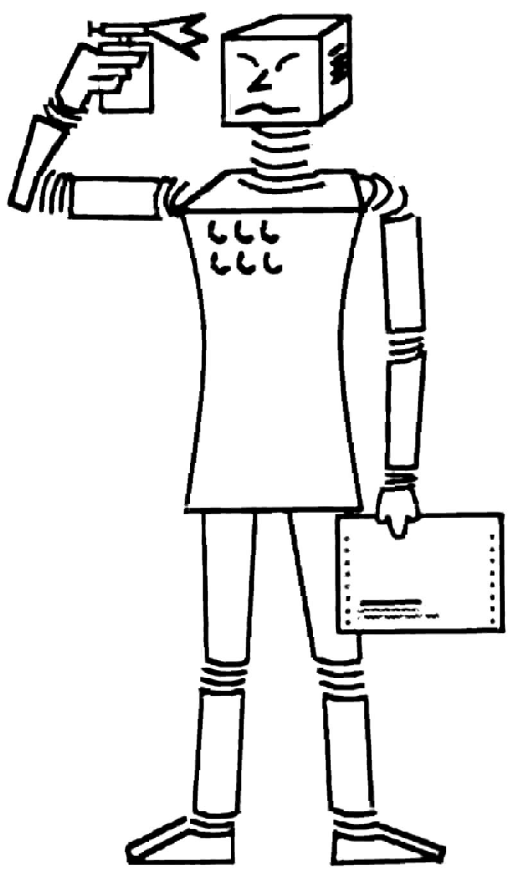
Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern

SEKI - REPORT



Check your Ordering - Termination Proofs and Open Problems

Joachim Steinbach, Ulrich Kühler

SEKI Report SR-90-25 (SFB)

Check Your Ordering

—

Termination Proofs and Open Problems*

Joachim Steinbach
Ulrich Kühler

Universität Kaiserslautern
Fachbereich Informatik
Postfach 3049
D–6750 Kaiserslautern
GERMANY

December 17, 1990

*This research was supported by the Deutsche Forschungsgemeinschaft, SFB314(D4)

Abstract

Termination is a decisive property of term rewriting systems that is often needed. Simplification orderings provide a mechanism for proving this property. The power of such an ordering depends on the number of examples (occurring in practice) to which it can successfully be applied.

In addition to a brief introduction to the presented simplification orderings, this report contains a large collection of term rewriting systems all of which were tested for completion by our completion environment COMTES. In a lot of cases simplification orderings can be found that enable the generation of canonical systems. However, a lot of examples exist for which the problem of completion cannot be solved.

1 Preface

The use of term rewriting systems presumes termination. In [Der87] several different methods which guarantee this important property are presented. The strength of these methods, called orderings, should be investigated in practice.

Therefore, we have collected various examples of rewriting systems and will present them together with some orderings proving their termination. The orderings investigated in this report are all taken from [Ste88] except for the ordering on polynomial interpretations ([Lan79], [BL87], [SZ90]). Besides the latter, the list of orderings includes the recursive path ordering with status (RPOS), a new recursive decomposition ordering with status (IRDS) and the Knuth-Bendix ordering with status (KBOS). The following subsection contains a brief description of these methods.

We have investigated 155 exmples. They belong to several different classes including algebraic structures, arithmetic theories, lists and string rewriting systems. We used two criteria for selecting these rule systems: i) We examined systems which often appear in the literature. ii) The remaining systems are interesting from a termination point of view, i.e., the orderings cannot be applied straightforward in order to prove the termination. More details about the examples examined (in particular, their presentation) will be given in subsection 1.2.

1.1 Used Orderings

First of all, we briefly introduce the used orderings. For more details on orderings, see for example [Der87], [BL87], [Ste88] and [SZ90]. [HO80], [JL87], [Klo87], [DJ89] and [AM90] contain general aspects and formal definitions of term rewriting systems.

A (partial) ordering $\succ$ on any set $\mathcal{M}$ is a transitive and irreflexive binary relation on $\mathcal{M}$. The application of an ordering $\succ$ w.r.t. term rewriting requires the well-foundedness to prevent infinite derivations of terms.

Note that $\succ$ is used to compare terms. Since operators have terms as arguments we define an extension of $\succ$, called lexicographically greater ($\succ^{lex}$), on tuples of elements as follows:

$$\begin{array}{lll} (m_1, m_2, ..., m_p) \succ^{lex} (n_1, n_2, ..., n_q) & & \\ \quad if \;\; either & p > 0 & \wedge \;\; q = 0 \\ \quad\quad\quad or & m_1 \succ n_1 & \\ \quad\quad\quad or & m_1 \sim n_1 & \wedge \;\; (m_2, ..., m_p) \succ^{lex} (n_2, ..., n_q). \end{array}$$

Multisets are treated like sets, that allow multiple occurrences of identical elements. The extension of $\succ$ on multisets of elements is defined as follows: A multiset $\mathcal{M}_1$ is greater than a multiset $\mathcal{M}_2$ over $\mathcal{M}$, denoted by

$$\begin{array}{ll} \mathcal{M}_1 \succ_{mul} \mathcal{M}_2 & \\ \quad iff \;\; i) & \mathcal{M}_1 \neq \mathcal{M}_2 \;\; \wedge \\ \quad\quad\quad ii) & (\forall y \in \mathcal{M}_2 \setminus \mathcal{M}_1)(\exists x \in \mathcal{M}_1 \setminus \mathcal{M}_2) \; x \succ y. \end{array}$$

All orderings used in this paper share one essential characteristic: Each operator has a status that determines the order according to which the direct subterms of f are compared ([KL80]). Formally, the status is a function τ which maps the set of operators $\mathcal{F}$ into the set $\{mul, left, right\}$:

$$\tau : \mathcal{F} \rightarrow \{mul, left, right\}.$$

Therefore, the status of a function symbol can have one of the following three values: *mul* (the arguments will be compared as multisets), *left* (lexicographical comparison from left to right) and *right* (the arguments will be compared lexicographically from right to left).

The index '$\tau(f)$' in '$\succ_{ord,\tau(f)}$' denotes the extension of the ordering $\succ_{ord}$ w.r.t. the status of the operator f:

$$\begin{array}{lll} (s_1, ..., s_m) \succ_{ord,\tau(f)} (t_1, ..., t_n) & & \\ \quad iff \quad\quad & \tau(f) = mul & \wedge \;\; \{s_1, ..., s_m\} \succ_{ord,mul} \{t_1, ..., t_n\} \\ \quad\quad\quad or & \tau(f) = left & \wedge \;\; (s_1, ..., s_m) \succ^{lex}_{ord} (t_1, ..., t_n) \\ \quad\quad\quad or & \tau(f) = right & \wedge \;\; (s_m, ..., s_1) \succ^{lex}_{ord} (t_n, ..., t_1) \end{array}$$

All of the following orderings uniquely define a congruence $\sim$ depending on $\mathcal{F}$ and τ via $f(s_1, ..., s_m) \sim g(t_1, ..., t_n)$ iff $f = g$ and $m = n$ and i) $\tau(f) = mul$ and there is a permutation π of the set $\{1, ..., n\}$ such that $s_i \sim t_{\pi(i)}$, for all $i \in [1, n]$ or ii) $\tau(f) \neq mul \wedge s_i \sim t_i$, for all $i \in [1, n]$.

Most of the orderings are based on the principle of root orderings, i.e., the comparison of two terms depends on their leading function symbols. The presentation of the case distinctions which result is of the following form: $s \succ t$ iff i) *condition*$_1$ ii) *condition*$_2$ ... denoting $s \succ t$ iff *condition*$_1$ or *condition*$_2$ or ... holds. If the conditions of the case distinctions are marked by hyphens they are to be evaluated 'lexicographically', e.g., $s \succ t$ iff $-\ s \succ_1 t\ \ -\ s \succ_2 t$ stands for $s \succ t$ iff $s \succ_1 t$ or $(s \sim_1 t \ \wedge\ s \succ_2 t)$. Here, the symbol $\sim_1$ denotes the congruence relation induced by the quasi-ordering $\succeq_1$.

When writing s, t and $\succ$ we will always assume that s and t are terms and $\succ$ is a precedence on the set $\mathcal{F}$ of operators.

Definition 1.1 ([KL80]) *Recursive Path Ordering with Status :* **RPOS**[1]

$s \succ_{RPOS} t \iff$

$i)$ $\mathcal{H}ead(s) \succ \mathcal{H}ead(t) \ \wedge\ \{s\} \succ_{RPOS,mul} \mathcal{A}rgs(t)$

$ii)$ $\mathcal{H}ead(s) = \mathcal{H}ead(t) \ \wedge\ \tau(\mathcal{H}ead(s)) = mul \ \wedge\ \mathcal{A}rgs(s) \succ_{RPOS,mul} \mathcal{A}rgs(t)$

$iii)$ $\mathcal{H}ead(s) = \mathcal{H}ead(t) \ \wedge\ \tau(\mathcal{H}ead(s)) \neq mul \ \wedge\ \mathcal{A}rgs(s) \succ_{RPOS,\tau(\mathcal{H}ead(s))} \mathcal{A}rgs(t)$
$\wedge\ \{s\} \succ_{RPOS,mul} \mathcal{A}rgs(t)$

$iv)$ $\mathcal{A}rgs(s) \succeq_{RPOS,mul} \{t\}$

To define the improved recursive decomposition ordering with status, we need some kind of formalism. A path of a term is a sequence of terms starting with the whole term followed by a path of one of its arguments:

- $path_\varepsilon(\triangle) \ = \ \triangle$ $\quad$ *if* $\triangle$ is a constant or a variable,
- $path_{i.u}(f(t_1,...,t_n)) \ = \ f(t_1,...,t_n); path_u(t_i)$ $\quad$ *if* $u \in \mathcal{P}os^*(t_i)$[2]

For a path $p = [t_1; t_2; ...; t_n]$ we denote the set $\{t_1, ..., t_n\}$ of all terms in p by *set*(p). This set will also be called path-decomposition and its abbreviation is $dec_u(t)$ (and is equal to $set(path_u(t))$).

The decomposition $dec(\{t_1,...,t_n\}) = \{dec_u(t_i) \mid i \in [1,n]\ ,\ u \in \mathcal{P}os^*(t_i)\}$ is the multiset of all path-decompositions of the terms $t_1, ..., t_n$.

The set of subterms of a path-decomposition P relative to a term t is defined as

$$sub(P,t) = \{s \in P \mid (\exists u \neq \varepsilon)\ t|_u = s\}.$$ [3]

Suppose $t = (x * y) + (x * z)$, then $dec_{21}(t) = \{t, x * z, x\}$, $dec(\{t\}) = \{\{t, x * y, x\}, \{t, x * y, y\}, \{t, x * z, x\}, \{t, x * z, z\}\}$ and $sub(dec_{21}(t), x) = \emptyset$.

Definition 1.2 ([Ste88] , [Ste89b]) *Improved Recursive Decomposition Ordering with Status :* **IRDS**[4]

$s \succ_{IRDS} t$

[1] The RPOS is based on the recursive path ordering of [Der82].

[2] where $\mathcal{P}os^*(t_i)$ is the set of all terminal occurrences of t_i.

[3] $t|_u$ denotes the subterm of t at occurrence u.

[4] The IRDS is based on the recursive decomposition ordering developed by Lescanne (see for example [Les83]). The power of the IRDS is identical to that of the path ordering with status of [KNS85].

$$
\begin{array}{ll}
\Longleftrightarrow & dec(\{s\}) \succ_{EL,mul}\succ_{EL,mul} dec(\{t\}) \\
& with\ dec_u(s) \ni s' \succ_{EL} t' \in dec_v(t) \\
& \quad \Longleftrightarrow \quad i) \quad \mathcal{H}ead(s') \succ \mathcal{H}ead(t') \\
& \qquad\qquad ii) \quad \mathcal{H}ead(s') = \mathcal{H}ead(t') \ \wedge\ \tau(\mathcal{H}ead(s')) = mul \ \wedge \\
& \qquad\qquad\qquad - \ sub(dec_u(s), s') \succ_{EL,mul} sub(dec_v(t), t') \\
& \qquad\qquad\qquad - \ dec(\mathcal{A}rgs(s')) \succ_{EL,mul}\succ_{EL,mul} dec(\mathcal{A}rgs(t')) \\
& \qquad\qquad iii) \quad \mathcal{H}ead(s') = \mathcal{H}ead(t') \ \wedge\ \tau(\mathcal{H}ead(s')) \neq mul \ \wedge \\
& \qquad\qquad\qquad \mathcal{A}rgs(s') \succ_{IRDS,\tau(\mathcal{H}ead(s'))} \mathcal{A}rgs(t') \ \wedge\ \{s'\} \succ_{IRDS,mul} \mathcal{A}rgs(t')
\end{array}
$$

To describe the Knuth-Bendix ordering with status and the polynomial ordering, more background information and helpful definitions will be presented.

If x is a variable and t is a term we denote the number of occurrences of x in t by $\#_x(t)$. We assign a non-negative integer $\varphi(f)$ – the weight of f – to each operator and a positive integer φ_0 to each variable such that

$$
\begin{array}{ll}
\varphi(c) \geq \varphi_0 & \text{if } c \text{ is a constant,} \\
\varphi(f) = 0 & \text{for one unary operator } f \text{ at most; } f \text{ has to be maximal w.r.t.} \succ .
\end{array}
$$

We extend this weight function on operators to terms:

$$\varphi(f(t_1, ..., t_n)) = \varphi(f) + \sum_{i=1}^{n} \varphi(t_i).$$

The weight of an operator f can be considered as a special linear function: $x_1 + ... + x_n + \varphi(f)$. A polynomial ordering even allows polynomials as the 'weight' of operators. For each n-ary operator f an interpretation $[f](x_1, ..., x_n)$ defines a polynomial with n variables which satisfies the following conditions:

- $[f]$ must have the same arity as f
- the coefficients of $[f]$ must be integers
- $(\forall x, y)\ [f](..., x, ...) > [f](..., y, ...)\ \ if\ \ x > y.$

The extension of an interpretation of an operator to a term $f(t_1, ..., t_n)$ is defined as follows:

$$[f(t_1, ..., t_n)](x_1, ..., x_m) = [f]([t_1](x_1, ..., x_m), ..., [t_n](x_1, ..., x_m)).$$

Definition 1.3 ([Ste88] , [Ste89b]) *Knuth-Bendix Ordering with Status:* **KBOS**[5]

$$
\begin{array}{lll}
s \succ_{KBOS} t & & \\
\Longleftrightarrow & (\forall x) & \#_x(s) \geq \#_x(t) \ \wedge \\
& i) & s \equiv f(...f(t)) \\
& ii) & - \ \varphi(s) > \varphi(t) \\
& & - \ \mathcal{H}ead(s) \succ \mathcal{H}ead(t) \\
& & - \ \mathcal{A}rgs(s) \succ_{KBOS,\tau(\mathcal{H}ead(s))} \mathcal{A}rgs(t)
\end{array}
$$

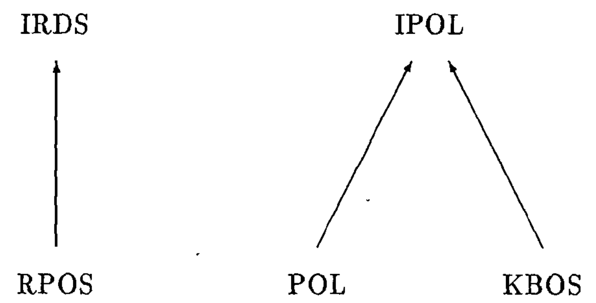
[5]The KBOS is an extension of the original Knuth-Bendix ordering of [KB70].

Definition 1.4 ([Lan79]) *Ordering on Polynomial Interpretations:* **POL**
$s \succ_{POL} t$
$\iff$ $[s] \sqsupset [t]$
where $p \sqsupset q$[6]
$\iff$ $(\forall x_i \in \mathcal{M} \subseteq \mathcal{N}at)\ p(x_1, ..., x_n) > q(x_1, ..., x_n)$

Definition 1.5 ([SZ90]) *Improved Polynomial Ordering:* **IPOL**
For each constant c, $[c]() \geq 2$.
$s \succ_{IPOL} t$
$\iff$
- $s \succ_{POL} t$
- $\mathcal{H}ead(s) \succ \mathcal{H}ead(t)$
- $\mathcal{A}rgs(s) \succ_{IPOL,\tau(\mathcal{H}ead(s))} \mathcal{A}rgs(t)$

Note that, if f is a unary operator interpreted by the identity function (i.e., $[f](x) = x$) there must not exist another operator being greater than f w.r.t. $\succ$. Furthermore, $\succ_{IPOL}$ is a simplification ordering only if the condition '$f(...f(s)) \succ_{IPOL} s$ if $[f](x) = x$' is additionally required.

The above described orderings were applied to prove the termination of rule systems. The following diagram depicts the comparison of their powers. The power of an ordering is described by the set of comparable pairs of terms. Thus, we examine the relation between two such sets. We use a kind of Hasse diagrams to represent these relations, i.e., if $\succ_1 \subset \succ_2$ then we arrange $\succ_2$ above $\succ_1$ joining them with an arrow:



For more details about the comparison of these orderings, see [Ste88], [Ste89b] and [SZ90].

