Open Problems in Rewriting

Nachum Dershowitz*
Department of Computer Science, University of Illinois
1304 West Springfield Avenue, Urbana, IL 61801, U.S.A.
nachum@cs.uiuc.edu

Jean-Pierre Jouannaud
Laboratoire de Recherche en Informatique, Bat. 490
Université de Paris Sud, 91405 Orsay, France
jouannau@lri.lri.fr

Jan Willem Klop†
CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands
Department of Mathematics and Computer Science, Free University
de Boelelaan 1081, 1081 HV Amsterdam, The Netherlands
jwk@cwi.nl

1 Introduction

Interest in the theory and applications of rewriting has been growing rapidly, as evidenced in part by four conference proceedings (including this one) [15, 26, 41, 66]; three workshop proceedings [33, 47, 77]; five special journal issues [5, 88, 24, 40, 67]; more than ten surveys [2, 7, 27, 28, 44, 56, 57, 76, 82, 81]; one edited collection of papers [1]; four monographs [3, 12, 55, 65]; and seven books (four of them still in progress) [8, 9, 35, 54, 60, 75, 84].

To encourage and stimulate continued progress in this area, we have collected (with the help of colleagues) a number of problems that appear to us to be of interest and regarding which we do not know the answer. Questions on rewriting and other equational paradigms have been included; many have not aged sufficiently to be accorded the appellation "open problem". We have limited ourselves to theoretical questions, though there are certainly many additional interesting questions relating to applications and implementations.

Previous lists of questions in this area include one distributed by Leo Marcus and one of us (Dershowitz) at the Sixth International Conference on Automated Deduction (New York, 1982), the questions posed in a set of lecture notes on "Term Rewriting Systems" by one of us (Klop) for a seminar on reduction machines (Ustica, 1985), another list by one of us (Jouannaud) in the *Bulletin of the European Association for Theoretical Computer Science* (Number 31, 1987), and electronic postings to the distribution list (rewriting@crin.crin.fr) maintained by Pierre Lescanne. We use primarily terminology and notation of [27].

*Research supported in part by the National Science Foundation under Grant CCR-9007195.
†Research partially supported by ESPRIT BRA projects 3020: Integration and 3074: Semagraph.

2 Problems

2.1 Rewriting

Problem 1. An important theme that is largely unexplored is definability (or implementability, or interpretability) of rewrite systems in rewrite systems. Which rewrite systems can be directly defined in lambda calculus? Here "directly defined" means that one has to find lambda terms representing the rewrite system operators, such that a rewrite step in the rewrite system translates to a reduction in lambda calculus. For example, Combinatory Logic is directly lambda definable. On the other hand, not every orthogonal rewrite system can be directly defined in lambda calculus. Are there universal rewrite systems, with respect to direct definability? (For alternative notions of definability, see [75].)

Problem 2 (M. Venturini-Zilli [91])**.** The reduction graph of a term is the set of its reducts structured by the reduction relation. These may be very complicated. The following notion of "spectrum" abstracts away from many inessential details of such graphs: If R is a term-rewriting system and t a term in R, let $Spec(t)$, the "spectrum" of t, be the space of finite and infinite reduction sequences starting with t, modulo the equivalence between reduction sequences generated by the following quasi-order: $t = t_1 \to_R t_2 \to_R \cdots \leq t = t'_1 \to_R t'_2 \to_R \cdots$ if for all i there is a j such that $t_i \to_R^* t'_j$. What are the properties of this cpo (complete partial order), in particular for orthogonal (left-linear, non-overlapping) rewrite systems? What influence does the non-erasing property have on the spectrum? (A rewrite system is "non-erasing" if both sides of each rule have exactly the same variables.) The same questions can be asked for the spectrum obtained for orthogonal systems by dividing out the finer notion of "permutation equivalence" due to J.-J. Lévy (see [14, 55, 57]).

Problem 3. A term t is "ground reducible" with respect to a rewrite system R if all its ground (variable-free) instances contain a redex. Ground reducibility is decidable for ordinary rewriting (and finite R) [20, 49, 80]. What is the complexity of this test?

Problem 4. One of the outstanding open problems in typed lambda calculi is the following: Given a term in ordinary untyped lambda calculus, is it decidable whether it can be typed in the second-order $\lambda 2$ calculus? See [7].

Problem 5 (A. Meyer, R. de Vrijer)**.** Do the surjective pairing axioms

$$\begin{aligned} D_1(Dxy) &= x \\ D_2(Dxy) &= y \\ D(D_1x)(D_2x) &= x \end{aligned}$$

conservatively extend $\lambda\beta\eta$-conversion on pure untyped lambda terms? More generally, is surjective pairing *always* conservative, or do there exist lambda theories, or extensions of Combinatory Logic for that matter, for which conservative extension by surjective pairing fails? (Surjective pairing is conservative over the pure $\lambda\beta$-calculus (see [92]). Of course, there are lots of other $\lambda\beta$, indeed $\lambda\beta\eta$, theories where conservative extension holds, simply because the theory consists of the valid equations in some λ model in which surjective pairing functions exist, e.g., D_∞.)

