Notre Dame Journal of Formal Logic
Volume 37, Number 2, Spring 1996

Algebra and Theory of Order-Deterministic Pomsets

AREND RENSINK

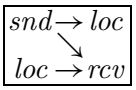
Abstract This paper is about partially ordered multisets (pomsets for short). We investigate a particular class of pomsets that we call *order-deterministic*, properly including all partially ordered *sets*, which satisfies a number of interesting properties: among other things, it forms a *distributive lattice* under pomset prefix (hence prefix closed sets of order-deterministic pomsets are prime algebraic), and it constitutes a *reflective subcategory* of the category of all pomsets. For the order-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. This theory is then extended in order to capture *refinement* of pomsets by incorporating homomorphisms between models as objects in the algebra and homomorphism application as a new operator.

1 Introduction We investigate a class of structures commonly called *partially ordered multisets* (a term proposed by Pratt [19]), or *pomsets* for short. Pomsets are node-labeled directed graphs where the edges constitute an irreflexive and transitive relation (i.e., a partial order) over the nodes, interpreted up to label- and edge-preserving *isomorphism* so that the identity of the nodes (but not their ordering) is abstracted away from. The *multiset* referred to in the term "pomset" is the multiset of *node labels*. Thus, pomsets can be thought of as generalizations of *traces* (also called *strings* or *words*), where the ordering of the nodes is linear.

It has been proposed by many researchers that pomsets can be used as mathematical representations of runs of a concurrent system; see, for instance, Grabowski [8], Pratt [19], Gischer [7], Nielsen et al. [17]. In this representation, the nodes of the graph model activities of the system on some abstract level of description where it is not necessary to model any finer-grained structure within such an activity. The nature of the activity associated to a node is described by an *action name* given as the node label. The edges represent *dependencies* or *causalities* between the activities which are due to, for instance, sequential composition (control flow) or communication (data flow). In this interpretation, activities are unordered, or independent, if they take place concurrently or in different parts of a distributed system and neither

of them uses data generated by the other.

For instance, if a system has a component sending a message and then doing some local activity, and a second component that first does something locally and then receives the message, the corresponding fragment of behavior may be modeled by the following pomset:

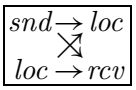


Because according to this point of view, the ordering of the nodes reflects an actual relation between the activities of the system, one might call the model and its underlying interpretation *intensional*. There would seem to be some obvious advantages to such an intensional model: since there is precise information about dependencies, it should be easier to analyze properties of system behavior, or, *vice versa*, to provide a distributed implementation of behavior specified through such a model.

Contrasting the intensional interpretation, there is the *extensional* interpretation that models a system according to what can be *observed* about it in terms of the activities it performs, i.e., the actions that it executes. This in turn depends on what counts as an observation. A very popular point of view is that it suffices to consider *linear* observations. This leads to the *interleaving* interpretation, according to which a system executing actions concurrently is no different from one that executes them in arbitrary sequential order. For instance, the system modeled by the pomset above would have the following sequential runs:

$$\begin{array}{cccc} snd & loc & loc & rcv \\ snd & loc & rcv & loc \\ loc & snd & loc & rcv \\ loc & snd & rcv & loc \end{array}$$

It is clear that some information is lost in the interleaving interpretation; for instance, the four sequential runs above could also have arisen from the following pomset, which specifies more dependencies than the previous one:



However, if one adheres to the point of view that this information was not relevant to the correct functioning of a system, then such an abstraction step is in fact desirable.

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. Our approach to this aim is outlined below. In short, we distinguish a class of pomsets with particularly nice properties, which we call *order-deterministic*; the main part of this paper is concerned with an exhaustive study of this subclass, especially including the development of a corresponding equational theory.

The paper is structured as follows. After an introductory account of our approach and an overview of existing pomset algebras and theories in the remainder of this section, in Section 2 we introduce the basic facts about order-deterministic pomsets by

studying the notion of *pomset prefix*. Specifically, in Section 2.3 we show that the subclass of order-deterministic pomsets is a distributive lattice with respect to the prefix ordering and that, given a suitable notion of morphism, it sits within the category of pomsets in a special way (it forms a reflective subcategory).

In Section 3 we investigate the equational theory of order-deterministic pomsets. It is proved sound and complete, and ω-complete in the presence of enough elements. In Section 4 we discuss *refinement* of order-deterministic pomsets, analogous to an operation that has been investigated for series-parallel pomsets: see especially Nielsen, Engberg, and Larsen [17]. This comes down to introducing automorphisms over **POM**[**E**] as objects in the algebra and homomorphism application as a new operator. The corresponding extension of the equations is again proved (ω-) complete.

In Section 5, 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. Among other things, it would seem possible to use the principles used here to generalize strings to pomsets in a similar way to generalize from ordinary modal logics, which are usually interpreted over strings, to logics interpreted over pomsets.

In the full report [21], we give some additional results. In particular, we discuss the notion of *termination* of pomsets and give complete equational theories for two variations: *distributed* termination (where a pomset may have multiple exit points) and *global* termination (where a pomset is either terminated as a whole or not terminated at all).

1.1 Approach Since strings are clearly a special case of pomsets, one way to study the latter is by generalizing 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 generalize 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 infima; 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 infima.

Example 1.1 The pomsets $\boxed{\begin{matrix}a\to b\\ a\to c\end{matrix}}$ and $\boxed{\begin{matrix}&b\\ \nearrow&\\ a\to c\end{matrix}}$ have common prefixes $\boxed{a\to b}$ and $\boxed{a\to c}$ but no largest common prefix; in particular, $\boxed{\begin{matrix}&b\\ \nearrow&\\ a\to c\end{matrix}}$ is itself not a prefix of $\boxed{\begin{matrix}a\to b\\ a\to c\end{matrix}}$.

We take the above observation as the starting point of our study. The order-deterministic pomsets that we concentrate on in this paper are precisely the class of pomsets in which infima are defined. Consequently, over order-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 order-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.

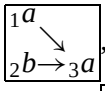
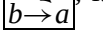
The *join* operation that yields the supremum of two pomsets is in fact a variant of the well-known *disjoint union* of pomsets that lies at the heart of most of the existing pomset theory. Concatenation and join give rise to a complete equational theory of order-deterministic pomsets, which forms the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

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 *labeled 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*;
- $\ell : V \to \mathbf{E}$ is a *labeling 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 $<_p$. A set $V \subseteq V_p$ will be called *left-closed* (with respect to $<_p$) if $v <_p w \in V$ implies $v \in V$.

