weakness of the Russian economy and believed that in spite of the bellicosity of the Cold War and episodes like the wars in Korea and Vietnam, the essential truth was that neither side really wanted to upset the stability of the postwar settlement. We felt that the Stalinist Empire was hollow in the middle and hoped to see it collapse from within. Therefore when the Hungarian Revolution broke out in 1956, and students in Moscow met in solidarity with the Hungarians, we agitated for arms to be made available to the Hungarian fighters. We distributed leaflets, marched with Hungarian immigrants, and organized public meetings.

CI meetings in New York were contentious affairs. Speakers were often interrupted followed by their cries of "Let me finish!". Nevertheless consensus was generally achieved, and our activities were carried out with verve and enthusiasm. What tore us apart was not disagreement about contemporary affairs, but a group study of Marx's classic *Capital*. The discussions and written documents to which this study gave rise over a period of three years ranged far and wide. Dissension arose over the first few pages in which Marx asserted what was later called his *Law of Value* to the effect that commodities exchange in the marketplace in proportion to the quantity of human labor required for their production. Jack and others found Marx's argument in favor of this proposition unpersuasive. Well, we were told, all this would be resolved in Volume III of *Capital*. Anyhow, we were assured, none of it could really be understood without a thorough grounding in Hegelian Philosophy. We were supposed to make sense of the impenetrable gobbledygook of Hegel's *Science of Logic*. Hegel's first propositions were the pair: *Being is nothing* and *Nothing is being*. Living in Eastern Connecticut at the time, I attended few of these meetings. But I do remember a meeting, apparently including people from the Friday and Saturday groups, in which Chester Manes (of the Friday group) recited verbatim Hegel's alleged demonstration of these assertions:

> **Being**, pure being, without any further determination. In its indeterminate immediacy it is equal only to itself. It is also not unequal relatively to an other; it has no diversity within itself nor any with a reference outwards. It would not be held fast in its purity if it contained any determination or content which could be distinguished in it or by which it could be distinguished from an other. It is pure indeterminateness and emptiness. There is nothing to be intuited in it, if one can speak here of intuiting; or, it is only this pure intuiting itself. Just as little is anything to be thought in it, or it is equally only this empty thinking. Being, the indeterminate immediate, is in fact nothing, and neither more nor less than nothing.
> **Nothing**, pure nothing: it is simply equality with itself, complete emptiness, absence of all determination and content—undifferentiatedness in itself. In so far as intuiting or thinking can be mentioned here, it counts as a distinction whether something or nothing is intuited or thought. To intuit or think nothing has, therefore, a meaning; both are distinguished and thus nothing is (exists) in our intuiting or thinking; or rather it is empty intuition and thought itself, and the same empty intuition or thought as pure being. Nothing is, therefore, the same determination, or rather absence of determination, and thus altogether the same as, pure being.[5]

Joe had been insisting that it was pointless to continue studying Marx without first obtaining a proper philosophical background. But people grew tired of this and asked him to desist. Despite being near the end of his life he published in CI a rambling fifty page paper dealing with philosophical matters.[6] In his article, Joe proposed that conceptual problems in physics and mathematics required "dialectical thinking" for their resolution. Mathematicians were urged to embrace contradictions rather than seek to avoid them. It was truly dispiriting to see someone whose breathtaking vision had once enthralled us produce such junk.

Marx understood quite well that there is something, that at least in "appearance", is problematical about his Law of Value, as is exemplified by the passage quoted at the very beginning of this article. Marx had worked out a very graphic way to demonstrate the exploitation of workers in a quantitative manner using his concept of *surplus value*. He pointed out that in the course of a working day, a laborer will reach a point where the value his efforts have produced so far are sufficient for his needs. Thus all the value produced beyond that point, the surplus value, accrues to the capitalist and is the source of his profit. Marx made use of this analysis to portray

---

[5] I found this text on the Internet. For the original German, see [5], pp. 43–44.

[6] Joe's article [16] was in #31 of CI dated October-November 1957, along with a brief article by my wife Virginia based on one of her KPFA broadcasts and an article by Jack on the dangers of nuclear experimentation.

capitalist society as in effect stealing from the worker. In an extensive address he gave to the First International in 1865 he did this very cleverly:[7]

> ...the peasant serf ... worked, for example, three days on his own field ... and the three subsequent days he performed compulsory and gratuitous labour on the estate of his lord. Here, then, the paid and unpaid parts of labour were visibly separated, separated in time and space; and our Liberals overflowed with moral indignation at the preposterous notion of making a man work for nothing.
> In point of fact, however, whether a man works three days of the week on his own field and three days for nothing on the estate of his lord, or whether he works in the factory or the workshop six hours daily for himself and six for his employer, comes to the same, although in the latter case the paid and unpaid portions of labour are inseparably mixed up with each other, and the nature of the whole transaction is completely masked by the *intervention of a contract* and the *pay* received at the end of the week.

This makes a fine polemic, but there is a conceptual problem. Tacitly, the Law of Value is being assumed, and it is a logical consequence of this "law" that Marx notes, is in "apparent contradiction" with "all experience based on appearance". Putting off for a subsequent volume the "many intermediate terms" needed "for the solution of this apparent contradiction", Marx sails blithely ahead and, basing his analysis on the Law of Value, he proves that the proletariat will eventually be placed in a position where they will be compelled to overthrow capitalism:

> Centralization of the means of production and socialization of labour at last reach a point where they become incompatible with their capitalist integument. This integument is burst asunder. The expropriators are expropriated.[8]

The "intermediate terms" needed were to be supplied in volume III of Capital. However Marx died before he had completed either volume II or volume III, and it was left to Engels to reconstruct them from Marx's rather chaotic notes.[9] Jack made a detailed study of the arguments in Volume III that were supposed to justify the Law of Value, and convinced himself that they were just wrong. He wrote this up in a detailed article in which he analyzed Marx's assertions in the context of a simple model economy in which the various quantities Marx discussed could be calculated explicitly using nothing more than high school algebra. The discussion in CI of Jack's work was bitterly contentious. As Jack said in a letter written to a friend at the time, "It became perfectly plain that I had touched upon a sensitive fiber: that the majority of the group felt that a number of fundamental views had been challenged, and were reacting with an angry blindness and hostility".[10] Jack recalled one complaint: "Marx gave us a palace, you leave us only with a shack."

In this atmosphere, Jack gave up trying to resolve the matter by further explanations and discussions during meetings. He submitted a short article for publication in our magazine. But the rupture had gone too far. Publication was refused despite the words in #5 of CI immediately preceding Joe's programmatic "Great Utopia":

---

[7] [8], p. 43. The same comparison appears, but less pointedly, in *Capital*, [9], p. 591.

[8] [9], p. 837.

[9] See [6], pp. 298–301.

[10] I'm grateful to Diana Schwartz who provided me with a copy of this letter.

> Only through thorough-going discussion, unrestrictedly free but informed and objective exchange of opinions of individuals of the most diverse origins, convictions, and experiences, can a comprehensive solution of the crucial issues of our time be arrived at.

Jack responded by resigning from CI. I had been working on a comprehensive reply to Joe's philosophical article in which I hoped to clarify some of the methodological issues for people with little education in those areas. Although my article had reached the point of a first rough draft, it seemed pointless to continue, and I also resigned along with two other members of the group. Our own group *Cornucopia*, in which we assumed a radical stance based on Keynesian economics, did not last long. CI continued for several years before it died as well, although Murray Bookchin continued pursuing a radical agenda including pioneering work on ecological issues.

2 Hegel and Marx[11]

Marxists and anti-Marxists disagree about many things, but all agree about Marx's relationship with Hegel's philosophy: he was a Young Hegelian in his early writings, and then, when he became a revolutionary communist, he adopted Hegel's dialectic to his own materialist philosophy. As we shall see the truth is more complicated.

By 1844, Marx and Engels had not only left the Young Hegelians behind, they were thoroughly disillusioned with Hegel's speculative philosophy as can be seen in this hilarious send-up of Hegelianism:

> If from real apples, pears, strawberries and almonds I form the general idea "Fruit", if I go further and imagine that my abstract idea "Fruit", derived from real fruit, is an entity existing outside me, is indeed the true essence of the pear, the apple, etc., then in the language of speculative philosophy—I am declaring that "Fruit" is the "Substance" of the pear, the apple, the almond, etc. I am saying, therefore, that to be a pear is not essential to the pear, that to be an apple is not essential to the apple; that what is essential to these things is not their real existence, perceptible to the senses, but the essence that I have abstracted from them and then foisted on them, the essence of my idea—"Fruit". I therefore declare apples, pears, almonds, etc., to be mere forms of existence, modi, of "Fruit". My finite understanding supported by my senses does of course distinguish an apple from a pear and a pear from an almond, but my speculative reason declares these sensuous differences inessential and irrelevant. It sees in the apple the same as in the pear, and in the pear the same as in the almond, namely "Fruit". Particular real fruits are no more than semblances whose true essence is "the substance"—"Fruit".
> ... Having reduced the different real fruits to the one "fruit" of abstraction—"the Fruit", speculation must, in order to attain some semblance of real content, try somehow to find its way back from "the Fruit", from the Substance to the diverse, ordinary real fruits, the pear, the apple, the almond, etc. It is as hard to produce real fruits from the abstract idea "the Fruit" as it is easy to produce this abstract idea from real fruits. Indeed, it is impossible to arrive at the opposite of an abstraction without relinquishing the abstraction.

---

[11] In bringing back to mind things I hadn't thought about for many years, I was greatly helped by the unpublished essay [3] by my son, Harold Davis. It was written as a term paper for an undergraduate course in January 1973. Harold had access to unpublished documents by Jack and by me.

> The speculative philosopher . . . argues somewhat as follows:
> If apples, pears, almonds and strawberries are really nothing but “the Substance”, “the Fruit”, the question arises: Why does “the Fruit” manifest itself to me sometimes as an apple, sometimes as a pear, sometimes as an almond? Why this semblance of diversity which so obviously contradicts my speculative conception of Unity, “the Substance”, “the Fruit”?
> This, answers the speculative philosopher, is because “the Fruit” is not dead, undifferentiated, motionless, but a living, self-differentiating, moving essence. The diversity of the ordinary fruits is significant not only for my sensuous understanding, but also for “the Fruit” itself and for speculative reason. The different ordinary fruits are different manifestations of the life of the “one Fruit”; they are crystallisations of “the Fruit” itself. Thus in the apple “the Fruit” gives itself an apple-like existence, in the pear a pear-like existence. We must therefore no longer say, as one might from the standpoint of the Substance: a pear is “the Fruit”, an apple is “the Fruit”, an almond is “the Fruit”, but rather “the Fruit” presents itself as a pear, “the Fruit” presents itself as an apple, “the Fruit” presents itself as an almond; and the differences which distinguish apples, pears and almonds from one another are the self-differentiations of “the Fruit” and make the particular fruits different members of the life-process of “the Fruit”. Thus “the Fruit” is no longer an empty undifferentiated unity; it is oneness as allness, as “totality” of fruits, which constitute an “organically linked series of members”. In every member of that series “the Fruit” gives itself a more developed, more explicit existence, until finally, as the “summary” of all fruits, it is at the same time the living unity which contains all those fruits dissolved in itself just as it produces them from within itself, just as, for instance, all the limbs of the body are constantly dissolved in and constantly produced out of the blood.
> . . . The ordinary man does not think he is saying anything extraordinary when he states that there are apples and pears. But when the philosopher expresses their existence in the speculative way he says something extraordinary. He performs a miracle by producing the real natural objects, the apple, the pear, etc., out of the unreal creation of the mind “the Fruit”, i.e., by creating those fruits out of his own abstract reason, which he considers as an Absolute Subject outside himself, represented here as “the Fruit”. And in regard to every object the existence of which he expresses, he accomplishes an act of creation.
> It goes without saying that the speculative philosopher accomplishes this continuous creation only by presenting universally known qualities of the apple, the pear, etc., which exist in reality, as determining features invented by him, by giving the names of the real things to what abstract reason alone can create, to abstract formulas of reason, finally, by declaring his own activity, by which he passes from the idea of an apple to the idea of a pear, to be the self-activity of the Absolute Subject, “the Fruit”.
> In the speculative way of speaking, this operation is called comprehending Substance as Subject, as an inner process, as an Absolute Person, and this comprehension constitutes the essential character of Hegel’s method.
> . . . On the one hand, Hegel with masterly sophistry is able to present as a process of the imagined creation of the mind itself, of the Absolute Subject, the process by which the philosopher through sensory perception and imagination passes from one subject to another. On the other hand, however, Hegel very often gives a real presentation, embracing the thing itself, within the speculative presentation. This real development within the speculative development misleads the reader into considering the speculative development as real and the real as speculative.[12]

Although this was written with the young Hegelians, and especially one F.Z. Zychlinski writing under the pen name Szeliga, in mind, the last paragraph particularly

[12] [12], pp. 68–72.

makes it clear that Marx and Engels are scoffing at Hegelian speculative philosophy in general.

Contrast this with what Marx had to say in 1873 in his preface to the second edition of *Capital* responding to criticism asserting that *Capital* used the Hegelian dialectic:

> My dialectic method is not only different from the Hegelian, but is its direct opposite. To Hegel, the life-process of the human brain, i.e., the process of thinking, which, under the name of "the Idea," he even transforms into an independent subject, is the demiurgos of the real world, and the real world is only the external, phenomenal form of "the Idea." With me, on the contrary, the ideal is nothing else than the material world reflected by the human mind, and translated into forms of thought.
> The mystifying side of Hegelian dialectic I criticised nearly thirty years ago, at a time when it was still the fashion. But just as I was working at the first volume of "Das Kapital," it was the good pleasure of the peevish, arrogant, mediocre Epigones [Büchner, Dühring and others] who now talk large in cultured Germany, to treat Hegel in same way as the brave Moses Mendelssohn in Lessing's time treated Spinoza, i.e., as a "dead dog." I therefore openly avowed myself the pupil of that mighty thinker, and even here and there, in the chapter on the theory of value, coquetted with the modes of expression peculiar to him. The mystification which dialectic suffers in Hegel's hands, by no means prevents him from being the first to present its general form of working in a comprehensive and conscious manner. With him it is standing on its head. It must be turned right side up again, if you would discover the rational kernel within the mystical shell.[13]

Coquetted? Here and there? Modes of expression? Does Marx's economic analysis really depend in some significant way on Hegelian methods or not? V.I. Lenin in Switzerland during the years before revolution broke out in Russia was studying philosophy and kept notes on his reading. Because of the Lenin hagiography in the Soviet Union, these scribbled notes were carefully edited and published as Lenin's "Philosophical Notebooks". In his notes on Hegel's Logic the following "aphorism" occurs:

> It is impossible completely to understand Marx's Capital, and especially its first chapter, without having thoroughly studied and understood the whole of Hegel's Logic. Consequently, half a century later none of the Marxists understood Marx!![14]

So, Lenin insisted that it's not a matter of a mere mode of expression, that the first chapter of *Capital*, "the chapter on the theory of value", can't be understood without a full understanding of Hegel's Logic. This was also very much what Joe Weber was trying to tell us in CI. They were telling us that Marx was not having a coquettish flirtation with Hegel, but rather a serious full-blooded relationship. However, if we look back to 1844, to *The Holy Family*, we have Marx and Engels telling us in effect that what the Hegelian method could contribute to a proper discussion of "common sense" matters was a "distorted" vacuity:

> *Speculative philosophy*, namely Hegel's philosophy, had to transpose all questions from the form of common sense to the form of speculative reason and convert the real question into a *speculative* one to be able to answer it. Having distorted my question on my lips and, like

---

[13] [9], p. 25.

[14] [7], p. 180.

> the catechism, put *its own* question into my mouth, it could of course, like the catechism, have its ready answer to all my questions.[15]

In order to begin to make sense of all of this, it is necessary to understand what Marx had been trying to accomplish. He was going to deepen the "political economy" of Adam Smith and Ricardo into a profound analysis of capitalism that would prove "scientifically" that capitalism as a socio-economic system was bound to collapse. Thus the coming of socialism would be seen as grounded in scientific law as opposed to the "utopian" socialism he and Engels viewed as pathetic. Engels recognized Marx's genius and was enthusiastic about this project. Engels work in his father's firm, which he hated, helped to finance this undertaking.[16]

While browsing in the voluminous correspondence of Marx and Engels, Jack Schwartz found a remarkable letter from Marx to Engels dated January 14, 1858:

> ... I am getting some nice developments. For instance, I have overthrown the whole doctrine of profit as it has existed up to now. In the *method* of treatment the fact that by mere accident I have again glanced through Hegel's *Logic* has been of great service to me—Freilgrath found some volumes of Hegel which originally belonged to Bakunin and sent them to me as a present.[17]

So after years of treating Hegel's philosophy with scorn, of dismissing it as transposing every "real question" into a "speculative" one, Marx found Hegel's Logic "of great service" in his economic theories. Here is Marx's exposition of his "Law of Value" in the first few pages of the first chapter of *Capital*:

> Exchange value, at first sight, presents itself as a quantitative relation, as the proportion in which values in use of one sort are exchanged for those of another sort, a relation constantly changing with time and place. Hence exchange value appears to be something accidental and purely relative, and consequently an intrinsic value, i.e., an exchange value that is inseparably connected with, inherent in commodities, seems a contradiction in terms. Let us consider the matter a little more closely.
> A given commodity, e.g., a quarter of wheat is exchanged for x blacking, y silk, or z gold, & c.—in short, for other commodities in the most different proportions. Instead of one exchange value, the wheat has, therefore, a great many. But since x blacking, y silk, or z gold & c., each represents the exchange value of one quarter of wheat, x blacking, y silk, z gold, & c., must, as exchange values, be replaceable by each other, or equal to each other. Therefore, first: the valid exchange values of a given commodity express something equal; secondly, exchange value, generally, is only the mode of expression, the phenomenal form, of something contained in it, yet distinguishable from it.
> Let us take two commodities, e.g., corn and iron. The proportions in which they are exchangeable, whatever those proportions may be, can always be represented by an equation in which a given quantity of corn is equated to some quantity of iron: e.g., 1 quarter corn $= x$ cwt. iron. What does this equation tell us? It tells us that in two different things—in 1 quarter of corn and x cwt. of iron, there exists in equal quantities something common to both. The two things must therefore be equal to a third, which in itself is neither the one nor the other. Each of them, so far as it is exchange value, must therefore be reducible to this third.

[15] [12], p. 106.

[16] Engels's unflagging support for Marx and his family continued for many years as the projected publication dates of the successive volumes of *Capital* receded into the future. See [6].

[17] [11], p. 102.

> A simple geometrical illustration will make this clear. In order to calculate and compare the areas of rectilinear figures, we decompose them into triangles. But the area of the triangle itself is expressed by something totally different from its visible figure, namely, by half the product of the base multiplied by the altitude. In the same way the exchange values of commodities must be capable of being expressed in terms of something common to them all, of which thing they represent a greater or less quantity.
> This common "something" cannot be either a geometrical, a chemical, or any other natural property of commodities. Such properties claim our attention only in so far as they affect the utility of those commodities, make them use values. But the exchange of commodities is evidently an act characterised by a total abstraction from use value. Then one use value is just as good as another, provided only it be present in sufficient quantity.
> ... As use values, commodities are, above all, of different qualities, but as exchange values they are merely different quantities, and consequently do not contain an atom of use value. If then we leave out of consideration the use value of commodities, they have only one common property left, that of being products of labour. But even the product of labour itself has undergone a change in our hands. If we make abstraction from its use value, we make abstraction at the same time from the material elements and shapes that make the product a use value; we see in it no longer a table, a house, yarn, or any other useful thing. Its existence as a material thing is put out of sight. Neither can it any longer be regarded as the product of the labour of the joiner, the mason, the spinner, or of any other definite kind of productive labour. Along with the useful qualities of the products themselves, we put out of sight both the useful character of the various kinds of labour embodied in them, and the concrete forms of that labour; there is nothing left but what is common to them all; all are reduced to one and the same sort of labour, human labour in the abstract.
> Let us now consider the residue of each of these products; it consists of the same unsubstantial reality in each, a mere congelation of homogeneous human labour, of labour power expended without regard to the mode of its expenditure. All that these things now tell us is, that human labour power has been expended in their production, that human labour is embodied in them. When looked at as crystals of this social substance, common to them all, they are—Values.[18]

On the basis of this discussion, and of nothing else, Marx goes on to develop his economic analysis presuming that commodities exchange in proportion to the quantity of labor "embodied in them". When the discussion of *Capital* in CI began, Jack and others objected at once complaining that in no way was this justified. One has no difficulty seeing that the relation of quantity x of commodity A exchanging with quantity y of commodity B is an equivalence relation. But then Marx concludes: "... in two different things—in 1 quarter of corn and x cwt. of iron, there exists in equal quantities something common to both. The two things must therefore be equal to a third, which in itself is neither the one nor the other. Each of them, so far as it is exchange value, must therefore be reducible to this third." Taking the example of the area of triangle being represented quantitatively by the usual formula, he states: "In the same way the exchange values of commodities must be capable of being expressed in terms of something common to them all, of which thing they represent a greater or less quantity." Must? Why so? By no means is every equivalence relation mediated by a quantitative measure in that manner. It is in the process of identifying this thing "common to them all" as a quantity of human labor that he adopts the "modes of expression" of Hegel's speculative philosophy. Like "the fruit" as an

[18][9], pp. 43–45.

abstraction, Marx offers us not the concrete actual labor of actual people, but rather "crystals of this social substance ... Values".

3 Marx's Enigmatic $\frac{0}{0}$

This essay begins with what Marx said when he explicitly confronted the contradiction between his Law of Value and the fact that in a market economy in which capital flows fairly readily towards profitable opportunities, the rate of profit tends to be constant across the entire economy. In more detail:

> Everyone knows that a cotton spinner ... does not pocket less profit or surplus value than a baker ... For the solution of this apparent contradiction, many intermediate terms are as yet wanted, as from the standpoint of elementary algebra many intermediate terms are wanted to understand that $\frac{0}{0}$ may represent an actual quantity.[19]

This apparently refers to the differential calculus in which the *derivative* is defined by

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}.$$

If we actually set h *equal* to 0 in the "difference quotient"

$$\frac{f(x+h) - f(x)}{h},$$

we do end up with the meaningless $\frac{0}{0}$. But what is involved in the definition of the derivative is the "limit" as $h \to 0$ of the difference quotient; it is not at all a question of setting $h = 0$. It was clear that the calculus developed by Newton and Leibniz yielded valuable results, but from the outset there were questions about the validity of the methods used. Consider the simple example of computing the derivative of x^2. The difference quotient

$$\frac{(x+h)^2 - x^2}{h} = \frac{2hx + h^2}{h} = 2x + h.$$

Therefore

$$\lim_{h \to 0} \frac{(x+h)^2 - x^2}{h} = \lim_{h \to 0} (2x + h) = 2x.$$

Now, when one divides by h to obtain $2x + h$, one tacitly assumes that $h \neq 0$ since division by 0 is undefined.[20] However, when one proceeds from $2x + h$ to $2x$, isn't the limit talk just a subterfuge? Isn't one just setting $h = 0$? This criticism was made in 1734 by the philosopher Bishop Berkeley in a witty polemic against the calculus:

---

[19] [9], p. 335.

[20] To emphasize the point, note that although $1 \times 0 = 2 \times 0$, if it were permissible to divide both sides of this equality by 0, we would reach the absurd conclusion $1 = 2$.

> For when it is said, let the Increments vanish, i.e. let the Increments be nothing, or let there be no Increments, the former Supposition that the Increments were something, or that there were Increments, is destroyed, and yet a Consequence of that Supposition, i.e. an Expression got by virtue thereof, is retained.[21]

It was not until well into the 19th century that these matters were cleared up. Hegel himself was not too shy to provide mathematicians with his explanation of the underlying concepts of the calculus, devoting almost fifty pages of his "Science of Logic" to this matter. How useful Hegel's insights were can be judged by his supposed clarification of how the "increment" h could both be and not be equal to 0:

> Although the mathematics of the infinite maintained that these quantitative determinations are vanishing magnitudes, i.e. magnitudes which are no longer any particular quantum and yet are not nothing but are still a determinateness relatively to an other, it seemed perfectly clear that such an intermediate state, as it was called, between being and nothing does not exist. What we are to think of this objection and the so-called intermediate state, has already been indicated above in [regard] to the category of becoming. The unity of being and nothing is, of course, not a state; a state would be a determination of being and nothing into which these moments might be supposed to have lapsed only by accident, as it were, into a diseased condition externally induced through erroneous thinking; on the contrary, this mean and unity, the vanishing or equally the becoming is alone their truth.[22]

So Hegel's doctrine of Being and Nothing makes everything clear!

Hegel's *Logic* appeared in 1812 well after d'Alembert's article in Diderot's Encyclopedia of 1754 had emphasized that the theory of limits was the proper foundation of the calculus. D'Alembert is not mentioned by Hegel although he does discuss at length a rather foolish proposal by his contemporary Lagrange. Cauchy's *Cours d'Analyse* of 1821 showed that the calculus could be developed as a rigorous deductive science with definitions and proofs. However, in his *Capital* of 1867, Marx had written as though the calculus was still sufficiently mysterious in a metaphysical way that he could use its alleged need for "many intermediate terms" such as those provided by Hegel to cover up the huge hole in his own economic analysis and help justify his own resort to Hegel.[23]

Marx died leaving for Engels the task of transforming Marx's notes and jottings into manuscripts suitable for publication as volumes II and III of *Capital*.[24] In his preface to Volume II, Engels promises that the problem of justifying equal profitability in the context of Marx's Law of Value would be solved in Volume III. In the meantime, he challenges critics:

> If they can show how an equal average rate of profit can and must come about, not only without a violation of the law of value, but rather on the very basis of it, we are willing to discuss the matter further with them.[25]

---

[21] [2], p. 57.

[22] [5], p. 165. Again the reference is to the German original; the translation is from the Internet.

[23] For a coherent account of the gradual development of a proper foundation for the calculus, see [1], pp. 260–273.

[24] See [6], pp. 298–301 for a description of the poor state of the papers from which Engels had to work in preparing Volume II and especially, Volume III.

[25] [10], p. 28.

In fact when Volume III did appear, no such solution was apparent.

In Engel's polemic *Anti-Dühring* [4] published in 1878, he continues to use the allegedly dialectical nature of the calculus as justification for Marx's reliance on Hegel:

> But, regardless of all protests made by common sense, the differential calculus assumes that under certain circumstances straight lines and curves are ... identical, and with this assumption reaches results which common sense, insisting on the absurdity of straight lines being identical with curves, can never attain.[26]

Of course the differential calculus assumes no such thing. For curves for which derivatives can be defined, the differential calculus associates with each point on the curve the *line* that is tangent to the curve at that point. So the differential calculus provides a curve with a whole collection of lines each of which shows the direction of the curve at a particular point on the curve. No contradiction and no Hegelian dialectics. Nevertheless, Engels continued to insist that with the use of "variable magnitudes" as in the calculus, "mathematics itself enters the field of dialectics.... The relation between the mathematics of variable and the mathematics of constant magnitudes is in general the same as the relation of dialectical to metaphysical thought."[27]

Berkeley's critique of the calculus, addressed to "an infidel mathematician" was intended to show that the reasoning used by mathematicians was no sounder than that used by theologians. Marx and Engels similarly invoke the alleged dependence of the calculus on the Hegelian dialectic to justify its use in Marx's economic system.

4 Conclusion

Jack Schwartz had come to the conclusion that Marx had left his Hegelian past behind when he became a revolutionary communist and that he returned to Hegel only when he found no other way to deal with the contradiction that threatened to ruin his economic system. If labor content did not really control profit in the mundane world of reality, one could still claim that it did so in some more profound conceptual domain. It was to provide such a framework that Marx returned to the Hegelian idealism of his youth. Although Marx had asserted that Hegel's philosophy "must be turned right side up ... if you would discover the rational kernel within the mystical shell", in fact it was precisely that "mystical shell" that he brought to bear on his economic system.

In the article written for CI, the one that was refused publication, Jack emphasized the burden that would-be Marxist thinkers faced in reconciling *Capital* with material reality. He analyzed the work of three of these Hilferding, Meek, and

[26] [4], p. 132.

[27] [4], p. 134.

Sweezy critically and in detail. In addition, it should be said that relying on an invalid theory based on an ideal world that differed too greatly from the world as it is has consequences. When the predicted collapse of capitalism from the lack of surplus value as industry became more and more machine driven failed to occur, the revisionism of Bernstein and Kautsky that so upset Lenin took hold.[28]

I conclude this article with a quotation from Jack's never published essay:

> Defenders of Marx have often claimed that *Capital* could not be comprehended in terms of ... customary logic, but required for its understanding some special "dialectical" mode of thought.... such a situation is typical for *every* false theory. Catholicism can also be elaborated logically in the logic of Saint Thomas, and on the basis of an initial act of faith.... All science, economics included, is ... the imitation of reality by notions. The imitation can be more or less exact in the sense of fidelity to detail; and it is an almost invariable rule in science that the more exact a picture becomes in this sense, the harder it is to work with, to see the forest for the trees ... Thus it is that at many points in science, one rough and ready generalization is worth more than a thousand exact theories which lose themselves in a mass of unwieldy theoretical detail. Thus it is that science so often develops ... from a broad and very qualified generalization, to a more exact and less qualified improved theory, of which the original approximate idea is seem to be an approximation valid in a common special case or in common, but theoretically extreme, conditions.
>
> Nevertheless, science is and remains the imitation of reality by notions. Thus the maxim "A is true in theory, but B is true in practice" is absurd, and proves only the falsity of the theory in which A is true. Theory, in order to simplify certain of the notions by which it seeks to imitate reality, is legitimately accustomed to ignoring certain aspects of reality, but this does not make these aspects disappear. Here science becomes an art. To say that the earth is a sphere is to express a relative truth; and is not to be refuted by the first person who points to a sharp cliff. But when it is pointed out that the earth is not a sphere but an oblate spheroid, and not an oblate spheroid but a spherical mass of plastic rock in which continental irregularities float, the relative truth must yield to the less relative. Nor must we think that the bulge at the earth's equator is to be taken as *its* perversity for not being perfectly spherical—it is only *our own* incompetence in the manipulation of the more complex notion of oblate spheroid, *our perverse* insistence on saying "sphere" rather than "oblate spheroid, *almost* a sphere" that is involved. To be blinded in theory by one's own constructions ... is among the most common of theoretical errors.

References

1. Bell, E.T.: The Development of Mathematics. McGraw-Hill, New York (1940)
2. Berkeley, G.: In: The Analyst or a Discourse Addressed to an Infidel Mathematician. The Works of George Berkeley Bishop of Cloyne, Nelson, London/Edinburgh, vol. 4, pp. 53–102 (1948)
3. Davis, H.: Hegelianism in Marx, unpublished essay
4. Engels, F.: Herr Eugen Dühring's Revolution in Science. Marxist Library, vol. 18. International Publishers, New York (1939)
5. Hegel, G.F.W.: Wissenschaft der Logik, Erster Band: Die Objektive Logik (1812/13). Gesamelte Werke, vol. 11. Felix Meiner Verlag, Hamburg (1978)

[28] As I write, we are experiencing economic conditions arising in considerable part from reliance on the monetary theories of Milton Friedman, theories that were roundly criticized in Jack's book [14].

6. Hunt, T.: Marx's General: The Revolutionary Life of Friedrich Engels. Henry Holt, New York (2009)
7. Lenin, V.I.: Philosophical Notebooks. Collected Works, vol. 38. Foreign Languages Publishing House, Moscow (1961)
8. Marx, K.: Value, Price and Profit. International Publishers, New York (1935)
9. Marx, K.: Capital: A Critique of Political Economy, vol. I. Modern Library, New York (1936)
10. Marx, K.: Capital: A Critique of Political Economy: The Process of Circulation of Capital, vol. II. Charles Kerr, Chicago (1915)
11. Marx, K., Engels, F.: Selected Correspondence. Marxist Library, vol. 29. International Publishers, New York (1935)
12. Marx, K., Engels, F.: The Holy Family, or Critique of Critical Criticism: Against Bruno Bauer and Company. Progress Publishers, Moscow (1975)
13. Schwartz, J.T.: Lectures on the Mathematical Method in Analytical Economics. Gordon and Breach, New York (1961)
14. Schwartz, J.T.: Theory of Money. Gordon and Breach, New York (1965)
15. Weber, J.: The great utopia. Contemp. Issues **2**(5), 3–22 (1950)
16. Weber, J. [Wilhelm Lunen, pseud.]: The problem of social consciousness in our time. Contemp. Issues **8**(31), 478–528 (1957)

SETL and the Evolution of Programming

Robert Dewar

Abstract The idea that programming should focus on the "what" rather than the "how" has generally been realized in language features for modularization and data type abstraction (classes and inheritance, for example). But equally important is having a notation to express functionality rather than algorithm and data structure detail, in effect to think like a mathematician rather than a programmer. That was Jack Schwartz's goal for the SETL project at NYU when it originated some 40 years ago. Based around the fundamental concepts of set and mapping, SETL has been used in practice to write executable specifications, effectively integrating requirements and design into compilable source code. As modern compiler technology, as well as modern hardware performance, make run-time efficiency issues much less of a concern, SETL's high-level worldview shows how to make programming easier and more productive. SETL has had a quiet but pervasive influence on language design, as seen for example in languages such as Python.

1 The Origins of the SETL Project

> *Computers are becoming very fast, with very large memories, and programmers will steadily become more and more expensive in relation to hardware costs. Given these circumstances, it makes sense to shift programming towards the use of very high level languages which will decrease programmer effort at the expense of some inefficiency, which becomes easy to accommodate given these trends.*

What makes this nearly four-decade old observation extraordinary was that the machine that inspired Jack's thoughts was the CDC 6600 that NYU had recently acquired. It is instructive to compare this machine with modern PCs. The cost was in the region of 5 million dollars, approximately 10,000 times the cost of a modern PC. The clock rate was 10 megahertz, and although this was one of the earliest pipelined

R. Dewar
Courant Institute, New York University, New York, USA

R. Dewar (✉)
AdaCore Technologies, New York, USA
e-mail: dewar@adacore.com

M. Davis, E. Schonberg (eds.), *From Linear Operators to Computational Biology*,
DOI 10.1007/978-1-4471-4282-9_4,

machines, it had nothing like the hardware parallelism of a modern processor, not to mention the multi-core trend, so it is probably reasonable to estimate that there is a factor of 2000 in processing performance. Finally the memory of the CDC 6600 was 128K 60-bit words, or the equivalent of about one megabyte, and our inexpensive PC would typically have 4000 times that much memory. So we are talking about a machine whose price/performance was at least 20 million times poorer than a cheap modern PC.

Yet Jack had the vision to see this trend far into the future, and although the notion of everyone buying multi-million dollar machines and then being prepared to waste a substantial factor in their processing power seemed at the time absurd, his view of the future was clearly remarkably accurate. The history of programming had already gone through a similar revolution with the replacement of efficient hand-coded machine language by the use of portable "high level" languages such as Algol-60, FORTRAN, and COBOL. The loss of efficiency (often a significant factor) was found acceptable in return for higher programming productivity and increased portability. By high level here, we refer to languages that allow writing in a form closer to the problem, and further away from low level implementation details. As a simple example, a writer in a language like FORTRAN no longer has to worry about which registers will hold which values.

Jack saw the SETL project as the next logical step in this progression. The overarching idea was simple: create a language in which the compiler and implementation take care of details so that the programmer does not have to waste time with low-level error-prone arcana. Conventional programming languages are characterized by design nits that are obviously undesirable but are needed for efficiency. For example, when we pass an integer to a procedure, we typically pass a copy, but when we pass an array, we typically pass a pointer, resulting in the entire set of issues that arise from undesirable aliasing. Why is this done? Simply because we have the view that copying an array is expensive. Arrays are in fact a source of other efficiency-imposed problems. In conventional programming languages we have to set definite limits for the bounds of an array in advance. Why? Simply because it allows a very efficient representation at run-time.

SETL [6] starts with some fundamental decisions. First, it is an entirely value-oriented language: composite objects are treated like integers, and copies are made freely. The implementation may strive to reduce copying (there are many SETL newsletters devoted to the interesting issues in such optimizations), but ultimately we don't care if we "waste" some time copying. There are no pointers of any kind, and no concerns about storage management, since we operate in an environment with secure type-accurate garbage collection. The fundamental data structures are sequences, sets, and maps. As mathematicians have known for at least a hundred years, sets and maps provide a universal mechanism for representing and manipulating all mathematical concepts. For example, the notion of pointer is all about representing mappings between arbitrary domains, and it is far more natural to think of these mappings as sets of ordered pairs, rather than as low level data structures.

For example, given the need to represent a binary tree structure, a conventional language such as C++ would use pointers to the structures representing the left and right subtrees. The use of such pointers comes with all the care necessary to avoid memory leaks, allocation errors, dangling references, and unintended aliasing, not to mention that setting such pointers often involves error-prone low level code. In SETL, we would represent such a structure using two mappings Left_Subtree and Right_Subtree which map from nodes to nodes. These mappings are simply values and just as we can add one to an integer, we can add an element to such a mapping. There are of course worrisome implications of having to copy the whole structure, e.g. to pass it to a procedure, or to return a modified value. But the whole point is to forget about such implications in the process of programming. The compiler will do its best to minimize such problems by playing various (transparent to the programmer) tricks, and in cases where we end up with extra copying and waste machine power, we don't care: we have plenty of machine power available to waste.

The built in notions of set and map abstraction allow convenient expression of algorithms at any level of detail, and in particular allow high level algorithms to be easily specified. Consider the following program to display the primes less than 1000. In addition to iteration over an (implicit) set it shows the use of the quantified expressions of first-order logic:

```
print {P in 2 .. 1000 | not exists D in 2 .. P-1 | P mod D = 0};
```

To most readers this looks more like a definition of what primes are. It is essentially just a high level specification of the problem statement. But in SETL this is an executable program. It can be executed and will print the primes in some arbitrary order (if you want them printed in order, you can replace the { } by [] to get an ordered sequence). Once I was teaching SETL in a programming class, and a mathematics professor was auditing the class because she thought that she should acquire some knowledge of computers and programming. Half way through, after we had written many sample programs, she came to me and said something like "This is interesting mathematics, but when do we get to programming?"

An important note here is that this program is indeed concise, but that's not its primary virtue. The important thing is that it is clear. A language like APL gives plenty of opportunity for amazingly concise representations of algorithms, and indeed clever compact C code can accomplish astounding feats in a small number of C symbols, but neither case is an example of clarity, and that is the virtue we seek here.

Now let's think about the efficiency of this program. It pretty clearly corresponds to two nested loops, and represents a rather inefficient implementation of finding primes. If we compare it to a conventional program with these same loops, it will execute in a comparable amount of time. But of course there are faster ways of finding primes, and of course we can explicitly program these in SETL easily enough. But let's ask two interesting questions.

2 Executable Specifications

Is the primes program in the previous section a specification or an implementation? If we wrote the specification in a non-executable specification language, it would look remarkably similar, so we can certainly regard this as a specification. But at the same time it is executable. In the context of the SETL project, the notion of "executable specifications" plays an important role. One significant advantage of an executable specification is that it can be tested. Testing does not ensure freedom from errors, but we sure feel more confident about a program that has been tested than one that has not, and this applies to specifications as well.

As an example of large scale use of SETL as an instrument for creating high level executable specifications, let's look at the early days of the Ada project at NYU [3]. Around 1980, I and some colleagues at NYU became interested in the Ada language, but it was clearly in a rather preliminary state with many detailed questions about the semantics still unanswered. We decided to build what was basically a denotational semantic definition of Ada, to understand these issues better, and we decided to build this as an executable specification in SETL. Now you may think that the fact that the specification was executable would mean that we were really doing an operational definition, but that's not the case. For example, an *if* statement was "executed" by replacing it in the executable representation of the program with either the *then* part or the *else* part depending on the condition. There was no notion of representing an *if* statement with a series of *goto*s. Indeed, as is typical of denotational specifications, *goto*s posed a challenging problem, and were represented as mappings from labels to program states.

Programming at this level required complex dynamic data structures, which in a traditional language would have overwhelmed the clarity of the definition with low level details, but in SETL these structures were represented with high level sets and mappings and were easily programmed at a level of clarity that compared well with existing attempts at producing denotational semantic definitions of programming languages. But of course the huge advantage of the SETL model was that the definition was executable, and indeed in 1983, this SETL "definition" was the first Ada translator to be validated by running the official test suite, and the use of this definition turned out to be helpful in defining the first version of the Ada standard.

As a contrast, a few years later, the EU decided to fund an effort (at a level of nearly two million dollars) to produce a formal definition of Ada, based on the work of Astesiano [1]. We participated in the discussions leading up to the grant award, and at our insistence, there was a requirement that the definition be executable, and everyone agreed this was a good idea. Unfortunately in subsequent negotiations with the European teams producing the definition, this requirement was dropped to save money. The result is that the definition was produced, but never had any significant impact. I still have sitting on my shelf two big telephone-directory-sized volumes full of formulae and really it is nothing more than impressive shelf decoration. Without the ability to show that the definition was consistent with the official test suite, it had no chance of being useful.

Of course this definition as an executable specification for Ada was drastically inefficient. Ed Schonberg once described it as an effective real time implementation of pencil and paper calculations. The generated code was perhaps six decimal orders of magnitude slower than code that would be produced by an efficient Ada compiler. Nevertheless it was fast enough to execute the test suite, even on the relatively slow machines of the day, and since student programs typically take close-to-zero time to execute and a million times close-to-zero is still close-to-zero, it was suitable for student use, and used in hundreds of universities teaching Ada to their students. It also served as a model for the implementation of a more efficient version written in C, and later of GNAT, which is one of the main commercial implementations of Ada.

3 Optimizing at the Algorithmic Level

The second question to ask about our simple primes example is what can be done to optimize the underlying algorithm. In general if we express a problem using a high level representation of what looks like on the surface a very inefficient algorithm, it is interesting to ask whether some automatic process can refine the program to use a more efficient algorithm. Let's take another example, the following SETL program outputs a partial order of the elements in a set S.

```
while S /= {} do
   Nopreds := {P in S | not exists Q in S | Q/= P and Q < P};
   Next := arb Nopreds
   Print (Next);
   S := S - {Next};
end;
```

Again this is pretty much the definition and specification of what a partial order is. You repeatedly choose an arbitrary element from the remaining elements for which there is no smaller element available. The expression in SETL uses the key operator **arb** to select an arbitrary element, avoiding any over-specification of the result. An attempt to program this in a language like C would not only be full of low level detail and data structures to represent the set, but would inevitably over-specify the result.

This is of course on its surface an inefficient algorithm, but let's ask the question of whether an automatic process could, starting with this clear high-level expression, derive an efficient algorithm. The answer at least for this example is a definite yes. The work of Robert Paige [4] shows how this and many other similar examples can be optimized to obtain an efficient algorithm. The basic approach, called "finite differencing", is to avoid redundant computations. Instead of repeatedly computing the set Nopreds, we keep it around and update it as elements are removed from S. This technique is widely applicable.

The ability to easily specify algorithms at a high level is very valuable, and the SETL notation facilitates this approach. Of course we can write things in a lower level language like C by implementing a set of library routines, but the messy syntax of using these routines gets badly in the way of clarity, and is not a natural way of using a language like C, which invites low level detail. For example, consider the following program for sorting a sequence S

```
print (arb (P in permutations (S) |
                    forall q in 1 .. #s - 1 | p (q) <= p (q + 1)));
```

This is again a high level specification of the notion of sorting, avoiding any over-specification of how the sorted sequence is to be found, and any implication of an efficient algorithm. Executed naively, this is a spectacularly inefficient algorithm, but again we can wonder if it can be optimized automatically. In the work "On Programming" [5] which is a remarkable collection of observations and musings on the future of programming, Jack wonders if there is a way of writing what you want instead of how to get there, something like

```
(forall q in 1 .. #s – 1 | p (q) < p (q + 1)) := True;
```

Here we simply say we want the sequence to be sorted, without giving any hint as to how that might be done. Jack postulates a set of assumptions under which the above statement can result in a reasonably efficient sorting algorithm automatically. We are certainly far from this level of programming, but it is a bolder speculation than what is found in existing declarative languages.

4 Improving Programming Productivity

Many studies show that programming productivity expressed in correct lines/day is pretty constant over a wide range of programming. This means that the fact that SETL programs are typically far smaller (by a factor of 5–10) than programs in conventional lower level languages of itself results in faster programming. The expense comes from giving away some performance, due to both the interpretive overhead and the natural tendency to use higher level less efficient algorithms that are easier to express. But that was a tradeoff that was at the heart of the decision to design SETL in the first place.

Another important boost to programmer productivity is to reduce debugging time. If you look at conventional program development in a language like C, the typical approach is to write a program that is initially full of bugs, and then painfully get rid of them by debugging. "Painfully" here refers to the costs associated with this extended debugging effort. One way to reduce this effort is to design the programming language to prevent bugs in the first place. SETL takes the viewpoint that we should never design bug-prone stuff into languages in the name of performance.

One well-known example is the buffer overrun error (exceeding the size of an array), which is a major source of errors in C programs (one Microsoft study of the Windows source code base shows thousands of instances of potential buffer overruns). A language like Ada makes some advances in clearly raising a run-time exception in such circumstances, rather than causing unpredictable chaos as in C, but you still have to decide how to handle the exception. As Ariane-5 [2] showed, an unhandled exception can be as deadly as undefined behavior. SETL simply arranges that arrays expand as needed. Yes, this involves some complex data structures. Yes, we could program these structures explicitly in lower level languages. But in SETL, this approach comes automatically.

As another example, pointers, aliasing, and storage management represent a huge source of errors in low level programming languages. SETL simply eliminates these concerns by using a pure value-oriented approach with no trace of pointers. There is no more effective way of preventing a particular class of bugs than by making it simply impossible to inject such a bug into the program in the first place.

5 The Status and Influence of SETL

SETL has continued to develop, and complete implementations of a follow-on version of the language SETL-2 are available on a wide variety of machines, including PCs. A community of researchers continues to explore the possibility of improving the programming process using this approach.

Has Jack's vision come to pass of wide spread use of very high level programming languages? A short answer is no. In the decades since SETL was first introduced, it has surprised Jack and others of us that such is the case, but the fact of the matter is that the vast majority of programming is still done in low level languages like C. Java shows some signs of advancing in the right direction, but ultimately is a disappointment, with many low level features like integer arithmetic that wraps silently on overflow, fixed-size arrays, and an error-prone thread model.

A more interesting and encouraging sign is the widespread adoption of Python and similar languages which indeed share SETL's vision of very high level languages with abstract data structures and operations (such as Dictionaries, which are general mapping structures). In the case of Python, we can trace a direct ancestry to SETL. The design of SETL strongly influenced the design of the ABC teaching language of Lambert Meertens, and in turn the design of ABC influenced the design of Python, so it is no accident to see some of the same design principles at work.

We are still far from reaching Jack's vision of most programming being done using very high level languages, even in this day of machines millions of times faster than the CDC 6600 which inspired Jack's original thoughts in this direction, but we do see definite progress in the right direction. The quote that started this article is even more true today than it was decades ago.

References

1. Astesiano, E., Reggio, G.: The SMoLCS approach to the formal semantics of programming languages—a tutorial introduction. In: Habermann, A.N., Montanari, U. (eds.) System Development and Ada: CRAI Workshop on Software Factories and Ada, pp. 81–116. Springer, London (1987)

2. Lions, J.L.: ARIANE 5, Flight 501 Failure, Report by the Inquiry Board (1996). http://www.ima.umn.edu/~arnold/disasters/ariane5rep.html
3. NYU Ada Project: Ada/Ed interpreter: Executable semantic model for Ada. Technical report, Courant Institute of Mathematical Sciences, New York University, July 1984
4. Paige, R., Koenig, S.: Finite differencing of computable expressions. ACM Trans. Programm. Lang. Syst. (TOPLAS) **4**(3), 402–454 (1982)
5. Schwartz, J.T.: On programming. An Interim Report on the SETL Project, Computer Science Department, Courant Institute of Mathematical Sciences, New York University (1973)
6. Schwartz, J.T., Dewar, R.B.K., Dubinsky, E., Schonberg, E.: Programming with Sets: An Introduction to SETL. Springer, New York (1986)

Decision Procedures for Elementary Sublanguages of Set Theory. XVII. Commonly Occurring Decidable Extensions of Multi-level Syllogistic

Domenico Cantone

Abstract The paper focuses on extending existing decision procedures for set theory and related theories commonly used in mathematics to handle such notions as monotonicity, ordering, inverse functions, etc. After presenting two decision procedures for the basic multilevel syllogistic fragment of set theory and studying the computational complexity of its decision problem, we illustrate a technique based on a syntactic translation of formulae with the special function and predicate symbols above into multilevel syllogistic that, in most cases, yields nondeterministic polynomial-time decision procedures. Such results can be quite useful for tool developers who aim at providing assistance to common mathematical reasoning. A semantically oriented approach is illustrated in the second part of the paper, where nondeterministic exponential-time decision procedures, of theoretical interest only, are briefly sketched for two extensions of multilevel syllogistic, with the general union operator and with the powerset operator.

1 Prologue

At the end of our first technical conversation in late August 1983, when I first met him and I was about to begin my PhD studies at NYU under his supervision, Jack Schwartz predicted that the (at that time novel) field of *Computable Set Theory* would have remained an active source of more and more challenging research problems "for at least five years, maybe ten, or even better for more than twenty years." Today I can say that he was completely right, while at that time I was just a little bit afraid that he meant it could have taken as many as twenty years for me to complete my Ph.D.!

Jack's interests in the field of proof correctness can be traced back to at least 1967, when he chaired a symposium on *Mathematical Aspects of Computer Science*, which was almost entirely devoted to various aspects of theorem proving (cf. [26]). Most certainly, they grew in parallel with his efforts in compiler and language design

D. Cantone (✉)
Dipartimento di Matematica e Informatica, Università di Catania, Viale Andrea Doria, 6, 95125 Catania, Italy
e-mail: cantone@dmi.unict.it

M. Davis, E. Schonberg (eds.), *From Linear Operators to Computational Biology*,
DOI 10.1007/978-1-4471-4282-9_5,

which culminated in the development of the SETL language, a pearl of elegance in the realm of programming languages (cf. [27, 30]).

The seminal ideas on program and proof verification systems were already contained in the paper *A survey of program proof technology* (cf. [28]) which in 1978 inaugurated the Technical Report series of the NYU Computer Science Department.

> *A proof checker is an interactive programmed system into which one can enter sequences of logical/mathematical formulae, each of which is a consequence, according to the laws of logic, of preceding formulae. Such a system will continue to accept new formulae as long as it can verify that each formula is indeed a correct consequence of what has gone before. Any step which is too complex for the verifier to follow (or which is incorrect) will be rejected, forcing the verifier's user to enter a number of intermediate formulae in order to get the verifier to accept the formula which really interests him. Thus the verifier ensures rigorously against logical error, possibly at the price of requiring its user to key in a burdensome mass of intermediate detail. The designer of such a proof verifier must aim to provide it with enough internal power for the mass of detail which is demanded to be reduced to reasonable levels.*
> [$\cdots$] *Full-scale proof verifiers can be expected to consist of three principal components:*
> (a) *An inner core of procedures which handle what the system regards and user perceives as elementary inferential steps.*
> [$\cdots$] *The most challenging and critical part of a proof verifier having this general structure is its inferential core. If this core is powerful enough, the system user will be able to make comfortably large and intuitive formal steps; if not, then large and counterintuitive masses of detail may be required to prove even rather simple statements. The size of the elementary inferential steps which a proof verifier permits directly controls the cost of program proofs in a system based upon the verifier.*
>
> (Jacob T. Schwartz, [28, pp. 6–7])

Much at that time, Jack started an investigation on decision procedures for fragments of set theory, which among others initially involved Alfredo Ferro and Eugenio Omodeo, aiming at the compilation of a large library of procedures to be included in the inferential core of the proof verifier he envisaged. Several of the initial results were collected in the series of papers on *Decision procedures for elementary sublanguages of set theory*, opened with the seminal paper [18] on a decision procedure for *multi-level syllogistic* and some of its extensions. The present paper revives the old paper series to honor Jack's memory.[1]

2 Introduction

Engaging formal proofs, such as those founding the field of mathematical analysis, very often rely on routine forms of set-theoretical reasoning which a human expert exploits almost unconsciously and a computerized proof-checker must encompass as basic inferential services. This paper focuses on situations when such reasoning services are implemented as decision algorithms for fragments of set theory.

[1]Thanks are due to Eugenio Omodeo and Andrea Formisano for their contribution to a very preliminary version of this paper which was presented at the *Workshop on Pragmatics of Decision Procedures in Automated Reasoning* held in Miami in 2003.

We will start with a specific satisfiability test which, despite being rather limited in the constructs it can deal with, proved to be very useful in practice (whereas decision algorithms for more expressive sub-languages of set theory tend to be utterly unmanageable due to their high complexity). We will address the issue: how can we extend the realm of applicability of this decision algorithm without substantial reworking of its internals, but rather via systematic preprocessing techniques? A variant of the method which we will propose as an answer is already at work in the proof-verification system ÆtnaNova/Referee described in [12, 20, 22, 25, 29].

The ideas reported in this paper have connections with the work of Armando et al. (cf. [1]).

3 Multi-level Syllogistic

MLS (*multilevel syllogistic*) is the unquantified fragment of set theory consisting of a denumerable infinity $u, v, w, x, y, z, \ldots$ of set variables, the 'null set' constant $\emptyset$, the set operators $\cdot \cap \cdot$, $\cdot \setminus \cdot$, $\cdot \cup \cdot$, the set predicates $\cdot \in \cdot$, $\cdot = \cdot$, $\cdot \supseteq \cdot$, $\cdot \subseteq \cdot$, and propositional connectives. If also the set operator $\{\cdot, \ldots, \cdot\}$ is admitted, one obtains the fragment **MLSS** (*multilevel syllogistic with singleton*). Simple examples of statements that can be formed using the constructs of **MLSS** are

$$(a \supseteq b \;\&\; b \supseteq c) \Rightarrow (a \supseteq c),$$
$$(a \supseteq b \;\&\; b \cap c = \emptyset) \Rightarrow (a \setminus c \supseteq b),$$
$$\big(a = \{b\} \;\&\; a = c \cup d \;\&\; b \notin c\big) \Rightarrow (a = d \;\&\; c = \emptyset).$$

These are easily seen to be universally valid or, equivalently, their negations are unsatisfiable. To be more precise on the intended semantics of **MLSS**, this is based upon the von Neumann's cumulative hierarchy $\mathcal{V}$ of sets defined as follows (where $\mathcal{O}rd$ and $\mathscr{P}(X)$ designate the class of all ordinals and the power-set of X):

$$\begin{aligned}
\mathcal{V}_0 &=_{\mathrm{Def}} \emptyset; \\
\mathcal{V}_{\alpha+1} &=_{\mathrm{Def}} \mathscr{P}(\mathcal{V}_\alpha), \quad \text{for each ordinal } \alpha; \\
\mathcal{V}_\lambda &=_{\mathrm{Def}} \bigcup_{\mu<\lambda} \mathcal{V}_\mu, \quad \text{for each limit ordinal } \lambda; \\
\mathcal{V} &=_{\mathrm{Def}} \bigcup_{\alpha \in \mathcal{O}rd} \mathcal{V}_\alpha.
\end{aligned}$$

An *assignment* $\mathcal{M}$ over a collection of variables V is any map from V into $\mathcal{V}$. Given an **MLSS**-formula F over a collection V of variables, and an assignment $\mathcal{M}$ over V, we denote by $F^{\mathcal{M}}$ the truth-value of F obtained by interpreting each variable $x \in V$ with the set $x^{\mathcal{M}}$ and the set operators and propositional connectives according to their standard meanings. Such an F is said to be *satisfiable* if it has a *model*, namely, if there exists an assignment $\mathcal{M}$ making $F^{\mathcal{M}}$ true. The satisfiability problem for **MLSS** is then the problem of determining for any given **MLSS**-formula F whether or not F is satisfiable. It was first solved in [18]. For the sake of completeness, we describe below a decision procedure for **MLSS**.

To begin with, given an **MLSS**-formula F with the goal of testing it for satisfiability, let P_F be its *propositional blobbing*, namely the propositional formula obtained by descending the syntax tree of F and reducing each node not marked with a propositional operator to a single propositional variable, in such a way that two subnodes are reduced to distinct propositional variables if and only if they correspond to distinct **MLSS**-atoms. Observe that the truth-value assignment over the propositional variables of P_F induced by a set model of F plainly satisfies P_F. Thus, to test whether F is satisfiable at the set-theoretical level one can first determine, using for instance the Davis-Putnam algorithm (or any other propositional-level algorithm of the same kind), all the truth-value assignments which satisfy P_F and then separately check for satisfiability the collections of negated and non-negated atomic formulae in F corresponding to each of these truth-value assignments. If any such collection of literals is satisfiable, then so is our original formula F. If no truth-value pattern satisfying P_F at the propositional level gives rise to a collection of **MLSS**-literals which can be satisfied at the underlying set-theoretic level, then our original formula F is plainly unsatisfiable. We shall refer to this preliminary propositional level step as *decomposition at the propositional level*.

Decomposition at the propositional level, as described above, may take exponential time in the number of distinct atoms present in the initial formula F. However, when the formula F is satisfiable, for the purpose of certifying its satisfiability one can just guess in linear time a truth-value assignment over the propositional variables of P_F (which satisfies P_F and) whose corresponding collection C of negated and non-negated atomic formulae in F is satisfiable at the set-theoretical level, and then verify that C is indeed satisfiable. If such verification can in turn be done in nondeterministic polynomial time, then it will plainly follow that the whole satisfiability verification process of our initial formula F can be accomplished in nondeterministic polynomial time in the size of F, yielding that the satisfiability problem for **MLSS** is in NP. This fact, coupled with the NP-hardness of (the satisfiability problem for) **MLSS**, which will be shown later, will also imply the NP-completeness of **MLSS**. So, our next step will be to exhibit such a decision test for conjunctions of **MLSS**-literals. In fact, we will further limit ourselves to the decision problem for conjunctions of **MLSS**-literals of very simple types, as argued next.

Any compound set term involving the available operators $\cap$, $\cup$, $\setminus$, $\{\cdot\}$ and the constant $\emptyset$ can be rewritten as a new auxiliary variable, subject to suitable conditions of particularly simple forms, possibly involving other auxiliary variables. For instance, a compound expression of the form $\{\emptyset, a\} \cap (b \setminus (c \cup d))$ can be expressed as the auxiliary set variable u_6, subject to the equational conditions

$$\begin{array}{llll} u_0 = u_0 \setminus u_0, & u_1 = \{u_0\}, & u_2 = \{a\}, & u_3 = u_1 \cup u_2, \\ u_4 = c \cup d, & u_5 = b \setminus u_4, & u_6 = u_3 \cap u_5, & \end{array}$$

each involving exactly one of the allowed set operators $\cap$, $\cup$, $\setminus$, $\{\cdot\}$.

The above observation, together with the fact that any equality of the form $x = y$ can be replaced by the equivalent atomic formula $x = y \cup y$, allows one to express any set equality of the form $t_1 = t_2$, where each of t_1 and t_2 is either a set variable or a compound set term, by a conjunction of equational conditions each involving exactly one set operator. Similarly, an inequality of the form $t_1 \neq t_2$ can be reduced to a conjunction of an inequality of the simpler form $u \neq v$, where u and v are variables, with equational conditions each involving one set operator. Likewise, literals of the form $t_1 \in t_2$ and $t_1 \notin t_2$, with t_1 and t_2 variables or compound terms, can be reduced to literals of the simpler forms $u \in v$ and $u \notin v$, respectively, conjoined with equational conditions each involving one set operator. Finally, inclusions like $y \supseteq x$ and $x \subseteq y$ can be replaced by the equivalent atomic formulae $y = x \cup y$.[2] By systematically applying simplifications of this second kind, which we will call *secondary decomposition*, and the decomposition at the propositional level discussed earlier, it turns out that the satisfiability problem for **MLSS** can be reduced to the satisfiability problem for conjunctions of **MLSS**-literals, each having one of the 'flat' forms

$$\begin{array}{llll} x = y \cup z, & x = y \cap z, & x = y \setminus z, & x \neq y, \\ x \in y, & x \notin y, & y = \{x\}, & \end{array} \tag{1}$$

where x, y, z stand for set variables. In fact, though we will not do it, it would be possible to further reduce the number of different forms present in (1) by observing, for instance, that

- $x \cap y$ can be rewritten as $x \setminus (x \setminus y)$,
- $x \neq y$ is equisatisfiable with $(u \in x \,\&\, u \notin y) \vee (u \notin x \,\&\, u \in y)$, where u is any newly introduced variable,
- etc.

Observe that secondary decomposition can be carried out in deterministic linear time.

We next solve the satisfiability problem for conjunctions of **MLSS**-literals of the forms (1). Thus, let C be such a conjunction.

We begin by deriving some necessary (computable) conditions for the satisfiability of C and later show that they are also sufficient, thereby proving the decidability of the theory **MLSS**. Thus, assume that C is satisfiable and let $\mathcal{M}$ be a model of C. Let $\sim_{\mathcal{M}}$ be the equivalence relation over the set variables occurring in C such that

$$x \sim_{\mathcal{M}} y \quad \text{iff} \quad x^{\mathcal{M}} = y^{\mathcal{M}}.$$

We select a representative in any of the equivalence classes of $\sim_{\mathcal{M}}$ and replace each occurrence in C of a variable by its selected representative. Let C' be the resulting **MLSS**-formula. It follows immediately that $\mathcal{M}$ satisfies C' (therefore C' cannot contain any clause of the form $x \neq x$), but now $x^{\mathcal{M}} \neq y^{\mathcal{M}}$ holds, for any two distinct set variables x, y occurring in C': in other words, $\mathcal{M}$ is an *injective* model of C'.

[2]From a practical point of view, it would be more convenient to eliminate equalities of the form $x = y$ by replacing all occurrences of x and y in our set of statements by a selected representative of x, y, and then drop the atom $x = y$.

Let V be the collection of set variables occurring in C' and let $U = \bigcup_{x \in V} x^{\mathcal{M}}$ be the *universe* of $\mathcal{M}$. We form the collection Σ of the nonempty disjoint regions σ of the Venn diagram of the sets $x^{\mathcal{M}}$, for $x \in V$, over the universe U (called *parts of the Venn diagram of* $\mathcal{M}$): these are the equivalence classes of the equivalence relation $\sim$ defined on U by

$$s \sim t \quad \text{iff} \quad s \in x^{\mathcal{M}} \Leftrightarrow t \in x^{\mathcal{M}}, \quad \text{for all set variables } x \in V.$$

Observe that for any set $x^{\mathcal{M}}$, each σ in Σ is either fully contained in $x^{\mathcal{M}}$ or is completely disjoint from it. Therefore, to each set $\sigma \in \Sigma$ we can associate a Boolean-valued map π_σ over V by putting

$$\pi_\sigma(x) = \begin{cases} \textbf{true} & \text{if } \sigma \subseteq x^{\mathcal{M}} \\ \textbf{false} & \text{if } \sigma \cap x^{\mathcal{M}} = \emptyset. \end{cases} \tag{2}$$

Let Π be the collection of all such maps. The following properties hold for the maps π in Π:

(a) $\pi(x) \Leftrightarrow \pi(y) \vee \pi(z)$ whenever $x = y \cup z$ appears in C',
(b) $\pi(x) \Leftrightarrow \pi(y) \,\&\, \pi(z)$ whenever $x = y \cap z$ appears in C',
(c) $\pi(x) \Leftrightarrow \pi(y) \,\&\, \neg\pi(z)$ whenever $x = y \setminus z$ appears in C'.

Boolean-valued maps defined on the set variables in V and satisfying the properties (a)–(c) above are called *places for* C'.

Observe that for each variable y occurring in any statements of C' we have

$$y^{\mathcal{M}} = \bigcup \{\sigma : \sigma \in \Sigma \mid \pi_\sigma(y)\}, \tag{3}$$

where $\bigcup S$ (the union of S) stands for $\bigcup_{s \in S} s$.

Next we look a bit more closely at the structure of the model $\mathcal{M}$, with an eye toward accumulating enough properties of its places to guarantee the existence of at least one model of C'. To begin with, we show that the collection Π of places for C' is *ample*, in the sense that for any two distinct set variables x, y occurring in C' there is a place π in Π such that $\pi(x) \neq \pi(y)$. Indeed, let x, y be two distinct set variables in C'. Then, as observed above, $x^{\mathcal{M}} \neq y^{\mathcal{M}}$, so that for some region $\sigma \in \Sigma$ of the Venn diagram of $\mathcal{M}$ we must have

$$\sigma \subseteq x^{\mathcal{M}} \quad \text{iff} \quad \sigma \not\subseteq y^{\mathcal{M}}$$

and therefore, for the place π in Π corresponding to σ, we have $\pi(x) \neq \pi(y)$.

It could be shown that the existence of an ample collection Π of places for C' is already sufficient for the injective satisfiability of C', provided that no literal of any of the forms $x \in y$, $x \notin y$, and $y = \{x\}$ be present in C'. To take into account also literals of the latter types, we reason as follows.

We first form the collection L of all the variables x in V which appear in statements of C' of the form $x \in y$ or $x = \{y\}$. These we call *left-hand* variables. All left-hand variables are modeled by $\mathcal{M}$ with sets belonging as elements to the universe U of $\mathcal{M}$. Hence, for any given left-hand variable x, there is a unique part σ^x of the Venn diagram of $\mathcal{M}$ for which $x^{\mathcal{M}} \in \sigma^x$. It is immediate to check that the following properties hold for the corresponding place $\pi^x \in \Pi$:

(a′) $\pi^x(y)$ is true if a statement $x \in y$ or a statement $y = \{x\}$ appears in C';
(b′) $\pi^x(y)$ is false if a statement $x \notin y$ appears in C'.

We call any place π having these two properties a *place at* x. Some of the places in Π are places at x for some left-hand variable x in the set C' of statements, others are not.

In terms of the definition just given, we have shown that there is a map $x \mapsto \pi^x$ from the collection L of left-hand variables into the collection of places Π, such that π^x is a place at x, for any left-hand variable x.

If a statement $y = \{x\}$ appears in C', then $y^{\mathcal{M}}$ must be a singleton, so that, by (3), there must be a unique place π in Π for which $\pi(y)$ is true, and since $\pi^x(y)$ is true, the place π^x at x must be indeed the only place π in Π such that $\pi(y)$ is true.

Finally, since set theory forbids all cycles

$$s_1 \in s_2 \in \cdots \in s_k \in s_1$$

of membership, it must be possible to arrange the sets $x^{\mathcal{M}}$ of our model into an order $\prec$ for which $x \prec y$ whenever $x^{\mathcal{M}} \in y^{\mathcal{M}}$. Note that for any left-hand variable x and variable y,

$$\text{if } \pi^x(y) \text{ then } x \prec y.$$

We will call any order of the variables of C' satisfying the latter property an *acceptable ordering*.

The following theorem shows that the conditions that we have just enumerated are also sufficient to guarantee the existence of a model of C, and so gives us a procedure for determining the satisfiability of conjunctions of **MLSS** literals of the form (1).

Theorem 1 *Let C be a collection of statements of the form* (1). *Then C is satisfiable, i.e. it has a model $\mathcal{M}$, if and only if there exists an equivalence relation $\sim$ over the set variables occurring in C such that by letting C' be the collection of statements obtained when each occurrence of any variable in C is replaced by the selected representative in its $\sim$-equivalence class, the following conditions hold*:

(i) *C' does not contain any literal of the form $x \neq x$.*
(ii) *C' has an ample set Π of places* (*for C'*), *i.e., for each pair x, y of distinct set variables occurring in C' there is a π in Π such that $\pi(x) \neq \pi(y)$.*
(iii) *For each left-hand variable x appearing in a statement of C', there is a place π^x at x in Π. Moreover, the variables appearing in the statements of C' can be arranged in an order such that $\pi^x(y)$ is false unless x precedes y in this order, where x is a left-hand variable.*

(iv) *If a statement $y = \{x\}$ appears in C', then π^x is the only place π in Π for which $\pi(y)$ is true.*

Proof We have already shown that conditions (i)–(iv) are necessary.

Suppose conversely that they are satisfied. For each place $\pi \in \Pi$ such that $\pi(x)$ holds for some left-hand variable x for which a statement $y = \{x\}$ appears in C', we put $\overline{\pi} = \emptyset$; for each remaining place π we choose a distinct singleton $\overline{\pi}$ in such way that its only member has cardinality larger than $m + n$, where m is the total number of left-hand variables appearing in C' and n is the cardinality of Π. Thus, all the sets $\overline{\pi}$ are pairwise disjoint. Then we use the formula

$$y^{\mathcal{M}} = \{x^{\mathcal{M}} : x \in L \mid \pi^x(y)\} \cup \bigcup\{\overline{\pi} : \pi \in \Pi \mid \pi(y)\} \tag{4}$$

to define $y^{\mathcal{M}}$ for each variable y appearing in C', following the order in which the variables of C' are arranged, according to condition (iii). This is possible, since if $x^{\mathcal{M}}$ is needed to define $y^{\mathcal{M}}$, then $\pi^x(y)$ holds and therefore the variable x must precede y, so that the value $x^{\mathcal{M}}$ is available when needed to construct the value $y^{\mathcal{M}}$. We will refer to $\mathcal{M}$ as a *canonical model* (after we prove that $\mathcal{M}$ satisfies C').

In view of the cardinality conditions on the member of each of the (singleton) sets $\overline{\pi}$, it follows from (4) that the cardinality of each set $y^{\mathcal{M}}$ is at most $m + n$. This ensures that

$$\{z^{\mathcal{M}} : z \in L\} \cap \bigcup\{\overline{\pi} : \pi \in \Pi\} = \emptyset,$$

and therefore

$$\{z^{\mathcal{M}} : z \in L \mid \pi^z(x)\} \cap \bigcup\{\overline{\pi} : \pi \in \Pi \mid \pi(y)\} = \emptyset \tag{5}$$

for every pair of variables x, y in C'.

We now show that the map $\mathcal{M}$ defined by (4) satisfies all the statements $x \neq y$ in C'. To this purpose, since, by (i), C' does not contain any literal of the form $x \neq x$, it is sufficient to show that $\mathcal{M}$ is injective.

So, let x and y be two distinct variables in C' and assume by way of contradiction that $x^{\mathcal{M}} = y^{\mathcal{M}}$; in fact, let us assume that x is the first variable, in the ordering mentioned in condition (iii), for which there exists a variable v in C', distinct from x, such that $x^{\mathcal{M}} = v^{\mathcal{M}}$. Then, by (5), we have

$$\{z^{\mathcal{M}} : z \in L \mid \pi^z(x)\} = \{z^{\mathcal{M}} : z \in L \mid \pi^z(y)\}$$

and

$$\bigcup\{\overline{\pi} : \pi \in \Pi \mid \pi(x)\} = \bigcup\{\overline{\pi} : \pi \in \Pi \mid \pi(y)\}.$$

Since the set Π of places is ample, there must exist a place π' in Π such that one of $\pi'(x), \pi'(y)$ is true and the other is false. Suppose for definiteness that $\pi'(x)$ is true, so $\pi'(y)$ is false. But then, because of the mutual disjointness of the sets $\overline{\pi}$, we must have both

$$\overline{\pi}' \subseteq \bigcup \{ \overline{\pi} : \pi \in \Pi \mid \pi(x) \}$$

and

$$\overline{\pi}' \cap \bigcup \{ \overline{\pi} : \pi \in \Pi \mid \pi(x) \} = \overline{\pi}' \cap \bigcup \{ \overline{\pi} : \pi \in \Pi \mid \pi(y) \} = \emptyset,$$

which imply that $\overline{\pi}' = \emptyset$, so that π' must be of the form $\pi' = \pi^z$, where z is some left-hand variable. Since $z^{\mathcal{M}} \in x^{\mathcal{M}} = y^{\mathcal{M}}$ and $\pi^z(y)$ is false, from (5) it follows that $z^{\mathcal{M}}$ must be identical with some $v^{\mathcal{M}}$ for which $\pi^v(y)$ is true. Both z and v must be left-hand variables, and they must be distinct, since $\pi^z(y)$ is false while $\pi^v(y)$ is true. But now $z^{\mathcal{M}} = v^{\mathcal{M}}$ contradicts our assumption that x is the first variable in the ordering mentioned in condition (iii) for which there exists a y such that $x^{\mathcal{M}} = y^{\mathcal{M}}$, as z precedes x in the same ordering. This contradiction proves our claim that the function $\mathcal{M}$ is injective and so shows that all clauses of the form $x \neq y$ in C' are correctly modeled by $\mathcal{M}$.

Next we show that all other statements of C' are correctly modeled also.

If a statement $y = \{x\}$ occurs in C', then from condition (iv) of our theorem π^x is the only place $\pi \in \Pi$ for which $\pi(y)$ is true. Hence, by (4),

$$y^{\mathcal{M}} = \{ x^{\mathcal{M}} \} \cup \overline{\pi}^x = \{ x^{\mathcal{M}} \},$$

as $\overline{\pi}^x = \emptyset$, i.e. $\mathcal{M}$ satisfies the statement $y = \{x\}$. Statements $x \in y$ are correctly modeled by $\mathcal{M}$ since the presence of such a statement implies that $\pi^x(y)$ must be true and therefore $x^{\mathcal{M}}$ must belong to the first term of (4). Statements $x \notin y$ are correctly modeled, since if $x^{\mathcal{M}} \in y^{\mathcal{M}}$, then $\pi^x(y)$ must be true (by (5) and the injectivity of $\mathcal{M}$), which is impossible if $x \notin y$ appears in C'.

Statements $x = y \cup z$ are correctly modeled since

$$\begin{aligned}
x^{\mathcal{M}} &= \{ u^{\mathcal{M}} : u \in L \mid \pi^u(x) \} \cup \bigcup \{ \overline{\pi} : \pi \in \Pi \mid \pi(x) \} \\
&= \{ u^{\mathcal{M}} : u \in L \mid \pi^u(y) \vee \pi^u(z) \} \cup \bigcup \{ \overline{\pi} : \pi \in \Pi \mid \pi(y) \vee \pi(z) \} \\
&= \left(\{ u^{\mathcal{M}} : u \in L \mid \pi^u(y) \} \cup \{ u^{\mathcal{M}} : u \in L \mid \pi^u(z) \} \right) \\
&\quad \cup \left(\bigcup \{ \overline{\pi} : \pi \in \Pi \mid \pi(y) \} \cup \bigcup \{ \overline{\pi} : \pi \in \Pi \mid \pi(z) \} \right) \\
&= \left(\{ u^{\mathcal{M}} : u \in L \mid \pi^u(y) \} \cup \bigcup \{ \overline{\pi} : \pi \in \Pi \mid \pi(y) \} \right) \\
&\quad \cup \left(\{ u^{\mathcal{M}} : u \in L \mid \pi^u(z) \} \cup \bigcup \{ \overline{\pi} : \pi \in \Pi \mid \pi(z) \} \right) \\
&= y^{\mathcal{M}} \cup z^{\mathcal{M}},
\end{aligned}$$

if $x = y \cup z$ occurs in C'.

Similarly, in view of the pairwise disjointness of the sets $\overline{\pi}$, the injectivity of $\mathcal{M}$ and condition (5), if a statement $x = y \setminus z$ appears in C' we have

$$
\begin{aligned}
x^{\mathcal{M}} &= \left\{u^{\mathcal{M}} : u \in L \,\middle|\, \pi^u(x)\right\} \cup \bigcup \left\{\overline{\pi} : \pi \in \Pi \,\middle|\, \pi(x)\right\} \\
&= \left\{u^{\mathcal{M}} : u \in L \,\middle|\, \pi^u(y) \,\&\, \neg\pi^u(z)\right\} \cup \bigcup \left\{\overline{\pi} : \pi \in \Pi \,\middle|\, \pi(y) \,\&\, \neg\pi(z)\right\} \\
&= \left(\left\{u^{\mathcal{M}} : u \in L \,\middle|\, \pi^u(y)\right\} \setminus \left\{u^{\mathcal{M}} : u \in L \,\middle|\, \pi^u(z)\right\}\right) \\
&\quad \cup \left(\bigcup \left\{\overline{\pi} : \pi \in \Pi \,\middle|\, \pi(y)\right\} \setminus \bigcup \left\{\overline{\pi} : \pi \in \Pi \,\middle|\, \pi(z)\right\}\right) \\
&= \left(\left\{u^{\mathcal{M}} : u \in L \,\middle|\, \pi^u(y)\right\} \cup \bigcup \left\{\overline{\pi} : \pi \in \Pi \,\middle|\, \pi(y)\right\}\right) \\
&\quad \Big\backslash \left(\left\{u^{\mathcal{M}} : u \in L \,\middle|\, \pi^u(z)\right\} \cup \bigcup \left\{\overline{\pi} : \pi \in \Pi \,\middle|\, \pi(z)\right\}\right) \\
&= y^{\mathcal{M}} \setminus z^{\mathcal{M}},
\end{aligned}
$$

proving that $\mathcal{M}$ models correctly the statement $x = y \setminus z$.