2.2 Normalization

Problem 6 (A. Middeldorp [71])**.** If R and S are two term-rewriting systems with disjoint vocabularies, such that for each of R and S any two convertible normal forms must be identical, then their union $R \cup S$ also enjoys this property [71]. Accordingly, we say that unicity of normal forms (UN) is a "modular" property of term-rewriting systems. "Unicity of normal forms with respect to reduction" ($UN^{\rightarrow}$) is the weaker property that any two normal forms of the same term must be identical. For non-left-linear systems, this property is not modular. The question remains: Is $UN^{\rightarrow}$ a modular property of left-linear term-rewriting systems?

Problem 7 (H. Comon, M. Dauchet)**.** Is it possible to decide whether the set of ground normal forms with respect to a given (finite) term-rewriting system is a regular tree language? See [34, 62].

Problem 8 (A. Middeldorp)**.** Is the decidability of strong sequentiality for orthogonal term rewriting systems NP-complete? See [39, 58].

Problem 9 (A. Middeldorp)**.** Thatte [87] showed that an orthogonal constructor-based rewrite system is left-sequential if and only if it is strongly sequential. Does this equivalence extend to the whole class of orthogonal term-rewriting systems? If not, is left-sequentiality a decidable property of orthogonal systems? See also [58].

Problem 10 (J. R. Kennaway)**.** Let a term-rewriting system (or more generally, a system with bound variables [57]) have the following properties: it is "finitely generated" (has finitely many function symbols and rules), it is "full" (its terms are all that can be formed from the function symbols), and it is Church-Rosser. Does it follow that it has a recursive, one-step, normalizing reduction strategy? (There are counterexamples if any of the three conditions is dropped.) Kennaway [50] showed that for "weakly" orthogonal systems the answer is yes. So, any counterexample must come from the murky world of non-orthogonal systems.

Problem 11 (A. Middeldorp [72])**.** A conditional term-rewriting system has rules of the form $p \Rightarrow l \rightarrow r$, which are only applied to instances of l for which the condition p holds. A "standard" (or "join") conditional system is one in which the condition p is a conjunction of conditions $u \downarrow v$, meaning that u and v have a common reduct (are "joinable"). Is unicity of normal forms (UN) a modular property of standard conditional systems?

2.3 Confluence

Problem 12. What is the complexity of the decision problem for the confluence of ground (variable-free) term-rewriting systems? Decidability was shown in [22, 78]; see also [23].

Problem 13 (J.-J. Lévy)**.** By a lemma of G. Huet [38], left-linear term-rewriting systems are confluent if, for every critical pair $t \approx s$ (where $t = u[r\sigma] \leftarrow u[l\sigma] = g\tau \rightarrow d\tau = s$, for some rules $l \rightarrow r$ and $g \rightarrow d$), we have $t \rightarrow^{\|} s$ (t reduces in one parallel step to s). (The condition $t \rightarrow^{\|} s$ can be relaxed to $t \rightarrow^{\|} r \leftarrow^{\|} s$ for some r when the critical pair is generated from two rules overlapping at the roots; see [89].) What if $s \rightarrow^{\|} t$ for every critical pair $t \approx s$? What if for every $t \approx s$ we have $s \rightarrow^{=} t$? (Here $\rightarrow^{=}$ is the reflexive closure of $\rightarrow$.) What if for every critical pair $t \approx s$, either $s \rightarrow^{=} t$ or $t \rightarrow^{=} s$? In the last case, especially, a confluence proof would be interesting; one would then have confluence after critical-pair completion without regard for termination. If these conditions are insufficient, the counterexamples will have to be (besides left-linear) non-right-linear, non-terminating, and non-orthogonal (have critical pairs). See [57].

Problem 14. Parallel rewriting with orthogonal term-rewriting systems is "subcommutative" (a "strong" form of confluence). Under which interesting syntactic restrictions do conditional rewrite systems enjoy the same property? It is known that orthogonal "normal" conditional rewriting systems (with conditions $u \rightarrow^{!} v$, where v is a ground normal form) are confluent, while "standard" (join) ones are not [13].

Problem 15 (Y. Toyama). Consider the following extension of Combinatory Logic (CL) with constants T (true), F (false), C (conditional):

$$\begin{array}{rcl} Ix & \to & x \\ Kxy & \to & x \\ Sxyz & \to & (xz)(yz) \\ CTxy & \to & x \\ CFxy & \to & y \\ x \leftrightarrow^* y \Rightarrow \quad Czxy & \to & x \end{array}$$

Is this (non-terminating) "semi-equational" (or "natural", as such are called in [31]) conditional rewrite system confluent? Note that if we take the above system plus the rule $x \leftrightarrow^* y \Rightarrow Czxy \to y$, the resulting conditional rewrite system *is* confluent (cf. [57, 93]).

