

KRZYSZTOF R. APT,  
PETER VAN EMDE BOAS  
AND ANGELO WELLING

**The STO-problem is NP-hard**

CT-94-08, *received:* April 1994

*ILLC Research Report and Technical Notes Series*  
Series editor: Dick de Jongh

*Computation and Complexity Theory (CT) Series, ISSN: 0928-3323*

Institute for Logic, Language and Computation (ILLC)  
University of Amsterdam  
Plantage Muidergracht 24  
NL-1018 TV Amsterdam  
The Netherlands  
e-mail: [ilc@fwi.uva.nl](mailto:ilc@fwi.uva.nl)

# The STO-problem is NP-hard

Krzysztof R. Apt  
CWI

*P.O. Box 94079, 1090 GB Amsterdam, The Netherlands*  
and

*Faculty of Mathematics and Computer Science*  
*University of Amsterdam, Plantage Muidergracht 24*  
*1018 TV Amsterdam, The Netherlands*

Peter van Emde Boas, Angelo Welling  
*Faculty of Mathematics and Computer Science*  
*University of Amsterdam, Plantage Muidergracht 24*  
*1018 TV Amsterdam, The Netherlands*

## Abstract

A finite set of term equations  $E$  is called subject to the occur-check (STO) if a sequence of actions of the Martelli-Montanari unification algorithm starts with  $E$  and ends with a positive occur-check. We prove here that the problem of deciding whether  $E$  is STO is NP-hard.

## 1 Introduction

For efficiency reasons in most Prolog implementations the so-called occur-check is omitted from the unification algorithm. This naturally calls for a definition of unification without the occur-check and for a characterization of the sets of term equations for which this omission might be of importance for unification purposes. The latter has been offered by Deransart, Ferrand and Tégua [1], who introduced the notion of a set of equations being STO (*Subject To Occur-check*). Informally, a set of equations is STO if some sequence of actions of the nondeterministic Martelli-Montanari unification algorithm leads to a situation in which the failure due to the occur-check arises. As the known unification algorithms - see, for example, Robinson [7, 8], Venturini-Zilli [10], Martelli and Montanari [5], Paterson and Wegman [6] - are special cases of the Martelli-Montanari algorithm, this concept describes when unification without the occur-check might lead to problems. This is apparently as close as one can get to a characterization of the sets of equations for which unification might depend on the presence of the occur-check. Therefore, not surprisingly, the definition of an STO set of equations entered the proposal for standard Prolog (see Scowen [9]).

The result of this paper indicates an unexpected difference between the two relevant properties of sets of equations. As was shown by Paterson and Wegman [6] the property of being unifiable can be tested in linear time. We prove that the property of being STO is NP-hard. Recall that a problem is NP-hard, if its solvability in polynomial time implies that every problem in the class NP is solvable in polynomial time. This shows that, for all practical purposes, the definition of standard Prolog refers to a computationally intractable concept.

## 2 Preliminaries

Throughout the paper, the symbol  $\equiv$  (resp.  $\neq$ ) is used to indicate syntactic equality (resp. inequality), the set of variables occurring in any syntactic object  $O$  is denoted by  $Var(O)$  and the arity of a function symbol  $f$  is denoted by  $Arity(f)$ . A function symbol of of arity 0 is called a constant.

From now on we fix a finite set of function symbols  $F$  and a finite set of variables  $V$ . The class of *terms over  $F$  and  $V$*  is defined recursively as follows:

- a variable is a term,
- if  $t_1, \dots, t_n$  are terms,  $f \in F$ ,  $Arity(f) = n$ , then  $f(t_1, \dots, t_n)$  is a term.

A *substitution* is a finite mapping from variables to terms which assigns to each variable  $x$  in its domain a term  $t$  different from  $x$ . We write it as

$$\{x_1/t_1, \dots, x_n/t_n\}$$

where

- $x_1, \dots, x_n$  are different variables,
- $t_1, \dots, t_n$  are terms,
- for  $i \in [1, n]$ ,  $x_i \neq t_i$ .

