Fixed points without completeness

M.W. Mislove\textsuperscript{a,\ast}, A.W. Roscoe\textsuperscript{b}, S.A. Schneider\textsuperscript{c}

\textsuperscript{a} Department of Mathematics, Tulane University, New Orleans, LA 70118, USA
\textsuperscript{b} Programming Research Group, Computing Laboratory, Oxford University, 8-11 Keble Road, Oxford OX1 3QD, UK
\textsuperscript{c} Department of Computer Science, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK

Abstract

This paper presents a simplification and generalisation of Barrett's "Fixed point theory of unbounded nondeterminism (FACS 3)" for untimed CSP. The difficulties of modeling unbounded nondeterminism in the untimed world persist to the timed case, where it remains the case that there is no reasonable complete partial order over the timed infinite traces model. The fact that the timed predeterministic processes are not a complete partial order means that the untimed approach is not directly applicable to the timed setting. The approach is extended here to a general theory of locally complete partial orders and dominating spaces. If every CSP operator is dominated by some operator on the dominating space, then the fixed point theory of the dominating space may be used to guarantee the existence of fixed points in the underlying CSP model. The application of this theory to untimed CSP is reviewed. The theory is then used to underpin the fixed point theory for timed CSP with infinite nondeterminism, by employing the complete metric space of deterministic timed processes to dominate the model.

1. Introduction

In this paper we develop the underlying mathematical theory we need to build models which support the semantics of unbounded nondeterminism. The two operators common to programming languages supporting concurrency which have been...
the focus of problems in finding models are recursion and nondeterminism. Having recursion in a language requires a model where sufficiently many functions have fixed points; indeed, it was the insight of Dana Scott that non-Hausdorff topological spaces could provide mathematical models where every continuous selfmap has a (canonical) fixed point. In this approach, recursion is modeled by using least fixed points, which exist because of the order structure of the spaces being used. The alternative approach has been to use complete metric spaces, where the mappings are contractions, so that the Banach Fixed Point Theorem assures that every selfmap has a unique fixed point. Both of these approaches have been used in modeling CSP; the partial order approach was used in the failures and failures-divergences models for untimed CSP. The introduction of timing into the language forced the abandonment of the partial order approach in favor of the complete metric space approach, where all operators in the language were modeled as contraction mappings on the appropriate semantic domain. But, this discussion applies only to the case where all of the operators of the language are n-ary for some finite n.

It became apparent from the outset that neither of these "war horses" of modeling theory would be applicable to languages with operators of infinite arity. Indeed, Roscoe [14] established that the "natural" model for unbounded nondeterminism was neither a complete metric space nor a complete partial order. The method used to overcome this problem was found by Barrett [2, 1]. The approach was to fall back into a world of "pure fixed point theory." This involved partially ordered spaces where just enough directed sets had suprema so that all of the operators of the language could be given meanings in terms of least fixed points. In fact, in this setting, least fixed points were sometimes needed for selfmaps which were not continuous (i.e. which did not preserve existing suprema of directed sets), and the theory then required an additional argument to justify the computational validity of the meaning of such an operator, since it might take more than $\omega$ iterations from the least point of the poset to reach the least fixed point of such a selfmap. That justification came by way of a related operational semantics for the language relative to which it was shown that all the terms of the language had the same meaning as that given by the denotational semantics.

These same complications persisted to the case of timed CSP, and they were successfully overcome in the work of Schneider [15], where equivalence with an operational semantics was used to underpin the fixed point theory, following the approach of [14]. It has since become clear that Barrett’s approach may be simplified and adapted to apply to the models for timed CSP which are based on complete metric spaces rather than complete partial orders, yielding the purely denotational fixed point theory presented in Section 4. In fact, the results in this paper arose from discussions among the authors during which they realised that there is a common underlying mathematical theory supporting both the approach for the untimed case and for the timed case. That theory involves what we call local cpo’s. The next section presents that theory, and the rest of the paper contains the application of the theory to untimed and timed CSP.
2. Local cpo's, almost complete semilattices and dominating functions

To begin, a partially ordered set (or poset) is a nonempty set endowed with a reflexive, antisymmetric and transitive relation, which we denote by ≤. A subset $D$ of a poset $P$ is directed if each finite subset of $D$ has an upper bound in $D$. Note that this means that $D$ is nonempty. Dually, a subset $F \subseteq P$ is filtered if each finite subset of $F$ has a lower bound in $F$. A function $f : P \to Q$ is monotone if $x \leq y \in P$ implies $f(x) \leq f(y) \in Q$, and $f$ is continuous if $f(\bigvee D) = \bigvee f(D)$ for all $D \subseteq P$ which are directed and for which $\bigvee D$ exists. A poset $P$ with a least element is a complete partial order (cpo) if each directed subset of $P$ has a least upper bound in $P$. A slightly weaker concept forms the heart of the theory we develop.

**Definition 2.1.** A partially ordered set with a least element is a local cpo if each directed subset having an upper bound has a least upper bound.

**Lemma 2.2.** For a poset $P$ with least element, the following are equivalent:

1. $P$ is a local cpo.
2. Each principal lower set $\downarrow x = \{ y \in P \mid y \leq x \}$ is a cpo, and if $D \subseteq P$ directed and bounded, then $\{ x \in P \mid D \subseteq \downarrow x \}$ is filtered.

**Proof.** If the poset $P$ is a local cpo and $x \in P$, then $\downarrow x$ has the least element of $P$ as its least element. Moreover, each directed subset $D \subseteq \downarrow x$ is bounded in $P$ by $x$, and so it has a least upper bound in $P$. This least upper bound must be in $\downarrow x$ since $x$ is an upper bound of $D$. And, any upper bound of $D$ in $\downarrow x$ is also an upper bound of $D$ in $P$. Hence $\bigvee D$ is the least upper bound of $D$ in $\downarrow x$, so $\downarrow x$ is a cpo. Also, if $D \subseteq P$ is directed and $x$ is any upper bound of $D$, then $\bigvee D$ exists, and $\bigvee D$ is clearly the least element of $\{ x \in P \mid D \subseteq \downarrow x \}$. Consequently $\{ x \in P \mid D \subseteq \downarrow x \}$ is filtered. Thus (1) implies (2).

Conversely, suppose the conditions of (2) hold. Let $D \subseteq P$ be directed with $x$ an upper bound of $D$. Then $\downarrow x$ is a cpo, so $D$ has a least upper bound, $\bigvee_{\downarrow x} D$ in $\downarrow x$. If $y$ is another upper bound of $D$ in $P$, then there is some $z \in P$ which is an upper bound for $D$ satisfying $z \leq x, y$. Then, $\bigvee_{\downarrow x} D \in \downarrow x \cap \downarrow y$, and so $\bigvee_{\downarrow x} D = \bigvee_{\downarrow y} D = \bigvee_{\downarrow x} D$. It follows that $\bigvee_{\downarrow x} D = \bigvee D$ is the least upper bound of $D$ in $P$. □

A partially ordered set $P$ is an inf-semilattice if each pair of elements of $P$ has a greatest lower bound; for $x, y \in P$, this element is denoted $x \wedge y$. An inf-semilattice is almost complete if every nonempty subset has a greatest lower bound.

**Corollary 2.3.** (1) An inf-semilattice $P$ in which $\downarrow x$ is a cpo for each $x \in P$ is a local cpo.

(2) An almost complete inf-semilattice is a local cpo.

**Proof.** Suppose that $P$ is an inf-semilattice for which $\downarrow x$ is a cpo for each $x \in P$. If $x, y \in P$, then $\downarrow x \cap \downarrow y$ is not empty, so $\downarrow x$ and $\downarrow y$ have the same least element, and this is also the least element of $P$. Thus, we only need to show that the second condition of
part (2) of Lemma 2.2 holds. But, if $D \subseteq P$ is directed and $x, y$ are upper bounds of $D$, then clearly $x \wedge y$ is also an upper bound of $D$, so \{x \in P \mid D \subseteq \downarrow x\} is filtered. Hence, (1) holds.

For (2), given an almost complete inf-semilattice $P$ and a directed subset $D \subseteq P$ having some upper bound, the set $U$ of all upper bounds of $D$ is then nonempty, and so it has an infimum. This infimum is clearly $\bigvee D$, so $P$ is a local cpo. $\square$

In our applications, the models we use invariably will be almost complete inf-semilattices. But we set out theory in the greater generality of local cpo's and monotone maps. In both of these instances, there are selfmaps with no fixed point. Such an example is the almost complete inf-semilattice $P = \{(n - 1)/n \mid n > 0\}$ and the semilattice mapping $f : P \to P$ satisfying $f((n - 1)/n) = n/(n + 1)$. What this mapping $f$ lacks is a prefixed point: an element $x \in P$ satisfying $f(x) \leq x$.

**Theorem 2.4.** Suppose $P$ is a local cpo with a least element $\bot$ and $f : P \to P$ is a monotone selfmap.

1. If $f(x) \leq x$ for some $x \in P$, then $f$ has a least fixed point given by
   \[
   \text{fix}(f) = \bigvee \{ f^n(\bot) \mid \alpha \text{ an ordinal} \}.
   \]

2. If $P$ is also an almost complete semilattice, then
   \[
   \text{fix}(f) = \bigwedge \{ x \mid f(x) \leq x \}
   \]
   is the infimum of the set of prefixed points of $f$.

**Proof.** Suppose $P$ is a local cpo with least element $\bot$ and let $f : P \to P$ be a monotone selfmap having $x$ as a prefixed point. For each ordinal $\alpha$, we define $f^\alpha(\bot)$ as follows. If $\alpha = \beta + 1$ and $f^\beta(\bot)$ is defined and satisfies $f^\beta(\bot) \leq x$, then $f^\alpha(\bot) = f(f^\beta(\bot)) \leq f(x) \leq x$ is also well defined. If $\alpha$ is a limit ordinal and $f^\beta(\bot)$ is defined and satisfies $f^\beta(\bot) \leq x$ for all $\beta < \alpha$, then $D_\alpha = \{ f^\beta(\bot) \mid \beta < \alpha \}$ is a directed set in $\downarrow x$. It follows that $\bigvee D_\alpha$ exists, so we define $f^\alpha(\bot) = \bigvee D_\alpha$, and clearly $f^\alpha(\bot) \leq x$. An argument using transfinite induction then shows that $f^\alpha(\bot)$ exists and $f^\alpha(\bot) \leq x$ for all ordinals $\alpha$. Then, $D = \{ f^\alpha(\bot) \mid \alpha \text{ an ordinal} \}$ is a directed subset of $\downarrow x$, and so $\bigvee D$ exists. But, since $D$ is indexed over all $\alpha$, it follows that there is some ordinal $\alpha$ with $f^\alpha(\bot) = f^{\alpha + 1}(\bot)$, so that $\bigvee D \in D$ is a fixed point of $f$. If $p$ is any fixed point of $f$, then $\bot \leq p$ implies $f^\alpha(\bot) \leq f^\alpha(p) = p$ for all $\alpha$, so $\bigvee D \leq p$, and $\bigvee D$ is the least fixed point of $f$. This shows part (1) holds.