[6]Note that $[s]$ and $[t]$ are polynomials. Thus, $\sqsupset$ is an ordering on polynomials (p,q). $p(x_1, ..., x_n)$ represents the polynomial $\sum_{r_1,...,r_n} c_{r_1...r_n} x_1^{r_1} \cdot ... \cdot x_n^{r_n}$ based on n variables. $\mathcal{N}at$ denotes the set of natural numbers.

1.2 Presentation of the Examples

All examples will be presented in the following way:

$$\begin{array}{lcl} l_1 & \to & r_1 \\ & \vdots & \\ l_n & \to & r_n \end{array}$$

$$\begin{array}{lcl} l_{n+1} & \to & r_{n+1} \\ & \vdots & \\ l_m & \to & r_m \\ IPOL & : & \ldots^7 \\ IRDS & : & \ldots \\ KBOS & : & \ldots \\ POL & : & \ldots \\ RPOS & : & \ldots \end{array}$$

The initial system consists of the rules $l_1 \to r_1 \cdots l_n \to r_n$. Throughout the completion process new rules, $l_{n+1} \to r_{n+1} \cdots l_m \to r_m$, are generated. The canonical system contains all of the presented rules except those marked with '$\diamond$'. Appropriate term orderings are specified by their parameters (such as the precedence, the status function, the weight function, the polynomial interpretations). Note that each set of orderings is minimal w.r.t. the power of the orderings, e.g., if there exists an RPOS we do not provide an appropriate IRDS (since the IRDS is stronger than the RPOS, see subsection 1.1). However, if none of the two strongest orderings of subsection 1.1 (either IRDS or IPOL) is enumerated the termination of the corresponding example cannot be proved (with the help of the presented orderings).

We use x, y, z, u, v, w to represent variables. Function symbols will be denoted by $f, g, h, h_1, h_2, i, j, k, p, s$ and a, b, b', c, d, d', e (constant symbols). Some additional function symbols which occur in the examples require no further elucidation.

The tested examples belong to several domains. The list of these classes consists of *algebraic structures* (groups, rings, etc.), *arithmetic theories* (addition, multiplication, etc.), *boolean theories* (if-then-else, etc.), *lists* (append, reverse, flatten, etc.), *string rewriting systems*[8] and *other systems* (systems that do not fall into one of the five former classes). Each example is characterized by an abbreviation of the class it belongs to (G: algebraic structures, A: arihmetical theories, B: Boolean theories, L: lists, S: string rewriting systems, O: other systems).

For completing the initial rule systems we used our completion environment COMTES (see [AMS89] , [Ste89a]) in which all presented orderings have been integrated. Obviously[9], we did not employ 'special tricks' such as postponing incomparable pairs, introducing new operators, etc. More precisely, the generation of an incomparable pair caused the completion process to break off.

---

[7] $\ldots$ represents the parameters of the orderings.

[8] The structure of string rewriting systems is easier than that of lists since they use unary operators, only.

[9] We are only interested in testing the power of the presented orderings.

In the next section we deal with the termination of systems which are already canonical (sometimes the systems have to be normalized). The third section contains rewriting systems where at least one non-trivial critical pair has to be generated in order to complete them. Finally, in the fourth section, we present a lot of systems for which the termination cannot be guaranteed using the methods described in subsection 1.1. However, all of these examples are terminating. We know that there exist some other orderings[10] with which the termination property of some of these examples can be proved. However, this report stands at the end of our research on simplification orderings and represents a beginning of investigations about orderings that can handle homeomorphic embeddings (in particular extremely recursive definitions).

[10] For example, semantic path orderings of [KL80] and transformation orderings of [BD86] and [BL88].

2 Canonical Systems

EXAMPLE 2.1 G: Group Theory ([KB70])

$$
\begin{array}{lcl}
i(0) & \to & 0 \\
0+y & \to & y \\
x+0 & \to & x \\
i(i(x)) & \to & x \\
i(x)+x & \to & 0 \\
x+i(x) & \to & 0 \\
i(x+y) & \to & i(y)+i(x) \\
x+(y+z) & \to & (x+y)+z \\
(x+i(y))+y & \to & x \\
(x+y)+i(y) & \to & x
\end{array}
$$

KBOS: $\varphi(i)=0, \varphi(+)>\varphi(0)$
$i \succ +$
$\tau(+)=right$

POL:
$$
\begin{array}{lcl}
[+](x,y) & = & 2xy+y \\
[i](x) & = & x^2 \\
[0]() & = & 2
\end{array}
$$

RPOS: $i \succ + \succ 0$
$\tau(+)=right$ □

EXAMPLE 2.2 G: Associativity and Endomorphism ([BL88])

$$
\begin{array}{lcl}
(x+y)+z & \to & x+(y+z) \\
f(x)+f(y) & \to & f(x+y) \\
f(x)+(f(y)+z) & \to & f(x+y)+z
\end{array}
$$

KBOS: $\varphi(f)>0$
$\tau(+)=left$

POL:
$$
\begin{array}{lcl}
[+](x,y) & = & xy+x \\
[f](x) & = & 2x
\end{array}
$$
□

EXAMPLE 2.3 G: [Kir87]

$$
\begin{array}{lcl}
--x & \to & x \\
-h(x) & \to & h(-x) \\
-f(x,y) & \to & f(-y,-x)
\end{array}
$$

KBOS: $\varphi(-)=0$
$- \succ h, - \succ f$

POL:
$$
\begin{array}{lcl}
[-](x) & = & 2x \\
[f](x,y) & = & x+y+1 \\
[h](x) & = & x+1
\end{array}
$$

RPOS: $- \succ h, - \succ f$ □

EXAMPLE 2.4 G: [Kir87]

$$\begin{array}{lcl} f(x+0) & \to & f(x) \\ x+(y+z) & \to & (x+y)+z \end{array}$$

KBOS: $\tau(+) = right$

POL:
$$\begin{array}{lcl} [+](x,y) & = & x+2y \\ [f](x) & = & x \\ [0]() & = & 1 \end{array}$$

RPOS: $\tau(+) = right$ □

EXAMPLE 2.5 G: Associativity and Distributivity

$$\begin{array}{lcl} x+(y+z) & \to & (x+y)+z \\ x*(y+z) & \to & (x*y)+(x*z) \\ (x+(y*z))+(y*u) & \to & x+(y*(z+u)) \end{array} \quad \diamond$$

POL:
$$\begin{array}{lcl} [+](x,y) & = & x+2y+1 \\ [*](x,y) & = & xy \end{array}$$
□

EXAMPLE 2.6 G: [Mar87]

$$\begin{array}{lcl} (x+y)+z & \to & x+(y+z) \\ (x*y)+(x*z) & \to & x*(y+z) \\ (x*y)+((x*z)+u) & \to & (x*(y+z))+u \end{array}$$

KBOS: $\varphi(*) > 0$
$\tau(+) = left$

POL:
$$\begin{array}{lcl} [+](x,y) & = & xy+x \\ [*](x,y) & = & x+y \end{array}$$
□

EXAMPLE 2.7 G: [Bel86]

$$\begin{array}{lcl} f(0,y) & \to & y \\ f(x,0) & \to & x \\ f(i(x),y) & \to & i(x) \\ f(f(x,y),z) & \to & f(x,f(y,z)) \\ f(g(x,y),z) & \to & g(f(x,z),f(y,z)) \\ f(1,g(x,y)) & \to & x \\ f(2,g(x,y)) & \to & y \end{array}$$

POL: $[f](x,y) = xy + x$
$[g](x,y) = x + y + 1$
$[i](x) = x$
$[0]() = 1$
$[1]() = 1$
$[2]() = 1$

RPOS: $f \succ g$
$\tau(f) = left$

□

EXAMPLE 2.8 G: [KNZ86]

$$\begin{array}{lcl} a + b & \rightarrow & b + a \\ a + (b + z) & \rightarrow & b + (a + z) \\ (x + y) + z & \rightarrow & x + (y + z) \\ f(a, y) & \rightarrow & a \\ f(b, y) & \rightarrow & b \\ f(x + y, z) & \rightarrow & f(x, z) + f(y, z) \end{array}$$

IPOL: $[+](x,y) = x + y + 1$
$[f](x,y) = xy$
$[a]() = 3$
$[b]() = 2$
$\tau(+) = left$

RPOS: $a \succ b, f \succ +$
$\tau(+) = left$

□

EXAMPLE 2.9 A: [PF86]

$$\begin{array}{lcl} 0 + y & \rightarrow & y \\ s(x) + y & \rightarrow & s(x + y) \\ s(x) + y & \rightarrow & x + s(y) \quad \diamond \end{array}$$

$$x + s(y) \rightarrow s(x + y)$$

KBOS: $+ \succ s$
$\tau(+) = left$

POL: $[+](x,y) = 3x + 2y$
$[s](x) = x + 1$
$[0]() = 1$

RPOS: $+ \succ s$
$\tau(+) = left$

□

EXAMPLE 2.10 A: Unary Integer Addition ([Der85])

$$\begin{array}{lcl} -0 & \to & 0 \\ x+0 & \to & x \\ 0+y & \to & y \\ (-1)+1 & \to & 0 \\ --x & \to & x \\ x+(-y) & \to & -((-x)+y) \\ x+(y+z) & \to & (x+y)+z \\ (-(x+1))+1 & \to & -x \end{array}$$

POL: $$\begin{array}{lcl} [+](x,y) & = & x+3y \\ [-](x) & = & x+1 \\ [0]() & = & 1 \\ [1]() & = & 1 \end{array}$$

RPOS: $+ \succ -, + \succ 0$

$\tau(+) = right$

□

EXAMPLE 2.11 A: Addition and Subtraction

$$\begin{array}{lcl} 0+y & \to & y \\ s(x)+y & \to & s(x+y) \\ 0-y & \to & 0 \\ x-0 & \to & x \\ s(x)-s(y) & \to & x-y \end{array}$$

KBOS: $\varphi(s) > 0$

$+ \succ s$

POL: $$\begin{array}{lcl} [+](x,y) & = & 2x+y \\ [-](x,y) & = & x+y \\ [s](x) & = & x+1 \\ [0]() & = & 1 \end{array}$$

RPOS: $+ \succ s$

□

EXAMPLE 2.12 A: [NRS89]

$$\begin{array}{lcl} 0+y & \to & y \\ s(x)+y & \to & s(x+y) \\ p(x)+y & \to & p(x+y) \\ -0 & \to & 0 \\ -s(x) & \to & p(-x) \\ -p(x) & \to & s(-x) \\ 0*y & \to & 0 \\ s(x)*y & \to & (x*y)+y \\ p(x)*y & \to & (x*y)+(-y) \end{array}$$

IPOL:	$[+](x,y)$	$=$	$x+y$
	$[*](x,y)$	$=$	$xy+1$
	$[-](x)$	$=$	$x+1$
	$[p](x)$	$=$	$x+2$
	$[s](x)$	$=$	$x+2$
	$[0]()$	$=$	1
	$* \succ + \succ p, + \succ s, - \succ p, - \succ s$		
RPOS:	$* \succ - \succ p, - \succ s, * \succ + \succ p, + \succ s$		

□

EXAMPLE 2.13 A[11]:

$double(0)$	$\rightarrow$	0	◇
$double(s(x))$	$\rightarrow$	$s(s(double(x)))$	◇
$x+0$	$\rightarrow$	x	
$x+s(y)$	$\rightarrow$	$s(x+y)$	
$s(x)+y$	$\rightarrow$	$s(x+y)$	
$double(x)$	$\rightarrow$	$x+x$	

POL:	$[+](x,y)$	$=$	$2x+2y$
	$[double](x)$	$=$	$4x+1$
	$[s](x)$	$=$	$x+1$
	$[0]()$	$=$	1
RPOS:	$double \succ + \succ s$		

□

EXAMPLE 2.14 A: [Toy86]

$double(0)$	$\rightarrow$	0
$double(s(x))$	$\rightarrow$	$s(s(double(x)))$
$half(0)$	$\rightarrow$	0
$half(s(0))$	$\rightarrow$	0
$half(s(s(x)))$	$\rightarrow$	$s(half(x))$
$x-0$	$\rightarrow$	x
$s(x)-s(y)$	$\rightarrow$	$x-y$
$if(0,y,z)$	$\rightarrow$	y
$if(s(x),y,z)$	$\rightarrow$	z
$half(double(x))$	$\rightarrow$	x

POL:	$[-](x,y)$	$=$	$x+y$
	$[double](x)$	$=$	$3x$
	$[half](x)$	$=$	$x+1$
	$[if](x,y,z)$	$=$	$x+y+z$
	$[s](x)$	$=$	$x+1$
	$[0]()$	$=$	1
RPOS:	$double \succ s, half \succ s$		

□

[11] The 'inductive theorem' described by the last rule would lead to divergence if the fifth rule were not included.

EXAMPLE 2.15 A: [JSS85]

$$\begin{array}{lll} f(0) & \rightarrow & 1 \\ f(s(x)) & \rightarrow & g(f(x)) \quad \diamond \\ g(x) & \rightarrow & x + s(x) \end{array}$$

$$f(s(x)) \rightarrow f(x) + s(f(x))$$

POL:

$$\begin{array}{lll} [+](x,y) & = & x+y \\ [f](x) & = & x^2 \\ [g](x) & = & 3x+1 \\ [s](x) & = & 2x \\ [0]() & = & 2 \\ [1]() & = & 2 \end{array}$$

RPOS: $f \succ 1, f \succ g \succ +, g \succ s$ □

EXAMPLE 2.16 A: [Toy86]

$$\begin{array}{lll} f(0) & \rightarrow & 0 \\ f(s(x)) & \rightarrow & g(x, s(x)) \\ g(0,y) & \rightarrow & y \\ g(s(x),y) & \rightarrow & g(x, y+s(x)) \quad \diamond \\ x+0 & \rightarrow & x \\ x+s(y) & \rightarrow & s(x+y) \end{array}$$

$$g(s(x),y) \rightarrow g(x, s(y+x))$$

POL:

$$\begin{array}{lll} [+](x,y) & = & x+2y \\ [f](x) & = & x^2 \\ [g](x,y) & = & x^2+y \\ [s](x) & = & x+2 \\ [0]() & = & 2 \end{array}$$

RPOS: $f \succ g \succ + \succ s$
$\tau(g) = left$ □

EXAMPLE 2.17 A: Sum of Natural Numbers

$$\begin{array}{lll} sum(0) & \rightarrow & 0 \\ sum(s(x)) & \rightarrow & sum(x) + s(x) \\ sum1(0) & \rightarrow & 0 \\ sum1(s(x)) & \rightarrow & s(sum1(x) + (x+x)) \end{array}$$

POL:

$$\begin{array}{lll} [+](x,y) & = & x+y \\ [s](x) & = & x+2 \\ [sum](x) & = & x^2 \\ [sum1](x) & = & x^2 \\ [0]() & = & 2 \end{array}$$

RPOS: $sum \succ +, sum1 \succ +, sum1 \succ s$ □

EXAMPLE 2.18 A[12]:

$$\begin{array}{lll} sum(0) & \rightarrow & 0 \\ sum(s(x)) & \rightarrow & s(x) + sum(x) \\ x + 0 & \rightarrow & x \\ x + s(y) & \rightarrow & s(x+y) \end{array}$$

IPOL:

$$\begin{array}{lll} [+](x,y) & = & x + y \\ [s](x) & = & x + 1 \\ [sum](x) & = & x^2 \\ [0]() & = & 1 \\ + \succ s & & \end{array}$$

RPOS: $sum \succ + \succ s$ □

EXAMPLE 2.19 A: Square Numbers ([HL89])

$$\begin{array}{lll} sqr(0) & \rightarrow & 0 \\ sqr(s(x)) & \rightarrow & sqr(x) + s(double(x)) \quad \diamond \\ double(0) & \rightarrow & 0 \\ double(s(x)) & \rightarrow & s(s(double(x))) \\ x + 0 & \rightarrow & x \\ x + s(y) & \rightarrow & s(x+y) \end{array}$$

$$sqr(s(x)) \rightarrow s(sqr(x) + double(x))$$

POL:

$$\begin{array}{lll} [+](x,y) & = & x + 2y \\ [double](x) & = & 3x \\ [s](x) & = & x + 3 \\ [sqr](x) & = & x^2 \\ [0]() & = & 2 \end{array}$$

RPOS: $sqr \succ + \succ s, sqr \succ double \succ s$ □

EXAMPLE 2.20 A: Sum of Square Numbers

$$\begin{array}{lll} sum(0) & \rightarrow & 0 \\ sum(s(x)) & \rightarrow & sqr(s(x)) + sum(x) \quad \diamond \\ sqr(x) & \rightarrow & x * x \end{array}$$

$$sum(s(x)) \rightarrow (s(x) * s(x)) + sum(x)$$

POL:

$$\begin{array}{lll} [+](x,y) & = & x + y \\ [*](x,y) & = & x + y \\ [s](x) & = & x + 2 \\ [sqr](x) & = & 2x + 1 \\ [sum](x) & = & x^2 \\ [0]() & = & 2 \end{array}$$

RPOS: $sum \succ +, sum \succ sqr \succ *$ □

[12]The following exponential functions can also guarantee the termination of the system: $[+](x,y) = x + 2y$, $[s](x) = x + 2$, $[sum](x) = 2^x$, $[0]() = 2$.

