HILDESHEIMER INFORMATIK-BERICHTE

ISSN 0941-3014

Arend Rensink

Deterministic Pomsets

30/94 (November 1994)

UNIVERSITÄT HILDESHEIM

Dieser Bericht ist herausgegeben vom

Institut für Informatik

Postfach 10 13 63
31113 Hildesheim

(this page intentionally left blank)

Deterministic Pomsets*

Arend Rensink
Institut für Informatik, Universität Hildesheim
Postfach 101363, D–31113 Hildesheim
rensink@informatik.uni-hildesheim.de

Abstract

This paper is about partially ordered multisets (pomsets for short). We investigate a particular class of pomsets that we call *deterministic*, properly including all partially ordered *sets*, which satisfies a number of interesting properties: among other things, it forms a *distributive lattice* under pomset prefix (hence prefix closed sets of deterministic pomsets are prime algebraic), and it constitutes a *reflective subcategory* of the category of all pomsets.

For the deterministic pomsets we develop an algebra with a sound and (ω-)complete equational theory. The operators in the algebra are *concatenation* and *join*, the latter being a variation on the more usual *disjoint union* of pomsets with the special property that it yields the *least upper bound with respect to pomset prefix*.

This theory is then extended in several ways. We capture *refinement* of pomsets by incorporating homomorphisms between models as objects in the algebra and homomorphism application as a new operator. This in turn allows to formulate *distributed termination* and *sequential composition* of pomsets, where the latter is different from concatenation in that it is right-distributive over union. To contrast this we also formulate a notion of *global termination*. Each variation is captured equationally by a sound and ω-complete theory.

Contents

*The research reported in this paper was partially supported by the HCM Cooperation Network "EXPRESS" (Expressiveness of Languages for Concurrency) and the Esprit Basic Research Working Group 6067 CALIBAN (Causal Calculi Based on Nets). A preliminary version was presented at the European Summer School on Logic, Language and Information, Copenhagen, Denmark, August 1994.

1 Introduction

Partially ordered multisets, or *pomsets*, have been proposed by many authors to model the behaviour of concurrent systems. In this regard they could generalise the role that *traces* (sometimes called *words*) play for the description of sequential behaviour. Traces can be regarded as total orderings, and are therefore seen to correspond naturally to the runs of a system in which actions are in fact executed sequentially. Concurrent behaviour on the other hand, in which actions in different parts of a system are executed independently, would seem to lend itself less naturally to such a description. Because according to this point of view, the ordering among the elements in the model reflects an actual relation between the actions of the system, one might call it the *intensional interpretation* of ordering.

It should be mentioned that the argument that partial orders are more suitable to describe concurrent behaviour than total orders is contended by many. Indeed the intensional interpretation, however appealing, is by no means necessary. A major body of theory has been developed on the basis of the *interleaving assumption*, which says that *extensionally*, a system executing actions concurrently is no different from one that executes them in arbitrary order; and therefore, that total orders may describe concurrent behaviour with sufficient precision.

The debate between the adherents of the intensional, partial-order school and the extensional, interleaving school has been going on for quite some time, and a definitive answer does not yet seem to be forthcoming. In the meanwhile, the least one can do is to study and compare the models that are being proposed. This paper aims to contribute to the already considerable amount of material that has been collected in the course of that study.

Since strings are clearly a special case of pomsets, one way to study the latter is by generalising and extending existing theory about the former. This is indeed the approach that one generally finds in the literature. In particular, one may introduce, in addition to the usual notion of (string) concatenation, an operation to put elements in parallel, and study the objects that are generated in this way. Thus the concept of *regular languages* is extended to pomsets.

One aspect of strings that does not generalise well along these lines is that of *prefix* or *initial segment*. The property that one string is the initial segment of another induces a partial ordering relation over the set of strings, which has arbitrary greatest lower bounds; it is this fact that allows us to regard an arbitrary set of strings as a tree, and to unfold arbitrary transition systems into trees. For pomsets however, although a prefix relation may be defined, it no longer has greatest lower bounds.

1.1 Example. The pomsets $\boxed{\begin{matrix}a\to b\\ a\to c\end{matrix}}$ and $\boxed{\begin{matrix}&b\\ a\nearrow&\\ &c\end{matrix}}$ have common prefixes $\boxed{a\to b}$ and $\boxed{a\to c}$ but no largest

common prefix; in particular, $\boxed{\begin{array}{c} b \\ \nearrow \\ a \!\!\rightarrow\!\! c \end{array}}$ is itself not a prefix of $\boxed{\begin{array}{c} a \!\!\rightarrow\!\! b \\ a \!\!\rightarrow\!\! c \end{array}}$.

We take the above observation as the starting point of our study. Throughout this paper, we essentially concentrate on the subclass of pomsets in which infima are defined. We call these pomsets *deterministic*, for reasons to be explained below. Over deterministic pomsets, the prefix relation has a very rich structure: apart from infima it also has all suprema (of finite sets, since we regard finite pomsets only), and moreover the two distribute over one another. This in turn implies that every set of deterministic pomsets can be interpreted as a *prime algebraic basis* in *exactly* the same way a set of strings can be interpreted as a tree. Since prime algebraic bases play an important role in partial order semantics, which in fact mirrors the role of trees in interleaving semantics, we regard this as an encouraging result.

The *join* operation that yields the supremum of two pomsets is in fact a variant of the well-known *disjoint union* of pomsets that lies at the heart of most of the existing pomset theory. Concatenation and join give rise to a complete equational theory of deterministic pomsets. This theory and its variations form the main subject of this paper.

To continue this discussion on a slightly more concrete level, we now recall the basic definitions of pomsets and some theories that have been developed for (special cases of) pomsets. Throughout the paper, we consider pomsets abstractly, without taking into account the nature of the elements that are being ordered. The elements are assumed to be collected in a set $\mathbf{E}$; we will use the letters a–e to refer to arbitrary elements.

A *labelled partially ordered set* or *lposet* over $\mathbf{E}$ is a triple $p = \langle V, <, \ell \rangle$ where

- V is an arbitrary set of *vertices*;
- $< \subseteq V \times V$ is an irreflexive and transitive *ordering relation*;
- $\ell \colon V \to \mathbf{E}$ is a *labelling function*.

We will assume the existence of a large enough universe of vertices, closed under pairing. In examples we sometimes use the natural numbers for this purpose. The class of lposets over $\mathbf{E}$ is denoted $\mathbf{LPO}[\mathbf{E}]$. We use V_p, $<_p$ and ℓ_p to denote the components of an lposet p, and $\leq_p$ to denote the reflexive closure of $<_p$.

Two lposets p and q are *isomorphic*, denoted $p \cong q$, if there exists a bijection $f \colon V_p \to V_q$ (the so-called *isomorphism*) such that for all $v, w \in V_p$, $v <_p w$ iff $f(v) <_q f(w)$ and $\ell_p(v) = \ell_q(f(w))$. A *partially ordered multiset* or *pomset*, finally, is an *isomorphism class* of lposets. The class of pomsets over $\mathbf{E}$ is denoted $\mathbf{POM}[\mathbf{E}]$ $(= \mathbf{LPO}[\mathbf{E}]/\!\cong)$. We use $[p] = \{q \in \mathbf{LPO}[\mathbf{E}] \mid q \cong p\}$ or $[V_p, <_p, \ell_p]$ to denote the pomset with representative p; by abuse of notation, we sometimes also write p for the pomset itself.

Graphically, we depict lposets by diagrams $\boxed{\begin{array}{c} {}_1a \\ \searrow \\ {}_2b \!\!\rightarrow\!\! {}_3a \end{array}}$, where $1, 2, 3$ are vertices and a, b their labels; and the corresponding pomsets by $\boxed{\begin{array}{c} a \\ \searrow \\ b \!\!\rightarrow\!\! a \end{array}}$, i.e. by deleting the vertex identifiers.

Strings. A very special case of partially ordered multisets are the *strings* over a given set of elements. Here the partial ordering is actually total. It is well known that strings are free monoids, meaning that they are freely generated by the signature $\Sigma_{str} = \langle \varepsilon, \cdot \rangle$ with the following equations:

$$\varepsilon \cdot x = x \tag{1}$$

$$x \cdot \varepsilon = x \tag{2}$$

$$(x \cdot y) \cdot z = x \cdot (y \cdot z) \tag{3}$$

ε denotes the *empty string* and $\cdot$ *concatenation of strings*; the equations state that concatenating the empty string to the left or right does not change a string, and that concatenation is associative.

The pomsets that can be generated in this way are precisely those p whose ordering is total, i.e., such that either $v \leq_p w$ or $w \leq_p v$ for all $v, w \in V_p$. Hence for instance $\boxed{a \to b \to a}$ can be generated but $\boxed{\begin{matrix} a \to b \\ a \end{matrix}}$ cannot. The empty string ε is modelled by the empty pomset $[\varnothing, \varnothing, \varnothing]$, a single element $e \in \mathbf{E}$ by $[\{0\}, \varnothing, \{(0, e)\}]$ (where 0 is a simple placeholder without intrinsic meaning), and the concatenation of p and q is defined by

$$p \cdot q = [V_p \cup V_q, <_p \cup (V_p \times V_q) \cup <_q, \ell_p \cup \ell_q]$$

where the representatives p and q are *disjoint*, i.e. are chosen such that $V_p \cap V_q = \varnothing$.

Multisets. Another very special case of partially ordered multisets are the *multisets* (sometimes called *bags*) over a given set of elements. Here the elements are actually completely unordered. Multisets are known to constitute free *commutative* monoids; that is, they are freely generated by the signature $\Sigma_{mul} = \langle \varepsilon, \uplus \rangle$ with the following equations:

$$\varepsilon \uplus x = x \tag{4}$$
$$(x \uplus y) \uplus z = x \uplus (y \uplus z) \tag{5}$$
$$x \uplus y = y \uplus x \tag{6}$$

ε now denotes the *empty multiset* and $\uplus$ *multiset addition*; the latter is associative and commutative, whereas adding the empty multiset does not change a multiset. Note that the symmetric variant of (4), $x \uplus \varepsilon = x$, follows directly from the commutativity of $\uplus$.

The pomsets that can be generated in this way are precisely those without any ordering whatsoever, i.e. those p with $<_p = \varnothing$. Hence, for instance, $\boxed{\begin{matrix} a \\ a \end{matrix}}$ can be generated but $\boxed{a \to a}$ cannot. The empty multiset and single-element multisets are modelled in the same way as the empty string and single-element strings above; multiset addition is modelled by disjoint pomset union:

$$p \uplus q = [V_p \cup V_q, <_p \cup <_q, \ell_p \cup \ell_q]$$

where again the representatives p and q should be disjoint.

Mazurkiewicz traces. An interesting mixture of strings and multisets can be found in the *Mazurkiewicz traces*, sometimes also called *partially commutative monoids*; see e.g. Mazurkiewicz [15] and Aalbersberg and Rozenberg [1]. Here one assumes not a standard set of elements $\mathbf{E}$, but rather a set with structure $\langle \mathbf{E}, I \rangle$ (sometimes called a *concurrent alphabet*) where $I \subseteq \mathbf{E} \times \mathbf{E}$ is an irreflexive and symmetric *independency relation*. This relation controls the degree to which the concatenation operator (which we will denote $\odot$ rather than $\cdot$ to distinguish it from string concatenation) is commutative: $d \odot e = e \odot d$ precisely when d and e are independent. Mazurkiewicz traces, then, are freely generated by $\Sigma_{Maz} = \langle \varepsilon, \odot, I \rangle$ with equations

$$\varepsilon \odot x = x \tag{7}$$
$$(x \odot y) \odot z = x \odot (y \odot z) \tag{8}$$

and rules for I to extend it from elements to traces:

$$\vdash \varepsilon \; I \; x \tag{9}$$
$$x \; I \; y \vdash y \; I \; x \tag{10}$$
$$x \; I \; y, x \; I \; z \vdash x \; I \; (y \odot z) \tag{11}$$
$$x \; I \; y \vdash x \odot y = y \odot x \tag{12}$$

Note that $x \odot \varepsilon = x$ is derivable from (7), (9) and (12). It follows that if $I = \varnothing$ (no independent elements) then the above system collapses to that for strings, whereas if $I = (\mathbf{E} \times \mathbf{E}) \setminus \{(e, e) \mid e \in \mathbf{E}\}$ (total irreflexive independence) then it collapses to that for multisets.

The ordering in the pomsets generated by the above signature satisfies the following condition: for all $v, w \in V_p$, if $v \not<_p w$ and $w \not<_p v$ then $\ell_p(v)\ I\ \ell_p(w)$, whereas if $v <_p w$ then $\exists u \in V_p . v <_p u \leq_p w$ and $\neg(\ell_p(v)\ I\ \ell_p(u))$. This implies that in principle, only the dependent elements are ordered; some additional orderings must be due to transitive closure. Hence, for instance, if $c\ I\ a\ I\ d\ I\ b$ then $\boxed{\begin{array}{c} a \to b \\ \nearrow \\ c \to d \end{array}}$ is a valid trace; on the other hand, $\boxed{\begin{array}{c} a \\ a \end{array}}$ cannot be generated (independence is irreflexive), and neither can $\boxed{\begin{array}{c} a \\ \searrow \\ b \to b \end{array}}$ (if $a\ I\ b$ then the a should not be ordered w.r.t. the second b, otherwise it should also be ordered w.r.t. the first b).

The independence relation is extended to pomsets by putting $p\ I\ q$ iff $\ell_p(v)\ I\ \ell_q(w)$ for all $v \in V_p$ and $w \in V_q$. The empty trace and single-element traces are modelled in the same manner as before; the partially commutative concatenation operation is defined by

$$p \odot q = [V_p \cup V_q, <_p \cup <_{pq} \cup <_q, \ell_p \cup \ell_q]$$

where p and q are disjoint representatives and $v <_{pq} w$ iff there exist $v' \in V_p$ and $w' \in V_q$ such that $v \leq_p v'$, $\neg(\ell_p(v')\ I\ \ell_q(w'))$ and $w' \leq_q w$. Note that the only difference w.r.t. ordinary string concatenation lies in the fact that essentially only the dependent vertices of p and q are ordered.

1.2 Example. If $c\ I\ a\ I\ d\ I\ b$ as above then $(a \odot c) \odot (b \odot d) = \boxed{\begin{array}{c} a \\ c \end{array}} \odot \boxed{\begin{array}{c} b \\ d \end{array}} = \boxed{\begin{array}{c} a \to b \\ \nearrow \\ c \to d \end{array}}$.

Series-parallel pomsets. Probably the most intensively studied approach to obtain a more extensive theory of pomsets is the direct combination of the algebras of strings and multisets, where the neutral elements of both are made to coincide. This leads to the theory of *series-parallel* or *N-free pomsets*, described in e.g. Aceto [2], Gischer [9], Grabowski [10], Jónsson [13], Pratt [19]. Series-parallel pomsets, therefore, are freely generated by the signature $\Sigma_{sp} = \langle \varepsilon, \cdot, \uplus \rangle$ with the following equations:

$$\begin{aligned}
\varepsilon \cdot x &= x && (13)\\
x \cdot \varepsilon &= x && (14)\\
(x \cdot y) \cdot z &= x \cdot (y \cdot z) && (15)\\
\varepsilon \uplus x &= x && (16)\\
(x \uplus y) \uplus z &= x \uplus (y \uplus z) && (17)\\
x \uplus y &= y \uplus x && (18)
\end{aligned}$$

It is seen that concatenation (*serial* composition) and disjoint union (*parallel* composition, hence *series-parallel*) do not interact at all. The models that can be generated using this signature are *N-free* in the sense that the figure N cannot occur as a substructure of the ordering relation: if $v <_p v' >_p w <_p w'$ for distinct $v, v', w, w' \in V_p$ then $v' < w'$ or $w < v$ or $v < w'$.

1.3 Example. $\boxed{\begin{array}{c} a \to b \\ \nearrow \\ c \to d \end{array}}$ is not N-free: it must be *augmented* at least to one of the following:

1. $(a \uplus c) \cdot b \cdot d = \boxed{\begin{array}{c} a \\ \searrow \\ c \to b \to d \end{array}}$;
2. $c \cdot ((a \cdot b) \uplus d) = \boxed{\begin{array}{c} c \to a \to b \\ \searrow \\ d \end{array}}$;
3. $(a \uplus c) \cdot (b \uplus d) = \boxed{\begin{array}{c} a \to b \\ \nearrow\!\!\searrow \\ c \to d \end{array}}$.

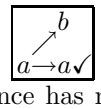
The empty and singleton pomsets are clearly N-free, and N-freedom is preserved by concatenation and disjoint union. It is less obvious that *all* N-free pomsets can be generated in the above algebra; see however any of the papers cited above.

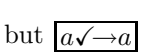
It should be mentioned that the theory of pomsets presented in the above papers, especially [13, 19, 9], extends far beyond this brief exposition. Apart from the fact that they also deal with *sets* of pomsets (in particular such sets as are closed under *augmentation*), a number of additional operators are defined, including one that allows *refinement* of elements into new pomsets, or in other words the *substitution* of a pomset for an element. (The latter is also dealt with in Nielsen, Engberg and Larsen [17].) Moreover, many results are obtained concerning pomsets that are *not* series-parallel. All the same, it is worthwhile to note that the characterisation of series-parallel pomsets as the initial model of the above algebra is the only *completeness* result mentioned.

Trees. Trees are pomsets with the special property that all predecessors of a given element are totally ordered. Algebraically this can be seen as an extension of multisets with an associative concatenation operator with respect to which the empty tree is left cancellative (rather than left and right neutral as for strings), and which distributes over addition from the right. We denote this operator by ';' to distinguish it from string concatenation. Hence, trees are freely generated by the signature $\Sigma_{tr} = \langle \varepsilon, ;, \uplus \rangle$ with the following equations:

$$\begin{aligned}
\varepsilon; x &= \varepsilon && (19)\\
(x; y); z &= x; (y; z) && (20)\\
\varepsilon \uplus x &= x && (21)\\
(x \uplus y) \uplus z &= x \uplus (y \uplus z) && (22)\\
x \uplus y &= y \uplus x && (23)\\
(x \uplus y); z &= x; z \uplus y; z && (24)
\end{aligned}$$

Note that Baeten and Weijland [3] present the above algebra with the additional axiom $x \uplus x = x$. Intuitively, concatenation of two trees p and q appends q to all the *termination points* of p, which are essentially its maximal elements —although some maximal elements may fail to be termination points; see below.

The pomsets generated by this system are *hierarchical orders with termination*, i.e. of the form $p = [V, <, \ell, \checkmark]$ such that $u < w > v$ implies $u \leq v$ or $v \leq u$ for all $u, v, w \in V$, and where the termination points are modelled by the extra component $\checkmark \subseteq \max_< V$. For instance, 

is a tree but 

and 

are not. The empty tree is modelled by $[\varnothing, \varnothing, \varnothing, \varnothing]$ (hence has no termination points) and single-element trees are modelled by $[\{0\}, \varnothing, \{(0, e)\}, \{0\}]$ (hence the single vertex is a termination point). Tree addition coincides with multiset addition (taking the union of the termination points); concatenation is defined by $p; q = [V, <, \ell, \checkmark]$ such that

$$\begin{aligned}
V &= V_p \cup (\checkmark_p \times V_q)\\
< &= <_p \cup \{(u, (v, w)) \mid u <_p v \in \checkmark_p, w \in V_q\} \cup \{((u, v), (u, w)) \mid u \in \checkmark_p, v <_q w\}\\
\ell &= \ell_p \cup \{((u, v), \ell_q(v)) \mid u \in \checkmark_p, v \in V_q\}\\
\checkmark &= \{(u, v) \mid u \in \checkmark_p, v \in \checkmark_q\}
\end{aligned}$$

It follows that appending the empty tree has the sole effect of removing all termination points (hence it is not right neutral w.r.t. ';'). Note that tree concatenation coincides with string concatenation if p is a nonempty terminated string.

1.4 Example.

1. $a;(b \uplus a);c = \boxed{\begin{array}{c} \nearrow b\checkmark \\ a{\to}a\checkmark \end{array}};\boxed{c\checkmark} = \boxed{\begin{array}{c} \nearrow b{\to}c\checkmark \\ a{\to}a{\to}c\checkmark \end{array}}$
2. $a;(b \uplus a;\varepsilon);b;(c;\varepsilon \uplus c) = \boxed{\begin{array}{c} \nearrow b\checkmark \\ a{\to}a \end{array}};\boxed{\begin{array}{c} \nearrow c \\ b{\to}c\checkmark \end{array}} = \boxed{\begin{array}{ccc} a{\to}b & & b{\to}c \\ & \searrow \quad \searrow & \\ & a \qquad c\checkmark & \end{array}}$

Note that such trees can easily be interpreted as *labelled transition systems* if one takes adjacent pairs of vertices as nodes and the vertices themselves as element-labelled transitions. This is in fact the usual interpretation of the above axioms in *process algebra*; see for instance Baeten and Weijland [3].