For part (2), the first part shows that $f$ has a least fixed point $p$, ad $p \in \{ x \mid f(x) \leq x \}$ since any fixed point is a prefixed point, so $\bigwedge \{ x \mid f(x) \leq x \}$ exists. Moreover, since $\{ x \mid f(x) < x \}$ is nonempty, the same is true of $\{ f(x) \mid f(x) \leq x \}$, and
\[
\bigwedge \{ f(x) \mid f(x) \leq x \} \leq \bigwedge \{ x \mid f(x) \leq x \}.
\]
Now, \( f \) is monotone and \( \land \{ f(x) \mid f(x) \leq x \} \leq y \) for each prefixed point \( y \) of \( f \), which implies \( f(\land \{ x \mid f(x) \leq x \}) \leq f(y) \) for each prefixed point \( y \) of \( f \). Thus, \( f(\land \{ x \mid f(x) \leq x \}) \) is a lower bound for \( \{ f(x) \mid f(x) \leq x \} \), so

\[
\land \{ x \mid f(x) \leq x \} \leq f(\land \{ x \mid f(x) \leq x \}).
\]

Combining these two inequalities shows \( \land \{ x \mid f(x) \leq x \} \) is a prefixed point of \( f \). Part (1) shows that \( p \in \land \{ x \mid f(x) \leq x \} \) for any prefixed point \( x \) of \( f \), and so \( p \in \land \{ x \mid f(x) \leq x \} \). But, \( \land \{ x \mid f(x) \leq x \} \leq p \) since \( p \in \{ x \mid f(x) \leq x \} \). Hence the two are equal. \( \square \)

Theorem 2.4 only guarantees a fixed point for a monotone selfmap of a local cpo which already has a prefixed point; even the stronger setting of a selfmap defined on an almost complete inf-semilattice and which preserves all nonempty infima provides no guarantee that a prefixed point exists. One remedy which has served the semantics community well has been to make the additional hypothesis that the underlying poset is directed complete. But the setting in which we are forced to work does not support this hypothesis, and so we devise another method for assuring that the functions in which we are interested have prefixed points, and hence least fixed points by Theorem 2.4. That method amounts to a “dominated convergence theorem.”

Theorem 2.5. Suppose \( P \) is a local cpo, and suppose that \( E \) is a set and \( i : E \to P \) is a function. Let \( f : P \to P \) be a monotone selfmap of \( P \), and suppose there is a related function \( F : E \to E \) such that \( f \circ i \leq i \circ F : E \to P \); i.e. suppose Fig. 1 “subcommutes:”

\[
\begin{array}{ccc}
E & F & E \\
i \downarrow & \leq & \downarrow i \\
P & f & P
\end{array}
\]

Fig. 1

If \( x \in E \) is a fixed point of \( F \), then \( i(x) \in P \) is a prefixed point of \( f \).

Proof. The subcommutativity of the diagram means that \( f(i(x)) \leq i(F(x)) \), and so \( f(i(x)) \leq i(x) \) since \( F(x) = x \). Thus, \( i(x) \) is a prefixed point of \( f \), as claimed. \( \square \)

To apply this result in our setting, we must clarify the types of local cpo’s we will encounter as models. Recall that a (single sorted) \( \mathcal{F} \)-algebra is a set \( S \) which is closed under a family of operators \( \mathcal{F} \). Each operator \( \tau : \mathcal{F} \) has an arity, \( n(\tau) \), which indicates that the operator is a function \( \tau : S^{n(\tau)} \to S \). Since we want to include the infinitary operator \( \bigwedge \), we must expand the usual definition from the case of finitary operators. If \( \kappa \) is an ordinal number, then \( S^\kappa \) denotes the set of functions \( f : \kappa \to S \). An operator \( \tau \in \mathcal{F} \) has arity \( \kappa \) if \( \tau : S^\kappa \to S \). In this case, we denote the arity of \( \tau \) by \( n(\tau) \).
We define a family of subsets \( \{S_\kappa | \kappa \in \text{Ord} \} \) of subsets of \( S \) indexed by the set of ordinal numbers, as follows:

\[
S_0 = \{ \tau | n(\tau) = 0 \},
\]

\[
S_\kappa = \bigcup_{\eta < \kappa} S_\eta \cup \bigcup_{\tau \in \mathcal{F}} \left( \bigcup_{\eta < \kappa} S_\eta \right)^{n(\tau)}, \quad \kappa > 0.
\]

Note that in the case of a successor ordinal, \( S_{\kappa + 1} = S_\kappa \cup \bigcup_{\tau \in \mathcal{F}} \tau((S_\kappa)^{n(\tau)}) \).

We can then define the rank of the terms in the \( \mathcal{F} \)-algebra \( S \) as follows:

\[
\text{rank}(s) = \begin{cases} 
\min \{ \kappa | s \in S_\kappa \}, & \text{if } s \in \bigcup_\kappa S_\kappa, \\
0, & \text{otherwise.}
\end{cases}
\]

In addition to the constants of \( \mathcal{F} \), the terms of rank 0 are those which do not appear in \( S_\kappa \) for any ordinal \( \kappa \). Since such terms have rank 0, and any other term of \( S \) must have the form \( s = \tau(f) \) for some \( f \in (\bigcup_{\eta < \kappa} S_\eta)^{n(\tau)} \), \( \text{rank}(s) \) is well defined. We say that \( S \) is full if each term of rank 0 is the value of some constant from \( \mathcal{F} \).

**Theorem 2.6.** Suppose that \( S \) is a full \( \mathcal{F} \)-algebra and a local cpo, and that each operator in \( \mathcal{F} \) is monotone. Let \( T \) be a \( \mathcal{F}' \)-algebra, and \( i: T \to S \) a function, and suppose that for each operator \( \tau \in \mathcal{F} \), there is an operator \( \tau' \in \mathcal{F}' \) such that \( n(\tau) = n(\tau') \) and \( \tau \circ i^{n(\tau)} \leq i \circ \tau' \). Then, for each term \( s \in S \), there is a term \( t \in T \) such that \( s \leq i(t) \).

**Proof.** We proceed by induction on the rank of a term \( s \in S \). Since \( S \) is full, the terms of rank 0 are constants, i.e. they are each the value of a nullary operator of \( \mathcal{F} \). To each such \( \tau \), there is a constant \( \tau' \in \mathcal{F}' \) such that \( \tau \circ i^0 \leq i \circ \tau' \), by hypothesis. But, this means that \( \tau \leq i(\tau') \), since both are nullary. This proves the result for terms of rank 0.

The other case to consider is for terms of rank greater than 0, when the result holds for all terms of rank less than \( \kappa \), and \( s \in S \) has rank \( \kappa \). Then there is some \( \tau \in \mathcal{F} \) and \( f \in (\bigcup_{\eta < \kappa} S_\eta)^{n(\tau)} \) so that \( s = \tau(f) \). The inductive hypothesis implies that for each \( \alpha < n(\tau) \), there is a term \( t_\alpha \in T \) so that \( f(\alpha) \leq i(t_\alpha) \). We then define \( f' \in T^{n(\tau')} \) by \( f'(x) = t_x \). Also, there is an operator \( \tau' \in \mathcal{F}' \) such that \( n(\tau) = n(\tau') \) and \( \tau \circ i^{n(\tau)} \leq i \circ \tau' \). Then, \( s = \tau(f) \leq \tau(i^{n(\tau)}(f')) \) since the operators of \( \mathcal{F} \) are monotone. Thus, \( s = \tau(i(f)) \leq (\tau \circ i^{n(\tau)})(f') \leq i(\tau'(f)) \). This proves the result for terms of rank \( \kappa \), so the result holds by induction. \( \square \)

Of course, to apply Theorems 2.5 and 2.6, we must find algebras \( S \) and \( T \) of appropriate signatures \( \mathcal{F} \) and \( \mathcal{F}' \), respectively, and the appropriate function \( i: T \to S \). Moreover, \( S \) must be a local cpo such that the operations of \( \mathcal{F} \) are monotone. The details of the construction of these algebras are given in the following sections, both for timed CSP and for untimed CSP. We use the rest of this section to outline how general semantic techniques can be used to build up models for a language supporting recursion from a model which supports only the "finitary part" of the language – the part without variables or recursion.
Our goal is to provide a denotational model for a dialect of timed CSP which supports unbounded nondeterminism, and we also demonstrate how the theory we are developing applies to the untimed case. In each case, the algebra $S$ will be a proposed semantic model for the language under study. For the sake of discussion, we consider a simpler language whose syntax is given by the following BNF-like production rules:

$$P ::= \text{STOP} \mid \text{SKIP} \mid X \mid a \rightarrow P \mid P;P \mid P \parallel P \mid P \sqcap P \mid P \sqcup P \mid \mu X \circ P.$$ 

Here, $a$ ranges over the set $\Sigma$ of atomic actions, and $X$ ranges over the set $\text{VAR}$ of identifiers. We take for the signature $\mathcal{F}$ the operators other than identifiers and recursion, so, in this case,

$$\mathcal{F} = \{\text{SKIP}, \text{STOP}, \;, \parallel, \sqcap, \sqcup \} \cup \{a \rightarrow \mid a \in \Sigma\}.$$ 

This signature generates a sublanguage, $\mathcal{L}$, of the language generated by the full set of production rules. Standard arguments show that initial $\mathcal{F}$-algebras exist, so we can regard $S$ as an initial $\mathcal{F}$-algebra.

