Index

[Var88] M. Vardi. Fundamentals of dependency theory. In E. Börger, editor, *Trends in Theoretical Computer Science*, chapter 5. Computer Science Press, 1988.

[Wai70] S.S. Wainer. A classification of ordinal recursive functions. *Archiv für Mathematische Logik und Grundlagen der Mathematik*, 13:136–153, 1970.

[Wil81] R.L. Wilder. *Mathematics as a Cultural System.* Pergamon Press, 1981.

[Zie82] Martin Ziegler. Einige unentscheidbare Körpertheorien. In V. Strassen E. Engeler, H. Läuchli, editor, *Logic and Algorithmic, An international Symposium held in honour of E. Specker*, pages 381–392. L'enseignement mathématique, 1982.

[She90] S. Shelah. *Classification Theory and the number of non–isomorphic models.* Studies in Logic and the Foundations of Mathematics. North Holland, 2 edition, 1990.

[She91] Enshao Shen. Manuscript. unpublished, Department of Mathematics, University of Freiburg, Germany, 1991.

[Sho67] J. Shoenfield. *Mathematical Logic.* Addison-Wesley Series in Logic. Addison-Wesley, 1967.

[SM81] E. Servi-Machlin. *The Classical Cuisine of the Italian Jews.* Everest House, 1981.

[Smo80] C. Smorynski. Some rapidly growing functions. *Mathematical Intelligencer*, 2:149–154, 1980.

[Smo83] C. Smorynski. Big news from Archimedes to Friedman. *Notices of the AMS*, 30:251–256, 1983.

[SSH87] J.T. Schwartz, M. Sharir, and J. Hopcroft, editors. *Planning, Geometry, and Complexity of Robot Motion.* Ablex Series in Artificial Intelligence. Ablex Publishing Corporation, 1987.

[Ste10] E. Steinitz. Algebraische Theorie der Körper. *Journal fúr reine und angewandte Mathematik*, 137:167–309, 1910.

[Ste91a] I.A. Stewart. Comparing the expressibility of languages formed using np–complete operators. *Journal of Logic and Computation*, 1(3):305–330, 1991.

[Ste91b] I.A. Stewart. On completeness for NP via projection translations. to appear in CSL'91, 1991.

[Tra50] B. Trakhtenbrot. On the impossibility of an algorithm for the decision problem in finite domains. *Doklady Akademy Nauk SSSR*, 70:569–572, 1950.

[Tra61] B. Trakhtenbrot. Finite automata and the logic of monadic predicates. *Doklady Akademy Nauk SSSR*, 140:326–329, 1961.

[TS88] R.W. Topor and E.A. Sonenberg. On domain independent databases. In J. Minker, editor, *Foundations of deductive data bases and logic programming*, chapter 6. Morgan Kaufmann Publishers, 1988.

[Ull82] J.D. Ullman. *Principles of Database Systems.* Principles of Computer Science Series. Computer Science Press, 2 edition, 1982.

[MV86] J.A. Makowsky and M. Vardi. On the expressive power of data dependencies. *Acta Informatica*, 23.3:231–244, 1986.

[MZ80] J. A. Makowsky and M. Ziegler. Topological model theory with an interior operator: Consistency properties and back and forth arguments. *Archiv für mathematische Logik und Grundlagenforschung*, 20:37–54, 1980.

[Pas90] A. Pasztor. Recursive programs and denotational semantics in absolut logics of programs. *Theoretical Computer Science*, 70:127–150, 1990.

[Res] J.P. Ressayre. Formal languages defined by the underlying structure of their words, part II. to appear in the Journal of Symbolic Logic.

[Res88] J.P. Ressayre. Formal languages defined by the underlying structure of their words. *Journal of Symbolic Logic*, 53(4):1009–1026, 1988.

[Ris69] R.H. Risch. The problem of integration in finite terms. *Transactions of the American Mathematical Society*, 139:167–189, 1969.

[Ros72] M. Rosenlicht. Integration in finite terms. *American Mathematical Monthly*, 79(9):963–972, 1972.

[Ros82] J.G. Rosenstein. *Linear orderings*. Academic Press, 1982.

[RS23] M. Ryan and M. Sadler. Valuation systems and consequence relations. In D. Gabbay and al., editors, *Handbook of Logic in Computer Science*. Oxford University Press, 1992/3.

[Ser71] M. Servi. Una questione die teoria dei modelli nelle categorie con prodotti finiti. *Matematiche*, 26:307–324, 1971.

[She84] J.C. Shepherdson. Negation as failure: A comparison of clark's completed data base and reiter's closed world assumption. *Journal of Logic Programming*, 1:51–81, 1984.

[She85] J.C. Shepherdson. Negation as failure II. *Journal of Logic Programming*, 3:185–202, 1985.

[She88] J.C. Shepherdson. Negation in logic programming. In J. Minker, editor, *Foundations of deductive data bases and logic programming*, chapter 1. Morgan Kaufmann Publishers, 1988.

[Mak85] J.A. Makowsky. Compactness, embeddings and definability. In *Model-Theoretic Logics*, Perspectives in Mathematical Logic, chapter 18. Springer Verlag, 1985.

[Mak87] J. A. Makowsky. Why Horn formulas matter for computer science: Initial structures and generic examples. *Journal of Computer and System Sciences*, 34.2/3:266–292, 1987.

[Mak88] J.A. Makowsky. Mental images and the architecture of concepts. In R. Herken, editor, *The Universal Turing Machine. A Half–Century Survey.* Oxford University Press, 1988.

[Mak92a] J.A. Makowsky. Database theory. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 5. Oxford University Press, 1992.

[Mak92b] J.A. Makowsky. The impact of model theory on theoretical computer science. To appear in the Proceedings of the 9th International Congress of Logic, Methodology and Philosophy of Science, Uppsala 1991, 1992.

[Mat70] Ju. V. Matijasevič. Enumerable sets are diophantine. *Sov. Math. Dokl.*, 11:354–357, 1970.

[MM77] J.A. Makowsky and A. Marcja. Completeness theorems for modal model theory with the Montague–Chang semantics, I. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 23:97–104, 1977.

[MM84] B. Mahr and J.A. Makowsky. Characterizing specification languages which admit initial semantics. *Theoretical Computer Science*, 31:49–60, 1984.

[MM85] J.A. Makowsky and D. Mundici. Abstract equivalence relations. In *Model-Theoretic Logics*, Perspectives in Mathematical Logic, chapter 19. Springer Verlag, 1985.

[Mos74] Y. Moschovakis. *Elementary Induction on Abstract Structures.* Studies in Logic, vol 77. North–Holland, 1974.

[MS89] J. A. Makowsky and I. Sain. Weak second order characterizations of various program verification systems. *Theoretical Computer Science*, 66:299–321, 1989.

[MS92] T. Maibaum and M. Sadler. Axiomatizing specification theory. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2. Oxford University Press, 1992.

[Imm88] N. Immerman. Nondeterministic space is closed under complement. *SIAM Journal on Computing*, 17:935–938, 1988.

[Kan90] P.C. Kannelakis. Elements of relational database theory. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume 2, chapter 17. Elsevier Science Publishers, 1990.

[KK67] G. Kreisel and J.L. Krivine. *Elements of Mathematical Logic: Model Theory*. North Holland, 1967.

[KK82] S. Kochen and S. Kripke. Non–standard models of Peano arithmetic. In V. Strassen E. Engeler, H. Läuchli, editor, *Logic and Algorithmic, An international Symposium held in honour of E. Specker*, pages 275–296. L'enseignement mathématique, 1982.

[KM87] A. Kanamori and K. McAloon. On Gödel incompleteness and finite combinatorics. *Annals of Pure and Applied Logic*, 33:23–41, 1987.

[Kol90] P.G. Kolaitis. Implicit definability on finite structures and unambiguous computations. In *FOCS'90*, pages 168–180. IEEE, 1990.

[KP82] L. Kirby and J. Paris. Accessible independence results for Peano arithmetic. *Bulletin of the London Mathematical Society*, 14:285–293, 1982.

[Kre52] G. Kreisel. On the concepts of completeness and interpretation of formal systems. *Fundamenta Mathematicae*, 39:103–127, 1952.

[Lad77] R.E. Ladner. Apllication of model theoretic games to discrete linera orders and finite automata. *Information and Control*, 33:281–303, 1977.

[Lyn82] J.F. Lynch. Complexity classes and theories of finite models. *Mathematical Systems Theory*, 15:127–144, 1982.

[Mak81] J.A. Makowsky. Characterizing database dependencies. In *ICALP'81*, volume 115 of *Lecture Notes in Computer Science*, pages 86–97. Springer Verlag, 1981.

[Mak84] J.A. Makowsky. Model theoretic issues in theoretical computer science, part I: Relational databases and abstract data types. In G. Lolli and al., editors, *Logic Colloquium '82*, Studies in Logic, pages 303–343. North Holland, 1984.

[GS87] Y. Gurevich and S. Shelah. Expected computation time for hamiltonian path problem. *SIAM Journal for Computing*, 16.3:486–502, 1987.

[GTWW77] J. Goguen, J.W. Thatcher, E.G. Wagner, and J.B. Wright. Initial algebra semantics and continuous algebras. *Journal of ACM*, 24:68–95, 1977.

[Gur90] Y. Gurevich. On finite model theory. In S.R. Buss and P.J. Scott, editors, *Perspectives in Computer Science*. Birkhäuser Verlag, 1990.

[Gur91] Y. Gurevich. Average case completeness. *Journal of Computer and System Sciences*, 42:346–398, 1991.

[Har84] D. Harel. Dynamic logic. In D. Gabbay and F. Günthner, editors, *Handbook of Philosophical Logic*, volume 2, chapter 10. D. Reidel Publishing Company, 1984.

[Hei83] J. Heintz. Definability and fast quantifier elimination in algebraically closed fields. *Theoretical Computer Science*, 24:239–277, 1983.

[Hen77] L. Henkin. The logic of equality. *American Mathematical Monthly*, 84:597–612, 1977.

[HMvS85] L.A. Harrington, M.D. Morley, A. Ščedrov, and S.G. Simpson, editors. *Harvey Friedman's Research in the Foundations of Mathematics*, volume 117 of *Studies in Logic and the Foundations of Mathematics*. North Holland, 1985.

[Hod92] W. Hodges. Logical features of horn clauses. In D. Gabbay and al., editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*. Oxford University Press, 1992.

[HU80] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley Series in Computer Science. Addison-Wesley, 1980.

[Ier89] D. Ierardi. Quantifier elimination in the theory of an algebraically closed field. In *STOCS'89*, pages 138–147. ACM, 1989.

[Imm87] N. Immerman. Languages that capture complexity classes. *SIAM Journal on Computing*, 16(4):760–778, Aug 1987. Also appeared as a preliminary report in Proceedings of the 15th Annual ACM Symposium on the Theory of Computing.

[Eng81] E. Engeler. Generalized Galois theory and its application to complexity. *Theoretical Computer Science*, 13:271–293, 1981.

[Fag74] R. Fagin. Generalized first-order spectra and polynomial time recognizable sets. In R. Karp, editor, *Complexity of Computation*, pages 27–41. American Mathematical Society Proc, 7, Society for Industrial and Applied Mathematics, 1974.

[Fag82] R. Fagin. Horn clauses and database dependencies. *Journal of ACM*, 29.4:952–985, 1982.

[Fag90] R. Fagin. Finite model theory - a personal perspective. In S. Abiteboul and P.C. Kannelakis, editors, *ICDT'90*, volume 470 of *Lecture Notes in Computer Science*, pages 3–24. Springer Verlag, 1990.

[FLO83] S. Fortune, D. Leivant, and M. O'Donnel. The expressiveness of simple and second–order type structures. *Journal of ACM*, 30.1:151–185, 1983.

[Flu91] J. Flum. On bounded theories. to appear in CSL'91, 1991.

[FR79] J. Ferrante and C.W. Rackoff. *The Computational Complexity of Logical Theories*, volume 718 of *Lecture Notes in Mathematics*. Springer Verlag, 1979.

[Ga92] D. Gabbay and al., editors. *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 1-4. Oxford University Press, 1992.

[Gai82] Haim Gaifman. On local and non-local properties. In J. Stern, editor, *Logic Colloquium '81*, pages 105–135. North-Holland publishing Company, 1982.

[Gal91] J.H. Gallier. What is so special about Kruskal's theorem and the ordinal γ_0 ? A survey of some results in proof theory. *Annals of Pure and Applied Logic*, 53:199–260, 1991.

[GJ79] M.G. Garey and D.S. Johnson, editors. *Computers and Intractability*. Mathematical Series. W.H. Freeman and Company, 1979.

[GPSS80] D. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. On temporal analysis of fairness. In *Proceedings of the 7th ACM-POPL*, pages 163–173, 1980.

[Gra83] E. Grandjean. Complexity of the first–order theory of almost all structures. *Information and Control*, 52:180–204, 1983.

[CH80] Ashok K. Chandra and David Harel. Computable queries for relational data bases. *Journal of Computer and System Sciences*, 21(2):156–178, Oct 1980.

[Cho88] Shang-Ching Chou. *Mechanical Geometry Theorem Proving*. Mathematics and its Applications. D. Reidel Publishing Company, 1988.

[CK90] C.C. Chang and H.J. Keisler. *Model Theory*. Studies in Logic, vol 73. North–Holland, 3rd edition edition, 1990.

[CM91] A. Calò and J.A. Makowsky. The Ehrenfeucht–Fraïssé games for transitive closure. to appear in the Proceedings of LFCS'92: Logic at Tver, LNCS 1992, 1991.

[Com87] K.J. Compton. A logical approach to asymptotic combinatorics I: First–order properties. *Advances in Mathematics*, 65:65–96, 1987.