EXAMPLE 2.21 A: Binomial Coefficients

$$\begin{array}{lcl} bin(x,0) & \rightarrow & s(0) \\ bin(0,s(y)) & \rightarrow & 0 \\ bin(s(x),s(y)) & \rightarrow & bin(x,s(y)) + bin(x,y) \end{array}$$

POL: $$\begin{array}{lcl} [+](x,y) & = & x+y \\ [bin](x,y) & = & xy+x \\ [s](x) & = & 2x \\ [0]() & = & 2 \end{array}$$

RPOS: $bin \succ s, bin \succ +$ □

EXAMPLE 2.22 A: Arithmetical Theory[13]

$$\begin{array}{lcl} exp(x,0) & \rightarrow & s(0) \\ exp(x,s(y)) & \rightarrow & x * exp(x,y) \\ 0 * y & \rightarrow & 0 \\ s(x) * y & \rightarrow & y + (x * y) \\ 0 - y & \rightarrow & 0 \\ x - 0 & \rightarrow & x \\ s(x) - s(y) & \rightarrow & x - y \end{array}$$

RPOS: $exp \succ * \succ +, exp \succ s$ □

EXAMPLE 2.23 A: Factorial Function[14] ([Pad89]) ◇

$$\begin{array}{lcl} fac(0) & \rightarrow & 1 \\ fac(s(x)) & \rightarrow & s(x) * fac(x) \\ floop(0,y) & \rightarrow & y \\ floop(s(x),y) & \rightarrow & floop(x, s(x) * y) \\ x * 0 & \rightarrow & 0 \\ x * s(y) & \rightarrow & (x * y) + x \\ x + 0 & \rightarrow & x \\ x + s(y) & \rightarrow & s(x+y) \\ 1 & \rightarrow & s(0) \end{array}$$

$fac(0) \rightarrow s(0)$

RPOS: $floop \succ * \succ + \succ s, fac \succ *, fac \succ 1 \succ 0, 1 \succ s$
$\tau(floop) = left$ □

EXAMPLE 2.24 A: Fibonacci Function

$$\begin{array}{lcl} fib(0) & \rightarrow & 0 \\ fib(s(0)) & \rightarrow & s(0) \\ fib(s(s(x))) & \rightarrow & fib(s(x)) + fib(x) \end{array}$$

[13]The system can be oriented by the exponential functions $[+](x,y) = x+y$, $[*](x,y) = xy$, $[-](x,y) = x+y$, $[exp](x,y) = 2^{xy}$, $[s](x) = 2x$, $[0]() = 2$.

[14]There exist exponential interpretations of the operators with which the rules of the system can be oriented except for the fourth rule.

POL:	$[+](x,y) = x+y$	
	$[fib](x) = x+1$	
	$[s](x) = 2x$	
	$[0]() = 2$	
RPOS:	$fib \succ +$	□

EXAMPLE 2.25 A: Fibonacci Function[15] ([Toy86])

$$
\begin{array}{lcl}
fib(0) & \rightarrow & 0 \\
fib(s(0)) & \rightarrow & s(0) \\
fib(s(s(x))) & \rightarrow & fib(s(x)) + fib(x) \\
x + 0 & \rightarrow & x \\
x + s(y) & \rightarrow & s(x+y)
\end{array}
$$

RPOS: $fib \succ + \succ s$ □

EXAMPLE 2.26 A: Fibonacci Function[16] ([Toy86])

$$
\begin{array}{lcll}
f(0) & \rightarrow & s(0) & \\
f(s(0)) & \rightarrow & s(0) & \\
f(s(s(x))) & \rightarrow & p(h(g(x))) & \diamond \\
g(0) & \rightarrow & <s(0), s(0)> & \\
g(s(x)) & \rightarrow & h(g(x)) & \diamond \\
h(x) & \rightarrow & <p(x) + q(x), p(x)> & \\
p(<x,y>) & \rightarrow & x & \\
q(<x,y>) & \rightarrow & y & \\
x + 0 & \rightarrow & x & \\
x + s(y) & \rightarrow & s(x+y) &
\end{array}
$$

$$
\begin{array}{lcl}
f(s(s(x))) & \rightarrow & p(g(x)) + q(g(x)) \\
g(s(x)) & \rightarrow & <p(g(x)) + q(g(x)), p(g(x))>
\end{array}
$$

RPOS: $f \succ g \succ h \succ + \succ s, h \succ <>, h \succ p; h \succ q$ □

[15]The system can be oriented by the exponential functions $[+](x,y) = x+2y$, $[fib](x) = 2^x$, $[s](x) = x+2$, $[0]() = 1$.

[16]The system can be oriented by the following exponential functions: $[+](x,y) = x+y$, $[<>](x,y) = x+y$, $[f](x) = 2^x$, $[g](x) = 2^x$, $[h](x) = 4x+1$, $[p](x) = x$, $[q](x) = x$, $[s](x) = x+3$, $[0]() = 1$.

EXAMPLE 2.27 A: Fibonacci Function[17] ([Red89])

$$\begin{array}{lll} fib(0) & \rightarrow & 0 \\ fib(s(0)) & \rightarrow & s(0) \\ fib(s(s(0))) & \rightarrow & s(0) & \diamond \\ fib(s(s(x))) & \rightarrow & s_p(g(x)) \\ g(0) & \rightarrow & <s(0),0> \\ g(s(0)) & \rightarrow & <s(0),s(0)> & \diamond \\ g(s(x)) & \rightarrow & n_p(g(x)) \\ s_p(<x,y>) & \rightarrow & x+y \\ n_p(<x,y>) & \rightarrow & <x+y,x> \\ x+0 & \rightarrow & x \\ x+s(y) & \rightarrow & s(x+y) \end{array}$$

RPOS: $fib \succ g \succ n_p \succ <> \succ + \succ s, g \succ s_p$ □

EXAMPLE 2.28 A: [Les83]

$$dfib(s(s(x)),y) \rightarrow dfib(s(x), dfib(x,y))$$

POL:

$$\begin{array}{lll} [dfib](x,y) & = & x+y \\ [s](x) & = & 2x \end{array}$$

RPOS: $\tau(dfib) = left$ □

EXAMPLE 2.29 A: Prime Numbers ([BM79])

$$\begin{array}{lll} prime(0) & \rightarrow & false \\ prime(s(0)) & \rightarrow & false \\ prime(s(s(x))) & \rightarrow & prime1(s(s(x)),s(x)) \\ prime1(x,0) & \rightarrow & false \\ prime1(x,s(0)) & \rightarrow & true \\ prime1(x,s(s(y))) & \rightarrow & (\neg divp(s(s(y)),x)) \wedge prime1(x,s(y)) \\ divp(x,y) & \rightarrow & rem(x,y) \equiv 0 \end{array}$$

POL:

$$\begin{array}{lll} [\wedge](x,y) & = & x+y \\ [\equiv](x,y) & = & x+y \\ [\neg](x) & = & x \\ [divp](x,y) & = & x+y+3 \\ [prime](x) & = & x^3 \\ [prime1](x,y) & = & xy^2 \\ [rem](x,y) & = & x+y \\ [s](x) & = & 2x \\ [0]() & = & 2 \\ [false]() & = & 2 \end{array}$$

[17] The system can be oriented by the exponential functions $[+](x,y) = x + 2y$, $[<>](x,y) = x + y$, $[f](x) = 2^x$, $[g](x) = 2^x + 1$, $[n_p](x) = 2x + 1$, $[s](x) = x + 2$, $[s_p](x) = 2x$, $[0]() = 1$.

RPOS: $prime \succ prime1 \succ false, prime1 \succ true, prime1 \succ divp \succ rem, prime1 \succ \wedge, prime1 \succ \neg, divp \succ \equiv, divp \succ 0$ □

EXAMPLE 2.30 B: Boolean Ring ([Hsi85])

$$\begin{array}{lcl} \neg x & \rightarrow & x \oplus true \\ x \supset y & \rightarrow & (x \wedge y) \oplus (x \oplus true) \\ x \vee y & \rightarrow & (x \wedge y) \oplus (x \oplus y) \\ x \equiv y & \rightarrow & x \oplus (y \oplus true) \end{array}$$

POL:

$$\begin{array}{lcl} [\wedge](x,y) & = & x + y \\ [\vee](x,y) & = & 2x + 2y + 1 \\ [\equiv](x,y) & = & x + y + 2 \\ [\supset](x,y) & = & 2x + y + 2 \\ [\oplus](x,y) & = & x + y \\ [\neg](x) & = & x + 2 \\ [true]() & = & 1 \end{array}$$

RPOS: $\neg \succ \oplus, \neg \succ true, \supset \succ \oplus, \supset \succ \wedge, \supset \succ true, \vee \succ \oplus, \vee \succ \wedge, \equiv \succ \oplus, \equiv \succ true$ □

EXAMPLE 2.31 B: [Pue81]

$$\begin{array}{lcl} \neg true & \rightarrow & false \\ \neg false & \rightarrow & true \\ odd(0) & \rightarrow & false \\ odd(s(x)) & \rightarrow & \neg odd(x) \\ x + 0 & \rightarrow & x \\ x + s(y) & \rightarrow & s(x + y) \\ s(x) + y & \rightarrow & s(x + y) \end{array}$$

KBOS: $\varphi(s) > \varphi(\neg) > \varphi(false), \varphi(\neg) > \varphi(true)$
$+ \succ s$

POL:

$$\begin{array}{lcl} [+](x,y) & = & 3x + 2y \\ [\neg](x) & = & x + 1 \\ [odd](x) & = & x + 1 \\ [s](x) & = & x + 2 \\ [0]() & = & 1 \\ [false]() & = & 1 \\ [true]() & = & 1 \end{array}$$

RPOS: $odd \succ \neg \succ false, \neg \succ true, + \succ s$ □

EXAMPLE 2.32 B: [Mus80]

$$\begin{array}{lcll}
\neg x & \to & if(x, false, true) & \\
x \wedge y & \to & if(x, y, false) & \\
x \vee y & \to & if(x, true, y) & \\
x \supset y & \to & if(x, y, true) & \\
x \equiv x & \to & true & \diamond \\
x \equiv y & \to & if(x, y, \neg y) & \diamond \\
if(true, x, y) & \to & x & \\
if(false, x, y) & \to & y &
\end{array}$$

$$\begin{array}{lcl}
if(x, x, if(x, false, true)) & \to & true \\
x \equiv y & \to & if(x, y, if(y, false, true))
\end{array}$$

POL:
$$\begin{array}{lcl}
[\wedge](x,y) & = & x+y+2 \\
[\vee](x,y) & = & x+y+2 \\
[\equiv](x,y) & = & x+2y+3 \\
[\supset](x,y) & = & x+y+2 \\
[\neg](x) & = & x+3 \\
[if](x,y,z) & = & x+y+z \\
[false]() & = & 1 \\
[true]() & = & 1
\end{array}$$

RPOS: $\wedge \succ if, \wedge \succ false, \vee \succ if, \vee \succ true, \supset \succ if, \supset \succ true, \equiv \succ \neg \succ if, \neg \succ false, \neg \succ true$ □

EXAMPLE 2.33 B: Disjunctive Normal Form[18]

$$\begin{array}{lcl}
x \vee x & \to & x \\
x \wedge x & \to & x \\
\neg\neg x & \to & x \\
\neg(x \wedge y) & \to & (\neg x) \vee (\neg y) \\
\neg(x \vee y) & \to & (\neg x) \wedge (\neg y)
\end{array}$$

KBOS: $\varphi(\neg) = 0$
$\neg \succ \vee, \neg \succ \wedge$

POL:
$$\begin{array}{lcl}
[\wedge](x,y) & = & x+y \\
[\vee](x,y) & = & x+y \\
[\neg](x) & = & x^2
\end{array}$$

RPOS: $\neg \succ \vee, \neg \succ \wedge$ □

EXAMPLE 2.34 B: If-then-else

$$\begin{array}{lcl}
if(true, x, y) & \to & x \\
if(false, x, y) & \to & y \\
if(x, y, y) & \to & y \\
if(if(x, y, z), u, v) & \to & if(x, if(y, u, v), if(z, u, v))
\end{array}$$

---

[18] The orientation of the two last rules in reverse order implies divergence.

POL: $[if](x,y,z) = xy + xz + x$
$[false]() = 1$
$[true]() = 1$

RPOS: $\tau(if) = left$ □

EXAMPLE 2.35 B: Implication ([Ste89c])

$$\begin{array}{lcl} x \wedge false & \to & false \\ x \wedge (\neg false) & \to & x \\ \neg\neg x & \to & x \\ false \supset y & \to & \neg false \\ x \supset false & \to & \neg x \\ (\neg x) \supset (\neg y) & \to & y \supset (x \wedge y) \end{array}$$

POL: $[\wedge](x,y) = x + y$
$[\supset](x,y) = 2x + 2y$
$[\neg](x) = 2x$
$[false]() = 1$

IRDS: $\supset \succ \neg \succ \wedge$ □

EXAMPLE 2.36 B: Implication

$$\begin{array}{lcl} (\neg x) \supset y & \to & x \vee y \\ (\neg x) \supset (y \vee z) & \to & y \supset (x \vee z) \quad \diamond \end{array}$$

$$x \supset (y \vee z) \to y \vee (x \supset z)$$

KBOS: $\varphi(\neg) + \varphi(\supset) > \varphi(\vee), \varphi(\neg) > 0$
$\supset \succ \vee$

POL: $[\vee](x,y) = x + y$
$[\supset](x,y) = x + 2y$
$[\neg](x) = 3x$

IRDS: $\neg \succ \supset \succ \vee$ □

EXAMPLE 2.37 B: Ternary $\wedge$-Operator ([Ste88])

$$and(\neg\neg x, y, \neg z) \to and(y, x \mathbin{\bar{\wedge}} z, x)$$

POL: $[and](x,y,z) = x + y + z$
$[\bar{\wedge}](x,y) = x + y$
$[\neg](x) = 2x$

IRDS: $\neg \succ \bar{\wedge}$ □

EXAMPLE 2.38 L: Functional List Append[19] ([DP88])

$$\begin{array}{lcl} nil \circ y & \rightarrow & y \\ x \circ nil & \rightarrow & x \\ (x.y) \circ z & \rightarrow & x.(y \circ z) \\ (x \circ y) \circ z & \rightarrow & x \circ (y \circ z) \end{array}$$

KBOS: $\circ \succ .$
$\tau(\circ) = left$

POL:
$$\begin{array}{lcl} [\circ](x,y) & = & 2x + y \\ [.](x,y) & = & x + y \\ [nil]() & = & 1 \end{array}$$

RPOS: $\circ \succ .$
$\tau(\circ) = left$ □

EXAMPLE 2.39 L: List ([Ait85])

$$\begin{array}{lcl} rev(nil) & \rightarrow & nil \\ rev(x.y) & \rightarrow & rev(y) \circ (x.nil) \\ car(x.y) & \rightarrow & x \\ cdr(x.y) & \rightarrow & y \\ null(nil) & \rightarrow & true \\ null(x.y) & \rightarrow & false \\ nil \circ y & \rightarrow & y \\ (x.y) \circ z & \rightarrow & x.(y \circ z) \end{array}$$

POL:
$$\begin{array}{lcl} [\circ](x,y) & = & xy \\ [.](x,y) & = & xy + y + 1 \\ [car](x) & = & x \\ [cdr](x) & = & x \\ [null](x) & = & x + 1 \\ [rev](x) & = & x^2 \\ [nil]() & = & 2 \\ [false]() & = & 2 \\ [true]() & = & 2 \end{array}$$

RPOS: $rev \succ \circ \succ .,rev \succ nil, null \succ true, null \succ false$ □

EXAMPLE 2.40 L: Member ([HO82])

$$\begin{array}{lcl} true \vee y & \rightarrow & true \\ x \vee true & \rightarrow & true \\ false \vee false & \rightarrow & false \\ mem(x, nil) & \rightarrow & false \\ mem(x, set(y)) & \rightarrow & x \equiv y \\ mem(x, union(y,z)) & \rightarrow & mem(x,y) \vee mem(x,z) \end{array}$$

---

[19] Using the KBOS with $\tau(\circ) = left$ and $. \succ \circ$, the completion process diverges. The completion also diverges if we use the KBOS with $\tau(\circ) = right$.

