The RTA List of Open Problems

Nachum Dershowitz

January 22, 2014

The RTA list of open problems summarizes open problems in the field of the International Conference on Rewriting Techniques and Applications (RTA). For the RTA 2002 conference, the topics of RTA were given as

- Applications: case studies; rule-based programming; symbolic and algebraic computation; theorem proving; functional and logic programming; proof checking.
- Foundations: matching and unification; completion techniques; strategies; constraint solving; explicit substitutions.
- Frameworks: string, term, and graph rewriting; lambda-calculus and higher-order rewriting; conditional rewriting; proof nets; constrained rewriting and deduction; categorical and infinitary rewriting.
- Implementation: compilation techniques; parallel execution; rewriting tools.
- Semantics: equational logic; rewriting logic.

The RTA list of open problems was created in 1991 by Nachum Dershowitz, Jean-Pierre Jouannaud and Jan Willem Klop on occasion of the RTA'91 conference. Updated lists have since been published at RTA'93, RTA'95 and RTA'98.

Since October 1997 the list of open problems is maintained as a web service. This effort is, at the moment, led by Nachum Dershowitz. The pages that you can access reflect an update of the lists presented to the RTA conferences. Currently, the list comprises 104 problems, 33 of which are solved. Note that problems marked as *solved* may still leave open the question of extensions or special cases.

Problem #1

Originator: Jan Willem Klop
Date: April 1991

Summary: Which rewrite systems can be directly defined in lambda calculus?

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 [O'D85].)

Remark

Some progress has been made in [BB92].

Problem #2

Originator: M. Venturini-Zilli [VZ84]
Date: April 1991

Summary: Investigate the properties of spectri *for special classes of rewrite systems.*

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 [BL79][Klo80][Klo92]).

Problem #3 (Solved !)

Originator: Deepak Kapur
Date: April 1991

Summary: What is the complexity of deciding ground-reducibility?

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) [Com88, KNZ87, Pla85], but n^{n^n} is the best known upper bound in general, $2^{dn \log n}$ and $2^{cn/\log n}$ are the best upper and lower bounds, respectively, for left-linear systems, where n is the size of the system R and c, d are constants [KNZ87]. Can these bounds be improved?

Remark

Ground-reducibility is EXPTIME-complete [CJ97a, CJ03].

Problem #4 (Solved !)

Originator:
Date: April 1991

Summary: Is it decidable whether a term is is typable in the second-order $\lambda 2$ calculus?

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 [Bar91][Hue90].

Remark

This question has been solved in the negative. In [Wel94] J.B. Wells proves that given a closed, type-free lambda term, the question whether it is typable in second-order $\lambda 2$ calculus, is undecidable. Moreover, given a closed type-free lambda term M and a type σ, then it is also undecidable in second-order $\lambda 2$ calculus whether M has type σ.

Problem #5

Originator: Albert Meyer, Roel C. de Vrijer
Date: April 1991

> *Summary: Does surjective pairing conservatively extend $\lambda\beta\eta$-conversion?*

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 [dV89]). 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_∞.

Comment sent by Kristian Støvring

Date: Tue, 22 Nov 2005 00:18:13 +0100

The problem has been solved with a positive answer [Stø05, Stø06]. The generalization to arbitrary lambda theories remains open.

Problem #6 (Solved !)

Originator: Aart Middeldorp [Mid89]
Date: April 1991

> *Summary: Is* unicity of normal forms with respect to reduction *a modular property of left-linear term-rewriting systems?*

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 [Mid89]. 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" ($\mathrm{UN}^{\rightarrow}$) 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?

Remark

A positive solution is given in [Mar94].

Problem #7 (Solved !)

Originator: Hubert Comon, Max Dauchet
Date: April 1991

> *Summary: 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?*

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 [Gil91][Kuc91].

Remark

This has been answered in the affirmative [VG92, KT93, KT95, HH93].

Problem #8

Originator: Aart Middeldorp
Date: April 1991

Summary: Is the decidability of strong sequentiality for orthogonal term rewriting systems NP-complete?

Is the decidability of strong sequentiality for orthogonal term rewriting systems NP-complete? See [HL91a, KM91].

Problem #9

Originator: Aart Middeldorp
Date: April 1991

Summary: Is left-sequentiality a decidable property of orthogonal systems?

Thatte [Tha87] 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 [KM91].

Problem #10

Originator: J. R. Kennaway
Date: April 1991

Summary: Has any full, finitely-generated and Church-Rosser term-rewriting system (or system with bound variables) a recursive, one-step, normalizing reduction strategy?

Let a term-rewriting system (or more generally, a system with bound variables [Klo92]) 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 [Ken89] showed that for “weakly” orthogonal systems the answer is yes. So, any counterexample must come from the murky world of non-orthogonal systems. See also [AM96].

Problem #11 (Solved !)

Originator: Aart Middeldorp [Mid90]
Date: April 1991

> *Summary: Is unicity of normal forms a modular property of standard conditional systems?*

A conditional term-rewriting system has rules of the form $p \Rightarrow l \to r$, 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? See also [Mid93].

Remark

This has been answered in the negative by giving a counterexample [Sch02].

Problem #12 (Solved !)

Originator: Wayne Snyder
Date: April 1991

> *Summary: What is the complexity of the decision problem for the confluence of ground term-rewriting systems?*

What is the complexity of the decision problem for the confluence of ground (i.e., variable-free) term-rewriting systems? Decidability was shown in [DHLT90][Oya87]; see also [DT90].

Remark

The problem is PTIME-complete, as has been shown independently by [CGN01] and [Tiw02].

Problem #13

Originator: Jean-Jacques Lévy
Date: April 1991

Summary: Give decidable criteria for left-linear rewriting systems to be Church-Rosser.

By a lemma of Gérard Huet [Hue80], left-linear term-rewriting systems are confluent if, for every critical pair $t \approx s$ (where $t = u[r\sigma] \leftarrow u[l\sigma] = g\tau \to d\tau = s$, for some rules $l \to r$ and $g \to d$), we have $t \to^{\|} s$ (t reduces in one parallel step to s). (The condition $t \to^{\|} s$ can be relaxed to $t \to^{\|} r \leftarrow^{\|} s$ for some r when the critical pair is generated from two rules overlapping at the roots; see [Toy88].) What if $s \to^{\|} t$ for every critical pair $t \approx s$? What if for every $t \approx s$ we have $s \to^{=} t$? (Here $\to^{=}$ is the reflexive closure of $\to$.) What if for every critical pair $t \approx s$, either $s \to^{=} t$ or $t \to^{=} s$? 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 [Klo92].

Remark

Significant progress is reported in [OO97].

A new criterion based on so-called *simultaneous critical pairs* has been presented in [Oku98].

The history of the problem and the attempts to solve it are told in [Der05].

Problem #14

Originator: Jean-Pierre Jouannaud
Date: April 1991

Summary: Which conditional rewrite systems are subcommutative?

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 $u \to^! v$, where v is a ground normal form) are confluent, while "standard" (join) ones are not [BK86].

Problem #15

Originator: Yoshihito Toyama
Date: April 1991

Summary: Is the extension of Combinatory Logic by Boolean constants confluent?

Consider the following extension of Combinatory Logic with constants T (true), F (false), C (conditional):

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

Is this (non-terminating) "semi-equational" (or "natural", as such are called in [DO90]) 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. [Klo92][de 90]).

Problem #16

Originator: Yoshihito Toyama
Date: April 1991

Summary: Under what conditions does confluence of a normal semi-equational conditional term rewriting system imply confluence of the associated oriented system?

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 $R_{se} = \{s \leftrightarrow^* t \Rightarrow l \to r\}$. Under what conditions does confluence of R_{se} imply confluence of R? In general, this is not the case, as can be seen from the following non-confluent system R (due to Aart Middeldorp):

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

Remark

Solutions have been provided by [YASM00]. They show that confluence of R follows from confluence of R_{se} if any of the two following conditions is satisfied:

- R_{se} is semi-decreasing
- R_{se} is level-confluent

See [YASM00] for definitions of these properties.

Problem #17

Originator: Roel C. de Vrijer
Date: April 1991

Summary: Is a certain conditional rewrite system, which is a linearization of Combinatory Logic extended with surjective pairing, confluent?

Is the following semi-equational conditional term rewriting system (a linearization of Combinatory Logic extended with surjective pairing) confluent:

$$\begin{array}{rcl} Ix & \to & x \\ Kxy & \to & x \\ Sxyz & \to & (xz)(yz) \\ D_1(Dxy) & \to & x \end{array}$$

$$
\begin{array}{rcl}
D_2(Dxy) & \to & y \\
x \leftrightarrow^* y \Rightarrow D(D_1x)(D_2y) & \to & x \\
x \leftrightarrow^* y \Rightarrow D(D_1x)(D_2y) & \to & y
\end{array}
$$

If yes, does an effective normal form strategy exist for it? See [Kd89][dV89].

Problem #18 (Solved !)

Originator: Richard Kennaway, Jan Willem Klop, Ronan Sleep, Fer-Jan de Vries [KKSd91]
Date: April 1991

Summary: Does "almost-confluence" hold for convergent infinite reduction sequences?

If one wants to consider reductions of transfinite length in the theory of orthogonal term-rewriting systems, one has to be careful. In [KKSd90][KKSd91] 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 [KKSd91] is based upon "strong convergence" of infinite sequences of terms in order to define (possibly infinite) limit terms. In related work, Dershowitz, et al. [DKP91] use a more "liberal" notion of convergent sequences (which is referred to in [KKSd91] 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?

Remark

This has been answered to the negative by [Sim04]. However, the counter-example given there is quite peculiar: The rewrite system is not right-linear, the right-hand sides of the rules are not in normal form, and there is no bound on the depths of the left-hand sides of the rules (the rewrite system has an infinite number of rules). Thus, the question remains under which reasonable conditions (Cauchy-)convergent and orthogonal rewrite systems are almost-confluent.

Problem #19

Originator: Jean-Jacques Lévy
Date: April 1991

Summary: Can strong normalization of the typed lambda calculus be proved by a reasonably straightforward mapping from typed terms to a well-founded ordering?

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 (Solved !)

Originator: Yves Métivier [Mét85]
Date: April 1991

Summary: What is the best bound on the length of a derivation for a one-rule length-preserving string-rewriting system?

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 [Mét85], or $O(n^k)$ (k is the size of the rule) as proved there.

Remark

The upper bound is $n^2/4$ where n denotes the length of the initiating string [Ber94]. The bound is reached by the derivation from $b^{n/2}a^{n/2}$ for the string rewriting system $\{ba \to ab\}$.

More about the history of this problem in the context of the question of one-rule termination can be found in [Der05].

Problem #21

Originator: Max Dauchet
Date: April 1991

Summary: Is termination of one linear rule decidable?

Is termination of one linear (left and right) rule decidable? Left linearity alone is not enough for decidability [Dau89].

Remark

A less ambitious, long-standing open problem (mentioned in [DJ90]) is decidability for *one* (length-increasing) monadic (string, semi-Thue) rule. Termination is undecidable for non-length-increasing monadic systems of rules [Car91]. For one monadic rule, confluence is decidable [Kur90][Wra90]. What about confluence of one non-monadic rule?

Partial results for string rewrite rules have been obtained in [Ges03].

The history of the problem and the attempts to solve it are told in [Der05].

Problem #22 (Solved !)

Originator: Nachum Dershowitz
Date: April 1991

Summary: Devise practical methods for proving termination of conditional rewriting systems.

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

Remark

Termination and decreasingness of CTRSs can be proved by transforming CTRSs into unconditional TRSs such that termination of the TRS is sufficient for decreasingness of the CTRS. Several variants of this transformation are studied in [BK86, DP86, GA01, GM87, Siv89, Mar96b, Ohl01]. Termination of the TRSs resulting from this transformation can often be proved automatically using dependency pairs [AG00, GA01]. The transformation (together with the dependency pair approach) is implemented in the tools TALP [OCM00] and AProVE [GTSKF04]. Both tools use this transformation in order to show termination of logic programs, but AProVE can also prove termination and decreasingness of CTRSs in this way. A different approach for termination proofs of CTRSs with the general path order [DH95] is described in [Hoo96].

Problem #23 (Solved !)

Originator: E. A. Cichon [Cic90]
Date: April 1991

> *Summary: Must any termination* ordering *used for proving termination of the* Battle of Hydra and Hercules-*system have the Howard ordinal as its order type?*

The following system [DJ90], based on the "Battle of Hydra and Hercules" in [KP82], is terminating, but not provably so in Peano Arithmetic:

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

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 [Cic90]?

Remark

If the notion of termination ordering is formalized by using ordinal notations with variables, then a termination proof using such orderings yields a slow growing bound on the lengths of derivations. If the order type is less than the Howard-Bachmann ordinal then, by Girard's Hierarchy Theorem, the derivation lengths are provably total in Peano Arithmetic. Hence a termination proof for this particular rewrite system for the Hydra game cannot be given by such an ordering [Andreas Weiermann, personal communication].

Remark

This has been answered to the negative by Georg Moser [Mos09], by giving a reduction order that is compatible with the above rewrite system, and whose order type is at most ϵ_0 (the proof theoretic ordinal of Peano arithmetic).

Problem #24

Originator: Jean-Pierre Jouannaud
Date: April 1991

> *Summary: Is satisfiability of lpo or rpo ordering constraints decidable in case of non-total precedences?*

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 [Com90, JO91b], but is undecidable for arbitrary formulas. Is the existential fragment decidable for partial precedences?

Remark

The Σ_4 ($\exists^*\forall^*\exists^*\forall^*$) fragment is undecidable, in general [Tre92]. The positive existential fragment for the empty precedence (that is, for homeomorphic tree embedding) is decidable [BC93]. One might also ask whether the first-order theory of *total* recursive path orderings is decidable. Related results include the following: The existential fragment of the subterm ordering is decidable, but its Σ_2 ($\exists^*\forall^*$) fragment is not [Ven87]. The first-order theory of encompassment (the instance-of-subterm relation) is decidable [CCD93]. The satisfiability problem for the existential fragment in the total case is NP-complete [Nie93b].

Though the first-order theory of encompassment is decidable [CCD93], the first-order (Σ_2) theory of the recursive (lexicographic status) path ordering, assuming certain simple conditions on the precedence, is not [CT97].

Problem #25 (Solved !)

Originator: Ralf Treinen [Tre90]
Date: April 1991

> *Summary: Is the Σ_2-fragment of the first-order theory of ground terms modulo AC decidable?*

Consider a finite set of function symbols containing at least one AC (associative-commutative) function symbol. Let T be the corresponding set of terms (modulo the AC properties). It is known from [Tre92] that the first-order theory (Σ_3 fragment) of T is undecidable when F contains at least a non-constant symbol (besides the AC symbol). When F only contains an AC symbol and constants, the theory reduces to Presburger's arithmetic and is hence decidable. On the other hand the Σ_1 fragment of T is always decidable [Com93a]. The decidability of the Σ_2 fragment of the theory of T remains open.

Remark

Even more, the solvability of the following important particular case is open: given $t, t_1, \ldots, t_n \in T(F, X)$, is there an instance of t which is not an instance of $t_1, \ldots, t_n$ modulo the AC axioms? This is known as *complement problems* modulo AC.

Several special cases have been solved [Fer93][LM93], and in unpublished work in progress.

The undecidability of the Σ_2-fragment of the first-order theory of ground terms modulo AC has been shown by [Mar99].

Problem #26

Originator: Jan Willem Klop
Date: April 1991

> *Summary: Is it true for non-orthogonal systems that decreasing redexes implies termination? If not, can some decent subclasses be delineated for which the implication does hold?*

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 $t \to t'$ 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 [Klo80] 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?

Comment sent by Vincent van Oostrom

Date: Thu Mar 4 11:28:20 MET 1999

Contrary to what was claimed in [Klo80] (and in the statement of problem 26), decreasingness does not imply termination for orthogonal combinatory reduction systems. A counterexample can be found in Section 6.2.2 of the PhD thesis [Mel96], pp. 158-160.

The main application of the lemma, termination of rewrite systems having 'bounded production-depth', was recovered there ([Mel96], Theorem 6.5) in an axiomatic setting. For the case of higher-order rewriting this was shown in [Oos97b].

Problem #27

Originator: Pierre Lescanne
Date: April 1991

Summary: How can the notion of well-rewrite-ordering *be used to as the basis for some new kind of "recursive path ordering"?*

In [Les90] an extension of term embedding, called "well-rewrite orderings", was introduced, leading to an extension of the concept of simplification ordering. How can those ideas best be extended to form the basis for some new kind of "recursive path ordering"?

Remark

Progress in this direction has been reported in [Wei92].

Problem #28

Originator: Pierre Lescanne
Date: April 1991

Summary: Develop effective methods to decide whether a system decreases with respect to some exponential interpretation*.*

Polynomial and exponential interpretations have been used to prove termination. For the former there are some reasonable methods [CL87, Lan79] that can help determine if a particular interpretation decreases with each application of a rule. Are there other implementable methods suitable for exponential interpretations?

Remark

Some work on this problem has been reported in [Les92].

Problem #29

Originator: Jean-Pierre Jouannaud
Date: April 1991

Summary: Which is the coarsest relation such that its union with any rewrite relation preserves termination?

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). Which is the coarsest (maximal) relation with these properties? (A relation R is said to be *coarser* than a relation S if xSy implies xRy).

The answer is not "the subterm relation". Is *encompassment* ("containment", the combination of subterm and subsumption) the coarsest relation which preserves termination (without full invariance)?

Remark

The coarsest relation we know of which could answer the first question is the variant of subterm that allows multiple occurrences of variables to be renamed apart.

Problem #30

Originator: Wayne Snyder
Date: April 1991

> *Summary: What are the complexities of various term ordering decision problems?*

What are the complexities of the various term ordering decision problems in the literature (see [Der87])? Determining if a precedence exists that makes two ground terms comparable in the recursive path ordering is NP-complete [KN85b], but an inequality can be decided in $O(n^2)$, using a dynamic programming algorithm. Snyder [Sny91] 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

Originator: Albert Meyer
Date: April 1991

> *Summary: Is there a decidable uniform word problem for which there is no variant on the rewriting theme that can decide it—without adding new symbols?*

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, [Squ87]); 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) [DMT85], or with a two-phased version of rewriting, wherein normal forms of the first system are inputs to the second [Bau85].

Problem #32

Originator: John Pedersen
Date: April 1991

Summary: Is there a finite term-rewriting system of some kind for free lattices?

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

Remark

As mentioned in a remark to Problem #77, it has been shown in [Fre93] that there is no finite, normal form, associative-commutative term-rewriting system for lattices.

Comment sent by Jordi Levy

Date: Fri Apr 17 20:20:41 MET DST 1998

There are bi-rewriting systems for free lattices and distributive free lattices [LA96]. Bi-rewriting systems are used to automatize deduction in theories with monotonic order relations. They are composed of a pair of rewriting systems. One is used for rewriting terms into smaller terms, and the other for rewriting terms into bigger terms.

Problem #33 (Solved !)

Originator: Jean-Pierre Jouannaud
Date: April 1991

Summary: How can completion modulo ACUI be made effective?

Completion modulo associativity and commutativity (AC) [PS81] is probably the most important case of *extended completion*; the general case of finite congruence classes is treated in [JK86]. Adding an axiom (U) for an identity element ($x + 0 = x$) gives rise to infinite classes. This case was viewed as conditional completion in [BPW89], and solved completely in [JM90]. The techniques, however, do not carry over to completion with idempotence (I) added ($x + x = x$). How to handle ACUI-completion effectively is open.

