Priority rewrite systems

Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

J.C.M. Baeten, J.A. Bergstra, J.W. Klop

Priority rewrite systems

Department of Computer Science Report CS-R8407 May

PRIORITY REWRITE SYSTEMS

J.C.M. BAETEN
Technische Hogeschool Delft

J.A. BERGSTRA, J.W. KLOP
Centre for Mathematics and Computer Science, Amsterdam

Term rewrite systems with rules of different priority are introduced. The semantics are explained in detail and several examples are discussed, including a rewrite rule interpretation of Backus functional programming.

1980 MATHEMATICS SUBJECT CLASSIFICATION: 03F65, 68C01, 68C99.
1982 CR. CATEGORIES: D.3.3, F.3.2, F.4.2.
KEY WORDS & PHRASES: term rewrite systems, priority rewrite set, signature, modularity, specification.
NOTE: This report will be submitted for publication elsewhere.

Report CS-R8407
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Introduction

The aim of this paper is twofold. It serves as an introduction to the first principles of term rewrite systems, (TRS), focusing on signatures, term algebras, congruences and rewrite sets.

Moreover, the conventional notion of a TRS is extended to that of a PRS (priority rewrite system). In a PRS differ ent rules may have relative priorities. Much attention is paid to precise semantics of the PRS mechanism.

In several examples, stacks, queues and in particular Backus' FP (functional programming) the use of the PRS mechanism is shown.

Motivation for the topic

In order to place the paper in an appropriate context we consider the issue of equational (algebraic) abstract datatype specifications. The simplest kind of specification is a pair (Σ,E) with Σ a signature and E a finite set of equations over this signature.

This type of specification and its initial algebra semantics has been thoroughly investigated in GTW[78],GTWr[75], see also Kl[83], and for a survey of recent literature on datatypes: KL[83].

In the subsequent development of datatype specification theory three sub-issues play a prominent role:

(i) executable specifications.

(ii) modular structure of specifications.

(iii) expressive power of specifications.

The most promising way to turn an equational specification into an executable module is to connect with it a term rewrite system which is terminating and confluent and which describes the same congruence on closed terms. For a survey on these matters see HO[80].

In BT[80] it was shown that every computable data type can be specified equationally (through using auxiliary functions) in such a way that one may provide each equation with a direction and then obtain a

confluent and terminating rewrite system. In this sense equational specifications with initial algebra semantics are adequate. (Leaving out the requirements on the rewrite system one may obtain much more concise specifications however BT[83].)

In connection with modularity the research has focused on how a specification S may be composed from smaller specifications $S_1,\ldots,S_n$. In particular much attention has been paid to modularisation via parametrisation, in which case S is made from S_1 and S_2, where S_1 serves as a parameter (specification) and S_2 specifies some functorial type. (See E[79], EKTWW[79], G[83]).

A need for specification formalisms of more expressive power arises from the fact that various mechanisms can hardly be modeled by means of algebraic equations only. This leads us to more and more complex specification languages with less and less algorithmic contents (e.g. B&Te[83]).

In principle the expressive power of algebraic specification can be increased (or at least modified) by changing to a more sophisticated semantics than the usual initial algebra semantics. For instance final algebra semantics (see K[80], W[79]) allows the specification of algebras that cannot be specified initially, see B&Tu[83]. However if we adhere to rewrite rules as an implementation mechanism the initial algebra semantics stands out as most plausible. Within these constraints two extensions of a purely equational formalism come to mind:

(i) conditional equations.

(ii) rewrite rules with priorities.

Rewrite rules based on conditional equations have been introduced and studied in PEE[81].

The present paper introduces a mechanism with priorities. It is shown that this mechanism allows certain modularisations that are not expressible by means of equational specifications. Moreover we feel that the priority mechanism is rather appealing from an intuitive point of view (indeed it has been used by many authors, but in a rather implicit way). Experience in practice will have to determine the practical value of the priority rewrite systems.

The contents of the paper are as follows:

(i) signatures

(ii) term algebras

(iii) equations and equational specifications

(iv) term rewrite systems

(v) priority rewrite systems

(vi) examples

CHAPTER I. SIGNATURES

§1. Definition and examples

A signature contains information about names, especially of sorts, functions and constants. Sorts are also types; functions have an arity $f: S_1 x \dots x S_n \to S$; constants have a sort (type).

We denote signatures with Σ, Γ, Δ, Θ,...

Each signature description has three sections:

S: sorts,
F: functions,
C: constants.

Examples:

i. Σ_B

S: B
F: V: B x B → B
¬: B → B
C: T ∈ B
F ∈ B.

ii. $\Sigma_N^{S,P,+}$

S: N
F: S: N → N
P: N → N
+: N x N → N
C: 0 ∈ N.

iii. $\Sigma_N^{S,P,+,\cdot}$

S: N
F: S: N → N
P: N → N
+: N x N → N
·: N x N → N
C: 0 ∈ N.

iv. $\Sigma_N^{S,P,ADD}$

S: N

F: S: N → N

P: N → N

ADD: N x N → N

C: 0 ∈ N.

v. $\Sigma_N^{P,S,+}$

S: N

F: P: N → N

S: N → N

+: N x N → N

C: 0 ∈ N.

vi. $\Sigma_{N,B}$

S: N, B

F: V: B x B → B

¬: B → B

S: N → N

P: N → N

eq: N x N → B

C: 0 ∈ N

T ∈ B

F ∈ B.

vii. $\Sigma_N^{\perp}$

S: N

F: S: N → N

P: N → N

C: $\perp$ ∈ N

0 ∈ N

(Purpose: to have the option P(0) = $\perp$ instead of P(0) = 0).

viii. $\Sigma_{D,SETS}$

S: D, B, SETS

F: V: B x B → B

¬: B → B

ins: D x SETS → SETS

elt: D x SETS → B

$$\left|\begin{array}{l} C\text{: } \emptyset \in \text{SETS} \\ \quad T \in B \\ \quad F \in B. \end{array}\right.$$

ix. $\Sigma^{\text{STACK}}_{D,\perp}$
$$\left|\begin{array}{l} S\text{: } D,\ ST \\ F\text{: } \text{push: } D \times ST \to ST \\ \quad \text{pop: } ST \to ST \\ \quad \text{top: } ST \to D \\ C\text{: } \emptyset \in ST \\ \quad \perp_D \in D \\ \quad \perp_S \in ST. \end{array}\right.$$

§2. Unordered signatures

All the above signature descriptions describe ordered signatures. This means that sorts, functions and constants constitute an ordered list.

Unordered signatures have sets of sorts, functions and constants:

Examples:

i. $\overline{\Gamma}_1$
$$\left|\begin{array}{l} S\text{: } \{N\} \\ F\text{: } \{P\text{: } N \to N, \\ \quad S\text{: } N \to N, \\ \quad +\text{: } N \times N \to N\} \\ C\text{: } \{0\}. \end{array}\right.$$

ii. $\overline{\Gamma}_2$
$$\left|\begin{array}{l} S\text{: } \{B\} \\ F\text{: } \{\neg\text{: } B \to B, \\ \quad V\text{: } B \times B \to B\} \\ C\text{: } \{F, \\ \quad T\}. \end{array}\right.$$

Notation: We use $\overline{\Sigma}$, $\overline{\Gamma}$, $\overline{\Theta}$ etc. for unordered signatures.

Fact: There is a canonical mapping:

$\overline{\cdot}\text{: } \Sigma \to \overline{\Sigma}$

mapping an ordered signature to an unordered one.

Examples: i. $\overline{\Sigma_B} = \overline{\Gamma_2}$,

ii. $\overline{\Sigma_N^{S,P,+}} = \overline{\Gamma_1}$.

§3. Inclusion of signatures

$\Sigma \subseteq \Gamma$ and $\overline{\Sigma} \subseteq \overline{\Gamma}$ are both defined in the natural way.

Examples: i. $\Sigma_N^{S,P,+} \subseteq \Sigma_N^{S,P,+,\cdot}$

ii. $\overline{\Sigma_N^{S,P,+}} \subseteq \overline{\Sigma_N^{S,P,+,\cdot}}$

iii. $\Sigma_N^{S,P,+} \not\subseteq \Sigma_N^{P,S,+}$

iv. $\overline{\Sigma_N^{S,P,+}} \subseteq \overline{\Sigma_N^{P,S,+}}$

v. $\Sigma_B \subseteq \Sigma_{N,B}$

vi. $\Sigma_B \subseteq \Sigma_{D,SETS}$

4. Union of unordered signatures

For unordered signatures $\overline{\Sigma}$, $\overline{\Gamma}$

$$\overline{\Sigma} + \overline{\Gamma}$$

is determined by pairwise taking unions:

$\overline{\Sigma} + \overline{\Gamma}$ | S: $S(\overline{\Sigma}) \cup S(\overline{\Gamma})$
| F: $F(\overline{\Sigma}) \cup F(\overline{\Gamma})$
| C: $C(\overline{\Sigma}) \cup C(\overline{\Gamma})$.

Taking unions of ordered signatures is problematic. We "do" this as follows: || Given Σ and Γ choose Δ with:
$\overline{\Delta} = \overline{\Sigma} + \overline{\Gamma}$.

Of course, this Δ is not uniquely determined.

§5. Restriction of signatures

Given $\overline{\Sigma}$ and Γ one obtains

$$\rho_{\overline{\Sigma}}(\Gamma)$$

by deleting all sorts, functions and constants of Γ that don't occur in $\overline{\Sigma}$, but preserving the order in what remains of Γ.

Likewise one defines $\rho_{\overline{\Sigma}}(\overline{\Gamma})$.

Examples: i. $\rho_{\Sigma_{N,B}}(\Sigma_{D,SETS}) = \Sigma_B$

ii. $\rho_{\Sigma_N^{\perp}}(\Sigma_N^{S,P,+}) = \Sigma_N^{S,P}$, where $\Sigma_N^{S,P} =$

S: N
F: S: N → N
P: N → N
C: 0

iii. $\rho_{\Sigma_{D,SETS}}(\Sigma_{D,\perp}^{STACK}) = \Sigma_D$, where $\Sigma_D =$

S: D
F:
C:

§6. Isomorphism of signatures

Both for ordered and for unordered signatures we need a notion of isomorphism,

$$\Sigma \cong \Gamma \text{ and } \overline{\Sigma} \cong \overline{\Gamma}.$$

For (un)ordered signatures Σ and Γ, they are isomorphic if a consistent renaming transforms Σ into Γ and conversely. This definition works in both cases. Some examples clarify the matter:

i. $\Sigma_N^{S,P,ADD} \cong \Sigma_N^{S,P,+}$

renamings:	⇒	⇐
	N → N	N → N
	S → S	S → S
	P → P	P → P
	ADD → +	+ → ADD
	0 → 0	0 → 0

ii. $\Sigma_N^{P,S,ADD} \cong \Sigma_N^{S,P,+}$

renamings:	$\Rightarrow$	$\Leftarrow$
	$N \to N$	$N \to N$
	$S \to P$	$S \to P$
	$P \to S$	$P \to S$
	$ADD \to +$	$+ \to ADD$
	$0 \to 0$	$0 \to 0$

iii. $\Sigma_N^{S,+,P} \ncong \Sigma_N^{S,P,+}$

(where $\Sigma_N^{S,+,P}$ is $\left|\begin{array}{l} S\text{: } N \\ F\text{: } S\text{: } N \to N \\ \quad +\text{: } N \times N \to N \\ \quad P\text{: } N \to N \\ C\text{: } 0 \in N \end{array}\right.$)

§7. Renamings of signatures

Suppose $\Sigma \cong \Gamma$, $\rho_{\overline{\Gamma}}(\Delta) = \emptyset$ and $\Sigma \subseteq \Delta$.

By $\{\Sigma := \Gamma\}\ \Delta$

we denote the signature that is obtained by replacing each Σ-name in Δ by its corresponding Γ-name.

Similarly one defines $\{\Sigma := \Gamma\}\ \overline{\Delta}$.

Remark: It is amazing to see that already at this elementary level of signatures it is nontrivial (and perhaps even new) to produce a workable set of notions, notations and operations with watertight definitions.

§8. Calculations with signatures

Especially in connection with so-called parametrized datatypes a simple kind of computations is frequently used.

Examples:

i. Let $\Sigma_N = \left|\begin{array}{l} S\colon N \\ F\colon \\ C\colon \end{array}\right.$

Then $\Sigma_N \tilde{=} \Sigma_D$.

Now here are two interesting unordered signatures (with ordered representations Γ and Δ):

$$\overline{\Gamma} = \overline{\Sigma_N^{S,P,+}} + \overline{\{\Sigma_D := \Sigma_N\}\ \Sigma_{D,SETS}}$$

$$\overline{\Delta} = \overline{\{\Sigma_N := \Sigma_D\}\ \Sigma_N^{S,P,+}} + \overline{\Sigma_{D,SETS}}\ .$$

Remark: $\overline{\Gamma}$ and $\overline{\Delta}$ both have to do with substituting N for D in $\Sigma_{D,SETS}$.

ii. Another (slightly more complex) example is:

$$\overline{\Gamma'} = \overline{\Sigma_{N,B}} + \overline{\{\Sigma_D := \Sigma_N\}\ \Sigma_{D,\perp}^{STACK}}$$

$$\overline{\Delta'} = \overline{\{\Sigma_N := \Sigma_D\}\ \Sigma_{N,B}} + \overline{\Sigma_{D,\perp}^{STACK}}$$

iii. Still more complex:

$$\overline{\Sigma_{N,B}} + \overline{\{\Sigma_D := \Sigma_N\}\ (\Sigma_{D,\perp}^{STACK} + \overline{\Sigma_{D,SETS}}}$$

iv. Let $\Sigma_{ST} = \left|\begin{array}{l} S\colon ST \\ F\colon \\ C\colon \end{array}\right.$

and choose Γ' as in the second example. Then

$$\overline{\Theta} = \overline{\Gamma'} + \overline{\{\Sigma_D := \Sigma_{ST}\}\ \Sigma_{D,SETS}}$$

corresponds somehow with sets of stacks of nonnegative integers.

CHAPTER II. (TERM) ALGEBRAS

In this chapter we focus interest on the following matters:

i. T_Σ, the Σ-algebra of terms over signature Σ;

ii. the notion of a rewrite set R over T_Σ; this is a subset of $T_\Sigma \times T_\Sigma$;

iii. The congruence $\equiv(R)$ generated by a rewrite set R;

iv. The reduction relation $\twoheadrightarrow(R)$ generated by R.

§1. Algebras

Definition: An algebra is a triple

$$((A_1,\ldots,A_k), (f_1,\ldots,f_\ell), (c_1,\ldots,c_m))$$

where: 1. $k>0$, $\ell\geq 0$, $m\geq 0$;

2. the A_i are nonempty and pairwise disjoint;

3. the f_i are functions: $A_{j(i,1)} \times \ldots \times A_{j(i,t)} \to A_{u(i)}$ for appropriate $j(i,1),\ldots,j(i,t)$, $u(i) \in \{1,\ldots,k\}$;

4. the c_i are elements of $A_1 \cup \ldots \cup A_k$.

