Universität zu Köln

Institut für Informatik
Prof. Dr. Ewald Speckenmeyer

UNIVERSITÄT ZU KÖLN

Pohligstr. 1
D–50969 Köln

Tel.: +49 221 470-5378
Fax: +49 221 470-5387
e–mail: theotag@informatik.uni–koeln.de

Programm des 28. Theorietages

Köln, 27. Febr. 1996

Hörsaal 301, Pohligstr. 1

9.25 Uhr	**Begrüßung**	
9.30 – 9.55 Uhr	U. Hertrampf, Trier	Über many-one lokal-selbstreduzierbare Mengen
9.55 – 10.20 Uhr	A. Kick, Karlsruhe	Erzeugung von Gegenbeispielen beim globalen μ-Kalkül Model checking
10.20 – 10.45 Uhr	M. Kutyłowski, K. Loryś, T. Wierzbicki, Paderborn, Wrocław	Complexity of Boolean functions on QRQW and EREW PRAMs
10.45 – 11.15 Uhr	**Pause**	
11.15 – 11.40 Uhr	J. Rethmann, E. Wanke, Düsseldorf	Storage Controlled Pile-Up Systems
11.40 – 12.05 Uhr	K. Jansen, J. Reiter, München, Trier	Approximation Algorithms for Register Allocation
12.05 – 12.30 Uhr	M. Böhm, P. Heusch, F. Meisgen, T. Seifert, E. Speckenmeyer, Köln	Precomputation–based Load Balancing
12.30 – 14.00 Uhr	**Pause**	
14.00 – 14.25 Uhr	K. Kühnle, E. W. Mayr, München	Exponential space computation of G bases
14.25 – 14.50 Uhr	U. Koppenhagen, E. W. Mayr, München	The Complexity of the Equivalence Problem for Commutative Semigroups
14.50 – 15.15 Uhr	O. Kullmann, Frankfurt	Worst case–Analyse und untere Schranken für bessere SAT-Algorithmen
15.15 – 15.45 Uhr	**Pause**	
15.45 – 16.10 Uhr	A. Hett, R. Drechsler, B. Becker, Freiburg	MORE: Alternative Implementation of BDD-Packages by Multi-Operand Synthesis
16.10 – 16.35 Uhr	Ch. Meinel, A. Slobodov, Trier	Ein neuer Reduktionsbegriff für OBDDs
16.35 – 17.00 Uhr	T. Theobald, Ch. Meinel, Trier	Zustandscodierungen und OBDD-Größen

Über many-one lokal-selbstreduzierbare Mengen

Ulrich Hertrampf
Universität Trier
Fachbereich IV – Theoretische Informatik
D-54286 Trier

Für $k \in \mathbb{N}$ heißt eine Menge A von natürlichen Zahlen *k-lokal-selbstreduzierbar*, wenn die charakteristische Funktion c_A so beschaffen ist, daß es eine polynomialzeitberechenbare Funktion f mit Werten in der Menge der k-stelligen booleschen Funktionen und eine Zahl $n_0 \geq k$ gibt, so daß

$$c_A(n) = f(n)(c_A(n-k), \ldots, c_A(n-1))$$

für alle $n \geq n_0$.

Für $k \geq 1$ heißt A *many-one-k-lokal-selbstreduzierbar*, wenn es eine polynomialzeitberechenbare Funktion g mit Werten in der Menge $\{1, \ldots, k\}$ und eine Zahl $n_0 \geq k$ gibt, so daß

$$c_A(n) = c_A(n - f(n))$$

für alle $n \geq n_0$.

Wir definieren für alle $k \geq 0$ die Komplexitätsklasse $\mathrm{LSR}_k^{\mathrm{P}}$ als den Abschluß der Menge aller k-lokal-selbstreduzierbaren Mengen unter polynomieller many-one Reduktion, d.h. $\mathrm{LSR}_k^{\mathrm{P}}$ besteht aus allen Mengen B, für die es eine Menge A gibt, so daß A k-lokal-selbstreduzierbar ist und $B \leq_m^{\mathrm{P}} A$.

