(viii) J.A. Makowsky and Y.B. Pnueli, Computable Quantifiers and Logics over Finite Structures, in "Quantifiers: Logics, Models and Computation, Volume I", M. Krynicki, M. Mostowski and L.W. Szczerba eds., Kluwer Academic Publishers, 1995, pp. 313-357.

Conference Publications

(ix) Y. Bargury and J.A.Makowsky, The expressive power of transitive closure and 2–way multihead automata, Computer Science Logic, Refereed papers from CSL'91, LNCS vol. 626 (1992) pp. 1-14.

(x) A. Calò and J.A. Makowsky, Ehrenfeucht–Fraïssé games for transitive closure, Logical Foundations of Computer Science–Tver '92, LNCS vol. 620, 1992, pp. 57-69.

(xi) J.A. Makowsky and Y.B. Pnueli, Oracles and Quantifiers. Computer Science Logic, Refereed papers from CSL'93, LNCS, vol. 832, 1994, pp. 189-222.

(xii) J.A. Makowsky, Capturing Complexity Classes with Lindström Quantifiers, Invited Lecture, Proceedings of the 19th International Symposium, MFCS'94 Kosice, Slovakia, LNCS vol. 841, 1994, pp. 68-71.

(xiii) J.A. Makowsky, Capturing Relativized Complexity Classes with Lindström Quantifiers, Invited Lecture, *The Foundational Debate*, W. DePauli-Schimanovich ed., Kluwer Academic Publishers, 1995, pp. 133-140.

(xiv) J.A. Makowsky, The impact of model theory on computer science, Proceedings of the IX Congress of Logic, Methodology and Philosophy of Science, Uppsala 1991, D. Prawitz et al. eds., North Holland 1994, pp. 239-262.

(xv) J.A. Makowsky and Y.B. Pnueli: Logics Capturing Relativized Complexity Classes Uniformly, Proceedings of LCC'94, Logic and Computational Complexity, D. Leivant ed., Lecture Notes in Computer Science vol. 960, 1995, pp. 463-479.

(xvi) J.A. Makowsky and E. Ravve, Incremental model checking for decomposable structures, Mathematical Foundations of Computer Science 1995, J. Wiedermann and P. Hajek eds., Lecture Notes in Computer Science, vol. 969 (1995), pp. 540-551.

(xvii) J.A. Makowsky, Invariant Definability, in *Computational Logic and Proof Theory*, Proceedings of the 5th Kurt Gödel Colloquium, KGC'97, Vienna, August 1997, LNCS vol. 1289 (1997), pp. 186-202.

Relevant Publications: J.A. Makowsky

Publications in Refereed Journals

Computer Science Logic

(i) J.A. Makowsky and Y.B. Pnueli, Arity vs. Alternation in Second Order Logic, Annals of Pure and Applied Logic, vol. 78 (1-3), 1996, pp. 189-202. Conference version in: Logical Foundations of Computer Science–St. Petersburg 1994, LNCS, vol. 813, 1994, pp. 240-252.

(ii) M. Kaminski, J. Makowsky and M. Tiomkin, Extensions for open default theories via the domain closure assumption, to appear in *Logic and Computation*, 1997.
Conference version in: M. Kaminski, J. Makowsky and M. Tiomkin, Extensions for open default theories via the domain closure assumption, in *Proceedings of the 5th European Workshop on Logics in Artificial Intelligence - JELIA'96*, J.J. Alfers, L.M. Pereira, and E. Orlowska eds., Springer, Berlin 1996, pp. 373-387 (*Lecture Notes in Artificial Intelligence* **1126**).

(iii) J.A. Makowsky and Y. Pnueli, Oracles and Lindström Quantifiers on Ordered Structures, to appear in *Information and Computation*, 1997

Theory of Algorithms

(iv) J.A. Makowsky and A. Sharell, On the average case complexity of SAT for symmetric distributions, Logic and Computation, vol. 5.1 (1995), pp. 71-92.

(v) G.Venkatesan, U.Rotics, M.S.Madanlal, J.Makowsky and C.Pandu Rangan, Restrictions of Minimum Spanner Problems, *Information and Computation*, 136 (1997) pp. 143-164.

(vi) G. Kogan and J.A. Makowsky, The Complexity of Schur Functions in Finite Fields of Characteristic 2 , submitted to Computational Complexity, October 1997.

Chapters in Research Monographs

(vii) J.A.Makowsky, Model Theory and Computer Science: An Appetizer, Chapter I.6 in the "Handbook of Logic in Computer Science, vol. 1 (Background: Mathematical structures)", S. Abramsky, D.M. Gabbay, T.S.E. Maibaum eds., Oxford University Press, 1992, pp. 763-814.

(i) K. Ma and J. von zur Gathen, Counting value sets of functions and testing permutation functions. Abstracts Int. Conf. Number Theoretic and Algebraic Methods in Computer Science, Moscow, Russia, 1993, 62-65. (See 6.iv.)

(ii) J. von zur Gathen and I. E. Shparlinski, Finding points on curves over finite fields. Proc. 36th Annual IEEE Symp.Foundations of Computer Science, Milwaukee WI, 1995, 284-292.

(iii) J. von zur Gathen and J. Gerhard, Arithmetic and factorization of polynomials over $\mathbf{F}_2$. Proc. ISSAC'96, Zürich, Switzerland, ACM Press, 1996, 1-9.