Examples:

1. $((\{0,1,\ldots\}), (\lambda x.x+1, \lambda x.x\dot{-}1), (0))$.

2. $((\{0,1,\ldots\}), (\lambda x.x+1, \lambda xy.x+y), (0,1))$.

3. $((\{t,f\}, \{0,1,\ldots\}), (\lambda x.x+1), (0,1,t,f))$.

We denote algebras with Gothic capitals $\mathfrak{A}$, $\mathfrak{B}$,...

A signed algebra is a pair $(\Sigma,\mathfrak{A})$ of a signature and an algebra where the (ordered) signature and the algebra correspond in such a way that Σ can be used as a naming scheme for the domains, functions and distinguished elements of $\mathfrak{A}$.

Example: $(\Sigma_N^{S,P},\mathfrak{A})$, with $\mathfrak{A}$ as in (1) above.

We denote signed algebras with A, B, C,...

With $\sigma(A)$ we denote the signature of A.

§2. Term algebras

Let Σ be a given signature. Using a simultaneous induction one defines for each $S \in S(\Sigma)$ the collection T_Σ^S of (closed) Σ-terms of sort S:

i. if $c \in S$ is in $C(\Sigma)$ then $c \in T_\Sigma^S$;

ii. if $f: S_{i_1} x \ldots x S_{i_n} \to S$ is in $F(\Sigma)$ and $t_1 \in T_\Sigma^{S_{i_1}}, \ldots, t_n \in T_\Sigma^{S_{i_n}}$, then $f(t_1,\ldots,t_n) \in T_\Sigma^S$.

Let $S(\Sigma) = \{S_1,\ldots,S_m\}$, then one can use $(T_\Sigma^{S_1},\ldots,T_\Sigma^{S_m})$ as domains for a Σ-algebra. The function $\hat{f}$ corresponding to f as in (ii) above then works as follows: $\hat{f}$: $t_1,\ldots,t_n \to f(t_1,\ldots,t_n)$, and to $c \in S$ corresponds the distinguished element $c \in T_\Sigma^S$ (we must require here that the T_Σ^S are nonempty).

We denote this term algebra with T_Σ. Clearly $\sigma(T_\Sigma) = \Sigma$.

Examples:

1. $\Sigma = \Sigma_N^{S,P}$.
$T_\Sigma^N = \{0,\ S(0),\ P(0),\ S(S(0)),\ S(P(0)),\ P(S(0)),\ P(P(0)),\ S(S(S(0))),\ldots\}$.

2. $\Sigma = \Sigma_{N,B}$. Typical terms of T_Σ^B are:
T, F, $((\neg T) \vee F) \vee (\neg(\neg T))$, eq(S(0),P(P(0))), eq(S(S(0)),0),
$\neg(\neg(\text{eq}(0,S(0))) \vee F) \vee \text{eq}(S(0),P(0)))$.

3. $\Sigma = \Sigma_{D,SETS}$. T_Σ^D is empty, and T_Σ^{SETS} contains just $\emptyset$.

4. $\overline{\Sigma} = \Sigma_N^{S,P,+} + \overline{\{\Sigma_D := \Sigma_N\}\ \Sigma_{D,SETS}}$.
T_Σ^{SETS} contains: ins(S(0) + P(0), ins(0,$\emptyset$)) ;
T_Σ^B contains: $\neg(\neg\ \text{elt}(S(0),\ \text{ins}(P(S(S(0))),\emptyset))\ \vee F)$.

§3. Congruences

Suppose all T_Σ^S are nonempty, so T_Σ exists.

Definition: A congruence on T_Σ is a family
$H^S \subseteq T_\Sigma^S \times T_\Sigma^S$, for $S \in S(\Sigma)$,
that satisfies the following conditions:

i. for all $t \in T_\Sigma^S$ $(t,t) \in H^S$;

ii. if (t,r), $(r,u) \in H^S$ then $(t,u) \in H^S$;

iii. if $(t,r) \in H^S$ then $(r,t) \in H^S$;

iv. if $(t_1,r_1) \in H^{S_{i_1}}, \ldots, (t_n,r_n) \in H^{S_{i_n}}$, and $f: S_{i_1} x \ldots x S_{i_n} \to S$ is in $F(\Sigma)$, then $(f(t_1,\ldots,t_n),f(r_1,\ldots,r_n)) \in H^S$.

§4. Fundamental construction

Let $H = \{H^S : S \in S(\Sigma)\}$ be a congruence on T_Σ.

We denote with $[t]$ the class of all u such that $(t,u) \in H^S$ (for $t \in T_\Sigma^S$); $[t]$ is the congruence class of t.

With T_Σ^S/H^S we denote the set of all congruence classes of T_Σ^S.

If we define $\hat{c} = [c]$, and $\hat{f}([t_1],\ldots,[t_n]) = [f(t_1,\ldots,t_n)]$, then $(\{T_\Sigma^S/H^S : S \in S(\Sigma)\}, \{\hat{f} : f \in F(\Sigma)\}, \{\hat{c} : c \in C(\Sigma)\})$ is a Σ-algebra. It is denoted by

$$T_\Sigma/H.$$

Notation: One often writes $\equiv$ for congruences, and superscripts for sorts are usually omitted.

$T_\Sigma/\equiv$ is also called T_Σ modulo $\equiv$.

§5. Rewrite sets

Definitions: 1. Let $R \subseteq T_\Sigma \times T_\Sigma$. We call such R a rewrite set.

2. With $\equiv_R$ we denote the smallest congruence that includes R; $\equiv_R$ is the congruence generated by R.

3. With R^0 we denote the reduction (rewrite) relation generated by R.

Formally: i. for all $t \in T_\Sigma$ $(t,t) \in R^0$;

ii. $(t,t') \in R \Rightarrow (t,t') \in R^0$;

iii. $(t,t') \in R^0$ & $(t',t'') \in R^0 \Rightarrow (t,t'') \in R^0$;

iv. $(t,t') \in R^0 \Rightarrow (\tau(t),\tau(t')) \in R^0$,

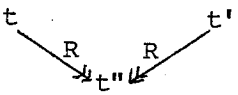
(here $\tau(t) \in T_\Sigma$; $\tau(t)$ is called a context of t, i.e. a term with t as subterm).

We write $t \xrightarrow{R} t'$ for $(t,t') \in R^0$, and $t \xrightarrow{R} t'$ for $(t,t') \in R$.

Given the rewrite set R, we can now define some important properties relating to the global behavior of Ξ_R and R^0:

Definitions: 1. $t \in T_\Sigma$ is a normal form if there is no $t' \in T_\Sigma$, $t \neq t'$ with $t \xrightarrow{R}\!\!\!\!\!\rightarrow t'$.

2. R^0 is confluent (or Church-Rosser) if for all t,t' such that $t \Xi_R t'$ there is a t" with



3. R^0 is weakly normalising if for all t there is a t' such that $t \xrightarrow{R}\!\!\!\!\!\rightarrow t'$ and t' is in normal form.

4. R^0 is strongly normalising (or terminating) if there exists no infinite sequence $t_1, t_2, t_3, \ldots$ such that $t_i \neq t_{i+1}$ and $t_i \xrightarrow{R}\!\!\!\!\!\rightarrow t_{i+1}$ for all $i \in \{1,2,\ldots\}$.

§6. Meaning of (term) algebras

Minimal algebras are algebras of the form $A = T_\Sigma/\Xi$.
If A is minimal then each object of A is the interpretation of a closed term. In Gu [75] and GTWr[75], GTW [78] it has been explained that abstract datatypes can be modeled as minimal algebras.

CHAPTER III. EQUATIONS

§1. Definitions and examples

Let us again consider a signature Σ. For each sort S of Σ we assume the presence of variables $X = \{x_i^S : i \in \omega, S \in S(\Sigma)\}$.
In the present framework we look at such a variable x_i^S as a meta variable ranging over all terms in T_Σ^S. This view is not the purely algebraic one. In algebra one would view a variable as ranging over all objects of its type (sort). But because we are interested in minimal algebras, these views coincide for all practical purposes.

If we allow the x_i^S as terms of type S we obtain the sets

$$T_\Sigma^S(X) \qquad (\text{for } S \in S(\Sigma))$$

of terms with variables. Clearly $T_\Sigma^S \subseteq T_\Sigma^S(X)$ for all $S \in S(\Sigma)$.
A Σ-equation is a pair (t_1^S, t_2^S), with $t_i^S \in T_\Sigma^S(X)$.
Equations are always written as follows:

$$t_1^S = t_2^S .$$

It we want to display the free variables we may write

$$t(x_1,\ldots,x_n) = t'(x_1,\ldots,x_n)$$

(omitting the sort superscripts as usual).

Definition: Let E be a set of equations. The rewrite set R(E) generated by E is the set of all pairs $(t(t_1,\ldots,t_n), t'(t_1,\ldots,t_n))$ with $t(x_1,\ldots,x_n) = t'(x_1,\ldots,x_n)$ an equation of E and $t_1,\ldots,t_n$ closed terms (i.e. terms without variables) of the right sorts.

Note: If $\equiv_{R(E)}$ is the congruence generated by E, then

$$\boxed{T_\Sigma/\equiv_{R(E)} = T_I(\Sigma,E)}$$

is the initial algebra of (Σ,E).
A pair (Σ,E) is often called an equational specification.
(Σ,E) is called an initial algebra specification of an algebra A if

$$T_I(\Sigma,E) \cong A.$$

Examples of specifications of familiar algebras with signatures taken from chapter I:

i. (Σ_B, E) with

E: $\neg(T) = F$, $\neg(F) = T$, $T \vee x = T$, $F \vee T = T$, $F \vee F = F$

or E: $\neg(T) = F$, $\neg(F) = T$, $T \vee T = T$, $F \vee T = T$, $x \vee y = y \vee x$, $F \vee F = F$

ii. $(\Sigma_N^{S,P,+}, E)$ with

E: $P(0) = 0$, $P(S(x)) = x$, $x + 0 = x$, $x + S(y) = S(x + y)$

or E: $P(0) = 0$, $P(S(x)) = x$, $0 + x = x$, $S(x) + y = S(x + y)$

iii. $(\Sigma_N^{S,P,+,\cdot}, E)$ with

E: $P(0) = 0$, $P(S(x)) = x$, $x + 0 = x$, $x + S(y) = S(x + y)$, $x.0 = 0$, $x.S(y) = x.y + x$

iv. $(\Sigma_{N,B}, E)$ with

E: $\neg(T) = F$, $\neg(F) = T$, $T \vee T = T$, $T \vee F = T$, $F \vee T = T$, $F \vee F = F$, $eq(x,x) = T$, $eq(0,S(x)) = F$, $eq(S(x),0) = F$, $eq(S(x),S(y)) = eq(x,y)$

v. $(\Sigma_N^{\perp}, E)$ with E: $S(\perp) = \perp$, $P(\perp) = \perp$, $P(0) = \perp$, $P(S(x)) = x$

In all these cases we have that $T_I(\Sigma,E)$ is isomorphic to a familiar algebra.

Remark: it usually requires a proof that $T_I(\Sigma,E) \cong A$. This proof is almost always tedious and occasionally quite nontrivial.

Warning: $T_I(\Sigma,E)$ need not always exist. Indeed if some T^S_Σ is empty then T_Σ is not defined and $T_I(\Sigma,E)$ isn't either. Of course it is easy to decide whether or not all T^S_Σ are nonempty.

Now consider $\Sigma_{D,SETS}$. In this case T^D_Σ is indeed empty. Still there is an interesting set of equations E for $\Sigma_{D,SETS}$:

vi. E:
$\neg(T) = F$
$\neg(F) = T$
$T \vee T = T$
$T \vee F = T$
$F \vee T = T$
$F \vee F = F$
$ins(x, ins(y,V)) \doteq ins(y, ins(x,V))$
$ins(x, ins(x,V)) = ins(x,V)$
$eq(x,x) = T$
$elt(x, ins(y,V)) = eq(x,y) \vee elt(x,V)$
$elt(x,\emptyset) = F$

To understand the properties of $(\Sigma_{D,SETS},E)$ we must see the sort D as a parameter that must be matched with another sort.

§2. Operations on specifications

Let (Σ,E) be a specification. The following definitions are all analogous to the ones in chapter I.

Definitions: i. $\rho_{\overline{\Gamma}}(\Sigma,E) = (\rho_{\overline{\Gamma}}(\Sigma), \rho_{\overline{\Gamma}}(E))$, where $\rho_{\overline{\Gamma}}(E)$ is obtained from E by leaving out all equations involving sorts or names outside $\overline{\Gamma}$.

ii. $\{\Sigma_0 := \Sigma_1\}\ (\Sigma,E) = (\{\Sigma_0 := \Sigma_1\}\ \Sigma, \{\Sigma_0 := \Sigma_1\}\ E)$, where $\{\Sigma_0 := \Sigma_1\}\ E$ is obtained from E by changing each Σ_0-name to the corresponding Σ_1-name.

iii. $\overline{(\Sigma,E)} = (\overline{\Sigma},E)$.

iv. $(\overline{\Sigma_1}, E_1) + (\overline{\Sigma_2}, E_2) = (\overline{\Sigma_1 + \Sigma_2}, E_1 \cup E_2)$.

Examples:

i. Let $\Sigma_{D,B} = \left|\begin{array}{l} S\text{: } D,\ B \\ F\text{: } eq\text{: } D \times D \to B \\ C\text{: } T \in B \\ \quad F \in B. \end{array}\right.$

Consider the specifications

$(\Sigma_{N,B}, E_1)$ with E_1 as E from example (iv) above;

$(\Sigma_{D,SETS}, E_2)$ with E_2 as E from (vi) above.

Now consider (Σ, E) with

$$(\overline{\Sigma, E}) = (\overline{\Sigma_{N,B}, E_1}) + \overline{\{\Sigma_{D,B} := \Sigma_{N,B}\}\ (\Sigma_{D,SETS}, E_2)}.$$

(Σ, E) indeed specifies finite sets of integers (with equality function) provided with insertion and test of elementhood.

ii. Let $\Sigma_D^{a,b} = \left|\begin{array}{l} S\text{: } D \\ F\text{: } \\ C\text{: } \perp_D,\ a,\ b. \end{array}\right.$

Consider $\Sigma_D^{a,b} + \Sigma_{D,\perp}^{STACK}$. For this signature we have the following specification: we have $(\Sigma_D^{a,b} + \Sigma_{D,\perp}^{STACK}, E)$ with

E: $\left|\begin{array}{ll} & pop(\emptyset) = \perp_S \\ & pop(\perp_S) = \perp_S \\ & push(\perp_S, y) = \perp_S \\ & push(x, \perp_D) = \perp_S \\ * & pop(push(x,a)) = x \\ * & pop(push(x,b)) = x \\ & top(\emptyset) = \perp_D \\ & top(\perp_S) = \perp_D \\ * & top(push(\emptyset,a)) = a \\ * & top(push(\emptyset,b)) = b \\ * & top(push(push(x,a),a)) = top(push(x,a)) \\ * & top(push(push(x,b),a)) = top(push(x,a)) \\ * & top(push(push(x,a),b)) = top(push(x,b)) \\ * & top(push(push(x,b),b)) = top(push(x,b)) \end{array}\right.$

Now these equations essentially involve the extra signature of $\Sigma_D^{a,b}$ (see the equations with *). In particular is an equation

pop(push(x,y)) = x

wrong because then

$\perp_S$ = pop(push(Ø,$\perp_D$)) = Ø.