Entsprechend definieren wir für alle $k \geq 1$ die Komplexitätsklasse $\mathrm{LSR}_{m,k}^{\mathrm{P}}$ als den Abschluß der Menge aller many-one-k-lokal-selbstreduzierbaren Mengen unter polynomieller many-one Reduktion, d.h. $\mathrm{LSR}_{m,k}^{\mathrm{P}}$ besteht aus allen Mengen B, für die es eine Menge A gibt, so daß A many-one-k-lokal-selbstreduzierbar ist und $B \leq_m^{\mathrm{P}} A$.

In [BS95] werden lokal-selbstreduzierbare Mengen untersucht und die folgenden Resultate erzielt:

a) $\oplus\mathrm{OptP} = \mathrm{LSR}_1^{\mathrm{P}} \subseteq \mathrm{LSR}_2^{\mathrm{P}} \subseteq \mathrm{MOD}_6\mathrm{PH}$

b) $k \geq 3 \implies \mathrm{LSR}_k^{\mathrm{P}} = \mathrm{PSPACE}$

c) $k \geq 6 \implies \mathrm{LSR}_{m,k}^{\mathrm{P}} = \mathrm{PSPACE}$

Wir ergänzen diese Resultate wie folgt:

d) $\oplus \cdot \mathrm{MOD}_3 \cdot \oplus\mathrm{P} \subseteq \mathrm{LSR}_2^{\mathrm{P}}$

e) $\mathrm{P} = \mathrm{LSR}_{m,1}^{\mathrm{P}} = \mathrm{LSR}_{m,2}^{\mathrm{P}} \subseteq \mathrm{LSR}_{m,3}^{\mathrm{P}} = \oplus \cdot \mathrm{NP}$

f) $\mathrm{MOD}_3 \cdot \oplus \cdot \mathrm{NP} \subseteq \mathrm{LSR}_{m,4}^{\mathrm{P}} \subseteq \mathrm{LSR}_{m,5}^{\mathrm{P}} \subseteq \mathrm{MOD}_6\mathrm{PH}$

Dabei benutzen wir für die unteren Schranken eine neue allgemeine Technik, die auch die Ergebnisse b) und c) von Beigel und Straubing noch einmal als leichte Folgerungen bringt.

Referenz

[BS95] R. Beigel, H. Straubing. *The Power of Local Self-Reductions.*
Proc. 10th IEEE Conference on Structures in Complexity Theory, 1995.

Erzeugung von Gegenbeispielen beim globalen μ-Kalkül Model checking

Alexander Kick

Lehrstuhl Informatik für Ingenieure und Naturwissenschaftler
Universität Karlsruhe
Am Fasanengarten 5
D-76128 Karlsruhe
Germany
Email: kick@ira.uka.de
Voice: +49 721 608 4337
FAX : +49 721 698675
WWW : http://liinwww.ira.uka.de/ kick

Temporal logic model checking is an automatic verification method for finite-state systems. An important feature of model checking is that a counterexample can be constructed when a temporal formula does not hold for the model. Such a counterexample can help locating the error in the system design. The very procedure of local model checking already constructs a witness for a formula in form of a tableau proof tree. However, local model checking has bad worst-case time and space complexities. For this and other reasons, global model checking has been applied much more often in practice. How to construct counterexamples in the case of global model checking is described in {Clarke 94} for the case that the temporal logic is restricted to fair CTL. This paper shows how counterexamples and witnesses for the much more expressive modal μ-calculus can be constructed if the model checking procedure is global. The witness construction presented in this paper is polynomial in the size of the model and the length of the formula.