Deterministic pomsets. To enable a better comparison, we also show the algebra we present in this paper, without going into details at this point. Rather than changing the nature of concatenation, as in trees, we replace pomset union by a new operator called *join*. The resulting signature is given by $\Sigma_{det} = \langle \varepsilon, \cdot, \sqcup \rangle$ with the following equations:

$$\begin{aligned} \varepsilon{\cdot}x &= x && (25)\\ x{\cdot}\varepsilon &= x && (26)\\ (x{\cdot}y){\cdot}z &= x{\cdot}(y{\cdot}z) && (27)\\ \varepsilon \sqcup x &= x && (28)\\ (x \sqcup y) \sqcup z &= x \sqcup (y \sqcup z) && (29)\\ x \sqcup y &= y \sqcup x && (30)\\ x{\cdot}(y \sqcup z) &= (x{\cdot}y) \sqcup (x{\cdot}z) && (31) \end{aligned}$$

We call the pomsets generated by this algebra *deterministic*. This terminology is derived from common usage in the case of *trees* interpreted as labelled transition systems (see above): such a transition system is called deterministic if every transition is completely determined by its start node in combination with its label. Likewise, in deterministic pomsets, as we will see, every vertex is completely determined by its set of predecessors and its label.

The deterministic pomsets properly include all deterministic trees and all Mazurkiewicz traces (and therefore all posets) but not all series-parallel pomsets. Note that the only syntactical difference with the theory of series-parallel pomsets is that concatenation distributes over join from the left. As a special case, using (26), (28) and (31) it is straightforward to derive $x = x \sqcup x$, i.e., join is idempotent.

The remainder of the paper is structured as follows. In Section 2 we introduce the basic facts about deterministic pomsets by elaborating on the notion of *pomset prefix*. Especially, we show that the subclass of deterministic pomsets has a particularly nice structure with respect to the prefix ordering (it is a distributive lattice; see Section 2.3) and that, given a suitable notion of morphism, it sits within the category of pomsets in a special way (it forms a reflective subcategory; see Section 2.4).

Section 3 investigates the above equational theory for deterministic pomsets. It is proved sound and complete, and ω-complete in the presence of enough elements. In Section 4 we define a refinement operator for deterministic pomsets, analogous to the definition of refinement in series-parallel pomsets mentioned above. This comes down to introducing automorphisms over $\mathbf{POM}[\mathbf{E}]$ as objects in the algebra, and homomorphism application as a new operator. The corresponding extension of the equations is again proved (ω-)complete.

Concatenation is left-distributive but not right-distributive over join. This corresponds to the fact (already true for strings) that concatenation is prefix-monotonic in the right operand but not in the left. For some applications, one would prefer to have left-monotonicity as well. This can be achieved by introducing a notion of *termination* and changing concatenation into *sequential*

composition, much as in the case of trees. In fact, there are two possible interpretations of termination: one may regard it either as a *local* concept (different parts of a system may terminate independently) or as a *global* one (the system is either terminated as a whole or not terminated at all). Both interpretations, with corresponding algebras and (ω-complete) equational theories, are discussed in Section 5.

In Section 6, after a summary of the results, we come back to the comparison with some of the theories described above. We also give an overview of some possible ways to follow up on the results of the paper.

2 An investigation of pomset prefix

In this section we consider the prefix ordering over pomsets. After showing that this notion is not very well-behaved over arbitrary pomsets, we restrict ourselves to those pomsets over which it *is* well-behaved, and show that there it is very well-behaved indeed.

2.1 Prefix relations and morphisms

Recall that a binary relation R is *one-to-one* if $x\ R\ y$ and $x\ R\ z$ implies $y = z$, and *injective* if $x\ R\ z$ and $y\ R\ z$ implies $x = y$. The *domain* of R is defined as $\operatorname{dom} R = \{x \mid \exists y.\ x\ R\ y\}$ and the *codomain* as $\operatorname{cod} R = \{x \mid \exists y.\ y\ R\ x\}$. Now let $p, q \in \mathbf{LPO}[\mathbf{E}]$ be arbitrary.

2.1 Definition (prefix relations and morphisms).

1. A *prefix relation* between p and q is a one-to-one injective relation $R \subseteq V_p \times V_q$ such that both $\operatorname{dom} R \subseteq V_p$ and $\operatorname{cod} R \subseteq V_q$ are left-closed (according to $<_p$ resp. $<_q$) and for all $v, v' \in V_p$ and $w, w' \in V_q$, if $v\ R\ w$ and $v'\ R\ w'$ then $\ell_p(v) = \ell_q(w)$ and $v <_p v' \iff w <_q w'$. (In other words, a prefix relation is an isomorphism between left-closed segments of p and q.) Because of the latter property, there is an unambiguous extension of R to sets of vertices.
2. A *maximal prefix relation* between p and q is a prefix relation R such that for all $v \in V_p, w \in V_q$, if $\ell_p(v) = \ell_q(w)$ and $\{v' \in V_p \mid v' <_p v\}\ R\ \{w' \in V_q \mid w' <_q w\}$ then either $v \in \operatorname{dom} R$ or $w \in \operatorname{cod} R$.
3. A *prefix morphism* from p to q is a function $f\colon V_p \to V_q$ whose underlying relational graph $\{(v, f(v)) \mid v \in V_p\}$ is a (maximal) prefix relation. If there exists a prefix morphism from p to q then we say that p is a *prefix* of q, denoted $p \sqsubseteq q$.

2.2 Example.

- $\{(1,3)\}$ is a maximal prefix relation between $p = \boxed{{}_1a \to {}_2b}$ and $\boxed{{}_3a \to {}_4c}$, but just a (non-maximal) prefix relation between p and $\boxed{\begin{array}{c} \nearrow {}^4c \\ {}_3a \to {}_5b \end{array}}$ since it can be extended with $(2,5)$.
- $\boxed{{}_1a \to {}_2b} \not\sqsubseteq \boxed{\begin{array}{c} {}^3a \\ \searrow \\ {}_4c \to {}_5b \end{array}}$; in particular, $\{(1,3)\}$ is a prefix relation but not a prefix morphism since it is undefined on 2, whereas $R = \{(1,3),(2,5)\}$ is not a prefix relation since $\operatorname{cod} R = \{3,5\}$ is not left-closed.
- $\boxed{{}_1a \to {}_2b} \sqsubseteq \boxed{\begin{array}{c} \nearrow {}^4c \\ {}_3a \to {}_5b \end{array}}$ due to the prefix morphism $\{(1,3),(2,5)\}$.
- From $\boxed{{}_1a \to {}_2b}$ to $\boxed{\begin{array}{l} {}^3a \to {}^4c \\ {}_5a \to {}_6b \end{array}}$ there are two maximal prefix relations, $\{(1,3)\}$ and $\{(1,5),(2,6)\}$; only the latter is a prefix morphism.

- Maximal prefix relations are not closed under composition. For instance, $R = \{(1,3)\}$ is a maximal prefix relation between $\boxed{{}_1a \to {}_2b}$ and $p = \boxed{{}_3a \to {}_4c}$ and $S = \{(3,5),(4,7)\}$ between p and $\boxed{\begin{array}{l} {}^{\nearrow}{}^6b \\ {}_5a \to {}_7c \end{array}}$, but $R;S = \{(1,5)\}$ is not maximal since it can be extended with $(2,6)$.

Some facts about prefix relations and morphisms (straightforward to check) are collected in the following proposition.

2.3 Proposition (prefix relations and morphisms).

1. If the union of two prefix relations (between the same lposets) is injective and one-to-one, then it is a prefix relation;
2. Prefix relations and maximal prefix relations (but not prefix morphisms) are closed under inverse;
3. Prefix relations and prefix morphisms (but not maximal prefix relations) are closed under composition;
4. Every identity function id_{V_p} is a prefix morphism from p to p.

Remark. On the existence of maximal prefix relations: note that such relations are indeed maximal, in an order-theoretic sense, in the space of all prefix morphisms between a given pair of lposets (ordered by $\subseteq$). Since this space is necessarily finite (we deal only with finite lposets), it follows that every prefix relation is a subrelation of a maximal prefix relation. Furthermore, for arbitrary pairs of lposets, the empty relation is a prefix relation. It follows that there is at least one maximal prefix relation between every pair of lposets.

As a consequence of the fact that prefix morphisms are closed under composition, $\sqsubseteq$ is transitive; in fact it is a preorder over **LPO** that contains lposet isomorphism as its largest symmetric subrelation.

2.4 Proposition. $\sqsubseteq$ is a reflexive and transitive relation such that $p \sqsubseteq q \sqsubseteq p \iff p \cong q$.

It follows immediately that prefix is well-defined up to isomorphism and lifts to a partial order over pomsets: $[p] \sqsubseteq [q]$ iff $p \sqsubseteq q$. Also the number of maximal prefix relations is invariant under isomorphism, although on the level of pomsets, the prefix relations themselves are in general difficult to represent extensionally.

2.5 Example. There are two prefix morphisms from $\boxed{{}_1a}$ to $\boxed{\begin{array}{l} {}^2a \\ {}_3a \end{array}}$, viz. $\{(1,2)\}$ and $\{(1,3)\}$, but their difference cannot be seen on the level of pomsets; the same holds for $\boxed{\begin{array}{l} {}^1a \\ {}_2a \end{array}}$ and $\boxed{\begin{array}{l} {}^3a \\ {}_4a \to {}_5b \end{array}}$.

The prefix ordering as defined above in fact coincides with the standard notion of pomset prefix, according to which $[p] \sqsubseteq [q]$ if p is isomorphic to a left-closed fragment of q; indeed such a fragment is given by $f(p)$ where f is the prefix morphism. For the special case of strings, our definition of pomset prefix comes down to the usual notion of string prefix, as the following proposition shows.

2.6 Proposition. If p, q are total orders then $p \sqsubseteq q$ iff there is a p' such that $q \cong p \cdot p'$.

Proof sketch. First note that there is exactly one maximal prefix relation between every pair p, q of total orders. If $p \sqsubseteq q$ then apparently this is in fact a prefix morphism f. Now p' defined as that part of q not covered by $f(p)$ (or possibly an isomorphic variant to satisfy the disjointness condition of concatenation) satisfies $p \cdot p' \cong q$. □

When one further investigates the structure of the subclass of total orders under the prefix ordering, the following becomes apparent.

2.7 Proposition. Every nonempty set of total orders has an infimum with respect to $\sqsubseteq$.

This follows basically from the fact that the prefixes of a given total order are totally ordered under prefix; hence so are the common prefixes of a set of total orders; moreover this set of common prefixes is finite and nonempty (it contains at least the empty string), hence it has a greatest element.

In general, sets of pomsets fail to have infima. We have shown a counterexample in the introduction. One may therefore ask if the existence of infima expresses something particular about strings, or rather something that holds more generally but not as generally as the class of all pomsets. It turns out that the latter is the case. In fact, uniqueness of maximal prefix relations is sufficient to guarantee the existence of infima.

2.8 Lemma. If p_1 and p_2 have the property that between an arbitrary q and p_i $(i = 1, 2)$ there is exactly one maximal prefix relation, then p_1 and p_2 have a $\sqsubseteq$-infimum, which moreover also has this property.

Proof. Let R be the unique maximal prefix relation between p_1 and p_2, and define $p = p_1 \restriction \mathrm{dom}\, R$ (where restriction $p \restriction V$ is defined in the natural way). It follows that id_{V_p} is a prefix morphism from p to p_1, and R, taken as a function, is a prefix morphism from p to p_2; hence p is a lower bound of p_1 and p_2. Now assume that q is also a lower bound; let $f_i \colon V_q \to V_{p_i}$ be the unique prefix morphisms from q to p_i $(i = 1, 2)$. $f_1; R$ is then a prefix relation from q to p_2; hence $f_1; R \subseteq f_2$. Similarly, $f_2; R^{-1} \subseteq f_1$. It follows that $f_1 = f_2; R^{-1}$. The right hand side in fact describes a prefix relation between q and p; because its domain equals V_q this is a prefix morphism, hence $q \sqsubseteq p$. Finally, let f and g be two prefix morphisms from an arbitrary q to p; then they are also prefix morphisms from q to p_1, hence $f = g$. □

2.2 Deterministic pomsets

Lemma 2.8 suggests that it may be important to study the conditions for uniqueness of maximal prefix relations. A *maximal auto-prefix relation of* p will be a maximal prefix relation between p and itself. The identity relation over V_p is a trivial maximal auto-prefix relation; however, some lposets also have non-trivial maximal auto-prefix relations.

2.9 Example. $\boxed{\begin{matrix} {}_1a \\ {}_2a \to {}_3b \end{matrix}}$ has the nontrivial maximal auto-prefix relation $\{(1,2),(2,1)\}$.

We call an lposet *prefix unique* if it has no nontrivial maximal auto-prefix relations. Clearly, if we want to restrict ourselves to pomsets with unique maximal prefix relations, we must stay within the class of prefix unique lposets. The following lemma shows that we need no further restrictions.

2.10 Lemma. If p and q are prefix unique then there is exactly one maximal prefix relation between them.

Proof. Let R and S be maximal prefix relations between p and q. It follows that $R; S^{-1}$ is a prefix relation from p to p, hence gives rise to a maximal auto-prefix relation of p, which must equal id_{V_p}; hence $R \cup S$ is injective. On the other hand, also $R^{-1}; S \subseteq id_{V_q}$; hence $R \cup S$ is one-to-one. It follows that $R \cup S$ is a prefix relation; however, it cannot be larger than either R or S since those are maximal; therefore we may conclude that $R = S$ $(= R \cup S)$. □

Lemma 2.8 then gives rise to the following result.

2.11 Corollary. The class of prefix unique pomsets has $\sqsubseteq$-infima of nonempty sets.

(The existence of infima of infinite sets follows from the fact that the set of lower bounds of such an infinite set P is bound to be finite; we can then apply the independently proved Proposition 2.12 below, plus the general fact that the supremum of the set of lower bounds of P equals the infimum

of P.) In fact, from the proof of Lemma 2.8 it is clear that the infimum of p and q is defined as follows:

$$p \sqcap q \;:=\; p \restriction \mathrm{dom}\, R$$

where R is the unique maximal prefix relation between p and q.

We now have that the class of prefix unique pomsets generalises the strings in such a way that the existence of prefix infima is preserved. Moreover, it turns out that this class also has prefix *suprema*:

2.12 Proposition. The class of prefix unique pomsets has $\sqsubseteq$-suprema of finite sets.

Proof. The empty set has supremum ε, and the supremum of a singleton set is given by its element. We show the existence of suprema of pairs $p_i = \langle V_i, <_i, \ell_i \rangle$ $(i = 1, 2)$. Consider the lposet q such that

$$\begin{array}{rcl} V_q & = & ((V_1 \smallsetminus \mathrm{dom}\, R) \times \{*\}) \cup (\{*\} \times (V_2 \smallsetminus \mathrm{cod}\, R)) \cup R \\ <_q & = & \{((v, v'), (w, w')) \mid v <_1 w \vee v' <_2 w'\} \\ \ell_q & = & \{((v, v'), a) \mid \ell_1(v) = a \vee \ell_2(v') = a\} \end{array}$$

where R is the unique maximal prefix relation between p and q and $* \notin V_1 \cup V_2$ is an arbitrary vertex identifier.[1] For $i = 1, 2$ let π_i denote the *partial projections* from V_q to V_i; these are in fact maximal prefix relations, and the π_i^{-1} are prefix morphisms from p_i to q.

First we prove that q is prefix unique. Let S be a maximal auto-prefix relation of q; then $S; \pi_i$ is a prefix relation between q and p_i; hence $S; \pi_i \subseteq \pi_i$ for $i = 1, 2$, which implies $S \subseteq id_{V_q}$ since S is injective.

Now we prove that q is the $\sqsubseteq$-supremum of p and q. Because the π_i^{-1} are prefix morphisms, q is certainly a $\sqsubseteq$-upper bound. Now assume $p_i \sqsubseteq q'$ for $i = 1, 2$ where q' is prefix unique; let the relevant prefix morphisms be given by f_i. It follows that the $\pi_i; f_i$ are prefix relations between q and q' such that $\pi_1; f_1 \cup \pi_2; f_2$ is one-to-one and injective, hence $\pi_1; f_1 \cup \pi_2; f_2$ is a prefix morphism, proving $q \sqsubseteq q'$. □

In the remainder of this paper, we will essentially restrict ourselves to the prefix unique pomsets. We will in fact use a more explicit characterisation of prefix uniqueness. The *principal ideals* of an lposet p are sets $\downarrow_p v = \{w \in V_p \mid w \leq_p v\}$ for $v \in V_p$. We call v the *top* of $\downarrow_p v$ and $\Downarrow_p v = (\downarrow_p v) \smallsetminus \{v\}$ the *pre-set*. We omit the index p whenever this does not give rise to confusion.

2.13 Definition (determinism). An lposet $p \in \mathbf{LPO}$ is called *deterministic* if every vertex of p is completely determined by the combination of its pre-set and its label, i.e., if

$$\forall v, w \in V.\; \Downarrow v = \Downarrow w \wedge \ell(v) = \ell(w) \Longrightarrow v = w \;\; .$$

The class of deterministic lposets will be denoted $\mathbf{DLPO}[\mathbf{E}]$; we also use $\mathbf{DPOM}[\mathbf{E}] = \mathbf{DLPO}[\mathbf{E}]/\cong$ to denote the deterministic *pomsets*. The following proposition states that this property of determinism in fact precisely coincides with the uniqueness of auto-prefix morphisms.

2.14 Proposition. An lposet is deterministic iff it is prefix unique.

Proof: if Assume that $p \in \mathbf{POM}$ is not deterministic. Let $v, w \in V_p$ be such that $\Downarrow v = \Downarrow w$ and $\ell(v) = \ell(w)$ but $v \neq w$; then $R = \{(u, u) \mid u < v\} \cup \{(v, w)\}$ is a prefix relation from p to p, hence can be extended to a nontrivial maximal auto-prefix relation, which implies that p is not prefix unique.

[1] Those who are familiar with *event structures* will recognise the similarity of this construction to the *synchronisation* of two event structures; see e.g. Winskel [23], Boudol and Castellani [5].

only if Assume that $p \in \mathbf{POM}$ is not prefix unique. Let R be a nontrivial maximal auto-prefix relation, and let $S \subseteq R$ be the smallest prefix relation that is not a subrelation of id_V. It follows that there is a unique $(v, w) \in S$ such that $v \neq w$, hence $(\Downarrow v)\ S\ (\Downarrow w)$ implies $\Downarrow v = \Downarrow w$; moreover $\ell(v) = \ell(w)$. It follows that p is not deterministic. □

The empty pomset and all single-element pomsets are trivially deterministic, and concatenation preserves determinism; not so however disjoint union, since for instance $\boxed{a} \uplus \boxed{a} = \boxed{\begin{matrix} a \\ a \end{matrix}}$. Instead of disjoint union we will therefore use the *supremum* as defined in the proof of Proposition 2.12 as a constructor, which we will call *join* in the remainder of this paper. The join of deterministic pomsets can be formulated alternatively as a slight variation of disjoint union, where instead of taking disjoint representatives, *isomorphic common ideals are merged together*. Similarly, the *meet* of deterministic pomsets corresponds to the *intersection* of such representatives.

2.15 Example. $\boxed{\begin{matrix} & b \\ \nearrow & \\ a \!\!\rightarrow\!\! c & \end{matrix}}$ and $\boxed{\begin{matrix} a \!\!\rightarrow\!\! b \\ c \end{matrix}}$ have the isomorphic common ideal $\boxed{a \!\!\rightarrow\!\! b}$, whereas for instance the respective ideals $\boxed{a \!\!\rightarrow\!\! c}$ and $\boxed{c}$ are not isomorphic. Their join is given by $\boxed{\begin{matrix} a \!\!\rightarrow\!\! b \\ \searrow \\ c & c \end{matrix}}$ and their meet by $\boxed{a}$.

Formally, this is defined as follows. We call lposets p and q *compatible* if

$$\forall v \in V_p, w \in V_q.\ \Downarrow_p v = \Downarrow_q w \wedge \ell_p(v) = \ell_q(w) \Longrightarrow v = w\ .$$

Note that pairs of deterministic pomsets always have compatible representatives: for if p and q are *disjoint* representatives with maximal prefix relation R between them, then the lposet obtained from p by replacing the vertices in the domain of R by their R-images is isomorphic to p and compatible with q. On the other hand, if p and q are compatible then both are deterministic. Now the meet and join are characterised by

$$\begin{aligned} p \sqcap q &= [V_p \cap V_q, <_p \cap <_q, \ell_p \cap \ell_q] \\ p \sqcup q &= [V_p \cup V_q, <_p \cup <_q, \ell_p \cup \ell_q] \end{aligned}$$

where p and q are compatible. Hence the only difference between disjoint union and join is the choice of representatives.