The application of a substitution to a (set of) term(s) and the relation "more general than" between the substitutions is defined in the usual way. A *set of equations  $E$*  is a finite set of the form  $\{s_1 = t_1, s_2 = t_2, \dots, s_n = t_n\}$ , where  $s_i$  and  $t_i$  are terms, for  $1 \leq i \leq n$ . A substitution  $\sigma$  such that  $s_1\sigma \equiv t_1\sigma, \dots, s_n\sigma \equiv t_n\sigma$  is called a *unifier* of  $E$ . A unifier of  $E$  is called a *most general unifier* (in short: *mgu*) of  $E$  if it is more general than all unifiers of  $E$ . Finally, we denote by  $|E|$  the number of equations in  $E$ .

The problem of deciding whether a set of equations has a unifier is called the *unification problem*. This problem was introduced and solved by Robinson [7] by providing a unification algorithm. For our purposes we need the following nondeterministic unification algorithm due to Martelli and Montanari [5] (and informally introduced by Herbrand [4]).

**Martelli-Montanari algorithm.** Given a set of equations, choose any equation of a form indicated below and perform the associated action. If no action applies to any equation, stop with success.

1.  $E \cup \{f(s_1, \dots, s_n) = g(t_1, \dots, t_m)\} \rightarrow \text{fail: clash}$   
 $f \neq g$
2.  $E \cup \{f(s_1, \dots, s_n) = f(t_1, \dots, t_n)\} \rightarrow E \cup \{s_1 = t_1, \dots, s_n = t_n\}$
3.  $E \cup \{x = x\} \rightarrow E$   
 $x \in V$
4.  $E \cup \{t = x\} \rightarrow E \cup \{x = t\}$   
 $x \in V, t \notin V$
5.  $E \cup \{x = t\} \rightarrow \text{fail: positive occur-check}$   
 $x \in V, t \notin V, x \in Var(t)$
6.  $E \cup \{x = t\} \rightarrow E\{x/t\} \cup \{x = t\}$   
 $x \in V, x \in Var(E), x \notin Var(t)$

The condition  $x \notin \text{Var}(t)$  in action 6. is called the *occur-check test*. The following result is due to Martelli and Montanari [5].

**Theorem 2.1** *The Martelli-Montanari algorithm always terminates. If the original set of equations  $E$  has a unifier, then the algorithm terminates with success and produces an mgu of  $E$  written in an equational form, and otherwise it terminates with failure.  $\square$ .*

Deransart, Ferrand and Tégua [1] introduced the following notion.

**Definition 2.2** *A set of equations  $E$  is subject to the occur-check (STO) iff a sequence of actions of the Martelli-Montanari algorithm starts with  $E$  and ends with action 5.  $E$  is not subject to the occur-check (NSTO) iff it is not STO.  $\square$ .*

Intuitively,  $E$  is NSTO iff unification and unification without the occur-check coincide for  $E$ . By Theorem 2.1 if an execution of the Martelli-Montanari algorithm terminates with success, the initial set of equations is NSTO. On the other hand, if an execution of the algorithm terminates with failure, the initial set of equations may be NSTO or STO. Consider for example the sets  $\{a = f(a)\}$  and  $\{x = f(x)\}$  with  $a$  a constant. Moreover, for some sets of equations different executions of the algorithm can terminate with failure for different reasons. Consider for example the set  $\{a = f(a), x = f(x)\}$ .

Scowen [9] lists the requirements for a formal definition of unification within standard Prolog. One of them (see top of page 934), when properly formalized, states that unification is undefined if the original set of equations is STO.

We show in this paper that the problem of deciding whether a set of equations is STO (in short: *the STO-problem*) is NP-hard.

### 3 The STO-problem is NP-hard

The following lemma allows us to reduce the STO test to simpler sets of equations.

**Lemma 3.1** [STO]

1. If  $f \neq g$ , then  $E \cup \{f(s_1, \dots, s_n) = g(t_1, \dots, t_m)\}$  is STO iff  $E$  is STO.
2. If  $x \in V, x \notin \text{Var}(E) \cup \text{Var}(t)$ , then  $E \cup \{x = t\}$  is STO iff  $E$  is STO.
3.  $E \cup \{f(s_1, \dots, s_n) = f(t_1, \dots, t_n)\}$  is STO iff  $E \cup \{s_1 = t_1, \dots, s_n = t_n\}$  is STO.
4. If  $x \in V$ , then  $E \cup \{t = x\}$  is STO iff  $E \cup \{x = t\}$  is STO.