Two lposets p and q are *isomorphic*, denoted $p \cong q$, if there exists a bijection $f : V_p \to V_q$ such that for all $v, w \in V_p$, $v <_p w$ if and only if $f(v) <_q f(w)$ 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  , where $1, 2, 3$ are vertices and a, b their labels; and the corresponding pomsets by  , i.e. by deleting the vertex identifiers.

1.2 Existing theories

1.2.1 Strings A very special case of partially ordered multisets comprises 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 latter is associative and has the empty string as a left and right neutral element. 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 modeled 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 $V_p \cap V_q = \varnothing$.

1.2.2 Multisets Another very special case of partially ordered multisets comprises 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, and has the empty multiset as its neutral element. 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 modeled in the same way as the empty string and single-element strings above; multiset addition is modeled 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.

1.2.3 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 does not assume 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 $v \not\leq_p w \not\leq_p v$ 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 $\boxed{\begin{matrix} a \to b \\ \nearrow \\ c \to d \end{matrix}}$ is a valid trace; on the other hand, $\boxed{\begin{matrix} a \\ a \end{matrix}}$ cannot be generated (independence is irreflexive), and neither can $\boxed{\begin{matrix} a \\ \searrow \\ b \to b \end{matrix}}$ (if $a\ I\ b$ then the a should not be ordered with respect to the second b, otherwise it should also be ordered with respect to the first b).

The independence relation is extended to pomsets by putting $p\ I\ q$ if and only if $\ell_p(v)\ I\ \ell_q(w)$ for all $v \in V_p$ and $w \in V_q$. The empty trace and single-element traces are modeled 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 $v <_{pq} w$ if and only if 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 with respect to ordinary string concatenation lies in the fact that essentially only the dependent vertices of p and q are ordered.

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

1.2.4 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 [7], Grabowski [8], Jónsson [11], Pratt [19]. Series-parallel pomsets are freely generated by the signature $\Sigma_{sp} = \langle \varepsilon, \cdot, \uplus \rangle$ with the following equations:

$$\begin{aligned}
\varepsilon \cdot x &= x && (1)\\
x \cdot \varepsilon &= x && (2)\\
(x \cdot y) \cdot z &= x \cdot (y \cdot z) && (3)\\
\varepsilon \uplus x &= x && (4)\\
(x \uplus y) \uplus z &= x \uplus (y \uplus z) && (5)\\
x \uplus y &= y \uplus x. && (6)
\end{aligned}$$

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 $v < w'$.

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

$$(a \uplus c)\cdot b\cdot d = \boxed{\begin{smallmatrix} a \\ \searrow \\ c\to b\to d \end{smallmatrix}} \quad c\cdot((a\cdot b) \uplus d) = \boxed{\begin{smallmatrix} c\to a\to b \\ \searrow \\ d \end{smallmatrix}} \quad (a \uplus c)\cdot(b \uplus d) = \boxed{\begin{smallmatrix} a\to b \\ \nearrow\!\!\searrow \\ c\to d \end{smallmatrix}}.$$

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 [11], [19], and [7], extends far beyond this brief exposition. More details are given in Section 5.

1.2.5 Forests Forests (i.e., multisets of 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 forest 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, forests are freely generated by the signature $\Sigma_{tr} = \langle \varepsilon, ;, \uplus \rangle$ with the following equations:

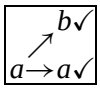
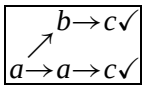
$$\begin{aligned} \varepsilon \uplus x &= x && (4)\\ (x \uplus y) \uplus z &= x \uplus (y \uplus z) && (5)\\ x \uplus y &= y \uplus x && (6)\\ \varepsilon \,;\, x &= \varepsilon && (14)\\ (x \,;\, y) \,;\, z &= x \,;\, (y \,;\, z) && (15)\\ (x \uplus y) \,;\, z &= x \,;\, z \uplus y \,;\, z && (16) \end{aligned}$$

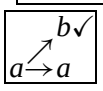
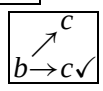
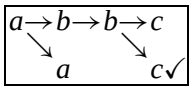
Baeten and Weijland [3] present the above algebra with the additional axiom $x \uplus x = x$. Intuitively, concatenation of two forests p and q appends q to all the *termination points* of p, which are essentially its maximal elements—although some maximal elements may fail to be termination points (see 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 $u \le v$ or $v \le u$ for all $u, v, w \in V$, and where the termination points are modeled by the extra component $\checkmark \subseteq \max_< V$. For instance, $\boxed{\begin{smallmatrix} b \\ \nearrow \\ a\to a\checkmark \end{smallmatrix}}$ is a forest but $\boxed{a\checkmark\to a}$ and $\boxed{\begin{smallmatrix} a \\ \searrow \\ b\to a \end{smallmatrix}}$ are not. The empty forest is modeled by $[\varnothing, \varnothing, \varnothing, \varnothing]$ (hence has no termination points) and single-element forests are modeled by $[\{0\}, \varnothing, \{(0, e)\}, \{0\}]$ (hence the single vertex is a termination point). Forest addition coincides with multiset addition (taking the union of the termination points); concatenation is defined by $p \,;\, q = [V, <, \ell, \checkmark]$ such that

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

It follows that appending the empty forest has the sole effect of removing all termination points. Forest concatenation coincides with string concatenation if p is a nonempty terminated string.

Example 1.4

$a\ ;\ (b \uplus a)\ ;\ c =$  ; = 

$a\ ;\ (b \uplus a\ ;\ \varepsilon)\ ;\ b\ ;\ (c\ ;\ \varepsilon \uplus c) =$  ;  = 

Note that such forests can easily be interpreted as *labeled transition systems* where the vertices become element-labeled transitions. This is in fact the usual interpretation of the above axioms in *process algebra*; see for instance Baeten and Weijland [3].

1.2.6 Order-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 forests, 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:

$$\begin{aligned}
\varepsilon \cdot x &= x && (1)\\
x \cdot \varepsilon &= x && (2)\\
(x \cdot y) \cdot z &= x \cdot (y \cdot z) && (3)\\
\varepsilon \sqcup x &= x && (17)\\
(x \sqcup y) \sqcup z &= x \sqcup (y \sqcup z) && (18)\\
x \sqcup y &= y \sqcup x && (19)\\
x \cdot (y \sqcup z) &= (x \cdot y) \sqcup (x \cdot z). && (20)
\end{aligned}$$

The term *order-deterministic* for the pomsets generated by this algebra is derived from common usage in the case of *forests* interpreted as labeled transition systems (see above): such a transition system is called deterministic if every transition is completely determined by its source node in combination with its label. Likewise, in order-deterministic pomsets, as we will see, every vertex is completely determined by its set of predecessors and its label. (See Rensink [23] for a more extensive discussion of various notions of determinism in partial order models; the notion of order-determinism used here is called *causal* determinism there.)