A similar argument handles the case of statements $x = y \cap z$, thus completing the proof that $\mathcal{M}$ models correctly C'.

It is an easy matter to show that $\mathcal{M}$ can be extended to a model of our initial conjunction C, by putting $x^{\mathcal{M}} =_{\mathrm{Def}} \mathsf{repr}(x)^{\mathcal{M}}$, for every variable x in C, where $\mathsf{repr}(x)$ is the representative of the $\sim$-equivalence class of x.

This completes the proof of our theorem. □

Theorem 1, together with the correctness of the decomposition at the propositional level and of the secondary decomposition seen earlier, readily implies the decidability of the full **MLSS** fragment. However, from it we can only infer that the satisfiability problem for conjunctions of **MLSS**-literals of the simple forms (1) takes nonderministic *exponential* time, rather than nonderministic polynomial time, as anticipated earlier. This is due to the fact that Theorem 1 does not provide any polynomial upper bound on the size of the set of places Π mentioned in condition (ii): we know only that Π must have size at most 2^p, where p is the number of variables occurring in the collection of statements C' cited in the same theorem.

In the informal derivation of the (necessity of the) conditions of Theorem 1, as set of places Π we selected the collection of all maps π_σ defined by (2), for $\sigma \in \Sigma$, where Σ was the collection of *all* nonempty disjoint regions σ of the Venn diagram of the sets $x^{\mathcal{M}}$ in the universe $U = \bigcup_{x \in V} x^{\mathcal{M}}$ and V was the collection of set variables occurring in C'. However, a closer examination of the line of reasoning given there makes it clear that it is enough that Π contain a place π^x at x for each left-hand variable x occurring in C' and a place π such that $\pi(x)$ and $\pi(y)$ have different truth values, for every pair of distinct variables x, y in C'. It therefore follows that the size of Π can be constrained to be not larger than $\frac{p(p+1)}{2}$, where p is again the number of distinct variables in C', yielding, as promised, a nondeterministic polynomial time satisfiability test for conjunctions of flat **MLSS**-literals. In fact, the size of Π can be further constrained to be linear in p, as will be shown next, for the sake of completeness.

Let C', V, L, $\mathcal{M}$, Σ, Π be as before, so that, in particular, Π is the collection of all places π_σ defined by (2), for $\sigma \in \Sigma$ and $\mathcal{M}$ is an injective model of C'. Given $\Sigma' \subseteq \Sigma$ and $V' \subseteq V$, we say that Σ' *distinguishes* V' if for every pair of distinct variables $x, y \in V'$ there exists a set $\sigma \in \Sigma'$ such that σ is contained in exactly one of the two sets $x^{\mathcal{M}}$ and $y^{\mathcal{M}}$ and is disjoint from the other (in this case we say that σ *distinguishes* the variables x and y). A useful property is the following: if Σ' distinguishes V', then for every x in V there exists a set $\sigma \in \Sigma$ such that $\Sigma' \cup \{\sigma\}$ distinguishes $V' \cup \{x\}$. Indeed, if Σ' does not distinguish $V' \cup \{x\}$ (but distinguishes V'), there must be a variable y in V' (therefore distinct from x) such that x and y are not distinguished by Σ'. On the other hand, $x^{\mathcal{M}} \neq y^{\mathcal{M}}$ (by the injectivity of $\mathcal{M}$) and therefore for some region σ of the Venn diagram Σ of $\mathcal{M}$ it must be the case that σ is contained in exactly one of the two sets $x^{\mathcal{M}}$ and $y^{\mathcal{M}}$ and is disjoint from the other. We claim that $\Sigma' \cup \{\sigma\}$ distinguishes $V' \cup \{x\}$. Indeed, if this were not the case, then, as before, there must be a variable z in V', distinct from x, such that x and z are not distinguishable by $\Sigma' \cup \{\sigma\}$. In addition, z must be distinct from y, since by construction σ distinguishes x and y. Hence z and y are distinguished by a region $\sigma' \in \Sigma'$ (as, by assumption, Σ' distinguishes V'). Since x and y are not distinguished by Σ', it follows that σ' distinguishes also x and z, contradicting our initial assumption that x and z were not distinguishable by $\Sigma' \cup \{\sigma\}$, thus proving that $\Sigma' \cup \{\sigma\}$ distinguishes $V' \cup \{x\}$. The property that we have just proved together with the fact that $\{x\}$ is vacuously distinguished by the empty set of Venn regions, for every variable $x \in V$, allow us to conclude, by induction, that there exist a subset Σ' of Σ of size at most $p - 1$, where p is the cardinality of V, which distinguishes V. It can easily be checked then that any subset of Π which contains the set of places $\{\pi_\sigma : \sigma \in \Sigma'\}$ is ample. This, for instance, is the case for the set of places

$$\Pi' = \left\{\pi_\sigma : \sigma \in \Sigma'\right\} \cup \left\{\pi^x : x \in L\right\},$$

whose cardinality is bounded by $p+m-1$, where m is the total number of left-hand variables appearing in C'.

In view of the above observations, it follows that condition (ii) in Theorem 1 can be strengthened as follows:

(ii$'$) *C' has an ample set Π of places (for C'), whose cardinality is bounded by $p + m - 1$, where p and m are respectively the number of variables and of left-hand variables in C'.*

Theorem 1, but with condition (ii) replaced by condition (ii$'$) above, now implies that one can verify in nondeterministic polynomial time the satisfiability of a given satisfiable conjunction C of flat **MLSS**-literals. Indeed, one has to guess

- a suitable equivalence relation $\sim$ over the set variables occurring in C (linear nondeterministic time),

- a suitable set Π of places for C', where C' is the collection of statements obtained when each occurrence of any variable in C is replaced by the selected representative in its $\sim$-equivalence class (linear nondeterministic time),
- a suitable ordering of the variables in C' (linear nondeterministic time)

and then verify that conditions (i), (ii′), (iii), and (iv) are satisfied (polynomial deterministic time).

Since, as already observed, decomposition at the propositional level and secondary decomposition can be performed in nondeterministic linear time, it follows that the satisfiability problem for the whole fragment **MLSS** is in NP.

To prove the NP-completeness of **MLSS**, it is now enough to show its NP-hardness, which we do by exhibiting a polynomial-time reduction of SAT to **MLSS**.

Let X be any propositional formula. For each propositional variable P in X, introduce a set variable x_P. Let also z be any new set variable, distinct from all variables x_P just introduced. Finally, let C_X be the **MLSS**-formula obtained by replacing in X each occurrence of a propositional variable P by the corresponding **MLSS**-atom $x_P \in z$. It is an easy matter to show that X is propositionally satisfiable if and only if C_X is satisfiable at the set-theoretical level, thus proving the NP-hardness of the satisfiability problem for **MLSS**.

Summing up, we have proved the following result.

Theorem 2 *The satisfiability problem for* **MLSS** *is* NP*-complete.*

Similarly, it is easy to exhibit a polynomial-time reduction of 3SAT to the collection of conjunctions of flat **MLSS**-literals (of the form (1)), thus establishing also the following result.

Theorem 3 *The satisfiability problem for conjunctions of flat* **MLSS**-*literals is* NP-*complete.*[3]

Another consequence of the strengthened form of Theorem 1 is that a conjunction of flat **MLSS**-literals C is satisfiable if and only if it has a set model $\mathcal{M}$ whose size is bounded by $p + m - 1$ (i.e., $x^{\mathcal{M}}$ has at most $p + m - 1$ members, for each set variable x in C), where p and m are respectively the number of variables and the number of left-hand variables in C. It would not be hard to derive a bound on the size of a "canonical" model for general **MLSS**-formulae. Similar bounds could also be derived for the *rank* of canonical models of **MLSS**-formulae, where the rank measures the nesting depth of a set.

[3] NP-completeness of flat conjunctions of **MLSS**-literals was first proved in [10], using a different approach.

3.1 A Tableau-Based Decision Procedure for MLSS

The decision procedure for **MLSS** contained in Theorem 1 is not very efficient from a practical point of view, though initially all implementations of the decision test for **MLSS** were based on the ‘place’ approach. A more efficient approach uses semantic tableaux (for general notions on semantic tableaux, the reader is referred to [19].)

The advantages of implementing decision procedures as complete and effective saturation strategies for tableau calculi over *ad hoc* methods are manyfold: firstly, tableaux maintain information on proof attempts in a very natural and readable way; such information can then be used either to reconstruct proofs or, in case of formulae which are not theorems, to construct counter-examples (this can be particularly useful to understand what was wrong in the conjecture one was about to prove); secondly, tableau calculi can easily be extended by new rules, thus allowing, in favourable cases, smooth generalizations to more expressive decidable fragments; thirdly, implementations of saturation strategies for tableau calculi can be equipped with heuristics (for instance a user could easily deactivate some of the rules, impose restrictions on their applicability, etc.).

The current implementation of the decision test for **MLSS** in the proof-verification system ÆtnaNova/Referee is mainly based on a variant of the tableau system examined in [16]. Other related versions have been presented in [4, 11].

Since one of the rules of the tableau calculus for **MLSS**, which we are about to illustrate, can introduce literals of the form $x = y$ (specifically, rule (9) in Table 1), we will not bother to eliminate these equalities from the flat forms, as was done previously. We therefore assume that after decomposition at the propositional level and subsequent secondary decomposition, we are left with a collection of formulae to be tested for satisfiability each of which is a conjunction of literals of the following forms

$$\begin{array}{llll} x = y \cup z, & x = y \cap z, & x = y \setminus z, & x = y, \\ x \neq y, & x \in y, & x \notin y, & y = \{x\}, \end{array} \tag{6}$$

where, as before, x, y, z range over set variables.

The rules of the tableau calculus for **MLSS** of our interest are listed in Table 1. Rules (2), (7), and (11) are called *splitting* rules, while the remaining ones are the *linear* rules.

Given a finite collection $\mathcal{S}$ of **MLSS**-literals of the form (6), an INITIAL **MLSS**-TABLEAU for $\mathcal{S}$ is a one-branch tree whose nodes are labeled by the literals in $\mathcal{S}$ (in any order). Then, an **MLSS**-TABLEAU for $\mathcal{S}$ is a finite tree whose nodes are labeled by **MLSS**-literals and which can be constructed from an initial tableau for $\mathcal{S}$ by a finite number of applications of the rules (1)–(13) in Table 1. More specifically, the application of one of the rules (1)–(13) to a given **MLSS**-tableaux $\mathcal{T}$ consists in (i) selecting a branch ϖ of $\mathcal{T}$, then (ii) selecting on ϖ a number of nodes equal to the number of premises present in the rule, whose labels match the rule premises via a suitable matching substitution, and finally (iii) prolonging ϖ with new nodes labeled by the literals present in the rule conclusion, after the same matching substitution has been applied to them. In particular, in the case of a linear rule, the branch

Table 1 Tableau rules for **MLSS**

$$\frac{\begin{array}{c} x = y_1 \cup y_2 \\ z \in y_i \end{array}}{z \in x}\ \text{(i)} \qquad \frac{\begin{array}{c} x = y_1 \cup y_2 \\ z \in x \end{array}}{z \in y_1 \mid z \in y_2}\ \text{(ii)} \qquad \frac{\begin{array}{c} x = y_1 \cap y_2 \\ z \in y_1 \\ z \in y_2 \end{array}}{z \in x}\ \text{(iii)} \qquad \frac{\begin{array}{c} x = y_1 \cap y_2 \\ z \in x \end{array}}{\begin{array}{c} z \in y_1 \\ z \in y_2 \end{array}}\ \text{(iv)}$$

$$\frac{\begin{array}{c} x = y_1 \setminus y_2 \\ z \in x \end{array}}{\begin{array}{c} z \in y_1 \\ z \notin y_2 \end{array}}\ \text{(v)} \qquad \frac{\begin{array}{c} x = y_1 \setminus y_2 \\ z \in y_1 \\ z \notin y_2 \end{array}}{z \in x}\ \text{(vi)} \qquad \frac{\begin{array}{c} x = y_1 \setminus y_2 \\ z \in y_1 \end{array}}{z \in y_2 \mid z \notin y_2}\ \text{(vii)} \qquad \frac{x = \{y\}}{y \in x}\ \text{(viii)}$$

$$\frac{\begin{array}{c} x = \{y\} \\ z \in x \end{array}}{z = y}\ \text{(ix)} \qquad \frac{\begin{array}{c} y_1 \in x \\ y_2 \notin x \end{array}}{y_1 \neq y_2}\ \text{(x)} \qquad \frac{x \neq y}{\begin{array}{c|c} w \in x & w \notin x \\ w \notin y & w \in y \end{array}}\ \text{(xi)}^{a}$$

$$\frac{\begin{array}{c} x = y \\ \ell \end{array}}{\ell^x_y}\ \text{(xii)}^{b} \qquad \frac{\begin{array}{c} y = x \\ \ell \end{array}}{\ell^x_y}\ \text{(xiii)}^{b}$$

[a] w must be a new variable not occurring on the branch to which the rule is applied

[b] By ℓ^x_y we denote the literal resulting from ℓ by replacing each occurrence of x by y

is prolonged linearly, whereas in the case of a splitting rule the branch is split into two branches. Figure 1 shows an **MLSS**-tableau for the collection of literals $x = \{y\}$, $x = z \cup v$, $y \notin z$, $x \neq v$.

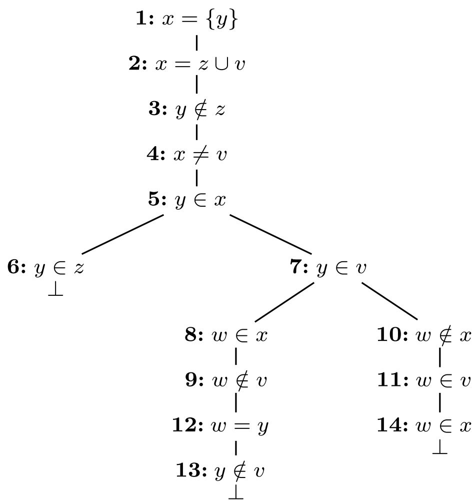
Let $\mathcal{T}$ be an **MLSS**-tableau for $\mathcal{S}$. A branch ϖ of $\mathcal{T}$ is said to be

- STRICT, if no rule has been applied more than once on ϖ to the same literal occurrences;
- SATURATED, if each of the tableau rules (1)–(13) has been applied at least once on each instance of its premises on ϖ;
- CLOSED, if either ϖ contains a set of literals of the form

$$x \in x_1, \qquad x_1 \in x_2, \qquad \ldots, \qquad x_{n-1} \in x_n, \qquad x_n \in x,$$

  (with $n \geqslant 0$) forming a membership cycle, or it contains a pair of complementary literals $X, \neg X$;
- OPEN, if it is not closed;
- SATISFIABLE, if there exists a set model for the literals occurring on ϖ.

Fig. 1 A closed **MLSS**-tableau



A tableau $\mathcal{T}$ is said to be

- STRICT, or SATURATED, or CLOSED, if such are all of its branches;
- SATISFIABLE, or OPEN, if such is at least one of its branches.

Notice that according to the above definition, any closed branch, and therefore any closed **MLSS**-tableau, is clearly unsatisfiable.

The system of rules (1)–(13) is plainly *sound*, namely every **MLSS**-tableau for a satisfiable collection of **MLSS**-literals must be satisfiable, and therefore must be open.

In addition, it can be shown that the tableau calculus in Table 1 is *complete*, in the sense that any unsatisfiable collection of **MLSS**-literals has a closed **MLSS**-tableau.

Though our tableau calculus is complete, in order to use it effectively to decide whether a given collection of **MLSS**-literals is satisfiable or not, one has to find a systematic way to apply its rules in such a way that after a finite number of rule applications (no greater than a suitable computable function of the size of the initial formula of **MLSS**) one ends up either with a closed tableau (in which case the initial formula is unsatisfiable) or with the knowledge that no closed tableau for the initial formula exists (in which case the initial formula is satisfiable). Two are the possible sources of problems with such approach. The first problem is that the same rule could be applied repeatedly with the same premises on a same branch. This problem is solved by imposing the restriction:

R1. All applications of any tableau rule are strict.

The second problem is that each use of rule (11) introduces a fresh set variable and, therefore, can lead in some cases to an unbounded sequence of new applications of the same rule, even under the strictness restriction R1. We address such problem by imposing the following restriction:

R2. Rule (11) can be applied only to literals of the form $x \neq y$ involving variables already occurring in the initial formula.

We observe that completeness of our tableau calculus is not disrupted by restrictions R1 and R2. More generally, it can be shown that starting with an initial collection $\mathcal{S}$ of flat **MLSS**-literals, any tableau construction strategy subject to the above restrictions R1 and R2 terminates in a finite number of steps, generating a saturated tableau $\mathcal{T}_{\mathcal{S}}$ for $\mathcal{S}$. Hence the decidability of **MLSS** follows from the fact that $\mathcal{S}$ is satisfiable if and only if the tableau $\mathcal{T}_{\mathcal{S}}$ is open. In view of the soundness of the tableau rules (1)–(13), one only needs to check that if $\mathcal{T}_{\mathcal{S}}$ is open then $\mathcal{S}$ is satisfiable. We sketch the proof of this fact, so let us assume that $\mathcal{T}_{\mathcal{S}}$ is open and let ϖ be an open (saturated) branch of $\mathcal{T}_{\mathcal{S}}$. It is enough to show that the branch ϖ is satisfiable. To this purpose, let us introduce the following notations

$V_{\mathcal{S}}$: the collection of variables occurring in $\mathcal{S}$;
T: the collection of variables occurring on ϖ other than $V_{\mathcal{S}}$;
$\sim_{\mathcal{S}}$: the equivalence relation induced on $V_{\mathcal{S}} \cup T$ by equality literals $x = y$ in ϖ;
T': the set $\{t \in T : t \not\sim_{\mathcal{S}} x, \text{for all } x \in V_{\mathcal{S}}\}$;
V': the set $(V_{\mathcal{S}} \cup T) \setminus T'$;
$\widehat{\in}_{\varpi}$: the dyadic relation on $V' \cup T'$ defined as follows:

$$x \widehat{\in}_{\varpi} y \quad \text{iff} \quad \text{the literal } x \in y \text{ is in } \varpi.$$

In addition, for each $t \in T'$, let $\boldsymbol{u}_t$ be an assigned set.

Since the branch ϖ is not closed, the relation $\widehat{\in}_{\varpi}$ is acyclic. Therefore we can recursively define the following assignment, called the *realization* of the branch ϖ relative to $\mathcal{S}$ and to the sets $\boldsymbol{u}_t$, for $t \in T'$:

$$\begin{aligned} x^{R_\varpi} &= \left\{ y^{R_\varpi} \mid y \widehat{\in}_{\varpi} x \right\}, && \text{if } x \in V'; \\ t^{R_\varpi} &= \boldsymbol{u}_t, && \text{if } t \in T'. \end{aligned}$$

It can easily be checked that if the sets $\boldsymbol{u}_t$ satisfy the conditions

(a) $\boldsymbol{u}_{t_1} \neq \boldsymbol{u}_{t_2}$, for every pair of distinct $t_1, t_2 \in T'$,
(b) $\boldsymbol{u}_t \neq x^{R_\varpi}$, for all $t \in T'$ and $x \in V'$,

then the realization R_ϖ is a model for ϖ, and in turn for $\mathcal{S}$. Since conditions (a) and (b) can always be enforced, for instance by choosing $|T'|$ distinct sets $\boldsymbol{u}_t$ of large enough cardinalities (cf. the cardinality condition in the proof of Theorem 1), completeness of our tableau calculus follows.

It is also interesting to note that the realization of an open non-saturated branch, called *partial realization*, can guide the saturation process, as discussed in depth in [4].

Figure 1 contains a closed **MLSS**-tableau for the collection

$$\mathcal{S} = \left\{ x = \{y\},\ x = z \cup v,\ y \notin z,\ x \neq v \right\}$$

of flat **MLSS**-literals. Closed branches are terminated with the symbol $\perp$.

Notice that in the above **MLSS**-tableau

- literals 1–4 form the initial tableau for $\mathcal{S}$;
- literal 5 has been added by rule (8);
- literals 6 and 7 have been added by rule (2);
- literals 8–11 have been added by rule (11);
- literal 12 has been added by rule (9);
- literal 13 has been added by rule (12);
- literal 14 has been added by rule (1).

To better understand the above remark about using the realization of an open non-saturated branch to guide the saturation process, consider the tableau in Fig. 1 just after the introduction of literals 6 and 7 (and therefore before any of the literals 8–14 have been added to the tableau). The sub-branch ending at node 6 is closed because it contains the pairs of complementary literals $y \notin z$ and $y \in z$ and therefore does not require any further attention. Next, let us consider the subbranch ϖ ending at node 7 and let $\widehat{\in}_{\varpi}$ and R_{ϖ} be their associated dyadic relation and realization, respectively. Plainly, we have $\widehat{\in}_{\varpi} = \{\langle y, x\rangle, \langle y, v\rangle\}$, so that:

$$y^{R_{\varpi}} = z^{R_{\varpi}} = \emptyset \quad \text{and} \quad x^{R_{\varpi}} = v^{R_{\varpi}} = \{\emptyset\}.$$

It can easily be checked that R_{ϖ} already satisfies all literals on the branch ϖ but literal 4. This hints at the fact that some tableau rule must be applied to literal 4. The only possible rule is (11). After its application, and subsequent saturation with respect to the other rules, we end up with a closed tableau and the procedure stops, proving the unsatisfiability of the initial set of **MLSS** literals.

It may also happen that a partial realization R_{ϖ} satisfies its branch ϖ, well before the tableau is saturated. In such lucky cases, there is no need to bring the saturation process to completion, since one already knows that the initial formula is satisfiable (and, in fact, R_{ϖ} is a model of it). In fact, the interleaving of model checking and deduction steps can lead to relevant practical speed-up of the decision test, as argued in [4, 14].

4 Extensions of Multi-level Syllogistic

The satisfiability algorithm for **MLSS** presented in the preceding section can be extended in several ways by allowing otherwise uninterpreted function symbols, constants and predicates, subject to certain implicit universally quantified statements (side conditions), to be intermixed with the other operators of **MLSS**. Note however that the statements decided by the method to be described are not explicitly unquantified.

The pairing operator cons and the two associated component extraction operators car and cdr exemplify the operator families to which our extension technique is applicable. Assume that an 'arbitrary selection' operator arb is available which satisfies the condition

$$\big[\forall x \mid \mathsf{arb}(x) \in x \cup \{x\} \,\&\, \mathsf{arb}(x) \cap x = \emptyset\big],$$

that is, $\mathsf{arb}(x)$ returns an element of x which—as a set—does not have elements in common with x, *except* when $x = \emptyset$, in which case $\mathsf{arb}(x) = \emptyset$.[4] Then cons, car, and cdr could be characterized by the following formal set-theoretic definitions:

$$\begin{aligned}\mathsf{cons}(x, y) &=_{\mathrm{Def}} \big\{\big\{\{x\}, \big\{\{y\}, y\big\}\big\}, \{x\}\big\},\\ \mathsf{car}(p) &=_{\mathrm{Def}} \mathsf{arb}\big(\mathsf{arb}(p)\big),\\ \mathsf{cdr}(p) &=_{\mathrm{Def}} \mathsf{arb}\big(\mathsf{arb}\big(\mathsf{arb}\big(p \setminus \big\{\mathsf{arb}(p)\big\}\big) \setminus \big\{\mathsf{arb}(p)\big\}\big)\big).\end{aligned}$$

However, in most settings, the details of these definitions are irrelevant. Only the following properties of these operators matter:

- the object $\mathsf{cons}(x, y)$ can be formed for any two sets x, y;
- both of the sets x, y from which $\mathsf{cons}(x, y)$ is formed can be recovered uniquely from the single object $\mathsf{cons}(x, y)$, since
  - $\mathsf{car}(\mathsf{cons}(x, y)) = x$, and
  - $\mathsf{cdr}(\mathsf{cons}(x, y)) = y$.

Almost all proofs in which the operators cons, car, and cdr appear use only these facts about this triple of operators. That is, they implicitly treat these operators as a family of three otherwise uninterpreted operators, subject only to the conditions

$$\big[\forall\, x, y \,\big|\, \mathsf{car}\big(\mathsf{cons}(x, y)\big) = x\big] \,\&\, \big[\forall\, x, y \,\big|\, \mathsf{cdr}\big(\mathsf{cons}(x, y)\big) = y\big].$$

The treatment indicated throws away information about these operators (e.g. it hides that $\mathsf{car}(x)$ is always a member of a member of x) that may become relevant only in quite unusual situations.

Even the more fundamental issue of extending **MLSS** with the arb operator can be tackled in this frame of mind. It was shown in [17] how to constrain the semantics of arb so strongly as to take a commitment concerning the truth value of the existential closure of *any* conjunction φ of **MLSS** extended with arb. On the other hand, it is more in the spirit of this paper to remain neutral concerning infinitely many such statements which involve arb: by leaving, e.g., the value of $[\exists x, y \mid \mathsf{arb}(\{\{x\}, \{x, y\}\}) \neq \{x\}]$ undefined.

Similar remarks apply to other important families of operators. We list some of these, along with their associated universally quantified statements:

(i) *monotone functions*:

$$\big[\forall x, y \,\big|\, x \supseteq y \Rightarrow f(x) \supseteq f(y)\big];$$

(ii) *monotone functions having a known order relationship*:

$$\begin{aligned}&\big[\forall x, y \,\big|\, x \supseteq y \Rightarrow f(x) \supseteq f(y)\big] \,\&\, \big[\forall x, y \,\big|\, x \supseteq y \Rightarrow g(x) \supseteq g(y)\big]\\ &\quad\&\, \big[\forall x \,\big|\, f(x) \supseteq g(x)\big];\end{aligned}$$

[4] Note, in passing, that a choice set for any family $\mathcal{F}$ of disjoint non-null sets can be formed as $\{\mathsf{arb}(x) : x \in \mathcal{F}\}$; hence, the assumed availability of arb, jointly with the replacement axiom of set theory, yields as a consequence the somewhat controversial *postulate of choice*.

(iii) *monotone functions of several variables*:

$$\big[\forall x, y, u, v \,\big|\, (x \supseteq y \,\&\, u \supseteq v) \Rightarrow f(x, u) \supseteq f(y, v)\big];$$

(iv) *additive functions*:

$$\big[\forall x, y \,\big|\, f(x \cup y) = f(x) \cup f(y)\big];$$

(v) arb *function*:

$$\big[\forall x \,\big|\, \big(x = \emptyset \,\&\, \mathsf{arb}(x) = \emptyset\big) \vee \big(\mathsf{arb}(x) \in x \,\&\, \mathsf{arb}(x) \cap x = \emptyset\big)\big];$$

(vi) *idempotent functions on a set* w:

$$\big[\forall x \in w \,\big|\, f(x) \in w \,\&\, f\big(f(x)\big) = f(x)\big];$$

(vii) *total ordering relationships on sets*:

$$\big[\forall x \in w, y \in w \,\big|\, R(x, y) \vee R(y, x)\big]$$
$$\&\ \big[\forall x \in w, y \in w, z \in w \,\big|\, R(x, y) \,\&\, R(y, z) \Rightarrow R(x, z)\big];$$

(viii) *multiple functions with known ranges* w_j *and domains* v_j:

$$\mathop{\&}_{j=1}^{k} \big[\forall x \in v_j \,\big|\, f_j(x) \in w_j\big];$$

(ix) *pairs of mutually inverse functions*:

$$\big[\forall x \in w \,\big|\, f(x) \in w \,\&\, g(x) \in w \,\&\, f\big(g(x)\big) = x \,\&\, g\big(f(x)\big) = x\big];$$

(x) *self-inverse functions*:

$$\big[\forall x \in w \,\big|\, f(x) \in w \,\&\, f\big(f(x)\big) = x\big];$$

(xi) car, cdr, *and* cons *functions*:

$$\big[\forall x, y \,\big|\, \mathsf{car}\big(\mathsf{cons}(x, y)\big) = x\big] \,\&\, \big[\forall\, x, y \,\big|\, \mathsf{cdr}\big(\mathsf{cons}(x, y)\big) = y\big].$$

These are all mathematically significant relationships, as the existence of names associated with them attests.

These cases can all be handled by a common method under the following conditions. Suppose that we are given an unquantified collection C of statements involving the operators of **MLSS** plus certain other function symbols f, g of various numbers of arguments. We can suppose that all occurrences of these additional symbols are in simple flat statements of forms like $y = f(x)$, $y = g(x, z)$, etc. From these initially given statements we must be able to draw a 'complete' collection S of consequences involving the variables which appear in them, along with some finite number of additional variables that it may be necessary to introduce. The translated formula, comprising S and some residue of the original C, will be entirely within the language of **MLSS**. 'Completeness' means that we can be sure that any model $\mathcal{M}$ of the translated formula can be extended to include the original function symbols f in such a way that their interpretation $f^{\mathcal{M}}$ actually satisfies the desired properties (monotonicity, etc.).

Call these added statements S the *extension conditions* for the given set of functions: e.g., in most of the cases listed above, S will comprise *single-valuedness conditions* $x = u \Rightarrow y = v$ for all pairs $y = f(x)$, $v = f(u)$ originally present in C. If we can find them, the extension conditions will encapsulate everything which the appearance of the functions in question tells us about the set variables which also appear. Hence satisfiability can be determined by replacing all the statements $y = f(x)$, $y = g(x, z)$ in our original collection by the extension conditions derived from them.

This gives us a systematic way of reducing various languages extending **MLSS** to pure **MLSS**. As we will see, this approach can be exploited, to some extent, with predicates too, thanks to the fact that certain properties of predicates can be rendered via representing functions.

Note that the 'extension conditions' technique can be applied even if the recipe for removing function or predicate occurrences by adding compensating extension clauses is not complete, as long as it is sound, i.e. all the clauses added do follow from known properties of the functions or predicates removed.

4.1 Monotone Functions

Consider the extension of **MLSS** with statements of the forms

$$y = f(x), \quad \mathit{increasing}(f), \quad \mathit{decreasing}(f) \tag{7}$$

involving uninterpreted function symbols. The extension conditions can be derived as follows. Let the function symbols known to designate monotone functions be f, g, etc. For each triple of statements $y = f(x)$, $v = f(u)$, *increasing*(f), originally present in our collection C, add the following two clauses

$$x \supseteq u \Rightarrow y \supseteq v, \qquad u \supseteq x \Rightarrow v \supseteq y. \tag{8}$$

Likewise, for each triple of statements $y = f(x)$, $v = f(u)$, *decreasing*(f), originally present in our collection C, add the following two clauses

$$x \subseteq u \Rightarrow y \supseteq v, \qquad u \subseteq x \Rightarrow v \supseteq y. \tag{9}$$

After having added clauses (8) and (9),[5] we drop from C all clauses of the types (7). The added clauses ensure that if a model $\mathcal{M}$ exists, the set of pairs $\langle x^{\mathcal{M}}, y^{\mathcal{M}} \rangle$,

[5]Note that these imply the single-valuedness conditions for all functions involved.

formed for all the x and y initially appearing in clauses $y = f(x)$ for a function symbol f, defines a function F which is monotone increasing (resp., decreasing) on its domain, provided that a clause *increasing*(f) (resp., *decreasing*(f)) was originally in C. This can be extended to a function F' defined everywhere by letting $F'(s)$ be the union (resp., intersection) of all the $F(t)$, extended over all the elements t of the domain of F for which $s \supseteq t$.[6] It is clear that the F' defined in this way is also monotone and extends F. This proves that the clauses (8) and (9) are adequate extension conditions for the extension of **MLSS** with monotone functions. Note that the required number of such clauses is roughly as large as the square of the number of clauses $y = f(x)$ originally present in C. Therefore, the satisfiability problem for the extension of **MLSS** with clauses of types (7) is still NP-complete.

To make this method of proof entirely clear we give an example. Suppose that we need to prove the following conjunction

$$f\big(f(x \cup y)\big) \supseteq f\big(f(x)\big) \;\&\; \mathit{increasing}(f). \tag{10}$$

By flattening the compound terms which appear in this statement, we get the collection

$$\begin{array}{llll} z = x \cup y, & u = f(z), & w = f(u), & u_1 = f(x), \\ v_1 = f(u_1), & \mathit{increasing}(f), & \neg(w \supseteq v_1), & \end{array}$$

which we must prove to be unsatisfiable. The four statements

$$u = f(z), \qquad w = f(u), \qquad u_1 = f(x), \qquad v_1 = f(u_1) \tag{11}$$

in this collection give rise to the 12 extension conditions

$$\begin{array}{lll} z \supseteq u \Rightarrow u \supseteq w, & z \supseteq x \Rightarrow u \supseteq u_1, & z \supseteq u_1 \Rightarrow u \supseteq v_1, \\ u \supseteq z \Rightarrow w \supseteq u, & x \supseteq z \Rightarrow u_1 \supseteq u, & u_1 \supseteq z \Rightarrow v_1 \supseteq u, \\ x \supseteq u \Rightarrow u_1 \supseteq w, & x \supseteq u_1 \Rightarrow u_1 \supseteq v_1, & u \supseteq u_1 \Rightarrow w \supseteq v_1, \\ u \supseteq x \Rightarrow w \supseteq u_1, & u_1 \supseteq x \Rightarrow v_1 \supseteq u_1, & u_1 \supseteq u \Rightarrow v_1 \supseteq w, \end{array}$$

which replace (11). It is now an easy matter to check that

$$z = x \cup y, \qquad z \supseteq x \Rightarrow u \supseteq u_1, \qquad u \supseteq u_1 \Rightarrow w \supseteq v_1, \qquad \neg(w \supseteq v_1)$$

is an unsatisfiable conjunction, proving the validity of (10).

4.2 Monotone Functions Having a Known Order Relationship

This case can be treated in much the same way as the somewhat simpler case in Sect. 4.1. For example, given monotone, say increasing, f, g, where it is known

---

[6] To avoid intersections of empty families of sets, it is convenient to assume that C contains a clause $x = f(\emptyset)$, for each function symbol f present in C.

that $f(x) \supseteq g(x)$ is universally true, first force the known part of their domains to be equal by introducing a u satisfying $g(x) = u$ for each initially given clause $y = f(x)$ and vice versa. Then proceed as in the preceding case, but now add inclusions $x = v \Rightarrow y \supseteq u$ for every pair $g(v) = u$, $f(x) = y$ of clauses originally present. It is clear that the extensions of g and f eventually satisfied stand in the proper ordering relationship.

The NP-completeness of the resulting decision problem can be easily established.

4.3 Monotone Functions of Several Variables

The case of monotone functions of several variables is also easy. We can proceed as follows. Given a function $f(x, y)$ which is to be monotone in both its variables, and also a set of clauses like $z = f(x, y)$, $w = f(u, v)$, introduce clauses

$$(x \supseteq u \,\&\, y \supseteq v) \Rightarrow (z \supseteq w).$$

Then plainly the set of pairs $\langle\langle x^{\mathcal{M}}, y^{\mathcal{M}}\rangle, z^{\mathcal{M}}\rangle$, formed for all the x, y, z initially appearing in clauses $z = f(x, y)$ defines a function F of two arguments which is monotone on its domain. This can be extended to a function F' defined everywhere by defining $F'(s, t)$ as the union of all the $F(p, q)$, extended over all the pairs p, q of the domain of F for which $s \supseteq p$ and $t \supseteq q$. Again, the NP-completeness of the related decision problem can be easily shown.

The reader is referred to [15, 31] for further elaborations.

4.4 Additive Functions

The related case of *additive functions* of a set variable can also be treated in the way which we will now explain (but the very many clauses which this technique introduces hints that 'additivity' is a significantly harder case than 'monotonicity'). A set-valued function f of sets is called 'additive' if $f(x \cup y) = f(x) \cup f(y)$ for all x and y.

Given an otherwise uninterpreted function f which is supposed to be additive, and clauses $y = f(x)$, introduce all the 'atomic parts' of all the variables x which appear in such clauses. These are variables, named u_j in what follows, representing all the intersections of some of the sets represented by variables occurring in the original clauses with the complements of the other similar sets. In terms of these intersections, which clearly are all disjoint, express each x in terms of its atomic parts u_j, namely as

$$x = u_{j_1} \cup \cdots \cup u_{j_k}.$$

Likewise, after introducing clauses

$$v_j = f(u_j)$$

giving names to the range elements $f(u_j)$, write out all the relationships

$$y = v_{j_1} \cup \cdots \cup v_{j_k}$$

that derive from clauses $y = f(x)$. Finally, writing $\emptyset$ and $f(\emptyset)$ for uniformity as u_0 and v_0, add implications

$$u_j = u_0 \Rightarrow v_j = v_0 \quad \text{and} \quad v_0 \subseteq v_j,$$

along with statements

$$u_j \cap u_i = \emptyset$$

(for $i \neq j$) which express the disjointness of distinct sets u_j.

Now suppose that the set of clauses we have written has a model $\mathcal{M}$ in which the u_j, v_j, x, y, etc. appearing above are represented by sets $(u_j)^{\mathcal{M}}$, $(v_j)^{\mathcal{M}}$, $x^{\mathcal{M}}$, $y^{\mathcal{M}}$, etc. and for each s, define the set-valued function $F(s)$ to be the union of all the sets $(v_j)^{\mathcal{M}}$ for which s intersects $(u_j)^{\mathcal{M}}$. The function F defined in this way is clearly additive. It is also clear that if a clause $y = f(x)$ is present in our initial collection, and the variables x and y are represented by sets $x^{\mathcal{M}}$ and $y^{\mathcal{M}}$, then $y^{\mathcal{M}} = F(x^{\mathcal{M}})$. Hence F can represent f in the model we have constructed; so f can be represented by an additive function, proving that the clauses we have added to our original collection are the appropriate extension conditions.

Observe that the above reduction generates **MLSS**-formulae of exponential size, yielding an exponential nondeterministic decision procedure. Though we have not proved it, we strongly believe that the decision problem for the extension of **MLSS** with additive functions is not NP-complete.

A more formal treatment of such an extension of **MLSS**, also with other constructs, can be found in [15, 31].

4.5 arb *Function*

To seek a model for a collection of **MLSS** clauses, plus statements of the form $w = \mathsf{arb}(z)$, we could proceed, in analogy with the extension of **MLSS** with monotone functions considered in Sect. 4.1, by firstly adding for each statement $y = \mathsf{arb}(x)$ the extension condition

$$(x = \emptyset \,\&\, y = \emptyset) \vee (y \in x \,\&\, y \cap x = \emptyset), \tag{12}$$

then adding, for each pair of statements $y = \mathsf{arb}(x)$ and $v = \mathsf{arb}(u)$ originally given, the corresponding extension condition

$$x = u \Rightarrow y = v \tag{13}$$

(namely the single-valued functional dependence condition), and finally dropping all clauses of the form $y = \mathsf{arb}(x)$ from the original collection of statements.

Suppose now that $\mathcal{M}$ models the resulting collection of **MLSS** clauses. Then plainly the set of pairs $\langle x^{\mathcal{M}}, y^{\mathcal{M}}\rangle$, formed for all the x and y appearing in the statements $y = \mathsf{arb}(x)$ originally present, defines a single-valued function A on its finite domain which satisfies

$$\big(s = \emptyset \,\&\, A(s) = \emptyset\big) \vee \big(A(s) \in s \,\&\, A(s) \cap s = \emptyset\big),$$

for all the elements of its domain. We can extend this to a function A' defined everywhere by writing

$$A'(s) = \textbf{if } s \text{ in } \mathsf{domain}(A) \textbf{ then } A(s) \textbf{ else } \mathsf{arb}(s) \textbf{ end if},$$

where arb is the built-in choice operator of our version of set theory. A' then satisfies the originally universally quantified condition for arb, verifying that the clauses (12) and (13) are the proper extension conditions.

Again, it is immediate to check that the decision problem for the extension of **MLSS** with the arb function is NP-complete.

4.6 Idempotent Functions on a Set

The case of idempotent functions is also easy. We can proceed as before, but adding a clause $y = f(y)$ whenever a clause $y = f(x)$ is present. Then we add the 'single-valuedness' implications $u = x \Rightarrow z = y$ whenever two clauses $y = f(x)$, $z = f(u)$ are present, and remove all the clauses $y = f(x)$. The added clauses ensure that if a model $\mathcal{M}$ exists, the mapping F which sends $x^{\mathcal{M}}$ to $y^{\mathcal{M}}$ for each clause $y = f(x)$ initially present is single-valued; moreover, thanks to the added clauses $y = f(y)$, this mapping is clearly idempotent where defined. It can be extended by mapping all elements not in the domain of F to any selected element of the range of F.

The 'self-inverse' function case

$$\big[\forall x \in w \,\big|\, f(x) \in w \,\&\, f\big(f(x)\big) = x\big]$$

can be handled in much the same way, but we omit the details since this can also be viewed as a special subcase of the extension of **MLSS** with pairs of mutually inverse functions which we will discuss later in Sect. 4.9.

Plainly, **MLSS** with idempotent functions on a set is still NP-complete.

4.7 Total Ordering Relationships on Sets

The case of total ordering relationships on sets can be handled in the following way, which derives from the immediately preceding remarks. Let R be such a relationship. Introduce the representing function f for it, i.e. a function such that

$f(y) = \{x \mid R(x, y)\}$, so that $f(x) \supseteq f(y) \Leftrightarrow R(x, y)$ holds. Then R is a total ordering if and only if the range elements $f(x)$ all belong to a collection of sets totally ordered by inclusion. So write a clause $y \supseteq v \vee v \supseteq y$ for each pair of clauses $y = f(x)$, $v = f(u)$, and also write the conditions needed to ensure that f is single-valued. In the resulting model f plainly maps its domain into a collection of sets totally ordered by inclusion, and then f can be extended to all other sets by sending them to $\emptyset$.

The above remarks readily yield that the decision problem for **MLSS** with total ordering relationships on sets is still NP-complete.

4.8 Multiple Functions with Known Ranges and Domains

The present case is also very easy. For clarity, we will consider the special subcase in which two functions f, g are given, along with two domain sets d_1, d_2, and two range sets r_1, r_2. The universally quantified conditions which must be satisfied are then

$$\left[\forall x \in d_1 \,\middle|\, f(x) \in r_1\right], \tag{14}$$

$$\left[\forall x \in d_2 \,\middle|\, g(x) \in r_2\right], \tag{15}$$

along with some collection of unquantified clauses of **MLSS**.

We proceed as follows. For any two clauses $y = f(x)$, $y' = f(x')$ and any two clauses $y = g(x)$, $y' = g(x')$ present in our set C of statements write a 'single-valuedness' condition

$$x = x' \Rightarrow y' = y'. \tag{16}$$

For any clause $y = f(x)$ in S, write a condition

$$x \in d_1 \Rightarrow y \in r_1, \tag{17}$$

and, similarly, for any clause $y = g(x)$ in S, write a condition

$$x \in d_2 \Rightarrow y \in r_2. \tag{18}$$

Finally, write the conditions

$$d_1 \neq \emptyset \Rightarrow r_1 \neq \emptyset, \qquad d_2 \neq \emptyset \Rightarrow r_2 \neq \emptyset, \tag{19}$$

and drop from C all clauses involving the functions f, g.

Plainly, if our original set of clauses is consistent, so is the resulting set S of clauses. Conversely, suppose that the clauses S have a model $\mathcal{M}$. Define a preliminary function F (resp. G) as the set of all pairs $\langle x^{\mathcal{M}}, y^{\mathcal{M}} \rangle$ for which a clause $y = f(x)$ (resp. $y = g(x)$) is present in S. The clauses (16) plainly imply that F and G are single-valued on their domain, and the clauses (17) ensure that F maps the intersection of its domain with ${d_1}^{\mathcal{M}}$ into ${r_1}^{\mathcal{M}}$. Likewise, the clauses (18) ensure that G maps the intersection of its domain with ${d_2}^{\mathcal{M}}$ into ${r_2}^{\mathcal{M}}$. If ${d_1}^{\mathcal{M}} = \emptyset$, the quantified condition (14) is automatically satisfied. On the other hand, if ${d_1}^{\mathcal{M}} \neq \emptyset$, the clause (19) ensures that ${r_1}^{\mathcal{M}} \neq \emptyset$, so we can extend F to map all elements of ${d_1}^{\mathcal{M}}$ not in its initial domain to any element of ${r_1}^{\mathcal{M}}$ we choose. Repeating this construction for g, d_2, and r_2 plainly gives us a model of all our clauses in which f and g are represented by single-valued functions satisfying (14) and (15). Hence the clauses (16), (17), (18), and (19) we have added are the extension conditions we require.

Observe that the number of clauses added as extension conditions is at most quadratic in the size of the initial set C of statements.[7] Therefore we have immediately the NP-completeness of the decision problem for the fragment at hand.

4.9 Pairs of Mutually Inverse Functions f, g

Extension conditions for the present case of mutually inverse functions f and g can be formulated as follows. Write the clauses, described above, that force f and g to be single-valued, namely

$$x = u \Rightarrow y = v, \tag{20}$$

for all pairs $y = f(x)$, $v = f(u)$ and all pairs $y = g(x)$, $v = g(u)$ originally present in the given set of clauses. To these, add clauses

$$y = v \Rightarrow x = u \tag{21}$$

derived from all the given statements $y = f(x)$, $v = f(u)$. These force f to be 1-1 on the collection of elements x known to be in its domain. (Note that this much also handles the case of functions known to be 1-1.) Do the same thing for g. Then add clauses

$$y = u \Leftrightarrow x = v \tag{22}$$

derived from all the statement pairs $y = f(x)$, $v = g(u)$ to take care of fact that f and g must be mutually inverse functions. Then, in the resulting model $\mathcal{M}$, the model functions F and G of f and g must both be 1-1 on their domains (e.g. for F this is the collection of sets $x^{\mathcal{M}}$ modeling points x for which some clause $y = f(x)$ appears in our original set of statements), and G must be the inverse of F on $\mathsf{domain}(G) \cap \mathsf{range}(F)$. Since G is 1-1 on its domain, it follows that the range of G on $\mathsf{domain}(G) \setminus \mathsf{range}(F)$ must be disjoint from $\mathsf{domain}(F)$. Indeed, if a set s is in $\mathsf{domain}(F) \cap \mathsf{range}(G)$ it must have the form $s = x^{\mathcal{M}}$ where clauses $y = f(x)$ and $x = g(u)$ both appear in our original set of statements. But then $u^{\mathcal{M}} =$

[7] In fact, introduction of 'single-valuedness' conditions (16) can easily be constrained so as only a linear number of such clauses need to be added.

$y^{\mathcal{M}}$ is implied by one of the clauses of type (22), and hence $u^{\mathcal{M}}$ is in the range of F. Similarly the range of F on $\mathsf{domain}(F) \setminus \mathsf{range}(G)$ must be disjoint from the $\mathsf{domain}(G)$. Therefore, F can be extended to

$$\mathsf{range}(G|_{\mathsf{domain}(G)\setminus\mathsf{range}(F)}) \quad \text{(the range on the restriction)}$$

as the inverse of G, and similarly G extended to

$$\mathsf{range}(F|_{\mathsf{domain}(F)\setminus\mathsf{range}(G)})$$

as the inverse of F. Let F' and G' be these extensions. Then plainly $\mathsf{domain}(F') = \mathsf{domain}(F) \cup \mathsf{range}(G)$, and so $\mathsf{range}(G') = \mathsf{range}(G) \cup \mathsf{domain}(F) = \mathsf{domain}(F')$ and vice versa. Hence the extensions F' and G' are mutually inverse with $\mathsf{domain}(F') = \mathsf{range}(G')$ and $\mathsf{domain}(G') = \mathsf{range}(F')$. F' and G' can now be extended to mutually inverse maps defined everywhere by using any 1-1 map of the complement of $\mathsf{domain}(F')$ onto the complement of $\mathsf{range}(F')$. This shows that the clauses listed above are the correct extension conditions for the extension of **MLSS** with pairs of mutually inverse functions.

Again, it is an easy matter to check that the number of extension conditions of the forms (20), (21), and (22) is at most quadratic in the size of the initial collection C of statements, so that even the present extension of **MLSS** turns out to have an NP-complete satisfiability problem.

4.10 Self-inverse Functions

The case of self-inverse functions is a special case of the preceding case of pairs of mutually inverse functions f, g.

4.11 car, cdr, *and* cons *Functions*

The extension conditions for the important car, cdr, and cons case can be worked out in somewhat similar fashion as follows. Regard $\mathsf{cons}(x, y)$ as a family of one parameter functions $\mathsf{cons}_x(y)$ dependent on the subsidiary parameter x. The ranges of all the functions cons_x in the family are disjoint (since $\mathsf{cons}(x, y)$ can never equal $\mathsf{cons}(u, v)$ if $x \neq u$). For the same reason, each cons_x is 1-1, and cdr is its (left) inverse, i.e. $\mathsf{cdr}(\mathsf{cons}_x(y)) = y$. Also, $\mathsf{car}(\mathsf{cons}_x(y)) = x$ everywhere. The required extension conditions are the following. For each pair of initial clauses $z = \mathsf{cons}(x, y)$, $w = \mathsf{cons}(u, v)$, add the clauses

$$(x \neq u \vee y \neq v) \Rightarrow z \neq w \quad \text{and} \quad (x = u \,\&\, y = v) \Rightarrow z = w \tag{23}$$

to force cons to be 'doubly 1-1' and well defined. Also, for each pair of initial clauses $z = \mathsf{cons}(x, y)$, $v = \mathsf{car}(u)$, add the clause

$$u = z \Rightarrow x = v \tag{24}$$

to force car to stand in the proper inverse relationship to cons. And, likewise, for each pair of initial clauses $z = \mathsf{cons}(x, y)$, $v = \mathsf{cdr}(u)$, add the clause

$$u = z \Rightarrow y = v \tag{25}$$

to force cdr to stand in the proper inverse relationship to cons.

The number of extension conditions (23), (24), and (25) added for the case at hand is easily seen to be at most quadratic in the size of the initial set of clauses, showing that also the extension of **MLSS** with the operators car, cdr, and cons has an NP-complete decision problem.

4.12 Other Cases

We note a few more cases, whose details we omit, which can be handled by the 'extension conditions' technique. Uninterpreted commutative arguments of two variables, having the property

$$\big[\forall x, y \,\big|\, f(x, y) = f(y, x)\big],$$

can readily be handled by methods like those shown above. It might be possible to treat associativity also, possibly based on a prior **MLSS**-like theory of the concatenation operator. A further known case (cf. [2] and [6, Chap. 7]) is that in which we allow the special constant symbols $\mathbb{N}$ and $\mathcal{O}rd$ for the set of non-negative integers and for the class of all ordinals, respectively.

4.13 Predicates Representable by Functions

Predicates representable by functions in one of the above classes can be removed automatically by first replacing them by the functions that represent them, and then removing these functions by writing the appropriate extension conditions. For example, equivalence relationships $R(x, y)$ can be written using a representing function f as $f(x) = f(y)$; f only needs to be single-valued. Any partial ordering relationship $R(x, y)$ can be written as $f(x) \supseteq f(y)$, where f only needs to be single-valued, and f is monotone if and only if the ordering relationship $R(x, y)$ is compatible with inclusion in the sense that

$$\big[\forall x,\ y \,\big|\, x \supseteq y \Rightarrow R(x, y)\big].$$

For the purpose of decidability, monadic predicates $P(x)$ satisfying the condition

$$\big[\forall x, y \,\big|\, P(x) \,\&\, P(y) \Rightarrow P(x \cup y)\big] \,\&\, \big[\forall x, y \,\big|\, P(x) \,\&\, x \supseteq y \Rightarrow P(y)\big]$$

can be written in the form

$$P(x) \;\Leftrightarrow_{\mathrm{Def}}\; p \supseteq f(x),$$

where f is additive and p is a set large enough to include as subset the set $f(x)$, for each variable x such that $P(x)$ is present in the formula. The predicates $\mathit{Finite}(x)$ (which states that x is finite), $\mathit{Countable}(x)$ (which states that x is countable, i.e. either finite or denumerable), and $\mathit{Is\_map}(x)$, where

$$\mathit{Is\_map}(s) \;\Leftrightarrow_{\mathrm{Def}}\; \big[\forall x \in s \,\big|\, \big[\exists u, v \,\big|\, x = \mathsf{cons}(u, v)\big]\big],$$

illustrate this last remark. However, though sound, in general the extension conditions derived just from the additivity of the representing function f could be incomplete and therefore might need to be integrated with additional conditions.

We illustrate the latter point by studying the generalization of **MLSS** by the two additional set predicates $\mathit{Finite}(x)$ and $\mathit{Countable}(x)$ just discussed. Thus, let C be a collection of statements which in addition to literals of the form (1) contains also literals of the following form

$$\mathit{Finite}(x), \quad \mathit{Countable}(x), \quad \neg\mathit{Finite}(x), \quad \neg\mathit{Countable}(x).$$

We introduce two new variables Fi and Co, and for these variables add to C the following statements:

- $\mathsf{Co} \supseteq \mathsf{Fi}$;
- $\mathsf{Fi} \supseteq x$, for each x for which a statement $x = \{y\}$ or a statement $\mathit{Finite}(x)$ is present in C;
- $\mathsf{Co} \supseteq x$, for each x for which a statement $\mathit{Countable}(x)$ is present in C;
- $\neg(\mathsf{Fi} \supseteq x)$, for each x for which a statement $\neg\mathit{Finite}(x)$ is present in C;
- $\neg(\mathsf{Co} \supseteq x)$, for each x for which a statement $\neg\mathit{Countable}(x)$ is present in C.[8]

Let C' be the resulting collection of statements after all statements of the form $\mathit{Finite}(x)$, $\mathit{Countable}(x)$, $\neg\mathit{Finite}(x)$, and $\neg\mathit{Countable}(x)$ have been dropped.

It is plain from what was said above that if our original collection C of statements has a model, so does our modified collection C'. Conversely, if C' has a model, then, as shown in Theorem 1, there must exist an equivalence relation $\sim$ such that conditions (i)–(iv) of the same theorem hold for the collection C'' of statements obtained when each occurrence of any variable in C' is replaced by the selected representative in its $\sim$-equivalence class. In particular, let Π be the ample set of places for C'', whose existence is stated in condition (ii) of Theorem 1. To the places π in Π we assign disjoint sets $\overline{\pi}$ according to the following rule:

[8] Thus, in this case, the function f is just the identity.

- if π is of the form π^x for some variable x appearing in a statement $y = \{x\}$, let $\overline{\pi}$ be null;
- otherwise, if $\pi(\mathsf{Fi}) = \textbf{true}$, then let $\overline{\pi}$ be some single element;
- otherwise, if $\pi(\mathsf{Co}) = \textbf{true}$, then let $\overline{\pi}$ be some countably infinite set;
- otherwise, let $\overline{\pi}$ be some uncountable set.

We also suppose, much as in the proof of Theorem 1, that each member of $\overline{\pi}$ has larger cardinality than the largest cardinality of any set $\overline{\pi}$ (here we are assuming that some set $\overline{\pi}$ is infinite), and then use (4) to define a model $\mathcal{M}$. The analysis given in the proof of Theorem 1 shows that this $\mathcal{M}$ correctly models all statements not involving the predicates '*Finite*' and '*Countable*'. It is plain that $\mathsf{Fi}^{\mathcal{M}}$ is finite and $\mathsf{Co}^{\mathcal{M}}$ is countable; hence all statements $Finite(x)$ and $Countable(x)$ originally present are correctly modeled also.

If any statement $\neg Finite(x)$ is present in C, then there exists a place π in Π such that $\pi(x)$ is true and $\pi(\mathsf{Fi})$ is false.[9] The place π cannot have the form π^z for any variable x appearing in any statement $y = \{z\}$ in C'', since if it did then the facts that $\pi^z(y)$ must be true and the statement $\mathsf{Fi} \supseteq y$ has been added to C'' would imply that $\pi^z(\mathsf{Fi})$ is true. Hence $\overline{\pi}$ is infinite and so, by (4), $x^{\mathcal{M}}$ is infinite also. This shows that all statements $\neg Finite(x)$ are correctly modeled. The case of statements $\neg Countable(x)$ can be handled in much the same way, showing that our original and modified sets of statements are equisatisfiable.

Again, it is an easy matter to observe that the number of extension conditions is linear in the size of the initial collection C of statements, and therefore the extension of **MLSS** with the set predicates '*Finite*' and '*Countable*' has an NP-complete decision problem.

5 Further Extensions of Multi-level Syllogistic

The extension technique discussed at length in the preceding section is not able to deal with constructs which allow one to establish connections among the disjoint regions of the Venn diagram of a model that are hardly describable by cardinality constraints. This is the case, for instance, of the general union operator $\bigcup(\cdot)$ and the powerset operator $\mathscr{P}(\cdot)$. Consider for example a formula C involving a clause $x = \mathscr{P}(y)$. Then in any model $\mathcal{M}$ of C, one must have that any member of any disjoint Venn region of $x^{\mathcal{M}}$ is a subset of the union of all disjoint Venn regions of $y^{\mathcal{M}}$, and conversely. Plainly, such relationship can not be described by cardinality constraints only.

As exemplified in the sufficiency part of the proofs of Theorem 1 and of the correctness of the extension conditions for the **MLSS** extension with the predicates $Finite(x)$ and $Countable(x)$ in Sect. 4.13, a common approach was to find effectively verifiable conditions on the places and variables of a given formula C which

[9]For simplicity, we are assuming that the $\sim$-representative of the variable x in $\neg Finite(x)$ is x itself.

when satisfied enable one to construct a model $\mathcal{M}$ of C by executing an instantiation procedure of the following form:

$$\text{Initialization} \left\{ \begin{array}{l} \text{For every place } \pi \in \Pi \text{ let} \\ \quad \overline{\pi} := \overline{\pi}^{(0)} \end{array} \right.$$

$$\in\text{-phase} \left\{ \begin{array}{l} \text{Following an admissible ordering} < \text{of the variables } y \text{ in } C \text{ do} \\ \quad y^{\mathcal{M}} := \{x^{\mathcal{M}} : x \in L \mid \pi^x(y)\} \cup \bigcup\{\overline{\pi} : \pi \in \Pi \mid \pi(y)\}, \end{array} \right.$$

where Π is a collection of places for C, L is a suitable subset of the collection of the variables occurring in C, and the $\overline{\pi}^{(0)}$'s are suitable sets assigned to the places for C. In connection with it (and with its extension below), it is convenient to introduce the following useful notion of *approximant assignment* $\mathcal{M}_{|i}$. Let $x_1 < x_2 < \cdots < x_m$ be the admissible ordering used in the $\in$-phase. Then, we put

$$y^{\mathcal{M}_{|i}} =_{\text{Def}} \left\{x^{\mathcal{M}} : x \in L \,\middle|\, \pi^x(y) \,\&\, x < x_i\right\} \cup \bigcup\left\{\overline{\pi} : \pi \in \Pi \,\middle|\, \pi(y)\right\}, \tag{26}$$

for each variable y in C and $1 \leqslant i \leqslant m$. Thus, for instance, for the first approximant we have $y^{\mathcal{M}_{|1}} = \bigcup\{\overline{\pi} : \pi \in \Pi \mid \pi(y)\}$. Additionally, $y^{\mathcal{M}} = y^{\mathcal{M}_{|i}}$, provided that $y \leqslant x_i$, or, more generally, provided that $\pi^x(y)$ is true for no x such that $x_i \leqslant x$.

The rationale behind the above instantiation procedure is that at the end of the initialization phase all equalities and cardinality predicates in C are satisfied by the initial approximant assignment $\mathcal{M}_{|1}$ and that membership clauses are made true by the subsequent $\in$-phase (obviously, without disrupting satisfaction of the equality and membership clauses already established).

To take into consideration also relationships among places such as the ones hinted to above that are forced by operators like $\bigcup$ and $\mathscr{P}$, a suitable 'stabilization' phase follows the initialization phase, so that equality clauses are satisfied (at least as inclusions) and, likewise, a stabilization step follows each single step in the $\in$-phase, yielding an instantiation procedure of the more general type:

$$\text{Initialization} \left\{ \begin{array}{l} \text{For every place } \pi \in \Pi \text{ let} \\ \quad \overline{\pi} := \overline{\pi}^{(0)} \\ \text{Stabilize} \end{array} \right.$$

$$\in\text{-phase} \left\{ \begin{array}{l} \text{Following an admissible ordering} < \text{of the variables } y \text{ in } C \text{ do} \\ \quad y^{\mathcal{M}} := \{x^{\mathcal{M}} : x \in L \mid \pi^x(y)\} \cup \bigcup\{\overline{\pi} : \pi \in \Pi \mid \pi(y)\} \\ \quad \text{Stabilize} \end{array} \right.$$

In the case of the instantiation procedure with stabilization, (26) becomes

$$y^{\mathcal{M}_{|i}} =_{\text{Def}} \left\{x^{\mathcal{M}} : x \in L \,\middle|\, \pi^x(y) \,\&\, x < x_i\right\} \cup \bigcup\left\{\overline{\pi}^{(i)} : \pi \in \Pi \,\middle|\, \pi(y)\right\}, \tag{27}$$

where $\overline{\pi}^{(i)}$ is the value of the variable $\overline{\pi}$ in the instantiation procedure after the ith stabilization step in the $\in$-phase. It is understood that stabilization is monotone, namely $\overline{\pi}^{(0)} \subseteq \overline{\pi}^{(1)} \subseteq \cdots \subseteq \overline{\pi}^{(m)}$ hold for each place π in Π.

We illustrate such a technique with two extensions of **MLS**, with the general union operator $\bigcup$ and with the powerset operator $\mathscr{P}$.

5.1 MLS with the General Union Operator

We first consider the unquantified theory **MLSU** obtained from **MLS** by also allowing an unrestricted number of occurrences of the general union operator $\bigcup$, where we recall that $\bigcup S = \bigcup_{s \in S} s$.

By way of a decomposition at the propositional level and a secondary decomposition of the kind shown in Sect. 3, we can limit ourselves to considering only formulae C of **MLSU** which are conjunctions of **MLS**-literals having one of the flat forms

$$\begin{array}{lll} x = y \cup z, & x = y \cap z, & x = y \setminus z, \\ x \neq y, & x \in y, & x \notin y, \end{array} \tag{28}$$

plus atoms of the following type

$$u = \bigcup y. \tag{29}$$

Thus, let C be a conjunction of **MLSU**-literals of the forms (28) and (29) and let $\mathcal{M}$ be a model of C. Let Σ, Π, etc., be as in Sect. 3 and let $<$ be an ordering of the variables in C such that if $x^{\mathcal{M}} \in y^{\mathcal{M}}$ then $x < y$. The heuristic technique used in order to derive conditions which are both necessary and sufficient for C to be satisfiable resembles that used in the **MLSS** case.

A first necessary condition is suggested by the following observation. Let x be a left-hand variable such that $\pi^x(y)$ is true, for a variable y occurring in a clause $u = \bigcup y$ in C. Then $x^{\mathcal{M}} \in \sigma^x \subseteq y^{\mathcal{M}}$, where π^x is the place in Π corresponding to the Venn region $\sigma_x \in \Sigma$. But since $u^{\mathcal{M}} = \bigcup y^{\mathcal{M}}$, it follows that $x^{\mathcal{M}} \subseteq u^{\mathcal{M}}$, so that for each Venn region $\sigma \in \Sigma$ such that $\sigma \subseteq x^{\mathcal{M}}$, we have also $\sigma \subseteq u^{\mathcal{M}}$. Therefore $\pi(u)$ is true, for each $\pi \in \Pi$ such that $\pi(x)$ is true. Thus, a first obviously necessary condition is:

(C1) if $\pi^x(y)$ is true for variables x, y in C such that C contains a clause of type $u = \bigcup y$, then $\pi(u)$ is true whenever $\pi(x)$ is true, for every place $\pi \in \Pi$.

The remaining conditions necessary and sufficient for satisfiability can be derived by closer analysis of the initialization- and $\in$-phases. A first problem is how to find values $\overline{\pi}^{(0)}$ such that the initial approximant assignment

$$x^{\mathcal{M}_{|1}} = \bigcup \left\{ \overline{\pi}^{(0)} : \pi \in \Pi \,\middle|\, \pi(x) \right\}$$

satisfies all conjuncts in C which do not involve the membership relation $\in$. For this, we begin with a set of places in which individuals can be put without exercising any special care (except that certain rank restrictions must be satisfied). Then we proliferate individuals in such a way as to initialize the sets $\overline{\pi}$ appropriately. It turns out that all inclusion relationships

$$\bigcup\Big(\bigcup\{\overline{\pi} : \pi \in \Pi \,\big|\, \pi(y)\}\Big) \subseteq \bigcup\{\overline{\pi} : \pi \in \Pi \,\big|\, \pi(u)\},$$

where $u = \bigcup y$ occurs in C, can be forced by the first part of the initialization phase (*proper initialization*). A subsequent *stabilization subphase* then turns these inclusions into equalities. During the $\in$-phase, the insertion of a set $z^{\mathcal{M}}$ into a $y^{\mathcal{M}}$ (precisely in the $(i+1)$st step of the $\in$-phase, where z is the ith variable in the given admissible ordering) can disrupt an already established inclusion of the form $u^{\mathcal{M}|_i} \subseteq \bigcup y^{\mathcal{M}|_i}$, where the clause $u = \bigcup y$ occurs in C. This may happen if $\pi^z(u)$ is true and $z^{\mathcal{M}|_i} \not\subseteq \bigcup y^{\mathcal{M}|_i}$ (the reverse of such inclusions are maintained because of condition (C1) above). To get around this problem it is enough that each step in the $\in$-phase is followed by a suitable stabilization step which enlarges the sets $\overline{\pi}$'s in such a way that $z^{\mathcal{M}|_i} \subseteq \bigcup\bigcup\{\overline{\pi}^{(i+1)} : \pi \in \Pi \mid \pi(y)\}$ holds, without disrupting other already established equalities or inclusions.

In order that the various stabilization steps that we have just sketched function properly, some relationship among places is required. This suggests the following definition, which associates a graph to the conjunction C and its set Π of places.

Definition 1 Given a conjunction C and a set Π of places of C as above, the *Ugraph* G of C, Π is the graph whose set of nodes is Π, plus one additional node Ω, and whose edges are as follows:

(i) a directed edge connects π to Ω if and only if $\pi(y)$ is false for every variable y for which $u = \bigcup y$ is in C (intuitively this means that clauses $u = \bigcup y$ of C tell us nothing about the set $\bigcup\overline{\pi}$, which allows the proper initialization phase to start with such places);

(ii) otherwise, a directed edge connects the place α to the place β if and only if $\beta(u)$ is true for all clauses $u = \bigcup y$ such that $\alpha(y)$ is true (intuitively, the nodes β such that $\alpha \to \beta$ is an edge of G represent all the sets $\overline{\beta}$ in which elements of $\bigcup\overline{\pi}$ can appear. Indeed, let $y_{i_1}, \ldots, y_{i_k}$ be all variables y such that $u = \bigcup y$ is in C and $\alpha(y)$ is true, so that $\overline{\alpha} \subseteq y_{i_1}^{\mathcal{M}} \cup \cdots \cup y_{i_k}^{\mathcal{M}}$. It follows that $\bigcup\overline{\alpha} \subseteq u_{i_1}^{\mathcal{M}} \cup \cdots \cup u_{i_k}^{\mathcal{M}}$, i.e. $\bigcup\overline{\alpha}$ is contained in the union of all sets $\overline{\beta}$ such that $\alpha \to \beta$ is an edge of G.)

In the graph just introduced we can distinguish three kinds of nodes. Those from which there is a directed path which reaches Ω are called *safe*. A node is called *trapped* if every sufficiently long path from it eventually reaches a node from which no edge branches off (*null* node $\pi^{\emptyset}$). Finally, a node is *cyclic* if it is neither safe nor trapped. Intuitively speaking, trapped places are those places whose elements are subject to severe restrictions. Consider for example the following formula:

$$\bigcup x = \emptyset \;\&\; \bigcup y = x \;\&\; \bigcup z = y \;\&\; x = \emptyset.$$

It is easy to see that in any set of places of the formula above, all places are trapped. It is also evident that variables x, y, z can assume just a few values; this 'semantic' fact reflects the 'syntactic' fact that we have just noted. Trapped places are dealt with by observing that such places can be assigned only sets having rank at most one more than the maximum length of a longest path forward from each of them to a null node. Therefore only a finite number of possible choices must be checked in order to determine the value $\overline{\pi}$ to associate with such a π. On the other hand it turns out that each nontrapped place π can be assigned an infinite $\overline{\pi}$. This fact simplifies greatly the initialization phase, since the 'individuals' which we initially place in $\overline{\pi}$ easily propagate along the Ugraph via singletons or pairs.

A rough description of the first initialization phase is as follows (for simplicity we only consider the case in which no trapped place exists). First of all infinitely many individuals are associated with every place π of C such that $\pi \to \Omega$ is an edge of the graph G. Then any safe place can iteratively be given an infinite supply of elements by drawing elements from its descendants and forming their singletons. The same technique can also be used to initialize cyclic places, once we observe that the null node $\pi^{\emptyset}$ must lie on a cycle which can be given elements by successive formation of singletons of the empty set $\emptyset$ (which is assigned to $\overline{\pi}^{\emptyset}$) and that the null node must be reachable along edges of G from every other node, by the regularity axiom of set theory. This observation, which in substance gives us a second condition for the satisfiability of C, guarantees that proper initialization can be accomplished successfully. Once this phase is completed, all equality clauses of C are correctly modeled; however for literals $u = \bigcup y$ in C all we can say is that $\bigcup(\bigcup\{\overline{\pi} : \pi \in \Pi \mid \pi(y)\}) \subseteq \bigcup\{\overline{\pi} : \pi \in \Pi \mid \pi(u)\}$. To get equalities in place of these inclusions, the following stabilization phase is then performed. For each element s which has been put into $\overline{\pi}$ and for every clause $u = \bigcup y$ such that $\pi(u)$ is true (i.e., intuitively, $\overline{\pi} \subseteq u^{\mathcal{M}}$), an element A is found such that after inserting A into $\overline{\pi}$ no inclusion of the type above is disrupted. Then the pair $\{p, A\}$ is inserted in a place β such that $\beta \to \pi$ is an edge of the Ugraph G. We refrain from stating the exact conditions which guarantee that such a stabilization phase can actually take place, since they are quite involved. The interested reader can find a complete account in [7]. It turns out that the resulting decision procedure has a nondeterministic exponential-time complexity in the size of the input formula.

5.2 MLS with the Powerset Operator

Next we consider the case of the unquantified theory **MLSP**, namely the extension of **MLS** with an unrestricted number of occurrences of the powerset operator $\mathscr{P}(\cdot)$, where we recall that $\mathscr{P}(s) = \{t \mid t \subseteq s\}$.

Much as in the previous case, by way of a decomposition at the propositional level and a secondary decomposition of the kind shown in Sect. 3, the satisfiability problem for **MLSP** can be reduced to the satisfiability problem for conjunctions C

of flat **MLS**-literals having the form (28) plus conjuncts of the form

$$p = \mathscr{P}(q) \tag{30}$$

In the preceding section it was convenient to introduce a graph structure among places of flat conjunctions of **MLSU**. In the present case we will see that what is needed instead is a relation between sets of places and places.

We begin with some general considerations on the powerset operator. Let $s_1, s_2, \ldots, s_n$ be nonempty disjoint sets. Then we have

$$\mathscr{P}(s_1 \cup s_2 \cup \cdots \cup s_n) = \bigcup \big\{ \mathscr{P}^*(A) \,\big|\, A \subseteq \{s_1, s_2, \ldots, s_n\} \big\},$$

where $\mathscr{P}^*(A)$ stands for the collection of those subsets of $\bigcup_{s \in A} s$ which have nonempty intersection with every element of A. The above formula can be easily verified by observing that every element on the right-hand side of the equality is a subset of $s_1 \cup s_2 \cup \cdots \cup s_n$. On the other hand, if t is an element of $\mathscr{P}(s_1 \cup s_2 \cup \cdots \cup s_n)$, then $t \in \mathscr{P}^*(A_t)$, where $A_t = \{s_i \mid s_i \cap t \neq \emptyset,\ i = 1, \ldots, n\}$. Hence, if $p = \mathscr{P}(q)$ is a powerset clause in a given flat **MLSP**-conjunction C, and $\alpha_1, \ldots, \alpha_\ell$ are places for C such that $\alpha_1(q), \ldots, \alpha_\ell(q)$ are all true, there must exist places $\beta_1, \ldots, \beta_k$ such that $\beta_1(p), \ldots, \beta_k(p)$ are true, and such that elements of $\mathscr{P}^*(\overline{\alpha}_1, \ldots, \overline{\alpha}_\ell)$ can lie in $\overline{\beta}_1 \cup \cdots \cup \overline{\beta}_k$ only. We indicate such relationship by writing $\{\alpha_1, \ldots, \alpha_\ell\} \to \beta_i$, for each $i = 1, \ldots, k$. (The notation used suggests the idea of a "flow" from the places $\alpha_1, \ldots, \alpha_\ell$ to the place β.) In order to be more precise, we give the following definition.

Definition 2 A nonempty set $\{\alpha_1, \ldots, \alpha_\ell\}$ of places of C is called a *P-node* if there is a powerset clause $p = \mathscr{P}(q)$ in C such that $\alpha_j(q)$ is true, for all $j = 1, \ldots, \ell$.

If A is a P-node, then a place β is called a *target* of A if for every powerset clause $p = \mathscr{P}(q)$ we have that $\beta(p) \leftrightarrow \alpha(q)$ is true, for all $\alpha \in A$.

A place β is called *initial* if it is not the target of any P-node A. (Intuitively speaking, initial places are those places which are not constrained by powerset clauses. It is then reasonable to start initialization from these places.)

A first condition for C to be satisfiable follows immediately from the fact that if $s = \mathscr{P}(t)$, then $u \in s$ if and only if $u \subseteq t$. In terms of places, this translates into the following:

> "if $p = \mathscr{P}(q)$ is a clause in C, then $\pi^x(p)$ is true if and only if $\pi(x) \to \pi(q)$ is true, for every place π."

This condition ensures that during the $\in$-phase, insertion of $z^{\mathcal{M}}$ in sets of type $\{x^{\mathcal{M}} : x \in L \mid \pi^x(p)\ \&\ x \leqslant z\}$ will not disrupt any (already established) inclusion of the type

$$\begin{aligned}\big\{x^{\mathcal{M}} : x \in L \,\big|\, \pi^x(p)\ \&\ x < z\big\} \cup \bigcup \big\{\overline{\pi} : \pi \in \Pi \,\big|\, \pi(p)\big\} \\ \subseteq \mathscr{P}\Big(\big\{x^{\mathcal{M}} : x \in L \,\big|\, \pi^x(q)\ \&\ x < z\big\} \cup \bigcup \big\{\overline{\pi} : \pi \in \Pi \,\big|\, \pi(q)\big\}\Big),\end{aligned}$$

for any powerset clause $p = \mathscr{P}(q)$ in C.

In order to force equalities in place of the above inclusions, a stabilization phase is needed each time a new variable z of C is processed during the $\in$-phase. Such a stabilization will proceed in a manner defined by certain special edges of the type $\{\alpha_1, \ldots, \alpha_\ell\} \to \beta$. (These special edges are those ones whose target is a "set" of maximum rank; this idea guarantees against circularity.)

In this case stabilization steps are easy to describe; they just consist of assignments of the form

$$\overline{\beta} := \mathscr{P}^*(\overline{\alpha}_1, \ldots, \overline{\alpha}_\ell) \Big\backslash \bigcup_{\{\alpha_1,\ldots,\alpha_\ell\} \to \gamma} \overline{\gamma},$$

where $\{\alpha_1, \ldots, \alpha_\ell\} \to \beta$ is a special edge.

The initialization phase can be described roughly as follows. Since initial places are not restricted by any powerset clause, we can initialize them freely using a sufficiently large number of individuals. Moreover the empty set can be assigned to the place $\pi^{\emptyset}$ (for simplicity, it is convenient to assume that $\emptyset$ is a variable occurring in C which stands for the empty set). At this point proliferation of elements can start. This will continue until each place has been assigned at least one element. More specifically, for each P-node $\{\alpha_1 \ldots, \alpha_\ell\}$, with $\overline{\alpha}_1, \ldots, \overline{\alpha}_\ell$ nonempty, elements in $\mathscr{P}^*(\overline{\alpha}_1, \ldots, \overline{\alpha}_\ell) \setminus \bigcup_{\{\alpha_1,\ldots,\alpha_\ell\} \to \gamma} \overline{\gamma}$ are distributed among all its targets γ in a suitable manner. Reference [8] states conditions which ensure that the initialization and subsequent stabilization phases can execute properly.