Literatur:

- E. Clarke and O. Grumberg and K. McMillan and X. Zhao, Efficient Generation of Counterexamples and Witnesses in Symbolic Model Checking, School of Computer Science, Carnegie Mellon University, 1994, CMU-CS-94-204, Pittsburgh, PA 15213, October

Complexity of Boolean functions on QRQW and EREW PRAMs

Mirosław Kutyłowski,
University of Paderborn, mirekk@uni-paderborn.de

Krzysztof Loryś, Tomasz Wierzbicki,
Uniwersytet Wrocławski, Wrocław, Poland
lorys,tomasz@ii.uni.wroc.pl

We consider parallel random access machines (PRAMs) with restricted access to the shared memory resulting from handling congestion of memory requests. We study the (SIMD) QRQW PRAM model where multiply requests are queued and serviced one at a time. We also consider exclusive read exclusive write (EREW) PRAM and its modification obtained by adding a single bus.

For the QRQW PRAMs we consider the case when the machine can measure the duration of a single step. For such a (powerful) QRQW PRAM we show that OR can be computed deterministically in a constant time while PARITY of n bits (PARITY$_n$) requires $\Omega(\log n)$ time.

We prove two lower bounds on time complexity of PARITY$_n$ on QRQW PRAM: $\Omega(\sqrt{\log n / \log\log n})$ for algorithms with success probability $\frac{1}{2} + \epsilon$ ($\epsilon > 0$), and $\Omega(\sqrt{\log n})$ on average for algorithms that err with probability bounded by $2^{-\sqrt{\log n}}$.

The 2-compaction problem is known to have runtime $\Omega(\sqrt{\log n})$ on both deterministic and randomized EREW PRAMs. We show that the time complexity of this problem is the same for randomized and deterministic EREW PRAMs. The technique which we apply is quite general and may be used to obtain similar results for any problem where the number of input configurations is small.

It seems that the time complexity of 2-compaction on EREW PRAM is higher than $\sqrt{\log n}$. On the other hand, we consider an EREW$^+$ PRAM, where an arbitrary processor may interrupt the computation of the whole machine at any time. For such a machine we show that the 2-compaction problem has time complexity $\Theta(\sqrt{\log n})$. While OR can be computed in $O(1)$ time on EREW$^+$ PRAM, we show that PARITY$_n$ requires in average $\Omega(\sqrt{\log n})$ steps on randomized EREW$^+$ PRAM that errs with probability bounded by $2^{-\sqrt{\log n}}$ (and $\Omega(\log n)$ steps if computed deterministically).

Storage Controlled Pile-Up Systems

Jochen Rethmann Egon Wanke

University of Düsseldorf, Department of Computer Science, D-40225 Düsseldorf
e-mail: {rethmann, wanke}@cs.uni-duesseldorf.de

The input of a pile-up system is a sequence q of *bins* $q = (b_1, \ldots, b_n)$. Each bin b_i, $1 \leq i \leq n$, is associated with a *pallet* $plt(b_i)$ by a so-called *pallet mapping*. The bins are piled-up on pallets at *places* $P_1, \ldots, P_k$ with the help of a storage S. Each place P_k, $1 \leq k \leq p$ and the storage S can be considered as a subset of $\{b_1, \ldots, b_n\}$. A place is either *free* ($P_k = \emptyset$) or *occupied* ($P_k \neq \emptyset$). All bins placed onto some pallet P_k must be destined for the same pallet, i.e., $plt(b_{i_1}) = plt(b_{i_2})$ for all $b_{i_1}, b_{i_2} \in P_k$. The storage S can store at most s bins for arbitrary pallets.

A pile-up system can move a bin b_i from S onto some place P_k, if P_k is occupied and $plt(b_i) = plt(b_j)$ while $b_j \in P_k$. If b_i is the last bin in the sequence for $plt(b_i)$ then the pallet is removed from place P_k and the place becomes free. If there is no such place then P_k has to be free, or the next bin of the sequence q is moved into storage S. This can be done only if the capacity of the storage is not exhausted and there is another bin in the sequence.

