BRICS

Basic Research in Computer Science

BRICS RS-02-15 J. M. Nielsen: On the Number of Maximal Independent Sets in a Graph

On the Number of Maximal Independent Sets in a Graph

Jesper Makholm Nielsen

BRICS Report Series RS-02-15

ISSN 0909-0878 April 2002

See back inner page for a list of recent BRICS Report Series publications.
Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark

Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

http://www.brics.dk
ftp://ftp.brics.dk
This document in subdirectory RS/02/15/

On the Number of Maximal Independent Sets in a Graph

Jesper Makholm Nielsen
BRICS*
Department of Computer Science
University of Aarhus
jespermn@brics.dk

April, 2002

Abstract

We show that the number of maximal independent sets of size *exactly* k in any graph of size n is at most $\lfloor n/k \rfloor^{k-(n \bmod k)}(\lfloor n/k \rfloor+1)^{n \bmod k}$. For maximal independent sets of size *at most* k the same bound holds for $k \leq n/3$. For $k > n/3$ a bound of approximately $3^{n/3}$ is given. All the bounds are exactly tight and improve Eppstein (2001) who give the bound $3^{4k-n}4^{n-3k}$ on the number of maximal independent sets of size at most k, which is the same for $n/4 \leq k \leq n/3$, but larger otherwise. We give an algorithm listing the maximal independent sets in a graph in time proportional to these bounds (ignoring a polynomial factor), and we use this algorithm to construct algorithms for 4- and 5- colouring running in time $\mathcal{O}(1.7504^n)$ and $\mathcal{O}(2.1593^n)$, respectively.

1 Introduction

In their 1965 paper Moon and Moser [6] show that the number of maximal independent sets in any graph is at most $3^{n/3}$ where n is the number of vertices in the graph. They use a carefully designed transformation on the extremal graphs to fully characterise these. Croitoru [1] uses

*Basic Research in Computer Science (www.brics.dk),
funded by the Danish National Research Foundation.

the same technique to show that all graphs with independence number (the size of the largest independent set) no larger than k have at most $\lfloor n/k \rfloor^{k-(n \bmod k)}(\lfloor n/k \rfloor + 1)^{n \bmod k}$ maximal independent sets.

Eppstein [3] uses an algorithmic proof to show that the number of maximal independent sets of size *at most* k is no larger than $3^{4k-n}4^{n-3k}$. His bound is both tight and equal to that of Croitoru for $n/4 \leq k \leq n/3$. Eppstein advocates the search for a tight bound, for all values of k.

In this paper we give such a tight bound for all values of k. This bound equals the bound of Croitoru without the restriction on the independence number. In the proof we use the same technique as Croitoru and Moon and Moser do, but we furthermore give an algorithmic proof similar to that of Eppstein, to show that the maximal independent sets can be listed with only polynomial overhead.

Eppstein uses his bound to improve the time bound for colouring graphs. Although we do not improve this time bound, our result is used to give better time bounds for 4- and 5-colouring than what was previously known.

2 Preliminaries

The graphs in this paper are all simple, undirected graphs $G = (V, E)$. By "subgraphs" we mean vertex-induced subgraphs, and the subgraph induced by a subset of the vertices $S \subseteq V$ is denoted $G[S]$. An edge (v, w) is *adjacent* to a vertex v', if v' equals v or w. The *neighbourhood* of a vertex is $\mathcal{N}(v) = \{w \in V \mid (v, w) \in E\}$, and the neighbourhood of a set of vertices is the union of their neighbourhoods. Two vertices v and w are *neighbours*, if $(v, w) \in E$, or equivalently if $w \in \mathcal{N}(v)$. For neighbours v and w we let $G_{v \to w}$ denote the graph obtained from G by removing all edges adjacent to v, except (v, w), and adding edges from v to all neighbours of w except for v itself. Note that $G_{v \to w}$ is equal to the graph obtained by removing v from G, making a copy of w with the same neighbours as w and adding an edge between them.

A *maximal independent set* in a graph is an independent set contained in no other independent set in the graph. Let $I^{=k}(G)$ $\left(I^{\leq k}(G)\right)$ denote the number of maximal independent sets of size exactly (at most) k in G. Similarly let $I_G^{=k}(x)$ $\left(I_G^{\leq k}(x)\right)$ denote the number of maximal independent sets of size exactly (at most) k in G that contains x. Also let $\mathcal{I}^{=k}(n)$ $\left(\mathcal{I}^{\leq k}(n)\right)$ denote the maximum number of maximal independent sets of size exactly (at most) k in any n-vertex graph.