Note, finally, that it can be proved that if m is the number of different variables occurring in C, then C is satisfiable if and only if it has a model of rank at most $2^{2^{m+1}+m+2} + 1$. The resulting search-based decision procedure for **MLSP** is not even elementary and is of theoretical interest only.

It is interesting to contrast this last result with the fact that there are formulae of the theory **MLSU** which admit only infinite models. For example, the formula $x \neq \emptyset \,\&\, x \subseteq \bigcup x$ is not finitely satisfiable, but the assignment $Mx = \{\emptyset_0, \emptyset_1, \ldots\}$, where $\emptyset_0 = \emptyset$ and $\emptyset_{n+1} = \{\emptyset_n\}$, for $n = 0, 1, \ldots$, clearly satisfies it.

We close this section by noticing that the extension **MLSSP** of **MLSP** with also the singleton operator has been shown decidable in [3] and, using the formative processes approach, in [13]. Roughly speaking, it is shown that an **MLSSP**-formula C is injectively satisfiable if and only if a certain nondeterministic association procedure can produce a canonical model of C in time bounded by a doubly exponential expression in the number m of variables occurring in C. To prove the necessity of such condition, an existing model $\mathcal{M}$ of C is used as an oracle to instantiate a computation of the association procedure. Again, matters are complicated by the presence of singleton clauses, which cause some of the places of C to become trapped. Trapped places are handled by maintaining a one-one partial map from the canonical model under construction into the oracle model. Such a map is intended to guide a correct instantiation of the association algorithm. At each step of a computation of the association algorithm, the rank of any set can increase at most by one. Therefore, as a by-product, it follows that any formula C of **MLSSP** is injectively satisfiable if and only if it has a model of rank doubly exponential in the number of variables in C.

6 Epilogue

Much work still remains to be done to bring to completion Jack's initial project. This includes the reimplementation of the ÆtnaNova/Referee verifier in a currently supported programming language, like Javascript and Java, and the completion of the full formalization in Referee of the proof of the Cauchy integral theorem (cf. [29]), as pointed out in [21].

In addition, Computable Set Theory is still a source of challenging problems and interesting applications. Indeed, just very recently the long-standing open problem concerning the decidability of the prenex sentences in the set-theoretic Bernays-Shönfinkel-Ramsey (BSR) class over von Neumann's cumulative hierarchy has received a positive solution (cf. [23, 24]). We recall that BSR-sentences have the form

$$(\exists x_1)\dots(\exists x_n)(\forall y_1)\dots(\forall y_m)\varphi(x_1,\dots,x_n,y_1,\dots,y_m),$$

where φ is any Boolean combination of atoms of the types $x = y$ and $x \in y$. We also mention a recent proof of the NP-completeness of the consistency problem for certain description logic knowledge bases via a reduction to the satisfiability problem for a quantified fragment of set theory involving ordered pairs and some operators to manipulate them (cf. [9]).

A further line of research (still active) concerns the investigation of the satisfiability problem for extensions of **MLSS** with various combinations of constructs such as the finiteness predicate *Finite*, the general union operator $\bigcup$, the powerset operator $\mathscr{P}$, the Cartesian product $\times$, etc. We expect that the technique of formative processes, developed in [13], can be a valuable starting point to tackle such problems. A particularly significant problem in this area is the satisfiability problem for **MLSS** with the Cartesian product, which can be regarded as a set-theoretic counterpart of the celebrated Hilbert's tenth problem, concerning the solvability of Diophantine polynomial equations by a mechanical procedure.[10] As is well known, a negative answer to Hilbert's tenth problem was the result of the combined work of Yuri Matiyasevich, Hilary Putnam, Julia Robinson, and Martin Davis.

References

1. Armando, A., Ranise, S., Rusinowitch, M.: Uniform derivation of decision procedures by superposition. In: Fribourg, L. (ed.) Proc. of Computer Science Logic 2001. LNCS, vol. 2142, pp. 513–527 (2001)

[10] A partial negative result for the satisfiability problem of **MLSS** with the Cartesian product is contained in [5], when a cardinality operator is also admitted.

2. Breban, M., Ferro, A., Omodeo, E.G., Schwartz, J.T.: Decision procedures for elementary sublanguages of set theory. II. Formulas involving restricted quantifiers, together with ordinal, integer, map, and domain notions. Commun. Pure Appl. Math. **34**, 177–195 (1981)
3. Cantone, D.: Decision procedures for elementary sublanguages of set theory. X. Multilevel syllogistic extended by the singleton and powerset operators. J. Autom. Reason. **7**(2), 193–230 (1991)
4. Cantone, D.: A fast saturation strategy for set-theoretic tableaux. In: Galmiche, D. (ed.) Proc. of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods. LNAI, vol. 1227, pp. 122–137. Springer, Berlin (1997)
5. Cantone, D., Cutello, V., Policriti, A.: Set-theoretic reductions of Hilbert's tenth problem. In: Börger, E., Kleine Büning, H., Richter, M.M. (eds.) Proceedings of 3rd Workshop Computer Science Logic—CSL'89, Kaiserslautern, 1989, vol. 440, pp. 65–75. Springer, Berlin (1990)
6. Cantone, D., Ferro, A., Omodeo, E.G.: Computable Set Theory. Oxford Science Publications of International Series of Monographs in Computer Science, vol. 6. Clarendon Press, Oxford (1989)
7. Cantone, D., Ferro, A., Schwartz, J.T.: Decision procedures for elementary sublanguages of set theory. V. Multi-level syllogistic extended by the general union operator. J. Comput. Syst. Sci. **34**(1), 1–18 (1987)
8. Cantone, D., Ferro, A., Schwartz, J.T.: Decision procedures for elementary sublanguages of set theory. VI. Multi-level syllogistic extended by the powerset operator. Commun. Pure Appl. Math. **38**, 549–571 (1985)
9. Cantone, D., Longo, C., Nicolosi Asmundo, M.: A decidable quantified fragment of set theory involving ordered pairs with applications to description logics. In: Bezem, M. (ed.) Computer Science Logic (CSL'11)—25th International Workshop/20th Annual Conference of the EACSL. Leibniz International Proceedings in Informatics (LIPIcs), Dagstuhl, Germany, April 2011, vol. 12, pp. 129–143 (2011). Schloss Dagstuhl–Leibniz-Zentrum für Informatik
10. Cantone, D., Omodeo, E.G., Policriti, A.: The automation of syllogistic. II. Optimization and complexity issues. J. Autom. Reason. **6**(2), 173–187 (1990)
11. Cantone, D., Omodeo, E.G., Policriti, A.: Set Theory for Computing. From Decision Procedures to Declarative Programming with Sets. Monographs in Computer Science. Springer, New York (2001)
12. Cantone, D., Omodeo, E.G., Schwartz, J.T., Ursino, P.: Notes from the logbook of a proof-checker's project. In: Dershowitz, N. (ed.) Proc. of the International Symposium on Verification: theory and practice. LNCS, vol. 2772. Springer, Berlin (2012)
13. Cantone, D., Omodeo, E.G., Ursino, P.: Formative processes with applications to the decision problem in set theory: I. Powerset and singleton operators. Inf. Comput. **172**, 165–201 (2002)
14. Cantone, D., Terranova, R., Ursino, P.: Experimental comparison of two tableau-based decision procedures for MLSS. In: CILC08—Convegno Italiano di Logica Computazionale, 10–12 July, Perugia (2008)
15. Cantone, D., Zarba, C.G., Schwartz, J.T.: A decision procedure for a sublanguage of set theory involving monotone, additive, and multiplicative functions, II: the multi-level case. Matematiche **LX**(1), 133–162 (2005)
16. Cantone, D., Zarba, C.G.: A new fast tableau-based decision procedure for an unquantified fragment of set theory. In: Caferra, R., Salzer, G. (eds.) Proc. of the International Workshop on First-Order Theorem Proving (FTP'98). LNAI, vol. 1761, pp. 126–136. Springer, Berlin (2000)
17. Ferro, A., Omodeo, E.G.: Decision procedures for elementary sublanguages of set theory. VII. Validity in set theory when a choice operator is present. Commun. Pure Appl. Math. **40**, 265–280 (1987)
18. Ferro, A., Omodeo, E.G., Schwartz, J.T.: Decision procedures for elementary sublanguages of set theory. I. Multi-level syllogistic and some extensions. Commun. Pure Appl. Math. **33**(5), 599–608 (1980)
19. Fitting, M.C.: First-Order Logic and Automated Theorem Proving, 2nd edn. Graduate Texts in Computer Science. Springer, Berlin (1996). 1st edn. (1990)

20. Formisano, A., Omodeo, E.G.: Theory-specific automated reasoning. In: Dovier, A., Pontelli, E. (eds.) A 25-Year Perspective on Logic Programming, pp. 37–63. Springer, Berlin (2010)
21. Omodeo, E.G.: The Ref proof-checker and its "common shared scenario". This volume, pp. 121–167 (2012)
22. Omodeo, E.G., Cantone, D., Policriti, A., Schwartz, J.T.: A computerized referee. In: Schaerf, M., Stock, O. (eds.) Reasoning, Action and Interaction in AI Theories and Systems—Essays Dedicated to Luigia Carlucci Aiello. LNAI, vol. 4155, pp. 117–139. Springer, Berlin (2006)
23. Omodeo, E.G., Policriti, A.: The Bernays-Schönfinkel-Ramsey class for set theory: semidecidability. J. Symb. Log. **75**(2), 459–480 (2010)
24. Omodeo, E.G., Policriti, A.: The Bernays-Schönfinkel-Ramsey class for set theory: decidability. J. Symbol. Log. (2012, to appear)
25. Omodeo, E.G., Schwartz, J.T.: A 'Theory' mechanism for a proof-verifier based on first-order set theory. In: Kakas, A., Sadri, F. (eds.) Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert Kowalski, Part II. LNAI, vol. 2408, pp. 214–230. Springer, Berlin (2002)
26. Schwartz, J.T. (ed.): Mathematical Aspects of Computer Science. Proc. of Symposia in Applied Mathematics, vol. 19. Am. Math. Soc., Providence (1967)
27. Schwartz, J.T.: Set theory as a language for program specification and programming. Courant Institute of Mathematical Sciences, New York University (1970)
28. Schwartz, J.T.: A survey of program proof technology. Technical report 001, New York University, Department of Computer Science, September 1978
29. Schwartz, J.T., Cantone, D., Omodeo, E.G.: Computational Logic and Set Theory: Applying Formalized Logic to Analysis. Springer, Berlin (2011). Foreword by M. Davis
30. Schwartz, J.T., Dewar, R.B., Schonberg, E., Dubinsky, E.: Programming with Sets: An Introduction to SETL. Springer, New York (1986)
31. Zarba, C.G., Cantone, D., Schwartz, J.T.: A decision procedure for a sublanguage of set theory involving monotone, additive, and multiplicative functions, I: the two-level case. J. Autom. Reason. **33**(3–4), 251–269 (2004)

Jack Schwartz and Robotics: The Roaring Eighties

Micha Sharir

Abstract This article reviews the birth and growth of algorithmic motion planning in robotics, and the immense contributions of Jack Schwartz to the creation and development of this area during the 1980s. These contributions have started with the series of works on the "Piano Movers" problem, by Jack and myself (reviewed in detail in this article), and continued in many other works, mostly theoretical but with a strong practical motivation. These works, by Jack and others, have brought together many disciplines in mathematics and computer science, and have had enormous impact on the development of computational geometry.

1 Prologue

1.1 Jack Schwartz and Robotics, and Me

It is a great honor for me to contribute this essay, attempting to summarize Jack Schwartz's work in robotics during the 1980s, to this volume, commemorating the life and work of this amazing person. It has been a double honor for me to have had the opportunity to meet Jack and to serve for about a decade as one of his disciples. This has had a tremendous influence on my scientific work. Jack has been a role model whom I desperately tried to imitate, impossible as it may have been. Working with him was like being with Lewis Carroll's red queen: One had to run twice as fast to get somewhere, and this is one of the main things I learnt from him, although he himself had no difficulty running ten times as fast, without the slight indication of any perspiration.

Work on this paper was partially supported by NSF Grant CCF-08-30272, by Grant 2006/194 from the U.S.–Israel Binational Science Foundation, by Grant 338/09 from the Israel Science Fund, and by the Hermann Minkowski–MINERVA Center for Geometry at Tel Aviv University.

M. Sharir (✉)
School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel
e-mail: michas@post.tau.ac.il

M. Sharir
Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA

M. Davis, E. Schonberg (eds.), *From Linear Operators to Computational Biology*,
DOI 10.1007/978-1-4471-4282-9_6,

My collaboration with Jack has started in 1977, when I joined the SETL project as a new postdoc and worked with him on high-level program optimization. For me this was a rather shocking metamorphosis, as my Ph.D. dissertation was on extreme operators in Banach spaces. Of course, I already knew of Jack from that period, mainly through the monumental trilogy of books, *Linear Operators*, by Dunford and Schwartz (the first book [16] cost only $23 at the time of publication, 1958), which was at that time the bible of functional analysis. Thus working on optimizing compilers was a rather drastic change in my career, but I survived it reasonably well, with many fond memories of this period. I remember the weekly lunch meetings of the group that Jack has organized, in which everything, technical and non-technical, was discussed, and the routine lunches, on most other days, at the legendary Szechuan Taste restaurant (which closed one day "for renovation" and never opened again). There were the many SETL Newsletters, most of which remained unpublished, so it was a non-trivial task to convince Tel Aviv University to recognize them as "real" papers during my appointment process. I also remember Jack's way of correcting the technical papers I have written in my poor English back then, with many iterations of red miniature-size handwritten scribbles. Semi-unconsciously, I have been doing the same to my students ever since.

The work on SETL has proceeded quite well, but in the early 80s Jack was already looking for a new area to explore. I remember quite sharply one day in 1981, as I was sitting in his office during a visit to NYU, when all of a sudden he has proposed to me the problem of planning a collision-free motion of a line segment amid polygonal obstacles in the plane, and already sketched a possible solution, based on the notion of critical curves. In no time at all, we were "trapped" in this problem, Jack in his usual zeal of looking over the horizon for something new, and me in great relief, as this has been a topic much closer to my mathematical heart than optimizing compilers. This has started an avalanche of research, that has drawn me into the fascinating area of robotics and computational geometry, and has led to the series of joint papers with Jack "On the Piano Movers' Problem", which have laid down the foundation for *algorithmic motion planning*.

The first paper in the series [36] presented an algorithm for solving Jack's initial problem, but it was the second one [37] which has been the most influential, as it gave a general solution to the motion planning problem, using and refining techniques from algebraic geometry. Specifically, given a moving system B with k degrees of freedom, a collection of obstacles, whose shape and locations are known to the planning system, and a start and goal placements of B, we want to determine whether there exists a continuous collision-avoiding motion of B from the start placement to the goal placement, and, if so, produce such a motion. The algorithm is based on Collins' *cylindrical algebraic decomposition* [15], but refines it to handle the topology of the resulting *free configuration space* of B, namely, the subset of the parametric k-space (in which placements of B are represented as points), consisting of those placements in which B does not intersect any obstacle. Under reasonable assumptions on the shape of B and the obstacles, and on the motion of B, the resulting algorithm is exact, and takes time which is doubly exponential in k, but polynomial in the other parameters of the problem. Thus, for any fixed system B, one gets an exact polynomial-time algorithm for planning its motion amid a given collection of obstacles (again, under reasonable assumptions on the shape of the moving system and the obstacles).

(I feel "safe" in praising that paper, because it was to a large extent Jack's creation. My role has been that of a disciple, running after the master, twice as fast, learning a lot from him, and helping wherever I could.)

This was only the beginning. There have been two main developments that grew out of this initial venture. First, as in the famous story about a person who finds a button and makes a suit out of it, Jack entered the world of robotics, with his usual unbounded energy, vigor, and stamina. With the aid of a large NSF Infrastructural Grant, he has founded, a few years later, the Robotics Lab at the Courant Institute, which has become one of the major centers for robotics research during the 1980s. I had the privilege of serving as the Deputy Director of the Lab for four years, from 1985 to 1989, during which I experienced intensive and productive interaction with Jack. As usual, he was interested in everything: Many fundamental questions in motion planning, questions related to shape and pattern recognition (for which he has introduced the technique of *geometric hashing* [9, 24], which is still used today, quite effectively, in the analysis of molecular structures in bioinformatics [54]), questions related to friction and grips of objects [32, 43], and many others. A large portion of this work, and several other parallel developments, is recorded in a book edited by John Hopcroft, Jack and myself [23], and in several surveys [39, 41, 44].

The other development was an extensive study of motion planning algorithms as a new sub-area within *computational geometry*, a field which was founded about 10 years earlier, and which experienced a tremendous growth during the 1980s, for which the motion planning problem has been a major motivating factor. Among the main developments resulting from this interaction were the study of arrangements, Davenport-Schinzel sequences and their geometric applications, and space-decomposition techniques. See [49] for details.

During the 1980s I have drifted from robotics into computational geometry, and have stayed there ever since. Jack dabbled a bit in this area too (see, e.g., [21, 42]), but was never really attracted to it. Towards the end of the 1980s and beginning of the 1990s, his interests have shifted to other topics (ultracomputers, multimedia, logic, bioinformatics, and what not), and our scientific ways, but not our friendship, have slowly drifted apart.

2 Jack and Motion Planning: The Piano Movers Papers I, II

As already noted, the theory of algorithmic motion planning has been presented in numerous survey papers, some by Jack and me [41, 44]; see also [45–47], so I prefer not to recreate here yet another survey of this kind. Instead, I will present the theory as it was developed by Jack and myself. I will then describe several other developments in the theory of robotics, in which Jack played an instrumental role too.

2.1 Piano Movers I

As described above, this was Jack's first encounter with motion planning. The (simplest instance of the) problem is: Let e be a given line segment, and let $\mathcal{O}$ be a collection of pairwise-disjoint polygonal obstacles in the plane, consisting of a total of n edges. Let s and t denote the start the target *configurations* (i.e., placements) of e, both assumed to be *free*, i.e., not to intersect any obstacle in $\mathcal{O}$. The goal is to determine whether there exists a continuous *collision-free motion* of e from s to t, during which it never collides with any obstacle, and if so, to plan such a motion.

A few observations are in order. First, placements of a line segment in the plane have three degrees of freedom. For example, we may represent each placement by the triple (x, y, θ), where (x, y) are the coordinates of a fixed endpoint a of e, and where θ is its orientation.[1] We can therefore represent placements of e as points in a 3-dimensional parametric space; topologically, it is the cylinder $\mathbb{R}^2 \times \mathbb{S}^1$, but, for simplicity of presentation, let us think of it as the real Euclidean 3-space $\mathbb{R}^3$. We refer to this parametric space as the *configuration space* of e.

Now, each obstacle C in $\mathcal{O}$ is mapped to a region C^* in configuration space, consisting of all (points representing) placements of e in which e intersects C. Many names have been given to such a region C^*, including *expanded obstacle*, *C-obstacle*, *configuration-space obstacle*, and *forbidden region*. The simple observation is that the union $\mathcal{K}$ of the regions C^*, over all obstacles $C \in \mathcal{O}$, is the *non-free portion* of the configuration space. It consists precisely of those placements in which a collision occurs between e and some obstacle. Hence, its complement $\mathcal{F} = \mathcal{K}^c$ is the *free configuration space*, the subset of all free placements of e.

By assumption, both s and t belong to $\mathcal{F}$. Moreover, a collision-free motion of e from s to t maps to a connected arc between s and t which lies fully in $\mathcal{F}$. In other words, there exists a collision-free motion of e from s to t if and only if s and t belong to the same connected component of $\mathcal{F}$. Hence, one approach to the problem is to construct $\mathcal{F}$ and to decompose it into its connected components. We want to represent the components in an effective manner that will allow us to (a) to determine whether s and t lie in the same component of $\mathcal{F}$, and (b) of they do, construct a connected arc from s to t within $\mathcal{F}$.

(We note that in this formulation we only consider the purely geometric aspect of the problem, and ignore issues related to physical constraints that might affect the realizability of certain kinds of motions that we might want e to undertake, such as sharp turns or some other combinations of translation and rotation. For example, if e represents a moving vehicle, we cannot translate it sideways. There have been many studies of this more realistic *nonholonomic* motion planning [29]. Other realistic aspects of motion planning are motion planning in unknown environments, motion

[1]We note in passing that θ is not a good choice of parameter, because it makes the equations that will arise in the analysis non-algebraic, whereas algebraicity will be a crucial ingredient of the analysis. This is not a serious issue, because we can replace θ by $\tan(\theta/2)$ and make all the relevant equations algebraic. Still, to make the presentation more readable, we stick to using θ.

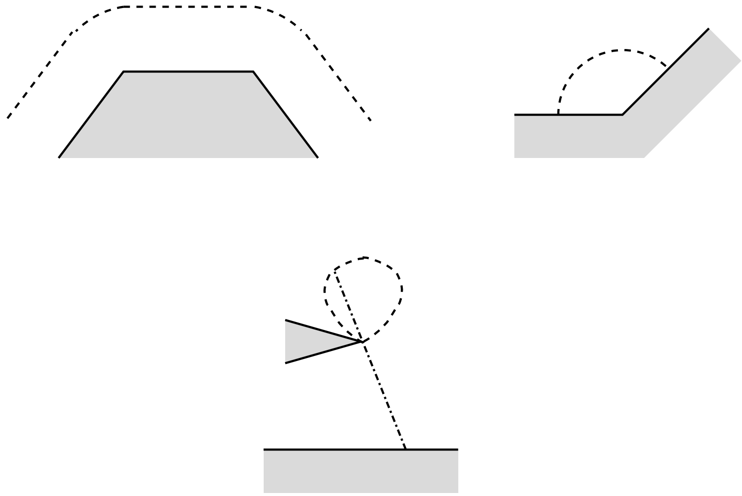


Fig. 1 Critical curves in the motion of a line segment

planning with uncertainty, optimal motion planning, and many others. However, in this initial encounter with motion planning, we push aside all these issues.)

The considerations made so far are very general, and apply to essentially all motion planning problems. We will get back to them later on, but for now let us go back to our simple special case of a line segment in the plane.

Jack's idea was to project the 3-dimensional free configuration space $\mathcal{F}$ onto the xy-plane, and to construct a cell decomposition of the projected $\mathcal{F}^*$. Specifically, consider a fixed point (x, y) in the plane. When we place e with its endpoint a at (x, y), it is left with one degree of freedom of rotation about a. As e rotates about a from a free placement it may swing freely through the entire 360° range of orientations, but typically it will stop being free by hitting some obstacle, both in the clockwise and the counterclockwise directions. This defines a collection of free arcs, each of which can be described symbolically by the pair of obstacles that determine its endpoint orientations.

In general, as we move (x, y) slightly, each of the free arcs will vary continuously, and will continue to have the same discrete label of the pair of delimiting obstacles. However, at certain *critical placements*, there will be a discrete change in the number of arcs or in its labels. For example, two free arcs may fuse into a single free arc, a free arc may split into two subarcs, a free arc may shrink to a point and then disappear, a new free arc may emerge, or the label of a free arc may change, so that instead of terminating at a contact with one obstacle it now terminates at a contact with a different obstacle. See Fig. 1 for illustrations of such critical placements.

The locus of all points (x, y) at which one of these criticalities occurs, for a fixed type of transition involving a fixed set of labeling obstacles, is a curve, referred to as a *critical curve*. These curves are of several kinds:

(i) Displacements of edges and convex corners of the obstacles by distance $|e|$. They are straight segments and circular arcs.
(ii) Curves that trace the position of a as e moves so that its relative interior touches a convex corner of an obstacle and either its other endpoint touches some obstacle edge or its relative interior also touches a second convex obstacle corner. These are either straight segments (in the latter case) or fourth-degree curves, known as *conchoids of Nicomedes* [30] (in the former case).

Altogether we have $O(n^2)$ critical curves, and they partition the xy-plane into $O(n^4)$ *non-critical regions*, each being a maximal connected portion of the complement of the union of the critical curves. Technically, each non-critical region is a face in the *arrangement* $\mathcal{A}$ of the critical curves. Each such face f has the property that the sequence σ_f of the labels of the free arcs around (x, y) is the same for all points $(x, y) \in f$.

This leads to the following cell decomposition $\mathcal{D}$ of $\mathcal{F}$. Each cell c of $\mathcal{D}$ is of the form (f, o_1, o_2), where f is a face of the planar arrangement $\mathcal{A}$, and o_1, o_2 are the obstacles delimiting a free arc around any point in f. Formally, c consists of all placements (x, y, θ), such that $(x, y) \in f$ and $\theta_{o_1}(x, y) < \theta < \theta_{o_2}(x, y)$, where $\theta_{o_1}(x, y)$, $\theta_{o_2}(x, y)$ are the orientations that delimit the corresponding free arc around (x, y).

Clearly, each cell in the above decomposition is an open connected portion of $\mathcal{F}$, the cells are pairwise disjoint, and the closure of $\mathcal{F}$ is the union of the closures of the cells.

This is not yet sufficient to construct the connected components of $\mathcal{F}$, because a connected component consists in general of several cells of $\mathcal{D}$. The extra information that we need is the *adjacency* relationship between the cells. We want to find all pairs of cells (c_1, c_2) for which there exists a free motion from some point in c_1 to some point in c_2, which stays within the union of the closures of c_1 and c_2. (Informally, the boundaries of the two cells should overlap in an appropriately defined manner.) As one can show, adjacency is always realized by crossing a critical curve which separates the xy-projections of c_1 and c_2.

To recap, the algorithm first constructs all the critical curves and computes their arrangement $\mathcal{A}$. For each face f of $\mathcal{A}$, we can then simply pick an arbitrary point (x, y) in f, construct the free arcs about (x, y), obtain their labels, and thereby obtain the stack of cells of $\mathcal{D}$ that lie above f. The adjacency relation is also easy to construct, but we omit details of this step.

Now we can view the adjacency relation as a graph, called the *adjacency graph* (or *connectivity graph* G, whose nodes are the cells of $\mathcal{D}$ and whose edges record adjacencies between cells. It is now fairly obvious that the connected components of $\mathcal{F}$ correspond in a one-to-one manner to the connected components of G, which we compute by a trivial depth-first search, say.

Now, given the start and target configurations s and t, we find the cells c_s, c_t that contain s and t, respectively, and test whether they lie in the same connected component of G. If so, we determine that e can be moved from s to t without colliding with any obstacle. Otherwise, no collision-free motion is possible between s and t.

In the former case we can also plan a collision-free motion. For this, we store with each edge (c_1, c_2) of G, a point on the common boundary of c_1 and c_2, through which we can freely cross from c_1 to c_2. Navigation between any two points u, v of a fixed cell c, or between a point in the cell and a "border-crossing" point, or between two border-crossing points is easy: We move the endpoint a of e within the xy-projection f of c from the xy-projection of u to that of v, while varying the orientation of e so that it stays within the corresponding free arc; the further easy details are omitted. With some care, we can output the motion plan as a finite sequence of "elementary motions", each consisting of a pure translation, a pure rotation, or a mixed "sliding" motion which moves in configuration space along some small-degree algebraic curve. Again, we omit the details.

The case of a line segment (a "ladder" or a "rod"), as just described, is a special case of the more general problem where the moving object B is an arbitrary rigid polygonal region. This latter case was also treated in the Piano Movers I paper. The same basic approach was used, except that the critical curves becomes somewhat more complicated and more diverse.

Discussion The algorithm described above runs in $O(n^5)$ time. In retrospect, this is a rather inefficient solution. As the study of motion planning has progressed during the 1980s, much faster algorithms have been designed. The current best solutions run in nearly-quadratic time, both for the case of a ladder and for the case of a *convex* polygonal object (with a constant number of edges) see [4, 26–28, 51]. Nevertheless, the algorithm had several significant merits. First, it was the first exact "combinatorial" algorithm for this problem with a provable worst-case bound on its running time, which depends (polynomially) only on the number n of edges of the obstacles and not on any physical parameter, like how cluttered together are the obstacles. Second, it has introduced the idea of constructing and representing the free configuration space via the cell decomposition method, which Jack has later applied in fairly full generality in the Piano Movers II paper (reviewed next). Third, it has defined algorithmic motion planning as a solid subdiscipline within computational geometry, which has been a strong motivating force for the study of arrangements and for several major discoveries that were made during this study; see below for a few comments on these developments.

There has been one additional merit, very significant from my personal point of view. As my earlier collaboration with Jack was on the SETL project, I came to be fond of implementing complex algorithms in this language, using its high-level set-theoretical features. So I thought of trying to implement the Piano Movers I algorithm in SETL. This attempt has aborted right away: One of the very first steps was to test whether the input is valid: Assuming that one specifies each polygonal obstacle as a cyclic sequence of its vertices along its boundary, we need to verify (a) that the resulting boundary is not self-intersecting, and (b) that no pair of distinct obstacles intersect each other. Of course, each of these steps could be implemented, by a brute force approach, in quadratic time, but I felt that there ought to be a better way of doing that. I snooped around, and discovered computational geometry, a budding young field at that time, which nevertheless had already possessed several important tools and techniques (including solutions to the intersection detection problem I got stuck at). I was fascinating by this field and got drawn into it, and got stuck there ever since, for better or worse.

2.2 Piano Movers II

While the Piano Movers I paper was a first step into the realm of algorithmic motion planning, the second paper, Piano Movers II [37], has covered the entire landscape. In a sense, it provided a complete solution to any motion planning problem, using standard, albeit somewhat complex, techniques from algebraic geometry and topology. I will not describe here the full details of the technique, but will try to present its main highlights.

Let B be a mechanical system with k degrees of freedom, moving in d-space (for $d = 2$ or 3) amid a collection $\mathcal{O}$ of obstacles of known shapes and locations. One assumes that B, its degrees of freedom, and the obstacles are such that the conditions for a placement of B to be collision-free can be specified as a Boolean combination of finitely many polynomial equalities and inequalities in a fixed number of variables (the degrees of freedom, possibly plus some auxiliary variables), possibly defined with the aid of some quantifiers. In other words, we assume that $\mathcal{F}$ is a real *semi-algebraic set*, specified in this manner, and we want to construct it (or, more precisely, its connected components) in a more explicit manner.

This was not a really new problem. Back around 1950, Tarski [52] has considered the decidability of predicates in the first-order theory of the reals, and has provided a decision procedure, which was guaranteed to terminate, but with no elementary upper bound on its running time. This has been considerably improved by Collins [15], who, in 1975, has devised the method of *cylindrical algebraic decomposition*, or *CAD* in short, which constructs a cell decomposition of the input semi-algebraic set, with an elementary upper bound (which is doubly exponential in k but polynomial in the number of constraints and in their maximum degree) on the number of cells that it produces. The analysis in the Piano Movers II paper was based on constructing a CAD of $\mathcal{F}$, but has further extended the analysis to obtain the adjacency relation between the cells of the CAD, from which the connectivity (and in fact the full topological structure) of $\mathcal{F}$ could be derived.

Here is a brief, rather informal review of the CAD technique. Let k be the number of variables in the specification of $\mathcal{F}$. We proceed recursively on k. For $k = 1$, $\mathcal{F}$ is a collection of intervals and points, which we can find by computing and collecting the roots of each of the input polynomials, and then by testing explicitly each root, and each (open) interval between two consecutive roots, whether it belongs to $\mathcal{F}$. The resulting collection of points and intervals is the CAD representation of $\mathcal{F}$. For $k > 1$ we eliminate the last variable x_k from the given collection of polynomials, using the theory of *resultants* and *sub-resultants*, and obtain a new collection of $(k-1)$-variate polynomials, and we construct its CAD recursively. For each cell c of the $(k-1)$-dimensional CAD, we consider the cylinder $c \times \mathbb{R}$, and note that $\mathcal{F}$ meets this cylinder in pairwise-disjoint layers, stacked above each other, where each layer is either of the form $x_k = f(x_1, \ldots, x_{k-1})$ or of the form $f(x_1, \ldots, x_{k-1}) < x_k < g(x_1, \ldots, x_{k-1})$, where f and g are algebraic functions. The union of these layers, over all cells c, is the desired CAD for the k-dimensional $\mathcal{F}$.

Note that each cell c is homeomorphic to an open ball of some dimension $\le k$. Moreover, c can be specified in terms of at most $2k$ algebraic functions, at most two for each dimension. However, the degrees of these function keep growing as we go down the recursion, essentially doubling at each recursive stage. This is because each of these functions arises from some resultant or sub-resultant constructed from a pair of functions from the preceding stage. Thus, the maximum degree is doubly exponential in k. The number of cells is $O(n^{2k})$ times the degree bound. Hence, so far we have an algorithm which is doubly exponential in k, the number of degrees of freedom, but polynomial in all the other parameters (the number of constraints n and their maximum degree b) of the problem.

However, as in the Piano Movers I case, we need to augment this CAD with the adjacency information between cells. One of Jack's main contributions was to provide a procedure for doing that, yielding complete information about the topological structure of the CAD, and thus of $\mathcal{F}$. This part of the analysis is too technical, and I will omit its details in this survey. The cost of the adjacency computation is asymptotically the same as the cost of constructing the CAD, namely doubly exponential in k. (There is a slight technical issue here: One needs to ensure that the coordinate frame is in generic position, for otherwise the adjacency structure may involve singularities that might hamper proper reconstruction of the topology of $\mathcal{F}$. A simple solution is to apply a random rotation to the frame, which will then ensure, with probability 1, that no degeneracy of this sort can arise.)

To recap, under reasonable assumptions on the moving system B and on the obstacles, the free configuration space $\mathcal{F}$ is a semi-algebraic set in k-space, and the Piano Movers II algorithm provides a general procedure for decomposing it into finitely many simple cells, each homeomorphic to a ball and having succinct representation, together with the adjacency relation between the cells. The structure can be further processed, so that, given any two free placements s, t of B, we can determine whether t can be reached from s by a collision-free motion, and, if so, plan such a motion. For the last part, we simply locate (by brute force) the cells c_s, c_t which contain s and t, respectively, and check whether c_s and c_t lie in the same connected component of the adjacency graph G. If so, we retrieve the path π in G from c_s to c_t, and transform it into a free motion from s to t, using a fairly straightforward mechanism for navigating between any pair of points within the same cell, including crossings between cells through points on their common boundary.

3 Further Developments

Piano Movers III, IV, and V The three consecutive papers in the Piano Movers sequence were definitely an anti-climax (and one of them, the fourth, was not even co-authored with Jack). They presented concrete motion planning algorithms for some specific systems, using ad-hoc techniques (which nevertheless were inspired by the methodology of the first paper) which aimed to improve the rather monstrous bounds yielded by the general technique in the Piano Movers II paper.

The Piano Movers III paper [38] dealt with coordinated motion planning of several independent bodies. Specifically, it studied the cases of two or three disks moving simultaneously in the plane, where, in addition to avoiding the obstacles, they also need to avoid colliding with each other. The Piano Movers IV paper [50] considered the problem of moving a "spider", namely a multi-arm robot, consisting of many straight segments, all hinged around a common endpoint, amid polygonal obstacles in the plane. The Piano Movers V paper [40] considered the problem of moving a line segment (a "rod") in three dimensions amid polyhedral obstacles.

As these papers have explored a yet uncharted area, they were very careful and detailed in the analysis, enumerating a large number of types of critical curves, with detailed analysis of the effect of crossing such a curve on the free motion of the system. This is clearly a trait that Jack could not be blamed for not having it.

Roadmaps and Other Techniques for Semi-algebraic Sets The main disadvantage of the CAD decomposition used in the Piano Movers II paper is its size, which is typically doubly exponential in the number of degrees of freedom. This has motivated extensive further studies, looking for alternative techniques with smaller (singly exponential) storage and running time. We briefly describe two such improvements. The first one is based on *roadmaps*, where a roadmap is a network N of 1-dimensional curves within $\mathcal{F}$ which captures the topology of $\mathcal{F}$ in the following sense. (i) There is a mapping (retraction) $\varphi : \mathcal{F} \mapsto N$ which sends each $w \in \mathcal{F}$ to a point $\varphi(w) \in N$ along a trajectory contained in $\mathcal{F}$, so w and $\varphi(w)$ lie in the same connected component of $\mathcal{F}$. (ii) Two points $w, w' \in \mathcal{F}$ lie in the same connected component of $\mathcal{F}$ if and only if $\varphi(w)$ and $\varphi(w')$ lie in the same connected component of N. (iii) For algorithmic purposes, the complexity of the retraction φ, and the algebraic and combinatorial complexity of N, should all be reasonably small.

Given such a network, motion planning becomes simple. Given the start and target configurations s, t, we compute $\varphi(s)$ and $\varphi(t)$, and then search for a path π in N from (the edge containing) $\varphi(s)$ to (the edge containing) $\varphi(t)$. If such a path exists, we concatenate it to the motion from s to $\varphi(s)$ and to the reverse motion from $\varphi(t)$ to t to obtain the desired motion from s to t. If no such path is found, t cannot be reached from s.

Canny [10, 11] was the first to come up with a method for constructing roadmaps in general semi-algebraic sets, whose combinatorial complexity is only singly exponential in the number of degrees of freedom. Basu et al. [7] have later refined and improved his construction. See a detailed description in [8, Chap. 15]. The general idea is to use Morse theory, compute the critical points of $\partial\mathcal{F}$ in some direction v, connect the critical points by monotone curves along ∂F, and then recursively complete the roadmap within each of the v-orthogonal hyperplanes passing through the critical points. More details can be found in [8].

Cell Decompositions One obvious disadvantage of the CAD structure is that it contains too many cells. To see this, consider the situation in the plane. In this case

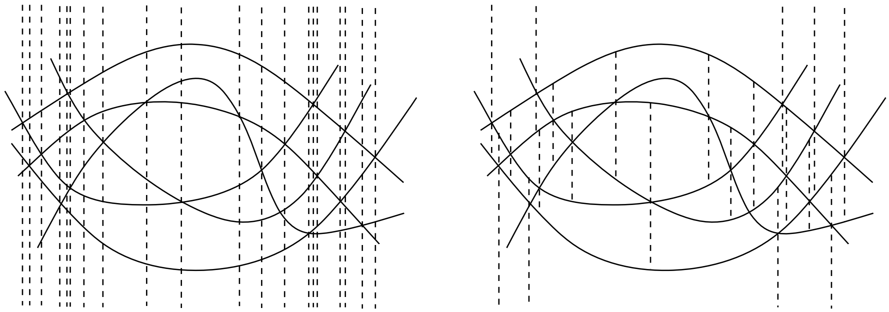


Fig. 2 *Left*: The CAD of a collection of curves in the plane. *Right*: The vertical decomposition

we have a collection of algebraic curves, and the CAD decomposes its arrangement into simple, trapezoidal-like cells. (More precisely, in the CAD construction each cell is either a point, an open vertical segment, an open x-monotone arc, or a trapezoidal-like open cell bounded by up to two vertical segments and by up to two arcs of the above kind.) The CAD construction finds all intersection points between the curves, all the singular points, and all the points with vertical tangency, and then draws a vertical line through each of these points. The vertical lines and the original curves partition the plane into cells of the desired shape, and the concrete set $\mathcal{F}$ is the union of some of these cells. See Fig. 2 (left).

In this case, the number of cells is $O(n^3)$, where n is the number of curves (ignoring the dependence on the algebraic degree), because there are $O(n^2)$ critical points of the above kind, and the vertical line through each of them (potentially) crosses each of the curves. This is too much. An obvious simple improvement is to draw, from each of the critical points a vertical line *segment*, upwards and downwards, only until it hits the next curve. This produces the same kind of cells as does the CAD, but their number drops to only $O(n^2)$. See Fig. 2 (right). This improved decomposition, known as *vertical decomposition*, is trickier to define in higher dimensions. See [12, 49] for details. The best known bound on the number of resulting cells is close to $O(n^{2d-4})$ (again, ignoring the dependence on the degree). This bound was not easy to obtain, and required a rather sophisticated analysis in three and four dimensions. While strongly suspected not to be (nearly) tight for $d \geq 5$, it beats by far the bound on the size of a CAD. One price we have to pay for this improvement is that the resulting decomposition is not a cell complex (as is the CAD), so deriving topological properties from this decomposition becomes more involved.

It is one of the hard open problems in computational geometry to derive sharper upper bounds on the complexity of vertical decompositions. Even more challenging is to find alternative, more efficient decomposition techniques for the general case. So far, the CAD and the improved vertical decomposition are the only known general-purpose techniques for this problem.

Lower Bounds Given that all general-purpose solutions of the motion planning problem are so inefficient (at least exponential in the number of degrees of free-

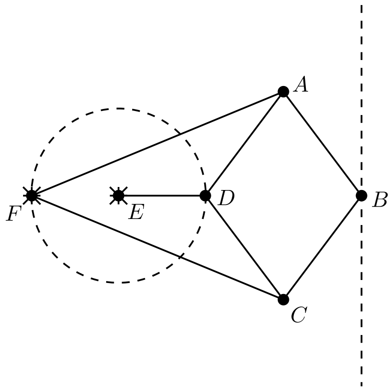


Fig. 3 The Peaucellier straight line motion linkage. E and F are fixed to the plane. As D rotates around E along the *dashed circle*, B traces the *dashed vertical line* segment

dom), one is led to the problem of showing that the problem is indeed computationally hard. Several papers have addresses this question during the 1980s, and I would like to mention three of them, each showing that the motion planning problem is PSPACE-hard, when the number of degrees of freedom is part of the input. Each of the papers uses a different system of (many) moving parts, and reduces some classical PSPACE-hard problem to the respective motion planning problem. Interestingly, what these papers effectively produce is a recipe for simulating any Turing machine with bounded tape "mechanically", where each step of the machine is realized by a sequence of moves of the mechanical system.

The first paper [22], by Jack, John Hopcroft (who was also interested in robotics at that time, and with whom Jack tried to establish some sort of collaboration) and myself, considered the "warehouseman's problem", in which one has a warehouse (a large rectangle), nearly packed with many rectangular boxes (of different sizes) lying on the floor, leaving very little free space. The goal is to move one specific box from one corner of the warehouse to another corner, by a sequence of moves, each sliding a single box on the floor of the warehouse. This is reminiscent of Sam Lloyd's 15 puzzle (see a brief history in [33, Chap. 9]), but is considerably more involved. It is shown in [22] that such a motion sequence can simulate a Turing machine with bounded tape, which implies that this instance of the motion planning problem is indeed PSPACE-hard.

The second paper [20], by Hopcroft, Joseph and Whitesides, considers mechanical linkages. These are collections of rigid links (line segments) in the plane, attached to each other at certain joints, about which they can rotate, and fixed to the plane at certain other points. The goal is to bring a designated endpoint of one of the links to a specified point or region in the plane.

Mechanical linkages have been studied already in the 19th century. An ingenious construction by Peaucellier [34] gives such a linkage to convert circular motion to linear motion (see Fig. 3), and many other sophisticated mechanisms were also developed. What Hopcroft et al. have shown is that one can construct such mechanisms for simulating the addition and multiplication of real numbers, and that a suitable combination of such components leads to a procedure for simulating an arbitrary linear bounded automaton, implying that this motion planning instance is also PSPACE-hard.

The third paper [35], by Reif (it originally appeared in 1979, so it is the oldest in this trilogy), uses a multi-arm robot navigating through a complex system of narrow tunnels in 3-space. It shows that such a system can simulate any symmetric Turing machine with bounded tape, making this instance also PSPACE-hard.

Efficient Motion Planning Algorithms The studies of the general motion planning problem, as reviewed above, have sent mixed signals to the community. On the positive side, they provided general frameworks for solving the problem in exact (algebraic) form and in fairly full generality. On the negative side, the general solutions were hopelessly inefficient. Moreover, the lower bounds just reviewed indicate that this inefficiency is probably inherent in the problem (when the number of degrees of freedom is arbitrarily large). This has led to an intensive study of techniques for solving efficiently the motion planning problem for specific systems with a small number of degrees of freedom.

For example, the problem with which it all started, namely motion planning of a ladder (line segment) in a planar polygonal environment, was solved in the Piano Movers I paper by an $O(n^5)$ algorithm, where n is the overall number of edges of the obstacles. Considerably improved algorithms, with (nearly) quadratic running time, have been obtained in [27, 51, 53], by a better exploitation of the geometry and combinatorics of the free configuration space.

Overall, the study of efficient motion planning algorithms has been very successful. As a matter of fact, motion planning was a driving force in the development of a large battery of sophisticated tools and techniques in computational geometry, including the study of arrangements of curves and surfaces, Davenport–Schinzel sequences, union of geometric objects, and many others. See [49] for a book summarizing many of these developments, and [1–3, 19, 48] for several related surveys.

Here is one example that illustrates this direction. The general techniques aim to construct some representation of the *entire* free configuration space. However, assuming that the start configuration s of the moving system is known in advance, we only need to construct the connected component C of $\mathcal{F}$ that contains s, because any target configuration not in C cannot be reached from s by a collision-free motion. This has led to the problem of computing a *single cell* in an arrangement of surfaces in higher dimensions, which in turn has led to the combinatorial question of bounding the combinatorial complexity of such a cell. For systems with two degrees of freedom, it was shown in [17] that the complexity of a single face in an arrangement of n low-degree algebraic curves is nearly linear in n, as opposed to the complexity of the entire arrangement, which is typically quadratic in n. This paper also gave a nearly-linear algorithm for constructing a single face (see also [13]). The problem was more challenging in higher dimensions. After an initial analysis in [18] for the three-dimensional case, the problem was solved in [6] in any dimension d, showing that the complexity of a single cell is close to n^{d-1}. See also [5] for the special case where the underlying surfaces are simplices.

On the theoretical side, this effort has been very successful, but the practical side was still suffering. The efficient solutions were quite complicated to implement, and were still time-consuming to be of any real practical value. This has led researchers back to the heuristic domain, albeit now backed up by better understanding of the overall structure of the problem. This has resulted, in the mid 1990s, in a general technique, called *probabilistic roadmaps* [25], which has proved successful in solving fairly complex motion planning problems in practice. Briefly, the idea is to sample many random free configurations of the system (essentially by sampling many configurations and selecting those that are free), including also the start and target configurations s, t, and then to use some simple-minded local planner to find pairs of these configurations that can be reached from one another by some sort of simple collision-free motion. This yields a *connectivity graph*, and the hope is that if the sampling is sufficiently dense then the graph will contain a path from s to t, if they are indeed reachable from one another within $\mathcal{F}$. Of course, many details have to be confronted, such as the possible existence of "narrow passages" in $\mathcal{F}$, which can be missed by the sampling process. Nevertheless, the method has been fairly successful in practice. See [14] for a recent account.

Motion Planning and Computational Geometry Although Jack realized the significance of computational (and combinatorial) geometry as a tool for tackling the motion planning problem, these fields have never really "gripped" him. He wrote with me a few papers in these areas, such as [42], but generally stayed away. I think he regarded computational geometry as being too low-level and technical for his taste, which was always to look for general principles of broader nature. As the 1980s have ended, Jack has drifted away from Robotics, turning his attention to multimedia as well as to his ongoing project of proof systems and computer logic, while I was "stuck" in computational geometry, still using motion planning as a major motivation and application area.

3.1 Jack and Robotics at Large

Motion planning was a cornerstone in Jack's involvement in robotics during the 1980s, but Jack's interest in the area grew in many other directions too. The Robotics Lab, put up in the 12th floor of 715 Broadway, was in fact a huge playground in which Jack could play with many shiny toys, some bought and some constructed on site. This has triggered many fundamental questions that Jack encountered, most of which without any known good solution at the moment, and Jack studied them too. Among those was the problem of pattern recognition, for which Jack developed several simple but very effective techniques for matching curves and point sets, including the *geometric hashing* technique mentioned above. Another problem was that of multi-finger grips of objects, which Jack has related, very elegantly, to basic machinery in convexity theory. See Mishra's paper in this collection [31] for a detailed account of this problem.

4 Epilogue

After Jack passed away in early 2009, the family and the Courant Institute held a memorial event "to celebrate the life of Jack Schwartz". I was unable to attend it, and instead sent a eulogy that was read at the event. I would like to conclude this tribute to Jack with that eulogy.

> I am very sorry that I was unable to attend this celebration of Jack's life. My deepest condolences to Diana and the family.
> In hebrew we say "Mori ve-Rabi"—my teacher and my rabbi, or, rather, my teacher and my spiritual guide. This is what Jack was to me, since I first met him in 1977, when I started my postdoc studies at NYU with him, working on the SETL project. I have never seen a person that better fits the notion of a scholar than Jack—a person with an infinite and unsatiable intellectual curiosity and prowess, with immense knowledge of practically everything, and with the ability to grasp and cope with new topics, diverse as they might be, and to form his own theories and discoveries, almost in a flash.
> In the first years of our acquaintance, I more or less served as his disciple, running after him and trying to catch up. It was indeed quite difficult to do so. His mind would continuously swivel around, always looking for new ideas and new fields and problems to invest in. Jack seemed to me like a butterfly, or perhaps, more appropriately, a busy bee, hopping from one flower to another, without ever getting tired. This was way too much for me; I only managed to execute one hop with him—from programming languages and optimizing compilers to robotics. This took place as I was sitting in his office, during a visit to NYU, where all of a sudden, he suggested the motion planning problem, sketched an initial solution, and started the ball rolling, leading to the founding of the NYU robotics lab, and to a very intense and successful collaboration between us. I stayed put, while he has continued to hop—to parallel computers, program verification, bioinformatics, multimedia, you name it. Jack kept on trying to get me to join him in this roller coaster ride, always selling me new ideas and new problems, and at some point almost managed to drag me into bioinformatics, but in the end it didn't work out—I just didn't have his energy.
> There are many other aspects of Jack that can be told, as I am sure other people will do in this gathering, such as his great interest in history, his special fondness for china (and for chinese food!), his generosity and caring, and so on. To me he was a friend and colleague, and I feel very fortunate to have met him. Meeting him has changed my life, and he has played a key role in the development of my career, for which I owe him my deepest and infinite gratitude. To me he was a giant, a constant source of admiration on one hand, and of frustration, realizing the impossibility of getting even close to what he was.
> The last years have been difficult. Jack fought the disease with courage, and at times with his usual approach of scientific curiosity. We have lost a great man, and stand in awe, clinging to his memory and vowing to continue with his legacy.
> Yehi zikhro barukh—may his memory be blessed.
>
> Micha Sharir

References

1. Agarwal, P.K., Pach, J., Sharir, M.: State of the union (of geometric objects). In: Proc. Joint Summer Research Conf. on Discrete and Computational Geometry: 20 Years Later. Contemp. Math., vol. 452, pp. 9–48. AMS, Providence (2008)
2. Agarwal, P.K., Sharir, M.: Davenport-Schinzel sequences and their geometric applications. In: Sack, J.R., Urrutia, J. (eds.) Handbook of Computational Geometry, pp. 1–47. North-Holland, Amsterdam (2000)

3. Agarwal, P.K., Sharir, M.: Arrangements of surfaces in higher dimensions. In: Sack, J.R., Urrutia, J. (eds.) Handbook of Computational Geometry, pp. 49–119. North-Holland, Amsterdam (2000)
4. Agarwal, P., Aronov, B., Sharir, M.: Motion planning for a convex polygon in a polygonal environment. Discrete Comput. Geom. **22**, 201–221 (1999)
5. Aronov, B., Sharir, M.: Castles in the air revisited. Discrete Comput. Geom. **12**, 119–150 (1994)
6. Basu, S.: The combinatorial and topological complexity of a single cell. Discrete Comput. Geom. **29**, 41–59 (2003)
7. Basu, S., Pollack, R., Roy, M.-F.: Computing roadmaps of semi-algebraic sets on a variety. J. Am. Math. Soc. **3**, 55–82 (1999)
8. Basu, S., Pollack, R., Roy, M.-F.: Algorithms in Real Algebraic Geometry. Algorithms and Computations in Mathematics, vol. 10. Springer, Berlin (2003)
9. Bastuscheck, C.M., Schonberg, E., Schwartz, J.T., Sharir, M.: Object recognition by 3-dimensional curve matching. Int. J. Intell. Syst. **1**, 105–132 (1986)
10. Canny, J.: The Complexity of Robot Motion Planning. MIT Press, Cambridge (1988)
11. Canny, J.: Computing roadmaps of general semi-algebraic sets. Comput. J. **36**, 504–514 (1993)
12. Chazelle, B., Edelsbrunner, H., Guibas, L., Sharir, M.: A singly exponential stratification scheme for real semi-algebraic varieties and its applications. Theor. Comput. Sci. **84**, 77–105 (1991). Also in Proc. 16th Int. Colloq. on Automata, Languages and Programming, pp. 179–193 (1989)
13. Chazelle, B., Edelsbrunner, H., Guibas, L., Sharir, M., Snoeyink, J.: Computing a face in an arrangement of line segments and related problems. SIAM J. Comput. **22**, 1286–1302 (1993)
14. Choset, H., Lynch, K.M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L.E., Thrun, S.: Principles of Robot Motion: Theory, Algorithms, and Implementation. MIT Press, Boston (2005)
15. Collins, G.E.: Quantifier elimination for real closed fields by cylindric algebraic decomposition. In: 2nd GI Conf. Aut. Theory and Formal Lang. LNCS, vol. 33, pp. 134–183. Springer, Berlin (1975)
16. Dunford, N., Schwartz, J.T.: Linear Operators, Part I: General Theory. Interscience Publishers, New York (1958)
17. Guibas, L., Sharir, M., Sifrony, S.: On the general motion planning problem with two degrees of freedom. Discrete Comput. Geom. **4**, 491–521 (1989)
18. Halperin, D., Sharir, M.: Almost tight upper bounds for the single cell and zone problems in three dimensions. Discrete Comput. Geom. **14**, 385–410 (1995)
19. Halperin, D., Sharir, M.: Arrangements and their applications in robotics: recent developments. In: Goldberg, K., Halperin, D., Latombe, J.C., Wilson, R. (eds.) The Algorithmic Foundations of Robotics, pp. 495–511. A.K. Peters, Boston (1995)
20. Hopcroft, J., Joseph, D., Whitesides, S.: Movement problems in 2-dimensional linkages. SIAM J. Comput. **13**, 610–629 (1984)
21. Hopcroft, J.E., Schwartz, J.T., Sharir, M.: Efficient detection of intersections among spheres. Int. J. Robot. Res. **2**(4), 77–80 (1983)
22. Hopcroft, J.E., Schwartz, J.T., Sharir, M.: On the complexity of motion planning for multiple independent objects; PSPACE hardness of the 'Warehouseman's problem'. Int. J. Robot. Res. **3**(4), 76–88 (1984)
23. Hopcroft, J.E., Schwartz, J.T., Sharir, M. (eds.): Planning, Geometry, and Complexity of Robot Motion. Ablex, Norwood (1987)
24. Kalvin, A., Schonberg, E., Schwartz, J.T., Sharir, M.: Two dimensional model based boundary matching using footprints. Int. J. Robot. Res. **5**(4), 38–55 (1986)
25. Kavraki, L., Svestka, P., Latombe, J.-C., Overmars, M.: Probabilistic roadmaps for path planning in high dimensional configuration spaces. IEEE Trans. Robot. Autom. **12**, 566–580 (1996)

26. Kedem, K., Sharir, M., Toledo, S.: On critical orientations in the Kedem-Sharir motion planning algorithm. Discrete Comput. Geom. **17**, 227–239 (1997)
27. Leven, D., Sharir, M.: An efficient and simple motion planning algorithm for a ladder moving in 2-dimensional space amidst polygonal barriers. J. Algorithms **8**, 192–215 (1987)
28. Leven, D., Sharir, M.: On the number of critical free contacts of a convex polygonal object moving in two-dimensional polygonal space. Discrete Comput. Geom. **2**, 255–270 (1987)
29. Li, Z., Canny, J.F. (eds.): Nonholonomic Motion Planning. Kluwer Academic, Norwell (1993)
30. Lockwood, E.H., Prag, A.: Book of Curves. Cambridge University Press, Cambridge (1961)
31. Mishra, B.: Mathematics mortua manus: Discovering dexterity. This volume
32. Mishra, B., Schwartz, J.T., Sharir, M.: On the existence and synthesis of multifinger positive grips. Algorithmica **2**, 541–558 (1987)
33. Pach, J., Sharir, M.: Combinatorial Geometry and Its Algorithmic Applications: The Alcalá Lectures. Am. Math. Soc. Press, Providence (2009). Lecture Notes, Alcalá, Spain, August–September (2006)
34. Peaucellier, M.: Correspondence. Nouv. Ann. Math. **3**, 414–415 (1864)
35. Reif, J.: Complexity of mover's problem and generalizations. In: Hopcroft, J., Schwartz, J., Sharir, M. (eds.) Planning, Geometry and Complexity of Robot Motion, pp. 267–281 (1987)
36. Schwartz, J.T., Sharir, M.: On the Piano Movers' problem: I. The case of a rigid polygonal body moving amidst polygonal barriers. Commun. Pure Appl. Math. **36**, 345–398 (1983)
37. Schwartz, J.T., Sharir, M.: On the Piano Movers' problem: II. General techniques for computing topological properties of real algebraic manifolds. Adv. Appl. Math. **4**, 298–351 (1983)
38. Schwartz, J.T., Sharir, M.: On the Piano Movers' problem: III. Coordinating the motion of several independent bodies: the special case of circular bodies moving amidst polygonal barriers. Int. J. Robot. Res. **2**(3), 46–75 (1983)
39. Schwartz, J.T., Sharir, M.: Mathematical problems and training in robotics. Not. Am. Math. Soc. **30**, 478–481 (1983)
40. Schwartz, J.T., Sharir, M.: On the Piano Movers' problem: V. The case of a rod moving in three-dimensional space amidst polyhedral obstacles. Commun. Pure Appl. Math. **37**, 815–848 (1984)
41. Schwartz, J.T., Sharir, M.: A survey of motion planning and related geometric algorithms. Artif. Intell. **37**, 157–169 (1988). Also in: D. Kapur, J. Mundy (eds.) Geometric Reasoning, pp. 157–169. MIT Press, Cambridge (1989), and in: S.S. Iyengar, A. Elfes (eds.) Autonomous Mobile Robots, vol. I, pp. 365–374. IEEE Comput. Soc., Los Alamitos (1991)
42. Schwartz, J.T., Sharir, M.: On the two-dimensional Davenport Schinzel problem. J. Symb. Comput. **10**, 371–393 (1990)
43. Schwartz, J.T., Sharir, M.: Finding effective 'force targets' for two-dimensional multifinger frictional grips. Algorithmica **8**, 1–20 (1992)
44. Schwartz, J.T., Sharir, M.: Algorithmic motion planning in robotics, an invited chapter. In: Van Leeuwen, J. (ed.) Algorithms and Complexity. Handbook of Theoretical Computer Science, vol. A, pp. 391–430. Elsevier, Amsterdam (1990)
45. Sharir, M.: Algorithmic motion planning in robotics. Computer **22**, 9–20 (1989)
46. Sharir, M.: Robot motion planning. In: Schonberg, E. (ed.) The Houses that Jack Built, pp. 287–300. Courant Institute, New York University (1995). Also in Commun. Pure Appl. Math. **48**, 1173–1186 (1995)
47. Sharir, M.: Motion planning. In: Goodman, J.E., O'Rourke, J. (eds.) Handbook of Discrete and Computational Geometry, pp. 733–754. CRC Press, Boca Raton (1997). Second Edition, pp. 1037–1064 (2004)
48. Sharir, M.: Arrangements of surfaces in higher dimensions. In: Chazelle, B., Goodman, J.E., Pollack, R. (eds.) Advances in Discrete and Computational Geometry (Proc. 1996 AMS Mt. Holyoke Summer Research Conference). Contemporary Mathematics, vol. 223, pp. 335–353. Am. Math. Soc., Providence (1999)
49. Sharir, M., Agarwal, P.K.: Davenport-Schinzel Sequences and Their Geometric Applications. Cambridge University Press, New York (1995)

50. Sharir, M., Ariel-Sheffi, E.: On the Piano Movers' problem: IV. Various decomposable two-dimensional motion planning problems. Commun. Pure Appl. Math. **37**, 479–493 (1984)
51. Sifrony, S., Sharir, M.: A new efficient motion planning algorithm for a rod in two-dimensional polygonal space. Algorithmica **2**, 367–402 (1987)
52. Tarski, A.: A Decision Method for Elementary Algebra and Geometry, 2nd edn. University of California Press, Berkeley (1951)
53. Vegter, G.: The visibility diagram: a data structure for visibility problems and motion planning. In: Proc. 2nd Scand. Workshop Algorithm Theory. Lecture Notes in Computer Science, vol. 447, pp. 97–110. Springer, Berlin (1990)
54. Wolfson, H.J., Rigoutsos, I.: Geometric hashing: an overview. IEEE Comput. Sci. Eng. **4**, 10–21 (1997)

Mathematics' Mortua Manus: Discovering Dexterity

B. Mishra

Abstract Dexterous manipulation, a major subfield of robotics and manufacturing, experienced a mathematical rebirth in the mid 80's, when this nascent field established many beautiful connections to convexity theory and computational geometry. Jack Schwartz played a seminal role in its inception and development. Here, I speculate on where Jack might have liked this field to go in the future.

1 Opening

After meeting Jack Schwartz, I promptly made up my mind to abandon theoretical (FOCS/STOC) computer science and embark upon a new career, combining mathematics, computer science and robotics. Jack promised to help. During my first year at Courant, a suspiciously simple-looking but thorny robotics problem kept popping up at our lunch and dinner conversations. Eventually, it led to the discovery of a surprisingly intimate relation between robot grasping and an elegant theorem due to Constantin Carathéodory.

Later, it dawned on me that Jack might have been mentoring me on the art of blending mathematics, computer science and robotics (or for that matter, any other applied field). After that experience, it has never been too difficult to be a "Bud-of-all-trades." But that's only a small part of all I have learned from Jack.

Carathéodory's theorem (belonging to a larger family of Helly-type theorems) [2, 3] is usually stated as follows: *If a point p of $\mathbb{R}^d$ lies in the convex hull of a set X, there is a subset $Y = \{y_1, \ldots, y_{r+1}\}$ of X consisting of $d+1$ or fewer points such that p lies in the convex hull of Y. Equivalently, p lies in an r-simplex with vertices in X, where $r \leq d$.*

Carathéodory proved his theorem in 1907 [2] for the case when X is compact. In 1914 Steinitz expanded Carathéodory's theorem for any sets X in $\mathbb{R}^d$. If one visualizes Carathéodory's theorem in 2 dimensions, it can be seen to state the existence of a triangle consisting of points from X that encloses any point enclosed by X—the theorem can be made constructive (Fig. 1). For instance, when X has finitely

B. Mishra (✉)
Courant Institute, 251 Mercer St., New York, NY, USA
e-mail: mishra@nyu.edu

M. Davis, E. Schonberg (eds.), *From Linear Operators to Computational Biology*,
DOI 10.1007/978-1-4471-4282-9_7,

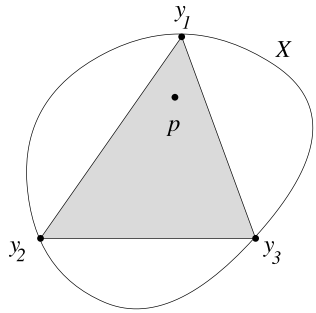


Fig. 1 Example of Carathéodory's theorem for $d = 2$

many points, a triangulation of X's convex-hull will have a triangle containing any point in the convex hull of X. Consider a set $X = \{(0,0), (0,1), (1,0), (1,1)\}$, a subset of $\mathbb{R}^2$. The convex hull of this set is a square. Consider now a point $p = (1/4, 1/4) \in \operatorname{conv} X$. We can then construct a set $\{(0,0), (0,1), (1,0)\} = Y$ ($|Y| = 3$), the convex hull of which is a triangle and encloses p. Another set, containing p in its convex hull, of course is $\{(0,0), (1,1)\} = Y'$ ($|Y'| \leq 3$), but it presents a degenerate example. Similar arguments extend the theorem to higher dimensions.

Carathéodory was born in Berlin in 1873, to a prominent Greek family, closely involved with the Ottoman Empire. After attending a variety of schools in Belgium, Carathéodory finally enrolled as a student of artillery and engineering at the Belgian Military Academy in 1891, where he received extensive technical training in engineering. It covered some antiquated calculus but also courses in mechanics, probability, astronomy, geography, and thermodynamics. His lifelong fascination with descriptive geometry, a core area of engineering mechanics, began at the academy.

When in 1897 an annual Nile flood interrupted his job as an engineer, to kill time, he started studying mathematics: Jordan's *Cours d'Analyze*, Salmon's book on conics, etc. During this process, he became enamored with pure mathematics and decided—to the chagrin of his entire extended aristocratic family—to relinquish engineering. Soon, he was attending lectures in pure mathematics by Schwartz, Fuchs, and Frobenius, and on symbolic logic by Carl Friedrich Stumpf.

Carathéodory came to Göttingen in the summer of 1902, and met Zermelo, Born, Blumenthal, the Youngs (William and Grace), Minkowski, Klein and Hilbert. When he proved the theorem presented earlier, with its centrality in convexity theory, neither he nor his colleagues could foresee any possible application of the theorem—physical or otherwise. The purity (rather absence of any obvious usefulness) seemed to have delighted Carathéodory. Five decades later, when Carathéodory's work began to find applications in economic theories of markets and equilibria, they were dismissed as non-physical (hence artificial) applications, not affecting the utter purity with which Carathéodory had held his theorem.

However, our initial work (started with Schwartz and Sharir) [11] and its sequels [5, 12, 20] showed how the theorem can be directly related to static problems in classical mechanics and applied for robots to plan "grasping," "work-holding" and "fixturing." With that, alas, whatever purity (imagined or real) Carathéodory might have bestowed on his theorem, seemed to have evaporated irrevocably. On the other hand, this might be seen as yet another example of marvels of mathematics: "The Unreasonable Effectiveness of Mathematics in the Natural Sciences [21]," of which Wigner wrote about so eloquently. "The miracle of the appropriateness of the language of mathematics for the formulation of the laws of physics is a wonderful gift, which we neither understand nor deserve. We should be grateful for it…"

Jack seemed to have been rather skeptical of the claims of mathematics' unreasonable effectiveness. "…The intellectual attractiveness of a mathematical argument, as well as the considerable mental labor involved in following it, makes mathematics a powerful tool of intellectual prestidigitation—a glittering deception in which some are entrapped, and some, alas, entrappers," Jack wrote in his 1986 essay entitled "*The Pernicious influence of Mathematics on Science* [18]."

In the following few paragraphs, I will outline the intellectual prestidigitation necessary to claim the robot grasping problem as solved—with its elegant reformulation in convexity theory [3] and computational geometry [4, 16]. It is worth pondering how dexterously the assumptions might have been manipulated to bring about this mental entrapment—a reach exceeding our grasp, perhaps. But then, can we rebuild a more realistic theory and algorithms for robot grasping, which would also include hand design as well as kinematics, dynamics and control in their formulations? Few such ideas have been explored preliminarily and tentatively, as in the paradigm of "reactive robotics," a topic to which we will return eventually.

2 Gripping

Imagine an idealized dextrous hand, consisting of several independently movable force-sensing fingers. These fingers move as points in three-dimensional space. The problem of *grip selection* for an object is to study how to hold that object in equilibrium with point fingers—in the absence of static friction between the surface of the object and the fingers. Since the fingers are assumed to be point fingers, a finger can only apply a force on the object along the surface-normal at the point of contact, directed inward.

When the shape of the object is precisely known, the problem of *grip selection* reduces to that of choosing a set of GRIP POINTS and a set of associated FORCE TARGETS. We may then ask two questions:

- *Can an arbitrary object be gripped with a finite number of fingers*?
- *If so, what are the grip points and the magnitudes of the forces exerted by the fingers (force targets) for such a grip*?

From elementary study of *statics* in classical mechanics, we know how an object in equilibrium can be characterized. We may think of the forces as *polygenic* (the force/torques applied at the fingers are generated by some actuators whose mechanics need not concern us). Equilibrium can be characterized by the *resultant* force and torque equation, as in the classical Newtonian mechanics.

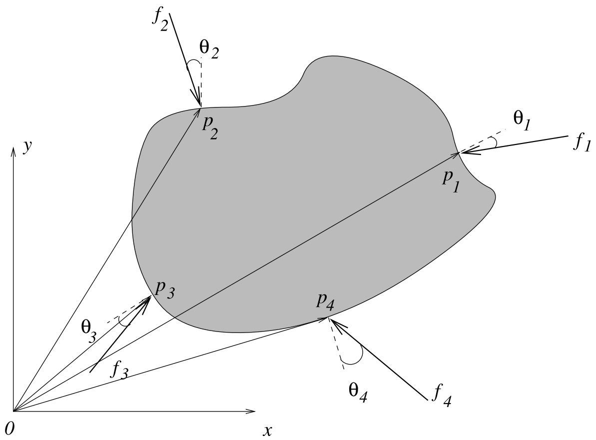


Fig. 2 A planar object subject to four forces f_1, f_2, f_3 and f_4

Consider a rigid body subject to a set of external polygenic forces $f_1, \ldots, f_k$, applied respectively at the points $p_1, \ldots, p_k$, as in Fig. 2. Then the necessary and sufficient condition for the rigid body to be in equilibrium is that *the resultant force and the resultant torque must be null vectors*. In mathematical notations, this condition can be stated as follows:

$$\sum_{i=1}^{k} f_i = 0 \quad \text{and} \quad \sum_{i=1}^{k} p_i \times f_i = 0,$$

where the cross product $\tau = p \times f$ gives a torque.[1]

Thus, in order to hold an object in equilibrium with a multi fingered hand (say, with k fingers), we need to place these fingers at points $p_1, \ldots, p_k$ on the boundary of the objects and apply forces $f_1, \ldots, f_k$ in such a manner that the equilibrium condition is satisfied.

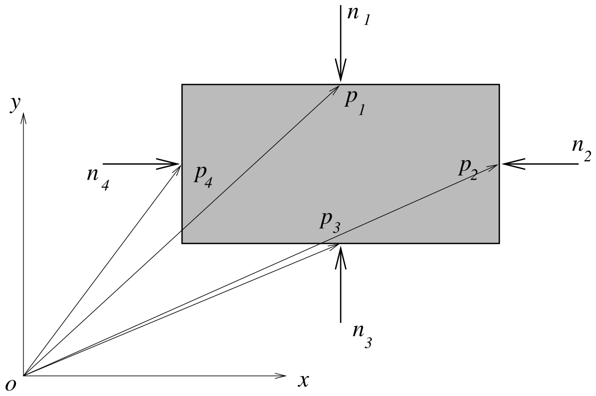
For example, consider a planar rectangular object with four grip points at the mid points of the edges (shown in Fig. 3). In this example, let the grip points be denoted as p_1, p_2, p_3 and p_4 and the respective unit surface normals as n_1, n_2, n_3 and n_4. Then we wish to determine if there are four scalar quantities α_1, α_2, α_3 and α_4 such that

$$\begin{aligned}
&\alpha_1 n_1 + \alpha_2 n_2 + \alpha_3 n_3 + \alpha_4 n_4 = 0 \\
&\alpha_1(p_1 \times n_1) + \alpha_2(p_2 \times n_2) + \alpha_3(p_3 \times n_3) + \alpha_4(p_4 \times n_4) = 0 \\
&\quad \alpha_1 \geq 0, \alpha_2 \geq 0, \alpha_3 \geq 0, \alpha_4 \geq 0 \text{ and not all 0.}
\end{aligned}$$

[1]The cross product $\tau = p \times f$ is defined as

$$\begin{aligned}
\tau_x &= p_y f_z - p_z f_y, \\
\tau_y &= p_z f_x - p_x f_z, \quad \text{and} \\
\tau_z &= p_x f_y - p_y f_x.
\end{aligned}$$

Fig. 3 A planar rectangular object with designated grip points $\{p_1, p_2, p_3, p_4\}$



Note that, for this example, any choice of $\alpha_1 = \alpha_3$ and $\alpha_2 = \alpha_4$ will satisfy the conditions (assuming that at least two of them are nonzero and all of them are non-negative). In particular, we could have chosen all the α's to be $1/4$!

To make matters little more abstract, we may define a *wrench map*, Γ, taking a point on the boundary of the object B to a point in the d-dimensional wrench space $\mathbb{R}^d$. Note that the term wrench space is used to denote a vector space consisting of all the wrenches. Its dimension d is 1, 3 or 6, depending on whether the object belongs to 1, 2 or 3-dimensional space.

$$\begin{aligned}\Gamma : \partial B &\to \mathbb{R}^d \\ : p_i &\mapsto (n_i, p_i \times n_i).\end{aligned}$$

Thus the wrench map Γ maps a point $p_i \in \partial B$ on the boundary of the body B to a wrench (a force/torque combination) that would be created if we apply a unit normal force directed inward at the point p_i. Then the feasibility of a positive grip can be expressed in terms of the existence of a solution of the following system of linear equations and inequalities:

$$\begin{aligned}\sum_{i=1}^{k} \alpha_i \Gamma(p_i) = 0 \\ \alpha_i \geq 0, \quad i = 1, \ldots, k, \\ \sum_{i=1}^{k} \alpha_i = 1.\end{aligned}$$

The last condition is added only for convenience. Geometrically, we were then asking if some convex combination of the $\Gamma(p_i)$'s would yield the null vector. More compactly,

$$0 \in \text{convex hull}\big(\Gamma(p_1), \ldots, \Gamma(p_k)\big)?$$

If the answer to the preceding question is yes, then we can hold the object in equilibrium with the given grip points by applying forces whose magnitudes simply correspond to the coefficients used in the convex combination to express the null vector.

2.1 Closure Grasps

One of the simplest problems in grasping theory can be stated as below:

Given: An arbitrary rigid 3-dimensional object B and some number k.
Determine: Whether one can choose k (finite) grip points, $\{p_1, p_2, \ldots, p_k\} \subseteq \partial B$ on the boundary of B such that the object can be grasped (positively) by placing fingers at those grip points.

$$\bigl(\exists ?\{p_1, \ldots, p_k\} \subseteq \partial B\bigr) \quad \bigl[0 \in \operatorname{conv}\bigl(\Gamma(p_1), \ldots, \Gamma(p_k)\bigr)\bigr].$$

The answer to the problem turns out to be "yes" and the necessary number of fingers is SEVEN (and not five!).

The proof proceeds in three simple steps:

STEP 1: Show that

$$0 \in \operatorname{conv} \Gamma(\partial B),$$

where $\Gamma\colon \partial B \to \mathbb{R}^6 : p \mapsto (n, p \times n)$. This is a simple consequence of the fact that an object under uniform pressure remains in equilibrium. The proof of this claim can be given rigorously using the *Divergence theorem of Gauss*.

STEP 2: By Carathéodory's theorem

$$\bigl(\exists \bigl\{\Gamma(p_1), \ldots, \Gamma(p_k)\bigr\} \subseteq \Gamma(\partial B)\bigr) \quad \bigl[k \le 7 \text{ and } 0 \in \operatorname{conv}\bigl(\Gamma(p_1), \ldots, \Gamma(p_k)\bigr)\bigr].$$

Hence there are positive nonnegative scalar quantities $\alpha_1, \ldots, \alpha_k$ such that:

$$\alpha_1 n_1 + \cdots + \alpha_k n_k = 0,$$
$$\alpha_1(p_1 \times n_1) + \cdots + \alpha_k(p_k \times n_k) = 0.$$

STEP 3: The positive grip is then selected by choosing as grip points

$$\begin{aligned} \text{Grip Points } &= \{p_1, \ldots, p_k\} \subseteq \partial B, \\ \text{Force Magnitudes } &= \alpha_1, \ldots, \alpha_k, \end{aligned}$$

with k no larger than 7.