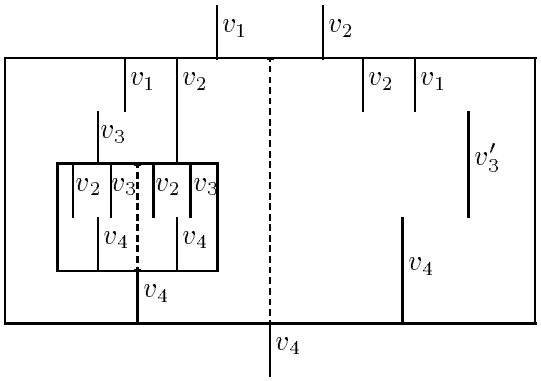
A configuration is called *blocked*, if no further action can be performed. The configuration in which the sequence, the storage and all places are empty is called *final*. A configuration is called *critical* if the storage is filled with bins, none of these bins is destined for a pallet already piled up on a place, but one place is free.

Such pile-up systems are used by supply industry. Control algorithms for pile-up systems have to compute the action to execute in a critical configuration such that no blocking configuration is obtained and the final configuration can be reached if this is possible. We have shown that such control algorithms have to solve a NP-complete problem whereas restricted pile-up systems in which the capacity of the storage or the number of pile-up places is constant can be controlled very efficiently. We also give a polynomial time algorithm for detecting blocking configurations.

Approximation Algorithms for Register Allocation

Klaus Jansen* Joachim Reiter†

High level synthesis starts with a behavioral specification of a digital system and produces a register transfer level structure that realizes the behavior. The two major optimization tasks in high level synthesis are scheduling and hardware allocation. Hardware allocation is an assignment of functional units to arithmetic and logical operations, of registers to variables, and of interconnections between registers and functional units. The goal of hardware allocation is to minimize the total amount of hardware elements.



In this talk, we study the problem of register allocation. In order to minimize the number of register, the possibility of register sharing is used. The lifetime of a variable is the time period in which the value of the variable must be saved in a register. The Figure above shows two nested conditional blocks each with two branches and the lifetimes of a set of variables. There are several ways in which registers can be shared by variables. For example, the variables v_3 and v_4 can share the same register since their lifetimes do not overlap. On the other hand, the variables v_3 and v_3' in different branches can share the same register even if their lifetimes overlap since only one of them is used during a program execution. For flow graphs with branching depth 1, the register allocation problem is already NP-complete. Several algorithms are proposed for the register allocation problem for flow graphs without conditional blocks. Only few heuristics are proposed to handle conditional register sharing (e.g. by Kurdahi, Parker (87) and by Park, Kim and Liu (93)).

We denote with $\chi(G)$ the number of colors in a minimum coloring of the conflict graph G. For a flow graph with branching depth d and at most m execution paths or branches in a conditional block, the known heuristics have worst case bound $m^d \cdot \chi(G)$. We propose an approximation algorithm with constant worst case bound for flow graphs with constant branching depth d. In the first step called equalization, the lifetimes are enlarged such that the lifetimes form an interval across different execution paths for each variable. We prove that the conflict graph $\bar{G} = (V, \bar{E})$ with enlarged lifetimes satisfies $\omega(\bar{G}) \leq (2d+1) \cdot \omega(G)$. In the second step we use an algorithm with approximation value $(d+2) \cdot \chi(\bar{G})$ for conflict graphs $\bar{G}$ with equalized variables.