2.16 Example.

1. $(a \cdot b) \uplus (a \cdot c) = \boxed{{}_1a \!\!\rightarrow\!\! {}_2b} \uplus \boxed{{}_3a \!\!\rightarrow\!\! {}_4c} = \boxed{\begin{matrix} {}_1a \!\!\rightarrow\!\! {}_2b \\ {}_3a \!\!\rightarrow\!\! {}_4c \end{matrix}} = \boxed{\begin{matrix} a \!\!\rightarrow\!\! b \\ a \!\!\rightarrow\!\! c \end{matrix}}$
2. $(a \cdot b) \sqcup (a \cdot c) = \boxed{{}_1a \!\!\rightarrow\!\! {}_2b} \sqcup \boxed{{}_1a \!\!\rightarrow\!\! {}_3c} = \boxed{\begin{matrix} & {}_2b \\ \nearrow & \\ {}_1a \!\!\rightarrow\!\! {}_3c & \end{matrix}} = \boxed{\begin{matrix} & b \\ \nearrow & \\ a \!\!\rightarrow\!\! c & \end{matrix}}$
3. $((a \sqcup c) \cdot b) \sqcup (c \cdot a) = \boxed{\begin{matrix} a & \\ \searrow & \\ c \!\!\rightarrow\!\! b & \end{matrix}} \sqcup \boxed{c \!\!\rightarrow\!\! a} = \boxed{\begin{matrix} a \!\!\rightarrow\!\! b \\ \nearrow \\ c \!\!\rightarrow\!\! a \end{matrix}}$.

The latter example shows that we can in fact construct N-shaped pomsets, which is not possible in the theory of series-parallel pomsets, as mentioned in the Introduction. As a final fact concerning the relation between disjoint union and join we mention the following:

2.17 Proposition. If $p, q \in \mathbf{DPOM}$ then $p \sqcup q = p \uplus q$ iff $p \sqcap q = \varepsilon$.

The following property lies at the heart of all completeness proofs in Sections 3–5, since it allows to decompose pomsets into (the join of) all its principal ideals, which themselves are constructed by appending a single element to an arbitrary pomset.

2.18 Proposition. If $p \in \mathbf{DPOM}$ then $p = \bigsqcup_{v \in V_p} (p \restriction \Downarrow v) \cdot \ell(v)$.

2.3 Prime algebraic bases

In this and the following subsection we discuss some additional properties of deterministic pomsets. First we discuss the structure of the class of deterministic pomsets under prefix; then we investigate, in a category theoretic setting, the manner in which the deterministic pomsets sit inside the full class of pomsets.

The characterisation above of prefix suprema and infima in terms of union and intersection immediately gives rise to the following distributivity property: for all $p_1, p_2, q \in \mathbf{DPOM}$

$$(p_1 \sqcap p_2) \sqcup q = (p_1 \sqcup q) \sqcap (p_2 \sqcup q) \ .$$

An ordered structure $\langle X, \sqsubseteq \rangle$ is called *distributive* if the above property is satisfied whenever the relevant infima and suprema exist. Moreover, we call an ordered structure $\langle X, \sqsubseteq \rangle$ a *basis* if it has all nonempty infima but no infinite suprema. (Note that the existence of nonempty infima implies *consistent completeness*, this being the property that all sets with an upper bound have a supremum; hence the absence of infinite suprema in a basis implies the absence of upper bounds of infinite sets, which in turn implies that no element of a basis may have an infinite number of predecessors.)[2] We then have the following strong order structure of the deterministic pomsets.

2.19 Corollary. $\langle \mathbf{DPOM}, \sqsubseteq \rangle$ is a distributive basis with all finite suprema.

Note that this property is *stronger* than the fact that $\langle \mathbf{DPOM}, \sqcup, \sqcap \rangle$ is a finitary distributive lattice (where finitariness is the property that compact elements have only finitely many predecessors — compactness of elements in turn being defined by the nonexistence of certain suprema, in particular infinite ones), since as remarked above, in a basis *all* elements have only finitely many predecessors. (Another way of stating this is that in a basis, all elements are compact.) A further consequence of Corollary 2.19 is that all prefix closed subclasses of $\langle \mathbf{DPOM}, \sqsubseteq \rangle$ form distributive bases, too, although these do not necessarily contain all finite suprema.

Distributivity of a basis can be characterised in quite a different way as well. A basis $\langle X, \sqsubseteq \rangle$ is called *prime algebraic* if for all $x \in X$,

$$x = \bigsqcup \{ y \sqsubseteq x \mid y \text{ is prime} \}$$

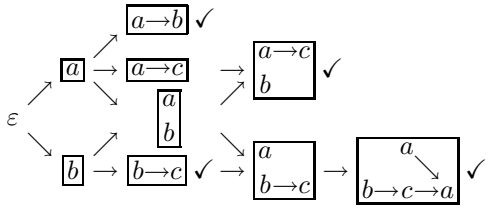
where $y \in X$ is called prime if for all consistent $Y \subseteq X$ (i.e., such that Y has an upper bound and hence a supremum), $y \sqsubseteq \bigsqcup Y$ implies $y \sqsubseteq z$ for some $z \in Y$. Prime algebraic bases play an important role in *partial order semantics*. For instance, Winskel has shown in [22] that every prime algebraic domain arises as the set of configurations of a prime event structure; Corradini et al. in [6] give a similar result for *safe parallel graph grammars*, which include all *safe Petri nets*. Now distributive bases are known to be exactly the same objects as prime algebraic bases (see e.g. [22]); therefore Corollary 2.19 is equivalent to the following.

2.20 Corollary. Every $\sqsubseteq$-left-closed subset of $\langle \mathbf{DPOM}, \sqsubseteq \rangle$ is a prime algebraic basis.

It follows that every set of deterministic pomsets P determines a prime algebraic basis given by its left-closure w.r.t. $\sqsubseteq$, with certain "terminated" elements corresponding to the members of P. This is analogous to the total order case, where every prefix-closed set of strings determines a (deterministic) tree ordered by string prefix, and every (arbitrary) set of strings L a tree with termination points corresponding to the elements of L. It is also not difficult to see that just as every deterministic tree arises in this way as a prefix closed set of strings, so every prime algebraic domain can be obtained as a prefix closed set of deterministic pomsets. For the restricted case of *posets* (i.e., pomsets with injective labelling functions) more details can be found in [20].

[2] In fact there is a one-to-one correspondence between bases in the above sense and *consistently complete partial orders* (ccpo's for short); the latter are obtained from the former by adding suprema of directed sets, whereas the inverse operation consists of omitting all elements with infinitely many predecessors; see [20] for details. We will henceforth ignore the difference between bases and ccpo's.

2.21 Example. The set of deterministic pomsets consisting of $\boxed{a{\to}b}$, $\boxed{\begin{matrix}a{\to}c\\ b\end{matrix}}$, $\boxed{b{\to}c}$ and $\boxed{\begin{matrix}a\\ \searrow\\ b{\to}c{\to}a\end{matrix}}$ gives rise to the following prime algebraic basis (where terminated elements are marked ✓):



2.4 Determinisation

When considering the class of pomsets and the subclass of deterministic pomsets, a natural question is whether anything can be said about the nature of this subclass, and about the relation (if any) between the pomsets outside to those inside the subclass. To make the question precise and provide an answer to it, we make a brief excursion to the field of *category theory*. For the duration of this excursion we once more view our objects as lposets rather than pomsets.

It turns out that under an appropriate notion of *morphism*, one may characterise the deterministic lposets as a *reflective subcategory* of the lposets.

2.22 Definition (determinising morphisms). Let $p, q \in \mathbf{LPO}$. A *determinising morphism* from p to q is a function $f\colon V_p \to V_q$ that preserves labelling and ordering and is *image left-closed* in the following sense: if $v <_q f(w)$ then $v = f(u)$ for some $u <_p w$.

The typical effect of a determinising morphism is to merge vertices with the same predecessors and the same label, i.e., precisely such vertices as should coincide in deterministic lposets, according to Definition 2.13.

2.23 Example. From $p = \boxed{\begin{matrix}{}_1a{\to}{}_2b\\ {}_3a{\to}{}_4c\end{matrix}}$ to $q = \boxed{\begin{matrix}{}_5a{\to}{}_6b\\ {}_8a\ \searrow\ {}_7c\end{matrix}}$ there is s single determinising morphism, viz. $\{(1,5),(2,6),(3,5),(4,7)\}$. Note that there is no prefix morphism from p to q.

The following facts are straightforward to establish:

2.24 Proposition (determinising morphisms).

1. Prefix morphisms are determinising morphisms, but not necessarily vice versa.
2. There is at most one determinising morphism from a given lposet *to* any deterministic lposet.
3. Every determinising morphism *from* a deterministic lposet is a prefix morphism.
4. Every identity function on vertices is a determinising morphism.
5. Determinising morphisms are closed under composition.

From the latter two facts it follows that determinising morphisms give rise to a category of lposets (where isomorphism corresponds to standard lposet isomorphism); moreover, in the full subcategory of deterministic lposets, the morphisms coincide with prefix morphisms. This subcategory is in fact a preorder category (at most one morphism between every pair of objects); hence meets and joins are products and coproducts, respectively.

2.25 Theorem. The lposets with determinising morphisms forms a category $\mathbf{LPO}_{det}$ with full subcategory $\mathbf{DLPO}_{det} = \mathbf{DLPO}$ (where the latter has prefix morphisms).

Now from an arbitrary lposet p we can construct a deterministic lposet Dp by collapsing all isomorphic prefixes of p, as follows: let $\sim_p \subseteq V_p \times V_p$ be the largest label and prefix preserving equivalence relation in V_p, i.e. such that if $v \sim_p w$ then $\ell_p(v) = \ell_p(w)$ and for all $v' <_p v$ there is a $w' <_p w$ such that $v' \sim_p w'$. Such a largest equivalence exists because the identity relation is a label and prefix preserving equivalence, and label and prefix preservation are preserved by union and transitive closure. (Note the analogy of $\sim_p$ to *bisimilarity*, which is an equivalence over transition systems —cf. e.g. Milner [16]. This is not coincidental: lposets can be seen as finite labelled transition systems in such a way that isomorphism of deterministic lposets is fully abstract w.r.t. bisimilarity.) Now for Dp take $V_p/\!\sim_p$ as a new vertex set, with the ordering and labelling induced from p; hence such that

$$\begin{aligned} V <_{Dp} W \quad &:\Leftrightarrow \quad \exists v \in V, w \in W.\, v <_p w \\ \ell_{Dp}(V) = a \quad &:\Leftrightarrow \quad \exists v \in V.\, \ell_p(v) = a \end{aligned}$$

where $V, W \in V_{Dp} = V_p/\!\sim_q$. It should be clear that Dp is indeed deterministic. In fact, since $\sim_p = id_{V_p}$ if p is deterministic already, it follows that in that case $Dp \cong p$. Furthermore, from an arbitrary determinising morphism f from p to q we can define a prefix morphism Df from Dp to Dq (which is therefore in fact determinising) as follows: for all $v \in V_p$,

$$Df\colon [v]_{\sim_p} \mapsto [f(v)]_{\sim_q} \ .$$

It follows that D is left adjoint to the inclusion functor $U\colon \mathbf{DLPO} \hookrightarrow \mathbf{LPO}_{det}$; the existence of such a left adjoint is called *reflectivity* of the subcategory.

2.26 Theorem. **DLPO** is a reflective subcategory of $\mathbf{LPO}_{det}$.

Proof. We have to show that for all lposets $p \in \mathbf{LPO}$ and $q \in \mathbf{DLPO}$ there are as many prefix morphisms from Dp to q as there are determinising morphisms from p to Uq, i.e., from p to q. We have already remarked above that there is at most one determinising morphism to any deterministic lposet; hence we have to show that $Dp \sqsubseteq q$ iff there is a determinising morphism from p to q. Since $Dq \cong q$, any determinising morphism f from p to q gives rise to a prefix morphism Df from Dp to Dq, hence $Dp \sqsubseteq q$. On the other hand, if f is a prefix morphism from Dp to q then $g\colon V_p \to V_q$ defined by

$$g\colon v \mapsto f([v]_{\sim_p})$$

is a determinising morphism from Dp to q. □

Among other things, it is known that right adjoints preserve colimits, in particular coproducts. It follows that $\mathbf{LPO}_{det}$ has coproducts, and indeed for arbitrary pomsets p and q, $p \uplus q$ with identity injections id_{V_p} and id_{V_q} is the coproduct of p and q in $\mathbf{LPO}_{det}$ (but not in **LPO** with prefix morphisms, as we have seen).

The object part of the functor D also preserves the A_{sp}*-structure* of $\mathbf{LPO}_{det}$ modulo isomorphism, i.e. the structure induced by the signature $\Sigma_{sp} = \langle \varepsilon, \cdot, \uplus \rangle$ and the corresponding equations. To be precise, ε and $\cdot$ are mapped to themselves whereas $\uplus$ is mapped to $\sqcup$, hence

$$D(p \uplus q) = Dp \sqcup Dq \ .$$

Note that the equations of A_{sp} automatically remain valid under this mapping, since joins are commutative and associative, and ε is a neutral element w.r.t. join. This property is formulated in the following theorem.

2.27 Theorem. The object part of D is an A_{sp}-homomorphism from **POM** to **DPOM**, where disjoint union in **POM** is carried over to join in **DPOM**.

3 An equational theory of deterministic pomsets

We have seen that deterministic pomsets arise rather naturally from an attempt to preserve the properties of string prefix in the more general class of pomsets. The investigation so far has been based solely on the *models* we have defined for strings and pomsets. However, it is well known that strings can be characterised *algebraically*: they are the free model generated by **E** in the algebra of *monoids*. That is, if we take the signature $\Sigma_{str} = \langle \varepsilon, \cdot \rangle$ with the equations

$$\begin{aligned} \varepsilon{\cdot}x &= x & (1)\\ x{\cdot}\varepsilon &= x & (2)\\ (x{\cdot}y){\cdot}z &= x{\cdot}(y{\cdot}z) & (3) \end{aligned}$$

(see also the Introduction), then the class of strings is isomorphic to $T_{str}(\mathbf{E})/\simeq$, where $T_{str}(\mathbf{E})$ is the set of terms obtained by applying the operators of Σ_{str} to the elements of **E**, and $\simeq \subseteq T_{str}(\mathbf{E}) \times T_{str}(\mathbf{E})$ is "provable equality," i.e. the equivalence generated by the equations above plus the rules of reflexivity, symmetry, transitivity, instantiation and congruence.

Now let us regard once more the standard definition of string prefix:

$$x \sqsubseteq y \;:\Leftrightarrow\; \exists z.\, y = x{\cdot}z \;\;.$$

Using the equations above it can be deduced, besides the fact that $\sqsubseteq$ is a partial ordering relation with smallest element ε, that string concatenation is monotonic in its right operand: for if $y \sqsubseteq z$ then $z = y{\cdot}y'$ for some y', hence $x{\cdot}z = x{\cdot}(y{\cdot}y') = (x{\cdot}y){\cdot}y'$, implying $x{\cdot}y \sqsubseteq x{\cdot}z$. (However, concatenation is not monotonic in its *left* operand, as is apparent from $a \sqsubseteq a{\cdot}b = ab$ but $ac \not\sqsubseteq abc$.)

As a next step, we can *algebraise* the prefix ordering by introducing a join-like operator, which for the moment is only partial; in other words, we let

$$x \sqcup y \;:=\; \begin{cases} x & \text{if } y \sqsubseteq x \\ y & \text{if } x \sqsubseteq y \\ \text{undefined} & \text{otherwise.} \end{cases}$$

In addition to the equations we already had, we then can express various properties of the prefix ordering equationally:

Reflexivity: $x \sqcup x = x$.

Symmetry: $x \sqcup y = y \sqcup x$.

Transitivity: $(x \sqcup y) \sqcup z = x \sqcup (y \sqcup z)$.

Smallest element: $\varepsilon \sqcup x = x$.

Right-monotonicity: $x{\cdot}(y \sqcup z) = (x{\cdot}y) \sqcup (x{\cdot}z)$.

These equations should be understood as follows: for all valuations of x, y, z, either both sides are undefined, or both are defined and provably equal. (Note, by the way, that the reflexivity equation can be proved from the others.)

3.1 The algebra A_{det}

To obtain a theory of deterministic pomsets, all one has to do now is turn the join operator into a constructor of the algebra rather than a derived notion. This implies that join is now totally defined, i.e., we have to add objects to represent the joins that were heretofore undefined. Of course, these "new" objects are exactly those where there are unordered elements. We obtain a signature $\Sigma_{det} = \langle \varepsilon, \cdot, \sqcup \rangle$ with equations

$$\varepsilon{\cdot}x = x \tag{25}$$
$$x{\cdot}\varepsilon = x \tag{26}$$
$$(x{\cdot}y){\cdot}z = x{\cdot}(y{\cdot}z) \tag{27}$$
$$\varepsilon \sqcup x = x \tag{28}$$
$$(x \sqcup y) \sqcup z = x \sqcup (y \sqcup z) \tag{29}$$
$$x \sqcup y = y \sqcup x \tag{30}$$
$$x{\cdot}(y \sqcup z) = (x{\cdot}y) \sqcup (x{\cdot}z) \tag{31}$$

This is the theory that we already announced in the introduction, and indeed we have the following characterisation.

3.1 Theorem. $\mathbf{DPOM}[\mathbf{E}]$ is the free A_{det}-model generated by $\mathbf{E}$.

Proving this involves showing that $\mathbf{DPOM}[\mathbf{E}]$ is closed under the intended interpretation of ε, $\cdot$ and $\sqcup$ and the equations hold under this interpretation (*soundness*), that every object in $\mathbf{DPOM}[\mathbf{E}]$ can be denoted using a term of the algebra (*no junk*), and that terms denoting the same object are provably equal (*no confusion*). The latter two properties together are also known as *completeness* of the theory, and we will in fact prove a slightly stronger version of it.

It is now important to distinguish carefully between objects and their denotations: the former correspond to pomsets, the latter to terms of the signature Σ_{det}. $T_{det}(\mathbf{E}, \mathbf{X})$, ranged over by s, t, will denote the set of Σ_{det}-terms on generators $\mathbf{E}$ and variables $\mathbf{X}$, and $T_{det}(\mathbf{E})$ the corresponding set of *ground terms*, i.e. terms without variables. We will drop the parameter $\mathbf{E}$ when this does not give rise to confusion. A *substitution* is a function $\rho\colon \mathbf{X} \to T_{det}(\mathbf{X})$ mapping variables to terms; ρ is called *ground* if its images are ground terms. Substitutions inductively give rise to functions $\rho\colon T_{det}(\mathbf{X}) \to T_{det}(\mathbf{X})$ (note the overloading of the symbol ρ); applications of the latter are postfix denoted, e.g. $t\rho$. The *semantics* of terms, i.e. the corresponding pomsets, are returned by a function $[\![\_]\!]\colon T_{det} \to \mathbf{DPOM}$ defined inductively on the structure of ground terms.

Theorem 3.1 is equivalent to Theorems 3.2, 3.3 and 3.5 below. The first of these states that the semantic function is well-behaved in that it maps to the intended class of models (the deterministic pomsets) and preserves provable equality as pomset equality (=lposet isomorphism); in other words, that $\mathbf{DPOM}$ is indeed a model of A_{det}.

3.2 Theorem (A_{det} is sound). For arbitrary $s, t \in T_{det}$, $[\![t]\!] \in \mathbf{DPOM}$, and $\mathrm{A}_{det} \vdash s = t$ implies $[\![s]\!] = [\![t]\!]$.

Next, we state that all the objects of our model can be denoted. For the proof, the following meta-notation is convenient: $\bigsqcup T$ for finite sets $T \subseteq T_{det}$ stands for the join of all $t \in T$, where $\bigsqcup \varnothing = \varepsilon$ and $\bigsqcup \{t\} = t$. We also use $\bigsqcup_{i \in I} t_i$ where I is an index set such that $t_i = t_j$ implies $i = j$, corresponding to $\bigsqcup \{t_i \mid i \in I\}$. This meta-notation is well-defined up to provable equality of terms, due to the fact that $\sqcup$ is commutative, associative and idempotent with identity ε (Equations (28)–(31)). Now we recursively define a function $R\colon \mathbf{DPOM} \to \mathit{Fin}(T_{det})$ (the latter denoting the set of *finite* subsets of T_{det}) as follows:

$$R(p) := \{(\textstyle\bigsqcup R(p \restriction (\Downarrow_p v))){\cdot}\ell_p(v) \mid v \in V_p\} \ .$$

Hence p is decomposed into all prefixes with a unique top element, and R is recursively applied to the predecessors of those prefixes. This can be shown to be well-defined by induction on the size of p. The following theorem then states that this yields a denotation for all deterministic pomsets. It can be proved by induction on the size of p, using the fact that $p = \bigsqcup_{v \in V_p} (p \restriction \Downarrow v){\cdot}\ell(v)$ for all deterministic pomsets p (Proposition 2.18).