[Cou90a] B. Courcelle. Graph rewriting: An algebraic approach. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume 2, chapter 5. Elsevier Science Publishers, 1990.

[Cou90b] B. Courcelle. The monadic second–order theory of graphs I: Recognizable sets of finite graphs. *Information and Computation*, 85:12–75, 1990.

[Dah82] E. Dahlhaus. *Combinatorial and Logical Properties of Reductions to some Complete Problems in NP and NL*. PhD thesis, Technische Universität Berlin, Germany, 1982.

[Dah83] E. Dahlhaus. Reductions to NP–complete problems by interpretations. In E. Börger et. al., editor, *Logic and Machines: Decision Problems and Complexity*, volume 171 of *Lecture Notes in Computer Science*. Springer Verlag, 1983.

[dR87] M. de Rougemont. Second–order and inductive definability on finite structures. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 33:47–63, 1987.

[Ebb85] H.D. Ebbinghaus. Extended logics: The general framework. In *Model-Theoretic Logics*, Perspectives in Mathematical Logic, chapter 2. Springer Verlag, 1985.

[Eme90] E.A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume 2, chapter 16. Elsevier Science Publishers, 1990.

References

[AF90] M. Ajtai and R. Fagin. Reachability is harder for directed than for undirected finite graphs. *Journal of Symbolic Logic*, 55.1:113–150, 1990.

[ANS82] H. Andréka, I. Nemeti, and I. Sain. A complete logic for reasoning about programs via non–standard model theory, parts I and II. *Theoretical Computer Science*, 17:193–212 and 259–278, 1982.

[B̈60] J.R. Büchi. Weak second–order arithmetic and finite automata. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 6:66–92, 1960.

[Bak75] A. Baker. *Transcendental Number Theory*. Cambridge University Press, 1975.

[Bak84] A. Baker. *A Concise Introduction to the Theory of Numbers*. Cambridge University Press, 1984.

[BDD91] S. Ben-David and M. Dvir. Non–standard models for independent arithmetical statements. Technical report, Technion–Israel Institute of Technology, Haifa, Israel, 1991.

[BDH91] S. Ben-David and S. Halevi. On the independence of P versus NP. Technical report, Technion–Israel Institute of Technology, Haifa, Israel, 1991.

[BF85] J. Barwise and S. Feferman, editors. *Model-Theoretic Logics*. Perspectives in Mathematical Logic. Springer Verlag, 1985.

[BS84] R.A. Bull and K. Segerberg. Basic modal logic. In D. Gabbay and F. Günthner, editors, *Handbook of Philosophical Logic*, volume 2, chapter 1. D. Reidel Publishing Company, 1984.

[BSTW85] A. Baudisch, D. Seese, P. Tuschik, and M. Weese. Decidability and quantifier–elimination. In *Model-Theoretic Logics*, chapter 7. Springer Verlag, 1985.

[Bur84] J. Burgess. Basic tense logic. In D. Gabbay and F. Günthner, editors, *Handbook of Philosophical Logic*, volume 2, chapter 2. D. Reidel Publishing Company, 1984.

[Cal89] A. Caló. The expressive power of transitive closure. M.Sc. Thesis, Faculty of Computer Science, Technion–Israel Institute of Technology, Haifa, Israel, 1989.

Marcja. We then worked together on the model theory of modal logic and wrote [MM77] Some time ago you could still find street vendors in winter selling castagnaccio which was baked on an open fire. The ingredients are simple: Chestnut flour, olive oil, sugar, raisins and pinenuts. The original receipe does not appear in Artusi's classical cooking book, but it does appear in the very enjoyable and nostalgic book by E. Servi–Machlin [SM81]. Incidentally, she is the sister of the Italian logician M. Servi who did some work on the model theory of categories with finite products [Ser71]. As it is difficult to find chestnut flour outside of Tuscany, let me give a modified version of the dish.

Take one can of canned, unsweetend chestnut purée. Add a third of the can of olive oil and half the can of sugar and mix in a food processor to a homogeneous paste. Add a third of the can of regular flour and keep mixing till smooth.

Spread the paste on a flat cooking sheet, not more than a small finger thick. It is advisable to rub the sheet with margarine or olive oil before spreading the paste. Now stick the pinenuts and raisins into the paste at your liking. Bake medium hot for about 40 minutes (till the paste hardens) but be careful not to burn it.

Let cool and break into pieces. Best served with coffee (expresso) and Grappa (di Brunello).

8.13: Corollary
$HALFCLIQUE$ and HAM are not definable in $\mathbf{DTC^1}, \mathbf{TC^1}, \mathbf{ATC^1}$, the First Order Logics augmented by the predicate transformers DTC, TC, ATC of size 1.

This gives us a quick example for interesting families of pairs of ordered finite structures which are n-isomorphic in $\mathbf{TC}^1$.

8.14: Corollary
There are functions $f, g : \mathbf{N} \mapsto \mathbf{N}$ such that for every $n \in \mathbf{N}$ $f(n) \neq g(n)$ and the words $a^{f(n)}b^{f(n)}$ and $a^{f(n)}b^{g(n)}$ are n-isomorphic (equivalent) in $\mathbf{TC}^1$.

In [dR87] it is only proved that HAM is not definable in existential Monadic Second Order Logic.

8.5 The Games and Pumping Lemmas

The various games described in this section are used to show non–definability results. Corollary 8.14 is a model theoretic version of the well known Pumping–Lemma for regular languages. Theorem 8.11 is a model theoretic version of a Pumping–Lemma for **L**. Similar Pumping–Lemmas can be formulated for other complexity classes **C**, provided **C** is captured by some logic, for which there are suitable games. It remains a challenging open problem, how to exploit this observation. Ressayre has done a first step into this direction [Res88, Res].

9 Conclusions

We, that is if the reader is still with me, have travelled through the landscape of Model Theory coming from the land of Theoretical Computer Science. There are many places we have not visited, and even where we did, we did not explore them enough. We have seen some of the land's history and cultural system and hinted at its connections with Database Theory and Logic Programming. We have explored its deeper connections to Computer Aided Geometry. But we have spent most of our time exploring the model theoretic aspects of the question $\mathbf{P} \neq \mathbf{NP}$ and its provability in formal systems of arithmetic.

The chapter is called an 'Appetizer'. It should tempt more than one reader to travel more. If your appetite has been whetted, let me invite you for a treat:

Castagnaccio: This is a popular sweet in Tuscany of which I have learnt during my own travels in the land of Model Theory (and Tuscany) from A.

8.12: Theorem
$HALFCLIQUE$ and HAM are not definable in Monadic Second Order Logic (with arbitrary alternation of quantifiers) and hence not definable in $\mathbf{DTC^1}, \mathbf{TC^1}, \mathbf{ATC^1}$.

Proof. We present here a new and quick but surprising proof of the above result, cf. [Mak92b]. Recall that Büchi's theorem states that a language L is regular (=recognizable by a finite automaton) iff the set L of words considered as first order structures with a linear order is definable by a formula in Monadic Second Order Logic. Note that the proof of Büchi's theorem in [Lad77] also uses Ehrenfeucht-Fraïssé games.

The proof has several stages:

(i) First we note that the language $\{a^n b^n : n \in \mathbf{N}\}$ is not regular, cf. [HU80]. This is usually proved by the Pumping Lemma, which is in some sense an automata theoretic counterpart of the Ehrenfeucht-Fraïssé games.

(ii) Next we use Büchi's theorem and conclude that the set of words corresponding to $a^n b^n$ is not definable in Monadic Second Order Logic.

(iii) Note that a complete bipartite graph is Hamiltonian iff both sets have the same cardinality.

(iv) Now assume, for contradiction, that HAM were definable by a τ–formula ϕ of Monadic Second Order Logic. Let $w \in \{a, b\}^*$ be a word. We define a binary relation E_w on w by $(i, j) \in E_w$ iff $i \in P_a$ and $j \in P_b$. E_w makes w into a complete bipartite graph. E_w is definable by a first order formula θ over τ_{words}. Substituting R in ϕ by θ would give us a formula in Monadic Second Order Logic which assures that this graph is Hamiltonian, hence would define the language $\{a^n b^n\}$, a contradiction.

(v) For $HALFCLIQUE$ we proceed similarly. We first note that the language $HALF(a)$ where at least half the letters are a's, is not regular, again by the Pumping Lemma. Hence $HALF(a)$ is not definable in Monadic Second Order Logic.

(vi) We now define E_w by $(i, j) \in E_w$ iff $i \in P_a$ or $j \in P_a$. E_w makes w into a graph where the a's form a clique. Using this in the above argument completes the proof.

Note however, the specific sequence of pairs of ordered graphs G_n, H_n, which we construct to show this, can be separated by a $\mathbf{TC^2}$–formula, i.e. a formula with the predicate transformer TC of size 2. ∎

similarly motivated games are introduced for Fixed Point Logic which is, on ordered structures, of the same expressive power as **ATC**. However, the game introduced in [dR87] is only shown to be sound, which suffices for the non–definability result discussed there.

8.4 More Non–definability Results

We discuss here briefly the case of **TC**, the other cases being similar, but not trivially so. The games naturally induce a sequence of equivalence relations ${\equiv_n}^{TC}$ between structures which we call n–isomorphic for **TC**. In [Cal89] Calò proves soundness and completeness of these equivalence relations in the sense that two structures are n–isomorphic for **TC** iff they satisfy the same formulas of quantifier depth n. As all these logics contain predicate transformers of arbitrary arity k, the logic **TC** can be viewed as a sequence of logics $\mathbf{TC}^k$ with the arity of the predicate transformer not exceeding k. $\mathbf{TC}^1$ is the logic with the transitive closure applied only to binary relations of 1–tuples. Our games also allow to characterize definability in $\mathbf{TC}^k$.

In [Imm87] it is shown that a class of finite ordered structures is definable in these logics iff its recognizability is in certain complexity classes. In this sense we speak of logics capturing complexity classes and we have for ordered structures, by abuse of notation, $\mathbf{L} = \mathbf{DTC}$, $\mathbf{TC} = \mathbf{NL}$ and $\mathbf{ATC} = \mathbf{P}$. **NP** was shown in [Fag74] to capture existential Second Order Logic. As an application of our work we state a necessary and sufficient condition for separating the complexity classes **L, NL, P** and **NP** respectively which is of pure model theoretic character. In the case of $\mathbf{NL} \neq \mathbf{NP}$ this condition can be stated as follows:

Let $HALFCLIQUE$ be the set of ordered graphs which contain a clique of half its size. Let HAM be the set of ordered graphs which contain a hamiltonian path. Note that $HALFCLIQUE$ and HAM are **NP**–complete, cf. [GJ79].

8.11: Theorem
$\mathbf{NL} \neq \mathbf{NP}$ *iff there is a sequence of pairs of ordered graphs* G_n, H_n *such that*

(i) $G_n {\equiv_n}^{TC} H_n$ *and*

(ii) $G_n \notin HALFCLIQUE$ *but* $H_n \in HALFCLIQUE$.

The same holds for $HALFCLIQUE$ *replaced by* HAM *or any other* **NP**–*complete problem.*

The construction of such families of graphs may be very hard and possibly requires probabilistic methods similar to the ones used in [AF90]. The following result nevertheless sheds some light on the problem.

(ii) $\mathcal{A}$ and $\mathcal{B}$ are n-equivalent iff they satisfy the same formulas of quantifier depth n.

For many applications the following corollary is most useful:

8.9: Corollary
A class of finite τ-structures K, τ finite and without function symbols, is definable by a formula ϕ of quantifier depth n iff K is closed under n-equivalence.

Ehrenfeucht–Fraïssé–games can be used to obtain non–definability results. To prove them we combine theorem 8.8 and proposition 8.6.

8.10: Theorem
The following classes of finite structures are not first order definable:

(i) The class of connected graphs;

(ii) the class of planar graphs;

(iii) the class of hamiltonian graphs;

(iv) the class of finite linear orders with an even number of elements.

For a detailed discussion, cf. [Gai82, AF90].

8.3 Second Order Logic and its Sublogics

Ehrenfeucht–Fraïssé–games can be generalized and adapted for various extensions of first order logics. In the case of Second Order Logic (Monadic Second Order Logic), one simple adds a new type of moves where the players choose relations (unary relations). Additionally the winning condition has to be modified correspondingly. For the transitive closure logics **DTC** $= FOL(DTC)$, **TC** $= FOL(TC)$ and **ATC** $= FOL(ATC)$ introduced in [Imm87] and section 7, such games were explicitly studied in [Cal89, CM91]. The existence of similar games as introduced in this paper, already follows from successive papers cumulating in [MM85]. There, they are defined for logics with generalized quantifiers rather than predicate transformers. Recently, a logic equivalent to **TC** based on generalized quantifiers only was exhibited in [She91]. However, the explicit use of such games for the case of predicate transformers stemming from transitive closure operators was first introduced in [Cal89]. It has been motivated by but is distinctly different from the games introduced in [MZ80]. It should also be noted that our explicit definition of these games is more straightforward than their derivation in the framework of abstract model theory and generalized quantifiers as described in [BF85] and [MM85]. In [dR87] *(i)* $\mathcal{A}$ *is* n*-equivalent to* $\mathcal{B}$ *iff* $\mathcal{A}$ *is* n*-isomorphic to* $\mathcal{B}$*;*

(ii) $\mathcal{A}$ *is* n*-equivalent to* $\mathcal{B}$ *for every* $n \in \mathbf{N}$ *iff* $\mathcal{A}$ *is finitely isomorphic to* $\mathcal{B}$*;*

(iii) $\mathcal{A}$ *is* ∞*-equivalent to* $\mathcal{B}$ *iff* $\mathcal{A}$ *is partially isomorphic to* $\mathcal{B}$*.*

As an exercice, prove proposition 8.4 both, in the formalism of games and in the formalism of families of partial isomorphisms.