Remark

Normalized Rewriting as introduced by Claude Marché in [Mar96a] is the right way of handling axiom systems like ACUI.

Problem #34 (Solved !)

Originator: Nachum Dershowitz, Jean-Pierre Jouannaud
Date: April 1991

> *Summary: Is there a set of inference rules that always succeeds in computing a convergent set of rewrite rules for a given set of equations and an ordering, provided that it exists?*

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 $>$?

Remark

A proposal appears in [Dev91]; more work is called for.

A positive answer has been given in [BGNR99].

Problem #35 (Solved !)

Originator: Nachum Dershowitz, L. Marcus
Date: April 1991

> *Summary: Can completion be incomplete when the ordering is changed en route?*

Huet's proof [Hue81] 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)?

Remark

The answer is yes, even when completion terminates with finitely many rules [SK94].

Problem #36

Originator: H. Zhang
Date: April 1991

> *Summary: Find more restrictive strategies in Boolean-ring based methods for resolution-like first-order theorem proving.*

Since the work of Hsiang [Hsi85], several Boolean-ring based methods have been proposed for resolution-like first-order theorem proving. In [KN85a], 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 [MSA88][Zha91].) In other respects, too, the set of inference rules in [BD87, KN85a] may be larger than necessary and the simplification weaker than possible.

Problem #37

Originator: U. Reddy, F. Bronsard
Date:

> *Summary: Is there a notion of "complete theory" for which* contextual deduction *is complete for refutation of ground clauses?*

In [BR91] 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 [BR91] it was taken to be obvious, but that is not clear; see also [NO90].) It was shown in [BR91] 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?

Contextual deduction as defined in [BR91] does not terminate. Bronsard and Reddy have gone on to solve this [BR93] by using a more restricted, decidable mechanism. A completeness proof, incorporating equational inference with complete systems, is given in [Bro95].

Problem #38 (Solved !)

Originator: Jörg Siekmann
Date: April 1991

> *Summary: Is unification modulo* distributivity *decidable?*

Is satisfiability of equations in the theory of distributivity (unification modulo modulo one right- and one left-distributivity axiom) decidable? (With just one of these, the problem had already been solved in [TA87].) A partial positive solution is given in [Con93a], based on a striking result on the structure of certain proofs modulo distributivity. Although many more cases are described in [Con92][Con93b], the general case remains open.

Remark

This theory is decidable [SS94a][SS97].

Problem #39

Originator: Jean-Pierre Jouannaud
Date: April 1991

> *Summary: Can the application condition on the* Merge *rule in the computation of dag-solved forms of unification problems be improved?*

Rules are given in [JK91a] 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?)

Remark

The problem has been solved by Hubert Comon [Com93b] using an extended *Check* rule (requiring a congruence closure step). The original question—for whatever it may be worth—stands.

Problem #40

Originator: Participants at Unif Val d'Ajol
Date: April 1991

> *Summary: Does AC unification terminate under more flexible control?*

Fages [Fag87] proved that associative-commutative unification terminates when "variable replacement" is made after each step. Boudet, et al. [BCD90] 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 (Solved !)

Originator: Participants at Unif Val d'Ajol
Date: April 1991

> *Summary: What is the complexity of the first-order theory of trees?*

The complexity of the theory of finite trees when there are finitely many symbols is known to be PSPACE-hard [Mah88]. Is it in PSPACE? The same question applies to infinite trees.

Remark

The problem is non-elementary [Vor96].

Problem #42 (Solved !)

Originator: Hubert Comon
Date: April 1991

Summary: Can negations be effectivly eliminated from first-order formulas over trees, where equality is the only predicate?

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 [Com88, LM87].

Remark

A positive solution is given in [Taj93].

Problem #43

Originator: Jean-Pierre Jouannaud
Date: April 1991

Summary: Design a framework for combining constraint solving algorithms.

Design a framework for combining constraint solving algorithms.

Remark

Some particular cases have been attacked: In [BS92] it was shown how decision procedures for solvability of unification problems can be combined. In [BS93] a similar technique is applied to (unquantified) systems of equations and disequations. In [Rin92] the combination of unification algorithms is extended to the case where alphabets share constants. In related work [Bou92], unification is performed in the combination of an equational theory and membership constraints.

Some further progress is in [Rin92].

The combination approach of [BS92] has been extended in [BS95a] to constraints involving predicate symbols other that equality, and [BS95b] in turn extends this approach to constraint-solving over solution domains that are not free structures. These results are presented in a uniform framework by [BS98].

The work of [Rin92] has been extended to the case of “shared constructors” by [DKR94].

Comment sent by Miki Hermann

Date: Mon Apr 27 12:05:20 MET DST 1998

Unification algorithms (and therefore also constraint solvers) cannot be combined in polynomial time, as proved by Hermann and Kolaitis in [HK96].

Problem #44

Originator: Hubert Comon
Date: April 1991

> *Summary: How to compute finite and complete sets of unifiers for any finitary unification problem of a syntactic equational theory.*

“Syntactic” theories enjoy the property that a (semi) unification algorithm can be derived from the axioms [JK91b][Kir86]. 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 [JK91b] 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.

Remark

The core of this problem has been solved [BC94].

Problem #45

Originator: M. Venturini-Zilli
Date: December 1991

> *Summary: Which ordinals correspond to reduction graphs in the λ-calculus?*

Some reduction graphs in λ-calculus [VZ84] are isomorphic to ordinals. For example, the reduction graph of $(\lambda x.y)((\lambda z.zzz)(\lambda z.zzz))$ is isomorphic to $\omega + 1$. Which ordinals appear in this way as reduction graphs? It is known that all ordinals less than ϵ_0 can be so represented.

Problem #46

Originator: Deepak Kapur
Date: December 1991

> *Summary: For which equational theories is ground reducibility of extended rewriting decidable?*

Ground reducibility of extended rewrite systems, modulo congruence, like associativity and commutativity (AC), is undecidable [KNZ87]. For left-linear AC systems, on the other hand, it is decidable [JK89]. What can be said more generally about restrictions on extended rewriting that give decidability? This problem is related to Problem #25.

Remark

Progress has been made in [KR94], where it is proven that ground reducibility remains undecidable when a single non-constant function symbol is associative.

Problem #47 (Solved !)

Originator: Jan Willem Klop
Date: December 1991

Summary: Prove a Parallel Moves Lemma *for reductions of infinite length.*

For reductions of transfinite length, a version of the Parallel Moves Lemma can be proved if one considers only "strongly converging" infinite reductions in the sense of [KKSd91]. However, if one wants to consider converging reductions, as in [DKP91], then it is not difficult to construct a counterexample, not to the infinite Parallel Moves Lemma itself, but to the method of proof (cf. [KKSd90]). An infinite Parallel Moves Lemma might involve a different notion of "descendant".

Remark

[Sim04] shows that it is not possible to obtain a Parallel Moves Lemma for (Cauchy-)convergent infinite reductions which relies on a notion of *residual* maintaining some of the basic properties of residuals known from the finite case. His counterexamples, however, are somewhat particular in that the right-hand sides of the rewrite rule are not normalized. The question remains whether it is possible to salvage a Parallel Moves Lemma for (Cauchy-)convergent reductions for restricted classes of rewrite systems.

Problem #48 (Solved !)

Originator: H.-C. Kong
Date: December 1991

Summary: Is embedding a well-quasi-ordering on strings?

Consider the following relation on strings over an infinite set $\mathcal{X}$ of variables: $x_1x_2\cdots x_m \hookrightarrow y_1y_2\cdots y_n$ if there exists a renaming $\rho : \mathcal{X} \to \mathcal{X}$ such that $x_i\rho = y_{j_i}$ for $1 \leq j_1 < j_2 < \cdots < j_m \leq n$. Is this "embedding" relation $\hookrightarrow$ a well-quasi-ordering (that is, must every infinite sequence of strings contain two strings, such that the first embeds in the second)?

Remark

The answer is "yes". (Map each variable to the position of its leftmost occurrence and use the fact that strings of natural numbers are well-quasi-ordered by the embedding extension of $\leq$ to strings.)

Problem #49

Originator: Miki Hermann
Date: December 1991

Summary: Can completion always be made terminating when limiting the depth of occurrences of critical pairs?

Suppose ordinary completion (as in [DJ90], for example, is non-terminating for some initial set of equations E, completion strategy, and reduction ordering. Must there be a finite depth N for E such that for any $n > N$ restricting the generation of critical pairs to overlaps at positions that are no deeper than n in the overlapped left-hand side (but otherwise not changing the strategy) also produces a non-terminating completion sequence?

Problem #50

Originator: Jean-Pierre Jouannaud
Date: December 1991

Summary: Investigate confluence and termination of combinations of typed lambda-calculi with term rewriting systems.

Combinations of typed λ-calculi with term-rewriting systems have been studied extensively in the past few years [Bar90][BTG89][DO90][Dou91]. The strongest termination result allows first-order rules as well as higher-order rules defined by a generalization of primitive recursion. Suppose all rules for functional constant F follow the schema:

$$F(\bar{l}[\bar{X}], \bar{Y}) \to v[F(\bar{r}_1[\bar{X}], \bar{Y}), ..., F(\bar{r}_m[\bar{X}], \bar{Y}), \bar{Y})]$$

where the (not necessarily disjoint) variables in $\bar{X}$ and $\bar{Y}$ are of arbitrary order, each of $\bar{l}, \bar{r}_1, ..., \bar{r}_m$ is in $\mathcal{T}(\mathcal{F}, \{\bar{X}\})$, $v[\bar{z}, \bar{Y}]$ is in $\mathcal{T}(\mathcal{F}, \{\bar{Y}, \bar{z}\})$, for new variables $\bar{z}$ of appropriate types, and $\bar{r}_1, \ldots, \bar{r}_m$ are each less than $\bar{l}$ in the multiset extension of the strict subterm ordering. If $\mathcal{T}(\mathcal{F}, \mathcal{X})$ is the term-algebra which includes only *previously* defined functional constants—forbidding the use of mutually recursive functional constants—termination is ensured [JO91a]. Does termination also hold when there are mutually recursive definitions? Does this also hold when the subterm assumption is unfulfilled? (In [JO91a] an alternative schema is proposed, with the subterm assumption weakened at the price of having only first-order variables in $\bar{X}$.) Questions of confluence of combinations of typed λ-calculi and higher-order systems also merit investigation. These results have been extended to combinations with more expressive type systems [BF93b][BF93a].

Remark

An extension to the Calculus of Constructions has been reported in [BFG94]. One can also allow the use of lexicographic and other "statuses" for the higher-order constants when comparing the subterms of F in left and right hand sides [Jouannaud and Okada, unpublished]. Finally, this can also be done when the rewrite rules follow from the induction schema in the initial algebra of the constructors [Wer94].

Important improvements of the previous works have been achieved in [Bla03] and [WC03].

Problem #51 (Solved !)

Originator: Hubert Comon, Max Dauchet
Date: June 1993

> *Summary: Is the first order theory of one-step rewriting decidable?*

For an arbitrary finite term rewriting system R, is the first order theory of one-step rewriting ($\rightarrow_R$) decidable? Decidability would imply the decidability of the first-order theory of encompassment (that is, being an instance of a subterm) [CCD93], as well as several known decidability results in rewriting. (It is well known that the theory of $\rightarrow^*_R$ is in general undecidable.)

Remark

This has been answered negatively in [Tre96, Tre98]. Sharper undecidability results have been obtained for the following subclasses of rewrite systems:

- linear, shallow, $\exists^*\forall^*$-fragment ([STT97], [STTT01]);
- linear, terminating, $\exists^*\forall^*\exists^*$-fragment ([Vor97a]), $\exists^*\forall^*$-fragment ([Mar97]).
- right-ground, terminating, $\exists^*\forall^*$-fragment ([Mar97]).

Decidbability results have been obtained for

- the positive existential theory ([NPR97])
- unary signatures ([Jac96])
- left-linear right-ground systems ([Tis90])

Problem #52 (Solved !)

Originator: Richard Statman
Date: June 1993

Summary: Is there a fixed point combinator Y for which $Y \leftrightarrow^* Y(SI)$?

It has been remarked by C. Böhm [Bar84] that Y is a fixed point combinator if and only if $Y \leftrightarrow^* (SI)Y$ (Y and SIY are convertible). Also, if Y is a fixed point combinator, then so is $Y(SI)$. Is there is a fixed point combinator Y for which $Y \leftrightarrow^* Y(SI)$?

Remark

This was solved by Benedetto Intrigila [Int97] who showed that there is no such fixed point combinator.

Problem #53

Originator: Richard Statman
Date: June 1993

Summary: Are there hyper-recurrent *combinators?*

A term M in Combinatory Logic or λ-calculus is *recurrent* if $N \to^* M$ whenever $N \leftrightarrow^* M$ (this notion is due to M. Venturini-Zilli.) Let's call M *hyper-recurrent* if N is recurrent for all $N \leftrightarrow^* M$. (Equivalently, M is hyper-recurrent if $P \to^* Q \to^* P$ whenever $P \leftrightarrow^* Q \leftrightarrow^* M$.) Are there any hyper-recurrent combinators? (The problem comes up immediately when the Ershov-Visser theory [Vis80] for $\leftrightarrow^*$ is applied to $\to^*$. It is known that hyper-recurrent combinators don't exist for Combinatory Logic [Sta91].)

Problem #54

Originator: Richard Statman
Date: June 1993

> *Summary: In combinatory logic, is there a* uniform universal generator*?*

Recall that M is a *universal generator* if each combinator P has a superterm Q such that $M \to^* Q$. Call M a *uniform universal generator* if there exists a context $C[\cdot]$ such that, for each combinator P, we have $M \to^* C[P]$. Is there a uniform universal generator? (For Combinatory Logic, if we restrict the context $C[\cdot]$ to be of the form $(N\cdot)$, no such term exists [Sta92].)

Problem #55

Originator: Richard Statman
Date: June 1993

> *Summary: In the λ-calculus, which sets have the form $\{M|Q \to^* PM\}$?*

It has been proved that (in λ-calculus or Combinatory Logic) every recursively enumerable set of ground terms that is closed under conversion has the form $\{M|PM \leftrightarrow^* Q\}$ for some P and Q. Which sets have the form $\{M|Q \to^* PM\}$?

Problem #56

Originator: V. van Oostrom
Date: June 1993

Summary: Does the Church-Rosser property of abstract reduction systems imply decreasing Church-Rosser?

An abstract reduction system is "decreasing Church-Rosser", if there exists a labelling of the reduction relation by a well-founded set of labels, such that all local divergences can be completed to form a "decreasing diagram" (see [Oos92] for precise definitions). Does the Church-Rosser property imply decreasing Church-Rosser? That is, is it always possible to localize the Church-Rosser property? This is known to be the case for (weakly) normalizing and finite systems.

Remark

It is now known to hold for countable systems [Man93],[vO94, Cor. 2.3.30].

Problem #57

Originator: Franz Baader [Baa90]
Date: June 1993

Summary: Does there exist a semigroup theory for which there is a reduced canonical term-rewriting system that is not length decreasing?

Does there exist a semigroup theory (without constants in the equations) for which there is a reduced canonical term-rewriting system (with the right-hand side and subterms of the left in normal form) that is not length decreasing?

Problem #58

Originator: Michio Oyamaguchi
Date: June 1993

Summary: Is any "strongly" non-overlapping right-linear term-rewriting system confluent?

Is any "strongly" non-overlapping right-linear term-rewriting system confluent? ("Strong" in the sense that left-hand sides are non-overlapping even when the occurrences of variables have been renamed apart [Che81].) On the one hand, strongly non-overlapping systems need not be confluent [Hue80]; on the other hand, strongly non-overlapping right-ground systems are [OO93].

Remark

A partial positive solution is given in [OOTar][TO94], namely, any strongly non-overlapping right-linear term-rewriting system is confluent if it satisfies the condition that for any rewrite rule, no variables occurring more than once in the left-hand-side occur in the right-hand-side.

Problem #59

Originator: M. Kurihara, M. Krishna Rao
Date: June 1993

Summary: What are sufficient condition for the modularity of confluence?

One of the earliest results established on modularity of combinations of term-rewriting systems is the confluence of the union of two confluent systems which share no symbols [Toy87]; if symbols are shared modularity is not preserved by union [KO92]. Some sufficient conditions for modularity of confluence of constructor-sharing systems that are terminating have been found [KO92][MT91]. Are there interesting sufficient conditions that are independent of termination?

Remark

Left-linearity is a sufficient condition, as shown long ago in [RV80]. In [Ohl94a], it is established that confluence is modular in the presence of the weak normalization property. (This result has been extended in [Rao95, Rao98] for hierarchical combinations.) In [Der97], some results are given when only one of the systems is terminating.

There are other sufficient conditions for modularity of confluence that do not require termination of the combined system even when function symbols are shared. One set of conditions, viz., "persistence", "relative termination", and lr-disjointness, is given in [Ver95, Ver96a]. An abstract confluence theorem without termination is given in [Ges90].

Problem #60 (Solved !)

Originator: Hans Zantema
Date: June 1993

Summary: Does termination of a many-sorted rewrite system reduce to the one-sorted case in case all variables are of the same sort?

Let R be a many-sorted term-rewriting system and R' the one-sorted system consisting of the same rules, but in which all operation symbols are considered to be of the same sort. Any rewrite in R is also a rewrite in R'. The converse does not hold, since terms and rewrite steps in R' are allowed that are not well-typed in R. In [Zan94a] it was shown that termination of R is in general not equivalent to termination of R', but it is if R does not contain both collapsing and duplicating rules. Are termination of R and of R' equivalent in the case where all variables occurring in R are of the same sort? If this statement holds, it would follow that simulating operation symbols of arity n greater than 2 by $n-1$ binary symbols in a straightforward way does not affect termination behavior.

Remark

This has been solved positively by Takahito Aoto [Aot01].

Problem #61 (Solved !)

Originator: Tobias Nipkow, Masako Takahashi
Date: June 1993

Summary: Are weakly orthogonal higher-order rewrite systems confluent?

For higher-order rewrite formats as given by combinatory reduction systems [Klo80] and higher-order rewrite systems [Nip91, Tak93], confluence has been proved in the restricted case of orthogonal systems. Can confluence be extended to such systems when they are weakly orthogonal (all critical pairs are trivial)? When critical pairs arise only at the root, confluence is known to hold.

Remark

Weakly orthogonal higher-order rewriting systems are confluent. This has been shown both via the Tait-Martin-Löf method and via finite developments [vOvR94].

The details and further extensions similar to Huet's parallel closure condition can be found in [Oos94, Oos97a, Raa96].

Problem #62 (Solved !)

Originator: Vincent van Oostrom
Date: June 1993

> *Summary: Is the union of two left-linear, confluent combinatory reduction systems over the same alphabet, where the rules of the first system do not overlap the rules of the second, confluent?*

Let R and S be two left-linear, confluent combinatory reduction systems with the same alphabet. Suppose the rules of R do not overlap the rules of S. Is $R \cup S$ confluent? This is true for the restricted case when R is a term-rewriting system (an easy generalization of a result by F. Müller [Mül92]), or if neither system has critical pairs. (The restriction to the same alphabet is essential, since confluence is in general not preserved under the addition of function symbols, not even for left-linear systems.)