POL:

$$
\begin{array}{lcl}
[\vee](x,y) & = & x+y \\
[\equiv](x,y) & = & x+y \\
[mem](x,y) & = & xy \\
[set](x) & = & x+2 \\
[union](x,y) & = & x+y+1 \\
[nil]() & = & 2 \\
[false]() & = & 1 \\
[true]() & = & 1
\end{array}
$$

RPOS: $mem \succ false, mem \succ \equiv, mem \succ \vee$ □

EXAMPLE 2.41 L: Finite Sequences of Natural Numbers ([KM87])

$$
\begin{array}{lcl}
\|nil\| & \to & 0 \\
\|g(x,y)\| & \to & s(\|x\|) \\
f(x,nil) & \to & g(nil,x) \\
f(x,g(y,z)) & \to & g(f(x,y),z) \\
rem(nil,y) & \to & nil \\
rem(g(x,y),0) & \to & g(x,y) \\
rem(g(x,y),s(z)) & \to & rem(x,z)
\end{array}
$$

KBOS: $\varphi(f) > \varphi(g) > \varphi(s), \varphi(nil) > \varphi(0)$
$f \succ g$

POL:

$$
\begin{array}{lcl}
[\|\cdot\|](x) & = & x \\
[f](x,y) & = & x+2y \\
[g](x,y) & = & x+y \\
[rem](x,y) & = & x+y \\
[s](x) & = & x \\
[0]() & = & 1 \\
[nil]() & = & 2
\end{array}
$$

RPOS: $\|.\| \succ 0, \|.\| \succ s, f \succ g$ □

EXAMPLE 2.42 L: Flattening and Reverse Operation

$$
\begin{array}{lcl}
flatten(nil) & \to & nil \\
flatten(unit(x)) & \to & flatten(x) \\
flatten(x \circ y) & \to & flatten(x) \circ flatten(y) \\
flatten(unit(x) \circ y) & \to & flatten(x) \circ flatten(y) \quad \diamond \\
flatten(flatten(x)) & \to & flatten(x) \\
rev(nil) & \to & nil \\
rev(unit(x)) & \to & unit(x) \\
rev(x \circ y) & \to & rev(y) \circ rev(x) \\
rev(rev(x)) & \to & x \\
x \circ nil & \to & x \\
nil \circ y & \to & y \\
(x \circ y) \circ z & \to & x \circ (y \circ z)
\end{array}
$$

KBOS:	$\varphi(unit) > 0, \varphi(flatten) = \varphi(rev) = 0$ $rev \sim flatten \succ \circ$[20] $\tau(\circ) = left$	
POL:	$[\circ](x,y) = 3xy + 2x$ $[flatten](x) = x^2$ $[rev](x) = x^2$ $[unit](x) = x + 1$ $[nil]() = 2$	
RPOS:	$flatten \succ \circ, rev \succ \circ$ $\tau(\circ) = left$	□

EXAMPLE 2.43 L: [Nar89]

$$
\begin{array}{lcl}
merge(nil, y) & \rightarrow & y \\
merge(x, nil) & \rightarrow & x \\
merge(x.y, u.v) & \rightarrow & if(x < u, x.merge(y, u.v), u.merge(x.y, v)) \\
nil \circ y & \rightarrow & y \\
(x.y) \circ z & \rightarrow & x.(y \circ z) \\
if(true, x, y) & \rightarrow & x \\
if(false, x, y) & \rightarrow & y
\end{array}
$$

POL:	$[\circ](x,y) = x^2y$ $[.](x,y) = x + 2y$ $[<](x,y) = x + y$ $[if](x,y,z) = x + y + z$ $[merge](x,y) = x^2y^2$ $[nil]() = 2$ $[false]() = 1$ $[true]() = 1$	
RPOS:	$merge \succ if, merge \succ <, merge \succ ., \circ \succ .$	□

EXAMPLE 2.44 L: [MS86]

$$
\begin{array}{lcl}
del(x.(y.z)) & \rightarrow & f(x \equiv y, x, y, z) \\
f(true, x, y, z) & \rightarrow & del(y.z) \\
f(false, x, y, z) & \rightarrow & x.del(y.z) \\
nil \equiv nil & \rightarrow & true \\
(x.y) \equiv nil & \rightarrow & false \\
nil \equiv (y.z) & \rightarrow & false \\
(x.y) \equiv (u.v) & \rightarrow & (x \equiv u) \wedge (y \equiv v)
\end{array}
$$

---

[20] Using a quasi-ordering on the operators, we may admit more than one unary operator with weight zero (see [Ste88]).

POL:

$$\begin{aligned}
[\wedge](x,y) &= x+y \\
[\equiv](x,y) &= xy \\
[.](x,y) &= x+y \\
[del](x) &= x^2 \\
[f](x,y,z,u) &= x+y+z^2+2zu+u^2 \\
[nil]() &= 2 \\
[false]() &= 2 \\
[true]() &= 2
\end{aligned}$$

□

EXAMPLE 2.45 L: [Nar89]

$$\begin{aligned}
admit(x, nil) &\to nil \\
admit(x, u.v.w.z) &\to cond(sum(x,u,v) \equiv w, u.v.w.admit(carry(x,u,v),z)) \\
cond(true, y) &\to y
\end{aligned}$$

POL:

$$\begin{aligned}
[\equiv](x,y) &= x+y \\
[.](x,y) &= xy \\
[admit](x,y) &= xy^3 \\
[carry](x,y,z) &= x+y+z \\
[cond](x,y) &= x+y \\
[sum](x,y,z) &= x+y+z \\
[nil]() &= 2 \\
[true]() &= 2
\end{aligned}$$

RPOS: $admit \succ cond, admit \succ sum, admit \succ \equiv, admit \succ . \succ carry$
$\tau(admit) = right$

□

EXAMPLE 2.46 S:

$$\begin{aligned}
aa &\to bb \\
bba &\to abb
\end{aligned}$$

KBOS: $\varphi(a) > \varphi(b)$
$b \succ a$

POL:

$$\begin{aligned}
[a](x) &= 2x+1 \\
[b](x) &= 2x
\end{aligned}$$

□

EXAMPLE 2.47 S: [HP86][21]

$$\begin{aligned}
ab &\to ba \\
ac &\to \lambda
\end{aligned}$$

KBOS: $a \succ b$

POL:

$$\begin{aligned}
[a](x) &= 2x \\
[b](x) &= x+1 \\
[c](x) &= x
\end{aligned}$$

[21]The process diverges if the first rule is oriented in reverse order. Note that λ denotes the empty word.

RPOS: $a \succ b$ □

EXAMPLE 2.48 S: [HP86][22]

$$\begin{array}{lll} d & \rightarrow & eu \\ du & \rightarrow & c \quad \diamond \\ cu & \rightarrow & b \quad \diamond \\ ve & \rightarrow & \lambda \\ bu & \rightarrow & ae \quad \diamond \end{array}$$

$$\begin{array}{lll} c & \rightarrow & euu \\ b & \rightarrow & euuu \\ ae & \rightarrow & euuuu \end{array}$$

KBOS: $\varphi(e) = \varphi(u) = 1, \varphi(a) = 5, \varphi(b) = \varphi(c) = \varphi(d) = 6$

POL:
$$\begin{array}{lll} [a](x) & = & x+1 \\ [b](x) & = & x+1 \\ [c](x) & = & x+1 \\ [d](x) & = & x+1 \\ [e](x) & = & x \\ [u](x) & = & x \\ [v](x) & = & x+1 \end{array}$$

RPOS: $c \succ b \succ a \succ e \succ d \succ u$ □

EXAMPLE 2.49 S: Monoid

$$\begin{array}{lll} acd & \rightarrow & c \\ ubdd & \rightarrow & b \\ vaa & \rightarrow & uv \\ vac & \rightarrow & ubd \\ vc & \rightarrow & b \\ waa & \rightarrow & uw \\ wac & \rightarrow & ubd \\ wc & \rightarrow & b \end{array}$$

KBOS: $\varphi(b) = \varphi(d) = \varphi(u) = \varphi(w) = 1, \varphi(a) = \varphi(v) = 2, \varphi(c) = 3$

POL:
$$\begin{array}{lll} [a](x) & = & x+1 \\ [b](x) & = & x \\ [c](x) & = & x \\ [d](x) & = & x+1 \\ [u](x) & = & x \\ [v](x) & = & x+1 \\ [w](x) & = & x+1 \end{array}$$

[22]The original system of [HP86] is parameterized, such that $a_{n+1}a_{n-1} \rightarrow \lambda$, $a_1 a_n \rightarrow a_0 a_{n-1}$, $a_i a_n \rightarrow a_{i-1}$ for all $i = n-1, ..., 2 (n \geq 3)$. The termination of this general system can be proved with the help of the following polynomial ordering: $[a_i](x) = x+1$ for all $i = 0, ..., n-2, n+1$ and $[a_i](x) = x$ for $i = n-1, n$.

RPOS: $v \succ u, v \succ b, v \succ d, w \succ u, w \succ b, w \succ d$ □

EXAMPLE 2.50 S:

$$ab \rightarrow bba$$

POL:

$$[a](x) = 3x$$
$$[b](x) = x+1$$

RPOS: $a \succ b$ □

EXAMPLE 2.51 O: Ackermann Function

$$ack(0,y) \rightarrow s(y)$$
$$ack(s(x),0) \rightarrow ack(x,s(0))$$
$$ack(s(x),s(y)) \rightarrow ack(x,ack(s(x),y))$$

RPOS: $ack \succ s$

$\tau(ack) = left$ □

EXAMPLE 2.52 O: [Thi84]

$$f(x,0,0) \rightarrow s(x)$$
$$f(0,y,0) \rightarrow s(y)$$
$$f(0,0,z) \rightarrow s(z)$$
$$f(s(0),y,z) \rightarrow f(0,s(y),s(z))$$
$$f(s(x),s(y),0) \rightarrow f(x,y,s(0))$$
$$f(s(x),0,s(z)) \rightarrow f(x,s(0),z)$$
$$f(0,s(0),s(0)) \rightarrow s(s(0))$$
$$f(s(x),s(y),s(z)) \rightarrow f(x,y,f(s(x),s(y),z))$$
$$f(0,s(s(y)),s(0)) \rightarrow f(0,y,s(0))$$
$$f(0,s(0),s(s(z))) \rightarrow f(0,s(0),z)$$
$$f(0,s(s(y)),s(s(z))) \rightarrow f(0,y,f(0,s(s(y)),s(z)))$$

RPOS: $f \succ s$

$\tau(f) = left$ □

EXAMPLE 2.53 O: [BD86]

$$f(x,y) \rightarrow g(x,y)$$
$$g(h(x),y) \rightarrow h(f(x,y)) \quad \diamond$$

$$g(h(x),y) \rightarrow h(g(x,y))$$

KBOS: $\varphi(f) = \varphi(g)$

$f \succ g \succ h$

POL: $[f](x,y) = 2x+y+1$
$[g](x,y) = 2x+y$
$[h](x) = x+2$ □

EXAMPLE 2.54 O: [Gra88][23]

$$f(x,a) \rightarrow x$$
$$f(x,g(y)) \rightarrow f(g(x),y)$$

KBOS: $\tau(f) = right$

POL: $[f](x,y) = x+2y$
$[g](x) = x+1$
$[a]() = 1$

RPOS: $f \succ g$
$\tau(f) = right$ □

EXAMPLE 2.55 O: [Gra88][24]

$$f(x,g(x)) \rightarrow x$$
$$f(x,h(y)) \rightarrow f(h(x),y)$$

KBOS: $\tau(f) = right$

POL: $[f](x,y) = x+2y$
$[g](x) = x$
$[h](x) = x+1$

RPOS: $f \succ h$
$\tau(f) = right$ □

EXAMPLE 2.56 O: [You88]

$$f(a,x) \rightarrow g(a,x) \quad \diamond$$
$$g(a,x) \rightarrow f(b,x)$$

$$f(a,x) \rightarrow f(b,x)$$

KBOS: $\varphi(f) = \varphi(g), \varphi(a) > \varphi(b)$
$f \succ g$

POL: $[f](x,y) = 2x+y$
$[g](x,y) = x+y$
$[a]() = 3$
$[b]() = 1$ □

EXAMPLE 2.57 O: [RJ81]

$$f(g(h(x,y)),f(a,a)) \rightarrow f(h(x,x),g(f(y,a)))$$

---

[23] If the last rule is oriented from right to left, the completion process will diverge.
[24] The orientation of the second rule in reverse order implies divergence.

POL: $[f](x,y) = x + y$
$[g](x) = 2x$
$[h](x,y) = x + y + 1$
$[a]() = 1$

RPOS: $f \succ g \succ h$
$\tau(f) = left$

□

EXAMPLE 2.58 O: [Mid88]

$$f(x,y) \rightarrow x$$
$$g(a) \rightarrow h(a,b,a)$$
$$i(x) \rightarrow f(x,x) \qquad \diamond$$
$$h(x,x,y) \rightarrow g(x)$$

$$i(x) \rightarrow x$$

POL: $[f](x,y) = x + y$
$[g](x) = x^2$
$[h](x,y,z) = x + y^2 + z$
$[i](x) = 2x + 1$
$[a]() = 3$
$[b]() = 1$

□

EXAMPLE 2.59 O: [Lév86]

$$f(g(x),y,y) \rightarrow g(f(x,x,y))$$

POL: $[f](x,y,z) = x^2 + y + z$
$[g](x) = x + 1$

RPOS: $f \succ g$

□

EXAMPLE 2.60 O:

$$f(g(f(a),h(a,f(a)))) \rightarrow f(h(g(f(a),a),g(f(a),f(a))))$$

POL: $[f](x) = x$
$[g](x,y) = xy$
$[h](x,y) = x + y + 1$
$[a]() = 2$

RPOS: $g \succ h$

□

EXAMPLE 2.61 O: [Lan89]

$$
\begin{array}{lcl}
f(j(x,y),y) & \to & g(f(x,k(y))) \\
f(x,h_1(y,z)) & \to & h_2(0,x,h_1(y,z)) \\
g(h_2(x,y,h_1(z,u))) & \to & h_2(s(x),y,h_1(z,u)) \\
h_2(x,j(y,h_1(z,u)),h_1(z,u)) & \to & h_2(s(x),y,h_1(s(z),u)) \\
i(f(x,h(y))) & \to & y \\
i(h_2(s(x),y,h_1(x,z))) & \to & z \\
k(h(x)) & \to & h_1(0,x) \\
k(h_1(x,y)) & \to & h_1(s(x),y)
\end{array}
$$

KBOS: $\varphi(h)=\varphi(h_1)=\varphi(h_2)=\varphi(s)=\varphi(0)=1, \varphi(g)=\varphi(k)=2, \varphi(f)=3, \varphi(j)=4$

POL:

$$
\begin{array}{lcl}
[f](x,y) & = & x+y+2 \\
[g](x) & = & x+1 \\
[h](x) & = & x+1 \\
[h_1](x,y) & = & x+y \\
[h_2](x,y,z) & = & x+y+z \\
[i](x) & = & x \\
[j](x,y) & = & x+y+2 \\
[k](x) & = & x+1 \\
[s](x) & = & x \\
[0]() & = & 1
\end{array}
$$

□

3 Systems to Complete

EXAMPLE 3.1 G: [Les84]

$$\begin{array}{lcl} 1 * y & \to & y \\ i(x) * x & \to & 1 \\ (x * y) * z & \to & x * (y * z) \\ x/y & \to & x * i(y) \end{array}$$

$$\begin{array}{lcl} x * 1 & \to & x \\ i(1) & \to & 1 \\ i(i(x)) & \to & x \\ x * i(x) & \to & 1 \\ i(x * y) & \to & i(y) * i(x) \\ i(x) * (x * y) & \to & y \\ x * (i(x) * y) & \to & y \end{array}$$

KBOS: $\varphi(*) = \varphi(i) = 0, \varphi(/) = \varphi(1) = 1$
$\tau(*) = left$

POL:
$$\begin{array}{lcl} [*](x,y) & = & 2xy + x \\ [/](x,y) & = & 2xy^2 + x + 1 \\ [i](x) & = & x^2 \\ [1]() & = & 2 \end{array}$$

RPOS: $/ \succ i \succ *, i \succ 1$
$\tau(*) = left$

□

EXAMPLE 3.2 G: [BL88]

$$\begin{array}{lcll} f(f(x)) & \to & x & \\ f(x) + f(y) & \to & f(x+y) & \diamond \\ (x+y)+z & \to & x + (y+z) & \end{array}$$

$$f(x) + y \to f(x + f(y))$$

POL:
$$\begin{array}{lcl} [+](x,y) & = & 3x + y \\ [f](x) & = & x + 1 \end{array}$$

RPOS: $+ \succ f$
$\tau(+) = left$

□

EXAMPLE 3.3 G: Group ([KS83])

$$\begin{array}{lcll} (x/x)/((y/y)/y) & \to & y & \diamond \\ (x/y)/(z/y) & \to & x/z & \diamond \\ x/x & \to & 1 & \diamond \\ 1/y & \to & i(y) & \diamond \\ x/i(y) & \to & x * y & \diamond \end{array}$$

$$
\begin{array}{lll}
i(1) & \rightarrow & 1 \\
i(i(x)) & \rightarrow & x \\
i(x*y) & \rightarrow & i(y)*i(x) \\
i(x)*x & \rightarrow & 1 \\
x*i(x) & \rightarrow & 1 \\
x*1 & \rightarrow & x \\
1*y & \rightarrow & y \\
(x*y)*z & \rightarrow & x*(y*z) \\
x*(i(x)*y) & \rightarrow & y \\
i(x)*(x*y) & \rightarrow & y \\
x/y & \rightarrow & x*i(y)
\end{array}
$$

KBOS: $\varphi(*) = \varphi(/) = \varphi(i) = 0, \varphi(1) = 1$
$/ \succ *, i \succ *$
$\tau(*) = left$

POL:

$$
\begin{array}{lll}
[*](x,y) & = & 2xy + x \\
[/](x,y) & = & 2xy^2 + x + 1 \\
[i](x) & = & x^2 \\
[1]() & = & 2
\end{array}
$$

RPOS: $/ \succ i \succ * \succ 1$
$\tau(*) = left$

□

EXAMPLE 3.4 G: Group ([KB70])

$$
\begin{array}{lll}
(x*y)*z & \rightarrow & x*(y*z) \\
1*y & \rightarrow & y \\
g(x) & \rightarrow & f(x)*x \\
g(x)*y & \rightarrow & y \quad \diamond
\end{array}
$$

$$
\begin{array}{lll}
f(x)*(x*y) & \rightarrow & y \\
f(1)*y & \rightarrow & y \\
f(f(x))*y & \rightarrow & x*y \\
x*(f(x)*y) & \rightarrow & y \\
f(x*y)*z & \rightarrow & f(y)*(f(x)*z)
\end{array}
$$