3.3 Theorem (no junk). For all $p \in \mathbf{DPOM}$, $p = [\![\bigsqcup R(p)]\!]$.

3.4 Example. The R-constructed denotation for $\boxed{\begin{array}{ccc} a & \to & c \\ & \nearrow & \\ b & \to & a \end{array}}$ is $\varepsilon\cdot a \sqcup \varepsilon\cdot b \sqcup (\varepsilon\cdot a \sqcup \varepsilon\cdot b)\cdot c \sqcup (\varepsilon\cdot b)\cdot a$, or in meta-notation, $\bigsqcup\{(\bigsqcup\varnothing)\cdot a, (\bigsqcup\varnothing)\cdot b, (\bigsqcup\{(\bigsqcup\varnothing)\cdot a, (\bigsqcup\varnothing)\cdot b\})\cdot c, (\bigsqcup\{(\bigsqcup\varnothing)\cdot b\})\cdot a\}$. A simpler denotation for the same pomsets is e.g. $(a \sqcup b)\cdot c \sqcup b\cdot a$.

Finally, we show that our equational theory is strong enough to prove all equalities that hold in the model; in other words, that terms which according to the theory describe distinct objects are not mapped to the same objects. Yet another way to express this is by saying that denotations of objects are unique up to provable equality.

3.5 Theorem (no confusion). For all $s, t \in T_{det}$, if $[\![s]\!] = [\![t]\!]$ then $\mathrm{A}_{det} \vdash s = t$.

As usual, this theorem is proved by rewriting terms to *normal forms*. In the course of this paper we will in fact encounter several different kinds of normal forms; this is the first and simplest.

3.6 Definition (normal forms). Consider the following production rule for terms in $T_{det}(\mathbf{E})$:

$$N ::= (\bigsqcup \textbf{saturated set of } N)\cdot e$$

where $e \in \mathbf{E}$ and a set T of N-produced terms is *saturated* if $T' \subseteq T$ for all $(\bigsqcup T')\cdot e' \in T$. A term is in *normal form* if it equals $\bigsqcup T$ for some saturated set T of N-produced terms.

Notation. For the sake of readability, we will in practice not write (pre-)normal forms using the meta-notation $\bigsqcup T$, but rather write ε, t and $t_1 \sqcup \cdots \sqcup t_n$ for (respectively) $\bigsqcup\varnothing$, $\bigsqcup\{t\}$ and $\bigsqcup\{t_1, \ldots, t_n\}$.

The closure condition is used to guarantee uniqueness of normal forms, as the following example shows.

3.7 Example. $\boxed{\begin{array}{ccc} & & b \\ & \nearrow & \\ a & \to & c \end{array}}$ is generated by $\bigsqcup T_1$ and $\bigsqcup T_2$ where T_1 and T_2 are sets of N-produced terms:

$$\begin{aligned} T_1 &= \{(\varepsilon\cdot a)\cdot b, (\varepsilon\cdot a)\cdot c\} \\ T_2 &= T_1 \cup \{\varepsilon\cdot a\} \ . \end{aligned}$$

T_1 is not saturated since $(\varepsilon\cdot a)\cdot b \in T_1$ but $\varepsilon\cdot a \notin T_1$. This is remedied in T_2, and indeed T_2 is saturated and $\bigsqcup T_2$ is in normal form.

The function R used in the proof of Theorem 3.3 in fact yields normal forms; moreover, on normal forms R is the left inverse of the semantic mapping. It follows that there is at most one normal form term describing a given pomsets; in other words, *normal forms are unique*. This is stated in the following lemma.

3.8 Lemma (normal forms are unique). $\bigsqcup R([\![t]\!]) = t$ for all normal form terms t.

Proof. By induction on the structure of normal forms. First note that if $s = s'\cdot e$ then $[\![s]\!]$ has a unique greatest vertex v with $\ell_{[\![s]\!]}(v) = e$ and $[\![s]\!] \upharpoonright (\Downarrow v) = [\![s']\!]$. Furthermore, if T is a saturated set of N-produced terms then there is a one-to-one correspondence between the elements of T and the vertices of $p = [\![\bigsqcup T]\!]$, i.e., for all $v \in V_p$ there is exactly one $s \in T$ with $[\![s]\!] = p \upharpoonright (\downarrow_p v) = (p \upharpoonright \Downarrow_p v)\cdot \ell_p(v)$.

Assume that T is a saturated set of N-produced terms such that $R([\![\bigsqcup T]\!]) = T$ and $R([\![\bigsqcup T_s]\!]) = T_s$ for all $s = (\bigsqcup T_s)\cdot e_s \in T$, and consider the N-produced term $t = (\bigsqcup T)\cdot e$. It follows that

$$\begin{aligned} R([\![t]\!]) &= \{(\bigsqcup R([\![t]\!] \upharpoonright \Downarrow v))\cdot \ell(v) \mid v \in V_{[\![t]\!]}\} \\ &= \{(\bigsqcup R([\![\bigsqcup T]\!] \upharpoonright \Downarrow v))\cdot \ell(v) \mid v \in V_{[\![\sqcup T]\!]}\} \cup \{(\bigsqcup R([\![\bigsqcup T]\!]))\cdot e\} \\ &= \{(\bigsqcup R([\![\bigsqcup T_s]\!]))\cdot e_s \mid s \in T\} \cup \{t\} \\ &= \{(\bigsqcup T_s)\cdot e_s \mid s \in T\} \cup \{t\} \\ &= T \cup \{t\} \ . \end{aligned}$$

Now let $\bigsqcup T$ be a normal form term such that $R([\![\bigsqcup T_s]\!]) = T_s$ for all $s \in T$; since $T_s \subseteq T$ (T is saturated), it follows that

$$R([\![\bigsqcup T]\!]) = \bigcup_{s\in T} R([\![s]\!]) = \bigcup_{s\in T} T_s \cup \{s\} = T \ .$$

This proves the lemma. □

It follows that syntactically different normal form terms yield different pomsets, which is one of the two crucial properties of normal forms. The second crucial property is that every term can be rewritten up to provable equality to a normal form term. To see that this holds, consider the following inductively defined algorithm:

$$\begin{aligned} \mathit{norm}(\varepsilon) &:= \varnothing \\ \mathit{norm}(e) &:= \{\varepsilon\cdot e\} \\ \mathit{norm}(s\cdot t) &:= \mathit{norm}(s) \cup \{(\bigsqcup \mathit{norm}(s\cdot t'))\cdot e \mid t'\cdot e \in \mathit{norm}(t)\} \\ \mathit{norm}(s \sqcup t) &:= \mathit{norm}(s) \cup \mathit{norm}(t) \ . \end{aligned}$$

It can be proved by induction on the term structure that for all $t \in T_{det}(\mathbf{E})$, $\mathit{norm}(t)$ yields a finite saturated set of N-produced terms whose join is provably equal to t. Hence every term can be rewritten to a normal form term up to provable equality. This is stated in the following lemma.

3.9 Lemma (normal forms exist). For all terms $t \in T_{det}$, $\bigsqcup \mathit{norm}(t)$ is a normal form such that $\mathrm{A}_{det} \vdash t = \bigsqcup \mathit{norm}(t)$.

Proof sketch. Both the fact that $\mathit{norm}(t)$ is a saturated set of N-produced terms and the fact that $\mathrm{A}_{det} \vdash t = \bigsqcup \mathit{norm}(t)$ are proved by induction on the terms structure of t, by referring to the definition of *norm*. We just show the (most interesting) case of concatenation. Assume that the lemma holds for s and t and for all $s\cdot t'$ where $t' \sqsubset t$, and regard the term $s\cdot t$. The elements of $\mathit{norm}(s\cdot t)$ are by construction N-produced terms. To see that $\mathit{norm}(s\cdot t)$ itself is saturated, consider $(\bigsqcup T')\cdot e' \in \mathit{norm}(s\cdot t)$; then by construction of $\mathit{norm}(s\cdot t)$, one of the following cases holds.

- $(\bigsqcup T')\cdot e' \in \mathit{norm}(s)$, in which case $T' \subseteq \mathit{norm}(s)$ due to the closure of $\mathit{norm}(s)$;
- $T' = \mathit{norm}(s\cdot t')$ where $t'\cdot e' \in \mathit{norm}(t)$, meaning that for all $t_1 \in T'$, either $t_1 \in \mathit{norm}(s)$ and hence $t_1 \in \mathit{norm}(s\cdot t)$, or $t_1 = (\bigsqcup \mathit{norm}(s\cdot t_2))\cdot e_2$ where $t_2\cdot e_2 \in \mathit{norm}(t')$; but then also $t_2\cdot e_2 \in \mathit{norm}(t)$ and hence $t_1 \in \mathit{norm}(s\cdot t)$.

Finally, we prove that the *norm*-rule for $s\cdot t$ preserves provable equality.

$$\begin{aligned} \bigsqcup \mathit{norm}(s\cdot t) &= \bigsqcup\big(\mathit{norm}(s) \cup \{(\bigsqcup \mathit{norm}(s\cdot t'))\cdot e \mid t'\cdot e \in \mathit{norm}(t)\}\big) \\ &= \bigsqcup \mathit{norm}(s) \sqcup \bigsqcup_{t'\cdot e\in \mathit{norm}(t)} (\bigsqcup \mathit{norm}(s\cdot t'))\cdot e \\ &= s \sqcup \bigsqcup_{t'\cdot e\in \mathit{norm}(t)} (s\cdot t')\cdot e \end{aligned}$$

$$
\begin{aligned}
&= s\cdot \bigsqcup_{t'\cdot e\in norm(t)} (\varepsilon \sqcup t'\cdot e)\\
&= s\cdot \bigsqcup norm(t)\\
&= s\cdot t \ .
\end{aligned}
$$

This concludes the proof of this case. The other cases are analogous. □

Proof of Theorem 3.5. If $[\![s]\!] = [\![t]\!]$ for two terms $s, t \in T_{det}$ then by applying Lemma 3.9 and Lemma 3.8 we can prove

$$\mathrm{A}_{det} \vdash s = \bigsqcup norm(s) = \bigsqcup R([\![s]\!]) = \bigsqcup R([\![t]\!]) = \bigsqcup norm(t) = t \ .$$

This concludes the completeness proof. □

3.2 ω-completeness of A_{det}

If there are enough elements around (**E** is large enough) then not only the above completeness property holds, but one which is even stronger. Whereas Theorem 3.5 expresses that A_{det} is *complete for ground terms*, a more interesting notion is *completeness for open terms*. This is the property that if two terms denote the same object *under arbitrary ground substitutions* then they are are provably equal *before substitution*. This is also called *inductive completeness* (because it implies that all theorems that can be proved by induction on the structure of terms can also be proved equationally) or *ω-completeness*. See e.g. [11, 12, 14] for a general discussion.

3.10 Theorem (A_{det} is ω-complete). Assume $|\mathbf{E}| = \omega$. For all $s, t \in T_{det}(\mathbf{E}, \mathbf{X})$, if $[\![s\rho]\!] = [\![t\rho]\!]$ for all ground substitutions $\rho\colon \mathbf{X} \to T_{det}(\mathbf{E})$ then $\mathrm{A}_{det} \vdash s = t$.

The side condition $|\mathbf{E}| = \omega$ is needed to ensure that for any pair of terms $s, t \in T_{det}(\mathbf{E}, \mathbf{X})$ there are enough "unused elements," i.e. not occurring in s or t, to "encode" the free variables of t.

3.11 Example. If $|\mathbf{E}| = 1$ then A_{det} is not ω-complete. The deterministic pomsets over a one-element set are in fact totally ordered; hence they are isomorphic to the natural numbers (by mapping p to $|V_p|$), where pomset concatenation corresponds to addition and pomset join to taking the maximum. It follows that the following additional equalities hold under all ground substitutions:

$$
\begin{aligned}
x\cdot y &= y\cdot x\\
(x \sqcup y)\cdot z &= (x\cdot y) \sqcup (x\cdot z)
\end{aligned}
$$

However, these equations are not provable in A_{det} (and indeed do not hold in general), hence we do not have ω-completeness.

To prove ω-completeness, two general techniques can be found in the literature. One technique, proposed by Groote [11], is to construct for any pair of open terms $s, t \in T_{det}(\mathbf{X})$ a "characteristic" ground substitution $\rho_{s,t}$ with the property that $\mathrm{A}_{det} \vdash s\rho_{s,t} = t\rho_{s,t}$ if and only if $\mathrm{A}_{det} \vdash s = t$. Clearly, if such characteristic substitutions exist then ω-completeness reduces to ordinary (ground) completeness.

A more specialised variant of this technique, described by Heering in [12] and by Lazrek, Lescanne and Thiel in [14], is to use *normal forms* once more, in particular *open* normal forms, with the following properties:

- for any open term there is a normal form that is provably equal to it;
- for any pair of different normal forms there is a ground substitution that maps them to (closed) terms denoting different objects.

The difference with the first proof idea is that the substitutions required by the latter property, which correspond to the characteristic substitutions of Groote's, are only applied to normal forms, which makes their characteristicness a good deal easier to prove. This advantage is offset by the need to define an appropriate normal form in the first place.

3.12 Definition (open normal forms). Consider the following grammar for terms of $T_{det}(\mathbf{E}, \mathbf{X})$:

$$N ::= (\bigsqcup \textbf{sat'd set of } N)\cdot e \mid (\bigsqcup \textbf{sat'd set of } N)\cdot x$$

where $e \in \mathbf{E}$, $x \in \mathbf{X}$ and a set T of N-produced terms is *saturated* if $T' \subseteq T$ for all $(\bigsqcup T')\cdot t' \in T$. A term is in *open normal form* if it equals $\bigsqcup T$ for some saturated set T of N-produced terms.

This format is a simple variation on Definition 3.6 in which variables x are treated in the exact same way as elements e. Since all our equations allow variables to be handled in the same way as elements (there are no special equations for elements), the first step of the ω-completeness proof (every open term has a provably equal open normal form) is immediate. The characteristic substitution required in the second step (for every pair of different open normal forms there is a characteristic substitution mapping them onto different ground terms) is also easy: every variable is mapped to a distinct new element not yet occurring in the normal forms being compared.

Proof sketch of Theorem 3.10. For every open term t there is an open normal form term t' such that $\mathrm{A}_{det} \vdash t = t'$. The proof is analogous to that of Theorem 3.5.

Now let s, t be syntactically different open normal forms, and let $E_{s,t} \subseteq \mathbf{E}$ be the set of elements that occur syntactically in s or t. For all $x \in \mathbf{X}$ let $e_x \in \mathbf{E} \setminus E_{s,t}$ be a distinct element (note that since $E_{s,t}$ is certainly finite, the cardinality of $\mathbf{E}$ guarantees that there are enough such e_x), and define $\rho_{s,t} \colon x \mapsto e_x$ for all $x \in \mathbf{X}$. Then $s\rho_{s,t}$ and $t\rho_{s,t}$ are two syntactically different (ground) normal forms, hence $[\![s\rho_{s,t}]\!] \neq [\![t\rho_{s,t}]\!]$. Hence $\mathrm{A}_{det} \vdash s = t$ for open normal forms iff $s = t$ (syntactically).

Now if s, t are arbitrary open terms such that $[\![s\rho]\!] = [\![t\rho]\!]$ for all ground substitutions ρ, and s' and t' are corresponding open normal forms (i.e., $\mathrm{A}_{det} \vdash s = s', t = t'$), then also $[\![s'\rho_{s',t'}]\!] = [\![t'\rho_{s',t'}]\!]$ for the specific characteristic ground substitution $\rho_{s',t'}$ and hence $s' = t'$; it follows that $\mathrm{A}_{det} \vdash s = t$. □

4 Refinement of pomsets

In this and the next section we discuss extensions of the algebra of pomsets presented above. In this section we will be looking at *refinement*, which is the principle of replacing the elements of a pomset by entire pomsets. After discussing in detail the relation between refinement and *homomorphism application*, we proceed to introduce it as an operator in the algebra of deterministic pomsets. For the extended algebra we once more give an ω-complete equational theory.

4.1 Homomorphisms and refinement

Let us consider A_{det}-homomorphisms from **DPOM** to itself, i.e. functions h mapping deterministic pomsets to deterministic pomsets while preserving the operations of Σ_{det}. This preservation comes down to the following equations:

$$\begin{aligned} h(\varepsilon) &= \varepsilon \\ h(p\cdot q) &= h(p)\cdot h(q) \\ h(p \sqcup q) &= h(p) \sqcup h(q) \end{aligned}$$

Because in **DPOM** there is no junk, h is completely determined by its action on the generators $\mathbf{E}$, i.e. by the images $h(e)$ for all $e \in \mathbf{E}$. On the other hand, because there is no confusion in the

model, every function $h\colon \mathbf{E} \to \mathbf{DPOM}[\mathbf{E}]$ can be extended to a homomorphism. We will overload the symbols h, k to denote both kinds of functions.

A homomorphism h has the effect of a *substitution* or *refinement*: in principle, its application to a pomset has the effect that every element of the pomset is replaced by (a copy of) its h-image. We can define this operation directly as follows: $p[h] = [V, <, \ell]$ where

$$\begin{aligned} V &= \{(v,w) \mid v \in V_p, w \in V_{h(\ell_p(v))}\} \\ < &= \{((v,w),(v',w')) \mid v <_p v' \vee (v = v' \wedge w <_{h(\ell_p(v))} w')\} \\ \ell &= \{((v,w),e) \mid \ell_{h(\ell_p(v))}(w) = a\} \ . \end{aligned}$$

Hence vertices $v \in V_p$ are replaced by vertices (v,w) for all w from the h-image of $\ell_p(v)$; these new (v,w) receive their label from w. The ordering is inherited partly from p (as far as ordering between (v,w) and (v',w') for $v \neq v'$ is concerned) and partly from $h(\ell_p(v))$ (as far as the ordering of (v,w) and (v,w') is concerned).

4.1 Example. Let h map a to itself, b to ε and c to $\boxed{\begin{matrix} c\to d \\ e \end{matrix}}$; then for instance, $\boxed{c\to b\to a}[h] = \boxed{\begin{matrix} c\to d \\ \searrow \\ e\to a \end{matrix}}$ and $\boxed{\begin{matrix} a\to a \\ \searrow\!\!\nearrow \\ c\to b \end{matrix}}[h] = \boxed{\begin{matrix} \quad a \quad \\ \quad\searrow \\ c\to d\to a \\ \quad\nearrow \\ \quad e \quad \end{matrix}}$.

Unfortunately, refinement does not always yields a deterministic pomset even if p and the images of h are deterministic.

4.2 Example. Let $p = \boxed{\begin{matrix} a \\ b \end{matrix}}$ and let h map a to $\boxed{c\to d}$ and b to $\boxed{c\to e}$. Now $p[h] = \boxed{\begin{matrix} c\to d \\ c\to e \end{matrix}}$ which is not deterministic; on the other hand, $h(p) = h(a) \sqcup h(b) = \boxed{c\to d} \sqcup \boxed{c\to e} = \boxed{\begin{matrix} \quad d \\ \nearrow \\ c\to e \end{matrix}}$.

Hence in general it is not the case that $h(p) = p[h]$. In particular, as the above example shows, refinement does not distribute over join, i.e. $(p \sqcup q)[h] = p[h] \sqcup q[h]$ does not hold in general. On the other hand, refinement does distribute over concatenation and disjoint union.

4.3 Proposition. For all $p, q \in \mathbf{POM}$ and $h\colon \mathbf{POM} \to \mathbf{POM}$ the following equations hold:

$$\begin{aligned} p[h]\cdot q[h] &= (p\cdot q)[h] \\ p[h] \uplus q[h] &= (p \uplus q)[h] \ . \end{aligned}$$

This follows directly from the definitions of concatenation, disjoint union and refinement. The reason why refinement fails to distribute over join is basically that the images of different elements may fail to be sufficiently different themselves; in particular, they may share initial elements, as $h(a)$ and $h(b)$ in Example 4.2, in which case refinement no longer yields a deterministic pomset. We can, however, formulate necessary and sufficient conditions on h under which $h(p) = p[h]$ does hold for all p. Let us call a homomorphism *image distinct* if the following conditions hold:

- the images are nonempty: $h(e) \neq \varepsilon$ for all $e \in \mathbf{E}$;
- different images have nothing in common: $d \neq e$ implies $h(d) \sqcap h(e) = \varepsilon$ for all $d, e \in \mathbf{E}$.

4.4 Proposition. Let $h\colon \mathbf{DPOM} \to \mathbf{DPOM}$ be an arbitrary homomorphism; then $h(p) = p[h]$ for all $p \in \mathbf{DPOM}$ iff h is image distinct.

The proof follows below. The proof of the "if" part depends on the following lemma.

4.5 Lemma. If $h\colon \mathbf{DLPO} \to \mathbf{DLPO}$ is image distinct with pairwise compatible images and $p, q \in \mathbf{DLPO}$ are compatible, then $p[h]$ and $q[h]$ are also compatible.