Remark

The answer is yes (Thm. 3.1 of [vOvR94]).

Problem #63 (Solved !)

Originator: Michio Oyamaguchi
Date: June 1993

> *Summary: Is confluence of right-ground term-rewriting systems decidable?*

Is confluence of right-ground term-rewriting systems decidable? Compare [Oya87, DHLT90, DT90, OO93].

Remark

Related is [Oya90].

This has been solved positively by [GTV04].

Problem #64

Originator: Nachum Dershowitz
Date: June 1993

> *Summary: Is confluence of ordered rewriting decidable when the (existential fragment of the) ordering is?*

Is confluence of ordered rewriting (using the intersection of one step replacement of equals and a reduction ordering that is total on ground terms) decidable when the (existential fragment of the) ordering is? This question was raised in [Nie93a], where some results were given for the lexicographic path ordering.

Remark

This was answered positive for the case of lexicographic path orderings, which is probably the most important special case, in [CNNR98]. The general question, however, remains open.

Problem #65

Originator: D. Cohen, Phil Watson [CW91]
Date: June 1993

> *Summary: Is the system of Cohen and Watson for arithmetic terminating?*

An interesting system for doing arithmetic by rewriting was presented in [CW91]. Unfortunately, its termination has not been proved.

Remark

Termination of a related system is proved in [WZ95].

Problem #66 (Solved !)

Originator: Franz Baader, Klaus Schulz
Date:

Summary: Is there an equational theory for which unification with constants is decidable, but general unification is undecidable?

Is there an equational theory for which unification with constants is decidable, but general unification (where free function symbols of arbitrary arity may occur) is undecidable? From the results in [BS92] it follows that this question can be reformulated as follows: Is there an equational theory for which unification with constants is decidable, but unification with linear constant restrictions is undecidable? Another way of formulating the question is: Consider *positive* first-order formulæ containing equality as the only predicate symbol, and function symbols from a given alphabet $\mathcal{F}$. Is there an equational theory E with alphabet $\mathcal{F}$ such that whether $E \models \phi$ is decidable for closed formulae ϕ with quantifier prefix $\forall^*\exists^*$, but undecidable for arbitrary quantifier prefixes?

Remark

This has been answered in the affirmative [Oto12] by exhibiting such an equational theory.

Problem #67

Originator: Franz Baader, Klaus Schulz [BS92]
Date: June 1993

> *Summary: Investigate the exact difference between* linear constant restrictions *and* arbitrary constant restrictions *in unification problems.*

It was shown in [BS92] that being able to solve unification problems with linear constant restrictions is a necessary and sufficient condition for the possibility of combining unification algorithms. Other approaches [SS89][Bou90] require solvability of constant elimination problems, which was shown to be equivalent to presupposing solvability of unification problems with arbitrary constant restrictions [BS92]. Is there an equational theory for which solvability of unification problems with linear constant restrictions is decidable, but solvability of unification problems with arbitrary constant restrictions is undecidable? Is there an equational theory for which unification problems with linear constant restrictions always have a finite complete set of solutions, but unification problems with arbitrary constant restrictions sometimes don't?

Problem #68 (Solved !)

Originator: Hubert Comon
Date: June 1993

> *Summary: Is satisfiability of* set constraints with projection and boolean operators *decidable?*

Consider the existential fragment of the theory defined by a binary predicate symbol $\subseteq$, a finite set of function symbols $f_1, \ldots, f_n$, the function symbols $\cap, \cup, \neg$, and the projection symbols $f_{i,j}^{-1}$ for $j \leq arity(f_i)$. Variables are interpreted as subsets of the Herbrand Universe. With the obvious interpretation of these symbols, is satisfiability of such formulas decidable? Special cases have been solved in [HJ90, AW92, BGW93, GTT93a].

Remark

This has been solved positively by [CP94b].

Partial solutions have been given by [GTT93b][CP94a][AKW93].

Problem #69

Originator: Claude Kirchner, J. Zhang
Date: June 1993

Summary: What is the syntactic type of (mid-, three-way) distributivity?

What is the syntactic type (maximum number of top-level steps needed in an equational proof [BC92]) of the distributivity axiom? What is the syntactic type of "three-way" commutativity:

$$f(x,y,z) = f(x,z,y) = f(y,x,z) = f(y,z,x) = f(z,x,y) = f(z,y,x)$$
$$f(f(x,y,z),u,x) = f(x,y,f(z,u,x))$$

What are the unification type, decidability, and syntactic type of "mid-commutativity": $(x+y)+(u+v) = (x+u)+(y+v)$?

Problem #70

Originator: Jean-Claude Raoult
Date: June 1993

Summary: Design a notion of automata *for graphs.*

There exist finite automata for words, trees, and dags. No really good comparable notion is available for graphs. (Perhaps there is one akin to the ideas in [LMSar] on label rewriting.)

Remark

A well motivated notion of "graph acceptor" has been presented in [Tho97].

Comment sent by Bruno Courcelle

Date: Mon, 31 Jan 2005 10:20:21 +0100

In my opinion, there cannot exist graph automata yielding satisfactory results. There can only exist ad hoc definitions working for very special cases, giving no general theory. However, there is a well established notion of recognizability based on finite congruences. The reason why there cannot exist graph automata is that the "structure" of the graph, on which automata computations must be based (like on trees or words) is not given. Because graphs have far more complex structure than trees. There is no good notion of dag automaton. Of course, one can propose tools and claim "I have the good notion of an automaton".

References: [Cou97]

Problem #71

Originator: Jean-Claude Raoult
Date: June 1993

Summary: Design pattern-matching algorithms for graphs.

There are good algorithms for pattern-matching for words and trees, but not yet for graphs.

Remark

An algorithm for finding the rules of a graph grammar that are applicable to a graph has been given in [BGT91].

Comment sent by Bruno Courcelle

Date: Mon, 31 Jan 2005 10:20:21 +0100

Many types of graph embeddings exist. Thus pattern-matching is not uniquely defined. However, the difficulty of graph isomorphism indicates there cannot exist general algorithms. There may exist in particular cases (bounded degree, for other constraints).

Problem #72

Originator: Jean-Claude Raoult
Date: June 1993

Summary: Give a definition of graph transduction that extends rational word transductions.

Graph rewritings, like term or word rewritings, are usually finitely branching. There are relations that are not finitely branching, yet satisfy good properties: rational transductions of words, tree-transductions. A good definition of graph transduction, that extends rational word transductions is still lacking.

Remark

See [Cou94, Cou97].

Comment sent by Bruno Courcelle

Date: Mon, 31 Jan 2005 10:20:21 +0100

My notion of monadic second-order transduction of graphs, hypergraphs and relational structures is in my opinion the equivalent of tree and word transductions. However, if a graph is given with a tree-structure, then a transducer can be based on this tree: it can produce an algebraic expression defining, after evaluation, the desired object graph, hypergraph etc...

References: [Cou94, Cou97, CK02]

Problem #73

Originator: Jean-Claude Raoult
Date: June 1993

Summary: Find an embedding theorem for directed graphs.

Termination is, as we know, undecidable. Yet, there are several sufficient conditions ensuring termination for word and term rewritings. Most are suitable extensions of Higman's or Kruskal's embeddings [Kru60]. Robertson and Seymour [RS] have achieved a similar theorem for undirected graphs. However, no embedding theorem has yet been proved for directed graphs, and (consequently?) powerful termination orderings remain to be designed.

This problem is related to Problem #100.

Remark

In [RS96], embedding theorems are proved for directed wqo-labelled graphs and hypergraphs.

Comment sent by Bruno Courcelle

Date: Mon, 31 Jan 2005 10:20:21 +0100

Graph rewriting termination: it is usually no problem because there is no duplication of subgraph, and the size reduces. One can of course interpret a term rewriting system as a graph rewriting system, if the symbols of the term denote graph operations. Hence, the termination is handled at the level of terms, with the well-known tools and criteria.

Problem #74

Originator: D. Plump, Bruno Courcelle
Date: June 1993, January 1998

Summary: How can termination orderings for term rewriting be adapted to cover those cases in which graph rewriting is terminating although term rewriting is not?

Graph rewriting systems that implement term rewriting systems (see, for example, [BvG$^+$87][HP91]) are terminating whenever term rewriting is. The converse, however, does not hold [Plu91]. How can termination orderings for term rewriting be adapted to cover those cases in which graph rewriting is terminating although term rewriting is not?

It would be interesting to see an example not too artificial where the termination proof is difficult. Graph rewriting systems implementing term rewriting do not duplicate subgraphs. So the major source of difficulty in termination proofs disappears.

Comment sent by Klaus Guenter Dabisch

Date: Mon Mar 9 13:44:57 MET 1998

This problem was solved intuitively in [Ohl97]. One could solve the difficult termination proofs in this way: The Gross-Knuth reduction is defined by: Contract all redexes simultaneously [BvG$^+$87]. Then, however, one has to prove that the result is unequivocal.

Problem #75

Originator: D. Plump
Date: June 1993

> *Summary: What sufficient conditions make confluence of general (hyper-)graph rewriting decidable?*

In contrast to term rewriting, confluence of general (hyper-)graph rewriting—in the "Berlin approach"—is undecidable, even for terminating systems [Plu93]. What sufficient conditions make confluence decidable?

Problem #76 (Solved !)

Originator: Jean-Pierre Jouannaud
Date: June 1993

> *Summary: Is* cycle unification *decidable?*

Cycle unification has been defined in [BHW92]. Is cycle unification decidable?

Remark

Cycle unification [BHW92] is undecidable [Dev93][HW93]. This was a long standing open problem, related to the non-termination of simple logic programs.

Problem #77 (Solved !)

Originator: Freese
Date: June 1993

> *Summary: Is there a finite, normal form, associative-commutative term-rewriting system for lattices?*

Remark

J Jezek, J. B. Nation, and R. Freese [Fre93] have shown that there is no finite, normal form, associative-commutative term-rewriting system for lattices. This is somewhat surprising because every lattice term is equivalent under lattice theory to a shortest term which is unique up to associativity and commutativity (known as "Whitman canonical form").

Problem #78 (Solved !)

Originator: Pierre Lescanne
Date: April 1995

Summary: Is there a calculus of explicit substitution that is both confluent and preserves termination?

There are confluent calculi of explicit substitutions, but these do not preserve termination (strong normalization) [CHL92, Mel95], and there are calculi that are not confluent on open terms, but which do preserve termination [LRD94]. Is there a calculus of explicit substitution that is both confluent and preserves termination?

Remark

The calculus presented in [Muñ96] enjoys both properties. This has led to Problem #88.

Problem #79

Originator: Mizuhito Ogawa
Date: April 1995

Summary: Does a system that is nonoverlapping under unification with infinite terms have unique normal forms?

Does a system that is nonoverlapping under unification with infinite terms (unification without "occur-check" [MR84]) have unique normal forms? This conjecture was originally proposed in [OO89] with an incomplete proof, as an extension of the result on strongly nonoverlapping systems [Klo80][Che81]. Related results appear in [OO93][TO94][MO94], but the original conjecture is still open. This is related to Problem 58. This problem is also related with modularity of confluence of systems sharing constructors, see [Ohl94b].

Remark

The answer is yes if the system is also nonduplicating [Ver96b]. A direct technique is given in [Ver96b]. The nonduplicating condition can be relaxed under a certain technical condition [Ver96b]. Some extensions to handle root overlaps are given in [Ver97] and a restricted version of the result in [Che81] is also proved using the direct technique in [Ver97].

Problem #80

Originator: Hubert Comon
Date: April 1995

Summary: Is strong sequentiality decidable for arbitrary rewrite systems?

Strong sequentiality is a property of rewrite systems introduced in [HL78] (see [HL91a]), which ensures the existence of optimal reduction strategies. Is strong sequentiality decidable for arbitrary rewrite systems? What is the complexity of strong sequentiality in the linear case? in the orthogonal case? Decidability results for particular rewrite systems are given in [HL91b, Toy92, JS94], among others.

Problem #81 (Solved !)

Originator: Andreas Weiermann
Date: April 1995

Summary: Is it possible to bound the derivation lengths of simply terminating rewrite systems by a multiply recursive function?

If the termination of a finite rewrite system over a finite signature can be proved using a simplification ordering, then the derivation lengths are bounded by a Hardy function of ordinal level less than the small Veblen number $\phi_{\Omega^\omega}0$. (See [Wei93].) Is it possible to lower this bound by replacing the Hardy function by a slow growing function? That is, is it possible to bound the derivation lengths by a multiply recursive function?

Remark

Hélène Touzet [Tou97] has shown in her thesis that the answer is negative, exhibiting a simplifying rewrite system which has derivation bounds "longer" than multiply recursive. This work leaves open the question about what complexity can be achieved using simpifying rewrite systems. An improved version of the proof is given in [Tou98].

In [Tou99], Touzet has shown that for any multiple recursive function f there is a simplifying string rewriting system whose derivation length function dominates f.

The complete solution to the problem is contained in [Lep04], where it is shown that the upper bound from Weiermann is tight, hence for any Hardy function h of ordinal level below the small Veblen ordinal there is a simplifying term rewrite system whose derivation length function dominates h (see also [Lep01]).

Problem #82

Originator: J. Zhang
Date: April 1995

Summary: Is there a convergent extended rewrite system for ternary boolean algebra, in which certain equations hold?

Is there a convergent extended rewrite system for ternary boolean algebra, for which the following permutative equations hold:

$$f(x,y,z) = f(x,z,y) = f(y,x,z) = f(y,z,x) = f(z,x,y) = f(z,y,x)$$
$$f(f(x,y,z),u,x) = f(x,y,f(z,u,x))$$

See [Wos][Zhaar][Chr][Fri85].

Comment sent by Hansjörg Lehner

Date: Wed Dec 20 17:46:07 MET 2000

```
The following permutative equations hold for every ternary boolean algebra:

  f(f(x, y, z), u, x) = f(x, y, f(z, u, x))
  f(x, y, z) = f(x, z, y) = f(y, x, z) = f(y, z, x) = f(z, x, y) = f(z, y, x)


Consider the following set of axioms:
```

```
  Axiom 1: f(f(x1,x2,x3),x4,f(x1,x2,x5)) = f(x1,x2,f(x3,x4,x5))
  Axiom 2: f(x1,x1,x2) = x1

This theorem holds true:

  Theorem 1: f(f(A,B,C),D,A) = f(A,B,f(C,D,A))

Proof:

  Lemma 1: z = f(z,x4,z)

    z
 =    by Axiom 2 RL
    f(z,z,f(y,x4,x5))
 =    by Axiom 1 RL
    f(f(z,z,y),x4,f(z,z,x5))
 =    by Axiom 2 LR
    f(z,x4,f(z,z,x5))
 =    by Axiom 2 LR
    f(z,x4,z)

  Theorem 1: f(f(A,B,C),D,A) = f(A,B,f(C,D,A))

    f(f(A,B,C),D,A)
 =    by Lemma 1 LR
    f(f(A,B,C),D,f(A,B,A))
 =    by Axiom 1 LR
    f(A,B,f(C,D,A))




Consider the following set of axioms:

  Axiom 1: x1 = f(x2,x1,x1)
  Axiom 2: x1 = f(x1,x2,g(x2))
  Axiom 3: f(x1,x2,f(x3,x4,x5)) = f(f(x1,x2,x3),x4,f(x1,x2,x5))

This theorem holds true:

  Theorem 1: f(A,B,C) = f(B,A,C)

Proof:
```

```
 Lemma 1: f(f(g(x1),x0,q),x0,x1) = f(g(x1),x0,x1)

   f(f(g(x1),x0,q),x0,x1)
=    by Axiom 2 LR
   f(f(f(g(x1),x0,q),x1,g(x1)),x0,x1)
=    by Axiom 1 LR
   f(f(f(g(x1),x0,q),x1,g(x1)),x0,f(f(g(x1),x0,q),x1,x1))
=    by Axiom 3 RL
   f(f(g(x1),x0,q),x1,f(g(x1),x0,x1))
=    by Axiom 3 RL
   f(g(x1),x0,f(q,x1,x1))
=    by Axiom 1 RL
   f(g(x1),x0,x1)

 Lemma 2: f(g(g(y)),y,q) = g(g(y))

   f(g(g(y)),y,q)
=    by Axiom 2 LR
   f(f(g(g(y)),y,q),y,g(y))
=    by Lemma 1 LR
   f(g(g(y)),y,g(y))
=    by Axiom 2 RL
   g(g(y))

 Lemma 3: g(g(z)) = z

   g(g(z))
=    by Lemma 2 RL
   f(g(g(z)),z,z)
=    by Axiom 1 RL
   z

 Lemma 4: f(y,y,q) = y

   f(y,y,q)
=    by Lemma 3 RL
   f(g(g(y)),y,q)
=    by Lemma 2 LR
   g(g(y))
=    by Lemma 3 LR
   y

 Lemma 5: f(g(x1),y,x1) = y
```

```
   f(g(x1),y,x1)
=    by Lemma 1 RL
   f(f(g(x1),y,y),y,x1)
=    by Axiom 1 RL
   f(y,y,x1)
=    by Lemma 4 LR
   y

 Lemma 6: f(v,u,x4) = f(u,x4,v)

   f(v,u,x4)
=    by Lemma 5 RL
   f(v,u,f(g(v),x4,v))
=    by Axiom 3 LR
   f(f(v,u,g(v)),x4,f(v,u,v))
=    by Axiom 1 LR
   f(f(v,u,g(v)),x4,f(f(v,v,v),u,v))
=    by Axiom 1 LR
   f(f(v,u,g(v)),x4,f(f(v,v,v),u,f(v,v,v)))
=    by Axiom 3 RL
   f(f(v,u,g(v)),x4,f(v,v,f(v,u,v)))
=    by Lemma 4 LR
   f(f(v,u,g(v)),x4,v)
=    by Lemma 3 RL
   f(f(g(g(v)),u,g(v)),x4,v)
=    by Lemma 5 LR
   f(u,x4,v)

 Theorem 1: f(A,B,C) = f(B,A,C)

   f(A,B,C)
=    by Lemma 6 RL
   f(C,A,B)
=    by Lemma 5 RL
   f(C,A,f(g(A),B,A))
=    by Axiom 3 LR
   f(f(C,A,g(A)),B,f(C,A,A))
=    by Axiom 1 RL
   f(f(C,A,g(A)),B,A)
=    by Axiom 2 RL
   f(C,B,A)
=    by Lemma 6 LR
   f(B,A,C)
```

```
 Consider the following set of axioms:

  Axiom 1: x1 = f(x2,x1,x1)
  Axiom 2: x1 = f(x1,x2,g(x2))
  Axiom 3: f(x1,x2,f(x3,x4,x5)) = f(f(x1,x2,x3),x4,f(x1,x2,x5))

This theorem holds true:

  Theorem 1: f(A,B,C) = f(A,C,B)

Proof:

  Lemma 1: f(v,u,f(g(u),x4,u)) = f(v,x4,u)

    f(v,u,f(g(u),x4,u))
 =    by Axiom 3 LR
    f(f(v,u,g(u)),x4,f(v,u,u))
 =    by Axiom 1 RL
    f(f(v,u,g(u)),x4,u)
 =    by Axiom 2 RL
    f(v,x4,u)

  Lemma 2: f(f(g(x1),x0,q),x0,x1) = f(g(x1),x0,x1)

    f(f(g(x1),x0,q),x0,x1)
 =    by Lemma 1 RL
    f(f(g(x1),x0,q),x1,f(g(x1),x0,x1))
 =    by Axiom 3 RL
    f(g(x1),x0,f(q,x1,x1))
 =    by Axiom 1 RL
    f(g(x1),x0,x1)

  Lemma 3: f(g(g(y)),y,q) = g(g(y))

    f(g(g(y)),y,q)
 =    by Axiom 2 LR
    f(f(g(g(y)),y,q),y,g(y))
 =    by Lemma 2 LR
    f(g(g(y)),y,g(y))
 =    by Axiom 2 RL
```

```
    g(g(y))

 Lemma 4: g(g(z)) = z

    g(g(z))
=     by Lemma 3 RL
    f(g(g(z)),z,z)
=     by Axiom 1 RL
    z

 Lemma 5: f(y,y,q) = y

    f(y,y,q)
=     by Lemma 4 RL
    f(g(g(y)),y,q)
=     by Lemma 3 LR
    g(g(y))
=     by Lemma 4 LR
    y

 Lemma 6: f(v,u,x4) = f(v,x4,u)

    f(v,u,x4)
=     by Lemma 5 RL
    f(v,u,f(x4,x4,u))
=     by Axiom 1 LR
    f(v,u,f(f(g(u),x4,x4),x4,u))
=     by Lemma 2 LR
    f(v,u,f(g(u),x4,u))
=     by Lemma 1 LR
    f(v,x4,u)

 Theorem 1: f(A,B,C) = f(A,C,B)

    f(A,B,C)
=     by Lemma 6 LR
    f(A,C,B)
```

Problem #83

Originator: Jean-Pierre Jouannaud
Date: April 1995

Summary: Extend combination results on rewrite orderings to systems involving $\beta\eta$ reductions.