Problem 16 (Y. Toyama). For a "normal" conditional term-rewriting system $R = \{s \to^! t \Rightarrow l \to r\}$, where t must be a ground normal from of s, we can consider the corresponding semi-equational conditional rewrite system $R' = \{s \leftrightarrow^* t \Rightarrow l \to r\}$. Under what conditions does confluence of R' imply confluence of R? In general, this is not the case, as can be seen from the following non-confluent system R (due to A. Middeldorp):

$$\begin{array}{rl} & a \to b \\ & a \to c \\ b \to^! c \Rightarrow & b \to c \end{array}$$

Problem 17 (R. de Vrijer). Is the following semi-equational conditional term rewriting system (a linearization of Combinatory Logic extended with surjective pairing) confluent:

$$\begin{array}{rcl} Ix & \to & x \\ Kxy & \to & x \\ Sxyz & \to & (xz)(yz) \\ D_1(Dxy) & \to & x \\ D_2(Dxy) & \to & y \\ x \leftrightarrow^* y \Rightarrow \quad D(D_1x)(D_2y) & \to & x \\ x \leftrightarrow^* y \Rightarrow \quad D(D_1x)(D_2y) & \to & y \end{array}$$

If yes, does an effective normal form strategy exist for it? See [59, 92].

Problem 18 (J. R. Kennaway, J. W. Klop, M. R. Sleep, F.-J. de Vries). If one wants to consider reductions of transfinite length in the theory of orthogonal term-rewriting systems, one has to be careful. In [51] it is shown that the confluence property "almost" holds for infinite rewriting with orthogonal term-rewriting systems. The only situation in which "infinitary confluence" may fail is when collapsing rules are present. (A rule $t \to s$ is "collapsing" if s is a variable.) Without collapsing rules, or even when only one collapsing rule of the form $f(x) \to x$ is present, infinitary confluence does hold. Now the notion of infinite reduction in [51] is based upon "strong convergence" of infinite sequences of terms in order to define (possibly infinite) limit terms. In related work, Dershowitz, et al. [29] use a more "liberal" notion of convergent sequences (which is referred to in [51] as "Cauchy convergence"). What is unknown (among other questions in this new area) is if this "almost-confluent" result is also valid for the more liberal convergent infinite reduction sequences?

2.4 Termination

Problem 19 (J.-J. Lévy). Can strong normalization (termination) of the typed lambda calculus be proved by a reasonably straightforward mapping from typed terms to a well-founded ordering? Note that the type structure can remain unchanged by β-reduction. The same question arises with polymorphic (second-order) lambda calculus.

Problem 20 (Y. Metivier [70]). What is the best bound on the length of a derivation for a one-rule length-preserving string-rewriting (semi-Thue) system? Is it $O(n^2)$ (n is the size of the initial term) as conjectured in [70], or $O(n^k)$ (k is the size of the rule) as proved there.

Problem 21 (M. Dauchet). Is termination of one linear (left and right) rule decidable? Left linearity alone is not enough for decidability [21].

Problem 22. Devise practical methods for proving termination of (standard) conditional rewriting systems. Part of the difficulty stems from the interdependence of normalization and termination.

Problem 23 (E. A. Cochin [18]). The following system [27], based on the "Battle of Hydra and Hercules" in [52], is terminating, but not provably so in Peano Arithmetic:

$$\begin{array}{rcl} h(z,e(x)) & \rightarrow & h(c(z),d(z,x)) \\ d(z,g(0,0)) & \rightarrow & e(0) \\ d(z,g(x,y)) & \rightarrow & g(e(x),d(z,y)) \\ d(c(z),g(g(x,y),0)) & \rightarrow & g(d(c(z),g(x,y)),d(z,g(x,y))) \\ g(e(x),e(y)) & \rightarrow & e(g(x,y)) \end{array}$$

Transfinite (ϵ_0-) induction is required for a proof of termination. Must any termination *ordering* have the Howard ordinal as its order type, as conjectured in [18]?

Problem 24. The existential fragment of the first-order theory of the "recursive path ordering" (with multiset and lexicographic "status") is decidable when the precedence on function symbols is total [19, 46], but is undecidable for arbitrary formulas. Is the existential fragment decidable for partial precedences?

2.5 Validity

Problem 25 (R. Treinen). Is the theory of multisets (AC) completely axiomatizable? In other words, is it decidable whether a first-order formula containing only equality as predicate symbol is valid in the algebra $\mathcal{T}(\mathcal{F})/AC(F)$? It is known that the Σ_3 fragment is undecidable when there are at least one unary function symbol (besides the AC one) and one constant; the Σ_1 fragment is decidable; the full theory is decidable even when there are no other symbols (besides constants) [90].

Problem 26. Let R be a term-rewriting or combinatory reduction system. Let "decreasing redexes" (DR) be the property that there is a map # from the set of redexes of R, to some well-founded linear order (or ordinal), satisfying:

- if in rewrite step $t \rightarrow_R t'$ redex r in t and redex r' in t' are such that r' is a descendant (or "residual") of r, then $\#r \geq \#r'$;
- if in rewrite step $t \rightarrow t'$ the redex r in t is reduced and r' in t' is "created" (t' is not the descendant of any redex in t), then $\#r > \#r'$.