The order-deterministic pomsets properly include all Mazurkiewicz traces (and therefore all posets) but not all forests or 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. Using (2), (17), and (20) it is straightforward to derive $x = x \sqcup x$, i.e., join is idempotent.

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 *functional* if $x\ R\ y$ and $x\ R\ z$ implies $y = z$, *injective* if $x\ R\ z$ and $y\ R\ z$ implies $x = y$, and *one-to-one* if it is both functional and injective. The *domain* of R is defined as $\mathrm{dom}\ R = \{x \mid \exists y.\ x\ R\ y\}$ and the *codomain* as $\mathrm{cod}\ R = \{x \mid \exists y.\ y\ R\ x\}$.

Definition 2.1 (prefix relations and morphisms) Let $p, q \in \mathbf{LPO}[\mathbf{E}]$ be arbitrary.

1. A *prefix relation* between p and q is a one-to-one relation $R \subseteq V_p \times V_q$ such that both $\mathrm{dom}\ R \subseteq V_p$ and $\mathrm{cod}\ R \subseteq V_q$ 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 $v \in \mathrm{dom}\ R$ or $w \in \mathrm{cod}\ R$.
3. A *prefix morphism* from p to q is a function $f: V_p \to V_q$ 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 $p \sqsubseteq q$.

Example 2.2

- $\{(1,3)\}$ is a maximal prefix relation between $p = \boxed{{}_1a \to {}_2b}$ and $\boxed{{}_3a \to {}_4c}$, but just a (nonmaximal) prefix relation between p and $\boxed{\begin{array}{c} {}^{\nearrow}{}^{4}c \\ {}_3a \to {}_5b \end{array}}$ since it can be extended with $(2,5)$.
- $\boxed{{}_1a \to {}_2b} \not\sqsubseteq \boxed{\begin{array}{c} {}^{3}a \\ {}^{\searrow} \\ {}_4c \to {}_5b \end{array}}$; 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 $\mathrm{cod}\ R = \{3,5\}$ is not left-closed.
- $\boxed{{}_1a \to {}_2b} \sqsubseteq \boxed{\begin{array}{c} {}^{\nearrow}{}^{4}c \\ {}_3a \to {}_5b \end{array}}$ due to the prefix morphism $\{(1,3), (2,5)\}$.
- From $\boxed{{}_1a \to {}_2b}$ to $\boxed{\begin{array}{l} {}^{3}a \to {}_4c \\ {}_5a \to {}_6b \end{array}}$ 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 $\boxed{\begin{array}{c} {}^{\nearrow}{}^{6}b \\ {}_5a \to {}_7c \end{array}}$, 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.

Proposition 2.3 (prefix relations and morphisms)

1. *If the union of two prefix relations (between the same lposets) is 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 2.4 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 relations 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 kernel.

Proposition 2.5 *$\sqsubseteq$ 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: $[p] \sqsubseteq [q]$ if and only if $p \sqsubseteq q$. 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.

Example 2.6 There are two prefix morphisms from $\boxed{{}_1a}$ to $\boxed{\begin{matrix}{}_2a\\ {}_3a\end{matrix}}$, viz. $\{(1,2)\}$ and $\{(1,3)\}$, but their difference cannot be seen on the level of pomsets; the same holds for $\boxed{\begin{matrix}{}_1a\\ {}_2a\end{matrix}}$ and $\boxed{\begin{matrix}{}_3a\\ {}_4a \to {}_5b\end{matrix}}$.

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.

Proposition 2.7 *If p, q are total orders then $p \sqsubseteq q$ if and only if 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.

Proposition 2.8 *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 for 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.

Lemma 2.9 *If there is a unique maximal prefix relation from p_1 to p_2, then p_1 and p_2 have a $\sqsubseteq$-infimum with a unique maximal prefix relation to p_1 and p_2.*

Proof: Let R be the unique maximal prefix relation from p_1 to 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 $V_p \to V_{p_2}$, is a prefix morphism from p to p_2; hence p is a $\sqsubseteq$-lower bound of p_1 and p_2.

Now assume that q is also a $\sqsubseteq$-lower bound of p_1 and p_2; let $f_i\colon V_q \to V_{p_i}$ be prefix morphisms from q to p_i ($i = 1, 2$). $f_1^{-1}\ ;\ f_2$ is then a prefix relation from p_1 to p_2; hence $f_1^{-1}\ ;\ f_2 \subseteq R$, implying $f_2\ ;\ R^{-1} \subseteq f_1$. Furthermore, for arbitrary $v \in V_q$ we have $(f_1(v), f_2(v)) \in f_1^{-1}\ ;\ f_2 \subseteq R$ and hence $f_2(v) \in \mathrm{cod}\, R$, implying $v \in \mathrm{dom}(f_2\ ;\ R^{-1})$; hence $V_q \subseteq \mathrm{dom}(f_2\ ;\ R^{-1})$. We may conclude that $f_1 = f_2\ ;\ R^{-1}$, and since the $f_2\ ;\ R^{-1}$ is a prefix relation from q to p and f_1 is a total function, it follows that f_1 is a prefix morphism from q to p.

Finally, if S is a maximal prefix relation from p to p_1 such that vSw for some $w \neq v$ then $S^{-1}\ ;\ R$ is a prefix relation from p_1 to p_2 such that $w\ (S^{-1}\ ;\ R)\ w'$ where w' is uniquely determined by $v\ R\ w'$. Since R is functional, it follows that $S^{-1}\ ;\ R \not\subseteq R$, contradicting the uniqueness of R. It follows that $S \subseteq id_{V_p}$, and since S is assumed to be maximal, $S = id_{V_p}$. We may conclude that id_{V_p} is the unique maximal prefix relation from p to p_1. Similarly, if S is a maximal prefix relation from p to p_2, one can prove $S = R$; hence R is the unique maximal prefix relation from p to p_2. □

2.2 Order-determinism Lemma 2.9 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 nontrivial maximal auto-prefix relations.

Example 2.10 $\boxed{\begin{matrix} {}_1a \\ {}_2a \to {}_3b \end{matrix}}$ 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 lposets 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.