8.6: Proposition

(i) Let $\tau = \emptyset$*.* τ*-structures then consist of their universe only. Let* $\mathcal{A}$ *and* $\mathcal{B}$ *two* τ*-structures.* $\mathcal{A}$ *and* $\mathcal{B}$ *are* n*-equivalent iff either both have at most* n *elements and have the same number of elements or both have at least* n *elements.*
(Find the corresponding statement for linear orders).

(ii) Let τ *have no function symbols. There are for every* n *two* τ*-structures* $\mathcal{A}$ *and* $\mathcal{B}$ *such that* $\mathcal{A}$ *and* $\mathcal{B}$ *are* n*-equivalent,* $\mathcal{A}$ *has exactly* n *elements and* $\mathcal{B}$ *is infinite.*

(iii) Let τ *have exactly one binary relation symbol. There are for every* n *two* τ*-structures* $\mathcal{A}$ *and* $\mathcal{B}$ *such that* $\mathcal{A}$ *and* $\mathcal{B}$ *are* n*-equivalent, and* $\mathcal{A}$ *is a connected (planar, hamiltonian) graph and* $\mathcal{B}$ *is not connected (planar, hamiltonian).*

8.2 Completeness of the Game

We are now ready to formulate the connection between First Order Logic and the winning strategies for the game $\mathcal{E}_n^\tau(\mathcal{A}, \mathcal{B})$.

8.7: Notation
We denote by $FOL_{k,n}(\tau)$ *the set of* τ*-formulas with all its variables among* $v_1, \ldots, v_n$ *and all its free variables among* $v_{k+1}, \ldots, v_n$*.* $FOL_{n,n}$ *are the formulas without free variables of quantifier depth* n*.*

8.8: Theorem (Ehrenfeucht–Fraïssé)
Let τ *be without function symbols and finite. Let* $\mathcal{A}$ *and* $\mathcal{B}$ *be two* τ*-structures.*

Let $a_1, \ldots, a_{n-k}$ *and* $b_1, \ldots, b_{n-k}$ *be elements of* $\mathcal{A}$ *and* $\mathcal{B}$ *respectively.*

(i) Player II has a winning strategy for k *more moves in the game* $\mathcal{E}_n^\tau(\mathcal{A}, \mathcal{B})$ *starting in the position described by* $a_1, \ldots, a_{n-k}$ *and* $b_1, \ldots, b_{n-k}$ *iff* $\mathcal{A}$ *and* $\mathcal{B}$ *satisfy the same formulas of* $FOL_{k,n}(\tau)$*, where the variable* v_{k+m} *takes the value* a_{k+m} *or* b_{k+m} *respectively.*

After this playful definitions we are motivated enough to give a definition of n-equivalence of τ-structures in terms of partial functions. The winning strategies for $\mathcal{E}_n^\tau(\mathcal{A},\mathcal{B})$ will be described by a finite family of non-empty sets of partial isomorphisms $\{I_m\}_{m=0}^n$ in the following way.

8.2: Definition
Let $\mathcal{A}$ and $\mathcal{B}$ be two τ-strutures. We say that $\mathcal{A}$ is finitely isomorphic *to $\mathcal{B}$ if and only if there is a family $\{I_n\}_{n=0}^\infty$ of non empty sets of partial isomorphism from A to B such that:*

constants: *for every $f \in I_n$ and for every constant symbol $c \in \tau$, $I_A(c) \in Dom(f)$.*

(1) forth: *for every $f \in I_n$ and for every $a \in A$ there is a $g \in I_{n-1}$ extending f such that $a \in Dom(g)$.*

(1) back: *for every $f \in I_n$ and for every $b \in B$ there is a $g \in I_{n-1}$ extending f such that $b \in Im(g)$.*

The following are two special cases of this definition which are of particular interest:

8.3: Definition

(i) If $I_n = I$ constant for every $n \in N$, we say that $\mathcal{A}$ is partially isomorphic *to $\mathcal{B}$.*

(ii) If there is only a finite family $\{I_m\}_{m=0}^n$ of partial isomorphisms with the back and forth properties, we say that $\mathcal{A}$ is n-isomorphic *to $\mathcal{B}$.*

8.4: Proposition
Let $\mathcal{A}$, $\mathcal{B}$ be two τ-structures.

(i) If $\mathcal{A}$ and $\mathcal{B}$ are partially isomorphic then $\mathcal{A}$ and $\mathcal{B}$ are finitely isomorphic.

*(ii) (**Cantor**) If $\mathcal{A}$ and $\mathcal{B}$ are both countable and partially isomorphic then $\mathcal{A}$ and $\mathcal{B}$ are isomorphic.*

(iii) If $\mathcal{A}$, $\mathcal{B}$ are finite, then $\mathcal{A}$ is finitely isomorphic to $\mathcal{B}$ if and only if $\mathcal{A}$ is isomorphic to $\mathcal{B}$.

The relationship between the game and the families of partial isomorphisms is given in the following:

8.5: Proposition
Let $\mathcal{A}$ and $\mathcal{B}$ be two τ-structures.

- $\mathcal{E}_n^\tau(\mathcal{A}, \mathcal{B})$ is played by two players I and II, called *spoiler* and *duplicator* respectively. The game is played for n moves, where n is a natural number. The 'board' of the game consists of two τ-structures $\mathcal{A}$ and $\mathcal{B}$ with universe A and B respectively. Here we make a restriction on τ in as much as τ may not contain any function symbols.
- The moves of the game look as follows:
  In the k-th move, player I chooses one of the structures $\mathcal{A}$ (or $\mathcal{B}$) and an element $a_k \in A$ ($b_k \in B$) and player II replies by choosing an element $b_k \in B$ ($a_k \in A$).
- After the n moves they have chosen elements $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$ Player II has won, if the map $f(a_i) = b_i, i \leq n$ is a partial isomorphism between the τ-structures $\mathcal{A}$ and $\mathcal{B}$.
- We look also at the infinite game $\mathcal{E}_\infty^\tau(\mathcal{A}, \mathcal{B})$ which is defined similarly, but with the moves numbered n for every $n \in \mathbf{N}$.

The interesting case here is when player II has a *winning strategy*. To make this notion precise we need some notational effort. Intuitively, a winning strategy is a catalogue of moves which lists all possible moves of the opponent with at least one answer which will guarantee ultimately that player II wins. With this informal definition in mind, we proceed now further:

- Let $\mathcal{A}$ and $\mathcal{B}$ be two τ-structures. We say that $\mathcal{A}$ and $\mathcal{B}$ are n-equivalent (∞-equivalent) and write $\mathcal{A} \equiv_n \mathcal{B}$, if player II has a winning strategy in the game $\mathcal{E}_n^\tau(\mathcal{A}, \mathcal{B})$ ($\mathcal{E}_\infty^\tau(\mathcal{A}, \mathcal{B})$).
- The relation $\mathcal{A} \equiv_n \mathcal{B}$ between τ-structures is indeed an equivalence relation, i.e. we have:
  (i) $\mathcal{A} \equiv_n \mathcal{A}$;
  (ii) $\mathcal{A} \equiv_n \mathcal{B}$ iff $\mathcal{B} \equiv_n \mathcal{A}$, and
  (iii) If $\mathcal{A} \equiv_n \mathcal{B}$ and $\mathcal{B} \equiv_n \mathcal{C}$ then $\mathcal{A} \equiv_n \mathcal{C}$.

8.1: Examples

(i) If $\mathcal{A} \simeq \mathcal{B}$ then $\mathcal{A} \equiv_n \mathcal{B}$ for every $n \in \mathbf{N}$.

(ii) Let τ consist of one binary relation symbol and let $\mathcal{G}_n$ be the τ-structure which, viewed as a graph, is the complete graph on n elements, and let $\mathcal{H}_n$ be the τ-structure which, viewed as a graph, is the complete bipartite graph on n elements. Analyze for small l, m and n whether $\mathcal{G}_l \equiv_m \mathcal{H}_n$.

7.5 Logics Capturing Complexity Classes

The framework developed so far allows us to state a very general theorem about logics capturing complexity classes. Theorems of this type were first stated in this form by Immerman [Imm87], but were preceeded and motivated by Fagin's work [Fag74]. We first need a definition.

7.22: Definition
Let K be a class of natural structures which is $\mathbf{C}$–complete for $\mathcal{L}$ reductions. We denote by $\mathcal{L}(K)$ the logic obtained from $\mathcal{L}$ by adding the quantifiers (or predicate transformers) associated with K for all sizes $k \in \mathbf{N}$. If $\mathcal{L}$ is First Order Logic we write $FOL(K)$ instead of $\mathcal{L}(K)$.

7.23: Theorem
Let $\mathbf{C}$ be a complexity class which contains $\mathbf{L}$. Let $\mathcal{L}$ be a semi–regular $\mathbf{C}$–computable logic. Let K be a class of natural structures $\mathbf{C}$–complete for $\mathcal{L}$–reductions. Then $\mathcal{L}(K)$ captures $\mathbf{C}$.

7.24: Corollary (Immerman)

(i) $FOL(DTC)$ captures Logarithmic Space $\mathbf{L}$.

(ii) $FOL(TC)$ captures Nondeterministic Logarithmic Space $\mathbf{NL}$.

(iii) $FOL(ATC)$ captures Polynomial Time $\mathbf{P}$.

Recently, I. Stewart [Ste91a, Ste91b] has taken a similar, but less general approach to show that

7.25: Corollary (Stewart)
$FOL(3COL)$ captures Nondeterministic Polynomial Time $\mathbf{NP}$.

These results make the problems $\mathbf{L} \neq (?)\mathbf{NL}$, $\mathbf{P} \neq (?)\mathbf{NP}$ or even $\mathbf{L} \neq (?)\mathbf{NP}$ into problems of definability by computable quantifiers. We shall see in the next section how to convert them into problems of model theory.

8 Ehrenfeucht–Fraïssé Games

In this section we introduce a very powerful tool with many applications in the analysis of the expressive power of First Order Logic. The tool is commonly known under the name *Ehrenfeucht–Fraïssé Games* $\mathcal{E}_n^\tau(\mathcal{A}, \mathcal{B})$.

8.1 The Games

Informally the *Ehrenfeucht–Fraïssé* Games are described as follows:

An analogous definition can be given for predicate transformers instead of quantifiers.

7.19: Definition
Let K be a class of natural σ–structures. Let $\mathbf{C} \subseteq \mathbf{D}$ be complexity classes and let $\mathcal{L}$ be a $\mathbf{C}$–computable logic. We say that K is $\mathbf{D}$–complete for $\mathcal{L}$–reductions *if*

(i) K is in $\mathbf{D}$ and

(ii) every class K_1 of σ_1–structures in $\mathbf{D}$ is $\mathcal{L}$–reducible to K.

The traditional definition of **NP**–completeness says that K is **NP**–complete iff K is **NP**–complete for **P**–reductions. Our definition is model theoretic counterpart of the more usual computational definition of K being **D**–complete for **C**–reductions. The exact relationship between the two definitions is given by the following proposition and is proven using theorem 7.12 or theorem 7.17

7.20: Proposition
Let K be a class of natural σ–structures. Let $\mathbf{C} \subseteq \mathbf{D}$ be complexity classes and let $\mathcal{L}$ be a $\mathbf{C}$–computable logic.

(i) If K is $\mathbf{D}$–complete for $\mathcal{L}$–reductions *then K is $\mathbf{D}$–complete for $\mathbf{C}$–reductions.*

(ii) There is are K's which are $\mathbf{D}$–complete for $\mathbf{C}$–reductions, but which are not $\mathbf{D}$–complete for $\mathcal{L}$–reductions.

The following are some examples of such complete classes K, cf. [Imm87, Imm88, Ste91a, Ste91b].

7.21: Proposition

(i) **(Immerman)** *DTC is $\mathbf{L}$–complete, TC is $\mathbf{NL}$–complete and ATC is $\mathbf{P}$–complete for first order reductions.*

(ii) **Stewart** *The class $3COL$ of three–colourable graphs is $\mathbf{NP}$–complete for first order reductions.*

Note that DTC, TC and ATC give rise to computable quantifiers with constants or, better to predicate transformers, whereas $3COL$ can best be used as a computable quantifier.

extended to a unique σ'–structure $\mathcal{B} \in J$ for which $R(z(y_1), z(y_2), ..., z(y_r))$ is true.

7.16: Example

(i) Let σ consist of one binary relation symbol R and $\sigma' = \sigma \cup \{S\}$ with S another binary relation symbol. TC (DTC) consists of the class of σ'–structures such that the interpretation of S is the (deterministic) transitive closure of the interpretation of R. Both can be used to define predicate transformers of arbitrary size.

(ii) Similarly one can define ATC for the alternate transitive closure of a relation over a unary predicate, cf. [Imm87].

7.17: Theorem
Let $\mathcal{L}$ be a **C***–computable logic. If J is in* **C** *and in co-***C** *then $\mathcal{L}_{pJ}$ is a* **C***–computable logic.*

A similar theorem can be formulated for semi–regular logics by only requiring that K be in **C** and restricting the predicate transformer to positive occurrences.

7.4 $\mathcal{L}$–Reducibility

Immerman and Dahlhaus [Dah82, Dah83, Imm87] independently introduced the notion of classes K of σ–structures complete for a complexity class **C** by first order reductions. We present here a slight generalization:

7.18: Definition
Let K_1 be a class of σ_1–structures and K_2 be a class of σ_2–structures.

*(i) K_2 is $k - \mathcal{L}$–*reducible *to K_1 for a natural number k if K_2 is definable by a formula of the form*

$$Q^k{}_{K_1}(\phi_1, \ldots, \phi_n)$$

where all the ϕ's are $\mathcal{L}(\sigma_2)$ formulas and $Q^k{}_{K_1}$ is the quantifier defined by K_1 of size k.