POL:

$$
\begin{array}{lll}
[*](x,y) & = & 2x + y \\
[f](x) & = & 2x \\
[g](x) & = & 5x + 1 \\
[1]() & = & 1
\end{array}
$$

RPOS: $g \succ f \succ *$
$\tau(*) = left$

□

EXAMPLE 3.5 G: Central Groupoid ([KB70])

$$
\begin{array}{lll}
(x*y)*(y*z) & \rightarrow & y \quad \diamond \\
(x*x)*x & \rightarrow & f(x) \quad \diamond \\
x*(x*x) & \rightarrow & g(x) \quad \diamond \\
g(x)*y & \rightarrow & x*y
\end{array}
$$

$f(f(x))$	$\rightarrow$	$f(x)$
$f(g(x))$	$\rightarrow$	$g(x)$
$g(g(x))$	$\rightarrow$	$g(x)$
$g(f(x))$	$\rightarrow$	$f(x)$
$f(x) * x$	$\rightarrow$	$f(x)$
$f(x) * g(x)$	$\rightarrow$	x
$x * f(y)$	$\rightarrow$	$x * y$
$f(x * y)$	$\rightarrow$	$g(x)$
$x * g(x)$	$\rightarrow$	$g(x)$
$g(x * y)$	$\rightarrow$	$f(y)$
$x * (y * z)$	$\rightarrow$	$x * g(y)$
$(x * y) * z$	$\rightarrow$	$f(y) * z$

KBOS: $\varphi(*) = \varphi(f) = \varphi(g) = 1$
$* \succ g$

POL:
$$[*](x,y) = x + y$$
$$[f](x) = x + 1$$
$$[g](x) = x + 1$$

RPOS: $* \succ f, * \succ g$ □

EXAMPLE 3.6 G: Taussky Group ([KB70])

$x * (y * z)$	$\rightarrow$	$(x * y) * z$	
$1 * 1$	$\rightarrow$	1	$\diamond$
$x * i(x)$	$\rightarrow$	1	
$g(x * y, y)$	$\rightarrow$	$f(x * y, x)$	$\diamond$
$f(1, y)$	$\rightarrow$	y	

$x * 1$	$\rightarrow$	x
$1 * y$	$\rightarrow$	y
$i(i(x))$	$\rightarrow$	x
$i(x * y)$	$\rightarrow$	$i(y) * i(x)$
$i(x) * x$	$\rightarrow$	1
$g(x, y)$	$\rightarrow$	$f(x, x * i(y))$
$(x * y) * i(y)$	$\rightarrow$	x
$(x * i(y)) * y$	$\rightarrow$	x

POL:
$$[*](x,y) = 2xy + y$$
$$[f](x,y) = x + y$$
$$[g](x,y) = x + y^2 + 2xy^2 + 1$$
$$[i](x) = x^2$$
$$[1]() = 2$$

RPOS: $g \succ i \succ * \succ 1, g \succ f \succ *$
$\tau(*) = right$ □

EXAMPLE 3.7 G: [Sak84]

$$
\begin{array}{lll}
(x+y)+z & \to & x+(y+z) \\
0+0 & \to & 0 \quad \diamond \\
x+(-x) & \to & 0 \\
f(0,y,z) & \to & y \\
g(x+y,y) & \to & f(x+y,x,y) \quad \diamond
\end{array}
$$

$$
\begin{array}{lll}
-0 & \to & 0 \\
--x & \to & x \\
-(x+y) & \to & (-y)+(-x) \\
x+0 & \to & x \\
0+y & \to & y \\
(-x)+x & \to & 0 \\
(-x)+(x+y) & \to & y \\
x+((-x)+y) & \to & y \\
g(x,y) & \to & f(x,x+(-y),y) .
\end{array}
$$

POL:

$$
\begin{array}{lll}
[+](x,y) & = & 2xy+x \\
[-](x) & = & x^2 \\
[f](x,y,z) & = & x+y+z \\
[g](x,y) & = & 2xy^2+2x+y+1 \\
[0]() & = & 2
\end{array}
$$

RPOS: $g \succ - \succ + \succ 0, g \succ f$
$\tau(+) = left$ □

EXAMPLE 3.8 G: Quasigroup ([Hul80b])

$$
\begin{array}{lll}
x*(x\backslash y) & \to & y \quad \diamond \\
x\backslash(x*y) & \to & y \quad \diamond \\
(x*y)/y & \to & x \quad \diamond \\
(x/y)*y & \to & x \quad \diamond \\
x*(y*x) & \to & y
\end{array}
$$

$$
\begin{array}{lll}
x\backslash y & \to & y*x \\
y/x & \to & x*y \\
(x*y)*x & \to & y
\end{array}
$$

KBOS: $\varphi(*) = \varphi(\backslash) = \varphi(/) = 1$
$\backslash \succ *, / \succ *$

POL:

$$
\begin{array}{lll}
[*](x,y) & = & x+y \\
[\backslash](x,y) & = & x+y+1 \\
[/](x,y) & = & x+y+1
\end{array}
$$

RPOS: $\backslash \succ *, / \succ *$ □

EXAMPLE 3.9 G: [Chr89][25]

$$
\begin{array}{lll}
f(f(x,y),z) & \rightarrow & f(x,f(y,z)) \\
f(a_k,y) & \rightarrow & y \qquad for\ k=1,...,12 \\
f(x,i_1(x)) & \rightarrow & a_1 \\
f(x,i_k(x)) & \rightarrow & a_k \qquad for\ k=2,...,12 \qquad \diamond
\end{array}
$$

$$
\begin{array}{lll}
i_k(x) & \rightarrow & f(i_1(x),a_k) \qquad for\ k=2,...,12 \\
i_1(a_k) & \rightarrow & a_1 \qquad for\ k=1,...,12 \\
i_1(i_1(x)) & \rightarrow & f(x,a_1) \\
i_1(f(x,y)) & \rightarrow & f(i_1(y),i_1(x)) \\
f(x,f(i_1(x),y)) & \rightarrow & y \\
f(i_1(x),f(x,y)) & \rightarrow & y \\
f(i_1(x),a_1) & \rightarrow & i_1(x)
\end{array}
$$

KBOS: $\varphi(f)=\varphi(i_1)=0$
$\varphi(a_k)=1 \qquad for\ k=1,...,12$
$\varphi(i_k)=2 \qquad for\ k=2,...,12$
$i_1 \succ a_1$
$\tau(f)=left$

POL:

$$
\begin{array}{lll}
[f](x,y) & = & 2xy+x \\
[i_1](x) & = & x^2 \\
[i_k](x) & = & 5x^2+1 \qquad for\ k=2,...,12 \\
[a_k]() & = & 2 \qquad for\ k=1,...,12
\end{array}
$$

RPOS: $i_2 \succ i_1$
$i_3 \succ i_k \succ i_1 \succ f \succ a_k \qquad for\ k=1,...,12$
$\tau(f)=left$

□

EXAMPLE 3.10 G: Cancellation Law ([KB70])

$$
\begin{array}{lll}
f(x,x*y) & \rightarrow & y \\
g(x*y,y) & \rightarrow & x \\
x*1 & \rightarrow & x \\
1*y & \rightarrow & y
\end{array}
$$

$$
\begin{array}{lll}
f(1,y) & \rightarrow & y \\
f(x,x) & \rightarrow & 1 \\
g(x,1) & \rightarrow & x \\
g(x,x) & \rightarrow & 1
\end{array}
$$

KBOS: $\varphi(*)=\varphi(f)=\varphi(g)=\varphi(1)=1$

POL:

$$
\begin{array}{lll}
[*](x,y) & = & x+y \\
[f](x,y) & = & x+y \\
[g](x,y) & = & x+y \\
[1]() & = & 1
\end{array}
$$

[25] Using the KBOS, the canonical system is supplemented by $i_1(i_1(i_1(x))) \rightarrow i_1(x)$ and the equation $i_1(i_1(x)) = f(x,a_1)$ is oriented in reverse order.

RPOS: $f \succ 1, g \succ 1$ □

EXAMPLE 3.11 A: [BBK87]

$p(0)$	$\rightarrow$	0	$\diamond$
$p(s(x))$	$\rightarrow$	x	$\diamond$
$x+0$	$\rightarrow$	x	
$s(x+p(y))$	$\rightarrow$	$x+y$	$\diamond$

$p(x)$	$\rightarrow$	x
$s(x)$	$\rightarrow$	x

KBOS: $\varphi(+) = \varphi(p) = \varphi(s) = \varphi(0) = 1$

POL:
$$[+](x,y) = x+y$$
$$[p](x) = x+1$$
$$[s](x) = x+1$$
$$[0]() = 1$$

RPOS: empty $\succ$ □

EXAMPLE 3.12 A: Division ([Dic87])

x/x	$\rightarrow$	1
$x/1$	$\rightarrow$	x
$i(x/y)$	$\rightarrow$	y/x
$(x/y)/z$	$\rightarrow$	$x/(z/i(y))$

$i(1)$	$\rightarrow$	1
$1/y$	$\rightarrow$	$i(y)$
$i(i(x))$	$\rightarrow$	x
$x/(y/i(x))$	$\rightarrow$	$i(y)$
$i(x)/(y/x)$	$\rightarrow$	$i(y)$
$x/(y/(i(x)/z))$	$\rightarrow$	$i(z)/y$
$i(x)/(y/(x/z))$	$\rightarrow$	$i(z)/y$

KBOS: $\varphi(/) = \varphi(i) = 0, \varphi(1) = 1$
$i \succ /$
$\tau(/) = left$ □

EXAMPLE 3.13 A: [Zeh89]

$(x+y)+z$	$\rightarrow$	$x+(y+z)$
$x+0$	$\rightarrow$	x
$(x*y)*z$	$\rightarrow$	$x*(y*z)$
$x*1$	$\rightarrow$	x
$exp(0)$	$\rightarrow$	1
$exp(x+y)$	$\rightarrow$	$exp(x)*exp(y)$

$x+(0+y)$	$\rightarrow$	$x+y$
$x*(1*y)$	$\rightarrow$	$x*y$

KBOS: $\varphi(exp) = 0, \varphi(+) = \varphi(*) = \varphi(0) = \varphi(1) = 1$
$exp \succ *, exp \succ 1$
$\tau(+) = \tau(*) = left$

POL:

$$\begin{aligned} [+](x,y) &= 3x + y \\ [*](x,y) &= 2x + y \\ [exp](x) &= 2x \\ [0]() &= 1 \\ [1]() &= 1 \end{aligned}$$

RPOS: $exp \succ *, exp \succ 1$
$\tau(+) = \tau(*) = left$

□

EXAMPLE 3.14 A: [KNZ86]

$$\begin{aligned} s(s(x)) &\rightarrow x \\ f(0,y) &\rightarrow y \\ f(s(x),y) &\rightarrow s(f(x,y)) \\ f(f(g(x,y),0),0) &\rightarrow g(x,y) \quad \diamond \\ g(0,y) &\rightarrow y \\ g(s(x),y) &\rightarrow f(g(x,y),0) \\ h(0) &\rightarrow s(0) \end{aligned}$$

$$f(f(x,0),0) \rightarrow x$$

KBOS: $\varphi(f) = 0, \varphi(g) = \varphi(0) = 1, \varphi(s) = 2, \varphi(h) = 3$
$f \succ s$

POL:

$$\begin{aligned} [f](x,y) &= xy + x \\ [g](x,y) &= xy + x \\ [h](x) &= 2x + 2 \\ [s](x) &= 2x + 1 \\ [0]() &= 1 \end{aligned}$$

RPOS: $g \succ f \succ s, g \succ 0, h \succ s$

□

EXAMPLE 3.15 A: Integers With Equality (Boolean Algebra) ([Com86])

$$\begin{aligned} p(s(x)) &\rightarrow x \\ x \equiv x &\rightarrow true \\ s(x) \equiv x &\rightarrow false \quad \diamond \\ x \equiv s(x) &\rightarrow false \quad \diamond \\ s(x) \equiv s(y) &\rightarrow x \equiv y \quad \diamond \\ x \equiv p(x) &\rightarrow false \\ p(x) \equiv x &\rightarrow false \\ p(x) \equiv p(y) &\rightarrow x \equiv y \end{aligned}$$

$$\begin{aligned} x \equiv s(y) &\rightarrow p(x) \equiv y \\ s(x) \equiv y &\rightarrow x \equiv p(y) \end{aligned}$$

KBOS: $\varphi(\equiv) = \varphi(p) = \varphi(false) = \varphi(true) = 1, \varphi(s) = 2$

POL:

$$\begin{aligned} [\equiv](x,y) &= x+y \\ [p](x) &= x+1 \\ [s](x) &= x+2 \\ [false]() &= 1 \\ [true]() &= 1 \end{aligned}$$

□

EXAMPLE 3.16 A: [Bur68]

$$\begin{aligned} car(x.y) &\to x \\ cdr(x.y) &\to y \\ car(x).cdr(x) &\to x \\ atom(x.y) &\to false \quad \diamond \end{aligned}$$

$$atom(x) \to false$$

KBOS: $\varphi(.) = \varphi(atom) = \varphi(car) = \varphi(cdr) = \varphi(false) = 1$

POL:

$$\begin{aligned} [.](x,y) &= x+y \\ [atom](x) &= x+1 \\ [car](x) &= x \\ [cdr](x) &= x \\ [false]() &= 1 \end{aligned}$$

RPOS: $atom \succ false$

□

EXAMPLE 3.17 B:

$$\begin{aligned} (x \wedge y) \vee (z \wedge y) &\to (x \vee z) \wedge y \\ x \wedge x &\to x \\ x \vee x &\to x \end{aligned}$$

$$\begin{aligned} (x \wedge y) \vee y &\to (x \vee y) \wedge y \\ x \vee (y \wedge x) &\to (x \vee y) \wedge x \end{aligned}$$

KBOS: $\varphi(\wedge) = \varphi(\vee) = 1$
$\vee \succ \wedge$

POL:

$$\begin{aligned} [\wedge](x,y) &= x+y+1 \\ [\vee](x,y) &= 2x+2y \end{aligned}$$

□

EXAMPLE 3.18 L: Reverse ([Küc87])

$$\begin{aligned} nil \circ y &\to y \\ (x.y) \circ z &\to x.(y \circ z) \\ rev(nil) &\to nil \\ rev(x.y) &\to rev(y) \circ (x.nil) \\ rev(rev(x)) &\to x \end{aligned}$$

$$rev(x \circ (y.nil)) \to y.rev(x)$$

POL:	$[\circ](x,y) = xy$	
	$[.](x,y) = xy+y+1$	
	$[rev](x) = x^2$	
	$[nil]() = 2$	
RPOS:	$rev \succ \circ \succ . \succ nil$	□

EXAMPLE 3.19 L: [HO80]

$$\begin{array}{lll}
nil \circ y & \rightarrow & y \\
(x.y) \circ z & \rightarrow & x.(y \circ z) \\
(x \circ y) \circ z & \rightarrow & x \circ (y \circ z) \\
reviter(nil, y) & \rightarrow & y \\
reviter(x.y, z) & \rightarrow & reviter(y, x.z) \\
rev(nil) & \rightarrow & nil \quad \diamond \\
rev(x.y) & \rightarrow & rev(y) \circ (x.nil) \quad \diamond \\
rev(x) \circ y & \rightarrow & reviter(x, y) \quad \diamond \\
rev(x) & \rightarrow & reviter(x, nil)
\end{array}$$

$$reviter(x,y) \circ z \rightarrow reviter(x, y \circ z)$$

POL:	$[\circ](x,y) = xy+x$	
	$[.](x,y) = xy+y+1$	
	$[rev](x) = 2x^2$	
	$[reviter](x,y) = xy+x$	
	$[nil]() = 2$	
RPOS:	$rev \succ \circ \succ reviter \succ . \succ nil$	
	$\tau(\circ) = \tau(reviter) = left$	□

EXAMPLE 3.20 L: [Mus80]

$$\begin{array}{lll}
x \equiv x & \rightarrow & true \\
nil \equiv end(y,z) & \rightarrow & false \\
end(x,y) \equiv nil & \rightarrow & false \\
end(x,y) \equiv end(u,v) & \rightarrow & (y \equiv v) \wedge (x \equiv u) \\
f(x, nil) & \rightarrow & end(nil, x) \\
f(x, end(y,z)) & \rightarrow & end(f(x,y), z) \\
nil.y & \rightarrow & y \\
end(x,y).z & \rightarrow & x.f(y,z) \\
null(nil) & \rightarrow & true \\
null(end(x,y)) & \rightarrow & false
\end{array}$$

$$true \wedge true \rightarrow true$$

KBOS: $\varphi(\wedge) = 0, \varphi(\equiv) = \varphi(.) = \varphi(end) = \varphi(f) = \varphi(null) = \varphi(nil) = \varphi(false) = \varphi(true) = 1$
$\tau(.) = left$

POL:

$$\begin{array}{lcl} [\wedge](x,y) & = & xy \\ [\equiv](x,y) & = & xy \\ [.](x,y) & = & x^2y \\ [end](x,y) & = & xy+1 \\ [f](x,y) & = & 2xy \\ [null](x) & = & x+1 \\ [nil]() & = & 2 \\ [false]() & = & 2 \\ [true]() & = & 2 \end{array}$$