*Institut für Informatik, TU München, 80 290 München, email: jansenk@informatik.tu-muenchen.de
†Fachbereich IV, Wirtschaftsinformatik, Universität Trier, 54 286 Trier, email: JR@wiinfo.uni-trier.de

Precomputation-based Load Balancing

Max Böhm, Peter Heusch, Frank Meisgen, Thomas Seifert,
Ewald Speckenmeyer

Universität zu Köln, Institut für Informatik

D–50969 Köln
e–mail: boehm@informatik.uni-koeln.de

Wir stellen einen effizienten verteilten Algorithmus für den dynamischen Lastausgleich in netzgekoppelten MIMD Systemen — den *Precomputation-based Load Balancing (PLB)* Algorithmus — vor. Der Algorithmus wird zunächst für die Netztopologie Baum definiert und anschließend auf mehrdimensionale Netztopologien erweitert. PLB kann bei beliebig teilbaren Lasten der Prozessoren einen Lastausgleich in $O(\Delta)$ parallelen Schritten durchführen, wobei Δ der Durchmesser des Netzwerkes ist. Wir analysieren die erwartete Summe der von PLB verschobenen Lastmenge für unterschiedliche Netztopologien, wobei die Lasten der Prozessoren durch unabhängige und identisch verteilte Zufallsvariablen mit der Standardabweichung σ gegeben sind. In einem linearen Array mit n Prozessoren ist die erwartete Summe der verschobenen Lasten $O(n\sqrt{n}\,\sigma)$, im vollständigen binären Baum ist sie $O(n\,\sigma)$. Wir zeigen, daß im Mittel in einem vollständigen binären Baum nur ca. 4 mal soviel Last wie in einem Cliquennetzwerk verschoben wird. Dies ist optimal, da in einem Cliquennetzwerk die beim Lastausgleich zu verschiebende Lastmenge eine untere Schranke für jede andere Netztopologie angibt. Weitere Analysen werden für Gitter- und Hypercubetopologien, sowie für die im schlimmsten Fall zu verschiebende Lastmenge durchgeführt. Experimente mit PLB bei der parallelen Lösung des Satisfiability Problems auf Transputersystemen mit bis zu 1024 Prozessoren lieferten sehr gute Effizienzen. Die mittlere Zeit eines Prozessors ohne sinnvolle Arbeit läßt sich mit PLB auf wenige Sekunden reduzieren, die nur zu Beginn und am Ende der gesamten Berechnung auftreten.

Exponential space computation of Gröbner bases

Klaus Kühnle Ernst W. Mayr
Institut für Informatik
Technische Universität München
D-80290 München
{kuehnle|mayr}@informatik.tu-muenchen.de
http://hpmayr1.informatik.tu-muenchen.de/

Let $\mathbf{Q}[x_1, \ldots, x_n]$ be the polynomial ring in the indeterminates $x_1, \ldots, x_n$ over the rationals and let some term order (i.e. a total order on the power products extending the divisibility relation and respecting multiplication) be given. We will consider an ideal in the polynomial ring $\mathbf{Q}[x_1, \ldots, x_n]$. With respect to this ideal and the given term order we then have, for any polynomial, a unique normal form, namely the smallest (w.r.t. the term order) monic polynomial in the same coset (there is a canonical extension of the term order to monic polynomials).

We consider the problem of finding this normal form for any given polynomial. As a solution, we will represent the difference of the given polynomial and its normal form as a linear combination of the ideal generators, transform this representation into a system of linear equations and solve this system. This is possible because we can bound the degrees of the normal form and of the polynomials being the coefficients in the linear combination by virtue of two already known degree bounds (Dubé 1990, Hermann 1926).

As an application of this normal form calculation we will compute the unique reduced (=irredundant) Gröbner basis of the given ideal with respect to the given term order. A Gröbner basis is a set of generators that allows normal forms to be calculated by a simple division algorithm. In such a division algorithm the given polynomial is repeatedly replaced by its remainder upon division by some generator as long as this is possible. It is a characteristic property of Gröbner bases that this division algorithm will in any case yield the normal form of the given polynomial.

The computation of the Gröbner basis runs as follows: We will enumerate all terms up to the known bound (Dubé 1990) on the degrees of the polynomials in the Gröbner basis and calculate their normal forms. If a term is not irreducible (i.e. is not equal to its normal form) but all its divisors are, then we will add the difference of this term and its normal form to the Gröbner basis.