Similar arguments in the plane imply that one would need FOUR fingers. The number four is arrived at by taking the dimension of the wrench space and adding one to it, as implied by the Carathéodory's theorem. It is also instructive to examine a set of equilibrium grasps for three planar objects: a rectangle, a triangle and a disc (Fig. 4). First consider the grasps for the rectangle. Clearly, the grasps (a) and (d) are not as secure as (g)—a horizontal external force will break the grasp (a) and an external torque about the center of the rectangle will break the grasp (d). In comparison, the grasp (g) is immune to such external disturbances, provided of course that

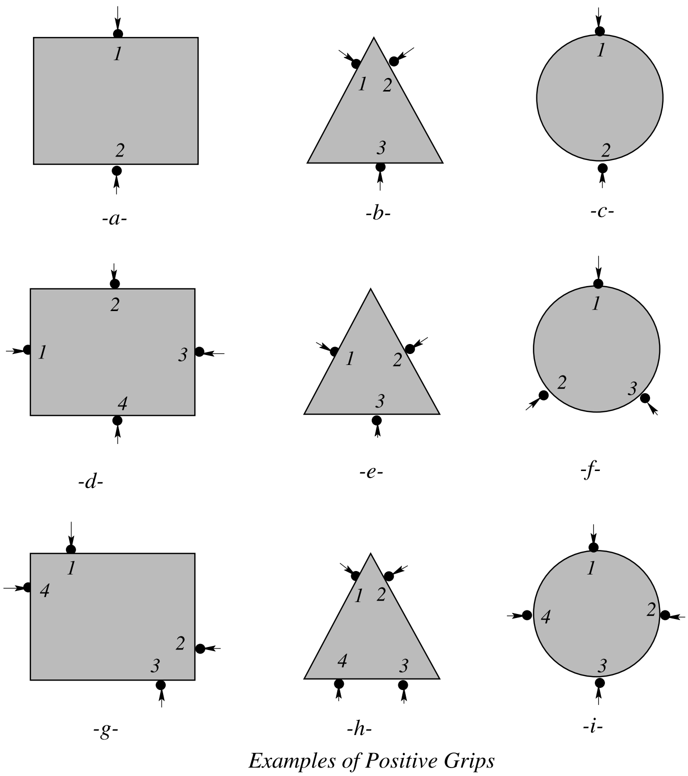


Fig. 4 Grasping planar objects

such disturbances are relatively small in magnitude. Similar examination will show that the grasp (h) is the most secure for a triangle. However, in the case of the disc, while the grasps (f) and (i) are better than (c), there is simply no way to resist an external torque about the center irrespective of how many fingers are used.

The kinds of secure grasps described in the preceding paragraph have been characterized as *closure grasps*. Furthermore, exactly those objects that do not allow closure grasps can also be characterized in purely geometric terms, and are referred to as *exceptional objects*. While we shall not go into a detailed description of closure grasps and exceptional objects (see [11]), it should suffice for the present purpose to say that the only planar bounded exceptional object is a disc and the only spatial bounded exceptional object is an object bounded by a surface of revolution.[2]

[2]If one allows unbounded objects then in 3-D, we have to include unbounded prisms and helical objects and in 2-D an unbounded strip of constant width. These objects in 3-D describe the so-called Reuleux pairs, studied almost a century ago.

2.2 *Synthesizing a Grasp*

At this point, it is natural for a roboticist to ask how one (a robot) can construct a grasp for a specific object and what sorts of computation this may entail. The answer turns out to be very interesting and shows a close connection of this problem to a classical algorithm, "*the simplex method*," used for solving linear programming problems.

Thus, suppose we have a polyhedral object with n faces. We proceed in a manner not very dissimilar from the ways we proved the existences of such a grasp. We first create a grasp with extremely large number of fingers: about $15n$ grip points, where n is the number of faces of the polyhedron. Next, step by step, we can eliminate one finger in each step while maintaining grasp as long as the number of grip points at the beginning of that step is strictly larger than the lower bound. The algorithm terminates when we are left with appropriate number of grip points (or fewer).

In order to understand the process by which the fingers are eliminated, we shall digress to consider an algorithmic approach to *algebraic manipulation with positive linear combinations*.

Given: A set of vectors $\{V_1, V_2, \ldots, V_l\} \subseteq \mathbb{R}^d$ and $V \in \mathbb{R}^d$ such that

$$\alpha_1 V_1 + \cdots + \alpha_l V_l = \alpha V$$
$$\alpha_i \geq 0, \alpha > 0, V \neq 0.$$

Find: A subset $m \leq d$ vectors

$$\{V_{i_1}, V_{i_2}, \ldots, V_{i_m}\} \subseteq \{V_1, \ldots, V_l\} \quad \text{and} \quad \alpha' > 0$$

such that

$$\alpha'_1 V_{i_1} + \cdots + \alpha'_m V_{i_m} = \alpha' V$$
$$\alpha'_i \geq 0 \ (\alpha' > 0, V \neq 0).$$

Reduction Algorithm

if $l \leq d$ then HALT;
else repeat

Choose d vectors from $\{V_1, \ldots, V_l\}$
(Say, the first d): $\{V_1, \ldots, V_d\}$
There are two cases to consider, depending on whether the vectors $V_1, \ldots, V_d$ are *linearly dependent* or not.

Case 1: $V_1, \ldots, V_d$ *are linearly dependent.*

We can write

$$\beta_1 V_1 + \cdots + \beta_d V_d = 0,$$

not all $\beta_i = 0$.
Assume that at least one $\beta_i < 0$ (otherwise, replace each β_i by $-\beta_i$ in the equation to satisfy the condition).

Let

$$\gamma = \min_{\beta_i < 0} (\alpha_i / \beta_i) < 0.$$

(For specificity, we may assume $\gamma = \alpha_1/\beta_1$.)
Put $\alpha_i' = \alpha_i - \gamma\beta_i$ for $1 \le i \le d$.
Hence by adding the equation $(\sum_{i=1}^{l} \alpha_i V_i = \alpha V)$ to $(-\gamma \sum_{i=1}^{d} \beta_i V_i = 0)$, we get

$$\alpha_2' V_2 + \cdots + \alpha_d' V_d + \alpha_{d+1} V_{d+1} + \cdots + \alpha_l V_l = \alpha V,$$

and by construction $\alpha_2', \ldots, \alpha_d' \ge 0$.

Case 2: *$V_1, \ldots, V_d$ are linearly independent.*
We can write

$$\beta_1 V_1 + \cdots + \beta_d V_d = V.$$

Assume that at least one $\beta_i < 0$ (otherwise, we have nothing more to do!).
Let

$$\gamma = \min_{\beta_i < 0} (\alpha_i / \beta_i) < 0.$$

(For specificity, we may assume $\gamma = \alpha_1/\beta_1$.)
Put $\alpha_i' = \alpha_i - \gamma\beta_i$ for $1 \le i \le d$, and $\alpha' = \alpha - \gamma > 0$.
Hence by adding the equation $(\sum_{i=1}^{l} \alpha_i V_i = \alpha V)$ to $(-\gamma \sum_{i=1}^{d} \beta_i V_i = -\gamma V)$, we get

$$\alpha_2' V_2 + \cdots + \alpha_d' V_d + \alpha_{d+1} V_{d+1} + \cdots + \alpha_l V_l = \alpha' V,$$

and by construction $\alpha_2', \ldots, \alpha_d' \ge 0$.

In algorithmic terminology, we can prove that "the reduction algorithm has a time complexity of $O(ld^3)$." In our grasping application, d will turn out to be a constant (=6) and l no more than $15n$.

Let us get back to our original question about grasping a polyhedron B with n faces. As hinted earlier, we shall start with a closure grasp of B using no more than $15n$ grip points. Assume that B is provided with a triangulation of each face, and

$$t_1, t_2, \ldots, t_N$$

is the set of triangles partitioning ∂B. For each triangle t_i, choose three non-collinear grip points p_{i_1}, p_{i_2} and $p_{i_3} \in t_i$ such that $(p_{i_1} + p_{i_2} + p_{i_3})/3$ is the centroid of t_i. In totality they will give us the initial $3N$ grip points. Using Euler's formula and some simple combinatorics, one can show that $N \le 5n - 12$ and the total number of grip points is no more than $15n - 36$ (see [11]).

Now, it can be shown that if one chooses p_{i_j}'s, $1 \le i \le N$, $j = 1, 2, 3$, as the grip points then they give rise to a closure grasp. In particular, we can see [11] (by using linear algebraic manipulations) that

$$\frac{\text{Area}\,(t_1)}{3}\Gamma(p_{1_1}) + \frac{\text{Area}\,(t_1)}{3}\Gamma(p_{1_2}) + \frac{\text{Area}\,(t_1)}{3}\Gamma(p_{1_3})$$
$$+\cdots+\frac{\text{Area}\,(t_N)}{3}\Gamma(p_{N_1}) + \frac{\text{Area}\,(t_N)}{3}\Gamma(p_{N_2}) + \frac{\text{Area}\,(t_N)}{3}\Gamma(p_{N_3}) = 0,$$

and that

$$\text{pos}\big(\Gamma(p_{1_1}), \Gamma(p_{1_2}), \ldots, \Gamma(p_{N_3})\big) = \mathbb{R}^6.$$

Henceforth, rewriting these grip points as $\{p_1, p_2, \ldots, p_l\}$, and the "area terms" as magnitude of coefficients: $\alpha_1, \alpha_2, \ldots, \alpha_l$, we have

$$\alpha_1\Gamma(p_1) + \alpha_2\Gamma(p_2) + \cdots + \alpha_l\Gamma(p_l) = 0, \tag{1}$$

where $\alpha_i > 0$. Furthermore, since

$$\text{lin}\big(\Gamma(p_1), \Gamma(p_2), \ldots, \Gamma(p_l)\big) = \mathbb{R}^6,$$

without loss of generality, assume that the first six wrenches are linearly independent, thus spanning the entire wrench space, i.e.,

$$\text{lin}\big(\Gamma(p_1), \ldots, \Gamma(p_6)\big) = \mathbb{R}^6.$$

Synthesizing a Equilibrium Grasp with Seven Fingers Let us now see how we can go from here to get a simple equilibrium grasp with no more than seven fingers. Note first that we can rewrite our equation 1 (for l-fingered grip) as

$$\frac{\alpha_1}{\alpha_l}\Gamma(p_1) + \cdots + \frac{\alpha_{l-1}}{\alpha_l}\Gamma(p_{l-1}) = -\Gamma(p_l),$$

where $\alpha_i > 0$ and $\Gamma(p_i) \in \mathbb{R}^6$. Now, we can use the "Reduction Algorithm" to find

$$\{p_{i_1}, p_{i_2}, \ldots, p_{i_m}\} \subseteq \{p_1, \ldots, p_{l-1}\}$$

satisfying the conditions below:

$$\alpha'_1\Gamma(p_{i_1}) + \cdots + \alpha'_m\Gamma(p_{i_m}) = -\alpha'\Gamma(p_l),$$

and $m \le 6$. Thus we have

$$\alpha'_1\Gamma(p_{i_1}) + \cdots + \alpha'_m\Gamma(p_{i_m}) + \alpha'\Gamma(p_l) = 0,$$

with $\alpha'_1 \ge 0, \ldots, \alpha'_m \ge 0$ and $\alpha' > 0$. Of course, this is our equilibrium grasp using no more than $m+1 \le 7$ fingers, placed at grip points $p_{i_1}, \ldots, p_{i_m}, p_l$ with associated force magnitudes $\alpha'_1, \ldots, \alpha'_m, \alpha'$.

As analyzed earlier, our grasping algorithm could be shown to take $O(n)$ time with a constant in the complexity growing as $O(d^3)$.

A few years later, in 1990, Papadimitriou and his colleagues revisited the problem [7], and proved (without appealing to Carathéodory-like theorems) similar bounds on number of fingers. They also showed how to turn the algorithmic problem into a linear programming problem in certain special cases (e.g., planar convex objects or non-convex objects with bounded number of concave angles). As Megiddo had shown that these linear programming problems have linear time solutions, when the dimension d is treated as a constant, Papadimitriou et al. had also demonstrated that grasping could be done in linear time—at least, for certain special geometries.

Thus, it seemed that any comprehensible formulation of the grasping problem would unavoidably appeal to convexity theory (to Carathéodory's dismay). However, the complexity of the algorithms (thus applicability) depended crucially on the exact formulation—going the Megiddo [9] route meant that the algorithm would have an $O(d2^{O(d)}n)$ time complexity. Big Ouch!

However, except for few such theoretical quibbles, the grasping problem had been more or less solved and with *panache blanc*—or so we thought.

3 Groping

In an article [1] appearing about a decade later, it was lamented that, "Notwithstanding the great effort spent, and the [impressive] technological and theoretical results achieved by the robotics community in building and controlling dexterous robot hands, the number of applications in the real-world and the performance of such devices in operative conditions should be frankly acknowledged as not yet satisfactory. In particular, the high degree of sophistication in the mechanical design prevented so far dexterous robotics hand to succeed in applications where factors such as reliability, weight, small size, or cost, are at a premium. One figure partially representing such complexity is the number of actuators, which ranges between 9 and 32 for hands considered above. Further reduction of hardware complexity, even below the theoretically minimum number of 9, is certainly one of the avenues for overcoming this impasse." Thus, while the elegant theory we had developed gave many insights into how to create a field of dexterous manipulation, the industrial (or elsewhere) applications of robot hands have never really embraced the needed complexity. Instead, simple parallel-jaw grippers still rules the manufacturing world.

What could be done? How can we connect the mathematical theories with applications. Jack [18] had worried that, "Related to this deficiency of mathematics ... is the simple-mindedness of mathematics—its willingness to elaborate upon any idea, however absurd; to dress scientific brilliancies and scientific absurdities alike in the impressive uniform of formulae and theorems. Unfortunately however, an absurdity in uniform is far more persuasive than an absurdity unclad." We may wish to return to the various underlying assumptions of the grasping theories to separate the ones that are apt from those that are absurd.

Setting aside the issues of finger properties (friction, softness, compliance, etc.) [15], object properties (degrees of freedom, deformability, elasticity, etc.), closure grasps [12], grasp quality [5, 10], grasp stability, robustness, gaiting, grasp planning, hand kinematics, dynamics and control, one may just focus on one issue: why simple hands have done so much better. For instance, a parallel-jaw gripper works well only with objects with antipodal grip points and of simple geometry (e.g., $2\frac{1}{2}$-dimensional), and yet it is ubiquitous.

In a recent publication, Matt Mason and colleagues [8] asked, "While complex hands offer the promise of generality, simple hands are more practical for most robotic and telerobotic manipulation tasks, and will remain so for the foreseeable future. This raises the question: how do generality and simplicity trade off in the design of robot hands?" Their answer was to focus on using "knowledge of stable grasp poses as a cue for object localization." Yet, a different approach is to integrate the hand design, grasp control algorithm and grasp selection into one framework—as done in our work on "Reactive Robotics."

With my students and colleagues, we invented a clever parallel-jaw reactive gripper, and showed how to drive its grasp control algorithm by a set of discrete rules, that simply translate certain boolean conditions determined by the sensors into immediate ("reactive") actions of the actuators. The gripper could very quickly grasp any convex object in just two antipodal points and enjoys many robustness properties.

Similar ideas can be extended to three-finger-hands, such as the commercially-available Barrett hand [19], which is stiff, not frictionless and has 7 DOFs (degrees of freedom), four of which are active. The key idea is to use the local geometry of the object to find a set of grip points, while solving the local motion planning problem of getting the fingers to their grasping positions. The object is assumed to have a smooth boundary and be convex. The grasping algorithm can be shown to be "non-disturbing," (i.e., it does not affect the object's location or motion, until it is grasped).

The gripper consists of 3 fingers, simplified by the constraint to have their endpoints move in a plane. The fingers move arbitrarily, but their order (around the triangle they form) remains fixed. The "reactive 3-finger hand" searches for three grip points by following the object boundary until some geometric condition is satisfied. Each finger is equipped with simple sensors that allow them to follow the object's contour and can determine the angle of the object boundary (it is close to). The sensors that may be considered are: (1) an omni-directional distance sensor (measuring distance to the object in any direction), and (2) an angle sensor (measuring angle of the object boundary at the closest point). Such sensors can be easily built using a pair of simple IR reflective sensors.

The key idea behind the grasping algorithm is for the hand to discover "reactively" a locally minimal area triangle that encloses the object. The grip points can be determined from this triangle via a theorem of Klee [6]: *if T has a locally minimum area among all triangles containing a convex body B, then the midpoints of each side of T touches B*. It can also been shown that [17] if the midpoint of an edge e of a triangle does not touch the object then e can be perturbed such that its midpoint after perturbation lies inside the original triangle. This perturbation reduces the triangle area. Thus, the grasping algorithm has two phases:

- *Phase 1*: Find a triangle that contains the object, by, say, closing the fingers along three concurrent lines spaced at equal angles ($120°$) from each other, until they come to close proximity of the object boundary. If the triangle is not "bounded," the hand can fix it by a small perturbing rotation.
- *Phase 2*: Find a locally minimal triangle enclosing the object. The basic step requires a finger to do the following: the finger divides its triangle edge into two segments and moves in the direction of the larger segment. Consequently, both

the ratio of the larger segment to smaller segment for each edge will be reduced; so will the area of the enclosing triangle. It may seem that each finger has to move one at a time synchronously, but that is really not necessary, if certain care is taken as one approaches convergence to a grasp.

The reactive hand works (like an analog computer) by minimizing a potential function defined by the triangle area, and from this an appropriate notion of stability and robustness can be derived. Once the triangle is determined, since the lines through edge mid-points and perpendicular to the corresponding edges are concurrent at the point which is at the center of the circumscribing circle of the triangle, we can use these midpoints to get a planar (force-closure) grasp without relying on friction. If the object and finger-tips have some static friction (which is true of Barrett hands) then the resulting grasp also has a planar torque closure. More details can be found in [13, 14].

What is more interesting is the way the reactive gripper may be anthropomorphized: The corresponding reactive algorithm will appear to have three fingers groping around an object blindly (as they have not performed any a priori computation on a model of the object) until deciding on a grip. What separates groping from gripping? Isn't groping just an analog computation performed by the finger sensors and actuators to solve an optimization problem (namely, the minimal point of a potential function, determined by the area of an enclosing triangle)? So then what exactly is a computation in robotics? How does a robotic algorithm separate sensing, planning and actuation?

I wish I knew how Jack might have thought about these questions ....

4 Closing

Over the last year, I have realized how much we all miss Jack, his polymathic and eclectic conversational topics and gentle mentoring. With Jack's death, Courant seems to have lost a significant part of its basic character.

Soon after my arrival at Courant in 1985, Jack had walked me over to the intersection of Mercer and fourth, and given me my first and the shortest tour of Manhattan: At the time, there was a Swensen's right across from Courant, a music place called Bottom Line on fourth and a Yeshiva in the opposite corner which still stands. He pointed out that without going too far I could now have food, religion, music and mathematics—that was all the Manhattan I needed. Jack never ever mentioned religion after that.

Later Jack took it upon himself to introduce me to all sorts of exotic food and information: Alexander's campaign route through Bactria and Parthia to India, explained over Matzo ball soup in Second Avenue Deli; Ferdowsi's Shahnameh and its significance to Persian culture over some ultra-hot vindaloo in Curry in a Hurry and how to design a balloon robot over many many servings of twice-cooked pork. He also told me that he considered himself a gourmet diner who likes to try the best and the tastiest from every cuisine—that too was his style in science and mathematics.

Jack seemed to have found interesting mathematics in almost everything: how to move a piano, how to grasp a greasy pig, how to manage personal relationships, how to trade in foreign exchange markets, how to visualize a genome, how to write music to be read by a computer, how to use cartoons to explain special theory, how to become immortal and a zillion other things like that.

With Jack, everything led to voracious gourmet feasting. Jack always skipped his appetizers, and never lingered on for the desserts.

Acknowledgements The paper has improved considerably following many insightful suggestions from several colleagues: most notably, S. Kleinberg and E. Schonberg of NYU, M. Mason of CMU and M. Wigler of CSHL.

References

1. Bicchi, A.: Hands for dextrous manipulation and robust grasping: a difficult road towards simplicity. IEEE Trans. Robot. Autom. **16**(6), 652–662 (2000)
2. Carathéodory, C.: Über den Variabilitätsbereich der Koeffizienten von Potenzreihen die gegebene Werte nicht annehmen. Math. Ann. **64**, 95–115 (1907)
3. Danzer, L., Grünbaum, B., Klee, V.: Helly's theorem and its relatives. Convexity **7**, 101–180 (1963)
4. Edelsbrunner, H.: Algorithms in Combinatorial Geometry. Springer, Berlin (1987)
5. Kirkpatrick, D., Mishra, B., Yap, C.: Quantitative Steinitz's theorem with applications to multifingered grasping. Discrete Comput. Geom. **7**, 295–318 (1992)
6. Klee, V.: Facet Centroids and Volume Minimization (1986)
7. Markenscoff, X., Ni, L., Papadimitriou, C.: The geometry of grasping. Int. J. Robot. Res. **9**, 61–74 (1990)
8. Mason, M., Srinivasa, S., Vazquez, A.: Generality and simple hands. In: International Symposium on Robotics Research (2009)
9. Megiddo, N.: Linear programming in linear time when the dimension is fixed. J. ACM **31**(1), 114–127 (1984)
10. Mishra, B.: Grasp Metrics: Optimality and Complexity, pp. 137–166. AK Peters, Wellesley (1995)
11. Mishra, B., Schwartz, J., Sharir, M.: On the existence and synthesis of multifinger positive grips. Algorithmica **2**, 541–558 (1987)
12. Mishra, B., Silver, N.: Some discussion of static gripping and its stability. IEEE Trans. Syst. Man Cybern. **19**, 783–796 (1989)
13. Mishra, B., Teichmann, M.: Reactive algorithms for 2 and 3 finger grasping. In: Proceedings of the 1994 International Workshop on Intelligent Robots and Systems, IRS 94 (1994)
14. Mishra, B., Teichmann, M.: Reactive algorithms for grasping using a modified parallel jaw gripper. In: Proceedings of the 1994 IEEE International Conference on Robotics and Automation, ICRA 94 (1994)
15. Mishra, B., Teichmann, M.: The Power of Friction: Quantifying the 'Goodness' of Frictional Grasps, pp. 311–320. AK Peters, Wellesley (1997)
16. O'Rourke, J.: Computational Geometry in C. Cambridge University Press, Cambridge (1994)
17. O'Rourke, J., Aggarwal, A., Maddila, S., Baldwin, M.: An optimal algorithm for finding minimal enclosing triangles. J. Algorithms **7**(2), 258–269 (1986). doi:10.1016/0196-6774(86)90007-6
18. Schwartz, J.: The Pernicious Influence of Mathematics on Science, pp. 230–235. Springer, Berlin (2006)

19. Technolgies, B.: The Barrett Hand. http://www.barrett.com/robot/products-hand.html
20. Teichmann, M.: Grasping and fixturing: a geometric study and an implementation. Ph.D. thesis, New York University, New York (1995)
21. Wigner, E.: The unreasonable effectiveness of mathematics in the natural sciences. Commun. Pure Appl. Math. **13**, 1–14 (1960)

The Ref Proof-Checker and Its "Common Shared Scenario"

Eugenio G. Omodeo

Abstract In his later years, Jack Schwartz devoted much energy to the implementation of a proof-checker based on set theory and to the preparation of a large script file to be fed into it. His goal was to attain a verified proof of the Cauchy integral theorem of complex analysis.

This contribution to the memorial volume for Jack reflects that effort: it briefly reports the chronicle of his proof-checking project and highlights some features of the system as implemented; in an annex, it presents a proof scenario leading from bare set theory to two basic theorems about claw-free graphs.

> ...l'histoire géologique nous montre que la vie n'est qu'un court épisode entre deux éternités de mort, et que, dans cet épisode même, la pensée consciente n'a duré et ne durera qu'un moment. La pensée n'est qu'un éclair au milieu d'une longue nuit.
> Mais c'est cet éclair qui est tout.[1]
>
> (H. Poincaré, *La valeur de la science*, 1905)

1 Introduction

When I visited him at the New York University in June 2000, Jack Schwartz invited me to read what he called the "*common shared scenario*": a wide, carefully assembled, sequence of definitions, theorems, and proofs, leading from the bare rudiments

[1]... *geological history shows us that life is only a short episode between two eternities of death, and that, even inside that episode, conscious thinking has only lasted and will last only one moment. Thought is but a flash in the middle of a long night.*

But this flash is all we have.

An epigraph (in French) drawn from here appears at the beginning of the 2nd part of Dunford–Schwartz's *Linear Operators*.

E.G. Omodeo (✉)
Dipartimento di Matematica e Informatica, Università di Trieste, Via Valerio, 12/1, 34127 Trieste, Italy
e-mail: eomodeo@units.it

M. Davis, E. Schonberg (eds.), *From Linear Operators to Computational Biology*,
DOI 10.1007/978-1-4471-4282-9_8,

of set theory to the beginning of mathematical analysis. Proofs began to be gappy after a few hundred pages, and then totally absent, but the flow of definitions and theorem statements went on, to culminate in the definition of complex line integral and in the following claim, closely akin to the celebrated *Cauchy integral theorem* of complex analysis:

$$\begin{aligned}&\mathsf{Is\_analytic}_{\mathbb{CF}}(\mathsf{F}) \;\rightarrow\; \Big\langle \exists \varepsilon \in \mathbb{R} \mid \varepsilon >_{\mathbb{R}} \mathbf{0}_{\mathbb{R}} \;\&\; \big\langle \forall \mathsf{crv}_1, \mathsf{crv}_2 \mid \\ &\quad \mathsf{Is\_CD\_curv}(\mathsf{crv}_1, \mathbf{0}_{\mathbb{R}}, \mathbf{1}_{\mathbb{R}}) \;\&\; \mathsf{Is\_CD\_curv}(\mathsf{crv}_2, \mathbf{0}_{\mathbb{R}}, \mathbf{1}_{\mathbb{R}}) \;\& \\ &\quad \mathsf{crv}_1 \restriction \mathbf{0}_{\mathbb{R}} = \mathsf{crv}_1 \restriction \mathbf{1}_{\mathbb{R}} \;\&\; \mathsf{crv}_2 \restriction \mathbf{0}_{\mathbb{R}} = \mathsf{crv}_2 \restriction \mathbf{1}_{\mathbb{R}} \;\& \\ &\quad \big\langle \forall \mathsf{x} \in \mathsf{Interval}(\mathbf{0}_{\mathbb{R}}, \mathbf{1}_{\mathbb{R}}) \mid \varepsilon \geqslant_{\mathbb{R}} \big|\mathsf{crv}_1 \restriction \mathsf{x} -_{\mathbb{C}} \mathsf{crv}_2 \restriction \mathsf{x}\big|_{\mathbb{C}} \big\rangle \;\rightarrow \\ &\qquad\qquad\qquad\qquad \oint_{\mathbf{0}_{\mathbb{R}}}^{\mathbf{1}_{\mathbb{R}}} (\mathsf{F}, \mathsf{crv}_1) = \oint_{\mathbf{0}_{\mathbb{R}}}^{\mathbf{1}_{\mathbb{R}}} (\mathsf{F}, \mathsf{crv}_2) \big\rangle \Big\rangle \,.\end{aligned}$$

This was followed by the statement of Cauchy's integral formula, and by the following conclusive comment:

> *Beyond this point, the number of steps of definition needed to reach any concept, say, of classical functional analysis can be estimated by counting the number of definitions needed to reach the corresponding point in any standard reference on this subject, e.g. Dunford–Schwartz.*

What charmed me as a monumental craft-work was, in Jack's own words, "an essential part of the feasibility study that must precede the development of any ambitious proof-checker" [3, p. 229].—Eventually it would also serve as a testing-bench for the concrete implementation, in sight of which Jack had begun to cast a significant piece of mathematics in formal detail, constantly asking himself whether a computer program could conceivably process and validate every single step.

At the time, Jack debated a proof-modularization mechanism named 'theory', which would enable one to "avoid repeating similar steps when the proofs of two theorems are closely analogous" and to "conceal the details of a proof once they have been fed into the system and successfully certified". 'Theories' added a touch of second-order logic capability to the first-order system of set theory underlying the scenario; and their prominence among Jack's concerns revealed how large was the scale of the goals he had in mind.

Jack and I made plans to write a book on computational logic together with Domenico Cantone, who was about to visit New York in his turn; actually, it is under his solicitation that, short after my departure, Jack set ahead the implementation work, speedily bringing into existence the proof-checker Referee, also known as Ref, or as ÆtnaNova. Ref is written in the SETL programming language [7], save for a small part written in PHP, which makes it accessible on the Web.

Year after year, I witnessed the progresses of Ref, of the book (now posthumously published as [6]), and of the main proof scenario, titled *The Formal Foundations of Mathematical Analysis*. When I was his guest, Jack left his many other interests aside, and gave absolute priority to the work on Ref, showing unbelievable resources of energy in debugging it and in "scrabbling"—as he used to say—the scenario.

I became myself fluent in the formal proof system underlying Ref, and contributed variously to the scenario. Among others, when Jack became disappointed with the length of the proofs resulting from the definition

$$\mathbb{R} =_{\mathrm{Def}} \{\, c \subseteq \mathbb{Q} \mid \langle \forall y \in c,\ \exists z \in c \mid y < z \rangle \ \& \\ \langle \forall y \in c,\ \forall z \in \mathbb{Q} \mid z < y \rightarrow z \in c \rangle \,\} \backslash \{\emptyset, \mathbb{Q}\}$$

of the reals as Dedekind cuts (cf. [8, p. 43]), I was invited to formalize anew the chapter about the field of reals, with Cantor's alternative definition that sees real numbers as equivalence classes of rational Cauchy sequences:[2]

$$\begin{aligned}
\mathsf{Seq}_{\mathbb{Q}} &=_{\mathrm{Def}} \{\mathsf{f} \subseteq \mathbb{N} \times \mathbb{Q} \mid \mathbf{domain}(\mathsf{f}) = \mathbb{N} \ \&\ \mathsf{Svm}(\mathsf{f})\},\\
\mathsf{Cau}_{\mathbb{Q}} &=_{\mathrm{Def}} \Big\{\mathsf{f} \in \mathsf{Seq}_{\mathbb{Q}} \mid \Big\langle \forall \varepsilon \in \mathbb{Q} \mid \varepsilon >_{\mathbb{Q}} \mathbf{0}_{\mathbb{Q}} \rightarrow \\
&\qquad \mathsf{Finite}\Big(\Big\{\mathsf{i} \cap \mathsf{j} : \mathsf{i} \in \mathbb{N}, \mathsf{j} \in \mathbb{N} \mid \big|\mathsf{f}{\upharpoonright}\mathsf{i} -_{\mathbb{Q}} \mathsf{f}{\upharpoonright}\mathsf{j}\big|_{\mathbb{Q}} >_{\mathbb{Q}} \varepsilon\Big\}\Big)\Big\rangle\Big\},\\
\mathsf{F} \approx_{\mathbb{QS}} \mathsf{G} &\leftrightarrow_{\mathrm{Def}} \Big\langle \forall \varepsilon \in \mathbb{Q} \mid \varepsilon >_{\mathbb{Q}} \mathbf{0}_{\mathbb{Q}} \rightarrow \\
&\qquad \mathsf{Finite}\Big(\Big\{\mathsf{x} : \mathsf{x} \in \mathbf{domain}(\mathsf{F}) \mid \big|\mathsf{F}{\upharpoonright}\mathsf{x} -_{\mathbb{Q}} \mathsf{G}{\upharpoonright}\mathsf{x}\big|_{\mathbb{Q}} >_{\mathbb{Q}} \varepsilon\Big\}\Big)\Big\rangle.
\end{aligned}$$

But I was reluctant to become over-committed to the Cauchy integral theorem; and tended, rather than working on the foundations of analysis, to belabor various asides (Stone representation theorem for Boolean algebra, finite state automata, correctness of the Davis–Putnam procedure, von Neumann's cumulative hierarchy, bisimulations, etc. [5, 9, 10]). On occasions, even Jack swerved from the royal road: when for example, "out of self-indulgence" as he said, he wrote proofs of Zorn's lemma and of the ultrafilter lemma.

His and my diversions may partly explain why Jack's original work program is still unachieved. A self-contained section of the "common shared scenario" appears in [6, Chap. 7]: this reaches in a small number of pages many results about ordinals, various properties of the transitive closure operation, transfinite induction, and then Zorn's lemma. A scenario accompanying this article presents new proofs, likewise certified correct by Ref, of two basic theorems about claw-free graphs.

2 Architecture of a Proof Scenario

The Ref verifier is fed script files, called scenarios, consisting of successive definitions, theorems, and auxiliary commands, which Ref either certifies as constituting a valid sequence or rejects as defective. In the case of rejection, the verifier attempts to pinpoint the troublesome locations within a scenario, so that errors can be located and repaired. Step timings are produced for all correct proofs, to help the user in spotting places where appropriate modifications could speed up proof processing.

The bulk of the text normally submitted to the verifier consists of theorems and proofs. Some theorems (and their proofs) are enclosed within so-called theories, whose external conclusions are justified by these internal theorems. This lets scenarios be subdivided into modules, which increases the readability and supports proof reuse.[3]

[2] Within these definitions: Svm means 'single-valued map', i.e. a function represented as a set of pairs; and $F{\upharpoonright}x$ designates the value resulting from application of a single-valued map F to an element x of the domain of F.

[3] This is the beginning of *A user's manual for the Ref verifier*, see http://setl.dyndns.org/EtnaNova/login/Ref_user_manual.html.

THEORY eq_classes(s, Eq(X, Y))
$\langle\forall v, w, z \mid \{v, w, z\} \subseteq \mathsf{s} \rightarrow \mathsf{Eq}(v, v)\ \&\ \big(\mathsf{Eq}(v, w)\ \&\ \mathsf{Eq}(z, w) \rightarrow \mathsf{Eq}(v, z)\big)\rangle$
$\Rightarrow$ (quot_Θ, $\mathsf{cl\_of}_\Theta$) -- *quotient-set and canonical embedding*
$\emptyset \notin \mathsf{quot}_\Theta\ \&\ \langle\forall b \in \mathsf{quot}_\Theta, \forall x \in b \mid x \in \mathsf{s}\ \&\ \mathsf{cl\_of}_\Theta(x) = b\rangle$
$\langle\forall x \in \mathsf{s} \mid x \in \mathsf{cl\_of}_\Theta(x)\ \&\ \mathsf{cl\_of}_\Theta(x) \in \mathsf{quot}_\Theta\rangle$
$\langle\forall x \in \mathsf{s}, \forall y \in \mathsf{s} \mid \mathsf{Eq}(x, y) \leftrightarrow \mathsf{cl\_of}_\Theta(x) = \mathsf{cl\_of}_\Theta(y)\rangle$
END eq_classes

Fig. 1 Split of a set into equivalence classes, specified as a reusable module

Like procedures in a programming language, Ref's theories have lists of formal input parameters (s and $\mathsf{Eq}(-,-)$ in the example of Fig. 1). Each THEORY requires its parameters to meet a set of assumptions (in the example at hand, the assumptions state that Eq behaves as an equivalence relation over s). When "applied" to a list of actual parameters that have been shown to meet the assumptions, a theory will instantiate several additional "output" symbols (such as the quot_Θ and $\mathsf{cl\_of}_\Theta(-)$ of Fig. 1)[4] standing for sets, predicates, and functions, and then supply a list of claims initially proved explicitly by the user inside the theory itself. These are theorems generally involving the new symbols. (In the example at hand, the new symbol quot_Θ designates the quotient set s/Eq, and the output function symbol $\mathsf{cl\_of}_\Theta(-)$ designates a *global* function, i.e. one whose operand ranges over all sets.)

For example, the following "invocation" of the THEORY 'eq_classes' shown in Fig. 1 produces the set $\mathbb{R}$ of all real numbers from the rational Cauchy sequences (cf. Introduction), along with the canonical mapping Cau_to_$\mathbb{R}$ sending every such sequence to its limit:

$$\begin{array}{l}\mathsf{APPLY}\ \langle\mathsf{quot}_\Theta : \mathbb{R},\ \ \mathsf{cl\_of}_\Theta : \mathsf{Cau\_to\_}\mathbb{R}\rangle \\ \qquad\qquad\qquad \mathsf{eq\_classes}\big(\mathsf{Eq(f,g)} \mapsto (\mathsf{f} \approx_{\mathbb{QS}} \mathsf{g}),\ \ \mathsf{s} \mapsto \mathsf{Cau}_{\mathbb{Q}}\big) \Rightarrow \\ \text{THEOREM. } \langle\forall \mathsf{f} \in \mathsf{Cau}_{\mathbb{Q}} \mid \mathsf{f} \in \mathsf{Cau\_to\_}\mathbb{R}(\mathsf{f})\ \&\ \mathsf{Cau\_to\_}\mathbb{R}(\mathsf{f}) \in \mathbb{R}\rangle\ \& \\ \qquad \langle\forall \mathsf{f} \in \mathsf{Cau}_{\mathbb{Q}}, \mathsf{g} \in \mathsf{Cau}_{\mathbb{Q}} \mid \mathsf{f} \approx_{\mathbb{QS}} \mathsf{g} \leftrightarrow \mathsf{Cau\_to\_}\mathbb{R}(\mathsf{f}) = \mathsf{Cau\_to\_}\mathbb{R}(\mathsf{g})\rangle.\end{array}$$

Besides defining $\mathbb{R}$ and Cau_to_$\mathbb{R}$, this invocation simultaneously acquires various facts about these two entities, while putting into oblivion other facts drawable from the THEORY, namely the information that $\emptyset \notin \mathbb{R}$, that $\bigcup\mathbb{R} \subseteq \mathsf{Cau}_{\mathbb{Q}}$, and that $\mathsf{Cau\_to\_}\mathbb{R}(x) = b$ always ensues from $x \in b \in \mathbb{R}$.

If taken alone, this down-to-earth example may be inadequate to illustrate the strength of the modularization construct THEORY: after all, we have simply stated that $\mathbb{R} =_{\mathrm{Def}} \mathsf{Cau}_{\mathbb{Q}}/\approx_{\mathbb{QS}}$. But we can, at times, raise considerably the import of a THEORY: look for example, in Fig. 2, at the much enhanced version of the THEORY just seen. Here Eq is no longer required to behave as an equivalence relation locally, i.e. over a set s, but *globally*, i.e. over the universe of all sets. Notwithstanding, as we will now discuss, one can select a representative from each equivalence class, thanks

[4]Such output symbols, whose meanings are specified inside the theory, carry the Θ subscript.

THEORY $\mathsf{circumscribed\_eq\_classes}(\mathsf{Eq}(\mathsf{X},\mathsf{Y}),\mathsf{R}(\mathsf{X}),\mathsf{c}(\mathsf{X}))$
$\langle\forall v,w,z \mid \mathsf{Eq}(v,v)\ \&\ (\mathsf{Eq}(v,w)\ \&\ \mathsf{Eq}(z,w) \rightarrow \mathsf{Eq}(v,z))\rangle$
$\langle\forall v \mid \langle\exists u \mid \mathsf{Eq}(u,v)\ \&\ \mathsf{R}(u)\rangle\rangle$
$\langle\forall v,u \mid \mathsf{Eq}(u,v)\ \&\ \mathsf{R}(u) \rightarrow u \in \mathsf{c}(v)\rangle$
$\Rightarrow (\mathsf{ch}_\Theta)$ -- *choice of an* $\in$*-minimal representative from each* Eq*-class*
$\langle\forall v \mid \mathsf{Eq}(\mathsf{ch}_\Theta(v),v)\rangle$
$\langle\forall v,w \mid \mathsf{Eq}(v,w) \leftrightarrow \mathsf{ch}_\Theta(v)=\mathsf{ch}_\Theta(w)\rangle$
$\langle\forall v,w \mid \mathsf{Eq}(v,w) \rightarrow v \notin \mathsf{ch}_\Theta(w)\rangle$
END $\mathsf{circumscribed\_eq\_classes}$

Fig. 2 Enhanced version of the selection of class representatives

to the criterion supplied by the parameters $\mathsf{R}(-)$ and $\mathsf{c}(-)$ for “circumscribing”, given any set X, a non-null set all of whose members are equivalent to X.

In developing the internals of this new THEORY, one should bear in mind that by von Neumann’s regularity axiom any nonnull set k owns an $\in$-minimal member; that is, an element $\mathbf{arb}(k) \in k$ such that $\mathbf{arb}(k) \cap k = \emptyset$. Thanks to the regularity axiom, ensuring that the membership relation is well-founded, a recursive definition such as

$$\mathsf{ult\_membs}(S) =_{\mathrm{Def}} S \cup \bigcup\{\,\mathsf{ult\_membs}(t) : t \in S\,\}$$

(where $\bigcup T =_{\mathrm{Def}} \{y : u \in T,\ y \in u\}$) makes sense (its base-case being $\mathsf{ult\_membs}(\emptyset) = \emptyset$), always producing a *set* as its result. Actually, Ref will accept without any ado this definition of the set of all “ultimate members” of S, precisely in this formulation. As should be intuitive, $\mathsf{ult\_membs}(S)$ consists of those y’s which can reach S through a membership chain in a finite number $n+1$ of steps: $y = y_0 \in y_1 \in \cdots \in y_n \in S$.

Inside $\mathsf{circumscribed\_eq\_classes}$ one can define $\mathsf{ch}_\Theta(\mathsf{X})$ to be an $\in$-minimal element of $\{w \in \mathsf{ult\_membs}(x_0) \mid \mathsf{Eq}(w,\mathsf{X})\}$, where $x_0 = \{u \in \mathsf{c}(\mathsf{X}) \mid \mathsf{Eq}(u,\mathsf{X})\ \&\ \mathsf{R}(u)\}$. Thus, even when the Eq-class of X is not a set, $\mathsf{ch}_\Theta(\mathsf{X})$ will be an $\in$-minimal element of this class, depending solely on the class and not on X.

The construction just outlined is paradigmatic of a method by which one often reduces a class of intimidating or unknown size first to a set x_1 and then to an $\in$-minimal member x_2 of x_1. In its most basic form, the method is applied to a property $P(Y)$ whatsoever, so that *a priori* the x’s for which $P(x)$ holds might form a proper class κ; anyway, unless κ is void, we can pick an x_0 in it, then consider the set $x_1 = \{w \in \mathsf{ult\_membs}(\{x_0\}) \mid P(w)\}$ (plainly nonnull, since $x_0 \in x_1$), and then put $x_2 = \mathbf{arb}(x_1)$. Thus $P(x_2)$ will hold, whereas $\neg P(z)$ holds for any $z \in x_2$.

Once cast in the form of a THEORY, this construction gives us a *transfinite induction principle*: seen here, rather than as a new inference rule, as a mechanism enabling us to construct an entity which promises to play a key role in a refutation. Its exploitation goes as follows: Suppose we must prove $\langle\forall x \mid Q(x)\rangle$. Arguing by contradiction, assume that $\neg Q(x_1)$; by actualizing the input parameter $P(Y)$ of transfinite induction as $\neg Q(Y)$, get an $\in$-minimal x_2 for which $\neg Q(x_2)$ holds.

Through details that depend on the peculiarities of $Q(Y)$, strive to get a contradiction from the alleged minimality of x_2.

Let us now look at the issue of specifying transfinite induction as a 'theory' to be placed among the exordia of a brand new scenario, relying on very few preexisting theorems and definitions. I suspect that Jack found himself at odds with the definition of ult_membs seen earlier, because he momentarily resorted to an alternative but equivalent notion. Indeed, that definition is not readily usable in an almost empty scenario; and so he adopted something akin in spirit to the following characterization of the *transitive closure* of a set S:

$$\mathsf{trCl}(S) = \bigcup_{i\in\mathbb{N}} \mathsf{descs}_i(S), \quad \text{where}$$
$$\mathsf{descs}_0(S) = S, \quad \text{and} \quad \mathsf{descs}_{j+1}(S) = \bigcup \mathsf{descs}_j \quad \text{for all} \;\; j \in \mathbb{N}.$$

But, unfortunately, neither the set $\mathbb{N}$ of all integers, nor this form of recursive definition, are available at the outset: one only knows (from the axioms) of the existence of an infinite 'ur-set' s_∞ meeting the conditions $\mathsf{s}_\infty \neq \emptyset$, and $\langle \forall x \in \mathsf{s}_\infty \mid \{x\} \in \mathsf{s}_\infty \rangle$. A formally impeccable surrogate of the above must hence be specified in slightly more roundabout terms:

$$\mathsf{trCl}(S) =_{\mathrm{Def}} \{\, x : \, i \in \mathsf{s}_\infty, \; x \in \mathsf{descs}(i, S) \,\}, \quad \text{where } \mathsf{descs}(I, S) =_{\mathrm{Def}}$$
$$\textbf{if } I = \textbf{arb}(\mathsf{s}_\infty) \textbf{ then } S \textbf{ else } \bigcup \textbf{arb}(\{\mathsf{descs}(j, S) : \; j \in I \mid j \in \mathsf{s}_\infty\}) \textbf{ fi}.$$

The construction just seen is so ingenious and so ugly that it well deserves being encapsulated and concealed inside a THEORY which only shows, of it, what is relevant for subsequent use; but we would miss an opportunity to increase reusability if we did not generalize it somewhat. In fact the reader can find in [6, pp. 378–386] a THEORY of 'reachability' which gets in input a digraph $G = (V, E)$ and produces as output the function sending every node, i.e. element υ of V, to the set of all nodes reachable from υ through paths of G. The digraph can be 'big', in the sense that its nodes might form a proper class—e.g., the universe of all sets—and its edges a proper class of pairs—e.g., the converse $\ni$ of membership. Actually, a very simple assumption suffices for a generalized notion of the set of all descendants; namely, that the children (i.e. immediate descendants) of each node form a set.

3 Definition Mechanisms

Definitions serve various purposes. At their simplest they are merely abbreviations which concentrate attention on interesting constructs by assigning them names which shorten their syntactic form. (But of course the compounding of such abbreviations can change the appearance of a discourse completely, transforming what would otherwise be an exponentially lengthening welter of bewildering formulae into a sequence of sentences which carry helpful intuitions.) Beyond this, definitions serve to 'instantiate', that is, to introduce the objects whose special properties are crucial to an intended argument. Like the selection of crucial lines, points, and circles from the infinity of geometric elements that might be considered in a Euclidean argument, definitions of this kind often carry a proof's most vital ideas.

> *We use the dictions of set theory, in particular its general* set formers, *as an essential means of instantiating new objects. As we will show by writing a hundred or so short statements which define all the essential foundations of standard mathematics, set theory gives us a very flexible and powerful tool for making definitions.* [6, p. 9]

The key role played by definitions in proof development should have already emerged from Sect. 2, where we have insisted on the usefulness of the construct THEORY as a means to “associate some highly compound meaning”[5] with its output symbols. Ref owns, among others, built-in theories which perform first-order *Skolemization*: these are used to constrain a brand new function symbol f to meet the condition

$$\big\langle \forall x_1, \dots, x_n \mid \varphi\big(x_1, \dots, x_n, f(x_1, \dots, x_n)\big)\big\rangle$$

(in any number n of arguments) once a theorem of the form

$$\big\langle \forall x_1, \dots, x_n,\ \exists y \mid \varphi(x_1, \dots, x_n, y)\big\rangle$$

has been proved.

Already at a much lower level than the THEORY construct, Ref offers many possibilities. The strength of Ref’s definition mechanisms originates in part from the SET-FORMER notation: By (possibly transfinite) element-iteration over the sets represented by the terms $t_0, t_1 \equiv t_1(z_0), \dots, t_m \equiv t_m(z_0, \dots, z_{m-1})$, we can form the set

$$\{e :\ z_0 \in t_0,\ z_1 \in t_1, \dots, z_m \in t_m \mid \gamma\},$$

where $e \equiv e(z_0, \dots, z_m)$ and $\gamma \equiv \gamma(z_0, \dots, z_m)$ are a set-term and a condition in which the pairwise distinct variables z_i can occur free (similarly, each t_{j+1} may involve $z_0, \dots, z_j$). If the condition γ is omitted, then γ is understood to be true, and if the term e is omitted, then e is understood to be the same as the first variable inside the braces.

Set-formers can be boosted by $\in$-recursion in definitions, as we have illustrated above through the introduction of $\mathsf{ult\_membs}(-)$ and $\mathsf{trCl}(-)$. To now see a series of interrelated low-level definitions at work, let us consider one way (by no means standard) of characterizing a finite *ordered list* $[h_n, \dots, h_2, h_1]$, with $n \geqslant 0$. We can view this as just being a special set $\{q_1, \dots, q_n\}$ within which each q_i indicates

---

[5]This is part of the passage

> *Wir haben oft ein Zeichen nötig, mit dem wir einen sehr zusammengesetzten Sinn verbinden. Dieses Zeichen dient uns sozusagen als Gefäß, in dem wir diesen Sinn mit uns führen können, immer in dem Bewußtsein, daß wir dieses Gefäß öffnen können, wenn wir seines Inhalts bedürfen.*

by Gottlob Frege as worded by Jack at the beginning of [1]:

> *We often need to associate some highly compound meaning with a symbol. Such a symbol serves us as a kind of container carrying this meaning, always with the understanding that it can be opened if we need its content.*

$$
\begin{array}{lll}
\mathsf{add}(\mathsf{H},\mathsf{B}) & =_{\mathrm{Def}} & \Big\{\big\{\ \{\mathsf{H},\emptyset\}\ \big\}\cup\mathsf{B}\Big\}\cup\mathsf{B}\\
\mathsf{final}(\mathsf{L}) & =_{\mathrm{Def}} & \big\{\mathsf{q}\in\mathsf{L}\mid\mathsf{L}\backslash\{\mathsf{q}\}\subseteq\mathsf{q}\big\}\\
\mathsf{sel}(\mathsf{X},\mathsf{Y}) & =_{\mathrm{Def}} & \mathbf{arb}\Big(\mathbf{arb}\big(\mathbf{arb}(X)\backslash\mathsf{Y}\big)\backslash\{\emptyset\}\Big)\\
\mathsf{top}(\mathsf{L}) & =_{\mathrm{Def}} & \mathsf{sel}\big(\mathsf{final}(\mathsf{L}),\mathsf{L}\big)\\
\mathsf{bot}(\mathsf{L}) & =_{\mathrm{Def}} & \mathsf{sel}\big(\mathsf{L},\mathsf{L}\big)\\
[\mathsf{X},\mathsf{Y}] & =_{\mathrm{Def}} & \mathsf{add}\big(\mathsf{Y},\mathsf{add}(\mathsf{X},\emptyset)\big)\\
\mathsf{del}(\mathsf{L}) & =_{\mathrm{Def}} & \mathsf{L}\backslash\mathsf{final}(\mathsf{L})\\
\mathsf{sub}(\mathsf{I},\mathsf{L}) & =_{\mathrm{Def}} & \textbf{if } \mathsf{I}=\emptyset \textbf{ then } \mathsf{L} \textbf{ else } \mathsf{del}\big(\mathbf{arb}(\{\mathsf{sub}(\mathsf{j},\mathsf{L}):\ \mathsf{j}\in\mathsf{I}\mid\mathsf{j}\in\mathsf{final}(\mathsf{I})\})\big) \textbf{ fi}\\
\mathsf{next}(\mathsf{X}) & =_{\mathrm{Def}} & \mathsf{X}\cup\{\mathsf{X}\}\\
\mathsf{len}(\mathsf{L}) & =_{\mathrm{Def}} & \textbf{if } \emptyset\in\mathsf{next}(\mathsf{L}) \textbf{ then } \emptyset \textbf{ else } \mathsf{next}\big(\bigcup\{\mathsf{len}(\mathsf{q}):\ \mathsf{q}\in\mathsf{L}\}\big) \textbf{ fi}\\
\mathsf{th}(\mathsf{I},\mathsf{L}) & =_{\mathrm{Def}} & \textbf{if } \mathsf{I}=\emptyset \textbf{ then } \mathsf{len}(\mathsf{L}) \textbf{ else } \mathsf{top}\Big(\mathsf{sub}\big(\mathsf{del}(\mathsf{I}),\mathsf{L}\big)\Big) \textbf{ fi}\\
\mathsf{cat}(\mathsf{L},\mathsf{M}) & =_{\mathrm{Def}} & \textbf{if } \emptyset\in\mathsf{M} \textbf{ then } \mathsf{M} \textbf{ else } \mathsf{L}\cup\{\mathsf{cat}(\mathsf{L},\mathsf{q}):\ \mathsf{q}\in\mathsf{M}\} \textbf{ fi}\\
\mathsf{Is\_list}(\mathsf{L}) & \leftrightarrow_{\mathrm{Def}} & \big\langle\forall\mathsf{q}\in\mathsf{L},\mathsf{q}'\in\mathsf{L}\mid\mathsf{q}\in\mathsf{q}'\vee\mathsf{q}'\in\mathsf{q}\vee\mathsf{q}=\mathsf{q}'\big\rangle\ \&\\
 & & \big\langle\forall\mathsf{q}\in\mathsf{L},\exists\mathsf{h}\mid\mathsf{q}\backslash\mathsf{L}=\big\{\{\mathsf{h},\emptyset\}\big\}\big\rangle\ \&\\
 & & \big\langle\forall\mathsf{q}'\in\mathsf{L},\exists\mathsf{q}\in\mathsf{L}\mid\mathsf{q}'\cap\mathsf{L}\in\big\{\emptyset,(\mathsf{q}\cap\mathsf{L})\cup\{\mathsf{q}\}\big\}\big\rangle
\end{array}
$$

Fig. 3 Basic definitions referring to lists

that h_i must occur in the i-th position from the right. To this aim, it suffices to put $q_i = \{q_1, \ldots, q_{i-1}, \{\emptyset, h_i\}\}$ for each i. One easily recognizes in the constructs $\mathsf{add}(-,-)$, $\mathsf{top}(-)$, and $\mathsf{del}(-)$ shown in Fig. 3 analogues of LISP's classical triad $\mathsf{cons}(H, L)$, $\mathsf{car}(L)$, and $\mathsf{cdr}(L)$. Various other basic operations related to lists are also shown: in particular, we specify the extraction $\mathsf{th}(I, L)$ of the I-th component (reading from the left) of a list L in terms of a recursive operation $\mathsf{sub}(J, L)$ which gives (for $J = 0, 1, \ldots$) successive sublists $[h_n, \ldots, h_1]$, $[h_{n-1}, \ldots, h_1], \ldots, [h_1]$, $[\ ]$ of a given $L = [h_n, \ldots, h_1]$. Ordered pair formation $[X, Y]$ has its associated projections specified as $\mathsf{bot}(-)$ and $\mathsf{top}(-)$. To end, we specify the general form of a list by means of the predicate $\mathsf{Is\_list}(-)$.

The definitions we have been examining are neither particularly transparent nor very profound: nevertheless they can serve to illustrate how the definition mechanisms of a set-based verifier like Ref retain, even "at their simplest", a lot of semantics. The case when definitions are mere abbreviations is relatively rare: in Fig. 3, the only construct introduced for the 'technical reason' of saving ink is $\mathsf{sel}(-,-)$.

Beyond training examples, since a large-scale scenario will certainly be pervaded by numbers, one will prefer to define a list as being a function whose domain is a natural number—identified, *à la* von Neumann, with the set of all its predecessors.

4 Inferential Armory

What is understanding? Has the word the same meaning for everybody? Does understanding the demonstration of a theorem consist in examining each of the syllogisms of which it is composed in succession, and being convinced that it is correct and conforms to the rules

DEF powerset: [family of all subsets of a set] $\mathscr{P}\mathsf{X} =_{\mathrm{Def}} \{\mathsf{y} : \mathsf{y} \subseteq \mathsf{X}\}$
DEF transitivity: [transitive set] $\mathsf{Trans(T)} \leftrightarrow_{\mathrm{Def}} \{\mathsf{y} \in \mathsf{T} \mid \mathsf{y} \not\subseteq \mathsf{T}\} = \emptyset$
THEOREM xxx: [Peddicord's lemma]
$\mathsf{Trans(T)} \;\&\; \mathsf{S} \subseteq \mathsf{T} \;\&\; \mathsf{S} \neq \mathsf{T} \rightarrow \emptyset \neq (\mathsf{T} \backslash \mathsf{S}) \cap \mathscr{P}\mathsf{S}$. PROOF:
$\mathsf{Suppose\_not}(\mathsf{t}_0, \mathsf{s}_0) \Rightarrow$ AUTO
$\mathsf{Use\_def}(\mathscr{P}\mathsf{s}_0) \Rightarrow$ AUTO
‖ For if our assertion has a counterexample $\mathsf{t}_0, \mathsf{s}_0$, then s_0 must be strictly included in t_0 and hence the axiom of regularity tells us that $\mathsf{t}_0 \backslash \mathsf{s}_0$ has an element, $\mathsf{a} = \mathbf{arb}(\mathsf{t}_0 \backslash \mathsf{s}_0)$, disjoint from $\mathsf{t}_0 \backslash \mathsf{s}_0$. Plainly a is also a member of the superset t_0 of $\mathsf{t}_0 \backslash \mathsf{s}_0$ and is not included in s_0.
$\mathsf{Loc\_def} \Rightarrow$ *Stat1* : $\mathsf{a} = \mathbf{arb}(\mathsf{t}_0 \backslash \mathsf{s}_0)$
$\mathsf{Use\_def(Trans)} \Rightarrow$ *Stat2* : $\{\mathsf{y} \in \mathsf{t}_0 \mid \mathsf{y} \not\subseteq \mathsf{t}_0\} = \emptyset \;\&\; \mathsf{a} \notin \{\mathsf{y}' : \mathsf{y}' \subseteq \mathsf{s}_0\}$
‖ But then, by the definition of transitive set, a must be a subset of s_0, because it is disjoint from $\mathsf{t}_0 \backslash \mathsf{s}_0$. This readily leads us to the desired contradiction.
$\langle \mathsf{a}, \mathsf{a} \rangle \hookrightarrow \mathit{Stat2}(\mathit{Stat1}) \Rightarrow$ false
$\mathsf{Discharge} \Rightarrow$ QED

Fig. 4 Tiny scenario, consisting of two definitions and a 6-line proof

of the game? In the same way, does understanding a definition consist simply in recognizing that the meaning of all the terms employed is already known, and being convinced that it involves no contradiction?
Yes, for some it is; when they have arrived at the conviction, they will say, I understand. But not for the majority. Almost all are more exacting; they want to know not only whether all the syllogisms of a demonstration are correct, but why they are linked together in one order rather than in another. As long as they appear to them engendered by caprice, and not by an intelligence constantly conscious of the end to be attained, they do not think they have understood.[6]

(H. Poincaré, *Science et méthode*, 1909)

For a presentation of the inferential armory of Ref, the interested reader is referred to [4]: here I will limit myself to sparse remarks about some salient peculiarities of this system.

Figure 4 displays the six-liner proof of a variant of Theorem 4a of the scenario accompanying this paper. An arrow within each line separates a HINT, specifying the inference rule that justifies the step, from the STATEMENT being derived at that particular line. In two lines the statement is represented by the placeholder AUTO. This happens once when, as the keyword Suppose_not suggests, an argument by contradiction begins: here AUTO stands for

$$\neg\big(\mathsf{Is\_full}(\mathsf{t}_0) \;\&\; \mathsf{s}_0 \subseteq \mathsf{t}_0 \;\&\; \mathsf{s}_0 \neq \mathsf{t}_0 \rightarrow \emptyset \neq (\mathsf{t}_0 \backslash \mathsf{s}_0) \cap \mathscr{P}\mathsf{s}_0\big),$$

opposite to the theorem's claim, where the new names $\mathsf{t}_0, \mathsf{s}_0$ have superseded the variables T, S, as requested by the hint. AUTO is exploited again in the next step, whose hint suggests that the instantiated definition $\mathscr{P}\mathsf{s}_0 = \{\mathsf{y} : \mathsf{y} \subseteq \mathsf{s}_0\}$ be transcribed *verbatim*.

It is very unusual, though, that the statement of a proof line is fully determined by its hint. One cause of this lack of uniqueness is that most inference primitives do not

[6]Taken from: Henri Poincaré, "Science and method", London: Routledge, 1996, p. 118.

DEF transitivity: [transitive set] $\mathsf{Trans}(\mathsf{T}) \leftrightarrow_{\mathrm{Def}} \{\mathsf{y} \in \mathsf{T} \mid \mathsf{y} \not\subseteq \mathsf{T}\} = \emptyset$

THEOREM yyy: [Peddicord's lemma]

$\mathsf{Trans}(\mathsf{T})\ \&\ \mathsf{S} \subseteq \mathsf{T}\ \&\ \mathsf{S} \neq \mathsf{T}\ \&\ \mathsf{A} = \mathbf{arb}(\mathsf{T} \backslash \mathsf{S}) \rightarrow \mathsf{A} \subseteq \mathsf{S}\ \&\ \mathsf{A} \in \mathsf{T} \backslash \mathsf{S}$. PROOF:

Suppose_not(t, s, a) $\Rightarrow$ $\mathsf{Trans}(\mathsf{t})\ \&\ \mathsf{s} \subseteq \mathsf{t}\ \&\ \mathsf{s} \neq \mathsf{t}\ \&\ \mathsf{a} = \mathbf{arb}(\mathsf{t} \backslash \mathsf{s})\ \&\ (\mathsf{a} \not\subseteq \mathsf{s} \vee \mathsf{a} \notin \mathsf{t} \backslash \mathsf{s})$

Use_def(Trans) $\Rightarrow$ *Stat1* : $\mathsf{a} \notin \{\mathsf{y} \in \mathsf{t} \mid \mathsf{y} \not\subseteq \mathsf{t}\}$

$\langle \mathsf{a} \rangle \hookrightarrow$ *Stat1* $\Rightarrow$ $\mathsf{a} \subseteq \mathsf{t}$

Discharge $\Rightarrow$ QED

THEOREM zzz: [for any transitive t: $\emptyset \in t$ if $t \neq \emptyset$, $\{\emptyset\} \in t$ if $t \not\subseteq \{\emptyset\}$, etc.]

$\mathsf{Trans}(\mathsf{T})\ \&\ \mathsf{N} \in \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}\ \&\ \mathsf{T} \not\subseteq \mathsf{N} \rightarrow$

$\mathsf{N} \subseteq \mathsf{T}\ \&\ \big(\mathsf{N} \in \mathsf{T} \vee (\mathsf{N} = \{\emptyset, \{\emptyset\}\}\ \&\ \{\{\emptyset\}\} \in \mathsf{T})\big)$. PROOF:

Suppose_not(t, n) $\Rightarrow$ AUTO

‖ The '($\star$)' context restriction in the following three lines serves to hide the semantics of **arb**: which, to the limited extent necessary here, has been captured by the preceding Peddicord's lemma.

$\langle \mathsf{t}, \emptyset, \mathbf{arb}(\mathsf{t} \backslash \emptyset) \rangle \hookrightarrow T\mathsf{yyy}(\star) \Rightarrow$ $\emptyset \in \mathsf{t}$

$\langle \mathsf{t}, \{\emptyset\}, \mathbf{arb}(\mathsf{t} \backslash \{\emptyset\}) \rangle \hookrightarrow T\mathsf{yyy}(\star) \Rightarrow$ $\{\emptyset\} \in \mathsf{t}$

$\langle \mathsf{t}, \{\emptyset, \{\emptyset\}\}, \mathbf{arb}(\mathsf{t} \backslash \{\emptyset, \{\emptyset\}\}) \rangle \hookrightarrow T\mathsf{yyy}(\star) \Rightarrow$ false

Discharge $\Rightarrow$ QED

Fig. 5 Variant of the tiny scenario of Fig. 4, showing use of a lemma

have a fixed number of premisses from which their consequent must follow, and precise indication of the premisses is not mandatory. Different consequents can hence be drawn, at a given place and by means of the same form of inference, depending on which subset of the preceding lines of the proof enters into play. Consider, for example, the penultimate proof line of Fig. 4: here the hint indicates that a must be used twice, to instantiate both bound variables, y and y′, of the statement labeled *Stat2*; moreover, the (optional) context restriction '(*Stat1*)' appearing in the hint of this line indicates that *Stat1* and *Stat2* are the only statements worth being taken into account to get the consequent false. It would be perfectly legitimate to replace the statement false by

$$(\mathsf{a} \in \mathsf{t}_0 \rightarrow \mathsf{a} \subseteq \mathsf{t}_0)\ \&\ \mathsf{a} \not\subseteq \mathsf{s}_0,$$

although the latter would only depend on the line labeled *Stat2*. Notice that after this change—and, of course, also if the context restriction were tightened into '(*Stat2*)'—the entire proof would remain correct, as the Discharge primitive would still be able to see the contradiction.

Another source of freedom in the statement of a line arises from the presence, among the fifteen or so inference mechanisms which constitute the inferential armory of Ref, of an inference primitive which gets often tacitly combined with other forms of inference. This is the mechanism ELEM, which implements the *multi-level syllogistic* decision algorithm [1, 2], also invocable directly on its own right. The tacit use of ELEM explains, in particular, why Discharge can still do its duty after the change discussed in the preceding paragraph. One can catch ELEM at work in almost every proof line of Fig. 5.

Much of the usability of Ref stems from ELEM and, to a lesser extent, from other behind-the-scenes proof mechanisms, e.g., *proof-by-structure* which, once switched on, performs many kinds of type inference as explained in [4, Sect. 4]. These mechanisms shroud the collection of statements already accepted as proved in a 'penumbra' of additional statements which follow from them as elementary consequences. In a talk given at City College in New York in September 2003, Jack said:

> *Putting mathematical discourse into a form every one of whose details can be checked by a computer forces us to 'walk in shackles'*
> *—but then we want these shackles to be as light as possible.*

5 Conclusions

This is my valedictory to Jack, who honored me with a valedictory when, at the beginning of 1981, I left New York and returned to Italy.

I hope I will be energetic enough to bring to completion various items that Jack left unaccomplished. These include a full formalization of the proof of the Cauchy integral theorem on which (years ago) Jack encouraged me to work "maniacally".

I spent many days working with Jack in August 2008, when he began to regard as an absolute priority the reimplementation of Ref/ÆtnaNova in Javascript (and partly in Java), because nobody was any longer in charge of maintaining SETL. To be frank, neither of us was very fluent in Javascript, and we had 25,000 lines or so of SETL code to translate; so the amount of work we then did was immense but inconclusive. Not discouraged, when I had my last chances to speak with Jack in February 2009, he recommended that this translation work was brought forward.

Definitely I would prefer to see (and would be, with all my limits, ready to concur to) an effort to bring SETL out of its current limbo, possibly in a more modern implementation that Salvatore Paxia seems to have in mind: this would be, I deem, a worthwhile tribute to a long-winged project which also absorbed much of Jack's energies over the years. But this, of course, is just my personal taste—and relates to a different story.

Appendix: Claw-Free Graphs as Sets

Eugenio G. Omodeo and Alexandru I. Tomescu

This scenario contains the formal proofs, checked by J. T. Schwartz's proof-verifier Ref, of two classical results on connected claw-free graphs; namely, that any such graph:

- owns a perfect matching if its number of vertices is even,
- has a Hamiltonian cycle in its square if it owns three or more vertices.

The original proofs (cf. [14, 15] and [11]) referred to undirected graphs, the ones to be presented refer to a special class of digraphs whose vertices are hereditarily finite sets and whose edges reflect the membership relation. Ours is a legitimate change of perspective in the light of [12], as we will briefly explain at the end.

To make our formal development self-contained, we proceed from the bare set-theoretic foundation built into Ref (cf. [13]). The lemmas exploited without proof in what follows are indeed very few, and their full proofs are available in [13].

A.1 Basic Laws on the Union-Set Global Operation

DEF unionset: [Members of members of a set] $\bigcup X =_{\mathrm{Def}} \{u : v \in X, u \in v\}$

The proof of the following claim, that the union set of a set s is the set-theoretic 'least upper bound' of all its elements, can be found in [13, p. 387].

THEOREM 2: [l.u.b.] $(X \in S \rightarrow X \subseteq \bigcup S)$ & $(\langle \forall y \in S \mid y \subseteq X\rangle \rightarrow \bigcup S \subseteq X)$.

THEORY imageOfDoubleton$\big(f(X), x_0, x_1\big)$
END imageOfDoubleton

ENTER_THEORY imageOfDoubleton

THEOREM imageOfDoubleton: [Image of an 'elementary set']
$\{f(v) : v \in \emptyset\} = \emptyset$ & $\{f(v) : v \in \{x_0\}\} = \{f(x_0)\}$ &
$\{f(v) : v \in \{x_0, x_1\}\} = \{f(x_0), f(x_1)\}$. PROOF:
Suppose_not() $\Rightarrow$ AUTO

Ref has the built-in ability to reduce $\{f(v) : v \in \emptyset\}$ to $\emptyset$ and $\{f(v) : v \in \{x_0\}\}$ to $\{f(x_0)\}$; hence we are left with only the doubleton to consider. Let c belong to one of $\{f(v) : v \in \{x_0, x_1\}\}$ and $\{f(x_0), f(x_1)\}$ but not to the other. After excluding, through variable-substitution, the case $c \notin \{f(v) : v \in \{x_0, x_1\}\}$, we easily exclude both possibilities $c = f(x_0)$ and $c = f(x_1)$, through variable-substitution and equality propagation.

SIMPLF $\Rightarrow$ *Stat1* : $\{f(v) : v \in \{x_0, x_1\}\} \neq \{f(x_0), f(x_1)\}$
$\langle c\rangle \hookrightarrow$ *Stat1* $\Rightarrow$ $c \in \{f(v) : v \in \{x_0, x_1\}\} \nleftrightarrow c \in \{f(x_0), f(x_1)\}$
Suppose $\Rightarrow$ *Stat2* : $c \notin \{f(v) : v \in \{x_0, x_1\}\}$
$\langle x_0\rangle \hookrightarrow$ *Stat2* $\Rightarrow$ AUTO
$\langle x_1\rangle \hookrightarrow$ *Stat2* $\Rightarrow$ AUTO

Discharge ⇒ *Stat3* : $c \in \{f(v) : v \in \{x_0, x_1\}\}$ & $c \notin \{f(x_0), f(x_1)\}$
$\langle x' \rangle \hookrightarrow$ *Stat3* ⇒ $x' \in \{x_0, x_1\}$ & $f(x') \neq f(x_0)$ & $f(x') \neq f(x_1)$
Suppose ⇒ $x' = x_0$
EQUAL ⇒ false; Discharge ⇒ $x' = x_1$
EQUAL ⇒ false; Discharge ⇒ QED

ENTER_THEORY Set_theory
DISPLAY imageOfDoubleton

THEORY imageOfDoubleton$(f(X), x_0, x_1)$
$\{f(v) : v \in \emptyset\} = \emptyset$ & $\{f(v) : v \in \{x_0\}\} = \{f(x_0)\}$ &
$\{f(v) : v \in \{x_0, x_1\}\} = \{f(x_0), f(x_1)\}$
END imageOfDoubleton

THEOREM 2*a*: [$\bigcup$ of double-/single-tons] $Z = \{X, Y\} \rightarrow \bigcup Z = X \cup Y$. PROOF:
Suppose_not(z_0, x_0, y_0) ⇒ AUTO

Under the assumption that $z_0 = \{x_0, y_0\}$ & $\bigcup z_0 \neq x_0 \cup y_0$ can hold, two citations of Theorem 2 readily yield $x_0 \subseteq \bigcup z_0$ and $y_0 \subseteq \bigcup z_0$.

$\langle x_0, z_0 \rangle \hookrightarrow T2$ ⇒ AUTO
$\langle y_0, z_0 \rangle \hookrightarrow T2$ ⇒ AUTO

A third citation of the same Theorem 2 enables us to derive from $\bigcup z_0 \neq x_0 \cup y_0$ that some element of $z_0 = \{x_0, y_0\}$ is not included in $x_0 \cup y_0$, which is manifestly absurd.

$\langle x_0 \cup y_0, z_0 \rangle \hookrightarrow T2$ ⇒ *Stat1* : $\neg \langle \forall y \in z_0 \mid y \subseteq x_0 \cup y_0 \rangle$
$\langle v \rangle \hookrightarrow$ *Stat1* ⇒ $v \in \{x_0, y_0\}$ & $v \not\subseteq x_0 \cup y_0$
(*Stat1*$\star$)Discharge ⇒ QED

THEOREM 2*b*: [Union of union] $\bigcup \bigcup X = \bigcup \left\{ \bigcup y : y \in X \right\}$. PROOF:
Suppose_not(x_0) ⇒ AUTO
Use_def$(\bigcup)$ ⇒ $\{z : y \in \{u : v \in x_0, u \in v\}, z \in y\} \neq$
$\left\{ s : r \in \left\{ \bigcup y : y \in x_0 \right\}, s \in r \right\}$
SIMPLF ⇒ *Stat1* : $\{z : v \in x_0, u \in v, z \in u\} \neq \left\{ s : y \in x_0, s \in \bigcup y \right\}$
$\langle z_0 \rangle \hookrightarrow$ *Stat1* ⇒ AUTO
Suppose ⇒ *Stat3* : $z_0 \in \{z : v \in x_0, u \in v, z \in u\}$ &
$z_0 \notin \left\{ s : y \in x_0, s \in \bigcup y \right\}$
Use_def$(\bigcup v_0)$ ⇒ AUTO
$\langle v_0, u_0, z, v_0, z_0 \rangle \hookrightarrow$ *Stat3*(*Stat1*$\star$) ⇒
Stat4 : $z_0 \notin \{z : u \in v_0, z \in u\}$ & $v_0 \in x_0$ & $u_0 \in v_0$ & $z_0 \in u_0$
$\langle u_0, z_0 \rangle \hookrightarrow$ *Stat4*(*Stat4*$\star$) ⇒ false
Discharge ⇒ *Stat5* : $z_0 \in \left\{ s : y \in x_0, s \in \bigcup y \right\}$ &
Stat6 : $z_0 \notin \{z : v \in x_0, u \in v, z \in u\}$

$\mathsf{Use\_def}(\bigcup y_0) \Rightarrow$ AUTO
$\langle y_0, s_0\rangle \hookrightarrow Stat5(Stat5\star) \Rightarrow$ $Stat7:\ z_0 \in \{s:\ u \in y_0, s \in u\}\ \&\ y_0 \in x_0$
$\langle u_1, s_1\rangle \hookrightarrow Stat7(Stat7\star) \Rightarrow$ $z_0 \in u_1\ \&\ u_1 \in y_0$
$\langle y_0, u_1, z_0\rangle \hookrightarrow Stat6(Stat7\star) \Rightarrow$ false; Discharge $\Rightarrow$ QED

THEOREM 2*c*: [Additivity and monotonicity of monadic union]
$\bigcup(X \cup Y) = \bigcup X \cup \bigcup Y\ \&\ (Y \supseteq X \rightarrow \bigcup Y \supseteq \bigcup X)$. PROOF:
$\mathsf{Suppose\_not}(x_0, y_0) \Rightarrow$ AUTO
Suppose $\Rightarrow$ $\bigcup(x_0 \cup y_0) \neq \bigcup x_0 \cup \bigcup y_0$
$\langle \{x_0, y_0\}\rangle \hookrightarrow T2b \Rightarrow$ $\bigcup\bigcup\{x_0, y_0\} = \bigcup\{\bigcup v:\ v \in \{x_0, y_0\}\}$
APPLY $\langle\ \rangle$ imageOfDoubleton$\big(f(X) \mapsto \bigcup X, x_0 \mapsto x_0, x_1 \mapsto y_0\big) \Rightarrow$
$\{\bigcup v:\ v \in \{x_0, y_0\}\} = \{\bigcup x_0, \bigcup y_0\}$
$\langle \{x_0, y_0\}, x_0, y_0\rangle \hookrightarrow T2a \Rightarrow$ $\bigcup\{x_0, y_0\} = x_0 \cup y_0$
$\langle \{\bigcup x_0, \bigcup y_0\}, \bigcup x_0, \bigcup y_0\rangle \hookrightarrow T2a \Rightarrow$ $\bigcup\{\bigcup x_0, \bigcup y_0\} = \bigcup x_0 \cup \bigcup y_0$
EQUAL $\Rightarrow$ false
Discharge $\Rightarrow$ $\bigcup(x_0 \cup y_0) = \bigcup x_0 \cup \bigcup y_0\ \&\ y_0 = x_0 \cup y_0\ \&\ \bigcup y_0 \not\supseteq \bigcup x_0$
EQUAL $\Rightarrow$ $\bigcup y_0 = \bigcup x_0 \cup \bigcup y_0$
Discharge $\Rightarrow$ QED

THEOREM 2*e*: [Union of adjunction] $\bigcup(X \cup \{Y\}) = Y \cup \bigcup X$. PROOF:
$\mathsf{Suppose\_not}(x_0, y_0) \Rightarrow$ $Stat0:\ \bigcup(x_0 \cup \{y_0\}) \neq y_0 \cup \bigcup x_0$
$\langle a\rangle \hookrightarrow Stat0 \Rightarrow$ $a \in \bigcup(x_0 \cup \{y_0\}) \neq a \in y_0 \cup \bigcup x_0$

Arguing by contradiction, let x_0, y_0 be a counterexample, so that in either one of $\bigcup(x_0 \cup \{y_0\})$ and $y_0 \cup \bigcup x_0$ there is an a not belonging to the other set. Taking the definition of $\bigcup$ into account, by monotonicity we must exclude the possibility that $a \in \bigcup x_0 \setminus \bigcup(x_0 \cup \{y_0\})$; through variable-substitution, we must also discard the possibility that $a \in \bigcup(x_0 \cup \{y_0\}) \setminus \bigcup x_0 \setminus y_0$.