A collection of rewrite orderings operating on disjoint signatures can be extended to an ordering operating on the union of the signatures, while still preserving part of the properties [Rub94]. Such constructions can be used for proving modular termination properties of rewrite systems. Do they extend to the case where one of the starting orderings is given by $\beta\eta$ reductions on typed lambda terms?

Problem #84

Originator: Jean-Pierre Jouannaud
Date: April 1995

Summary: Is unification of patterns modulo any set of variable-preserving equations decidable?

Unification of patterns (à la [Mil91]) modulo associativity and commutativity has been shown decidable [BC97], repairing the incomplete solution in [QW94]. Does it extend to equational theories whose axioms have the same set of variables on left and right hand side?

Comment sent by Evelyne Contejean

Date: Mon Jan 12 15:20:45 MET 1998

In his conference paper, Qian claimed that he has solved the problem of unifying patterns a la Miller modulo AC, but in fact he never succeeded to prove the completeness of his algorithm. Actually his algorithm is not complete, since he uses a first-order unification algorithm for pure AC-patterns as a black box. The problem was solved last year by Boudet and Contejean [BC97]: the case of pure AC-patterns requieres is handled in the same spirit as the first order case, by counting things, but technically this is not exactly identical. In [BC97], the proof of completeness of the algorithm is given. I must admit that [BC97] takes advantage of the paper of Qian, in particular, the remark that the equations of the form

$$\lambda x_1...x_n F(x_1, ..., x_n) = \lambda x_1...x_n F(x_{\pi(1)}, ..., x_{\pi(n)})$$

have an infinite set of solutions $\{\sigma_1, \sigma_2, ...\}$ such that σ_{i+1} is strictly more general than σ_i. This leads to the notion of constrained solution of a unification problem, and every unification problem of patterns with AC symbols admits a finite complete set of constrained unifiers, and the algorithm proposed in [BC97] computes such a set.

Problem #85 (Solved !)

Originator: Michaël Rusinowitch
Date: April 1995

> *Summary: Can the restrictions on orderings for the use in ordered theorem proving strategies be relaxed?*

Ordered paramodulation is known to be complete for simplification orderings that are total on ground terms [HR86]. Other theorem proving strategies are similarly restricted. How can these restrictions be relaxed?

Remark

[BGNR99] shows that it is sufficient for the ordering to be well-founded and to have the subterm property.

Problem #86

Originator: Hans Zantema
Date: April 1995

> *Summary: Is the union of two totally terminating rewrite systems, which do not share any symbols, totally terminating?*

When there exists a monotonic well-ordering (*monotonic* means that replacing a subterm with a smaller one decreases the whole term) of ground terms that shows termination of a rewrite system, the system is called *totally terminating*. The union of two totally terminating rewrite systems which do not share any symbols is totally terminating if at least one of them does not contain a rule that has more occurrences of some variable on the right than on the left [FZ93, FZ96]. What if variables are duplicated?

Problem #87 (Solved !)

Originator: Hans Zantema
Date: April 1995

> *Summary: Is it decidable whether a single term rewrite rule can be proved terminating by a monotonic ordering that is total on ground terms?*

Termination of string-rewriting systems is known to be undecidable [HL78]. Termination of a single term-rewriting rule was proved undecidable in [Dau92, Les94]. It is also undecidable whether there exists a simplification ordering that proves termination of a single term rewriting rule [MG95] (cf. [JK84]). Is it decidable whether a single term rewrite rule can be proved terminating by a monotonic ordering that is total on ground terms? (With more rules it is not [Zan94b].)

Remark

A negative solution has been given in [GMOZ97]. More about the history of this problem in the context of the question of one-rule termination can be found in [Der05].

Problem #88

Originator: Delia Kesner
Date: January 1998

> *Summary: Is there a calculus of explicit substitution that is confluent on open terms, simulates one-step beta-reduction and preserves beta-strong normalization?*

There are confluent calculi of explicit substitutions but these do not preserve termination (strong normalization) [CHL92, Mel95], and there are calculi that are not confluent on open terms but which do preserve termination [LRD94]. César Muñoz presented in [Muñ96] a calculus enjoying both properties (answering Problem #78), however, the calculus is not able to simulate one-step of beta-reduction: if a beta-reduces to b in the lambda-calculus then a does not necessarily reduce to b in the calculus of Muñoz. Is there a calculus of explicit substitution that is confluent on open terms, simulates one-step beta-reduction and preserves beta-strong normalization?

Comment sent by Jean Goubault-Larrecq

Date: Mon Nov 27 16:37:43 MET 2000

This problem was solved positively in [GL99]. The calculus SKInT, introduced in [GGL00], is confluent on open terms and simulates one-step beta-reduction (although in a slightly contorted way, see [GGL00]; the obvious translation only simulates a bit more that one-step call-by-value beta-reduction). The paper [GL99] characterizes strongly normalizing, weakly normalizing and solvable terms through intersection types, and preservation of strong normalization follows. SKInT is also standardizing, has a terminating subcalculus of substitutions ΣT, but is based on an infinite signature and finitely many rule schemes parameterized by integers. Can we lift the latter restriction?

Problem #89

Originator: Hubert Comon, Robert Nieuwenhuis
Date: January 1998

> *Summary: Is the satisfiablity of ordering constraints (lpo) in conjunction with predicates like irreducibility by a fixed rewrite system or membership in a regular tree language decidable?*

Satisfiability of ordering constraints (lpo) for total precedences has been shown decidable in [Com90, Nie93b]. Is the satisfiablity of total lpo ordering constraints together with the constraint $Irr(x)$, expressing that x is not reducible by some fixed rewrite system, decidable? This would imply decidability of the confluence of ordered rewriting (see Problem #64).

Besides the irreducibility predicate the following related predicates are of interest:

- membership in a fixed regular tree language
- a predicate expressing that a fixed symbol does not occur in a term.

Problem #90

Originator: Hubert Comon, Manfred Schmidt-Schauß, Jordi Levy [Com91], [SS94b], [Lev96]
Date: September 1991, 1994, July 1996

> *Summary: Are* context unification *and* linear second order unification *decidable?*

Context unification and *linear second order unification* are closely related, they both generalize string unification (which is known to be decidable, [Mak77]) and are special cases of second order unification (which is know to be undecidable, [Gol81]).

Context unification ([Com91], [SS94b]) is unification of first-order terms with context variables that range over terms with one hole. Linear Second Order Unification is second-order unification where the domain of functions is restricted to λ-terms with exactly one occurrence of any bound variable (there can be several bound variables in contrast to context unification allowing for just one hole) Applications are

- solving membership constraints in completion of contraint rewriting ([Com98a])
- solving constraints occurring in Distributive Unification (Problem #38, [SS97])
- Extended Critical Pairs in Bi-Rewriting Systems ([LA96])
- Semantics of ellipses in natural language ([NPR97])
- One-Step Rewriting constraints ([NPR97])

Some special cases have been solved:

- Hubert Comon [Com98b] solved a special case where any occurrence of the same context variable is always applied to the same term,
- Manfred Schmidt-Schauß [SS94b] (see also [SS97]) solved the case of so-called *stratified* context unification, where for any occurrence of the same second-order variable the string of second-order variables from this occurrence to the root of the containing term is the same,
- Jordi Levy [Lev96] (see also [NPR97]) showed that linear-second order unification is decidable when any variable has at most two occurrences.
- Manfred Schmidt-Schauß and Klaus Schulz [SSS99a] showed that solvability is decidable for systems of context equations containing only two context variables (having an arbitrary number of occurrences in the system) and an arbitray number of first-order variables.

Progress towards a decidability proof along the lines of Makanin's proof for string-unification has been reported in [SSS98]. Levy and Villaret [LV00] show how to reduce linear second-order unification to context unification plus membership predicates in regular tree languages, and discuss a possible way of showing decidability of the latter. [LV02] shows that it is sufficient, both for linear 2nd-order and for context unification, to consider signatures consisting of an arbitrary number of constants and one binary function symbol.

Problem #91

Originator: Friedrich Otto [OSKM98]
Date: March 1998

Summary: Does every automatic group have a presentation through some finite convergent string-rewriting system?

Does every automatic monoid have an automatic structure such that the set of representatives is a prefix-closed cross-section?

For a finite alphabet Σ, we define the *padded extension* $\Sigma_\#$ of Σ as

$$\Sigma_\# := ((\Sigma \cup \{\#\}) \times (\Sigma \cup \{\#\})) \setminus \{(\#, \#)\},$$

where $\#$ is an additional symbol. A mapping $\nu : \Sigma^* \times \Sigma^* \to \Sigma_\#^*$ is then used to encode pairs of strings from Σ^* as strings from $\Sigma_\#^*$ as follows:
if $u := a_1 a_2 \cdots a_n$ and $v := b_1 b_2 \cdots b_m$, where $a_1, \ldots, a_n, b_1, \ldots, b_m \in \Sigma$, then

$$\nu(u, v) := \begin{cases} (a_1, b_1)(a_2, b_2) \cdots (a_m, b_m)(a_{m+1}, \#) \cdots (a_n, \#), & \text{if } m < n, \\ (a_1, b_1)(a_2, b_2) \cdots (a_m, b_m), & \text{if } m = n, \\ (a_1, b_1)(a_2, b_2) \cdots (a_n, b_n)(\#, b_{n+1}) \cdots (\#, b_m), & \text{if } m > n. \end{cases}$$

Now a subset $L \subseteq \Sigma^* \times \Sigma^*$ is called *synchronously regular*, *s-regular* for short, if $\nu(L) \subseteq \Sigma_\#^*$ is accepted by some finite state acceptor (fsa).

An *automatic structure* for a finitely generated monoid-presentation $(\Sigma; R)$ consists of a fsa W over Σ, a fsa $M_=$ over $\Sigma_\#$, and fsa's M_a ($a \in \Sigma$) over $\Sigma_\#$ satisfying the following conditions:

1. $L(W) \subseteq \Sigma^*$ is a complete set of (not necessarily unique) representatives for the monoid M_R presented by $(\Sigma; R)$, that is, $L(W) \cap [w]_R \neq \emptyset$ holds for each $w \in \Sigma^*$,
2. $L(M_=) = \{\nu(u, v) \mid u, v \in L(W) \text{ and } u \leftrightarrow_R^* v\}$, and

3. for all $a \in \Sigma$, $L(M_a) = \{\nu(u,v) \mid u,v \in L(W) \text{ and } ua \leftrightarrow_R^* v\}$.

A monoid-presentation is called *automatic* if it admits an automatic structure, and a monoid is called *automatic* if it has an automatic presentation.

Groups with automatic structure have been investigated thoroughly [Eps92], while the automatic monoids have been investigated only recently [CRRT96]. It is known that there exists monoids (in fact, groups) that can be presented through finite convergent string-rewriting systems, but that are not automatic [Ger92a].

QUESTION 1: Does every automatic group have a presentation through some finite convergent string-rewriting system?

For monoids in general the answer is negative as proved by an example given in [OSKM98].

If $(W, M_=, M_a(a \in \Sigma))$ is an automatic structure for a monoid-presentation $(\Sigma; R)$, then the language $L(W)$ contains one or more strings from every congruence class $[w]_R(w \in \Sigma^*)$. Actually, it can be required without loss of generality that $L(W)$ is a *cross-section* for $(\Sigma; R)$, that is, it contains exactly one string from every congruence class [Eps92].

Instead of requiring uniqueness one can also transform the given automatic structure in such a way as to obtain one for which the set of representatives is prefix-closed. However, the following question is still open.

QUESTION 2: Does every automatic monoid have an automatic structure such that the set of representatives is a prefix-closed cross-section?

Gersten stated this question for the special case of groups [Ger92b]. If the language $L(W)$ is a prefix-closed cross-section, then there exists an s-regular convergent prefix-rewriting system P on Σ such that the right-congruence generated by P coincides with the congruence generated by R, and $L(W)$ coincides with the set of irreducible strings mod P. Conversely, if a monoid-presentation admits an s-regular convergent prefix-rewriting system, then it has an automatic structure $(W, M_=, M_a(a \in \Sigma))$ such that the set $L(W)$ is a prefix-closed cross-section. Thus, QUESTION 2 can be reformulated as follows.

QUESTION 2 (restated): Does every finitely presented automatic monoid admit an s-regular convergent prefix-rewriting system?

For additional information on monoid-presentations and convergent string-rewriting systems see e.g. [BO93], and for the notion of prefix-rewriting systems see e.g. [KM89].

Problem #92

Originator: Klaus Schulz
Date: September 1998

> *Summary: What is the exact complexity of word unification?*

Satisfiability of word equations, that is unifiability in the algebra of ground terms built on a set of constants and a binary, associative concatenation operator, has been shown decidable by [Mak77], see [Die02] for a recent presentation of Makanin's algorithm. The best known upper bounds for its complexity are *exponential space* and *doubly exponential time* ([Gut98]), leaving a wide gap to the best known lower bound of its complexity which is just NP (see [Die02]). There is an strange discrepancy between this weak lower bound and the enormous difficulty in designing word unification algorithms. So, what is the exact complexity of word unification?

Remark

Satisfiability of word equations is in PSPACE [Pla99].

Problem #93

Originator: Ralf Treinen [Tre96]
Date: 1996

> *Summary: Are the existential fragment or the positive fragment of the theory of one-step rewriting decidable?*

For a given signature Σ and rewrite system R, the theory of one-step rewriting by R is the first order theory of the model comprising all Σ-ground-terms, and the binary predicate *x rewrites to y in one rewrite step of R*.

It is well-known that the full first-order theory is undecidable, even for strong restrictions on the class of rewrite systems (see Problem #51). Is the existential fragment of this theory (in other words: satisfiability of quantifier-free formulas) decidable? Is the positive fragment (arbitrary quantification, but no negation or implications) decidable?

It is known that the positive existential fragment is decidable [NPR97], and there are decidability results for the full existential fragment in case of restricted classes of rewrite systems [CSTT99, LR99].

Problem #94

Originator: Gérard Huet [Hue76]
Date: 1979

Summary: *Is* higher-order matching *decidable?*

Higher-order matching is the following problem:

Given a set of equations $s_i = t_i$ between typed lambda-terms where the t_i are ground, is there a substitution σ such that $\sigma s_i = t_i$ for all i. The order of the matching problem is the maximal height of function arrows in the types of the terms. Is higher-order matching decidable for arbitrary order? The problem has non-elementary complexity [Vor97b].

The following results are known:

- First-order matching is, of course, decidable.
- Second-order matching is decidable [Hue76].
- Third-order matching is decidable [Dow93].
- Fourth-order matching is decidable [Pad96] and NEXPTIME-hard [Wie99]. The solutions can be described by a tree automaton [CJ97b], which gives an 2-NEXPTIME upper bound.
- A restricted case of fifth-order matching has been shown decidable in [Sch97].
- Linear higher-order matching is decidable and NP-complete [dG00a].

More on the complexity of higher-order matching can be found in [Wie99].

This problem is also listed as *Problem #21* in the TLCA list of open problems.

Remark

It has recently been announced [Sti06] that the higher-order matching problem is decidable. However, the proof method only applies to the "classical" case of the problem, that is when all types are built from a single atom. Therefore, the problem remains open for the generalized case of types built from an arbitrary number of type variables.

Problem #95

Originator: Hans Zantema
Date: 1999

Summary: Is there a one-rule string rewriting system that is non-terminating but also non-looping?

Is there a one-rule string rewriting system that is non-terminating but also non-looping, that is not allowing a rewrite sequence $x \to^* uxv$? Motivation and examples of two rule string rewriting systems and a one rule term rewriting system that is non-terminating and non-looping, can be found in [ZG96].

This problem is related to Problem #21 and Problem #87. More about the history of this problem in the context of the question of one-rule termination can be found in [Der05].

Problem #96

Originator: Richard Statman [Sta00]
Date: July 2000

Summary: Is the word problem for all proper combinators of order smaller than 3 decidable?

The order of a proper combinator is the number of variables on the left hand side of its defining equation. For instance, the K combinator has order 2. Is the word problem for all proper combinators of order smaller than 3 decidable? See [Sta00] for related results.

A related question is the word problem for the S-combinator (of order 3), see Problem #97.

Problem #97

Originator: Henk Barendregt [Bar75]
Date: 1975

Summary: Is the word problem for the S-combinator decidable?

The word problem for the S-combinator is: Given two ground terms build only of the constant S in combinatory logic (that is with an application operator written as juxtaposition, and parantheses), are they convertible in the system consisting only of the definition of the S-combinator

$$Sxyz \to (xz)(yz)$$

Is the word-problem for the S-combinator decidable? See also [Wal98b] and [Wal98a] for more background.

A related problem is the word problem for proper combinators of order smaller than 3 (S is of order 3), see Problem #96.

Problem #98

Originator: Dan Dougherty (Talk at RTA 2000)
Date: July 2000

Summary: Is unification modulo the theory of allegories decidable?

Let ALL be the equational theory of Allegories. Is unification modulo ALL decidable?

Background:

The notion of "Allegory" has defined by Peter Freyd and Andre Scedrov in their monograph [FS90]. Allegories are to binary relations between sets as categories are to functions between sets. By ALL we refer to the untyped version of the theory (see page 195 of [FS90]).

Validity in this equational theory is decidable (Gutiérrez' dissertation, Wesleyan University 1999, also see [DG00b]). The universal-existential theory over these axioms is undecidable (reduction from the universal-existential theory of free semigroups with constants).

Problem #99 (Solved !)

