High Failure Probability for Cuckoo Hashing with too Simple Hash Functions

Ulf Schellbach

5th February 2008

Cuckoo hashing (Pagh/Rodler 2001) is a dictionary implementation that uses two hash tables T_1, T_2 of the same size m, and corresponding hash functions h_1, h_2. A key $x \in U$, for some set U (universe), is either stored in cell $h_1(x)$ of T_1, or in cell $h_2(x)$ of T_2. Thus, a *lookup* of x costs no more than two memory accesses.

Let $S \subset U$ denote any set of n keys. Pagh and Rodler showed that, if $m > (1 + \varepsilon)n$ for an arbitrary constant $\varepsilon > 0$, and h_1, h_2 map keys fully random or at least $c \log n$-wise independently, then S can be stored with high probability, and the amortized expected insertion time is $O(1)$. However, in experiments it was observed that a simple hash class containing the functions

$$x \mapsto ax \bmod 2^k \operatorname{div} 2^{k-l} \quad (1 \leq a \leq 2^k \text{ odd})$$

does not seem to work together well with cuckoo hashing. We prove that, for arbitrary functions h_1, h_2 of this class, and a randomly chosen set S of size $m/2$, the failure probability tends to 1 as k and l grow, if they satisfy the conditions $l \leq k - 2$, and $l/k > 10/11$. That is, given large enough parameters k and l satisfying the conditions, it is very unlikely that, even if the table size exceeds n by a factor of 2, a random set of n keys can be inserted successfully. A similar result is true for the popular hash class consisting of the *linear functions*

$$x \mapsto ((ax + b) \bmod p) \bmod m.$$

This should be taken as a warning that, when using cuckoo hashing, the degree of independence of hash functions must not be too low. In fact, experiments seem to suggest the hypothesis that a little more independence of hash values, as for instance given by quadratic polynomials, yields a very good performance.

(Joint work with Martin Dietzfelbinger.)

Über Variablen-Gewichtete X3SAT Optimierungsprobleme

Galina Plagge und Stefan Porschen

Institut für Informatik
Universität zu Köln
{plagge,porschen}@informatik.uni-koeln.de

Zusammenfassung

Das XSAT Problem für KNF Formeln ist ein $\mathcal{NP}$-vollständiges Problem, bei dem man eine exakt erfüllende Belegung für die Formel sucht, d.h. in jeder Klausel muß genau ein Literal auf 1 gesetzt werden. Das X3SAT Problem ist eine Variante des XSAT Problems für 3-KNF Formeln. Dieses Problem ist trotz Einschränkungen immer noch $\mathcal{NP}$-vollständig [3]. In dieser Arbeit betrachten wir zwei $\mathcal{NP}$-schwere Optimierungsvarianten des X3SAT Problems, bei denen die Variablen mit reellen Gewichten versehen werden: minimal gewichtete exakte Erfüllbarkeit für 3-KNF Formeln (MinW-X3SAT) und maximal gewichtete exakte Erfüllbarkeit für 3-KNF Formeln (MaxW-X3SAT). Wir stellen einen exakten Algorithmus vor, der MinW-X3SAT (bzw. MaxW-X3SAT) in Zeit $O(2^{0.1625n})$ löst, dabei bezeichnet n die Anzahl der Variablen in der Formel.

Die beste zuvor bekannte Möglichkeit diese Probleme zu lösen war ein Algorithmus von S. Porschen zur Lösung des MinW-XSAT Problems mit der Laufzeit $O(2^{0.2441n})$ [2]. Um diese Laufzeit zu verbessern, haben wir die Einschränkung in der Klausellänge ausgenutzt. Unserem Wissen nach, ist es der erste Algorithmus, der explizit MinW-X3SAT und MaxW-X3SAT löst. Dementsprechend ist es auch der schnellste exakte Algorithmus zur Lösung solcher Probleme.

Die zur Zeit beste bekannte Laufzeit für das ungewichtete X3SAT Problem von $O(2^{0.1379n})$ liefert ein Algorithmus von J.M. Byskov, B.A. Madsen und B. Skjernaa [1].

Literatur

[1] J.M. Byskov, B.A. Madsen, and B. Skjernaa. New Algorithms for Exact Satisfiability. *Theoretical Computer Science*, 332(1–3):515–541, 2005.

[2] S. Porschen. Solving Minimum Weight Exact Satisfiability in Time $O(2^{0.2441n})$. In *Proceedings of ISAAC 2005*, volume 3827 of *LNCS*, pages 654–664. Springer-Verlag, 2005.

[3] T.J. Schaefer. The Complexity of SAT Problems. *Proceedings of the 10th ACM Symposium on Theory of Computing*, pages 216–226, 1978.

Some fixed-parameter tractable classes of hypergraph duality and related problems

Khaled Elbassioni, Matthias Hagen, and Imran Rauf

Max-Planck-Institut für Informatik, D–66123 Saarbrücken,
{elbassio,irauf}@mpi-inf.mpg.de

Friedrich-Schiller-Universität Jena, Institut für Informatik, D–07737 Jena,
hagen@cs.uni-jena.de

We present fixed-parameter algorithms for the problem Dual—given two hypergraphs, decide if one is the transversal hypergraph of the other—and related problems.

Briefly, a parameterized problem with parameter k is *fixed-parameter tractable* if it can be solved by an algorithm running in time $O(f(k) \cdot poly(n))$, where f is a function depending on k only, n is the size of the input, and $poly(n)$ is any polynomial in n. The class FPT contains all fixed-parameter tractable problems.