Leaving out the * equations from E however, we do not obtain a workable specification of the STACK. This is an essential difference between the cases $\Sigma_{D,SETS}$ and $\Sigma_{D,\perp}^{STACK}$.

In the section on priority rewrite systems we will suggest a solution to this problem.

CHAPTER IV. TERM REWRITE SYSTEMS

Let (Σ,E) be a specification. An important difficulty is to find an implementation of $T_I(\Sigma,E)$.

In particular one needs a method to decide the word problem:

$$\boxed{T_I(\Sigma,E) \models t_1 = t_2}$$

for closed terms t_1 and t_2.

The paradigm is as follows:

Let $R(E)^0$ be the reduction relation generated by the rewrite set belonging to the equations E.

Assumption: Assume that $R(E)^0$ is both confluent and terminating.

Then $t_1=t_2$ is decided as follows: apply repeated rewrite steps on t_1 and t_2 until both have reached a normal form (which will happen because $R(E)^0$ is terminating). Let $\overline{t_1}$, $\overline{t_2}$ be these normal forms.

If $T_\Sigma \models \overline{t_1} = \overline{t_2}$ then $T_I(\Sigma,E) \models t_1 = t_2$,

otherwise $T_I(\Sigma,E) \models t_1 \neq t_2$.

To see this note that if $T_I(\Sigma,E) \models t_1=t_2$ then $T_I(\Sigma,E) \models \overline{t_1}=\overline{t_2}$. Thus by confluence, $\overline{t_1}$ and $\overline{t_2}$ have a common reduct. As both terms are in normal form, they must be identical.

There is a solid amount of theory about term rewrite systems. For more information, see the survey paper HO [80].

Examples: examples (i) - (v) on equational specifications provide confluent and terminating rewrite systems as well.

Modularisation

Let us now consider the problem of modular specifications:

1. Modularity is fundamental for specifications because large equational specifications are notoriously hard to read.
2. Modularity is not a matter of modular notation (using blocks of equations etc.), but of true decomposition into subsystems.
3. Theoretically minded people study decomposition by developing appropriate composition principles.
4. In our setting the main composition principle is +:

(Σ_1,E_1), (Σ_2,E_2) $\rightarrow$ $(\overline{\Sigma_1,E_1})$ + $(\overline{\Sigma_2,E_2})$.

The theory of parametrized datatypes essentially studies this composition principle. (Usually in the context of many more features, and semantically based on categories.)

5. The main point now is to provide specifications that can be used in a flexible way as parts of a "sum".

Example: Let $(\Sigma_{D,SETS},E_1)$ be the specification given on page 17.

Moreover let $\Sigma^{eq}_{D,B}$ be: $\Sigma^{eq}_{D,B}$ = | S: D, B
| F: eq: D x D $\rightarrow$ B
| C: T $\in$ B
| F $\in$ B.

Also let $\Gamma^{eq}_{D,B} \supseteq \Sigma^{eq}_{D,B}$ such that $\rho_{\overline{\Sigma}_{D,SETS}}(\Gamma^{eq}_{D,B}) = \Sigma^{eq}_{D,B}$

(or rather: $\overline{\Gamma}^{eq}_{D,B} \cap \overline{\Sigma}_{D,SETS} = \overline{\Sigma}^{eq}_{D,B}$).

Now assume that $(\Gamma^{eq}_{D,B},E_0)$ is a specification with an initial algebra in which B = {T,F}, T $\neq$ F and eq(x,y)=T $\Leftrightarrow$ x=y.

Also choose (Σ,E) with

$$(\overline{\Sigma,E}) = (\overline{\Gamma^{eq}_{D,B},E_0}) + (\overline{\Sigma_{D,SETS},E_1}).$$

Then $T_I(\Sigma,E)$ is indeed the algebra of finite subsets of D with insertion and is-element-of function.

Therefore we have that $(\Sigma_{D,SETS},E_1)$ is a very useful module for specification.

Remark: As a term rewrite system $(\Sigma_{D,SETS},E_1)$ poses nontrivial but surmountable difficulties, because of the equation

$$ins(x, ins(y,V)) = ins(y, ins(x,V)).$$

As a reduction this is a so-called permutative one. Such reductions stand in the way of strong normalisation.

The problem:

Once more consider $\Sigma^{eq}_{D,B}$. It seems most obvious to look at this signature as a parametrized one. In this sense we look at equations E such that $(\overline{\Sigma^{eq}_{D,B},E})$ + $(\overline{\Gamma_D,E'})$ describes "D + equality function", whenever Γ_D is a signature with sort D

(and not naming B, T, F, eq).

Fact: Such E cannot be found. (Not even when one uses auxiliary sorts and functions, or even conditional equations.)
Proof: Otherwise each initial algebra would be decidable!
A solid proof of this fact requires a very systematic analysis of initial algebra semantics in the light of computability theory. In essence this work has been carried out in BK [83].

Conclusion: Equational specifications do not support proper modularisation (in unexpected cases).

CHAPTER V. PRIORITY REWRITE SYSTEMS

In this chapter we will propose a new mechanism for specifying term algebras (i.e. congruences) via rewrite sets.

It is claimed that this mechanism, called "Priority Rewrite Systems" (PRS), by its higher expressive power, supports modularity better than the TRS mechanism.

Because of the always important issue of automatic implementation, and the prejudice that automatic implementation for algebraic specifications is deeply connected with term rewriting, we search for more expressive power in such a way that the spirit of term rewriting is preserved.

(We had inspiration from discussions with Jan Heering, Paul Klint and Ed Kuypers. They pointed out the inadequacy of TRS theory in several examples. Of course this document does not necessarily reflect their views.)

§1. Definition

A priority rewrite system is a triple

$$(\Sigma, E, P)$$

where Σ is a signature, E is a finite set of named equations (rewrite rules) over Σ and P is a partial ordering on the names of equations in E.

Before giving a detailed semantics some examples are given. The names of rules are indicated thus:

$$\text{name: } t(\vec{x}) = t'(\vec{x}).$$

If no confusion arises names can be omitted. The order between equations is indicated by vertical arrows, for instance:

$$\begin{array}{l} \Big\downarrow \\ \\ \\ \Big\downarrow \\ \\ \Big\downarrow \end{array} \begin{array}{l} r\text{: } t_1 = t_2 \\ u\text{: } t_3 = t_4 \\ v\text{: } t_5 = t_6 \\ w\text{: } t_7 = t_8 \\ h\text{: } t_9 = t_{10} \\ i\text{: } t_{11} = t_{12}. \end{array}$$

This system corresponds to the partial ordering $<$: $r<u<v$, $h<i$.

The notation
$$\begin{array}{l} r\colon\ t_1 = t_2 \\ u\colon\ t_3 = t_4 \\ \downarrow \\ v\colon\ t_5 = t_6 \end{array}$$
means that r and u are incomparable, but both precede v, i.e. $r<v$ and $u<v$ but not $r<u$ or $u<r$.

We will not display $<$ as a third component of a PRS if no confusion can arise. If rule r_1 precedes r_2 in the partial order P ($(r_1,r_2) \in P$) we say that r_1 is of higher priority than r_2.

§2. Examples of notation

i. $\Sigma_{D,B}$:
$$\begin{array}{ll} S\colon & D \\ & B \\ F\colon & eq\colon D \times D \to B \\ C\colon & T \in B \\ & F \in B. \end{array}$$
$E_{D,B}$:
$$\begin{array}{l} eq(x,x) = T \\ \downarrow \\ eq(x,y) = F. \end{array}$$

ii. $\Sigma_U^{f,g}$:
$$\begin{array}{ll} S\colon & U \\ F\colon & f\colon U \times U \to U \\ & g\colon U \to U \\ C\colon & a \in U \\ & b \in U \\ & \bot \in U. \end{array}$$
$E_U^{f,g}$:
$$\begin{array}{l} g(\bot) = \bot \\ f(\bot,x) = \bot \\ f(x,\bot) = \bot \\ \downarrow \\ f(x,a) = a \end{array}$$

iii. Each TRS is a PRS with the empty ordering.

The intuitive meaning of priority between rules is this: if

r_1: $t_1(\vec{x}) \to t_1'(\vec{x})$ precedes

r_2: $t_2(\vec{x}) \to t_2'(\vec{x})$,

then we may apply r_2 only on a redex if r_1 cannot be applied, and could never be applied, after other rewrite steps.

§3. Informal example

Let $\Sigma_D^{d_1,d_2} = \left|\begin{array}{l} S\text{: } D \\ F\text{: } \mathrm{swap}\text{: } D \to D \\ C\text{: } d_1 \\ \quad\ d_2 \end{array}\right.$

and $E_D^{d_1,d_2} = \begin{array}{l} \mathrm{swap}(d_1) = d_2 \\ \mathrm{swap}(d_2) = d_1. \end{array}$

Note that this is just a TRS.

Now consider $(\overline{\Sigma}_{D,B}, E_{D,B}) + (\overline{\Sigma}_D^{d_1,d_2}, E_D^{d_1,d_2})$,

where the partial ordering on the set of equations is just the union of the partial orderings. The result of this union is this PRS:

$$\left|\begin{array}{l} \mathrm{swap}(d_1) = d_2 \\ \mathrm{swap}(d_2) = d_1 \\ \mathrm{eq}(x,x) = T \\ \mathrm{eq}(x,y) = F. \end{array}\right.$$

We experiment with some reductions (of closed terms):

$\mathrm{swap}(\mathrm{swap}(\mathrm{swap}(d_1))) \to \mathrm{swap}(\mathrm{swap}(d_2)) \to \mathrm{swap}(d_1) \to d_2.$

$\mathrm{eq}(\mathrm{swap}(d_1),\mathrm{swap}(\mathrm{swap}(d_2))) \to \mathrm{eq}(d_2,\mathrm{swap}(d_1)) \to \mathrm{eq}(d_2,d_2) \to T.$

$\mathrm{eq}(\mathrm{swap}(\mathrm{swap}(d_2)),\mathrm{swap}(d_2)) \to \mathrm{eq}(\mathrm{swap}(d_1),\mathrm{swap}(d_2)) \to \mathrm{eq}(d_2,\mathrm{swap}(d_2))$

$\mathrm{eq}(d_2,d_1) \overset{*}{\to} F.$

Here the reduction $\mathrm{eq}(d_2,d_1) \overset{*}{\to} F$ is allowed because the only rule with higher priority cannot be applied, which is clear from the fact that d_1 and d_2 are in normal form.

Remarks about the informal example:

i. $(\Sigma_{D,B}, E_{D,B})$ acts as the module which was impossible to describe using

a TRS, according to what we said on page 22.

ii. We consider the PRS $(\overline{\Sigma},E) = (\overline{\Sigma}_{D,B},E_{D,B}) + (\overline{\Sigma}_D^{d_1,d_2},E_D^{d_1,d_2})$ to be the result of putting the parameter $(\Sigma_D^{d_1,d_2},E_D^{d_1,d_2})$ into a parametrized datatype (given by $(\Sigma_{D,B},E_{D,B})$).

As we shall later see formally, (Σ,E) determines a rewrite set and a congruence $\equiv$ on T_Σ. It turns out that $T_\Sigma/\equiv$ can also be described without priorities by this TRS:

$$\begin{cases} swap(d_1) = d_2 \\ swap(d_2) = d_1 \\ eq(x,x) = T \\ eq(d_1,d_2) = F \\ eq(d_2,d_1) = F. \end{cases}$$

If we are interested in an equational specification only we can use this one:

$$\begin{cases} swap(d_1) = d_2 \\ swap(swap(x)) = x \\ eq(x,x) = T \\ eq(swap(x),x) = F. \end{cases}$$

(As a TRS this one is useless!)

iii. Let $\Sigma_D^{S,0} = \begin{cases} S\colon D \\ F\colon S\colon D \to D \\ C\colon 0. \end{cases}$

Consider $(\overline{\Sigma}_{D,B},E_{D,B}) + (\overline{\Sigma}_D^{S,0},\emptyset)$.

This specification describes integers with equality function; again in the resulting PRS the arrows can be eliminated (see the example on page 16, after renaming N by D).

§4. Formal semantics of a PRS.

In the informal explanation on page 25, one observes a circularity. We will now introduce a method to deal with semantical problems of this type.

4.1 Definition: let (Σ,E) be a PRS. A labeled rewrite set $R^\ell = R^\ell(\Sigma,E)$ is a set of triples (r,t_1,t_2), with

r: $t(\vec{x}) = t'(\vec{x})$ a rule in E (so $t,t' \in T_\Sigma(X)$), and
$t_1 \in T_\Sigma$ an instance of $t(\vec{x})$,
$t_2 \in T_\Sigma$ the corresponding instance of $t'(\vec{x})$,
(so only outermost rewrites are allowed).

4.2 Note: Clearly there is a maximal labeled rewrite set $R^{\ell}_{max}(\Sigma,E)$ for (Σ,E), and a minimal one, $R^{\ell}_{min}(\Sigma,E) = \emptyset$.
If (Σ,E) is a TRS then $R^{\ell}_{max}(\Sigma,E) = R(\Sigma,E) = R(E)$, as defined on page 15 (without the labels, which are not essential anyway in the case of a TRS).

4.3 Outline of semantics: A semantics for (Σ,E) is a labeled rewrite set R as defined in 4.1. We'll look at the following properties:

(1) R is sound;

(2) R is complete;

(3) R is perfect.

We are especially interested in cases (Σ,E) where there is a unique perfect rewrite set R for (Σ,E).
Recall that R^0 is the reduction relation generated by R (this is done for a labeled R just as for an unlabeled one).

4.4 Note: First we are going to define what it means for a rule-application to be correct. The definition has to be more complicated than we informally stated on page 25, as is illustrated by the following example:
we consider finite sets of nonnegative integers with insertion and deletion:

$\Sigma_{N,SET}$ =
S: N, SET
F: S: N → N
ins: N x SET → SET
del: N x SET → SET
C: 0 ∈ N
∅ ∈ SET

$E_{N,SET}$ =
r_1: ins(x, ins(x,A)) = ins(x,A)
r_2: ins(x, ins(y,A)) = ins(y, ins(x,A))
r_3: del(x, ins(x,A)) = del(x,A)
r_4: del(x,A) = A.