Set_monot $\Rightarrow$ $\{u:\ v \in x_0, u \in v\} \subseteq \{u:\ v \in x_0 \cup \{y_0\}, u \in v\}$
Suppose $\Rightarrow$ $Stat1:\ a \in \{u:\ v \in x_0 \cup \{y_0\}, u \in v\}\ \&$
$a \notin \{u:\ v \in x_0, u \in v\}\ \&\ a \notin y_0$
$\langle v_0, u_0, v_0, u_0\rangle \hookrightarrow Stat1 \Rightarrow$ false; Discharge $\Rightarrow$ AUTO
$\mathsf{Use\_def}(\bigcup) \Rightarrow$ $Stat2:\ a \notin \{u:\ v \in x_0 \cup \{y_0\}, u \in v\}\ \&\ a \in y_0$

The only possibility left, namely that $a \in y_0 \setminus \bigcup(x_0 \cup \{y_0\})$, is also manifestly absurd. This contradiction leads us to the desired conclusion.

$\langle y_0, a\rangle \hookrightarrow Stat2 \Rightarrow$ false; Discharge $\Rightarrow$ QED

A.2 Transitive Sets

DEF transitivity: [Transitive set] $\mathsf{Trans}(T) \leftrightarrow_{\mathrm{Def}} \{y \in T \mid y \not\subseteq T\} = \emptyset$

THEOREM 3*a*: [Transitive sets include their unionsets] $\mathsf{Trans}(T) \leftrightarrow T \supseteq \bigcup T$.

$\mathsf{Suppose\_not}(t) \Rightarrow$ AUTO

$\mathsf{Use\_def}(\bigcup t) \Rightarrow$ AUTO

$\mathsf{Use\_def}(\mathsf{Trans}(t)) \Rightarrow$ AUTO

$\mathsf{Suppose} \Rightarrow$ $\mathit{Stat1}:\ t \not\supseteq \bigcup t\ \&\ \mathsf{Trans}(t)$

$\langle c\rangle \hookrightarrow \mathit{Stat1}(\star) \Rightarrow$ $\mathit{Stat2}:\ c \in \{u :\ v \in t, u \in v\}\ \&$
$\{y \in t \mid y \not\subseteq t\} = \emptyset\ \&\ c \notin t$

$\langle v, u, v\rangle \hookrightarrow \mathit{Stat2}(\mathit{Stat2}\star) \Rightarrow$ false

$\mathsf{Discharge} \Rightarrow$ $\mathit{Stat3}:\ \{y \in t \mid y \not\subseteq t\} \neq \emptyset\ \&\ t \supseteq \{u :\ v \in t, u \in v\}$

$\mathsf{Loc\_def} \Rightarrow$ $a = \mathbf{arb}(d \setminus t)$

$\langle d\rangle \hookrightarrow \mathit{Stat3}(\mathit{Stat3}) \Rightarrow$
$\mathit{Stat4}:\ a \notin \{u :\ v \in t, u \in v\}\ \&\ d \in t\ \&\ a \in d\ \&\ a \notin t$

$\langle d, a\rangle \hookrightarrow \mathit{Stat4}(\mathit{Stat4}\star) \Rightarrow$ false; $\mathsf{Discharge} \Rightarrow$ QED

THEOREM 3*c*: [For a transitive set, elements are also subsets]
$\mathsf{Trans}(T)\ \&\ X \in T \rightarrow X \subseteq T$. PROOF:

$\mathsf{Suppose\_not}(t, x) \Rightarrow$ AUTO

$\langle t\rangle \hookrightarrow T3a \Rightarrow$ $\mathit{Stat1}:\ t = t \cup \{x\}\ \&\ \bigcup t \not\supseteq x \cup \bigcup t$

$\langle t, x\rangle \hookrightarrow T2e \Rightarrow$ AUTO

$\mathsf{EQUAL}(\mathit{Stat1}) \Rightarrow$ false; $\mathsf{Discharge} \Rightarrow$ QED

THEOREM 3*d*: [Trapping phenomenon for trivial sets]
$\mathsf{Trans}(S)\ \&\ X, Z \in S\ \&\ X \notin Z\ \&\ Z \notin X\ \&\ S \setminus \{X, Z\} \subseteq \{\emptyset, \{\emptyset\}\} \rightarrow$
$S \subseteq \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\emptyset, \{\emptyset\}\}\}$. PROOF:

$\mathsf{Suppose\_not}(s, x, z) \Rightarrow$ AUTO

$\langle s, x\rangle \hookrightarrow T3c \Rightarrow$ AUTO

$\langle s, z\rangle \hookrightarrow T3c \Rightarrow$ AUTO

$\mathsf{Discharge} \Rightarrow$ QED

|| Any strict subset of a transitive set t, owns a subset in t which does not belong to it.

THEOREM 4*a*: [Peddicord's lemma] $\mathsf{Trans}(T)\ \&\ Y \subseteq T\ \&$
$Y \neq T\ \&\ A = \mathbf{arb}(T \setminus Y) \rightarrow A \subseteq Y\ \&\ A \in T \setminus Y$. PROOF:

$\mathsf{Suppose\_not}(t, y, a) \Rightarrow$ AUTO

$\langle t, a\rangle \hookrightarrow T3c \Rightarrow$ $a \subseteq t$

$\mathsf{Discharge} \Rightarrow$ QED

THEOREM 4*b*: [$\emptyset$ belongs to any transitive $t \neq \emptyset$, so does $\{\emptyset\}$ if $t \not\subseteq \{\emptyset\}$, etc.]
$\mathsf{Trans}(T)\ \&\ N \in \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}\ \&\ T \not\subseteq N \rightarrow$
$N \subseteq T\ \&\ \big(N \in T \vee (N = \{\emptyset, \{\emptyset\}\}\ \&\ \{\{\emptyset\}\} \in T)\big)$. PROOF:

$\mathsf{Suppose\_not}(t, n) \Rightarrow$ AUTO

|| The '($\star$)' context restriction in the following three steps serves to hide the semantics of **arb**: which, to the limited extent necessary here, has been captured by the preceding Peddicord's lemma.

$$\langle \mathsf{t}, \emptyset, \mathbf{arb}(\mathsf{t}\backslash\emptyset) \rangle \hookrightarrow T4a(\star) \Rightarrow \quad \emptyset \in \mathsf{t}$$
$$\langle \mathsf{t}, \{\emptyset\}, \mathbf{arb}(\mathsf{t}\backslash \{\emptyset\}) \rangle \hookrightarrow T4a(\star) \Rightarrow \quad \{\emptyset\} \in \mathsf{t}$$
$$\langle \mathsf{t}, \{\emptyset, \{\emptyset\}\}, \mathbf{arb}(\mathsf{t}\backslash \{\emptyset, \{\emptyset\}\}) \rangle \hookrightarrow T4a(\star) \Rightarrow \quad \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow \quad \text{QED}$$

THEOREM 4*c*: [Source removal does not disrupt transitivity]
$\mathsf{Trans(S)}$ & $\mathsf{S} \supseteq \mathsf{T}$ & $(\mathsf{S}\backslash\mathsf{T}) \cap \bigcup \mathsf{S} = \emptyset \rightarrow \mathsf{Trans(T)}$. PROOF:
$\mathsf{Suppose\_not(s,t)} \Rightarrow \quad$ AUTO
$\mathsf{Use\_def(Trans)} \Rightarrow \quad Stat1: \{\mathsf{y} \in \mathsf{t} \mid \mathsf{y} \not\subseteq \mathsf{t}\} \neq \emptyset$ & $\{\mathsf{y} \in \mathsf{s} \mid \mathsf{y} \not\subseteq \mathsf{s}\} = \emptyset$

‖ Assuming that s is transitive, that t equals s deprived of some sources and that t is not transitive, there must be an element y of t which is not a subset of t, so that a $\mathsf{z} \in \mathsf{y}$ exists which does not belong to t. Due to the transitivity of s, y is included in s and hence z belongs to s; hence, under the assumption that $\mathsf{s}\backslash\mathsf{t}$ and $\bigcup \mathsf{s}$ are disjoint, z does not belong to $\bigcup \mathsf{s}$.

$$\langle \mathsf{y},\mathsf{y} \rangle \hookrightarrow Stat1 \Rightarrow \quad Stat2: \mathsf{y} \not\subseteq \mathsf{t} \;\&\; \mathsf{y} \in \mathsf{s} \;\&\; \mathsf{y} \subseteq \mathsf{s}$$
$$\mathsf{Use\_def}(\textstyle\bigcup \mathsf{s}) \Rightarrow \quad \text{AUTO}$$
$$\langle \mathsf{z} \rangle \hookrightarrow Stat2 \Rightarrow \quad Stat3: \mathsf{z} \notin \{\mathsf{u}: \mathsf{v} \in \mathsf{s}, \mathsf{u} \in \mathsf{v}\} \;\&\; \mathsf{z} \in \mathsf{y}$$

‖ However, this is untenable.

$$\langle \mathsf{y},\mathsf{z} \rangle \hookrightarrow Stat3 \Rightarrow \quad \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow \quad \text{QED}$$

A.3 Basic Laws on the Finitude Property

‖ To begin developing an acceptable treatment of finiteness without much preparatory work, we adopt here the definition (reminiscent of Tarski's 1924 paper "Sur les ensembles finis"): a set F is finite if every non-null family of subsets of F owns an inclusion-minimal element. This notion is readily specified in terms of the power-set operator, as follows:

DEF $\mathscr{P}$: [Family of all subsets of a given set] $\mathscr{P}\mathsf{S} =_{\text{Def}} \{\mathsf{x}: \mathsf{x} \subseteq \mathsf{S}\}$

DEF Fin: [Finitude] $\mathsf{Finite(F)} \leftrightarrow_{\text{Def}} \langle \forall \mathsf{g} \in \mathscr{P}(\mathscr{P}\mathsf{F})\backslash \{\emptyset\}, \exists \mathsf{m} \mid \mathsf{g} \cap \mathscr{P}\mathsf{m} = \{\mathsf{m}\} \rangle$

‖ The lemma on the monotonicity of finitude and the THEORY of finite induction displayed below are proved in full—together with various other laws on finiteness which we will not need here—in [13, pp. 405–407].

THEOREM 24: [Monotonicity of finitude] $\mathsf{Y} \supseteq \mathsf{X}$ & $\mathsf{Finite(Y)} \rightarrow \mathsf{Finite(X)}$.

THEORY finiteInduction$(\mathsf{s_0}, \mathsf{P(S)})$
$\quad \mathsf{Finite(s_0)} \;\&\; \mathsf{P(s_0)}$
$\Rightarrow (\mathsf{fin}_\Theta)$
$\quad \langle \forall \mathsf{S} \mid \mathsf{S} \subseteq \mathsf{fin}_\Theta \rightarrow \mathsf{Finite(S)} \;\&\; (\mathsf{P(S)} \leftrightarrow \mathsf{S} = \mathsf{fin}_\Theta) \rangle$
END finiteInduction

A.4 Some Combinatorics of the Union-Set Operation

THEOREM 31*d*: [Unionset of $\emptyset$ and $\{\emptyset\}$] $\mathsf{Y} \subseteq \{\emptyset\} \leftrightarrow \bigcup \mathsf{Y} = \emptyset$. PROOF:

$\mathsf{Suppose\_not}(\mathsf{x_0}) \Rightarrow$ AUTO
$\quad \mathsf{Use\_def}(\bigcup \mathsf{x_0}) \Rightarrow$ AUTO
$\quad \mathsf{Suppose} \Rightarrow \; \mathsf{x_0} \subseteq \{\emptyset\}$
$\quad\quad \mathsf{ELEM} \Rightarrow \; Stat1: \{\mathsf{z}: \mathsf{y} \in \mathsf{x_0}, \mathsf{z} \in \mathsf{y}\} \neq \emptyset$
$\quad\quad \langle \mathsf{y_0}, \mathsf{z_1} \rangle \hookrightarrow Stat1 \Rightarrow$ false
$\quad \mathsf{Discharge} \Rightarrow \; Stat2: \mathsf{x_0} \not\subseteq \{\emptyset\} \;\&\; \{\mathsf{z}: \mathsf{y} \in \mathsf{x_0}, \mathsf{z} \in \mathsf{y}\} = \emptyset$
$\langle \mathsf{y_1}, \mathsf{y_1}, \mathbf{arb}(\mathsf{y_1}) \rangle \hookrightarrow Stat2 \Rightarrow$ false; $\quad$ Discharge $\Rightarrow$ QED

THEOREM 31*e*: [Unionset of a set obtained through single removal]
$\quad \bigcup(\mathsf{X} \setminus \{\mathsf{Y}\}) \supseteq \bigcup \mathsf{X} \setminus \mathsf{Y} \;\&\; \bigcup \mathsf{X} \supseteq \bigcup(\mathsf{X} \setminus \{\mathsf{Y}\})$. PROOF:

$\quad \mathsf{Suppose\_not}(\mathsf{x}, \mathsf{y}) \Rightarrow$ AUTO
$\quad \langle \mathsf{x} \setminus \{\mathsf{y}\}, \mathsf{x} \rangle \hookrightarrow T2c(\star) \Rightarrow \; Stat1: \bigcup(\mathsf{x} \setminus \{\mathsf{y}\}) \not\supseteq \bigcup \mathsf{x} \setminus \mathsf{y}$
$\quad \langle \mathsf{c} \rangle \hookrightarrow Stat1(Stat1\star) \Rightarrow \; Stat2: \mathsf{c} \in \bigcup \mathsf{x} \setminus \mathsf{y} \;\&\; \mathsf{c} \notin \bigcup(\mathsf{x} \setminus \{\mathsf{y}\})$
$\quad \mathsf{Use\_def}(\bigcup) \Rightarrow \; Stat3: \mathsf{c} \in \{\mathsf{u}: \mathsf{v} \in \mathsf{x}, \mathsf{u} \in \mathsf{v}\} \;\&$
$\quad\quad\quad \mathsf{c} \notin \{\mathsf{u}: \mathsf{v} \in \mathsf{x} \setminus \{\mathsf{y}\}, \mathsf{u} \in \mathsf{v}\} \;\&\; \mathsf{c} \notin \mathsf{y}$
$\langle \mathsf{v_0}, \mathsf{u_0}, \mathsf{v_0}, \mathsf{u_0} \rangle \hookrightarrow Stat3(Stat3\star) \Rightarrow$ false; $\quad$ Discharge $\Rightarrow$ QED

THEOREM 31*f*: [Unionset, after a removal followed by two adjunctions]
$\quad \bigcup \mathsf{M} \supseteq \mathsf{P} \;\&\; \mathsf{Q} \cup \mathsf{R} = \mathsf{P} \cup \mathsf{S} \rightarrow \bigcup(\mathsf{M} \setminus \{\mathsf{P}\} \cup \{\mathsf{Q}, \mathsf{R}\}) = \bigcup \mathsf{M} \cup \mathsf{S}$. PROOF:

$\mathsf{Suppose\_not}(\mathsf{m}, \mathsf{p}, \mathsf{q}, \mathsf{r}, \mathsf{s}) \Rightarrow$ AUTO
$\quad \mathsf{TELEM} \Rightarrow \; \mathsf{m} \setminus \{\mathsf{p}\} \cup \{\mathsf{q}\} \cup \{\mathsf{r}\} = \mathsf{m} \setminus \{\mathsf{p}\} \cup \{\mathsf{q}, \mathsf{r}\}$
$\quad \mathsf{EQUAL} \Rightarrow \; \bigcup(\mathsf{m} \setminus \{\mathsf{p}\} \cup \{\mathsf{q}\} \cup \{\mathsf{r}\}) = \bigcup(\mathsf{m} \setminus \{\mathsf{p}\} \cup \{\mathsf{q}, \mathsf{r}\})$
$\quad \langle \mathsf{m} \setminus \{\mathsf{p}\}, \mathsf{q} \rangle \hookrightarrow T2e \Rightarrow$ AUTO
$\quad \langle \mathsf{m} \setminus \{\mathsf{p}\} \cup \{\mathsf{q}\}, \mathsf{r} \rangle \hookrightarrow T2e(\star) \Rightarrow \bigcup(\mathsf{m} \setminus \{\mathsf{p}\} \cup \{\mathsf{q}, \mathsf{r}\}) = \bigcup(\mathsf{m} \setminus \{\mathsf{p}\}) \cup (\mathsf{p} \cup \mathsf{s})$
$\langle \mathsf{m}, \mathsf{p} \rangle \hookrightarrow T31e(\star) \Rightarrow$ false; $\quad$ Discharge $\Rightarrow$ QED

THEOREM 31*g*: [Incomparability of pre-pivotal elements]
$\quad \mathsf{Y} \in \mathsf{X} \;\&\; \mathsf{X} \in \mathsf{Z} \;\&\; \mathsf{X}, \mathsf{Z} \in \mathsf{S} \rightarrow \mathsf{Y} \in \bigcup(\mathsf{S} \cap \bigcup \mathsf{S})$. PROOF:

$\mathsf{Suppose\_not}(\mathsf{y}, \mathsf{x}, \mathsf{z}, \mathsf{s}) \Rightarrow \; \mathsf{y} \in \mathsf{x} \;\&\; \mathsf{x} \in \mathsf{z} \;\&\; \mathsf{x}, \mathsf{z} \in \mathsf{s} \;\&\; \mathsf{y} \notin \bigcup(\mathsf{s} \cap \bigcup \mathsf{s})$
$\quad \mathsf{Use\_def}(\bigcup) \Rightarrow \; Stat1: \mathsf{y} \notin \{\mathsf{v}: \mathsf{u} \in \mathsf{s} \cap \bigcup \mathsf{s}, \mathsf{v} \in \mathsf{u}\}$
$\quad \mathsf{Use\_def}(\bigcup \mathsf{s}) \Rightarrow$ AUTO
$\quad \langle \mathsf{x}, \mathsf{y} \rangle \hookrightarrow Stat1(\star) \Rightarrow \; Stat2: \mathsf{x} \notin \{\mathsf{t}: \mathsf{w} \in \mathsf{s}, \mathsf{t} \in \mathsf{w}\}$
$\langle \mathsf{z}, \mathsf{x} \rangle \hookrightarrow Stat2(\star) \Rightarrow$ false; $\quad$ Discharge $\Rightarrow$ QED

Preparatory to a technique to which we will resort for extending perfect matchings, we introduce the following trivial combinatorial lemma:

THEOREM 31*h*: [Less-one lemma for unionset]

$$\bigcup \mathsf{M} = \mathsf{T} \setminus \{\mathsf{C}\} \ \&\ \mathsf{S} = \mathsf{T} \cup \mathsf{X} \cup \{\mathsf{V}\} \ \&\ \big(\mathsf{Y} = \mathsf{V} \vee (\mathsf{C} = \mathsf{Y} \ \&\ \mathsf{Y} \in \mathsf{S})\big) \rightarrow \langle \exists \mathsf{d} \mid \bigcup(\mathsf{M} \cup \{\mathsf{X} \cup \{\mathsf{Y}\}\}) = \mathsf{S} \setminus \{\mathsf{d}\} \rangle.$$ PROOF:

$$\mathsf{Suppose\_not}(\mathsf{m},\mathsf{t},\mathsf{c},\mathsf{s},\mathsf{x},\mathsf{v},\mathsf{y}) \Rightarrow \quad \mathit{Stat0}: \neg \langle \exists \mathsf{d} \mid \bigcup(\mathsf{m} \cup \{\mathsf{x} \cup \{\mathsf{y}\}\}) = \mathsf{s} \setminus \{\mathsf{d}\} \rangle \& \bigcup \mathsf{m} = \mathsf{t} \setminus \{\mathsf{c}\} \ \&\ \mathsf{s} = \mathsf{t} \cup \mathsf{x} \cup \{\mathsf{v}\} \ \&\ \big(\mathsf{y} = \mathsf{v} \vee (\mathsf{c} = \mathsf{y} \ \&\ \mathsf{y} \in \mathsf{s})\big)$$

For, supposing the contrary, $\bigcup(\mathsf{m} \cup \{\mathsf{x} \cup \{\mathsf{y}\}\})$ would differ from each of $\mathsf{s} \setminus \{\mathsf{s}\}$, $\mathsf{s} \setminus \{\mathsf{c}\}$, and $\mathsf{s} \setminus \{\mathsf{v}\}$, the first of which equals s. Thanks to Theorem 2e, we can rewrite $\bigcup(\mathsf{m} \cup \{\mathsf{x} \cup \{\mathsf{y}\}\})$ as $\mathsf{x} \cup \{\mathsf{y}\} \cup \bigcup \mathsf{m}$; but then the decision algorithm for a fragment of set theory known as ‘multi-level syllogistic with singleton’ yields an immediate contradiction.

$$\langle \mathsf{s} \rangle \hookrightarrow \mathit{Stat0} \Rightarrow \quad \bigcup(\mathsf{m} \cup \{\mathsf{x} \cup \{\mathsf{y}\}\}) \neq \mathsf{s}$$
$$\langle \mathsf{c} \rangle \hookrightarrow \mathit{Stat0} \Rightarrow \quad \bigcup(\mathsf{m} \cup \{\mathsf{x} \cup \{\mathsf{y}\}\}) \neq \mathsf{s} \setminus \{\mathsf{c}\}$$
$$\langle \mathsf{v} \rangle \hookrightarrow \mathit{Stat0} \Rightarrow \quad \bigcup(\mathsf{m} \cup \{\mathsf{x} \cup \{\mathsf{y}\}\}) \neq \mathsf{s} \setminus \{\mathsf{v}\}$$
$$\langle \mathsf{m}, \mathsf{x} \cup \{\mathsf{y}\} \rangle \hookrightarrow T2e \Rightarrow \quad \text{AUTO}$$
$$\mathsf{EQUAL} \Rightarrow \quad \mathit{Stat1}: \mathsf{x} \cup \{\mathsf{y}\} \cup \bigcup \mathsf{m} \neq \mathsf{s} \setminus \{\mathsf{c}\} \ \&\ \mathsf{x} \cup \{\mathsf{y}\} \cup \bigcup \mathsf{m} \neq \mathsf{s} \setminus \{\mathsf{v}\} \ \&\ \mathsf{x} \cup \{\mathsf{y}\} \cup \bigcup \mathsf{m} \neq \mathsf{s}$$
$$(\mathit{Stat0}, \mathit{Stat1})\mathsf{Discharge} \Rightarrow \quad \text{QED}$$

THEOREM 32: [Finite, non-null sets own sources]

$$\mathsf{Finite}(\mathsf{F}) \ \&\ \mathsf{F} \neq \emptyset \rightarrow \mathsf{F} \setminus \bigcup \mathsf{F} \neq \emptyset.$$ PROOF:

$$\mathsf{Suppose\_not}(\mathsf{f}_1) \Rightarrow \quad \text{AUTO}$$

Arguing by contradiction, suppose that there are counterexamples to the claim. Then, by exploiting finite induction, we can pick a minimal counterexample, f_0.

$$\mathsf{APPLY} \ \langle \mathsf{fin}_\Theta : \mathsf{f}_0 \rangle \ \mathsf{finiteInduction}\big(\mathsf{s}_0 \mapsto \mathsf{f}_1, \mathsf{P}(\mathsf{S}) \mapsto (\mathsf{S} \neq \emptyset \ \&\ \mathsf{S} \setminus \bigcup \mathsf{S} = \emptyset)\big) \Rightarrow$$
$$\mathit{Stat0}: \langle \forall \mathsf{s} \mid \mathsf{s} \subseteq \mathsf{f}_0 \rightarrow \mathsf{Finite}(\mathsf{s}) \ \&\ (\mathsf{s} \neq \emptyset \ \&\ \mathsf{s} \setminus \bigcup \mathsf{s} = \emptyset \leftrightarrow \mathsf{s} = \mathsf{f}_0) \rangle$$
$$\mathsf{Loc\_def} \Rightarrow \quad \mathsf{a} = \mathbf{arb}(\mathsf{f}_0)$$
$$\langle \mathsf{f}_0 \rangle \hookrightarrow \mathit{Stat0} \Rightarrow \quad \mathit{Stat1}: \mathsf{Finite}(\mathsf{f}_0) \ \&\ \mathsf{a} \in \mathsf{f}_0 \ \&\ \mathsf{f}_0 \setminus \bigcup \mathsf{f}_0 = \emptyset$$

Momentarily supposing that $\mathsf{f}_0 = \{\mathsf{a}\}$, one gets $\bigcup \mathsf{f}_0 \not\subseteq \mathsf{a}$, because $\bigcup \mathsf{f}_0 \subseteq \mathsf{a}$ would imply $\mathsf{f}_0 \setminus \bigcup \mathsf{f}_0 \supseteq \{\mathsf{a}\} \setminus \mathsf{a}$ and hence would imply the emptiness of $\{\mathsf{a}\} \setminus \mathsf{a}$, whence the manifest absurdity $\mathsf{a} \in \mathsf{a}$ follows. But, on the other hand, $\bigcup \{\mathsf{a}\} \subseteq \mathsf{a}$ trivially holds; therefore we must exclude that f_0 is a singleton $\{\mathsf{a}\}$.

$$\mathsf{Suppose} \Rightarrow \quad \mathsf{f}_0 = \{\mathsf{a}\} \ \&\ \bigcup \mathsf{f}_0 \not\subseteq \mathsf{a}$$
$$\mathsf{EQUAL} \Rightarrow \quad \bigcup \{\mathsf{a}\} \not\subseteq \mathsf{a}$$
$$\mathsf{Use\_def}(\bigcup) \Rightarrow \quad \{\mathsf{u} : \mathsf{v} \in \{\mathsf{a}\}, \mathsf{u} \in \mathsf{v}\} \not\subseteq \mathsf{a}$$
$$\mathsf{SIMPLF} \Rightarrow \quad \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow \quad \text{AUTO}$$

Due to our minimality assumption, the strict non-null subset $\mathsf{f}_0 \setminus \{\mathbf{arb}(\mathsf{f}_0)\}$ of f_0 cannot be a counterexample to the claim; therefore it has sources and hence $\mathsf{f}_0 \setminus \bigcup(\mathsf{f}_0 \setminus \{\mathbf{arb}(\mathsf{f}_0)\}) \neq \emptyset$.

$$\langle \mathsf{f}_0 \setminus \{\mathsf{a}\}, \mathsf{a} \rangle \hookrightarrow T2e(\star) \Rightarrow \quad \bigcup(\mathsf{f}_0 \setminus \{\mathsf{a}\} \cup \{\mathsf{a}\}) = \bigcup(\mathsf{f}_0 \setminus \{\mathsf{a}\}) \cup \mathsf{a} \ \& \ \mathsf{f}_0 \setminus \{\mathsf{a}\} \cup \{\mathsf{a}\} = \mathsf{f}_0$$
$$\langle \mathsf{f}_0 \setminus \{\mathsf{a}\} \rangle \hookrightarrow Stat0(\star) \Rightarrow \quad \mathsf{f}_0 \setminus \bigcup(\mathsf{f}_0 \setminus \{\mathsf{a}\}) \neq \emptyset$$

Since $\mathbf{arb}(\mathsf{f}_0)$ does not intersect f_0, the inequality just found conflicts with the equality $\mathsf{f}_0 \setminus \big(\bigcup(\mathsf{f}_0 \setminus \{\mathbf{arb}(\mathsf{f}_0)\}) \cup \mathbf{arb}(\mathsf{f}_0)\big) = \emptyset$ which one gets from THEOREM 2e through equality propagation.

$$\text{EQUAL} \Rightarrow \quad \mathsf{f}_0 \setminus \big(\bigcup(\mathsf{f}_0 \setminus \{\mathsf{a}\}) \cup \mathsf{a}\big) = \emptyset$$
$$\text{Discharge} \Rightarrow \quad \text{QED}$$

A.5 Claw-Free, Transitive Sets and Their Pivots

A *claw* is defined to be a pair Y, F such that (1) F has at least three elements, (2) no element of F belongs to any other element of F, (3) either Y belongs to all elements of F or there is a W in Y such that Y belongs to all elements of $\mathsf{F} \setminus \{\mathsf{W}\}$.

DEF claw: [Pair forming a claw, perhaps endowed with more than 3 el'ts]

$$\mathsf{Claw}(\mathsf{Y}, \mathsf{F}) \leftrightarrow_{\mathrm{Def}} \quad \mathsf{F} \cap \bigcup \mathsf{F} = \emptyset \ \& \ \langle \exists \mathsf{x}, \mathsf{z}, \mathsf{w} \mid \mathsf{F} \supseteq \{\mathsf{x}, \mathsf{z}, \mathsf{w}\} \ \& \ \mathsf{x} \neq \mathsf{z} \ \& \ \mathsf{w} \notin \{\mathsf{x}, \mathsf{z}\} \ \& \ \{\mathsf{w}\} \cap \mathsf{Y} \supseteq \{\mathsf{v} \in \mathsf{F} \mid \mathsf{Y} \notin \mathsf{v}\} \rangle$$

To really interest us, a claw-free set must be transitive: we omit this requirement in the definition given here below, but we will make it explicit in the major theorems pertaining to claw-freeness (Fig. 6).

DEF clawFreeness: [Claw-freeness in a membership digraph]

$$\mathsf{ClawFree}(\mathsf{S}) \leftrightarrow_{\mathrm{Def}} \langle \forall \mathsf{y} \in \mathsf{S}, \mathsf{e} \subseteq \mathsf{S} \mid \neg \mathsf{Claw}(\mathsf{y}, \mathsf{e}) \rangle$$

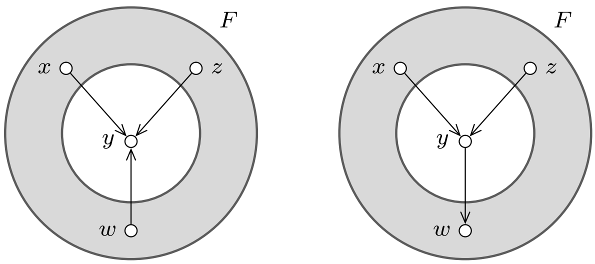


Fig. 6 The forbidden orientations of a claw in a claw-free set

THEOREM $\mathsf{clawFreeness_a}$: [Subsets of claw-free sets are claw-free]

$\mathsf{ClawFree}(S)\ \&\ T \subseteq S \rightarrow \mathsf{ClawFree}(T)$. PROOF:

$\mathsf{Suppose\_not}(s,t) \Rightarrow$ AUTO

$\mathsf{Use\_def}(\mathsf{ClawFree}) \Rightarrow$ $Stat1:\ \neg\langle\forall y \in t, e \subseteq t \mid \neg\mathsf{Claw}(y,e)\rangle\ \&$
$\langle\forall y \in s, e \subseteq s \mid \neg\mathsf{Claw}(y,e)\rangle$

$\langle y,e,y,e\rangle \hookrightarrow Stat1(\star) \Rightarrow$ false; $\mathsf{Discharge} \Rightarrow$ QED

THEOREM $\mathsf{clawFreeness_b}$: [Any potential claw must have a bypass]

$\mathsf{ClawFree}(S)\ \&\ S \supseteq \{Y,X,Z,W\}\ \&\ Y \in X \cap Z\ \&$
$W \in Y\ \&\ X \notin Z \cup \{Z\}\ \&\ Z \notin X \rightarrow W \in X \cup Z$. PROOF:

$\mathsf{Suppose\_not}(s,y,x,z,w) \Rightarrow$ AUTO

$\mathsf{Use\_def}(\mathsf{ClawFree}) \Rightarrow$ $Stat0:\ \langle\forall y \in s, e \subseteq s \mid \neg\mathsf{Claw}(y,e)\rangle\ \&$
$x \notin w\ \&\ z \notin w\ \&\ x \notin z\ \&\ w \notin x\ \&\ w \notin z\ \&$
$z \notin x\ \&\ x \neq z\ \&\ w \in y\ \&\ y \in x \cap z$

$\mathsf{Loc\_def} \Rightarrow$ $Stat1:\ e = \{x,z,w\}$

$\mathsf{Use\_def}(\mathsf{Claw}(y,e)) \Rightarrow$ AUTO

$\langle y,e\rangle \hookrightarrow Stat0(Stat1\star) \Rightarrow$ $\neg\Big(e \cap \bigcup e = \emptyset\ \&\ \langle\exists x,z,w \mid e \supseteq \{x,z,w\}\ \&$
$x \neq z\ \&\ w \notin \{x,z\}\ \&\ \{w\} \cap y \supseteq \{v \in e \mid y \notin v\}\rangle\Big)$

$\mathsf{EQUAL} \Rightarrow$ $\bigcup e = \bigcup\{x,z,w\}$

$\mathsf{Suppose} \Rightarrow$ $Stat2:\ e \cap \bigcup e \neq \emptyset$

$\mathsf{Use\_def}(\bigcup e) \Rightarrow$ AUTO

$\langle c\rangle \hookrightarrow Stat2(\star) \Rightarrow$ $Stat3:\ c \in \{u : v \in e, u \in v\}\ \&\ c \in e$

$\langle v_0,u_0\rangle \hookrightarrow Stat3(Stat1, Stat1\star) \Rightarrow$ $Stat4:\ v_0, c \in \{x,z,w\}\ \&\ c \in v_0$

$(Stat0, Stat4\star)\mathsf{Discharge} \Rightarrow$ $Stat5:\ \langle\exists x,z,w \mid e \supseteq \{x,z,w\}\ \&\ x \neq z\ \&$
$w \notin \{x,z\}\ \&\ \{w\} \cap y \supseteq \{v \in e \mid y \notin v\}\rangle$

$\langle x,z,w\rangle \hookrightarrow Stat5(Stat0\star) \Rightarrow$ $Stat6:\ \{w\} \cap y \not\supseteq \{v \in e \mid y \notin v\}$

$\langle d\rangle \hookrightarrow Stat6(Stat6\star) \Rightarrow$ $Stat7:\ d \in \{v \in e \mid y \notin v\}\ \&\ d \notin \{w\} \cap y$

$\langle\ \rangle \hookrightarrow Stat7(Stat1, Stat1\star) \Rightarrow$ $Stat8:\ d \in \{x,z,w\}\ \&\ y \notin d$

$(Stat0, Stat8, Stat7\star)\mathsf{Discharge} \Rightarrow$ QED

THEORY $\mathsf{pivotsForClawFreeness}(s_0)$

$\mathsf{ClawFree}(s_0)\ \&\ \mathsf{Finite}(s_0)\ \&\ \mathsf{Trans}(s_0)$

$s_0 \not\subseteq \{\emptyset\}$

END pivotsForClawFreeness

ENTER_THEORY pivotsForClawFreeness

By way of first approximation, we want to select from each finite transitive set s not included in $\{\emptyset\}$ a 'pivotal pair' consisting of an element x of maximum rank in s and an element y of maximum rank in x. To avoid introducing the recursive notion of rank of a set, we slightly generalize the idea: for any set s (not necessarily finite or transitive) we define the *frontier* of s to consist of those elements x of s which own elements y belonging to s such that the length of no membership chain issuing from y, ending in s, and contained in s ever exceeds 2. Any element y which is thus related to an element x of the frontier of s will be called a *pivot* of s.

DEF frontier: [Frontier of a set] $\mathsf{front}(\mathsf{S}) =_{\mathrm{Def}} \left\{\mathsf{x} \in \mathsf{S} \mid \mathsf{x} \cap \mathsf{S} \backslash \bigcup(\mathsf{S} \cap \bigcup \mathsf{S}) \neq \emptyset\right\}$

THEOREM $\mathsf{frontier}_1$: [Non-trivial finite sets have a non-null frontier]
$\mathsf{Finite}(\mathsf{S} \cap \bigcup \mathsf{S})$ & $\mathsf{S} \cap \bigcup \mathsf{S} \neq \emptyset \rightarrow \mathsf{front}(\mathsf{S}) \neq \emptyset$. PROOF:
$\mathsf{Suppose\_not}(\mathsf{s}) \Rightarrow$ AUTO
$\langle \mathsf{s} \cap \bigcup \mathsf{s} \rangle \hookrightarrow T32 \Rightarrow$ *Stat1* : $\mathsf{s} \cap \bigcup \mathsf{s} \backslash \bigcup(\mathsf{s} \cap \bigcup \mathsf{s}) \neq \emptyset$
$\mathsf{Use\_def}(\bigcup \mathsf{s}) \Rightarrow$ AUTO
$\langle \mathsf{y} \rangle \hookrightarrow Stat1 \Rightarrow$ *Stat2* : $\mathsf{y} \in \{\mathsf{u} : \mathsf{v} \in \mathsf{s}, \mathsf{u} \in \mathsf{v}\}$ & $\mathsf{y} \in \mathsf{s}$ & $\mathsf{y} \notin \bigcup(\mathsf{s} \cap \bigcup \mathsf{s})$
$\mathsf{Use\_def}(\mathsf{front}(\mathsf{s})) \Rightarrow$ AUTO
$\langle \mathsf{x}, \mathsf{u} \rangle \hookrightarrow Stat2 \Rightarrow$
Stat3 : $\mathsf{x} \notin \left\{\mathsf{x}_1 \in \mathsf{s} \mid \mathsf{x}_1 \cap \mathsf{s} \backslash \bigcup(\mathsf{s} \cap \bigcup \mathsf{s}) \neq \emptyset\right\}$ & $\mathsf{x} \in \mathsf{s}$ & $\mathsf{y} \in \mathsf{x}$
$\langle \mathsf{x} \rangle \hookrightarrow Stat3 \Rightarrow$ false; $\mathsf{Discharge} \Rightarrow$ QED

Our next claim is that if we choose a pivot element y of a transitive set s from an element of the frontier of s, then removal of all predecessors of y from s leads to a transitive set t such that y is a source of t.

THEOREM $\mathsf{frontier}_2$: [Transitivity-preserving reduction of a transitive set]
$\mathsf{Trans}(\mathsf{S})$ & $\mathsf{X} \in \mathsf{front}(\mathsf{S})$ & $\mathsf{Y} \in \mathsf{X} \backslash \bigcup \bigcup \mathsf{S}$ & $T = \{\mathsf{z} \in \mathsf{S} \mid \mathsf{Y} \notin \mathsf{z}\} \rightarrow$
$\mathsf{Trans}(T)$ & $T \subseteq \mathsf{S}$ & $\mathsf{X} \notin T$ & $\mathsf{Y} \in T \backslash \bigcup T$. PROOF:
$\mathsf{Suppose\_not}(\mathsf{s}, \mathsf{x}, \mathsf{y}, \mathsf{t}) \Rightarrow$ AUTO

Arguing by contradiction, let $\mathsf{s}, \mathsf{x}, \mathsf{y}, \mathsf{t}$ be a counterexample to the claim. Taking the definition of t into account to exploit monotonicity, we readily get $\mathsf{t} \subseteq \mathsf{s}$ and $\mathsf{x} \in \mathsf{t}$.

$\mathsf{Set\_monot} \Rightarrow$ $\{\mathsf{z} \in \mathsf{s} \mid \mathsf{y} \notin \mathsf{z}\} \subseteq \{\mathsf{z} : \mathsf{z} \in \mathsf{s}\}$
$\mathsf{Suppose} \Rightarrow$ *Stat0* : $\mathsf{x} \in \{\mathsf{z} \in \mathsf{s} \mid \mathsf{y} \notin \mathsf{z}\}$
$\langle\ \rangle \hookrightarrow Stat0 \Rightarrow$ false; $\mathsf{Discharge} \Rightarrow$ $\mathsf{x} \notin \mathsf{t}$

Now taking the definition of front into account, we can simplify our initial assumption to the following:

$\mathsf{Use\_def}(\mathsf{front}) \Rightarrow$ *Stat1* : $\mathsf{x} \in \left\{\mathsf{x}' \in \mathsf{s} \mid \mathsf{x}' \cap \mathsf{s} \backslash \bigcup(\mathsf{s} \cap \bigcup \mathsf{s}) \neq \emptyset\right\}$
$\langle\ \rangle \hookrightarrow Stat1 \Rightarrow$ $\mathsf{Trans}(\mathsf{s})$ & $\mathsf{x} \in \mathsf{s}$ & $\mathsf{x} \cap \mathsf{s} \backslash \bigcup(\mathsf{s} \cap \bigcup \mathsf{s}) \neq \emptyset$ & $\mathsf{y} \in \mathsf{x} \backslash \bigcup \bigcup \mathsf{s}$ &
$\mathsf{t} = \{\mathsf{z} \in \mathsf{s} \mid \mathsf{y} \notin \mathsf{z}\}$ & $\mathsf{Trans}(\mathsf{s})$ & $\neg\big(\mathsf{Trans}(\mathsf{t})$ & $\mathsf{y} \in \mathsf{t} \backslash \bigcup \mathsf{t}\big)$

Since s is transitive, if t were not transitive then by Theorem 4c $\mathsf{s}\backslash\mathsf{t}$ would have an element z not being a source of s. But then y would belong to $\mathsf{z} \in \bigcup\mathsf{s}$, which conflicts with y being a pivot.

$$
\begin{array}{l}
\mathsf{Suppose} \Rightarrow \quad \neg\mathsf{Trans}(\mathsf{t}) \\
\quad \langle \mathsf{s}, \mathsf{t}\rangle \hookrightarrow T4c \;\Rightarrow\quad \mathit{Stat2}:\ (\mathsf{s}\backslash\mathsf{t}) \cap \bigcup\mathsf{s} \neq \emptyset \\
\quad \mathsf{Use\_def}(\bigcup\mathsf{s}) \Rightarrow \quad \mathrm{AUTO} \\
\quad \langle \mathsf{z}\rangle \hookrightarrow \mathit{Stat2} \Rightarrow \quad \mathit{Stat3}:\ \mathsf{z} \in \left\{\mathsf{u}':\ \mathsf{w}' \in \mathsf{s}, \mathsf{u}' \in \mathsf{w}'\right\}\ \& \\
\qquad\qquad\qquad\qquad \mathsf{z} \notin \left\{\mathsf{z}' \in \mathsf{s} \mid \mathsf{y} \notin \mathsf{z}'\right\}\ \&\ \mathsf{z} \in \mathsf{s} \\
\quad \langle \mathsf{v}, \mathsf{a}, \mathsf{z}\rangle \hookrightarrow \mathit{Stat3}(\mathit{Stat3}\star) \Rightarrow \quad \mathsf{y} \in \mathsf{z}\ \&\ \mathsf{z} \in \mathsf{v}\ \&\ \mathsf{v} \in \mathsf{s} \\
\quad \mathsf{Use\_def}\big(\bigcup\bigcup\mathsf{s}\big) \Rightarrow \quad \mathrm{AUTO} \\
\quad \mathsf{EQUAL}(\mathit{Stat1}) \Rightarrow \quad \mathsf{y} \notin \left\{\mathsf{u}:\ \mathsf{w} \in \left\{\mathsf{u}':\ \mathsf{w}' \in \mathsf{s}, \mathsf{u}' \in \mathsf{w}'\right\}, \mathsf{u} \in \mathsf{w}\right\} \\
\quad \mathsf{SIMPLF} \Rightarrow \quad \mathit{Stat4}:\ \mathsf{y} \notin \left\{\mathsf{u}:\ \mathsf{w}' \in \mathsf{s}, \mathsf{w} \in \mathsf{w}', \mathsf{u} \in \mathsf{w}\right\} \\
\langle \mathsf{v}, \mathsf{z}, \mathsf{y}\rangle \hookrightarrow \mathit{Stat4}(\mathit{Stat1}\star) \Rightarrow \quad \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow \quad \mathsf{y} \in \bigcup\mathsf{t} \vee \mathsf{y} \notin \mathsf{t}
\end{array}
$$

Now knowing that $\mathsf{Trans}(\mathsf{t})$, we must consider the other possibility, namely that $\mathsf{y} \notin \mathsf{t}\backslash\bigcup\mathsf{t}$. However, after expanding t and $\bigcup\mathsf{t}$ according to their definitions, …

$$
\begin{array}{l}
\mathsf{Use\_def}(\bigcup\mathsf{t}) \Rightarrow \quad \mathrm{AUTO} \\
\mathsf{Use\_def}\big(\mathsf{Trans}(\mathsf{s})\big) \Rightarrow \quad \mathrm{AUTO} \\
\mathsf{EQUAL} \Rightarrow \quad \mathit{Stat5}:\ \{\mathsf{y} \in \mathsf{s} \mid \mathsf{y} \not\subseteq \mathsf{s}\} = \emptyset\ \& \\
\qquad\qquad \big(\mathsf{y} \in \{\mathsf{u}:\ \mathsf{v} \in \{\mathsf{z} \in \mathsf{s} \mid \mathsf{y} \notin \mathsf{z}\}, \mathsf{u} \in \mathsf{v}\} \vee \mathsf{y} \notin \{\mathsf{z} \in \mathsf{s} \mid \mathsf{y} \notin \mathsf{z}\}\big)
\end{array}
$$

… we see that neither one of the possibilities $\mathsf{y} \in \bigcup\mathsf{t}$, $\mathsf{y} \notin \mathsf{t}$ is tenable.

$$
\begin{array}{l}
\quad \langle \mathsf{x}\rangle \hookrightarrow \mathit{Stat5}(\mathit{Stat1}\star) \Rightarrow \quad \mathsf{y} \in \mathsf{s} \\
\quad \mathsf{SIMPLF} \Rightarrow \quad \mathit{Stat6}:\ \mathsf{y} \in \{\mathsf{u}:\ \mathsf{v} \in \mathsf{s}, \mathsf{u} \in \mathsf{v} \mid \mathsf{y} \notin \mathsf{v}\} \vee \mathsf{y} \notin \{\mathsf{z} \in \mathsf{s} \mid \mathsf{y} \notin \mathsf{z}\} \\
\langle \mathsf{w}, \mathsf{u}, \mathsf{y}\rangle \hookrightarrow \mathit{Stat6}(\mathit{Stat5}\star) \Rightarrow \quad \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow \quad \mathrm{QED}
\end{array}
$$

DEF $\mathsf{clawFreeness}_{\mathsf{frontEl}}$: [Frontier el't of a claw-free transitive non-trivial set]

$$\mathsf{x}_\Theta =_{\mathrm{Def}} \mathbf{arb}(\mathsf{front}(\mathsf{s}_0))$$

DEF $\mathsf{clawFreeness}_{\mathsf{pivotEl}}$: [Pivotal el't of a claw-free transitive non-trivial set]

$$\mathsf{y}_\Theta =_{\mathrm{Def}} \mathbf{arb}\big(\mathsf{x}_\Theta\backslash\bigcup\bigcup\mathsf{s}_0\big)$$

THEOREM $\mathsf{clawFreeness}_{\mathsf{c}}$: [$\mathsf{x}_\Theta$ truly belongs to the frontier of s_0]

$\mathsf{x}_\Theta \in \mathsf{front}(\mathsf{s}_0)\ \&\ \mathsf{x}_\Theta\backslash\bigcup\bigcup\mathsf{s}_0 \neq \emptyset\ \&\ \mathsf{x}_\Theta \in \mathsf{s}_0$. PROOF:

$$
\begin{array}{l}
\mathsf{Suppose\_not}() \Rightarrow \quad \mathrm{AUTO} \\
\quad \mathsf{Assump} \Rightarrow \quad \mathit{Stat0}:\ \mathsf{ClawFree}(\mathsf{s}_0)\ \&\ \mathsf{Finite}(\mathsf{s}_0 \cap \bigcup\mathsf{s}_0)\ \&\ \mathsf{Trans}(\mathsf{s}_0)\ \& \\
\qquad\qquad\qquad\qquad \mathsf{s}_0 \not\subseteq \{\emptyset\} \\
\quad \langle \mathsf{s}_0\rangle \hookrightarrow T3a \;\Rightarrow\quad \mathsf{s}_0 \cap \bigcup\mathsf{s}_0 = \bigcup\mathsf{s}_0 \\
\quad \langle \mathsf{s}_0\rangle \hookrightarrow T31d \;\Rightarrow\quad \mathsf{s}_0 \cap \bigcup\mathsf{s}_0 \neq \emptyset \\
\quad \langle \mathsf{s}_0\rangle \hookrightarrow T\mathsf{frontier}_1 \;\Rightarrow\quad \mathit{Stat1}:\ \mathsf{front}(\mathsf{s}_0) \neq \emptyset \\
\quad \mathsf{Use\_def}\big(\mathsf{front}(\mathsf{s}_0)\big) \Rightarrow \quad \mathrm{AUTO} \\
\quad \mathsf{Use\_def}(\mathsf{x}_\Theta) \Rightarrow \quad \mathit{Stat2}:\ \mathsf{x}_\Theta \in \left\{\mathsf{x} \in \mathsf{s}_0 \mid \mathsf{x} \cap \mathsf{s}_0\backslash\bigcup(\mathsf{s}_0 \cap \bigcup\mathsf{s}_0) \neq \emptyset\right\}\ \& \\
\qquad\qquad\qquad\qquad \mathsf{x}_\Theta \in \mathsf{front}(\mathsf{s}_0)
\end{array}
$$

$\langle\ \rangle \hookrightarrow Stat2(Stat2\star) \Rightarrow \quad \mathsf{x}_\Theta \in \mathsf{s}_0 \ \&\ \mathsf{x}_\Theta \setminus \bigcup(\mathsf{s}_0 \cap \bigcup \mathsf{s}_0) \neq \emptyset$

EQUAL $\Rightarrow$ false; Discharge $\Rightarrow$ QED

Pivotal elements, in a transitive claw-free set such as the one treated in this THEORY, own at most two predecessors.

THEOREM clawFreeness$_0$: [Pivots own at most two predecessors]
$\mathsf{Y} \in \mathsf{X} \setminus \bigcup\bigcup \mathsf{s}_0 \ \&\ \mathsf{X} \in \mathsf{s}_0 \rightarrow \langle \exists \mathsf{z} \mid \{\mathsf{v} \in \mathsf{s}_0 \mid \mathsf{Y} \in \mathsf{v}\} = \{\mathsf{X}, \mathsf{z}\} \rangle$. PROOF:

Suppose_not(y, x) $\Rightarrow$ $Stat1: \ \neg\langle \exists \mathsf{z} \mid \{\mathsf{v} \in \mathsf{s}_0 \mid \mathsf{y} \in \mathsf{v}\} = \{\mathsf{x}, \mathsf{z}\} \rangle \ \&$
$\mathsf{x} \in \mathsf{s}_0 \ \&\ \mathsf{y} \in \mathsf{x} \setminus \bigcup\bigcup \mathsf{s}_0$

Suppose that y, x constitute a counter-example, so that y has, in addition to x, at least two predecessors z and w in s_0.

Suppose $\Rightarrow$ $Stat2: \ \mathsf{x} \notin \{\mathsf{v} \in \mathsf{s}_0 \mid \mathsf{y} \in \mathsf{v}\}$

$\langle \mathsf{x} \rangle \hookrightarrow Stat2(\star) \Rightarrow$ false; Discharge $\Rightarrow$ AUTO

$\langle \mathsf{x} \rangle \hookrightarrow Stat1(\star) \Rightarrow$ $Stat3: \ \{\mathsf{v} \in \mathsf{s}_0 \mid \mathsf{y} \in \mathsf{v}\} \neq \{\mathsf{x}\}$

$\langle \mathsf{z} \rangle \hookrightarrow Stat3(\star) \Rightarrow$ $Stat4: \ \mathsf{z} \in \{\mathsf{v} \in \mathsf{s}_0 \mid \mathsf{y} \in \mathsf{v}\} \ \&\ \mathsf{x} \neq \mathsf{z}$

$\langle\ \rangle \hookrightarrow Stat4(Stat4\star) \Rightarrow$ $Stat5: \ \mathsf{z} \in \mathsf{s}_0 \ \&\ \mathsf{y} \in \mathsf{z}$

$\langle \mathsf{z} \rangle \hookrightarrow Stat1(\star) \Rightarrow$ $Stat6: \ \{\mathsf{v} \in \mathsf{s}_0 \mid \mathsf{y} \in \mathsf{v}\} \neq \{\mathsf{z}, \mathsf{x}\}$

$\langle \mathsf{w} \rangle \hookrightarrow Stat6(\star) \Rightarrow$ $Stat7: \ \mathsf{w} \in \{\mathsf{v} \in \mathsf{s}_0 \mid \mathsf{y} \in \mathsf{v}\} \ \&\ \mathsf{w} \notin \{\mathsf{x}, \mathsf{z}\}$

$\langle\ \rangle \hookrightarrow Stat7(Stat7\star) \Rightarrow$ $Stat8: \ \mathsf{w} \in \mathsf{s}_0 \ \&\ \mathsf{y} \in \mathsf{w}$

Loc_def $\Rightarrow$ $\mathsf{e} = \{\mathsf{x}, \mathsf{z}, \mathsf{w}\}$

Suppose $\Rightarrow$ $Stat9: \ \{\mathsf{v} \in \mathsf{e} \mid \mathsf{y} \notin \mathsf{v}\} \neq \emptyset$

$\langle \mathsf{v} \rangle \hookrightarrow Stat9(\star) \Rightarrow$ false; Discharge $\Rightarrow$ AUTO

The transitivity of s_0, since $\mathsf{y} \in \mathsf{x}$ and $\mathsf{x} \in \mathsf{s}_0$, implies that $\mathsf{y} \in \mathsf{s}_0$; therefore, in view of the claw-freeness of s_0, y and $\mathsf{e} = \{\mathsf{x}, \mathsf{z}, \mathsf{w}\}$ do not form a claw.

Assump $\Rightarrow$ $\mathsf{ClawFree}(\mathsf{s}_0) \ \&\ \mathsf{Trans}(\mathsf{s}_0)$

Use_def(ClawFree) $\Rightarrow$ $Stat10: \ \langle \forall \mathsf{y} \in \mathsf{s}_0, \mathsf{e} \subseteq \mathsf{s}_0 \mid \neg \mathsf{Claw}(\mathsf{y}, \mathsf{e}) \rangle$

$\langle \mathsf{s}_0, \mathsf{x} \rangle \hookrightarrow T3c(\star) \Rightarrow$ $\mathsf{y} \in \mathsf{s}_0$

$\langle \mathsf{y}, \mathsf{e} \rangle \hookrightarrow Stat10(\star) \Rightarrow$ $\neg \mathsf{Claw}(\mathsf{y}, \mathsf{e})$

It readily follows from the definition of claw that $\{\mathsf{x}, \mathsf{z}, \mathsf{w}\}$ and $\bigcup \{\mathsf{x}, \mathsf{z}, \mathsf{w}\}$ intersect; therefore, we can pick an element a common to the two.

Use_def(Claw) $\Rightarrow$ $Stat11: \neg\langle \exists \mathsf{x}, \mathsf{z}, \mathsf{w} \mid \mathsf{e} \supseteq \{\mathsf{x}, \mathsf{z}, \mathsf{w}\} \ \&\ \mathsf{x} \neq \mathsf{z} \ \&\ \mathsf{w} \notin \{\mathsf{x}, \mathsf{z}\}$
$\&\ \{\mathsf{w}\} \cap \mathsf{y} \supseteq \{\mathsf{v} \in \mathsf{e} \mid \mathsf{y} \notin \mathsf{v}\} \rangle \vee \mathsf{e} \cap \bigcup \mathsf{e} \neq \emptyset$

$\langle \mathsf{x}, \mathsf{z}, \mathsf{w} \rangle \hookrightarrow Stat11(Stat4\star) \Rightarrow$ $\mathsf{e} \cap \bigcup \mathsf{e} \neq \emptyset$

$\mathsf{EQUAL}(\mathit{Stat8}) \Rightarrow \quad \mathit{Stat12}: \{x,z,w\} \cap \bigcup\{x,z,w\} \neq \emptyset$
$\langle a\rangle \hookrightarrow \mathit{Stat12}(\mathit{Stat12}\star) \Rightarrow \quad \mathit{Stat13}: a \in \{x,z,w\} \;\&\; a \in \bigcup\{x,z,w\}$

‖ But then $y \in a$, $a \subseteq \bigcup\bigcup\{x,z,w\}$, and $\bigcup\bigcup\{x,z,w\} \subseteq \bigcup\bigcup s_0$ must hold, implying that $y \in \bigcup\bigcup s_0$; but we have started with the assumption that $y \notin \bigcup\bigcup s_0$. This contradiction proves the claim.

$\langle \{x,z,w\}, s_0\rangle \hookrightarrow T2c(\mathit{Stat1}, \mathit{Stat5}, \mathit{Stat8}, \mathit{Stat13}\star) \Rightarrow \quad y \in a \;\&$
$\bigcup s_0 \supseteq \bigcup\{x,z,w\}$
$\langle \bigcup\{x,z,w\}, \bigcup s_0\rangle \hookrightarrow T2c(\mathit{Stat13}\star) \Rightarrow \quad \bigcup\bigcup s_0 \supseteq \bigcup\bigcup\{x,z,w\}$
$\langle a, \bigcup\{x,z,w\}\rangle \hookrightarrow T2\,(\mathit{Stat13}\star) \Rightarrow \quad \mathit{Stat14}: y \in \bigcup\bigcup s_0$
$(\mathit{Stat1}, \mathit{Stat14}\star)\mathsf{Discharge} \Rightarrow \quad$ QED

THEOREM $\mathsf{clawFreeness_d}$: [Shape of the frontier at a pivotal pair]
$\langle \exists z \mid \{v \in s_0 \mid y_\Theta \in v\} = \{x_\Theta, z\} \;\&\; y_\Theta \in z\rangle$. PROOF:
$\mathsf{Suppose\_not}() \Rightarrow \quad$ AUTO
$\langle\,\rangle \hookrightarrow T\,\mathsf{clawFreeness_c} \Rightarrow \quad \mathit{Stat1}: x_\Theta \setminus \bigcup\bigcup s_0 \neq \emptyset \;\&\; x_\Theta \in s_0$
$\mathsf{Use\_def}(y_\Theta) \Rightarrow \quad y_\Theta \in x_\Theta \setminus \bigcup\bigcup s_0$
$\langle y_\Theta, x_\Theta\rangle \hookrightarrow T\,\mathsf{clawFreeness_0} \Rightarrow$
$\mathit{Stat2}: \langle \exists z \mid \{v \in s_0 \mid y_\Theta \in v\} = \{x_\Theta, z\}\rangle \;\&$
$\neg\langle \exists z \mid \{v \in s_0 \mid y_\Theta \in v\} = \{x_\Theta, z\} \;\&\; y_\Theta \in z\rangle$
$\langle z_0, z_0\rangle \hookrightarrow \mathit{Stat2} \Rightarrow \quad \mathit{Stat3}: z_0 \in \{v \in s_0 \mid y_\Theta \in v\} \;\&\; y_\Theta \notin z_0$
$\langle\,\rangle \hookrightarrow \mathit{Stat3} \Rightarrow \quad \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow \quad$ QED

‖ Via Skolemization, we give a name to the third item in a pivotal tripleton:

$\mathsf{APPLY}\ \langle v1_\Theta : z_\Theta\rangle\ \mathsf{Skolem} \Rightarrow$
THEOREM $\mathsf{clawFreeness_e}$. $\{v \in s_0 \mid y_\Theta \in v\} = \{x_\Theta, z_\Theta\} \;\&\; y_\Theta \in z_\Theta$.

THEOREM $\mathsf{clawFreeness_f}$: [Tripleton pivot in claw-free, transitive set]
$\{v \in s_0 \mid y_\Theta \in v\} = \{x_\Theta, z_\Theta\} \;\&\; \{x_\Theta, y_\Theta, z_\Theta\} \subseteq s_0 \;\&$
$y_\Theta \in x_\Theta \cap z_\Theta \setminus \bigcup\bigcup s_0 \;\&\; x_\Theta \notin z_\Theta \;\&\; z_\Theta \notin x_\Theta$. PROOF:
$\mathsf{Suppose\_not}() \Rightarrow \quad$ AUTO
$\langle\,\rangle \hookrightarrow T\,\mathsf{clawFreeness_c} \Rightarrow \quad \mathit{Stat3}: x_\Theta \setminus \bigcup\bigcup s_0 \neq \emptyset$
$\mathsf{Use\_def}(y_\Theta) \Rightarrow \quad y_\Theta \notin \bigcup\bigcup s_0 \;\&\; y_\Theta \in x_\Theta$
$\langle\,\rangle \hookrightarrow T\,\mathsf{clawFreeness_e} \Rightarrow \quad \mathit{Stat1}: x_\Theta \in \{v : v \in s_0 \mid y_\Theta \in v\} \;\&$
$z_\Theta \in \{v \in s_0 \mid y_\Theta \in v\} \;\&$
$\{v : v \in s_0 \mid y_\Theta \in v\} = \{x_\Theta, z_\Theta\} \;\&\; y_\Theta \in z_\Theta$
$\langle v_0, v_1\rangle \hookrightarrow \mathit{Stat1} \Rightarrow \quad x_\Theta \in s_0 \;\&\; z_\Theta \in s_0$
$\mathsf{Assump} \Rightarrow \quad \mathsf{Trans}(s_0)$
$\langle s_0, z_\Theta\rangle \hookrightarrow T3c \Rightarrow \quad y_\Theta \in s_0$
$\langle s_0\rangle \hookrightarrow T3a \Rightarrow \quad s_0 \cap \bigcup s_0 = \bigcup s_0$
$\mathsf{EQUAL} \Rightarrow \quad y_\Theta \notin \bigcup(s_0 \cap \bigcup s_0)$

$\langle y_\Theta, x_\Theta, z_\Theta, s_0\rangle \hookrightarrow T31g \Rightarrow \quad x_\Theta \notin z_\Theta$
$\langle y_\Theta, z_\Theta, x_\Theta, s_0\rangle \hookrightarrow T31g \Rightarrow \quad$ false; Discharge $\Rightarrow$ QED

DEF $\mathsf{clawFreeness_{rmv}}$: [Removing el’ts above pivot] $t_\Theta =_{\mathrm{Def}} \{v \in s_0 \mid y_\Theta \notin v\}$

The removal of the predecessors of a pivot from a claw-free, transitive non-trivial set such as the one treated by this THEORY does not disrupt transitivity.

THEOREM $\mathsf{clawFreeness_g}$: [Removing el’ts above pivot preserves transitivity]
$\mathrm{Trans}(t_\Theta)$ & $\mathrm{ClawFree}(t_\Theta)$ & $t_\Theta \subseteq s_0$ & $x_\Theta \notin t_\Theta$ & $y_\Theta \in t_\Theta \setminus \bigcup t_\Theta$ &
$t_\Theta = s_0 \setminus \{x_\Theta, z_\Theta\}$. PROOF:
Suppose_not() $\Rightarrow$ AUTO
Use_def(t_Θ) $\Rightarrow$ $\mathit{Stat1}: \ t_\Theta = \{v \in s_0 \mid y_\Theta \notin v\}$
Set_monot $\Rightarrow$ $\{v \in s_0 \mid y_\Theta \notin v\} \subseteq \{v : v \in s_0\}$
Assump $\Rightarrow$ $\mathrm{Trans}(s_0)$ & $\mathrm{ClawFree}(s_0)$
$\langle s_0, t_\Theta\rangle \hookrightarrow T\,\mathsf{clawFreeness_a}(\mathit{Stat1}\star) \Rightarrow$ $\mathrm{ClawFree}(t_\Theta)$
$\langle\ \rangle \hookrightarrow T\,\mathsf{clawFreeness_c} \Rightarrow$ $x_\Theta \in \mathrm{front}(s_0)$
$\langle\ \rangle \hookrightarrow T\,\mathsf{clawFreeness_f} \Rightarrow$ $\mathit{Stat2}: \ \{v \in s_0 \mid y_\Theta \in v\} = \{x_\Theta, z_\Theta\}$ &
$y_\Theta \in x_\Theta \setminus \bigcup\bigcup s_0$ & $y_\Theta \notin \bigcup\bigcup s_0$
$\langle s_0, x_\Theta, y_\Theta, t_\Theta\rangle \hookrightarrow T\,\mathsf{frontier_2}(\star) \Rightarrow$ $\mathit{Stat3}: \ t_\Theta \neq s_0 \setminus \{x_\Theta, z_\Theta\}$ &
$\mathrm{Trans}(t_\Theta)$ & $t_\Theta \subseteq s_0$ & $x_\Theta \notin t_\Theta$ & $y_\Theta \in t_\Theta \setminus \bigcup t_\Theta$
$\langle e\rangle \hookrightarrow \mathit{Stat3}(\mathit{Stat3}\star) \Rightarrow$ $e \in t_\Theta \neq e \in s_0 \setminus \{x_\Theta, z_\Theta\}$
Suppose $\Rightarrow$ $\mathit{Stat4}: \ e \in \{v \in s_0 \mid y_\Theta \notin v\}$ & $e \notin s_0 \setminus \{x_\Theta, z_\Theta\}$
$\langle\ \rangle \hookrightarrow \mathit{Stat4}(\mathit{Stat2}\star) \Rightarrow$ $\mathit{Stat5}: \ e \in \{v \in s_0 \mid y_\Theta \in v\}$ & $e \in s_0$ & $y_\Theta \notin e$
$\langle\ \rangle \hookrightarrow \mathit{Stat5}(\mathit{Stat5}\star) \Rightarrow$ false; Discharge $\Rightarrow$
$\mathit{Stat6}: \ e \notin \{v \in s_0 \mid y_\Theta \in v\}$ & $e \notin \{v \in s_0 \mid y_\Theta \notin v\}$ & $e \in s_0$
$\langle e, e\rangle \hookrightarrow \mathit{Stat6}(\mathit{Stat6}\star) \Rightarrow$ false; Discharge $\Rightarrow$ QED

ENTER_THEORY Set_theory
DISPLAY pivotsForClawFreeness

THEORY pivotsForClawFreeness(s_0)
$\mathrm{ClawFree}(s_0)$ & $\mathrm{Finite}(s_0)$ & $\mathrm{Trans}(s_0)$
$s_0 \not\subseteq \{\emptyset\}$
$\Rightarrow (x_\Theta, y_\Theta, z_\Theta, t_\Theta)$
$\langle \forall y, x \mid y \in x \setminus \bigcup\bigcup s_0$ & $x \in s_0 \rightarrow \langle \exists z \mid \{v \in s_0 \mid y \in v\} = \{x, z\}\rangle\rangle$
$\{v \in s_0 \mid y_\Theta \in v\} = \{x_\Theta, z_\Theta\}$ & $\{x_\Theta, y_\Theta, z_\Theta\} \subseteq s_0$ & $y_\Theta \in x_\Theta \cap z_\Theta \setminus \bigcup\bigcup s_0$ &
$x_\Theta \notin z_\Theta$ & $z_\Theta \notin x_\Theta$
$t_\Theta = \{v \in s_0 \mid y_\Theta \notin v\}$
$\mathrm{ClawFree}(t_\Theta)$ & $\mathrm{Trans}(t_\Theta)$ & $t_\Theta \subseteq s_0$ & $x_\Theta \notin t_\Theta$ & $y_\Theta \in t_\Theta \setminus \bigcup t_\Theta$ &
$t_\Theta = s_0 \setminus \{x_\Theta, z_\Theta\}$
END pivotsForClawFreeness

A.6 Hanks, Cycles, and Hamiltonian Cycles

The following notion approximately models the concept of a graph where every vertex has at least two incident edges. However, we neither require that (1) edges be doubletons, nor that (2) the set H of edges and the one of vertices—which is understood to be $\bigcup\mathsf{H}$—be disjoint.

DEF cycle_0: [Collection of edges whose endpoints have degree greater than 1]

$$\mathsf{Hank}(\mathsf{H}) \leftrightarrow_{\mathrm{Def}} \emptyset \notin \mathsf{H} \,\&\, \langle \forall \mathsf{e} \in \mathsf{H} \mid \mathsf{e} \subseteq \bigcup(\mathsf{H} \setminus \{\mathsf{e}\}) \rangle$$

DEF cycle_1: [Cycle (unless null)]

$$\mathsf{Cycle}(\mathsf{C}) \leftrightarrow_{\mathrm{Def}} \mathsf{Hank}(\mathsf{C}) \,\&\, \langle \forall \mathsf{d} \subseteq \mathsf{C} \mid \mathsf{Hank}(\mathsf{d}) \,\&\, \mathsf{d} \neq \emptyset \rightarrow \mathsf{d} = \mathsf{C} \rangle$$

THEOREM hank_0: [Alternative characterization of a hank]
$\mathsf{Hank}(\mathsf{H}) \leftrightarrow \big(\emptyset \notin \mathsf{H} \,\&\, \langle \forall \mathsf{e} \in \mathsf{H}, \mathsf{x} \in \mathsf{e}, \exists \mathsf{q} \in \mathsf{H} \mid \mathsf{q} \neq \mathsf{e} \,\&\, \mathsf{x} \in \mathsf{q} \rangle\big)$. PROOF:

$\mathsf{Suppose\_not}(\mathsf{h}) \Rightarrow$ AUTO

$\mathsf{Suppose} \Rightarrow \neg\langle \forall \mathsf{e} \in \mathsf{h}, \mathsf{x} \in \mathsf{e}, \exists \mathsf{q} \in \mathsf{h} \mid \mathsf{q} \neq \mathsf{e} \,\&\, \mathsf{x} \in \mathsf{q} \rangle \,\&$
$\quad \langle \forall \mathsf{e} \in \mathsf{h} \mid \mathsf{e} \subseteq \bigcup(\mathsf{h} \setminus \{\mathsf{e}\}) \rangle$

$\mathsf{Use\_def}(\bigcup) \Rightarrow$ *Stat1* : $\neg\langle \forall \mathsf{e} \in \mathsf{h}, \mathsf{x} \in \mathsf{e}, \exists \mathsf{q} \in \mathsf{h} \mid \mathsf{q} \neq \mathsf{e} \,\&\, \mathsf{x} \in \mathsf{q} \rangle \,\&$
$\quad \langle \forall \mathsf{e} \in \mathsf{h} \mid \mathsf{e} \subseteq \{\mathsf{v} : \mathsf{u} \in \mathsf{h} \setminus \{\mathsf{e}_0\}, \mathsf{v} \in \mathsf{u}\} \rangle$

$\langle \mathsf{e}_0, \mathsf{x}_0, \mathsf{e}_0 \rangle \hookrightarrow$ *Stat1* $\Rightarrow$ *Stat2* : $\mathsf{x}_0 \in \{\mathsf{v} : \mathsf{u} \in \mathsf{h} \setminus \{\mathsf{e}_0\}, \mathsf{v} \in \mathsf{u}\} \,\&$
$\quad \neg\langle \exists \mathsf{q} \in \mathsf{h} \mid \mathsf{q} \neq \mathsf{e}_0 \,\&\, \mathsf{x}_0 \in \mathsf{q} \rangle \,\&\, \mathsf{e}_0 \in \mathsf{h} \,\&\, \mathsf{x}_0 \in \mathsf{e}_0$

$\langle \mathsf{q}_0, \mathsf{v}_0, \mathsf{q}_0 \rangle \hookrightarrow$ *Stat2* $\Rightarrow$ false; $\quad \mathsf{Discharge} \Rightarrow$ AUTO

$\mathsf{Use\_def}(\mathsf{Hank}) \Rightarrow$ *Stat3* : $\neg\langle \forall \mathsf{e} \in \mathsf{h} \mid \mathsf{e} \subseteq \bigcup(\mathsf{h} \setminus \{\mathsf{e}\}) \rangle \,\&$
$\quad \langle \forall \mathsf{e} \in \mathsf{h}, \mathsf{x} \in \mathsf{e}, \exists \mathsf{q} \in \mathsf{h} \mid \mathsf{q} \neq \mathsf{e} \,\&\, \mathsf{x} \in \mathsf{q} \rangle$

$\langle \mathsf{e}_1 \rangle \hookrightarrow$ *Stat3* $\Rightarrow$ *Stat4* : $\mathsf{e}_1 \not\subseteq \bigcup(\mathsf{h} \setminus \{\mathsf{e}_1\}) \,\&$
$\quad \langle \forall \mathsf{e} \in \mathsf{h}, \mathsf{x} \in \mathsf{e}, \exists \mathsf{q} \in \mathsf{h} \mid \mathsf{q} \neq \mathsf{e} \,\&\, \mathsf{x} \in \mathsf{q} \rangle \,\&\, \mathsf{e}_1 \in \mathsf{h}$

$\mathsf{Use\_def}\big(\bigcup(\mathsf{h} \setminus \{\mathsf{e}_1\})\big) \Rightarrow$ AUTO

$\langle \mathsf{x}_1, \mathsf{e}_1, \mathsf{x}_1 \rangle \hookrightarrow$ *Stat4* $\Rightarrow$ *Stat5* : $\langle \exists \mathsf{q} \in \mathsf{h} \mid \mathsf{q} \neq \mathsf{e}_1 \,\&\, \mathsf{x}_1 \in \mathsf{q} \rangle \,\&$
$\quad \mathsf{x}_1 \notin \{\mathsf{v} : \mathsf{u} \in \mathsf{h} \setminus \{\mathsf{e}_1\}, \mathsf{v} \in \mathsf{u}\} \,\&\, \mathsf{x}_1 \in \mathsf{e}_1$

$\langle \mathsf{q}_1, \mathsf{q}_1, \mathsf{x}_1 \rangle \hookrightarrow$ *Stat5* $\Rightarrow$ false; $\quad \mathsf{Discharge} \Rightarrow$ QED

THEOREM hank_1: [No singleton or doubleton set is a cycle]
$\mathsf{H} \subseteq \{\mathsf{X}, \mathsf{U}\} \,\&\, \mathsf{Hank}(\mathsf{H}) \rightarrow \mathsf{H} = \emptyset$. PROOF:

$\mathsf{Suppose\_not}(\mathsf{h}_0, \mathsf{x}_0, \mathsf{u}_0) \Rightarrow$ *Stat0* : $\mathsf{h}_0 \neq \emptyset \,\&\, \mathsf{h}_0 \subseteq \{\mathsf{x}_0, \mathsf{u}_0\} \,\&\, \mathsf{Hank}(\mathsf{h}_0)$

For, assuming that h_0 is a hank, non-null, and a subset of a doubleton $\{\mathsf{x}_0, \mathsf{u}_0\}$, we will reach a contradiction arguing as follows. If a is one of the (at most two) elements of h_0, since $\emptyset \neq \mathsf{a} \,\&\, \mathsf{a} \subseteq \bigcup(\mathsf{h}_0 \setminus \{\mathsf{a}\})$ ensues from the definition of hank, $\bigcup(\mathsf{h}_0 \setminus \{\mathsf{a}\})$ must be non-null; hence $\mathsf{h}_0 = \{\mathsf{a}, \mathsf{b}\}$, where $\mathsf{b} \neq \mathsf{a}$. But then $\bigcup(\mathsf{h}_0 \setminus \{\mathsf{a}\}) = \bigcup\{\mathsf{b}\}$ and $\bigcup(\mathsf{h}_0 \setminus \{\mathsf{b}\}) = \bigcup\{\mathsf{a}\}$, i.e., $\bigcup(\mathsf{h}_0 \setminus \{\mathsf{a}\}) = \mathsf{b}$ and $\bigcup(\mathsf{h}_0 \setminus \{\mathsf{b}\}) = \mathsf{a}$; therefore $\mathsf{a} \subseteq \mathsf{b}$ and $\mathsf{b} \subseteq \mathsf{a}$ ensue from the definition of hank, leading us to the identity $\mathsf{a} = \mathsf{b}$, which contradicts an earlier inequality.

$\langle \mathsf{a} \rangle \hookrightarrow Stat0(Stat0\star) \Rightarrow \quad Stat1: \ \mathsf{a} \in \mathsf{h}_0$
$\mathsf{Use\_def(Hank)} \Rightarrow \quad Stat2: \ \langle \forall \mathsf{e} \in \mathsf{h}_0 \mid \mathsf{e} \subseteq \bigcup(\mathsf{h}_0 \setminus \{\mathsf{e}\}) \rangle \ \& \ \emptyset \notin \mathsf{h}_0$
$\langle \mathsf{a} \rangle \hookrightarrow Stat2 \Rightarrow \quad \mathsf{a} \subseteq \bigcup(\mathsf{h}_0 \setminus \{\mathsf{a}\})$
$\langle \mathsf{h}_0 \setminus \{\mathsf{a}\} \rangle \hookrightarrow T31d \ \Rightarrow \quad Stat3: \ \mathsf{h}_0 \neq \{\mathsf{a}\}$
$\langle \mathsf{b} \rangle \hookrightarrow Stat3 \Rightarrow \quad Stat4: \ \mathsf{b} \in \mathsf{h}_0 \ \& \ \mathsf{b} \neq \mathsf{a}$
$\langle \mathsf{b} \rangle \hookrightarrow Stat2 \Rightarrow \quad \mathsf{b} \subseteq \bigcup(\mathsf{h}_0 \setminus \{\mathsf{b}\})$
$\langle \{\mathsf{a}\}, \mathsf{a}, \mathsf{a} \rangle \hookrightarrow T2a \ \Rightarrow \quad \bigcup\{\mathsf{a}\} = \mathsf{a}$
$\langle \{\mathsf{b}\}, \mathsf{b}, \mathsf{b} \rangle \hookrightarrow T2a \ \Rightarrow \quad \bigcup\{\mathsf{b}\} = \mathsf{b}$
$(Stat0, Stat1, Stat4\star)\mathsf{ELEM} \Rightarrow \quad \mathsf{h}_0 \setminus \{\mathsf{a}\} = \{\mathsf{b}\} \ \& \ \mathsf{h}_0 \setminus \{\mathsf{b}\} = \{\mathsf{a}\}$
$\mathsf{EQUAL} \Rightarrow \quad \mathsf{false};$ $\qquad \mathsf{Discharge} \Rightarrow \quad$ QED

The following is the basic case of a general theorem scheme where the length of the chain can be any number > 2.

