DOWNLOAD PDF 7.4 TERMINATION OF TERM REWRITING SYSTEMS141

Chapter 1 : Search Techniques for Rational Polynomial Orders - CORE

In contrast to string rewriting systems, whose objects are flat sequences of symbols, the objects a term rewriting system works on, i.e. the terms, form a term algebra. A term can be visualized as a tree of symbols, the set of admitted symbols being fixed by a given signature.

Logic programming Algorithms Term-rewriting systems. These are essentially sets of equations directed from left to right, indicating that instances of the left-hand side may be replaced by the corresponding instances of the right-hand side. Such systems may be used as programming languages, and are also useful for theorem-proving applications. We are investigating methods of combining term-rewriting systems with first-order theorem provers. We developed a way of handling AC symbols using permutations and implemented it. We also studied techniques for applying efficient permutation group algorithms to equational theorem proving. We are investigating methods to automate the process of searching for and checking proofs. This has applications to program verification and to teaching logic and mathematics, since a proof checker can be helpful for instructional purposes. We have developed a sequence of theorem-proving methods, the most recent being clause linking with semantics and ordered semantic hyper-linking. At the same time, we have developed analytical methods for studying the search efficiencies of theorem-proving strategies. As a result, we have developed provers that are among the best in terms of being able to prove a wide variety of non-equality problems with a minimum of human guidance. We have applied these provers to problems involving concept description languages in AI and planning problems, with good results. We have also investigated rigid E-unification, which is a version of unification relevant for certain theorem proving methods based on "matings". We have developed some simple but effective tests that permit the occurrence check in Prolog to be eliminated in most cases, while guaranteeing the same semantics as true unification. We have developed theoretical methods for automatically generating the inductive assertions in program verification. We would like to see to what extent this can be implemented and applied to program verification. We are studying the proofs-as-programs paradigm, in which programs are expressed as proofs in a logical system. This system permits the generation of programs that are correct with respect to their specifications. Our approach combines constructive and classical logic, with classical logic being used for the parts of the proofs that have no computational content. Also, our approach can handle nondeterminism and nonterminating programs. Book Ad Selected Publications For copies of my publications, send me email. The following articles may also be obtained from your library by interlibrary loan. Reprints of some of my publications are available. Programming by Term Rewriting. Oxford, UK, July 7, Maria Paola Bonacina and David A. Constraint manipulation in SGGS. Semantically-guided goal-sensitive theorem proving Abstract. Swaha Miller and David A. David Plaisted and Adnan Yahya, A relevance restriction strategy for automated deduction, Artificial Intelligence Plaisted, General algorithms for permutations in equational inference, Journal of Automated Reasoning, Vol. Plaisted and Gregory Kucherov, The complexity of some complementation problems, Information Processing Letters 71 Webster, editor, , pp. Paramasivam and David A. Heng Chu and David A. Banff, Canada, July 27 -- August 1, Yunshan Zhu and D. To order, contact elisabeth. Lee, Shie-Jue and D. Plaisted, Controlling the consumption of storage with sliding priority search in a hyper-linking based theorem prover, Computers and Artificial Intelligence Lee, Shie-Jue and Plaisted, D. Ninth Conference on Automated Deduction. Oxford University Press, ,

DOWNLOAD PDF 7.4 TERMINATION OF TERM REWRITING SYSTEMS141

Chapter 2 : Table of contents for Library of Congress control number

A rewrite (term-rewriting) system ~ over a set of terms 3- is a (finite) set of rewrite rules, each of the form l~r, where l and r are terms containing variables ranging over ~,, and such that r only contains variables also in l.

Click here to display publication list. Click to retract publication list. Learning Specifications from Demonstrations. Declarative vs Rule-based Control for Flocking Dynamics. Passive STL learning using only positive examples. Look for the proof to find the program: Smolka and Radu Grosu. A search-based procedure for nonlinear real arithmetic. Formal Methods in System Design: Smolka, and Ashish Tiwari. V-formation as optimal control. In 4th Workshop on Biological Distrbuted Algorithms. Trusted machine learning for probabilistic models. In Reliable Machine Learning in the Wild. Workshop colocated with ICML Severity levels of inconsistent code. Two-restricted one context unification is in polynomial time. One context unification problems solvable in polynomial time. With Adria Gascon and Bruno Dutertre. Synthesis Using Dual Interpretation. Click here for benchmarks, code, and pdf of the paper. See here for SAL models. A slightly modified version is here. Seshia, "Reverse engineering digital circuits using structural and functional analyses". TETC special issue on emerging nanoscale architectures for security, trust and reliability. Finding word-level structures in a sea of bit-level gates".

Chapter 3 : Contents of Term Rewriting and All That

A rewrite (term-rewriting) system!11 over a set of terms:T is a (finite) set of rewrite rules. each of the form {-r. where I and r are terms containing variables ranging over and.