In 6.7 we'll prove that $(\Sigma_{N,SET}, E_{N,SET})$ has a unique perfect rewrite set. Intuitively, the application:

(*) $\quad del(0, del(0, ins(0,\emptyset))) \xrightarrow{r_4} del(0, ins(0,\emptyset))$

is correct, since $del(0, ins(0,\emptyset)) \twoheadrightarrow \emptyset$, so 0 "is not an element of" $del(0, ins(0,\emptyset))$.

However, we can apply r_3 to the result of (*), which seems to contradict our informal definition of correctness. The solution is, that the outermost redex on the lefthand side of (*) is not the same as the outermost redex on the righthand side. Therefore, we say that an application

$$del(x,A) \xrightarrow{r_4} A$$

is correct unless there is a $y \in T_\Sigma^N$ and a $B \in T_\Sigma^{SET}$ such that $x \twoheadrightarrow y$ and $A \twoheadrightarrow ins(y,B)$. This leads us to the following definition.

4.5 Definition: Let (Σ,E) be a PRS and let r_1: $t_1=s_1$ and r_2: $t_2=s_2$ be two rules in E (so $t_1,t_2,s_1,s_2 \in T_\Sigma(X)$).
The matching context of r_1 and r_2, notation C^{r_1,r_2}, is the largest context such that both t_1 and t_2 are substitution instances of it, i.e.
i. there are $\vec{u},\vec{v} \in T_\Sigma(X)$ such that $C^{r_1,r_2}(\vec{u})$ is t_1 and $C^{r_1,r_2}(\vec{v})$ is t_2; and
ii. if C' satisfies property (i), then C^{r_1,r_2} is a substitution instance of C'.

4.6 Example: For $(\Sigma_{N,SET}, E_{N,SET})$, defined in 4.4, we have
$C^{r_1,r_2}(x_1,x_2,A_1) = ins(x_1,ins(x_2,A_1))$;
$C^{r_1,r_3}(x_1) = C^{r_1,r_4}(x_1) = C^{r_2,r_3}(x_1) = C^{r_2,r_4}(x_1) = x_1$ (the empty context);
and $C^{r_3,r_4}(x_1,A_1) = del(x_1,A_1)$.

4.7 Definition: Let $(r,t,t') \in R^{\ell}_{max}(\Sigma,E)$ and let $R \subseteq R^{\ell}_{max}(\Sigma,E)$.
i. We say that (r,t,t') is incorrect with respect to R if
there is a rule r_1 with higher priority than r, and
there are $t_1,t_1',u_1,\ldots,u_n,v_1,\ldots,v_n \in T_\Sigma$ such that
1. t is $C^{r,r_1}(\vec{u})$ and t_1 is $C^{r,r_1}(\vec{v})$;
2. $u_i \xrightarrow{R^0} v_i$ for each $i \in \{1,\ldots,n\}$;
3. $(r_1,t_1,t_1') \in R$.

ii. (r,t,t') is incorrect if it is incorrect w.r.t. $R^{\ell}_{max}(\Sigma,E)$.
iii. (r,t,t') is correct (w.r.t. R) if it is not incorrect (w.r.t. R).

4.8 Definition: Let R be a labeled rewrite set for a PRS (Σ,E).
i. R is sound if each $(r,t,t') \in R$ is correct w.r.t. R;
ii. the closure of R, notation $\overline{R}$, is the set of all rewrites correct w.r.t. R;
iii. R is complete if $R \supseteq \overline{R}$;
iv. R is perfect if R is sound and complete.

Thus we see that for each PRS (Σ,E):

R is sound iff $R \subseteq \overline{R}$
R is complete iff $R \supseteq \overline{R}$
R is perfect iff $R = \overline{R}$

4.9 Remark: In case (Σ,E) is a TRS, R(E) is perfect and this perfect rewrite set is unique.

§5. Theoretical matters

First of all we note that for a given PRS (Σ,E), $\emptyset$ is a sound rewrite set and $R^{\ell}_{max}(\Sigma,E)$ is complete. It is the existence and uniqueness of perfect rewrite sets that poses problems. With respect to implementations confluence and termination of R^0 are quite important (but not investigated here).

5.1 Lemma: Let R, P be rewrite sets for (Σ,E) then:
i. $R \subseteq P \Rightarrow \overline{R} \supseteq \overline{P}$;
ii. $R \supseteq P$ and P perfect $\Rightarrow$ R complete;
iii. $R \subseteq P$ and P perfect $\Rightarrow$ R sound.
Proof: i. follows immediately from the definition of $\overline{R}$;
ii. suppose $R \supseteq P$ then by (i) $\overline{R} \subseteq \overline{P} = P$, thus $R \supseteq \overline{R}$;
iii. if $R \subseteq P$ then $\overline{R} \supseteq \overline{P} = P$, whence $R \subseteq \overline{R}$.

5.2 Notation: $\overline{R} = R$; $\overline{R}^{n+1} = \overline{\overline{R}^{n}}$.

Now suppose that $R = R^{\ell}_{max}(\Sigma,E)$ and that P is some perfect rewrite set. Then we find the following picture:

$$\begin{array}{ccccccc} R = \overline{R}^0 & \supseteq & \overline{R}^2 & \supseteq & \overline{R}^4 & \supseteq & \ldots \\ \cup & & \cup & & \cup & & \\ P & = & P & = & P & = & \ldots \\ \cup & & \cup & & \cup & & \\ \overline{R}^1 & \subseteq & \overline{R}^3 & \subseteq & \overline{R}^5 & \subseteq & \ldots \end{array}$$

Moreover $\bigcap_{n=0}^{\infty} \overline{R}^{2n}$ is a complete rewrite set (by lemma 5.1.ii) and $\bigcup_{n=0}^{\infty} \overline{R}^{2n+1}$ is sound (by 5.1.iii).

5.3 Lemma: If for some n $\overline{\overline{R^{\ell}_{max}(\Sigma,E)}}^{n} = \overline{\overline{R^{\ell}_{max}(\Sigma,E)}}^{n+1}$, then there is a unique perfect rewrite set for (Σ,E).

Proof: immediate.

5.4 Let us now consider the special case that the TRS, obtained by omitting the partial order, is strongly normalising (this occurs in several of the examples later on).

5.5 Lemma: Let (Σ,E) be a PRS. Assume that $R^{\ell}_{max}(\Sigma,E)^0$ is strongly normalising, and that R is a unique perfect rewrite set for (Σ,E). Then R is decidable.

Proof: Let $\langle t_n : n < \omega \rangle$ be an effective enumeration of the closed terms over Σ (this is possible, since Σ is given effectively). Any rewrite set for (Σ,E) can be represented by a subset S of $\omega \times k \times \omega$, where k is the number of rules in E.

A close inspection of the definition of soundness yields that soundness is a Π^0_1-property of S. Completeness is a Π^0_2-property in general, but since $R^{\ell}_{max}(\Sigma,E)^0$ is strongly normalising, it is a Π^0_1-property.

Now we consider the definability of R.

We have $(r_i : t_j \to t_\ell) \in R \iff$

$\forall S \subseteq \omega \times k \times \omega$ [$(j,i,\ell) \in S \vee$ (S not sound) $\vee$ (S not complete)].

The part between the square brackets is Σ^0_1, so by König's lemma, R is Σ^0_1

as well. Similarly $(r_i: t_j \to t_\ell) \notin R \iff$

$\forall S \subseteq \omega \times k \times \omega$ [$(j,i,\ell) \notin S$ V (S not sound) V (S not complete)],

which gives that the complement of R is also Σ_1^0, from which it follows that R is decidable.

5.6 Definition: We call a PRS (Σ,E) ambiguous if there are two rewrites (r_1,t,t_1), $(r_2,t,t_2) \in R^{\ell}_{max}(\Sigma,E)$ such that r_1 and r_2 are incomparable (i.e. $r_1 \neq r_2$, $r_1 \not< r_2$ and $r_2 \not< r_1$).

5.7 Theorem: Suppose (Σ,E) is an unambiguous PRS and $R^{\ell}_{max}(\Sigma,E)$ is lenghth-decreasing. Then there is at most one perfect rewrite set for (Σ,E).

Proof: Suppose R_1 and R_2 are two different perfect rewrite sets for (Σ,E). Choose $(r: s \to t) \in (R_1-R_2) \cup (R_2-R_1)$ such that s is of minimal lenghth. Without loss of generality, we can assume that $(r: s \to t) \in R_1-R_2$. Since $(r: s \to t) \notin R_2$, and R_2 is complete, it is incorrect with respect to R_2, so there is a rule $r'<r$ and reductions

$$s \xrightarrow{r_1} t_1 \xrightarrow{r_2} \ldots \xrightarrow{r_n} t_n \quad \text{in } {R_2}^0$$

(where each reduction does not change the matching context of r and r'), and a reduction $t_n \xrightarrow{r'} t_{n+1}$ in R_2 (so outermost).
Because R_1 is sound, not all these reductionsteps can be in ${R_1}^0$, so at least one step is in ${R_2}^0-{R_1}^0$. However, every reduction is lenghth-decreasing, and the lenghth of s was chosen to be minimal, so we must have $(s \xrightarrow{r_1} t_1) \in R_2-R_1$. Then, since (Σ,E) is unambiguous, we must have that r_1 and r are comparable.
If $r=r_1$, we get an immediate contradiction; if $r<r_1$, we get that R_2 is not sound, and if $r_1<r$, we get that R_1 is not sound.
Therefore, we have a contradiction, and the proof is finished.

§6. Examples

6.1 Take $\Sigma = \begin{vmatrix} S: V \\ F: \\ C: a, b \end{vmatrix}$ and $E = \begin{vmatrix} r_1: a = b \\ \downarrow \\ r_2: b = a. \end{vmatrix}$

Claim: (Σ,E) has no perfect rewrite set.

Proof: The only complete rewrite set is $\{(r_1,a,b), (r_2,b,a)\}$, but that one is not sound.

6.2 Take $\Sigma = \begin{cases} S: V \\ F: \\ C: a, b, c, d \end{cases}$ and $E = \begin{cases} r_1: a = d \\ r_2: c = b \\ r_3: b = d \\ r_4: c = a. \end{cases}$

Then (Σ,E) has two different perfect rewrite sets R_1 and R_2:

$R_1 = \{(r_1,a,d), (r_2,c,b), (r_3,b,d)\}$ and

$R_2 = \{(r_1,a,d), (r_3,b,d), (r_4,c,a)\}$.

6.3 In this and the following examples we will show the existence of a unique perfect rewrite set.

Let the specification NAT be given by:

$\Sigma = \Sigma_N^{S,P,+}$ (see Chapter I) and E: $r_1: P(0) = 0$

$r_2: P(S(x)) = x$

$r_3: x + 0 = x$

$r_4: x + y = S(x + P(y))$.

Theorem: (Σ,E) has a unique perfect rewrite set, which is confluent.

Proof: Define the algebra $N = (\mathbb{N}, (\lambda x.x+1, \lambda x.x\dot{-}1, \lambda xy.x+y), (0))$ as usual, and an interpretation $\phi: T_\Sigma \to N$ in the natural way.

We define a rewrite set P by:

$P = \{\text{all instances of } r_1, r_2, r_3\} \cup$

$\cup \{t_1+t_2 \xrightarrow{r_4} S(t_1+P(t_2)) : t_1,t_2 \in T_\Sigma, \phi(t_2) \neq 0\}$.

We prove that P is perfect by first proving two lemmas.

soundness lemma: reductions in P preserve ϕ, i.e.

if $t_1 \xrightarrow{P^0} t_2$, then $\phi(t_1) = \phi(t_2)$.

proof: just check that each reduction-step is correct:

r_1: $\phi(P(0)) = 0 \dot{-} 1 = 0 = \phi(0)$;

r_2: $\phi(P(S(x))) = \phi(S(x)) \dot{-} 1 = (\phi(x) + 1) \dot{-} 1 = \phi(x)$;

r_3: $\phi(x + 0) = \phi(x) + \phi(0) = \phi(x) + 0 = \phi(x)$;

r_4: if $\phi(y) \neq 0$, then $\phi(x + y) = \phi(x) + \phi(y) = \phi(x) + ((\phi(y) \dot{-} 1) + 1) =$

$= \phi(x) + \phi(P(y)) + 1 = \phi(x + P(y)) + 1 = \phi(S(x + P(y)))$.

definition: for $n \in \mathbb{N}$ we define $S^n(0) \in T_\Sigma$ by induction:

$S^0(0)$ is 0 and $S^{n+1}(0)$ is $S(S^n(0))$.

completeness lemma: for all $t \in T_\Sigma$ there is an $n \in \mathbb{N}$ such that $t \xrightarrow{P^0}\!\!\!\!\!\rightarrow S^n(0)$.

proof: we first show by induction on n, that if $m,n \in \mathbb{N}$, then

$$S^m(0) + S^n(0) \xrightarrow{P^0}\!\!\!\!\!\rightarrow S^{m+n}(0).$$

- If n=0, $S^m(0) + S^0(0) = S^m(0) + 0 \xrightarrow{r_3} S^m(0) = S^{m+0}(0)$
  (Note: here = stands for identity between terms).
- If it is true for n, then $\phi(S^{n+1}(0)) = n+1 \neq 0$, so
  $(S^m(0) + S^{n+1}(0) \xrightarrow{r_4} S(S^m(0) + P(S^{n+1}(0))) = S(S^m(0) + P(S(S^n(0)))) \xrightarrow{r_2}$
  $S(S^m(0) + S^n(0)) \xrightarrow{P^0}\!\!\!\!\!\rightarrow S(S^{m+n}(0)) = S^{m+n+1}(0)) \in P^0$.

Now we prove the lemma by induction on t:

a) t = 0. immediate.

b) t = P(t'). By induction hypothesis, there is $n \in \mathbb{N}$ with $t' \xrightarrow{P^0}\!\!\!\!\!\rightarrow S^n(0)$.
   case 1: n=0. Then $S^n(0) = 0$, so $(t = P(t') \xrightarrow{P^0}\!\!\!\!\!\rightarrow P(0) \xrightarrow{r_1} 0 = S^0(0)) \in P^0$.
   case 2: n>0. Then $(t = P(t') \xrightarrow{P^0}\!\!\!\!\!\rightarrow P(S^n(0)) = P(S(S^{n-1}(0))) \xrightarrow{r_2} S^{n-1}(0))$
   $\in P^0$.

c) t = S(t'). immediate.

d) t = t' + t". By induction hypothesis, there are $m,n \in \mathbb{N}$ with
   $t' \xrightarrow{P^0}\!\!\!\!\!\rightarrow S^m(0)$ and $t'' \xrightarrow{P^0}\!\!\!\!\!\rightarrow S^n(0)$. Then $t = t' + t'' \xrightarrow{P^0}\!\!\!\!\!\rightarrow S^m(0) + S^n(0) \xrightarrow{P^0}\!\!\!\!\!\rightarrow$
   $S^{m+n}(0)$ by the above.