In the rest of the paper, when we talk about the time complexity of an algorithm we will ignore polynomial factors, as all time complexities are exponential. We will use the notation $\mathcal{O}^*\left(c^n\right)$ to denote a function within a polynomial factor of c^n.

3 Results

Theorem 1. *Let $1 \leq k \leq n$. If $k \leq n/3$, then*

$$\mathcal{I}^{\leq k}\left(n\right) = \lfloor n/k \rfloor^{k-(n \bmod k)} \left(\lfloor n/k \rfloor + 1\right)^{n \bmod k}.$$

If $k > n/3$, then

$$\mathcal{I}^{\leq k}\left(n\right) = \begin{cases} 3^{n/3} & \text{if } n \equiv 0 \pmod 3, \\ 4 \cdot 3^{\lfloor n/3 \rfloor - 1} & \text{if } n \equiv 1 \pmod 3, \\ 2 \cdot 3^{\lfloor n/3 \rfloor} & \text{if } n \equiv 2 \pmod 3. \end{cases}$$

Moreover, there is an algorithm that takes a graph as input and outputs all maximal independent sets of size at most k in time $\mathcal{O}^*\left(\mathcal{I}^{\leq k}\left(n\right)\right)$.

Theorem 2. *Let $1 \leq k \leq n$. Then*

$$\mathcal{I}^{=k}\left(n\right) = \lfloor n/k \rfloor^{k-(n \bmod k)} \left(\lfloor n/k \rfloor + 1\right)^{n \bmod k}.$$

Moreover, there is an algorithm that takes a graph as input and outputs all maximal independent sets of size exactly k in time $\mathcal{O}^*\left(\mathcal{I}^{=k}\left(n\right)\right)$.

Remark. The time needed to output all maximal independent sets is bounded by the maximum number of these in any n-vertex graphs, not the actual number in the graph given as input.

The proof of the bounds in the two theorems is essentially the same as that used by Moon and Moser [6].

Proof of Theorem 1 and 2. Let v and w be neighbours in a graph G, and look at the number of maximal independent sets of size *at most k* in $G_{v \to w}$. None of these contain both v and w, since they are neighbours. All maximal independent sets of size at most k in G that neither contains v nor w are still maximal independent sets in $G_{v \to w}$, since if v or w could be added in $G_{v \to w}$ so could w in G. Those maximal independent sets containing w also remains maximal in $G_{v \to w}$, and in addition we get a new maximal independent set in $G_{v\to w}$, with v replacing w. The maximal independent sets containing v are no longer maximal in $G_{v\to w}$, except for those that are maximal independent sets in $G[V \setminus \{v\}]$ when v is removed from them. Let $M(v)$ denote the number of these. Thus $I^{\leq k}(G_{v\to w}) = I^{\leq k}(G) + I^{\leq k}_{G}(w) - I^{\leq k}_{G}(v) + M(v)$, and by the same argument $I^{\leq k}(G_{w\to v}) = I^{\leq k}(G) + I^{\leq k}_{G}(v) - I^{\leq k}_{G}(w) + M(w)$. Suppose G has the maximum number of maximal independent sets of size at most k among all graphs with n vertices. If v and w are neighbours in G, neither $G_{v\to w}$ nor $G_{w\to v}$ can have more maximal independent sets than G, but then $I^{\leq k}_{G}(v) = I^{\leq k}_{G}(w)$ and $M(v) = M(w) = 0$ by the above equations. This implies that we can replace G by $G_{v\to w}$ for all neighbours v and w in G without changing $I^{\leq k}(G)$. Let $w \in V$ and $\mathcal{N}(w) = \{v_1, v_2, \ldots, v_k\}$. By successively replacing G by $G_{v_i\to w}$ we end up with $\{w, v_1, v_2, \ldots, v_k\}$ being a connected component of the graph forming a clique. Doing this for all connected components we end up with a graph consisting of disconnected cliques. Let the number of vertices in these cliques be $i_1, i_2, \ldots, i_l$, where $i_1 + i_2 + \cdots + i_l = n$. Then the number of maximal independent sets of size at most k in G is

$$I^{\leq k}(G) = \begin{cases} i_1 \cdot i_2 \cdot \cdots \cdot i_l & \text{if } l \leq k, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