(iv) J. von zur Gathen and M. Nöcker, Exponentiation in finite fields: theory and practice. Proc. AAECC'97, Toulouse, France, Springer Lecture Notes in Computer Science, 1997, 88-113.

(v) J. von zur Gathen and J. Gerhard, Computing Taylor shifts and solving difference equations, Proc. ISSAC'97, Maui HI, USA, ACM Press, 1997, 40-47.

Articles in books

(i) Parallel linear algebra. In *Synthesis of parallel algorithms*, ed. J. Reif, Morgan & Kaufman, 1993, 573-617.

Non-refereed publications

(i) A polynomial factorization challenge. SIGSAM Bulletin **26** (1992), 22-24.

(ii) Faktorisieren von Polynomen. Computeralgebra in Deutschland, Fachgruppe Computeralgebra, Passau und Heidelberg, 1993, 20-22 and 25-26.

(iii) Counting curves over finite fields. Abstracts of papers presented to the AMS **14** (1993), 575.

Relevant Publications: J. von zur Gathen

Refereed Publications

(i) J. von zur Gathen, Processor-efficient exponentiation in finite fields. Inform. Process. Letters **41** (1992), 81-86, Zbl. 762.68026.

(ii) J. von zur Gathen and V. Shoup, Computing Frobenius maps and factoring polynomials. Computational complexity **2** (1992), 187-224. MR 94d:12011, Zbl. 778.11076.

(iii) S. Gao and J. von zur Gathen, Berlekamp's and Niederreiter's polynomial factorization algorithms. Contemporary Mathematics **168** (1994), 101-116. MR 95f:11106.

(iv) K. Ma and J. von zur Gathen, Tests for permutation functions. Finite Fields and Their Applications **1** (1995), 31-56.

(v) K. Ma and J. von zur Gathen, The computational complexity of recognizing permutation functions. Computational complexity **5** (1995), 76-97.

(vi) J. von zur Gathen and J. Weiß, Bivariate homogeneous decompositions. Journal of Symbolic Computation **19** (1995), 409-434.

(vii) J. von zur Gathen, M. Karpinski, and I. Shparlinski. Counting curves and their projections. Computational complexity **6** (1996), 64-99.

(viii) J. von zur Gathen and I. E. Shparlinski, Components and projections of curves over finite fields. SIAM J. Computing (1997), to appear.

(ix) S. Gao, J. von zur Gathen, and D. Panario, Gauss periods: orders and cryptographical applications. Mathematics of Computation (1997), to appear.

(x) J. von zur Gathen and S. Hartlieb, Factoring modular polynomials. Journal of Symbolic Computation, to appear.

(xi) J. von zur Gathen and I. E. Shparlinski, Orders of Gauss periods in finite fields. AAECC, to appear.

Publications in Conference Proceedings

Preliminary version of all Refereed Publications, except (vi), appeared at various conferences: FOCS, STOC, ISSAC, Latin, ISAAC. Furthermore:

Budget Details

As the research is theoretical, no special equipment is needed. Both, the Computer Science Department of the Technion and and the Mathematics Department in Paderborn are well equipped with word processing and computing facilities to accomodate all our needs.

The funding will be spend mostly for funding research assistants and graduate students working in topics related to this proposal. As smaller part of the money will be used for travel, communication cost (fax, telephone, express mail), acquisition of literature and other non-durable equipment.

There is a natural candidate for the one of the Haifa research assistant positions: G. Kogan, a new immigrant from Russia. He arrived in Israel in 1995 from Nishny Novgorod University and has many quite spectacular results concerning the computation of counting functions in finite fields (see preliminary results above). Another two research assistants (5o%, Ph.D. students) will be added to each of the two teams and have to be determined later.

[Val79] L.G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.

[MR96] J.A. Makowsky and E. Ravve. Translation schemes and the fundamental problem of database design. In *Conceptual Modeling - ER'96*, volume 1157 of *Lecture Notes in Computer Science*, pages 5–26. Springer Verlag, 1996.

[MRST97] W. McCuaig, N. Robertson, P. Seymour, and R. Thomas. Permanents, pfaffian orientations, and even directed circuits. In *STOCS'97*, pages 402–405. ACM, 1997.

[Ott97] M. Otto. *Bounded Variable Logics and Counting*, volume 9 of *Lecture Notes in Logic*. Springer Verlag, 1997.

[Pap94] C. Papadimitriou. *Computational Complexity*. Addison Wesley, 1994.

[PZ83] C. Papadimitriou and E. Zachos. Two remarks on the power of counting. In *6th GI Conference in Theoretical Computer Science*, volume 145 of *Lecture Notes in Computer Science*, pages 269–276. Springer Verlag, 1983.

[Raz87] A. Razborov. Lower bounds on the size of bounded-depth networks over a complete basis with logical addition. *Mathematical Notes of the Academy of Sciences of the USSR*, 41:333–338, 1987.

[Rei93] J. Reif, editor. *Parallel linear algebra*. Morgan & Kaufman, 1993.

[Sha93] A. Shamir. On the generation of polynomials which are hard to factor. In *Proceedings of the 25th Annual ACM Symposium on the Theory of Computing*, pages 796–804, 1993.