Lemma 2.11 *Between a pair of prefix unique lposets there is exactly one maximal prefix relation.*

Proof: Let R and S be maximal prefix relations between prefix unique lposets 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 functional. 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 2.9 then gives rise to the following result.

Corollary 2.12 *The class of prefix unique pomsets has $\sqsubseteq$-infima of nonempty sets.*

(The existence of the infimum of an infinite set P follows from the fact that the set of lower bounds of P is bound to be finite; in fact, it is also the set of lower bounds of a finite subset of P, and thus has a greatest element.) In fact, from the proof of Lemma 2.9 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 generalizes 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*.

Proposition 2.13 *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 $\{p\}$ is given by p. 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

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

where R is the unique maximal prefix relation between p and q and $* \notin V_1 \cup V_2$ is an arbitrary vertex identifier. (Those who are familiar with *event structures* will recognize the similarity of this construction to the *synchronization* of two event structures; see e.g. Winskel [26], Boudol and Castellani [4].) 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. Combining the facts that $R' = \{(\pi_1 v, \pi_2 w) \mid v S w\}$ is a prefix relation between p_1 and p_2 and hence $R' \subseteq R$, and $R_i = \{(\pi_i v, \pi_i w) \mid v S w\}$ is an auto-prefix relation of p_i and hence $R_i \subseteq id_{V_i}$, it can be derived that $S = id_{V_q}$.

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, 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 characterization 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.

Definition 2.14 (order-determinism) An lposet $p \in \mathbf{LPO}$ is called *order-deterministic* if every vertex of p is completely determined by the combination of its pre-set and its label, i.e., if

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

Example 2.15 $\boxed{a \to b}$ and $\boxed{a \to c}$ have $\boxed{\begin{array}{c} \quad b \\ \nearrow \\ a \to c \end{array}}$ as their supremum, whereas for instance $\boxed{a \to b}$ and the non-prefix-unique $\boxed{\begin{array}{c} a \to c \\ a \end{array}}$ have no supremum ($\boxed{\begin{array}{c} a \to c \\ a \to b \end{array}}$ and $\boxed{\begin{array}{c} a \to c \\ \searrow \\ a \quad b \end{array}}$ are upper bounds with no common lower bound).

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

Proposition 2.16 *An lposet is order-deterministic if and only if it is prefix unique.*

Proof: ($\Rightarrow$) Assume that $p \in \mathbf{POM}$ is not order-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.

($\Leftarrow$) 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 a minimal 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 order-deterministic. □

The empty pomset and all single-element pomsets are trivially order-deterministic, and concatenation preserves order-determinism; not so however disjoint union, since for instance $\boxed{a} \uplus \boxed{a} = \boxed{\begin{array}{c} a \\ a \end{array}}$. Instead of disjoint union we will therefore use the *supremum* as defined in the proof of Proposition 2.13 as a constructor, which we will call *join* in the remainder of this paper. The join of order-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 order-deterministic pomsets corresponds to the *intersection* of such representatives.

Example 2.17 $\boxed{\begin{array}{c} \quad b \\ \nearrow \\ a \to c \end{array}}$ and $\boxed{\begin{array}{c} a \to b \\ c \end{array}}$ have the isomorphic common ideal $\boxed{a \to b}$. Their join is given by $\boxed{\begin{array}{c} a \to b \\ \searrow \\ c \quad c \end{array}}$ and their meet by $\boxed{a \to b}$.

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 order-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. In fact, we have the following slightly stronger result.

Proposition 2.18 *Every set of order-deterministic pomsets has a set of pairwise compatible representatives.*

Now the meet and join are characterized as follows: if p and q are compatible representatives then

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

Hence the only difference between disjoint union and join is the choice of representatives.

Example 2.19

1. $(a{\cdot}b) \uplus (a{\cdot}c) = \boxed{{}_1a{\to}{}_2b} \uplus \boxed{{}_3a{\to}{}_4c} = \boxed{\begin{matrix} a\to b \\ a\to c \end{matrix}}$ whereas $(a{\cdot}b) \sqcup (a{\cdot}c) = \boxed{{}_1a{\to}{}_2b} \sqcup \boxed{{}_1a{\to}{}_3c} = \boxed{\begin{matrix} & b \\ a\nearrow & \\ a\to c & \end{matrix}}$.
2. $((a \sqcup c){\cdot}b) \sqcup (c{\cdot}a) = \boxed{\begin{matrix} a & \\ & \searrow \\ c\to & b \end{matrix}} \sqcup \boxed{c\to a} = \boxed{\begin{matrix} a\to b & \\ & \nearrow \\ c\to & a \end{matrix}}$; hence we can 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:

Proposition 2.20 *If $p, q \in$ **DPOM** then $p \sqcup q = p \uplus q$ if and only if $p \sqcap q = \varepsilon$.*

The following property lies at the heart of the completeness proofs in Sections 3 and 4.

Proposition 2.21 *If $p \in$ **DPOM** then $p = \bigsqcup_{v \in V_p} (p \restriction \Downarrow v){\cdot}\ell(v)$.*

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

The characterization above of prefix suprema and infima in terms of union and intersection immediately gives rise to the following distributivity property: for all $p_1, p_2, q \in$ **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. In fact, there is a one-to-one correspondence between bases in the above sense and *consistently complete partial orders* (ccpo 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 Rensink [20] for details. We will henceforth ignore the difference between bases and ccpos.) We then have the following strong order-theoretic structure of the order-deterministic pomsets.

Corollary 2.22 $\langle \mathbf{DPOM}, \sqsubseteq \rangle$ *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 2.22 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 characterized in quite a different way as well. A basis $\langle X, \sqsubseteq \rangle$ is called *prime algebraic* if for all $x \in X$,

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

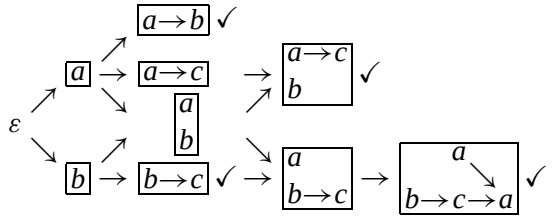
where $y \in X$ is called prime if for all consistent $Y \subseteq X$ (i.e., such that Y has an upper bound and hence a supremum), $y \sqsubseteq \bigsqcup Y$ implies $y \sqsubseteq z$ for some $z \in Y$. Prime algebraic bases play an important role in *partial order semantics*. For instance, Winskel [25] has shown that every prime algebraic domain arises as the set of configurations of a prime event structure; Corradini et al. [5] 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. [25]); therefore Corollary 2.22 implies the following.