We want to find those values of i_j maximising this expression.[1] It is clear that none of the i_j's can differ by more than one, since if $a > b+1$, $(a-1)(b+1) = ab + (a-b-1) > ab$. Also since $a \leq \lfloor a/2 \rfloor \cdot \lceil a/2 \rceil$ for $a \geq 4$, if $l < k$ any $i_j \geq 4$ can be split in two, namely $\lfloor i_j/2 \rfloor \cdot \lceil i_j/2 \rceil$ without lowering the value of the expression. Thus for $k \leq n/3$ this is largest, if $(k - (n \bmod k))$ of the i_j's are $\lfloor n/k \rfloor$ and $(n \bmod k)$ of them are $\lfloor n/k \rfloor + 1$, giving $I^{\leq k}(G) = \lfloor n/k \rfloor^{k-(n \bmod k)}(\lfloor n/k \rfloor + 1)^{n \bmod k}$. For $k > n/3$, since $2^3 < 3^2$ three two's are replaced by two three's. Thus the maximum is $3^{n/3}$, if $n \equiv 0 \pmod 3$, $4 \cdot 3^{\lfloor n/3 \rfloor - 1}$, if $n \equiv 1 \pmod 3$ and $2 \cdot 3^{\lfloor n/3 \rfloor}$, if $n \equiv 2 \pmod 3$. As we started out from a graph with the maximum number of maximal independent sets this is an upper bound on $\mathcal{I}^{\leq k}(n)$, and since we explicitly found the extremal graphs this is also a lower bound obtained for the graphs consisting of a union of $(k - (n \bmod k))$ $K_{\lfloor n/k \rfloor}$'s and $(n \bmod k)$ $K_{\lfloor n/k \rfloor + 1}$'s, for $k \leq n/3$ and an appropriate number of K_2 and K_3's, for $k > n/3$.[2] For $I^{=k}(G)$ a similar

[1] Moon and Moser [6] gave the above proof with $I^{\leq k}$ replaced by the total number of maximal independent sets, and thus got the upper bound $3^{n/3}$.

[2] These graphs are the complement of the Turán graphs.

expression to (1), except that there must be exactly k i_j's, is found, and therefore the bound $I^{=k}(G) = \lfloor n/k \rfloor^{k-(n \bmod k)} (\lfloor n/k \rfloor + 1)^{n \bmod k}$ is valid for all $k \geq 1$. $\square$

Remark. The extremal graphs are unique, as already Moon and Moser [6] noted using the following proof. Look at the second-to-last graph G in the above procedure s.t. $G_{v \to w}$ is the last. Then v and w must be neighbours, and G consists of v and a bunch of disconnected cliques. Now v is connected to all vertices in w's component; otherwise, the maximal independent set consisting of v, a vertex from w's component, except for w itself and a maximal independent set of the remaining graph is still a maximal independent set in $G[V \setminus \{v\}]$ with v removed, contradicting $M(v) = 0$. Since $I_G^{\leq k}(v) = I_G^{\leq k}(w)$, v is connected to no other vertices. But then $G = G_{v \to w}$, and since the process started out with any extremal graph, these are unique.

Proof (continued). We now devise an algorithm for finding all maximal independent sets of size *exactly* k. Suppose a graph $G = (V, E)$ has a maximal independent set I of size k. By maximality $I \cup \mathcal{N}(I) = V$; thus, I must have a vertex say v of degree $\geq \lceil n/k \rceil - 1$. All maximal independent sets in G either consists of v and a maximal independent set in $G\left[V \setminus \left(\{v\} \cup \mathcal{N}(v)\right)\right]$, or a maximal independent set in $G[V \setminus \{v\}]$ that intersects $\mathcal{N}(v)$. Our algorithm chooses the vertex with the highest degree and branches on the two cases of either including it in the maximal independent set, and removing it from the graph together with all its neighbours, or excluding it from the maximal independent set, and removing it from the graph. If $k = n$ the algorithm just tests whether the remaining vertices form an independent set, and if $k = 0$, or $k > n$ there are no appropriate maximal independent sets, and the branching stops. Let $T(n, k)$ denote the maximum number of leaves in the recursion tree of the algorithm when finding all maximal independent sets of size exactly k in a graph with n vertices. We want to show that $T(n, k) \leq \lfloor n/k \rfloor^{k-(n \bmod k)} (\lfloor n/k \rfloor + 1)^{n \bmod k}$. For $k = 1$ and $k = n$ this is trivially so. We proceed by induction in k. Let $1 < k < n$ and suppose it holds for all smaller values of k. Then by the above discussion

$$T(n, k) \leq T(n - \lceil n/k \rceil, k - 1) + T(n - 1, k)$$

which by induction hypothesis is at most

$$\left\lfloor \frac{n-\lceil n/k\rceil}{k-1} \right\rfloor^{k-1-((n-\lceil n/k\rceil) \bmod (k-1))} \left(\left\lfloor \frac{n-\lceil n/k\rceil}{k-1} \right\rfloor + 1\right)^{(n-\lceil n/k\rceil) \bmod (k-1)}$$
$$+ \left\lfloor \frac{n-1}{k} \right\rfloor^{k-((n-1) \bmod k)} \left(\left\lfloor \frac{n-1}{k} \right\rfloor + 1\right)^{(n-1) \bmod k} .$$