Proof. Let $(v,w) \in V_{p[h]}$ and $(v',w') \in V_{q[h]}$ be arbitrary such that $\Downarrow_{p[h]}(v,w) = \Downarrow_{q[h]}(v',w')$ and $\ell_{p[h]}(v,w) = \ell_{q[h]}(v',w')$. The set $\Downarrow_{p[h]}(v,w)$ can be split up into the disjoint subsets

$$\begin{aligned} X_{(v,w)} &= \{(v'',w'') \in V_{p[h]} \mid v'' <_p v\} \\ Y_{(v,w)} &= \{(v,w'') \in V_{p[h]} \mid w'' <_{h(\ell_p(v))} w\} \ . \end{aligned}$$

Likewise, $\Downarrow_{q[h]}(v',w')$ can be split up into

$$\begin{aligned} X_{(v',w')} &= \{(v'',w'') \in V_{q[h]} \mid v'' <_q v'\} \\ Y_{(v',w')} &= \{(v',w'') \in V_{q[h]} \mid w'' <_{h(\ell_q(v'))} w'\} \ . \end{aligned}$$

If $Y_{(v,w)} \subseteq X_{(v',w')}$ then apparently $v <_q v'$, which would imply $(v,w) <_{q[h]} (v',w')$, contradicting $\Downarrow_{q[h]}(v',w') = \Downarrow_{p[h]}(v,w)$. Hence $Y_{(v,w)} \cap Y_{(v',w')} \neq \varnothing$, immediately implying $Y_{(v,w)} = Y_{(v',w')}$. This in turn implies $\Downarrow_{h(\ell_p(v))} w = \Downarrow_{h(\ell_q(v'))} w'$. We also have $\ell_{h(\ell_p(v))}(w) = \ell_{p[h]}(v,w) = \ell_{q[h]}(v',w') = \ell_{h(\ell_q(v'))}(w')$, implying $w = w'$ since all images of h are compatible. Because clearly $h(\ell_p(v)) \sqcap h(\ell_q(v'))$ contains at least w, by the distinctness of h it also follows that $\ell_p(v) = \ell_q(v')$.

Furthermore, $Y_{(v,w)} = Y_{(v',w')}$ also implies $X_{(v,w)} = X_{(v',w')}$, and therefore

$$\{v'' <_p v \mid \exists w''.\ (v'',w'') \in V_{p[h]}\} = \{v'' <_q v' \mid \exists w''.\ (v'',w'') \in V_{q[h]}\} \ .$$

Because by assumption $h(e) \neq \varepsilon$ for all $e \in \mathbf{E}$, for all $v'' <_p v$ there is a $w'' \in V_{h(\ell_p(v))}$, hence $(v'',w'') \in V_{p[h]}$; likewise, for all $v'' <_q v'$ there is a $w'' \in V_{h(\ell_q(v'))}$ and hence $(v'',w'') \in V_{q[h]}$. Hence the above equality is equivalent to $\Downarrow_p v = \Downarrow_q v'$. Together with $\ell_p(v) = \ell_q(v')$, already deduced above, and the fact that p and q are compatible, this implies $v = v'$. In combination with $w = w'$, already deduced above, this proves the compatibility of $p[h]$ and $q[h]$. □

Proof of Proposition 4.4: if Assume that the h-images, regarded as lposets, are compatible, and that p and q are compatible; then according to Lemma 4.5, $p[h]$ and $q[h]$ are compatible as well. A straightforward application of the definitions of join and refinement then establishes that $(p \sqcup q)[h] = p[h] \sqcup q[h]$ for all $p,q \in \mathbf{DPOM}$. $h(p) = p[h]$ can then be shown by induction on the structure of p.

only if If $h(a) = \varepsilon$ and $h(b) = p \neq \varepsilon$ then e.g. $\boxed{\begin{matrix} a \to b \\ b \end{matrix}}[h] = \boxed{a \to b}[h] \uplus \boxed{b}[h] = p \uplus p \neq p = h\left(\boxed{\begin{matrix} a \to b \\ b \end{matrix}}\right)$. On the other hand, if $h(a) \sqcap h(b) \neq \varepsilon$ then $h(a) \sqcup h(b) \neq h(a) \uplus h(b)$ and hence for instance $\boxed{\begin{matrix} a \\ b \end{matrix}}[h] = h(a) \uplus h(b) \neq h\left(\boxed{\begin{matrix} a \\ b \end{matrix}}\right)$. □

Another consequence of Lemma 4.5 is the following.

4.6 Proposition. Every image distinct homomorphism $h\colon \mathbf{DPOM} \to \mathbf{DPOM}$ is injective.

Proof. Assume $h(p) = h(q)$ where $p \neq q$, with p and q compatible. It follows that $p[h] = h(p) = h(q) = q[h]$ according to Proposition 4.4; let f be the (unique) isomorphism from $p[h]$ to $q[h]$. Let $(v,w) \in V_{p[h]}$ be $<_{p[h]}$-minimal such that $f(v,w) = (v',w') \neq (v,w)$. It follows by minimality that $\Downarrow_{p[h]}(v,w) = f(\Downarrow_{p[h]}(v,w)) = \Downarrow_{q[h]}(v',w')$, and $\ell_{p[h]}(v,w) = \ell_{q[h]}(v',w')$ because f is an isomorphism. Because $p[h]$ and $q[h]$ are compatible (Lemma 4.5) it follows that $(v,w) = (v',w')$, which contradicts the assumptions; hence such p,q do not exist. □

4.2 Refinement and determinisation

If we are working with arbitrary homomorphisms h rather than image distinct ones, there is still a clear relation between refinement and homomorphism application, through the *determinisation*

of a refined pomset (see Section 2.4). Namely, if we determinise $p[h]$ then the resulting deterministic pomset does correspond to $h(p)$ for arbitrary h. For the combination of refinement and determinisation we introduce a new operator $*$, defined by

$$h * p \;:=\; D(p[h]) \;\; .$$

The following lemma states that it does not matter if we first determinise p before applying $h * \_$.

4.7 Lemma. For all $p \in \mathbf{POM}$ and $h\colon \mathbf{POM} \to \mathbf{POM}$, $h * p = h * Dp$.

Proof. Established by comparing $\sim_{p[h]}$ with $\sim_{(Dp)[h]}$. In particular, it can be seen that for all $v \in V_p$ and $w \in V_{h(\ell_p(v))}$, $([v]_{\sim_p}, w) <_{(Dp)[h]} ([v']_{\sim_p}, w')$ iff there is a $v'' \sim_p v'$ such that $(v, w) <_{p[h]} (v'', w')$, and that $v \sim_p v'$ implies $(v, w) \sim_{p[h]} (v', w)$. It follows that

$$(v, w) \sim_{p[h]} (v', w') \iff ([v]_{\sim_p}, w) \sim_{(Dp)[h]} ([v']_{\sim_p}, w') \;\; ,$$

hence the function $f\colon V_{h*p} \to V_{h*Dp}$ defined by

$$f\colon [(v, w)]_{\sim_{p[h]}} \mapsto [([v]_{\sim_p}, w)]_{\sim_{(Dp)[h]}}$$

is an isomorphism. □

We are now ready to state and prove the correspondence of refinement followed by determinisation to homomorphism application.

4.8 Theorem. For all $p \in \mathbf{DPOM}$ and $h\colon \mathbf{DPOM} \to \mathbf{DPOM}$, $h(p) = h * p$.

Proof. First recall Theorem 2.27 which states that D takes $\cdot$ over **POM** to $\cdot$ over **DPOM**, and $\uplus$ to $\sqcup$. Using also Proposition 4.3, we can derive

$$h * (p{\cdot}q) = D((p{\cdot}q)[h]) = D(p[h]{\cdot}q[h]) = D(p[h]){\cdot}D(q[h]) = (h * p){\cdot}(h * q) \;\; .$$

Furthermore, by applying Lemma 4.7 we get

$$h * (p \sqcup q) = h * D(p \uplus q) = h * (p \uplus q) = D((p \uplus q)[h]) = D(p[h] \uplus q[h]) = (h * p) \sqcup (h * q) \;\; .$$

Finally, it is clear that $h{*}\varepsilon = D(\varepsilon[h]) = D\varepsilon = \varepsilon$ and for all $e \in \mathbf{E}$, $h{*}e = D(e[h]) = D(h(e)) = h(e)$. The theorem therefore follows by induction on the structure of terms in T_{det}. □

The following corollary supplements Proposition 4.4 in that it states some more circumstances in which refinement corresponds directly to homomorphism application, without the intermediate step of determinisation.

4.9 Corollary. For all $p \in \mathbf{DPOM}$ and $h\colon \mathbf{E} \to \mathbf{DPOM}$, $p[h] = h(p)$ iff $p[h]$ is deterministic.

In the remainder of this paper we will apply the term "refinement" as equivalent to "homomorphism application", hence ignore the fact that a determinisation step takes place in between. Accordingly, we will refer to $*$ as the "refinement operator."

4.3 Refinement algebraically: the algebra A^*_{det}

Having established that for deterministic pomsets, homomorphism application corresponds to a refinement-like operator, we now want to introduce this operator into the algebra of deterministic pomsets, resulting in an extended algebra A^*_{det} with $\Sigma^*_{det} = \langle \varepsilon, \cdot, \sqcup, * \rangle$. This entails introducing denotations for refinement functions (i.e., homomorphisms). We will restrict ourselves to refinement functions that are *the identity almost everywhere*, i.e. which map only a finite number of events to terms other than themselves. The refinement of t according to h is denoted $h * t$.

To denote a refinement function h, we will simply list the pairs of events and images for which the image does not equal the event: e.g., $h = [t_1/e_1, \ldots, t_n/e_n]$ (abbreviated $[t_i/e_i]_{i\in I}$) denotes the function mapping e_i to t_i for all $i \in I = \{1, \ldots, n\}$, and e to itself for all events $e \in \mathbf{E} \setminus \{e_i\}_{i\in I}$; in other words,

$$h\colon e \mapsto \begin{cases} t_i & \text{if } e = e_i \\ e & \text{if } e \neq e_i \text{ for all } i \in I. \end{cases}$$

We sometimes refer to $\{e_i \mid i \in I\}$ as the *syntactic domain* of h. The empty list, corresponding to the identity function over $\mathbf{E}$, is denoted id. Finally, for all finite $E \subseteq \mathbf{E}$ we introduce functions $h_E = [x_e/e]_{e\in E}$ and $k_E = [y_e/e]_{e\in E}$ mapping the events in E to distinct variables, i.e., such that $x_d \neq x_e$ if $d \neq e$, and $x_d \neq y_e$ for all $d, e \in \mathbf{E}$. (Hence, in this notation, $h_\varnothing = id = k_\varnothing$.)

The resulting algebra is two-sorted: on the one hand we have pomsets and on the other refinement functions. The refinement operator is also extended pointwise to refinement functions as right hand operands, by setting $h * k = \lambda e.\, h * k(e)$; in our chosen notation, this becomes

$$[s_e/e]_{e\in E} * [t_d/d]_{d\in F} = [([s_e/e]_{e\in E} * t_d)/d, s_e/e]_{d\in F, e\in E\setminus F} \ .$$

We then have the following additional equations for all finite $E, F \subseteq \mathbf{E}$:

$$h_E * e = \begin{cases} x_e & \text{if } e \in E \\ e & \text{otherwise.} \end{cases} \tag{32}$$

$$h_E * \varepsilon = \varepsilon \tag{33}$$

$$h_E * (x \cdot y) = (h_E * x) \cdot (h_E * y) \tag{34}$$

$$h_E * (x \sqcup y) = (h_E * x) \sqcup (h_E * y) \tag{35}$$

$$h_E * (k_F * x) = (h_E * k_F) * x \tag{36}$$

$$id * x = x \tag{37}$$

Note that every line (except the last one) actually corresponds to a (countable) infinity of equations, one for each instantiation of E resp. F. The alternative would be to introduce second-order variables for refinement functions, for which a complete theory would be much more difficult to obtain.

For A^*_{det} we can prove basically the same soundness and completeness properties as for A_{det}. First we state soundness and ordinary (ground) completeness:

4.10 Theorem (A^*_{det} is sound and complete). For all $s, t \in T^*_{det}$, $\mathrm{A}^*_{det} \vdash s = t$ iff $[\![s]\!] = [\![t]\!]$.

Proof. The soundness of (32)–(37) is immediate; this together with Theorem 3.2 proves the "only if" part of the theorem. For the "if" part, note that every $t \in T^*_{det}$ can be rewritten modulo provable equality to a pomset normal form in the sense of Definition 3.6, by application of (32)–(35); in particular, one may add the following rule to the algorithm presented in the proof of Theorem 3.5:

$$norm(h * t) := \bigcup_{t'\cdot e \in norm(t)} norm((h * t') \cdot h(e)) \ .$$

Note that equations (36) and (37) are not necessary for the purpose of this proof; indeed, they are required only if we want to prove the stronger property of ω-completeness, as we will see below. □

4.4 ω-completeness of A^*_{det}

The theory of deterministic pomsets with refinement is stronger than is apparent from the results so far: just as for the basic theory A_{det} we can also prove completeness for open terms. The relevant statement of this property is as follows:

4.11 Theorem (A^*_{det} is ω-complete). Assume $|\mathbf{E}| = \omega$. For all $s, t \in T^*_{det}(\mathbf{E}, \mathbf{X})$, if $[\![s\rho]\!] = [\![t\rho]\!]$ for all ground substitutions $\rho\colon \mathbf{X} \to T^*_{det}(\mathbf{E})$ then $\mathrm{A}^*_{det} \vdash s = t$.

To prove this, we use the same technique as before, but its application this time around has become a good deal more complicated. In particular, it is not the case that refinement-free open normal forms suffice to capture all open A^*_{det}-terms: for instance, $[t/e] * x$ cannot be reduced to a refinement-free term since we know nothing in general about the presence of e in the term to be substituted for x. We are therefore forced to introduce a new kind of normal form.[3]

4.12 Definition (open $*$-normal forms). Consider the following production rule for terms of $T^*_{det}(\mathbf{E}, \mathbf{X})$:

$$N ::= (\bigsqcup \textbf{sat'd set of } N)\cdot e \mid (\bigsqcup \textbf{sat'd set of } N)\cdot([\bigsqcup \textbf{sat'd set of } N/e]_{e\in E} * x)$$

where $e \in \mathbf{E}$, $x \in \mathbf{X}$, $E \subseteq_{\mathrm{fin}} \mathbf{E}$ and a set T of N-produced terms is *saturated* if $T' \subseteq T$ for all $(\bigsqcup T')\cdot e' \in T$, and furthermore, if $[t_e/e]_{e\in E}$ is a refinement function appearing in an N-produced term, then $t_e \neq \varepsilon\cdot e$ for all $e \in E$. A term is in *open $*$-normal form* if it equals $\bigsqcup T$ for some saturated set T of N-produced terms.

Hence the "tail pieces" of open N-produced terms are (apart from the usual elements e) not simply variables x but *refined* variables $h * x$, where the refinement function h is itself also in normal form. For instance, the above term $[t/e] * x$ corresponds to the open $*$-normal form $\varepsilon\cdot([t'/e] * x)$ where t' is the open $*$-normal form of t. To turn arbitrary open A^*_{det}-terms into open $*$-normal form terms, we define a recursive function which is a variation on *norm*:

$$\begin{aligned}
norm_*(\varepsilon) &:= \varnothing \\
norm_*(e) &:= \{\varepsilon\cdot e\} \\
norm_*(x) &:= \{\varepsilon\cdot(id * x)\} \\
norm_*(s\cdot t) &:= norm_*(s) \cup \{(\textstyle\bigsqcup norm_*(s\cdot t'))\cdot t'' \mid t'\cdot t'' \in norm_*(t)\} \\
norm_*(s \sqcup t) &:= norm_*(s) \cup norm_*(t) \\
norm_*(h * t) &:= \bigcup_{s\cdot e\in norm_*(t)} norm_*((h * s)\cdot h(e)) \\
&\quad \cup \bigcup_{s\cdot(k*x)\in norm_*(t)} \{(\textstyle\bigsqcup norm_*(h * s))\cdot(norm_*(h * k) * x)\}
\end{aligned}$$

where the normalisation of refinement functions is defined by pointwise extension

$$norm_*([t_e/e]_{e\in E}) := [\textstyle\bigsqcup norm_*(t_e)/e]_{e\in F} \quad (F = \{e \in E \mid norm_*(t_e) \neq \{\varepsilon\cdot e\}}) \ .$$

Note that we remove mappings t_e/e where t_e normalises to $\varepsilon\cdot e$ $(= e)$; in our chosen notation, such mappings are implicit for all events not in the syntactic domain of a refinement function. The role of $norm_*$ is formulated in the following lemma, which is proved by a tedious but straightforward induction on the term structure.

4.13 Lemma (open $*$-normal forms exist). For all terms $t \in T^*_{det}(\mathbf{E}, \mathbf{X})$, $\bigsqcup norm_*(t)$ is an open $*$-normal form such that $\mathrm{A}^*_{det} \vdash t = \bigsqcup norm_*(t)$.

We now come to the characteristic substitutions used to establish the normality of normal forms. Again, the substitutions used in the proof of Theorem 3.10 no longer suffice. We say that e *does not occur* in a refinement function $[t_e/e]_{e\in E}$ if it is neither in the syntactic domain E nor in any of the images t_e.

[3] The fact that open normal forms for A^*_{det} are not trivially derived from closed normal forms can be regarded as a consequence of an axiom in the theory that deals specifically with elements, viz. Equation (32).

4.14 Example. If $\rho_{s,t}(x) = e_x$ where e_x is a "fresh" event not occurring in s or t, then $h(e_x) = e_x$ for any refinement function h occurring in s or t; hence for instance, if $s = [s'/e] * x$ and $t = [t'/e] * x$ where s', t' are ground terms such that $[\![s']\!] \neq [\![t']\!]$, then $\mathrm{A}_{det} \vdash s\rho_{s,t} = e_x = t\rho_{s,t}$ but $[\![s\rho]\!] \neq [\![t\rho]\!]$ if $\rho(x) = e$.

Basically, the problem is that the characteristic substitution must preserve enough structure of the normal forms to which it is applied to be injective; this structure includes especially the "tail ends" $h * x$ allowed by Definition 4.12. To achieve this, then, $\rho_{s,t}(x)$ must contain copies of all elements with a nontrivial h-image, in such a way, moreover, that these images can be re-retrieved from $h * (\rho_{s,t}(x))$.

Again, let $E_{s,t}$ be the set of events occurring syntactically in s or t. Assume a fixed ordering over $E_{s,t}$, such that $E_{s,t} = \{e_1, \ldots, e_n\}$. Let $\{d_x, e_x\}_{x \in \mathbf{X}}$ be a set of pairwise distinct events disjoint from $E_{s,t}$. Now $\rho_{s,t} \colon \mathbf{X} \to T^*_{det}$ is defined as follows:

$$\rho_{s,t} \colon x \mapsto d_x \sqcup e_x \cdot e_1 \cdot e_x \cdot e_2 \cdots e_x \cdot e_n \ .$$

The d_x and e_x play the role of special markers: d_x signals the start of a subterm $\rho_{s,t}(x)$ whereas the e_x separate the e_i. The e_i themselves are needed to record the effect of refinements that $\rho_{s,t}(x)$ may be submitted to; by keeping this record one avoids the accidental confusion of $s\rho_{s,t}$ and $t\rho_{s,t}$ as in Example 4.14.

The pomsets constructed by terms of the form $t\rho_{s,t}$ therefore have a specific format that allows to retrieve essentially t (up to provable equality). We call p *characteristic* if it has this format. Characteristicness is defined as follows.

4.15 Definition (characteristic pomsets). Let $E_{s,t} = \{e_1, \ldots, e_n\}$ and $\{d_x, e_x\}_{x \in \mathbf{X}}$ be sets of elements as above. A pomset p is called *characteristic* if for all $v \in V_p \setminus \ell_p^{-1}(E_{s,t})$

- if $\ell_p(v) = d_x$ then the set of *characteristic vertices* $C_v \subseteq V_p$ defined by

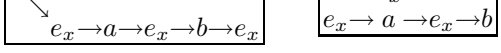
$$C_v := \{w \in V_p \mid \forall u \in V_p.\ (u <_p v \Rightarrow u <_p w) \wedge (u >_p v \Rightarrow u >_p w)\}$$

  has the property that for all $w \in C_v$ and $u \in V_p \setminus C_v$, $u <_p w$ implies $u <_p v$ and $u >_p w$ implies $u >_p v$. Moreover, $p \restriction C_v = d_x \sqcup (e_x \cdot p_1 \cdot e_x \cdot p_2 \cdots e_x \cdot p_n)$ where for all $1 \leq i \leq n$, p_i is a characteristic pomset, sometimes denoted $C_v(e_i)$;
- if $\ell_p(w) = e_x$ there is a $v \in V_p$ such that $\ell_p(v) = d_x$ and $w \in C_v$.

We will not mention the sets $E_{s,t}$ and $\{d_x, e_x\}_{x \in \mathbf{X}}$ with respect to which this property is defined when they are implicitly clear. If p is characteristic and $C_v \subseteq V_p$ is a set of characteristic vertices, then C_v can be contracted into a single node w, yielding a pomset q from which p can be reconstructed by refining q according to $w \mapsto p \restriction C_v$. Note that $C_v \cap C_w \neq \varnothing$ for $v, w \in V_p$ such that $\ell_p(v) = d_x$ and $\ell_p(w) = d_y$ implies $C_v \subseteq C_w$ or $C_w \subseteq C_v$. It follows that for all $v \in V_p$, either there is no set C_w such that $v \in C_w$, or there is a unique largest such C_w. Very important is the property that for any characteristic p, if $\ell_p(v) = d_x$ then the $C_v(e_i)$ are uniquely defined.