RPOS: $. \succ f \succ end, \equiv \succ false, \equiv \succ true, null \succ false, null \succ true$
$\tau(.) = left$ □

EXAMPLE 3.21 L: Finite Sequences of Natural Numbers ([KM87])

$$\begin{array}{lcll} f(x,nil) & \rightarrow & g(nil,x) & \diamond \\ f(x,g(y,z)) & \rightarrow & g(f(x,y),z) & \diamond \\ g(g(x,y),y) & \rightarrow & g(x,y) & \diamond \\ g(cdr(g(x,y)),y) & \rightarrow & cdr(g(x,y)) & \diamond \\ x \circ nil & \rightarrow & x & \\ x \circ g(y,z) & \rightarrow & g(x \circ y, z) & \diamond \\ cdr(nil) & \rightarrow & nil & \\ cdr(g(nil,y)) & \rightarrow & nil & \diamond \\ cdr(g(g(x,y),z)) & \rightarrow & g(cdr(g(x,y)),z) & \diamond \end{array}$$

$$\begin{array}{lcl} f(x,nil) & \rightarrow & nil \\ g(x,y) & \rightarrow & x \end{array}$$

KBOS: $\varphi(\circ) = \varphi(cdr) = \varphi(f) = \varphi(g) = \varphi(nil) = 1$
$f \succ g, \circ \succ g, cdr \succ g$

POL:

$$\begin{array}{lcl} [\circ](x,y) & = & x+2y \\ [cdr](x) & = & 2x \\ [f](x,y) & = & x+2y \\ [g](x,y) & = & x+y \\ [nil]() & = & 2 \end{array}$$

RPOS: $f \succ g, \circ \succ g, cdr \succ g$ □

EXAMPLE 3.22 S: Fibonacci Group

$$\begin{array}{lcll} ae & \rightarrow & d & \diamond \\ ba & \rightarrow & e & \diamond \\ a & \rightarrow & cb & \diamond \\ b & \rightarrow & dc & \diamond \\ c & \rightarrow & ed & \diamond \end{array}$$

$$\begin{array}{lcl} a & \rightarrow & eee \\ b & \rightarrow & eeeeeeeee \\ c & \rightarrow & eeeee \\ d & \rightarrow & eeee \\ eeeeeeeeeee & \rightarrow & e \end{array}$$

KBOS: $\varphi(e) = 1, \varphi(d) = 5, \varphi(c) = 7, \varphi(b) = 13, \varphi(a) = 21$
$d \succ e$

POL:
$$\begin{aligned}[a](x) &= 8x + 41\\ [b](x) &= 4x + 17\\ [c](x) &= 2x + 6\\ [d](x) &= 2x + 4\\ [e](x) &= x + 1\end{aligned}$$

RPOS: $a \succ b \succ c \succ d \succ e$ □

EXAMPLE 3.23 S: [HP86]

$$\begin{aligned} bd &\to \lambda\\ da &\to e\\ ec &\to a\\ ce &\to a \end{aligned}$$

$$\begin{aligned} be &\to a\\ ca &\to ac\\ ea &\to ae\\ ba &\to ac \end{aligned}$$

KBOS: $\varphi(a) = \varphi(b) = \varphi(c) = \varphi(d) = \varphi(e) = 1$
$e \succ a, b \succ c \succ a$

POL:
$$\begin{aligned}[a](x) &= x + 1\\ [b](x) &= 2x\\ [c](x) &= 2x\\ [d](x) &= 2x\\ [e](x) &= 2x\end{aligned}$$

RPOS: $d \succ e \succ a, b \succ c \succ a$ □

EXAMPLE 3.24 S: [Sat88]

$$\begin{aligned} aba &\to ba \quad \diamond\\ c &\to a \quad \diamond\\ c &\to b \quad \diamond \end{aligned}$$

$$\begin{aligned} c &\to b\\ a &\to b\\ bbb &\to bb \end{aligned}$$

KBOS: $\varphi(a) = \varphi(b) = \varphi(c)$
$c \succ a \succ b$

POL:
$$\begin{aligned}[a](x) &= 3x\\ [b](x) &= 2x\\ [c](x) &= 5x\end{aligned}$$

RPOS: $c \succ a \succ b$ □

EXAMPLE 3.25 S: [Mar86]

$abaab \rightarrow a$

$aaab \rightarrow abaa$

KBOS: $\varphi(a) = \varphi(b)$
$a \succ b$

POL: $[a](x) = 2x$
$[b](x) = x + 1$

RPOS: $a \succ b$ □

EXAMPLE 3.26 S: [Mar86]

$ac \rightarrow cbb$ ◇
$ba \rightarrow acc$ ◇
$cb \rightarrow baa$ ◇
$ua \rightarrow \lambda$ ◇
$vb \rightarrow \lambda$ ◇
$wc \rightarrow \lambda$ ◇

$a \rightarrow ccccbccccc$
$u \rightarrow bc$
$v \rightarrow cccccbccccc$
$w \rightarrow cccccc$
$bb \rightarrow cccbcccccc$
$bcb \rightarrow ccbc$
$bccb \rightarrow ccccccbccccc$
$bcccb \rightarrow cbcccc$
$bccccb \rightarrow ccccbcc$
$bcccccb \rightarrow cc$
$bccccccb \rightarrow cccccbccc$
$ccccccc \rightarrow \lambda$

KBOS: $\varphi(c) = 1, \varphi(b) = 10, \varphi(a) = 20, \varphi(u) = \varphi(v) = \varphi(w) = 21$
$v \succ a \succ c, b \succ c$

POL: $[a](x) = 11x + 11$
$[b](x) = 5x + 5$
$[c](x) = x + 1$
$[u](x) = 13x + 13$
$[v](x) = 13x + 13$
$[w](x) = 13x + 13$

RPOS: $a \succ b \succ c, u \succ a, v \succ a, w \succ a$ □

EXAMPLE 3.27 S: [Mar86][26]

a	$\rightarrow$	$cccbcc$	$\diamond$
bcb	$\rightarrow$	$ccbc$	$\diamond$
$bccb$	$\rightarrow$	$ccccccbccccc$	$\diamond$
$bcccb$	$\rightarrow$	$cbcccc$	$\diamond$
$bbcccb$	$\rightarrow$	$ccccbcc$	$\diamond$
$bccccb$	$\rightarrow$	cc	$\diamond$
$bcccccb$	$\rightarrow$	$cccccbccc$	$\diamond$
$ccccccc$	$\rightarrow$	λ	$\diamond$

$a \rightarrow \lambda$
$b \rightarrow \lambda$
$c \rightarrow \lambda$

KBOS: $\varphi(c) = 1, \varphi(b) = 10, \varphi(a) = 16$
$b \succ c$

POL:
$$[a](x) = 2x + 8$$
$$[b](x) = 2x$$
$$[c](x) = x + 1$$

RPOS: $a \succ b \succ c$ □

EXAMPLE 3.28 S: [Mar86][27]

bu	$\rightarrow$	ab	$\diamond$
au	$\rightarrow$	λ	$\diamond$
aw	$\rightarrow$	ca	
cv	$\rightarrow$	bc	$\diamond$
bv	$\rightarrow$	λ	$\diamond$
cw	$\rightarrow$	λ	
ua	$\rightarrow$	λ	$\diamond$
vb	$\rightarrow$	λ	$\diamond$
wc	$\rightarrow$	λ	

---

u	$\rightarrow$	$wbcab$
v	$\rightarrow$	wbc
aba	$\rightarrow$	b
abb	$\rightarrow$	bba
$abca$	$\rightarrow$	$wwbc$
ac	$\rightarrow$	wa
$bbcab$	$\rightarrow$	$ccabc$
bcb	$\rightarrow$	c
bcc	$\rightarrow$	ccb
bw	$\rightarrow$	$wwbc$

[26] Using the given orderings, the completion produces only 39 rules and 8 critical pairs while the ordering on the polynomial interpretations $[a](x) = x + 1$, $[b](x) = 23x + 23$ and $[c](x) = 11x + 11$ leads to 530 rules and 1006 critical pairs (before generating the canonical system above).

[27] The above canonical system is generated w.r.t. the given POL. Using the RPOS, the rule $bbcab \rightarrow ccabc$ is redundant, and $abc \rightarrow wwbbcab$ is included instead of $abca \rightarrow wwbc$.

POL:

$$\begin{aligned}
[a](x) &= 7x+7\\
[b](x) &= 5x+5\\
[c](x) &= x+1\\
[u](x) &= 176x+224\\
[v](x) &= 5x+13\\
[w](x) &= x+2
\end{aligned}$$

RPOS: $u \succ a \succ v \succ b \succ w \succ c$ □

EXAMPLE 3.29 S:

$$\begin{aligned}
ac &\to ea\\
ad &\to va\\
bc &\to eb\\
bd &\to vb\\
au &\to bu \quad \diamond\\
u &\to cw\\
dw &\to u \quad \diamond
\end{aligned}$$

$$\begin{aligned}
dw &\to cw\\
eaw &\to vbw\\
vaw &\to vbw\\
ebw &\to vbw
\end{aligned}$$

KBOS: $\varphi(b) = \varphi(v) = \varphi(w) = 1, \varphi(a) = 2, \varphi(e) = 3, \varphi(c) = 4, \varphi(d) = 5, \varphi(u) = 7$

POL:

$$\begin{aligned}
[a](x) &= x+2\\
[b](x) &= x\\
[c](x) &= x+3\\
[d](x) &= x+4\\
[e](x) &= x+2\\
[u](x) &= x+5\\
[v](x) &= x+1\\
[w](x) &= x
\end{aligned}$$

RPOS: $u \succ d \succ c \succ e \succ a \succ v \succ b \succ w$ □

EXAMPLE 3.30 O[28]:

$$\begin{aligned}
f(g(x),x) &\to a \quad \diamond\\
f(g(x),y) &\to h(y) \quad \diamond\\
f(g(x),f(y,z)) &\to k(f(g(x),y),f(g(x),z)) \quad \diamond
\end{aligned}$$

$$\begin{aligned}
f(g(x),y) &\to a\\
h(x) &\to a\\
k(a,a) &\to a
\end{aligned}$$

[28] If the third rule of the initial rule system is normalized to $k(a,a) \to a$ the following KBOS can prove its termination: $\varphi(a) = \varphi(f) = \varphi(g) = \varphi(h) = \varphi(k) = 1$.

POL: $$\begin{aligned}[f](x,y) &= x+y\\ [g](x) &= x+1\\ [h](x) &= x+1\\ [k](x,y) &= x+y\\ [a]() &= 1\end{aligned}$$

RPOS: $f \succ h \succ a$ □

EXAMPLE 3.31 O:

$$\begin{aligned}f(x,h(y)) &\to j(x)\\ f(h(x),y) &\to j(h(x))\\ g(f(x,x)) &\to i(x)\end{aligned}$$

$$g(j(h(x))) \to i(h(x))$$

KBOS: $\varphi(f)=\varphi(g)=\varphi(h)=\varphi(i)=\varphi(j)=1$

POL: $$\begin{aligned}[f](x,y) &= x+y\\ [g](x) &= x+1\\ [h](x) &= x\\ [i](x) &= x\\ [j](x) &= x\end{aligned}$$

RPOS: $g \succ i, f \succ j$ □

EXAMPLE 3.32 O:

$$\begin{aligned}f(x,x) &\to x && \diamond\\ f(g(x),y) &\to g(x) && \diamond\\ g(g(x)) &\to x\end{aligned}$$

$$f(x,y) \to x$$

KBOS: $\varphi(f)=\varphi(g)$

POL: $$\begin{aligned}[f](x,y) &= x+y\\ [g](x) &= x+1\end{aligned}$$

RPOS: empty $\succ$ □

EXAMPLE 3.33 O: [Hul80a]

$$\begin{aligned}f(g(x)) &\to g(x) && \diamond\\ g(a) &\to a\\ g(g(x)) &\to x\end{aligned}$$

$$f(x) \to x$$

KBOS: $\varphi(f)=\varphi(g)=\varphi(a)$

POL: $$\begin{aligned}[f](x) &= x+1\\ [g](x) &= x+1\\ [a]() &= 1\end{aligned}$$

RPOS: empty $\succ$ □

4 Open Problems

EXAMPLE 4.1 G: [JK86]

$$\begin{array}{lcl} --x & \rightarrow & x \\ -(x+y) & \rightarrow & (-y)+(-x) \\ (-x)+(x+y) & \rightarrow & y \\ (x+y)+(-y) & \rightarrow & x \end{array}$$

Problem: Divergence[29] □

EXAMPLE 4.2 G: Commutator ([Mar86])

$$\begin{array}{lcl} x*1 & \rightarrow & x \\ 1*y & \rightarrow & y \\ i(x)*x & \rightarrow & 1 \\ x*i(x) & \rightarrow & 1 \\ x*(y*z) & \rightarrow & (x*y)*z \\ i(1) & \rightarrow & 1 \\ (x*y)*i(y) & \rightarrow & x \\ (x*i(y))*y & \rightarrow & x \\ i(i(x)) & \rightarrow & x \\ i(x*y) & \rightarrow & i(y)*i(x) \\ k(x,1) & \rightarrow & 1 \\ k(x,x) & \rightarrow & 1 \\ k(x,y)*k(y,x) & \rightarrow & 1 \\ (i(x)*k(y,z))*x & \rightarrow & k((i(x)*y)*x,(i(x)*z)*x) \\ k(x*i(y),y*i(x)) & \rightarrow & 1 \end{array}$$

Problem: Divergence □

EXAMPLE 4.3 G: [Pau84]

$$\begin{array}{lcl} x+0 & \rightarrow & x \\ (-x)+x & \rightarrow & 0 \\ -0 & \rightarrow & 0 \\ --x & \rightarrow & x \\ -(x+y) & \rightarrow & (-x)+(-y) \\ x*1 & \rightarrow & x \\ x*0 & \rightarrow & 0 \\ x*(y+z) & \rightarrow & (x*y)+(x*z) \\ x*(-y) & \rightarrow & -(x*y) \end{array}$$

Problem: Divergence[30] □

[29] The completion process will converge if + is used as an AC-operator and a new operator is introduced.

[30] Using + as an AC-operator, the completion converges.

EXAMPLE 4.4 G: [Kir87]

$$\begin{array}{lcl} (x*y)+(a*y) & \to & (x+a)*y \\ (x*y)*z & \to & x*(y*z) \end{array}$$

Problem: Divergence □

EXAMPLE 4.5 G: Distributivity ([Der87])

$$\begin{array}{lcl} x*(y+z) & \to & (x*y)+(x*z) \\ (x+y)*z & \to & (x*z)+(y*z) \\ x*1 & \to & x \\ 1*y & \to & y \end{array}$$

Problem: Divergence & Termination[31] □

EXAMPLE 4.6 G: Associativity and Distributivity ([Der87])

$$\begin{array}{lcl} (x*y)*z & \to & x*(y*z) \\ (x+y)*z & \to & (x*z)+(y*z) \\ x*(y+f(z)) & \to & g(x,z)*(y+y) \end{array}$$

Problem: Termination □

EXAMPLE 4.7 G: [Dic86]

$$\begin{array}{lcl} i(x)*x & \to & 1 \\ 1*y & \to & y \\ x*0 & \to & 0 \\ (x*y)*z & \to & x*(y*z) \end{array}$$

Problem: Termination[32] □

EXAMPLE 4.8 G: Idempotent Semigroup ([Hul80b])

$$\begin{array}{lcl} x*(y*z) & \to & (x*y)*z \\ x*x & \to & x \end{array}$$

Problem: Divergence[33] □

EXAMPLE 4.9 G: Ring ([Hul80b])

$$\begin{array}{lcl} x+0 & \to & x \\ x+i(x) & \to & 0 \\ (x+y)+z & \to & x+(y+z) \\ x*(y+z) & \to & (x*y)+(x*z) \\ (x+y)*z & \to & (x*z)+(y*z) \end{array}$$

Problem: Termination & Divergence □

---

[31] The AC-completion (+ is AC) leads to a canonical system.

[32] The pair $x = 1$ will be generated.

[33] AC-completion terminates.

EXAMPLE 4.10 G: A-Modules ([Hul80b])

$$\begin{array}{lcl} x * (y * z) & \rightarrow & (x \otimes y) * z \\ 1 * y & \rightarrow & y \\ (x + y) * z & \rightarrow & (x * z) \oplus (y * z) \\ x * (y \oplus z) & \rightarrow & (x * y) \oplus (x * z) \end{array}$$

Problem: Termination & Divergence □

EXAMPLE 4.11 A: Addition ([JL87])

$$\begin{array}{lcl} x + 0 & \rightarrow & x \\ x + s(y) & \rightarrow & s(x + y) \\ 0 + s(y) & \rightarrow & s(y) \\ s(0 + y) & \rightarrow & s(y) \end{array}$$

Problem: Divergence □

EXAMPLE 4.12 A: Addition

$$\begin{array}{lcl} 0 + y & \rightarrow & y \\ s(x) + 0 & \rightarrow & s(x) \\ s(x) + s(y) & \rightarrow & s(s(x) + (y + 0)) \end{array}$$

Problem: Termination □

EXAMPLE 4.13 A: Difference

$$\begin{array}{lcl} 0 - y & \rightarrow & 0 \\ x - 0 & \rightarrow & x \\ x - s(y) & \rightarrow & if(x > s(y), s(x - p(s(y))), 0) \\ p(0) & \rightarrow & 0 \\ p(s(x)) & \rightarrow & x \end{array}$$

Problem: Termination □

EXAMPLE 4.14 A: [NRS89]

$$\begin{array}{lcl} p(s(x)) & \rightarrow & x \\ s(p(x)) & \rightarrow & x \\ 0 + y & \rightarrow & y \\ s(x) + y & \rightarrow & s(x + y) \\ p(x) + y & \rightarrow & p(x + y) \\ -0 & \rightarrow & 0 \\ -s(x) & \rightarrow & p(-x) \\ -p(x) & \rightarrow & s(-x) \\ 0 * y & \rightarrow & 0 \\ s(x) * y & \rightarrow & (x * y) + y \\ p(x) * y & \rightarrow & (x * y) + (-y) \end{array}$$

Problem: Divergence[34] □

[34] The above system is ground confluent.