Using the fact that

$$n - \lceil n/k\rceil = \begin{cases} (k-1)\lfloor n/k\rfloor & \text{if } k \mid n, \\ (k-1)\lfloor n/k\rfloor + (n \bmod k) - 1 & \text{if } k \nmid n, \end{cases}$$

we get that, if $k \mid n$, $T(n,k)$ is at most

$$\left\lfloor \frac{n}{k} \right\rfloor^{k-1} \left(\left\lfloor \frac{n}{k} \right\rfloor + 1\right)^{0} + \left(\left\lfloor \frac{n}{k} \right\rfloor - 1\right)^{1} \left\lfloor \frac{n}{k} \right\rfloor^{k-1}$$

which is equal to $\mathcal{I}^{=k}(n)$, and if $k \nmid n$, $T(n,k)$ is at most

$$\left\lfloor \frac{n}{k} \right\rfloor^{k-(n \bmod k)} \left(\left\lfloor \frac{n}{k} \right\rfloor + 1\right)^{(n \bmod k)-1}$$
$$+ \left\lfloor \frac{n}{k} \right\rfloor^{k-(n \bmod k)+1} \left(\left\lfloor \frac{n}{k} \right\rfloor + 1\right)^{(n \bmod k)-1}$$

which again equals $\mathcal{I}^{=k}(n)$. Since the algorithm at every step takes at most polynomial time the total running time of the algorithm is $\mathcal{O}^*\left(\mathcal{I}^{=k}(n)\right)$.

Next we devise an algorithm for finding all maximal independent sets of size *at most* k rather than exactly k. If the maximum degree in the graph is at most two the graph consists of only paths and cycles. For each of these we can calculate all possible sizes of maximal independent sets. The algorithm branches as before on the cases of including or excluding a vertex, but keeps track of the possible sizes of maximal independent sets in the remaining graph, such that it only makes recursive calls that will find maximal independent sets. Then every leaf in the recursion tree corresponds to a maximal independent set of size at most k in the graph, and the running time is thus proportional to $\mathcal{I}^{\leq k}(n)$.

If the maximum degree is larger than two an algorithm similar to the one for maximal independent sets of size exactly k is used, the only difference being that it no longer stops if $k \geq n$, since we are looking for maximal independent sets of size *at most* k. We now prove by induction in k that the number of leaves in the recursion tree $T(n,k)$ is at most $\mathcal{I}^{\leq k}(n)$. For $n=1$ and $k=1$ this is trivially so. We want to show it for $1<k\leq n$, so suppose it is true for all smaller values of k. If $k<n/3$ then $k\leq(n-1)/3$ and the proof is similar to that for $\mathcal{I}^{=k}(n)$. If $k=n/3$ then

$$T(n,k)\leq 3^{(n-3)/3}+2\cdot 3^{\lfloor(n-1)/3\rfloor}=3^{n/3}$$

which is equal to $\mathcal{I}^{\leq k}(n)$. For $k>n/3$, if the maximum degree is at most two the running time is proportional to $\mathcal{I}^{\leq k}(n)$ by the above. Suppose that the maximum degree is at least three. Then

$$T(n,k)\leq T(n-4,k-1)+T(n-1,k)$$

which is at most

$$2\cdot 3^{\lfloor(n-4)/3\rfloor}+2\cdot 3^{\lfloor(n-1)/3\rfloor}=8/9\cdot 3^{n/3}$$

if $n\equiv 0 \pmod 3$,

$$3^{(n-4)/3}+3^{(n-1)/3}=4\cdot 3^{\lfloor n/3\rfloor-1}$$

if $n\equiv 1 \pmod 3$ and

$$4\cdot 3^{\lfloor(n-4)/3\rfloor-1}+4\cdot 3^{\lfloor(n-1)/3\rfloor-1}=16/9\cdot 3^{\lfloor n/3\rfloor}$$

if $n\equiv 2 \pmod 3$. In all cases this is at most $\mathcal{I}^{\leq k}(n)$; thus, the algorithm runs in time $\mathcal{O}^*\left(\mathcal{I}^{\leq k}(n)\right)$. □

Hujter and Tuza [4] have shown that triangle-free graphs (i.e. graphs having no triangles as subgraphs) can have at most $2^{n/2}$ maximal independent sets. This we can show by an algorithmic proof similar to the above.