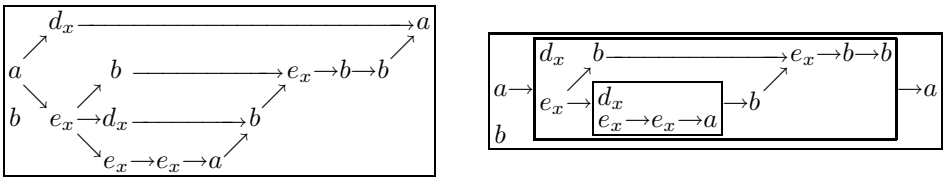
4.16 Example.

1. Any pomset in which there are no d_x- or e_x-labelled vertices is characteristic.
2. If $\rho_{s,t}$ is a characteristic substitution then $p = [\![\rho_{s,t}(x)]\!]$ is a characteristic pomset for all $x \in \mathbf{X}$: there is exactly one v such that $\ell_p(v) = d_x$, where $C_v = V_p$; $p \restriction C_v = p = d_x \sqcup e_x \cdot e_1 \cdots e_x \cdot e_n$ by construction, hence $C_v(e_i) = e_i$ for all $1 \leq i \leq n$.
3. Assume $E_{s,t} = \{a, b\}$ and $\mathbf{X} = \{x\}$. Then the following pomsets are *not* characteristic:



In the left hand pomset, the subpomset $e_x{\cdot}a{\cdot}e_x{\cdot}b{\cdot}e_x$ cannot be subdivided into $e_x{\cdot}p_a{\cdot}e_x{\cdot}p_b$ such that p_a and p_b are again characteristic, since either p_a or p_b must contain an e_x-element but neither can contain a d_x-element. In the right hand pomset, on the other hand, there is no appropriate set C_v to the d_x-element, since the initial a-element is not a predecessor of the e_x's.

4. Let $E_{s,t}$ and $\mathbf{X}$ be as above, and consider the left hand side pomset:



This pomset is characteristic: the right hand side indicates its division into principal subpomsets. It can be regarded as $\boxed{\begin{matrix}a{\to}x{\to}a\\ b\end{matrix}}$ where x is refined by $d_x \sqcup (e_x{\cdot}p_a{\cdot}e_x{\cdot}p_b)$, such that $p_a = C_v(a) = h * \boxed{\begin{matrix}b\\ x{\to}b\end{matrix}}$ and $p_b = C_v(b) = \boxed{b{\to}b}$, where the refinement function h in p_a is given by $x \mapsto d_x \sqcup (e_x{\cdot}\varepsilon{\cdot}e_x{\cdot}a)$.

One can prove, by induction on the term structure, that pomsets obtained by applying a characteristic ground substitution to an open A^*_{det}-term are always characteristic in the above sense.

4.17 Lemma. For all $s, t \in T^*_{det}(\mathbf{X})$, $[\![t\rho_{s,t}]\!]$ is a characteristic pomset.

The next task consists of reconstructing a (normal form) term from an arbitrary characteristic pomset, with the property that applying the characteristic substitution to that term once more yields the pomset we started with. For this purpose we need one more auxiliary notion. If p is a characteristic pomset, then $v \in V_p$ is called *principal* if either there is no $w \in V_p$ such that $v \in C_w$, or $\ell_p(v) = d_x$ and $v \in C_w$ implies $v = w$. (The latter is equivalent to saying that C_v is *maximal* among all characteristic sets of vertices containing v; we have seen above that such maximal C_v always exist.) The principal vertices of p are denoted VP_p.

Now we recursively define a partial function $R_*\colon \mathbf{DPOM} \to \mathit{Fin}(T^*_{det}(\mathbf{X}))$ from characteristic pomsets to finite sets of open $*$-normal terms, as follows:

$$\begin{aligned} R_*(p) \quad = \quad & \{(\bigsqcup R_*(p \restriction \Downarrow v)){\cdot}\ell(v) \mid v \in VP, \ell(v) \neq d_x\} \cup \\ & \{(\bigsqcup R_*(p \restriction \Downarrow v)){\cdot}([\![\bigsqcup R_*(C_v(e))/e]\!]_{e \in E_{s,t}, C_v(e) \neq e} * x) \mid v \in VP, \ell(v) = d_x\} \ . \end{aligned}$$

In words: the principal vertices v are turned into N-produced terms (see Definition 4.12), where the vertex label $\ell_p(v)$ determines if the produced subterm has a "simple tail" consisting of a single element $\ell_p(v) \neq d_x$, or a "complex tail" $h * x$ corresponding to the refinement of a variable if $\ell_p(v) = d_x$. In the latter case, the refinement function h is reconstructed from the subpomset determined by the characteristic vertices C_v.

Note that the saturation requirement of normal forms is fulfilled due to the fact that if $v <_p w$ for two principal vertices $v, w \in VP_p$ then $v \in \Downarrow_p w$ and hence the $R_*(p \restriction \Downarrow w)$ will include the subterm $(\bigsqcup T_v){\cdot}t_v$ constructed for v.

4.18 Example. For the pomset in Example 4.16.4, R_* yields the following set:

$$\{\varepsilon{\cdot}a, \varepsilon{\cdot}b, (\varepsilon{\cdot}a){\cdot}(h_1 * x), (\varepsilon{\cdot}a \sqcup (\varepsilon{\cdot}a){\cdot}(h_1 * x)){\cdot}a\}$$

where

$$\begin{aligned} h_1\colon a \quad &\mapsto \quad \varepsilon{\cdot}b \sqcup \varepsilon{\cdot}(h_2 * x) \sqcup (\varepsilon{\cdot}(h_2 * x)){\cdot}b \\ b \quad &\mapsto \quad \varepsilon{\cdot}b \sqcup (\varepsilon{\cdot}b){\cdot}b \ ; \\ h_2\colon a \quad &\mapsto \quad \varepsilon \\ b \quad &\mapsto \quad \varepsilon{\cdot}a \ . \end{aligned}$$

The following lemma states the role of the function R_*. It is analogous to Lemma 3.8 and proved by induction on the structure of open $*$-normal form terms.

4.19 Lemma (open $*$-normal forms are unique). If $t \in T^*_{det}(\mathbf{X})$ is an open $*$-normal form terms and $s \in T^*_{det}(\mathbf{X})$ is arbitrary then $\bigsqcup R_*([\![t\rho_{s,t}]\!]) = t$.

Proof sketch of Theorem 4.11. Let $s, t \in T^*_{det}(\mathbf{E}, \mathbf{X})$ be arbitrary, and let s', t' be the corresponding open $*$-normal form terms, i.e., such that $\mathrm{A}^*_{det} \vdash s = s', t = t'$. The existence of s' and t' is ensured by Lemma 4.13. If $[\![s\rho]\!] = [\![t\rho]\!]$ for all ground substitutions ρ, then also $[\![s'\rho_{s',t'}]\!] = [\![t'\rho_{s',t'}]\!]$; hence $s' = \bigsqcup R[\![s'\rho_{s',t'}]\!] = \bigsqcup R[\![t'\rho_{s',t'}]\!] = t'$ (Lemma 4.19). It follows that $\mathrm{A}^*_{det} \vdash s = t$. □

5 Termination of pomsets

One of the aspects of the algebra A_{det} that may be a disadvantage is that concatenation is not $\sqsubseteq$-monotone in its left operand —or, stated algebraically, that concatenation fails to distribute over join from the right. For instance, if we should want to solve equations in A_{det} then the technique of fixpoint constructions requires monotonicity of all operators.[4] In this section we discuss how to obtain left-monotonicity by introducing a notion of *termination*.

The basic idea is that a pomset either is or is not terminated, and if it is not, then it cannot be extended by concatenation. To capture this effect, we introduce a new binary operation called *sequential composition*, denoted ';', with the effect that if the first of two sequentially composed pomsets is not terminated then the second one disappears entirely, whereas if it is terminated then the sequential composition corresponds to concatenation of the two. However, there are at least two ways in which this basic idea may be given substance. An algebraically quite pleasing solution is to allow *multiple* or *distributed* termination, where different parts of a pomset may terminate independently, i.e., there may be more than one exit point. Another solution takes the point of view that termination should be regarded as a *global* concept, hence a pomset is terminated as a whole or not terminated at all. We describe both solutions below, albeit the former in more detail than the latter.

5.1 Distributed termination: the algebra $\mathrm{A}^{\checkmark}_{det}$

Let us lead up to the idea of distributed termination gradually. To signal termination of a pomset we introduce a special element $\mathbb{T} \in \mathbf{E}$. This label may only occur as *maximal element* of a pomset, since it would contradict our intuition if anything could happen following termination.

5.1 Example.

1. $\boxed{a \to b}$ is not terminated but $\boxed{a \to b \to \mathbb{T}}$ is; $\boxed{a \to \mathbb{T} \to b}$ is not allowed.
2. $\boxed{\begin{matrix} a \\ b \end{matrix}}; \boxed{c} = \boxed{\begin{matrix} a \\ b \end{matrix}}$ whereas $\boxed{\begin{matrix} a \\ \searrow \\ b \to \mathbb{T} \end{matrix}}; \boxed{c} = \boxed{\begin{matrix} a \\ \searrow \\ b \to c \end{matrix}}$.

Immediately however the problem arises that for instance $\boxed{a \to \mathbb{T}} \sqcup \boxed{b \to \mathbb{T}} = \boxed{\begin{matrix} a \to \mathbb{T} \\ b \to \mathbb{T} \end{matrix}}$ in which there are two occurrences of $\mathbb{T}$, rather than just one. In the distributed termination approach therefore, instead of saying that a pomset is either terminated or not, we say that it has a number of *exit points*, labelled by the element $\mathbb{T}$; the second operand of sequential composition is appended at all exit points of the first operand.

[4] In fact, there are other shortcomings of the theory in its current form that prevent the construction of fixpoints; most notably, one would need infinitary joins —which are actually straightforward to add— and continuity rather than monotonicity. A detailed discussion is outside the scope of this paper.

5.2 Example. Some special cases of exit points; left-distributivity holds.

1. $(\boxed{a{\to}\mathbb{T}} \sqcup \boxed{b{\to}\mathbb{T}}); \boxed{c} = \boxed{\begin{matrix} a{\to}\mathbb{T} \\ b{\to}\mathbb{T} \end{matrix}}; \boxed{c} = \boxed{\begin{matrix} a{\to}c \\ b{\to}c \end{matrix}} = \boxed{a{\to}c} \sqcup \boxed{b{\to}c} = (\boxed{a{\to}\mathbb{T}}; \boxed{c}) \sqcup (\boxed{b{\to}\mathbb{T}}; \boxed{c})$.
2. $(\boxed{a{\to}\mathbb{T}} \sqcup \boxed{b{\to}c}); \boxed{d} = \boxed{\begin{matrix} a{\to}\mathbb{T} \\ b{\to}c \end{matrix}}; \boxed{d} = \boxed{\begin{matrix} a{\to}d \\ b{\to}c \end{matrix}} = \boxed{a{\to}d} \sqcup \boxed{b{\to}c} = (\boxed{a{\to}\mathbb{T}}; \boxed{d}) \sqcup (\boxed{b{\to}c}; \boxed{d})$.

To perform the operation "append at all exit points", rather than concatenation we use *refinement* of the $\mathbb{T}$-event in the first operand, where all elements other than $\mathbb{T}$ are unchanged. This is captured by the following definition:

$$x; y \;:=\; [y/\mathbb{T}] * x \;\;. \tag{38}$$

It should be noted that in some cases, this definition may have unexpected consequences, due to the fact that we are working in the context of *deterministic* pomsets; especially if the first operand has both exit points and other maximal elements.

5.3 Example. $\boxed{\begin{matrix} a \\ \mathbb{T} \end{matrix}}; \boxed{a{\to}b} = [a{\cdot}b/\mathbb{T}] * (a \sqcup \mathbb{T}) = a \sqcup (a{\cdot}b) = \boxed{a{\to}b}$.

In addition we introduce an operator to turn non-terminated pomsets into terminated ones, basically by appending $\mathbb{T}$; however, we have to be careful to ensure that this does not result in non-maximal exit points, therefore we first remove all existing $\mathbb{T}$'s by refining them to ε. The operator is denoted postfix by $\checkmark$:

$$x\checkmark \;:=\; ([\varepsilon/\mathbb{T}] * x){\cdot}\mathbb{T} \tag{39}$$

Hence we have extended A_{det} with a binary operator for sequential composition and a unary operator to ensure termination. It turns out that these new operators may in some sense replace concatenation. First we list a number of equations that can be derived for the new operators using the A_{det}-theory in combination with (38) and (39). We use $\mathbf{0}$ instead of ε to denote the bottom element of join, and $\mathbf{1}$ to denote the constant corresponding to the single-element pomset $\mathbb{T}$.

$$\begin{aligned}
\mathbf{1}; x &= x && (40)\\
x; \mathbf{1} &= x && (41)\\
(x; y); z &= x; (y; z) && (42)\\
\mathbf{0}; x &= \mathbf{0} && (43)\\
e; \mathbf{0} &= e && (44)\\
x\checkmark; y\checkmark &= (x\checkmark; y)\checkmark && (45)\\
x\checkmark; \mathbf{0} &= x; \mathbf{0} && (46)\\
\mathbf{0}\checkmark &= \mathbf{1} && (47)\\
x; (y \sqcup z) &= x; y \sqcup x; z && (48)\\
(x \sqcup y); z &= x; z \sqcup y; z \;\;. && (49)
\end{aligned}$$

In fact, (49) is not derivable in this way but must be proved by analysis of the model. In addition, the following equations from A_{det} remain valid:

$$\begin{aligned}
\mathbf{0} \sqcup x &= x && (50)\\
(x \sqcup y) \sqcup z &= x \sqcup (y \sqcup z) && (51)\\
x \sqcup y &= y \sqcup x && (52)\\
x \sqcup x &= x && (53)
\end{aligned}$$

Note that other than in A_{det}, the idempotence of join (53) is no longer derivable from the other equations, basically because the neutral element of sequential composition ($\mathbf{1}$) is different from that of join ($\mathbf{0}$), whereas for concatenation and join they were the same (ε). We can also derive a number of auxiliary equations:

- $x\checkmark = x\checkmark; \mathbf{1} = x\checkmark; \mathbf{0}\checkmark = (x\checkmark; \mathbf{0})\checkmark = (x; \mathbf{0})\checkmark$
- $x\checkmark\checkmark = (x\checkmark; \mathbf{0})\checkmark = (x; \mathbf{0})\checkmark = x\checkmark$
- $(x\checkmark \sqcup y)\checkmark = ((x\checkmark \sqcup y); 0)\checkmark = (x\checkmark; 0 \sqcup y)\checkmark = (x; 0 \sqcup y; 0)\checkmark = ((x \sqcup y); 0)\checkmark = (x \sqcup y)\checkmark$.

Now we define the theory $\mathrm{A}^{\checkmark}_{det}$ consisting of the signature $\Sigma^{\checkmark}_{det} = \langle \mathbf{0}, \mathbf{1}, \checkmark, ;, \sqcup \rangle$ and the equations (45)–(53). Models of $\mathrm{A}^{\checkmark}_{det}$ are pomsets in which all $\mathbb{T}$-labelled vertices are maximal; such pomsets are called *terminating* and are collected in $\mathbf{DPOM}^{\checkmark}[\mathbf{E}] \subset \mathbf{DPOM}[\mathbf{E} \cup \{\mathbb{T}\}]$ (where $\mathbb{T} \notin \mathbf{E}$).

On the other hand, note that the following holds for all $\mathbb{T}$-free pomsets x, i.e., such that $x; \mathbf{0} = x$:

$$x \cdot y = x\checkmark; y \tag{54}$$

Using this equality, we can alternatively derive the A_{det}-equations for concatenation from $\mathrm{A}^{\checkmark}_{det}$:

- $(x \cdot y) \cdot z = (x\checkmark; y)\checkmark; z = (x\checkmark; y\checkmark); z = x\checkmark; (y\checkmark; z) = x \cdot (y \cdot z)$ (note that $x \cdot y$ is $\mathbb{T}$-free iff both x and y are);
- $\varepsilon \cdot x = \varepsilon\checkmark; x = \mathbf{1}; x = x$;
- $x \cdot \varepsilon = x\checkmark; \mathbf{0} = x; \mathbf{0} = x$ (the last step by assumption).

We would therefore be equally justified in regarding A_{det} as having been derived from $\mathrm{A}^{\checkmark}_{det}$, such that the models of A_{det} form a subclass $\mathbf{DPOM}[\mathbf{E}] \subset \mathbf{DPOM}^{\checkmark}[\mathbf{E}]$ of those of $\mathrm{A}^{\checkmark}_{det}$, rather than the other way around. Through the connection with basic pomsets established by (39) and (38), we can immediately prove the soundness of theory of terminating pomsets.

5.4 Theorem ($\mathrm{A}^{\checkmark}_{det}$ is sound). For all $s, t \in T^{\checkmark}_{det}(\mathbf{X})$, if $\mathrm{A}^{\checkmark}_{det} \vdash s = t$ then $[\![s\rho]\!] = [\![t\rho]\!]$ for all ground substitutions $\rho \colon \mathbf{X} \to T^{\checkmark}_{det}$.

Proof sketch. By translation to A^{*}_{det}; using (38) and (39), the equations of $\mathrm{A}^{\checkmark}_{det}$ directly translate to provable equations or A^{*}_{det}, with the exception of (49), which translates to

$$[(x \sqcup y)x/\mathbb{T}] * z = ([x/\mathbb{T}] * z) \sqcup ([y/\mathbb{T}] * z) \ .$$

It is however easy to establish that this is valid in $\mathbf{DPOM}^{\checkmark}$, due to the fact that the elements being refined are always maximal. □

Proving completeness is somewhat more involved. Other than for A^{*}_{det}, we can no longer use the standard normal form (Definition 3.6) for proving (ground) completeness, since (apart from the fact that we are working with a different signature) we have imposed a restriction on the pomsets under consideration, namely that all $\mathbb{T}$-labelled vertices are to be maximal. Let us first show how to construct denotations in $\Sigma^{\checkmark}_{det}$ for all such pomsets. Consider the following recursively defined function $R_{\checkmark} \colon \mathbf{DPOM}^{\checkmark} \to \mathit{Fin}(T^{\checkmark}_{det})$:

$$R_{\checkmark}(p) := \{(\bigsqcup R_{\checkmark}(p \upharpoonright \Downarrow v))\checkmark; \mathbf{1} \mid v \in V, \ell(v) = \mathbb{T}\} \cup \{(\bigsqcup R_{\checkmark}(p \upharpoonright \Downarrow v))\checkmark; \ell(v) \mid v \in V, \ell(v) \neq \mathbb{T}\} \ .$$

(For readability, we will actually write normal forms using $\mathbf{0}$, $\mathbf{1}$, t and $t_1 \sqcup \cdots \sqcup t_n$ rather than $\bigsqcup \varnothing$, $(\bigsqcup \varnothing)\checkmark$, $\bigsqcup \{t\}$ and $\bigsqcup \{t_1, \ldots, t_n\}$, respectively.) It can be proved by induction on the size of V_p that $p = [\![\bigsqcup R_{\checkmark}(p)]\!]$ for all $p \in \mathbf{DPOM}^{\checkmark}$.

5.5 Example. The denotation for $\boxed{\begin{matrix} a \to \mathbb{T} \\ \nearrow \\ b \to a \end{matrix}}$ obtained from $R_{\checkmark}$ is $\mathbf{1}; a \sqcup \mathbf{1}; b \sqcup (\mathbf{1}; a \sqcup \mathbf{1}; b)\checkmark \sqcup (\mathbf{1}; b)\checkmark; a$.

A more parsimonious denotation is for instance $(a \sqcup b)\checkmark \sqcup b\checkmark; a$.

The general property that every terminating deterministic pomset has a denotation in $\mathrm{A}^{\checkmark}_{det}$ is formulated in the following theorem, which is the counterpart of Theorem 3.3:

5.6 Theorem (no junk in $\mathrm{A}^{\checkmark}_{det}$). For all $p \in \mathbf{DPOM}^{\checkmark}[\mathbf{E}]$ there is a $t \in T^{\checkmark}_{det}(\mathbf{E})$ such that $p = [\![t]\!]$.

In addition, as mentioned above, to prove completeness we need a more restricted format for normal form terms, preferably formulated in terms of $\Sigma^{\checkmark}_{det}$. This is given by the following definition.

5.7 Definition (✓-normal forms). Consider the following grammar for terms of $T^{\checkmark}_{det}(\mathbf{E})$:

$$\begin{aligned} M &:= (\bigsqcup \textbf{sat'd set of } N)\checkmark; \mathbf{1} \ . \\ N &:= (\bigsqcup \textbf{sat'd set of } N)\checkmark; e \end{aligned}$$

where $e \in \mathbf{E}$ and a set T of N-produced terms is *saturated* if $T' \subseteq T$ whenever $(\bigsqcup T'); t' \in T$. A term is in *✓-normal form* if it equals $\bigsqcup T$ for some saturated set T of M- and N-produced terms.