EXAMPLE 4.15 A: Addition and Subtraction ([Bac87])

$$\begin{array}{lcl} (x-y)+z & \rightarrow & (x+z)-y \\ (x+y)-y & \rightarrow & x \end{array}$$

Problem: Divergence □

EXAMPLE 4.16 A: [Zeh89]

$$\begin{array}{lcl} f(0) & \rightarrow & s(0) \\ f(s(0)) & \rightarrow & s(s(0)) \\ f(s(0)) & \rightarrow & s(s(0)) * f(0) \\ f(x+s(0)) & \rightarrow & s(s(0)) * f(x) \\ f(x+y) & \rightarrow & f(x) * f(y) \end{array}$$

Problem: Divergence □

EXAMPLE 4.17 A: Factorial Function ([KL80])

$$\begin{array}{lcl} fac(s(x)) & \rightarrow & fac(p(s(x))) * s(x) \\ p(s(0)) & \rightarrow & 0 \\ p(s(s(x))) & \rightarrow & s(p(s(x))) \end{array}$$

Problem: Termination □

EXAMPLE 4.18 A: Greatest Common Divisor ([BM79])

$$\begin{array}{lcl} gcd(x,0) & \rightarrow & x \\ gcd(0,y) & \rightarrow & y \\ gcd(s(x),s(y)) & \rightarrow & if(x < y, gcd(s(x), y-x), gcd(x-y, s(y))) \end{array}$$

Problem: Termination □

EXAMPLE 4.19 A: [Ste88]

$$((-x)-(-x))-((-y)-(-y)) \rightarrow (x-y)-(x-y)$$

Problem: Divergence[35] □

EXAMPLE 4.20 B: Boolean Ring ([Ait85])

$$\begin{array}{lcl} \neg x & \rightarrow & x \oplus true \\ x \vee y & \rightarrow & (x \wedge y) \oplus (x \oplus y) \\ x \supset y & \rightarrow & (x \wedge y) \oplus (x \oplus true) \\ x \wedge true & \rightarrow & x \\ x \wedge false & \rightarrow & false \\ x \wedge x & \rightarrow & x \\ x \oplus false & \rightarrow & x \\ x \oplus x & \rightarrow & false \\ (x \oplus y) \wedge z & \rightarrow & (x \wedge z) \oplus (y \wedge z) \end{array}$$

[35]Note that this rule cannot be oriented with the help of the RPOS. However, it is possible using the IRDS or a polynomial ordering.

Problem: Divergence[36] □

EXAMPLE 4.21 B: Distributive Lattices ([BPW89])

$x \wedge (y \vee z)$	$\rightarrow$	$(x \wedge y) \vee (x \wedge z)$
$x \wedge (y \wedge y)$	$\rightarrow$	$x \wedge y$
$(x \vee y) \vee (y \wedge z)$	$\rightarrow$	$x \vee y$
$x \vee (x \wedge y)$	$\rightarrow$	x
$true \vee y$	$\rightarrow$	$true$
$x \vee false$	$\rightarrow$	x
$x \vee x$	$\rightarrow$	x
$x \vee (y \vee y)$	$\rightarrow$	$x \vee y$
$x \wedge true$	$\rightarrow$	x
$false \wedge y$	$\rightarrow$	$false$
$x \wedge x$	$\rightarrow$	x

Problem: Divergence[37] □

EXAMPLE 4.22 B: Disjunctive Normal Form ([Der83b])

$\neg(x \wedge y)$	$\rightarrow$	$(\neg x) \vee (\neg y)$
$\neg(x \vee y)$	$\rightarrow$	$(\neg x) \wedge (\neg y)$
$x \wedge (y \vee z)$	$\rightarrow$	$(x \wedge y) \vee (x \wedge z)$

Problem: Divergence □

EXAMPLE 4.23 B: If-then-else

$if(true, x, y)$	$\rightarrow$	x
$if(false, x, y)$	$\rightarrow$	y
$if(x, y, y)$	$\rightarrow$	y
$if(if(x, y, z), u, v)$	$\rightarrow$	$if(x, if(y, u, v), if(z, u, v))$
$if(x, if(x, y, z), z)$	$\rightarrow$	$if(x, y, z)$
$if(x, y, if(x, y, z))$	$\rightarrow$	$if(x, y, z)$

Problem: Divergence □

EXAMPLE 4.24 L: Reverse

$rev(nil)$	$\rightarrow$	nil
$rev(x \circ y)$	$\rightarrow$	$rev_1(x, y) \circ rev_2(x, y)$
$rev_1(x, nil)$	$\rightarrow$	x
$rev_1(x, y \circ z)$	$\rightarrow$	$rev_1(y, z)$
$rev_2(x, nil)$	$\rightarrow$	nil
$rev(x, y \circ z)$	$\rightarrow$	$rev(x \circ rev(rev_2(y, z)))$

Problem: Termination □

[36] Using $\wedge$ and $\oplus$ as AC-operators, the completion process terminates.
[37] The AC-completion method will converge, if $\wedge$ and $\vee$ are used as AC-operators.

EXAMPLE 4.25 L: Reverse

$$\begin{array}{lcl} rev(a) & \to & a \\ rev(b) & \to & b \\ rev(x \circ y) & \to & rev(y) \circ rev(x) \\ rev(x \circ x) & \to & rev(x) \end{array}$$

Problem: Divergence □

EXAMPLE 4.26 L: [Par88]

$$\begin{array}{lcl} rev(nil) & \to & nil \\ rev(rev(x)) & \to & x \\ rev(x \circ y) & \to & rev(y) \circ rev(x) \\ nil \circ y & \to & y \\ x \circ nil & \to & x \\ (x.y) \circ z & \to & x.(y \circ z) \\ x \circ (y \circ z) & \to & (x \circ y) \circ z \\ make(x) & \to & x.nil \end{array}$$

Problem: Divergence □

EXAMPLE 4.27 L: List of Integers

$$\begin{array}{lcl} int(0,0) & \to & 0.nil \\ int(0,s(y)) & \to & 0.int(s(0),s(y)) \\ int(s(x),0) & \to & nil \\ int(s(x),s(y)) & \to & int_{list}(int(x,y)) \\ int_{list}(nil) & \to & nil \\ int_{list}(x.y) & \to & s(x).int_{list}(y) \end{array}$$

Problem: Termination □

EXAMPLE 4.28 L: Finite Sequences of Natural Numbers ([KM87])

$$\begin{array}{lcl} f(x,nil) & \to & g(nil,x) \\ f(x,g(y,z)) & \to & g(f(x,y),z) \\ x \circ nil & \to & x \\ x \circ g(y,z) & \to & g(x \circ y,z) \\ null(nil) & \to & true \\ null(g(x,y)) & \to & false \\ mem(nil,y) & \to & false \\ mem(g(x,y),z) & \to & (y \equiv z) \vee mem(x,z) \\ mem(x,max(x)) & \to & \neg null(x) \\ max(g(g(nil,x),y)) & \to & max'(x,y) \\ max(g(g(g(x,y),z),u)) & \to & max'(max(g(g(x,y),z)),u) \end{array}$$

Problem: Divergence □

EXAMPLE 4.29 L: Merging ([Bou83])

$$
\begin{array}{lcl}
merge(x, nil) & \rightarrow & x \\
merge(nil, y) & \rightarrow & y \\
merge(x \circ y, u \circ v) & \rightarrow & x \circ merge(y, u \circ v) \\
merge(x \circ y, u \circ v) & \rightarrow & u \circ merge(x \circ y, v)
\end{array}
$$

Problem: Divergence □

EXAMPLE 4.30 L: Binary Trees

$$
\begin{array}{lcl}
f(nil) & \rightarrow & nil \\
f(nil.y) & \rightarrow & nil.f(y) \\
f((x.y).z) & \rightarrow & f(x.(y.z)) \\
g(nil) & \rightarrow & nil \\
g(x.nil) & \rightarrow & g(x).nil \\
g(x.(y.z)) & \rightarrow & g((x.y).z)
\end{array}
$$

Problem: Termination □

EXAMPLE 4.31 L: Purging ([Wal88])

$$
\begin{array}{lcl}
purge(nil) & \rightarrow & nil \\
purge(x.y) & \rightarrow & x.purge(remove(x, y)) \\
remove(x, nil) & \rightarrow & nil \\
remove(x, y.z) & \rightarrow & if(x \equiv y, remove(x, z), y.remove(x, z))
\end{array}
$$

Problem: Termination □

EXAMPLE 4.32 S: [Mar86]

$$
\begin{array}{lcl}
ab & \rightarrow & baa \\
bc & \rightarrow & cbb \\
ca & \rightarrow & acc \\
ua & \rightarrow & \lambda \\
vb & \rightarrow & \lambda \\
wc & \rightarrow & \lambda \\
au & \rightarrow & \lambda \\
bv & \rightarrow & \lambda \\
cw & \rightarrow & \lambda
\end{array}
$$

Problem: Divergence □

EXAMPLE 4.33 S: [KN85]

$$aba \rightarrow bab$$

Problem: Divergence & Termination □

EXAMPLE 4.34 S: [Der87]

$$ab \rightarrow bbaa$$

Problem: Termination □

EXAMPLE 4.35 S: [BL88]

$$ab \rightarrow acb$$

Problem: Termination □

EXAMPLE 4.36 S: [BL88]

$$aa \rightarrow aba$$

Problem: Termination □

EXAMPLE 4.37 S: [HP86]

$$aba \rightarrow ba$$

Problem: Divergence □

EXAMPLE 4.38 O: Associative Ring ([Hul80b])

$$\begin{array}{lcl} g(f(x,y),z) & \rightarrow & f(x,g(y,z)) \\ g(h(x,y),z) & \rightarrow & g(x,f(y,z)) \\ g(x,h(y,z)) & \rightarrow & h(g(x,y),z) \end{array}$$

Problem: Divergence □

EXAMPLE 4.39 O: [Ste88]

$$x * ((-y) * y) \rightarrow (-(y * y)) * x$$

Problem: Divergence □

EXAMPLE 4.40 O: Combinatory Logic ([Klo87])

$$\begin{array}{lcl} f(f(f(a,x),y),z) & \rightarrow & f(f(x,z),f(y,z)) \\ f(f(b,x),y) & \rightarrow & x \\ f(c,y) & \rightarrow & y \end{array}$$

Problem: Termination □

EXAMPLE 4.41 O: Forward Chains ([Der83a])

$$\begin{array}{lcl} f(a,y) & \rightarrow & f(y,g(y)) \\ g(a) & \rightarrow & b \\ g(b) & \rightarrow & b \end{array}$$

Problem: Termination □

EXAMPLE 4.42 O:

$$\begin{array}{lcl} f(a,g(y)) & \rightarrow & g(g(y)) \\ f(g(x),a) & \rightarrow & f(x,g(a)) \\ f(g(x),g(y)) & \rightarrow & h(g(y),x,g(y)) \\ h(g(x),y,z) & \rightarrow & f(y,h(x,y,z)) \\ h(a,y,z) & \rightarrow & z \end{array}$$

Problem: Termination □

EXAMPLE 4.43 O: [TJ89]

$$\begin{array}{lcl} x+0 & \to & x \\ x+s(y) & \to & s(x+y) \\ 0+y & \to & y \\ s(x)+y & \to & s(x+y) \\ x+(y+z) & \to & (x+y)+z \\ f(g(f(x))) & \to & f(h(s(0),x)) \\ f(g(h(x,y))) & \to & f(h(s(x),y)) \\ f(h(x,h(y,z))) & \to & f(h(x+y,z)) \end{array}$$

Problem: Divergence □

EXAMPLE 4.44 O: [Bac87]

$$\begin{array}{lcl} f(h(x)) & \to & f(i(x)) \\ g(i(x)) & \to & g(h(x)) \\ h(a) & \to & b \\ i(a) & \to & b \end{array}$$

Problem: Termination □

EXAMPLE 4.45 O: [Bau88]

$$\begin{array}{lcl} f(x,x) & \to & a \\ f(g(x),y) & \to & f(x,y) \end{array}$$

Problem: Divergence □

EXAMPLE 4.46 O: [BL88]

$$\begin{array}{lcl} f(a) & \to & f(b) \\ g(b) & \to & g(a) \end{array}$$

Problem: Termination □

EXAMPLE 4.47 O: [Die89]

$$\begin{array}{lcl} f(g(i(a,b,b'),c),d) & \to & if(e,f(b.c,d'),f(b'.c,d')) \\ f(g(h(a,b),c),d) & \to & if(e,f(b.g(h(a,b),c),d),f(c,d')) \end{array}$$

Problem: Termination □

EXAMPLE 4.48 O: Ternary Boolean Algebra ([Wos89])

$$\begin{array}{lcl} f(f(x,y,z),u,f(x,y,v)) & \to & f(x,y,f(z,u,v)) \\ f(x,y,y) & \to & y \\ f(x,y,g(y)) & \to & x \\ f(x,x,y) & \to & x \\ f(g(x),x,y) & \to & y \end{array}$$

Problem: Divergence □

EXAMPLE 4.49 O:

$$f(x,y,f(z,u,v)) \rightarrow f(f(x,y,z),u,f(x,y,v))$$

Problem: Termination □

EXAMPLE 4.50 O: [Toy87]

$$f(0,1,x) \rightarrow f(x,x,x)$$

Problem: Termination □

EXAMPLE 4.51 O: [Pla86]

$$\begin{array}{lcl} f(a) & \rightarrow & g(h(a)) \\ h(g(x)) & \rightarrow & g(h(f(x))) \\ k(x,h(x),a) & \rightarrow & h(x) \\ k(f(x),y,x) & \rightarrow & f(x) \end{array}$$

Problem: Termination □

EXAMPLE 4.52 O: [Nip89]

$$\begin{array}{lcl} s(a) & \rightarrow & a \\ s(s(x)) & \rightarrow & x \\ s(f(x,y)) & \rightarrow & f(s(y),s(x)) \\ s(g(x,y)) & \rightarrow & g(s(x),s(y)) \\ f(x,a) & \rightarrow & x \\ f(a,y) & \rightarrow & y \\ f(g(x,y),g(u,v)) & \rightarrow & g(f(x,u),f(y,v)) \\ g(a,a) & \rightarrow & a \end{array}$$

Problem: Divergence □

EXAMPLE 4.53 O:

$$\begin{array}{lcl} f(a) & \rightarrow & b \\ f(c) & \rightarrow & d \\ f(g(x,y)) & \rightarrow & g(f(x),f(y)) \\ f(h(x,y)) & \rightarrow & g(h(y,f(x)),h(x,f(y))) \\ g(x,x) & \rightarrow & h(e,x) \end{array}$$

Problem: Divergence □

EXAMPLE 4.54 O:

$$g(f(x,y)) \rightarrow f(f(g(g(x)),g(g(y))),f(g(g(x)),g(g(y))))$$

Problem: Termination □

EXAMPLE 4.55 O:

$$\begin{array}{lcl} f(x,x) & \rightarrow & f(a,b) \\ b & \rightarrow & c \end{array}$$

Problem: Termination □

EXAMPLE 4.56 O:

$$\begin{array}{lcl} f(a,b) & \to & f(a,c) \\ f(c,d) & \to & f(b,d) \end{array}$$

Problem: Termination □

EXAMPLE 4.57 O:

$$\begin{array}{lcl} f(x,y,z) & \to & g(x \leq y, x, y, z) \\ g(true,x,y,z) & \to & z \\ g(false,x,y,z) & \to & f(f(p(x),y,z), f(p(y),z,x), f(p(z),x,y)) \\ p(0) & \to & 0 \\ p(s(x)) & \to & x \end{array}$$

Problem: Termination □

EXAMPLE 4.58 O: Battle of Hydra and Hercules ([DJ89])

$$\begin{array}{lcl} f(x,g(y)) & \to & f(h(x), i(x,y)) \\ i(x,j(0,0)) & \to & g(0) \\ i(x,j(y,z)) & \to & j(g(y), i(x,z)) \\ i(h(x), j(j(y,z),0)) & \to & j(i(h(x), j(y,z)), i(x, j(y,z))) \\ j(g(x), g(y)) & \to & g(j(x,y)) \end{array}$$

Problem: Termination □

EXAMPLE 4.59 O: Quick-Sort ([Gra90])

$$\begin{array}{lcl} qsort(nil) & \to & nil \\ qsort(x.y) & \to & qsort(lowers(x,y)) \circ (x.qsort(greaters(x,y))) \\ lowers(x,nil) & \to & nil \\ lowers(x,y.z) & \to & if(y \leq x, y.lowers(x,z), lowers(x,z)) \\ greaters(x,nil) & \to & nil \\ greaters(x,y.z) & \to & if(y \leq x, greaters(x,z), y.greaters(x,z)) \end{array}$$

Problem: Termination □

EXAMPLE 4.60 O: Minimum-Sort ([Gra90])

$$\begin{array}{lcl} msort(nil) & \to & nil \\ msort(x.y) & \to & min(x,y).msort(del(min(x,y), x.y)) \\ min(x,nil) & \to & x \\ min(x,y.z) & \to & if(x \leq y, min(x,z), min(y,z)) \\ del(x,nil) & \to & nil \\ del(x,y.z) & \to & if(x \equiv y, z, y.del(x,z)) \end{array}$$

Problem: Termination □

EXAMPLE 4.61 O: Bubble-Sort ([Gra90])

$$
\begin{array}{lcl}
bsort(nil) & \rightarrow & nil \\
bsort(x.y) & \rightarrow & last(bubble(x.y)).bsort(butlast(bubble(x.y))) \\
bubble(nil) & \rightarrow & nil \\
bubble(x.nil) & \rightarrow & x.nil \\
bubble(x.(y.z)) & \rightarrow & if(x \leq y, y.bubble(x.z), x.bubble(y.z)) \\
last(nil) & \rightarrow & 0 \\
last(x.nil) & \rightarrow & x \\
last(x.(y.z)) & \rightarrow & last(y.z) \\
butlast(nil) & \rightarrow & nil \\
butlast(x.nil) & \rightarrow & nil \\
butlast(x.(y.z)) & \rightarrow & x.butlast(y.z)
\end{array}
$$

Problem: Termination □

References

[Ait85] Hassan Ait-Kaci. An algorithm for finding a minimal recursive path ordering. *R.A.I.R.O. Informatique Theorique (Theoretical Informatics)*, 19(4):359–382, 1985.