Originator: Konstantin Korovin and Andrei Voronkov [KV00]
Date: June 2000

> *Summary: Is the first-order theory of any* Knuth-Bendix ordering *decidable?*

Is there an algorithm which, given a term signature Σ, a weight function w and a precedence $>>$, decides whether a first-order formula is valid in the term-algebra with the Knuth-Bendix ordering defined by $(w, >>)$?

Positive partial results have been given for

- the existential fragment [KV00, KV01];
- signatures consisting only of constants and unary function symbols [KV02a].

Remark

This has been answered in the affirmative [ZSM05].

Problem #100

Originator: Nachum Dershowitz
Date: October 16, 2002

> *Summary: Design new termination methods based on the gap-embedding theorems of Friedman and Kriz.*

Harvey Friedman [Sim85] modified Kruskal's Tree Theorem to restrict labels that appear along the path between the images of adjacent nodes to what is called *gap embedding*. Whereas Friedman's result applied only to labellings with the natural numbers, Igor Kříž [K8̆9] extended it to arbitrary ordinal labellings. See also [Gor90]. The question is whether new and useful termination methods can be based on these gap-embedding theorems. One step in this directions is [Oga95].

This problem is related to Problem #73.

Problem #101

Originator: Hitoshi Ohsaki [OT02]
Date: July 2002

Summary: Are universality and inclusion of AC-recognizable languages decidable?

An *AC-tree automaton* as defined by [Ohs01] is given by a signature Σ, a set of AC-axioms (that is, associativity and commutativity) for some function symbols of Σ, and a set of rewrite rules R of the form

$$f(q_1, \ldots, q_n) \rightarrow q \quad (1)$$
$$f(q_1, \ldots, q_n) \rightarrow f(p_1, \ldots, p_n) \quad (2)$$
$$q \rightarrow p \quad (3)$$

where the q's and p's are state symbols. Such an automaton accepts a term t iff it rewrites t modulo the given AC-axioms to some final state. $L(A)$ denotes the language recognized by an AC-tree automaton A; a language L is called *AC-recognizable* iff $L = L(A)$ for some AC-tree automaton A.

Are the following questions decidable?

- *Universality*: Given an AC-tree automaton A, is $L(A)$ equal to the set of all ground terms over the given signature Σ?
- *Inclusion*: Given AC-tree automata A and B, is $L(A)$ a subset of $L(B)$?

It has been shown [OT02] that emptiness of AC-recognizable languages is decidable. Furthermore, as a consequence of the results of [ZL03], universality and inclusion are decidable if transition rules of the form $f(q_1, \ldots, q_n) \rightarrow f(p_1, \ldots, p_n)$ are not allowed (this is the sub-class of so-called *regular* AC tree-automata). However, both questions are still open in the general case.

Remark

The inclusion problem of AC-tree automata is undecidable [OTTR05]. Decidability of universality is still an open question.

Problem #102

Originator: Bruno Courcelle
Date: January 2005

Summary: Investigate normalization by a canonical term rewrite system in the setting of second-order monadic logic

Consider a confluent and terminating term rewriting system and the mapping from a term to its normal form. When is this mapping a monadic second-order transduction? When does it preserve decidability of the monadic second-order theory of a set of terms?

See [Cou94, CK02]

Problem #103

Originator: Bruno Courcelle
Date: January 2005

Summary: Equational axiomatization of graph operations

Given a set of graph operations like disjoint union, edge-complement, quantifier-free definable operations (like relabellings). When can one axiomatize its equational theory by finitely many equational rules or by infinitely many rules describable in finitary ways?

See [BC87].

Problem #104 (Solved !)

Originator: Hans Zantema
Date: July 2005

Summary: Termination of replacing two successive occurrences of the same symbol in a string

Start by a finite string over the alphabet $\{a, b, c\}$. As long as two consecutive symbols are the same, they may be replaced by the other two symbols in alphabetic order. So

- aa may be replaced by bc,
- bb may be replaced by ac, and

- cc may be replaced by ab.

Can this go on forever?

This problem coincides with establishing termination of the string rewrite system consisting of the three rules

$$\begin{array}{rcl} aa & \to & bc \\ bb & \to & ac \\ cc & \to & ab \end{array}$$

Up to renaming it coincides with problem SRS/Zantema/z086 in the termination problem data base TPDB, on which all tools failed in the Termination Competition 2005. A variant of this problem on multisets, the Chamelon Problem, is known to be non-terminating.

Remark

Termination of this system has been shown by Hofbauer and Waldmann [HW05]. The derivational complexity of this system is open, see Problem 105.

Problem #105 (Solved !)

Originator: Johannes Waldmann (Talk at RTA'06)
Date: August 2006

Summary: Derivational complexity of replacing two successive occurrences of the same symbol in a string

The following string rewrite system is known to be terminating [HW05], see Problem 104.

$$\begin{array}{rcl} aa & \to & bc \\ bb & \to & ac \\ cc & \to & ab \end{array}$$

Is the derivational complexity polynomially bounded? (It is at least quadratic.).

Remark

There is a quadratic bound on the length of derivation sequences [Adi09].

Problem #106

Originator: Jürgen Giesl and Hans Zantema
Date: July 2010

Summary: Can we use the dependency pair method to prove relative termination?

The key of the success of the dependency pair method in proving termination is the following property from [AG00, GTSKF06], stated in more recent terminology:

> A TRS R is terminating if and only if the dependency pair problem $(DP(R), R)$ is terminating.

A dependency pair problem is a pair (P, R) of TRSs. Such a dependency pair problem is called terminating if it admits no infinite chain, that is, there is no $P \cup R$ reduction containing infinitely many P-steps, where P-steps only occur at the root.

Can we use the dependency pair method to prove relative termination? Here for a pair (R, S) of TRSs, R is said to be terminating modulo S if there is no $R \cup S$ reduction containing infinitely many R-steps. This is the same requirement as for termination of a dependency pair problem, except that the first TRS in a dependency pair problem may only be used for root steps. So, more precisely, the open problem is:

> Find a "useful" effectively computable function ϕ from pairs of TRSs to dependency pair problems, such that for every two TRSs R, S the TRS R is terminating modulo S if and only if the dependency pair problem $\phi(R, S)$ is terminating.

Here, "useful" means that the resulting dependency pair problem $\phi(R, S)$ should be "easy" (i.e., suitable for automated termination analysis by existing tools).

Problem #107

Originator: Georg Moser and Harald Zankl
Date: 2010

Summary: Give a complete (resource free) characterisation of rewrite systems with polynomial derivational complexity.

It is well-known that well-founded monotone algebras form a complete characterisation for termination while such a result is currently unknown for polynomial derivational complexity. The notion of *resource freeness* is borrowed from implicit computational complexity theory. Here it refers to characterisations devoid of *direct* references to polynomial derivational complexity.

Currently suitably restricted matrix interpretations (see [MSW08, Wal10, NZM10]) form *the* method for proving polynomial upper bounds on the derivational complexity. Thus it is perhaps important to emphasise that matrix interpretations as studied in [EWZ08] are not sufficient as a starting point to solve the problem. Consider the one-rule TRS $\mathsf{g}(x,x) \to \mathsf{g}(\mathsf{a},\mathsf{b})$. This TRS has linear derivational complexity, but no compatible matrix interpretation can exist.

Bibliography

[Abr94] Samson Abramsky, editor. *Ninth Symposium on Logic in Computer Science*, Paris, France, July 1994. IEEE.

[Adi09] Sergei Adian. Upper bound on the derivational complexity in some word rewriting system. *Doklady Mathematics*, 80(2):679–683, October 2009.

[AG00] Thomas Arts and Jürgen Giesl. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 236:133–178, 2000.

[AKW93] A. Aiken, D. Kozen, and E. Wimmers. Decidability of systems of set constraints with negative constraints. Technical Report 93-1362, Computer Science Department, Cornell University, 1993.

[AM96] Sergio Antoy and Aart Middeldorp. A sequential reduction strategy. *Theoretical Computer Science*, 165(1):75–95, 1996.

[Aot01] Takahito Aoto. Solution to the problem of Zantema on a persistent property of term rewriting systems. *Journal of Functional and Logic Programming*, 2001(11), December 2001.

[AW92] A. Aiken and E. Wimmers. Solving systems of set constraints. In Scedrov [Sce92], pages 329–340.

[Baa90] Franz Baader. Rewrite systems for varieties of semigroups. In M. Stickel, editor, *Proceedings of the Tenth International Conference on Automated Deduction (Kaiserslautern, West Germany)*, volume 449 of *Lecture Notes in Computer Science*, pages 381–395, Berlin, July 1990. Springer-Verlag.

[Bac00] Leo Bachmair, editor. *Rewriting Techniques and Applications*, volume 1833 of *Lecture Notes in Computer Science*, Norwich, UK, July 2000. Springer-Verlag.

[Bar75] Henk Barendregt. Open problems. In Corrado Böhm, editor, *Lambda Calculus and Computer Science Theory*, volume 37 of *Lecture Notes in Computer Science*, pages 367–370. Springer-Verlag, 1975.

[Bar84] Henk Barendregt. *The Lambda Calculus, its Syntax and Semantics*. North-Holland, Amsterdam, second edition, 1984.

[Bar90] Franco Barbanera. Combining term rewriting and type assignment systems. *IJFCS*, 1:165–184, 1990.

[Bar91] Henk Barendregt. Lambda calculi with types. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*. Oxford University Press, Oxford, 1991. To appear.

[Bau85] G. Bauer. n-level rewriting systems. *Theoretical Computer Science*, 40:85–99, 1985.

[BB92] Alessandro Berarducci and Corrado Böhm. A self-interpreter of lambda calculus having a normal form. Rapporto tecnico 16, Dip. di Matematica Pura ed Applicata, Universita di L'Aquila, October 1992.

[BC87] Michel Bauderon and Bruno Courcelle. Graph expressions and graph rewritings. *Mathematical Systems Theory*, 20(2–3):83–127, 1987.

[BC92] Alexandre Boudet and Evelyne Contejean. On n-syntactic equational theories. In H. Kirchner and G. Levi, editors, *Proceedings of the Third International Conference on Algebraic and Logic Programming*, volume 632 of *Lecture Notes in Computer Science*, pages 446–457, Pisa, Italy, September 1992. Springer-Verlag.

[BC93] Alexandre Boudet and Hubert Comon. About the theory of tree embedding. In M. C. Gaudel and J.-P. Jouannaud, editors, *4th International Joint Conference on Theory and Practice of Software Development*, volume 668 of *Lecture Notes in Computer Science*, Orsay, France, April 1993. Springer-Verlag.

[BC94] Alexandre Boudet and Evelyne Contejean. "Syntactic" AC-unification. In Jouannaud [Jou94], pages 136–151.

[BC97] Alexandre Boudet and Evelyne Contejean. AC-unification of higher-order patterns. In Gert Smolka, editor, *Principles and Practice of Constraint Programming*, volume 1330 of *Lecture*

Notes in Computer Science, pages 267–281, Linz, Austria, October 1997. Springer-Verlag.

[BCD90] Alexandre Boudet, Evelyne Contejean, and Hervé Devie. A new *AC* unification algorithm with an algorithm for solving diophantine equations. In Mitchell [Mit90], pages 289–299.

[BD87] Leo Bachmair and Nachum Dershowitz. Inference rules for rewrite-based first-order theorem proving. In *Second Symposium on Logic in Computer Science*, pages 331–337. IEEE, 1987.

[BD97] Michel Bidoit and Max Dauchet, editors. *Theory and Practice of Software Development*, volume 1214 of *Lecture Notes in Computer Science*, Lille, France, April 1997. Springer-Verlag.

[Ber94] A. Bertrand. Sur une conjecture d'Yves Métivier. *Theoretical Computer Science*, 123(1):21–30, 1994.

[BF93a] Franco Barbanera and Maribel Fernández. Combining first and higher order rewrite systems with type assignment systems. In *International Conference of Typed Lambda Calculi and Applications*, 1993.

[BF93b] Franco Barbanera and Maribel Fernández. Modularity of termination and confluence in combinations of rewrite systems with λ_ω. In Andrzej Lingas, Rolf Karlsson, and Svante Carlsson, editors, *20th International Colloquium on Automata, Languages and Programming*, volume 700 of *Lecture Notes in Computer Science*, Lund, Sweden, July 1993. Springer-Verlag.

[BFG94] Franco Barbanera, Maribel Fernández, and Herman Geuvers. Modularity of strong normalization and confluence in the λ-algebraic-cube. In Abramsky [Abr94].

[BGNR99] Miquel Bofill, Guillem Godoy, Robert Nieuwenhuis, and Albert Rubio. Paramodulation with non-monotonic orderings. In *Fourteenth IEEE Annual Symposium on Logic in Computer Science*, Trento, Italy, July 1999. IEEE.

[BGT91] Horst Bunke, T. Glauser, and T.-H. Tran. An efficient implementation of graph grammars based on the RETE-matching algorithm. In H. Ehrig, H.-J. Kreowski, and G. Rozenberg, editors, *Graph Grammars and Their Application to Computer Science*, volume 532 of *Lecture Notes in Computer Science*, pages 174–189, 1991.

[BGW93] Leo Bachmair, Harald Ganzinger, and Uwe Waldmann. Set constraints are the monadic class. In Moshe Y. Vardi, editor, *Eigth Symposium on Logic in Computer Science*, pages 75–83, Montreal, Canada, June 1993. IEEE.

[BHW92] Wolfgang Bibel, S. Hölldobler, and Jörg Würtz. Cycle unification. In Kapur [Kap92], pages 94–108.

[BK86] J. A. Bergstra and Jan Willem Klop. Conditional rewrite rules: Confluency and termination. *Journal of Computer and System Sciences*, 32(3):323–362, 1986.

[BL79] Gérard Berry and Jean-Jacques Lévy. Mimimal and optimal computations of recursive programs. *J. of the Association for Computing Machinery*, 26:148–175, 1979.

[Bla03] Frédéric Blanqui. Definitions by rewriting in the Calculus of Constructions, 2003. To appear in Mathematical Structures in Computer Science.

[BO93] R. V. Book and Friedrich Otto. *String-Rewriting Systems*. Springer-Verlag, 1993.

[Boo91] Ronald. V. Book, editor. *4th International Conference on Rewriting Techniques and Applications*, volume 488 of *Lecture Notes in Computer Science*, Como, Italy, April 1991. Springer-Verlag.

[Bou90] Alexandre Boudet. Unification in combination of equational theories: An efficient algorithm. In *Proceedings of the Tenth International Conference on Automated Deduction (Kaiserslautern, Germany)*, volume 449 of *Lecture Notes in Computer Science*. Springer-Verlag, 1990.

[Bou92] Alexandre Boudet. Unification in order-sorted algebras with overloading. In Kapur [Kap92].

[BPW89] Timothy Baird, Gerald Peterson, and Ralph Wilkerson. Complete sets of reductions modulo Associativity, Commutativity and Identity. In Dershowitz [Der89], pages 29–44.

[BR91] Francois Bronsard and Uday S. Reddy. Conditional rewriting in Focus. In M. Okada, editor, *Proceedings of the Second International Workshop on Conditional and Typed Rewriting Systems*, volume 516 of *Lecture Notes in Computer Science*, Montreal, Canada, 1991. Springer-Verlag.

[BR93] F. Bronsard and U. S. Reddy. Reduction techniques for first-order reasoning. In Michaël Rusinowitch and Rémy [MRR93], pages 242–256.

[Bro95] Francois Bronsard. *Using Term Orders to Control Deductions*. PhD thesis, University of Illinois, 1995. Forthcoming.

[BS92] Franz Baader and Klaus Schulz. Unification in the union of disjoint equational theories: Combining decision procedures. In Kapur [Kap92].

[BS93] Franz Baader and Klaus Schulz. Combination techniques and decision problems for disunification. In Kirchner [Kir93], pages 301–315.

[BS95a] Franz Baader and Klaus Schulz. Combination of constraint solving techniques: An algebraic point of view. In Hsiang [Hsi95], pages 352–366.

[BS95b] Franz Baader and Klaus Schulz. On the combination of symbolic constraints, solution domains, and constraint solvers. In Ugo Montanari and Francesca Rossi, editors, *Principles and Practice of Constraint Programming*, volume 976 of *Lecture Notes in Computer Science*, Cassis, France, September 1995. Springer-Verlag.

[BS98] Franz Baader and Klaus Schulz. Combination of constraint solvers for free and quasi-free structures. *Theoretical Computer Science*, 1998. To appear.

[BTG89] Val Breazu-Tannen and Jean Gallier. Polymorphic rewriting conserves algebraic strong normalization. In *16th International Colloquium on Automata, Languages and Programming*, volume 372 of *Lecture Notes in Computer Science*, pages 137–150, Stresa, Italy, July 1989. Springer-Verlag.

[BvG+87] Henk Barendregt, M. C. J. D. van Eekelen, J. R. W. Glauert, J. R. Kennaway, M. J. Plasmeijer, and M. R. Sleep. Term graph rewriting. In *Proceedings of the European Workshop on Parallel Architectures and Languages*, volume 259 of *Lecture Notes in Computer Science*, pages 141–158, Berlin, 1987. Springer-Verlag.

[Car91] Anne-Cécile Caron. Linear bounded automata and rewrite systems: Influence of initial configurations on decision properties. In *Proceedings of the International Joint Conference on Theory and Practice of Software Development, volume 1: Colloquium* *on Trees in Algebra and Programming (Brighton, U.K.)*, volume 493 of *Lecture Notes in Computer Science*, pages 74–89, Berlin, April 1991. Springer-Verlag.

[CCD93] Anne-Cécile Caron, Jean-Luc Coquidé, and Max Dauchet. Encompassment properties and automata with constraints. In Kirchner [Kir93], pages 328–342.

[CGN01] Hubert Comon, Guillem Godoy, and Robert Nieuwenhuis. The confluence of ground term rewrite systems is decidable in polynomial time. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 298–307, Las Vegas, Nevada, USA, 2001.

[Che81] Paul Chew. Unique normal forms in term rewriting systems with repeated variables. In *Proceedings of the Thirteenth Annual Symposium on Theory of Computing*, pages 7–18. ACM, 1981.

[CHL92] Pierre-Louis Curien, Thérèse Hardin, and Jean-Jacques Lévy. Confluence properties of weak and strong calculi of explicit substitutions. RR 1617, Institut National de Rechereche en Informatique et en Automatique, Rocquencourt, France, February 1992.

[Chr] Jim Christian. Problem corner: An experiment with Grau's ternary Boolean algebra. Submitted.

[Cic90] 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.

[CJ97a] Hubert Comon and Florent Jacquemard. Ground reducibility is EXPTIME-complete. In Winskel [Win97], pages 26–34.

[CJ97b] Hubert Comon and Yan Jurski. Higher-order matching and tree automata. In M. Nielsen and Wolfgang Thomas, editors, *Proceedings of the Conference on Computer Science Logic*, volume 1414 of *Lecture Notes in Computer Science*, pages 157–176, Aarhus, Denmark, August 1997. Springer-Verlag.

[CJ03] Hubert Comon and Florent Jacquemard. Ground reducibility is EXPTIME-complete. *Information and Computation*, 187(1):123–153, 2003.

[CK02] Bruno Courcelle and Teodor Knapik. The evaluation of first-order substitution is monadic second-order compatible. *Theoretical Computer Science*, 281(1–2):177–206, June 2002.

[CL87] Ahlem Ben Cherifa and Pierre Lescanne. Termination of rewriting systems by polynomial interpretations and its implementation. *Science of Computer Programming*, 9(2):137–159, October 1987.