In the first part, we consider the number of edges of the hypergraphs, the maximum degree of a vertex, and vertex complementary degrees as our parameters.

In the second part, we use an Apriori approach to obtain FPT results for generating all maximal independent sets of a hypergraph, all minimal transversals of a hypergraph, and all maximal frequent sets where parameters bound the intersections or unions of edges.

Tight bounds for blind search on the integers

Martin Dietzfelbinger, *Technische Universität Ilmenau*

(Joint work with Jonathan Rowe, Ingo Wegener, and Philipp Woelfel)

Summary

Consider the following game, played on a segment of the integers: Given is a probability distribution μ on $\{1, \ldots, n\}$. Start at a point R_0 chosen uniformly at random in $\{0, \ldots, n\}$, and then repeat the following steps, for $t = 1, 2, 3, \ldots$:

1. Choose (a distance) D_t from $\{1, \ldots, n\}$ randomly, according to μ;
2. If $D_t \leq R_{t-1}$, let $R_t = R_{t-1} - D_t$ (use the step), otherwise let $R_t = R_{t-1}$ (can't use the step).

Obviously, the process $R_0, R_1, R_2, \ldots$ has 0 as an absorbing state. Let

$$T = \min\{t \mid R_t = 0\}.$$

What can we say about the expected time $\mathbf{E}(T) = \mathbf{E}_\mu(T)$ until the process hits 0?

Theorem

(a) There is some μ such that $\mathbf{E}_\mu(T) = O((\log n)^2)$.
(b) For all μ we have $\mathbf{E}_\mu(T) = \Omega((\log n)^2)$.

Features of *proof*: The upper bound is easy. The lower bound has two interesting features: delayed decisions and a potential function argument.

Motivation: The game considered here is a toy version of a simple randomized search heuristic for a black-box optimization problem, where one tries to find the minimum of a function $f\colon \{0, \ldots, n\} \to \mathbb{N}$, by jumping blindly around in $\{0, \ldots, n\}$ by randomly chosen distances, accepting a step if it improves (decreases) the function value. The game is closely connected to the behaviour of the strategy in case f is unimodal (first falling, then increasing).

(STACS 2008, `http://drops.dagstuhl.de/opus/volltexte/2008/1348/`)

Smoothed Analysis of Trie Height

Stefan Eckhardt *Sven Kosub* *Johannes Nowak*

Fakultät für Informatik
Technische Universität München
{eckhardt, nowakj}@in.tum.de

Fachbereich Informatik & Informationswissenschaft
Universität Konstanz
kosub@inf.uni-konstanz.de

Tries are general purpose data structures for information retrieval. The most significant parameter of a trie is its height H which equals the length of the longest common prefix of any two string in the set A over which the trie is built. Analytical investigations of random tries suggest that $\mathbf{E}[H]$ is in $O(\log(\|A\|))$, although H is unbounded in the worst case. Moreover, sharp results on the distribution function of H are known for many different random string sources. But because of the inherent weakness of the modeling behind average-case analysis—analyses being dominated by random data—these results can utterly explain the fact that in many practical situations the trie height is logarithmic.

We propose a new semi-random string model and perform a smoothed analysis in order to give a mathematically more rigorous explanation for the practical findings. The perturbation functions which we consider are based on probabilistic finite automata (PFA) and we show that the transition probabilities of the representing PFA completely characterize the asymptotic growth of the smoothed trie height. Our main result is of dichotomous nature—logarithmic or unbounded—and is certainly not surprising at first glance, but we also give quantitative upper and lower bounds, which are derived using multivariate generating function in order to express the computations of the perturbing PFA. A direct consequence is the logarithmic trie height for edit perturbations (i.e., random insertions, deletions and substitutions).

Das Deduktionstheorem für starke aussagenlogische Beweissysteme

Olaf Beyersdorff

Theoretische Informatik, Universität Hannover
Appelstr. 4, 30167 Hannover, Germany
beyersdorff@thi.uni-hannover.de

In this talk we focuse on the deduction theorem for propositional logic. We define and investigate different deduction properties and show that the presence of these deduction properties for strong proof systems is powerful enough to characterize the existence of optimal and even polynomially bounded proof systems. We also exhibit a similar, but apparently weaker condition that implies the existence of complete disjoint NP-pairs. In particular, this yields a sufficient condition for the completeness of the canonical pair of Frege systems and provides a general framework for the search for complete NP-pairs.

The Uniformity Duality Property

Michael Thomas

Theoretische Informatik, Universität Hannover
Appelstr. 4, 30167 Hannover, Germany
thomas@thi.uni-hannover.de

The usual intensional uniformity conditions for Boolean circuits require that a resource-bounded machine be able to exhibit the circuits in the circuit family defining $\mathcal{C}$. In this talk, we focus on uniformity imposed by intersecting formal logics (or circuit complexity) with formal languages. We say that $(\mathcal{C}, \mathcal{L})$ has the *Uniformity Duality Property* if the extensionally uniform class $\mathcal{C} \cap \mathcal{L}$ can be captured intensionally by means of adding so-called $\mathcal{L}$*-numerical predicates* to the first-order descriptive complexity apparatus describing the connection language of the circuit family defining $\mathcal{C}$. We exhibits positive and negative instances of the Uniformity Duality Property.