Claim 1: P is perfect.

Proof: Let $t_1,t_2 \in T_\Sigma$. Then $(t_1 + t_2 \xrightarrow{r_4} S(t_1 + P(t_2))) \in P \iff \phi(t_2) \neq 0$
$\iff$ there is n>0 with $t_2 \xrightarrow{P^0}\!\!\!\!\!\rightarrow S^n(0) \iff$ not$(t_2 \xrightarrow{P^0}\!\!\!\!\!\rightarrow 0) \iff$
the reduction $(t_1 + t_2 \xrightarrow{r_4} S(t_1 + P(t_2)))$ is correct w.r.t. P.

Claim 2: P is unique.

Proof: Suppose $P' \neq P$ is also perfect. Take $(t \xrightarrow{r} t') \in (P'-P) \cup (P-P')$ such that the lenghth of t is minimal. Then we must have $r=r_4$, and there are $t_1,t_2 \in T_\Sigma$ with $t = t_1 + t_2$. Then $(t \xrightarrow{r} t') \notin P' \iff t_2 \xrightarrow{P'^0}\!\!\!\!\!\rightarrow 0 \iff$
$t_2 \xrightarrow{P^0}\!\!\!\!\!\rightarrow 0 \iff \phi(t_2) = 0 \iff (t \xrightarrow{r} t') \notin P$, contradiction.

Claim 3: P is confluent.

Proof: each term has a normal form $S^n(0)$ by the completeness lemma.

6.4 In this and the following examples we'll look at a PRS that can be used as a module.

Let the module STACK be given by:

Σ_{ST}: S: D, ST
F: pop: ST → ST
top: ST → D
push: ST x D → ST
C: $\perp_D \in D$
$\emptyset, \perp_S \in ST$,

E_{ST}: r_1: push($\perp_S$,y) = $\perp_S$
r_2: push(x,$\perp_D$) = $\perp_S$
r_3: pop($\emptyset$) = $\perp_S$
r_4: top($\emptyset$) = $\perp_D$
r_5: pop($\perp_S$) = $\perp_S$
r_6: pop(push(x,y)) = x
r_7: top($\perp_S$) = $\perp_D$
r_8: top(push(x,y)) = y.

Note that $R^{\ell}_{max}(\Sigma_{ST},E_{ST})$ is lenghth-decreasing and unambiguous.
This is only a module, and does not give interesting reductions (the only datum is $\perp_D$, the only stacks are $\perp_S$ and $\emptyset$).
Therefore, we consider a specification (Σ_1,E_1) with $\rho_{\overline{\Sigma_{ST}}}(\Sigma_1) = \Sigma_{D,\perp}$, where

$\Sigma_{D,\perp}$ = S: D
F:
C: $\perp_D$.

(Think e.g. of $\Sigma_1 = \{\Sigma_N := \Sigma_D\}\ \Sigma_N^{\perp}$).

Definition: a specification (Σ_1,E_1) with $\Sigma_{D,\perp} \subseteq \Sigma_1$ and rewrite set P_1 has property (*) if $x \equiv_{P_1} \perp_D \Rightarrow x \xrightarrow{P_1}{}^{0}\!\!\twoheadrightarrow \perp_D$.

Note: if P_1 is confluent and $\perp_D$ is in normal form, then P_1 has property (*).

Theorem: Put $(\Sigma,E) = (\Sigma_{ST},E_{ST}) + (\Sigma_1,E_1)$.
i. If (Σ_1,E_1) has a unique perfect rewrite set P_1 with property (*), then (Σ,E) has a unique perfect rewrite set P.
ii. If moreover P_1 is confluent, then P is confluent.

Proof: i) The term algebra for (Σ_1,E_1) has as elements equivalenceclasses $[x]$ (under P_1) for $x \in T_{\Sigma_1}$. An algebra for (Σ,E) will have finite sequences of these as elements, so it will consist of:
a) $\{[x] : x \in T_{\Sigma_1}\}$;
b) elements $\emptyset$, $\perp_S$ (here $\emptyset$ is the empty sequence);
c) $\{<[x_1],\ldots,[x_n]> : \text{for } i \leq n \text{ we have } x_i \in T^D_{\Sigma_1}, \text{ and } x_i \not\equiv_{P_1} \perp_D\}$.
Now we define an interpretation ϕ from T_{Σ} to this algebra by induction on terms:

a) $\phi(x) = [x]$ if $x \in T_{\Sigma_1}$,

b) $\phi(\lfloor_S) = \lfloor_S$,

c) $\phi(\emptyset) = \emptyset$,

d) $\phi(\text{push}(x,y)) = \begin{cases} \lfloor_S & \text{if } \phi(y)=[\lfloor_D] \text{ or } \phi(x)=\lfloor_S, \\ <\phi(y)>\hat{}\,\phi(x) & \text{otherwise,} \end{cases}$

(here $\hat{}$ is concatenation of sequences),

e) $\phi(\text{pop}(y)) = \begin{cases} \lfloor_S & \text{if } \phi(y) \in \{\lfloor_S,\emptyset\}, \\ \text{tail}(\phi(y)) & \text{otherwise,} \end{cases}$

f) $\phi(\text{top}(y)) = \begin{cases} [\lfloor_D] & \text{if } \phi(y) \in \{\lfloor_S,\emptyset\}, \\ \text{first}(\phi(y)) & \text{otherwise,} \end{cases}$

Note that this is well-defined, and if $x \in T_\Sigma^{ST}$, then $\phi(x)$ is a finite sequence or $\phi(x) \in \{\lfloor_S,\emptyset\}$.

Now define P = {all instances of E_1^0 and rules r_1-r_5,r_7 of E_{ST}} $\cup$
$\cup$ $\{\text{pop}(\text{push}(x,y)) \xrightarrow{r_6} x : x \in T_\Sigma^{ST},\ y \in T_\Sigma^D,\ \phi(y)\neq[\lfloor_D],\ \phi(x)\neq\lfloor_S\}$ $\cup$
$\cup$ $\{\text{top}(\text{push}(x,y)) \xrightarrow{r_8} x : x \in T_\Sigma^{ST},\ y \in T_\Sigma^D,\ \phi(y)\neq[\lfloor_D],\ \phi(x)\neq\lfloor_S\}$.

We will prove that P is the unique perfect rewrite set of (Σ,E).

soundness lemma: reductions in P preserve ϕ.

proof: as before, just check that each rule-application is correct.
For example, if $\phi(x)\neq[\lfloor_D]$ and $\phi(y)\neq\lfloor_S$, then $\phi(\text{push}(x,y)) = <\phi(y)>\hat{}\,\phi(x) \notin \{\lfloor_S,\emptyset\}$, so $\phi(\text{pop}(\text{push}(x,y))) = \text{tail}(\phi(\text{push}(x,y)) = \text{tail}(<\phi(y)>\hat{}\,\phi(x)) = \phi(x)$.

completeness lemma: for all $x \in T_{\Sigma_1}^D$ and $y \in T_\Sigma^{ST}$:
i. if $\phi(y)=\lfloor_S$, then $y \xrightarrow{P^0} \lfloor_S$;
ii. if $\phi(y)=\emptyset$, then $y \xrightarrow{P^0} \emptyset$;
iii. if $\phi(y) \notin \{\lfloor_S,\emptyset\}$, there are $v \in T_{\Sigma_1}^D$ and $w \in T_\Sigma^{ST}$ such that $v \notin [\lfloor_D]$, $\phi(w) \neq \lfloor_S$ and $y \xrightarrow{P_0} \text{push}(w,v)$;
iv. if $\phi(\text{top}(y)) = [x]$, then there is a $v \in [x]$ such that $\text{top}(y) \xrightarrow{P^0} v$ (so $v \in T_{\Sigma_1}^D$).

proof: First we define the lenghth of a term t, l(t), by induction on t:
a) if $t \in T_{\Sigma_1}$, $l(t)=1$;
b) if t is $\lfloor_S$ or $\emptyset$, $l(t)=1$;
c) $l(\text{push}(t_1,t_2)) = l(t_1) + l(t_2) + 1$;
d) $l(\text{pop}(t_1)) = l(t_1) + 1$;

e) $l(top(t_1)) = l(t_1) + 1$.

With this definition, $R^{\ell}_{max}(\Sigma,E)$ becomes lenghth-decreasing.

We prove the lemma by simultaneous induction on $l(y)$.

a) y is $\bot_S$. Then $\phi(y)=\bot_S$, so (i), (ii), (iii) are immediate.

(iv) $(top(\bot_S) \xrightarrow{r_7} \bot_D) \in P$, and $\phi(top(\bot_S)) = [\bot_D]$.

b) y is $\emptyset$. Then $\phi(y)=\emptyset$, so (i), (ii), (iii) are immediate.

(iv) $(top(\emptyset) \xrightarrow{r_4} \bot_D) \in P$, and $\phi(top(\emptyset)) = [\bot_D]$.

c) y is push(a,b) with $a \in T^{ST}_{\Sigma}$, $b \in T^{D}_{\Sigma}$, and suppose the lemma holds for a.

<u>case 1</u>: $\phi(b)=[\bot_D]$. If $b \in T^{D}_{\Sigma}-T^{D}_{\Sigma_1}$, b will contain a subterm of the form top(z), with $z \in T^{ST}_{\Sigma}$. However, $l(z) < l(y)$, so the induction hypothesis allows us to reduce top(z) to a term in $T^{D}_{\Sigma_1}$. By repeating this procedure, b reduces to a term $b' \in T^{D}_{\Sigma_1}$.

Then $[b'] = \phi(b') = \phi(b) = [\bot_D]$, so, since P_1 has property (*), $b' \xrightarrow{P^0} \bot_D$.

$\phi(y) = \phi(push(a,b)) = \bot_S$, since $\phi(b) = [\bot_D]$, so (i) applies and $(push(a,b) \xrightarrow{P^0} push(a,b') \xrightarrow{P^0} push(a,\bot_D) \xrightarrow{r_2} \bot_S) \in P^0$.

(ii) and (iii) don't apply and

(iv) $(top(push(a,b)) \xrightarrow{P^0} top(\bot_S) \xrightarrow{r_7} \bot_D) \in P^0$.

<u>case 2</u>: $\phi(a)=\bot_S$. By induction, $a \xrightarrow{P^0} \bot_S$, and $\phi(y) = \phi(push(a,b)) = \bot_S$, so (i) applies and $(push(a,b) \xrightarrow{P^0} push(\bot_S,b) \xrightarrow{r_1} \bot_S) \in P^0$.

(ii) and (iii) don't apply and

(iv) $(top(push(a,b)) \xrightarrow{P^0} top(\bot_S) \xrightarrow{r_7} \bot_D) \in P^0$.

<u>case 3</u>: otherwise. Then $\phi(y) \notin \{\bot_S,\emptyset\}$, so (i) and (ii) don't apply.

(iii) As in case 1 we can reduce b to a term $b' \in T^{D}_{\Sigma_1}$.

Then $[b'] = \phi(b') = \phi(b) \neq [\bot_D]$, so we are done.

(iv) $(top(push(a,b)) \xrightarrow{r_8} b) \in P$, and again reduce b to a $b' \in T^{D}_{\Sigma_1}$.

d) y is pop(a), with $a \in T^{ST}_{\Sigma}$, and suppose the lemma holds for a.

<u>case 1</u>: $\phi(a)=\bot_S$. By induction hypothesis $a \xrightarrow{P^0} \bot_S$. Also $\phi(y) = \bot_S$.

(i) $(pop(a) \xrightarrow{P^0} pop(\bot_S) \xrightarrow{r_5} \bot_S) \in P^0$.

(ii) and (iii) don't apply and

(iv) $(top(pop(a)) \xrightarrow{P^0} top(\bot_S) \xrightarrow{r_7} \bot_D) \in P^0$.

<u>case 2</u>: $\phi(a)=\emptyset$. By induction hypothesis $a \xrightarrow{P^0} \emptyset$. Also $\phi(y) = \bot_S$.

(i) $(pop(a) \xrightarrow{P^0} pop(\emptyset) \xrightarrow{r_3} \bot_S) \in P^0$.

(ii) and (iii) don't apply and

(iv) $(\mathrm{top}(\mathrm{pop}(a)) \xrightarrow{P^0} \mathrm{top}(\lfloor_S) \xrightarrow{r_7} \lfloor_D) \in P^0$.

case 3: Otherwise. By induction hypothesis, there are $v \in T^D_{\Sigma_1}$ and $w \in T^{ST}_\Sigma$ such that $v \notin [\lfloor_D]$, $\phi(w) \neq \lfloor_S$ and $a \xrightarrow{P^0} \mathrm{push}(w,v)$.

case 3.1: $\phi(w)=\emptyset$. Since $l(w) < l(a) < l(y)$, $w \xrightarrow{P^0} \emptyset$ by induction hypothesis. Then $\phi(y) = \phi(\mathrm{pop}(\mathrm{push}(w,v)) = \mathrm{tail}(\mathrm{push}(w,v)) = \mathrm{tail}(\langle\phi(v)\rangle\hat{}\,\phi(w)) = \phi(w) = \emptyset$, so (ii) applies and
$(\mathrm{pop}(a) \xrightarrow{P^0} \mathrm{pop}(\mathrm{push}(w,v)) \xrightarrow{r_6} w \xrightarrow{P^0} \emptyset) \in P^0$.

(i) and (iii) don't apply and
(iv) $(\mathrm{top}(\mathrm{pop}(a)) \xrightarrow{P^0} \mathrm{top}(\emptyset) \xrightarrow{r_4} \lfloor_D) \in P^0$.

case 3.2: Otherwise. Note that then $\phi(w) \notin \{\lfloor_S, \emptyset\}$, since $\phi(w)=\lfloor_S$ gives $\phi(a)=\lfloor_S$. Again $\phi(y) = \phi(w)$, so (iii) applies and as in case 3.1 we have $y \xrightarrow{P^0} w$, and use the induction hypothesis on w.

(i) and (ii) don't apply and
(iv) $\mathrm{top}(y) \xrightarrow{P^0} \mathrm{top}(w)$, again use the induction hypothesis on w.

Now we can finish the proof of (i) with three claims.

Claim 1: P is sound.

Proof: Suppose not, so $(t \xrightarrow{r} t') \in P$, but is incorrect w.r.t. P.

a) $r=r_6$. Then there are $x \in T^{ST}_\Sigma$ and $y \in T^D_\Sigma$ such that t is $\mathrm{pop}(\mathrm{push}(x,y))$ and $\mathrm{push}(x,y) \xrightarrow{P^0} \lfloor_S$ (for $C^{r_5,r_6} = \mathrm{pop}(\,)$).