The Complexity of the Equivalence Problem for Commutative Semigroups

Ulla Koppenhagen Ernst W. Mayr
Institut für Informatik
Technische Universität München
D-80290 München, GERMANY
e-mail: {KOPPENHA|MAYR}@INFORMATIK.TU-MUENCHEN.DE
WWW: HTTP://WWWMAYR.INFORMATIK.TU-MUENCHEN.DE/

Commutative semi-Thue systems (or equivalently, Petri nets or vector addition systems) are well-known models for parallel processes. For an important subclass of commutative semi-Thue systems, the class of commutative Thue-systems (or equivalently, reversible Petri nets or reversible vector addition systems), we will focus on the equivalence problem.

The equivalence problem for commutative semigroups is the problem of deciding for any two given congruence classes $[u]_{\mathcal{P}}$, $[v]_{\mathcal{Q}}$, where $\mathcal{P}$, $\mathcal{Q}$ are two commutative semigroup presentations over an alphabet X, and u, v are two words in X^*, whether $[u]_{\mathcal{P}}$ is equal to $[v]_{\mathcal{Q}}$.

As main result we derive that a closed representation of a congruence class $[u]_{\mathcal{P}}$ as a uniformly semilinear set can be generated in exponential space. The basic concepts which we use are the close relationship between commutative semigroups and binomial ideals, and an exponential space algorithm for constructing the reduced Gröbner basis of a binomial ideal. First we show that the minimal elements of $[u]_{\mathcal{P}}$ can be determined requiring at most space $2^{c\cdot\mathrm{size}(u,\mathcal{P})}$ for some constant $c > 0$ independent of u and $\mathcal{P}$. We project $[u]_{\mathcal{P}}$ onto its bounded coordinates and then use an algorithm for the subword reachability problem in commutative semigroups. The bounded coordinates can be found by an exponential space algorithm for the coverability problem, and we will see that for the subword reachability problem there is also an exponential space algorithm. Then, using again the subword reachability algorithm, we show an analogous bound for the minimal periods of $[u]_{\mathcal{P}}$.

Thus, for the equivalence problem for commutative semigroups, the gap between the

$$2^{d\cdot\mathrm{size}(u,v,\mathcal{P},\mathcal{Q})\cdot\log(\mathrm{size}(u,v,\mathcal{P},\mathcal{Q}))}$$

space upper bound and the exponential space lower bound resulting from the exponential space completeness of the uniform word problem is closed, and the exponential space completeness established.

Worst case–Analyse und untere Schranken für bessere SAT-Algorithmen

Oliver Kullmann
Johann Wolfgang Goethe-Universität, Fachbereich Mathematik
60054 Frankfurt, Germany
e-mail: kullmann@mi.informatik.uni-frankfurt.de

25. Januar, 1996

In meinem Vortrag möchte ich, ausgehend von den Ergebnissen und Methoden in [KuLu95], [Ku95a] und [Ku95b], diskutieren (und auch spekulieren), wie durch das Zusammenspiel von worst case-Analyse, algorithmischen Betrachtungen und Reflektion der Methoden mittels unterer Schranken neue Methoden zur SAT-Entscheidung entstehen können.

Die worst case-Analyse ermöglicht hierbei, als eine Art Kompaß, durch ihre numerische Bewertung der algorithmisch-kombinatorischen Optionen eine Richtung zu finden und auszubauen, in der auch beweisbar ein Fortschritt liegt. Beispiele für neu entwickelte algorithmische Methoden, motiviert durch eine verfeinerte Analyse, sind der Einbezug der Entstehung neuer 2-Klauseln für die 3-SAT-Entscheidung, sowie hierbei die Methode der "Verallgemeinerte Autarkie" und das Konzept der "Blockierten Klauseln" ([Ku95a]) zur Erzeugung neuer 2-Klauseln.