Corollary 2.23 *Every* $\sqsubseteq$*-left-closed subset of* $\langle \mathbf{DPOM}, \sqsubseteq \rangle$ *is a prime algebraic basis.*

It follows that every set P of order-deterministic pomsets determines a prime algebraic basis given by its left-closure with respect to $\sqsubseteq$, 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 order-deterministic pomsets. For the restricted case of unlabeled *posets* (which correspond to pomsets with injective labeling functions) more details can be found in Rensink [20].

Example 2.24 The set of order-deterministic pomsets containing $\boxed{a \to b}$, $\boxed{\begin{matrix} a \to c \\ b \end{matrix}}$, $\boxed{b \to c}$ and $\boxed{\begin{matrix} a & \\ & \searrow \\ b \to c & \to a \end{matrix}}$ gives rise to the following prime algebraic basis (where termi-

nated elements are marked ✓):



When considering the class of pomsets and the subclass of order-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 characterize the order-deterministic lposets as a *reflective subcategory* of the lposets.

Definition 2.25 (determinizing morphisms) Let $p, q \in \mathbf{LPO}$. A *determinizing morphism* from p to q is a function $f: V_p \to V_q$ that preserves labeling 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 determinizing morphism is to merge vertices with the same predecessors and the same label, i.e., precisely such vertices as should coincide in order-deterministic lposets, according to Definition 2.14.

Example 2.26 From $p = \boxed{{}_1a \to {}_2b \;\; {}_3a \to {}_4c}$ to $q = \boxed{{}_5a \to {}_6b \;\; {}_8a \searrow {}_7c}$ there is a single determinizing 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.

Proposition 2.27 (determinizing morphisms)

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

From the latter two facts it follows that determinizing morphisms give rise to a category of lposets (where isomorphism corresponds to standard lposet isomorphism); moreover, in the full subcategory of order-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.

Theorem 2.28 *The lposets with determinizing morphisms form 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 an order-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 labeled transition systems in such a way that isomorphism of order-deterministic lposets is fully abstract with respect to bisimilarity.) Now for Dp take $V_p/\sim_p$ as a new vertex set, with the ordering and labeling induced from p; hence

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

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

Theorem 2.29 **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 determinizing morphisms from p to Uq, i.e., from p to q. We have already remarked above that there is at most one determinizing morphism to any order-deterministic lposet; hence we have to show that $Dp \sqsubseteq q$ if and only if there is a determinizing morphism from p to q. Since $Dq \cong q$, any determinizing 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 determinizing 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 with respect to join. This property is formulated in the following theorem.

Theorem 2.30 *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 order-deterministic pomsets We have seen that order-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 characterized *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

$$\varepsilon \cdot x = x \tag{1}$$
$$x \cdot \varepsilon = x \tag{2}$$
$$(x \cdot y) \cdot z = x \cdot (y \cdot z) \tag{3}$$

(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 *algebraize* the prefix ordering by introducing a join-like operator, which for the moment is only partial; in other words, we let

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

Using this definition we can express various properties of the prefix ordering equationally, in the sense that the equation holds if and only if the corresponding property holds for $\sqsubseteq$:

Reflexivity: $x \sqcup x = x$.
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.) In addition, the following is obvious:

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

3.1 The algebra A_{det} To obtain a theory of order-deterministic pomsets, all one has to do now is turn the join operator into a constructor of the algebra rather than a derived notion, with the above equations as axioms. 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 pomsets that are not linear. We obtain a signature $\Sigma_{det} = \langle \varepsilon, \cdot, \sqcup \rangle$ with equations (1)–(3) and in addition

$$\varepsilon \sqcup x = x \tag{17}$$

$$(x \sqcup y) \sqcup z = x \sqcup (y \sqcup z) \tag{18}$$

$$x \sqcup y = y \sqcup x \tag{19}$$

$$x \cdot (y \sqcup z) = (x \cdot y) \sqcup (x \cdot z). \tag{20}$$

This is the theory that we already announced in the introduction. Denoting the resulting algebra by A_{det}, we have the following result.

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

Proving this involves showing that **DPOM**[**E**] is closed under the intended interpretation of ε, $\cdot$ and $\sqcup$ and the equations hold under this interpretation (*soundness*), that every object in **DPOM**[**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 **E** and variables **X**, and $T_{det}(\mathbf{E})$ the corresponding set of *ground terms*, i.e., terms without variables. We will drop the parameter **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 order-deterministic pomsets) and preserves provable equality as pomset equality (= lposet isomorphism); in other words, that **DPOM** is indeed a model of A_{det}.

Theorem 3.2 (A_{det} is sound) *For all* $s, t \in T_{det}$, $[\![t]\!] \in \mathbf{DPOM}$, *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 (17)–(19)). Now we recursively define a function $R\colon \mathbf{DPOM} \to \mathrm{Fin}(T_{det})$ (the latter denoting the set of *finite* subsets of T_{det}) as follows:

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

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 order-deterministic pomsets. It can be proved by induction on the size of p, using the fact that $p = \bigsqcup_{v \in V_p} (p \upharpoonright \Downarrow v) \cdot \ell(v)$ for all order-deterministic pomsets p (Proposition 2.21).

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

Example 3.4 The R-constructed denotation for $\boxed{\begin{smallmatrix} a \to c \\ \nearrow \\ b \to a \end{smallmatrix}}$ 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 pomset 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 denotations of objects are unique up to provable equality.

Theorem 3.5 (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*.

Definition 3.6 (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 3.7 For the sake of readability, we will in practice not write normal forms using the meta-notation $\bigsqcup T$, but rather use ε, t and $t_1 \sqcup \cdots \sqcup t_n$ for (respectively) $\bigsqcup \varnothing$, $\bigsqcup \{t\}$ and $\bigsqcup \{t_1, \ldots, t_n\}$.

Saturation is required to guarantee uniqueness of normal forms, as the following example shows.

Example 3.8 $\boxed{\begin{smallmatrix} & b \\ \nearrow & \\ a \to c & \end{smallmatrix}}$ 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 defined above 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 pomset; in other words, *normal forms are unique*. This is stated in the following lemma.

Lemma 3.9 (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:

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

It can be proved by induction on the term structure that for all $t \in T_{det}(\mathbf{E})$, $norm(t)$ 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.