Proposition 1. *If G is a triangle-free graph the number of maximal independent sets in G is at most $2^{n/2}$, and the bound is tight. Moreover, there is an algorithm that takes a triangle-free graph as input and outputs all maximal independent sets in the graph in time $\mathcal{O}^*\left(2^{n/2}\right)$.*

Proof. Let G be a triangle-free graph. As long as the maximum degree of the graph is greater than two, we just branch on the maximum-degree vertex. Either it is in the maximal independent set, and thus none of it neighbours are, or it is not. This removes at least four vertices in one branch and one in the other; thus, the number of leaves in the recursion tree is at most $T(n) \leq T(n-1) + T(n-4)$, which has the solution $T(n) = 2^{0.4650 \cdot n}$. If at some point the maximum degree is at most two the remaining graph consists of only paths and cycles. Since there are no triangles the worst case is paths of length two, yielding $2^{n/2}$ maximal independent sets, which is worse than the above. The bound is tight since a graph consisting of $n/2$ disconnected K_2's achieves the bound for even n. □

4 Application to colouring

Consider a k-colouring of a graph. Each colour class is an independent set, and the largest has size at least n/k and can be extended to a maximal independent set. Thus to check whether a graph is k-colourable it suffices to check for all maximal independent sets of size at least n/k, whether the remaining graph is $(k-1)$-colourable, which can be done by applying the method recursively and using a polynomial time algorithm for checking 2-colourability. For checking 3-colourability, Eppstein [2] has devised an algorithm running in time $T_3(n) = \mathcal{O}(1.3289^n)$ which is faster than the above. Using his algorithm for checking 3-colourability we get the following running times for 4- and 5-colourability.

Theorem 3. *It can be checked in time $\mathcal{O}(1.7504^n)$, whether a graph is 4-colourable.*

Proof. The running time of the above-mentioned algorithm is at most

$$\sum_{k=\lceil \frac{n}{4} \rceil}^{n} I^{=k}(G) \cdot T_3(n-k).$$

By Eppstein [3], $3^{4k-n}4^{n-3k}$ is an upper bound on the number of maximal independent sets for all k, and they can be generated in this time. Thus the sum is at most

$$\sum_{k=\lceil \frac{n}{4} \rceil}^{n} 3^{4k-n}4^{n-3k} \cdot T_3(n-k).$$

As the terms are decreasing as a function of k, this is at most

$$n \cdot 4^{n/4} \cdot T_3(3n/4).$$

Thus the algorithm runs in time $\mathcal{O}(1.7504^n)$, where we have dropped the factor of n since we have rounded up. □

Theorem 4. *It can be checked in time $\mathcal{O}(2.1593^n)$, whether a graph is 5-colourable.*

Proof. Let T_4 denote the running time of the above algorithm for checking 4-colourability. The running time for checking 5-colourability is at most

$$\sum_{k=\lceil \frac{n}{5} \rceil}^{n} I^{=k}(G) \cdot T_4(n-k)$$

which is no larger than

$$\sum_{k=\lceil \frac{n}{5} \rceil}^{\lfloor \frac{n}{4} \rfloor} 4^{5k-n} 5^{n-4k} \cdot T_4(n-k) + \sum_{k=\lceil \frac{n}{4} \rceil}^{n} 3^{4k-n} 4^{n-3k} \cdot T_4(n-k).$$

Since the terms in both sums are decreasing as functions of k, and the largest is the first term in the first sum, this is at most

$$n \cdot 5^{n/5} \cdot T_4(4n/5),$$

which is equal to

$$n^2 \cdot 20^{n/5} \cdot T_3(3n/5).$$

Thus the algorithm runs in time $\mathcal{O}(2.1593^n)$. □

5 Conclusion

In this paper we give new bounds on the number of maximal independent sets of size k, and thus strengthening Eppstein's [3] bound to be tight for all values of k. Eppstein uses his bound to prove better time bounds for colouring graphs with the minimum possible number of colours, but for our result to improve this, 4-colouring algorithms running in time $o(1.4150^n)$ are needed (see [3]), which is far from the running time of $\mathcal{O}(1.7504^n)$ proven in this paper.