[Sip97] Michael Sipser. *Introduction to the theory of computation*. PWS Publishing Company, Boston, 1997.

[Smo87] R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proc. ACM Symp. on Theory of Computing*, pages 77–82, 1987.

[SS76] E. Specker and V. Strassen, editors. *Komplexität von Entscheidungsproblemen*, volume 43 of *Lecture Notes in Computer Science*. Springer Verlag, 1976.

[Str90] V. Strassen. Algebraic complexity theory. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume 1, chapter 11. Elsevier Science Publishers, 1990.

[Str94] H. Straubing. *Finite Automata, Formal Logic, and Circuit Complexity*. Progress in Theoretical Computer Science. Birkhäuser, 1994.

[vzGH96] J. von zur Gathen and S. Hartlieb. Factoring modular polynomials. In *Proc. ISSAC '96, Zürich, Switzerland*, pages 10–17. ACM press, 1996.

[GJ79a] M. R. Garey and D. S. Johnson. *Computers and intractability: A guide to the theory of NP-Completeness*. W. H. Freeman, San Francisco CA, 1979.

[GJ79b] M.G. Garey and D.S. Johnson. *Computers and Intractability*. Mathematical Series. W.H. Freeman and Company, 1979.

[Joh90] D.S. Johnson. A catalog of complexity classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume 1, chapter 2. Elsevier Science Publishers, 1990.

[KMa] G. Kogan and J.A. Makowsky. The complexity of Schur functions in finite fields of characteristic 2. May 1997, submitted.

[KMb] G. Kogan and J.A. Makowsky. Computing Schur functions for Borchardt matrices. in preparation.

[KMc] G. Kogan and J.A. Makowsky. Computing the permanent over fields of characteristic 3: Where and why it becomes difficult. Technion–Israel Institute of Technology, February 1996, Journal version in preparation.

[Kog] G. Kogan. The complexity of Schur functions over finite fields. Research proposal for candidacy examination, June 1997.

[Kog96] G. Kogan. Computing the permanent over fields of characteristic 3: Where and why it becomes difficult. In *FOCS'97*, pages 108–114. IEEE, 1996.

[Maka] J.A. Makowsky. A logic capturing P/poly. in preparation.

[Makb] J.A. Makowsky. Translations, interpretations and reductions. Lecture Notes of a course given at ESSLLI'97 in Aix-en-Provence, August 1997.

[Mak97] J.A. Makowsky. Invariant definability. In *Computational Logic and Proof Theory*, volume 1289 of *Lecture Notes in Computer Science*, pages 186–202. Springer Verlag, 1997.

[MR95] J.A. Makowsky and E.V. Ravve. Incremental model checking for decomposable structures. In *Mathematical Foundations of Computer Science (MFCS'95)*, volume 969 of *Lecture Notes in Computer Science*, pages 540–551. Springer Verlag, 1995.

References

[Bar96] A.I. Barvinok. Two algorithmic results for the traveling salesman problem. *Mathematics of Operations Research*, 21:65–84, 1996.

[BCS97] P. Bürgisser, M. Clausen, and M.A. Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren*. Springer Verlag, 1997.

[BCSSar] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation*. Springer Verlag, to appear.

[BG92] R. Beigel and J. Gill. Counting classes: Threshols, parity, mods and fewness. *Theoretical Computer Science*, 103:3–23, 1992.

[BS90] R.B. Boppana and M. Sipser. The complexity of finite functions. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume 1, chapter 14. Elsevier Science Publishers, 1990.

[CH90] J. Cai and L. Hemachandra. On the power of parity polynomial time. *Mathematical Systems Theory*, 23:95–106, 1990.

[Coo71] S. A. Cook. The complexity of theorem proving procedures. In *Proc. 3rd Ann. ACM Symp. Theory of Computing*, pages 151–158, 1971.

[Cou] B. Courcelle. The monadic second–order theory of graphs VI: On several representations of graphs by logical structures. To appear in Discrete and Applied Mathematics.

[Cou92a] B. Courcelle. Monadic second order definable graph transductions. In *CAAP'92*, volume 581 of *Lecture Notes in Computer Science*, pages 124–144. Springer, 1992.

[Cou92b] B. Courcelle. Monadic second-order logic of graphs vii: Graphs as relational structures. *Theoretical Computer Science*, 101:3–33, 1992.

[EF95] H. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer Verlag, 1995.

[vzG86] J. von zur Gathen. Parallel arithmetic computations: a survey. In *Proc. 12th Int. Symp. Math. Foundations of Computer Science*, volume 233 of *Lecture Notes in Computer Science*, pages 93–112. Springer Verlag, 1986.

[vzG87] J. von zur Gathen. Permanent and determinant. *Linerar Algebra and its Applications*, 96:87–100, 1987.

[vzG93] J. von zur Gathen. *Parallel linear algebra*, pages 573–617. Morgan & Kaufman, 1993.

An important first step was to settle on a general way in which such rings can generally and efficiently be presented.

Besides the above, we have concentrated on rings of the form $\mathbb{Z}_n[x]/(f)$ i.e., the factorization of univariate polynomials modulo an integer n. Much of our intuition from factoring over a field goes overboard: the innocuous polynomial x factors in a nontrivial way ([Sha93]), and a polynomial may have an exponential number of irreduclible factors. But we found a way of computing in polynomial time representations for these many factors [vzGH96].