Lemma 3.10 (normal forms exist) *For all terms $t \in T_{det}$, $\bigsqcup norm(t)$ is a normal form such that $\mathrm{A}_{det} \vdash t = \bigsqcup norm(t)$.*

Proof sketch: By induction on the term structure of t. 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 $norm(s{\cdot}t)$ are by induction N-produced terms. To see that $norm(s{\cdot}t)$ itself is saturated, consider $(\bigsqcup T'){\cdot}e' \in norm(s{\cdot}t)$; then by construction of $norm(s{\cdot}t)$, one of the following cases holds.

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

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

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

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 3.10 and Lemma 3.9 we can prove $\mathrm{A}_{det} \vdash s = \bigsqcup norm(s) = \bigsqcup R([\![s]\!]) = \bigsqcup R([\![t]\!]) = \bigsqcup norm(t) = t$. □

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., Groote [9], Heering [10], or Lazrek et al. [13] for a general discussion.

Theorem 3.11 (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.

Example 3.12 If $|\mathbf{E}| = 1$ then A_{det} is not ω-complete. The order-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 the maximum. It follows that under all ground substitutions, the following equations are valid:

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

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 [9], 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 specialized variant of this technique, described by Heering in [10] and by Lazrek, Lescanne and Thiel in [13], 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 from the first proof idea is that the substitutions required by the latter property, which correspond to the characteristic substitutions of Groote's, are applied only 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.

Definition 3.13 (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 3.11: 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 $\rho_{s,t}: x \mapsto e_x$ 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 if and only if $s = t$ (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 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 order-deterministic pomsets. For the extended algebra we once more give an ω-complete equational theory.

4.1 Homomorphisms, refinement, and determinization Let us consider A_{det}-homomorphisms from **DPOM** to itself, i.e., functions h mapping order-deterministic pomsets to order-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 and none of the equations refer to single elements, every function $h: \mathbf{E} \to \mathbf{DPOM}[\mathbf{E}]$ 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

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

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).

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

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

Example 4.2 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 order-deterministic; on the other hand, $h(p) = h(a) \sqcup h(b) = \boxed{c \to d} \sqcup \boxed{c \to e} = \boxed{\begin{matrix} & d \\ \nearrow & \\ c & \to e \end{matrix}}$.

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.

Proposition 4.3 *For all $p, q \in$ **POM** and h: **POM** $\to$ **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 an order-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 $d, e \in \mathbf{E}$.

Proposition 4.4 *Let h: **DPOM** $\to$ **DPOM** be an arbitrary homomorphism; then $h(p) = p[h]$ for all $p \in$ **DPOM** if and only if h either is image distinct or maps all pomsets to ε.*

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

Lemma 4.5 *If h: **DLPO** $\to$ **DLPO** is image distinct with pairwise compatible images and $p, q \in$ **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: ($\Leftarrow$) If $h(a) = \varepsilon$ for all a then $h(p) = p[h] = \varepsilon$ for all p. Otherwise assume that the h-images, regarded as lposets, are compatible (there are such compatible representatives according to Proposition 2.18), 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.

($\Rightarrow$) Assume h is not constantly ε. If $h(a) = \varepsilon$ and $h(b) = p \neq \varepsilon$ then e.g. $\left[\begin{smallmatrix} a \to b \\ b \end{smallmatrix}\right][h] = \boxed{a \to b}[h] \uplus \boxed{b}[h] = p \uplus p \neq p = h\left(\left[\begin{smallmatrix} a \to b \\ b \end{smallmatrix}\right]\right)$. 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 $\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right][h] = h(a) \uplus h(b) \neq h\left(\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]\right)$. □

Another consequence of Lemma 4.5 is the following.

Proposition 4.6 *Every image distinct homomorphism h: **DPOM** $\to$ **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. □

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 *determinization* of a refined pomset (see Section 2.3). Namely, if we determinize $p[h]$ then the resulting order-deterministic pomset does correspond to $h(p)$

for arbitrary h. For the combination of refinement and determinization we introduce a new operator $*$, defined by

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

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

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

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 $w \in V_{h(\ell_p(v))}$, $([v]_{\sim_p}, w) <_{(Dp)[h]} ([v']_{\sim_p}, w')$ if and only if 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 $f: V_{h*p} \to V_{h*Dp}$ defined by

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

is an isomorphism. □

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

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

Proof: First recall Theorem 2.30 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

$$\begin{aligned} h * (p \sqcup q) = h * D(p \uplus q) = h * (p \uplus q) = D((p \uplus q)[h]) = \\ = D(p[h] \uplus q[h]) = (h * p) \sqcup (h * q). \end{aligned}$$

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 determinization.

Corollary 4.9 *For all* $p \in \mathbf{DPOM}$ *and* $h: \mathbf{E} \to \mathbf{DPOM}$, $p[h] = h(p)$ *if and only if* $p[h]$ *is order-deterministic.*

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

4.2 Refinement algebraically: the algebra A^*_{det} Having established that for order-deterministic pomsets, homomorphism application corresponds to a refinement-like operator, we now want to introduce this operator into the algebra of order-deterministic pomsets. This entails introducing denotations for refinement functions. 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. To denote a refinement function h, we then list the pairs of events and images for which the image does not syntactically 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. This gives rise to an extended algebra A^*_{det} with $\Sigma^*_{det} = \langle \varepsilon, \cdot, \sqcup, [\_/e]_{e\in E} * \_ \rangle$, where $[\_/e]_{e\in E} * \_$ is an E-indexed family of $|E| + 1$-ary operators. Hence, the refinement of t according to a refinement function h is denoted $h * t$. 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

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

Finally, for all finite $E \subseteq \mathbf{E}$ we introduce function terms $h_E = [x_e/e]_{e\in E}$ 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, $h_\varnothing = id = k_\varnothing$.) We then have the following additional equations for all finite $E, F \subseteq \mathbf{E}$:

$$h_E * e = \begin{cases} x_e & \text{if } e \in E \\ e & \text{otherwise.} \end{cases} \tag{21}$$

$$h_E * \varepsilon = \varepsilon \tag{22}$$

$$h_E * (x \cdot y) = (h_E * x) \cdot (h_E * y) \tag{23}$$

$$h_E * (x \sqcup y) = (h_E * x) \sqcup (h_E * y) \tag{24}$$

$$h_E * (k_F * x) = (h_E * k_F) * x \tag{25}$$

$$id * x = x \tag{26}$$

Note that (21)–(25) actually correspond 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.