THEOREM hank_2: [A membership chain and an extra edge form a hank]
$\mathsf{X} \in \mathsf{Y} \ \& \ \mathsf{Y} \in \mathsf{Z} \rightarrow \mathsf{Hank}(\{\{\mathsf{X},\mathsf{Y}\}, \{\mathsf{Y},\mathsf{Z}\}, \{\mathsf{Z},\mathsf{X}\}\})$. PROOF:
$\mathsf{Suppose\_not}(\mathsf{x}_0, \mathsf{y}_0, \mathsf{z}_0) \Rightarrow \quad$ AUTO
$\mathsf{Use\_def(Hank)} \Rightarrow \quad Stat0: \ \neg\langle \forall \mathsf{e} \in \{\{\mathsf{x}_0,\mathsf{y}_0\}, \{\mathsf{y}_0,\mathsf{z}_0\}, \{\mathsf{z}_0,\mathsf{x}_0\}\} \mid$
$\mathsf{e} \subseteq \bigcup(\{\{\mathsf{x}_0,\mathsf{y}_0\}, \{\mathsf{y}_0,\mathsf{z}_0\}, \{\mathsf{z}_0,\mathsf{x}_0\}\} \setminus \{\mathsf{e}\}) \rangle \ \& \ \mathsf{x}_0 \in \mathsf{y}_0 \ \& \ \mathsf{y}_0 \in \mathsf{z}_0$
$\langle \mathsf{e}_0 \rangle \hookrightarrow Stat0 \Rightarrow \quad \mathsf{e}_0 \in \{\{\mathsf{x}_0,\mathsf{y}_0\}, \{\mathsf{y}_0,\mathsf{z}_0\}, \{\mathsf{z}_0,\mathsf{x}_0\}\} \ \&$
$\mathsf{e}_0 \nsubseteq \bigcup(\{\{\mathsf{x}_0,\mathsf{y}_0\}, \{\mathsf{y}_0,\mathsf{z}_0\}, \{\mathsf{z}_0,\mathsf{x}_0\}\} \setminus \{\mathsf{e}_0\})$
$\mathsf{Suppose} \Rightarrow \quad \mathsf{e}_0 = \{\mathsf{x}_0,\mathsf{y}_0\} \ \&$
$\{\{\mathsf{x}_0,\mathsf{y}_0\}, \{\mathsf{y}_0,\mathsf{z}_0\}, \{\mathsf{z}_0,\mathsf{x}_0\}\} \setminus \{\mathsf{e}_0\} = \{\{\mathsf{y}_0,\mathsf{z}_0\}, \{\mathsf{z}_0,\mathsf{x}_0\}\}$
$\langle \{\{\mathsf{y}_0,\mathsf{z}_0\}, \{\mathsf{z}_0,\mathsf{x}_0\}\}, \{\mathsf{y}_0,\mathsf{z}_0\}, \{\mathsf{z}_0,\mathsf{x}_0\} \rangle \hookrightarrow T2a \ \Rightarrow$
$\bigcup\{\{\mathsf{y}_0,\mathsf{z}_0\}, \{\mathsf{z}_0,\mathsf{x}_0\}\} = \{\mathsf{y}_0,\mathsf{z}_0,\mathsf{x}_0\}$
$\mathsf{EQUAL} \Rightarrow \quad \{\mathsf{x}_0,\mathsf{y}_0\} \nsubseteq \{\mathsf{y}_0,\mathsf{z}_0,\mathsf{x}_0\};$ $\qquad \mathsf{Discharge} \Rightarrow \quad$ AUTO
$\mathsf{Suppose} \Rightarrow \quad \mathsf{e}_0 = \{\mathsf{y}_0,\mathsf{z}_0\} \ \&$
$\{\{\mathsf{x}_0,\mathsf{y}_0\}, \{\mathsf{y}_0,\mathsf{z}_0\}, \{\mathsf{z}_0,\mathsf{x}_0\}\} \setminus \{\mathsf{e}_0\} = \{\{\mathsf{x}_0,\mathsf{y}_0\}, \{\mathsf{z}_0,\mathsf{x}_0\}\}$
$\langle \{\{\mathsf{x}_0,\mathsf{y}_0\}, \{\mathsf{z}_0,\mathsf{x}_0\}\}, \{\mathsf{x}_0,\mathsf{y}_0\}, \{\mathsf{z}_0,\mathsf{x}_0\} \rangle \hookrightarrow T2a \ \Rightarrow$
$\bigcup\{\{\mathsf{x}_0,\mathsf{y}_0\}, \{\mathsf{z}_0,\mathsf{x}_0\}\} = \{\mathsf{x}_0,\mathsf{y}_0,\mathsf{z}_0\}$
$\mathsf{EQUAL} \Rightarrow \quad \{\mathsf{y}_0,\mathsf{z}_0\} \nsubseteq \{\mathsf{x}_0,\mathsf{y}_0,\mathsf{z}_0\};$ $\qquad \mathsf{Discharge} \Rightarrow \quad$ AUTO
$\langle \{\{\mathsf{x}_0,\mathsf{y}_0\}, \{\mathsf{y}_0,\mathsf{z}_0\}\}, \{\mathsf{x}_0,\mathsf{y}_0\}, \{\mathsf{y}_0,\mathsf{z}_0\} \rangle \hookrightarrow T2a \ \Rightarrow$
$\bigcup\{\{\mathsf{x}_0,\mathsf{y}_0\}, \{\mathsf{y}_0,\mathsf{z}_0\}\} = \{\mathsf{x}_0,\mathsf{y}_0,\mathsf{z}_0\} \ \& \ \mathsf{e}_0 = \{\mathsf{z}_0,\mathsf{x}_0\} \ \&$
$\{\{\mathsf{x}_0,\mathsf{y}_0\}, \{\mathsf{y}_0,\mathsf{z}_0\}, \{\mathsf{z}_0,\mathsf{x}_0\}\} \setminus \{\mathsf{e}_0\} = \{\{\mathsf{x}_0,\mathsf{y}_0\}, \{\mathsf{y}_0,\mathsf{z}_0\}\}$
$\mathsf{EQUAL} \Rightarrow \quad \{\mathsf{z}_0,\mathsf{x}_0\} \nsubseteq \{\mathsf{x}_0,\mathsf{y}_0,\mathsf{z}_0\};$ $\qquad \mathsf{Discharge} \Rightarrow \quad$ QED

The following is the basic case of a general theorem scheme where the length of the path can be any number > 1: Replacing an edge by a path with the same endpoints does not disrupt a hank.

THEOREM hank_3: [Hank enrichment] $\big(\mathsf{Hank(H)} \ \& \ \{\mathsf{W},\mathsf{Y}\} \in \mathsf{H} \ \& \ \mathsf{W} \neq \mathsf{Y} \ \&$
$\mathsf{X} \notin \bigcup \mathsf{H} \ \& \ \mathsf{H}' = \mathsf{H} \setminus \{\{\mathsf{W},\mathsf{Y}\}\} \cup \{\{\mathsf{W},\mathsf{X}\}, \{\mathsf{X},\mathsf{Y}\}\}\big) \rightarrow \mathsf{Hank}(\mathsf{H}')$. PROOF:
$\mathsf{Suppose\_not}(\mathsf{h}_0, \mathsf{w}_0, \mathsf{y}_0, \mathsf{x}_1, \mathsf{h}_1) \Rightarrow \quad Stat0: \ \big(\mathsf{Hank}(\mathsf{h}_0) \ \& \ \{\mathsf{w}_0,\mathsf{y}_0\} \in \mathsf{h}_0 \ \&$

$$\mathsf{w}_0 \neq \mathsf{y}_0 \mathbin{\&} \mathsf{x}_1 \notin \bigcup \mathsf{h}_0 \mathbin{\&} \mathsf{h}_1 = \mathsf{h}_0 \setminus \{\{\mathsf{w}_0, \mathsf{y}_0\}\} \cup \{\{\mathsf{w}_0, \mathsf{x}_1\}, \{\mathsf{x}_1, \mathsf{y}_0\}\}\big) \mathbin{\&} \neg \mathsf{Hank}(\mathsf{h}_1)$$

‖ Suppose that the premisses are met by $\mathsf{h}_0, \mathsf{w}_0, \mathsf{y}_0, \mathsf{x}_1$, and h_1. In order to prove $\mathsf{Hank}(\mathsf{h}_1)$, we assume it to be false, so that the definition of hank implies the existence of an $\mathsf{e}_1 \in \mathsf{h}_1$ and a $\mathsf{z}_1 \in \mathsf{e}_1$ such that $\mathsf{z}_1 \notin \bigcup(\mathsf{h}_1 \setminus \{\mathsf{e}_1\})$.

$\mathsf{Use\_def(Hank)} \Rightarrow \quad \mathit{Stat1}: \neg \langle \forall \mathsf{e} \in \mathsf{h}_1 \mid \mathsf{e} \subseteq \bigcup(\mathsf{h}_1 \setminus \{\mathsf{e}\}) \rangle \mathbin{\&}$
$\quad \mathit{Stat2}: \langle \forall \mathsf{e} \in \mathsf{h}_0 \mid \mathsf{e} \subseteq \bigcup(\mathsf{h}_0 \setminus \{\mathsf{e}\}) \rangle$
$\langle \mathsf{e}_1, \mathsf{e}_1 \rangle \hookrightarrow \mathit{Stat1} \Rightarrow \quad \mathit{Stat3}: \mathsf{e}_1 \not\subseteq \bigcup(\mathsf{h}_1 \setminus \{\mathsf{e}_1\}) \mathbin{\&} \mathsf{e}_1 \in \mathsf{h}_1 \mathbin{\&}$
$\quad \big(\mathsf{e}_1 \in \mathsf{h}_0 \rightarrow \mathsf{e}_1 \subseteq \bigcup(\mathsf{h}_0 \setminus \{\mathsf{e}_1\})\big)$
$\mathsf{Use\_def}\big(\bigcup(\mathsf{h}_1 \setminus \{\mathsf{e}_1\})\big) \Rightarrow \quad \textsc{Auto}$
$\langle \mathsf{z}_1 \rangle \hookrightarrow \mathit{Stat3}(\mathit{Stat3}\star) \Rightarrow \quad \mathit{Stat4}: \mathsf{z}_1 \notin \{\mathsf{v} : \mathsf{u} \in \mathsf{h}_1 \setminus \{\mathsf{e}_1\}, \mathsf{v} \in \mathsf{u}\} \mathbin{\&} \mathsf{z}_1 \in \mathsf{e}_1$

‖ Since x_1 belongs to the distinct edges $\{\mathsf{w}_0, \mathsf{x}_1\}, \{\mathsf{x}_1, \mathsf{y}_0\}$ of h_1, clearly $\mathsf{z}_1 \neq \mathsf{x}_1$.

$\mathsf{Suppose} \Rightarrow \quad \mathsf{z}_1 = \mathsf{x}_1$
$\quad \langle \{\mathsf{w}_0, \mathsf{x}_1\}, \mathsf{x}_1 \rangle \hookrightarrow \mathit{Stat4} \Rightarrow \quad \{\mathsf{w}_0, \mathsf{x}_1\} = \mathsf{e}_1$
$\langle \{\mathsf{x}_1, \mathsf{y}_0\}, \mathsf{x}_1 \rangle \hookrightarrow \mathit{Stat4} \Rightarrow \quad \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow \quad \textsc{Auto}$

‖ Moreover, e_1 cannot be one of the edges of $\mathsf{h}_1 \setminus \mathsf{h}_0$.

$\mathsf{Suppose} \Rightarrow \quad \mathsf{e}_1 \in \{\{\mathsf{w}_0, \mathsf{x}_1\}, \{\mathsf{x}_1, \mathsf{y}_0\}\}$
$\quad \mathsf{Use\_def}\big(\bigcup(\mathsf{h}_0 \setminus \{\{\mathsf{w}_0, \mathsf{y}_0\}\})\big) \Rightarrow \quad \textsc{Auto}$
$\quad \langle \{\mathsf{w}_0, \mathsf{y}_0\} \rangle \hookrightarrow \mathit{Stat2} \Rightarrow \quad \mathit{Stat5}: \mathsf{z}_1 \in \{\mathsf{v} : \mathsf{u} \in \mathsf{h}_0 \setminus \{\{\mathsf{w}_0, \mathsf{y}_0\}\}, \mathsf{v} \in \mathsf{u}\}$
$\quad \langle \mathsf{e}', \mathsf{z}' \rangle \hookrightarrow \mathit{Stat5} \Rightarrow \quad \mathsf{e}' \in \mathsf{h}_0 \setminus \{\{\mathsf{w}_0, \mathsf{y}_0\}\} \mathbin{\&} \mathsf{z}_1 \in \mathsf{e}'$
$\quad \mathsf{Use\_def}(\bigcup \mathsf{h}_0) \Rightarrow \quad \textsc{Auto}$
$\quad \langle \mathsf{e}', \mathsf{z}_1 \rangle \hookrightarrow \mathit{Stat4} \Rightarrow \quad \mathsf{x}_1 \in \mathsf{e}' \mathbin{\&} \mathit{Stat6}: \mathsf{x}_1 \notin \{\mathsf{v} : \mathsf{u} \in \mathsf{h}_0, \mathsf{v} \in \mathsf{u}\}$
$\langle \mathsf{e}', \mathsf{x}_1 \rangle \hookrightarrow \mathit{Stat6} \Rightarrow \quad \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow \quad \textsc{Auto}$
$\mathsf{Use\_def}\big(\bigcup(\mathsf{h}_0 \setminus \{\mathsf{e}_1\})\big) \Rightarrow \quad \textsc{Auto}$

‖ We know, at this point, that $\mathsf{e}_1 \in \mathsf{h}_0 \setminus \{\{\mathsf{w}_0, \mathsf{y}_0\}\}$. Since h_0 is a hank, z_1 has in h_0 at least one incident edge different from e_1; since the latter is no longer available in $\mathsf{h}_1 \setminus \{\mathsf{e}_1\}$, it must be $\{\mathsf{w}_0, \mathsf{y}_0\}$, and either $\mathsf{z}_1 = \mathsf{w}_0$ or $\mathsf{z}_1 = \mathsf{y}_0$ hence holds. Both cases lead to a contradiction, though; in fact $\{\mathsf{w}_0, \mathsf{x}_1\}, \{\mathsf{x}_1, \mathsf{y}_0\}$ differ from e_1 and these edges of h_1 are, respectively, incident to w_0 and to y_0.

$(\mathit{Stat0}\star)\mathsf{ELEM} \Rightarrow \quad \mathsf{e}_1 \in \mathsf{h}_0 \mathbin{\&}$
$\mathit{Stat7}: \mathsf{z}_1 \in \{\mathsf{v} : \mathsf{u} \in \mathsf{h}_0 \setminus \{\mathsf{e}_1\}, \mathsf{v} \in \mathsf{u}\} \mathbin{\&} \mathsf{z}_1 \notin \{\mathsf{v} : \mathsf{u} \in \mathsf{h}_1 \setminus \{\mathsf{e}_1\}, \mathsf{v} \in \mathsf{u}\}$
$\quad \langle \mathsf{e}_0, \mathsf{z}_0, \mathsf{e}_0, \mathsf{z}_1 \rangle \hookrightarrow \mathit{Stat7} \Rightarrow \quad \mathit{Stat8}: \mathsf{z}_1 \in \{\mathsf{w}_0, \mathsf{y}_0\}$
$\quad \langle \{\mathsf{w}_0, \mathsf{x}_1\}, \mathsf{w}_0 \rangle \hookrightarrow \mathit{Stat4} \Rightarrow \quad \mathsf{z}_1 \neq \mathsf{w}_0$
$\quad \langle \{\mathsf{x}_1, \mathsf{y}_0\}, \mathsf{y}_0 \rangle \hookrightarrow \mathit{Stat4} \Rightarrow \quad \mathsf{z}_1 \neq \mathsf{y}_0$
$(\mathit{Stat8}\star)\mathsf{Discharge} \Rightarrow \quad \textsc{Qed}$

THEOREM cycle_0: [A membership 2-chain and an extra edge make a cycle]
$\mathsf{X} \in \mathsf{Y} \,\&\, \mathsf{Y} \in \mathsf{Z} \rightarrow \mathsf{Cycle}(\{\{\mathsf{X},\mathsf{Y}\},\{\mathsf{Y},\mathsf{Z}\},\{\mathsf{Z},\mathsf{X}\}\})$. PROOF:
$\mathsf{Suppose\_not}(\mathsf{x}_0,\mathsf{y}_0,\mathsf{z}_0) \Rightarrow$ AUTO
$\mathsf{Use\_def}\big(\mathsf{Cycle}(\{\{\mathsf{x}_0,\mathsf{y}_0\},\{\mathsf{y}_0,\mathsf{z}_0\},\{\mathsf{z}_0,\mathsf{x}_0\}\})\big) \Rightarrow$ AUTO
$\langle \mathsf{x}_0,\mathsf{y}_0,\mathsf{z}_0\rangle \hookrightarrow T\mathsf{hank}_2(\star) \Rightarrow$ $\mathit{Stat0}: \neg\langle \forall \mathsf{d} \subseteq \{\{\mathsf{x}_0,\mathsf{y}_0\},\{\mathsf{y}_0,\mathsf{z}_0\},\{\mathsf{z}_0,\mathsf{x}_0\}\} \mid$
$\mathsf{Hank}(\mathsf{d}) \,\&\, \mathsf{d} \neq \emptyset \rightarrow \mathsf{d} = \{\{\mathsf{x}_0,\mathsf{y}_0\},\{\mathsf{y}_0,\mathsf{z}_0\},\{\mathsf{z}_0,\mathsf{x}_0\}\}\rangle$
$\langle \mathsf{d}\rangle \hookrightarrow \mathit{Stat0} \Rightarrow$ $\mathit{Stat2}: \mathsf{Hank}(\mathsf{d}) \,\&\, \mathsf{d} \neq \emptyset \,\&$
$\mathsf{d} \neq \{\{\mathsf{x}_0,\mathsf{y}_0\},\{\mathsf{y}_0,\mathsf{z}_0\},\{\mathsf{z}_0,\mathsf{x}_0\}\} \,\&\, \mathsf{d} \subseteq \{\{\mathsf{x}_0,\mathsf{y}_0\},\{\mathsf{y}_0,\mathsf{z}_0\},\{\mathsf{z}_0,\mathsf{x}_0\}\}$
$\langle \mathsf{d},\{\mathsf{y}_0,\mathsf{z}_0\},\{\mathsf{z}_0,\mathsf{x}_0\}\rangle \hookrightarrow T\mathsf{hank}_1(\mathit{Stat2}\star) \Rightarrow$ $\mathsf{d} \neq \{\{\mathsf{y}_0,\mathsf{z}_0\},\{\mathsf{z}_0,\mathsf{x}_0\}\} \,\&$
$\mathsf{d} \neq \{\{\mathsf{y}_0,\mathsf{z}_0\}\} \,\&\, \mathsf{d} \neq \{\{\mathsf{z}_0,\mathsf{x}_0\}\}$
$\langle \mathsf{d},\{\mathsf{x}_0,\mathsf{y}_0\},\{\mathsf{z}_0,\mathsf{x}_0\}\rangle \hookrightarrow T\mathsf{hank}_1(\mathit{Stat2},\mathit{Stat2}\star) \Rightarrow$ $\mathsf{d} \neq \{\{\mathsf{x}_0,\mathsf{y}_0\},\{\mathsf{z}_0,\mathsf{x}_0\}\} \,\&$
$\mathsf{d} \neq \{\{\mathsf{x}_0,\mathsf{y}_0\}\}$
$\langle \mathsf{d},\{\mathsf{x}_0,\mathsf{y}_0\},\{\mathsf{y}_0,\mathsf{z}_0\}\rangle \hookrightarrow T\mathsf{hank}_1(\mathit{Stat2},\mathit{Stat2}\star) \Rightarrow$ $\mathsf{d} \neq \{\{\mathsf{x}_0,\mathsf{y}_0\},\{\mathsf{y}_0,\mathsf{z}_0\}\}$
$(\mathit{Stat2}\star)\mathsf{ELEM} \Rightarrow$ false; $\mathsf{Discharge} \Rightarrow$ QED

The replacement of an edge by a 2-path with the same endpoints does not disrupt a cycle.

THEOREM cycle_1: [Cycle enrichment] $\mathsf{Cycle}(\mathsf{C}) \,\&\, \{\mathsf{W},\mathsf{Y}\} \in \mathsf{C} \,\&\, \mathsf{W} \neq \mathsf{Y} \,\&$
$\mathsf{X} \notin \bigcup \mathsf{C} \,\&\, \mathsf{C}' = \mathsf{C} \setminus \{\{\mathsf{W},\mathsf{Y}\}\} \cup \{\{\mathsf{W},\mathsf{X}\},\{\mathsf{X},\mathsf{Y}\}\} \rightarrow \mathsf{Cycle}(\mathsf{C}')$. PROOF:
$\mathsf{Suppose\_not}(\mathsf{h}_0,\mathsf{w}_0,\mathsf{y}_0,\mathsf{x}_1,\mathsf{h}_1) \Rightarrow$ AUTO

Supposing that $\mathsf{h}_0,\mathsf{w}_0,\mathsf{y}_0,\mathsf{x}_1,\mathsf{h}_1$ constitute a counter-example to the claim, observe that $\mathsf{Hank}(\mathsf{h}_1)$ must hold; hence we can consider a strictly smaller hank d_1 than h_1. It readily turns out that either $\{\mathsf{w}_0,\mathsf{x}_1\} \in \mathsf{d}_1$ or $\{\mathsf{x}_1,\mathsf{y}_0\} \in \mathsf{d}_1$; for otherwise h_0 would strictly include d_1, since $\mathsf{d}_1 \neq \mathsf{h}_0$ follows from $\{\mathsf{w}_0,\mathsf{y}_0\} \in \mathsf{h}_0 \setminus \mathsf{h}_1$.

$\langle \mathsf{h}_0,\mathsf{w}_0,\mathsf{y}_0,\mathsf{x}_1,\mathsf{h}_1\rangle \hookrightarrow T\mathsf{hank}_3 \Rightarrow$ AUTO
$\mathsf{Use\_def}(\mathsf{Cycle}) \Rightarrow$ $\mathit{Stat0}: \{\mathsf{w}_0,\mathsf{y}_0\} \in \mathsf{h}_0 \,\&\, \mathsf{w}_0 \neq \mathsf{y}_0 \,\&$
$\mathsf{x}_1 \notin \bigcup \mathsf{h}_0 \,\&\, \mathsf{h}_1 = \mathsf{h}_0 \setminus \{\{\mathsf{w}_0,\mathsf{y}_0\}\} \cup \{\{\mathsf{w}_0,\mathsf{x}_1\},\{\mathsf{x}_1,\mathsf{y}_0\}\} \,\&$
$\mathit{Stat1}: \neg\langle \forall \mathsf{d} \subseteq \mathsf{h}_1 \mid \mathsf{Hank}(\mathsf{d}) \,\&\, \mathsf{d} \neq \emptyset \rightarrow \mathsf{d} = \mathsf{h}_1\rangle \,\&$
$\mathit{Stat2}: \langle \forall \mathsf{d} \subseteq \mathsf{h}_0 \mid \mathsf{Hank}(\mathsf{d}) \,\&\, \mathsf{d} \neq \emptyset \rightarrow \mathsf{d} = \mathsf{h}_0\rangle \,\&\, \mathsf{Hank}(\mathsf{h}_0) \,\&\, \mathsf{Hank}(\mathsf{h}_1)$
$\langle \mathsf{d}_1,\mathsf{d}_1\rangle \hookrightarrow \mathit{Stat1}(\mathit{Stat0}\star) \Rightarrow$ $\mathit{Stat3}: \mathsf{d}_1 \subseteq \mathsf{h}_1 \,\&\, \mathsf{d}_1 \neq \mathsf{h}_1 \,\&\, \mathsf{d}_1 \neq \emptyset \,\&\, \mathsf{Hank}(\mathsf{d}_1) \,\&$
$\neg(\{\mathsf{w}_0,\mathsf{x}_1\} \notin \mathsf{d}_1 \,\&\, \{\mathsf{x}_1,\mathsf{y}_0\} \notin \mathsf{d}_1)$
$\mathsf{Use\_def}(\bigcup \mathsf{h}_0) \Rightarrow$ AUTO
$\langle \mathsf{d}_1\rangle \hookrightarrow T\mathsf{hank}_0 \Rightarrow$ $\mathit{Stat4}: \langle \forall \mathsf{e} \in \mathsf{d}_1, \mathsf{x} \in \mathsf{e}, \exists \mathsf{q} \in \mathsf{d}_1 \mid \mathsf{q} \neq \mathsf{e} \,\&\, \mathsf{x} \in \mathsf{q}\rangle \,\&$
$\mathit{Stat4a}: \mathsf{x}_1 \notin \{\mathsf{v} : \mathsf{u} \in \mathsf{h}_0, \mathsf{v} \in \mathsf{u}\} \,\&\, \emptyset \notin \mathsf{d}_1$

Should one of $\{\mathsf{w}_0,\mathsf{x}_1\}$, $\{\mathsf{x}_1,\mathsf{y}_0\}$, but not the other, belong to d_1, we would easily get a contradiction: the two cases are treated symmetrically. At this point we have derived that both $\{\mathsf{w}_0,\mathsf{x}_1\}$ and $\{\mathsf{x}_1,\mathsf{y}_0\}$ belong to d_1.

$\langle \{\mathsf{w}_0,\mathsf{x}_1\},\mathsf{x}_1,\mathsf{q}_0,\mathsf{q}_0,\mathsf{x}_1\rangle \hookrightarrow \mathit{Stat4}(\mathit{Stat0}\star) \Rightarrow$ $\neg(\{\mathsf{w}_0,\mathsf{x}_1\} \in \mathsf{d}_1 \,\&\, \{\mathsf{x}_1,\mathsf{y}_0\} \notin \mathsf{d}_1)$
$\langle \{\mathsf{x}_1,\mathsf{y}_0\},\mathsf{x}_1,\mathsf{q}_1,\mathsf{q}_1,\mathsf{x}_1\rangle \hookrightarrow \mathit{Stat4}(\mathit{Stat0}\star) \Rightarrow$ $\mathit{Stat5}: \{\mathsf{w}_0,\mathsf{x}_1\},\{\mathsf{x}_1,\mathsf{y}_0\} \in \mathsf{d}_1$

|| We will show that the set d_0 obtained by replacing $\{\mathsf{w}_0, \mathsf{x}_1\}$ and $\{\mathsf{x}_1, \mathsf{y}_0\}$ by $\{\mathsf{w}_0, \mathsf{y}_0\}$ in d_1 is non-null and is a cycle strictly included in h_0. Obviously $\{\mathsf{w}_0, \mathsf{x}_1\} \neq \{\mathsf{w}_0, \mathsf{y}_0\}$ & $\{\mathsf{x}_1, \mathsf{y}_0\} \neq \{\mathsf{w}_0, \mathsf{y}_0\}$, because x_1, y_0, and w_0 are distinct.

$\langle \{\mathsf{w}_0, \mathsf{y}_0\}, \mathsf{x}_1 \rangle \hookrightarrow \mathit{Stat4a}(\mathit{Stat0}\star) \Rightarrow \quad \mathit{Stat6}: \ \mathsf{x}_1 \neq \mathsf{w}_0 \ \& \ \mathsf{x}_1 \neq \mathsf{y}_0 \ \& \ \mathsf{w}_0 \neq \mathsf{y}_0 \ \&$
$\quad \{\mathsf{w}_0, \mathsf{y}_0\} \notin \mathsf{d}_1$
$\mathsf{Loc\_def} \Rightarrow \quad \mathit{Stat7}: \ \mathsf{d}_0 = \mathsf{d}_1 \cup \{\{\mathsf{w}_0, \mathsf{y}_0\}\} \setminus \{\{\mathsf{w}_0, \mathsf{x}_1\}, \{\mathsf{x}_1, \mathsf{y}_0\}\}$
$(\mathit{Stat5}, \mathit{Stat7}, \mathit{Stat6}, \mathit{Stat3}, \mathit{Stat0}, \mathit{Stat4}\star)\mathsf{ELEM} \Rightarrow \quad \mathsf{d}_0 \subseteq \mathsf{h}_0 \ \& \ \mathsf{d}_0 \neq \emptyset \ \&$
$\quad \mathsf{d}_0 \neq \mathsf{h}_0 \ \& \ \emptyset \notin \mathsf{d}_0$
$\mathsf{Use\_def}(\bigcup \mathsf{d}_0) \Rightarrow \quad \text{AUTO}$
$\langle \mathsf{d}_0, \mathsf{h}_0 \rangle \hookrightarrow T2c \ (\mathit{Stat0}\star) \Rightarrow \quad \mathit{Stat8}: \ \mathsf{x}_1 \notin \{\mathsf{u} : \mathsf{v} \in \mathsf{d}_0, \mathsf{u} \in \mathsf{v}\}$

|| Despite us having assumed at the beginning that h_0 contains no proper cycle, so that in particular $\mathsf{Hank}(\mathsf{d}_0)$ cannot hold, due to an edge e_0 of d_0 and to an endpoint x_0 of e_0 which is not properly covered in d_0, ...

$\langle \mathsf{d}_0 \rangle \hookrightarrow T\mathsf{hank}_0 \ \Rightarrow \quad \text{AUTO}$
$\langle \mathsf{d}_0 \rangle \hookrightarrow \mathit{Stat2}(\mathit{Stat7}\star) \Rightarrow \quad \mathit{Stat9}: \ \neg\langle \forall \mathsf{e} \in \mathsf{d}_0, \mathsf{x} \in \mathsf{e}, \exists \mathsf{q} \in \mathsf{d}_0 \mid \mathsf{q} \neq \mathsf{e} \ \& \ \mathsf{x} \in \mathsf{q} \rangle$
$\langle \mathsf{e}_0, \mathsf{x}_0 \rangle \hookrightarrow \mathit{Stat9} \Rightarrow$
$\qquad \mathit{Stat10}: \ \neg\langle \exists \mathsf{q} \in \mathsf{d}_0 \mid \mathsf{q} \neq \mathsf{e}_0 \ \& \ \mathsf{x}_0 \in \mathsf{q} \rangle \ \& \ \mathsf{e}_0 \in \mathsf{d}_0 \ \& \ \mathsf{x}_0 \in \mathsf{e}_0$

|| ... we now aim at showing that this offending edge e_0 of d_0 will also offend d_1, which contradicts a fact noted at the beginning. Here we shortly digress to prove that $\mathsf{e}_0 = \{\mathsf{w}_0, \mathsf{y}_0\}$ must hold, else e_0 would offend d_1.

$\mathsf{Suppose} \Rightarrow \quad \mathit{Stat11}: \ \mathsf{e}_0 \neq \{\mathsf{w}_0, \mathsf{y}_0\}$

|| Indeed, assuming $\mathsf{e}_0 \neq \{\mathsf{w}_0, \mathsf{y}_0\}$, e_0 would also belong to d_1, and each one of its endpoints must have edges distinct from e_0 incident to it in d_1. However, it will turn out that this cannot be the case for the endpoint x_0, which hence is not properly covered in d_1.

$\langle \mathsf{e}_0, \mathsf{x}_0 \rangle \hookrightarrow \mathit{Stat4}(\mathit{Stat7}, \mathit{Stat10}, \mathit{Stat11}\star) \Rightarrow \quad \mathit{Stat12}: \ \langle \exists \mathsf{q} \in \mathsf{d}_1 \mid \mathsf{q} \neq \mathsf{e}_0 \ \& \ \mathsf{x}_0 \in \mathsf{q} \rangle$
$\langle \mathsf{q}_2 \rangle \hookrightarrow \mathit{Stat12}(\mathit{Stat12}\star) \Rightarrow \quad \mathit{Stat13}: \ \mathsf{x}_0 \in \mathsf{q}_2 \ \& \ \mathsf{q}_2 \neq \mathsf{e}_0 \ \& \ \mathsf{q}_2 \in \mathsf{d}_1$

|| To see this, let $\mathsf{q}_2 \neq \mathsf{e}_0$ be the edge that covers x_0 in d_1.

$\mathsf{Suppose} \Rightarrow \quad \mathit{Stat14}: \ \mathsf{q}_2 = \{\mathsf{w}_0, \mathsf{x}_1\} \vee \mathsf{q}_2 = \{\mathsf{x}_1, \mathsf{y}_0\}$

|| If this edge q_2 were one of the two which have been removed, the edge $\{\mathsf{w}_0, \mathsf{y}_0\}$ would satisfactorily cover x_0 in d_0.

$\langle \{\mathsf{w}_0, \mathsf{y}_0\} \rangle \hookrightarrow \mathit{Stat10}(\mathit{Stat7}, \mathit{Stat6}, \mathit{Stat11}\star) \Rightarrow \quad \mathit{Stat15}: \ \mathsf{x}_0 \notin \{\mathsf{w}_0, \mathsf{y}_0\}$
$\langle \mathsf{e}_0, \mathsf{x}_0 \rangle \hookrightarrow \mathit{Stat8} \Rightarrow \quad \text{AUTO}$
$(\mathit{Stat10}\star)\mathsf{Discharge} \Rightarrow \quad \mathit{Stat16}: \ \neg(\mathsf{q}_2 = \{\mathsf{w}_0, \mathsf{x}_1\} \vee \mathsf{q}_2 = \{\mathsf{x}_1, \mathsf{y}_0\})$

q_2 must hence belong to d_0; but again, this implies that q_2 would satisfactorily cover x_0 in d_0. Therefore, $e_0 = \{w_0, y_0\}$ must hold.

$$\langle q_2 \rangle \hookrightarrow \mathit{Stat10}(\mathit{Stat7}, \mathit{Stat16}, \mathit{Stat13}\star) \Rightarrow \quad \mathsf{false};$$
$$\mathsf{Discharge} \Rightarrow \quad \mathit{Stat17}: \ e_0 = \{w_0, y_0\}$$

The only remaining possibility, $e_0 = \{w_0, y_0\}$, is also untenable. Indeed, d_1 has two edges incident to w_0, one of which is $\{w_0, x_1\}$; likewise, d_1 has two edges incident to y_0, one of which is $\{x_1, y_0\}$. For either one of the endpoints w_0, y_0 of e_0, the second incident edge belongs to d_1 and differs from $\{w_0, y_0\}$, so it must belong to d_0 as well; since d_0 also owns the edge $\{w_0, y_0\}$ incident to either endpoint, it is not true that e_0 is an offending edge for d_0, a fact that contradicts one of the assumptions made.

$$\langle \{w_0, x_1\}, w_0, q_4 \rangle \hookrightarrow \mathit{Stat4}(\mathit{Stat5}, \mathit{Stat7}, \mathit{Stat6}, \mathit{Stat17}\star) \Rightarrow$$
$$\mathit{Stat19}: \ q_4 \neq e_0 \ \& \ q_4 \in d_0 \ \& \ w_0 \in q_4$$
$$\langle \{x_1, y_0\}, y_0, q_5 \rangle \hookrightarrow \mathit{Stat4}(\mathit{Stat5}, \mathit{Stat5}\star) \Rightarrow$$
$$\mathit{Stat20}: \ y_0 \in q_5 \ \& \ q_5 \neq \{x_1, y_0\} \ \& \ q_5 \in d_1$$
$$(\mathit{Stat20}, \mathit{Stat7}, \mathit{Stat6}, \mathit{Stat17}\star)\mathsf{ELEM} \Rightarrow \quad \mathit{Stat21}: \ q_5 \neq e_0 \ \& \ q_5, e_0 \in d_0$$
$$\langle q_5 \rangle \hookrightarrow \mathit{Stat10}(\mathit{Stat21}, \mathit{Stat17}, \mathit{Stat20}\star) \Rightarrow \quad \mathit{Stat22}: \ x_0 \notin q_5 \ \& \ x_0 = w_0$$
$$\langle q_4 \rangle \hookrightarrow \mathit{Stat10}(\mathit{Stat19}, \mathit{Stat22}\star) \Rightarrow \quad \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow \quad \mathrm{QED}$$

DEF hamiltonian$_1$: [Hamiltonian cycle, in graph devoid of isolated vertices]

$$\mathsf{Hamiltonian}(H, S, E) \ \leftrightarrow_{\mathrm{Def}} \mathsf{Cycle}(H) \ \& \ \bigcup H = S \ \& \ H \subseteq E$$

In our specialized context, where edges are 2-sets whose elements satisfy a peculiar membership constraint, we do not simply require that a Hamiltonian cycle H touches every vertex, but also that every source has an incident membership edge in H.

DEF hamiltonian$_2$: [Edges in squared membership]

$$\mathsf{sqEdges}(S) =_{\mathrm{Def}} \{\{x, y\} : \ x \in S, y \in S \setminus \{x\}, z \in S \cap x \mid y = z \vee y \in z \vee z \in y\}$$

DEF hamiltonian$_3$: [Restraining condition for Hamiltonian cycles]

$$\mathsf{SqHamiltonian}(H, S) \ \leftrightarrow_{\mathrm{Def}} \mathsf{Hamiltonian}\big(H, S, \mathsf{sqEdges}(S)\big) \ \&$$
$$\langle \forall x \in S \setminus \bigcup S, \exists y \in x \mid \{x, y\} \in H \rangle$$

THEOREM hamiltonian$_1$: [Enriched Hamiltonian cycles]

$$S = T \cup \{X\} \ \& \ X \notin T \ \& \ Y \in X \ \& \ \mathsf{SqHamiltonian}(H, T) \ \& \ \{W, Y\} \in H \ \&$$
$$\big(W \in Y \vee (Y \in W \ \& \ K \neq Y \ \& \ \{W, K\} \in H \ \& \ K \in W)\big) \rightarrow$$
$$\mathsf{SqHamiltonian}(H \setminus \{\{W, Y\}\} \cup \{\{W, X\}, \{X, Y\}\}, S). \ \mathrm{PROOF}:$$

$$\mathsf{Suppose\_not}(s_0, t_0, x_1, y_0, h_0, w_0, k_0) \Rightarrow \quad \mathrm{AUTO}$$
$$\mathsf{Use\_def}(\mathsf{SqHamiltonian}) \Rightarrow \quad \mathit{Stat0}: \ \langle \forall x \in t_0 \setminus \bigcup t_0, \exists y \in x \mid \{x, y\} \in h_0 \rangle \ \&$$
$$\mathsf{Hamiltonian}\big(h_0, t_0, \mathsf{sqEdges}(t_0)\big) \ \&$$
$$\neg\Big(\mathsf{Hamiltonian}\big(h_0 \setminus \{\{w_0, y_0\}\} \cup \{\{w_0, x_1\}, \{x_1, y_0\}\}, s_0, \mathsf{sqEdges}(s_0)\big) \ \&$$

$\langle \forall x \in s_0 \backslash \bigcup s_0, \exists y \in x \mid \{x,y\} \in h_0 \backslash \{\{w_0,y_0\}\} \cup \{\{w_0,x_1\},\{x_1,y_0\}\}\rangle)$

Suppose $\Rightarrow$ *Stat1* :
$\neg\langle \forall x \in s_0 \backslash \bigcup s_0, \exists y \in x \mid \{x,y\} \in h_0 \backslash \{\{w_0,y_0\}\} \cup \{\{w_0,x_1\},\{x_1,y_0\}\}\rangle$

Suppose that $s_0, t_0, x_1, y_0, h_0, w_0, k_0$ make a counterexample to the claim. One reason why

$$\mathsf{SqHamiltonian}(h_0 \backslash \{\{w_0,y_0\}\} \cup \{\{w_0,x_1\},\{x_1,y_0\}\}, s_0)$$

is violated might be

$\neg\langle \forall x \in s_0 \backslash \bigcup s_0, \exists y \in x \mid \{x,y\} \in h_0 \backslash \{\{w_0,y_0\}\} \cup \{\{w_0,x_1\},\{x_1,y_0\}\}\rangle$

; if this is the case, we can choose an x' witnessing this fact.

$\langle x'\rangle \hookrightarrow$ *Stat1* $\Rightarrow$ *Stat2* :
$\neg\langle \exists k \in x' \mid \{x',k\} \in h_0 \backslash \{\{w_0,y_0\}\} \cup \{\{w_0,x_1\},\{x_1,y_0\}\}\rangle \;\&\; x' \in s_0 \backslash \bigcup s_0$

To see that $x' \in t_0 \backslash \bigcup t_0$ follows from the constraint $x' \in s_0 \backslash \bigcup s_0$, we assume the contrary and argue as follows: (1) Unless x' belongs to t_0, we must have $x' = x_1$, which however has an incident membership edge, namely $\{x_1, y_0\}$, in $h_0 \backslash \{\{w_0,y_0\}\} \cup \{\{w_0,x_1\},\{x_1,y_0\}\}$. (2) Thus, since $x' \in t_0$, we have $x' \in t_0 \cap \bigcup t_0$ and hence $x' \in s_0 \cap \bigcup s_0$, contradicting the constraint on x'.

Suppose $\Rightarrow$ $x' \notin t_0 \backslash \bigcup t_0$
$\langle y_0\rangle \hookrightarrow$ *Stat2*$(\star) \Rightarrow$ $x' \in t_0 \;\&\; s_0 = t_0 \cup \{x_1\} \;\&\; x_1 \notin t_0 \;\&\; y_0 \in x_1$
$\langle t_0, s_0\rangle \hookrightarrow$ *T2c* (*Stat2*$\star$) $\Rightarrow$ $\bigcup s_0 \supseteq \bigcup t_0$
(*Stat2*$\star$)Discharge $\Rightarrow$ AUTO

Knowing that $x' \in t_0 \backslash \bigcup t_0$, we can find a $y_1 \in x'$ such that $\{x', y_1\} \in h_0$. Since this membership edge is no longer available after the modification of h_0, it must be $\{w_0, y_0\}$; therefore, $x' = w_0$, for otherwise $x' = y_0$ would (in view of the fact $y_0 \in x_1$) contradict the assumption $x' \in s_0 \backslash \bigcup s_0$.

$\langle x', y_1\rangle \hookrightarrow$ *Stat0*$(\star) \Rightarrow$ *Stat3* : $y_1 \in x' \;\&\; \{x', y_1\} \in h_0 \;\&\; x_1 \in s_0 \;\&\; y_0 \in x_1$
Use_def$(\bigcup s_0) \Rightarrow$ AUTO
$\langle y_1\rangle \hookrightarrow$ *Stat2*(*Stat3*$\star$) $\Rightarrow$ *Stat4* : $x' \notin \{u : v \in s_0, u \in v\} \;\&\; \{x', y_1\} = \{w_0, y_0\}$
$\langle x_1, y_0\rangle \hookrightarrow$ *Stat4*(*Stat2*$\star$) $\Rightarrow$ $x' = w_0$

If $x' \in y_0$, the assumption $x' \in s_0 \backslash \bigcup s_0$ would be contradicted similarly: but then, by the initial assumption, we must have $\{x', k_0\} \in h_0 \;\&\; k_0 \neq y_0 \;\&\; k_0 \in x'$, conflicting with Stat2, because $\{x', k_0\} = \{w_0, y_0\}$ would imply $k_0 = y_0$.

$\langle x_1, y_0\rangle \hookrightarrow$ *Stat4*$(\star) \Rightarrow$ $x' \notin y_0 \;\&\; \{x', k_0\} \in h_0 \;\&\; k_0 \in x' \;\&\; k_0 \neq y_0$
$\langle k_0\rangle \hookrightarrow$ *Stat2*(*Stat4*$\star$) $\Rightarrow$ false; Discharge $\Rightarrow$
Stat5 : $\mathsf{Hamiltonian}(h_0, t_0, \mathsf{sqEdges}(t_0)) \;\&$
$\neg\mathsf{Hamiltonian}(h_0 \backslash \{\{w_0,y_0\}\} \cup \{\{w_0,x_1\},\{x_1,y_0\}\}, s_0, \mathsf{sqEdges}(s_0))$

At this point the reason why

$$\mathsf{SqHamiltonian}(h_0 \setminus \{\{w_0, y_0\}\} \cup \{\{w_0, x_1\}, \{x_1, y_0\}\}, s_0)$$

is false must be that

$$\mathsf{Hamiltonian}\big(h_0 \setminus \{\{w_0, y_0\}\} \cup \{\{w_0, x_1\}, \{x_1, y_0\}\}, s_0, \mathsf{sqEdges}(s_0)\big)$$

is false; however, we will derive a contradiction also in this case.

$\mathsf{Use\_def}\Big(\mathsf{Hamiltonian}\big(h_0, t_0, \mathsf{sqEdges}(t_0)\big)\Big) \Rightarrow$ AUTO
$\mathsf{ELEM} \Rightarrow$ *Stat6* : $s_0 = t_0 \cup \{x_1\}$ & $x_1 \notin t_0$ & $y_0 \in x_1$ & $\{w_0, y_0\} \in h_0$ &
$\quad\big(w_0 \in y_0 \vee (y_0 \in w_0$ & $k_0 \neq y_0$ & $\{w_0, k_0\} \in h_0$ & $k_0 \in w_0)\big)$
$\mathsf{Loc\_def} \Rightarrow$ *Stat7* : $h_1 = h_0 \setminus \{\{w_0, y_0\}\} \cup \{\{w_0, x_1\}, \{x_1, y_0\}\}$
$\mathsf{Use\_def}\Big(\mathsf{Hamiltonian}\big(h_1, s_0, \mathsf{sqEdges}(s_0)\big)\Big) \Rightarrow$ AUTO
$\mathsf{EQUAL}(\mathit{Stat5}) \Rightarrow$ *Stat8* : $\big(\mathsf{Cycle}(h_0)$ & $\bigcup h_0 = t_0$ & $h_0 \subseteq \mathsf{sqEdges}(t_0)\big)$ &
$\quad\neg\big(\mathsf{Cycle}(h_1)$ & $\bigcup h_1 = s_0$ & $h_1 \subseteq \mathsf{sqEdges}(s_0)\big)$

In fact, after observing that $\{w_0, y_0\} \subseteq \bigcup h_0$ must hold, we will be able to discard one by one each potential reason why $\mathsf{Hamiltonian}\big(h_0 \setminus \{\{w_0, y_0\}\} \cup \{\{w_0, x_1\}, \{x_1, y_0\}\}, s_0, \mathsf{sqEdges}(s_0)\big)$ should be false.

$\langle h_0, w_0, y_0, x_1, h_1\rangle \hookrightarrow T\mathsf{cycle}_1\ (\mathit{Stat6}\star) \Rightarrow$ *Stat9* : $\big(\mathsf{Cycle}(h_0)$ & $\bigcup h_0 = t_0$ &
$\quad h_0 \subseteq \mathsf{sqEdges}(t_0)\big)$ & $\neg\big(\bigcup h_1 = s_0$ & $h_1 \subseteq \mathsf{sqEdges}(s_0)\big)$
$\mathsf{Suppose} \Rightarrow$ *Stat10* : $\{w_0, y_0\} \not\subseteq \bigcup h_0$
$\mathsf{Use\_def}(\bigcup h_0) \Rightarrow$ AUTO
$\langle b\rangle \hookrightarrow \mathit{Stat10}(\mathit{Stat10}\star) \Rightarrow$ *Stat11* : $b \notin \{u : v \in h_0, u \in v\}$ & $b \in \{w_0, y_0\}$
$\langle \{w_0, y_0\}, b\rangle \hookrightarrow \mathit{Stat11}(\mathit{Stat11}, \mathit{Stat6}\star) \Rightarrow$ false; $\quad \mathsf{Discharge} \Rightarrow$
$\quad$ *Stat12* : $w_0, y_0 \in t_0$
$\mathsf{Suppose} \Rightarrow$ *Stat13* : $h_1 \not\subseteq \mathsf{sqEdges}(s_0)$
$\mathsf{Use\_def}\big(\mathsf{sqEdges}(s_0)\big) \Rightarrow$ AUTO
$\langle e\rangle \hookrightarrow \mathit{Stat13}(\mathit{Stat7}\star) \Rightarrow$ $(e \in h_0 \vee e = \{w_0, x_1\} \vee e = \{x_1, y_0\})$ & *Stat14* :
$\quad e \notin \{\{x, y\} : x \in s_0, y \in s_0 \setminus \{x\}, z \in s_0 \cap x \mid y = z \vee y \in z \vee z \in y\}$
$(\mathit{Stat6}, \mathit{Stat12}\star)\mathsf{ELEM} \Rightarrow$ *Stat15* :
$\quad x_1, y_0, w_0 \in s_0$ & $y_0 \in x_1$ & $(w_0 \in y_0 \vee y_0 \in w_0)$ & $x_1 \neq w_0$
$\langle x_1, y_0, y_0\rangle \hookrightarrow \mathit{Stat14}(\mathit{Stat15}\star) \Rightarrow$ $e \neq \{x_1, y_0\}$
$\langle x_1, w_0, y_0\rangle \hookrightarrow \mathit{Stat14}(\mathit{Stat15}\star) \Rightarrow$ $e \neq \{w_0, x_1\}$
$\mathsf{Use\_def}\big(\mathsf{sqEdges}(t_0)\big) \Rightarrow$ AUTO
$(\mathit{Stat8}\star)\mathsf{ELEM} \Rightarrow$ *Stat16* :
$\quad e \in \{\{x, y\} : x \in t_0, y \in t_0 \setminus \{x\}, z \in t_0 \cap x \mid y = z \vee y \in z \vee z \in y\}$
$\langle x_2, y_2, z_2\rangle \hookrightarrow \mathit{Stat16}(\mathit{Stat16}\star) \Rightarrow$ *Stat17* : $e = \{x_2, y_2\}$ & $x_2, y_2, z_2 \in t_0$ &
$\quad x_2 \neq y_2$ & $z_2 \in x_2$ & $(y_2 = z_2 \vee y_2 \in z_2 \vee z_2 \in y_2)$
$(\mathit{Stat6}\star)\mathsf{ELEM} \Rightarrow$ $s_0 \supseteq t_0$
$\langle x_2, y_2, z_2\rangle \hookrightarrow \mathit{Stat14}(\mathit{Stat17}\star) \Rightarrow$ false; $\quad \mathsf{Discharge} \Rightarrow$ *Stat18* : $\bigcup h_1 \neq s_0$

We prove first that $\bigcup h_1 \subseteq s_0$.

$\langle\,\{\{w_0, x_1\}, \{x_1, y_0\}\}, \{w_0, x_1\}, \{x_1, y_0\}\,\rangle \hookrightarrow T2a(Stat18\star) \Rightarrow$
$\quad \bigcup\{\{w_0, x_1\}, \{x_1, y_0\}\} = \{w_0, x_1, y_0\}$
$\langle h_0 \setminus \{\{w_0, y_0\}\}, \{\{w_0, x_1\}, \{x_1, y_0\}\}\,\rangle \hookrightarrow T2c(Stat18\star) \Rightarrow$
$\quad \bigcup(h_0 \setminus \{\{w_0, y_0\}\} \cup \{\{w_0, x_1\}, \{x_1, y_0\}\})$
$\quad = \bigcup(h_0 \setminus \{\{w_0, y_0\}\}) \cup \bigcup\{\{w_0, x_1\}, \{x_1, y_0\}\}$
$\langle h_0 \setminus \{\{w_0, y_0\}\}, h_0\rangle \hookrightarrow T2c(Stat8\star) \Rightarrow$
$\quad \bigcup h_0 \supseteq \bigcup(h_0 \setminus \{\{w_0, y_0\}\})\ \&\ \{w_0, y_0\} \subseteq \bigcup h_0$
EQUAL$(Stat7) \Rightarrow \quad \bigcup h_1 = \bigcup(h_0 \setminus \{\{w_0, y_0\}\}) \cup \{w_0, x_1, y_0\}$
$(Stat6\star)$ELEM $\Rightarrow \quad \bigcup h_1 \subseteq s_0$

The remaining case is $s_0 \not\subseteq \bigcup h_1$, which entails that we can find an $a \in t_0 \setminus \bigcup h_1$.

Use_def$(\bigcup h_1) \Rightarrow$ AUTO
$\langle a\rangle \hookrightarrow Stat18(Stat18\star) \Rightarrow \quad Stat23:\ a \notin \{u : v \in h_1, u \in v\}\ \&\ a \in s_0 \setminus \{x_1\}$

Since $a \in t_0$ and $t_0 = \bigcup h_0$, we can find an $e' \in h_0$ such that $a \in e'$.

Use_def$(\bigcup) \Rightarrow \quad Stat24:\ a \in \{u : v \in h_0, u \in v\}$
$\langle e', u'\rangle \hookrightarrow Stat24(Stat25\star) \Rightarrow \quad Stat25:\ e' \in h_0\ \&\ a \in e'$

Since $h_1 = h_0 \setminus \{\{w_0, y_0\}\} \cup \{\{w_0, x_1\}, \{x_1, y_0\}\}$, we conclude that $e' = \{w_0, y_0\}$ must hold. Hence, either $a = w_0$ or $a = y_0$ must hold, both of which yield a contradiction.

$\langle e', a\rangle \hookrightarrow Stat23(Stat7, Stat25\star) \Rightarrow \quad e' = \{w_0, y_0\}$
$\langle\,\{w_0, x_1\}, w_0\rangle \hookrightarrow Stat23(Stat7\star) \Rightarrow \quad a = y_0$
$\langle\,\{x_1, y_0\}, y_0\rangle \hookrightarrow Stat23(Stat7\star) \Rightarrow \quad$ false; Discharge $\Rightarrow$ QED

THEOREM hamiltonian$_2$: [Doubly enriched Hamiltonian cycles]
$S = T \cup \{X, Z\}\ \&\ \{X, Z\} \cap T = \emptyset\ \&\ X \neq Z\ \&\ Y \in X \cap Z\ \&$
$\quad$ SqHamiltonian$(H, T)\ \&\ \{W, Y\} \in H\ \&\ W \in Y \cap X \rightarrow$
$\quad\quad$ SqHamiltonian$(H \setminus \{\{W, Y\}\} \cup \{\{W, X\}, \{X, Z\}, \{Z, Y\}\}, S)$. PROOF:
Suppose_not$(s_0, t_0, x_0, z_0, y_0, h_0, w_0) \Rightarrow$ AUTO
$\langle t_0 \cup \{x_0\}, t_0, x_0, y_0, h_0, w_0, \emptyset\rangle \hookrightarrow T$hamiltonian$_1 \Rightarrow$
$\quad$ SqHamiltonian$(h_0 \setminus \{\{w_0, y_0\}\} \cup \{\{w_0, x_0\}, \{x_0, y_0\}\}, t_0 \cup \{x_0\})$
Loc_def $\Rightarrow \quad Stat1:$
$\quad h_1 = h_0 \setminus \{\{w_0, y_0\}\} \cup \{\{w_0, x_0\}, \{x_0, y_0\}\}\ \&\ t_1 = t_0 \cup \{x_0\}$
ELEM $\Rightarrow \quad Stat2:\ s_0 = t_1 \cup \{z_0\}\ \&\ x_0 \notin t_0$
EQUAL $\Rightarrow$ SqHamiltonian(h_1, t_1)
Suppose $\Rightarrow \quad \{x_0, y_0\} \in h_0$

Notice that since h_0 is a Hamiltonian path in t_0, its unionset must equal t_0; since x_0 does not belong to t_0, but it belongs to $\{x_0, y_0\}$, it follows that $\{x_0, y_0\}$ cannot belong to h_0.

$\mathsf{Use\_def}\big(\mathsf{SqHamiltonian}(h_0, t_0)\big) \Rightarrow \quad \text{AUTO}$

$\mathsf{Use\_def}\Big(\mathsf{Hamiltonian}\big(h_0, t_0, \mathsf{sqEdges}(t_0)\big)\Big) \Rightarrow \quad \text{AUTO}$

$\mathsf{Use\_def}(\bigcup) \Rightarrow \quad \mathit{Stat3}: \ x_0 \notin \{u : \ v \in h_0, u \in v\}$

$\langle \{x_0, y_0\}, x_0 \rangle \hookrightarrow \mathit{Stat3}(\mathit{Stat2}\star) \Rightarrow \quad \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow \quad \text{AUTO}$

$\mathsf{ELEM} \Rightarrow \quad \{x_0, y_0\} \neq \{w_0, x_0\} \ \& \ z_0 \notin t_1 \ \&$
$\quad y_0 \in z_0 \ \& \ y_0 \in x_0 \ \& \ w_0 \in y_0 \ \& \ w_0 \in x_0$

$\langle t_1 \cup \{z_0\}, t_1, z_0, y_0, h_1, x_0, w_0 \rangle \hookrightarrow T\mathsf{hamiltonian}_1 \ (\mathit{Stat1}\star) \Rightarrow$
$\quad \mathsf{SqHamiltonian}(h_1 \setminus \{\{x_0, y_0\}\} \cup \{\{x_0, z_0\}, \{z_0, y_0\}\}, t_1 \cup \{z_0\})$

$(\mathit{Stat1}\star)\mathsf{ELEM} \Rightarrow \quad h_1 \setminus \{\{x_0, y_0\}\} = h_0 \setminus \{\{w_0, y_0\}\} \cup \{\{w_0, x_0\}\}$

$(\mathit{Stat1}\star)\mathsf{ELEM} \Rightarrow \quad h_1 \setminus \{\{x_0, y_0\}\} \cup \{\{x_0, z_0\}, \{z_0, y_0\}\}$
$\quad = h_0 \setminus \{\{w_0, y_0\}\} \cup \{\{w_0, x_0\}, \{x_0, z_0\}, \{z_0, y_0\}\}$

$\mathsf{EQUAL} \Rightarrow \quad \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow \quad \text{QED}$

THEOREM $\mathsf{hamiltonian}_3$: [Trivial Hamiltonian cycles]
$S = \{X, Y, Z\} \ \& \ X \in Y \ \& \ Y \in Z \rightarrow$
$\quad \mathsf{SqHamiltonian}(\{\{X, Y\}, \{Y, Z\}, \{Z, X\}\}, S)$. PROOF:

$\mathsf{Suppose\_not}(s, x_0, y_0, z_0) \Rightarrow \quad \text{AUTO}$

Arguing by contradiction, assume that $\{\{x_0, y_0\}, \{y_0, z_0\}, \{z_0, x_0\}\}$ is not a ‘square Hamiltonian’ cycle for $s = \{x_0, y_0, z_0\}$, where $x_0 \in y_0$ and $y_0 \in z_0$ holds. We will first exclude the possibility that $\{\{x_0, y_0\}, \{y_0, z_0\}, \{z_0, x_0\}\}$ is not a Hamiltonian cycle in the ‘square edges’ of s; after discarding this, we will also exclude that this cycle may fail to satisfy the restraining condition that it has a genuine membership edge incident into each source of s.

$\mathsf{Use\_def}\big(\mathsf{SqHamiltonian}(\{\{x_0, y_0\}, \{y_0, z_0\}, \{z_0, x_0\}\}, s)\big) \Rightarrow \quad \text{AUTO}$

$\mathsf{Use\_def}\Big(\mathsf{Hamiltonian}\big(\{\{x_0, y_0\}, \{y_0, z_0\}, \{z_0, x_0\}\}, s, \mathsf{sqEdges}(s)\big)\Big) \Rightarrow \quad \text{AUTO}$

$\langle x_0, y_0, z_0 \rangle \hookrightarrow T\mathsf{cycle}_0 \ \Rightarrow \quad \text{AUTO}$

$\mathsf{ELEM} \Rightarrow \quad \mathit{Stat1}: \ s = \{x_0, y_0, z_0\} \ \& \ x_0 \in y_0 \ \& \ y_0 \in z_0$

$\mathsf{Suppose} \Rightarrow \quad \mathit{Stat8}: \ \{\{x_0, y_0\}, \{y_0, z_0\}, \{z_0, x_0\}\} \not\subseteq \mathsf{sqEdges}(s)$

$\quad \mathsf{Use\_def}\big(\mathsf{sqEdges}(s)\big) \Rightarrow \quad \text{AUTO}$

$\quad \langle e_0 \rangle \hookrightarrow \mathit{Stat8}(\mathit{Stat8}\star) \Rightarrow \quad \mathit{Stat9}:$
$\quad\quad e_0 \notin \{\{x, y\} : \ x \in s, y \in s \setminus \{x\}, z \in s \cap x \mid y = z \vee y \in z \vee z \in y\} \ \&$
$\quad\quad e_0 \in \{\{x_0, y_0\}, \{y_0, z_0\}, \{z_0, x_0\}\}$

$\quad \langle z_0, y_0, y_0 \rangle \hookrightarrow \mathit{Stat9}(\mathit{Stat1}, \mathit{Stat1}\star) \Rightarrow \quad e_0 \neq \{y_0, z_0\}$

$\quad \langle z_0, x_0, y_0 \rangle \hookrightarrow \mathit{Stat9}(\mathit{Stat1}, \mathit{Stat9}\star) \Rightarrow \quad e_0 \neq \{z_0, x_0\}$

$\quad \langle y_0, x_0, x_0 \rangle \hookrightarrow \mathit{Stat9}(\mathit{Stat1}, \mathit{Stat1}\star) \Rightarrow \quad e_0 \neq \{x_0, y_0\}$

$(\mathit{Stat9}\star)\mathsf{Discharge} \Rightarrow \quad \text{AUTO}$

$\mathsf{Suppose} \Rightarrow \quad \mathit{Stat4}: \ \bigcup \{\{x_0, y_0\}, \{y_0, z_0\}, \{z_0, x_0\}\} \neq s$

$\quad (\mathit{Stat1}, \mathit{Stat1}\star)\mathsf{ELEM} \Rightarrow \quad s = \{x_0, y_0, z_0\} \ \&$
$\quad\quad \{\{x_0, y_0\}, \{y_0, z_0\}, \{z_0, x_0\}\} = \{\{x_0, y_0\}, \{y_0, z_0\}\} \cup \{\{z_0, x_0\}\} \ \&$
$\quad\quad \{x_0, y_0, z_0\} \cup \{z_0, x_0\} = \{x_0, y_0, z_0\}$

$\quad \langle \{\{x_0, y_0\}, \{y_0, z_0\}\}, \{\{z_0, x_0\}\} \rangle \hookrightarrow T2c \ (\mathit{Stat5}\star) \Rightarrow \quad \mathit{Stat5}:$
$\quad\quad \bigcup(\{\{x_0, y_0\}, \{y_0, z_0\}\} \cup \{\{z_0, x_0\}\})$

$= \bigcup\{\{x_0, y_0\}, \{y_0, z_0\}\} \cup \bigcup\{\{z_0, x_0\}\}$

$\langle\, \{\{x_0, y_0\}, \{y_0, z_0\}\}, \{x_0, y_0\}, \{y_0, z_0\}\,\rangle \hookrightarrow T2a\ (Stat6\star) \Rightarrow \quad Stat6:$

$\bigcup\{\{x_0, y_0\}, \{y_0, z_0\}\} = \{x_0, y_0, z_0\}$

$\langle\, \{\{z_0, x_0\}, \{z_0, x_0\}\}, \{z_0, x_0\}, \{z_0, x_0\}\,\rangle \hookrightarrow T2a\ (Stat7\star) \Rightarrow \quad Stat7:$

$\bigcup\{\{z_0, x_0\}, \{z_0, x_0\}\} = \{z_0, x_0\}\ \&\ \{\{z_0, x_0\}, \{z_0, x_0\}\} = \{\{z_0, x_0\}\}$

$\mathsf{EQUAL}(Stat4) \Rightarrow\ \ \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow\ \ Stat10:$

$\neg\langle \forall z \in s \setminus \bigcup s, \exists y \in z \mid \{z, y\} \in \{\{x_0, y_0\}, \{y_0, z_0\}, \{z_0, x_0\}\}\,\rangle$

We conclude by checking that $\{\{x_0, y_0\}, \{y_0, z_0\}, \{z_0, x_0\}\}$ owns a genuine membership edge incident into each source of s; as a matter of fact, z_0 is the only source of s and $\{y_0, z_0\}$ is a membership edge.

$\langle z'\rangle \hookrightarrow Stat10(Stat10\star) \Rightarrow \quad Stat11:$

$\neg\langle \exists y \in z' \mid \{z', y\} \in \{\{x_0, y_0\}, \{y_0, z_0\}, \{z_0, x_0\}\}\,\rangle\ \&\ z' \in s \setminus \bigcup s$

$\langle y_0\rangle \hookrightarrow Stat11(Stat11\star) \Rightarrow \quad Stat12:$

$y_0 \notin z' \vee \{z', y_0\} \notin \{\{x_0, y_0\}, \{y_0, z_0\}, \{z_0, x_0\}\}$

$\mathsf{Use\_def}(\bigcup) \Rightarrow \quad Stat13:\ z' \notin \{u : v \in s, u \in v\}\ \&\ z' \in s$

$\langle z_0, y_0\rangle \hookrightarrow Stat13(Stat12, Stat1\star) \Rightarrow \quad Stat14:\ z' = x_0$

$\langle y_0, x_0\rangle \hookrightarrow Stat13(Stat14, Stat13, Stat12, Stat1\star) \Rightarrow \quad \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow \quad$ QED

Any non-trivial transitive set whose square is devoid of Hamiltonian cycles must strictly comprise certain sets.

THEOREM $\mathsf{hamiltonian}_4$: [Potential revealers of non-Hamiltonicity]

$\mathsf{Trans}(S)\ \&\ S \not\subseteq \{\emptyset, \{\emptyset\}\}\ \&\ \neg\langle \exists h \mid \mathsf{SqHamiltonian}(h, S)\rangle \rightarrow$

$S \neq \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}\}\ \&\ S \neq \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}\ \&$

$S \neq \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\emptyset, \{\emptyset\}\}\}\ \&\ S \supsetneq \{\emptyset, \{\emptyset\}\}\ \&$

$\big(\{\{\emptyset\}\} \in S \vee \{\emptyset, \{\emptyset\}\} \in S\big)$. PROOF:

$\mathsf{Suppose\_not}(t) \Rightarrow \quad$ AUTO

Indeed, any set satisfying the premises of our present claim must, due to its transitivity and non-triviality, include either one of the Hamiltonian cycles endowed with the vertices $\emptyset, \{\emptyset\}, \{\{\emptyset\}\}$ and $\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}$, respectively; but it must also own additional elements, else the last premise would be falsified. Moreover, it cannot have exactly the elements $\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\emptyset, \{\emptyset\}\}$, as these form a Hamiltonian cycle.

$\langle t, \{\emptyset, \{\emptyset\}\}\,\rangle \hookrightarrow T4b \ \Rightarrow \quad Stat1:\ \neg\langle \exists h \mid \mathsf{SqHamiltonian}(h, t)\rangle\ \&$

$\big(t \supseteq \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}\} \vee t \supseteq \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}\big)$

The cases $t = \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}\}$ and $t = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$ must be excluded, because we have the respective Hamiltonian cycles

$$\{\{\emptyset, \{\emptyset\}\}, \{\{\emptyset\}, \{\{\emptyset\}\}\}, \{\{\{\emptyset\}\}, \emptyset\}\},$$
$$\{\{\emptyset, \{\emptyset\}\}, \{\{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}, \{\{\emptyset, \{\emptyset\}\}, \emptyset\}\}.$$

$\langle \mathsf{t}, \emptyset, \{\emptyset\}, \{\{\emptyset\}\} \rangle \hookrightarrow T\mathsf{hamiltonian}_3 \Rightarrow$ AUTO
$\langle \{\{\emptyset, \{\emptyset\}\}, \{\{\emptyset\}, \{\{\emptyset\}\}\}, \{\{\{\emptyset\}\}, \emptyset\}\} \rangle \hookrightarrow \mathit{Stat1}(\star) \Rightarrow \quad \mathsf{t} \not= \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}\}$
$\langle \mathsf{t}, \emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\} \rangle \hookrightarrow T\mathsf{hamiltonian}_3 \Rightarrow$ AUTO
$\langle \{\{\emptyset, \{\emptyset\}\}, \{\{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}, \{\{\emptyset, \{\emptyset\}\}, \emptyset\}\} \rangle \hookrightarrow \mathit{Stat1}(\star) \Rightarrow$
$\mathit{Stat2}: \ \mathsf{t} = \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\emptyset, \{\emptyset\}\}\}$

|| Having thus established that $\mathsf{t} = \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\emptyset, \{\emptyset\}\}\}$, we can now exploit Theorem hamiltonian$_2$ to enrich the Hamiltonian cycle for $\{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}\}$ into one which does to our case.

$\langle \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}\}, \emptyset, \{\emptyset\}, \{\{\emptyset\}\} \rangle \hookrightarrow T\mathsf{hamiltonian}_3(\mathit{Stat2}\star) \Rightarrow$
$\mathsf{SqHamiltonian}(\{\{\emptyset, \{\emptyset\}\}, \{\{\emptyset\}, \{\{\emptyset\}\}\}, \{\{\{\emptyset\}\}, \emptyset\}\}, \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}\})$
$\langle \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}\}, \emptyset, \{\emptyset\}, \{\{\emptyset\}\} \rangle \hookrightarrow T\mathsf{hamiltonian}_3 \Rightarrow$
$\mathsf{SqHamiltonian}(\{\{\emptyset, \{\emptyset\}\}, \{\{\emptyset\}, \{\{\emptyset\}\}\}, \{\{\{\emptyset\}\}, \emptyset\}\}, \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}\})$
$\langle \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\emptyset, \{\emptyset\}\}\}, \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}\}, \{\emptyset, \{\emptyset\}\}, \{\emptyset\},$
$\{\{\emptyset, \{\emptyset\}\}, \{\{\emptyset\}, \{\{\emptyset\}\}\}, \{\{\{\emptyset\}\}, \emptyset\}\}, \emptyset \rangle \hookrightarrow T\mathsf{hamiltonian}_1(\mathit{Stat2}\star) \Rightarrow$
$\mathsf{SqHamiltonian}(\{\{\emptyset, \{\emptyset\}\}, \{\{\emptyset\}, \{\{\emptyset\}\}\}, \{\{\{\emptyset\}\}, \emptyset\}\} \setminus \{\{\emptyset, \{\emptyset\}\}\} \cup$
$\{\{\emptyset, \{\emptyset, \{\emptyset\}\}\}, \{\{\emptyset, \{\emptyset\}\}, \{\emptyset\}\}\},$
$\{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\emptyset, \{\emptyset\}\}\})$
$\mathsf{EQUAL}(\mathit{Stat2}) \Rightarrow$
$\mathsf{SqHamiltonian}(\{\{\emptyset, \{\emptyset\}\}, \{\{\emptyset\}, \{\{\emptyset\}\}\}, \{\{\{\emptyset\}\}, \emptyset\}\} \setminus \{\{\emptyset, \{\emptyset\}\}\} \cup$
$\{\{\emptyset, \{\emptyset, \{\emptyset\}\}\}, \{\{\emptyset, \{\emptyset\}\}, \{\emptyset\}\}\}, \mathsf{t})$
$\langle \{\{\emptyset, \{\emptyset\}\}, \{\{\emptyset\}, \{\{\emptyset\}\}\}, \{\{\{\emptyset\}\}, \emptyset\}\} \setminus \{\{\emptyset, \{\emptyset\}\}\} \cup \{\{\emptyset, \{\emptyset, \{\emptyset\}\}\}, \{\{\emptyset, \{\emptyset\}\}, \{\emptyset\}\}\} \rangle$
$\hookrightarrow \mathit{Stat1}(\mathit{Stat2}\star) \Rightarrow \quad \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow$ QED

A.7 Hamiltonicity of Squared Claw-Free Sets

|| Non-trivial, claw-free, finite transitive sets have Hamiltonian squares.

THEOREM $\mathsf{clawFreeness}_1$: [Hamiltonicity of non-trivial, claw-free sets]
$\mathsf{Finite}(\mathsf{S}) \ \& \ \mathsf{Trans}(\mathsf{S}) \ \& \ \mathsf{ClawFree}(\mathsf{S}) \ \& \ \mathsf{S} \not\subseteq \{\emptyset, \{\emptyset\}\} \rightarrow$
$\langle \exists \mathsf{h} \mid \mathsf{SqHamiltonian}(\mathsf{h}, \mathsf{S}) \rangle$. PROOF:
$\mathsf{Suppose\_not}(\mathsf{s}_1) \Rightarrow$ AUTO

|| For, assuming the opposite, there would exist an inclusion-minimal, finite transitive non-trivial claw-free set whose square lacks a Hamiltonian cycle.