Calling $\#r$ the "degree" of redex r, created redexes have a degree strictly less than the degree of the creator redex, while the degree of descendant redexes is not increased. The typical example is reduction in simply typed lambda calculus. In [55] it is proved that for orthogonal term-rewriting systems and combinatory reduction systems, decreasing redexes implies termination (strong normalization). Does this implication also hold for non-orthogonal systems? If not, can some decent subclasses be delineated for which the implication does hold?

Problem 27 (P. Lescanne)**.** In [68] an extension of term embedding, called "well-rewrite orderings", was introduced, leading to an extension of the concept of simplification ordering. Can those ideas be extended to form the basis for some new kind of "recursive path ordering"?

Problem 28 (P. Lescanne)**.** Polynomial and exponential interpretations have been used to prove termination. For the former there are some reasonable methods [11, 63] that can help determine if a particular interpretation decreases with each application of a rule. Are there other implementable methods suitable for exponential interpretations?

Problem 29. Any rewrite relation commutes with the strict-subterm relation; hence, the union of the latter with an arbitrary terminating rewrite relation is terminating, and also "fully invariant" (closed under instantiation). Is subterm the maximal relation with these properties? Is "encompassment" ("containment", the combination of subterm and subsumption) the maximal relation which preserves termination (without full invariance)?

Problem 30 (W. Snyder)**.** What are the complexities of the various term ordering decision problems in the literature (see [25])? Determining if a precedence exists that makes two ground terms comparable in the recursive path ordering is NP-complete [61], but an inequality can be decided in $O(n^2)$, using a dynamic programming algorithm. Snyder [85] has shown that the lexicographic path ordering can be done in $O(n \log n)$ in the ground case with a total precedence, but the technique doesn't extend to non-total precedences or to terms with variables.

Problem 31. Is there a decidable uniform word problem for which there is no variant on the rewriting theme (for example, rewriting modulo a congruence with a decidable matching problem, or ordered rewriting) that can decide it—without adding new symbols to the vocabulary? There are decidable theories that cannot be decided with ordinary rewriting (see, for example, [86]); on the other hand, any theory with decidable word problem can be solved by ordered-rewriting with some ordered system for some conservative extension of the theory (that is, with new symbols) [30], or with a two-phased version of rewriting, wherein normal forms of the first system are inputs to the second [10].

Problem 32. Is there a finite term-rewriting system of some kind for free lattices?

Problem 33. Completion modulo associativity and commutativity (AC) [79] is probably the most important case of "extended completion"; the general case of finite congruence classes is treated in [43]. Adding an axiom (Z) for an identity element, however, gives rise to infinite classes. This case was viewed as conditional completion in [6], and solved completely in [45]. The techniques, however, do not carry over to completion with idempotence (I) added; how to handle ACZI-completion effectively is open.

Problem 34. Ordered rewriting computes a given convergent set of rewrite rules for an equational theory E and an ordering $>$ whenever such a set R exists for $>$, provided $>$ can be made total on ground terms. Unfortunately, this is not always possible, even if $>$ is derivability ($\rightarrow_R^+$) in R. Is there a set of inference rules that will always succeed in computing R whenever R exists for $>$?

2.6 Theorem Proving

Problem 35. Huet's proof [37] of the "completeness" of completion is predicated on the assumption that the ordering supplied to completion does not change during the process. Assume that at step i of completion, the ordering used is able to order the current rewriting relation $\rightarrow_{R_i}$, but not necessarily $\rightarrow_{R_k}$ for $k < i$ (since old rules may have been deleted by completion). Is there an example showing that completion is then incomplete (the persisting rules are not confluent)?

Problem 36 (H. Zhang)**.** Since the work of Hsiang [36], several Boolean-ring based methods have been proposed for resolution-like first-order theorem proving. In [48], superposition rules were defined using multiple overlaps (requiring unifications of products of atoms). It is unknown whether single overlaps (requiring only unifications of atoms) are sufficient in these inference rules. Also, it is not known if unifications of maximal atoms (under a given term ordering) suffice. (The same problem for Hsiang's method was solved positively in [73, 94].) In other respects, too, the set of inference rules in [4, 48] may be larger than necessary and the simplification weaker than possible.

Problem 37 (U. Reddy, F. Bronsard)**.** In [17] a rewriting-like mechanism for clausal reasoning called "contextual deduction" was proposed. It specializes "ordered resolution" by using pattern matching in place of unification, only instantiating clauses to match existing clauses. Does contextual deduction always terminate? (In [17] it was taken to be obvious, but that is not clear; see also [74].) It was shown in [17] that the mechanism is complete for refuting ground clauses using a theory that contains all its "strong-ordered" resolvents. Is there a notion of "complete theory" (like containing all strong-ordered resolvents not provable by contextual refutation) for which contextual deduction is complete for refutation of ground clauses?

2.7 Satisfiability

Problem 38 (J. Siekmann [83])**.** Is satisfiability of equations in the theory of distributivity (unification modulo a distributivity axiom) decidable?