The important difference with the ordinary normal form is that here we have two production rules. N-produced terms *terminate nowhere*, i.e., the corresponding pomsets are $\mathbb{T}$-free. In terms of the equational theory, $\mathrm{A}^{\checkmark}_{det} \vdash t; \mathbf{0} = t$ for all N-produced t. Therefore a subterm $(\bigsqcup T)\checkmark; \mathbf{1}$ where all $t \in T$ are N-produced terms corresponds to the A_{det}-term $(\bigsqcup T)\cdot\mathbb{T}$ whereas $(\bigsqcup T)\checkmark; e$ corresponds to $(\bigsqcup T)\cdot e$. Using this insight, it is clear that ✓-normal forms in fact encode (ordinary) normal forms of terminating pomsets in the new signature $\Sigma^{\checkmark}_{det}$. The following lemma is the counterpart of Lemma 3.8.

5.8 Lemma (✓-normal forms are unique). $\bigsqcup R_{\checkmark}([\![t]\!]) = t$ for all ✓-normal forms t.

A recursively defined function along the lines of *norm* and $norm_*$ in the previous sections serves to construct saturated sets of M- and N-produced terms from arbitrary terms of $\mathrm{A}^{\checkmark}_{det}$.

$$\begin{aligned} norm_{\checkmark}(\mathbf{0}) &:= \varnothing \\ norm_{\checkmark}(\mathbf{1}) &:= \{\mathbf{1}; \mathbf{1}\} \\ norm_{\checkmark}(e) &:= \{\mathbf{1}; e\} \\ norm_{\checkmark}(t\checkmark) &:= \{t'; e \mid t'; e \in norm_{\checkmark}(t)\} \cup \{(\bigsqcup \{t'; e \mid t'; e \in norm_{\checkmark}(t)\})\checkmark; \mathbf{1}\} \\ norm_{\checkmark}(s; t) &:= \{s'; e \mid s'; e \in norm_{\checkmark}(s)\} \\ &\quad \cup \{(\bigsqcup norm_{\checkmark}(s'; t'))\checkmark; t'' \mid s'; \mathbf{1} \in norm_{\checkmark}(s), t'\checkmark; t'' \in norm_{\checkmark}(t)\} \\ norm_{\checkmark}(s \sqcup t) &:= norm_{\checkmark}(s) \cup norm_{\checkmark}(t) \ . \end{aligned}$$

5.9 Example. Let $t = (a \sqcup b)\checkmark \sqcup b\checkmark; a$ be the "parsimonious" term of Example 5.5; then $norm_{\checkmark}(t) = \{\mathbf{1}; a, \mathbf{1}; b, (\mathbf{1}; a \sqcup \mathbf{1}; b)\checkmark; \mathbf{1}, (\mathbf{1}; b)\checkmark; a\}$. Note that this equals $R_{\checkmark}([\![t]\!])$; indeed, one can prove that $norm_{\checkmark}(t) = R_{\checkmark}([\![t]\!])$ for all $t \in T^{\checkmark}_{det}(\mathbf{E})$.

The following lemma states that $norm_{\checkmark}(t)$ gives rise to a normal form representation of t. It is analogous to Lemma 3.9 and proved by induction on the structure of t.

5.10 Lemma (✓-normal forms exist). For all $t \in T^{\checkmark}_{det}(\mathbf{E})$, $\bigsqcup norm_{\checkmark}(t)$ is a ✓-normal form such that $\mathrm{A}^{\checkmark}_{det} \vdash t = \bigsqcup norm_{\checkmark}(t)$.

Using exactly the same proof strategy as for the previous completeness results, Lemma 5.8 and Lemma 5.10 together the following theorem.

5.11 Theorem ($\mathrm{A}^{\checkmark}_{det}$ is complete). For all $s, t \in T^{\checkmark}_{det}(\mathbf{E})$, $[\![s]\!] = [\![t]\!]$ implies $\mathrm{A}^{\checkmark}_{det} \vdash s = t$.

5.2 ω-completeness of $\mathrm{A}^{\checkmark}_{det}$

Again, we do not just have completeness but in fact ω-completeness. Proving it requires yet another normal form, defined as follows.

5.12 Definition (open ✓-normal forms). Consider the following production rules for terms of $T^{\checkmark}_{det}(\mathbf{E},\mathbf{X})$:

$$\begin{array}{rcl} M & := & (\bigsqcup \textbf{sat'd set of } N)\checkmark;\mathbf{1} \mid (\bigsqcup \textbf{sat'd set of } N)\checkmark;(x;M) \ . \\ N & := & (\bigsqcup \textbf{sat'd set of } N)\checkmark;e \mid (\bigsqcup \textbf{sat'd set of } N)\checkmark;(x;\mathbf{0}) \mid (\bigsqcup \textbf{sat'd set of } N)\checkmark;(x;N) \end{array}$$

where $e \in \mathbf{E}$, $x \in \mathbf{X}$ and a set T of N-produced terms is *saturated* if $(\bigsqcup T')\checkmark;t' \in T$ implies $T' \subseteq T$ and $s'\checkmark;(x;(\bigsqcup T')\checkmark;t')) \in T$ implies $\{s'\checkmark;(x;t'') \mid t'' \in T' \vee t'' = \mathbf{0}\} \subseteq T$. A term is in *open ✓-normal form* if it equals $\bigsqcup T$ for some saturated set T of M- and N-produced terms.

To generate open ✓-normal forms for arbitrary open $\mathrm{A}^{\checkmark}_{det}$-terms, the function $norm_{\checkmark}$ defined above is modified and extended by the following rules:

$$\begin{array}{rcl} norm_{\checkmark}(x) & := & \{\mathbf{1};(x;\mathbf{1})\} \\ norm_{\checkmark}(s;t) & := & \{s';e \mid s';e \in norm_{\checkmark}(s)\} \\ & & \cup\, \{(\bigsqcup norm_{\checkmark}(s';t'))\checkmark;t'' \mid s';\mathbf{1} \in norm_{\checkmark}(s), t'\checkmark;t'' \in norm_{\checkmark}(t)\} \\ & & \cup\, \{s';(x;t') \mid s';(x;s'') \in norm_{\checkmark}(s), t' \in norm_{\checkmark}(s'';t)\} \ . \end{array}$$

5.13 Example. If $t = x;a\checkmark;(a\checkmark;y \sqcup (x;c)\checkmark)$ then

$$norm_{\checkmark}(t) = \left\{ \begin{array}{l} \mathbf{1};(x;\mathbf{0}) \\ \mathbf{1};(x;(\mathbf{1};a)) \\ \mathbf{1};(x;((\mathbf{1};a)\checkmark;a) \\ \mathbf{1};(x;(((\mathbf{1};a)\checkmark;a)\checkmark;(y;\mathbf{1})) \\ \mathbf{1};(x;((\mathbf{1};a)\checkmark;(x;\mathbf{0})) \\ \mathbf{1};(x;((\mathbf{1};a)\checkmark;(x;c)) \\ \mathbf{1};(x;((\mathbf{1};a)\checkmark;((x;c)\checkmark;\mathbf{1})) \end{array} \right\} \ .$$

The function $norm_{\checkmark}$ gives rise to the following extension of Lemma 5.10:

5.14 Lemma (open ✓-normal forms exist). For all $t \in T^{\checkmark}_{det}(\mathbf{X})$, $\bigsqcup norm_{\checkmark}(t)$ is an open ✓-normal form such that $\mathrm{A}^{\checkmark}_{det} \vdash t = \bigsqcup norm_{\checkmark}(t)$.

To show the normality of open ✓-normal forms, we once more need a notion of characteristic pomsets as in Definition 4.15. Fortunately, a slight variation will do: the only difference is that previously we had to take into account that any element could be refined, and hence the characteristic vertices had to contain images of all those refinements. In $\mathrm{A}^{\checkmark}_{det}$ however, the only element that can be refined is $\mathbb{T}$, and hence just one image has to be recorded. This gives rise to the following definition:

5.15 Definition (✓-characteristic pomsets). Let $E_{s,t}$ and $\{d_x, e_x\}_{x\in\mathbf{X}}$ be disjoint sets of distinct elements. A pomset p is called *✓-characteristic* if for all $v \in V_p \setminus \ell_p^{-1}(E_{s,t})$:

- if $\ell_p(v) = d_x$ then the set of *characteristic vertices* $C_v \subseteq V_p$ defined by

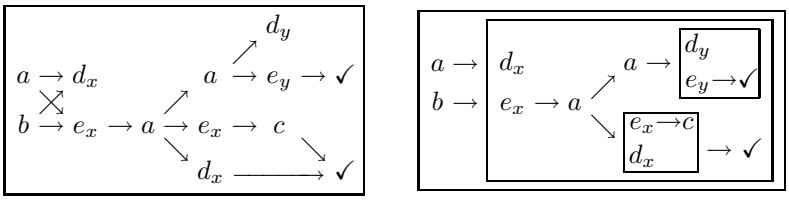
$$C_v := \{w \in V_p \mid \forall u \in V_p.\ (u <_p v \Rightarrow u <_p w) \wedge (u >_p v \Rightarrow u >_p w)\}$$

  has the property that for all $w \in C_v$ and $u \in V_p \setminus C_v$, $u <_p w$ implies $u <_p v$ and $u >_p w$ implies $u >_p v$. Furthermore, $p \upharpoonright C_v = d_x \sqcup (e_x \cdot p)$, where p is a characteristic pomset sometimes denoted $C_v(\mathbb{T})$.
- if $\ell_p(w) = e_x$ there is a $v \in V_p$ such that $\ell_p(v) = d_x$ and $w \in C_v$.

The corresponding ✓-characteristic ground substitutions $\rho_{s,t}$ are given by $x \mapsto d_x \sqcup e_x\checkmark$ for all $x \in \mathbf{X}$. We then have the counterpart of Lemma 4.17:

5.16 Lemma. For all $s, t \in T^{\checkmark}_{det}(\mathbf{X})$, $[\![t\rho_{s,t}]\!]$ is a ✓-characteristic pomset.

5.17 Example. Applying a characteristic substitution to the term $(a \sqcup b)\checkmark; (x; a\checkmark; (a\checkmark; y \sqcup (x; c)\checkmark)$ yields the left-hand pomset below. The right-hand side shows the principal subpomsets.



Continuing the developments in an analogous fashion, we extend $R_{\checkmark}$ to deal with ✓-characteristic pomsets:

$$\begin{aligned} R_{\checkmark}(p) \quad := \quad & \{(\bigsqcup R_{\checkmark}(p \upharpoonright \Downarrow v))\checkmark; \mathbf{1} \mid v \in VP, \ell(v) = \mathbb{T}\} \cup \\ & \{(\bigsqcup R_{\checkmark}(p \upharpoonright \Downarrow v))\checkmark; \ell(v) \mid v \in VP, \ell(v) \notin \{d_x, \mathbb{T}\}\} \cup \\ & \{(\bigsqcup R_{\checkmark}(p \upharpoonright \Downarrow v))\checkmark; (x; \mathbf{0}) \mid v \in VP, \ell(v) = d_x, C_v(\mathbb{T}) = \varepsilon\} \cup \\ & \{(\bigsqcup R_{\checkmark}(p \upharpoonright \Downarrow v))\checkmark; (x; t) \mid v \in VP, \ell(v) = d_x, t \in R_{\checkmark}(C_v(\mathbb{T}))\} \ . \end{aligned}$$

Again, it can be established that $R_{\checkmark}$ gives rise to the left inverse of the semantic function applied after a ✓-characteristic ground substitution:

5.18 Lemma (open ✓-normal forms are unique). If $s, t \in T^{\checkmark}_{det}(\mathbf{X})$ are open ✓-normal form terms, then $\bigsqcup R_{\checkmark}([\![t\rho_{s,t}]\!]) = t$.

Lemma 5.14 and Lemma 5.18 allow us to prove, in the now familiar way, the ω-completeness of $\mathrm{A}^{\checkmark}_{det}$:

5.19 Theorem ($\mathrm{A}^{\checkmark}_{det}$ is ω-complete). Assume $|\mathbf{E}| = \omega$. If for all $s, t \in T^{\checkmark}_{det}(\mathbf{E}, \mathbf{X})$, if $[\![s\rho]\!] = [\![t\rho]\!]$ for all ground substitutions $\rho\colon \mathbf{X} \to T^{\checkmark}_{det}(\mathbf{E})$ then $\mathrm{A}^{\checkmark}_{det} \vdash s = t$.

5.3 Global termination: the algebra $\mathrm{A}^{\delta}_{det}$

As an alternative to the notion of distributed termination, one may prefer to think of termination as a *global* property. This involves a strict division of the pomsets into terminated and nonterminated ones. An effective model is obtained by extending every pomset with a single boolean value indicating if it is terminated or not; hence we have objects $\pi = \langle V, <, \ell, \checkmark\rangle$ where $p_\pi = \langle V, <, \ell\rangle$ is a pomset and $\checkmark \in \{\mathbf{t}, \mathbf{f}\}$. The set of all such objects is denoted $\mathbf{DPOM}^{\delta}$ (for reasons that will become apparent below). We also write $\langle p, \checkmark\rangle$ to denote $\langle V_p, <_p, \ell_p, \checkmark\rangle$. Pictorially, appending ✓ to a pomset signifies that it is terminated, while the absence of ✓ denotes the opposite

5.20 Example. $\boxed{\begin{smallmatrix} & b \\ \nearrow & \\ a & \to c \end{smallmatrix}}\checkmark$ is globally terminated but $\boxed{\begin{smallmatrix} a & \\ & \searrow \\ b & \to c \end{smallmatrix}}$ is not.

In sequential composition, $\pi; \kappa = \pi$ if π is not terminated, whereas otherwise $\pi; \kappa$ corresponds to the concatenation of (the pomsets within) π and κ, which is terminated iff κ is.

Immediately the question rises what effect the join operator should have on the termination of its parameters, in particular, what should happen when a terminated pomset is joined with a nonterminating one. The most natural answer to this question would seem to be that since termination is global, for the parallel composition of two subpomsets to terminate they should both

terminate. Joining two pomsets is not quite the same as composing them in parallel, witness e.g. the equation $x \sqcup x{\cdot}y = x{\cdot}y$; still, this is the solution we will work out here. Consider the following set of equations over the signature $\Sigma^{\delta}_{det} = \langle \varepsilon, \delta, ;, \sqcup \rangle$.

$$\begin{aligned}
\varepsilon; x &= x && (55)\\
x; \varepsilon &= x && (56)\\
(x; y); z &= x; (y; z) && (57)\\
\varepsilon \sqcup x &= x && (58)\\
(x \sqcup y) \sqcup z &= x \sqcup (y \sqcup z) && (59)\\
x \sqcup y &= y \sqcup x && (60)\\
x; (y \sqcup z) &= x; y \sqcup x; z && (61)\\
x; \delta &= x \sqcup \delta && (62)\\
\delta; x &= \delta \ . && (63)
\end{aligned}$$

The resulting theory will be denoted $\mathrm{A}^{\delta}_{det}$. Note that just as in A_{det} but other than in $\mathrm{A}^{\checkmark}_{det}$, the neutral elements of the submonoids $\langle \varepsilon, ; \rangle$ and $\langle \varepsilon, \sqcup \rangle$ coincide, and therefore the distributivity property (61) implies idempotence of join, i.e., $x = x \sqcup x$ is derivable. The most curious of the $\mathrm{A}^{\delta}_{det}$-equations is probably (62), which expresses that appending δ has the same effect as joining δ: both of them make the pomset deadlocking. The intended semantics of the terms of $\mathrm{A}^{\delta}_{det}$ is given inductively as follows:

$$\begin{aligned}
[\![\delta]\!] &:= \langle \varnothing, \varnothing, \varnothing, \mathbf{f} \rangle\\
[\![\varepsilon]\!] &:= \langle \varnothing, \varnothing, \varnothing, \mathbf{t} \rangle\\
[\![e]\!] &:= \langle \{0\}, \varnothing, \{(0, e)\}, \mathbf{t} \rangle\\
[\![\pi; \kappa]\!] &:= \begin{cases} \pi & \text{if } \checkmark_\pi = \mathbf{f} \\ \langle p_\pi{\cdot}p_\kappa, \checkmark_\kappa \rangle & \text{if } \checkmark_\pi = \mathbf{t} \end{cases}\\
[\![\pi \sqcup \kappa]\!] &:= \langle p_\pi \sqcup p_\kappa, \checkmark_\pi \wedge \checkmark_\kappa \rangle \ .
\end{aligned}$$

It follows that A_{det} corresponds to the fragment of $\mathrm{A}^{\delta}_{det}$ without the constant δ, such that the pomsets described by this fragment are all terminated. Now let us investigate to what degree we have reached our goal of making sequential composition monotonic w.r.t. pomset prefix.

5.4 Pomset prefix with global termination

We have yet to extend the prefix relation to $\mathbf{DPOM}^{\delta}$. We will denote the extended relation $\preccurlyeq$, to distinguish it from ordinary pomset prefix. We certainly expect the extension to be consistent with pomset prefix, i.e., the following should hold:

$$\pi \preccurlyeq \kappa \Longrightarrow p_\pi \sqsubseteq p_\kappa \ .$$

In addition however, we want sequential composition to be monotonic w.r.t. $\preccurlyeq$ in its left operand; indeed, that was our reason for introducing the concept of termination in the first place. Now if we define $x \preccurlyeq y :\Leftrightarrow x \sqcup y = y$ as before, then it would follow that $\boxed{a}\checkmark \preccurlyeq \boxed{a{\to}b}\checkmark$ and hence $\boxed{a{\to}c} = \boxed{a}\checkmark;\boxed{c} \preccurlyeq \boxed{a{\to}b}\checkmark;\boxed{c} = \boxed{a{\to}b{\to}c}$ due to monotonicity; this however contradicts the consistency requirement above.

In fact, the only way to combine the criteria of monotonicity w.r.t. sequential composition and consistency with pomset prefix is to require that terminated pomsets are $\preccurlyeq$-maximal. This then gives rise to the following definition:

$$\pi \preccurlyeq \kappa \iff (p_\pi \sqsubseteq p_\kappa \wedge \neg\checkmark_\pi) \vee (\pi = \kappa) \ .$$

It follows that if we embed A_{det} in $\mathrm{A}^{\delta}_{det}$ in the manner suggested above, by letting the pomsets of **DPOM** correspond to the terminated pomsets in $\mathbf{DPOM}^{\delta}$, then $\preccurlyeq$ and $\sqsubseteq$ are really quite different relations: for instance, with respect to the former all terminated pomsets are incomparable. Still, the following property carries over from $\sqsubseteq$ over **DPOM** to $\preccurlyeq$ over $\mathbf{DPOM}^{\delta}$ (compare Corollary 2.20):

5.21 Theorem. Every $\preccurlyeq$-left-closed subset of $\langle \mathbf{DPOM}^{\delta}, \preccurlyeq \rangle$ is a prime algebraic basis, where the primes are (i) the nonterminated pomsets with a top element and (ii) the terminated pomsets.

Proof sketch. One way to prove prime algebraicity is to prove the following distributivity property (cf. Section 2.3), which should hold iff $\pi \curlyvee \kappa$ is defined:

$$(\pi \curlyvee \kappa) \curlywedge \lambda = (\pi \curlywedge \lambda) \curlyvee (\kappa \curlywedge \lambda)$$

where $\curlyvee$ and $\curlywedge$ denote the supremum resp. infimum w.r.t. $\preccurlyeq$, defined by:

$$\pi \curlyvee \kappa \quad := \quad \begin{cases} \pi & \text{if } \pi = \kappa \\ \langle p_\pi \sqcup p_\kappa, \checkmark_\pi \vee \checkmark_\kappa \rangle & \text{if } \neg(\checkmark_\pi \wedge \checkmark_\kappa) \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$\pi \curlywedge \kappa \quad := \quad \begin{cases} \pi & \text{if } \pi = \kappa \\ \langle p_\pi \sqcap p_\kappa, \mathbf{f} \rangle & \text{otherwise.} \end{cases}$$

The actual proof that these are indeed the supremum and infimum, and that the above distributivity property holds, is straightforward and omitted here. □

In terms of the $\mathrm{A}^{\delta}_{det}$-theory, the new prefix relation is defined by

$$x \preccurlyeq y \;:\Leftrightarrow\; (x = x;\delta \wedge x \sqcup y = \delta \sqcup y) \vee (x = y) \;\; .$$

The desired monotonicity properties are then provable in $\mathrm{A}^{\delta}_{det}$, as the following proposition shows.

5.22 Proposition. $\preccurlyeq$ is a partial ordering relation such that δ is the smallest element, all terminated objects are maximal, and $\sqcup$ and ; are monotonic in both operands.