Untere Schranken für Algorithmenklassen aufzuzeigen durch Simulation von Algorithmenläufen auf unerfüllbaren Eingaben mittels Beweissystemen, für die eine untere Schranke bekannt ist, lenkt auf der anderen Seite den Blick auf die prinzipielle Beschränktheit der verwendeten Methoden, und vermag, durch Vergleich mit stärkeren Beweissystemen, Hinweise auf neue Methoden zu geben, die einen qualitativen Sprung mit sich bringen könnten.

Literatur

[Ku95a] Kullmann, O.: A systematical approach to 3-SAT-decision, yielding 3-SAT-decision in less than 1.5045^n steps.

[Ku95b] Kullmann, O.: A note on a generalization of Extended Resolution. In Vorbereitung.

[KuLu95] Kullmann, O. and Luckhardt, H.: Various upper bounds on the complexity of algorithms for deciding propositional tautologies.

MORE: Alternative Implementation of BDD-Packages by Multi-Operand Synthesis[1)]

Andreas Hett Rolf Drechsler Bernd Becker

FB Informatik, Albert-Ludwigs-Universität, D-79110 Freiburg i.Br.
email: <name>@informatik.uni-freiburg.de

Abstract

We present a new approach for the realization of a BDD-package. This approach does not depend on recursive synthesis operations (i.e. the ternary *If-Then-Else*-operator (ITE)) to perform manipulations of Boolean functions; instead our basic operation *MORE* is based on exchanges of neighbouring variables and existential quantification. It is capable of combining an arbitrary number of Boolean functions in *parallel*.

We present experimental results to show that a *Computed Table* (CT), mostly a hashed-based cache that maps the three operands F, G and H of an ITE-function call to the result node *ite*(F,G,H) once this result has been computed, is essential to gain a good performance of ITE.

MORE does not need a CT to gain good performance and thus spares the resources (usually some megabytes) needed to maintain a CT.

One major drawback of an ITE-operation is that it is not possible to interrupt it in order to change the variable ordering whithout the loss of all subgraphs that were calculated during its recursive process to gain the final result. Thus, to handle an impending size limit crossing ITE must protocol every step taken to perform a synthesis operation in order to make a reversion of them possible. After this reversion a minimization of the BDDs (e.g. by means of dynamic variable reordering algorithms) can be done and the ITE-operation can be retried in hope that is will succeed this time.

Since *MORE* is based on *Levelexchanges*, an impending boundary crossing can be handled simply by interrupting the *Levelexchanges* and rearranging the variables to minimize the BDDs. Then *MORE* will continue right at the point where it was suspended. Thus, *MORE* works with higher granularity allowing to construct BDDs with a better controllability of resources.

We discuss these and other differences between *MORE* and ITE and give experimental results to show the advantages of our implementation approach.

[1)] This work was supported in part by DFG grant Be 1176/4-2.

Ein neuer Reduktionsbegriff für OBDDs

Christoph Meinel, Anna Slobodová[†‡]
FB IV – Informatik
Universität Trier
D-54286 Trier

Reduktionsmechanismen gehören zu den wichtigsten Werkzeugen der Berechenbarkeits- und Komplexitätstheorie. Mit ihrer Hilfe kann die Komplexität von Problemen relativ zur Komplexität von anderen Problemen charakterisiert und untere bzw. obere Schranken von einem Problem auf ein anderes übertragen werden. Üblicherweise heißt ein Problem A reduzierbar auf ein Problem B, falls A mit Hilfe eines Programmes für B berechnet werden kann. Die Reduktion selbst beschreibt dabei das Rahmenprogramm für A, das das Programm für B als Unterprogramm benutzt.