**Proof:** Properties 1. and 2. are obvious whereas 3. and 4. were proved in Deransart and Maluszynski [2].  $\square$

**Definition 3.2** *Given a set of equations  $E$ , we denote by  $\text{Stand}(E)$  the set of equations which is obtained from  $E$  by applying as many times as possible actions 2. and 4. of the Martelli-Montanari algorithm and by deleting the equations according to the STO Lemma 3.1.1 and 3.1.2.  $\square$*

This brings us to the following conclusion.

**Theorem 3.3**  *$E$  is STO iff  $\text{Stand}(E)$  is STO.*

**Proof:** By the STO Lemma 3.1.  $\square$

**Definition 3.4** *Consider a set of equations  $E$ . A subset  $E'$  of  $E$  is closed within  $E$  if for some variable  $x \notin \text{Var}(E - E')$  all equations of  $E'$  are of the form  $x = s$ , where  $x \notin \text{Var}(s)$ .  $\square$*

For example, the set  $E = \{x = f(y), x = y, z = f(u), y = a\}$  has two subsets closed within  $E$ :  $\{x = f(y), x = y\}$  and  $\{z = f(u)\}$ . Note that  $\{y = a\}$  is not closed within  $E$  since  $y \in \text{Var}(E - \{y = a\})$ .

Observe that when  $E'$  is closed within  $E$ , then only action 6. can be applied to an equation from  $E'$ . This brings us to the following definition.

**Definition 3.5** *Consider a set of equations  $E$  and its subset  $E'$  closed within  $E$ . Let  $|E'| = k$ . The set  $\text{Reduce}(E')$  consists of  $k$  sets of equations, each of which is obtained from  $E'$  by applying action 6. to a different equation from  $E'$ . We denote by  $\text{Reduce}(E')(m)$ , with  $1 \leq m \leq k$ , the  $m^{\text{th}}$  element of  $\text{Reduce}(E')$  in some enumeration.  $\square$*

**Lemma 3.6** *Consider a set of equations  $E$  and its subset  $E'$  closed within  $E$ . Then  $E$  is STO iff for some  $m, 1 \leq m \leq |E'|$ , the set  $(E - E') \cup \text{Reduce}(E')(m)$  is STO.*

**Proof:** ( $\Rightarrow$ ) Suppose  $E$  is STO. Consider a sequence of actions which leads to action 5. If this sequence does not select (an instance of) an equation from  $E'$  somewhere, the same sequence can be applied to  $E - E'$ , so a fortiori to  $(E - E') \cup \text{Reduce}(E')(m)$ , for each  $m$ . So suppose now that this sequence selects (an instance of) an equation  $x = s$  from  $E'$ . Consider the first such selection. By the form of  $E'$  the performed action is then action 6.

Let  $E_1$  be the resulting set of equations. Thanks to the fact that  $x \notin \text{Var}(E - E')$  the actions preceding this selection of  $x = s$  do not introduce new occurrences of  $x$  in the considered sets of equations. Consequently,  $E_1$  can also be obtained from  $E$  by a transposed sequence of actions in which the equation  $x = s$  is selected first and then the original sequence of actions up to the selection of  $x = s$  is performed. Consequently for some  $m, 1 \leq m \leq |E'|$ , the set  $E_1$  can be obtained from  $(E - E') \cup \text{Reduce}(E')(m)$ , so  $(E - E') \cup \text{Reduce}(E')(m)$  is STO.

( $\Leftarrow$ ) By the fact that for all  $m, 1 \leq m \leq |E'|$ , the set  $E$  reduces to  $(E - E') \cup \text{Reduce}(E')(m)$  by action 6.  $\square$

Intuitively, this lemma states that to determine whether  $E$  is STO it is sufficient to limit one's attention to the sequences of actions which start with action 6. applied to an equation in a subset of  $E'$  which is closed within  $E$ . We are now in position to prove the desired result.