Problem 39. Rules are given in [42] for computing dag-solved forms of unification problems in equational theories. The *Merge* rule $x \approx s, x \approx t \Rightarrow x \approx s, s \approx t$ given there assumes that s is not a variable and its size is less than or equal to that of t. Can this condition be improved by replacing it with the condition that the rule *Check** does not apply? (In other words, is *Check** complete for finding cycles when *Merge* is modified as above?)

Problem 40. Fages [32] proved that associative-commutative unification terminates when "variable replacement" is made after each step. Boudet, et al. [16] have proven that it terminates when variable replacement is postponed to the end. Does the same (or similar) set of transformation rules terminate with more flexible control?

Problem 41. The complexity of the theory of finite trees when there are finitely many symbols is known to be PSPACE-hard [69]. Is it in PSPACE? The same question applies to infinite trees.

Problem 42 (H. Comon)**.** Given a first-order formula with equality as the only predicate symbol, can negation be effectively eliminated from an arbitrary formula ϕ when ϕ is equivalent to a positive formula? Equivalently, if ϕ has a finite complete set of unifiers, can they be computed? Special cases were solved in [20, 64].

Problem 43. Design a framework for combining constraint solving algorithms.

Problem 44 (H. Comon)**.** "Syntactic" theories enjoy the property that a (semi-) unification algorithm can be derived from the axioms [42, 53]. This algorithm terminates for some particular cases (for instance, if all variable occurrences in the axioms are at depth at most one, and cycles have no solution) but does not in general. For the case of associativity and commutativity (AC), with a seven-axiom syntactic presentation, the derivation tree obtained by the non-deterministic application of the syntactic unification rules (*Decompose, Mutate, Merge, Coalesce, Check*, Delete*) in [42] can be pruned so as to become finite in most cases. The basic idea is that one unification problem (up to renaming) must appear infinitely times on every infinite branch of the tree (since there are finitely many axioms in the syntactic presentation). Hence, it should be possible to prune or freeze every infinite branch from some point on. The problem is to design such pruning rules so as to compute a finite derivation tree (hence, a finite complete set of unifiers) for every finitary unification problem of a syntactic equational theory.

3 Afterword

This list is by no means exhaustive. Please send any contributions by electronic or ordinary mail to the first author. We will periodically publicize new problems and solutions to old ones.

Acknowledgements

We thank all the individuals mentioned above who contributed questions, F.-J. de Vries for helping prepare this list, and Ron Book, for going out of his way to make it possible.

References

[1] H. Ait-Kaci and M. Nivat, eds. *Resolution of Equations in Algebraic Structures.* Vol. 2: Rewriting Techniques, Academic Press, New York, 1989.

[2] J. Avenhaus and K. Madlener. Term rewriting and equational reasoning. In R. B. Banerji, editor, *Formal Techniques in Artificial Intelligence: A Sourcebook*, pp. 1–41, Elsevier, Amsterdam, 1990.

[3] L. Bachmair. *Canonical Equational Proofs.* Birkhäuser, Boston, 1991. To appear.

[4] L. Bachmair and N. Dershowitz. Inference rules for rewrite-based first-order theorem proving. In *Proceedings of the Second IEEE Symposium on Logic in Computer Science*, pp. 331–337, Ithaca, NY, June 1987.

[5] L. Bachmair and J. Hsiang, eds. *Rewrite Techniques in Theorem Proving (Special Issue).* Vol. 11 (1) of *J. Symbolic Computation*, Academic Press, 1991. To appear.

[6] T. Baird, G. Peterson, and R. Wilkerson. Complete sets of reductions modulo Associativity, Commutativity and Identity. In N. Dershowitz, editor, *Proceedings of the Third International Conference on Rewriting Techniques and Applications*, pp. 29–44, Chapel Hill, NC, Apr. 1989. Vol. 355 of *Lecture Notes in Computer Science*, Springer, Berlin.

[7] H. P. Barendregt. Lambda calculi with types. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, eds., *Handbook of Logic in Computer Science*, Oxford University Press, Oxford, 1991. To appear.

[8] H. P. Barendregt. *The Lambda Calculus, its Syntax and Semantics.* North-Holland, Amsterdam, second edition, 1984.

[9] H. P. Barendregt. *The Typed Lambda Calculus, its Syntax and Semantics.* North-Holland, Amsterdam, 1991.

[10] G. Bauer. n-level rewriting systems. *Theoretical Computer Science*, 40:85–99, 1985.

[11] A. Ben Cherifa and P. Lescanne. Termination of rewriting systems by polynomial interpretations and its implementation. *Science of Computer Programming*, 9(2):137–159, Oct. 1987.

[12] B. Benninghofen, S. Kemmerich, and M. M. Richter. *Systems of Reductions.* Vol. 277 of *Lecture Notes in Computer Science*, Springer, Berlin, 1987.

[13] J. A. Bergstra and J. W. Klop. Conditional rewrite rules: Confluency and termination. *J. of Computer and System Sciences*, 32:323–362, 1986.

[14] G. Berry and J. Lévy. Mimimal and optimal computations of recursive programs. *J. of the Association for Computing Machinery*, 26:148–175, 1979.

[15] R. Book, ed. *Proceedings of the Fourth International Conference on Rewriting Techniques and Applications (Como, Italy)*, Springer, Berlin, Apr. 1991.