Our result is algorithmic in the sense that all maximal independent sets of size at most k can be listed, but only in time proportional to our bound, not their actual number. Algorithms for generating *all* maximal independent sets running in time proportional to their number exists; Johnson et al. [5] give an algorithm that in fact outputs all maximal independent sets with only polynomial delay. We would like to find such an algorithm for maximal independent sets of size k.

Our result gives no information about the distribution of sizes of maximal independent sets. It does not tell whether the existence of many maximal independent sets of one size limits the number of maximal independent sets of another size. Such bounds would be an interesting goal of further research.

Acknowledgements

I would like to thank my advisor, Peter Bro Miltersen, for many helpful comments and insights and also Bjarke Skjernaa for comments on and corrections to this paper.

References

[1] Cornelius Croitoru. On stables in graphs. In *Proc. 3rd Coll. Operations Research (Cluj-Napoca, 1978)*, pages 55–60. Univ. "Babeş-Bolyai", Cluj-Napoca, 1979.

[2] David Eppstein. Improved algorithms for 3-coloring, 3-edge-coloring, and constraint satisfaction. In *Proc. 12th Symp. Discrete Algorithms*, pages 329–337. ACM and SIAM, 2001.

[3] David Eppstein. Small maximal independent sets and faster exact graph coloring. In *Proc. 7th Worksh. Algorithms and Data Structures*, volume 2125 of *Lecture Notes in Computer Science*, pages 462–470. Springer-Verlag, 2001.

[4] Mihály Hujter and Zsolt Tuza. The number of maximal independent sets in triangle-free graphs. *SIAM Journal on Discrete Mathematics*, 6(2):284–288, 1993.

[5] David S. Johnson, Mihalis Yannakakis, and Christos H. Papadimitriou. On generating all maximal independent sets. *Information Processing Letters*, 27(3):119–123, 1988.

[6] J. W. Moon and L. Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3:23–28, 1965.

Recent BRICS Report Series Publications

RS-02-15 **Jesper Makholm Nielsen.** ***On the Number of Maximal Independent Sets in a Graph*****. April 2002. 10 pp.**

RS-02-14 **Ulrich Berger and Paulo B. Oliva.** ***Modified Bar Recursion*****. April 2002. 22 pp.**

RS-02-13 **Gerth Stølting Brodal, Rune B. Lyngsø, Anna Östlin, and Christian N.S̃. Pedersen.** ***Solving the String Statistics Problem in Time $O(n \log n)$*****. March 2002. To appear in ICALP '02.**

RS-02-12 **Olivier Danvy and Mayer Goldberg.** ***There and Back Again*****. March 2002. This report supersedes the earlier report BRICS RS-01-39.**

RS-02-11 **Aske Simon Christensen, Anders Møller, and Michael I. Schwartzbach.** ***Extending Java for High-Level Web Service Construction*****. March 2002.**

RS-02-10 **Ulrich Kohlenbach.** ***Uniform Asymptotic Regularity for Mann Iterates*****. March 2002. 17 pp.**

RS-02-9 **Anna Östlin and Rasmus Pagh.** ***One-Probe Search*****. February 2002. 17 pp.**

RS-02-8 **Ronald Cramer and Serge Fehr.** ***Optimal Black-Box Secret Sharing over Arbitrary Abelian Groups*****. February 2002. 19 pp.**

RS-02-7 **Anna Ingólfsdóttir, Anders Lyhne Christensen, Jens Alsted Hansen, Jacob Johnsen, John Knudsen, and Jacob Illum Rasmussen.** ***A Formalization of Linkage Analysis*****. February 2002. vi+109 pp.**

RS-02-6 **Luca Aceto, Zoltán Ésik, and Anna Ingólfsdóttir.** ***Equational Axioms for Probabilistic Bisimilarity (Preliminary Report)*****. February 2002. 22 pp.**

RS-02-5 **Federico Crazzolara and Glynn Winskel.** ***Composing Strand Spaces*****. February 2002. 30 pp.**

RS-02-4 **Olivier Danvy and Lasse R. Nielsen.** ***Syntactic Theories in Practice*****. January 2002. 34 pp. This revised report supersedes the earlier BRICS report RS-01-31.**

RS-02-3 **Olivier Danvy and Lasse R. Nielsen.** ***On One-Pass CPS Transformations*****. January 2002. 18 pp.**