Since $(t \xrightarrow{r_6} t') \in P$, we must have $\phi(y) \neq [\lfloor_D]$ and $\phi(x) \neq \lfloor_S$, but on the other hand $\phi(\mathrm{push}(x,y)) = \phi(\lfloor_S) = \lfloor_S$ by the soundness lemma, so by definition of ϕ we have $\phi(y)=[\lfloor_D]$ or $\phi(x)=\lfloor_S$. Contradiction.

b) Similar to (a).

Claim 2: P is complete.

Proof: Suppose not, so $(t \xrightarrow{r} t') \notin P$, but is correct w.r.t. P.

a) $r=r_6$. Take $x \in T^{ST}_\Sigma$ and $y \in T^D_\Sigma$ such that t is $\mathrm{pop}(\mathrm{push}(x,y))$.

Since $(t \xrightarrow{r_6} t') \notin P$, we must have $\phi(y)=[\lfloor_D]$ or $\phi(x)=\lfloor_S$.

If $\phi(y)=[\lfloor_D]$, we can reduce y to a $y' \in T^D_{\Sigma_1}$ by the completeness lemma (iv), and then $y' \xrightarrow{P_1^0} \lfloor_D$, since P_1 has property (*). Then
$((\mathrm{push}(x,y) \xrightarrow{P^0} \mathrm{push}(x,\lfloor_D) \xrightarrow{r_2} \lfloor_S) \in P^0$, so $(t \xrightarrow{r_6} t')$ is incorrect w.r.t. P, contradiction.

If $\phi(x)=\lfloor_S$, we have $x \xrightarrow{P^0} \lfloor_S$ by the completeness lemma (i), so

$(\mathrm{push}(x,y) \xrightarrow{P^0}\!\!\!\!\rightarrow \mathrm{push}(\lfloor_S, y) \xrightarrow{r_1} \lfloor_S) \in P^0$, and again $(t \xrightarrow{r_6} t')$ is incorrect w.r.t. P, contradiction.

b) $r=r_8$. Similar to (a).

Claim 3: P is unique.

Proof: By 5.7.

ii) For each element of the algebra we define a set of standard representatives, $SR \subseteq T_\Sigma$, as follows:

if $x \in T_{\Sigma_1}$, $SR([x]) = \{y \in T_{\Sigma_1} : y \in [x]\}$;

$SR(\emptyset) = \{\emptyset\}$; $SR(\lfloor_S) = \{\lfloor_S\}$;

if $x_1,\ldots,x_n \in T^D_{\Sigma_1}$, with $x_i \notin [\lfloor_D]$ (for $i \in \{1,\ldots,n\}$), then

$SR(\langle[x_1],\ldots,[x_n]\rangle) = \{\mathrm{push}(\mathrm{push}(\ldots(\mathrm{push}(\emptyset,y_1),y_2),\ldots,y_n) : y_i \in [x_i]\}$.

Claim: if $t \in T_\Sigma$, then there is a term $t' \in SR(\phi(t))$ such that $t \xrightarrow{P^0}\!\!\!\!\rightarrow t'$.

Proof: Follows immediately, using the completeness lemma (use induction for (iii)).

Corollary: if P_1 is confluent, then P is confluent.

Proof: Suppose $t \equiv_{P^0} t'$, $(t,t' \in T_\Sigma)$.

From the soundness lemma it follows by induction that $\phi(t)=\phi(t')$.

Take $t_1,t_2 \in SR(\phi(t))$ such that $t \xrightarrow{P^0}\!\!\!\!\rightarrow t_1$, $t' \xrightarrow{P^0}\!\!\!\!\rightarrow t_2$.

Then t_1 and t_2 only differ in a number of subterms from T_{Σ_1}. But these subterms must be pairwise congruent w.r.t. P_1, so have common reducts by assumption. Then t_1 and t_2 also have a common reduct, so t and t' do.

6.5 A perfect rewrite set for the remaining examples is found in a similar way. Also the proofs are similar.

Let the module QUEUE be given by:

Σ_Q:

S: D, Q

F: qout: Q → Q

out: Q → D

qin: D x Q → Q

C: $\lfloor_D \in D$

$\emptyset, \lfloor_Q \in Q$.

E_Q: r_1: $qin(x,\perp_Q) = \perp_Q$
r_2: $qin(\perp_D,y) = \perp_Q$
r_3: $qout(\emptyset) = \perp_Q$
r_4: $out(\emptyset) = \perp_D$
r_5: $qout(\perp_Q) = \perp_Q$
r_6: $qout(qin(x,\emptyset)) = \emptyset$
r_7: $qout(qin(x,y)) = qin(x,qout(y))$
r_8: $out(\perp_Q) = \perp_D$
r_9: $out(qin(x,\emptyset)) = x$
r_{10}: $out(qin(x,y)) = out(y)$

Let (Σ_1,E_1) be a specification with $\rho_{\Sigma_Q}(\Sigma_1) = \Sigma_{D,\perp}$, and define $(\Sigma,E) = (\Sigma_1,E_1) + (\Sigma_Q,E_Q)$.

Theorem: (i) If (Σ_1,E_1) has a unique perfect rewrite set P_1 with property (*), then (Σ,E) has a unique perfect rewrite set P.

(ii) If moreover P_1 is confluent, then P is confluent.

Proof: As this proof is so similar to the proof of 6.4, we'll just indicate the differences. The algebra is the same (just replace $\perp_S$ by $\perp_Q$).

Define ϕ by: (a) $\phi(x) = [x]$ if $x \in T_{\Sigma_1}$;

b) $\phi(\perp_Q) = \perp_Q$; (c) $\phi(\emptyset) = \emptyset$;

d) $\phi(qin(x,y)) = \begin{cases} \perp_Q & \text{if } \phi(x) = [\perp_D] \text{ or } \phi(y) = \perp_Q; \\ \phi(y)^\frown\langle\phi(x)\rangle & \text{otherwise;} \end{cases}$

e) $\phi(qout(y)) = \begin{cases} \perp_Q & \text{if } \phi(y) \in \{\perp_Q,\emptyset\}; \\ tail(\phi(y)) & \text{otherwise;} \end{cases}$

f) $\phi(out(y)) = \begin{cases} [\perp_D] & \text{if } \phi(y) \in \{\perp_Q,\emptyset\}; \\ first(\phi(y)) & \text{otherwise;} \end{cases}$

Define P by:

$P = \{\text{all instances of } P_1 \text{ and rules } r_1\text{-}r_5 \text{ and } r_8 \text{ of } E_Q\} \cup$
$\cup \{qout(qin(x,\emptyset)) \xrightarrow{r_6} \emptyset : x \in T^D_\Sigma,\ \phi(x) \neq [\perp_D]\} \cup$
$\cup \{qout(qin(x,y)) \xrightarrow{r_7} qin(x,qout(y)) : x \in T^D_\Sigma,\ y \in T^Q_\Sigma,\ \phi(x) \neq [\perp_D],$
$\phi(y) \notin \{\perp_Q,\emptyset\}\} \cup \{out(qin(x,\emptyset)) \xrightarrow{r_9} x : x \in T^D_\Sigma,\ \phi(x) \neq [\perp_D]\} \cup$
$\cup \{out(qin(x,y)) \xrightarrow{r_{10}} out(y) : x \in T^D_\Sigma,\ y \in T^Q_\Sigma,\ \phi(x) \neq [\perp_D],\ \phi(y) \notin \{\perp_Q,\emptyset\}\}.$

P is the unique perfect rewrite set of (Σ,E).

Again we have the

soundness lemma: reductions in P^0 preserve ϕ; and the

completeness lemma: for all $x \in T^D_{\Sigma_1}$ and $y \in T^Q_\Sigma$:

i) if $\phi(y) = \perp_Q$, then $y \xrightarrow{P^0}\!\!\!\!\!\rightarrow \perp_Q$;

ii) if $\phi(y) = \emptyset$, then $y \xrightarrow{P^0}\!\!\!\!\!\rightarrow \emptyset$;

iii) if $\phi(y) \notin \{\lfloor_Q, \emptyset\}$, there are $v \in T^D_{\Sigma_1}$ and $w \in T^Q_\Sigma$ such that $v \notin [\lfloor_D]$, $\phi(w) \neq \lfloor_Q$ and $y \xrightarrow{P^0} qin(v,w)$.

iv) if $\phi(out(y)) = [x]$, then there is a $v \in [x]$ such that $out(y) \xrightarrow{P^0} v$ (so $v \in T^D_{\Sigma_1}$).

Also like in 6.4, we use these lemma's to show that P is the unique perfect rewrite set for (Σ,E).

ii) We define SR on the algebra:

$SR([x]) = \{y \in T_{\Sigma_1} : y \in [x]\}$ if $x \in T_{\Sigma_1}$;

$SR(\emptyset) = \{\emptyset\}, SR(\lfloor_Q) = \{\lfloor_Q\}$;

$SR(<[x_1],\ldots,[x_n]>) = \{qin(y_n,qin(y_{n-1},\ldots,qin(y_1,\emptyset))\ldots)) : y_i \in [x_i]\}$ if $x_i \in T^D_{\Sigma_1}$, $x_i \notin [\lfloor_D]$ $(i \in \{1,\ldots,n\})$.

Then, we show as before:

If $t \in T_\Sigma$, then there is a $t' \in SR(\phi(t))$ with $t \xrightarrow{P^0} t'$.

If P_1 is confluent, then P is confluent.

6.6 Let the module TREE be given by:

Σ_{TR}:
- S: D, TR
- F: $<\ ,\ >$: TR x TR → TR
  - tr: D → TR
  - data: TR → D
  - R, L: TR → TR
- C: $\lfloor_D \in D$
  - $\lfloor_T \in TR$

E_{TR}:

r_1: $<x,\lfloor_T> = \lfloor_T$

r_2: $<\lfloor_T,x> = \lfloor_T$

r_3: $tr(\lfloor_D) = \lfloor_T$

r_4: $data(\lfloor_T) = \lfloor_D$

r_5: $data(<x,y>) = \lfloor_D$

r_6: $data(tr(x)) = x$

r_7: $L(tr(x)) = \lfloor_T$

r_8: $R(tr(x)) = \lfloor_T$

r_9: $L(\lfloor_T) = \lfloor_T$

r_{10}: $L(<x,y>) = x$

r_{11}: $R(\lfloor_T) = \lfloor_T$

r_{12}: $R(<x,y>) = y$.

Let (Σ_1,E_1) be such that $\rho_{\Sigma_{TR}}(\Sigma_1) = \Sigma_{D,\lfloor}$ and put $(\Sigma,E) = (\Sigma_1,E_1)+(\Sigma_{TR},E_{TR})$.

Theorem: if (Σ_1,E_1) has a unique perfect rewrite set P_1 with property (*), then (Σ,E) has a unique perfect rewrite set P.

If moreover P_1 is confluent, then P is confluent.

Proof: The initial algebra consists of equivalenceclasses of P_1, an element $\perp_T$ and finite binary trees with labels from $\{[x] : x \in T^D_{\Sigma_1}, x \notin [\perp_D]\}$.

Define ϕ by: (a) $\phi(x) = [x]$ if $x \in T_{\Sigma_1}$;

b) $\phi(\perp_T) = \perp_T$;

c) $\phi(\langle x,y\rangle) = \begin{cases} \perp_T & \text{if } \phi(x) = \perp_T \text{ or } \phi(y) = \perp_T; \\ \text{the tree with left part } \phi(x) \text{ and right part } \phi(y) & \text{otherwise;} \end{cases}$

d) $\phi(\text{data}(x)) = \begin{cases} [\perp_D] & \text{if } \phi(x)=\perp_T \text{ or } \phi(x) \text{ has more than one node;} \\ [d] & \text{if } \phi(x) \text{ has one node, labeled } [d]; \end{cases}$

e) $\phi(\text{tr}(d)) = \begin{cases} \perp_T & \text{if } \phi(d)=[\perp_D]; \\ \text{the tree with one node, labeled } \phi(d) & \text{otherwise;} \end{cases}$

f) $\phi(L(x)) = \begin{cases} \perp_T & \text{if } \phi(x)=\perp_T \text{ or } \phi(x) \text{ has one node;} \\ \text{the left part of } \phi(x) & \text{otherwise;} \end{cases}$

g) $\phi(R(x)) = \begin{cases} \perp_T & \text{if } \phi(x)=\perp_T \text{ or } \phi(x) \text{ has one node;} \\ \text{the right part of } \phi(x) & \text{otherwise.} \end{cases}$

Define P by:

$P = \{\text{all instances of } P_1 \text{ and rules } r_1\text{-}r_9 \text{ and } r_{11}\} \cup$
$\cup \{L(\langle x,y\rangle) \xrightarrow{r_{10}} x : \phi(x) \neq \perp_T, \phi(y) \neq \perp_T, x,y \in T^{TR}_\Sigma\} \cup$
$\cup \{R(\langle x,y\rangle) \xrightarrow{r_{12}} y : \phi(x) \neq \perp_T, \phi(y) \neq \perp_T, x,y \in T^{TR}_\Sigma\}.$

soundness lemma: reductions in P^0 preserve ϕ.

completeness lemma: for all $x \in T^{TR}_\Sigma$ and $d \in T^D_{\Sigma_1}$:

i) if $\phi(x) = \perp_T$, then $x \xrightarrow{P^0}\!\!\!\!\rightarrow \perp_T$;

ii) if $\phi(x)$ has one node, labeled $[d]$, then there is a $d_1 \in [d]$ such that $x \xrightarrow{P^0}\!\!\!\!\rightarrow \text{tr}(d_1)$;

iii) if $\phi(x)$ has more than one node, there are $y,z \in T^{TR}_\Sigma$ such that $\phi(y) \neq \perp_T$, $\phi(z) \neq \perp_T$ and $x \xrightarrow{P^0}\!\!\!\!\rightarrow \langle y,z\rangle$;

iv) if $\phi(\text{data}(x)) = [d]$, then there is a $d_1 \in [d]$ such that $\text{data}(x) \xrightarrow{P^0}\!\!\!\!\rightarrow d_1$.

As a result of these lemma's, we can prove that P is the unique perfect rewrite set for (Σ,E).

Next we define SR:

$SR([x]) = \{y \in T_{\Sigma_1} : y \in [x]\}$ if $x \in T_{\Sigma_1}$;

$SR(\lfloor_T) = \{\lfloor_T\}$;

if a is the tree with single node [x] ($x \in T^D_{\Sigma_1}$), then

$SR(a) = \{tr(y) : y \in [x]\}$; and

if a is a tree with more than one node, with left part b en right part c,

then $SR(a) = \{\langle x,y\rangle : x \in SR(b),\ y \in SR(c)\}$.

Then we can prove the following statements:

If $t \in T_\Sigma$, then there is a $t' \in SR(\phi(t))$ with $t \xrightarrow{P^0} t'$.

If P_1 is confluent, then P is confluent.