Leider ist dieses Schema für die Betrachtung sehr eingeschränkter Berechnungsmodelle in den meisten Fällen zu eng, da es aufgrund der in diesen Fällen dem Rahmenprogramm nur sehr beschränkt zur Verfügung stehenden Berechnungsressourcen nur in sehr geringem Maße Berechnungsmechanismen zur Verfügung stellen kann, die über die Möglichkeiten des Unterprogramms hinausgehen. Demzufolge bleiben die erzielbaren Komplexitätsaussagen häufig ohne weitergehendes Interesse. Dies ist um so bedauerlicher, als wirkungsvolle Reduktionstechniken gerade für eingeschränkte Modelle sowohl im theoretischen Bereich als auch bei praktischen Anwendungen von großer Wichtigkeit wären, da diese Modelle sehr oft die einzigen Modelle mit bewiesenen unteren Schranken sind bzw. in Anwendungen tatsächlich vorkommen.

In unserer Arbeit haben wir diese Problematik für das Berechnungsmodell der in den letzten 10 Jahren im CAD-Bereich am häufigsten benutzten Datenstruktur der OBDDs (ordered binary decision diagrams) untersucht. Basierend auf den aus der Schaltkreiskomplexitätstheorie bekannten Projektionsreduktionen führten Bollig und Wegener (STACS'96) dem oben skizzierten Schema folgend den Reduktionsbegriff der read-once Projektionen für OBDDs ein. Die dabei aufgetretenen Effekte, wie z.B. die Nichtreduzierbarkeit von 0 auf 1, oder die von MODULO-4 auf MODULO-2 trotz hochgradiger Ähnlichkeit der OBDD-Darstellungen der betrachteten Funktionen belegen die angesprochenen Schwierigkeiten. Zur Vermeidung dieser Schwierigkeiten schlagen wir einen neuen Reduktionsmechanismus für OBDDs vor. Wir nennen ein Problem Π auf ein Problem Σ reduzierbar, falls man mit Hilfe einer Folge von "elementaren" Operationen aus einem OBDD für Σ ein OBDD für Π erhalten kann, wobei "elementar" auf die Berechenbarkeit in konstanter Zeit hinweist. Das Ziel der Arbeit liegt darin, eine algorithmische Fundierung dieses dynamischen Reduktionsverfahrens zu entwickeln und die Möglichkeiten des neuen Reduktionsbegriffs anhand verschiedener Beispielen zu untersuchen.

[†]Unterstüzt von DFG, Projekt Me 1077/2-2
[‡]Wir sind dankbar für die Unterstützung von DAAD ACCIONES INTEGRADAS, Projekt-Nr. 322-ai-e-dr

Zustandscodierungen und OBDD-Größen

Thorsten Theobald, Christoph Meinel
FB IV - Informatik, Universität Trier
D-54286 Trier, Germany
{theobald, meinel} @ ti.uni-trier.de

Geordnete binäre Entscheidungsgraphen (OBDDs) haben sich in den letzten Jahren als sehr effiziente Datenstruktur zur Repräsentation Boolescher Funktionen erwiesen. Ein Hauptanwendungsgebiet von OBDD-basierten Algorithmen ist die Verifikation digitaler Systeme. Hierbei werden die zu verifizierenden Eigenschaften auf Boolesche Gleichungen zurückgeführt. Die Komplexität der Überprüfung der Gleichungen hängt maßgeblich von der Effizienz der Datenstruktur ab.

Von besonderer Bedeutung für die Verifikation sind zustandsendliche Systeme. Die Verbindung zwischen der OBDD-Datenstruktur und zustandsendlichen Systemen wird durch die sogenannte Transitionsrelation hergestellt, deren charakteristische Funktion eine Boolesche Funktion ist. Im Gegensatz zur kombinatorischen Schaltkreisverifikation hängt die Größe der OBDDs nicht nur von der Variablenordnung sondern auch von der Zustandscodierung ab.

In dem Vortrag werden Zähler als wichtige Teilklasse zustandsendlicher Systeme unter diesen Gesichtspunkten analysiert. Für die Klasse wird eine Verbindung zwischen Zustandscodierung und der resultierenden OBDD-Größe hergestellt. Es werden OBDD-Größen für wichtige Codierungen hergeleitet, untere Schranken bewiesen und worst-case Codierungen konstruiert, die auf exponentiell große OBDDs führen.