**Theorem 3.7** *The STO-problem is NP-hard.*

**Proof:** We provide a reduction from the known NP-Complete Satisfiability Problem (see e.g. Garey and Johnson [3]) to the STO-problem. Let  $U = \{u_1, u_2, \dots, u_n\}$  be a set of variables and  $C = \{c_1, \dots, c_m\}$  be a set of clauses making up an arbitrary instance of the Satisfiability Problem. A set of equations  $E$  is constructed such that  $E$  is STO if and only if  $C$  is satisfiable.  $E$  is a union of  $n$  disjoint subsets  $E_1, E_2, \dots, E_n$ . Each  $E_i$  consists of four equations; two of them are associated with  $u_i$  and two with  $\bar{u}_i$ , the complement of  $u_i$ .

First, we define a set  $V$  of variable symbols and a set  $F$  of function symbols over which the terms occurring in  $E$  are build:

$$V = \{x_i \mid 1 \leq i \leq n\} \cup \{z_j \mid 1 \leq j \leq m\}, F = \{f^i, g^i \mid 1 \leq i \leq n\} \cup \{h\}.$$

The arity of  $h$  is independent of the form of the particular instance of Satisfiability and is equal to one, whereas the arities of  $f^i$  and  $g^i$  do depend on this form in a way which will be described below. In the following, “+1” denotes the “increment modulo  $m$ ” over the set  $\{1, \dots, m\}$ , so  $m + 1 = 1$ .

We are now ready to define the sets  $E_i, 1 \leq i \leq n$ . Let  $C_i$  be the set of clauses of  $C$  which contain  $u_i$ . Two terms,  $s_{i,1}$  and  $s_{i,2}$ , are constructed with the function symbol  $f^i$  as the outer constructor, whose arity is now defined to be equal to the cardinality of  $C_i$ . Suppose the  $j^{\text{th}}$  clause of  $C_i$  is  $c_k$ . Then

$$s_{i,1} \equiv f^i(\dots, z_k, \dots) \text{ and } s_{i,2} \equiv f^i(\dots, h(z_{k+1}), \dots),$$

with  $z_k$  and  $h(z_{k+1})$  being the  $j^{\text{th}}$  arguments of, respectively  $s_{i,1}$  and  $s_{i,2}$ .  $C_i$  contributes to  $E_i$  two equations

$$x_i = s_{i,1} \text{ and } x_i = s_{i,2}.$$

Let  $\bar{C}_i$  be the set of clauses of  $C$  which contain  $\bar{u}_i$ . In the same way as above two terms  $t_{i,1}$  and  $t_{i,2}$  are constructed using the function symbol  $g^i$ .  $\bar{C}_i$  contributes to  $E_i$  two equations

$$x_i = t_{i,1} \text{ and } x_i = t_{i,2}.$$

As an example of this construction, consider the following instance of the Satisfiability Problem:  $U = \{u_1, u_2\}, C = \{c_1, c_2\}$ , with  $c_1 = \{u_1, \bar{u}_2\}$  and  $c_2 = \{u_1, \bar{u}_1, u_2\}$ . It yields the following set of equations:

$$\begin{aligned} & \{x_1 = f^1(z_1, z_2), x_1 = f^1(h(z_2), h(z_1)), x_1 = g^1(z_2), x_1 = g^1(h(z_1))\} \\ \cup & \{x_2 = f^2(z_2), x_2 = f^2(h(z_1)), x_2 = g^2(z_1), x_2 = g^2(h(z_2))\}. \end{aligned}$$

Given a truth assignment  $T : U \rightarrow \{T, F\}$  we denote below its restriction to the variable  $u_i$  by  $t[u_i]$ . Each subset  $E_i$  is closed within  $E$ , so applying Lemma 3.6  $n$  times we get

$E$  is STO iff there are  $l_1, \dots, l_n, 1 \leq l_i \leq 4$ , such that  $\bigcup_{i=1}^n \text{Reduce}(E_i)(l_i)$  is STO.

Fix such a sequence  $l_1, \dots, l_n$ . By Theorem 3.3

$\bigcup_{i=1}^n \text{Reduce}(E_i)(l_i)$  is STO iff  $\text{Stand}(\bigcup_{i=1}^n \text{Reduce}(E_i)(l_i))$  is STO.