6.7 Let the module SET be given by:

Σ_{SET}:
- S: D, SET
- F: ins: D x SET → SET
  del: D x SET → SET
- C: $\emptyset \in$ SET

E_{SET}:
- r_1: ins(x, ins(x,y)) = ins(x,y)
- r_2: ins(x, ins(v,y)) = ins(v, ins(x,y))
- r_3: del(x, ins(x,y)) = del(x,y)
- r_4: del(x,y) = y.

Let (Σ_1,E_1) be a specification with $\rho_{\Sigma_{SET}}(\Sigma_1) = \Sigma_D$ (as defined on page 8) and put $(\Sigma,E) = (\Sigma_1,E_1) + (\Sigma_{SET},E_{SET})$.

(An example is $(\Sigma_{N,SET},E_{N,SET})$, see page 27.)

This example is different from the previous ones in that we need confluency of P_1 to get a unique perfect rewrite set. This is because rule r_3 requires us to recognise when two elements of D are equal.

Theorem: If (Σ_1,E_1) has a unique perfect rewrite set P_1 which is confluent, then (Σ,E) has a unique perfect rewrite set P which is confluent.

Proof: The initial algebra consists of:

a) $\{[x] : x \in T_{\Sigma_1}\}$; (b) all finite subsets of $\{[x] : x \in T^D_{\Sigma_1}\}$.

Define ϕ by: (a) $\phi(x) = [x]$ if $x \in T_\Sigma$; (b) $\phi(\emptyset) = \emptyset$;

c) $\phi(ins(x,y)) = \phi(y) \cup \{[x]\}$ ($x \in T^D_\Sigma = T^D_{\Sigma_1}$, $y \in T^{SET}_\Sigma$);

d) $\phi(del(x,y)) = \phi(y) - \{[x]\}$ ($x \in T^D_\Sigma$, $y \in T^{SET}_\Sigma$).

Then define $P = \{$all instances of P_1 and rules r_1-r_3 of $E_{SET}\} \cup$

$\cup \{del(x,y) \xrightarrow{r_4} y : x \in T^D_\Sigma,\ y \in T^{SET}_\Sigma,\ [x] \notin \phi(y)\}$.

soundness lemma: reductions in P^0 preserve ϕ.

completeness lemma: for all $x \in T_\Sigma^D$ and $y \in T_\Sigma^{SET}$:
if $[x] \in \phi(y)$, there are $v \in T_\Sigma^D$ and $w \in T_\Sigma^{SET}$ such that
$x \xrightarrow{P_1{}^0\,*} v$, $[x] \notin \phi(w)$ and $y \xrightarrow{P^0\,*} ins(v,w)$.

We can again show that P is perfect and unique.

Then define SR by:

$SR([x]) = \{y \in T_{\Sigma_1} : y \in [x]\}$ if $x \in T_{\Sigma_1}$;

$SR(\emptyset) = \{\emptyset\}$;

$SR(\{[x_1],\ldots,[x_n]\}) = \{ins(y_{\sigma(1)},ins(y_{\sigma(2)},\ldots,ins(y_{\sigma(n)},\emptyset))\ldots) :$
σ is a permutation on n, $y_i \in [x_i]\}$,
if $x_i \in T_\Sigma^D$ $(i \in \{1,\ldots,n\})$ and the $[x_i]$ are distinct.

This definition enables us to show:
If $t \in T_\Sigma$, then there is a $t' \in SR(\phi(t))$ such that $t \xrightarrow{P^0\,*} t'$.
P is confluent. (Note that all elements of SR(x) can be reduced to each other by using P^0 and r_2.)

6.8 Our final example is a version of Backus' Functional Programming (B [78]).

Let the specification FP be given by:

Σ_{FP}:

S: N, S, D, F

F: i: N → D
j: S → D
Ap: F x D → D
*: D x S → S
⊛: F x F → F
†: D x D x D → D
→: F x F x F → F
S: N → N
$\overline{\bullet}$: D → F
∘: F x F → F

C: 0 ∈ N | T, F ∈ D
$\emptyset$, $\perp_S \in S$ | id, 1, tl, eq, apndl, +, •, p ∈ F

Notes: 1. Some abbreviations we'll use to improve legibility are: f(x) for Ap(f,x); <x,y> for j(x*(y*Ø)), and $\bot$ for j($\bot_S$).

2. Think of D as N U S U {T,F}, so think of i and j as injections.

3. E_{FP} gives reduction relations only on T_Σ^D.

E_{FP}: r_1: $\bot * s = \bot_S$
r_2: $x * \bot_S = \bot_S$

r_3: apndl(<x,j(y)>) = j(x * y)
r_4: apndl(x) = $\bot$

r_5: $f \circledast g(x)$ = apndl(<f(x),g(x)>)
r_6: $f \to g; h(x) = f(x) \overset{T}{\to} g(x); h(x)$

r_7: $T \overset{T}{\to} x; y = x$
r_8: $F \overset{T}{\to} x; y = y$
r_9: $z \overset{T}{\to} x; y = \bot$

r_{10}: $\overline{x}(\bot) = \bot$
r_{11}: $\overline{x}(y) = x$

r_{12}: id(x) = x

r_{13}: $f \circ g(x) = f(g(x))$

r_{14}: 1($\bot$) = $\bot$
r_{15}: 1(j(x * s)) = x
r_{16}: 1(x) = $\bot$

r_{17}: tl($\bot$) = $\bot$
r_{18}: tl(j(x * s)) = j(s)
r_{19}: tl(x) = $\bot$

r_{20}: eq($\bot$) = $\bot$
r_{21}: eq(<x,x>) = T
r_{22}: eq(<x,y>) = F
r_{23}: eq(x) = $\bot$

r_{24}: +(<i(x),i(0)>) = i(x)
r_{25}: +(<i(x),i(S(0))>) = i(S(x))
r_{26}: +(<i(x),i(S(y))>) = +(<+(<i(x),i(y)>), i(S(0))>)
r_{27}: +(x) = $\bot$

r_{28}: •(<i(x),i(0)>) = i(0)
r_{29}: •(<i(x),i(S(y))>) = +(<•(<i(x),i(y)>),i(x)>)
r_{30}: •(x) = $\bot$

r_{31}: p(i(0)) = i(0)
r_{32}: p(i(S(x))) = i(x)
r_{33}: p(x) = $\bot$

Theorem: (Σ_{FP}, E_{FP}) has a unique perfect rewrite set P, which is confluent.

Proof: is not hard, but tedious.

Define the initial algebra $\mathcal{F}$ by:

a) $(\mathbb{N}, (+, \bullet, \text{suc}, \dot{-}1), (0)) \subseteq \mathcal{F}$;

b) $\{\bot, Ø, T, F\} \subseteq \mathcal{F}$; and

c) $t_1,\ldots,t_n \in \mathcal{F}-\{\bot\} \Rightarrow \langle t_1,\ldots,t_n\rangle \in \mathcal{F}$.

Define an interpretation $\phi: T_\Sigma^D \to \mathcal{F}$ by induction:

a) $\phi(i(0)) = 0$, $\phi(T) = T$, $\phi(F) = F$, $\phi(j(\emptyset)) = \emptyset$, $\phi(\bot) = \bot$;

b) if $x \in T_\Sigma^D$ and $s \in T_\Sigma^S$, then

$$\phi(j(x * s)) = \begin{cases} \bot & \text{if } \phi(x)=\bot \text{ or } \phi(j(s))=\bot; \\ \langle\phi(x)\rangle\hat{}\,\phi(j(s)) & \text{otherwise (here } \hat{} \text{ is concatenation);} \end{cases}$$

c) if $x,y,z \in T_\Sigma^D$, then

$$\phi(x \rightarrow y;\ z) = \begin{cases} \phi(y) & \text{if } \phi(x) = T; \\ \phi(z) & \text{if } \phi(x) = F; \\ \bot & \text{otherwise;} \end{cases}$$

d) if $n \in T_\Sigma^N$, then $\phi(i(S(n))) = suc(\phi(i(n)))$;

e) if $x \in T_\Sigma^D$, then we define $\phi(Ap(f,x))$ by induction on $f \in T_\Sigma^F$:

e1) $\phi(id(x)) = \phi(x)$;

e2) $$\phi(apndl(x)) = \begin{cases} \langle first(\phi(x))\rangle\hat{}\,second(\phi(x)) & \text{if } \phi(x) \text{ is a pair} \\ & \text{with } second(\phi(x)) \text{ a sequence and} \\ & first(\phi(x)) \neq \bot; \\ \bot & \text{otherwise;} \end{cases}$$

(Note: $x \in \mathcal{F}$ is a sequence if $x=\emptyset$ or x is obtained by (c) in the definition of $\mathcal{F}$.)

e3) $$\phi(1(x)) = \begin{cases} first(\phi(x)) & \text{if } \phi(x) \text{ is a nonempty sequence;} \\ \bot & \text{otherwise;} \end{cases}$$

e4) $$\phi(tl(x)) = \begin{cases} tail(\phi(x)) & \text{if } \phi(x) \text{ is a nonempty sequence;} \\ \bot & \text{otherwise;} \end{cases}$$

e5) $$\phi(eq(x)) = \begin{cases} T & \text{if } \phi(x) \text{ is a pair with } first(\phi(x)) = \\ & = second(\phi(x)); \\ F & \text{if } \phi(x) \text{ is a pair with } first(\phi(x)) \neq \\ & \neq second(\phi(x)); \\ \bot & \text{otherwise;} \end{cases}$$

e6) $$\phi(+(x)) = \begin{cases} first(\phi(x)) + second(\phi(x)) & \text{if } \phi(x) \text{ is a pair from } \mathbb{N}; \\ \bot & \text{otherwise;} \end{cases}$$

e7) $$\phi(\cdot(x)) = \begin{cases} first(\phi(x)) \cdot second(\phi(x)) & \text{if } \phi(x) \text{ is a pair from } \mathbb{N}; \\ \bot & \text{otherwise;} \end{cases}$$

e8) $$\phi(p(x)) = \begin{cases} \phi(x) \dot{-} 1 & \text{if } \phi(x) \in \mathbb{N}; \\ \bot & \text{otherwise;} \end{cases}$$

e9) $\phi(f \circ g(x)) = \phi(apndl(\langle f(x), g(x)\rangle))$;

e10) $\phi(f \to g;\ h(x)) = \phi(f(x) \rightarrow g(x);\ h(x))$;

e11) $\phi(\overline{y}(x)) = \begin{cases} \phi(y) & \text{if } \phi(x) \neq \bot; \\ \bot & \text{otherwise;} \end{cases}$

e12) $\phi(f \circ g(x)) = \phi(f(g(x)))$.

Then define P = {all instances of r_1-r_3, r_5-r_8, r_{10}, r_{12}-r_{14}, r_{17}, r_{20}, r_{24}, r_{25}, r_{28}, r_{29}, r_{31}, r_{32}} $\cup$ {apndl(x) $\xrightarrow{r_4}$ $\bot$: $\phi(x)$ is not a pair or second($\phi(x)$) is not a sequence} $\cup$ {z $\rightarrow$ x; y $\xrightarrow{r_9}$ $\bot$: x,y,z $\in T_\Sigma^D$, $\phi(z) \notin$ {T,F}} $\cup$ {$\overline{x}(y) \xrightarrow{r_{11}} x$: x,y $\in T_\Sigma^D$, $\phi(y) \neq \bot$} $\cup$ {1(j(x * s)) $\xrightarrow{r_{15}}$ x : x $\in T_\Sigma^D$, s $\in T_\Sigma^S$, $\phi(x) \neq \bot$, $\phi(j(s)) \neq \bot$} $\cup$ {1(x) $\xrightarrow{r_{16}}$ $\bot$: x $\in T_\Sigma^D$, $\phi(x) \neq \bot$, $\phi(x) = \emptyset$ or $\phi(x)$ is not a sequence} $\cup$ {tl(j(x * s)) $\xrightarrow{r_{18}}$ j(s) : x $\in T_\Sigma^D$, s $\in T_\Sigma^S$, $\phi(x) \neq \bot$, $\phi(j(s)) \neq \bot$} $\cup$ {tl(x) $\xrightarrow{r_{19}}$ $\bot$: x $\in T_\Sigma^D$, $\phi(x) \neq \bot$, $\phi(x) = \emptyset$ or $\phi(x)$ is not a sequence} $\cup$ {eq(<x,x>) $\xrightarrow{r_{21}}$ T : x $\in T_\Sigma^D$, $\phi(x) \neq \bot$} $\cup$ {eq(<x,y>) $\xrightarrow{r_{22}}$ F : x,y $\in T_\Sigma^D$, $\phi(x) \neq \bot$, $\phi(y) \neq \bot$, $\phi(x) \neq \phi(y)$} $\cup$ {eq(x) $\xrightarrow{r_{23}}$ $\bot$: x $\in T_\Sigma^D$, $\phi(x) \neq \bot$, $\phi(x)$ is not a pair} $\cup$ {+(<i(x),i(S(y))>) $\xrightarrow{r_{26}}$ +(<+(<i(x),i(y)>),i(S(0))>) : x,y $\in T_\Sigma^N$, $\phi(y) \neq 0$} $\cup$ {+(x) $\xrightarrow{r_{27}}$ $\bot$: x $\in T_\Sigma^D$, $\phi(x)$ is not a pair from $\mathbb{N}$} $\cup$ $\cup$ {•(x) $\xrightarrow{r_{30}}$ $\bot$: x $\in T_\Sigma^D$, $\phi(x)$ is not a pair from $\mathbb{N}$} $\cup$ {p(x) $\xrightarrow{r_{33}}$ $\bot$: x $\in T_\Sigma^D$, $\phi(x) \notin \mathbb{N}$}.

Soundness lemma: Reductions in P^0 preserve ϕ.

Completeness lemma: for all x $\in T_\Sigma^D$:

i) if $\phi(x) = \bot$, then $x \xrightarrow{P^0} \bot$;

ii) if $\phi(x) = T$, then $x \xrightarrow{P^0} T$;

iii) if $\phi(x) = F$, then $x \xrightarrow{P^0} F$;

iv) if $\phi(x) \in \mathbb{N}$, there is an n $\in \mathbb{N}$ such that $x \xrightarrow{P^0} i(S^n(0))$;

v) if $\phi(x) = \emptyset$, then $x \xrightarrow{P^0} j(\emptyset)$;

vi) otherwise, $\phi(x)$ is a nonempty sequence, and there are y $\in T_\Sigma^D$ and s $\in T_\Sigma^S$ such that $\phi(y) \neq \bot$, $\phi(j(s)) \neq \bot$ and $x \xrightarrow{P^0} j(y * s)$.

Proof: By induction on the lenghth of the term x, similar to the proof in 6.4.

Corollary: P is perfect, unique and confluent.

Proof: as before. Note that each term in T_Σ^D has a normal form in NF, where NF is given by: (a) $\bot$, T, F, j$\emptyset$ $\in$ NF; (b) $i(S^n(0)) \in$ NF for n $\in \mathbb{N}$; (c) If x, j(s) $\in$ NF-{$\bot$} then j(x * s) $\in$ NF (so x $\in T_\Sigma^D$, s $\in T_\Sigma^S$).