There also are $\mathcal{F}$-algebras which are almost complete inf-semilattice cpo's in which $\sqcap$ is modeled by infimum, and in which all of the operators in $\mathcal{F}$ are monotone (in fact, they preserve all nonempty infima), e.g., the failures model or the failures-divergences model for CSP. For this discussion, let us assume that we have such a model, $S$, in hand. Then the initiality of $\mathcal{L}$ guarantees that there is a unique $\mathcal{F}$-algebra homomorphism $\mathcal{M} : \mathcal{L} \rightarrow S$. So, we assume we are given a $\mathcal{F}$-algebra $S$ which is an almost complete inf-semilattice cpo in which all of the operators in $\mathcal{F}$ are continuous, and a $\mathcal{F}$-algebra homomorphism $\mathcal{M} : \mathcal{L} \rightarrow S$. We now describe how we can build a model for the larger language $\mathcal{L}[\text{VAR}]$ with identifiers, and ultimately a model for the full language including recursion, $\mathcal{L}_\mu$, using this model. And we show that the resulting model satisfies the hypotheses of Theorems 2.5 and 2.6 if the model with which we begin satisfies them.

We begin with some comments about environments and models supporting identifiers. These issues were treated from the algebraic point of view in another context in [9], and a complete treatment of this approach in the current setting can be found in [8]; we limit our presentation here to an outline of the relevant results.

We indicated that we regard the language $\mathcal{L}$ as an initial $\mathcal{F}$-algebra. Analogously, the language $\mathcal{L}[\text{VAR}]$ which also includes the identifiers $X \in \text{VAR}$ can be regarded as the free $\mathcal{F}$-algebra over $\text{VAR}$. This view of $\mathcal{L}[\text{VAR}]$ serves to define the substitution map we need, as follows. Recall that the set of syntactic environments $\mathcal{E}$ for the language $\mathcal{L}$ is the family of functions $\mathcal{E} = \mathcal{L}[\text{VAR}] = \{\rho : \text{VAR} \rightarrow \mathcal{L}\}$. It is easy to show that $\mathcal{E}$ is a $\mathcal{F}$-algebra under the pointwise operations. For example, the term of $\mathcal{E}$ corresponding to $\text{STOP}$ is the constant environment which sends every identifier to the term $\text{STOP} \in \mathcal{L}$. Now, each environment $\rho \in \mathcal{E}$ defines a map from $\text{VAR}$ to the $\mathcal{F}$-algebra $\mathcal{L}$. Since $\mathcal{L}[\text{VAR}]$ is the free $\mathcal{F}$-algebra over $\text{VAR}$, it follows that there is a unique homomorphism $\rho : \mathcal{L}[\text{VAR}] \rightarrow \mathcal{L}$ of $\mathcal{F}$-algebras such that $\rho(X) = \rho(X)$ for
each $X \in VAR$. Expressed diagrammatically, we have Fig. 2. We then have the substitution mapping

$$Sub : \mathcal{L}[VAR] \times \mathcal{S} \to \mathcal{S} \quad \text{by} \quad Sub(P, \rho) = \hat{P}(\rho)$$

which clearly is a homomorphism of $\mathcal{T}$-algebras.

We now show how we can extend the semantic model $S$ of the language $\mathcal{L}$ to a semantic model for the language $\mathcal{L}[VAR]$ with identifiers. Given a $\mathcal{T}$-algebra $S$ as we have hypothesised, we define the set of semantic environments to be $\mathcal{S}_S = S^{VAR} = \{ \phi : VAR \to S \}$. Then, the (unique) $\mathcal{T}$-algebra homomorphism $\mathcal{M} : \mathcal{L} \to S$ leads to the mapping

$$\mathcal{M}_{VAR} : \mathcal{S} \to \mathcal{S}_S \quad \text{given by} \quad \mathcal{M}_{VAR}(\rho)(X) = \mathcal{M}(\rho(X))$$

for each $\rho \in \mathcal{S}$ and $X \in VAR$, which is an extension of $\mathcal{M}$ to a $\mathcal{T}$-algebra homomorphism on $\mathcal{S}$ (where we regard $\mathcal{L}$ as the subalgebra of $\mathcal{S}$ of constant environments, and $\mathcal{M}$ as the mapping which sends the constant syntactic environment with value $P \in \mathcal{L}$ to the constant semantic environment with value $\mathcal{M}(P) \in S$).

Furthermore, given a semantic environment $\phi : VAR \to S$, the facts that $\mathcal{L}[VAR]$ is the free $\mathcal{T}$-algebra over $VAR$ and that $S$ is a $\mathcal{T}$-algebra mean there is a unique $\mathcal{T}$-algebra homomorphism $\hat{\phi} : \mathcal{L}[VAR] \to S$ such that $\hat{\phi}(X) = \phi(X)$ for each $X \in VAR$ (see Fig. 3). So each term $P \in \mathcal{L}[VAR]$ defines a function

$$\mathcal{M}_{VAR}(P) : \mathcal{S}_S \to S \quad \text{by} \quad \mathcal{M}_{VAR}(P)(\phi) = \hat{\phi}(P),$$

and this is a homomorphism of $\mathcal{T}$-algebras. Of course, since $\mathcal{L}$ is a subalgebra of $\mathcal{L}[VAR]$, this mapping defines a homomorphism of $\mathcal{L}$ into $S$, which must coincide with $\mathcal{M}$ since the latter is unique. But, at a higher level, we now have a homomorphism of $\mathcal{T}$-algebras

$$\mathcal{M}_{VAR} : \mathcal{L}[VAR] \to (\mathcal{S}_S \to S) \quad \text{by} \quad \mathcal{M}_{VAR}(P) = \lambda \phi : \mathcal{S}_S \bullet \mathcal{M}_{VAR}(P)(\phi).$$

Bringing $\mathcal{M}_{VAR} : \mathcal{S} \to \mathcal{S}_S$ back into the picture, Fig. 4 summarises the situation: All the mappings are $\mathcal{T}$-algebra homomorphisms, and they are induced in a canonical fashion from the single homomorphism $\mathcal{M}$. Thus, from the semantic map $\mathcal{M} : \mathcal{L} \to S$ we have built a semantic model for the language $\mathcal{L}[VAR]$ including identifiers which relates the application map on the semantic side to substitution on the syntactic side.

The final step in our process is to interpret recursive terms of the language. Each term $P \in \mathcal{L}[VAR]$ has a meaning as a function $\mathcal{M}_{VAR}(P) : \mathcal{S}_S \to S$. And, given a semantic environment $\phi \in \mathcal{S}_S$, we can apply the mapping apply to the pair $(\mathcal{M}_{VAR}(P), \phi)$ to
obtain an element of \( S \). If \( P \) is a term of \( \mathcal{L}[\text{VAR}] \), then we can form the recursive term 
\[ \mu X \circ P, \]
and we would like to interpret this term in our model. Given a semantic environment \( \varphi : \text{VAR} \rightarrow S \), we define a self-map
\[ \mathcal{M}_{\text{VAR}}(P)(\varphi X) : S \rightarrow S \]
by
\[ \mathcal{M}_{\text{VAR}}(P)(\varphi X)(s) = \text{apply}(\mathcal{M}_{\text{VAR}}(P), \varphi [X \mapsto s]), \]
where the semantic environment \( \varphi [X \mapsto s] \) is defined by
\[ \varphi [X \mapsto s](Y) = \begin{cases} 
\varphi(Y) & \text{if } Y \neq X, \\
s & \text{if } Y = X.
\end{cases} \]
Since \( S \) is a complete partial order and a continuous \( \mathcal{T} \)-algebra, the interpretation of each operator \( \tau \in \mathcal{T} \) preserves directed suprema in each coordinate separately (as a mapping \( \tau : S^n ightarrow S \)), so each term \( P \in \mathcal{L}[\text{VAR}] \) and each semantic environment \( \varphi : \text{VAR} \rightarrow S \) give rise to a mapping \( \mathcal{M}_{\text{VAR}}(P)(\varphi X) : S \rightarrow S \) which also preserves directed suprema. Thus, we can define the meaning of the term \( \mu X \circ P \) to be
\[ \mathcal{M}_{\text{VAR}}(\mu X \circ P) : S \rightarrow S \]
by
\[ \mathcal{M}_{\text{VAR}}(\mu X \circ P)(\varphi) = \text{fix}(\mathcal{M}_{\text{VAR}}(P)(\varphi X)). \]
Note that \( \mathcal{M}_{\text{VAR}}(\mu X \circ P) = \mathcal{M}_{\text{VAR}}(P) \) if \( X \) is not free in \( P \), so we have lifted the semantic map \( \mathcal{M} : \mathcal{L} \rightarrow S \) to a semantic map \( \mathcal{M}_{\text{VAR}} : \mathcal{L}_\mu \rightarrow (S \rightarrow S) \), where \( \mathcal{L}_\mu \) is the language generated by the full set of production rules
\[ P ::= \text{STOP} \mid \text{SKIP} \mid X \mid a \rightarrow P \mid P ; P \mid P \parallel P \mid P \triangleright P \mid P \parallel P \mid \mu X \circ P. \]
In the following applications of the theory, we generalise these observations to the case that \( S \) is a local cpo or an almost complete inf-semilattice. In both cases which we consider, this is done by defining a related model \( T \) which is a cpo or a complete metric space and then applying Theorems 2.5 and 2.6.

Theorems 2.5 and 2.6 come into play here since the models we construct are not continuous algebras where all functions are Scott continuous. In each of the two cases we consider, we propose a semantic model for the language we study which contains
a model for a related language that does satisfy these domain-theoretic properties (or analogous ones in complete metric spaces), but the model for the full language does not satisfy them. For example, our approach in the case of untimed CSP is as follows. Referring to Theorem 2.6, we propose a model $S$ for the language of untimed CSP without variables or recursion which is an almost complete inf-semilattice and which contains a model $T$ of a related language of "predeterministic" processes and operators on them. The model $T$ is a complete partial order, and all of the operators on $T$ are continuous. Thus, $T$ is a subset of the proposed model $S$, and the mapping $i : T \rightarrow S$ is just inclusion. Showing that Theorem 2.6 applies then amounts to showing that each operator of $S$ has a related operator on the submodel $T$ which dominates its restriction to $T$. Once this is done, it is easy to argue that each term $P$ in the larger language with variables has a term $P'$ in the related language of predeterministic processes with variables whose meaning in $S_T \rightarrow (T \rightarrow T)$ dominates that of $P$ in $S_S \rightarrow (S \rightarrow S)$. First we note the following theorem.