Now for some truth assignment  $t : U \rightarrow \{T, F\}$ , (namely the one defined by  $t(u_i) = \mathbf{if } l_i \leq 2 \text{ then } T \text{ else } F \mathbf{fi}$ ,  $1 \leq i \leq n$ )

$$\text{Stand}\left(\bigcup_{i=1}^n \text{Reduce}(E_i)(l_i)\right) = \bigcup_{i=1}^n \{z_j = h(z_{j+1}) \mid c_j \text{ is true under } t[i]\}.$$

But for every truth assignment  $t : U \rightarrow \{T, F\}$

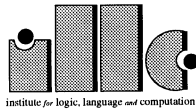
$$\bigcup_{i=1}^n \{z_j = h(z_{j+1}) \mid c_j \text{ is true under } t[i]\} = \{z_j = h(z_{j+1}) \mid c_j \text{ is true under } t\},$$

and the latter set is STO iff it equals  $\{z_j = h(z_{j+1}) \mid 1 \leq j \leq m\}$ . Now the equality between these last two sets holds iff all clauses of  $C$  are true under  $t$ . Thus  $E$  is STO iff  $C$  is satisfiable.

It is clear that the construction of  $E$  from  $C$  can be accomplished in polynomial time, as for each variable  $u_i \in U$  at most  $m$  clauses have to be checked for the occurrences of  $u_i$  and  $\bar{u}_i$ .  $\square$

## References

- [1] P. Deransart, G. Ferrand, M. Tégua. *NSTO programs (not subject to occur-check)*. In V. Saraswat and K. Ueda, editors, *Proceedings of the International Logic Symposium*, p. 533-547. The MIT Press, 1991.
- [2] P. Deransart, J. Maluszynski. *A Grammatical View of Logic Programming*. The MIT Press, 1993.
- [3] M. Garey, D. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.
- [4] J. Herbrand. *Recherches sur la théorie de la démonstration*. Thèse de l'Université de Paris, 1930. English translation in *Jacques Herbrand: Logical writings*. Ed. W. Goldfarb. Harvard, 1971, p. 148.
- [5] A. Martelli, U. Montanari. *An efficient unification algorithm*. In *ACM Transactions on Programming Languages and Systems*, Vol. 4, No. 2, 1982, p. 258 - 282.
- [6] M. Paterson, M. Wegman. *Linear unification*. In *Proceedings of the Symposium on the theory of computing*. ACM Special Interest Group for automata and computability theory, 1976, p. 181 - 186.
- [7] J. A. Robinson. *A machine oriented logic based on the resolution principle*. In *Journal of the Association for Computing Machinery*, Vol. 12, No. 1, 1965, p. 23-41.
- [8] J. A. Robinson. *Computational logic: The unification computation*. In *Machine Intelligence*, Vol. 6, 1971, p. 63-72.
- [9] R.S. Scowen. *An overview of Prolog standardization – progress, problems and solutions*. In K. Furukawa, editor, *Proceedings of the Eighth International Conference on Logic Programming*, pages 922–936, Paris, France, 1991. The MIT Press.
- [10] M. Venturini-Zilli. *Complexity of the unification algorithm for first-order expressions*. In *Calcolo*. Vol. 12, No. 4, 1975, p. 361-372.



## ILLC Research Reports and Technical Notes

Coding for Research Reports: *Series-Year-Number*, with LP = Linguistics and Philosophy of Language; ML = Mathematical Logic and Foundations; CL = Computational Linguistics; CT = Computation and Complexity Theory; X = Technical Notes.

All previous ILLC-publications are available from the ILLC bureau. For prepublications before 1993, contact the bureau.