Abbreviations: 1) we define n-tuples on T_Σ^D and T_Σ^F by:

for n=0: $\langle\rangle = j\emptyset$ and $[\,] = \overline{j\emptyset}$;

for n>0: $\langle x_1,\ldots,x_n\rangle = \text{apndl}(\langle x_1,\langle x_2,\ldots,x_n\rangle\rangle)$ and

$[f_1,\ldots,f_n] = f_1 \circledast [f_2,\ldots,f_n]$. Note that this tallies with page 44.

2) Define selection functions n (for $n \in \mathbb{N}-\{0\}$) in T_Σ^F by:

1 is a constant in $C(\Sigma)$ and if n>0, we put $n+1 = n \circ t1$.

3) Some other abbreviations:

$\text{pair} = \text{eq} \circ [\text{id},[1,2]]$;

$\neg = \text{id} \to \overline{F}; \overline{T}$;

$V = \text{pair} \to (1 \to (2 \to \overline{T};\overline{T}); (2 \to \overline{T};\overline{F}))$; $\bot$;

$\& = \text{pair} \to \neg \circ V \circ [\neg \circ 1, \neg \circ 2]$; $\bot$.

Extensions of FP: An important feature of Backus' FP is the existence of least fixed points.

Let $t(f) \in T_\Sigma^F(X)$, with one free function variable f.

Define (Σ_t,E_t) by: Σ_t: S: F; F:; C: $c_t \in F$ and E_t: r_t: $c_t(x) = t(c_t)(x)$.

and define $(\Sigma_{FP}^t,E_{FP}^t) = (\Sigma_{FP},E_{FP}) + (\Sigma_t,E_t)$.

Notations: $\mu f.t(f) = c$, or $c = t(c)$.

Definition: $P_t = P \cup \{\text{all instances of } r_t\}$.

Theorem: If, for all $x \in T_\Sigma^D$, $c_t(x) \xrightarrow{P_t\to} y \in T_\Sigma^D$ (so $c_t(x)$ has a terminating reduction, the subterm c_t can be eliminated), then P_t is the unique confluent perfect rewrite set for (Σ_{FP}^t,E_{FP}^t). (Here, as before, $\Sigma=\Sigma_{FP}$.)

Proof: immediate by assumption, since r_t has no prior rule.

Note: In the same way it is possible to make, successively, many extensions of FP, provided the assumption of the theorem is satisfied each time.

Example: Let $g \in T_\Sigma^F$. Define $\alpha g = \mu f.\text{eq} \circ [\text{id},\overline{j\emptyset}] \to \overline{j\emptyset}; \text{apndl} \circ [g\circ 1, f\circ t1]$.

Claim: For each $x \in T_\Sigma^D$, $\alpha g(x) \xrightarrow{P\alpha\to} y \in T_\Sigma^D$.

Proof: By first reducing x, we can assume $x \in NF$.

case 1: $x = \bot$. First note $[\text{id},\overline{j\emptyset}](\bot) = \text{id} \circledast (\overline{j\emptyset} \circledast \overline{j\emptyset})(\bot) \xrightarrow{r_5}$

$\text{apndl}(\langle \text{id}(\bot),\text{apndl}(\langle \overline{j\emptyset}(\bot),\overline{j\emptyset}(\bot)\rangle)\rangle) \xrightarrow{r_{10},r_{12}}$

$\text{apndl}(\langle \bot,\text{apndl}(\langle \bot,\bot\rangle)\rangle) \xrightarrow{r_3} j(\bot * (\bot * \bot_S)) \xrightarrow{r_2} j(\bot_S) = \bot$.

Therefore $eq_\circ[id,\overline{j\emptyset}](\bot) \xrightarrow{r_{13}} eq([id,\overline{j\emptyset}](\bot)) \xrightarrow{P^0} eq(\bot) \xrightarrow{r_{20}} \bot$.

Then $\alpha g(\bot) \xrightarrow{r_\alpha} (eq_\circ[id,\overline{j\emptyset}] \to \overline{j\emptyset};\ apndl_\circ[g_\circ 1,\alpha g_\circ tl])(\bot) \xrightarrow{r_6}$

$(eq_\circ[id,\overline{j\emptyset}](\bot) \,\mathrm{r}\, \overline{j\emptyset}(\bot);\ apndl_\circ[g_\circ 1,\alpha g_\circ tl](\bot) \xrightarrow{P^0}$

$\bot \,\mathrm{r}\, \overline{j\emptyset}(\bot);\ apndl_\circ[g_\circ 1,\alpha g_\circ tl](\bot) \xrightarrow{r_9} \bot$.

case 2: $x \in \{T,F\} \cup \{S^n(0) : n \in \mathbb{N}\}$.

Then $\alpha g(x) \xrightarrow{r_\alpha} (eq_\circ[id,\overline{j\emptyset}] \to \overline{j\emptyset};\ apndl_\circ[g_\circ 1,\alpha g_\circ tl])(x) \xrightarrow{r_6,r_{13}}$

$eq([id,\overline{j\emptyset}](x)) \,\mathrm{r}\, \overline{j\emptyset}(x);\ apndl([g_\circ 1,\alpha g_\circ tl](x)) \xrightarrow{r_5,r_{13}}$

$eq(apndl(<id(x),apndl(<\overline{j\emptyset}(x),\overline{j\emptyset}(x)>)>)) \,\mathrm{r}\, \overline{j\emptyset}(x);\ apndl(apndl(<g(1(x)),$

$apndl(<\alpha g(tl(x)),\overline{j\emptyset}(x)>)>)) \xrightarrow{r_{11},r_{12},r_{16},r_{19}}$

$eq(apndl(<x,apndl(<j\emptyset,j\emptyset>)>)) \,\mathrm{r}\, j\emptyset;\ apndl(apndl(<g(\bot),apndl(<\alpha g(\bot),j\emptyset>)>))$

$\xrightarrow{r_3} eq(j(x*(j\emptyset*\emptyset))) \,\mathrm{r}\, j\emptyset;\ apndl(j(g(\bot)*(\alpha g(\bot)*\emptyset)) =$

$eq(<x,j\emptyset>) \,\mathrm{r}\, j\emptyset;\ apndl(j(g(\bot)*(\alpha g(\bot)*\emptyset)) \xrightarrow{r_{22},\ case\ 1}$

$F \,\mathrm{r}\, j\emptyset;\ apndl(j(g(\bot)*(\bot*\emptyset))) \xrightarrow{r_1,r_8} apndl(<g(\bot),\bot>) \xrightarrow{r_3} j(g(\bot)*\bot_S) \xrightarrow{r_2}$

$j(\bot_S) = \bot$.

case 3: $x = j\emptyset$. Then $\alpha g(j\emptyset) \xrightarrow{r_\alpha} (eq_\circ[id,\overline{j\emptyset}] \to \overline{j\emptyset};\ apndl_\circ[g_\circ 1,\alpha g_\circ tl])(j\emptyset)$

$\xrightarrow{P^0} eq(<id(j\emptyset),\overline{j\emptyset}(j\emptyset)>) \,\mathrm{r}\, \overline{j\emptyset}(j\emptyset);\ apndl(<g_\circ 1(j\emptyset),\alpha g_\circ tl(j\emptyset)>) \xrightarrow{r_{11},r_{12}}$

$eq(<j\emptyset,j\emptyset>) \,\mathrm{r}\, j\emptyset;\ldots \xrightarrow{r_{21}} T \,\mathrm{r}\, j\emptyset;\ldots \xrightarrow{r_7} j\emptyset$.

case 4: otherwise. Then, $\phi(x)$ is a nonempty sequence, and by definition of NF there are y, $js \in NF-\{\bot\}$ such that x is of the form $j(y*s)$.

Then $\alpha g(x) \xrightarrow{P^0} eq(<x,j\emptyset>) \,\mathrm{r}\, j\emptyset;\ apndl(<g(1(j(y*s))),\alpha g(tl(j(y*s)))>)$

$\xrightarrow{r_{15},r_{18},r_{22}} F \,\mathrm{r}\, j\emptyset;\ apndl(<g(y),\alpha g(js)>) \xrightarrow{r_8} apndl(<g(y),\alpha g(js)>)$.

Now we use the induction hypothesis on the subterm $\alpha g(js)$.

Corollary: for all $x_1,\ldots,x_n \in T_\Sigma^D$ and $g \in T_\Sigma^F$, if $\phi(x_i) \neq \bot$ for $i=1,\ldots,n$, then $\alpha g(<x_1,\ldots,x_n>) \xrightarrow{P_0 \alpha} <g(x_1),\ldots,g(x_n)>$.

Proof: use reductions as those above.

Examples: some other examples of correct extensions of FP:

$apndr = \mu f.\&_\circ[pair,eq_\circ[1,\overline{j\emptyset}]] \to [2];\ apndl_\circ[1_\circ 1,f_\circ[tl_\circ 1,2]]$;

$distl = \mu f.\&_\circ[pair,eq_\circ[2,\overline{j\emptyset}]] \to \overline{j\emptyset};\ apndl_\circ[[1,1_\circ 2],f_\circ[1,tl_\circ 2]]$;

$distr = \mu f.\&_\circ[pair,eq_\circ[1,\overline{j\emptyset}]] \to \overline{j\emptyset};\ apndl_\circ[[1_\circ 1,2],f_\circ[tl_\circ 1,2]]$;

$iota = \mu f.eq_\circ[id,\overline{S(0)}] \to [\overline{S(0)}];\ apndr_\circ[f_\circ p,id]$;

and for each $g \in T_\Sigma^F$:

$/g = \mu f.eq_\circ[tl,\overline{j\emptyset}] \to 1;\ g_\circ[1,f_\circ tl]$.

Backus' FP: We define the following congruence relation on T_Σ^F:

$$f = g \iff \forall x \in T_\Sigma^D \ (\ f(x) \text{ and } g(x) \text{ have a common reduct }).$$

Then all of Backus' "laws" follow as theorems from (Σ_{FP}, E_{FP}).

Example: Backus' III.4: $\alpha(f \circ g) = \alpha f \circ \alpha g$.

Proof: let $x \in T_\Sigma^D$. We can assume $x \in NF$.

case 1: $x \in \{T, F, \bot\} \cup \{S^n(0) : n \in \mathbb{N}\}$.

By the claim on page 47, we have $\alpha(f \circ g)(x) \twoheadrightarrow \bot$ and $\alpha f \circ \alpha g(x) \twoheadrightarrow \alpha f(\bot) \twoheadrightarrow \bot$.

case 2: otherwise. Then $\phi(x)$ is a sequence, and we use induction. We write this down informally, using the corollary on page 48.

First, $\alpha(f \circ g)(j\emptyset) \twoheadrightarrow j\emptyset$ and $\alpha f \circ \alpha g(j\emptyset) \twoheadrightarrow \alpha f(j\emptyset) \twoheadrightarrow j\emptyset$.

Next, if $x = \langle x_1, \ldots, x_n \rangle$,

$\alpha(f \circ g)(\langle x_1, \ldots, x_n \rangle) \twoheadrightarrow \langle f \circ g(x_1), \ldots, f \circ g(x_n) \rangle$ and

$\alpha f \circ \alpha g(\langle x_1, \ldots, x_n \rangle) \twoheadrightarrow \alpha f(\langle g(x_1), \ldots, g(x_n) \rangle) \twoheadrightarrow \langle f(g(x_1)), \ldots, f(g(x_n)) \rangle$.

Now use r_{10}.

REFERENCES

B [78] J.Backus, *Can programming be liberated from the Von Neumann style? A functional style and its algebra of programs,* CACM 21 (8), 1978.

BK [83] J.A.Bergstra & J.W.Klop, *Initial algebra specifications for parametrized datatypes,* EIK 19, 1/2, 1983, pp 17 - 31.

BTe [83] J.A.Bergstra & J.Terlouw, *Standard model semantics for DSL, a datatype specification language,* Acta Informatica 19, 1983, pp 97 - 113.

BT [83] J.A.Bergstra & J.V.Tucker, *The completeness of the algebraic specification methods for computable datatypes,* Information & Control, 54 (3), 1983, pp 186 - 200.

BTu [83] J.A.Bergstra & J.V.Tucker, *Initial and final algebra semantics for datatype specifications, two characterisation theorems,* SIAM Journal of Computing, 12 (2), 1983, pp 366 - 387.

E [79] H.-D.Ehrich, *On the theory of specification, implementation and parametrization of abstract datatypes,* JACM, 29 (1), 1979, pp 206 - 227.

EKTWW [79] H.Ehrig, H.-J.Kreowski, J.W.Thatcher, E.G.Wagner & J.B.Wright, *Parametrized datatypes in algebraic specification languages,* Proc. 7th ICALP, LNCS 85, 1979, pp 157 - 168.

G [83] H.Ganzinger, *Parametrized specifications: parameter passing and implementation with respect to observability,* ACM TOPLAS, 5 (3), 1983, pp 318 - 354.

GTW [78] J.A.Goguen, J.W.Thatcher & E.G.Wagner, *An initial algebra approach to the specification, correctness and implementation of abstract datatypes,* R.T.Yeh (ed.), Current trends in Programming Methodology IV, Data Structuring, Prentice Hall, Englewood Cliffs, New Jersey.

GTWr [75] J.A.Goguen, J.W.Thatcher & J.B.Wright, *Abstract datatypes as initial algebras and correctness of datatype representations,* Proceedings of ACM Conference on Computer Graphics, Pattern Recognition and Data Structure, ACM, New York, 1975.

Gu [75] J.V.Guttag, *The specification and application to programming of abstract datatypes,* Ph.D. Thesis, University of Toronto,

Department of Computer Science, 1975.

HO [80] G.Huet & D.C.Oppen, *Equations and rewrite rules, a survey,* Formal Languages, Perspectives and Open Problems, Academic Press, 1980.

K [80] S.Kamin, *Final datatype specifications, a new datatype specification method,* Proc. 7th ACM Symp. Principles Progr. Lang. Las Vegas, 1980.

Kl [83] H.A.Klaeren, *Algebraische Spezification, Eine Einführung,* Springer Lehrbuchreihe Informatik, 1983.

KL [83] B.Kutzler & F.Lichtenberger, *Bibliography on abstract datatypes,* Informatische Fachberichte 68, Springer, 1983.

PEE [81] U.Pletat, G.Engels & H.-D.Ehrich, *Operational semantics of algebraic specifications with conditional equations,* Forschungsbericht Nr. 118/81, Abteilung Informatik, Universität Dortmund, 1981.

SW [83] D.Sanella & M.Wirsing, *A kernel language for algebraic specifications and implementations,* M.Karpinski (ed.), Foundations of Computing Theory, Springer LNCS 158, 1983, pp 413 - 427.

W [79] M.Wand, *Final algebra semantics and datatype extensions,* JCSS 19, 1979, pp 27 - 44.