[16] A. Boudet, E. Contejean, and H. Devie. A new *ac* unification algorithm with an algorithm for solving diophantine equations. In *Proceedings of the Fifth Annual IEEE Symposium on Logic in Computer Science*, pp. 289–299, Philadelphia, PA, June 1990.

[17] F. Bronsard and U. S. Reddy. Conditional rewriting in Focus. In M. Okada, editor, *Proceedings of the Second International Workshop on Conditional and Typed Rewriting Systems*, Springer, Montreal, Canada, 1991. To appear.

[18] E. A. Cichon. *Bounds on Derivation Lengths from Termination Proofs*. Technical Report CSD-TR-622, Department of Computer Science, University of London, Surrey, England, June 1990.

[19] H. Comon. Solving inequations in term algebras (Preliminary version). In *Proceedings of the Fifth Annual IEEE Symposium on Logic in Computer Science*, pp. 62–69, Philadelphia, PA, June 1990.

[20] H. Comon. *Unification et Disunification: Théorie et Applications*. PhD thesis, l'Institut National Polytechnique de Grenoble, 1988.

[21] M. Dauchet. Simulation of Turing machines by a left-linear rewrite rule. In N. Dershowitz, editor, *Proceedings of the Third International Conference on Rewriting Techniques and Applications*, pp. 109–120, Chapel Hill, NC, Apr. 1989. Vol. 355 of *Lecture Notes in Computer Science*, Springer, Berlin.

[22] M. Dauchet, T. Heuillard, P. Lescanne, and S. Tison. Decidability of the confluence of finite ground term rewriting systems and of other related term rewriting systems. *Information and Computation*, 88(2):187–201, October 1990.

[23] M. Dauchet and S. Tison. The theory of ground rewrite systems is decidable. In *Proceedings of the Fifth IEEE Symposium on Logic in Computer Science*, pp. 242–248, Philadelphia, PA, June 1990.

[24] N. Dershowitz, ed. *Rewriting Techniques and Applications III (Special Issue). J. of Symbolic Computation*, Academic Press, 1992. To appear.

[25] N. Dershowitz. Termination of rewriting. *J. of Symbolic Computation*, 3(1&2):69–115, February/April 1987. Corrigendum: *4*, 3 (December 1987), 409–410.

[26] N. Dershowitz, ed. *Third International Conference on Rewriting Techniques and Applications*, Chapel Hill, NC, April 1989. Vol. 355 of *Lecture Notes in Computer Science*, Springer, Berlin.

[27] N. Dershowitz and J. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science B: Formal Methods and Semantics*, chapter 6, pp. 243–320, North-Holland, Amsterdam, 1990.

[28] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In S. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, Wiley, 1991. In preparation.

[29] N. Dershowitz, S. Kaplan, and D. A. Plaisted. Rewrite, Rewrite, Rewrite, Rewrite, Rewrite,.... *Theoretical Computer Science*, 1991. In press.

[30] N. Dershowitz, L. Marcus, and A. Tarlecki. *Existence, Uniqueness, and Construction of Rewrite Systems*. Technical Report ATR-85(8354)-7, Computer Science Laboratory, The Aerospace Corporation, El Segundo, CA, December 1985.

[31] N. Dershowitz and M. Okada. A rationale for conditional equational programming. *Theoretical Computer Science*, 75:111–138, 1990.

[32] F. Fages. Associative-commutative unification. *J. Symbolic Computation*, 3(3):257–275, June 1987.

[33] *Proceedings of an NSF Workshop on the Rewrite Rule Laboratory*, Schenectady, NY, Sep. 1983. Report 84GEN008, General Electric Research and Development (April 1984).

[34] R. Gilleron. Decision problems for term rewriting systems and recognizable tree languages. In *Proceddings of STACS*, 1991. To appear.

[35] J. R. Hindley and J. P. Seldin. *Introduction to Combinators and λ-Calculus.* Cambridge University Press, 1986.

[36] J. Hsiang. Refutational theorem proving using term-rewriting systems. *Artificial Intelligence*, 25:255–300, March 1985.

[37] G. Huet. A complete proof of correctness of the Knuth-Bendix completion algorithm. *J. Computer and System Sciences*, 23(1):11–21, 1981.

[38] G. Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *J. of the Association for Computing Machinery*, 27(4):797–821, Oct. 1980.

[39] G. Huet and J.-J. Lévy. Computations in orthogonal term rewriting systems. In J.-L. Lassez and G. Plotkin, eds., *Computational Logic: Essays in Honour of Alan Robinson*, MIT Press, Cambridge, MA, to appear.

[40] J.-P. Jouannaud, ed. *Rewriting Techniques and Applications.* Academic Press, London, 1987.

[41] J.-P. Jouannaud, ed. *Rewriting Techniques and Applications (Proceedings, Dijon, France)*, Springer, Berlin, May 1985. Vol. 202 of *Lecture Notes in Computer Science.*

[42] J.-P. Jouannaud and C. Kirchner. Solving equations in abstract algebras: A rule-based survey of unification. In J.-L. Lassez and G. Plotkin, eds., *Computational Logic: Essays in Honor of Alan Robinson*, MIT Press, Cambridge, MA, 1991. To appear.