[CNNR98] Hubert Comon, Paliath Narendran, Robert Nieuwenhuis, and Michaël Rusinowitch. Decision problems in ordered rewriting. In Vaughan Pratt, editor, *Thirteenth IEEE Annual Symposium on Logic in Computer Science*, Indianapolis, IN, June 1998. IEEE.

[Com88] Hubert Comon. *Unification et Disunification: Théorie et Applications*. PhD thesis, Institut National Polytechnique de Grenoble, 1988. In French.

[Com90] Hubert Comon. Solving inequations in term algebras (Preliminary version). In Mitchell [Mit90], pages 62–69.

[Com91] Hubert Comon. Completion of rewrite systems with membership constraints. Rapport de Recherche 699, L.R.I., Université de Paris-Sud, September 1991.

[Com93a] Hubert Comon. Complete axiomatizations of some quotient term algebras. *Theoretical Computer Science*, 118(2), September 1993.

[Com93b] Hubert Comon. Personal communication, 1993.

[Com97] Hubert Comon, editor. *8th International Conference on Rewriting Techniques and Applications*, volume 1232 of *Lecture Notes in Computer Science*, Barcelona, Spain, June 1997. Springer-Verlag.

[Com98a] Hubert Comon. Completion of rewrite systems with membership constraints. Part I: Deduction rules. *Journal of Symbolic Computation*, 25:397–419, 1998.

[Com98b] Hubert Comon. Completion of rewrite systems with membership constraints. Part II: Constraint solving. *Journal of Symbolic Computation*, 25:421–453, 1998.

[Con92] Evelyne Contejean. *Eléments pour la Décidabilité de l'Unification modulo la Distributivité*. PhD thesis, Université Paris-Sud, Orsay, France, April 1992.

[Con93a] Evelyne Contejean. A partial solution for D-unification based on a reduction to AC1-unification. In *Proceedings of the EATCS International Conference on Automata, Languages and Programming*, pages 621–632, Lund, Sweden, July 1993. Springer-Verlag.

[Con93b] Evelyne Contejean. Solving linear Diophantine constraints incrementally. In David S. Warren, editor, *Proceedings of the Tenth International Conference on Logic Programming*, Logic Programming, pages 532–549, Budapest, Hungary, June 1993. The MIT Press.

[Cou94] Bruno Courcelle. Monadic-second order graph transductions: A survey. *Theoretical Computer Science*, 126:53–75, 1994.

[Cou97] Bruno Courcelle. The expression of graph properties and graph transformations in monadic second-order logic. In G. Rozenberg, editor, *Handbook of graph grammars and computing by graph transformations, vol. 1: Foundations*, chapter 5, pages 313–400. World Scientific, New-Jersey, London, 1997.

[CP94a] W. Charatonik and L. Pacholski. Negative set constraints with equality. In Abramsky [Abr94], pages 128–136.

[CP94b] W. Charatonik and L. Pacholski. Set constraints with projections are in NEXPTIME. In *Proceedings of the* 35^{th} *Symposium on Foundations of Computer Science*, pages 642–653, 1994.

[CRRT96] C. M. Campbell, E. F. Robertson, N. Ruškuc, and M. R. Thomas. On subsemigroups and ideals in free products of semigroups. *Int. J. of Algebra and Computation*, 6:571–591, 1996.

[CSTT99] Anne-Cécile Caron, Franck Seynhaeve, Sophie Tison, and Marc Tommasi. Deciding the satisfiability of quantifier free formulae on one-step rewriting. In Narendran and Rusinowitch [NR99], pages 103–117.

[CT97] Hubert Comon and Ralf Treinen. The first-order theory of lexicographic path orderings is undecidable. *Theoretical Computer Science*, 176(1–2):67–87, April 1997.

[CW91] D. Cohen and Phil Watson. An efficient representation of arithmetic for term rewriting. In Book [Boo91], pages 240–251.

[Dau89] Max Dauchet. Simulation of Turing machines by a left-linear rewrite rule. In Dershowitz [Der89], pages 109–120.

[Dau92] Max Dauchet. Simulation of Turing machines by a regular rewrite rule. *Theoretical Computer Science*, 103(2):409–420, 1992.

[de 90] Roel C. de Vrijer. Unique normal forms for Combinatory Logic with parallel conditional, a case study in conditional rewriting. Technical report, Free University, Amsterdam, 1990.

[Der87] Nachum Dershowitz. Termination of rewriting. *Journal of Symbolic Computation*, 3(1&2):69–115, February/April 1987. Corrigendum: *4*, 3 (December 1987), 409–410; reprinted in *Rewriting Techniques and Applications*, J.-P. Jouannaud, ed., pp. 69—115, Academic Press, 1987.

[Der89] Nachum Dershowitz, editor. *Rewriting Techniques and Applications*, volume 355 of *Lecture Notes in Computer Science*, Chapel Hill, NC, USA, April 1989. Springer-Verlag.

[Der97] Nachum Dershowitz. Innocuous constructor-sharing combinations. In Comon [Com97], pages 202–216.

[Der05] Nachum Dershowitz. Open. Closed. Open. In Jürgen Giesl, editor, *16th International Conference on Rewriting Techniques*, volume 3467 of *Lecture Notes in Computer Science*, Nara, Japan, April 2005. Springer-Verlag.

[Dev91] Hervé Devie. *Une approche algébrique de la réécriture de preuves équationnelles et son application à la dérivation de procédures de complétion.* PhD thesis, Université Paris-Sud, Orsay, France, October 1991.

[Dev93] Phillipe Devienne. Personal communication, 1993.

[dG00a] Philippe de Groote. Linear higher-order matching is NP-complete. In Bachmair [Bac00], pages 127–140.

[DG00b] Dan Dougherty and Claudio Gutiérrez. Normal forms and reduction for theories of binary relations. In Bachmair [Bac00], pages 95–109.

[DH95] Nachum Dershowitz and Charles Hoot. Natural termination. *Theoretical Computer Science*, 142:170–207, 1995.

[DHLT90] Max Dauchet, Thierry Heuillard, Pierre Lescanne, and Sophie 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.

[Die02] Volker Diekert. Makanin's algorithm. In M. Lothaire, editor, *Algebraic Combinatorics on Words*, chapter 12, pages 387–442. Cambridge University Press, 2002.

[DJ90] Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Methods and Semantics, chapter 6, pages 243–320. North-Holland, Amsterdam, 1990.

[DKP91] Nachum Dershowitz, Stéphane Kaplan, and David A. Plaisted. Rewrite, rewrite, rewrite, rewrite, rewrite,.... *Theoretical Computer Science*, 83(1):71–96, 1991.

[DKR94] E. Domenjoud, F. Klay, and C. Ringeissen. Combination techniques for non-disjoint equational theories. In A. Bundy, editor, *12th International Conference on Automated Deduction*, volume 814 of *Lecture Notes in Computer Science*, pages 267–281, Nancy, France, 1994. Springer-Verlag.

[DL94] Nachum Dershowitz and N. Lindenstrauss, editors. *Proceedings of the Fourth International Workshop on Conditional Rewriting Systems*, Jerusalem, Israel, July 1994. Springer-Verlag.

[DMT85] Nachum Dershowitz, Leo Marcus, and Andrzej 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.

[DO90] Nachum Dershowitz and Mitsuhiro Okada. A rationale for conditional equational programming. *Theoretical Computer Science*, 75:111–138, 1990.

[Dou91] Daniel Dougherty. Adding algebraic rewriting to the untyped lambda calculus (extended abstract). In Book [Boo91], pages 37–48.

[Dow93] Gilles Dowek. Third order matching is decidable. *Annals of Pure and Applied Logic*, 1993.

[DP86] Nachum Dershowitz and David Plaisted. Equational programming. *Machine Intelligence*, 11:21–56, 1986.

[DT90] Max Dauchet and Sophie Tison. The theory of ground rewrite systems is decidable. In Mitchell [Mit90], pages 242–248.

[dV89] Roel C. de Vrijer. Extending the lambda calculus with surjective pairing is conservative. In *Fourth Symposium on Logic in Computer Science*, pages 204–215. IEEE, 1989.

[Eps92] D. B. A. Epstein. *Word Processing In Groups*. XJones and Bartlett Publishers, 1992.

[EWZ08] Joerg Endrullis, Johannes Waldmann, and Hans Zantema. Matrix interpretations for proving termination of term rewriting. *Journal of Automated Reasoning*, 40(2-3):195–220, 2008.

[Fag87] François Fages. Associative-commutative unification. *Journal of Symbolic Computation*, 3(3):257–275, June 1987. Previous version in the *Proceedings of the Seventh International Conference on Automated Deduction*, Napa, CA, pp. 194-208 [May 1984].

[Fer93] Maribel Fernández. AC-complement problems: Validity and negation elimination. In Kirchner [Kir93], pages 358–373.

[Fre93] R. Freese. personal communication, 1993.

[Fri85] Laurent Fribourg. A superposition oriented theorem prover. *Theoretical Computer Science*, 35:161, 1985.

[FS90] Peter Freyd and Andre Scedrov. *Categories and Allegories*, volume 39 of *North Holland Mathematical Library*. North-Holland, 1990.

[FZ93] M. C. F. Ferreira and Hans Zantema. Total termination of term rewriting. In Kirchner [Kir93], pages 213–227. Extended version as [FZ96].

[FZ96] M. C. F. Ferreira and Hans Zantema. Total termination of term rewriting. *Applicable Algebra in Engineering, Communication and Computing*, 7(2):133–162, 1996.

[GA01] Jürgen Giesl and Thomas Arts. Verification of Erlang processes by dependency pairs. *Applicable Algebra in Engineering, Communication and Computing*, 12(1,2):39–72, 2001.

[Gan96] Harald Ganzinger, editor. *7th International Conference on Rewriting Techniques and Applications*, volume 1103 of *Lecture Notes in Computer Science*, New Brunswick, NJ, USA, July 1996. Springer-Verlag.

[Gan99] Harald Ganzinger, editor. *16th International Conference on Automated Deduction*, volume 1632 of *Lecture Notes in Artificial Intelligence*, Trento, Italy, July 1999. Springer-Verlag.

[Ger92a] S. M. Gersten. Dehn functions and l1-norms of finite presentations. In G. Baumslag and C. F. Miller III, editors, *Algorithms and Classification in Combinatorial Group Theory*, volume 23 of *Math. Sciences Research Institute Publ.*, pages 195–224. Springer-Verlag, 1992.

[Ger92b] S. M. Gersten. Problems on automatic groups. In G. Baumslag and C. F. Miller III, editors, *Algorithms and Classification in Combinatorial Group Theory*, volume 23 of *Math. Sciences Research Institute Publ.*, pages 225–232. Springer-Verlag, 1992.

[Ges90] Alfons Geser. *Relative Termination.* PhD thesis, qUniversität Passau, Passau, Germany, 1990.

[Ges03] Alfons Geser. Termination of string rewriting rules that have one pair of overlaps. In Nieuwenhuis [Nie03], pages 410–423.

[GGL00] Healfdene Goguen and Jean Goubault-Larrecq. Sequent combinators: A Hilbert system for the lambda calculus. *Mathematical Structures in Computer Science*, 10:1–79, 2000.

[Gil91] R. Gilleron. Decision problems for term rewriting systems and recognizable tree languages. In Christian Choffrut and Matthias Jantzen, editors, *Eighth Symposium on Theoretical Aspects of Computer Science*, volume 480 of *Lecture Notes in Computer Science*, Hamburg, Germany, February 1991. Springer-Verlag.

[GL99] Jean Goubault-Larrecq. Conjunctive types and SKInT. In *Proceedings of the TYPES'98 Workshop*, Lecture Notes in Computer Science. Springer-Verlag, Bad Irsee, Germany, 1999.

[GM87] E. Giovanetti and C. Moiso. Notes on the elimination of conditions. In *Proc. CTRS '87*, LNCS 308, pages 91–97, 1987.

[GMOZ97] Alfons Geser, Aart Middeldorp, Enno Ohlebusch, and Hans Zantema. Relative undecidability in the termination hierarchy of single rewrite rules. In Bidoit and Dauchet [BD97], pages 237–248.

[Gol81] Warren D. Goldfarb. The undecidability of the second-order unification problem. *Theoretical Computer Science*, 13:225–230, 1981.

[Gor90] L. Gordeev. Generalizations of the Kruskal-Friedman theorems. *Journal of Symbolic Logic*, 55(1):157–181, 1990.

[GTSKF04] Jürgen Giesl, René Thiemann, Peter Schneider-Kamp, and S. Falke. Automated termination proofs with AProVE. In Vincent van Oostrom, editor, *15th International Conference on Rewriting Techniques*, volume 3091 of *Lecture Notes in Computer Science*, pages 210–220, Aachen, Germany, June 2004. Springer-Verlag.

[GTSKF06] Jürgen Giesl, René Thiemann, Peter Schneider-Kamp, and Stephan Falke. Mechanizing and improving dependency pairs. *Journal of Automated Reasoning*, 37(3):155–203, 2006.

[GTT93a] Rémy Gilleron, Sophie Tison, and Marc Tommasi. Solving systems of set constraints using tree automata. In Patrice Enjalbert, Alain Finkel, and Klaus W. Wagner, editors, *Symposium on Theoretical Aspects of Computer Science*, volume 665 of *Lecture Notes in Computer Science*, Würzburg, Germany, February 1993. Springer-Verlag.

[GTT93b] Rémy Gilleron, Sophie Tison, and Marc Tommasi. Solving systems of set constraints with negated subset relationships. In *Proc. 34th Symposium on Foundations of Computer Science*, pages 372–380, Palo Alto, CA, November 1993. IEEE.

[GTV04] Guillem Godoy, Ashish Tiwari, and Rakesh Verma. Characterizing Confluence by Rewrite Closure and Right Ground Term Rewrite Systems. *Applicable Algebra in Engineering, Communication and Computing*, 15(1):13–36, June 2004.

[Gut98] C. Gutiérrez. Satisfiability of word equations with constants is in exponential space. In *Proceedings of the* 39^{th} *Symposium on Foundations of Computer Science*, 1998.

[HH93] D. Hofbauer and M. Huber. Computing linearizations using test sets. In Michaël Rusinowitch and Rémy [MRR93], pages 287–301.

[HJ90] Nevin Heintze and Joxan Jaffar. A decision procedure for a class of set constraints. In Mitchell [Mit90], pages 42–51.

[HK96] Miki Hermann and P. G. Kolaitis. Unification algorithms cannot be combined in polynomial time. In McRobbie and Slaney [MS96], pages 246–260. To appear in a special issue of Information and Computation.

[HL78] Gérard Huet and Dallas S. Lankford. On the uniform halting problem for term rewriting systems. Rapport laboria 283, Institut de Recherche en Informatique et en Automatique, Le Chesnay, France, March 1978.

[HL79] Gérard Huet and Jean-Jacques Lévy. Call by need computations in non-ambiguous linear term rewriting systems. Rapport Laboria 359, Institut National de Recherche en Informatique et en Automatique, Le Chesnay, France, August 1979.

[HL91a] Gérard Huet and Jean-Jacques Lévy. Computations in orthogonal rewriting systems, I and II. In Jean-Louis Lassez and Gordon Plotkin, editors, *Computational Logic: Essays in Honor of Alan Robinson*, pages 395–443. The MIT Press, Cambridge, MA, 1991. This is a revision of [HL79].

[HL91b] Gérard Huet and Jean-Jacques Lévy. Computations in orthogonal rewriting systems, II. In Jean-Louis Lassez and Gordon Plotkin, editors, *Computational Logic: Essays in Honor of Alan Robinson*, chapter 12, pages 415–443. The MIT Press, Cambridge, MA, 1991.

[Hoo96] Charles Hoot. *Termination of Non-Simple Rewrite Systems*. PhD thesis, University of Illinois, 1996.

[HP91] B. Hoffmann and D. Plump. Implementing term rewriting by jungle evaluation. *RAIRO Theoretical Informatics and Applications*, 25(5):445–472, 1991.

[HR86] Jieh Hsiang and Michaël Rusinowitch. A new method for establishing refutational completeness in theorem proving. In Jr̈g Siekmann, editor, *8th International Conference on Automated Deduction*, volume 230 of *Lecture Notes in Computer Science*, pages 141–152, Oxford, England, 1986. Springer-Verlag.

[Hsi85] Jieh Hsiang. Refutational theorem proving using term-rewriting systems. *Artificial Intelligence*, 25:255–300, March 1985.

[Hsi95] Jieh Hsiang, editor. *6th International Conference on Rewriting Techniques and Applications*, volume 914 of *Lecture Notes in Computer Science*, Kaiserslautern, Germany, April 1995. Springer-Verlag.

[Hue76] Gérard Huet. *Résolution d'équations dans les langages d'ordre* $1, 2, \ldots, \omega$. Thèse d'état, Université Paris 7, Paris, France, 1976.

[Hue80] Gérard Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *J. of the Association for Computing Machinery*, 27(4):797–821, October 1980.

[Hue81] Gérard Huet. A complete proof of correctness of the Knuth-Bendix completion algorithm. *J. Computer and System Sciences*, 23(1):11–21, 1981.

[Hue90] Gérard Huet, editor. *Logical Foundations of Functional Programming*. University of Texas at Austin Year of Programming. Addison-Wesley, Reading, MA, 1990.

[HW93] Philipp Hanschke and Jörg Würtz. Satisfiability of the smallest binary program. *Information Processing Letters*, 45(5):237–241, April 1993.

[HW05] Dieter Hofbauer and Johannes Waldmann. Termination of $\{aa \to bc, bb \to ac, cc \to ab\}$. Preprint, 2005.

[Int97] Benedetto Intrigila. Non-existent statman's double fixed point combinator does not exist, indeed. *Information and Computation*, 137(1):35–40, 1997.

[Jac96] Florent Jacquemard. *Automates d'arbres et Réécriture de termes*. PhD thesis, Université de Paris-Sud, 1996. In French.

[JK84] Jean-Pierre Jouannaud and Hélène Kirchner. Construction d'un plus petit ordre de simplification. *RAIRO Theoretical Informatics*, 18(3):191–207, 1984.

[JK86] Jean-Pierre Jouannaud and Hélène Kirchner. Completion of a set of rules modulo a set of equations. *SIAM Journal on Computing*, 15:1155–1194, November 1986.

[JK89] Jean-Pierre Jouannaud and Emmanuel Kounalis. Automatic proofs by induction in equational theories without constructors. *Information and Computation*, 81(1):1–33, 1989.

[JK91a] Jean-Pierre Jouannaud and Claude Kirchner. Solving equations in abstract algebras: A rule-based survey of unification. In Jean-Louis Lassez and Gordon Plotkin, editors, *Computational Logic: Essays in Honor of Alan Robinson*. The MIT Press, Cambridge, MA, 1991.

[JK91b] Jean-Pierre Jouannaud and Claude Kirchner. Solving equations in abstract algebras: A rule-based survey of unification. In Jean-Louis Lassez and Gordon Plotkin, editors, *Computational Logic: Essays in Honor of Alan Robinson*. The MIT Press, 1991.

[JM90] Jean-Pierre Jouannaud and Claude Marché. Completion modulo associativity, commutativity and identity. In Alfonso Miola, editor, *Proceedings of the International Symposium on the Design and Implementation of Symbolic Computation Systems (Capri, Italy)*, volume 429 of *Lecture Notes in Computer Science*, pages 111–120, Berlin, April 1990. Springer-Verlag.