- LP-93-01 *Martijn Spaan, Parallel Quantification*
- LP-93-02 *Makoto Kanazawa, Dynamic Generalized Quantifiers and Monotonicity*
- LP-93-03 *Nikolai Pankrat'ev, Completeness of the Lambek Calculus with respect to Relativized Relational Semantics*
- LP-93-04 *Jacques van Leeuwen, Identity, Quarrelling with an Unproblematic Notion*
- LP-93-05 *Jaap van der Does, Sums and Quantifiers*
- LP-93-06 *Paul Dekker, Updates in Dynamic Semantics*
- LP-93-07 *Wojciech Buszkowski, On the Equivalence of Lambek Categorical Grammars and Basic Categorical Grammars*
- LP-93-08 *Zisheng Huang, Peter van Emde Boas, Information Acquisition from Multi-Agent resources; abstract*
- LP-93-09 *Makoto Kanazawa, Completeness and Decidability of the Mixed Style of Inference with Composition*
- LP-93-10 *Makoto Kanazawa, Weak vs. Strong Readings of Donkey Sentences and Monotonicity Inference in a Dynamic Setting*
- LP-93-11 *Friederike Moltmann, Resumptive Quantifiers in Exception Sentences*
- LP-93-12 *Jaap van der Does, On Complex Plural Noun Phrases*
- LP-93-13 *Natasha Alechina, Binary Quantifiers and Relational Semantics*
- LP-93-14 *Mati Pentus, Lambek Calculus is L-complete*
- LP-93-15 *David Ian Beaver, What comes first in Dynamic Semantics*

- ML-93-01 *Maciej Kandulski, Commutative Lambek Categorical Grammars*
- ML-93-02 *Johan van Benthem, Natasha Alechina, Modal Quantification over Structured Domains*
- ML-93-03 *Mati Pentus, The Conjoinability Relation in Lambek Calculus and Linear Logic*
- ML-93-04 *Andreja Prijatelj, Bounded Contraction and Many-Valued Semantics*
- ML-93-05 *Raymond Hoofman, Harold Schellinx, Models of the Untyped  $\lambda$ -calculus in Semi Cartesian Closed Categories*
- ML-93-06 *J. Zaslavski, Categorical Generalization of Algebraic Recursion Theory*
- ML-93-07 *A.V. Chagrov, L.A. Chagrova, Algorithmic Problems Concerning First-Order Definability of Modal Formulas on the Class of All Finite Frames*
- ML-93-08 *Raymond Hoofman, Ieke Moerdijk, Remarks on the Theory of Semi-Functors*
- ML-93-09 *A.S. Troelstra, Natural Deduction for Intuitionistic Linear Logic*
- ML-93-10 *Vincent Danos, Jean-Baptiste Joinet, Harold Schellinx, The Structure of Exponentials: Uncovering the Dynamics of Linear Logic Proofs*
- ML-93-11 *Lex Hendriks, Inventory of Fragments and Exact Models in Intuitionistic Propositional Logic*
- ML-93-12 *V.Yu. Shavrukov, Remarks on Uniformly Finitely Precomplete Positive Equivalences*
- ML-93-13 *V.Yu. Shavrukov, Undecidability in Diagonizable Algebras*
- ML-93-14 *Dick de Jongh, Albert Visser, Embeddings of Heyting Algebras*
- ML-93-15 *G.K. Dzhabaridze, Effective Truth*
- ML-93-16 *Maarten de Rijke, Correspondence Theory for Extended Modal Logics*
- ML-93-17 *Alexander Chagrov, Michael Zakharyashev, On the Independent Axiomatizability of Modal and Intermediate Logics*
- ML-93-18 *Jaap van Oosten, Extensional Realizability*
- ML-93-19 *Raymond Hoofman, Comparing Models of the Non-Extensional Typed  $\lambda$ -Calculus*
- ML-93-20 *L.A. Chagrova, Dick de Jongh, The Decidability of Dependency in Intuitionistic Propositional Logic*
- ML-93-21 *Max I. Kanovich, The Relational Knowledge-Base Interpretation and Feasible Theorem Proving for Intuitionistic Propositional Logic*



ML-93-22 *Andreja Prijatelj, Connectification for  $n$ -contraction*

CT-93-01 *Marianne Kalsbeek, The Vanilla Meta-Interpreter for Definite Logic Programs and Ambivalent Syntax*

CT-93-02 *Sophie Fischer, A Note on the Complexity of Local Search Problems*

CT-93-03 *Johan van Benthem, Jan Bergstra, Logic of Transition Systems*

CT-93-04 *Karen L. Kwast, Sieger van Denneheuvel, The Meaning of Duplicates in the Relational Database Model*