6 Detailed Account of Available Resources

As the research is theoretical, no special equipment is needed. Both the Computer Science Department of the Technion and the Mathematics Department in Paderborn are well equipped with word processing and computing facilities to accomodate all our needs. The first applicant's research group comprises six Ph. D. students and Postdoctoral Fellows, funded by the university and outside resources. It is expected that some of these young people will collaborate on some of the issues proposed in this Project.

The funding will be spend mostly for funding research assistants and graduate students working in topics related to this proposal. As smaller part of the money will be used for travel, communication cost (fax, telephone, express mail), acquisition of literature and other non-durable equipment.

Schur Function		1-semi-unitary Matrix of Characteristic		
		0	2	3
det		**P**	**P**	**P**
per	(new for char 0,3)	#**P**	**P**	**P**
ham	(new)	#**P**	$\bigoplus_2$ **P**	$\bigoplus_3$ **P**
cycle$_-$	(new)	#**P**	$\bigoplus_2$ **P**	$\bigoplus_3$ **P**
cycle$_+$	(new)	#**P**	$\bigoplus_2$ **P**	$\bigoplus_3$ **P**

Table 3: The complexity of Schur functions on 1-semi-unitary matrices

Schur Function		General Matrix of Characteristic		
		0	2	3
det		**P**	**P**	**P**
per		#**P**	**P**	$\bigoplus_3$ **P**
ham		#**P**	$\bigoplus_2$ **P**	$\bigoplus_3$ **P**
cycle$_-$		#**P**	$\bigoplus_2$ **P**	$\bigoplus_3$ **P**
cycle$_+$		#**P**	$\bigoplus_2$ **P**	$\bigoplus_3$ **P**

Table 4: The complexity of Schur functions on general matrices, cf. [Val79, PZ83]

able in LFP.

5.4 Parallel algebraic complexity

Besides the sequential complextiy issues discussed so far, we also have results on parallel complexity. A chapter [vzG93] in the most extensive text [Rei93] on the theory of parallel computation collects those methods and results. Two "complexity classes" are central for parallel linear algebra: an "arithmetic" one defined by the determinant, and a "Boolean" one defined by the rank of matrices.

Many natural problems were proven complete for one of these classes, and there are also exciting open problems: how does the independence of vectors or the singularity of matrices fit into these hierarchies?

5.5 Complexity of finite rings

Several preliminary results on the computational structure theory of finite rings have been obtained at Paderborn. For example, we have a polynomial-time computation of the radical of a finite ring (unpublished); a further research goal is to find (relative) lower bounds, i.e., reduce some other problems to this one.

Schur Function		Borchardt Matrix or $C(\bar{x}, \bar{y})$ of Characteristic		
		0	2	3
det		**P**	**P**	**P**
per		**P**	**P**	**P**
ham	(new)	**P**	**P**	**P**
$cycle_-$	(new)	**P**	**P**	**P**
$cycle_+$	(new)	#**P**	**P**	$\bigoplus_3$ **P**

Table 1: The complexity of Schur functions on Borchardt matrices or matrices of the form $C(\bar{x}, \bar{y})$

Schur Function		Unitary Matrix of Characteristic		
		0	2	3
det		**P**	**P**	**P**
per	(new for char 0,3)	#**P**	**P**	**P**
ham	(new)	#**P**	**P**	$\bigoplus_3$ **P**
$cycle_-$	(new)	#**P**	**P**	$\bigoplus_3$ **P**
$cycle_+$	(new)	#**P**	**P**	$\bigoplus_3$ **P**

Table 2: The complexity of Schur functions on unitary matrices

This analysis seems to suggest a hierarchy of difficulty: The determinant is always easy. For a given class of matrices $\mathcal{M}$, whenever the permanent is difficult, so are *ham* and $cycle_\pm$. Finally, whenever $cycle_-$ is hard, so is $cycle_+$. Note that the unitary matrices and $C(\bar{x}, \bar{y})$ are closed under some kind of multiplication and the $C(\bar{x}, \bar{y})$ matrices are themselves unitary.

Problem 5.2 *Is this hierachy of complexity of Schur functions true for any multiplicatively closed class of matrices ?*

5.3 Logics capturing circuit and counting classes

As already mentioned in section 3 we have succeded in characterizing $\bigoplus_2$ **P** in logical terms, cf. theorem 3.3 using second order counting quantifiers. A similar result can also be formulated and proved for the class $\bigoplus_2$ **L**, where polynomial time is replace by logarithmic space.

We have also found logical characterizations of the classes **P**/*poly*, **NP/poly** and **NL/poly** in terms of weak invariant definability in *LFP*, respective the logics capturing **NP** and **NL** respectively, cf. [Maka].

The relationship between **P**/*poly* and $\bigoplus_2$ **P** is open. But it can be attacked by trying to show that second order counting is/isn't weakly invariantly defin-

5.1 Permanents of unitary matrices in fields of characteristic 3

In [Kog96, KMc] we proposed various new techniques for analyzing the complexity of the permanent over fields of characteristic 3. It is also known that if in this case (characteristic 3) the permanent were polynomial time computable, then also Hamiltonicity would be computable in randomized polynomial time. We showed in [Kog96]:

Theorem 5.1 *For the following cases there are polynomial time algorithms (in characteristic 3) for computing the permanent:*

- *For unitary matrices,*
- *For matrices M with $rg(MM^T - I) = 1$, (these matrices are called 1-semi-unitary).*
- *On the other hand, for matrices M with $rg(MM^T - I) > 1$, the problem is as difficult as the general case, i.e. $\bigoplus_3 \mathbf{P}$-complete.*

The proofs of these results are to be found in [Kog96, KMc].