[JO91a] Jean-Pierre Jouannaud and Mitsuhiro Okada. Executable higher-order algebraic specification languages. In Kahn [Kah91], pages 350–361.

[JO91b] Jean-Pierre Jouannaud and Mitsuhiro Okada. Satisfiability of systems of ordinal notations with the subterm property is decidable. In J. Leach Albert, B. Monien, and M. Rodríguez Artalejo, editors, *Proceedings of the Eighteenth EATCS Colloquium on Automata, Languages and Programming (Madrid, Spain)*, volume 510 of *Lecture Notes in Computer Science*, pages 455–468, Berlin, July 1991. Springer-Verlag.

[Jou94] Jean-Pierre Jouannaud, editor. *First International Conference on Constraints in Computational Logics*, volume 845 of *Lecture Notes in Computer Science*, München, Germany, September 1994. Springer-Verlag.

[JS94] Jean-Pierre Jouannaud and Walid Sadfi. Strong sequentiality of left-linear overlapping rewrite systems. In Dershowitz and Lindenstrauss [DL94].

[Kah91] Gilles Kahn, editor. *Sixth Symposium on Logic in Computer Science*, Amsterdam, The Netherlands, July 1991. IEEE.

[Kap92] Deepak Kapur, editor. *11th International Conference on Automated Deduction*, volume 607 of *Lecture Notes in Computer Science*, Saratoga Springs, NY, June 1992. Springer-Verlag.

[Kd89] Jan Willem Klop and Roel C. de Vrijer. Unique normal forms for lambda calculus with surjective pairing. *Information and Computation*, 80:97–113, 1989.

[Ken89] J. R. Kennaway. Sequential evaluation strategies for parallel-or and related reduction systems. *Annals of Pure and Applied Logic*, 43:31–56, 1989.

[Kir86] Claude Kirchner. Computing unification algorithms. In *First Symposium on Logic in Computer Science*, pages 206–216, Cambridge, MA, June 1986. IEEE.

[Kir93] Claude Kirchner, editor. *5th International Conference on Rewriting Techniques and Applications*, volume 690 of *Lecture Notes in Computer Science*, Montreal, Canada, June 1993. Springer-Verlag.

[KKSd90] J. R. Kennaway, Jan Willem Klop, M. R. Sleep, and F. J. de Vries. Transfinite reductions in orthogonal term rewriting systems. Technical Report CS-R9041, CWI, Amsterdam, 1990.

[KKSd91] J. R. Kennaway, Jan Willem Klop, M. R. Sleep, and F. J. de Vries. Transfinite reductions in orthogonal term rewriting systems (Extended abstract). In Book [Boo91], pages 1–12.

[Klo80] Jan Willem Klop. *Combinatory Reduction Systems*, volume 127 of *Mathematical Centre Tracts*. Mathematisch Centrum, Amsterdam, 1980.

[Klo92] Jan Willem Klop. Term rewriting systems. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, chapter 1, pages 1–117. Oxford University Press, Oxford, 1992.

[KM89] N. Kuhn and Klaus Madlener. A method for enumerating cosets of a group presented by a canonical system. In *Proc. ISSAC'89*, pages 338–350. ACM, 1989.

[KM91] Jan Willem Klop and Aart Middeldorp. Sequentiality in orthogonal term rewriting systems. *Journal of Symbolic Computation*, 12:161–195, 1991.

[KN85a] Deepak Kapur and Paliath Narendran. An equational approach to theorem proving in first-order predicate calculus. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 1146–1153, Los Angeles, CA, August 1985.

[KN85b] M. S. Krishnamoorthy and P. Narendran. On recursive path ordering. *Theoretical Computer Science*, 40:323–328, 1985.

[KNZ87] Deepak Kapur, Paliath Narendran, and Hantao Zhang. On sufficient completeness and related properties of term rewriting systems. *Acta Informatica*, 24(4):395–415, August 1987.

[KO92] M. Kurihara and A. Ohuchi. Modularity of simple termination of term rewriting systems with shared constructors. *Theoretical Computer Science*, 103:273–282, 1992.

[KP82] Laurie Kirby and Jeff Paris. Accessible independence results for Peano arithmetic. *Bulletin London Mathematical Society*, 14:285–293, 1982.

[KR94] Gregory Kucherov and Michaël Rusinowitch. On the ground reducibility problem for word rewriting systems with variables. *Information Processing Letters*, 1994. To appear. Earlier version appeared in the Proceedings of 1994 ACM/SIGAPP Symposium on Applied Computing, Phoenix, AZ.

[Kru60] Joseph B. Kruskal. Well-quasi-ordering, the Tree Theorem, and Vazsonyi's conjecture. *Transactions of the American Mathematical Society*, 95:210–225, May 1960.

[KT93] Gregory Kucherov and Mohamed Tajine. Decidability of regularity and related properties of ground normal form languages. In Michaël Rusinowitch and Rémy [MRR93], pages 272–286.

[KT95] Gregory Kucherov and Mohamed Tajine. Decidability of regularity and related properties of ground normal form languages. *Information and Computation*, 118(1):91–100, 1995.

[Kuc91] G. Kucherov. On relationship between term rewriting systems and regular tree languages. In Book [Boo91].

[Kur90] W. Kurth. *Termination und Konfluenz von Semi-Thue-Systems mit nur einer Regel.* PhD thesis, Technische Universitat Clausthal, Clausthal, Germany, 1990. In German.

[Kř89] I. Kříž. Well-quasiordering finite trees with gap-condition. proof of Harvey Friedman's conjecture. *Ann. of Math.*, 130:215–226, 1989.

[KV00] Konstantin Korovin and Andrei Voronkov. A decision procedure for the existential theory of term algebras with the Knuth-Bendix ordering. In *15th Annual IEEE Symposium on Logic in Computer Science*, pages 291–302, Santa Barbara, CA, USA, June 2000. IEEE.

[KV01] Konstantin Korovin and Andrei Voronkov. Knuth-Bendix constraint solving is NP-complete. In Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen, editors, *28th International Colloquium on Automata, Languages and Programming*, volume 2076 of *Lecture Notes in Computer Science*, pages 979–992, Crete, Greece, July 2001. Springer-Verlag.

[KV02a] Konstantin Korovin and Andrei Voronkov. The decidability of the first-order theory of the Knuth-Bendix orders in the case of unary signatures. In *Foundations of Software Technology and Theoretical Computer Science*, Lecture Notes in Computer Science, Kanpur, India, December 2002. Springer-Verlag. A preliminary report of the result appeared in [KV02b].

[KV02b] Konstantin Korovin and Andrei Voronkov. The decidability of the first-order theory of the Knuth-Bendix orders in the case of unary signatures. In Christophe Ringeissen, Cesare Tinelli, Ralf Treinen, and Rakesh M. Verma, editors, *Proceedings of the 16th*

International Workshop on Unification (UNIF 2002), Technical Report 02-05, Department of Computer Science, University of Iowa, pages 45–46, Copenhagen, Denmark, July 2002.

[LA96] Jordi Levy and Jaume Agustí. Bi-rewriting systems. *Journal of Symbolic Computation*, 22(3):279–314, September 1996.

[Lan79] Dallas S. Lankford. On proving term rewriting systems are Noetherian. Memo MTP-3, Mathematics Department, Louisiana Tech. University, Ruston, LA, May 1979. Revised October 1979.

[Lep01] Ingo Lepper. *Simplification Orders in Term Rewriting*. PhD thesis, Universität Münster, Germany, 2001. Available at `http://wwwmath.uni-muenster.de/logik/publ/diss/9.html`.

[Lep04] Ingo Lepper. Simply terminating rewrite systems with long derivations. *Archive for Mathematical Logic*, 43(1):1–18, 2004.

[Les90] Pierre Lescanne. Well rewrite orderings. In Mitchell [Mit90], pages 239–256.

[Les92] Pierre Lescanne. Termination of rewrite systems by elementary interpretations. In Hélène Kirchner and Giorgio Levi, editors, *3th International Conference on Algebraic and Logic Programming*, volume 632 of *Lecture Notes in Computer Science*, pages 21–36, Volterra, Italy, September 1992. Springer-Verlag.

[Les94] Pierre Lescanne. On termination of one rule rewrite systems. *Theoretical Computer Science*, 132(1–2):395–401, September 1994.

[Lev96] Jordi Levy. Linear second-order unification. In Ganzinger [Gan96], pages 332–346.

[LM87] J.-L. Lassez and K. G. Marriott. Explicit representation of terms defined by counter examples. *Journal of Automated Reasoning*, 3(3):1–17, September 1987.

[LM93] D. Lugiez and J.-L. Moysset. Complement problems and tree automata in AC-like theories. In *Proceedings of the Symposium on Theoretical Aspects of Computer Science (Würzburg, Germany)*, Lecture Notes in Computer Science, Berlin, 1993. Springer-Verlag.

[LMSar] Igor Litovski, Yves Métivier, and Eric Sopena. Definitions and comparisons of local computations on graphs. *Mathematical Systems Theory*, to appear. Available as internal report 91-43 of LaBRI, University of Bordeaux 1.

[LR99] Sébastien Limet and Pierre Réty. A new result about the decidability of the existential one-step rewriting theory. In Narendran and Rusinowitch [NR99], pages 118–132.

[LRD94] Pierre Lescanne and J. Rouyer-Degli. The calculus of explicit substitutions $\lambda\upsilon$. Technical Report RR-2222, INRIA-Lorraine, January 1994.

[LV00] Jordi Levy and Mateu Villaret. Linear second-order unification and context unification with tree-regular constraints. In Bachmair [Bac00], pages 156–171.

[LV02] Jordi Levy and Mateu Villaret. Currying second-order unification problems. In Tison [Tis02], pages 326–339.

[Mah88] Michael J. Maher. Complete axiomatizations of the algebras of the finite, rational and infinite trees. In Yuri Gurevich, editor, *Third Symposium on Logic in Computer Science*, pages 348–357, Edinburgh, Scotland, July 1988. IEEE.

[Mak77] G. S. Makanin. The problem of solvability of equations in a free semi-group. *Math. USSR Sbornik*, 32(2):129–198, 1977.

[Man93] Ken Mano, September 1993. Personal communication.

[Mar94] M. Marchiori. Modularity of $\mathrm{UN}^{\rightarrow}$ for left-linear term rewriting systems. Technical report, CWI, Amsterdam, 1994.

[Mar96a] Claude Marché. Normalized rewriting: an alternative to rewriting modulo a set of equations. *Journal of Symbolic Computation*, 21(3):253–288, March 1996.

[Mar96b] M. Marchiori. Unravelings and ultra-properties. In *Proc. ALP '96*, LNCS 1139, pages 107–121, 1996.

[Mar97] Jerzy Marcinkowski. Undecidability of the first order theory of one-step right ground rewriting. In Comon [Com97], pages 241–253.

[Mar99] Jerzy Marcinkowski. Undecidability of the $\exists^*\forall^*$ part of the theory of ground term algebra modulo an AC symbol. In Narendran and Rusinowitch [NR99], pages 92–102.

[Mel95] Paul-André Melliès. Typed λ-calculi with explicit substitutions may not terminate. In *Typed Lambda Calculi and Applications*, 1995.

[Mel96] Paul-André Melliès. *Description Abstraite des Systèmes de Réécriture.* Thèse de doctorat, Université Paris VII, 20 Décembre 1996.

[Mét85] Yves Métivier. Calcul de longueurs de chaînes de réécriture dans le monoïde libre. *Theoretical Computer Science*, 35(1):71–87, January 1985.

[MG95] Aart Middeldorp and Bernhard Gramlich. Simple termination is difficult. *Applicable Algebra in Engineering, Communication and Computing*, 6(2):115–128, 1995.

[Mid89] Aart Middeldorp. Modular aspects of properties of term rewriting systems related to normal forms. In Dershowitz [Der89], pages 263–277.

[Mid90] Aart Middeldorp. *Modular Properties of Term Rewriting Systems.* PhD thesis, Vrije Universiteit, Amsterdam, The Netherlands, 1990.

[Mid93] Aart Middeldorp. Modular properties of conditional term rewriting systems. *Information and Computation*, 104(1):110–158, 1993.

[Mil91] Dale Miller. A logic programming language with lambda-abstraction, function variables, and simple unification. In P. Schroeder-Heister, editor, *Extensions of Logic Programming*, volume 690 of *Lecture Notes in Computer Science.* Springer-Verlag, 1991.

[Mit90] John C. Mitchell, editor. *Fifth Symposium on Logic in Computer Science*, Philadelphia, PA, June 1990. IEEE.

[MO94] K. Mano and M. Ogawa. A new proof of Chew's theorem. Technical report, IPSJ PRG94-19-7, 1994.

[Mos09] Georg Moser. The Hydra battle and Cichon's principle. *Applicable Algebra in Engineering, Communication and Computing*, 20(2):133–158, 2009. doi:10.1007/s00200-009-0094-4.

[MR84] A. Martelli and G. Rossi. Efficient unification with infinite terms in logic programming. In *International conference on fifth generation computer systems*, pages 202–209, 1984.

[MRR93] M. Michaël Rusinowitch and J. L. Rémy, editors. *Proceedings of the Third International Workshop on Conditional Rewriting Systems*, volume 656 of *Lecture Notes in Computer Science*, Pont-à-Mousson, France, January 1993. Springer-Verlag.

[MS96] M. A. McRobbie and J. K. Slaney, editors. *13th International Conference on Automated Deduction*, Lecture Notes in Computer Science, New Brunswick, NJ, July/August 1996. Springer-Verlag.

[MSA88] Jürgen Müller and Rolf Socher-Ambrosius. Topics in completion theorem proving. SEKI-Report SR-88-13, Fachbereich Informatik, University of Kaiserslautern, Kaiserslautern, West Germany, 1988. To appear in *J. of Symbolic Computation*.

[MSW08] Georg Moser, Andreas Schnabl, and Johannes Waldmann. Complexity analysis of term rewriting based on matrix and context dependent interpretations. In Ramesh Hariharan, Madhavan Mukund, and V Vinay, editors, *Proceedings of the IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2008)*, volume 2 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 304–315. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2008.

[MT91] Aart Middeldorp and Yoshihito Toyama. Completeness of combinations of constructor systems. In Book [Boo91], pages 174–187.

[Mül92] Fritz Müller. Confluence of the lambda calculus with left-linear algebraic rewriting. *Information Processing Letters*, 41:293–299, April 1992.

[Muñ96] César Muñoz. Confluence and preservation of strong normalisation in an explicit substitutions calculus (extended abstract). In Edmund C. Clarke, editor, *Eleventh Symposium on Logic in Computer Science*, pages 440–447, New Brunswick, NJ, July 1996. IEEE.

[Nie93a] Robert Nieuwenhuis. A precedence-based total AC-compatible ordering. In Kirchner [Kir93], pages 374–388.

[Nie93b] Robert Nieuwenhuis. Simple LPO constraint solving methods. *Information Processing Letters*, 47:65–69, 1993.

[Nie03] Robert Nieuwenhuis, editor. *14th International Conference on Rewriting Techniques*, volume 2706 of *Lecture Notes in Computer Science*, Valencia, Spain, June 2003. Springer-Verlag.

[Nip91] Tobias Nipkow. Higher-order critical pairs. In Kahn [Kah91], pages 342–349.

[Nip98] Tobias Nipkow, editor. *9th International Conference on Rewriting Techniques and Applications*, volume 1379 of *Lecture Notes in Computer Science*, Tsukuba, Japan, April 1998. Springer-Verlag.

[NO90] Robert Nieuwenhuis and Fernando Orejas. Clausal rewriting. In S. Kaplan and M. Okada, editors, *Extended Abstracts of the Second International Workshop on Conditional and Typed Rewriting Systems*, pages 81–88, Montreal, Canada, June 1990. Concordia University. Revised version to appear in *Lecture Notes in Computer Science*, Springer-Verlag, Berlin.

[NPR97] Joachim Niehren, Manfred Pinkal, and Peter Ruhrberg. On equality up-to constraints over finite trees, context unification and one-step rewriting. In William McClune, editor, *14th International Conference on Automated Deduction*, volume 1249 of *Lecture Notes in Artificial Intelligence*, pages 34–48, Townsville, Australia, July 1997. Springer-Verlag.

[NR99] Paliath Narendran and Michael Rusinowitch, editors. *10th International Conference on Rewriting Techniques and Applications*, volume 1631 of *Lecture Notes in Computer Science*, Trento, Italy, July 1999. Springer-Verlag.

[NZM10] Friedrich Neurauter, Harald Zankl, and Aart Middeldorp. Revisiting matrix interpretations for polynomial derivational complexity of term rewriting. In *17th International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, volume 6397 of *ARCoSS*, October 2010. To appear.

[OCM00] Enno Ohlebusch, Claus Claves, and Claude Marché. TALP: A tool for the termination analysis of logic programs. In Bachmair [Bac00], pages 270–273.

[O'D85] Michael J. O'Donnell. *Equational Logic as a Programming Language.* The MIT Press, Cambridge, MA, 1985.

[Oga95] M. Ogawa. Simple gap termination on term graph rewriting systems. In *Theory of Rewriting Systems and Its Application*, pages 99–108, 1995. Research report 918, RIMS, Kyoto Univ, Kyoto., also available at http://www.brl.ntt.co.jp/people/mizuhito.

[Ohl94a] Enno Ohlebusch. On the modularity of confluence of constructor-sharing term rewriting systems. In *19th Colloquium on Trees in Algebra and Programming*, volume 787 of *LNCS*, pages 261–275. Springer-Verlag, 1994.

[Ohl94b] Enno Ohlebusch. On the modularity of confluence of constructor-sharing term rewriting systems. In Sophie Tison, editor, *Colloquium on Trees in Algebra and Programming*, volume 787 of *Lecture Notes in Computer Science*, Edinburgh, Scotland, 1994. Springer-Verlag.

[Ohl97] Enno Ohlebusch. Conditional term graph rewriting. In *6th International Conference on Algebraic and Logic Programming*, volume 1298 of *Lecture Notes in Computer Science*, pages 144–158. Springer-Verlag, 1997.

[Ohl01] Enno Ohlebusch. Termination of logic programs: Transformational methods revisited. *Applicable Algebra in Engineering, Communication and Computing*, 12(1,2):73–116, 2001.

[Ohs01] Hitoshi Ohsaki. Beyond regularity: Equational tree automata for associative and commutative theories. In Laurent Fribourg, editor, *15th International Workshop on Computer Science Logic*, volume 2142 of *Lecture Notes in Computer Science*, pages 539–553, Paris, France, September 2001. Springer-Verlag.

[Oku98] Satoshi Okui. Simultaneous critical pairs and Church-Rosser property. In Nipkow [Nip98], pages 2–16.

[OO89] M. Ogawa and S. Ono. On the uniquely converging property of nonlinear term rewriting systems. Technical report, IEICE COMP89-7, 1989.

[OO93] Michio Oyamaguchi and Yoshikatsu Ohta. On the confluent property of right-ground term rewriting systems. *Trans. IEICE*, J76-D-I:39–45, 1993.

[OO97] Michio Oyamaguchi and Yoshikatsu Ohta. A new parallel closed condition for Church-Rosser of left-linear term rewriting systems. In Comon [Com97], pages 187–201.