**Theorem 2.7.** Let $S$ be local cpo (respectively, almost complete inf-semilattice) and a $\mathcal{T}$-algebra in which all the operations are monotone.

1. The family $\mathcal{E}_S = S^{\text{VAR}} = \{ \phi : \text{VAR} \rightarrow S \}$ of semantic environments is a local cpo (respectively, almost complete inf-semilattice) in the pointwise order and under pointwise operations whose induced operations are monotone.

2. If $i : T \rightarrow S$ is a function from a $\mathcal{T}'$-algebra $T$ into $S$ satisfying the hypotheses of Theorem 2.6, then there is an induced mapping $i_{\mathcal{E}} : \mathcal{E}_T \rightarrow \mathcal{E}_S$ satisfying the same hypotheses. Hence the conclusions of Theorem 2.6 lift from $S$ to $\mathcal{E}_S$.

**Proof.** For part (1), it is easy to check that $\mathcal{E}_S$ is a $\mathcal{T}$-algebra in the pointwise operations, and that these operations are all monotone. Likewise, if $S$ is an almost complete inf-semilattice, then the same is true of the family $\mathcal{E}_S$ under pointwise operations.

For part (2), given $i : T \rightarrow S$, we can define $i_{\mathcal{E}} : \mathcal{E}_T \rightarrow \mathcal{E}_S$ by $i_{\mathcal{E}}(\phi)(X) = i \circ \phi(X)$. Since the operations on $\mathcal{E}_S$ and $\mathcal{E}_T$ are defined pointwise, it is routine to check that each operator on $\mathcal{E}_S$ has a dominating operator (in the sense of Theorem 2.6) on $\mathcal{E}_T$, which is defined environment-by-environment.

Now, the meaning of each term $P'$ in the related language of predeterministic processes is a function from $\mathcal{E}_T$ to $T \rightarrow T$. Since $T$ is a cpo and each operator on $T$ we use is continuous, this function has a least fixed point in each environment $\phi' \in \mathcal{E}_T$. If we now consider a term $P$ from the full language, Theorem 2.7 implies there is a term $P'$ from the related language of predeterministic processes whose meaning in each environment $\phi' : \text{VAR} \rightarrow T$ dominates that of $P$. Now, if we also assume that $T$ dominates $S$, by which we mean $S = \downarrow T = \{ s \in S \mid (\exists t \in T) s \leq t \}$, then each environment $\phi \in \mathcal{E}_S$ has a dominating environment $\phi' \in \mathcal{E}_T$, which means $\phi(X) \leq i(\phi'(X))$ for each $X \in \text{VAR}$. Then, for the term $P$ and environment $\phi \in \mathcal{E}_S$, there is a corresponding predeterministic term $P'$ and dominating environment $\phi' \in \mathcal{E}_T$ such
that $\mathcal{M}_{VAR}(P)(\varphi_x)(s) \leq \mathcal{M}_{VAR}(P')(\varphi'_x)(t)$ for $s \leq t$. Theorem 2.5 then implies that $\mathcal{M}_{VAR}(P)(\varphi)$ has a least fixed point, which is exactly what we need to define the meaning of recursion in the larger language we are considering.

3. Application to untimed CSP

Before developing the fixed point theory of unboundedly nondeterministic Timed CSP, we briefly review how it applies to the untimed case. This is both to set the existing work in the context of our general theory, and to give an example which is both slightly simpler than Timed CSP, and based around a partial order fixed point theory over the inner space $\mathcal{E}$ rather than the metric one we will see later.

The syntax of untimed, unboundedly nondeterministic CSP is as follows:

$P ::= \text{STOP} \mid \text{SKIP} \mid a \rightarrow P \mid a : A \rightarrow P_a \mid P \sqcup Q \mid P \sqcap Q \mid P \parallel Q \mid P ; Q \mid P \setminus A \mid f(P) \mid f^{-1}(P) \mid X \mid \mu X . P \mid \bigcap S$

This is the same as that used in the papers developing its theory [14, 2]. The important operators from the point of view of unbounded nondeterminism are hiding $(P \setminus A)$, generalised nondeterministic choice $(\bigcap S)$, direct image $(f(P))$ when $f$ is not finite-to-one, and recursion.

The BNF-style syntax above conceals one considerable subtlety, namely that two of the operators (prefix choice: $a : A \rightarrow P_a$, and generalised nondeterministic choice: $\bigcap S$) are infinitary in the sense that they can take an potentially infinite number of process arguments. In the case of prefix-choice, this number is bounded by the size of the overall alphabet $\Sigma$, but no such limit applies to nondeterministic choice. If we did not limit the size of sets permitted in $\bigcap S$, the "set" of objects represented by our BNF would contain a copy of its own powerset – a familiar impossibility. Our preferred way of circumventing this difficulty is to place a limit (which can be any cardinal whatsoever) on the size of the sets combinable by $\bigcap$. If we do this, then we can form an inductive hierarchy of syntactic terms, indexed by the ordinals, where the terms $S_\alpha$ with rank less than or equal to $\alpha$ are all those whose immediate subterms were born strictly earlier. If $\lambda$ is the smallest infinite regular cardinal strictly greater than both the size of $\Sigma$ and the limit on $\bigcap$, then it is straightforward to show that the terms of $S_\lambda$ are precisely the same as the union $\bigcup_{\alpha < \lambda} S_\alpha$ of those with rank less than $\lambda$, and that no further terms are added by taking the hierarchy further. Thus $S_\lambda$ represents a natural (and the least) fixed point of the "equation" represented by the BNF. For more details of these arguments, see [6].

The rank of a term provides an immediate justification for the principle of structural induction, since this then corresponds to ordinary transfinite induction.

When we are working over a mathematical model such as $\mathcal{U}$ (defined below, the model for unboundedly nondeterministic CSP), the restriction on nondeterministic branching is a pure formality, and does not cut down on expressiveness at all. For the
size chosen has no effect on the model itself, and so we can let the limit be bigger than
the size of the model. This means that the language will contain nondeterministic
compositions which are as big as could conceivably be significant.

We recall here the important points of Barrett's development of a fixed point theory
for $\mathcal{U}$. Readers wishing more details about the model or his construction should
consult [14, 2, 1].

The model itself is a natural development from the standard failures/divergences
model of untimed CSP [5]. A process is represented by a triple $\langle F, D, I \rangle$, where
$F \subseteq \Sigma^* \times \mathcal{P}(\Sigma)$ are its failures, $D \subseteq \Sigma^*$ are its divergences, and $I \subseteq \Sigma^\omega$ are its infinite
traces. The model $\mathcal{U}$ is the set of all triples satisfying eight healthiness conditions. The
first seven are straightforward:

\[
\begin{align*}
(st, \{\}) \in F & \Rightarrow (s, \{\}) \in F \\
(t, X) \in F \land Y \subseteq X & \Rightarrow (t, Y) \in F \\
(t, X) \in F \land \forall a \in Y \cdot (t \langle a \rangle, \{\}) \notin F & \Rightarrow (t, X \cup Y) \in F \\
s \in D & \Rightarrow st \in D \\
s \in D & \Rightarrow (st, X) \in F \\
su \in I & \Rightarrow (s, \{\}) \in F \\
s \in D & \Rightarrow su \in I
\end{align*}
\]

Here, and in the rest of this section, $a, b, \ldots$ range over $\Sigma$; $X, Y, A, B, \ldots$ over
$\mathcal{P}(\Sigma)$; $s, t, v, w$ over the finite traces $\Sigma^*$; and $u$ over infinite traces $\Sigma^\omega$.

Except in the presence of divergence, the first seven axioms do not force a process to
have any infinite traces at all, even though the structure of the model allows us to see
that a process $P$, say, never refuses an $a$, which means we can force the infinite trace $a^\omega$. The
axiom which characterises, correctly, this sort of property is rather subtle. One
formulation is given below, others can be found in [14, 3]. Further discussion can also
be found in the next section, since the corresponding axiom for Timed CSP is based on
the ideas developed for $\mathcal{U}$. Here $T$ ranges over prefix-closed sets of finite traces and
$
\tilde{T} = \{ u \in \Sigma^\omega \mid \forall t < u. t \in T \}$.

\[
(s, \{\}) \in F \Rightarrow \exists T \bullet (\forall t \in T \cdot (st, \{a | t \langle a \rangle \notin \tilde{T}\}) \in F \land \{su | \tilde{u} \subseteq I\})
\]

This was the first example known to the authors of a naturally-occurring model for
a programming language which was incomplete under the natural partial order
(refinement: $\langle F, D, I \rangle \sqsubseteq \langle F', D', I' \rangle$ is and only if $F \supseteq F'$, $D \supseteq D'$ and $I \supseteq I'$), and
furthermore where it could be shown [14] that there was no useful complete partial
order.

The first fixed point theory for CSP over $\mathcal{U}$ was based (a) on the fact that it was
a local cpo and (b) that the existence of a congruent operational semantics guaranteed
the existence of prefixed points via a complex structural induction run in parallel with
the congruence proof of the operational semantics. This was inelegant in the sense that
it required so much machinery outside the semantic model.

A member of $\mathcal{U}$ is said to be predeterministic if it never has the choice whether to
accept or refuse an event unless it can also diverge:

$$(s \notin D \land (s, X) \in F) \Rightarrow X \cap \{ a | (s \langle a \rangle, \{ \}) \in F \} = \{ \}$$

It is easily shown that each predeterministic process has as its set of infinite traces the
closure traces $\overline{(P)}$ of its set of finite traces. The set $\mathcal{P}$ of predeterministic processes form
a complete partial order within $\mathcal{U}$. They have the property [14] that each element of
$\mathcal{U}$ is the nondeterministic composition of all its predeterministic refinements.