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

Proof: The soundness of (21)–(26) 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 (21)–(24); 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 (25) and (26) 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. □

4.3 ω-completeness of A^*_{det} The theory of order-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:

Theorem 4.11 (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* $\rho: \mathbf{X} \to T^*_{det}(\mathbf{E})$ *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. (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 (21).)

Definition 4.12 (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 $norm$:

$$\begin{aligned} norm_*(\varepsilon) &:= \varnothing \\ norm_*(e) &:= \{\varepsilon \cdot e\} \end{aligned}$$

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

where the normalization of refinement functions is defined by pointwise extension

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

Note that we remove mappings t_e/e where t_e normalizes 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.

Lemma 4.13 (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 3.11 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.

Example 4.14 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.

Definition 4.15 (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

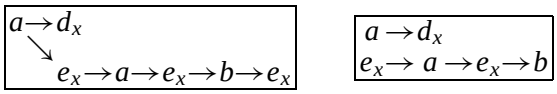
$$C_v := \{w \in V_p \mid \forall u \in V_p.\ (u <_p v \Rightarrow u <_p w) \wedge (u >_p v \Rightarrow u >_p w)\}$$

  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 $\ell_p(v) = e_x$ there is a $w \in V_p$ such that $\ell_p(w) = d_x$ and $v \in C_w$.

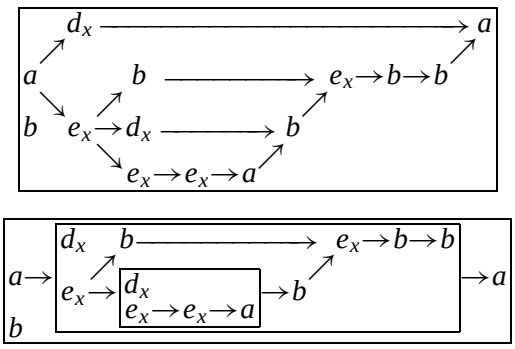
We will not mention the sets $E_{s,t}$ and $\{d_x, e_x\}_{x\in\mathbf{X}}$ with respect to which this property is defined when they are implicitly clear. If p is characteristic and $C_v \subseteq V_p$ is a set of characteristic vertices, then C_v can be contracted into a single node w, yielding a pomset 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.

Example 4.16

1. Any pomset in which there are no d_x- or e_x-labeled 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.
4. Let $E_{s,t}$ and $\mathbf{X}$ be as above, and consider the upper 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.

Lemma 4.17 *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 $R_*\colon \mathbf{DPOM} \to \mathrm{Fin}(T^*_{det}(\mathbf{X}))$ from characteristic pomsets to finite sets of open $*$-normal terms, as follows:

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

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.

Example 4.18 For the pomset in Example 4.16.4, R_* yields the 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

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

The following lemma states the role of the function R_*. It is analogous to Lemma 3.9 and proved by induction on the structure of open $*$-normal form terms.

Lemma 4.19 (open $*$-normal forms are unique) *If $t \in T^*_{det}(\mathbf{X})$ is an open $*$-normal form term 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 Concluding remarks It remains to summarize the results of this paper, to compare them in somewhat more detail with existing work, and to discuss extensions and future work.

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

- Order-deterministic pomsets arise as a generalization of strings, by freely adding objects corresponding to the prefix-suprema of arbitrary finite sets of strings.
- The class of order-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, order-deterministic lposets form a reflective subcategory of the lposets.

We have then formulated an algebra of order-deterministic pomsets by algebraizing 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 an algebraic theory of order-deterministic pomsets and proved it sound and complete, and ω-complete in the presence of sufficiently many elements. The algebra is denoted A_{det} (see Section 3), consisting of the signature $\Sigma_{det} = \langle \varepsilon, \cdot, \sqcup \rangle$ and equations (1)–(3) and (17)–(20) (see Section 1.2.6). ε is the *empty pomset*, $\cdot$ is *concatenation* of pomsets, and $\sqcup$ denotes pomset join. Models are order-deterministic pomsets.

Furthermore, we have defined an extension of A_{det} with a notion of *refinement* which basically algebraizes *homomorphism application*. This yields an algebra denoted A^*_{det} (see Section 4.2) with signature $\Sigma^*_{det} = \langle \varepsilon, \cdot, \sqcup, [\_/e]_{e \in E} * \_ \rangle$ and equations (1)–(3) and (17)–(20) (Section 1.2.6) and (21)–(26) (Section 4.2). Models are order-deterministic pomsets and finite refinement functions mapping elements to order-deterministic pomsets (*finite* meaning that they are the identity except on a finite number of elements).

5.2 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 [8], Jónsson [11], Pratt [19], Gischer [7], 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 order-deterministic pomsets. Hence although within the class of order-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 order-deterministic, is in the class of pomsets for which they are complete. These classes are incomparable: for instance, $\boxed{\begin{matrix} a \to c \\ \nearrow \\ b \to d \end{matrix}}$ is not series-parallel whereas $\boxed{\begin{matrix} a \\ a \end{matrix}}$ is not order-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*, introduced in Mazurkiewicz [14]; good references are Aalbersberg and Rozenberg [1] and Mazurkiewicz [15]. As we have remarked in the introduction, all Mazurkiewicz traces are in fact order-deterministic pomsets, and some of the facts proved for order-deterministic pomsets in this paper constitute a proper generalization 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 order-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 [6]. There are in fact two ways in which trees 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 order-deterministic pomsets in fact correspond to *deterministic forests*, where forests are multisets of trees (see also Section 1.2), and pomset join merges such forests 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 or forests. 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 [24]; however, due to the reversal in the interpretation, the extension to infinite trees requires non-well-founded pomsets.

The second interpretation is the one propagated by De Nicola and Labella [6]. 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.

5.3 Extensions We briefly review a number of directions in which the results of this paper can be extended.

5.3.1 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 order-deterministic pomsets*. These form a proper class, which may be seen as a direct generalization of the *ordinals* in which there exist, instead of a single successor function, a family of different ones (one for each element in **E**). A detailed discussion is outside the scope of this paper.

5.3.2 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 labeling 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 $\boxed{a \to b} \sqsubseteq \boxed{\begin{array}{c} a \to b \\ c \end{array}}$ 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.