(ii) K_2 is $\mathcal{L}$–reducible to K_1 if it is $k - \mathcal{L}$–reducible for some natural number k.

(iii) If $\mathcal{L}$ is First Order Logic we speak of k–first order reducible. (Immerman in [Imm87] considers the case where the ϕ's are even quantifier free).

contains at least one relation symbol. Let $\sigma' = \sigma \cup \{R\}$ where R is a relation symbol of arity r not in σ.

(i) We say that a σ'–structure $\mathcal{A}'$ is an expansion *of a σ–structure $\mathcal{A}$, if they have the same universe and all the relations of σ are identical.*

(ii) Let J be a set of natural structures over σ'. J is a predicate transformer *if for every σ–structure $\mathcal{A}$ there is at most one expansion $\mathcal{A}' \in J$.*

(iii) Let $\mathbf{C}$ be a complexity class and J be a predicate transformer. We now say that J is $\mathbf{C}$–computable *if, given a σ–structure $\mathcal{A}$ and an r-tuple of elements from the universe, $(b_1, b_2, ..., b_r)$, the problem whether $R(b_1, b_2, ..., b_r)$ is true in $\mathcal{A}'$ can be decided in $\mathbf{C}$. (Similarly we define when J is in co-$\mathbf{C}$).*

We define $\mathcal{L}_{pJ}$, the extension of $\mathcal{L}$ with a $\mathbf{C}$–computable predicate transformer by the addition of one more formation rule as follows:

7.14: Definition (Syntax)

(i) Let $\phi_1, \phi_2, ..., \phi_n, \psi_1$ be formulas in $\mathcal{L}_{pJ}$ such that for each relation symbol R_i of arity a_i in σ, there is a formula ϕ_i with a_i free variables which do not appear in any other of the ϕ's.
Then the following is a formula in $\mathcal{L}_{pJ}$:

$$Px_1, x_2, ..., x_m y_1, y_2, ..., y_r(\phi_1, \phi_2, ..., \phi_n)$$

where the variables $x_1, x_2, ..., x_m$ are bound, m is the sum of the a_i's and the variables $y_1, y_2, ..., y_r$ are free. We refer to such a formula as a formula of form (**).

(ii) As in the case of quantifiers we can consider the formula above as a predicate of size *1 and in a similar way define predicates of any size.*

Also the meaning function for $\mathcal{L}_{pJ}$ is very similar to the case of quantifiers. We only give the modifications needed.

7.15: Definition (Semantics)
The meaning function $M_{\mathcal{L}_{pJ}}(\theta, \mathcal{A}, z)$ is defined as the extension of the meaning function of $\mathcal{L}$ in the following way. Let θ be a τ–formula of the form (**), let $\mathcal{A}$ be a τ–structure. Let z be an assignment of the free variables in θ to elements in A the universe of $\mathcal{A}$ and z' be the restriction of z to the variables different from the y's.
Then $M_{\mathcal{L}_{pJ}}(\theta, \mathcal{A}, z) = 1$ iff the σ–structure defined by $\theta, \mathcal{A}$ and z' can be

(i) *Its universe is the universe A of $\mathcal{A}$ (in the case were the quantifier is of some size $k > 1$, each element in the defined structure is a vector of size k of elements from A).*

(ii) *Let $R_i^{a_i}$ be the i'th relation in σ (its arity is a_i). We associate with it the formula ϕ_i ,which has a_i free variables not appearing in any other of the formulas of (*) : $x_{j+1}, ..., x_{j+a_i}$, and define $R(b_1, b_2, ..., b_{a_i}) = M_{\mathcal{L}_{qK}}(\phi_i, \mathcal{A}, z')$. Where z' is a substitution as follows : for each free variable x_{j+p} from $x_{j+1}, ..., x_{j+a_i}$ we substitute the element b_{j+p}. For each other free variable in ϕ_i (which is also free in θ) we substitute the corresponding element from z.*

Now we put $M_{\mathcal{L}_{qK}}(\theta, \mathcal{A}, z) = 1$ iff $\mathcal{A}_\theta \in K$.

7.11: Example
Let σ consist of one binary predicate and let $3COL$ be the class σ–structures, which are 3–colourable graphs. A formula of the form (*) is true in a τ–structure $\mathcal{A}$ if ϕ_1 defines a 3–colourable graph on the universe of $\mathcal{A}$. The case of the same quantifier but choosing size 2 binds four variables. A formula of the form (*) is true in a structure $\mathcal{B}$ if the 4–ary relation defined by ϕ_1 defines a 3-colourable graph on the pairs of elements of the universe of $\mathcal{B}$.

We have the following general theorem:

7.12: Theorem
Let $\mathbf{C}$ be a complexity class containing at least $\mathbf{L}$. If K is in $\mathbf{C}$ and in co-$\mathbf{C}$ then $\mathcal{L}_{qK}$ is a $\mathbf{C}$–computable logic, i.e. every formula in the extended logic $\mathcal{L}_{qK}$ has a model checker of complexity $\mathbf{C}$.

This theorem requires some tedious checking, but is not surprising. However, if $\mathbf{C}$ does not contain $\mathbf{L}$, serious problems arise. A similar theorem can be formulated for semi–regular logics by only requiring that K be in $\mathbf{C}$ and restricting the quantifier to positive occurrences.

7.3 Computable Predicate Transformers

In some cases it is convenient to work with predicate transformers instead of generalized quantifiers. A predicate transformer is a mapping from relations to relations. A guiding example is the transitive closure of a binary relation. Other examples are computable queries an Database theory.

We are interested in extensions of computable logics $\mathcal{L}$ by predicate transformers. This generalizes the construction introduced in [Imm87].

To set up our framework we need some machinery.

7.13: Definition
Let σ be a vocabulary without function symbols or constant symbols that

(ii) K is in **PH** *iff K is definable by a formula of Second Order Logic.*

In our terminology, Second Order Logic captures **PH**.

7.2 Computable Quantifiers

We now define computable quantifiers. Our presentation follows the definition of the Lindström Quantifiers in [Ebb85] and combines the theory of generalized quantifiers with the idea of computable queries of [CH80].

Let **C** be a complexity class. Let σ be a vocabulary without function or constant symbols, K be a set of natural σ–structures in **C**. Further, let $\mathcal{L}$ be some regular **C**–computable logic. We define $\mathcal{L}_{qK}$, the extension of $\mathcal{L}$ with a **C**–computable quantifier, as follows :

7.9: Definition (Syntax)

(i) All the rules of forming formulas in $\mathcal{L}$ are rules of forming formulas in $\mathcal{L}_{qK}$.

(ii) Let $\phi_1, \phi_2, ..., \phi_n$ be formulas in $\mathcal{L}_{qK}$ satisfying the following condition:
For each relation symbol R_i of arity a_i in σ, there is a formula ϕ_i with a_i free variables which do not appear in any other of the ϕ and ψ formulas.
Then the following is a formula in $\mathcal{L}_{qK}$:

$$Qx_1x_2...x_m(\phi_1, \phi_2, ..., \phi_n)$$

where m is the sum of the arities a_i. We refer to such a formula as a formula of form (*).

(iii) The formula (*) above is considered as the syntax of a quantifier of size *1. A quantifier of* size *k is then defined as the formula we get by replacing each of the variables x_i with a vector of variables of size k (each element of each such vector can appear only in a single formula among the ϕ's and ψ's). The syntax of $\mathcal{L}_{qK}$ is then extended to include quantifiers of any size.*

7.10: Definition (Semantics)
The meaning function $M_{\mathcal{L}_{qK}}(\theta, \mathcal{A}, z)$ is defined by extending the meaning function $M_{\mathcal{L}}$ of $\mathcal{L}$ as follows:
Let θ be a τ–formula of the form (*), let $\mathcal{A}$ be a τ–structure and let z be a assignment of the free variables in θ to elements in A, the universe of $\mathcal{A}$. Now $\theta, \mathcal{A}$ and z define a σ–structure $\mathcal{A}_\theta$ as follows:

Now the analogue of Lindström's theorem over finite structures would be the identification of semi–regular logics $\mathcal{L}$ which capture complexity classes **C** for suitable **C**.

For the case where **C** equals the recursively enumerable (recursive) sets of natural structures such a characterization is easy. Let $\mathcal{L}_{r.e.}$ be like first order logic but with infinite disjunctions over r.e. sets of formulas and such that infinite disjunctions appear only under an even number of negations. Similarly, Let $\mathcal{L}_{rec}$ be like first order logic but with infinite disjunctions over recursive sets of formulas, iteratively applied a finite number of times without restrictions. Clearly $\mathcal{L}_{r.e.}$ and $\mathcal{L}_{rec}$ are semi–regular logics satisfying the above requirements. On the other hand any finite structure can be described up to isomorphisms by a first order sentence, therefore any r.e. set of finite structures can be described by a simple r.e. disjunction of such first order sentences. In other words:

7.6: Theorem

(i) $\mathcal{L}_{r.e.}$ *is a r.e.–computable semi–regular logic which captures the r.e. sets of structures.*

(ii) $\mathcal{L}_{rec}$ *is a recursive semi–regular logic which captures the recursive sets of structures.*

A different characterization of r.e.–computable logics was given in the fundamental paper of Chandra and Harel [CH80], in terms of computable queries and Pascal–like programs.

As already mentioned in section 3.7, Fagin noted that

7.7: Theorem (Fagin)
Let K be a class of natural structures. Then K is in **NP** *iff K is definable by an existential formula of Second Order Logic.*

In our terminology the set of existential formulas of Second Order Logic form a semi–regular logic which captures **NP**. This is a special case of a more general theorem due to J. Lynch, generalizing a result of Stockmeyer [Lyn82], noting that **NP** is just one level in the polynomial hierarchy **PH**. Recall, cf. [GJ79], that **PH** is the union of the complexity classes ${\Sigma_n}^P$ and ${\Pi_n}^P$, where ${\Sigma_1}^P$ is **NP** and ${\Pi_1}^P$ is **Co–NP**. Furthermore, a Σ_n–formula (Π_n–formula) of Second Order Logic is a formula in prenex normal form starting with existential (universal) second order quantifiers and having $n-1$ alternations of second order quantifiers.

7.8: Theorem (Lynch–Stockmeyer)
Let K be a class of natural structures.

(i) K is in ${\Sigma_n}^P$ (${\Pi_n}^P$) iff K is definable by an Σ_n–formula (Π_n–formula) of Second Order Logic.

7.1 Computable Logics

When we restrict ourselves to finite τ–structures first order logic is not anymore characterizable in a way similar to Lindström's theorem. We have to allow for extensions of first order logic. We still require that logics should be regular. We shall require that a formula is satisfiable iff it has a finite model. However, we cannot require that the set of valid sentences is recursively enumerable because of the following classical result [Tra50]:

7.1: Theorem (Trakhtenbrot)
There is a finite vocabulary τ such that the set of first order τ–sentences which is true in all finite τ–structures is not recursively enumerable.

On the other hand we have the following observation.

7.2: Definition

(i) *We say that a finite structure of cardinality n is* natural, *if its universe consists of the set $\{0, 1, \ldots, n-1\}$.*

(ii) *We say that a finite structure of cardinality n is* naturally ordered, *if it is natural and its vocabulary contains a binary relation symbol whose interpretation is the customary linear order on $\{0, 1, \ldots, n-1\}$.*

7.3: Theorem
Let τ be a finite vocabulary and ϕ be a first order τ–sentence. Then the set of finite natural τ–structures $\mathcal{A}$ such that $\mathcal{A} \models \phi$ is recursively enumerable. In fact, it is even in $\mathbf{L}$.

These considerations lead us to the following development.

7.4: Definition
Let $\mathbf{C}$ *be a complexity class. A semi–regular logic L is* $\mathbf{C}$–computable *if all of the following hold:*

(i) *For every finite vocabulary τ the set of τ–formulas is computable in* $\mathbf{C}$.

(ii) *The meaning function is invariant under isomorphisms. In other words, for every τ, every two τ–isomorphic τ–structures $\mathcal{A}, \mathcal{B}$ and every τ–formula ϕ we have $\mathcal{A} \models \phi$ iff $\mathcal{B} \models \phi$.*

(iii) *Let τ be a finite vocabulary and ϕ be a τ–sentence. Then ϕ has a model–checker of complexity less than* $\mathbf{C}$*, i.e. the set of finite natural τ–structures $\mathcal{A}$ such that $\mathcal{A} \models \phi$ is recognizable in* $\mathbf{C}$.

7.5: Definition
Let $\mathcal{L}$ be a semi–regular computable logic and $\mathbf{C}$ *be a complexity class. We say that $\mathcal{L}$* captures $\mathbf{C}$ *if for every class of naturally ordered τ–strucures K we have that K is in* $\mathbf{C}$ *iff K is definable in by a τ–formula of $\mathcal{L}$.*

6.7: Corollary
Let T be a finite set formulas over τ_{field} consistent with either ACF_0 or RCF. Then T does not admit elimination of quantifiers.

In other words, elimination of quantifiers is a rather rare phenomenon.

6.3 Elementary Geometry

The most important application of the method of elimination of quantifiers is in Elementary Geometry. Elementary Geometry is, by Descartes' reduction, identified with the normed vector spaces $\mathbf{R}^n$. Statements about configurations of points in $\mathbf{R}^n$ can be expressed in first order logic over the vocabulary τ_{field} using the axioms of RCF. Tarski's theorems, therefore, show that any such statement is decidable, using the method of elimination of quantifiers. Although the general method is doubly exponential in time and simply exponential space, special cases have been singled out with lower complexity. Applications of this method to Robotics are surveyed and discussed in [SSH87], applications to automated theorem proving in [Cho88].

6.4 Other Theories with Elimination of Quantifiers

We briefly list in this section other cases where the method of elimination of quantifiers was applied successfully.