Various CSP operators introduce nondeterminism so that they do not map $\mathcal{P}$ to
itself. Barrett observed, however, that the way they introduce nondeterminism could
be analysed, and each of the offending operators refined to a version which remained
monotone over $\mathcal{P}$, and furthermore now introduced no nondeterminism and so
mapped $\mathcal{P}$ to itself. This is precisely the situation we generalised in the previous
section: $\mathcal{U}$ plays the role of $P$, $\mathcal{P}$ plays the role of $E$, and the refinement of each
operator restricted to $\mathcal{P}$ to one on $\mathcal{P}$ provides, for each function $f: P \rightarrow P$, a related
function $F: E \rightarrow E$.

In this case we are, of course, using the least fixed point theory over the cpo $\mathcal{P}$ with
monotone functions to establish the existence of prefixed points.

4. The infinite timed failures model for CSP

The infinite timed failures model $TM_\infty$ was introduced in [15], although the fixed
point theory there was supported by an equivalence with the operational semantics. It
was felt at the time that it was not entirely satisfactory that so much work was
required outside the denotational framework in order to guarantee well definedness.
The theory presented in Section 2 supports a purely denotational approach.

We will use the standard syntax of timed CSP enhanced with an arbitrary nondeter-
ministic choice operator. The language $TCSP$ is defined by the following BNF
definition:

$$P ::= \text{CHAOS} \mid \text{STOP} \mid \text{SKIP} \mid \text{Wait } t \mid P : P \mid a \rightarrow P$$
$$\mid P > P \mid P \sqcap P \mid P \sqcap P \mid \prod_{i \in I} P_i \mid a : A \rightarrow P_a$$
$$\mid P \parallel A \mid P \parallel P$$
$$\mid P \setminus A \mid f(P) \mid f^{-1}(P) \mid X \mid \mu X \circ P$$

Variable $X$ is drawn from the set of variables $VAR$, $I$ is a subset of the set of indices $I$,
$A$ is a subset of the universal set of events $\Sigma$, and $f$ is a mapping $\Sigma \rightarrow \Sigma$. The $P_I$ and
$P(a)$ are sets of terms indexed by $I$ and $A$, respectively. The variable $t$ ranges over the
nonnegative reals; we will also allow arithmetic expressions, and consider syntactic equivalence to be modulo equal arithmetic expressions, identifying for example \( Wait \ 5 \) and \( Wait \ (7-2) \). Observe that the requirement that arguments to an arbitrary internal choice be indexed from \( \mathcal{A} \) ensures that the size of the choice is bounded by some cardinal, the requirement discussed in Section 3. By allowing the set \( \mathcal{A} \) to be larger than the cardinality of the semantic model \( TM_t \) (see below), we avoid any practical restrictions.

We require that the body \( P \) of a recursive term \( \mu X \circ P \) must be \( t \)-guarded for \( X \), for some \( t>0 \), given by the following definition.

**Definition 4.1.** The following rules determine the values of \( t \) for which a given timed CSP term \( P \) is \( t \)-guarded for \( X \):

- \( X \) is 0-guarded for \( X \)
- For any \( t \),
  1. \( CHAOS, STOP, SKIP, Wait \ t_0 \), and \( \mu X \circ P \) are all \( t \)-guarded for \( X \)
  2. If \( Y \neq X \) then \( Y \) is \( t \)-guarded for \( X \)
- If \( P \) is \( t \)-guarded for \( X \),
  1. \( a \to P, P \setminus A, \ f(P), \ f^{-1}(P) \), and \( \mu Y \circ P \) are all \( t \)-guarded for \( X \)
  2. \( P \) is \( t' \)-guarded for \( X \), for any \( t' < t \)
  3. \( Wait \ t_0; P \) is \( t_0 + t \)-guarded for \( X \)
- If \( P \) and \( Q \) are \( t \)-guarded for \( X \),
  1. \( P; Q, P \sqcap Q, P \cap Q, P \sqcup Q \), \( P \parallel Q, P \sqparallel Q \), are all \( t \)-guarded for \( X \)
- If \( P \) is \( t_1 \)-guarded for \( X \), and \( Q \) is \( t_2 \)-guarded for \( X \),
  1. \( P; Q \) is \( \min\{t_1, t_2 + t\} \)-guarded for \( X \)
  2. If each \( P_i \) and \( Q_i \) is \( t \)-guarded for \( X \), then
    1. \( a \to P_i \) is \( t \)-guarded for \( X \)
    2. \( \bigcap_{i \in I} P_i \) is \( t \)-guarded for \( X \)

Timed CSP programs are closed terms – those terms with no free occurrences of \( X \) – such that the body \( P \) of every recursive subterm \( \mu X \circ P \) is \( t \)-guarded for some \( t>0 \).

### 4.1. Notation

The events \( a, b, c \) are taken to range over the universal set of events \( \Sigma \), and \( A, B, \) and \( C \) range over \( P(\Sigma) \). The variables \( t \) and \( u \) range over \( R^+ \), the set of nonnegative real numbers. \( s \) ranges over \( (R^+ \times \Sigma)^\omega \), the finite and infinite sequences of timed events; we use \( \mathbb{N} \subseteq (R^+ \times \Sigma) \) to represent timed refusals, sets of timed visible events.

We use the following operations on sequences of events: \( \#s \) is the length of the sequence \( s \); \( s_1 \sqcup s_2 \) denotes the concatenation of \( s_1 \) and \( s_2 \). The notation \( \text{begin}(s) \) is the time of the first event in \( s \), and is \( \infty \) if \( s = \langle \rangle \); \( \text{end}(s) \) is the supremum of the times of events in \( s \), and is 0 if \( s = \langle \rangle \); thus if \( s \) is finite and nonempty, then \( \text{end}(s) \) is the time of
the last event in $s$: $\text{first}(\langle(t, a)\rangle^* s) = a$. The following projections on sequences are defined by list comprehension, where $(u, a) \leftarrow s$ means that $(u, a)$ is a timed event in $s$:

- $s << t = \langle(u, a)\rangle((u, a)\leftarrow s, u < t)$
- $s \ll t = \langle(u, a)\rangle((u, a)\leftarrow s, u < t)$
- $s \triangleright t = \langle(u, a)\rangle((u, a)\leftarrow s, u \geq t)$
- $s \uparrow t = \langle(u, a)\rangle((u, a)\leftarrow s, u = t)$
- $s \uparrow A = \langle(u, a)\rangle((u, a)\leftarrow s, a \in A)$
- $s \setminus A = \langle(u, a)\rangle((u, a)\leftarrow s, a \notin A)$
- $s - t = \langle(u - t, a)\rangle((u, a)\leftarrow s, u \geq t)$
- $s + t = \langle(u + t, a)\rangle((u, a)\leftarrow s)$
- $\sigma(s) = \{ a \mid s \uparrow \{ a \} \neq \langle \rangle \}$

We also define a number of projections on refusal sets:

- $\mathbb{N} \ll t = \{ (u, a) \mid (u, a) \in \mathbb{N}, u < t \}$
- $\mathbb{N} \triangleright t = \{ (u, a) \mid (u, a) \in \mathbb{N}, u \geq t \}$
- $\mathbb{N} \uparrow A = \{ (u, a) \mid (u, a) \in \mathbb{N}, a \in A \}$
- $\mathbb{N} - t = \{ (u - t, a) \mid (u, a) \in \mathbb{N}, u \geq t \}$
- $\sigma(\mathbb{N}) = \{ a \mid (u, a) \in \mathbb{N} \}$
- $\text{end}(\mathbb{N}) = \text{sup} \{ u \mid (u, a) \in \mathbb{N} \}$

We will use $(s, \mathbb{N}) - t$ as an abbreviation for $(s - t, \mathbb{N} - t)$, and $(s, \mathbb{N}) \ll t$ for $(s \ll t, \mathbb{N} \ll t)$.

**Behaviours**

We define the set of finite and infinite timed traces $T\Sigma^\omega_\tau$ and timed refusals $\text{IRSET}$ as follows:

- $T\Sigma^\omega_\tau = \{ s \in (\mathbb{R}^+ \times \Sigma)^\omega \mid s = s_1 \leftarrow s_1 \triangleright \langle(t_1, a_1), (t_2, a_2)\rangle \leftarrow s_3 \Rightarrow t_1 \leq t_2$
  \[ \wedge \# s = \infty \Rightarrow \text{end}(s) = \infty \}$

- $\text{RTOK} = \{ [b, e) \times A \mid 0 \leq b < e < \infty \wedge A \subseteq \Sigma \}$

- $\text{RSET} = \{ \bigcup R \mid R \subseteq \text{RTOK} \wedge R \text{ is finite} \}$

- $\text{IRSET} = \{ \bigcup R \mid \subseteq \text{RTOK} \wedge \forall t \bullet (\bigcup R) \ll t \in \text{RSET} \}$
Behavioural information

We define the set of behaviours $\text{BEH}$ to be the cartesian product $\Sigma^\omega \times \text{IRSET}$. We then define the information partial order $\preceq$ on $\text{BEH}$ as follows:

$$(s, N) \preceq (s', N') \iff \exists s'' \cdot s'' = s'' \land N \equiv N' \preceq \text{begin}(s'')$$

We understand by $(s, N) \preceq (s', N')$ that the left-hand behaviour contains less information than the right. As we would expect, $((\langle \rangle, \{\}) \preceq (s, N)$ for any $(s, N)$.

The upward and downward closure of a set of behaviours $B$ are defined as expected:

$\uparrow B = \{(s, N) \mid \exists (s', N') \in B \cdot (s', N') \preceq (s, N)\}$

$\downarrow B = \{(s, N) \mid \exists (s', N') \in B \cdot (s, N) \preceq (s', N')\}$

We may define a distance function between subsets of $\text{BEH}$:

$$d(S, T) = \inf \{2^{-t} \mid S \preceq t = T \preceq t\}$$

where

$$S \preceq t \triangleq \{(s, N) \preceq t \mid (s, N) \in S\}$$

Observe that this distance function is not a metric, since any two processes that agree on their finite behaviours will be distance 0 apart, even if they have distinct infinite behaviours. For example, the sets

$$\{(\langle \rangle, [0, t) \times \{a\}) \mid 0 \leq t < \infty\} \quad \text{and} \quad \{(\langle \rangle, [0, t) \times \{a\}) \mid 0 \leq t \leq \infty\}$$

are distance 0 apart, although they are distinct.