5.3.3 Prefix ideals In Gischer [7], an important role is played by *augmentation left-closed* sets of pomsets, which he calls (augmentation) *ideals*. An analogous extension

that we have studied in [22] is to consider *prefix closed sets of pomsets* as models; one might call such sets *prefix ideals*. The basic idea is to interpret the constants of A_{det} 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 modeling *choice*. Choice can be captured equationally as follows:

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

(This operator is in fact entirely analogous to the one described in e.g. Gischer [7] for arbitrary sets of processes.) In other words, we obtain a third monoid, whose neutral element ε equals those of concatenation and join, and whose operator allows all others to distribute over it. For the purpose of modeling prefix ideals this is not yet quite satisfactory, since in fact the models are not only closed under pomset *prefix* but in fact also under the weaker relation $\leq \subseteq \mathbf{DPOM} \times \mathbf{DPOM}$ discussed briefly above. This is due to the fact that concatenation is not left-monotonic with respect to pomset prefix. To repair it one needs a notion of *termination*; see [21], [22] for an extensive discussion.

5.3.4 Pomset logics Based on the results of this paper, it seems an interesting problem to define a *pomset logic*, whose models are pomsets and which has special modalities to deal with pomset join and concatenation. In fact, it would seem that pomset join in some sense corresponds to logical conjunction, and therefore the interpretation of the logic could contain the following rule:

$$p \vDash \varphi \wedge \psi \quad \text{when } p = q_1 \sqcup q_2 \text{ such that } q_1 \vDash \varphi \text{ and } q_1 \vDash \psi.$$

In particular, this corresponds to the fact that **DPOM** forms a complete lattice under $\sqcup$ and the dual $\sqcap$ (Section 2.3). Negation, however, does not let itself be defined easily in this way since the lattice is not complete, and hence certainly not Boolean. On the other hand, pomset concatenation would seem to correspond to the sequential composition of programs, for which there are well-known logical characterizations (see e.g. [12]). For instance, one could define a logical operator ';' with the following semantics:

$$p \vDash \varphi \,;\, \psi \quad \text{when } p = q_1{\cdot}q_2 \text{ such that } q_1 \vDash \varphi \text{ and } q_2 \vDash \varphi.$$

Acknowledgments 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).

REFERENCES

[1] Aalbersberg, J. J., and G. Rozenberg, "Theory of traces," *Theoretical Computer Science*, vol. 60 (1988), pp. 1–82.

[2] Aceto, L., "Full abstraction for series-parallel pomsets," pp. 1–25 in *TAPSOFT '91, Volume 1*, edited by S. Abramsky and T. S. E. Maibaum, vol. 493 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1991.

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

[4] Boudol, G., and I. Castellani, "Permutations of transitions: An event structure semantics for CCS and SCCS," pp. 411–427 in *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, edited by J. W. de Bakker, W.-P. de Roever, and G. Rozenberg, vol. 354 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1989.

[5] Corradini, A., H. Ehrig, M. Löwe, U. Montanari, and F. Rossi, "An event structure semantics for safe graph grammars," pp. 423–446 in *Programming Concepts, Methods and Calculi*, edited by E.-R. Olderog, vol. A–56 of *IFIP Transactions*, North-Holland, Amsterdam, 1994.

[6] De Nicola, R., and A. Labella, "A completeness theorem for nondeterministic Kleene algebras," pp. 536–545 in *Mathematical Foundations of Computer Science 1994*, edited by I. Prívara, B. Rovan, and P. Ružička, vol. 841 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1994.

[7] Gischer, J. L., "The equational theory of pomsets," *Theoretical Computer Science*, vol. 61 (1988), pp. 199–224.

[8] Grabowski, J., "On partial languages," *Fundamenta Informaticæ*, vol. IV (1981), pp. 427–498.

[9] Groote, J. F., *Process Algebra and Structured Operational Semantics*, Ph.D. thesis, University of Amsterdam, 1991.

[10] Heering, J., "Partial evaulation and ω-completeness of algebraic specifications," *Theoretical Computer Science*, vol. 43 (1986), pp. 149–167.

[11] Jónsson, B., "Arithmetic of ordered sets," pp. 3–41 in *Ordered Sets*, edited by I. Rival, Reidel, Dordrecht, 1982.

[12] Kröger, F., *Temporal Logic of Programs*, vol. 8 of *EATCS Monographs on Theoretical Computer Science*, Springer-Verlag, Berlin, 1987.

[13] Lazrek, A., P. Lescanne, and J.-J. Thiel, "Tools for proving inductive equalities, relative completeness, and ω-completeness," *Information and Computation*, vol. 84 (1990), pp. 47–70.

[14] Mazurkiewicz, A., "Concurrent program schemes and their interpretations," DAIMI Report PB–78, Aarhus University, 1977.

[15] Mazurkiewicz, A., "Basic notions of trace theory," pp. 285–363 in *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, edited by J. W. de Bakker, W.-P. de Roever, and G. Rozenberg, vol. 354 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1989.

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

[17] Nielsen, M., U. Engberg, and K. G. Larsen, "Fully abstract models for a process language with refinement," pp. 523–549 in *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, edited by J. W. de Bakker, W.-P. de Roever, and G. Rozenberg, vol. 354 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1989.

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

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

[20] Rensink, A., "Posets for configurations!" pp. 269–285 in *Concur '92*, edited by W. R. Cleaveland, vol. 630 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1992.

[21] Rensink, A., "Deterministic pomsets," Hildesheimer Informatik-Berichte 30/94, Institut für Informatik, University of Hildesheim, 1994.

[22] Rensink, A., "A complete theory of deterministic event structures," pp. 160–174 in *Concur '95: Concurrency Theory*, edited by I. Lee and S. A. Smolka, vol. 962 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1995.

[23] Rensink, A., "Denotational, causal, and operational determinism in event structures," pp. 272–286 in *Trees in Algebra and Programming—CAAP '96*, edited by H. Kirchner, vol. 1059 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1996.

[24] Rutten, J. J. M. M., "Processes as terms: Non well-founded models for bisimulation," *Mathematical Structures in Computer Science*, vol. 2 (1992), pp. 257–275.

[25] Winskel, G., "Event structures," pp. 325–392 in *Petri Nets: Applications and Relationships to Other Models of Concurrency*, edited by W. Brauer, W. Reisig, and G. Rozenberg, vol. 255 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1987.

[26] Winskel, G., "An introduction to event structures," pp. 364–397 in *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, edited by J. W. de Bakker, W.-P. de Roever, and G. Rozenberg, vol. 354 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1989.

Institut für Informatik
Universität Hildesheim
Postfach 101363
D–31113 Hildesheim
Germany
email: rensink@informatik.uni-hildesheim.de