5.2 Schur functions in characteristic 2 and 3

So far we have discussed the complexity of various Schur functions for Borchardt matrices in fields of arbitrary characteristic. The unitray case was discussed only in characteristic 3. Clearly the determinant can always be computed in polynomial time.

We can show that in characteristic 0, the permanent (and *ham* and $cycle_{\pm}$) all are $\#\mathbf{P}$-complete. For the permanent this is well known, [Val79]. The other cases are, to the best of our knowledge, new.

We have analyzed the complexity of these Schur functions completely also for characteristic 2 and 3. Our results can be summarized in the following tables.

We present in each table the complexity of the Schur function indicated for a certain class of matrices. Figure 1 gives the picture for Borchardt matrices and the matrices of the form $C(\bar{x}, \bar{y})$. The latter are also proved to be unitary in [KMb]. Figure 2 gives the picture for unitary matrices, figure 3 for 1-semi-unitary matrices, and, to complete our understanding, figure 4 for general matrices.

plexity of determining whether two finite rings are isomorphic? How difficult is it, given a finite ring R, to compute its

- radical
- group of units
- Wedderburn decomposition?

The corresponding question can also be asked in the descriptive context: In which logics are the above listed definable? The methods applied in [MR95, MR96, Makb] will be applied to such questions.

4 Distribution of Research

Both investigators studied mathematics at the ETH-Zurich and participated in the early 70ies in the Specker-Strassen Seminar, cf. [SS76], on Logic and Complexity, and took Prof. Strassen's course on Algebraic Complexity at Zurich University. Later their respective research interests concentrated in complementary fields: von zur Gathen did most of his work in algebraic complexity theory and computer algebra, whereas Makowsky worked in logic and its application to various fields in computer science. But both investigators kept being informed of each other's work and met personally at regular intervals.

In this project this complementarity is a major asset. The combination of methods from logic (descriptive complexity) and algebraic geometry (algebraic complexity) should lead to new ideas in understanding the complexity of combinatorial counting functions, the isomorphisms problem of finite rings, and of other properties of finite algebraic structures.

The Haifa group (Makowsky and Ph.D. students) will investigate the Turing complexity and logical definability of Schur functions, and the logical characterizability of counting complexity classes. They will also study the descriptive complexity of ring and matrix properties.

The Paderborn group (von zur Gathen and Ph.D. students) will investigate the complexity of the same function classes in various computational models for algebraic circuit complexity, and the comparative merits of such models. They will also study the algorithmic complexity of the ring isomorphisms problem.

The combinatorial counting functions and the ring isomorphism problem serve as joint benchmarks. Both groups depend on each other's work to get a thorough understanding of the computational difficulty of combinatorial counting functions.

5 Preliminary Results

In the first three subsections, we describe shortly some preliminary results which have been obtained in the last year.

In this proposal we mostly deal with $det(M)$, $ham(M)$ and $per(M)$. A generalization of these three Schur functions can be obtained as follows.

Definition 3.4 *We denote by $Cyc(\pi)$ the set of cycles of a permuation π (as a set of elements, i.e $\gamma \in Cyc(\pi)$ iff there is $g \in \gamma$ such that $\gamma = \{\pi^j(g)\}$. For an n-vector $\bar{\lambda}$ we put*

$$c_{\pi,\pm}(\bar{\lambda}) = \prod_{\gamma \in Cyc(\pi)} \left(1 \pm \prod_{i \in \gamma} \lambda_i\right)$$

and

$$cycle_{\pm}(M, \bar{\lambda}) = \sum_{\pi \in S_n} c_{\pi,\pm}(\bar{\lambda}) \cdot \prod_{i=1}^{n} m_{i,\pi(i)}.$$

Clearly we have $per(M) = cycle_{-}(M, \bar{0}) = cycle_{+}(M, \bar{0})$ and $ham(M) = cycle_{-}(M, (0, 1, \ldots, 1)^T)$. To express the determinant we use limits and get

$$det(M) = \lim_{\epsilon \to 0} (\epsilon^n \cdot cycle_{-}(M, \frac{1}{\epsilon}\bar{1}) \cdot (-1)^n$$

Research in the complexity of permanents (and other Schur functions) is not very active. But the permanent has found many surprising applications in complexity theory, cf. [Pap94]. Recently McCuaig, Robertson, Seymour and Thomas, cf. [MRST97], have announced polynomial algorithms for deciding whether a 0-1 matrix M can be modified by replacing some 1's by -1's such that the determinant $det(M_1)$ of the new matrix M_1 equals $per(M)$. This problem was also shown to be equivalent to the even circuit problem for directed graphs and also to the problem of finding Pfaffian orientation in a bipartite graph.