The evaluation domain $TM_1$

The model: This model is presented in full in [15]; we provide a summary of it here. We formally define $TM_1$ to be those subsets $S$ of $\Sigma^\omega \times \text{IRSET}$ satisfying axioms (1)–(4) given below, and axiom (5) to follow.

1. $(\langle \rangle, \{\}) \in S$

2. $(s \prec s', N) \in S \Rightarrow (s, N \preceq \text{begin}(s')) \in S$

3. $(s, N) \in S \land N' \in \text{IRSET} \land N' \equiv N \Rightarrow (s, N') \in S$

4. $(s, N) \in S \Rightarrow$

   $\exists N' \in \text{IRSET} \bullet N \equiv N' \land (s, N') \in S \land \forall (t, a) \in \mathbb{R}^+ \times \Sigma \bullet$

   $$(C1) (t, a) \notin N' \Rightarrow (s \prec t \langle (t, a) \rangle, N' \preceq t) \in S$$

   $\land$

   $$(C2) (t > 0 \land \neg \exists \varepsilon > 0 \bullet ((t - \varepsilon, t) \times \{a\} \equiv N')) \Rightarrow (s \prec t \langle (t, a) \rangle, N' \preceq t) \in S$$
Axioms (1)–(3) require that an element of $TM_l$ must be a nonempty set of behaviours, downward closed under the information ordering. Axiom (4) requires that on every execution, timed events must be either possible or refusible. We define $TM_4$ to be the elements of $P(BEH)$ that meet axioms (1)–(4).

Definition 4.2. Given $(s, \mathcal{N}) \in S$, we say that $\mathcal{N}'$ is a total extension of $(s, \mathcal{N})$ in $S$ if $\mathcal{N} \subseteq \mathcal{N}'$, and (C1) and (C2) hold for all $(t, a)$ in $\mathbb{R}^+ \times \Sigma$.

In [13] such extensions were termed complete; the terminology has been changed here since “complete” is already used in a different way. Axiom (4) states that every behaviour of a process has some total extension.

None of the above four axioms allow any deductions to be made concerning the presence of infinite trace behaviours in a process. Indeed, given any set $S$ that satisfies those axioms, the set obtained by removing all infinite trace behaviours from $S$ also satisfies them.

Consider a process that is never able to refuse the event $a$ beyond one second of the end of any trace; then it is clear that the process may be forced to perform an infinite sequence of $a$'s. Hence the finite trace behaviours in a process description do provide some information about the infinite behaviours. The fifth axiom of the model, introduced below, is concerned with ensuring that a process description contains sufficient infinite trace behaviours to be consistent with the finite trace behaviours.

The approach taken is to consider first the most deterministic processes. It is easy to deduce the infinite behaviours of such processes, since if all approximations to an infinite behaviour can be performed, then they must all be performed during the same execution (since the process is deterministic), and so the complete execution will exhibit the infinite behaviour.

Many flavours of nondeterminism are discussed in [13]. Nondeterministic behaviour is witnessed by the possibility of a process being able either to refuse or to perform an event at a given point during an execution. We might say that a process is nondeterministic if it contains both the behaviour $(s, \mathcal{N})$ and the behaviour $(s \sim \langle t, a \rangle, \emptyset)$ where $(t, a) \in \mathcal{N}$. Certainly this characterises as nondeterministic a process such as $(a \rightarrow STOP, b \rightarrow STOP)$, which allows both $(\langle \rangle, [0, 1) \times \{a\})$ and $(\langle (0, a) \rangle, \emptyset)$, the refusal and also the possibility of $(0, a)$.

However, unavoidable nondeterminism at a single point is present whenever an offer is withdrawn. For example, the process $(a \rightarrow STOP) \triangleright STOP$ is able both to refuse and to perform event $a$ at time 1. This nondeterminism is essential: Axiom (4) (C2) requires that $a$ is possible at time 1, since it is not refusible before that time; and any total refusal $\mathcal{N}'$ for the behaviour $(\langle \rangle, \emptyset)$ must contain $(1, a)$, since $a$ must be refused after time 1. Although the process is not deterministic in the above sense, it cannot be made any more deterministic. Such instances of nondeterminism are termed essential point nondeterminism; they arise at the moments when events are withdrawn, since the choice at an instant between the withdrawal of the event and its performance must be nondeterministic.
Not all point nondeterminism need be essential. For example, the process \((Wait 1 : b \rightarrow STOP)\) may perform the event \(b\) at time 1 but no earlier or later. Since \(b\) may be refused up until time 1, and also from time 1 onwards, the pair of behaviours \((\langle \rangle, [0,2) \times \{b\})\) and \((\langle (1,b), \{\} \rangle)\) are evidence of nondeterministic behaviour. But in this case, the possibility of \(b\) may be removed, to produce the deterministic process \(STOP\). Nonessential point nondeterminism arises when an event is possible for a single instant only. Since it was not available for some interval up to that instant, its possibility is not guaranteed by axiom (4).

**The nondeterminism partial order \(\subseteq\)**

**Definition 4.3.** Given \(S_1\) and \(S_2\) in \(P(T, \Sigma, X, IRSET)\), we define

\[ S_1 \subseteq S_2 \iff S_2 \subseteq S_1 \]

Since nondeterminism is always manifested in a finite time, it is sufficient to examine the finite behaviours of a process to establish that any nondeterminism present is essential. Any greater process must then have the same finite behaviours.

**Definition 4.4.** A process \(S \in TM_4\) is *finitely maximal* if for any \(S_1 \in TM_4\)

\[ S \subseteq S_1 \Rightarrow \forall t : S \ll t = S_1 \ll t \]

We have a characterisation for finitely maximal processes. It states that if an event is both possible and refusable at some time, it must be due to point nondeterminism required by axiom (4). This turns out to be equivalent to the second definition of quasi-deterministic presented in [13].

**Theorem 4.5.** A set \(S \in TM_4\) is finitely maximal if and only if \(S\) meets predicate (M) defined as follows:

\[(M) (s, \Sigma) \in S \Rightarrow \forall (t, a) : (t, a) \in \Sigma \land (s \rightarrow (t, a), \{\}) \in S \]

\[ \Rightarrow \exists t_1 : end(s) \leq t_1 < t \land (t_1, t) \times \{a\} \cap \Sigma = \{\} \]

**Proof.** See [15]. \(\square\)

The closure of such sets may be defined in the following way.

**Definition 4.6.** For any set \(S \subseteq BEH\) we define \(\bar{S}\), the **closure** of \(S\), as follows:

\[ \bar{S} = \{(s, \Sigma) \in BEH \mid \forall t : (s, \Sigma) \ll t \in S\} \]

Observe that the closure operator defines a retract: \(\bar{S} \subseteq S\), and \(S = \bar{S}\). If \(S = \bar{S}\) then we say \(S\) is closed.
Treating the closure of a process $S$ as the set of possible executions of some process amounts to adopting the assumption that all approximations to an infinite behaviour may be generated by the same execution, and hence that the limit must also be present in the set.

The final condition we require of a candidate maximally deterministic process is that all chains of behaviours do indeed approximate some legitimate behaviour; this is not always the case. For example, the process that is able to perform any finite number of $a$ events at time 0 should not be able to perform infinitely many, so we reject as unreasonable the claim that all of these behaviours arise from the same execution. The problem arises from a chain of behaviours whose production from a single execution would require infinitely many state changes in a finite time; this is precluded by the structure of behaviours.

We say that a set of behaviours is finitely variable when this does not occur.

**Definition 4.7.** A set $S \subseteq BEH$ is finitely variable if for every time $t < \infty$ the set $(S \langle t)$ is a complete partial order under $\leq$.

In order to reasonably assume that related behaviours are produced from the same execution, we disallow infinite chains of behaviours in finite intervals, by (impossibly) requiring them to have least upper bounds. Observe that finite variability is preserved by closure.

Then we may consider any finitely maximal, finitely variable set $S$ to contain the finite behaviours of a maximally deterministic process; the closure of $S$ yields the infinite behaviours as well, yielding the complete semantics of a maximally deterministic process.

**Definition 4.8.** We define $\mathcal{FM}$ to be the set of finitely maximal, finitely variable sets.

**Lemma 4.9.** If $D \in \mathcal{FM}$, then $D \in \mathcal{FM}$.

**Axiom (5)**

Maximal elements of $TM_I$ are the most deterministic processes, and will be completely deterministic apart from essential point nondeterminism. As we have argued, any such process must contain the limits of all its finite behaviours, since all approximations to an infinite behaviour must come from the same execution, whose infinite behaviour is described by the limit. One way of casting the new axiom is to say that after any finite behaviour the process has an almost deterministic implementation. This amounts to saying that all future nondeterminism may be resolved at any point in time.

**Notation (After).** Given a set $S$, define $S/(s, N), t$, (pronounced $S$ after $(s, N), t$) to be the set of behaviours $\{ (s', N') \mid (s^\rightarrow(s' + t), N \cup (N' + t)) \in S \}$. If the refusal set is empty, then
it may be omitted: \( S/s, t = S/(s, \{ \} ), t \); if the time is \( \text{end}(s, \mathcal{N}) \), then it may be omitted: \( S/s = S/s, \text{end}(s) \); and if the trace is empty, then we may omit it: \( S/t = S/\langle \rangle, t \).

**Lemma 4.10.** For any \( S \in T_{M_4} \), \( (s, \mathcal{N}) \in S, t \geq \text{end}(s, \mathcal{N}) \), we have that \( S/(s, \mathcal{N}), t \in T_{M_4} \).

We are now in a position to state the axiom:

\[
(5) \quad (s, \mathcal{N}) \in S \land t \geq \text{end}(s, \mathcal{N}) \Rightarrow \exists D \in \mathcal{F} \mathcal{M} \bullet S/(s, \mathcal{N}), t \subseteq D.
\]

Again, recall that \( T_{M_4} \) is the set of sets of behaviours satisfying axioms (1)–(5). This family is an almost complete inf-semilattice.

**Theorem 4.11.** The partial order \( \subseteq \) defined on \( T_{M_4} \) contains arbitrary nonempty infima. Hence, \( T_{M_4} \) is an almost complete inf-semilattice.