[43] J.-P. Jouannaud and H. Kirchner. Completion of a set of rules modulo a set of equations. *SIAM J. on Computing*, 15:1155–1194, Nov. 1986.

[44] J.-P. Jouannaud and P. Lescanne. Rewriting systems. *Technology and Science of Informatics*, 6(3):181–199, 1987. French version: "La réécriture", *Technique et Science de l'Informatique* (1986), vol. 5, no. 6, pp. 433-452.

[45] J.-P. Jouannaud and C. Marché. Completion modulo associativity, commutativity and identity. In A. Miola, editor, *Proceedings of DISCO*, Capri, Italy, Apr. 1990. Vol. 429 in *Lecture Notes in Computer Science*, Springer, Berlin.

[46] J.-P. Jouannaud and M. Okada. Satisfiability of systems of ordinal notations enjoying the subterm property is decidable. 1991. Submitted.

[47] S. Kaplan and J.-P. Jouannaud, eds. *Conditional Term Rewriting Systems (Proceedings, Orsay, France, July 1987)*, Springer, Berlin, 1988. Vol. 308 of *Lecture Notes in Computer Science.*

[48] D. Kapur and P. Narendran. An equational approach to theorem proving in first-order predicate calculus. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pp. 1146–1153, Los Angeles, CA, Aug. 1985.

[49] D. Kapur, P. Narendran, and H. Zhang. On sufficient completeness and related properties of term rewriting systems. *Acta Informatica*, 24(4):395–415, Aug. 1987.

[50] J. R. Kennaway. Sequential evaluation strategies for parallel-or and related reduction systems. *Annals of Pure and Applied Logic*, 43:31–56, 1989.

[51] J. R. Kennaway, J. W. Klop, M. R. Sleep, and F. J. de Vries. Transfinite reductions in orthogonal term rewriting systems (Extended abstract). In R. Book, editor, *Proceedings of the Fourth International Conference on Rewriting Techniques and Applications*, Como, Italy, Apr. 1991. In *Lecture Notes in Computer Science*, Springer, Berlin.

[52] L. Kirby and J. Paris. Accessible independence results for Peano arithmetic. *Bulletin London Mathematical Society*, 14:285–293, 1982.

[53] C. Kirchner. Computing unification algorithms. In *Proceedings of the First IEEE Symposium on Logic in Computer Science*, pp. 206–216, Cambridge, Massachussets, June 1986.

[54] C. Kirchner and H. Kirchner. *Rewriting: Theory and Applications.* North-Holland, 1991. In preparation.

[55] J. W. Klop. *Combinatory reduction systems.* Vol. 127 of *Mathematical Centre Tracts*, Mathematisch Centrum, 1980.

[56] J. W. Klop. Term rewriting systems: A tutorial. *Bulletin of the European Association for Theoretical Computer Science*, 32:143–183, June 1987.

[57] J. W. Klop. Term rewriting systems. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, eds., *Handbook of Logic in Computer Science*, Oxford University Press, Oxford, 1991. To appear.

[58] J. W. Klop and A. Middeldorp. *Sequentiality in Orthogonal Term Rewriting Systems.* Report CS-R8932, Centre for Mathematics and Computer Science, Amsterdam, 1989.

[59] J. W. Klop and R. C. de Vrijer. Unique normal forms for lambda calculus with surjective pairing. *Information and Computation*, 80:97–113, 1989.

[60] J. W. Klop and R. C. de Vrijer. *Term Rewriting Systems.* Cambridge University Press, Cambridge, 1991. In preparation.

[61] M. S. Krishnamoorthy and P. Narendran. On recursive path ordering. *Theoretical Computer Science*, 40:323–328, 1985.

[62] G. Kucherov. On relationship between term rewriting systems and regular tree languages. In R. Book, editor, *Proceedings of the Fourth International Conference on Rewriting Techniques and Applications*, Como, Italy, Apr. 1991. In *Lecture Notes in Computer Science*, Springer, Berlin.

[63] D. S. Lankford. *On Proving Term Rewriting Systems are Noetherian.* Memo MTP-3, Mathematics Department, Louisiana Tech. University, Ruston, LA, May 1979. Revised October 1979.

[64] J. Lassez and K. G. Marriott. Explicit representation of terms defined by counter examples. *J. Automated Reasoning*, 3(3):1–17, Sep. 1987.

[65] P. Le Chenadec. *Canonical Forms in Finitely Presented Algebras.* Pitman-Wiley, London, 1985.

[66] P. Lescanne, ed. *Rewriting Techniques and Applications (Proceedings, Bordeaux, France)*, Springer, Berlin, May 1987. Vol. 256 of *Lecture Notes in Computer Science.*

[67] P. Lescanne, ed. *Rewriting Techniques and Applications II (Special Issue).* Vol. 67 (2&3) of *Theoretical Computer Science*, North-Holland, 1989.