$\mathsf{APPLY} \ \langle \mathsf{fin}_\Theta : \mathsf{s}_0 \rangle \ \mathsf{finiteInduction}\Big(\mathsf{s}_0 \mapsto \mathsf{s}_1, \mathsf{P}(\mathsf{S}) \mapsto \big(\mathsf{Trans}(\mathsf{S}) \ \& \ \mathsf{ClawFree}(\mathsf{S}) \ \&$
$\mathsf{S} \not\subseteq \{\emptyset, \{\emptyset\}\} \ \& \ \neg \langle \exists \mathsf{h} \mid \mathsf{SqHamiltonian}(\mathsf{h}, \mathsf{S}) \rangle \big)\Big) \Rightarrow$

$\mathit{Stat1}: \ \langle \forall \mathsf{s} \mid \mathsf{s} \subseteq \mathsf{s}_0 \rightarrow \mathsf{Finite}(\mathsf{s}) \ \& \ \big(\mathsf{Trans}(\mathsf{s}) \ \& \ \mathsf{ClawFree}(\mathsf{s}) \ \&$
$\mathsf{s} \not\subseteq \{\emptyset, \{\emptyset\}\} \ \& \ \neg \langle \exists \mathsf{h} \mid \mathsf{SqHamiltonian}(\mathsf{h}, \mathsf{s}) \rangle \leftrightarrow \mathsf{s} = \mathsf{s}_0 \big) \rangle$

$$\langle \mathsf{s}_0 \rangle \hookrightarrow \mathit{Stat1}(\mathit{Stat1}\star) \Rightarrow \quad \mathit{Stat2}:\ \neg\langle \exists \mathsf{h} \mid \mathsf{SqHamiltonian}(\mathsf{h}, \mathsf{s}_0)\rangle \;\&\; \mathsf{Finite}(\mathsf{s}_0) \;\&\; \mathsf{Trans}(\mathsf{s}_0) \;\&\; \mathsf{ClawFree}(\mathsf{s}_0) \;\&\; \mathsf{s}_0 \not\subseteq \{\emptyset, \{\emptyset\}\}$$

Thanks to the finiteness of such an s_0, the THEORY pivotsForClawFreeness can be applied to s_0. We thereby pick an element x from the frontier of s_0, and an element y of x which is pivotal relative to s_0. This y will have at most two in-neighbors (one of the two being x) in s_0. We denote by z an in-neighbor of y in s_0, such that z differs from x if possible. Observe, among others, that neither one of x, z can belong to the other.

$$\mathsf{APPLY}\ \langle \mathsf{x}_\Theta : \mathsf{x}, \mathsf{y}_\Theta : \mathsf{y}, \mathsf{z}_\Theta : \mathsf{z}, \mathsf{t}_\Theta : \mathsf{t}\rangle\ \mathsf{pivotsForClawFreeness}(\mathsf{s}_0 \mapsto \mathsf{s}_0) \Rightarrow$$
$$\mathit{Stat3}:\ \{\mathsf{v} \in \mathsf{s}_0 \mid \mathsf{y} \in \mathsf{v}\} = \{\mathsf{x}, \mathsf{z}\} \;\&\; \mathsf{z} \in \mathsf{s}_0 \;\&\; \mathsf{y} \in \mathsf{z} \;\&\; \mathsf{y} \in \mathsf{x} \;\&\; \mathsf{y}, \mathsf{x} \in \mathsf{s}_0 \;\&\; \mathsf{y} \notin \bigcup\bigcup \mathsf{s}_0 \;\&\; \mathsf{t} = \{\mathsf{u} \in \mathsf{s}_0 \mid \mathsf{y} \notin \mathsf{u}\} \;\&\; \mathsf{Trans}(\mathsf{t}) \;\&\; \mathsf{ClawFree}(\mathsf{t}) \;\&\; \mathsf{s}_0 \supseteq \mathsf{t} \;\&\; \mathsf{x} \notin \mathsf{t} \;\&\; \mathsf{y} \in \mathsf{t} \setminus \bigcup \mathsf{t} \;\&\; \mathsf{t} = \mathsf{s}_0 \setminus \{\mathsf{x}, \mathsf{z}\} \;\&\; \mathsf{x} \notin \mathsf{z} \;\&\; \mathsf{z} \notin \mathsf{x}$$

Thus it turns out readily that removal of x, z from s_0 leads to a set t to which, unless t is 'trivial' (i.e. a subset of $\{\emptyset, \{\emptyset\}\}$), the inductive hypothesis applies; hence, by that hypothesis, there is a Hamiltonian cycle h_0 for t.

$$\mathsf{Suppose} \Rightarrow \quad \mathsf{t} \subseteq \{\emptyset, \{\emptyset\}\}$$

In order that t be trivial, we should have $\mathsf{s}_0 \subseteq \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\emptyset, \{\emptyset\}\}\}$; but then, as already shown in the proof of Theorem $\mathsf{hamiltonian}_4$, we have the ability, either directly, or by enrichment of a Hamiltonian cycle for $\{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}\}$, to construct a Hamiltonian cycle for s_0: thus, if we suppose $\mathsf{t} \subseteq \{\emptyset, \{\emptyset\}\}$ then we get a contradiction.

$$\langle \mathsf{s}_0, \mathsf{x}, \mathsf{z}\rangle \hookrightarrow T3d(\mathit{Stat2}\star) \Rightarrow \quad \mathit{Stat7}:\ \mathsf{s}_0 \subseteq \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\emptyset, \{\emptyset\}\}\}$$
$$\langle \mathsf{s}_0 \rangle \hookrightarrow T\mathsf{hamiltonian}_4(\mathit{Stat2}, \mathit{Stat7}\star) \Rightarrow \quad \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow \quad \text{AUTO}$$
$$\langle \mathsf{t} \rangle \hookrightarrow \mathit{Stat1}(\mathit{Stat3}\star) \Rightarrow \quad \mathit{Stat9}:\ \langle \exists \mathsf{h} \mid \mathsf{SqHamiltonian}(\mathsf{h}, \mathsf{t})\rangle \;\&\; \mathsf{t} \not\subseteq \{\emptyset, \{\emptyset\}\}$$
$$\langle \mathsf{h}_0 \rangle \hookrightarrow \mathit{Stat9}(\mathit{Stat9}\star) \Rightarrow \quad \mathit{Stat10}:\ \mathsf{SqHamiltonian}(\mathsf{h}_0, \mathsf{t})$$
$$\mathsf{Use\_def}\Big(\mathsf{Hamiltonian}\big(\mathsf{h}_0, \mathsf{t}, \mathsf{sqEdges}(\mathsf{t})\big)\Big) \Rightarrow \quad \text{AUTO}$$
$$\mathsf{Use\_def}(\mathsf{SqHamiltonian}) \Rightarrow$$
$$\mathit{Stat11}:\ \langle \forall \mathsf{x} \in \mathsf{t} \setminus \bigcup \mathsf{t}, \exists \mathsf{y} \in \mathsf{x} \mid \{\mathsf{x}, \mathsf{y}\} \in \mathsf{h}_0 \rangle \;\&\; \mathsf{Cycle}(\mathsf{h}_0) \;\&\; \bigcup \mathsf{h}_0 = \mathsf{t} \;\&\; \mathsf{h}_0 \subseteq \mathsf{sqEdges}(\mathsf{t})$$

It follows from y being a source of $\mathsf{t} = \bigcup \mathsf{h}_0$ that there is an edge $\{\mathsf{y}, \mathsf{w}\}$, with $\mathsf{w} \in \mathsf{y}$, in h_0, If $\mathsf{x} = \mathsf{z}$, to get a Hamiltonian cycle h_1 for s_0 (a fact conflicting with the inductive hypothesis) it will suffice to take $\mathsf{h}_1 = \mathsf{h}_0 \setminus \{\{\mathsf{y}, \mathsf{w}\}\} \cup \{\{\mathsf{x}, \mathsf{y}\}, \{\mathsf{x}, \mathsf{w}\}\}$, where $\{\mathsf{x}, \mathsf{w}\}$ is a square edge because $\mathsf{w} \in \mathsf{y}$ and $\mathsf{y} \in \mathsf{x}$ both hold. On the other hand, if $\mathsf{x} \neq \mathsf{z}$, claw-freeness implies that either $\mathsf{w} \in \mathsf{x}$ or $\mathsf{w} \in \mathsf{z}$. Assume the former for definiteness, and put $\mathsf{h}_2 = \mathsf{h}_0 \setminus \{\{\mathsf{y}, \mathsf{w}\}\} \cup \{\{\mathsf{y}, \mathsf{z}\}, \{\mathsf{z}, \mathsf{x}\}, \{\mathsf{x}, \mathsf{w}\}\}$, where $\{\mathsf{x}, \mathsf{z}\}$ is a square edge and $\{\mathsf{x}, \mathsf{w}\}$ and $\{\mathsf{y}, \mathsf{z}\}$ are genuine edges incident in the sources x, z. We are again facing a contradiction, because h_2 is a Hamiltonian cycle for s_0.

$\langle y, w\rangle \hookrightarrow \mathit{Stat11}(\mathit{Stat3}\star) \Rightarrow \quad \mathit{Stat12}:\ w \in y \ \&\ \{w, y\} \in h_0$

Suppose $\Rightarrow \quad x = z$

$\langle s_0, t, x, y, h_0, w, \emptyset\rangle \hookrightarrow T\text{hamiltonian}_1(\mathit{Stat3}\star) \Rightarrow$

$\text{SqHamiltonian}(h_0 \setminus \{\{w, y\}\} \cup \{\{w, x\}, \{x, y\}\}, s_0)$

$\langle h_0 \setminus \{\{w, y\}\} \cup \{\{w, x\}, \{x, y\}\}\rangle \hookrightarrow \mathit{Stat2}(\mathit{Stat2}\star) \Rightarrow \quad$ false;

Discharge $\Rightarrow \quad \mathit{Stat13}:\ x \neq z$

$\langle s_0, y\rangle \hookrightarrow T3c(\mathit{Stat2}\star) \Rightarrow \quad \mathit{Stat14}: w \in s_0$

$\langle s_0, y, x, z, w\rangle \hookrightarrow T\text{clawFreeness}_b(\mathit{Stat2}, \mathit{Stat3}, \mathit{Stat12}, \mathit{Stat13}, \mathit{Stat14}\star) \Rightarrow\ w \in x \cup z$

Suppose $\Rightarrow \quad \mathit{Stat15}:\ w \in x$

$\langle s_0, t, x, z, y, h_0, w\rangle \hookrightarrow T\text{hamiltonian}_2(\mathit{Stat3}, \mathit{Stat13}, \mathit{Stat10}, \mathit{Stat12}, \mathit{Stat15}\star) \Rightarrow$

$\text{SqHamiltonian}(h_0 \setminus \{\{w, y\}\} \cup \{\{w, x\}, \{x, z\}, \{z, y\}\}, s_0)$

$\langle h_0 \setminus \{\{w, y\}\} \cup \{\{w, x\}, \{x, z\}, \{z, y\}\}\rangle \hookrightarrow \mathit{Stat2}(\mathit{Stat2}\star) \Rightarrow \quad$ false;

Discharge $\Rightarrow \quad \mathit{Stat16}:\ w \in z$

$\langle s_0, t, z, x, y, h_0, w\rangle \hookrightarrow T\text{hamiltonian}_2(\mathit{Stat3}, \mathit{Stat13}, \mathit{Stat10}, \mathit{Stat12}, \mathit{Stat16}\star) \Rightarrow$

$\text{SqHamiltonian}(h_0 \setminus \{\{w, y\}\} \cup \{\{w, z\}, \{z, x\}, \{x, y\}\}, s_0)$

$\langle h_0 \setminus \{\{w, y\}\} \cup \{\{w, z\}, \{z, x\}, \{x, y\}\}\rangle \hookrightarrow \mathit{Stat2}(\mathit{Stat2}\star) \Rightarrow \quad$ false;

Discharge $\Rightarrow$ QED

A.8 Perfect Matchings

Next we introduce the notion of *perfect matching*. This is a partition consisting of doubletons one of whose elements belongs to the other. Special cases of a perfect matching are: the empty set and, more generally, all subsets of a perfect matching.

DEF perfect_matching: [Set of disjoint membership pairs]

$\text{PerfectMatching}(M) \ \leftrightarrow_{\text{Def}}$

$\langle \forall p \in M, \exists x \in p, y \in x, \forall q \in M \mid x \in q \vee y \in q \rightarrow \{x, y\} = q\rangle$

THEOREM perfectMatching$_0$: [The null set is a perfect matching]

$\text{PerfectMatching}(\emptyset)$. PROOF:

Suppose_not() $\Rightarrow$ AUTO

Use_def(PerfectMatching) $\Rightarrow \quad \mathit{Stat0}:\ \neg\langle \forall p \in \emptyset, \exists x \in p, y \in x, \forall q \in \emptyset \mid$

$x \in q \vee y \in q \rightarrow \{x, y\} = q\rangle$

$\langle p_1\rangle \hookrightarrow \mathit{Stat0} \Rightarrow \quad$ false; Discharge $\Rightarrow$ QED

THEOREM perfectMatching$_1$: [Perfect matchings consist of true doubletons]

$\text{PerfectMatching}(M) \ \&\ P \in M \rightarrow P \notin \{\emptyset, \{X\}\}$. PROOF:

Suppose_not$(m, p_0, x_0) \Rightarrow$ AUTO

Use_def(PerfectMatching) $\Rightarrow \quad \mathit{Stat1}:\ \langle \forall p \in m, \exists x \in p, y \in x, \forall q \in m \mid$

$x \in q \vee y \in q \rightarrow \{x, y\} = q\rangle$

$\langle p_0, x, y, p_0\rangle \hookrightarrow \mathit{Stat1}(\star) \Rightarrow \quad$ false; Discharge $\Rightarrow$ QED

THEOREM perfectMatching$_2$: [All subsets of a perfect matching are perfect]
PerfectMatching(M) & $\mathsf{M} \supseteq \mathsf{N} \rightarrow$ PerfectMatching(N). PROOF:
Suppose_not(m, n) $\Rightarrow$ AUTO
Set_monot $\Rightarrow$
$\langle \forall p \in m, \exists x \in p, y \in x, \forall q \in m \mid x \in q \vee y \in q \rightarrow \{x, y\} = q \rangle \rightarrow$
$\langle \forall p \in n, \exists x \in p, y \in x, \forall q \in n \mid x \in q \vee y \in q \rightarrow \{x, y\} = q \rangle$
Use_def(PerfectMatching) $\Rightarrow$ false; Discharge $\Rightarrow$ QED

By adjoining a pair $\{x, y\}$ with $y \in x$ to a perfect matching none of whose blocks has either x or y as an element, we always obtain a perfect matching.

THEOREM perfectMatching$_3$: [Bottom-up assembly of a perfect matching]
PerfectMatching(M) & $X \notin \bigcup M$ & $Y \notin \bigcup M$ & $Y \in X \rightarrow$
PerfectMatching($M \cup \{\{X, Y\}\}$). PROOF:
Suppose_not(m, x_0, y_0) $\Rightarrow$ *Stat2* : PerfectMatching(m) & $x_0 \notin \bigcup m$ &
$y_0 \notin \bigcup m$ & $y_0 \in x_0$ & $\neg$PerfectMatching($m \cup \{\{x_0, y_0\}\}$)
Suppose $\Rightarrow$ *Stat3* : $\neg \langle \forall q \in m \mid x_0 \notin q \;\&\; y_0 \notin q \rangle$
Use_def($\bigcup$) $\Rightarrow$ *Stat4* : $x_0 \notin \{v : u \in m, v \in u\}$ &
$y_0 \notin \{v : u \in m, v \in u\}$
$\langle q_2 \rangle \hookrightarrow$ *Stat3*(*Stat3*$\star$) $\Rightarrow$ $q_2 \in m \;\&\; x_0 \in q_2 \vee y_0 \in q_2$
$\langle q_2, x_0, q_2, y_0 \rangle \hookrightarrow$ *Stat4*(*Stat4*$\star$) $\Rightarrow$ false; Discharge $\Rightarrow$
Stat5 : $\langle \forall q \in m \mid x_0 \notin q \;\&\; y_0 \notin q \rangle$
Use_def(PerfectMatching) $\Rightarrow$
Stat6 : $\neg \langle \forall p \in m \cup \{\{x_0, y_0\}\}, \exists x \in p, y \in x,$
$\forall q \in m \cup \{\{x_0, y_0\}\} \mid x \in q \vee y \in q \rightarrow \{x, y\} = q \rangle$ &
Stat7 : $\langle \forall p \in m, \exists x \in p, y \in x,$
$\forall q \in m \mid x \in q \vee y \in q \rightarrow \{x, y\} = q \rangle$
$\langle p_0 \rangle \hookrightarrow$ *Stat6*(*Stat6*$\star$) $\Rightarrow$
Stat8 : $\neg \langle \exists x \in p_0, y \in x, \forall q \in m \cup \{\{x_0, y_0\}\} \mid x \in q \vee y \in q \rightarrow$
$\{x, y\} = q \rangle \;\&\; p_0 \in m \cup \{\{x_0, y_0\}\}$
Suppose $\Rightarrow$ *Stat9* : $p_0 = \{x_0, y_0\}$
$\langle x_0, y_0 \rangle \hookrightarrow$ *Stat8*(*Stat2*, *Stat9*$\star$) $\Rightarrow$
Stat10 : $\neg \langle \forall q \in m \cup \{\{x_0, y_0\}\} \mid x_0 \in q \vee y_0 \in q \rightarrow \{x_0, y_0\} = q \rangle$
$\langle q_1 \rangle \hookrightarrow$ *Stat9*(*Stat9*$\star$) $\Rightarrow$ $q_1 \in m \;\&\; x_0 \in q_1 \vee y_0 \in q_1$
$\langle q_1 \rangle \hookrightarrow$ *Stat5*(*Stat10*$\star$) $\Rightarrow$ false; Discharge $\Rightarrow$ AUTO
$\langle p_0, x_1, y_1 \rangle \hookrightarrow$ *Stat7* $\Rightarrow$ AUTO
$\langle x_1, y_1 \rangle \hookrightarrow$ *Stat8*(*Stat8*$\star$) $\Rightarrow$
Stat13 : $\neg \langle \forall q \in m \cup \{\{x_0, y_0\}\} \mid x_1 \in q \vee y_1 \in q \rightarrow \{x_1, y_1\} = q \rangle$&
Stat12 : $\langle \forall q \in m \mid x_1 \in q \vee y_1 \in q \rightarrow \{x_1, y_1\} = q \rangle \;\&\; p_0 \in m$ &
$x_1 \in p_0 \;\&\; y_1 \in x_1$
$\langle q_0, q_0 \rangle \hookrightarrow$ *Stat13*(*Stat13*$\star$) $\Rightarrow$ *Stat14* : $q_0 = \{x_0, y_0\} \;\&\; x_1 \in q_0 \vee y_1 \in q_0$
$\langle p_0 \rangle \hookrightarrow$ *Stat5*($\star$) $\Rightarrow$ $x_0 \notin p_0 \;\&\; y_0 \notin p_0$
$\langle p_0 \rangle \hookrightarrow$ *Stat12*(*Stat13*, *Stat13*$\star$) $\Rightarrow$ $\{x_1, y_1\} = p_0$
(*Stat14*$\star$)Discharge $\Rightarrow$ QED

If, in a perfect matching m, we replace one block $\{\mathsf{y},\mathsf{w}\}$ by pairs $\{\mathsf{y},\mathsf{z}\}, \{\mathsf{x},\mathsf{w}\}$, then, under suitable conditions ensuring disjointness between blocks and membership within each block, we get a perfect matching again.

THEOREM $\mathsf{perfectMatching}_4$: [Deviated perfect matching]

$$\mathsf{PerfectMatching}(\mathsf{M}) \;\&\; \{\mathsf{Y},\mathsf{W}\} \in \mathsf{M} \;\&\; \mathsf{X} \notin \bigcup \mathsf{M} \;\&\; \mathsf{Z} \notin \bigcup \mathsf{M} \;\&\; \mathsf{Y} \in \mathsf{Z} \;\&$$
$$\mathsf{Y} \neq \mathsf{X} \;\&\; \mathsf{X} \neq \mathsf{Z} \;\&\; \mathsf{W} \in \mathsf{X} \rightarrow$$
$$\mathsf{PerfectMatching}(\mathsf{M} \setminus \{\{\mathsf{Y},\mathsf{W}\}\} \cup \{\{\mathsf{Y},\mathsf{Z}\},\{\mathsf{X},\mathsf{W}\}\}).$$ PROOF:

$\mathsf{Suppose\_not}(\mathsf{m},\mathsf{y}_0,\mathsf{w}_0,\mathsf{x}_0,\mathsf{z}_0) \Rightarrow$ AUTO

For assuming that $\mathsf{m},\mathsf{y}_0,\mathsf{w}_0,\mathsf{x}_0,\mathsf{z}_0$ are a counterexample to the claim, we could get a contradiction arguing as follows. Begin by observing that neither y_0 nor w_0 can belong to the unionset of the perfect submatching $\mathsf{m}\setminus\{\{\mathsf{y}_0,\mathsf{w}_0\}\}$ of m.

$\mathsf{Suppose} \Rightarrow$ *Stat1* : $\{\mathsf{y}_0,\mathsf{w}_0\} \cap \bigcup(\mathsf{m}\setminus\{\{\mathsf{y}_0,\mathsf{w}_0\}\}) \neq \emptyset$

$\mathsf{Use\_def}(\mathsf{PerfectMatching}) \Rightarrow$ *Stat2* :

$\langle \forall \mathsf{p} \in \mathsf{m}, \exists \mathsf{x} \in \mathsf{p}, \mathsf{y} \in \mathsf{x}, \forall \mathsf{q} \in \mathsf{m} \mid \mathsf{x} \in \mathsf{q} \vee \mathsf{y} \in \mathsf{q} \rightarrow \{\mathsf{x},\mathsf{y}\} = \mathsf{q}\rangle$

$\mathsf{Use\_def}\big(\bigcup(\mathsf{m}\setminus\{\{\mathsf{y}_0,\mathsf{w}_0\}\})\big) \Rightarrow$ AUTO

$\langle \mathsf{w}_1 \rangle \hookrightarrow$ *Stat1* $\Rightarrow$ *Stat3* :

$\mathsf{w}_1 \in \{\mathsf{u} : \mathsf{v} \in \mathsf{m}\setminus\{\{\mathsf{y}_0,\mathsf{w}_0\}\}, \mathsf{u} \in \mathsf{v}\} \;\&\; \mathsf{w}_1 \in \{\mathsf{y}_0,\mathsf{w}_0\}$

$\langle \mathsf{p}_0,\mathsf{w}_2 \rangle \hookrightarrow$ *Stat3* $\Rightarrow$ $\mathsf{p}_0 \in \mathsf{m}\setminus\{\{\mathsf{y}_0,\mathsf{w}_0\}\} \;\&\; \mathsf{w}_1 \in \mathsf{p}_0$

$\langle \mathsf{p}_0,\mathsf{x}_2,\mathsf{y}_2 \rangle \hookrightarrow$ *Stat2* $\Rightarrow$ *Stat4* :

$\langle \forall \mathsf{q} \in \mathsf{m} \mid \mathsf{x}_2 \in \mathsf{q} \vee \mathsf{y}_2 \in \mathsf{q} \rightarrow \{\mathsf{x}_2,\mathsf{y}_2\} = \mathsf{q}\rangle \;\&\; \mathsf{x}_2 \in \mathsf{p}_0 \;\&\; \mathsf{y}_2 \in \mathsf{x}_2$

$\langle \mathsf{p}_0 \rangle \hookrightarrow$ *Stat4* $\Rightarrow$ $\mathsf{p}_0 = \{\mathsf{x}_2,\mathsf{y}_2\}$

$\langle \{\mathsf{y}_0,\mathsf{w}_0\} \rangle \hookrightarrow$ *Stat4*$(\star) \Rightarrow$ false; $\mathsf{Discharge} \Rightarrow$ AUTO

$\langle \mathsf{m}, \mathsf{m}\setminus\{\{\mathsf{y}_0,\mathsf{w}_0\}\} \rangle \hookrightarrow T\mathsf{perfectMatching}_2 \Rightarrow$ $\mathsf{PerfectMatching}(\mathsf{m}\setminus\{\{\mathsf{y}_0,\mathsf{w}_0\}\})$

Thus, taking into account that $\mathsf{w}_0 \in \mathsf{x}_0$ and that $\mathsf{x}_0 \notin \bigcup \mathsf{m}$ which is a superset of $\bigcup(\mathsf{m}\setminus\{\{\mathsf{y}_0,\mathsf{w}_0\}\})$, we can extend the perfect matching $\mathsf{m}\setminus\{\{\mathsf{y}_0,\mathsf{w}_0\}\}$ with the doubleton $\{\mathsf{x}_0,\mathsf{w}_0\}$.

$\langle \mathsf{m}\setminus\{\{\mathsf{y}_0,\mathsf{w}_0\}\}, \mathsf{m} \rangle \hookrightarrow T2c \Rightarrow$ $\mathsf{x}_0 \notin \bigcup(\mathsf{m}\setminus\{\{\mathsf{y}_0,\mathsf{w}_0\}\})$

$\langle \mathsf{m}\setminus\{\{\mathsf{y}_0,\mathsf{w}_0\}\}, \mathsf{x}_0, \mathsf{w}_0 \rangle \hookrightarrow T\mathsf{perfectMatching}_3 \Rightarrow$

$\mathsf{PerfectMatching}(\mathsf{m}\setminus\{\{\mathsf{y}_0,\mathsf{w}_0\}\} \cup \{\{\mathsf{x}_0,\mathsf{w}_0\}\})$

Observe next that $\mathsf{x}_0 \neq \mathsf{y}_0$ and $\mathsf{z}_0 \neq \mathsf{w}_0$, because $\mathsf{x}_0 \notin \bigcup \mathsf{m}$ and $\mathsf{z}_0 \notin \bigcup \mathsf{m}$ whereas $\mathsf{y}_0 \in \bigcup \mathsf{m}$ and $\mathsf{w}_0 \in \bigcup \mathsf{m}$. It then follows from $\mathsf{z}_0 \neq \mathsf{w}_0$, thanks to the assumption $\mathsf{z}_0 \neq \mathsf{x}_0$, that z_0 does not belong to the unionset of the perfect matching $\mathsf{m}\setminus\{\{\mathsf{y}_0,\mathsf{w}_0\}\} \cup \{\{\mathsf{x}_0,\mathsf{w}_0\}\}$.

$\mathsf{Suppose} \Rightarrow$ $\mathsf{x}_0 = \mathsf{w}_0 \vee \mathsf{z}_0 = \mathsf{w}_0$

$\mathsf{Use\_def}(\bigcup) \Rightarrow$ *Stat5* : $\mathsf{z}_0 \notin \{\mathsf{u} : \mathsf{v} \in \mathsf{m}, \mathsf{u} \in \mathsf{v}\} \;\&\; \mathsf{x}_0 \notin \{\mathsf{u} : \mathsf{v} \in \mathsf{m}, \mathsf{u} \in \mathsf{v}\}$

$\langle \{\mathsf{y}_0,\mathsf{w}_0\}, \mathsf{w}_0, \{\mathsf{y}_0,\mathsf{w}_0\}, \mathsf{y}_0 \rangle \hookrightarrow$ *Stat5* $\Rightarrow$ false; $\mathsf{Discharge} \Rightarrow$ AUTO

$\mathsf{Suppose} \Rightarrow$ $\mathsf{z}_0 \in \bigcup(\mathsf{m}\setminus\{\{\mathsf{y}_0,\mathsf{w}_0\}\} \cup \{\{\mathsf{x}_0,\mathsf{w}_0\}\})$

$\langle \mathsf{m}\setminus\{\{\mathsf{y}_0,\mathsf{w}_0\}\}, \mathsf{m} \rangle \hookrightarrow T2c \Rightarrow$ $\mathsf{z}_0 \notin \bigcup(\mathsf{m}\setminus\{\{\mathsf{y}_0,\mathsf{w}_0\}\})$

$\langle \mathsf{m}\setminus\{\{\mathsf{y}_0,\mathsf{w}_0\}\}, \{\mathsf{x}_0,\mathsf{w}_0\} \rangle \hookrightarrow T2e \Rightarrow$ false; $\mathsf{Discharge} \Rightarrow$ AUTO

$\mathsf{y_0}$ cannot equal $\mathsf{w_0}$ either, the reason being that the set $\{\mathsf{y_0}, \mathsf{w_0}\}$ is a block of a perfect matching and hence it cannot be a singleton. If follows, thanks to the assumption $\mathsf{y_0} \in \mathsf{x_0}$ (entailing that $\mathsf{y_0} \neq \mathsf{x_0}$), that $\mathsf{y_0}$ does not belong to $\bigcup(\mathsf{m} \setminus \{\{\mathsf{y_0}, \mathsf{w_0}\}\} \cup \{\{\mathsf{x_0}, \mathsf{w_0}\}\})$ either.

$$\langle \mathsf{m}, \{\mathsf{y_0}, \mathsf{w_0}\}, \mathsf{w_0} \rangle \hookrightarrow T\,\mathsf{perfectMatching}_1 \;\Rightarrow\; \mathsf{y_0} \neq \mathsf{w_0}$$
$$\langle \mathsf{m} \setminus \{\{\mathsf{y_0}, \mathsf{w_0}\}\}, \{\mathsf{x_0}, \mathsf{w_0}\} \rangle \hookrightarrow T2e \;\Rightarrow\; \mathsf{y_0} \notin \bigcup(\mathsf{m} \setminus \{\{\mathsf{y_0}, \mathsf{w_0}\}\} \cup \{\{\mathsf{x_0}, \mathsf{w_0}\}\})$$

We now know that the perfect matching $\mathsf{m} \setminus \{\{\mathsf{y_0}, \mathsf{w_0}\}\} \cup \{\{\mathsf{x_0}, \mathsf{w_0}\}\}$ can be extended with the doubleton $\{\mathsf{y_0}, \mathsf{z_0}\}$, which readily leads us to the sought contradiction.

$$\langle \mathsf{m} \setminus \{\{\mathsf{y_0}, \mathsf{w_0}\}\} \cup \{\{\mathsf{x_0}, \mathsf{w_0}\}\}, \mathsf{z_0}, \mathsf{y_0} \rangle \hookrightarrow T\,\mathsf{perfectMatching}_3 \;\Rightarrow$$
$$\mathsf{PerfectMatching}(\mathsf{m} \setminus \{\{\mathsf{y_0}, \mathsf{w_0}\}\} \cup \{\{\mathsf{x_0}, \mathsf{w_0}\}\} \cup \{\{\mathsf{y_0}, \mathsf{z_0}\}\})\ \&$$
$$\mathsf{m} \setminus \{\{\mathsf{y_0}, \mathsf{w_0}\}\} \cup \{\{\mathsf{x_0}, \mathsf{w_0}\}\} \cup \{\{\mathsf{y_0}, \mathsf{z_0}\}\} =$$
$$\mathsf{m} \setminus \{\{\mathsf{y_0}, \mathsf{w_0}\}\} \cup \{\{\mathsf{y_0}, \mathsf{z_0}\}, \{\mathsf{x_0}, \mathsf{w_0}\}\}$$
$$\mathsf{EQUAL} \Rightarrow\; \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow\; \textsc{Qed}$$

A.9 Each Claw-Free Set Admits a Near-Perfect Matching

Every claw-free finite, transitive set admits a perfect matching perhaps omitting one of its elements.

THEOREM $\mathsf{clawFreeness}_2$: [Claw-free sets admit near-perfect matchings]

$$\mathsf{Finite}(\mathsf{S})\ \&\ \mathsf{Trans}(\mathsf{S})\ \&\ \mathsf{ClawFree}(\mathsf{S}) \rightarrow$$
$$\langle \exists \mathsf{m}, \mathsf{y} \mid \mathsf{PerfectMatching}(\mathsf{m})\ \&\ \mathsf{S} \setminus \{\mathsf{y}\} = \bigcup \mathsf{m} \rangle. \ \textsc{Proof}:$$
$$\mathsf{Suppose\_not}(\mathsf{s_1}) \Rightarrow\; \textsc{Auto}$$

For, supposing the contrary, there would be an inclusion-minimal finite set $\mathsf{s_0}$ which is transitive and claw-free, and such that no perfect matching m partitions the set $\mathsf{s_0}$ possibly deprived of an element $\mathsf{y_0}$.

$$\mathsf{APPLY}\ \langle \mathsf{fin}_\Theta : \mathsf{s_0} \rangle\ \mathsf{finiteInduction}\Big(\mathsf{s_0} \mapsto \mathsf{s_1}, \mathsf{P(S)} \mapsto \big(\mathsf{Trans}(\mathsf{S})\ \&\ \mathsf{ClawFree}(\mathsf{S})\ \&$$
$$\neg \langle \exists \mathsf{m}, \mathsf{y} \mid \mathsf{PerfectMatching}(\mathsf{m})\ \&\ \mathsf{S} \setminus \{\mathsf{y}\} = \bigcup \mathsf{m} \rangle\big)\Big) \Rightarrow$$
$$\mathit{Stat1}: \ \langle \forall \mathsf{s} \mid \mathsf{s} \subseteq \mathsf{s_0} \rightarrow \mathsf{Finite}(\mathsf{s})\ \&\ \big(\mathsf{Trans}(\mathsf{s})\ \&\ \mathsf{ClawFree}(\mathsf{s})\ \&$$
$$\neg \langle \exists \mathsf{m}, \mathsf{y} \mid \mathsf{PerfectMatching}(\mathsf{m})\ \&\ \mathsf{s} \setminus \{\mathsf{y}\} = \bigcup \mathsf{m} \rangle \leftrightarrow \mathsf{s} = \mathsf{s_0}\big) \rangle$$
$$\langle \mathsf{s_0} \rangle \hookrightarrow \mathit{Stat1}(\mathit{Stat1}\star) \Rightarrow$$
$$\mathit{Stat2}: \ \neg \langle \exists \mathsf{m}, \mathsf{y} \mid \mathsf{PerfectMatching}(\mathsf{m})\ \&\ \mathsf{s_0} \setminus \{\mathsf{y}\} = \bigcup \mathsf{m} \rangle\ \&$$
$$\mathsf{Trans}(\mathsf{s_0})\ \&\ \mathsf{ClawFree}(\mathsf{s_0})\ \&\ \mathsf{Finite}(\mathsf{s_0})$$

We observe that such an $\mathsf{s_0}$ cannot equal $\emptyset$ or $\{\emptyset\}$, else the null perfect matching would cover it.

$$
\begin{array}{l}
\mathsf{Suppose} \Rightarrow \quad \mathit{Stat3}:\ \mathsf{s_0} \cap \bigcup \mathsf{s_0} = \emptyset \\
\quad \langle \mathsf{s_0} \rangle \hookrightarrow T3a \ \Rightarrow \quad \textsc{Auto} \\
\quad \langle \mathsf{s_0} \rangle \hookrightarrow T31d \ \Rightarrow \quad \textsc{Auto} \\
\quad \langle \emptyset \rangle \hookrightarrow T31d \ \Rightarrow \quad \textsc{Auto} \\
\quad \langle \emptyset, \emptyset \rangle \hookrightarrow \mathit{Stat2} \Rightarrow \quad \neg \mathsf{PerfectMatching}(\emptyset) \\
\langle\ \rangle \hookrightarrow T\mathsf{perfectMatching_0}(\mathit{Stat3}\star) \Rightarrow \quad \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow \quad \mathsf{s_0} \cap \bigcup \mathsf{s_0} \neq \emptyset
\end{array}
$$

Thanks to the finiteness of $\mathsf{s_0}$, the THEORY pivotsForClawFreeness can be applied to $\mathsf{s_0}$. We thereby pick an element x from the frontier of $\mathsf{s_0}$ and an element y of x which is pivotal relative to $\mathsf{s_0}$. This y will have at most two $\in$-predecessors (one of the two being x) in $\mathsf{s_0}$. We denote by z a predecessor of y in $\mathsf{s_0}$, such that z differs from x if possible.

$$
\begin{array}{l}
\mathsf{APPLY}\ \langle \mathsf{x}_\Theta : \mathsf{x}, \mathsf{y}_\Theta : \mathsf{y}, \mathsf{z}_\Theta : \mathsf{z}, \mathsf{t}_\Theta : \mathsf{t}' \rangle\ \mathsf{pivotsForClawFreeness}(\mathsf{s_0} \mapsto \mathsf{s_0}) \Rightarrow \\
\quad \mathit{Stat4}:\ \{\mathsf{v} \in \mathsf{s_0} \mid \mathsf{y} \in \mathsf{v}\} = \{\mathsf{x}, \mathsf{z}\}\ \&\ \mathsf{z} \in \mathsf{s_0}\ \&\ \mathsf{y} \in \mathsf{z}\ \&\ \mathsf{y} \in \mathsf{x}\ \&\ \mathsf{y}, \mathsf{x} \in \mathsf{s_0}\ \& \\
\qquad \mathsf{y} \notin \bigcup\bigcup \mathsf{s_0}\ \&\ \mathsf{t}' = \{\mathsf{z} \in \mathsf{s_0} \mid \mathsf{y} \notin \mathsf{z}\}\ \&\ \mathsf{Trans}(\mathsf{t}')\ \&\ \mathsf{ClawFree}(\mathsf{t}')\ \& \\
\qquad\quad \mathsf{s_0} \supseteq \mathsf{t}'\ \&\ \mathsf{x} \notin \mathsf{t}'\ \&\ \mathsf{y} \in \mathsf{t}' \setminus \bigcup \mathsf{t}'\ \&\ \mathsf{t}' = \mathsf{s_0} \setminus \{\mathsf{x}, \mathsf{z}\}\ \&\ \mathsf{x} \notin \mathsf{z}\ \&\ \mathsf{z} \notin \mathsf{x}
\end{array}
$$

Moreover, we take t' to be $\mathsf{s_0}$ deprived of the predecessors of y and, if $\mathsf{x} \neq \mathsf{z}$, we take $\mathsf{t} = \mathsf{t}'$ else we take $\mathsf{t} = \mathsf{t}' \setminus \{\mathsf{y}\}$.

$$
\mathsf{Loc\_def} \Rightarrow \quad \mathsf{t} = \textbf{if}\ \mathsf{x} = \mathsf{z}\ \textbf{then}\ \mathsf{t}' \setminus \{\mathsf{y}\}\ \textbf{else}\ \mathsf{t}'\ \textbf{fi}
$$

Thus it turns out readily that t is transitive; hence, by the inductive hypothesis, there is a perfect matching $\mathsf{m_0}$ for t.

$$
\begin{array}{l}
\langle \mathsf{t}', \mathsf{t} \rangle \hookrightarrow T\mathsf{clawFreeness_a}(\mathit{Stat4}\star) \Rightarrow \quad \mathit{Stat5}:\ \mathsf{ClawFree}(\mathsf{t})\ \&\ \mathsf{x} \notin \mathsf{t}\ \&\ \mathsf{s_0} \supseteq \mathsf{t}\ \& \\
\qquad\qquad \mathsf{t}' \supseteq \mathsf{t}\ \&\ \mathsf{t}' = \mathsf{s_0} \setminus \{\mathsf{x}, \mathsf{z}\}\ \&\ \mathsf{y} \in \mathsf{t}' \\
\langle \mathsf{t}', \mathsf{t} \rangle \hookrightarrow T4c(\mathit{Stat4}\star) \Rightarrow \quad \mathsf{Trans}(\mathsf{t}) \\
\langle \mathsf{t} \rangle \hookrightarrow \mathit{Stat1}(\mathit{Stat4}\star) \Rightarrow \mathit{Stat6}:\ \langle \exists \mathsf{m}, \mathsf{y} \mid \mathsf{PerfectMatching}(\mathsf{m})\ \&\ \mathsf{t} \setminus \{\mathsf{y}\} = \bigcup \mathsf{m} \rangle \\
\langle \mathsf{m_0}, \mathsf{y_0} \rangle \hookrightarrow \mathit{Stat6}(\mathit{Stat6}\star) \Rightarrow \quad \mathit{Stat7}:\ \mathsf{PerfectMatching}(\mathsf{m_0})\ \&\ \bigcup \mathsf{m_0} = \mathsf{t} \setminus \{\mathsf{y_0}\}
\end{array}
$$

The possibility that y does not belong to $\bigcup \mathsf{m_0}$ is then discarded; in fact, if this were the case, then by adding the pair $\{\mathsf{x}, \mathsf{y}\}$ to $\mathsf{m_0}$ we would get a perfect matching for $\mathsf{s_0}$, while we have assumed that such a matching does not exist. From the fact $\mathsf{y} \in \bigcup \mathsf{m_0}$ it follows that y belongs to t, hence that $\mathsf{t} = \mathsf{t}'$ and that x, z are distinct.

$$
\begin{array}{l}
\mathsf{Suppose} \Rightarrow \quad \mathit{Stat8}:\ \mathsf{y} \notin \bigcup \mathsf{m_0} \\
\quad \langle \mathsf{m_0}, \mathsf{x}, \mathsf{y} \rangle \hookrightarrow T\mathsf{perfectMatching_3}(\mathit{Stat4}\star) \Rightarrow \\
\qquad\qquad \mathit{Stat9}:\ \mathsf{PerfectMatching}(\mathsf{m_0} \cup \{\{\mathsf{x}, \mathsf{y}\}\}) \\
\quad \mathsf{Loc\_def} \Rightarrow \quad \mathit{Stat10}:\ \mathsf{v} = \textbf{if}\ \mathsf{x} = \mathsf{z}\ \textbf{then}\ \mathsf{y}\ \textbf{else}\ \mathsf{z}\ \textbf{fi} \\
\quad (\mathit{Stat4}\star)\mathsf{ELEM} \Rightarrow \quad \mathit{Stat11}:\ \mathsf{s_0} = \mathsf{t} \cup \{\mathsf{x}, \mathsf{v}\}\ \&\ \{\mathsf{x}\} \cup \{\mathsf{y}\} = \{\mathsf{x}, \mathsf{y}\} \\
\quad (\mathit{Stat5}, \mathit{Stat11}, \mathit{Stat8}, \mathit{Stat7}\star)\mathsf{ELEM} \Rightarrow \quad \mathsf{y} = \mathsf{v} \lor \mathsf{y} = \mathsf{y_0} \\
\quad \langle \mathsf{m_0}, \mathsf{t}, \mathsf{y_0}, \mathsf{s_0}, \{\mathsf{x}\}, \mathsf{v}, \mathsf{y} \rangle \hookrightarrow T31h(\mathit{Stat4}\star) \Rightarrow
\end{array}
$$

$$\mathit{Stat12}:\ \langle \exists d \mid \bigcup(m_0 \cup \{\{x\} \cup \{y\}\}) = s_0 \setminus \{d\}\rangle$$
$$\langle d_0\rangle \hookrightarrow \mathit{Stat12}(\mathit{Stat12}\star) \Rightarrow \quad \bigcup(m_0 \cup \{\{x\} \cup \{y\}\}) = s_0 \setminus \{d_0\}$$
$$\mathsf{EQUAL}(\mathit{Stat11}) \Rightarrow \quad \mathit{Stat13}:\ \bigcup(m_0 \cup \{\{x,y\}\}) = s_0 \setminus \{d_0\}$$
$$\langle m_0 \cup \{\{x,y\}\}, d_0\rangle \hookrightarrow \mathit{Stat2}(\mathit{Stat9}, \mathit{Stat13}\star) \Rightarrow \quad \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow$$
$$\mathit{Stat14}:\ y \in \bigcup m_0$$
$$\mathsf{Use\_def}(\bigcup) \Rightarrow \quad \mathit{Stat15}:\ y \in \{h : p \in m_0, h \in p\}\ \&\ y \in x \cap z\ \&\ x \neq z\ \&$$
$$t = t'\ \&\ x \notin z\ \&\ z \notin x\ \&\ \bigcup m_0 = \{h : p \in m_0, h \in p\}$$

It also follows that y is the tail of that arc p_1 of m_0 to which it belongs. In fact, if y were instead the head of p_1, then the tail x_2 of p_1, which must belong to $\bigcup m_0$ would belong to $s_0 \setminus \{x, z\}$, hence would be inside s_0 but outside the set of predecessors of y, which is absurd.

$$\mathsf{Suppose} \Rightarrow \quad \mathit{Stat16}:\ \neg\langle \exists w \mid \{y,w\} \in m_0\ \&\ w \in y\rangle$$
$$\langle p_1, h_1\rangle \hookrightarrow \mathit{Stat15}(\mathit{Stat16}\star) \Rightarrow \quad p_1 \in m_0\ \&\ y \in p_1$$
$$\mathsf{Use\_def}(\mathsf{PerfectMatching}) \Rightarrow \quad \mathit{Stat17}:\ \langle \forall p \in m_0, \exists x \in p, y \in x, \forall q \in m_0 \mid$$
$$x \in q \vee y \in q \rightarrow \{x,y\} = q\rangle$$
$$\langle p_1, x_2, w_2, p_1\rangle \hookrightarrow \mathit{Stat17}(\mathit{Stat16}\star) \Rightarrow \quad x_2 \in p_1\ \&\ w_2 \in x_2\ \&\ p_1 \in m_0\ \&$$
$$\{x_2, w_2\} = p_1$$
$$\langle w_2\rangle \hookrightarrow \mathit{Stat16}(\mathit{Stat16}\star) \Rightarrow \quad \mathit{Stat18}:\ y \in x_2\ \&\ \{y, x_2\} \in m_0$$
$$\mathsf{Suppose} \Rightarrow \quad \mathit{Stat19}:\ x_2 \notin \{u : v \in m_0, u \in v\}$$
$$\langle \{y, x_2\}, x_2\rangle \hookrightarrow \mathit{Stat19}(\mathit{Stat18}\star) \Rightarrow \quad \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow \quad \textsc{Auto}$$
$$\mathsf{EQUAL}(\mathit{Stat4}) \Rightarrow \quad \mathit{Stat20}:\ \{v \in s_0 \mid y \in v\} = \{x, z\}\ \&\ x_2 \in \bigcup m_0$$
$$(\mathit{Stat7}, \mathit{Stat5}, \mathit{Stat20}\star)\mathsf{ELEM} \Rightarrow \quad \mathit{Stat21}:\ x_2 \notin \{v \in s_0 \mid y \in v\}\ \&\ x_2 \in s_0$$
$$\langle x_2\rangle \hookrightarrow \mathit{Stat21}(\mathit{Stat18}, \mathit{Stat21}\star) \Rightarrow \quad \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow$$
$$\mathit{Stat22}:\ \langle \exists w \mid \{y,w\} \in m_0\ \&\ w \in y\rangle$$

Call w the head of the arc issuing from y in m_0. Then y, x, z, w form a potential claw; this implies, since s_0 is claw-free that either $w \in x$ or $w \in z$.

$$\langle w\rangle \hookrightarrow \mathit{Stat22}(\mathit{Stat22}\star) \Rightarrow \quad \mathit{Stat23}:\ w \in y\ \&\ \{y,w\} \in m_0$$
$$\langle s_0, y\rangle \hookrightarrow T3c(\mathit{Stat2}, \mathit{Stat4}, \mathit{Stat23}, \mathit{Stat4}, \mathit{Stat5}, \mathit{Stat15}\star) \Rightarrow$$
$$\mathit{Stat24}:\ w, y, x, z \in s_0$$
$$(\mathit{Stat2}, \mathit{Stat7}, \mathit{Stat15}\star)\mathsf{ELEM} \Rightarrow$$
$$\mathsf{ClawFree}(s_0)\ \&\ \mathsf{PerfectMatching}(m_0)\ \&\ y \in x \cap z$$
$$\langle s_0, y, x, z, w\rangle \hookrightarrow T\,\mathsf{clawFreeness_b}(\mathit{Stat15}\star) \Rightarrow \quad w \in x \cup z$$

Obviously, $w \in \bigcup m_0$. Moreover, through elementary reasoning we derive $\bigcup m_0 \cup \{z, x\} = s_0 \setminus \{y_1\}$, where y_1 lies outside s_0 if both x and z has been covered by the matching m_0, otherwise y_1 equals the one of x, z (which might be the same set) left over by m_0.

$$\mathsf{Suppose} \Rightarrow \quad \mathit{Stat25}:\ w \notin \{u : v \in m_0, u \in v\}$$
$$\langle \{y,w\}, w\rangle \hookrightarrow \mathit{Stat25}(\mathit{Stat23}, \mathit{Stat23}\star) \Rightarrow \quad \mathsf{false}; \qquad \mathsf{Discharge} \Rightarrow \quad \textsc{Auto}$$
$$(\mathit{Stat7}, \mathit{Stat5}\star)\mathsf{ELEM} \Rightarrow \quad x \notin \bigcup m_0\ \&\ z \notin \bigcup m_0$$

$\mathsf{Loc\_def} \Rightarrow \quad \mathit{Stat26}: \ \mathsf{y}_1 = \textbf{if } \mathsf{y}_0 \in \{\mathsf{x},\mathsf{z}\} \textbf{ then } \mathsf{s}_0 \textbf{ else } \mathsf{y}_0 \textbf{ fi}$
$(\mathit{Stat24}, \mathit{Stat26}\star)\mathsf{ELEM} \Rightarrow \quad \mathsf{s}_0 \setminus \{\mathsf{x},\mathsf{z},\mathsf{y}_0\} \cup \{\mathsf{z},\mathsf{x}\} = \mathsf{s}_0 \setminus \{\mathsf{y}_1\}$
$(\mathit{Stat7}, \mathit{Stat15}, \mathit{Stat5}\star)\mathsf{ELEM} \Rightarrow \quad \bigcup \mathsf{m}_0 \cup \{\mathsf{z},\mathsf{x}\} = \mathsf{s}_0 \setminus \{\mathsf{x},\mathsf{z},\mathsf{y}_0\} \cup \{\mathsf{z},\mathsf{x}\}$
$\mathsf{EQUAL}(\mathit{Stat15}) \Rightarrow \quad \mathit{Stat27}: \ \bigcup \mathsf{m}_0 \cup \{\mathsf{z},\mathsf{x}\} = \mathsf{s}_0 \setminus \{\mathsf{y}_1\} \ \& \ \mathsf{w} \in \bigcup \mathsf{m}_0$

If there is an edge between w and x, then we deviate the perfect matching by replacing $\{\mathsf{y},\mathsf{w}\}$ by $\{\mathsf{y},\mathsf{z}\}$ and $\{\mathsf{x},\mathsf{w}\}$; otherwise we replace $\{\mathsf{y},\mathsf{w}\}$ by $\{\mathsf{y},\mathsf{x}\}$ and $\{\mathsf{z},\mathsf{w}\}$. Plainly we get a perfect matching for s_0 in either case, which leads us to the desired contradiction.

$\mathsf{Suppose} \Rightarrow \quad \mathsf{w} \in \mathsf{x}$
$\langle \mathsf{m}_0, \mathsf{y}, \mathsf{w}, \mathsf{x}, \mathsf{z} \rangle \hookrightarrow T\mathsf{perfectMatching}_4(\mathit{Stat15}\star) \Rightarrow$
$\mathit{Stat28}: \ \mathsf{PerfectMatching}(\mathsf{m}_0 \setminus \{\{\mathsf{y},\mathsf{w}\}\} \cup \{\{\mathsf{y},\mathsf{z}\}, \{\mathsf{x},\mathsf{w}\}\})$
$\langle \mathsf{m}_0, \{\mathsf{y},\mathsf{w}\}, \{\mathsf{y},\mathsf{z}\}, \{\mathsf{x},\mathsf{w}\}, \{\mathsf{z},\mathsf{x}\} \rangle \hookrightarrow T31f(\mathit{Stat15}, \mathit{Stat27}\star) \Rightarrow$
$\bigcup(\mathsf{m}_0 \setminus \{\{\mathsf{y},\mathsf{w}\}\} \cup \{\{\mathsf{y},\mathsf{z}\}, \{\mathsf{x},\mathsf{w}\}\}) = \mathsf{s}_0 \setminus \{\mathsf{y}_1\}$
$\langle \mathsf{m}_0 \setminus \{\{\mathsf{y},\mathsf{w}\}\} \cup \{\{\mathsf{y},\mathsf{z}\}, \{\mathsf{x},\mathsf{w}\}\}, \mathsf{y}_1 \rangle \hookrightarrow \mathit{Stat2}(\mathit{Stat28}\star) \Rightarrow \quad \mathsf{false}$
$\mathsf{Discharge} \Rightarrow \quad \textsc{Auto}$
$\langle \mathsf{m}_0, \mathsf{y}, \mathsf{w}, \mathsf{z}, \mathsf{x} \rangle \hookrightarrow T\mathsf{perfectMatching}_4(\mathit{Stat15}\star) \Rightarrow$
$\mathit{Stat29}: \ \mathsf{PerfectMatching}(\mathsf{m}_0 \setminus \{\{\mathsf{y},\mathsf{w}\}\} \cup \{\{\mathsf{y},\mathsf{x}\}, \{\mathsf{z},\mathsf{w}\}\})$
$\langle \mathsf{m}_0, \{\mathsf{y},\mathsf{w}\}, \{\mathsf{y},\mathsf{x}\}, \{\mathsf{z},\mathsf{w}\}, \{\mathsf{z},\mathsf{x}\} \rangle \hookrightarrow T31f(\mathit{Stat15}, \mathit{Stat27}\star) \Rightarrow$
$\bigcup(\mathsf{m}_0 \setminus \{\{\mathsf{y},\mathsf{w}\}\} \cup \{\{\mathsf{y},\mathsf{x}\}, \{\mathsf{z},\mathsf{w}\}\}) = \mathsf{s}_0 \setminus \{\mathsf{y}_1\}$
$\langle \mathsf{m}_0 \setminus \{\{\mathsf{y},\mathsf{w}\}\} \cup \{\{\mathsf{y},\mathsf{x}\}, \{\mathsf{z},\mathsf{w}\}\}, \mathsf{y}_1 \rangle \hookrightarrow \mathit{Stat2}(\mathit{Stat29}\star) \Rightarrow \quad \mathsf{false};$
$\mathsf{Discharge} \Rightarrow \quad \textsc{Qed}$

A.10 From Membership Digraphs to General Graphs

Let us now place the results presented so far under the more general perspective that motivates this work. We display in this section the interfaces of two THEORYs (not developed formally with Ref, as of today), explaining why we can work with membership as a convenient surrogate for the edge relationship of general graphs.

One of these, THEORY finGraphRepr, will implement the proof that any finite graph $(\mathsf{v}_0, \mathsf{e}_0)$ is 'isomorphic', via a suitable orientation of its edges and an injection ϱ_Θ of v_0 onto a set ν_Θ, to a digraph $(\nu_\Theta, \{(x,y) : x \in \nu_\Theta, \ y \in x \cap \nu_\Theta\})$ that enjoys the *weak extensionality* property: "distinct non-sink vertices have different out-neighborhoods".

Although accessory, the weak extensionality condition is the clue for getting the desired isomorphism; in fact, for any weakly extensional digraph, acyclicity always ensures that a variant of *Mostowski's collapse* is well-defined: in order to get it, one starts by assigning a distinct set Mt to each sink t and then proceeds by putting recursively

$$Mw = \{ Mu : (w,u) \text{ is an arc} \}$$

for all non-sink vertices w; plainly, injectivity of the function $u \mapsto Mu$ can be ensured globally by a suitable choice of the images Mt of the sinks t.

DISPLAY finGraphRepr

THEORY finGraphRepr(v_0, e_0)
  $\mathsf{Finite}(v_0)$ & $e_0 \subseteq \{\{x, y\} : x, y \in v_0 \mid x \neq y\}$
$\Rightarrow$ $(\varrho_\Theta,\ \nu_\Theta)$
  $1\text{--}1(\varrho_\Theta)$ & $\mathbf{domain}(\varrho_\Theta) = v_0$ & $\mathbf{range}(\varrho_\Theta) = \nu_\Theta$
  $\langle \forall x \in v_0, y \in v_0 \mid \{x, y\} \in e_0 \leftrightarrow \varrho_\Theta \upharpoonright x \in \varrho_\Theta \upharpoonright y \vee \varrho_\Theta \upharpoonright y \in \varrho_\Theta \upharpoonright x \rangle$
  $\{x \in \nu_\Theta \mid x \cap \nu_\Theta \neq \emptyset\} \subseteq \mathscr{P}(\nu_\Theta)$
END finGraphRepr

The other THEORY, cfGraphRepr, will specialize finGraphRepr to the case of a *connected, claw-free* (undirected, finite) graph—connectedness and claw-freeness are specified, respectively, by the second and by the third assumption of this THEORY. For these graphs, we can insist that the orientation be so imposed as to ensure *extensionality* in full: "distinct vertices have different out-neighborhoods". Consequently, the following will hold:

- there is a unique sink, ∅; moreover,
- the set ν_Θ of vertices underlying the image digraph is transitive. And, trivially,
- ν_Θ is a claw-free set, in an even stronger sense than the definition with which we have been working throughout this proof scenario.

(As regards the third of these points, it should be clear that none of the four non-isomorphic membership renderings of a claw are induced by any quadruple of elements of ν_Θ; our definition forbade only two of them, though!)

DISPLAY cfGraphRepr

THEORY cfGraphRepr(v_0, e_0)
  $\mathsf{Finite}(v_0)$ & $e_0 \subseteq \{\{x, y\} : x, y \in v_0 \mid x \neq y\}$
  $\langle \forall x \in v_0, y \in v_0 \mid x \neq y$ & $\{x, y\} \notin e_0 \rightarrow \langle \exists p \subseteq e_0 \mid \mathsf{Cycle}(p \cup \{\{y, x\}\})\rangle\rangle$
  $\langle \forall w \in v_0, x \in v_0, y \in v_0, z \in v_0 \mid \{w, y\}, \{y, x\}, \{y, z\} \in e_0 \rightarrow$
    $x = z \vee w \in \{z, x\} \vee \{x, z\} \in e_0 \vee \{z, w\} \in e_0 \vee \{w, x\} \in e_0\rangle$
$\Rightarrow$ $(\rho_\Theta, \nu_\Theta)$
  $1\text{--}1(\rho_\Theta)$ & $\mathbf{domain}(\rho_\Theta) = v_0$ & $\mathbf{range}(\rho_\Theta) = \nu_\Theta$
  $\langle \forall x \in v_0, y \in v_0 \mid \{x, y\} \in e_0 \leftrightarrow \rho_\Theta \upharpoonright x \in \rho_\Theta \upharpoonright y \vee \rho_\Theta \upharpoonright y \in \rho_\Theta \upharpoonright x \rangle$
  $\mathsf{Trans}(\nu_\Theta)$ & $\mathsf{ClawFree}(\nu_\Theta)$
END cfGraphRepr

Via the THEORY cfGraphRepr, the above-proved existence results about perfect matchings and Hamiltonian cycles can be transferred from a realm of special membership digraphs to the *a priori* more general realm of the connected claw-free graphs.

References[7]

1. Ferro, A., Omodeo, E.G., Schwartz, J.T.: Decision procedures for some fragments of set theory. In: Bibel, W., Kowalski, R. (eds.) Proc. of the 5th Conference on Automated Deduction. LNCS, vol. 87, pp. 88–96, Les Arcs, France. Springer, Berlin (1980).
2. Ferro, A., Omodeo, E.G., Schwartz, J.T.: Decision procedures for elementary sublanguages of set theory. I: Multi-level syllogistic and some extensions. Commun. Pure Appl. Math. **XXXIII**, 599–608 (1980)
3. Omodeo, E.G., Schwartz, J.T.: A 'Theory' mechanism for a proof-verifier based on first-order set theory. In: Kakas, A., Sadri, F. (eds.) Computational Logic: Logic Programming and Beyond—Essays in Honour of Bob Kowalski, Part II, vol. 2408, pp. 214–230. Springer, Berlin (2002)
4. Omodeo, E.G., Cantone, D., Policriti, A., Schwartz, J.T.: A computerized Referee. In: Schaerf, M., Stock, O. (eds.) Reasoning, Action and Interaction in AI Theories and Systems—Essays Dedicated to Luigia Carlucci Aiello. Lecture Notes in Artificial Intelligence, vol. 4155, pp. 117–139. Springer, Berlin (2006)
5. D'Agostino, G., Omodeo, E.G., Schwartz, J.T., Tomescu, A.I.: Self-applied proof verification. (Extended abstract). In: Cordón-Franco, A., Fernández-Margarit, A., Félix Lara-Martin, F. (eds.) JAF, 26èmes Journées sur les Arithmétiques Faibles, Sevilla, 11–13 June 2007, pp. 113–117. Fénix Editora, Sevilla (2007). See http://www.cs.us.es/glm/jaf26
6. Schwartz, J.T., Cantone, D., Omodeo, E.G.: Computational Logic and Set Theory. Applying Formalized Logic to Analysis. Springer, Berlin (2011). Foreword by Martin Davis
7. Schwartz, J.T., Dewar, R.K.B., Dubinsky, E., Schonberg, E.: Programming with Sets: An Introduction to SETL. Texts and Monographs in Computer Science. Springer, Berlin (1986)
8. Landau, E.: Foundations of Analysis. The Arithmetic of Whole, Rational, Irrational and Complex Numbers, 3rd edn. Chelsea Publishing, New York (1966)
9. Omodeo, E.G., Tomescu, A.I.: Using ÆtnaNova to formally prove that the Davis-Putnam satisfiability test is correct. Matematiche **63**, 85–105 (2008). A preliminary version was presented at, CILC07 (Messina)
10. Cantone, D., Chiaruttini, C., Nicolosi Asmundo, M., Omodeo, E.G.: Cumulative hierarchies and computability over universes of sets. Matematiche **63**, 31–84 (2008)
11. Matthews, M.M., Sumner, D.P.: Hamiltonian results in $K_{1,3}$-free graphs. J. Graph Theory **8**, 139–146 (1984)
12. Milanič, M., Tomescu, A.I.: Set graphs. I. Hereditarily finite sets and extensional acyclic orientations. Discrete Appl. Math. (2012). doi:1016/j.dam.2011.11.027
13. Schwartz, J.T., Cantone, D., Omodeo, E.G.: Computational Logic and Set Theory. Springer, Berlin (2011). Foreword by Martin Davis
14. Sumner, D.P.: Graphs with 1-factors. Proc. Am. Math. Soc. **42**, 8–12 (1974)
15. Las Vergnas, M.: A note on matchings in graphs. Cah. Cent. Étud. Rech. Opér. **17**, 257–260 (1975)

[7]The references that follows aim at covering in full, in chronological order, Jack's publications that refer to computational logic and to proof-verification. I have received substantial help in this reconstruction from Domenico Cantone and from Alfredo Ferro. The few bibliographic items not coauthored by Jack, placed at the end, have to do with the Ref system.

Computational Approaches to RNAi and Gene Silencing

Alessandro Laganà, Rosalba Giugno, Alfredo Pulvirenti, and Alfredo Ferro

Abstract The discovery of small regulatory RNAs in the past few years has deeply changed the RNA molecular biology, revealing more complex pathways involved in the regulation of gene expression and in the defense of the genome against exogenous nucleic acids. These small RNA molecules play a crucial role in many physiological processes. Aberrations in their sequences and expression patterns are often related to the development of malignant diseases. The underlying biological mechanisms are known as gene silencing and RNA interference (RNAi). This discovery not only changes our conception of gene expression regulation but, at the same time, opens new frontiers for the development of therapeutic approaches, more specific and less toxic, especially against all those diseases which are still resistant to traditional treatment. Computational techniques constitute an essential tool in the study of these complex systems. Many existing bioinformatics methods and newly developed approaches have been used to analyze and classify RNAi data and more sophisticated tools are needed to allow a better understanding of the small RNAs

---

Foreword: This essay is dedicated to our unforgivable mentor and friend Jack Schwartz. His contribution to the birth and development of the Computer Science group in Catania will never be forgotten. For this reason we decided to entitle the well established Lipari School as "Jacob T. Schwartz International School for Scientific Research" (http://lipari.dmi.unict.it). Jack stimulated our interest in computational biology and bioinformatics, which is now the main subject of our research. In what follows we summarize some of the topics that were subjects of many stimulating discussions with Jack.

A. Laganà (✉) · R. Giugno · A. Pulvirenti · A. Ferro
Dept. of Mathematics and Computer Sciences, University of Catania, Catania, Italy
e-mail: lagana@dmi.unict.it

R. Giugno
e-mail: giugno@dmi.unict.it

A. Pulvirenti
e-mail: apulvirenti@dmi.unict.it

A. Ferro
e-mail: ferro@dmi.unict.it

A. Laganà · A. Ferro
Dept. of Biomedical Sciences, University of Catania, Catania, Italy

M. Davis, E. Schonberg (eds.), *From Linear Operators to Computational Biology*,
DOI 10.1007/978-1-4471-4282-9_9,

biogenesis, processing and functions. In this essay we will review the basics of gene expression regulation by small RNA molecules and discuss the main computational issues in RNAi research, focusing on the most popular algorithms and addressing the open challenges.

1 Introduction

In the last decade, a revolution has occurred in the field of RNA biology, with the discovery of the small non-coding RNAs (ncRNAs) involved in the regulation of gene expression and in the defense of the genome against exogenous nucleic acids [1]. This regulation can occur at different levels of the gene expression process, including transcription, mRNA processing and translation. The corresponding mechanisms are known as Gene Silencing and RNA interference (RNAi). These small RNA molecules play a specific role in guiding effector protein complexes towards their nucleic acids targets by partial or full complementarity bounds [2].

Although several classes of regulatory ncRNAs have been identified, they can be classified into three categories, based on their origin, structure, associated effector complexes and function: short interfering RNA (siRNA), microRNA (miRNA) and piwi interacting RNA (piRNA) [3]. These molecules seem to be present only in eukaryotes and in some viruses. siRNAs and miRNAs are the most abundant regulatory molecules in terms of both phylogeny and physiology and are characterized by double strand precursors. Conversely, piRNAs are mainly present in animals, exert their functions in germ lines and derive from precursor about which very little is known but for which single stranded nature has been hypothesized.

Many computational tools have been developed in the last years for the analysis and the classification of the RNAi related data. Machine learning, probabilistic models and heuristic approaches have been successfully applied for the identification of miRNA genes, prediction of miRNA and siRNA targets and design of synthetic regulatory RNAs. In this essay we will review the basics of gene expression regulation by miRNAs and siRNAs and discuss the main computational issues in RNAi research. We will give an overview of the most successful approaches and briefly describe the most popular tools, highlighting the significant results and open challenges.

2 Gene Silencing and RNA Interference

The first miRNA, the *C. elegans* lin-4, was discovered by Ambros et al. in 1993, as endogenous regulator of genes controlling the developmental timing [4]. Five years later, Fire, Mello et al., reported the capability of exogenous double stranded RNAs to silence genes in a specific manner, giving rise to the RNAi mechanism [5]. In 1999, a similar process was found to take place in plants, based on short RNA sequences (~20–25 nt), able to bind their target through perfect base complementarity [6].

In 2001 the two classes of regulatory ncRNAs were characterized: miRNAs as regulators of endogenous genes, and siRNAs as defenders of the genome integrity against exogenous nucleic acids such as transposons and viruses [4]. In 2004, single stranded forms of miRNAs and siRNAs were found associated to effector protein complexes known as RNA-induced silencing complexes (RISC) [7]. In both cases, the regulated genes are specified by the small RNA molecules, which recognize their targets through base complementarity mechanism.

miRNAs and siRNAs differ in terms of their origin. miRNAs are endogenous genome products, derived from partially complementary double strand hairpin precursors, while siRNAs are mainly exogenous molecules coming from viruses and transposons and obtained by perfectly complementary long double stranded precursors (dsRNA) [6]. Nevertheless, their common features, such as the length of their mature products and their sequence-specific inhibitory functions, suggest similar biogenesis and common mechanisms. Both classes of RNAs indeed depend on the same two protein families: the Dicer and the Ago proteins [7]. In the following subsections the main biological features of siRNAs and miRNAs and the implications of miRNAs in physiological and pathological pathways will be briefly discussed.

2.1 siRNA

siRNAs are RNA molecules which are usually 20–24 nucleotides long. Their role is the defense of the cell against exogenous nucleic acids, and the maintenance of genome integrity through the silencing of undesired transcripts (like transposones and repetitive elements) [2, 3].

siRNA precursors are linear long dsRNAs, processed by an enzyme called Dicer, which cleaves them into smaller double strand molecules. Dicer, together with the TRBP and Ago2 proteins, form the RISC-loading complex. Then, Ago2 cleaves one of the two strands of the siRNA, the passenger strand, generating the functional RISC [8–12]. The selection of the guide strand depends on the relative thermodynamic stabilities of the two duplex ends. The strand whose $5'$ end is less stably paired is usually recognized as the guide strand [6]. This directs the RISC towards perfectly complementary target mRNAs, causing their degradation. In particular, the PIWI domain in the Ago protein is responsible of the cleavage, which occurs between the nucleotides paired to the siRNA bases 10 and 11. This is a very precise process and pairing mismatches in the siRNA/target duplex can suppress the cleavage. However, partially complementary targets can still be translationally repressed by the RISC, as in the case of miRNA mediated silencing [6].

Endogenous siRNAs have been found only in plants and in some simple animal species. Trans-acting short interfering RNAs (ta-siRNA) are present in plants, in which they bind their targets with partial complementarity, leading to their translational repression [13, 14]. Repeat-associated short interfering RNAs (rasi-RNA) are derived from dsRNAs presumably originated from transposable elements. They play a fundamental role in the gametogenesis in flies and worms, through chromatin modelling and the silencing of viral transcripts [15, 16]. Scan RNAs (scnRNA) are relatively longer siRNAs, found in the protozoa Tetrahymena thermophila and involved in the control of genomic rearrangements [17]. Finally, Long siRNAs (lsiRNA) are longer siRNAs found in plants and induced in response to bacterial infections. It has been demonstrated the role of these siRNAs in conferring resistance to bacteria, through the silencing of a gene involved in the defense regulation [18].

2.2 miRNA

microRNAs (miRNAs) are small single stranded regulatory RNAs (20–22 nt long), able to modulate gene expression through the degradation or the translational repression of specific target molecules [4]. It has been estimated that miRNA coding genes represent the 1 % of the total gene population, being the biggest class of regulatory molecules.

miRNAs are present in plants, in higher eukaryotes and in some viruses. They are encoded by miRNA genes, which are usually located in the introns of protein coding genes, in intergenic regions or, more rarely, in the exons of coding genes. miRNAs are usually transcribed by polymerase II and are often encoded in clusters [4]. Primary transcripts (pri-miRNA) are single stem-loops or multi-hairpin structures, in the case of clustered miRNAs. The first step of miRNA biogenesis occurs in the nucleus, and consists of the excision of the stem-loop from the longer transcript, in order to obtain the pre-miRNA. This cleavage reaction is performed by Dcl1 in plants and by Drosha in animals [19].

The second step is the excision of the loop from the stem, in order to create the mature miRNA duplex, usually ~22 nt long. In plants, this reaction is carried out by Dcl1 in the nucleus, while in animals the pre-miRNA is exported to the cytoplasm and then processed by Dicer [4, 19].

As in siRNA processing, one of the two strands of the mature miRNA duplex is incorporated into the RISC complex, although the passenger strand is sometimes found associated to Ago proteins as well. The most commonly associated strand is called the mature miRNA, while the other one is called miRNA* [20].

miRNAs act as adaptors of RISC complexes to specific mRNA targets. miRNA binding sites in animals are usually located in the 3′ UTR sequences and are often present in multiple copies. Most animal miRNAs bind their target with partial complementarity, allowing bulges and loops in the duplexes. However, a key feature in target recognition is the perfect pairing of the nucleotides 2–8 of the miRNA, which is called the *seed* region (see Fig. 1). Conversely, most miRNAs in plants bind their targets in their coding regions with perfect complementarity.

The kind of binding of miRNAs to their targets is considered a key factor in the regulatory mechanism. The presence of mismatches in the central part of the duplex is usually associated to translational repression, which seems to be the default mechanism of miRNA mediated silencing. The cleavage of perfectly paired duplexes is considered an additional feature leading to the same effect on the protein level.

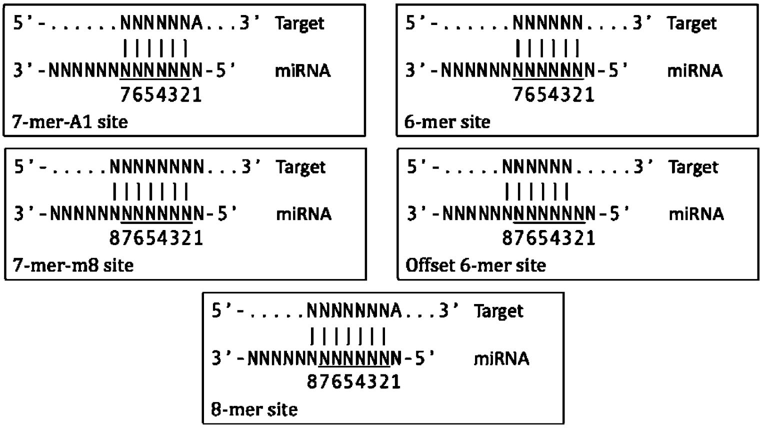


Fig. 1 Types of miRNA canonical target sites. *Seed* is underlined

2.3 miRNAs in Development and Disease

The biological functions of miRNAs are currently being intensively investigated and the involvement of miRNAs in fundamental processes, such as apoptosis, metabolism and cell proliferation, has been demonstrated. miRNAs play also an essential role in development [21]. Dicer knock-out mouse models provided significant evidence for the specific role of miRNAs in the morphogenesis of several organs, including lungs, limbs and muscles, and in the differentiation of T cells [22–26]. miRNAs can also act as switches of regulatory pathways, as in the control of alternative splicing contributing to tissue-specificity. For example, the muscle-specific miR-133 is able to silence a protein involved in alternative splicing during the differentiation of myoblasts, in order to control the splicing of certain exons combinations [27].

Although the miRNA regulatory mechanisms have not been yet completely elucidated, their importance in the normal development of many organs and their impact on many pathologies, including cancer, is evident.

The altered expression of miRNAs is often associated to the development of diseases [28]. Recent evidence suggests a potential involvement of miRNAs in neurodegeneration. A significant under-expression of miR-107 in Alzheimer's disease patients have been associated to higher levels of the BACE1 protein, which is responsible of the cleavage of the myeloid precursor protein and the consequent release of neurotoxic amyloid-beta peptides. This dysregulation could be one involved in the pathogenesis of Alzheimer's disease [29].

miRNAs are also involved in primary muscle disorders, including muscular dystrophy and congenital inflammatory myopathies. The frequent over-expression of 5 miRNAs (miR-146b, miR-221, miR-155, miR-214 and miR-222) observed in Duchenne muscular dystrophy, Miyoshi myopathy and dermatomyositis patients, suggests their possible involvement in a common regulatory pathway [30].

miRNAs also play a crucial role in cardiac pathologies. It has been recently described the correlation between miR-133, which regulates the proteins RhoA and Nelf-A/WHSC2, and cardiomyocyte hypertrophy [31]. miR-1 is over-expressed in coronary disease patients and this alteration is associated to arrhythmias through the silencing of the genes GJA1 and KCNJ2 [32]. The knock-out of miR-1 can inhibit ischemic arrhythmias, suggesting a possible therapeutic application.

There is now a wide literature demonstrating the crucial role of miRNAs in the cancer pathogenesis [33]. The first evidence of the involvement of miRNAs in cancer is the observed under-expression or deletion of miR-15 and miR-16 in a significant number of chronic lymphocytic leukemia (CLL) patients [34]. Following studies reveal the differential expression of miRNAs between normal and tumour tissues, but also between primary and metastatic tumours. These differences are often tumour-specific and sometimes have prognostic value. Evidence suggests a role of miRNAs both as oncogenes and tumour suppressors.

The let-7 family contains miRNAs able to regulate the activity of the oncogene Ras, through post-transcriptional repression [35]. A recent study shows that the expression of let-7g in lung cancer cells expressing K-Ras in mouse models, induces the arrest of cell cycle and cell death, thus revealing the therapeutic potential of let-7 family as oncosuppressors [36].

Another study reports the capability of a miRNA to induce a neoplastic disease. The over-expression of miR-155 in mice B-Cells, indeed, induces a preleukemic proliferation, followed by the malignant disease [37].

It has also been demonstrated the role of miRNAs in the development of metastasis. miR-10b is found to be highly expressed in the metastatic cells of breast cancer. Such over-expression, induced by the transcription factor Twist, starts the invasion and the metastatic process through the inhibition of the translation of the homeobox D10 gene, followed by the related over-expression of the pro-metastatic gene RHOC [38].

All these studies demonstrate the potential regulatory functions of miRNAs in the diverse cellular and tissue types. miRNAs could be involved in the pathophysiology of several human diseases, through the modulation of pathways involving hundreds or thousands of different genes. Moreover, a single miRNA could affect several pathological pathways, due to its different targets.

miRNA targets comprise genes involved in development and transformation, such as transcription factors and cell cycle control proteins. Diseases could be the result of the perturbation of these pathways due to the mutation of miRNA genes and their binding sites on targets or in the pathways regulating their expression.

The numerous computational tools designed for the study of RNAi are essential for the understanding of its basic mechanisms and the effect on the phenotype. The computational prediction of miRNA and siRNA genes and their targets, together with Data Mining analysis constitutes the fundamental basis of a promising research in the field of regulatory RNAs. The aim is to uncover significant correlations between regulatory RNAs, their targets and the physio-pathological processes in which they are involved. In the following sections, the main computational issues of RNAi will be described, with particular emphasis on the open questions and challenges still awaiting efficient solutions.

3 miRNA Genes Prediction

miRNAs are present in several eukaryotes and in some viruses and, to date, over 700 miRNAs have been identified in human [39]. As discussed in the previous section, miRNAs are encoded by particular non coding genes which are transcribed into hairpins.

Most eukaryotic genomes contain very high numbers of inverted repeats that, when transcribed, can form hairpins. It has been estimated that the human genome can encode about 11 million hairpins [40], thus the challenge is the selection of the right hairpins, that is, the true miRNA genes. The search for hairpins is the first step in miRNA genes prediction, common to many computational approaches. The next step is the evaluation of the "miRNA-ness" of the hairpins, based on the empirical rules inferred from the validated miRNA genes and on the thermodynamics of the candidate molecules.

Pre-miRNAs have low folding free energy, which is dependant on the sequence length [41]. The length of pre-miRNAs can be variable, sometimes depending on the organism [42]. For example, a typical human pre-miRNA is about 100 nt long, while miRNA precursors length in plants can range from 60 to 400 nucleotides [43]. Thus, the length of the sequence must be taken into account when evaluating the folding free energy. Recent studies showed that the average adjusted minimal folding free energy of a miRNA, obtained as a combination of folding free energy and miRNA length, is significantly lower than other RNAs, including tRNAs, rRNAs and mRNAs and thus can help discriminating between miRNA precursors and other RNAs [44].

Evolutionary conservation is another important feature of miRNAs. Mature miRNAs are usually more conserved than their precursors across the evolution but many non-conserved miRNAs have also been identified in several species [4, 45]. Nonetheless, conservation is one of the most used criteria for the identification of pre-miRNAs [46].

Many prediction algorithms make use of BLAST and other similar tools in order to perform homology searches in DNA or Expressed Sequence Tags (EST) databases against known experimentally validated miRNAs. By using this approach many miRNAs have been identified in human, mouse and plants genomes. miR-Align is a tool which uses a combination of homology search and secondary structure evaluation and it was used to identify 59 miRNA genes in Anopheles gambiae [47]. Also based on homology search, the tool microHarvester allowed the identification of novel miRNA genes in plants [48]. Sequence and structural features are integrated in a Hidden Markov Model (HMM) based tool, ProMir, in which the miRNA stem-loop is modelled as a paired sequence [49]. Candidate stem-loops are filtered using various structural criteria concerning stem length, loop size, Minimum Free Energy (MFE) and other features like the pattern of base-pairing and the location of the mature miRNA in the precursor. ProMir was used to identify 9 novel miRNAs in human.

The tool miRseeker performs an analysis of the genome for conserved sequences with stem-loop secondary structure and uses sequence features inferred from known miRNAs as a posterior filter [50]. The tool was able to identify 75 % of previously reported miRNAs in Drosophila.