**Proof.** For any nonempty subset \( R \) of \( T_{M_4} \) we have that \( \inf R = \bigcup R \in T_{M_4} \).

Axiom (5) may also be characterised in a different way. Define \( \mathcal{C} \mathcal{L} \) to be those elements of \( T_{M_4} \) that are closed and finitely variable. Then axiom (5) is equivalent to the property that any process is the nondeterministic composition of all the processes in \( \mathcal{C} \mathcal{L} \) that are stronger than it.

**Theorem 4.12.** Let \( S \in T_{M_4} \) be finitely variable. Then the following are equivalent.

1. \( S \) satisfies axiom (5).
2. \( S = \bigcap \{ Q \in \mathcal{C} \mathcal{L} \mid S \subseteq Q \} \)

**Proof.** See [15].

**Lemma 4.13.** If \( S \in T_{M_4} \) meets \( (M) \), then we have

\[
(s, \mathcal{N}) \in S \Rightarrow \exists \mathcal{N}' \bullet \mathcal{N}' \text{ is a total extension of } (s, \mathcal{N})
\]

**Proof.** See [15].

**The semantic function \( \mathcal{F}_1 \)**

We need to provide semantics only for the basic terms of the language without free variables; it is shown in Section 2 how this is lifted to the language including free variables.

The semantic function

\[
\mathcal{F}_1 : TCSP \rightarrow T_{M_4}
\]
is defined by the following set of equations:

\[ F_i[\text{CHAOS}] \triangleq \{ (s, \tau) | s \in T \Sigma^* \land \tau \in IRSET \} \]

\[ F_i[\text{STOP}] \triangleq \{ (\langle \rangle, \tau) | \tau \in IRSET \} \]

\[ F_i[\text{SKIP}] \triangleq \{ (\langle \rangle, \rho) | \rho \notin \sigma(\tau) \} \]

\[ \cup \{ (\langle t, \langle \rangle \rangle, \tau) | t \geq 0 \land \rho \notin \sigma(\tau \uparrow[0, t]) \} \]

\[ F_i[\text{Wait} t_0] \triangleq \{ (\langle \rangle, \tau) | \rho \notin \sigma(\tau \triangleright t_0) \} \]

\[ \cup \{ (\langle t, \langle \rangle \rangle, \tau) | t \geq t_0 \land \rho \notin \sigma(\tau \uparrow[t_0, t]) \} \]

\[ F_i[P ; Q] \triangleq \{ (s, \tau) | \rho \notin \sigma(s) \land (s, \tau \in \{ 0, \text{end}(s, \tau) \} \times \{ \langle \rangle \}) \in F_i[P] \lor \]

\[ s = s_p \sim s_Q \land \rho \notin \sigma(s_p) \land \]

\[ (s_Q, \tau) \in F_i[Q] \land \]

\[ (s_p \sim (t, \langle \rangle), \tau \triangleright t \cup \{ 0, t \} \times \{ \langle \rangle \}) \in F_i[P] \} \]

\[ F_i[a \rightarrow P] \triangleq \{ (\langle \rangle, \tau) | \rho \notin \sigma(\tau) \} \]

\[ \cup \{ (\langle t, a \rangle \sim s, \tau) | t \geq 0 \land \]

\[ a \notin \sigma(\tau \triangleright t) \land \]

\[ \text{begin}(s) \geq t \land \]

\[ (s, \tau) \in F_i[P] \} \]

\[ F_i[P \leftarrow Q] \triangleq \{ (s, \tau) | \text{begin}(s) \leq t_0 \land (s, \tau) \in F_i[P] \} \]

\[ \cup \{ (s, \tau) | \text{begin}(s) \geq t_0 \land (\langle \rangle, \tau \triangleright t_0) \in F_i[P] \land \]

\[ (s, \tau) \in F_i[Q] \} \]

\[ F_i[P \sqcap Q] \triangleq \{ (\langle \rangle, \tau) | (\langle \rangle, \tau) \in F_i[P] \cap F_i[Q] \}

\[ \cup \{ (s, \tau) | s \neq \langle \rangle \land (s, \tau) \in F_i[P] \cup F_i[Q] \land \]

\[ (\langle \rangle, \tau \triangleright \text{begin}(s)) \in F_i[P] \cap F_i[Q] \} \]

\[ F_i[P \setminus Q] \triangleq F_i[P] \cup F_i[Q] \]

\[ F_i[\bigcap_{i \in I} P_i] \triangleq \bigcup_{i \in I} F_i[P_i] \]

\[ F_i[a : A \rightarrow P_a] \triangleq \{ (\langle \rangle, \tau) | A \cap \sigma(\tau) = \{ \} \}

\[ \cup \{ (\langle t, a \rangle \sim (s + t), \tau) | a \in A \land t \geq 0 \land A \cap \sigma(\tau \triangleright t) = \{ \}

\[ \land (s, \tau - t) \in F_i[P(a)] \} \]
\[ \mathcal{F}_1[P \parallel Q] = \{(s, N) | \exists s_P, s_Q \bullet s_{SP} \parallel s_Q \wedge (s_P, N) \in \mathcal{F}_1[P] \wedge (s_Q, N) \in \mathcal{F}_1[Q] \} \]

\[ \mathcal{F}_1[P \setminus A] = \{(s \setminus A, N) | (s, N \cup ([0, \infty) \times A) \in \mathcal{F}_1[P] \} \]

\[ \mathcal{F}_1[f(P)] = \{(f(s), N) | (s, f^{-1}(N)) \in \mathcal{F}_1[P] \} \]

\[ \mathcal{F}_1[f^{-1}(P)] = \{(s, N) | (f(s), f(N)) \in \mathcal{F}_1[P] \} \]

where the auxiliary function on timed traces is defined as follows:

\[ s_{SP} \parallel s_Q = \{ s : T \Sigma^{\infty}_s | \forall t : TIME \bullet \forall a : \Sigma \bullet s \uparrow t \uparrow \{ a \} = s_{SP} \uparrow t \uparrow \{ a \} \} \]

**Fixed points**

The partial order \( \subseteq \) is not complete on the domain \( TM_I \). One way to see this is to consider the processes \( P_n \) defined by

\[ P_n = \bigcap_{n \leq t < \infty} Wait t ; a \rightarrow STOP \]

Then any upper bound of the chain \( \{ P_i \}_{i=0}^\infty \) will be unable to perform an event \( a \) at any finite time, but will not be able to refuse it for ever (since none of the components may do so). Axiom (4) forbids such behaviour for any process, so there is no possible upper bound.

We may also see the partial order is not complete by defining a monotone function (using \( P_0 \) defined above) that does not have a fixed point:

\[ F(X) = \begin{cases} Wait 1 ; P_0 & \text{if } P_0 \notin X \\ Wait 1 ; X & \text{if } P_0 \subseteq X \end{cases} \]

In a complete partial order, every monotonic function has a (least) fixed point.

Furthermore, the straightforward metric space approach based upon recursive programs as contraction mappings does not appear to be applicable to the model.
Consider for example the following processes:

\[ P_0 = \text{STOP} \]
\[ P_{n+1} = a \rightarrow \text{Wait} 1 ; P_n \]
\[ P = \bigcap_{n \in \mathbb{N}} P_n \]
\[ Q = P \cap (\mu X \circ a \rightarrow \text{Wait} 1 ; X) \]

The function \( F(X) = \text{STOP} \cap a \rightarrow \text{Wait} 1 ; X \) has both \( P \) and \( Q \) as fixed points. Hence any metric structure imposed upon \( TM_I \) would have to ensure that \( F(X) \) is not a contraction mapping, since otherwise this would admit a unique fixed point. Such a metric structure would be very different from those employed in other models for timed CSP, and it seems likely that many innocuous recursive definitions would have to be outlawed.

We have observed in Theorem 4.11 that the space \( TM_I \) with partial order \( \subseteq \) is an inf-semilattice. Our strategy will be to provide a dominating space \( TM_D \) for \( TM_I \), and a set of dominating functions, one for each of the TCSP operators, so that Theorems 2.5 and 2.7 will guarantee the existence of least fixed points of all functions that may be built from the TCSP operators. We will then be in a position to define the semantics of the term \( \mu X \circ P \) as the least fixed point of the function corresponding to \( P \).

### 4.2. Domination

We will use the space of maximally deterministic processes as the dominating space \( TM_D \). Elements of this space are deterministic apart from essential point nondeterminism. Since such processes are completely characterised by their finite behaviours, the distance function \( d \) will impose a metric structure upon \( TM_D \) - elements which agree on all finite behaviours must be identical.

We must also provide a family of operators to dominate those of TCSP. We will therefore provide a dominating language \( TCSP_D \) which contains one operator for each operator of TCSP; and provide a semantic function \( \mathcal{F}_D : TCSP_D \rightarrow TM_D \). We will use the injection function \( i : TM_D \rightarrow TM_I \) as our function from the dominating space; thus we need only establish that \( \mathcal{F}_I[P] \subseteq \mathcal{F}_I[P'] \) for any TCSP program \( P \) and corresponding program \( P' \). In fact, this will follow straightforwardly from the definition of \( \mathcal{F}_n \); the laborious part of the proof will be establishing that \( \mathcal{F}_n \) is well defined.

#### 4.2.1. The dominating space

**Definition 4.14.** The dominating space \( TM_D \) is defined to be the set of those elements of \( TM_I \) that are maximal in the partial order \( \subseteq \).
**Lemma 4.15.** A set $S \subseteq \text{BEH}$ is an element of $\text{TM}_D$ if and only if

- $S$ meets axioms (1)–(4)
- $S$ is closed
- $S$ is finitely variable
- $S$ meets $(M)$

**Theorem 4.16.** The space $\text{TM}_D$ with distance function $d$ is a complete metric space.

**Proof.** We first observe that the space of all closed sets of behaviours

$$\text{CLO} = \{ S \in \mathcal{P}(\text{BEH}) | S = \bar{S} \}$$

with distance function $d$ is a complete metric space, since elements of $\text{CLO}$ are determined completely by their finite time behaviours. We use the following variant of Theorem 9.6.3 from [10].