[AM90] Jürgen Avenhaus and Klaus E. Madlener. Term rewriting and equational reasoning. In R.B. Banerji, editor, *Formal Techniques in Artificial Intelligence – A Sourcebook*, pages 1–43. Elsevier Science Publishers B.V. (North-Holland), 1990.

[AMS89] Jürgen Avenhaus, Klaus E. Madlener, and Joachim Steinbach. COMTES – An experimental environment for the completion of term rewriting systems. In Nachum Dershowitz, editor, *Proc. 3rd RTA (LNCS 355)*, pages 542–546, Chapel Hill (North Carolina), April 1989.

[Bac87] Leo Bachmair. *Proof methods for equational theories*. PhD thesis, University of Illinois, Urbana-Champaign (Illinois), 1987.

[Bau88] Marianne Baudinet. Proving termination properties of PROLOG programs – A semantic approach. In *Proc. 3rd LICS*, pages 336–347, Edinburgh (Scotland), 1988.

[BBK87] Jos C.M. Baeten, Jan A. Bergstra, and Jan Willem Klop. Term rewriting systems with priorities. In *Proc. 2nd RTA*, pages 83–94, Bordeaux (France), May 1987.

[BD86] Leo Bachmair and Nachum Dershowitz. Commutation, transformation and termination. In *Proc. 8th CADE (LNCS 230)*, pages 52–60, Oxford (England), 1986.

[Bel86] Françoise Bellegarde. Rewriting systems on FP expressions to reduce the number of sequences yielded. *SCP*, 6:11–34, 1986.

[BL87] Ahlem BenCherifa and Pierre Lescanne. Termination of rewriting systems by polynomial interpretations and its implementation. *SCP*, 9(2):137–160, October 1987.

[BL88] Françoise Bellegarde and Pierre Lescanne. Termination proofs based on transformation techniques. Technical report, Centre de Recherche en Informatique de Nancy, Nancy (France), March 1988.

[BM79] Robert S. Boyer and J. Strother Moore. *A computational logic*. Academic Press, 1979.

[Bou83] G. Boudol. Computational semantics of term rewriting systems. Technical report, INRIA, France, July 1983.

[BPW89] Timothy B. Baird, Gerald E. Peterson, and Ralph W. Wilkerson. Complete sets of reductions modulo associativity, commutativity and identity. In Nachum Dershowitz, editor, *Proc. 3rd RTA (LNCS 355)*, pages 29–44, Chapel Hill (North Carolina), April 1989.

[Bur68] R.M. Burstall. Proving properties of programs by structural induction. *Computer Journal*, 12:41–48, 1968.

[Chr89] Jim Christian. Fast Knuth-Bendix completion: Summary. In Nachum Dershowitz, editor, *Proc. 3rd RTA (LNCS 355)*, pages 551–555, Chapel Hill (North Carolina), April 1989.

[Com86] Hubert Comon. Sufficient completeness, term rewriting systems and 'anti-unification'. In *Proc. 8th CADE (LNCS 230)*, pages 128–140, Oxford (England), 1986.

[Der82] Nachum Dershowitz. Orderings for term rewriting systems. *J. TCS*, 17(3):279–301, March 1982.

[Der83a] Nachum Dershowitz. Applications of the Knuth-Bendix completion procedure. Technical Report ATR-83(8478)-2, The Aerospace Corporation, El Segundo (California), May 1983.

[Der83b] Nachum Dershowitz. Well-founded orderings. Technical Report ATR-83(8478)-3, The Aerospace Corporation, El Segundo (California), May 1983.

[Der85] Nachum Dershowitz. Computing with rewrite systems. *Information and Control*, 65:122–157, 1985.

[Der87] Nachum Dershowitz. Termination of rewriting. *JSC*, 3:69–116, 1987.

[Dic86] A.J.J. Dick. Introduction to 'ERIL' (Equational reasoning – An interactive laboratory). Technical report, Rutherford Appleton Laboratory, Chilton, Didcot (England), 1986.

[Dic87] A.J.J. Dick. ERIL – Equational reasoning: An interactive laboratory. *EUROCAL*, March 1987.

[Die89] N.W.P. van Diepen. SMALL – Dynamic semantics of a language with GOTOs. In J.A. Bergstra, J. Heering, and P. Klint, editors, *Algebraic Specification*, chapter 3, pages 133–161. Addison-Wesley Publishing Company, ACM Press, 1989.

[DJ89] Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. Technical Report 478, Laboratoire de Recherche en Informatique, Université de Paris-Sud, Centre d'Orsay (France), April 1989.

[DP88] Nachum Dershowitz and David A. Plaisted. Equational programming. *Machine Intelligence*, 11:21–56, 1988.

[Gra88] Bernhard Gramlich. Unification of term schemes – Theory and applications. Technical Report SR-88-18, University of Kaiserslautern, Kaiserslautern (Germany), 1988.

[Gra90] Bernhard Gramlich. Completion based inductive theorem proving – A case study in verifying sorting algorithms. Technical Report SR-90-04, University of Kaiserslautern, Kaiserslautern (Germany), 1990.

[HL89] Dieter Hofbauer and Clemens Lautemann. Termination proofs and the length of derivations. In Nachum Dershowitz, editor, *Proc. 3rd RTA (LNCS 355)*, pages 167–177, Chapel Hill (North Carolina), April 1989.

[HO80] Gérard Huet and Derek C. Oppen. Equations and rewrite rules: A survey. In Ronald Book, editor, *Formal Languages – Perspectives and Open Problems*, pages 349–405. Academic Press, 1980.

[HO82] Christoph M. Hoffmann and Michael J. O'Donnell. Programming with equations. *ACM Transactions on Programming Languages and Systems*, 4(1), January 1982.

[HP86] Mikulas Hermann and Igor Privara. On nontermination of Knuth-Bendix algorithm. In *Proc. 13th ICALP*, pages 146–156, Rennes (France), July 1986.

[Hsi85] Jieh Hsiang. Two results in term rewriting theorem proving. In *Proc. 1st RTA*, pages 301–324, Dijon (France), May 1985.

[Hul80a] Jean-Marie Hullot. Canonical forms and unification. In *Proc. 5th CADE (LNCS 87)*, pages 318–334, Les Arc (France), 1980.

[Hul80b] Jean-Marie Hullot. A catalogue of canonical term rewriting systems. Technical Report CSL-113, SRI International, Menlo Park (California), April 1980.

[JK86] Jean-Pierre Jouannaud and Hélène Kirchner. Completion of a set of rules modulo a set of equations. *SIAM Journal on Computing*, 15(4):1155–1194, November 1986.

[JL87] Jean-Pierre Jouannaud and Pierre Lescanne. Rewriting systems. *Technology and Science of Informatics*, 6(3):181–199, 1987.

[JSS85] Neil D. Jones, Peter Sestoft, and Harald Søndergaard. An experiment in partial evaluation: The generation of a compiler generator. In *Proc. 1st RTA*, pages 124–140, Dijon (France), May 1985.

[KB70] Donald E. Knuth and Peter B. Bendix. Simple word problems in universal algebras. In John Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970.

[Kir87] Hélène Kirchner. Schematization of infinite sets of rewrite rules – Application to the divergence of completion processes. In *Proc. 2nd RTA*, pages 180–191, Bordeaux (France), May 1987.

[KL80] Sam Kamin and Jean-Jacques Lévy. Attempts for generalizing the recursive path orderings. Urbana (Illinois), February 1980.

[Klo87] Jan Willem Klop. Term rewriting systems – A tutorial. *Bulletin EATCS*, 32:143–182, 1987.

[KM87] Deepak Kapur and David R. Musser. Proof by consistency. *Artificial Intelligence Journal*, 31:125–157, 1987.

[KN85] Deepak Kapur and Paliath Narendran. A finite Thue system with decidable word problem and without equivalent finite canonical system. *TCS*, 35:337–344, 1985.

[KNS85] Deepak Kapur, Paliath Narendran, and G. Sivakumar. A path ordering for proving termination of term rewriting systems. In *Proc. 10th CAAP (LNCS 185)*, pages 173–187, Berlin (Germany), March 1985.

[KNZ86] Deepak Kapur, Paliath Narendran, and Hantao Zhang. Proof by induction using test sets. In *Proc. 8th CADE (LNCS 230)*, pages 99–117, Oxford (England), 1986.

[KS83] Deepak Kapur and G. Sivakumar. Experiments with and architecture of RRL, a rewrite rule laboratory. In *Proc. NSF Workshop on the Rewrite Rule Laboratory*, pages 33–56, General Electric Research and Development Center, Schenectady (New York), September 1983. appeared as Report 84GEN008.

[Küc87] Wolfgang Küchlin. Inductive completion by ground proof transformation. In *Proc. CREAS 5*, Lakeway (Texas), 1987.

[Lan79] Dallas S. Lankford. On proving term rewriting systems are noetherian. Memo MTP-3, Louisiana Technical University, Department of Mathematics, Ruston (Louisiana), May 1979.

[Lan89] Steffen Lange. Towards a set of inference rules for solving divergence in Knuth-Bendix completion. In *Proc. 1st Conference on Analogical and Inductive Inference (LNCS 397)*, pages 304–316, Reinhardsbrunn (Germany), October 1989.

[Les83] Pierre Lescanne. How to prove termination? An approach to the implementation of a new recursive decomposition ordering. In *Proc. NSF Workshop on the Rewrite Rule Laboratory*, pages 109–121, General Electric Research and Development Center, Schenectady (New York), September 1983. appeared as Report 84GEN008.

[Les84] Pierre Lescanne. Term rewriting systems and algebra. Technical report, CRIN, Nancy (France), 1984.

[Lév86] Jean-Jacques Lévy. Computation and deduction – Course notes 7. Lecture Notes given at Pittsburgh, 1986.

[Mar86] Ursula Martin. Doing algebra with REVE. Technical Report UMCS-86-10-4, Department of Computer Science, University of Manchester, Manchester (England), June 1986.

[Mar87] Ursula Martin. How to choose the weights in the Knuth-Bendix ordering. In Pierre Lescanne, editor, *Proc. 2nd RTA*, pages 42–52, Bordeaux (France), 1987.

[Mid88] Aart Middeldorp. A sufficient condition for the termination of the direct sum of term rewriting systems. Technical report, Department of Mathematics and Computer Science, Vrije University, Vrije (The Netherlands), September 1988.

[MS86] Chilukuri K. Mohan and Mandayam K. Srivas. Function definitions in term rewriting and applicative programming. *Information and Control*, 71:186–217, 1986.

[Mus80] David R. Musser. On proving inductive properties of abstract data types. *Journal of the ACM*, pages 154–162, 1980.

[Nar89] Sanjai Narain. Optimization by non-deterministic, lazy rewriting. In Nachum Dershowitz, editor, *Proc. 3rd RTA (LNCS 355)*, pages 326–342, Chapel Hill (North Carolina), April 1989.

[Nip89] Tobias Nipkow. Proof transformations for some simple equational theories. In *Proc. 3rd International Workshop on Unification*, pages 10–21, Lambrecht (Germany), 1989. appeared as SEKI-Report SR-89-17.

[NRS89] Werner Nutt, Pierre Rety, and Gert Smolka. Basic narrowing revisited. *JSC*, 7:295–317, 1989.

[Pad89] Peter Padawitz. Inductive proofs by resolution and paramodulation. In *Proc. 14th CAAP*, Barcelona (Spain), 1989.

[Par88] Francesco Parisi-Presicce. Product and iteration of module specifications. In *Proc. 13th CAAP*, pages 149–164, Nancy (France), March 1988.

[Pau84] Etienne Paul. Proof by induction in equational theories with relations between constructors. In *Proc. 9th CAAP*, pages 211–225, Bordeaux (France), March 1984.

[PF86] Sara Porat and Nissim Francez. Full-commutation and fair-termination in equational (and combined) term rewriting systems. In *Proc. 8th CADE (LNCS 230)*, pages 21–41, Oxford (England), July 1986.

[Pla86] David A. Plaisted. A simple non-termination test for the Knuth-Bendix method. In *Proc. 8th CADE (LNCS 230)*, pages 79–88, Oxford (England), 1986.

[Pue81] Laurence Puel. Proofs in the final algebra. Technical report, University of Paris and LITP, Paris (France), 1981.

[Red89] Uday S. Reddy. Rewriting techniques for program synthesis. In Nachum Dershowitz, editor, *Proc. 3rd RTA (LNCS 355)*, pages 388–403, Chapel Hill (North Carolina), April 1989.

[RJ81] Fernand Reinig and Jean-Pierre Jouannaud. Decomposition orderings: A new family of recursive simplification orderings. Technical Report 81-R-040, Centre de Recherche en Informatique de Nancy, Nancy (France), 1981.

[Sak84] Ko Sakai. An ordering method for term rewriting systems. Technical Report TR-062, ICOT Research Center, Tokyo (Japan), April 1984.

[Sat88] Andrea Sattler-Klein. The SMOV-strategy. Technical Report 185/88, Department of Computer Science, University of Kaiserslautern, Kaiserslautern (Germany), 1988.

[Ste88] Joachim Steinbach. Term orderings with status. Technical Report SR-88-12, University of Kaiserslautern, Kaiserslautern (Germany), 1988.

[Ste89a] Joachim Steinbach. COMTES - Vervollständigung von Termersetzungssystemen. Technical Report SWP-89-07, University of Kaiserslautern, Kaiserslautern (Germany), 1989.

[Ste89b] Joachim Steinbach. Extensions and comparison of simplification orderings. In Nachum Dershowitz, editor, *Proc. 3rd RTA (LNCS 355)*, pages 434–448, Chapel Hill (North Carolina), April 1989.

[Ste89c] Joachim Steinbach. Path and decomposition orderings for proving AC-termination. Technical Report SR-89-18, University of Kaiserslautern, Kaiserslautern (Germany), 1989.

[SZ90] Joachim Steinbach and Michael Zehnter. Vade-mecum of polynomial orderings. Technical Report SR-90-03, University of Kaiserslautern, Kaiserslautern (Germany), 1990.

[Thi84] Jean Jacques Thiel. Stop losing sleep over incomplete data type specifications. In *ACM-Symposium on Principles of Programming Languages*, 1984.

[TJ89] Muffy Thomas and Klaus P. Jantke. Inductive inference for solving divergence in Knuth-Bendix completion. In *Proc., 1st Conference on Analogical and Inductive Inference*, pages 288–303, Reinhardsbrunn (Germany), October 1989.

[Toy86] Yoshihito Toyama. How to prove equivalence of term rewriting systems without induction. In *Proc. 8th CADE (LNCS 230)*, pages 118–127, Oxford (England), 1986.

[Toy87] Yoshihito Toyama. Counterexamples to termination for the direct sum of term rewriting systems. *IPL*, 25(3):141–143, May 1987.

[Wal88] Christoph Walther. A digest of argument-bounded algorithms. Technical Report 18/88, University of Karlsruhe, Karlsruhe (Germany), 1988.

[Wos89] Larry Wos. The problem of determining the size of a complete set of reductions. *JAR*, 5:235–237, 1989.

[You88] Jia-Huai You. Solving equations in an equational language. In *Proc. 1st International Conference on Algebraic and Logic Programming*, pages 245–254, Gaussig (Germany), November 1988.

[Zeh89] Michael Zehnter. Theorievollständige Ordnungen – Ein überblick. Projektarbeit, Universität Kaiserslautern, November 1989.