- Dense orders, linear orders, well-orderings, monadic theory of linear order, [Ros82]
- Applications to temporal logic, such as Kamp's theorem, [GPSS80, Flu91].
- Boolean algebras, Abelian groups and modules and similar theories also with generalized quantifiers, [BSTW85].

7 Computable Logics over Finite Structures

The presentation of the material in this section has been developed by the author. It was never published but used in lectures since 1984. I am indebted to Y. Pnueli who once worked out the notes of my lectures on the subject.

In this section we shall frequently speak of complexity classes $\mathbf{C}$ such as $\mathbf{L}$ (LogSpace), $\mathbf{NL}$ (Non–deterministic LogSpace), $\mathbf{P}$ (Polynomial Time), $\mathbf{NP}$ (Non–deterministic Polynomial Time), ..., recursive, recursively enumerable. We do not give an abstract definition, and its enough to have these and similar examples in mind.

(iii) Σ *satisfies the submodel condition, if for every model* $\mathcal{B}$ *of* Σ *and every submodel* $\mathcal{A}$ *of* $\mathcal{B}$*, and every formula*

$$\phi = \exists x_1, \ldots, x_n B(x_1, \ldots, x_n B$$

with B *quantifierfree, we have that* $\mathcal{B} \models \phi$ *iff* $\mathcal{A} \models \phi$*.*

6.3: Theorem (Shoenfield)
If Σ *satisfies both the isomorphism condition and the submodel condition, then* Σ *admits elimination of quantifiers.*

To prove Tarski's theorem one has to verify the isomorphism and the submodel condition, which again uses particulars of the theory of algebraically closed fields.

Tarski proved the same theorem also for real closed fields. A τ_{field}-structure is a real closed field if it satisfies the field axioms and the following two axioms:

Square roots exist: $\exists y(y * y = x) \vee \exists y(y * y = -1 * x)$

Polynomials of odd degree have roots: For every term $t(x)$ of the form $y_{2n+1} * x^{2n+1} + y_{2n} * x^{2n} + \ldots + y_1 * x + y_0$ there is an x such that $t(x) = 0$.

We denote the axioms of real closed fields by RCF. Real closed fields are always of characteristic 0. Moreover, one can define an order relation $x < y$ by the formula $\exists z(y + (-1 * x) = z * z)$, which makes a real closed field into an ordered field.

6.4: Theorem (Quantifier elimination for real closed fields)
RCF *admits quantifier elimination. Furthermore, given* ϕ *the equivalent quantifierfree formula* ψ *can be computed in double exponential time and simple exponential space.*

6.5: Corollary
The first order theories ACF_0 *and* RCF *are decidable in double exponential time and simple exponential space.*

The complexity result are due to Collins and Ben–Or, Kozen and Reif, cf. [SSH87].

The question arises whether other first order theories of fields are decidable as well. Tarski conjectured and Ziegler [Zie82] proved the following

6.6: Theorem (Ziegler)
Let T *be a finite set formulas over* τ_{field} *consistent with either* ACF_0 *or* RCF*. Then every subset* T' *of* T *is undecidable. In particular, the first order theory of fields is undecidable.*

A field is of characteristic n if additionally

$$\underbrace{1 + \ldots + 1}_{n} = 0.$$

A field is algebraically closed if for every polynomial (τ_{field}-term) $t(x) = y_n * x^n + \ldots + y_1 * x + y_0$ with one free variable x and coefficients y_i we have

$$\exists x\ t(x) = 0 \vee \forall y, z\ t(y) = t(z).$$

Note that n is the degree of the polynomial provided that $y_n \neq 0$. We denote the axioms of algebraically closed fields of characteristic 0 by ACF_0. The field of complex numbers is an algebraically closed field of characteristic 0. The class of algebraically closed fields was structurally analyzed by Steinitz ([Ste10]), and his analysis served as a paradigm for algebra and model theory. Classical algebraic geometry is the study of algebraic varieties over the complex numbers. An algebraic variety is a set of n-tuples of complex numbers all of which satisfying some set of polynomial equations. In logic we replace equalities by arbitrary first order formulas, including quantifiers and inequalities. Tarski now proved the following:

6.1: Theorem (Tarski, Elimination of quantifiers for algebraically closed fields)
Let $\phi(x_1, \ldots, x_n)$ be a τ_{field}-formula. Then there is a quantifierfree τ_{field}-formula $\psi(x_1, \ldots, x_n)$ with the same free variables such that $ACF_0 \models \phi \leftrightarrow \psi$. Furthermore, ψ is computable from ϕ in double exponential time and simple exponential space.

The complexity result is due to J. Heintz [Hei83] and there several refinements. For an up to date discussion, cf. [Ier89]. The proof uses the field axioms to bring terms into polynomial normal form and then reduces the problem to the case where ϕ is an existential formula with no disjunctions. The last step exploits that every polynomial has a zero. This proof does not invite for generalizations. However, Shoenfield [Sho67] found the model theoretic contents of Tarski's theorem.

6.2: Definition
Let Σ be a set of first order formulas over some vocabulary τ.

(i) Σ admits elimination of quantifiers *if for every τ-formula. $\phi(x_1, \ldots, x_n)$ there is a quantifierfree τ-formula $\psi(x_1, \ldots, x_n)$ with the same free variables such that $\Sigma \models \phi \leftrightarrow \psi$.*

(ii) Σ satisfies the isomorphism condition, if for every two models $\mathcal{A}$ and $\mathcal{B}$ of Σ and every isomorphism g of a substructure of $\mathcal{A}$ and a substructure of $\mathcal{B}$ there is an extension of g which is an isomorphism of a submodel of $\mathcal{A}$ and a submodel of $\mathcal{B}$.

for a given function term $g(x)$. If there is such a function term the problem also requires to find it. Here the function terms are built from one variable x and the natural numbers by addition, subtraction, multiplication, division, the formation of exponentials and logarithms to basis e and the taking of n-th roots. The cumulative knowledge and expertise from Leibniz and Newton till 1969 have led to Risch's theorem [Ris69], which completely solves the problem. [Ros72] gives an elementary exposition. The solution is much more complicated than the solution of the corresponding problem for Algebraic Equations, i.e. whether zeros of polynomials with integer coefficients can represented by terms in the coefficients built from the natural numbers by addition, subtraction, multiplication, division and the formation of n-th roots. The latter problem was solved completely by Galois and the solution of the problem of Integration in Finite Terms is a grandiose completion of Galois' methods. Attempts of generalizing Galois theory to arbitrary structures were initiated by E.Engeler [Eng81]. This is a very promising and challenging program, which combines model theory and algorithmics, but which remains beyond the scope of this chapter.

Elementary Geometry is the geometry of *Ruler and Compass* in the Euclidean Plane and its generalization to the n-dimensional Euclidean Space. In contrast to the above problems, whose solutions are all based on the cumulative effort of generations of mathematicians, Elementary Geometry can be mechanized (and even implemented) based on rather little mathematical knowledge. The method which works here, however, is not based on a generally applicable method, like logical deduction, but on a method whose scope is limited and which has been completely characterized: The method of *Elimination of Quantifiers.* This section is exclusively dedicated to this method. A good text book introduction may be found in [KK67].

6.2 Tarski's Theorem

Tarski studied the first order theories of fields, in particular algebraically closed fields and real closed fields. Let $\tau^{field} = \{0, 1, -1, +, *\}$ where $0, 1, -1$ are constant symbols and $+, *$ are binary function symbols. A field is a structure over the vocabulary τ^{field} which satisfies the field axioms, where free variables are quantified universally:

Commutativity: $x + y = y + x$, $x * y = y * x$;

Associativity: $(x + y) + z = x + (y + z)$, $(x * y) * z = x * (y * z)$;

Neutral elements: $x + 0 = x$, $x * 1 = x$, $1 \neq 0$;

Distributivity: $x * (y + z) = (x * y) + (x * z)$;

Inverse elements: $x + (-1 * x) = 0$, $x \neq 0 \rightarrow \exists y(x * y = 1)$.

or the Fermat equation

$$x^n + y^n = z^n$$

where the variables range over the integers and the problem is to characterize the set of its solution quantitatively (empty, finite, infinite) or qualitatively (by another such equation holding between the values of the variables which form a solution). More precisely, a Diophantine Equation is of the form

$$Term = 0$$

where term are built from variables and natural numbers by addition, subtraction, multiplication, division and exponentiation. Concerning the above two equations it is conjectured that Catalan's equation has only one solution ($x = u = 3, y = z = 2$) and that Fermat's equation has none for $3 \leq n$. The solutions for $n \leq 2$ have been completely characterized. Both conjectures are still open, but in the last fifteen years Tijdeman, based on results of A. Baker [Bak75], has proved that Catalan's equation has only finitely many solutions and Faltings proved that for every $3 \leq n$ Fermat's equation has only finitely many solutions (cf. also [Bak84]). In 1900 D.Hilbert asked whether there is one general algorithm which decides whether a given Diophantine equation has a solution (Hilbert's Tenth Problem). The cumulative efforts of Davis, Putnam, Julia Robinson and Matijasevĉ finally showed in 1970 that *no such algorithm exists* [Mat70]. On the other hand the state of the art seems to suggest that if we restrict the problem to Diophantine equations in two variables, then all the successful methods known so far exhaust all the cases, which have been distinguished and, therefore such an algorithm might exists [Bak84].

A much simpler class of problems is the class of Arithmetical Identities. An Arithmetical Identity is an equation of the form

$$Term_1 = Term_2$$

such as

$$(a + b)^2 = a^2 + 2ab + b^2.$$

Here the terms are formed from variables and natural numbers by addition and multiplication only and the identity holds if it holds for all values the variables can assume. This problem has been mechanized completely. Its solution basically states that the methods you have (supposedly) learned in High School are complete. An excellent discussion of the arithmetical identities may be found in [Hen77].

Integration in Finite Terms is the problem whether there is a function term $f(x)$ such that

$$f(x) = \int g(x)$$

(i) $\mathcal{B}_0$ *is an initial segment of* $\mathcal{B}$ *and*

(ii) $\mathcal{B}_0 \models PA_1+$ *"f is not total".*

This theorem is a substantial improvement on [KM87] where a similar result is proven for some special combinatorial functions.

6 Elimination of Quantifiers

In this section we discuss a model theoretic method useful in Computer Aided Mathematics. We first discuss quite in detail the knowledge needed for turning mathematical practice into recordable and automatically reproducible experience. After all, the computer is a kind of a phonograph or rather epistograph, who will replay recorded procedures. If there is nothing to record, no programming technique can overcome the void.

6.1 Computer Aided Theorem Proving in Classical Mathematics

What we call here *classical mathematics* comprises Number Theory, Differential and Integral Calculus and Elementary Geometry. In each of these fields we state precisely a class of problems and discuss the prospects of mechanization and implementation of theorem proving. The point we wish to make here is that success or failure very much depend on a thorough understanding of particular features of the class of problems, and that, superficially speaking, *minor variations* may change the situation radically. Since very often advocates of the imminent breakthrough in Artificial Intelligence argue that mathematical Expert Systems like MACSYMA, REDUCE, CALEY, MAPLE, MATHEMATICA, etc. prove their point, we would, on the contrary, argue, that, upon closer analysis, these systems rather prove the point of Natural Ingenuity. More precisely, they show how the collective experience of generations of mathematicians, having reached a deep understanding of the subject, allows the mechanization of their knowledge, and that the invocation of Artificial Intelligence, at least in these cases, only adds confusion to an otherwise clearly understandable situation.

Number Theory was called by Gauss the Queen of Mathematics. It has many facets and recently number theoretic results have increasingly found applications in Computer Science. But the true purpose of Number Theory was always to test the depth of our mathematical understanding. The oldest branch of Number Theory is the theory of Diophantine Equations. Challenging examples are the Catalan equation

$$x^y - z^u = 1$$

Wainer hierarchy. The Wainer hierarchy is an extension to infinite countable ordinal indices of the more familiar Ackermann hierarchy of functions:

$$F_1(n) = 2n$$
$$F_{k+1}(n) = F_k^{(n)}(n)$$

where $F^{(n)}$ denotes the n'th iterate of F. The famous Ackermann Function is F_ω - quite low in this hierarchy.

5.4: Definition

(i) *We say that a function f* dominates *a function g if for all large enough n's, $g(n) < f(n)$.*
Note that for every $\alpha < \beta$, F_β dominates F_α. It is also worthwhile to recall that Ackermann Function, F_ω, dominates all primitive recursive functions.

(ii) *ϵ_0 is the first ordinal α satisfying $\omega^\alpha = \alpha$. (The exponentiation here is ordinal exponentiation and ϵ_0 turns out to be a countable ordinal - the limit of the sequence $\omega, \omega^\omega, \omega^{\omega^\omega}, \ldots$).*

(iii) *A function f is* provably recursive *in a formal theory $\mathcal{T}$ if there is an algorithm A that computes f for which $\mathcal{T}$ proves (using some fixed recursive coding of algorithms) that A halts on every input.*

5.5: Theorem (Wainer)

(i) *For every ordinal $\alpha < \epsilon_0$, the α'th function in the Wainer hierarchy, F_α, is provably recursive in PA.*

(ii) *If a total recursive function $f : N \to N$ is provably recursive in PA then it is dominated by some F_α in $\{F_\alpha : \alpha < \epsilon_0\}$.*

Furthermore, Fortune, Leivant and O'Donnel [FLO83] prove that the set of functions that are provably recursive in PA equals that set of functions that can be computed in (deterministic) time dominated by some F_α in $\{F_\alpha : \alpha < \epsilon_0\}$. Let $\mathcal{N}$ be the standard model of Peano Arithmetic. Let PA_1 be the first order theory consisting of the first order Peano Axioms together with all the Π_1 formulas true in $\mathcal{N}$. A formula is Π_1 is if it is of the form $\forall x \phi$ where all the quantifiers in ϕ are bounded. The model theoretic statement which shows the existence of functions not provably recursive in PA_1 is now the following theorem [BDD91].