3.7 Cryptography

A speculative thought that we will always have in the back of our minds is: can this theory be used for cryptographic purposes? We cannot realistically hope to make major progress here, but this seems worth a try. In particular, in the light of corollary 3.2, showing that $\bigoplus_m \mathbf{P} = \mathbf{P}$ for some $m \geq 2$, and hence $\mathbf{RP} = \mathbf{NP}$, would imply that no one-way functions exist. This, admittedly, unlikely result, would make most of the complexity theoretic cryptography vacuous. On the other hand, proving that some problem is in $\mathbf{NP} - \mathbf{RP}$ would give complexity theoretic cryptography its hitherto missing theoretical foundation.

3.8 Descriptive and algorithmic complexity of properties of finite rings

A further topic will be the computational structure theory of finite rings. The ultimate goal is to answer questions such as the following. What is the com-

We can generalize a theorem due to Toda, cf. [Pap94]. Recall that **RP** is the class problems solvable by a random polynomial Turing machine. Note that $\mathbf{RP} \subseteq \mathbf{NP}$. For a class of problems $\mathbf{C}$, $\mathbf{P^C}$ ($\mathbf{RP^C}$) denotes class of problems solvable by **P**-oracle (**RP**-oracle) Turing machines using oracles from **C**.

Theorem 3.1 *For every* $m \geq 2$ $\mathbf{NP} \subseteq \mathbf{RP}^{\bigoplus_\mathbf{m} \mathbf{P}}$.

Toda proved this for $m = 2$ and also that $\mathbf{NP} \subseteq \mathbf{P}^{\#\mathbf{P}}$. It is open whether $\mathbf{NP} \subseteq \bigoplus_\mathbf{m} \mathbf{P}$.

Corollary 3.2 *If* $\bigoplus_m \mathbf{P} = \mathbf{P}$ *for some* $m \geq 2$ *then* $\mathbf{RP} = \mathbf{NP}$.

The search for logics capturing **P** or **L** without the order present lead to the introduction of First Order Counting quantifiers, cf. [Ott97], counting the number of elements with certain properties modulo k. In this research we propose the introduction of Second Order Counting quantifiers, counting the number of relationswith certain properties modulo k.

Second Order Logic augmented with such quantifiers still defines classes of structures which are in **PSpace**. As $\bigoplus_2 \mathbf{P}$ is closed under complement, it is not difficult to prove:

Theorem 3.3 (Makowsky) *LFP augmented with second order counting modulo 2 captures the class* $\bigoplus_2 \mathbf{P}$.

As it is open whether $\bigoplus_k$ is closed under complement for general k, the corresponding theorem is not obvious. But the attempt to find a logic capturing the class $\bigoplus_k \mathbf{P}$ will also clarify the closure properties of $\bigoplus_k \mathbf{P}$.

3.6 Schur functions

Computing the permanent of matrices associated with graphs have often graph theoretic interpretations. The permanent of an adjacency matrix corresponds to the number of perfect matchings. The number of hamiltonian paths of a graph G can also be expressed using some Schur function of its adjacency matrix. But every Schur function of a matrix can be polynomially reduced to the permanent. This is not true, if we restrict the class of matrices. Recall the general Schur function of a matrix M is given by an expression

$$F = \sum_{\pi \in S_n} c_\pi \cdot \prod_{i=1}^{n} m_{i,\pi(i)}$$

Clearly the permanent is the case where $c_\pi = 1$ for every permutation, and $det(M)$ is the case where c_π is the parity of π. If c_π is the characteristic function of the permutations with exactly one cycle this function F is denoted by

$$ham(M)$$

the descriptive point of view.

3.4 Combinatorial Counting

Several open problem in complexity can be reduced to problems about the specific intrinsic difficulty of solving a particular problem. The Hamiltonian path problem is known to be **NP**–complete but no polynomial time algorithm is known, not even if randomization is allowed. The problem of computing the determinant of a matrix is solvable in polynomial time, but computing the permanent is considered hard, although its exact complexity is not known, [Val79, vzG87], but it is complete for the counting complexity class #**P**, defined below.

Ever since Valiant's work [Val79], the permanent has become a benchmark problem to test new algorithmic ideas or lower bound techniques. Recently, several papers were published exhibiting fast (polynomial) algorithms for the computation of the permanent on restricted classes of matrices. Barvinok [Bar96] showed that the problem is solvable in polynomial time for matrices of fixed rank k over fields of arbitrary characteristic, but his algorithm depends exponentially on k. Kogan showed, [Kog], that the same holds for k-sparse matrices, i.e. matrices with at most k non-zero elements in each row (or column). Kogan also showed that the problem is is solvable in polynomial time for unitary matrices in fields of characteristic 3, [Kog96, KMc]. Our renewed interest in the complexity of the permanent was triggered by these recent result of our student G. Kogan [Kog96, KMc, Kog, KMa, KMb].

Bürgisser (1997, unpublished) has recently obtained interesting results on the complexity of the permanent and other immanants; we will also explore that direction.

3.5 Complexity classes which count

For our purpose two polynomial counting classes from the literature are important. Recall that a counting Turing machine is a non-determinsitic Turing machine which counts the number of accepting paths of a computation. #**P** is the class of functions computable by a non-deterministic polynomial time counting Turing machine. $\bigoplus_2 \mathbf{P}$ is the class of problems decidable by a non-determinsitic polynomial time counting Turing machine which counts modulo 2. We also study the classes $\bigoplus_m \mathbf{P}$ where m is a natural number ≥ 2, first seriously studied in [CH90] and [BG92].