Proof. Transitivity: assume $x \preccurlyeq y \preccurlyeq z$. The interesting case is $x \neq y \neq z$; then

$$\mathrm{A}^{\delta}_{det} \vdash x = x;\delta,\;\; x \sqcup z = x;\delta \sqcup z = x \sqcup (\delta \sqcup z)) = (x \sqcup y) \sqcup z = \delta \sqcup (y \sqcup z) = \delta \sqcup z \;\; .$$

Antisymmetry: assume $x \preccurlyeq y \preccurlyeq x$ and $x \neq y$; then

$$\mathrm{A}^{\delta}_{det} \vdash x = x;\delta = x \sqcup \delta = x \sqcup y = y \sqcup \delta = y;\delta = y$$

contradicting $x \neq y$; hence it must be the case that $x = y$. δ is the smallest element:

$$\mathrm{A}^{\delta}_{det} \vdash \delta = \delta;\delta,\;\; \delta \sqcup x = \delta \sqcup x \;\; .$$

x is terminated iff $x;\delta \neq x$, hence $x \sqsubseteq y$ implies $x = y$, hence every terminated object is maximal. As for monotonicity: assume that $x \preccurlyeq y$; the interesting case is $x \neq y$, hence $x;\delta = x$ and $x \sqcup y = \delta \sqcup y$. Monotonicity in the right hand operand of sequential composition:

$$\mathrm{A}^{\delta}_{det} \vdash z;x = (z;x);\delta,\;\; z;x \sqcup z;y = z;(x \sqcup y) = z;(\delta \sqcup y) = (z;y);\delta = \delta \sqcup z;y \;\; .$$

Monotonicity in the left hand operand of sequential composition:

$$\begin{aligned} \mathrm{A}^{\delta}_{det} \;&\vdash\; x;z = x;(\delta;z) = x;\delta = x;(\delta;(z;\delta)) = (x;z);\delta \\ \mathrm{A}^{\delta}_{det} \;&\vdash\; x;z \sqcup y;z = x \sqcup (y;\delta \sqcup y;z) = (x \sqcup y) \sqcup (\delta \sqcup y;z) = \delta \sqcup (y;\delta \sqcup y;z) = \delta \sqcup y;z \;\; . \end{aligned}$$

Monotonicity w.r.t. $\sqcup$ is immediate. □

5.5 Soundness and completeness of $\mathrm{A}^{\delta}_{det}$

For the theory $\mathrm{A}^{\delta}_{det}$ we have the usual soundness and completeness properties, which we state here without proof:

5.23 Theorem ($\mathrm{A}^{\delta}_{det}$ is sound and complete). For all $s, t \in T^{\delta}_{det}(\mathbf{E})$, $\mathrm{A}^{\delta}_{det} \vdash s = t$ iff $[\![s]\!] = [\![t]\!]$.

The normal forms used to prove completeness equal the standard normal forms (Definition 3.6) with an optional additional join-component δ; that is, δ-normal forms are $\bigsqcup T[\cup\{\delta\}]$ where $\bigsqcup T$ is a normal form term. The presence of such a δ-component indicates that the term is not terminated. To create denotations for δ-pomsets we use a slight variation on R, which maps to ';' rather than '$\cdot$' and is defined by:

$$R_\delta(\pi) := \begin{cases} R(p_\pi) & \text{if } \checkmark_\pi \\ R(p_\pi) \cup \{\delta\} & \text{otherwise.} \end{cases}$$

To normalise terms we use a function $norm_\delta$ which essentially equals $norm$, except in dealing with δ-components.

$$\begin{aligned}
norm_\delta(\varepsilon) &:= \varnothing \\
norm_\delta(\delta) &:= \{\delta\} \\
norm_\delta(e) &:= \{\varepsilon \cdot e\} \\
norm_\delta(s;t) &:= norm_\delta(s) \\
&\quad \cup (norm_\delta(t) \cap \{\delta\}) \\
&\quad \cup \{(\textstyle\bigsqcup norm_\delta(s;t')); e \mid \delta \notin norm_\delta(s), t'; e \in norm_\delta(t)\} \\
norm_\delta(s \sqcup t) &:= norm_\delta(s) \cup norm_\delta(t) \ .
\end{aligned}$$

The following are the required lemmas to prove completeness.

5.24 Lemma (δ-normal forms exist). For all $t \in T^{\delta}_{det}$, $\bigsqcup norm_\delta(t)$ is a δ-normal form such that $\mathrm{A}^{\delta}_{det} \vdash t = \bigsqcup norm_\delta(t)$.

5.25 Lemma (δ-normal forms are unique). $\bigsqcup R_\delta([\![t]\!]) = t$ for all δ-normal forms t.

As in the case of A_{det} (i.e., without refinement), open δ-normal forms can be formulated by simply allowing variables to occur at every position where ordinary δ-normal forms allow elements; in other words, they equal $\bigsqcup T[\cup\{\delta\}]$ where $\bigsqcup T$ is in open normal form (see Definition 3.12). The corresponding characteristic ground substitutions simply map variables to distinct fresh elements. Since $\mathrm{A}^{\delta}_{det}$ contains no equations which deal specifically with elements, all proofs carry over smoothly from the ground case. Hence again we do not only have completeness but in fact ω-completeness:

5.26 Theorem ($\mathrm{A}^{\delta}_{det}$ is ω-complete). Assume $|\mathbf{E}| = \omega$. For all $s, t \in T^{\delta}_{det}(\mathbf{E}, \mathbf{X})$, if $[\![s\rho]\!] = [\![t\rho]\!]$ for all ground substitutions $\rho\colon \mathbf{X} \to T^{\delta}_{det}(\mathbf{E})$ then $\mathrm{A}^{\delta}_{det} \vdash s = t$.

6 Concluding remarks

It remains to summarise the results of this paper, to compare them in somewhat more detail with existing work, and to discuss extensions and future work.

6.1 Summary

We have introduced the class of *deterministic pomsets*, and have shown that this class satisfies the following properties:

- Deterministic pomsets arise as a generalisation of strings, by freely adding objects corresponding to the prefix-suprema of arbitrary finite sets of strings.
- The class of deterministic pomsets forms a distributive basis with all finite suprema; hence prefix-closed sets of pomsets form prime algebraic bases.
- Given an appropriate notion of (prefix-preserving) lposet morphisms, deterministic lposets form a reflective subcategory of the lposets.

We have then formulated an algebra of deterministic pomsets by algebraising the supremum of pairs of such pomsets, resulting in an operator for *pomset join*. Pomset join is a slight variation on pomset disjoint union: both can be defined by the union of lposet representatives, the only difference being the choice of representatives, which for disjoint union have to be *disjoint* in their sets of vertices, but for join should coincide precisely on isomorphic prefixes.

Based on pomset join, we have developed several algebraic theories, all of which have been proved sound and complete, and ω-complete in the presence of sufficiently many elements:

- A_{det} (see Section 3), consisting of the signature $\Sigma_{det} = \langle \varepsilon, \cdot, \sqcup \rangle$ and equations (25)–(31) (see Page 7). ε is the *empty pomset*, $\cdot$ is *concatenation* of pomsets, and $\sqcup$ denotes pomset join. Models are (unadorned) deterministic pomsets.
- A^*_{det} (see Section 4.3), extending A_{det} with a notion of *refinement* which basically algebraises *homomorphism application*: signature $\Sigma^*_{det} = \langle \varepsilon, \cdot, \sqcup, * \rangle$ and equations (25)–(31) (Page 7) and (32)–(37) (Page 25). Models are deterministic pomsets and finite refinement functions mapping elements to deterministic pomsets (*finite* meaning that they are the identity except on a finite number of elements).
- $\mathrm{A}^{\checkmark}_{det}$ (see Section 5.1), specialising A^*_{det} to obtain a notion of *distributed termination* through special elements $\mathbb{T}$ that denote *exit points* and may only appear as maximal elements; sequential composition corresponds to the refinement of the $\mathbb{T}$-elements. The corresponding signature is given by $\Sigma^{\checkmark}_{det} = \langle \mathbf{0}, \mathbf{1}, \checkmark, ;, \sqcup \rangle$ with equations (45)–(53) (Page 30): $\mathbf{0}$ denotes deadlock, $\mathbf{1}$ successful termination, $\checkmark$ is a unary operator appending $\mathbb{T}$ to a pomset, and ; denotes sequential composition. Models are deterministic pomsets with special $\mathbb{T}$-elements.
- $\mathrm{A}^{\delta}_{det}$ (see Section 5.3), extending A_{det} to obtain a notion of *global termination* by appending a boolean value to the pomsets denoting whether they are terminated or not. The signature is given by $\Sigma^{\delta}_{det} = \langle \varepsilon, \delta, ;, \sqcup \rangle$ with equations (55)–(63) (Page 35). Models are deterministic pomsets extended with a boolean component denoting whether or not they are terminated.

6.2 Related work

In the course of the paper we have already given a fairly detailed comparison with existing work on *series-parallel* pomsets, based as it is on the *disjoint union* of pomsets rather than pomset join. Relevant papers are for instance (in order of appearance) Grabowski [10], Jónsson [13], Pratt [19], Gischer [9] and Aceto [2].

One important point of difference that has not been stressed so far is the following: *pomset join is only partially defined*, namely only between pomsets which have compatible representatives (see Section 2.2); these are in fact precisely the deterministic pomsets. Hence although within the class of deterministic pomsets we have very satisfactory results, they appear to be difficult to extend to larger classes. This contrasts with disjoint union, which is totally defined on **POM**.

Another point of difference is that where we have concentrated on a small number of operators —basically pomset join, refinement, and sequential composition— the existing theory of series-parallel pomsets is much more extensive, covering many operators and considering *sets* of pomsets as well as single pomsets.

All other things being equal, the principal difference between the two theories, series-parallel versus deterministic, is in the class of pomsets for which they are complete. These classes are incomparable: for instance, $\boxed{\begin{array}{c} a \rightarrow c \\ \nearrow \\ b \rightarrow d \end{array}}$ is not series-parallel whereas $\boxed{\begin{array}{c} a \\ a \end{array}}$ is not deterministic. Any question concerning which of the two is the more appropriate can therefore only be answered in the context of some specific application.

Another well-developed theory of pomsets, which has received somewhat short shrift here, is that of *Mazurkiewicz traces*; good references are [1, 15]. As we have remarked in the introduction, all Mazurkiewicz traces are in fact deterministic pomsets, and some of the facts proved for deterministic pomsets in this paper constitute a proper generalisation of known Mazurkiewicz trace theory; in particular the fact that prefix closed sets of Mazurkiewicz traces form prime algebraic bases (see e.g. Nielsen, Sassone and Winskel [18], where it is in fact proved for the intermediate class of *pomsets without auto-concurrency*, which is properly in between the Mazurkiewicz traces and the deterministic pomsets). However, the concept of a *concurrent alphabet* which is central to Mazurkiewicz trace theory, and underlies the associated operators (especially concatenation), is totally absent from this paper, and indeed the actual algebraic theories have little in common.

The final related field we wish to mention here is the theory of *trees*, as developed especially in the context of process algebra (see e.g. [3] for a good exposition of the algebraic side), but also in a different setting for instance in [8]. There are in fact two ways in which tree may be related to pomsets: trees can either be directly regarded as pomsets themselves, with a specific condition on the ordering relation according to which all predecessors of a given vertex must be totally ordered; or they may be regarded as *prefix closed sets* of pomsets, which for the specific case of trees are then in fact prefix closed sets of *total* orders.

In the first interpretation, note that the deterministic pomsets in fact correspond to *deterministic trees*, and pomset join merges trees from their roots up to the first branch where they differ. However, pomset concatenation would not in general correspond to a very useful operator since it very easily leads outside the class of trees. There are a number of variations on this theme—for instance, one may choose to read pomsets *backwards* to obtain trees, which gets rid of the restriction to deterministic trees: for the finite models we have studied here this in fact yields a fully abstract model with respect to *strong bisimulation*, which has been studied e.g. by Rutten in [21]; however, due to the reversal in the interpretation, the extension to infinite trees requires non-well-founded pomsets. Another variation is to introduce exit points, much as we have done in $\mathrm{A}^{\surd}_{det}$; in fact the fragment of $\mathrm{A}^{\surd}_{det}$ excluding the termination operator $\surd$ constitutes a complete theory of deterministic trees.

The second interpretation is the one propagated by De Nicola and Labella in [8]. For an exhaustive comparison with the results of this paper, one would have to investigate the theory of prefix closed sets of A_{det}-pomsets; we briefly discuss this below as a possible extension. One observation that can be made right away, however, is that such an extension of A_{det} once more would be enable to describe only *deterministic* trees.

6.3 Extensions

We briefly review three interesting directions in which the results of this paper may be extended.

Infinite pomsets A straightforward extension is to consider *infinite* as well as finite pomsets. In fact all the theory developed in this paper extends smoothly to this more general case if we introduce *infinitary joins*. The relevant models are the *well-founded deterministic pomsets*. These form a proper class, which may be seen as a direct generalisation of the *ordinals* in which there exist, instead of a single successor function, a family of different ones (one for each element in $\mathbf{E}$). The resulting (infinitary) theory allows the solution of A_{det}-equations, as briefly hinted at in Section 5. A detailed discussion is outside the scope of this paper.

Augmentation Apart from the prefix relation, which we have studied in considerable detail here, there is another relation over pomsets that has received much attention in the literature, viz. that of *augmentation*; see for instance the papers on series-parallel pomsets cited above.

Basically, a pomset is said to augment another if it contains strictly more ordering but is the same otherwise. Currently we do not have any general results tying this relation into the framework of this paper. However, if we restrict our attention to *posets* rather than pomsets (which can be regarded as pomsets with an injective labelling function) then the following may be established: the smallest partial ordering relation over posets including prefix and *inverse* augmentation coincides with the *finest* pre-congruence with respect to join and concatenation that subsumes prefix: in other words, it is the smallest transitive relation $\leq$ over pomsets such that $p \sqsubseteq q$ implies $p \leq q$ and

$$p_1 \leq p_2 \Longrightarrow (p_1 \sqcup q \leq p_2 \sqcup q) \wedge (p_1; q \leq p_2; q) \wedge (q; p_1 \leq q; p_2)$$

where p_1, p_2 and q are arbitrary posets. For instance, pre-congruence allows to derive $\boxed{a \to b} \leq \boxed{\begin{array}{c} a \\ \searrow \\ c \to b \end{array}}$ from $\boxed{a} \sqsubseteq \boxed{\begin{array}{c} a \\ c \end{array}}$, and indeed it holds that $\boxed{a \to b} \sqsubseteq \boxed{\begin{array}{l} a \to b \\ c \end{array}}$ which is an inverse augmentation of $\boxed{\begin{array}{c} a \\ \searrow \\ c \to b \end{array}}$. This result is not directly useful however, since due to the inversion of the augmentation relation, left-closure with respect to $\leq$ would correspond to augmentation right-closure rather than left-closure. We have not pursued this matter further.

Prefix ideals In Gischer [9], an important role is played by *augmentation left-closed* sets of pomsets, which he calls (augmentation) *ideals*. An analogous extension that we intend to study in the future is to consider *prefix closed sets of pomsets* as models; one might call such sets *prefix ideals*. A theory of prefix ideals can be obtained by interpreting the constants we had as prefix ideals—in particular, letting each $e \in \mathbf{E}$ correspond to the set containing all prefixes of e— and introducing a union-like operator $+$, which may be thought of as modelling *choice*.

For instance, if we take $\mathrm{A}^{\delta}_{det}$ (see Section 5) as the basis for such an extension (meaning that we should use $\preccurlyeq$ rather than $\sqsubseteq$ as our prefix relation), then $+$ is captured equationally by the following properties.[5]

$$\begin{array}{rcl} \delta + x & = & x \\ x + y & = & y + x \\ (x + y) + z & = & x + (y + z) \\ x + x & = & x \\ x; (y + z) & = & x; y + x; z \\ (x + y); z & = & x; z + y; z \\ x \sqcup (y + z) & = & (x \sqcup y) + (x \sqcup z) \ . \end{array}$$

In other words, we obtain a third monoid, whose neutral element δ equals the constant we had introduced in Section 5.3 to model *deadlock*, and whose operator allows all others to distribute over it. It follows from these distribution properties that the ordering relation defined by

$$x \leq y \ :\Leftrightarrow \ x + y = y$$

(which corresponds to the *subset relation* over prefix ideals) is a pre-congruence for all the operators in the algebra. This corresponds to the property that our operators are *monotonic*, but they are in fact even *continuous* with respect to $\leq$ (to express this equationally would require the introduction of infinitary sums).

Moreover, it is a standard result that the union and intersection of arbitrary collections of ideals are ideals, hence the space of models is automatically seen to be a complete partial order under $\leq$

[5] Note this operator is entirely analogous to the one described in e.g. Gischer [9] for arbitrary sets of processes.

(in fact a complete lattice), allowing the construction of fixpoints. (Note that the limit points of this lattice are *infinite* sets of *finite* pomsets, hence we do not here need the extension to infinite pomsets mentioned above). In combination with the continuity of our operators, it follows that all context-free sets of equations give rise to standard minimal (or maximal) fixpoint solutions.

The resulting theory is actually closely related to the area of *causality-based semantics*, especially *event structures* as studied by e.g. Winskel in [23] and Boudol and Castellani in [4]. The connection is made through the theory of *prime algebraic bases* we have briefly discussed in Section 2.3. In fact, in the latter paper one may find the basic ideas of an algebra much like the one we have developed here.

References

[1] IJ. J. Aalbersberg and G. Rozenberg. Theory of traces. *Theoretical Comput. Sci.*, 60:1–82, 1988.

[2] L. Aceto. Full abstraction for series-parallel pomsets. In S. Abramsky and T. S. E. Maibaum, editors, *TAPSOFT '91, Volume 1*, volume 493 of *Lecture Notes in Computer Science*, pages 1–25. Springer-Verlag, 1991.

[3] J. C. M. Baeten and W. P. Weijland. *Process Algebra*. Cambridge University Press, 1990.

[4] G. Boudol and I. Castellani. On the semantics of concurrency: Partial orders and transition systems. In H. Ehrig, R. Kowalski, G. Levi, and U. Montanari, editors, *TAPSOFT '87, Volume 1*, volume 249 of *Lecture Notes in Computer Science*, pages 123–137. Springer-Verlag, 1987.

[5] G. Boudol and I. Castellani. Permutations of transitions: An event structure semantics for CCS and SCCS. In de Bakker et al. [7], pages 411–427.

[6] A. Corradini, H. Ehrig, M. Löwe, U. Montanari, and F. Rossi. An event structure semantics for safe graph grammars. In E.-R. Olderog, editor, *Programming Concepts, Methods and Calculi*, volume A–56 of *IFIP Transactions*, pages 423–446. IFIP, 1994.

[7] J. W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors. *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, volume 354 of *Lecture Notes in Computer Science*. Springer-Verlag, 1989.

[8] R. De Nicola and A. Labella. A completeness theorem for nondeterministic Kleene algebras. In: *Proceedings of MFCS*, 1994.

[9] J. L. Gischer. The equational theory of pomsets. *Theoretical Comput. Sci.*, 61:199–224, 1988.

[10] J. Grabowski. On partial languages. *Fundamenta Informaticae*, IV(2):427–498, 1981.

[11] J. F. Groote. *Process Algebra and Structured Operational Semantics*. PhD thesis, University of Amsterdam, 1991.

[12] J. Heering. Partial evaulation and ω-completeness of algebraic specifications. *Theoretical Comput. Sci.*, 43:149–167, 1986.

[13] B. Jónsson. Arithmetic of ordered sets. In I. Rival, editor, *Ordered Sets*, pages 3–41. Reidel Pub. Co., 1982.

[14] A. Lazrek, P. Lescanne, and J.-J. Thiel. Tools for proving inductive equalities, relative completeness, and ω-completeness. *Information and Computation*, 84:47–70, 1990.

[15] A. Mazurkiewicz. Basic notions of trace theory. In de Bakker et al. [7], pages 285–363.

[16] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[17] M. Nielsen, U. Engberg, and K. G. Larsen. Fully abstract models for a process language with refinement. In de Bakker et al. [7], pages 523–549.

[18] M. Nielsen, V. Sassone, and G. Winskel. Relationships between models for concurrency. In J. W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *A Decade of Concurrency*, volume 803 of *Lecture Notes in Computer Science*, pages 425–476. Springer-Verlag, 1994.

[19] V. R. Pratt. Modeling concurrency with partial orders. *International Journal of Parallel Programming*, 15(1):33–71, 1986.

[20] A. Rensink. Posets for configurations! In W. R. Cleaveland, editor, *Concur '92*, volume 630 of *Lecture Notes in Computer Science*, pages 269–285. Springer-Verlag, 1992.

[21] J. J. M. M. Rutten. Processes as terms: Non well-founded models for bisimulation. *Mathematical Structures in Computer Science*, 2:257–275, 1922.

[22] G. Winskel. Event structures. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Petri Nets: Applications and Relationships to Other Models of Concurrency*, volume 255 of *Lecture Notes in Computer Science*, pages 325–392. Springer-Verlag, 1987.

[23] G. Winskel. An introduction to event structures. In de Bakker et al. [7], pages 364–397.