5.6: Theorem (S. Ben–David and M.Dvir)
Let $f : \mathbf{N} \to \mathbf{N}$ be a recursive function which is not dominated by any F_α for $\alpha < \epsilon_0$. Then every non–standard model $\mathcal{B}$ elementary equivalent to $\mathcal{N}$ has a submodel $\mathcal{B}_0$ such that

A technical and philosophical discussion of such theorems may be found in [HMvS85]. A presentation of this theorem and related results accessible to computer scientists may be found in [Gal91].

5.2 Non–Provability in Complexity Theory

The following variation of the above theorems is due to S. Ben–David [BDH91]:

5.2: Theorem (S. Ben–David)
There is a language (sets of words recognizable by Turing machines) L_{PH} such that

(i) L_{PH} *is in* **Co–NP***;*

(ii) L_{PH} *is not context free, but*

(iii) it is not provable in Peano Arithmetic that L_{PH} *is not regular.*

Recently, S. Ben–David has analyzed these results further and related them to discuss the prospect of $\mathbf{P} \neq \mathbf{NP}$ not being provable in some formalized system such as Peano Arithmetic or fragments of Second Order Arithmetic [BDH91]. The key notion here are functions *extremely close to polynomials* where extremely close depends on the growth rate of functions not provably total in the formal system in question. His theorem states the following:

5.3: Theorem (S. Ben–David)
If $\mathbf{P} \neq \mathbf{NP}$ *is not provable in some fragment of second order Arithmetic* **S** *then every problem* P *in* **NP** *can be solved by an algorithm with run time upper bound* **S***–extremely close to a polynomial.*

5.3 Model Theory of Fast Growing Functions

The underlying theme of these theorems are fast growing functions. We review now some basic facts from the proof theory of Arithmetic. Our exposition is taken from [BDH91]. We refer the reader to [Smo80, Smo83] for an elaborated and truly enjoyable discussion of this topic. The basic idea goes back to Kreisel [Kre52]. For every recursive formal theory which is sound for Arithmetic there exist total recursive functions such that the theory cannot prove their totality. Such functions can be characterized by their rate of growth.

Wainer [Wai70] supplies a useful measuring rod for the rate of growth of recursive functions (from natural numbers to natural numbers) - the of Abstract Datatypes [MM84] extracting the model theoretic content of [GTWW77]. The latter was based on ideas from Category Theory and Universal Algebra. A more general discussion may be found in [Mak84]. Clearly, neither the closure under substructures nor under products has any explanatory power per se in these contexts. It would be more satisfactory, if the formation of products and the closure under substructures could be replaced by some activity stemming from handling databases. This was achieved with moderately satisfactory results in [Mak81, MV86].

The predominant rôle Horn formulas play in Logic Programming can be explained syntactically by the similarity of Horn formulas to deterministic rules or instructions. Semantically, the situation is similar to Abstract Data Types in as much as one thinks of a unique minimal interpretation. An exact model theoretic analysis of Horn formulas in Logic Programming was proposed in [Mak87]. Its relevance for Negation by Failure was discussed in Shepherdson's [She84, She85, She88]. The exact formulation of this analysis is unfortunately not possible in this survey. An excellent exposition of special properties of Horn formulas is [Hod92]. Formulas preserved under relativization play a vital rôle in relational database theory, especially in connection with *safe* queries, cf. [Ull82, TS88, MV86]. Horn formulas preserved under intersections were analyzed in [Mak87]. Finally, formulas with monotone predicates can be characterized as formulas with positive occurrence of the predicate and play an important rôle in the theory of computable fixed points and related topics [Mos74]. The use of preservation theorems in Database Theory will be discussed in my chapter [Mak92a].

5 Fast Growing Functions

5.1 Non–Provability Results in Second Order Arithmetic

It is questionable whether the model theoretic proof of the Paris-Harrington 3.6 theorem really captures the essence of the matter completely. The original proof has a proof theoretic flavour and for various generalization of this theorem no purely model theoretic proof is known. A prominent example is Friedman's theorem:

5.1: Theorem (H. Friedman)
There are programs (number theoretic functions) which

(i) always terminate (are total) but

(ii) such a termination proof does not exist within the formalization of various fragments of second order Arithmetic.

- Relativization
- Intersection of models
- Monotone predicates

and many more. All these preservation theorems were inspired by analyzing the constructions used by the *working algebraist*. They always exploit the fact that infinite structures are part of our discourse. In fact, most preservation theorems fail, if restricted to finite models. One exception was found by Gurevich and Shelah. cf. [Gur90].

4.1 Horn Formulas

Both, in Relational Database Theory and Logic Programming, first order formulas form the syntactic background of the field. In both fields it was observed that certain syntactically defined classes formulas play a special rôle. For a detailed discussion of first order logic's rôle in database theory one may consult [Var88, Kan90] and the corresponding chapter in this handbook [Mak92a] The most prominent such class of formulas are called *Universal Horn formulas*. They also play a certain rôle in the Specification of Abstract Data Types.

4.1: Definition (Horn formulas)

(i) A quantifierfree Horn formula is a formula of the form

$$P_1 \cap \ldots \cap P_k \rightarrow P_0$$

where all the $P_i, i \leq k$ are atomic formulas.

(ii) A Universal Horn formula is a formula of the form $\forall x_1, \ldots, x_m \Phi$ which Φ a quantifier free Horn formula.

The classical theorem of model theory gives the following characterization of Universal Horn formulas.

4.2: Theorem (Mal'cev)
Let K be a class of τ–structures which are exactly the models of a set of first order τ–formulas Σ. Then K is closed under substructures and products iff Σ is equivalent to a set Universal Horn formulas.

It is now tempting to find to use this characterization of Universal Horn formulas in order to explain their special properties in terms of Databases and Logic Programming. Fagin has done this in [Fag82] for the case of databases. Mahr and Makowsky have done this for the case of Specification as having the universe $\{0, 1, \ldots, n-1\}$. Let $S_\tau(n)$ be the number of τ-structures of size n. Recall that $I(\phi, n)$ is the number of different structures of size n satisfying ϕ. Let $P(n, \phi)$ be the fraction of $I(\phi, n)$ and $S_\tau(n)$.

3.16: Theorem (0–1-Law of First Order Logic)
For every τ without function symbols and every first order τ-formula ϕ the limit

$$\lim_{n \to \infty} P(n, \phi)$$

is well defined and is either 0 or 1.

3.17: Definition (Almost true formulas)
A First Order Formula ϕ is almost true if $\lim_{n \to \infty} P(n, \phi) = 1$.

In contrast to First Order Validity over finite structures, which is undecidable (cf. Trakhtenbrot's theorem 7.1), the set of first order sentences true in almost all structures is decidable. In fact, Grandjean proved [Gra83]:

3.18: Theorem (Grandjean)
Assume that τ has no function symbols. The problem of deciding, whether a first order τ–formula ϕ almost true, is **P**–*Space complete.*

0–1 Laws were investigated also for extensions of First Order Logic. For a further discussion of similar theorems the reader should consult [Com87, Fag90] and the literature quoted therein. Striking applications of 0–1 Laws in Computer Science are still missing. They may emerge in the context of Average Case Complexity Theory [Gur91], Graph Algorithms [GS87] and the like.

4 Preservation Theorems

Preservation theorems of First Order Logic characterize syntactic classes of formulas in terms of their semantic properties. In section 4 we have given the simplest example, the substructure theorem. It's proof is a simple but ingenious use of the compactness theorem. This specific application of compactness was termed the *Method of Diagrams* and may be found in every introduction to model theory. Alternative proofs of preservation theorems were given using ultraproducts [CK90].

The classical preservation theorems characterize formulas which are preserved under

- Unions of chains
- Homomorphisms
- Products and reduced products

richer semantic structures, found rich applications in theoretical computer science. Engeler was the first to observe that infinitary logic can serve as a framework to formulate the input/output behaviour of programs. His approach was considered awkward. V. Pratt D.Kozen used a Kripke-like semantics for his approach to axiomatize the input/output behaviour of programs, which was finally called 'Dynamic Logic'. This was received enthusiastically. However, it was soon observed that the two approaches were equivalent. Burstall suggested modal and Pnueli temporal logic for the axiomatic description of program behaviour. Kripke-structures are also abundant in foundational research in AI, especially in the theory of knowledge.

3.9 The Hidden Method

One model theoretic tool of central importance does usually not appear in the statement of theorems, but mostly in their proofs. This is the 'back-and-forth' characterization of n-equivalent structures, i.e. structures satisfying the same sentences of quantifier rank n. This characterization originated in the early work of R. Fraïssé and was popularized in an influential paper by A. Ehrenfeucht. Ehrenfeucht also generalized the method to monadic second order logic, and further generalizations for infinitary logic and logics with generalized quantifiers and predicate transformers were developed subsequently, cf. [BF85]. We shall devote section 8 to an extensive discussion of this method, which call Ehrenfeucht-Fraïssé games. Here we only list some of its application.

Originally, Ehrenfeucht-Fraïssé games were used to prove that certain concepts are not definable by first order formulas even if restricted to finite structures. Among such concepts we find the connectivity and planarity of graphs. The deepest and most surprising application of Ehrenfeucht-Fraïssé games occurs in the proof of Lindström's theorem. A close analysis of this proof also shows that Beth's theorem can be proven using this method, as well as various preservation theorems. Ehrenfeucht's generalization of the method to monadic second order logic can be used to give a model theoretic proof of Büchi's theorem. It was used in [FR79] to establish lower and upper bounds for the complexity of decidable theories such as Presburger Arithmetic and the theory of two successors functions. And finally, it can be also used to prove the 0–1 law for first order logic over finite structures, due independently to R. Fagin and Glebskiĭ, Kogan, Ligon'kiĭ and Talanov.

3.10 0–1 Laws

To state the 0–1 Theorem, let τ be a vocabulary without function symbols and let ϕ be a first order τ-formula. We think of a structure of size n

This theorem was followed by intense investigations of model theories of particular logics and the evolution of a framework for 'abstract model theory'. The fruits of these investigations were collected in the monumental volume [BF85].

In 1965 S. Kripke initiated the model theoretic study of logics different from classical first order or propositional logic, such as intuitionistic logics, modal logics and temporal logics. His main idea was to look at, say propositional modal or temporal logic, as a special case of first order logic. A Kripke-structure is a first order structure with a binary relation for accessibility to possible states (worlds in the case of modal logic, points in time in the case of temporal logic). Propositions then are unary predicates in Kripke-structures. The modal and temporal operators (necessarily/possibly, always/sometimes) now become first order definable. The axioms of modal or temporal logic shape the accessibility relation. In this way Kripke was able to state precisely the semantics of modal logic and prove, for the first time, completeness theorems. To illustrate this let us state here case of the modal system T, which captures the unproblematic aspects of 'necessity'. The formula $\Box\phi$ is read as 'necessarily ϕ'. The system T contains all substitutions of propositional tautologies, the axioms $\Box(\phi \to \psi) \to (\Box\phi \to \Box\psi)$ and $\Box\phi \to \phi$, and the two deduction rules Modus Ponens and from ϕ infer $\Box\phi$.

3.14: Theorem (Kripke)
A modal formula ϕ is provable in T iff ϕ is true in all Kripke-structures with a reflexive accessibility relation.

We speak of temporal logic when the accessibility relation is a partial order, in the most natural case a discrete linear order. The formula $\Box\phi$ is now read as 'always ϕ'. It is natural to ask whether the introduction of one temporal operator (or for that matter, modal operator) suffices, or whether there are many hitherto undiscovered temporal operators. Obviously we have operators corresponding to 'next'. 'previously' 'always in the future', 'always in the past', 'ϕ until ψ' and 'ϕ since ψ'. We note that all these operators are first order definable over linearly ordered Kripke-structures. H. Kamp now proved the following remarkable

3.15: Theorem (Kamp)
Let $TO(p_1, \ldots, p_n)$ be an n-ary temporal operator which is first order definable over discrete, complete linear orderings. Then $TO(p_1, \ldots, p_n)$ is definable from the operators 'next', 'previously', 'until' and 'since'.

The theorems of Kripke and Kamp are two prime examples of model theoretic theorems in non–standard logic. The underlying techniques, however, are applicable in a much wider context and have not yet been systematically developed. Good surveys are [Bur84, BS84, RS23].

Both types of generalizations of first order logic, more formulas and

3.11: Theorem (Büchi 1960)
A set of words is regular iff it is definable by an existential formula of monadic second order logic.

Trakhtenbrot [Tra61] independently found a similar theorem.

Fagin studied the finite spectrum and was led to the following theorem:

3.12: Theorem (Fagin)
A set of finite structures is in **NP** *iff it is definable by an existential (full) second order sentence.*

Let ϕ be a first order formula over a vocabulary τ. We note that $Spec(\phi)$ can be viewed as the set of finite models of Φ over the empty vocabulary, where Φ is obtained from ϕ by existentially quantifying all the predicate symbols of τ. So Fagin's theorem generalizes the both the spectrum problem as well as Büchi's theorem.

Immerman characterized similarly sets of ordered finite structures in **L**, **NL**, **P**. We shall discuss the interplay between model theory and complexity theory in section 7.

3.8 Beyond First Order Logic

In this introduction we already have come across features which go beyond first order logic. We have tacitly introduced quantification over relations in Büchi's theorem, and we have mentioned the semantic restriction to finite structures. These mark the two independent directions generalizations might take: More sentences vs. more complex models.