[68] P. Lescanne. Well rewrite orderings. In J. Mitchell, editor, *Proceedings of the Fifth IEEE Symposium on Logic in Computer Science*, pp. 239–256, Philadelphia, PA, 1990.

[69] M. J. Maher. Complete axiomatizations of the algebras of the finite, rational and infinite trees. In *Proceedings of the Third IEEE Symposium on Logic in Computer Science*, pp. 348–357, Computer Society Press, Edinburgh, UK, July 1988.

[70] Y. Métivier. Calcul de longueurs de chaînes de réécriture dans le monoïde libre. *Theoretical Computer Science*, 35(1):71–87, Jan. 1985.

[71] A. Middeldorp. Modular aspects of properties of term rewriting systems related to normal forms. In N. Dershowitz, editor, *Proceedings of the Third International Conference on Rewriting Techniques and Applications*, pp. 263–277, Chapel Hill, NC, Apr. 1989. Vol. 355 of *Lecture Notes in Computer Science*, Springer, Berlin.

[72] A. Middeldorp. *Modular Properties of Term Rewriting Systems.* PhD thesis, Vrije Universiteit, Amsterdam, 1990.

[73] J. Müller and R. Socher-Ambrosius. *Topics in completion theorem proving.* SEKI-Report SR-88-13, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern, West Germany, 1988. To appear in *J. of Symbolic Computation.*

[74] R. Nieuwenhuis and F. Orejas. Clausal rewriting. In S. Kaplan and M. Okada, eds., *Extended Abstracts of the Second International Workshop on Conditional and Typed Rewriting Systems*, pp. 81–88, Concordia University, Montreal, Canada, June 1990. Revised version to appear in *Lecture Notes in Computer Science*, Springer, Berlin.

[75] M. J. O'Donnell. *Equational Logic as a Programming Language.* MIT Press, Cambridge, MA, 1985.

[76] M. J. O'Donnell. Programming with equations. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, eds., *Handbook of Logic in Artificial Intelligence and Logic Programming*, Oxford University Press, Oxford, 1991. To appear.

[77] M. Okada, ed. *Proceedings of the Second International Workshop on Conditional and Typed Rewriting Systems (Montreal, Canada)*, 1991. *Lecture Notes in Computer Science*, Springer, Berlin.

[78] M. Oyamaguchi. The Church-Rosser property for ground term rewriting systems is decidable. *Theoretical Computer Science*, 49(1), 1987.

[79] G. E. Peterson and M. E. Stickel. Complete sets of reductions for some equational theories. *J. of the Association for Computing Machinery*, 28(2):233–264, Apr. 1981.

[80] D. A. Plaisted. Semantic confluence tests and completion methods. *Information and Control*, 65(2/3):182–215, May/June 1985.

[81] D. A. Plaisted. Term rewriting systems. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, eds., *Handbook of Logic in Artificial Intelligence and Logic Programming*, Oxford University Press, Oxford, 1991. To appear.

[82] D. A. Plaisted. Equational reasoning and term rewriting systems. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, eds., *Handbook of Logic in Artificial Intelligence and Logic Programming*, Oxford University Press, Oxford, 1991. To appear.

[83] J. Siekmann. Universal unification. In R. E. Shostak, editor, *Proceedings of the Seventh International Conference on Automated Deduction*, pp. 1–42, Napa, CA, May 1984. Vol. 170 of *Lecture Notes in Computer Science*, Springer, Berlin.

[84] M. R. Sleep. *An Introduction to Rewriting.* John Wiley, Chichester, England, 1992. In preparation.

[85] W. Snyder. *(Computing the Lexicographic Path Ordering).* Technical Report, Boston University, Boston, MA, 1990.

[86] C. Squier. Word problems and a homological finiteness condition for monoids. *J. of Pure and Applied Algebra*, 1991. To appear.

[87] S. Thatte. A refinement of strong sequentiality for term rewriting with constructors. *Information and Computation*, 72:46–65, 1987.

[88] M. Thomas, ed. *Term Rewriting (Special Issue).* Vol. 34 (1) of *Computer J.*, Feb. 1991.

[89] Y. Toyama. Commutativity of term rewriting systems. In K. Fuchi and L. Kott, eds., *Programming of Future Generation Computers II*, pp. 393–407, North-Holland, 1988.

[90] R. Treinen. A new method for undecidability proofs of first order theories. *LNCS* 472, 48–62.

[91] M. Venturini-Zilli. Reduction graphs in the Lambda Calculus. *Theoretical Computer Science*, 29:251–275, 1984.

[92] R. C. de Vrijer. Extending the lambda calculus with surjective pairing is conservative. In *Proceedings of the Fourth IEEE Symposium on Logic in Computer Science*, pp. 204–215, 1989.

[93] R. C. de Vrijer. *Unique normal forms for Combinatory Logic with Parallel Conditional, a case study in conditional rewriting.* Technical Report, Free University, Amsterdam, 1990.

[94] H. Zhang. *A New Strategy for the Boolean Ring Based Approach to First Order Theorem Proving.* Technical Report, Department of Computer Science, The University of Iowa, 1991.