A similar filter-based approach is used in miRScan, a tool which tries to identify miRNAs based on common characteristics of previously known miRNAs, like the base pairing rules [51]. miRScan initially identifies hairpins in the genome of *C. elegans* and looks for potential homologues in the genome of *C. briggsae*. Then, the aligned segments from the two different genomes are scored based on several features such as the amount of base pairing to the proposed mature miRNA, the amount of base pairing in the stem excluding the mature miRNA, the conservation in the 5′ and 3′ ends of the two aligned sequences, the bulge symmetry, the distance from mature miRNA to the loop of the hairpin and the presence of specific bases at the first five positions of the candidate mature miRNA. By using this method, the authors were able to identify 50 of the 53 miRNAs known at the time to be conserved in both species.

The tool MirFinder is a tool for finding miRNAs in plants, based on a comparative genomic approach. It finds miRNAs through the identification of conserved hairpin structures in the genomes of A. thaliana and Oryza sativa and the application of several filters, based on core features derived from known miRNAs [52].

All of these tools make extensive use of conservation criteria and are therefore unable to identify non conserved or poorly conserved miRNAs.

A different kind of approach, based on intragenomic matching, is built on the idea that a functional miRNA should have at least one target. The tool miMatcher tries to simultaneously predicts miRNAs and their targets [53]. Given one or more potential target mRNAs, it finds all the complementary matches between these targets and the genome. All the matches are candidate mature miRNA sequences, which are then filtered by assessing their "miRNAness" through structural and sequence analysis.

All the described tools were able to identify new or already reported miRNAs in various species and, although they differ in the implemented models, they are mostly based on conservation and on sequence and structural features inferred from known miRNAs. Table 1 summarizes the main features of the presented tools.

The combination of deep sequencing and computational methods is the key for the identification of new miRNAs and other non coding RNAs. The use of Data Mining based approaches could help to identify new sequence and structure features of the miRNA genes and their surroundings (e.g. promoters and protein binding sites). This will allow a better understanding of miRNA biogenesis and processing as well as the prediction of new miRNA genes.

4 Prediction of miRNA Targets

The fundamental step in determining miRNA functions is to find their targets. The computational prediction of miRNA targets in plants is easier than in animals due to the perfect complementarity that plant miRNAs usually exhibit to their

Table 1 Overview of the described tools for miRNA genes prediction

Name	Approach	Species	Link
MiR-Align	Homology search and secondary structure evaluation	Animals and plants	http://bioinfo.au.tsinghua.edu.cn/miralign/
ProMir	Pre-miRNA features (probabilistic-hidden Markov model)	Animals and plants	http://cbit.snu.ac.kr/~ProMiR2/index.php
MiRseeker	Conservation and Pre-miRNA features	Drosophila	–
microHarvester	Conservation and Pre-miRNA features	Plants	http://www-ab.informatik.uni-tuebingen.de/brisbane/tb/index.php?view=microharvester
MirScan	Conservation and Pre-miRNA features	Animals	http://genes.mit.edu/mirscan/
MiRFinder	Comparative genomics	Plants	http://www.bioinformatics.org/mirfinder/
miMatcher	Intra-genomic matching	Plants	http://wiki.binf.ku.dk/MiRNA/miMatcher2

targets. In animals, the perfect complementarity is usually limited to the 5′ end of the miRNA, which is usually referred to as the seed (∼6–9 nt long) [4]. The target sites are usually located in the 3′ UTR sequences of mRNAs. The short length of the miRNA seeds raises the probability of finding random matches that don't correspond to functional sites, thus other determinants are needed, in order to significantly reduce the number of false positives [54]. Such rules should be primarily inferred from experimentally verified targets, therefore having good sources of data is the basic step in the development of prediction tools. A significant amount of miRNA/target interactions data, usually coming from the literature, is publicly available on web databases, such as Tarbase [55] and miRecords [56]. These data usually provide information on the binding sites of miRNAs in their verified targets, but the details of the paired bases are generally computationally predicted. Recently, the high-throughput sequencing of RNAs isolated by crosslinking immunoprecipitation (HITS-CLIP) has identified functional RISC interaction sites on mRNAs, allowing the creation of a library of reliable miRNA binding sites [57]. The analysis of these sequences through Data Mining techniques could help to identify important discriminant features for the prediction of new binding sites.

The miRNA/target interaction rules are generally not sufficient to predict functional targets, due to the high number of false positives deriving from random matches of the short seed region of miRNAs to false targets. Thus, other kind of data is used in order to improve the prediction algorithms. A widely used criterion is target conservation. The alignment of miRNAs in different species, indeed, reveals high sequence conservation, especially in the seed regions, which often corresponds to high conservation of their targets. Thus, the identification of conserved regions in the 3′ UTR of a gene may help to detect functional sites, although this approach is not useful in the case of non-conserved miRNAs [58].

Many prediction methods make use of thermodynamics properties. The free energy ΔG can be used to evaluate the stability of the predicted duplexes. All the validated miRNA/target pairs are indeed characterized by low values of free energy, usually below -20 Kcal/mol [59]. However, low energy value is a necessary but not sufficient condition, since not all the energetically favourable miRNA/target duplexes are functional. Another thermodynamic feature used by computational methods is the structural accessibility of the target molecule. miRNA binding sites shouldn't be involved in any intra-molecular base pairing, and any existing secondary structure should be disrupted in order to make the site accessible to the miRNA [60]. This very complex problem mostly relies on secondary structure prediction computation, which is still one of the challenges of computational biology [61].

Other features used by prediction tools include the nucleotide composition surrounding the binding sites and the position of the sites in the UTR, as well as the presence of multiple sites on the same UTR. It is known, indeed, that a single miRNA can have more binding sites on the same target and that a target can have multiple sites for different miRNAs [62].

4.1 Tools for the Prediction of miRNA Targets

Many computational tools for the prediction of miRNA targets are currently available on the web [63]. In this subsection we will review the basic ideas behind the most popular ones, which are TargetScan, miRanda, Pictar, Diana-microT, RNA22, RNAHybrid, StarMir and PITA (Table 2).

One of the most popular tools for miRNA targets prediction is TargetScan, a sophisticated algorithm based on both conservation and base pairing rules [58, 62]. TargetScan searches for miRNA seed matches on UTRs, considering different kinds of seed (see Fig. 1). It also makes use secondary structure prediction in order to calculate the free energy of the predicted duplexes. The presence of multiple sites for the same miRNA on a target contributes positively to the score of the prediction. TargetScan also takes into account the conservation on different species, computed through sequence alignment, for the identification of the most probable targets. All the predictions, computed for different species like human, mouse and rat, are available on the TargetScan website.

The tool miRanda gives its predictions on human, mouse and rat on a website as well [64, 65]. It is based on an alignment algorithm which uses a weighted matrix aimed at promoting the binding of the seed of the miRNA rather than its 3′ end. It also uses the free energy of predicted duplexes and the conservation criteria to select the most probable targets.

Table 2 Overview of the described tools for miRNA target prediction

Name	Approach	Species	Link
miRanda	Conservation and empirical rules	Human, mouse and rat	http://www.microrna.org
TargetScan	Conservation and empirical rules	Human, mouse, worm and fly	http://www.targetscan.org/
PicTar	Conservation and empirical rules	Vertebrates, flies and nematodes	http://pictar.mdc-berlin.de/
Diana-MicroT	Empirical rules	Human	http://diana.cslab.ece.ntua.gr/microT/
RNA22	Pattern based	Human, mouse, worm and fly	http://cbcsrv.watson.ibm.com/rna22.html
RNAHybrid	Thermodynamics and empirical rules	Animals	http://bibiserv.techfak.uni-bielefeld.de/rnahybrid/
StarMir	Thermodynamics (structural accessibility)	Animals	http://sfold.wadsworth.org/starmir.pl
PITA	Thermodynamics (structural accessibility)	Animals	http://genie.weizmann.ac.il/pubs/mir07/

PicTar is another popular tool for the prediction of miRNA targets on vertebrates, nematodes and flies [66]. The algorithm is trained to identify binding sites for a single miRNA and multiple sites regulated by different miRNAs acting cooperatively. It makes use of a pairwise alignment algorithm in order to find sites conserved in many species (7 Drosophila species and 8 vertebrate species). It also considers the clustering and co-expression of miRNAs together with ontological information, such as the time and tissue specificity of miRNAs and their potential targets, to enhance its predictions.

The algorithm of Diana-MicroT is trained to identify targets with a single binding site for a miRNA [67]. It is based on a sequence alignment algorithm which focuses on the search for miRNA/target duplexes characterized by central bulges and paired $5'$ and $3'$ ends.

The RNA22 tool uses a different approach, based on the analysis of miRNA sequences to find intra- and inter-species patterns of conserved sequence features [68]. The algorithm generates the reverse complement of the most significant patterns and search for their instances in the UTRs. Then, the target islands supported by a minimum number of pattern hits are identified. A target island is defined as any hot spot where the reverse complement of mature miRNA patterns aggregate. The pairing of each target island with each candidate miRNA is then computed and the thermodynamic stability of the duplex is evaluated.

RNAHybrid is a miRNA target prediction tool conceived as an extension of the RNA secondary structure prediction algorithm by Zuker and Stiegler to two sequences [59]. The miRNA is hybridized to the target in an energetically optimal way, i.e. yielding the Minimum Free Energy (MFE). Intra-molecular base pairing and multi-loops are forbidden. The MFE hybridization of the miRNA and its candidate target is computed by dynamic programming, forcing the perfect match of the seed. Bulges and internal loops are restricted to a constant maximum length in either sequence.

The main feature of the tools StarMir and PITA is instead the computation of the structural accessibility of the targets. StarMir is based on the secondary structure of the target predicted by the tool Sfold [69]. The miRNA/target interaction is modelled as a two-step hybridization reaction: the nucleation at an accessible site and the hybrid elongation to disrupt local target secondary structure and form the complete duplex. PITA is based on a slightly different model which computes the difference between the free energy gained from the formation of the miRNA/target duplex and the energetic cost of unpairing the target to make it accessible to the miRNA [70].

Although all of the mentioned tools were rather successful in predicting effective miRNA targets, the problem still remains a big challenge. The high number of false positives and the use of conservation criteria reveal our partial knowledge in the targeting mechanisms. The combined approach of Data Mining, Pattern Discovery and Machine Learning techniques together with thermodynamics and the availability of more reliable experimental data, will allow the improvement of predictions and enhance our knowledge of RNAi mechanisms.

5 Design of Synthetic miRNAs

The experiments of Fire and Mello on gene silencing artificially induced by small dsRNAs complementary to their targets earned them the Nobel Prize for medicine in 2006 and gave rise to the development of new therapeutic strategies for cancer and other diseases [5]. Small siRNA molecules were initially designed for gene knock out experiments and, later, short-hairpin-RNAs (shRNA) and artificially designed miRNAs revealed their efficacy in inhibiting specific proteins, thus constituting a new potential class of smart drugs, for the treatment of infections and all the other diseases which are related to over-expressed proteins [71].

A siRNA can be introduced into a cell as a double strand RNA molecule perfectly complementary to a site of the target RNA and, then, it enters the RNAi cellular pathway directly in the RISC loading phase which occurs in the cytoplasm [72, 73]. shRNAs, which are hairpin shaped siRNAs, and artificial pre-miRNAs need to enter the nucleus first, in order to exert their function [74]. These molecules are exogenous regulators of mRNA levels and they can be used to partially or totally reduce the expression of one or more genes. In some cases, a partial regulation is needed in order to restore the normal expression of a protein and to let it perform its functions. In other cases, the complete silencing of the target molecule is required.

The main issue of the small RNA-based therapeutic approach is the high number of potential side effects due to the partial complementarity of the synthetic RNAs to undesired targets. However, the potential off target genes can be easily identified through simple sequence analysis and reduced, as far as possible, by refining the designed molecules.

As for the other classes of drugs, the main challenge of small RNA-based therapy is the delivery of the molecules into the cell and then, for shRNAs and miRNAs, into the nucleus [73]. Viral vectors are among the most used delivery systems. A recent work showed a significant effect on tumor growth of the injection of an adenovirus encoding a siRNA designed to target the gene HIF-1, combined with ionizing radiations [75]. Another delivery system consists of cancer cell-specific antibodies. Song et al. showed that an antibody against the gene ErbB2 is able to specifically deliver siRNAs only to ErbB2-expressing breast cancer cells [76].

Compared to traditional drugs, small RNAs allow a faster and simpler design and are capable of inhibiting whatever kind of protein, including those known as non-druggable proteins, which have conformations not favourable to small molecule binding [73]. So far, many classes of targets have been successfully inhibited by small RNAs, including neuro-transmitters and neuro-transmitter receptors, cytokines, growth and transcription factors [77–80].

The use of small regulatory RNAs as therapeutic agents is thus promising. The main challenges are related to the specificity, the efficacy and the safety of delivery systems. In the next subsection the most significant computational tools for the design of synthetic small regulatory RNAs will be reviewed.

5.1 Tools for the Design of siRNA and miRNA

Several online tools for the design of siRNA, shRNA and synthetic miRNA have been proposed in the last years for plants and animals. In 2004, Reynolds et al. published the guidelines for rational siRNA design [81]. They performed a systematic analysis of 180 siRNAs targeting the mRNA of two genes, in order to identify specific features likely to contribute to efficient processing at each step of the siRNA pathway. They identified a few characteristics associated with siRNA functionality like low G/C content, the lack of inverted repeats and sense strand base preferences like, for example, A at position 19 and U at position 10. All the identified determinants, applied together, enhanced the selection of functional siRNAs. Similarly, Ui-Tei et al. identified sequence conditions capable of inducing highly effective gene silencing in mammalian cells, like A/U at the $5'$ end of the antisense strand, G/C at the $5'$ end of the sense strand and the absence of any GC stretch of more than 9 nt length, thus confirming the findings of Reynolds [82]. Similar sequence properties are implemented into Deqor, a web-based tool for the design and quality control of siRNAs, which combines the previous knowledge about the features of effective siRNAs with a scoring system designed to evaluate the inhibitory potency of siRNAs and the off target effects [83].

A work by Siolas et al. showed that synthetic shRNAs, mimicking the natural small regulatory RNA molecules, were more potent inducers of RNAi than siRNAs [84]. Chang et al. found that expression of the artificial miRNA in the context of a natural miRNA primary transcript provides the highest levels of mature miRNA in RISC and generally effective silencing [85]. According to previous studies, indicating that the presence of sequences flanking the native miRNA is essential for its efficient processing, and to reports of artificial miRNAs based upon the precursor of the endogenous miR-30 [86–89], they constructed shRNA libraries modelled after the precursor and the primary transcript of miR-30. Furthermore, a recent work showed that artificial miRNAs are safer and processed more efficiently than shRNA-based siRNAs, suggesting their better suitability for therapeutic silencing [90].

Tools have also been developed for the design of artificial miRNAs for plants. A work by Schwab et al., in 2006, showed the highly specific gene silencing performed by artificial miRNAs in Arabidopsis [91]. The authors developed a web-based tool for the design of artificial miRNA sequences. Another similar work shows similar results in rice [92].

Finally, a new computational tool for the design of highly specific synthetic miRNAs, called miR-Synth, have been proposed [93]. The basic ideas behind miR-Synth is the design of a single miRNA able to bind different targets simultaneously, mimicking the endogenous miRNAs' mode of action. Furthermore, miR-Synth tries to avoid negative side effects, reducing the number of potential off-target genes by discarding all those miRNAs that might bind well to other UTRs sites. Given a target mRNA, the structural accessibility analysis is performed through the use of base pairing probabilities, computed by the RNA fold program from the Vienna Package [94]. In this phase, regions which are more likely to be single stranded are identified and then screened for repeated patterns. These will constitute the binding sites for the synthetic miRNAs seeds. Indeed, as already discussed earlier, the presence of multiple binding sites for a miRNA is a key feature for efficient silencing. In order to reduce as much as possible the number of potential off-target genes, all those patterns which appear in multiple copies in other UTRs are discarded. Other features coming from the experimentally validated endogenous miRNA/target pairs are also taken into account. These include, among others, the AU-rich nucleotide composition near the seeds' binding sites and constraints on the position of the sites. For each seed a synthetic miRNA is then designed, according to the endogenous miRNAs' features, by using a consensus criterion based on sequence profiles. miR-Synth's output consists of the designed miRNAs, their binding sites on the target gene and the list of all possible off-target genes to which a synthesized miRNA might bind. Although miR-Synth has not yet been experimentally validated, the produced miRNAs satisfy the empirical binding rules inferred from validated miRNA/target pairs, thus confirming the plausibility of the proposed method.

The development of more reliable synthetic miRNA design tools is strictly related to the progresses in the prediction of endogenous miRNA targets. Indeed, a better understanding of the rules underlying the natural miRNA processing and function could surely improve the design of the artificial ones. Moreover, further wet biology experiments on these synthetic molecules are needed, in order to investigate the relationships between the designed sequences and their effects on the cells and learn the determinants that allow to modulate the degree of silencing of the targets.

6 A miRNA Knowledge Base

As discussed in Sect. 4, many computational predictions of miRNA targets are available on the web, but a precise association between miRNAs and phenotypes has been demonstrated only for few cases. Much more is known about genes: for example, the Gene Ontology database (GO) [95] provides annotations for processes and functions in which they are involved. Moreover, there is a vast literature concerning genetic roles in pathologies. Nonetheless, miRNAs may be annotated with information about their validated or predicted targets.

A common approach in the study of diseases or biological processes involving miRNAs, requires the extraction of data from several independent sources, such as miRNA/target prediction databases, gene functional annotations, expression profiles and biomedical literature. Thus, there is a need for a system that integrates data from heterogeneous sources in order to build extensible and updatable knowledge bases. This system should be equipped with mining algorithms capable of inferring new knowledge.

The first system for functional annotation of miRNAs is miRGator [96], a database which integrates data from different sources, like target prediction tools and Gene Ontology, and makes it available through a set of standard queries. Although miRGator represents a first attempt at the integration of such data, it has several limitations. It does not provide information about the diseases, nor does it implement customizable queries or data mining facilities.

Here we will describe miRò, a new system which provides users with miRNA-phenotype associations in humans [97]. miRò is a web-based environment that allows users to perform simple searches and sophisticated data mining queries. The main goal of miRò is to provide users with powerful query tools for finding non-trivial associations among heterogeneous data and thereby to allow the identification of relationships among genes, processes, functions and diseases at the miRNA level. Finally, the data mining module includes a specificity function allowing selection of the most significant associations among validated data.

6.1 The miRò System

The miRò web-site integrates data from different sources, as shown in Fig. 2. miRNAs are annotated with information about their precursor and mature sequences coming from miRBase [39], and with expression profiles obtained from the Mammalian microRNA Atlas [98]. The miRNA Atlas contains expression patterns of pre-miRNAs and mature miRNAs in several kinds of tissues, both normal and malignant. miRNAs are also associated to GO terms and diseases through their targets: each miRNA inherits all the annotations of its target genes. Experimentally supported miRNA/target pairs come from miRecords [56]. The predicted targets are taken from the TargetScan [58, 62], PicTar [66] and miRanda [64, 65] web-sites. The target genes records are enriched with general information such as genomic

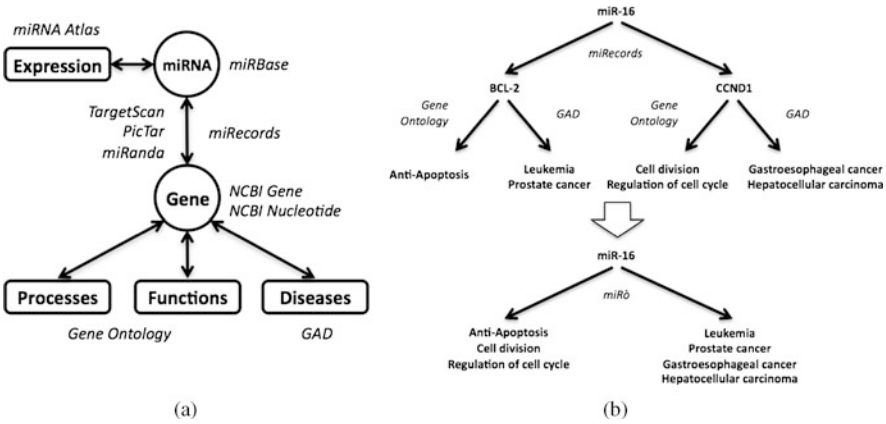


Fig. 2 The miRò knowledge base schema. (**a**) miRNAs are annotated with their features coming from miRBase and their expression profiles coming from the miRNA Atlas. They are linked to processes, functions and diseases through their predicted (by TargetScan, PicTar and miRanda) or validated target genes (miRecords). (**b**) In this case, miR-16 has two validated targets, BCL2 and CCND1, among others. These genes are annotated with GO terms and diseases, thus miR-16 inherits these annotations

context and transcript-related data coming from the NCBI Gene and Nucleotide databases. The ontological terms with which the target genes are annotated (processes and functions) are obtained from the Gene Ontology Database. Finally, the gene-disease relations come from the Genetic Association Database (GAD) [99], which is a database of human genetic association studies of complex diseases and disorders.

All the data are collected and maintained up-to-date in a MySQL database. In particular, the most relevant data about the miRNAs and the target genes, such as the genomic contexts and the sequences, are stored in the database for an immediate availability, while links to the original sources are provided for more detailed information. The data are retrieved from the source websites as flat files except for GO, which is provided as a mySQL db dump and the miRNA Atlas, which is given as a collection of Excel spreadsheets. Initially, miRNA information from the miRBase files is stored. Then, all the target prediction data are screened and stored together with information on the target genes, retrieved from the NCBI Gene files. Gene aliases are also stored, in order to facilitate the subsequent integrations. In the prediction files, the genes are identified by their NCBI IDs, while the miRNAs are identified either by their accessions (miRanda) or their IDs (TargetScan and PicTar). The genes are then annotated with their GO terms, coming from the GO Database, and their associated diseases, retrieved from GAD. Since in both cases the genes are identified by their names, the gene aliases are often used in this step, in order to correctly identify all of them. Finally, miRNA expression profiles are integrated. For each miRNA, identified by its ID, all the expressing tissues, together with the expression values, are stored. The percentage distribution of miRNA clones in the tissues is also computed and then stored for fast retrieval.

Inconsistencies in data integration are prevented by a semi-automatic process: the system automatically detects and reports in a logfile any mismatches, which are then resolved manually. For example, all the miRNA names in the prediction files (PicTar and TargetScan) which can not be found in the miRBase data (e.g. their IDs have been changed), are reported in the logfile. The automatic procedure also retrieves all the similar names in the database. These are then manually screened, sometimes using also sequence information, in order to find the correct ones.

The system also automatically checks for new releases of the source databases every three months and performs an update if needed.

6.2 The miRò Web Interface

The miRò web interface allows the user to perform four different types of queries. A simple search is used to get information about a single object, which can be a miRNA, a gene, a process, a function, a disease or a tissue. For example, it is possible to specify a miRNA or to choose one from the complete miRNA catalog to get the list of all the diseases and GO Terms (Processes and functions) which can be associated to that miRNA through its targets. The results are ordered by the number of tools which predict the corresponding miRNA/target pairs and the experimentally supported associations are always given first. Using the "AND" constraint enables users to select only the terms associated to the targets predicted by all the selected tools. This may help to identify the most strongly supported associations, and reduce the falsely predicted associations.

Similarly, the user can search for all the miRNAs associated to a certain gene, disease, process or function, and obtain a list of all the miRNAs expressed in a certain tissue with their expression levels.

A customized search also allows users to extend the knowledge base by a personal set of miRNA/target pairs. These pairs will be temporarily stored and used in all the session queries. This feature may be helpful in testing new miRNA/target data.

The advanced search form can be used to perform more sophisticated queries. The user chooses a "subject" among miRNA, gene, disease, process and function, then specifies a list of constraints that the subject must satisfy.

For example, it is to possible to ask miRò to show all the miRNAs (the subject is "miRNA") which are associated to heart failure, RNA binding and apoptosis, but not to congenital heart diseases. The user may also choose the sources of the miRNA/target pairs. This allows to tighten or relax the query conditions, in order to get a smaller or a larger output, respectively. The system will show the list of all the miRNAs which satisfy these constraints, with details about the involved targets (see Fig. 3). This query tool links objects through miRNA-based associations. For example, a disease d and a process p which are not linked through any common gene might be associated through a miRNA which regulates a gene gd, involved in d,

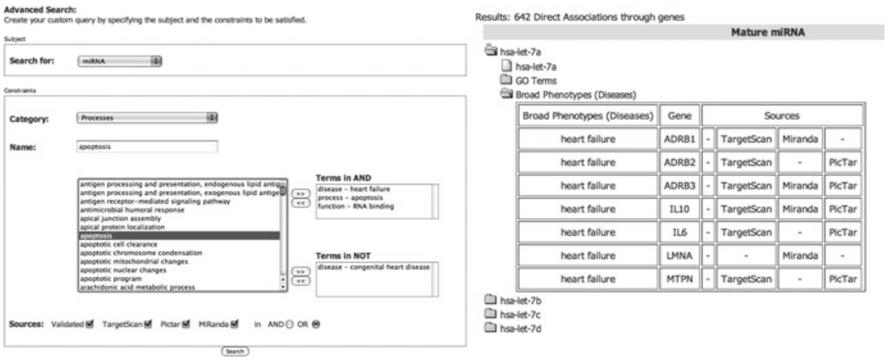


Fig. 3 An example of Advanced Query execution. (**a**) The advanced search form with the selected constraints: miRNAs involved in Heart failure, apoptosis and RNA binding, but not in Congenital heart disease will be returned. (**b**) A subset of the corresponding results. For example, the miRNA let-7a satisfies the specified requirements. In particular it is predicted to bind seven genes, which are involved in the Heart Failure disease. The details about the other constraints (apoptosis and RNA binding) are shown in the GO Terms folder

and a gene gp, involved in p. This introduces a new layer of associations between genes and processes inferred based on miRNAs annotations. These associations are given in the Advanced Search results pages, when the subject of the query is a GO term or a disease.

6.3 The miRò Data Mining Facilities

Since miRNAs are associated to GO terms and diseases, they can be clustered according to their common terms, i.e. miRNAs which are associated to the same set of terms are grouped together. miRò is equipped with a data mining module, based on a maximal frequent itemset computation [100, 101], which allows users to query the database and extract non-trivial subsets of miRNAs sharing some features. The analysis is performed using different support thresholds: a high threshold allows to obtain a small number of miRNA subsets associated to a great number of terms, while a low threshold gives more subsets associated to fewer terms. All the subsets have been pre-computed off-line on the dataset of the validated miRNA/target pairs.

In the miRò interface, the user may choose up to n miRNAs together with an association criteria (i.e. process or disease). The system will find all the subsets of the selected miRNAs and the processes or diseases which they are most closely associated to. This may suggest that a set of miRNAs acting cooperatively carry out certain biological functions, as will be shown in the next section. Moreover, miRNA/term associations are scored in order to highlight the most significant ones, as discussed in the next subsection.

Each miRNA/process or miRNA/disease pair, in each subset, is scored according to a specificity scoring function. This evaluates the relationships between the

Processes	(1) miR-124	(2) miR-137
G1 phase of mitotic cell cycle	0.000297	0.018519
negative regulation of transcription from RNA polymerase II promoter	0.000789	0.027778
positive regulation of cell-matrix adhesion	0.000297	0.018519
regulation of transcription, DNA-dependent	0.021306	0.027778
protein amino acid phosphorylation	0.004036	0.018519
cell cycle	0.004978	0.018519
signal transduction	0.00744	0.027778
regulation of gene expression	0.000297	0.018519
hemopoiesis	0.000297	0.018519
gliogenesis	0.000297	0.018519
cell dedifferentiation	0.000297	0.018519
regulation of erythrocyte differentiation	0.000297	0.018519
negative regulation of osteoblast differentiation	0.001061	0.018519
positive regulation of transcription from RNA polymerase II promoter	0.00126	0.027778
positive regulation of fibroblast proliferation	0.000297	0.018519
negative regulation of epithelial cell proliferation	0.000297	0.018519
cell division	0.002744	0.018519
Max (cluster)	0.021306	0.027778
Max (global)	0.075306	0.166667

Fig. 4 An example of an miRNA subset containing 2 miRNAs (miR-124 and miR-137) both involved in 17 processes. The entries contain the miRNA/process specificity scores. The entries are colored based on their value: the *red* entries indicate the maximum value of the subset, while the *blue* ones indicate the minimum values. In this case, the most relevant associations in the subset are between miR-137 and the four processes corresponding to the red entries. This may suggest a specific role of miR-137 in such processes and is due to the number of targets of the miRNA involved in such processes and to their specificity to the processes (Color figure online)

miRNAs and their annotation terms (processes and diseases). The specificity of a miRNA m_k for a process p_j is defined as follows:

$$S_{m_k,p_j} = \frac{|G_{m_k,p_j}|}{|G_{m_k}|} \cdot \frac{\sum_{g_i \in G_{m_k,p_j}} S_{g_i}}{|G_{m_k,p_j}|} = \frac{\sum_{g_i \in G_{m_k,p_j}} S_{g_i}}{|G_{m_k}|}$$

where G_{m_k,p_j} is the set of the target genes of miRNA m_k involved in the process p_j, and G_{m_k} is the set of all the target genes of m_k. The specificity of a gene S_{g_i} is inversely proportional to the number of processes in which the gene is involved:

$$S_{g_i} = \frac{1}{|P_{g_i}|}$$

where P_{g_i} is the set of the processes in which the gene g_i is involved.

Intuitively, a gene associated with fewer processes is more focused on the ones remaining. The specificity of a miRNA for a process relies on the number of targets and their specificity to the process.

This function has been applied to the set of validated miRNA/target interactions. The subsets of frequently associated miRNAs are visualized by tables showing the miRNAs and the processes/diseases to which they are all associated, with their specificity scores. The table entries are colored based on the specificity value ranging from blue (lowest value) to red (highest value) (see Fig. 4).

6.4 Validation of miRò

The system has been tested on some known cases coming from the literature. It has been able to identify miRNA-disease and miRNA-process associations previously reported. The role of the miR-17-92 cluster in development and disease had been well established [102]. The expression of these miRNAs promotes cell proliferation, suppresses apoptosis of cancer cells, and induces tumor angiogenesis. In particular they are involved in lymphoma, melanoma and other types of cancer (breast, colorectal, lung, ovarian, pancreas, prostate and stomach). The miR-17-92 cluster also plays an essential role during normal development of the heart, lungs, and immune system.

Performing an advanced search on miRò looking for the diseases related to subgroups of the miRNAs of the miR-17-92 cluster, one finds that four of them (miR-17, miR-19a, miR-19b and miR-92a) are associated to those tumors together with other pathologies. Moreover, an advanced search for the processes involving the cluster returns, among others, angiogenesis, apoptosis, cell cycle, cell growth and proliferation, heart and lung development. These processes are also linked to the diseases reported in [102].

The search through the miRò Advanced Search form for the miRNAs miR-1, miR-206 and miR-133a, independently known to be involved in muscle activity [103], shows the involvement of such miRNAs in muscle contraction. Moreover, the data mining analysis detects a high correlation of miR-1 and miR-206, which are frequently associated in terms of both biological processes and pathologies.

Similarly, miR-124 and miR-137, which have been independently reported to be involved in glioblastoma [104], are associated together to several processes, among which gliogenesis.

The specificity function, introduced in the previous section, aims at scoring the miRNA annotations in order to highlight the most significant ones.

Among the top ranking miRNA/disease and miRNA/process associations, there are cases which have been reported in literature, as shown in Tables 3 and 4. For example, the top scoring miRNA/disease association links miR-433 to Parkinson's disease. This result is confirmed by a study which has shown that the disruption of the miR-433 binding site of the gene FGF20, confers risk for Parkinson's disease. Indeed, the increase in translation of FGF20 is correlated with increased alpha-synuclein expression, which is known to cause Parkinson's diseases [105].

Similarly, the association between miR-224 and apoptosis is among the top ranking miRNA/Process associations. This is supported by a study showing that miR-224, which is up-regulated in Hepatocellular carcinoma patients, increases apoptotic cell death by targeting the apoptosis inhibitor-5 (API-5) [106].

7 Conclusions

The discovery of gene silencing and RNAi represents a revolution in the RNA molecular biology field. RNAi also constitutes a promising new therapeutic ap-

Table 3 Top ranking miRNA/disease associations reported in literature

Rank	miRNA	Disease	Reference
1	miR-23b	Leukemia	[107]
1	miR-433	Parkinson's disease	[105]
2	miR-107	Alzheimer's disease	[108]
2	miR-27b	Leukemia	[109]
2	miR-9	Alzheimer's disease	[110]
3	miR-20a	Lung carcinoma	[111]
3	miR-29a	Alzheimer's disease	[110]

Table 4 Top ranking miRNA/process associations reported in literature

Rank	miRNA	Process	Reference
2	miR-212	Cell-cell junction assembly	[112]
6	miR-224	Apoptosis	[106]
7	miR-433	Fibroblast growth factor receptor signaling pathway, Cell growth	[105]
9	miR-221	Cell cycle arrest	[113]
9	miR-222	Cell cycle arrest	[113]
11	miR-219-5p	Apoptosis	[114]

proach. In this essay we reviewed the basic principles of gene silencing and the computational approaches used for the analysis of this complex mechanism. The combination of both computational and experimental methods, working strictly together, have given a big contribution to this field, helping to uncover the biological mechanisms that underlie this powerful phenomenon. As new experimental techniques will allow to obtain more and more reliable data, bioinformatics methods will keep constituting a fundamental resource for their analysis and classification. The rules that govern RNAi are already written in the genome and in the complex interaction networks involving the small RNAs, the genes and their products. So there is a need for more intense application of Data Mining and Machine Learning techniques, able to infer and predict new knowledge, apart from evolutionary conservation which can be a useful criterion but unknown to the cell itself for the maintenance of its functions. This clearly indicates that more effort is needed in the development on new analysis tools and that small RNA molecules should be considered in the context of a more complex environment, involving other RNA molecules, proteins and chemicals [115]. The complex regulatory networks inferred from expression and interaction data needs to be mined in order to extract significant patterns that would allow a better comprehension of the sophisticated regulation mechanisms. The development of new therapeutic approaches based on small RNAs is strictly connected to the progresses in the target prediction. As our knowledge in miRNA/target interactions increases, we will become able to design more and more efficient molecules mimicking the endogenous ones, less toxic as possible. Deep laboratory experiments will also help to determine the relationships between the design rules and their effects on the synthesized molecules. Finally, a great contribution to biomedicine is given by knowledge bases equipped with sophisticated analysis tools, able to find direct and hidden associations between the expression of regulatory RNAs and the phenotype, allowing a better understanding of the molecular basis of diseases and the identification of the most important targets for therapeutic applications.

References

1. Dykxhoorn, D.M., Lieberman, J.: The silent revolution: RNA interference as basic biology, research tool, and therapeutic. Annu. Rev. Med. **56**, 401–423 (2005)
2. Chu, C.Y., Rana, T.M.: Small RNAs: regulators and guardians of the genome. J. Cell. Physiol. **213**(2), 412–419 (2007)
3. Kutter, C., Svoboda, P.: miRNA, siRNA, piRNA: knowns of the unknown. RNA Biol. **5**(4), 181–188 (2008)
4. Bartel, D.P.: MicroRNAs: genomics, biogenesis, mechanism, and function. Cell **116**, 281–297 (2004)
5. Fire, A., et al.: Potent and specific genetic interference by double-stranded RNA in caenorhabditis elegans. Nature **391**(6669), 806–811 (1998)
6. Tomari, Y., Zamore, P.D.: Perspective: machines for RNAi. Genes Dev. **19**, 517–529 (2005)
7. Meister, G., Tuschl, T.: Mechanisms of gene silencing by double-stranded RNA. Nature **431**, 343–349 (2004)
8. Matranga, C., et al.: Passenger-strand cleavage facilitates assembly of siRNA into Ago2-containing RNAi enzyme complexes. Cell **123**, 607–620 (2005)
9. Miyoshi, K., et al.: Slicer function of Drosophila Argonautes and its involvement in RISC formation. Genes Dev. **19**, 2837–2848 (2005)
10. Rand, T.A., et al.: Argonaute2 cleaves the anti-guide strand of siRNA during RISC activation. Cell **123**, 621–629 (2005)
11. Gregory, R.I., et al.: Human RISC couples microRNA biogenesis and post-transcriptional gene silencing. Cell **123**, 631–640 (2005)
12. Maniataki, E., Mourelatos, Z.: A human, ATP-independent, RISC assembly machine fueled by pre-miRNA. Genes Dev. **19**, 2979–2990 (2005)
13. Talmor-Neiman, M., et al.: Identification of trans-acting siRNAs in moss and an RNA-dependent RNA polymerase required for their biogenesis. Plant J. **48**(4), 511–521 (2006)
14. Fahlgren, N., et al.: Regulation of AUXIN RESPONSE FACTOR3 by TAS3 ta-siRNA affects developmental timing and patterning in arabidopsis. Curr. Biol. **16**(9), 939–944 (2006)
15. Mette, M.F., et al.: Transcriptional silencing and promoter methylation triggered by double stranded RNA. EMBO J. **19**(19), 5194–5201 (2000)
16. Vagin, V.V., et al.: A distinct small RNA pathway silences selfish genetic elements in the germline. Science **313**(5785), 320–324 (2006)
17. Mochizuki, K., Gorovsky, M.A.: Conjugation-specific small RNAs in tetrahymena have predicted properties of scan (scn) RNAs involved in genome rearrangement. Genes Dev. **18**(17), 2068–2073 (2004)
18. Katiyar-Agarwal, S., et al.: A novel class of bacteria-induced small RNAs in Arabidopsis. Genes Dev. **21**(23), 3123–3134 (2007)
19. Kim, V.N.: MicroRNA biogenesis: coordinated cropping and dicing. Nat. Rev. Mol. Cell Biol. **6**, 376–385 (2005)

20. Okamura, K., et al.: The regulatory activity of microRNA* species has substantial influence on microRNA and 3′ UTR evolution. Nat. Struct. Mol. Biol. **15**, 354–363 (2008)
21. Erson, A.E., Petty, E.M.: MicroRNAs in development and disease. Clin. Genet. **74**, 296–306 (2008)
22. Bernstein, E., et al.: Dicer is essential for mouse development. Nat. Genet. **35**, 215–217 (2003)
23. Harris, K.S., et al.: Dicer function is essential for lung epithelium morphogenesis. Proc. Natl. Acad. Sci. USA **103**, 2208–2213 (2006)
24. Harfe, B.D., et al.: The RNaseIII enzyme Dicer is required for morphogenesis but not patterning of the vertebrate limb. Proc. Natl. Acad. Sci. USA **102**, 10898–10903 (2005)
25. O'Rourke, J.R., et al.: Essential role for Dicer during skeletal muscle development. Dev. Biol. **311**, 359–368 (2007)
26. Muljo, S.A., et al.: Aberrant T cell differentiation in the absence of Dicer. J. Exp. Med. **202**, 261–269 (2005)
27. Boutz, P., et al.: MicroRNAs regulate the expression of the alternative splicing factor nPTB during muscle development. Genes Dev. **21**, 71–84 (2007)
28. Garofalo, M., Condorelli, G., Croce, C.M.: MicroRNAs in diseases and drug response. Curr. Opin. Pharmacol. **8**(5), 661–667 (2008)
29. Wang, W.X., et al.: The expression of microRNA miR-107 decreases early in Alzheimer's disease and may accelerate disease progression through regulation of beta-site amyloid precursor protein-cleaving enzyme 1. J. Neurosci. **28**, 1213–1223 (2008)
30. Eisenberg, I., et al.: Distinctive patterns of microRNA expression in primary muscular disorders. Proc. Natl. Acad. Sci. USA **104**, 17016–17021 (2007)
31. Carè, A., et al.: MicroRNA-133 controls cardiac hypertrophy. Nat. Med. **13**, 613–618 (2007)
32. Yang, B., et al.: The muscle-specific microRNA miR-1 regulates cardiac arrhythmogenic potential by targeting GJA1 and KCNJ2. Nat. Med. **13**, 486–491 (2007)
33. Visone, R., Croce, C.M.: MiRNAs and cancer. Am. J. Pathol. **174**(4), 1131–1138 (2009)
34. Cimmino, A., et al.: miR-15 and miR-16 induce apoptosis by targeting BCL2. Proc. Natl. Acad. Sci. USA **102**(39), 13944–13949 (2005)
35. Johnson, S.M., et al.: RAS is regulated by the let-7 microRNA family. Cell **120**, 635–647 (2005)
36. Kumar, M.S., et al.: Suppression of non-small cell lung tumor development by the let-7 microRNA family. Proc. Natl. Acad. Sci. USA **105**, 3903–3908 (2008)
37. Costinean, S., et al.: Pre-B cell proliferation and lymphoblastic leukemia/high-grade lymphoma in E(mu)-miR155 transgenic mice. Proc. Natl. Acad. Sci. USA **103**, 7024–7029 (2006)
38. Eisenberg, I., et al.: Tumour invasion and metastasis initiated by microRNA-10b in breast cancer. Nature **449**, 682–688 (2007)
39. Griffiths-Jones, S., et al.: miRBase: tools for microRNA genomics. Nucleic Acids Res. **36**(Database issue), D154–D158 (2008)
40. Bentwich, I., et al.: Identification of hundreds of conserved and nonconserved human microRNAs. Nat. Genet. **37**, 766–770 (2005)
41. Ambros, V., et al.: MicroRNAs and other tiny endogenous RNAs in C. elegans. Curr. Biol. **13**(10), 807–818 (2003)
42. Seffens, W., Digby, D.: mRNAs have greater negative folding free energies than shuffled or codon choice randomized sequences. Nucleic Acids Res. **27**(7), 1578–1584 (1999)
43. Zhang, B., et al.: Plant microRNA: a small regulatory molecule with big impact. Dev. Biol. **289**(1), 3–16 (2006)
44. Zhang, B.H., et al.: Evidence that miRNAs are different from other RNAs. Cell. Mol. Life Sci. **63**(2), 246–254 (2006)
45. Zhang, B., et al.: Computational identification of microRNAs and their targets. Comput. Biol. Chem. **30**(6), 395–407 (2006)
46. Lagos-Quintana, M., et al.: Identification of novel genes coding for small expressed RNAs. Science **294**(5543), 853–858 (2001)

47. Wang, X., et al.: MicroRNA identification based on sequence and structure alignment. Bioinformatics **21**(18), 3610–3614 (2005)
48. Dezulian, T., et al.: Identification of plant microRNA homologs. Bioinformatics **22**(3), 359–360 (2006)
49. Nam, J.W., et al.: Human microRNA prediction through a probabilistic co-learning model of sequence and structure. Nucleic Acids Res. **33**(11), 3570–3581 (2005)
50. Lai, E.C., et al.: Computational identification of Drosophila microRNA genes. Genome Biol. **4**(7), R42 (2003)
51. Lim, L.P., et al.: The microRNAs of Caenorhabditis elegans. Genes Dev. **17**(8), 991–1008 (2003)
52. Bonnet, E., et al.: Detection of 91 potential conserved plant microRNAs in Arabidopsis thaliana and Oryza sativa identifies important target genes. Proc. Natl. Acad. Sci. USA **101**(31), 11511–11516 (2004)
53. Lindow, M., et al.: Intragenomic matching reveals a huge potential for miRNA-mediated regulation in plants. PLoS Comput. Biol. **3**(11), e238 (2007)
54. Bartel, D.P.: MicroRNAs: target recognition and regulatory functions. Cell **136**(2), 215–233 (2009)
55. Sethupathy, P., et al.: TarBase: a comprehensive database of experimentally supported animal microRNA targets. RNA **12**, 192–197 (2006)
56. Xiao, F., et al.: miRecords: an integrated resource for microRNA-target interactions. Nucleic Acids Res. **37**(Database issue), D105–D110 (2009)
57. Chi, S.W., et al.: Argonaute HITS-CLIP decodes microRNA–mRNA interaction maps. Nature **460**(7254), 479–486 (2009)
58. Lewis, B.P., Burge, C.B., Bartel, D.P.: Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microRNA targets. Cell **120**(1), 15–20 (2005)
59. Rehmsmeier, M., et al.: Fast and effective prediction of microRNA/target duplexes. RNA **10**, 1507–1517 (2004)
60. Hofacker, I.L.: How microRNAs choose their targets. Nat. Genet. **39**(10), 1191–1192 (2007)
61. Mathews, D.H.: Revolutions in RNA secondary structure prediction. J. Mol. Biol. **359**(3), 526–532 (2006)
62. Grimson, A., et al.: MicroRNA targeting specificity in mammals: determinants beyond seed pairing. Mol. Cell **27**, 91–105 (2007)
63. Mazière, P., Enright, A.J.: Prediction of microRNA targets. Drug Discov. Today **12**(11–12), 452–458 (2007)
64. John, B., et al.: Human MicroRNA targets. PLoS Biol. **2**(11), 1862–1879 (2004)
65. Betel, D., et al.: The microRNA.org resource: targets and expression. Nucleic Acids Res. **36**(Database issue), D149–D153 (2007)
66. Krek, A., et al.: Combinatorial microRNA target predictions. Nat. Genet. **37**(5), 495–500 (2005)
67. Kiriakidou, M., et al.: A combined computational-experimental approach predicts human microRNA targets. Genes Dev. **18**(10), 1165–1178 (2004)
68. Miranda, K.C., et al.: A pattern-based method for the identification of MicroRNA binding sites and their corresponding heteroduplexes. Cell **126**, 1203–1217 (2006)
69. Long, D., et al.: Potent effect of target structure on microRNA function. Nat. Struct. Mol. Biol. **14**(4), 287–294 (2007)
70. Kertesz, M., et al.: The role of site accessibility in microRNA target recognition. Nat. Genet. **39**(10), 1278–1284 (2007)
71. Wadhwa, R., et al.: Know-how of RNA interference and its applications in research and therapy. Mutat. Res. **567**(1), 71–84 (2004)
72. Takeshita, F., Ochiya, T.: Therapeutic potential of RNA interference against cancer. Cancer Sci. **97**(8), 689–696 (2006)
73. Sah, D.W.: Therapeutic potential of RNA interference for neurological disorders. Life Sci. **79**(19), 1773–1780 (2006)

74. Rao, D.D., et al.: siRNA vs. shRNA: similarities and differences. Adv. Drug Deliv. Rev. **61**(9), 746–759 (2009)
75. Zhang, X., et al.: Enhancement of hypoxia-induced tumor cell death in vitro and radiation therapy in vivo by use of small interfering RNA targeted to hypoxia-inducible factor-1α. Cancer Res. **64**, 8139–8142 (2004)
76. Song, E., et al.: Antibody mediated in vivo delivery of small interfering RNAs via cell-surface receptors. Nat. Biotechnol. **23**, 709–717 (2005)
77. Luo, M., et al.: RNAi of neuro peptide Y (NPY) for neuropathic pain. Society for Neuroscience Abstracts (2005)
78. Filleur, S., et al.: SiRNA-mediated inhibition of vascular endothelial growth factor severly limits tumor resistance to antiangiogenic thrombospondin-1 and slows tumor vascularization and growth. Cancer Res. **63**(14), 3919–3922 (2003)
79. Zhang, Y., et al.: Intravenous RNA interference gene therapy targeting the human epidermal growth factor receptor prolongs survival in intracranial brain cancer. Clin. Cancer Res. **10**(11), 3667–3677 (2004)
80. Gaudilliere, B., et al.: RNA interference reveals a requirement for MEF2A in activity-dependent neuronal survival. J. Biol. Chem. **277**(48), 46442–46446 (2002)
81. Reynolds, A., et al.: Rational siRNA design for RNA interference. Nat. Biotechnol. **22**(3), 326–330 (2004)
82. Ui-Tei, K., et al.: Guidelines for the selection of highly effective siRNA sequences for mammalian and chick RNA interference. Nucleic Acids Res. **32**(3), 936–948 (2004)
83. Henschel, A., Buchholz, F., Habermann, B.: DEQOR: a web-based tool for the design and quality control of siRNAs. Nucleic Acids Res. **32**(Web Server issue), W113–W120 (2004)
84. Siolas, D., et al.: Synthetic shRNAs as potent RNAi triggers. Nat. Biotechnol. **23**(2), 227–231 (2005)
85. Chang, K., et al.: Lessons from nature: microRNA-based shRNA libraries. Nat. Methods **3**(9), 707–714 (2006)
86. Zeng, Y., Wagner, E.J., Cullen, B.R.: Both natural and designed micro RNAs can inhibit the expression of cognate mRNAs when expressed in human cells. Mol. Cell **9**, 1327–1333 (2002)
87. Silva, J.M., et al.: Second-generation shRNA libraries covering the mouse and human genomes. Nat. Genet. **37**, 1281–1288 (2005)
88. Chen, C.Z., et al.: MicroRNAs modulate hematopoietic lineage differentiation. Science **303**, 83–86 (2004)
89. Zeng, Y., Cullen, B.R.: Sequence requirements for micro RNA processing and function in human cells. RNA **9**, 112–123 (2003)
90. Boudreau, R.L., Martins, I., Davidson, B.L.: Artificial microRNAs as siRNA shuttles: improved safety as compared to shRNAs in vitro and in vivo. Molec. Ther. **17**(1), 169–175 (2009)
91. Schwab, R., et al.: Highly specific gene silencing by artificial microRNAs in arabidopsis. Plant Cell **18**(5), 1121–1133 (2006)
92. Warthmann, N., et al.: Highly specific gene silencing by artificial miRNAs in rice, PLoS. One. **3**(3), e1829 (2008)
93. Laganà, A., et al.: miR-Synth: a tool for designing highly specific synthetic miRNAs. In: ECSB 2007—The European Conference on Synthetic Biology: Design, Programming and Optimization of Biological Systems, Sant Feliu de Guixols, Spain, November 24–29 (2007)
94. Gruber, A.R., et al.: The Vienna RNA websuite. Nucleic Acids Res. **36**(Web Server issue), W70–W74 (2008)
95. Ashburner, M., et al.: Gene ontology: tool for the unification of biology. The gene ontology consortium. Nat. Genet. **25**(1), 25–29 (2000)
96. Nam, S., et al.: miRGator: an integrated system for functional annotation of microRNAs. Nucleic Acids Res. **36**(Database issue), D159–D164 (2007)
97. Laganà, A., et al: miRò: a miRNA knowledge base. Database, vol. 2009, bap008 (2009). doi:10.1093/database/bap008

98. Landgraf, P., et al.: A mammalian microRNA expression atlas based on small RNA library sequencing. Cell **129**(7), 1401–1414 (2007)
99. Becker, K.G., et al.: The genetic association database. Nat. Genet. **36**(5), 431–432 (2004)
100. Agrawal, R., et al.: Fast algorithms for mining association rules. In: Proc. 20th Int. Conf. Very Large Data Bases, VLDB 1994, pp. 487–499 (1994)
101. Burdick, D., et al.: MAFIA: a maximal frequent itemset algorithm for transactional databases. In: Proceedings of the 17th International Conference on Data Engineering 2001, pp. 443–452 (2001)
102. Ventura, A., et al.: Targeted deletion reveals essential and overlapping functions of the miR-17-92 family of miRNA clusters. Cell **132**, 875–886 (2008)
103. Rao, P.K., et al.: Myogenic factors that regulate expression of muscle-specific microRNAs. Proc. Natl. Acad. Sci. USA **103**(23), 8721–8726 (2006)
104. Silber, J., et al.: miR-124 and miR-137 inhibit proliferation of glioblastoma multiforme cells and induce differentiation of brain tumor stem cells. BMC Med. **6**, 14 (2008)
105. Wang, G., et al.: Variation in the miRNA-433 binding site of FGF20 confers risk for Parkinson disease by overexpression of alpha-synuclein. Am. J. Hum. Genet. **82**(2), 283–289 (2008)
106. Wang, Y., et al.: Profiling microRNA expression in hepatocellular carcinoma reveals microRNA-224 up-regulation and apoptosis inhibitor-5 as a microRNA-224-specific target. J. Biol. Chem. **283**(19), 13205–13215 (2008)
107. Calin, G.A., et al.: A MicroRNA signature associated with prognosis and progression in chronic lymphocytic leukemia. N. Engl. J. Med. **353**(17), 1793–1801 (2005)
108. Wang, W.X., et al.: The expression of microRNA miR-107 decreases early in Alzheimer's disease and may accelerate disease progression through regulation of beta-site amyloid precursor protein-cleaving enzyme 1. J. Neurosci. **28**(5), 1213–1223 (2008)
109. Mi, S., et al.: MicroRNA expression signatures accurately discriminate acute lymphoblastic leukemia from acute myeloid leukemia. Proc. Natl. Acad. Sci. USA **104**(50), 19971–19976 (2007)
110. Hebert, S.S., et al.: Loss of microRNA cluster miR-29a/b-1 in sporadic Alzheimer's disease correlates with increased BACE1/beta-secretase expression. Proc. Natl. Acad. Sci. USA **105**(17), 6415–6420 (2008)
111. Hayashita, Y., et al.: A polycistronic microRNA cluster, miR-17-92, is overexpressed in human lung cancers and enhances cell proliferation. Cancer Res. **65**(21), 9628–9632 (2005)
112. Tang, Y., et al.: Effect of alcohol on miR-212 expression in intestinal epithelial cells and its potential role in alcoholic liver disease. Alcohol. Clin. Exp. Res. **32**(2), 355–364 (2008)
113. Galardi, S., et al.: miR-221 and miR-222 expression affects the proliferation potential of human prostate carcinoma cell lines by targeting p27Kip1. J. Biol. Chem. **282**(32), 23716–23724 (2007)
114. Izzotti, A., et al.: Downregulation of microRNA expression in the lungs of rats exposed to cigarette smoke. FASEB J. **March**, 806–812 (2009)
115. Ferro, A., et al.: miRScape: a Cytoscape plugin to annotate biological networks with microRNAs. In: NETTAB 2009, Catania, June 10–12 (2009)

The Last Ten Yards

Michael Wigler

Abstract Jack Schwartz was drawn to the biological science late in life. He took a two year sabbatical at Cold Spring Harbor Laboratory, where he demonstrated how computer generated graphics could provide a window into visual processing, and found signatures in the permutation of mitochondrial gene order useful for phylogenetic analysis. He made a lasting impact on computational genomics at the lab by emphasizing the utility of the sort and exact matching. This led to the application of the Burrough-Wheeler algorithm, now a central component to many widely used genome mapping algorithms.

I met Jack just before he retired from the Courant Institute, and we developed a friendship that lasted through his final decade. These notes chronicle a slice of his thinking post retirement, and document Jack's creativity and analytical abilities enduring far past the time when he ceased to care about publishing.

By the end of last century I had become convinced of the need for an influx of sharply quantitative minds into biology, especially genomics. I actively sought such people to join my cancer genetics group at Cold Spring Harbor Laboratory. Bud Mishra, a Professor of Computer Science and Mathematics at the Courant Institute, represented NYU at a small meeting on genomics that we jointly attended. Bud seemed to be a deep fellow with broad interests. Upon my active pursuit, Bud and I collaborated (and still do) on several projects, and it was he who introduced me to Jack. Bud thought Jack and I would get along, and he was right.

1 Taming the Robot

I invited Jack to visit my lab in its bucolic setting on the harbor on the North Shore of Long Island, facing the Sound and Connecticut across the way. He came with his wife Diana on a sunny spring day. I cannot recall our discussions, but I do recall the

M. Wigler (✉)
Cold Spring Harbor Laboratory, Cold Spring Harbor, NY 11724, USA
e-mail: wigler@cshl.edu

M. Davis, E. Schonberg (eds.), *From Linear Operators to Computational Biology*, DOI 10.1007/978-1-4471-4282-9_10,

events. My lab had recently purchased an all purpose liquid handling robot which had proven difficult to operate. I knew Jack had worked on robotics, and I asked him and Diana to have a look. Shortly after, they reported that the operator's "programming" through a clunky interface of key board and buttons was translated into a text file, the real input to the robot. This file could be downloaded, and Jack and Diana decoded it. Thereafter we programmed the robot with text instructions.

The robot had been tamed, and the visit matured into a sabbatical, and then a friendship that lasted until Jack's death. Although Jack took interest in all matters biological, the historical vector of his own training aimed his thought and experimentation upon three topics: genomics, visual perception and cognitive structure. Of these, I have expert[1] knowledge in only the first, and so I will describe that area of Jack's interest in greatest detail.

2 Genomes as Searchable Strings

The parallels of an organism's genetic instructions to a programming language had great appeal to theoretical computer scientists.[2] These genetic instructions, or the "genome", can be (incompletely) described as a string of characters drawn from a four nucleotide alphabet which ultimately instructs the cell how to make other macromolecules such as RNA and protein.[3] The fundamental human query of the genomic string is the "substring exact match", and Jack argued persuasively that fundamental to that was the "sort". This simple idea germinated in a young computer scientist named John Healy, who had joined the lab at about the same time as Jack, and both shared my office. Computational efficiency of genome searches is a significant problem when the genomes are 3×10^9 nucleotides long, the size of the human genome. John saw how to use a sort on a transformed string so that the original string could be searched rapidly for exact matches. Jack grasped John's difficult algorithm immediately, and was able to explain it to me. After researching his method, John discovered that he had re-invented the then obscure Burroughs-Wheeler algorithm (BWA) [1], created for string compression. The utility of an indexed BWA for searches had also been previously discovered [2] but not widely appreciated. John had rediscovered both these inventions. The algorithm became the basis for our microarray designs for several years [3], the only paper on which Jack and I were co-authors. Today, the BWA is the core for many of the software programs used in DNA sequence analysis.

[1] Jack defined an expert as someone who has made every possible mistake.

[2] See for example Jack's 8th and 9th slides from his presentation at Johns Hopkins at http://roma.cshl.edu/jack.

[3] Genes are substrings of DNA. DNA is double stranded, so each element of the genome "string" is really a complementary pair of single stranded nucleotides. Messenger or coding RNA is made by transcribing one strand of the DNA as template. Parts of the RNA are edited, and the sections of RNA removed are called "introns", leaving intact but joining together the "exons". Only then is the RNA is used to make protein.

3 Phylogeny of Mitochondria

Beginning with the Human Genome Sequence Initiative there was a National Institutes of Health mandate to place DNA sequence data into the public domain. Jack loved data, especially free data. There was much that could be done with the deposited sequence data, but in particular Jack was interested in phylogeny, as it was natural to organize the genomes by descent. Most phylogenies are based on comparison of protein sequences and nucleic acid sequences, where the distance between strings is measured roughly as the number of mutations, one nucleotide at a time, that convert one string into another. Jack devised different methods.

In the early 2000's he showed me his results based on study of the circular mitochondrial genome.[4] The mitochondrial genes are highly conserved, so conserved that using single nucleotide mutation is insufficient to establish a phylogeny. Jack observed that the order of the genes (coherent blocks of nucleotides) was not conserved, and suffered permutation during the long course of evolution. Furthermore, Jack saw that there was a minimizing "rule" to describe the permutations: excise a segment of genes from the circle, circularize that segment, and insert that new circle anywhere between the remaining genes of the old circle such that the gene order after insertion is preserved or reversed. If reversed, there was also a reversal of the orientation of each gene.[5] There was a preference for reinsertion with reversal at the site of excision. Unknown to Jack at the time he made his observations, his rules corresponded precisely to the expectations of recombination within a circular genome predicted from the resolution of recombinant intermediates called Holliday structures (see Fig. 1). The minimum number of applications of Jack's rules to transform one genome into another was a measure of the evolutionary distance.

Using his method, the genomes of vertebrates and those of crustaceans could be lumped into two groupings that were very clearly separable but also related. The mitochondrial gene order of lobsters closely resembled mosquitoes. Hippos resembled fish. There were minor variations off the vertebrate theme, or the crustacean theme, and these variations illustrated the minimal rules. But three types of crustaceans stood out: ticks, fleas and mites. These bore no obvious relation to each other or to the crustaceans. This surprised and delighted Jack, as he had thought that all the modern animals with exoskeletons were derived from the same explosion of metazoans. On several occasions I urged Jack to formulate this work and write a paper, and I thought there was interesting mathematics in viewing mutations as elements of a group, and how sets of group generators established "efficient" descriptors of a distance metric on the group of mutations. But he balked, saying that computing precise distances was not feasible. He would have guided a graduate student, but he no longer had any interest in publishing findings or in expanding his Curriculum Vitae. So the work remained unpublished, although Jack spoke of it at various

[4]Mitochondria are the tiny intracellular "engines" within every eukaryotic cell that produce ATP, the energy currency. They have their own genomes.

[5]The gene unit itself, being a string, has an orientation.

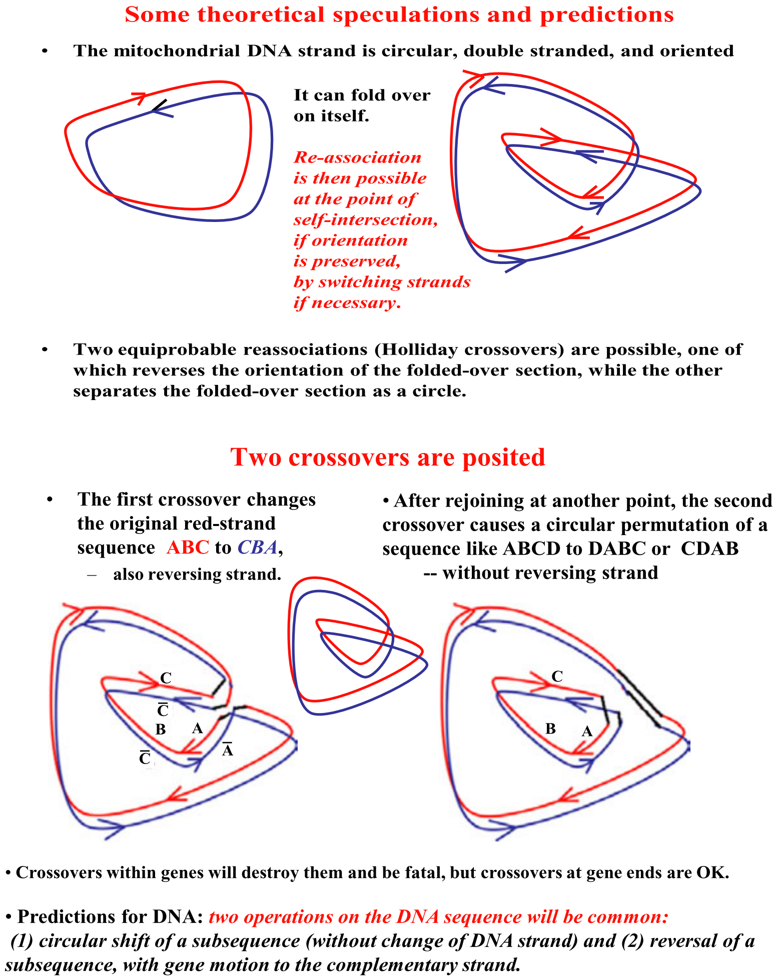


Fig. 1 Recombination in Mitochondrial circular genomes as inferred from gene order mutation. From Jack's 16th and 17th slide at his IBM talk (see footnote 6)

meetings on computational biology to which he was invited.[6] With hindsight, and the assistance of Google, I recently discovered a much earlier paper that had examined mitochondrial gene order as a means to compute phylogeny that had used far fewer genomes [5]. As far as I know, Jack's insight into the crustaceans was new, as was the precision he brought to the problem.

[6]See for example Jack's slide talk at IBM. http://roma.cshl.edu/jack. Recently, I looked again at the difficulty of computing precise distances between species, measured as the number of applications of Jack's rule. I now concur with his opinion about its difficulty.

4 The Clocks of Evolution

Jack discovered other methods of phylogenetic mapping. One was based on the conservation of the intron-exon structure of transcripts (see footnote 3). Overall, his various methods can be viewed as clocks that run on different time scales, as do the minute and hour hands of a clock. One hand might be useful for measuring fast events, and the other for slow events. What would it say for our confidence in phylogeny if these clocks depicted different trees? Or if the branches were of different duration, depending on the clock used to measure it? The synthesis of these ideas into a coherent method never occurred under Jack's watch. He was more interested in the thrill of a logically solid novel insight than in exhausting its implications. But these ideas will eventually be useful in understanding cancer evolution, where the rates of different types of genome instability (point mutation, rearrangements, and changes in copy number) become uncoupled, and each can be measured relative to each other, and to the absolute rate of mutation in normal cells. The rates of these clocks in individual tumors are likely to predict patient survival (see for example [4]).

5 Illusions

Jack had a keen interest in human cognition, especially vision. This choice was dictated by several considerations: Jack was gifted in visualization, he had mastery of the design of computer images, and he had himself as a willing subject. The five areas he explored were illusions of motion, distortion in parallelism, and depth perception, as well as the ability to distinguish textures and perceive objects against various backgrounds (see for example Fig. 2). He was especially adept at crafting three dimensional illusions, provided the user wore red-green colored glasses. He found how to make a static object on the computer screen glisten with metallic sheen by separately controlling left and right visual inputs. In all these instances he was able to use his command of the computed image to explore how the illusions fared under manipulation of the underlying parameters (such as contrast, graininess or rotation). He had little difficulty recruiting other willing subjects, and used this opportunity to probe human visual capability. To his surprise, he found some robust differences in the human responses. He never took this to the level of exploring patho-physiology, for example testing stroke victims or patients suffering from visual migraines, or individual genetic variation. This would have required far greater resources than was available to him. To my knowledge Jack never published his methods or observations.[7] Overall, Jack believed that vision was a composition of discrete hardwired

[7]Some of Jack's creations can be found at http://www.settheory.com/Glass_paper/cafe_wall_study.html, http://www.multimedialibrary.com/Articles/Jack/Illusions/index.asp, http://www.multimedialibrary.com/education/illusions/index.asp, http://www.settheory.com/hole_illusion/dill.html.

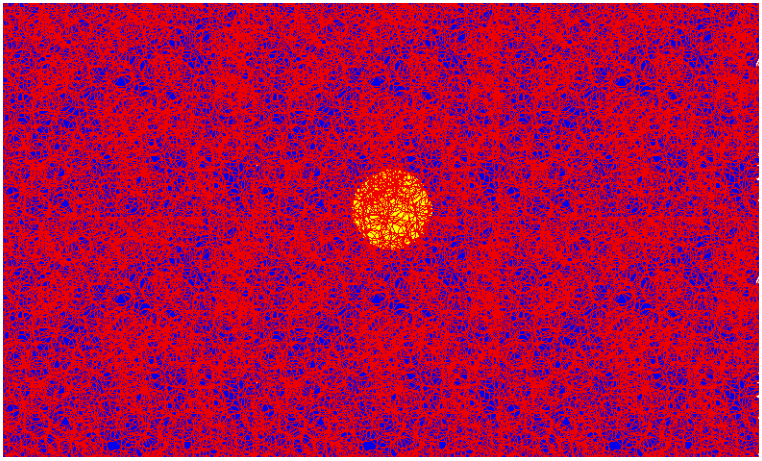


Fig. 2 Jacks's hole illusion. The *yellow* disk can be variously seen as floating or as a hole, or coplanar with the *blue* background, depending on the viewer's cognitive apparatus or disposition. See http://www.settheory.com/hole_illusion/dill.html (Color figure online)

processing subprograms. The discreteness is not ordinarily apparent to the user. For any new space of operation, he sought a "basis" of elementary operations, which when combined spanned the full space. I think this is what he was attempting to do in vision.

6 Memory

The organization of memories was a frequent subject of our discussions, although neither of us knew neurobiology. Jack would of course make use of his knowledge of computer architecture and algorithms to reason about the cognitive processes of the brain. A few of his observations had strong resonance with me. Here is the most important one. Human memory has a feature, as he put it, similar to "filename completion". If a Unix user specifies part of a filename, the rest of the filename is suggested by the computer system. Jack was in fact working on algorithms at the time that did much more than this: produce a list of candidate names given only a part of the last name of the person, perhaps the street name of their address, and perhaps a fragment of their home phone number (such as an area code plus a few digits). How would you organize your contact databases to execute such a task? Humans cannot do this particular task very well, but we do other things like it very well: "Who is the young actress, very pretty, the heroine in one of Woody Allen's murder mysteries and the victim in another?" Jack thought this was a fundamental property of memory.

In fact, it is easier to "imagine" how to set up such memory out of brain stuff than out of programmable silicon circuits. First, the "clues" to the full memory are disseminated broadly throughout the brain, recruiting various neurons[8] reactive to the clues. A pre-existant but dormant original circuit, created by prior experience, connects a subset of these recruited neurons along with many others, not yet activated. This circuit becomes activated by the initial recruits, firing up the other neurons of the original circuit that were not in the initially recruited set. The "name" or missing parts of the full memory reside in the full circuit. In other words, a neural circuit that can keep a set of member neurons stably stimulated, and be driven into that stable firing state by activation of only a few of its members, seems all that is needed for associative memory. The creation and management of such highly connected circuits does not have parallels in common computer architecture. Jack chose not to think too deeply about this, because his years as Director at DARPA/ISTO funding research[9] into neural nets left an indelibly negative impression. He was well aware of the gulf between how Nature conducts herself and how humans imagine she does.

7 The End Game

Jack remained very active after retirement. Aside from the mental preoccupations described here, Jack took an active interest in defense against bioterrorism and organized meetings on the subject. He developed a project to enhance the mathematical curriculum for gifted high school teachers and their students. He funded it himself, and scripted easily comprehensible and web accessible visual proofs of important theorems in mathematics, such as the fundamental theorem of algebra.

Jack loved Cold Spring Harbor, with its invigorating empirical tradition and its natural beauty, but his sabbatical ended upon his first serious illness. Once he recognized that his time was limited, he returned home to the city, teeming with the ethnic restaurants he adored, to the project on the application of computational logic to the foundations of mathematics. He maintained his contacts with his colleagues, and continued to visit from time to time, as I did him, often bringing vexing problems or new results. I love mathematics, and Jack loved biology, so we happily exchanged roles as teacher and student and kept each other amused. We discussed a myriad of subjects: cellular architecture, recombinant DNA, cloning, principles of experimental design, olfaction, transcriptional networks, cancer biology, Galois theory, the nature of proofs, the relationship of science to mathematics, the limits of the latter, computer architecture, programming languages, principles of algorithm design, and the uses of computers in musical annotation[10] among others. I still use Jack's lessons in my everyday professional research.

---

[8]Roughly speaking, a neuron can be resting and have a low rate of firing, or be in an active state with a high rate of firing.

[9]Jack summarized his experience as Director this way: "The fastest way to make enemies is to give away money."

[10]See for example http://www.settheory.com/drum_machine/drums.html.

I do not understand fully the intellectual journey he took from math to computation to empiricism. Jack was in some ways a thrill seeker, although the thrills had to have very great logical rigor. He loved to challenge his intellect in new ways, and I assume math failed to do this eventually. As related by his granddaughter, Adrienne Fainman, during her moving eulogy at Jack's memorial, Jack would comfort her when she was studying math with these words: be happy when you are confused, because you are poised to learn something. Jack was fascinated and endlessly surprised by the universe, of which life was the crowning part. Logic and computation were his sensory organs and muscles by which he grasped the things around him.

Jack held that a mathematician's emotional age, often measured in single digits or low teens, was set by the year in which he discovered his intellectual gifts. But Jack escaped this tyranny of arrested development. His intellect was matched by a graceful emotional maturity and an almost serene detachment. An indifferent shrug was the most he would concede to a temporary setback. He lived so as to be free of the folly of others. He neither internalized nor identified with such, and happily dealt with his own. He drew his ability to grow from his curiosity, his autonomous powers for discovery, and the joy he took in sharing his thoughts and findings with others. By such means, and with the support of a loving wife and sister, and highly appreciative friends and other relatives, he experienced a unique and richly rewarding life.

Acknowledgements I thank Diana Robinson Schwartz for her help with Jack's archives, for stimulating my memory, and for her patience. She and Bud Mishra helped with editorial comments. Dennis Sullivan and Jeff Cheeger helped me to understand the depth of the mathematics implicit in mitochondrial phylogeny.

References

1. Burrows, M., Wheeler, D.J.: A block sorting lossless data compression algorithm. Technical report 124, Digital Equipment Corporation, Palo Alto, CA (1994)
2. Ferragina, P., Manzini, G.: Opportunistic data structures with applications. In: 41st IEEE Symposium on Foundations of Computer Science, pp. 390–398 (2000)
3. Healy, J., Thomas, E., Schwartz, J.T., Wigler, M.: Annotating large genomes with exact word matches. Genome Res. **13**, 2306–2315 (2003)
4. Hicks, J., Krasnitz, A., Lakshmi, B., Navin, N., Riggs, M., Leibu, E., Esposito, D., Alexander, J., Troge, J., Grubor, V., Yoon, S., Wigler, M., Ye, K., Børresen-Dale, A.-L., Naume, B., Schlicting, E., Norton, L., Hagerstrom, T., Skoog, L., Auer, G., Maner, S., Lundin, P., Zetterberg, A.: Novel patterns of genomic rearrangement and their association with survival in breast cancer. Genome Res. **16**, 1465–1479 (2006)
5. Sankoff, D., Leduc, G., Antoine, N., Paquin, B., Lang, B.F., Cedergren, R.: Gene order comparisons for phylogenetic inference: evolution of the mitochondrial genome. Proc. Natl. Acad. Sci. USA **89**(14), 6575–6579 (1992)

Jacob T. Schwartz—Selected Bibliography

1955

1. De Rham's theorem for arbitrary spaces. Am. J. Math. **77**, 29–44

1956

2. Convergence almost everywhere of operator averages (with N. Dunford). J. Ration. Mech. **5**, 129–178
3. On Mautner's eigenfunction expansion (with W.G. Bade). Proc. Nat. Acad. Sci. **42**, 519–525

1958

4. Linear Operators: vol. I. General Theory (with Nelson Dunford). Interscience Publishers. Reprinted Wiley (1988)

1962

5. Lectures on the Mathematical Method in Analytical Economics. Gordon and Breach, New York

1963

6. Linear Operators: vol. II. Spectral Theory (with Nelson Dunford). Interscience Publishers. Reprinted Wiley (1988)

M. Davis, E. Schonberg (eds.), *From Linear Operators to Computational Biology*,
DOI 10.1007/978-1-4471-4282-9, © Springer-Verlag London 2013

7. Two finite, non-hyperfinite, non-isomorphic factors. Commun. Pure Appl. Math. **16**, 19–26
8. Non-isomorphic factors of type III. Commun. Pure Appl. Math. **16**, 111–120
9. Generators of the convolution algebra L_1 (with H.S. Shapiro and D.J. Newman). Trans. Am. Math. Soc. **107**, 466–484

1966

10. Large parallel computers. J. Assoc. Comput. Mach. **13**, 25–32

1967

11. W^* Algebras. Gordon and Breach, New York

1969

12. Programming Languages and Their Compilers (with John Cocke). Courant Institute

1970

13. Linear Operators: vol. III. Spectral Operators (with Nelson Dunford). Interscience Publishers. Reprinted Wiley (1988)

1975

14. Automatic data structure choice in a language of very high level. Commun. Assoc. Comput. Mach. **18**, 722–728

1979

15. Metamathematical extensibility for theorem verifiers and proof-checkers (with M. Davis). Comput. Math. Appl. **5**, 217–230

1980

16. Fast probabilistic algorithms for verification of polynomial identities. J. Assoc. Comput. Mach. **27**, 701–717
17. Decision procedures for elementary sublanguages of set theory, part I: multi-level syllogistic and its extensions (with A. Ferro and E. Omodeo). Commun. Pure Appl. Math. **33**, 599–608

1981

18. Decision procedures for elementary sublanguages of set theory, part II: a decidable class of quantified formulae (with A. Ferro, M. Breban, and E. Omodeo). Commun. Pure Appl. Math. **34**, 177–195

1982

19. Networks and algorithms for large-scale parallel computation (with A. Gottlieb). IEEE Comput. Mag. **15**, 27–36

1983

20. On the piano movers' problem: I. The special case of a rigid polygonal body moving amidst polygonal barriers (with M. Sharir). Commun. Pure Appl. Math. **36**, 345–398

1984

21. On the piano movers' problem: IV. The case of a rod moving in three-dimensional space amidst polyhedral obstacles (with M. Sharir). Commun. Pure Appl. Math. **37**, 815–848
22. On the complexity of motion planning for multiple independent objects; PSPACE hardness of the 'Warehouseman's Problem' (with J.E. Hopcroft, and M. Sharir). Int. J. Robot. Res. **3**, 76–88

1985

23. Efficient detection of intersections among spheres (with J.E. Hopcroft, and M. Sharir). Int. J. Robot. Res. **4**, 77–80

1986

24. Higher Level Programming: An Introduction to the Set-Theoretic Language SETL (with E. Schonberg, E. Dubinsky, and R. Dewar). Springer-Verlag

1995

25. T-theorem proving I (with Alberto Policriti). J. Symb. Comput. **20**, 315–342

Index

M. Davis, E. Schonberg (eds.), *From Linear Operators to Computational Biology*,
DOI 10.1007/978-1-4471-4282-9,