[Oos92] V. van Oostrom. Confluence by decreasing diagrams. IR 298, Vrije Universiteit, Amsterdam, The Netherlands, August 1992. To appear in *Theoretical Computer Science*.

[Oos94] Vincent van Oostrom. *Confluence for Abstract and Higher-Order Rewriting*. PhD thesis, Vrije Universiteit, Amsterdam, 1994.

[Oos97a] Vincent van Oostrom. Developing developments. *Theoretical Computer Science*, 175:159–181, 1997.

[Oos97b] Vincent van Oostrom. Finite family developments. In Comon [Com97], pages 308–322.

[OOTar] Yoshikatsu Ohta, Michio Oyamaguchi, and Yoshihito Toyama. On the Church-Rosser property of simple-right-linear term rewriting systems. *Trans. IEICE*, to appear.

[OSKM98] Friedrich Otto, Andrea Sattler-Klein, and Klaus Madlener. Automatic monoids versus monoids with finite convergent presentations. In Nipkow [Nip98], pages 32–46.

[OT02] Hitoshi Ohsaki and Toshinori Takai. Decidability and closure properties of equational tree languages. In Tison [Tis02], pages 114–128.

[Oto12] Jan Otop. E-unification with constants vs. general E-unification. *Journal of Automated Reasoning*, 48(3):363–390, March 2012.

[OTTR05] Hitoshi Ohsaki, Jean-Marc Talbot, Sophie Tison, and Yves Roos. Monotone AC-tree automata. In Geoff Sutcliffe and Andrei Voronkov, editors, *12th International Conference on Logic Programming and Automated Reasoning*, volume 3835 of *Lecture Notes in Artificial Intelligence*, pages 337–351, Montego Bay, Jamaica, December 2005. Springer-Verlag.

[Oya87] Michio Oyamaguchi. The Church-Rosser property for ground term rewriting systems is decidable. *Theoretical Computer Science*, 49(1):43–79, 1987.

[Oya90] Michio Oyamaguchi. On the word problem for right-ground term-rewriting systems. In *Trans. IEICE*, volume E73-5, pages 718–723, 1990.

[Pad96] Vincent Padovani. *Filtrage d'ordre supérieure*. Phd thesis, Université de Paris 7, Paris, France, 1996.

[Pla85] David A. Plaisted. Semantic confluence tests and completion methods. *Information and Control*, 65(2/3):182–215, May/June 1985.

[Pla99] Wojciech Plandowski. Satisfiability of word equations with constants is in PSPACE. In *Proceedings of the* 40^{th} *Symposium on Foundations of Computer Science*, pages 495–500, New York, NY, USA, October 1999. IEEE.

[Plu91] D. Plump. Implementing term rewriting by graph reduction: Termination of combined systems. In S. Kaplan and M. Okada, editors, *Proceedings of the Second International Workshop on Conditional and Typed Rewriting Systems (Montreal, Canada, June 1990)*, volume 516 of *Lecture Notes in Computer Science*, pages 307–317, Berlin, 1991. Springer-Verlag.

[Plu93] D. Plump. Hypergraph rewriting: Critical pairs and undecidability of confluence. In M. R. Sleep, M. J. Plasmeijer, and M. C. van Eekelen, editors, *Term Graph Rewriting: Theory and Practice*, chapter 15. Wiley, 1993. To appear.

[PS81] Gerald E. Peterson and Mark E. Stickel. Complete sets of reductions for some equational theories. *J. of the Association for Computing Machinery*, 28(2):233–264, April 1981.

[QW94] Zheniu Qian and Kang Wang. Modular AC-unification of higher-order patterns. In Jouannaud [Jou94], pages 105–120.

[Raa96] Femke van Raamsdonk. *Confluence and Normalization for Higher-Order Rewriting*. PhD thesis, Vrije Universiteit, Amsterdam, 1996.

[Rao95] M.R.K. Krishna Rao. Semi-completeness of hierarchical and super-hierarchical combinations of term rewriting systems. In *Theory and practice of Software Development, TAPSOFT'95*, volume 915 of *Lecture Notes in Computer Science*, pages 379–393. Springer-Verlag, 1995.

[Rao98] M.R.K. Krishna Rao. Modular aspects of term graph rewriting. *Theoretical Computer Science*, 208(1–2):59–86, 1998. Special issue on Rewriting Techniques and Applications conference RTA'96.

[Rin92] Christophe Ringeissen. Unification in a combination of equational theories with shared constants and its application to primal algebras. In A. Voronkov, editor, *International Conference on Logic Programming and Automated Reasoning*, volume 624 of *Lecture Notes in Artificial Intelligence*, St. Petersburg, Russia, July 1992. Springer-Verlag.

[RS] Neil Robertson and P. D. Seymour. Graph minors IV. Treewidth and well-quasi-ordering. Submitted 1982; revised January 1986.

[RS96] Neil Robertson and P. Seymour. Graph minors 23: Nash-williams' immersion conjecture. Preprint, March 1996.

[Rub94] Albert Rubio. *Automated deduction with constrained clauses.* PhD thesis, Univ. de Catalunya, 1994.

[RV80] Jean-Claude Raoult and Jean Vuillemin. Operational and semantic equivalence between recursive programs. *J. of the Association for Computing Machinery*, 27(4):772–796, October 1980.

[Sce92] Andre Scedrov, editor. *Seventh Symposium on Logic in Computer Science*, Santa Cruz, CA, June 1992. IEEE.

[Sch97] Aleksy Schubert. Linear interpolation for the higher-order matching problem. In Bidoit and Dauchet [BD97], pages 441–452.

[Sch02] Kai Scheffermann. A countererexample to the modularity of unicity of normal forms for join conditional term rewriting systems. Preprint pr309, Universität Hannover, Institut für Mathematik, April 2002.

[Sim85] S.G. Simpson. Nonprovability of certain combinatorial properties of finite trees. In L. A. Harrington, editor, *Harvey Friedman's research on the Foundation of Mathematics*, pages 87–117. Elsevier Science, 1985.

[Sim04] Jakob Grue Simonsen. On confluence and residuals in Cauchy convergent transfinite rewriting. *Information Processing Letters*, 91(3):141–146, August 2004.

[Siv89] G. Sivakumar. *Proofs and Computations in Conditional Equational Theories*. PhD thesis, Department of Computer Science, University of Illinois, Urbana, IL, May 1989. Report R-89-1642.

[SK94] Andrea Sattler-Klein. About changing the ordering during Knuth-Bendix completion. In *Proceedings of the Symposium on Theoretical Aspects of Computer Science*, pages 175–186, 1994.

[Sny91] Wayne Snyder. A note on the complexity of simplification orderings. Technical Report TR 90–009, Boston University, Boston, MA, 1991.

[Squ87] Craig Squier. Word problems and a homological finiteness condition for monoids. *J. of Pure and Applied Algebra*, 49:201–217, 1987.

[SS89] Manfred Schmidt-Schauß. Combination of unification algorithms. *Journal of Symbolic Computation*, 8, 1989.

[SS94a] Manfred Schmidt-Schauß. Unification of stratified second-order terms. Research Report 12/94, Fachbereich Informatik, Universität Frankfurt, Germany, December 1994.

[SS94b] Manfred Schmidt-Schauß. Unification of stratified second-order terms. Internal Report 12/94, Johann-Wolfgang-Goethe-Universität, Frankfurt, Germany, 1994.

[SS97] Manfred Schmidt-Schauß. A unification algorithm for distributivity and a multiplicative unit. *Journal of Symbolic Computation*, 22(3):315–344, 1997.

[SSS98] Manfred Schmidt-Schauß and Klaus Schulz. On the exponent of peridicity of minimal solutions of context equations. In Nipkow [Nip98], pages 61–75.

[SSS99a] Manfred Schmidt-Schauß and Klaus Schulz. Solvability of context unification with two context variables is decidable. In Ganzinger [Gan99], pages 67–81. Complete version as [SSS99b].

[SSS99b] Manfred Schmidt-Schauß and Klaus Schulz. Solvability of context unification with two context variables is decidable. CIS-Report 98-114, CIS, Universität München, München, Germany, 1999.

[Sta91] Richard Statman. There is no hyperrecurrent S,K combinator. Research Report 91–133, Department of Mathematics, Carnegie Mellon University, Pittsburgh, PA, 1991.

[Sta92] Richard Statman. A short note on a problem of Ray Smullyan. Technical report, INRIA-Rocquencourt, 1992.

[Sta00] Rick Statman. On the word problem for combinators. In Bachmair [Bac00], pages 203–213.

[Sti06] Colin Stirling. A game-theoretic approach to deciding higher-order matching. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP (2)*, volume 4052 of *Lecture Notes in Computer Science*, pages 348–359, Venice, Italy, July 2006. Springer-Verlag.

[Stø05] Kristian Støvring. Extending the extensional lambda calculus with surjective pairing is conservative. Research Report BRICS RS-05-35, DAIMI, Department of Computer Science, University of Aarhus, Aarhus, Denmark, November 2005.

[Stø06] Kristian Støvring. Extending the extensional lambda calculus with surjective pairing is conservative. *Logical Methods in Computer Science*, 2(2:1):1–14, 2006.

[STT97] Franck Seynhaeve, Marc Tommasi, and Ralf Treinen. Grid structures and undecidable constraint theories. In Bidoit and Dauchet [BD97], pages 357–368.

[STTT01] Franck Seynhaeve, Sophie Tison, Marc Tommasi, and Ralf Treinen. Grid structures and undecidable constraint theories. *Theoretical Computer Science*, 258(1–2):453–490, May 2001.

[TA87] Erik Tiden and Stefan Arnborg. Unification problems with one-sided distributivity. *Journal of Symbolic Computation*, 3:183–202, 1987.

[Taj93] Mohamed Tajine. Negation elimination for syntactic equational formula. In Kirchner [Kir93], pages 316–327.

[Tak93] Masako Takahashi. λ-calculi with conditional rules. In M. Bezem and J. F. Groote, editors, *Proceedings of the International Conference on Typed Lambda Calculi and Applications (Utrecht, The Netherlands)*, volume 664 of *Lecture Notes in Computer Science*, pages 406–417, Berlin, 1993. Springer-Verlag.

[Tha87] S. Thatte. A refinement of strong sequentiality for term rewriting with constructors. *Information and Computation*, 72:46–65, 1987.

[Tho97] Wolfgang Thomas. Automata theory on trres and partial orders. In *Theory and Practice of Software Development*, volume 1214 of *Lecture Notes in Computer Science*, pages 20–38, 1997.

[Tis90] Sophie Tison. Automates comme outil de décision dans les arbres. Dossier d'habilitation à diriger des recherches, December 1990. In French.

[Tis02] Sophie Tison, editor. *Rewriting Techniques and Applications*, volume 2378 of *Lecture Notes in Computer Science*, Copenhagen, Denmark, July 2002. Springer-Verlag.

[Tiw02] Ashish Tiwari. Deciding confluence of certain term rewriting systems in polynomial time. In Gordon Plotkin, editor, *IEEE Symposium on Logic in Computer Science, LICS 2002*, pages 447–456, Copenhagen, Denmark, July 2002. IEEE.

[TO94] Yoshihito Toyama and Michio Oyamaguchi. Church-Rosser property and unique normal form property of non-duplicating term rewriting systems. In Dershowitz and Lindenstrauss [DL94].

[Tou97] Hélène Touzet. *Propriétés combinatoires pour la terminaison de systèmes de réécriture.* PhD thesis, Université Henri Poincaré – Nancy 1, Nancy, France, September 1997. In French.

[Tou98] Hélène Touzet. Encoding the Hydra battle as a rewrite system. In Lubos Brim, Jozef Gruska, and Jirí Zlatuska, editors, *Mathematical Foundations of Computer Science*, volume 1450 of *Lecture Notes in Computer Science*, pages 267–276, Brno, Czech Republic, August 1998. Springer-Verlag.

[Tou99] Hélène Touzet. A characterization of multiply recursive functions with Higman's lemma. In Narendran and Rusinowitch [NR99], pages 163–174.

[Toy87] Yoshihito Toyama. On the Church-Rosser property for the direct sum of term rewriting systems. *J. of the Association for Computing Machinery*, 34(1):128–143, January 1987.

[Toy88] Yoshihito Toyama. Commutativity of term rewriting systems. In K. Fuchi and L. Kott, editors, *Programming of Future Generation Computers II*, pages 393–407. North-Holland, 1988.

[Toy92] Yoshihito Toyama. Strong sequentiality of left linear overlapping term rewriting systems. In Scedrov [Sce92].

[Tre90] Ralf Treinen. A new method for undecidability proofs of first order theories. In K. V. Nori and C. E. Veni Madhavan, editors, *Proceedings of the Tenth Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 472 of *Lecture Notes in Computer Science*, pages 48–62. Springer-Verlag, 1990.

[Tre92] Ralf Treinen. A new method for undecidability proofs of first order theories. *Journal of Symbolic Computation*, 14(5):437–458, November 1992.

[Tre96] Ralf Treinen. The first-order theory of one-step rewriting is undecidable. In Ganzinger [Gan96], pages 276–286.

[Tre98] Ralf Treinen. The first-order theory of linear one-step rewriting is undecidable. *Theoretical Computer Science*, 208(1–2):179–190, November 1998.

[Ven87] K. N. Venkataraman. Decidability of the purely existential fragment of the theory of term algebras. *J. of the Association for Computing Machinery*, 34(2):492–510, 1987.

[Ver95] Rakesh M. Verma. Unique normal forms and confluence for rewrite systems. In *Int'l Joint Conf. on Artificial Intelligence*, pages 362–368, 1995.

[Ver96a] Rakesh M. Verma. Unicity and modularity of confluence for term rewriting systems. Technical report, University of Houston, 1996.

[Ver96b] Rakesh M. Verma. Unicity and modularity of confluence for term rewriting systems. Technical report, University of Houston, 1996.

[Ver97] Rakesh M. Verma. Unique normal forms for nonlinear term rewriting systems: Root overlaps. In *Symp. on Fundamentals of Computation Theory*, volume 1279 of *Lecture Notes in Computer Science*, pages 452–462. Springer-Verlag, September 1997.

[VG92] S. Vágvölgyi and R. Gilleron. For a rewrite system it is decidable whether the set of irreducible, ground terms is decidable. *Bulletin of the European Association for Theoretical Computer Science*, 48:197–209, October 1992.

[Vis80] A. Visser. Numerations, lambda calculus, and arithmetic. In Hindley and Seldin, editors, *Essays on Combinatory Logic, Lambda-Calculus, and Formalism*, pages 259–284. Academic Press, 1980.

[vO94] Vincent van Oostrom. *Confluence for Abstract and Higher-Order Rewriting*. PhD thesis, Vrije Universiteit, Amsterdam, March 1994.

[Vor96] Sergei Vorobyov. An improved lower bound for the elementary theories of trees. In McRobbie and Slaney [MS96], pages 275–287.

[Vor97a] Sergei Vorobyov. The first-order theory of one step rewriting in linear noetheran systems is undecidable. In Comon [Com97], pages 254–268.

[Vor97b] Sergei Vorobyov. The "hardest" natural decidable theory. In Winskel [Win97], pages 294–305.

[vOvR94] Vincent van Oostrom and Femke van Raamsdonk. Weak orthogonality implies confluence: the higher-order case. In A. Nerode and Y. V. Matiyasevich, editors, *Third International Symposium on the Logical Foundations of Computer Science*, volume 813 of *Lecture Notes in Computer Science*, pages 379–392, St. Petersburg, Russia, July 1994. Springer-Verlag.

[VZ84] M. Venturini-Zilli. Reduction graphs in the Lambda Calculus. *Theoretical Computer Science*, 29:251–275, 1984.

[Wal98a] Johannes Waldmann. *The Combinator S.* PhD thesis, Fakultät für Mathematik und Informatik of the Friedrich-Schiller-Universität Jena, Jena, Germany, 1998.

[Wal98b] Johannes Waldmann. Normalisation of s-terms is decidable. In Nipkow [Nip98], pages 138–150.

[Wal10] Johannes Waldmann. Polynomially bounded matrix interpretations. In Christopher Lynch, editor, *Proceedings of the 21st International Conference on Rewriting Techniques and Applications*, volume 6 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 357–372, Edinburgh, UK, July 2010. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[WC03] Daria Walukiewicz-Chrzaszcz. Termination of rewriting in the Calculus of Constructions. *Journal of Functional Programming*, 13(2):339–414, 2003.

[Wei92] Andreas Weiermann. Well-rewrite orderings and the induced recursive path orderings. Unpublished note, Institut für Mathematische Logik und Grundlagenforschung, 1992.

[Wei93] Andreas Weiermann. Bounding derivation lengths with functions from the slow growing hierarchy. Preprint Münster, 1993.

[Wel94] Joe B. Wells. Typability and type checking in the second-order λ-calculus are equivalent and undecidable. In Abramsky [Abr94].

[Wer94] Benjamin Werner. *Méta-théorie du Calcul des Constructions Inductives.* Thèse Univ. Paris VII, France, 1994.

[Wie99] Tomasz Wierzbicki. Complexity of the higher-order matching. In Ganzinger [Gan99], pages 82–96.

[Win97] Glynn Winskel, editor. *Twelfth Symposium on Logic in Computer Science*, Warsaw, Poland, June 1997. IEEE.

[Wos] Larry Wos. Automated reasoning: 33 basic research problems.

[Wra90] C. Wrathall. Confluence of one-rule Thue systems. In *Proceedings of the First International Workshop on Word Equations and Related Topics (Tubingen)*, volume 572 of *Lecture Notes in Computer Science*, pages 237–246, Berlin, 1990. Springer-Verlag.

[WZ95] H. R. Walters and Hans Zantema. Rewrite systems for integer arithmetic. In Hsiang [Hsi95], pages 324–338.

[YASM00] Toshiyuki Yamada, Jürgen Avenhaus, Carlos Loría Sáenz, and Aart Middeldorp. Logicality of conditional rewrite systems. *Theoretical Computer Science*, 236(1–2):209–232, April 2000.

[Zan94a] Hans Zantema. Termination of term rewriting: interpretation and type elimination. *Journal of Symbolic Computation*, 17:23–50, 1994.

[Zan94b] Hans Zantema. Total termination of term rewriting is undecidable. Technical Report UU-CS-1994-55, Utrecht University, December 1994.

[ZG96] Hans Zantema and Alfons Geser. Non-looping rewriting. Technical Report UU-CS-96-03, Utrecht University, Department of Computer Science, January 1996. Extended version to appear in *RAIRO Théor. Inf* as "Non-looping string rewriting".

[Zha91] Hantao Zhang. A new strategy for the Boolean ring based approach to first order theorem proving. Technical report, Department of Computer Science, University of Iowa, 1991.

[Zhaar] J. Zhang. A 3-place commutative operator from TBA. *AAR Newsletters*, to appear.

[ZL03] Silvano Dal Zilio and Denis Lugiez. XML schema, tree logic and sheaves automata. In Nieuwenhuis [Nie03], pages 246–263.

[ZSM05] Ting Zhang, Henny Sipma, and Zohar Manna. The decidability of the first-order theory of Knuth-Bendix order. In Robert Nieuwenhuis, editor, *20th International Conference on Automated Deduction*, volume 3632 of *Lecture Notes in Computer Science*, pages 131–148, Tallinn, Estonia, July 2005. Springer-Verlag.