The model theoretic study of richer logics over τ-structures in the usual sense was initiated in the late 50ies independently by A. Tarski and his students, and E.Engeler for infinite first order formulas, and by A. Mostowski for generalized quantifiers. The book [BF85] contains an excellent bibliography and historic account. From a naive model theoretic point of view it is natural to ask whether for those generalized logics the compactness theorem and the Löwenheim-Skolem theorem are still true. For infinite formulas compactness fails trivially. It was also observed that in all the examples of generalized quantifiers studied one of the two usually failed. In 1966 P. Lindström published a paper which was hardly noticed till 1970. In it the following fundamental result was stated and proved:

3.13: Theorem (Lindström)
Let $\mathcal{L}$ be a regular logic over τ-structures which both satisfies the compactness theorem and the Löwenheim-Skolem theorem. Then $\mathcal{L}$ is, up to semantic equivalence, first order logic.

A logic is *regular* if it is closed under boolean operations, quantification, relativization and does not distinguish between isomorphic τ-structures.

countable atomless boolean algebras. Steinitz showed that any two uncountable algebraic closed fields of characteristic zero of the same cardinality are isomorphic. So Vaught's theorem quickly establishes that these theories are complete and therefore decidable.

3.7 Spectrum Problems

The study of categoricity of first order theories was the driving force behind the deepest results of model theory. Ryll-Nardzewski, Svenonius and Engeler independently characterized ω-categorical theories, and Morley proved the following generalization of Steinitz' theorem:

3.9: Theorem (Morley)
If Σ is categorical for some uncountable κ then Σ is categorical for every uncountable κ.

If Σ is not categorical, then it is natural to look at the following: Let Σ be a set of formulas and denote by $I(\Sigma,\kappa)$ the number of non-isomorphic models of cardinality κ. $I(\Sigma,\kappa)$ is called the spectrum of Σ. The study of $I(\Sigma,\kappa)$ for infinite κ was initiated by Morley and Vaught (cf. [CK90]). A complete analysis of the infinite case dominated the research efforts in model theory and culminated in Shelah's theorem [She90]:

3.10: Theorem (Shelah's Spectrum Theorem)
For uncountable κ $I(\Sigma,\kappa)$ is non-decreasing in κ and, in fact either

(i) $I(\Sigma,\kappa) = 2^{\kappa}$ *or*

(ii) $I(\Sigma,\omega_\alpha) < BETH_{\omega_\alpha}(card(\alpha))$.

The infinite spectrum and its ramifications are the core of a highly sophisticated development in model theory called stability theory. Although it is of extreme mathematical depth and beauty I can so far see no fruitful interplay between stability theory and computer science.

Instead of $I(\Sigma,\kappa)$ for finite κ, we shall look at the finite cardinal spectrum $Spec(\Sigma)$ of finite sets of formulas Σ. $Spec(\Sigma)$ is the set of natural numbers n such that there is a finite model of Σ of cardinality n. The study of $Spec(\Sigma)$ was initiated by Scholz. For the historic remarks cf. [Fag90]). In contrast to stability theory, the study of the finite cardinal spectrum $Spec(\Sigma)$ led to very interesting interactions between model theory and complexity theory, through the pioneering work of Büchi, Fagin and Immerman (cf. [B̈60, Fag74, Imm87].

Büchi studied the interplay between Monadic Second Order Logic and automata theory. He looked at words over a finite alphabet as finite linearly ordered structures with unary predicates. Recall that a set of words is regular if it is recognizable by a finite automaton. His theorem states:

against the Hydra, where the Hydra grows n new heads after the nth blow it receives. The underlying theme of this theorem are *fast growing functions*. We return to this topic in section 5.

3.6 Complete Theories and Elimination of Quantifiers

Another line of early investigations was the study of complete theories. A set Σ of formulas (over a fixed vocabulary τ) is complete if for every formula ϕ either $\Sigma \models \phi$ or $\Sigma \models \neg\phi$. The original interest for complete theories stems from questions of decidability. A set of formulas Σ is decidable if its set of consequences is recursive.

3.7: Theorem
If Σ is recursive and complete then Σ is decidable.

Proofs of completeness were often obtained using the method of elimination of quantifiers. Tarski used these ideas to show that there is a decision procedure for Elementary Geometry, which he identifies with the first order theory of real closed fields. This theorem led recently to interesting applications in robotics. But the method of elimination of quantifiers has not yet received the attention it deserves among researchers in automated theorem proving. The state of art in automated theorem proving for elementary geometry is best discussed in [Cho88, SSH87].

Another way of proving completeness of first order theories is based on a simple but ingenious observation due to Vaught, which shows the power of model theoretic reasoning. Let Σ be a complete theory. If Σ has a model $\mathcal{A}$ which is finite, then it is unique up to isomorphism. If $\mathcal{A}$ is infinite, then by the Löwenheim-Skolem Theorem, Σ has models of arbitrary infinite cardinalities. Now, if all models of Σ of infinite cardinality κ are isomorphic, we say that Σ is κ-categorical. Note that if $\mathcal{A}$ and $\mathcal{B}$ are isomorphic then they satisfy the same first order sentences.

3.8: Theorem (Vaught)
If Σ is κ-categorical for some infinite κ and Σ has no finite models, then Σ is complete.

Proof. Assume, for contradiction, that there is ϕ such that neither $\Sigma \models \phi$ nor $\Sigma \models \neg\phi$. As Σ has no finite models, using the Löwenheim-Skolem Theorem we can find models $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{A} \models \Sigma \cup \{\phi\}$ and $\mathcal{B} \models \Sigma \cup \{\neg\phi\}$, both of cardinality κ. But then $\mathcal{A}$ is isomorphic to $\mathcal{B}$, which contradicts the fact that $\mathcal{A} \models \phi$ and $\mathcal{B} \models \neg\phi$. ∎

Classical mathematical results establish categoricity of a few natural first order theories. Hausdorff and Cantor showed that any two countable dense linear orderings are isomorphic, and a similar argument shows the same for

Σ. By an ingenious application of the compactness theorem he proved the converse of this observation:

3.5: Theorem (Substructure Theorem)
A set Σ of first order formulas is preserved under substructures iff Σ is equivalent to some set of universal formulas.

The Substructure Theorem set a pattern for further investigations whose results are called preservation theorems. It led to similar syntactic characterizations for formulas preserved under unions of chains, homomorphisms, products, intersections and other algebraic operations. There are also some surprising interrelationships between a generalization of Beth's theorem and preservation theorems for a wide class of operations between structures, cf. [Mak85]. Some of these preservation theorems have variations and interpretations which are of importance in database theory [Mak84] and the foundations of logic programming, [Mak87]. Questions related to such preservation theorems also occur naturally in the compositional approach to model checking for various temporal logics [Eme90]. The latter is a subdiscipline of program verification. It still remains an open avenue of research to find the preservation theorems which will be useful for model checking, in particular those preservation theorems which will reflect the compositionality of programs.

3.5 Disappointing Ultraproducts

With these early investigations centering around the compactness theorem and the preservation theorems an alternative proof of the compactness theorem was discovered using ultraproducts. The method of ultraproducts also lead to alternative proofs of preservation theorems and dominated research in model theory throughout the 60ies (cf. [CK90]), but it had almost no impact on theoretical computer science. Although Kripke and Kochen [KK82] used bounded ultraproducts to give a model theoretic proof of the Paris-Harrington Theorem, Kanamori and McAloon [KM87] gave a model theoretic proof of this theorem without bounded ultraproducts. In the language of theoretical computer science this theorem can be stated as follows:

3.6: Theorem (Paris, Harrington)
There are programs (number theoretic functions) which

(i) always terminate (are total) but

(ii) such a termination proof does not exist within the formalization of Peano arithmetic.

A very picturesque version of this theorem is due to Kirby and Paris [KP82]. The function described there is a winning strategy for the fight of Hercules of mathematics, as it led to 'non–standard' models of the Natural Numbers, the Real Numbers and of Set Theory. However, A. Robinson realized that those non-standard models had their own usefulness for developing genuine first order mathematics. For theoretical computer science, non-standard models of number theory and set theory only recently started to play a rôle. We shall not discuss their use in this paper, but refer the reader to [ANS82, MS89, Pas90].

On the positive side we have Beth's theorem on implicit definitions and its various generalizations. Those theorems were mostly proven first by syntactic methods, but the model theoretic proofs found later make those theorems independent of the particular formalism of first order logic. Let Σ be a set of first order formulas over some vocabulary τ, and let P be an n-ary relation symbol not in τ. We say that a formula $\phi(P)$ over $\tau \cup \{P\}$ defines P *implicitly* using Σ, if in each model $\mathcal{A}$ of Σ there is *at most one* interpretation of P. We say that the predicate implicitly defined by ϕ using Σ has an *explicit* definition of there is a formula $\theta(x_1, x_2, \ldots, x_n)$ over τ such that

$$\Sigma \cup \phi(P) \models \forall x_1, x_2, \ldots, x_n(\theta(x_1, x_2, \ldots, x_n) \leftrightarrow P(x_1, x_2, \ldots, x_n)).$$

Now Beth's theorem can be stated as follows:

3.4: Theorem (Beth)
Let Σ be a set of first order formulas and let $\phi(P)$ be an implicit definition of P using Σ. Then there is an explicit definition of P using Σ.

Beth's theorem is trivially true for second order logic, and false for first order logic when restricted to finite structures. In the latter case, implicit definitions allow us to define classes of structures recognizable in **NP** $\cap$ **co–NP**, whereas first order formulas define classes recognizable in **L**. We shall discuss the consequences of this observation in section 7. Beth's theorem is mainly appealing as a closure property of a logic. There are surprisingly few genuine applications of Beth's theorem and its relatives. One of them, in the axiomatic treatment of specification theory, is relevant to theoretical computer science (cf. [MS92]). More recently Kolaitis has studied implicit definability on finite structures and related it to issues in complexity theory, [Kol90].

3.4 Preservation Theorems

Another line of explorations of the compactness theorem was initiated by Tarski. He observed that universal first order formulas are preserved under substructures. In other words, if Σ is a set of first order formulas in prenex normal form with universal quantifiers only and $\mathcal{A} \models \Sigma$ and $\mathcal{B} \subseteq \mathcal{A}$ is a substructure of $\mathcal{A}$ then $\mathcal{B} \models \Sigma$. The same is true for any Σ_1 equivalent to given formula holds in the given structure. Any mathematically proven statement about the meaning function is a model theoretic theorem.

The first result of mathematical logic which could be called model theoretic was the famous Löwenheim-Skolem Theorem:

3.1: Theorem (Löwenheim-Skolem Theorem)
Let Σ be a set of formulas of first order logic such that there is an infinite $\mathcal{A}$ with $\mathcal{A} \models \Sigma$. Then there are models $\mathcal{B}$ of arbitrary infinite cardinalities $\kappa \geq card(\tau) + \omega$ such that $\mathcal{B} \models \Sigma$.

The most basic model theoretic theorem is the compactness theorem for first order logic. We say that a set Σ of formulas is satisfiable if there is a structure $\mathcal{A}$ such that $\mathcal{A} \models \Sigma$. The compactness theorem now states that:

3.2: Theorem (Compactness Theorem)
A set Σ of first order formulas is satisfiable iff every finite subset of Σ is satisfiable.

It follows from Gödel's completeness theorem for countable Σ and was proven for arbitrary Σ by Mal'cev. A model theoretic proof of the completeness theorem was given independently by Hasenjäger, Henkin and Hintikka in 1949. This proof, most widely known as Henkin's method, was instrumental in shaping the further developments of logic and model theory.

The completeness theorem usually refers to some specific *deduction method* and states that a τ–formula ϕ is derivable from a set of τ–formulas Σ iff ϕ is a semantical consequence of Σ. The notion of semantical consequence is model theoretic. It says that for every τ–structure $\mathcal{A}$ and every assignment z such that $M(\Sigma, \mathcal{A}, z) = 1$ we also have $M(\phi, \mathcal{A}, z) = 1$. A purely model theoretic statement which captures the essence of the completeness theorem without reference to the particular deduction is the following:

3.3: Theorem
For every recursive enumerable set Σ of τ–formulas the set τ–formulas ϕ which are semantical consequences of Σ is recursive enumerable.

3.3 Definability Questions

The next ten years of evolving model theory were marked by explorations of the compactness theorem and the Löwenheim-Skolem Theorem. The first of this explorations concerns definability questions, both negative and positive results.

On the negative side we have that many important mathematical concepts cannot be captured by first order formulas. Among them are the concept of well-orderings, connectivity of binary relations and Cauchy completeness of linear orders. This was first perceived as blow to the foundation

3 Model Theory and Computer Science

As we discuss here applications of model theory to computer science we have to clarify what we intend both by model theory and theoretical computer science.

3.1 Computer Science

Concerning Computer Science we take a pragmatic approach. Any mathematically modelled situation which captures any issue arising in the dealings with computers is a possible topic for computer science. This includes hardware, software, data modelling, interfaces and more. Some of the more classical fields of theoretical computer science have already matured into well established subdisciplines. Among them we find computability theory, algorithmics, complexity theory, database theory, data and program specification, program verification and testing etc. However, we feel that a certain confusion in the definitions of these fields is obfuscating the issues involved. It very much depends whether our point of view is *method–oriented* or *application–oriented.* Computability and complexity theory deal with the clarification of our notion of what is computable. This represents a clear case of a well defined method–oriented subdiscipline of computer science and the foundations of mathematics. Database theory on the other hand is a field which grew from an application–oriented approach. From a method–oriented point of view, database theory tends to fall apart into subfields, such as *finite model theory*, *operating systems*, *file systems*, *user interfaces* and *algorithmics*, where each of these transcend the boundaries of the database applications. Scientifically speaking, the ad hoc collection of methods bound together by a vaguely defined common application is unsatisfactory. It is justified only for didactic purposes such as training application–oriented engineers. But such training is detrimental to a deeper understanding of the craft and the science and leads to chaotic duplicity (and multiplicity) of research and research subcultures each disguised in its own terminology and provincialisms.