$\bigoplus_m \mathbf{P}$ is the class of problems decidable by a non-determinsitic polynomial time counting Turing machine which counts modulo m. All these classes contain **P** (determinsitic polynomial time). Computing the permanent of $(0, 1)$-matrices and counting the number of Hamiltonian paths of a directed graph is #**P**-complete (via polynimal Turing reductions). Counting the number of Hamiltonian paths of a directed graph modulo 2 is $\bigoplus_2 \mathbf{P}$-complete.

a square matrix is given by a certain sum with alternating signs; the permanent is given by the same sum, but with all signs set to +1.)

In recent years, a model proposed by Blum, Smale and Shub, cf. [BCSSar] has been widely discussed. Some interesting results have been proved in it (such as the completeness of Julia sets), but it has failed to gain wide acceptance because some of its features scare Boolean complexity theorists away: the Halting Problem for (ordinary, Boolean) Turing machines can be solved in this model in essentially linear time.

A different model was proposed in [vzG86] which does not have these flaws, but it has never been fully evaluated. Its description is circuit-based and includes various notions of uniformity for constructing the circuit and its constants. The proposed research will investigate this model further: How can it be turned into a model of an "arithmetic Turing machine with uniform constants"? Is that the better approach, or is the circuit formulation better? Which of the results in Blum–Shub–Smale–model carry over to this model?

A further perceived flaw in the Blum–Shub–Smale-model is its neglect of the degree of the polynomials under consideration. It then becomes trivial to separate parallel complexity classes corresponding to $\mathbf{NC}^k$ and $\mathbf{NC}^l$ for any $k < l$. Rather than being a "result", this triviality points to the requirement of including the degree in the definition of algebraic complexity classes if one wants to maintain an analogy with the Boolean situation. However, as soon as one allows branching in computations, one does not have the rather intuitive degree of polynomials, but rather the intricate notion of the degree of algebraic varieties, say over an algebraically closed field. The interesting research questions are: How does the degree behave under these computations? How should it be incorporated in the definition of complexity classes? What kind of lower bounds can we expect? Instead of the "log deg" lower bound for polynomials, it seems that one can only use "log log deg".

3.3 Descriptive Complexity

On of the remarkable successes in complexity theory was the emergence of logics which give high level descriptions of problems in certain complexity classes usually defined in low level machine complexity (Turing machines, circuits). The classes **P** and **NP** have been identified with Least Fixed Point Logic FLP and the existential fragment of Second Order Logic (but arbitrary first order quantification). The polynomial hierarchy **PH** corresponds to Second Order Logic. But in the case of **P** one has to restrict oneself to ordered (labeled) input structures, cf. [EF95]. Similar results over ordered structures exist for the lower complexity classes **L** and **NL** (using various transitive closure operators) and for $\mathbf{AC_0}$ using invariant definability in First Order Logic. For the latter, cf. [BS90, Str94, Mak97].

It is therefore natural to study not only complexity classes, but also particular properties of finite structures (graphs, groups, rings, fields, matrices) from

- We shall look for efficient ways of determing the structure of finite rings: computing the radical and the group of units, Wedderburn decomposition, isomorphism.

3 Comprehensive Description

3.1 Complexity Theory

Boolean Complexity Theory (a branch of Theoretical Computer Science) deals with the computational complexity of combinatorial problems. Complexity is measured usually in terms of an abstract machine model (Turing machines) and the minimum Time or Space (Memory) needed by such a machine to solve the problem. Polynomial Time Algorithms play a particularly prominent role, and the search for polynomial time algorithm is an ongoing theme. Showing that no such algorithm exists is usually harder. Next best to polynomial time algorithms are randomized polynomial time algorithms.

The theories revolving around the Boolean complexity class **NP** and the notion of **NP**-completeness are of fundamental importance and highly successful in Theoretical Computer Science, cf. [Coo71, GJ79a, Joh90, Sip97]. As an example, the initial thrust of modern cryptography was based on complexity related notions.

The major question in this area, *Cook's Hypothesis* that $\mathbf{P} \neq \mathbf{NP}$, has remained unresolved since the beginning, in 1971. It is today arguably one of the central open questions of mathematics and computer science, at par with, say, the Riemann Hypothesis or natural language translation.

From the beginning, the difficulty of the problem was realized, and researchers tried to make progress by not tackling the question itself, but rather variants that might be more easily amenable to a solution. The monotone circuit lower bounds are a successful instance of such an attack, cf. [BS90, Smo87, Raz87].

3.2 Algebraic Complexity

Another approach of this kind is *algebraic complexity theory*. The hope here is that the highly structured computations will make problems more manageable. Strassen and Valiant were the pioneers in the area, setting up the framework, asking the right questions, and giving many answers, cf. [Str90]. For a modern account and history, cf. [BCS97]. The hope was fulfilled at the "low" end of arithmetic computations, where matching upper and lower bounds were proved for many natural problems—the complexity theorist's dream come true. At the "high" end, corresponding to the Boolean **P** vs. **NP**, Valiant identified "p-complete" problems like the permanent, but the task of proving that it cannot be computed in polynomial time is still unresolved. (Recall that the determinant of

Our goals include

- The algebraic complexity of combinatorial counting functions over various (finite and infinite) fields and the computational model of Valiant-Strassen.
- The descriptive complexity of combinatorial counting functions in various extensions of second order logic.
- The circuit complexity of combinatorial counting functions in various models of algebraic nature.
- Transfer theorems between the three approaches of algebraic, descriptive and circuit complexity.
- The algorithmic and descriptive complexity of finite algebraic structures, such as finite rings.