Gramlich, Bernhard Article Type: We investigate restricted termination and confluence properties of term rewriting systems, in particular weak termination, weak innermost termination, strong innermost termination, strong termination, and their interrelations. New criteria are provided which are sufficient for the equivalence of these properties. These criteria provide interesting possibilities to infer completeness, i. Our main result states that any strongly innermost terminating, locally confluent overlay system is terminating, and hence confluent and complete. Using these basic results we are also able to prove some new results about modular termination of rewriting. In particular, we show that â€¦termination is modular for some classes of innermost terminating and locally confluent term rewriting systems, namely for non-overlapping and even for overlay systems. As an easy consequence this latter result also entails a simplified proof of the fact that completeness is a decomposable property of constructor systems. Similarly, a combined overlay system with shared constructors is complete if and only if its component sytems are complete overlay systems. Interestingly, these modularity results are obtained by means of a proof technique which itself constitutes a modular approach. Term rewriting systems, confluence, termination, weak termination, innermost termination, modularity, disjoint union, constructor systems, combination of constructor systems, combined systems with shared constructors DOI: Fundamenta Informaticae , vol. Martin, Ursula Article Type: We describe two new families of orderings on words, and show that each has continuum many distinct members. Steinbach, Joachim Article Type: We focus on termination proof techniques for unconditional term rewriting systems using simplification orderings. Throughout the last few years numerous simplification orderings have been defined by various authors. This paper provides an overview on different aspects of these techniques. Additionally, we introduce a formalism that allows clear representations of orderings. A new kind of transformation of term rewriting systems TRS is proposed, depending on a choice for a model for the TRS. The labelled TRS is obtained from the original one by labelling operation symbols, possibly creating extra copies of some rules. This construction has the remarkable property that the labelled TRS is terminating if and only if the original TRS is terminating. Although the labelled version has more operation symbols and may have more rules sometimes infinitely many , termination is often easier to prove for the labelled TRS than for the original one. Zhang, Hantao Article Type: Contextual rewriting as a generalization of conditional rewriting has been found in different forms in the papers whose major subject is not contextual rewriting. Here, we put the scattered information together and give it a systematic study. Among various properties of contextual rewriting, we show that contextual rewriting is a powerful simplification rule for the first-order theorem proving with equality and preserves the refutational completeness of many reasoning systems. We provide a detailed procedure for simplifying clauses using contextual â€¦rewriting and a solution on how to handle variables which appear only in the condition of a conditional rewrite rule. Narrowing is a universal unification procedure for equational theories defined by a,canonical term rewriting system. In its original form it is extremely inefficient. Therefore, many optimizations have been proposed during the last years. In this paper, we present the narrowing strategies for arbitrary canonical systems in a uniform framework and introduce the new narrowing strategy LSE narrowing. LSE narrowing is complete and improves all other strategies which are complete for arbitrary-canonical systems. It is optimal in the sense that two different LSE narrowing derivations cannot generate the same narrowing substitution. Moreover, LSE narrowing computes only normalized narrowing substitutions. We present a collection of results on regular tree languages and rewrite systems. Moreover we prove the undecidability of the preservation of regularity by rewrite systems. More precisely we prove that it is undecidable whether or not for a set E of equations the set E R congruence closure of set R is a regular tree language whenever R is a regular tree language. It is equally undecidable whether or not for a confluent and terminating rewrite system S the set S R of ground S-normal forms of set R is a regular tree language whenever R is a regular tree language. This paper describes a methodology for parallel theorem proving in a

distributed environment, called deduction by Clause-Diffusion. This methodology utilizes parallelism at the search level, by having concurrent, asynchronous deductive processes searching in parallel the search space of the problem. The search space is partitioned among the processes by distributing the clauses and by subdividing certain classes of inferences. The processes communicate by exchanging data. Policies for distributing the clauses and for scheduling inference and communication steps complete the picture. A distributed derivation is made of the collection of the derivations computed by the concurrent deductive processes and it halts â€¦successfully as soon as one of them does. While the Clause-Diffusion methodology applies to theorem proving in general, it has been designed to provide solutions to the problems in the parallelization of contraction-based strategies, such as rewriting-based methods. We identify backward contraction, i. In parallel implementations of contraction-based strategies in shared memory, this difficulty appears as a write-bottleneck, which we have termed the backward contraction bottleneck. The Clause-Diffusion approach avoids this problem by adopting a mostly distributed memory and distributed global contraction schemes. We conclude by reporting some of our results with an implementation of Clause-Diffusion.

Chapter 4 : Rewriting - Wikipedia

Termination De nition A term rewriting system Ris terminating i! R is terminating, i.e., there is no in nite reduction chain t 0! R t 1! R t 2! R. Termination is.

Chapter 5 : Ashish Tiwari's Homepage

Abstract. We study the termination of rewriting modulo a set of equations in the Calculus of Algebraic Constructions, an extension of the Calculus of Constructions with functions and predicates defined by higher-order rewrite rules.

Chapter 6 : David A. Plaisted's Home Page

We propose a modular approach of term rewriting systems, making the best of their hierarchical structure. We definerewriting modules and then provide a new method to prove termination incrementally. We obtain new and powerful termination criteria for standard rewriting, thanks to the combination of.

Chapter 7 : CiteSeerX â€” Type Assignment and Termination of Interaction Nets

existing techniques for termination detection of term rewriting systems can be found in [5, 22]. In the literature termination is also called strong normali~ation.

Chapter 8 : Fundamenta Informaticae - Volume 24, issue 1,2 - Journals - IOS Press

Introduction The direct sum of two term rewriting systems (TRSs) [2,8] Ro and R, is confluent if Ro and R, are confluent [10], but the direct sum is not necessarily terminating even if Ro and R, are terminating [ll]; that is, termination is not a "modular" [6] property for general term rewriting systems.

Chapter 9 : CiteSeerX â€” Type Assignment and Termination of Interaction Nets

Bibliographic record and links to related information available from the Library of Congress blog.quintoapp.com: Electronic data is machine generated. May be incomplete or contain other coding.