In this paper we try to exhibit a method and a scientific framework, *model theory*, and discuss typical problems whose discussion in this framework is beneficial to our understanding.

3.2 The Birth of Model Theory

Model theory deals with the mathematical study of the satisfaction relation or its characteristic function, the meaning function. For a specific syntactic system which we call logic, the meaning function singles out the pairs of first order structures and formulas which we interpret as asserting that the explicitness and technicalities of the field of application.

2.3 Logics

The most prominent logic is First Order Logic. Although we argued that τ–structures are sufficiently general to model all situations which can be treated mathematically, First Order Logic has a limited expressive power. This means that a description in First Order Logic of a situation will allow what are called *non–standard models*. In other words, it will have models of that description which do not capture all the intended features.

Other logics we shall consider are Second Order Logic (allowing quantification over subsets and relations without making them into objects of the model), Monadic Second Order Logic (allowing quantification only over subsets), infinitary logics (allowing infinite conjunctions and disjunctions) and logics with generalized quantifiers. The latter will be discussed in detail in section 7.

A logic itself again can be modelled within set theory. It consists of a family of τ–formulas $Fm(\tau)$ with associated meaning functions M_τ subject to several conditions. The most fundamental among them is the *Isomorphism Condition* which asserts that isomorphic τ–structures cannot be distinguished by τ–formulas. The other conditions assert that the most basic operations such as conjunction, disjunction, negation, relativization and quantification over elements are well defined. Such logics are called regular logics. If negation is omitted we call the logics semi–regular. The model theory of such logics has been extensively studied, cf. [BF85].

For applications in computer science the relevant logics have two additional features: The set of τ–formulas Fm_τ is recursive for finite τ and the meaning functions M_τ are *absolute* for set theory, i.e., they do not depend on the particular model of Zermelo-Fränkel set theory we are working in. If we additionally require that the tautologies of such a logic are recursively enumerable, we call such a logic a *Leibniz Logic*. It now follows from work of Lindström and Barwise that every Leibniz Logic is in some precise sense equivalent to First Order Logic, cf. [BF85]. In other words, a proper extension of First Order Logic is either not regular or not absolute or its tautologies are not recursively enumerable. If we restrict ourselves to finite structures the latter is unavoidable even for First Order Logic, but then the satisfiable formulas are recursively enumerable. Semi–regular logics on finite structures where the satisfiable formulas are recursively enumerable have many applications to computer science and are studied in 7.

the set of substructures.

In the case of graphs the modelling issue is more subtle. It is customary to describe a graph as a set with an incidence relation. Thus there is quantification over vertices but not over edges. If we choose to allow quantification over edges we change the notion of structure. To what extent this matters has been studied by Courcelle in a series of papers [Cou90a, Cou90b]. Finite graphs can also be described by their incidence matrix, which does not fit the notion of a τ–structure in a natural way. However, we can consider the incidence matrix itself as a τ–structure in many ways.

Logic and model theory take the notion of τ–structures for granted. How to choose the particular vocabulary depends on many extra–logical issues. Discussing some of these issues is a discipline in itself called Data Modelling. The issues discussed there come from data processing and data bases.

First order logic allows quantification only for elements of the underlying universe. This looks like a severe restriction, as in mathematics we quantify very often also over subsets and more complex objects. However, this restriction only affects the modelling issue. In set theory all objects are sets, and second order arithmetic can be formalized using first order τ–structures, where the universe consists of points and sets with a unary predicate distinguishing between them. It is in this sense that the notion of τ–structures is as universal as the set theoretic modelling of mathematical situations.

More surprisingly, τ–structures can also capture situations of modal and temporal propositional logic. A propositional variable may be true in some moments of time and false on others. So let the universe of our discourse be time and propositions be unary predicates [Bur84]. This is almost obvious. In the case of modal logic it needed Kripke's ingenuity to make use of this idea [BS84]. The universe now is a set, the set of all possible worlds or situations, propositions are again unary predicates, but the relationship between possible worlds is described by an accessibility relation. From here, it is natural to continue and consider several accessibility relations (to model for example the distinction between the legally and the morally possible). In the theory of program verification this was used to model the behaviour of abstract programs (Dynamic Logic [Har84]). In AI this approach was extended further to model reasoning about knowledge [Eme90]. The interested reader will find more also in section 3.8 and [Ga92]. For reasons of space we shall not treat these issues much further in this chapter.

The point we want to emphasize here is that the framework of τ–structures is flexible enough to model everything which can be modelled in mathematics, more precisely in set theory. The choice of vocabulary is sometimes difficult and guided by various issues, including user friendliness, *plication*, a unary function, called the *inverse*, and a constant called the *unit element*. An ordered group is additionally equipped with a binary relation, called the *order relation*. In similar ways we can define *fields*, *rings* or the structure of *arithmetic on the natural numbers*. In computer science other *data structures* are defined similarly, such as *words, stacks, lists, trees, graphs, Turing machines etc.*. A word of length n over the alphabet $\{0,1\}$ can be viewed as a set of n elements with a binary relation which linearly orders that set and a unary relation, which indicates which places in the word are occupied by the letter 1. A graph is just a set with a binary relation. In each case it is required that the functions, relations and constants satisfy some interrelating properties which make it into a group (field, word, graph, Turing machine etc.).

Sometimes, it is more practical to model structures with several underlying sets, as in the case of vector spaces. These sets form several universes and are called *sorts*. We then speak of *many–sorted structures*. A Turing machine consists of two sets: a set of states and a set of letters; a binary relation between states and letters; a unary relation, the set of final states; and a constant, the initial state. Many–sorted structures allow us to model also concepts which involve sets of sets, such as topologies, families of subgroups or whatever comes to ones mind. This last statement is not just a sloppy way of saying something vague. It really expresses a belief, or rather experience, that everything which can be modelled in set theoretic terms with finitely many basic concepts can be modelled by such structures.

In modern terms a structure is a tuple of sets of specified characteristics. The primitive concepts have *names* and these names form again a set, called the *vocabulary*. A structure then is an *interpretation* of a vocabulary. More precisely, a (first order) vocabulary τ is a set of sort symbols, function symbols, relation symbols and constant symbols. The function, relation and constant symbols have an *arity* which specifies the number and sorts of the arguments and values. The arity is mostly assumed to be *finite*. In this way we can naturally associate with a vocabulary τ the proper class of all τ–structures, which we denote by $STR(\tau)$.

2.2 The Choice of the Vocabulary

The notion of a τ–structure evolved naturally in mathematics, more precisely in algebra. Groups and fields are usually described as sets with operations, ordered fields are sets with operations and relations. The choice of the basic operations is in no way trivial. Should we add the inverse operation as basic or not ? In the case of arithmetic we have the successor relation, addition and multiplication. The first order theory of arithmetic is undecidable, but if we leave out multiplication, it becomes decidable. This is a dramatic change. Subtraction is definable by a first order formula, so leaving it out or adding it, does not affect decidability. But it does affect to number theory (Skolem, Gödel) and finally, of modelling logic within set theory (Tarski, Vaught). The first two reductions were motivated by the fundamental questions of the foundations of mathematics, whereas the latter accepts Bourbaki's view that set theory is the foundational framework of mathematics. It is this latter approach which forms the background of model theory proper. Let us elaborate this further: We take some Naive Set Theory for granted and attempt to model all objects of mathematical study within this Set Theory. Without having to bother too much about the choice of set theory we can model the natural numbers, finite strings, finite graphs within set theory. We accept the axiom of choice as a fact of life. With this we can model also most of the concepts of classical algebra (field theory, ring theory, group theory, but not necessarily cohomology theory) within set theory. The natural numbers, fields, graphs are mathematical structures which serve as the prime examples for models of logical theories. We usually think of models of a logical theory rather than of a single model, and the models form usually a proper class (the class of all groups, rings, etc.). If we restrict ourselves to finite mathematical structures we can additionally consider recursive sets of models or sets models of lower complexity classes (Logarithmic Space or Polynomial Time recognizable classes of models). Next we observe that logical theories are just sets of formulas and that formulas can be viewed again as either strings over some alphabet or as some kind of labeled trees. Most people think of formulas as inherently finite objects, but infinite formulas (then better viewed as trees) are easily conceivable. So formulas and sets of formulas can also be modelled in our set theory. If we think of finite formulas as strings it makes sense to bring in also concepts of recursion theory and complexity theory.

The basic relationship between sets of formulas and models is the satisfaction relation. We view it here as a ternary relation $M(\Sigma, \mathcal{A}, z)$, where Σ is a set of formulas, $\mathcal{A}$ is a structure, i.e. a generalized algebra over some vocabulary (similarity type) and z is an assignment function mapping free variables of the formulas Σ into elements of the universe of the structure $\mathcal{A}$. If $M(\Sigma, \mathcal{A}, z)$ holds for every z we simply write $\mathcal{A} \models \Sigma$ and say that $\mathcal{A}$ is a model of Σ. The characteristic function of the satisfaction relation is often called meaning function. The meaning function can also be modelled in set theory.

2.1 First Order Structures

It is customary to model algebraic structures as sets equipped with functions and relations. This view has its origins in algebra as understood in the 19th century. A structure consists of a set, the *universe*, equipped with some relations, functions and constant, which model the *primitives*.

A *group* then is a set equipped with a binary function, called *multi-*

competing) cultural systems which are themselves embedded in host cultural systems; that nevertheless the fame and prestige of the protagonists of science and scientific progress do play an important, possibly also counterproductive rôle; that cultural stress and cultural lag play a crucial rôle in the evolution of concepts; that periods of turmoil are followed by periods of consolidation after which concepts stabilize; that diffusion between different fields usually will lead to new concepts and accelerated growth of science; that environmental stresses created by the host culture and its subcultures will elicit observable response from the scientific culture in question; and, finally, revolutions may occur in the metaphysics, symbolism and methodology of computing science, but not in the core of computing itself. Wilder has developed in [Wil81] a general theory of 'Laws' governing the evolution of mathematics, from which I have adapted the above statements. It remains a vast research project to assimilate Wilder's theory into our context, but it is an indispensable project if we want to adjust our expectation of progress in computing science to realistic hopes. Wilder's work also sheds some light into the real problems underlying the so called 'software crisis': The cultural lag of programming practice behind computing science and the absence of various cultural stresses may account for the abundance of programming paradigms without the evolution of rigorous standards of conceptual specifications.

I can only hope that I may contribute my small share to the slow process of bridging that gap and further the logical foundations of computer science.

Acknowledgements: I am indebted to many colleagues who encouraged me at several stages to pursue my research of model theoretic methods in computer science. Among them I would like to mention Erwin Engeler, Eli Shamir, Shimon Even, Vaughan Pratt, Catriel Beeri, Saharon Shelah, David Harel and Yuri Gurevich. I am also indebted to the graduate students of my department whose work contributed to my understanding. Among them are Ariel Calò, Yaniv Bargury, Yachin Pnueli and Avy Sharel. I would like to thank S. Ben–David who allowed me to include his unpublished results in section 5 and Y. Pnueli, who helped writing section 7. Finally I would like to thank D. Gabbay for inviting me to write this chapter and for his insisting that I give him this version for publication.

2 The Set Theoretic Modelling of Syntax and Semantics

Model theory is the mathematical (set theoretical) study of the interplay between Syntax and Semantics. Historically it has its roots in the various attempts of reducing first mathematics to logic (Frege, Hilbert), then logic

For M.N.Y., who emerged into my life
while I was preparing this chapter.

1 Introduction

The purpose of this chapter is to give an account of those aspects of model theory which we think are relevant to theoretical computer science. We start with a general outline of the evolution of model theory, which will serve as an exposition of the major themes. In the subsequent section we shall elaborate some of these themes and put them into the context of theoretical computer science.

We assume the reader is familiar with the basics of First Order Logic, Computability Theory, Complexity Theory and Basic Algebra. Whenever possible we shall refer to textbooks and monographs rather than the original papers. Only material not treated in standard texts will be quoted in the original (or by referring to a subsequent paper which contains the result in the most readable form).

This chapter is not meant to be an exhaustive scholarly survey of model theoretic methods in theoretical computer science. It is more of a personal guided tour into a well mapped but still largely unexplored landscape. It has definite autobiographical traits. No author can completely escape that. It proposes to some extent a unifying view which ultimately should lead to the disappearance of the personal touch. However, for that to happen more research and reinterpretation of classical results is needed. Logic and model theory are relatively old disciplines which enjoy renewed interest. They can serve as one explanatory paradigm for foundational problems in theoretical computer science. But the gap between the traditional logicians and mathematicians and the working computer scientists is first of all cultural in the sense of R. Wilder's [Wil81]. His studies deserve special attention especially when one has in mind the evolution and development of programming languages, operating systems, user interfaces and other paradigms of computing, but also in addressing foundational questions, cf. [Mak88].

Wilder's studies clearly show several phenomena: that the evolution of concepts to widely accepted norms of practice takes much longer and needs more than just the availability of such concepts; that the evolution of concepts is not due to individuals but is embedded in one (or several

Contents

Model Theory and Computer Science: An Appetizer

J.A. Makowsky
Department of Computer Science
Technion - Israel Institute of Technology, Haifa, Israel

January 27, 1995

To appear as chapter I.6 in volume I of the 'Handbook of Logic in Computer Science' (S. Abramsky, D.M. Gabbay and T.S.E. Maibaum eds), Oxford University Press 1992.

Abstract

We first review the main developments of model theory as it evolved as a branch of mathematical logic and examine which developments have a potential for research in theoretical computer science. We then discuss those aspects of model theory which we think have the greatest impact on theoretical computer science in further detail. Our presentation is not topic oriented but method oriented.