2 Objectives

In our research we propose to work on the following problems:

- We shall study the complexity of Schur polynomials (which include the permanent and the hamiltonian polynomials) over finite fields. We investigate how far polynomial algorithms can be extended and search for parameters (such as rank of M or the rank of $MM^T - 1$) for which the problems gets hard.

  We currently work in generalizing the positive results to larger and the negative results to smaller classes of matrices. We also investigate to what extent these results are restricted to characteristic 3. The new techniques introduced so far make us optimistic in narrowing the gap between the polynomial and difficult(#**P**-complete, $\bigoplus_m$ **P**-complete) case of computing the permanent over finite fields.
- We shall look for logics which capture the complexity of combinatorial counting functions (and other multivariate polynomials). As most of these counting functions are in **PSpace** they can be defined withing First Order Logic augmented by partial fixed point operator PFP. Showing that they are definable within Second Order Logic SOL would establish their membership in the polynomial hierarchy **PH**. We shall in particular look for logics between SOL and PFP. Good candidates are SOL augmented by second order counting quantifiers, cf. section 3.
- We shall further investigate a circuit based model of computation for algebraic problems. Various notions of uniformity play a central role. A particular goal is to see which of the results in the Blum–Shub–Smale model carry over to this more realistic model, whose power is not that much greater than that of standard Boolean models.

1 Brief Description

Complexity theory deals with the algorithmic complexity of decision problems (is there a Hamiltonian path in a graph?), of search problems (find a Hamiltonian path in a graph), and counting problems (how many Hamiltonian paths are there in a graph?). There are several ways of understanding the inherent complexity of such problems.

- Find provable lower and upper bounds.
- Find suitable reductions to problems complete in some complexity class.

The second approach is rich in variations: we have some freedom in choosing

- the model of computation: boolean complexity, Turing complexity, algebraic complexity, parallel vs. sequential, etc...
- the class of reductions: many-one transductions, Turing reductions, circuit reductions, reductions expressible in some algebraic formalisms, reductions definable in some logical formalisms, etc...

In the last decades much work has been invested in the better understanding of the complexity of decision and search problems, but counting problems have been studied less prominently. In the GIF research 0293-80-06/93 (Definability and Computability over Finite Structures, 1995-1997), the second investigator has explored both decision and search problems. The continuation of this research into the complexity of counting functions was triggered by his work with G.Kogan on the complexity of the permanent and attempts to capture counting complexity classes with logical means, cf. section 5.

Combinatorial counting functions are often representable as multivariate polynomials over some field. In the case of adjacency matrices of graphs the permanent function computes the number of perfect matchings and the hamiltonian polynomial the number of hamiltonian circuits. The first investigator has done considerable work in the past in studying the algebraic complexity of the permanent and polynomials in general. Of particular relevance is his lower bound on the size of determinants of which the permanent is a projection. In this research project we want to study the complexity of combinatorial counting functions from various points of view.

In past twenty years the algorithmic complexity of computing related structures such as graphs, words, automata, etc. have been studied extensively, [GJ79b]. The study of descriptive complexity of graph properties was initiated by Courcelle in a sequence of papers [Cou92a, Cou, Cou92b]. To the best of our knowledge no systematic study of descriptive complexity of properties of groups or rings has been undertaken. As our study concentrates on combinatorial counting functions over finite fields and rings, we also plan to study the complexity of various ring invriants which allow us to solve the isomorphisms problem for finite rings.

Abstract

Complexity theory deals with the algorithmic complexity of decision problems, of search problems, and counting problems. There are several ways of understanding the inherent complexity of such problems: by finding exact or approximate upper and lower bounds; by finding reductions to equally complex problems; and by analyzing different computational models.

Combinatorial counting functions are often representable as multivariate polynomials over some field. In the case of adjacency matrices of graphs the permanent function computes the number of perfect matchings and the hamiltonian polynomial the number of hamiltonian circuits. In this research project we want to study the complexity of combinatorial counting functions from various points of view. The polynomial representation of these counting functions depends on an underlying algebraic structure, usually a field, or more generally, a ring.

Our specific goals include

- The algebraic complexity of combinatorial counting functions over various (finite and infinite) fields and the computational model of Valiant-Strassen.
- The descriptive complexity of combinatorial counting functions in various extensions of second order logic.
- The circuit complexity (sequential and parallel) of combinatorial counting functions in various models of algebraic nature.
- Transfer theorems between the three approaches of algebraic, descriptive and circuit complexity.
- The descriptive and algorithmic complexity of finite algebraic structures, such as defining or computing invariants of finite rings.

Algebraic and Descriptive Complexity of Combinatorial Counting Functions

Research Proposal for GIF project

Joachim von zur Gathen
Fachbereich Mathematik–Informatik
Universität-GH Paderborn
Paderborn, Germany
and
Johann A. Makowsky
Department of Computer Science
Technion–Israel Institute of Technology
Haifa, Israel

October 27, 1997