CT-93-05 *Erik Aarts, Proving Theorems of the Lambek Calculus of Order 2 in Polynomial Time*

CT-93-06 *Krzysztof R. Apt, Declarative programming in Prolog*

CT-93-07 *Janusz A. Pomykala, Approximation, Similarity and Rough Constructions, Part I. Elementary Introduction*

CL-93-01 *Noor van Leusen, László Kálmán, Computational Linguistics*

CL-93-02 *Theo M.V. Janssen, An Algebraic View On Rosetta*

CL-93-03 *Patrick Blackburn, Claire Gardent, Wilfried Meyer-Viol, Talking about Trees*

X-93-01 *Paul Dekker, Existential Disclosure, revised version*

X-93-02 *Maarten de Rijke, What is Modal Logic?*

X-93-03 *Michiel Leezenberg, Gorani Influence on Central Kurdish: Substratum or Prestige Borrowing*

X-93-04 *A.S. Troelstra (editor), Metamathematical Investigation of Intuitionistic Arithmetic and Analysis, Corrections to the First Edition*

X-93-05 *A.S. Troelstra (editor), Metamathematical Investigation of Intuitionistic Arithmetic and Analysis, Second, corrected Edition*

X-93-06 *Michael Zakharyashev, Canonical Formulas for K4. Part II: Cofinal Subframe Logics*

ML-94-01 *Domenico Zambella, Notes on polynomially bounded arithmetic*

LP-94-01 *Dimitar Gelev, Introducing Some Classical Elements of Modal Logic to the Propositional Logics of Qualitative Probabilities*

LP-94-02 *Andrei Arsov, Basic Arrow Logic with Relation Algebraic Operators*

LP-94-03 *Jerry Seligman, An algebraic appreciation of diagrams*

LP-94-04 *Kazimierz Świrydowicz, A Remark on the Maximal Extensions of the Relevant Logic R*

LP-94-05 *Natasha Kurtonina, The Lambek Calculus: Relational Semantics and the Method of Labelling*

CT-94-01 *Harry Buhrman and Leen Torenvliet, On the Cutting Edge of Relativization: the Resource Bounded Injury Method*

CT-94-02 *Alessandro Panconesi, Marina Papatriantafidou, Philippas Tsigas, Paul Vitányi, Randomized Wait-Free Distributed Naming*

CT-94-03 *Ming Lee, John Tromp, Paul Vitányi, Sharpening Occam's Razor (extended abstract)*

CT-94-04 *Ming Lee and Paul Vitányi, Inductive Reasoning*

CT-94-05 *Tao Jiang, Joel I. Seiferas, Paul M.B. Vitányi, Two heads are Better than Two Tapes*

CT-94-06 *Guido te Brake, Joost N. Kok, Paul Vitányi, Model Selection for Neural Networks: Comparing MDL and NIC*

CT-94-07 *Charles H. Bennett, Péter Gács, Ming Li, Paul M.B. Vitányi, Wojciech H. Zurek, Thermodynamics of Computation and Information Distance*

CT-94-08 *Krzysztof R. Apt, Peter van Emde Boas and Angelo Welling, The STO-problem is NP-hard*

X-94-01 *Johan van Benthem, Two Essays on Semantic Modelling*

*Titles in the ILLC Dissertation Series:*

- 1993-1 *Transsentential Meditations; Ups and downs in dynamic semantics*, Paul Dekker
- 1993-2 *Resource Bounded Reductions*, Harry Buhrman
- 1993-3 *Efficient Metamathematics*, Rineke Verbrugge
- 1993-4 *Extending Modal Logic*, Maarten de Rijke
- 1993-5 *Studied Flexibility*, Herman Hendriks
- 1993-6 *Aspects of Algorithms and Complexity*, John Tromp
- 1994-1 *The Noble Art of Linear Decorating*, Harold Schellinx
- 1994-2 *Generating Uniform User-Interfaces for Interactive Programming Environments*, Jan Willem Cornelis Koorn
- 1994-3 *Process Theory and Equation Solving*, Nicoline Johanna Drost