**Lemma 4.17.** Suppose $S$ is a predicate on $\text{CLO}$ such that, for any $P \in \text{CLO}$,

$$\neg S(P) \Rightarrow \exists t \in \mathbb{R}^+ \rhd \forall Q \in \text{CLO} \bullet (P \ll t = Q \ll t \Rightarrow \neg S(Q))$$

Then $S$ is closed in $\text{CLO}$. □

This lemma may be used to show that axioms (1)–(3), finite variability, and $(M)$, are all closed in $\text{CLO}$. For example, if $P$ is not finitely variable, then there is some $t$ such that $P \ll t$ is not a complete partial order under $\leq$. But if $P \ll t = Q \ll t$ then $Q \ll t$ is not a complete partial order under $\leq$, and so $Q$ is not finitely variable. It follows from the lemma that finite variability is closed in $\text{CLO}$.

However, axiom (4) is not closed in $\text{CLO}$. Consider the processes

$$P_n = \mathcal{F}_1 \left[ \left( \bigcap_{t \geq 1} ((\text{Wait } t ; a \rightarrow \text{STOP}) \uplus (\text{Wait } t^{-1} ; b \rightarrow \text{STOP})) \right) \right]$$

$$\text{Wait } n ; a \rightarrow \text{STOP}$$

$$P = \mathcal{F}_1 \left[ \left( \bigcup_{t \geq 1} ((\text{Wait } t ; a \rightarrow \text{STOP}) \uplus (\text{Wait } t^{-1} ; b \rightarrow \text{STOP})) \right) \right]$$

\[ \cup \downarrow \{(>, [0, \infty) \times \{a\})\} \]

The processes $P_1$ and $P$ are all elements of $\text{CLO}$. Each $P_i$ meets axiom (4), but $P$ does not. However,

$$\lim_{t \to \infty} P_i = P$$

and so axiom (4) is not closed in $\text{CLO}$. 
Lemma 4.17 may be used to show that the following alternative to axiom (4) is closed in $CLO$:

$$(4') \quad (s, \mathbf{K}) \in S, t \geq \text{end}(s, \mathbf{K}) \Rightarrow$$

$$\exists \mathbf{N}' \in \text{IRSET} \bullet \mathbf{N} \subseteq \mathbf{N}' \land (s, \mathbf{N}') \in S \land \forall t' < t \land \forall a \in \Sigma \bullet$$

$$(C1) \quad (t', a) \notin \mathbf{N}' \Rightarrow (s \lessdot t' \prec \langle (t', a) \rangle, \mathbf{N}' \ll t') \in S$$

$$\land$$

$$(C2) \quad (t' > 0 \land \neg \exists \varepsilon > 0 \bullet ((t' - \varepsilon, t') \times \{a\} \subseteq \mathbf{N}'))$$

$$\Rightarrow (s \lessdot t' \prec \langle (t', a) \rangle, \mathbf{N}' \ll t') \in S$$

Hence the intersection of the closed predicates axioms (1)–(3), axiom (4'), finite variability, and (M), is closed in $CLO$, and therefore constitutes a complete metric space with distance function $d$. Furthermore, axioms (4) and (4') are equivalent in the presence of (M), and so this complete metric space is the space $TM_D$.

### 4.2.2. Dominating the TCSP operators

Each TCSP operator will be dominated by a TCSP$_D$ operator. Those operators that introduce no nondeterminism (i.e. STOP, SKIP, Wait $t$, $a \rightarrow P$, $a : A \rightarrow P_a$, $f^{-1}(P)$) will serve to dominate themselves: their semantics given by $\mathcal{S}_D$ will be simply the restriction of $\mathcal{S}_I$ to the domain $TM_D$. The operators which may introduce nondeterminism when their arguments are maximally deterministic must be dominated by operators which resolve all of the nondeterminism. Thus for example we define the nondeterminism operator $\nabla$ on TCSP$_D$ which satisfies $\mathcal{S}_D[P \ominus Q] = \mathcal{S}_D[P]$. In each case, it is clear from its definition that the behaviours of $\mathcal{S}_D[P \ominus Q]$ are a subset of the behaviours of $\mathcal{S}_I[P \ominus Q]$, since the definition for $\ominus'$ will be similar to that for $\ominus$, but with some extra constraints on those behaviours that are permitted. It is less clear that the definition returns an element of $TM_D$ when supplied with arguments from that space; proofs of well definedness are provided in Appendix A.

Let $c : P(\Sigma) \rightarrow \Sigma$ be a choice function such that $c(A) \in A$ whenever $A \neq \emptyset$; and let $d : P(\mathcal{I}) \rightarrow \mathcal{I}$ be a choice function such that $d(I) \in I$ for nonempty indexing sets $I$.

The deterministic language TCSP$_D$ is defined by the following BNF:

$$P ::= \text{CHAO}\text{S}_0 \mid \text{STOP} \mid \text{SKIP} \mid \text{Wait } t \mid P 
\overset{d}{\rightarrow} P \mid a \rightarrow P \mid \prod_{i \in I} P_i \mid a : A \rightarrow P_a$$

$$\mid P \upharpoonright A \mid P \lhd P \mid P \ll P \mid P \lll P \mid P \hspace{1em} \mid f_c(P) \mid f^{-1}(P) \mid X \mid \mu X \circ P$$

The operators of TCSP$_D$ correspond to the operators of TCSP in the obvious way.
The semantic function $\mathcal{F}_D : TCSP_D \rightarrow TM_D$ is defined by the following equations. In the necessary cases, nondeterminism introduced by the corresponding $TCSP$ operator must be removed.

\[
\begin{align*}
\mathcal{F}_D[STOP] & \doteq \mathcal{F}_l[STOP] \\
\mathcal{F}_D[SKIP] & \doteq \mathcal{F}_l[SKIP] \\
\mathcal{F}_D[Wait t_0] & \doteq \mathcal{F}_l[Wait t_0] \\
\mathcal{F}_D[a \rightarrow P] & \doteq \mathcal{F}_l[a \rightarrow P] \\
\mathcal{F}_D[a : A \rightarrow P_a] & = \mathcal{F}_l[a : A \rightarrow P_a] \\
\mathcal{F}_D[f^{-1}(P)] & = \mathcal{F}_l[f^{-1}(P)]
\end{align*}
\]

**Bottom**

The constant program $CHAOS$ is maximally deterministic, and is dominated by any constant deterministic process. For definiteness we will use the semantics of $STOP$.

\[ \mathcal{F}_D[CHAOS_0] \doteq \{(\langle \rangle, N) \mid N \in ISET\} \]

**Sequential composition**

The nondeterminism introduced by sequential composition is similar to that introduced by timeout, since it is also associated with the passing of control between processes. Again, if both processes are able to perform an event at a possible instant where control is passed from one to the other, then it is nondeterministic which process performed the event, and hence whether or not control was passed. Point nondeterminism may also arise, if an event is made available at the moment control is passed. Consider for example the process

\[(a \rightarrow P \quad b \rightarrow Q \quad Wait 1 ; c \rightarrow R \quad Wait 1) ; a \rightarrow STOP\]

The left-hand process is prepared to terminate at time 1 (by having $Wait 1$ perform $\sqrt{1}$), passing control to $a \rightarrow STOP$. Therefore, if event $a$ occurs at time 1, the resulting process is nondeterministically either $P$ or $STOP$. Furthermore, the event $c$ becomes available at time 1 and is instantly disabled, resulting in point nondeterminism. On the other hand, the event $b$ is available for the interval leading up to time 1, so the point nondeterminism at the instant of retraction is essential, by axiom (4).

We dominate this sequential operator with one which removes all nonessential point nondeterminism, and which resolves all nondeterminism at the point of passing of control in favour of the right hand argument. When used in place of the standard
operator above, it will remove the possibility of \( c \) occurring at all, and will yield \( \text{STOP} \) if \( a \) occurs at time 1.

\[
\mathcal{F}_D[P \rightarrow Q] \triangleq \\
\{ (s, N) \mid \not\exists \sigma(s) \\
\quad s = u \rightarrow w \Rightarrow \forall t \in [\text{end}(u), \text{begin}(w), \bullet (u \rightarrow \langle t, \vee \rangle), \{ \} \} \not\in \mathcal{F}_D[P] \\
\quad \land (u \rightarrow \langle \text{begin}(w), \vee \rangle, \{ \} \} \in \mathcal{F}_D[P] \Rightarrow \\
\quad \land \forall u, w \bullet \forall t_1 < \text{begin}(w) \bullet (u, [t_1, \text{begin}(w)]) \times \{ \text{first}(w) \} \not\in \mathcal{F}_D[P] \\
\quad \land \langle \langle 0, \text{begin}(w) \rangle, \{ \} \} \not\in \mathcal{F}_D[Q] \} \\
\cup \\
\{ (s \leftarrow s', N) \mid \not\exists \sigma(s) \\
\quad \land \exists t \bullet (s \leftarrow \langle t, \vee \rangle, N \leftarrow t) \in \mathcal{F}_D[P] \\
\quad \land \text{begin}(s') \geq t \land (s - t, N - t) \in \mathcal{F}_D[Q] \\
\quad \land \forall u, w \bullet s = u \rightarrow w \Rightarrow \\
\quad \forall t' \in [\text{end}(u), \min \{ \text{begin}(w), t \} \bullet (u \rightarrow \langle t', \vee \rangle), \{ \} \} \not\in \mathcal{F}_D[P] \\
\quad \land (w \neq \langle \rangle \land (u \rightarrow \langle \text{begin}(w), \vee \rangle, \{ \} \} \in \mathcal{F}_D[P]) \Rightarrow \\
\quad \forall t_1 < \text{begin}(w) \bullet (u, [t_1, \text{begin}(w)]) \times \{ \text{first}(w) \} \not\in \mathcal{F}_D[P] \}
\]

**Timeout**

The timeout introduces nondeterminism at the moment of transfer of control between its two operands, in two ways: if both processes are able to perform the same event at that instant, then the choice between them is nondeterministic; and if the first process makes some event (that the second cannot perform) available precisely at the moment of transfer control, then point nondeterminism is introduced, since that event is immediately retracted. For example, consider the process

\[
((a \rightarrow P) \Box (\text{Wait} 2; b \rightarrow Q) \Box \text{Wait} 1; c \rightarrow S) \geq a \rightarrow R
\]

If event \( a \) occurs before or after time 2 then no nondeterminism is introduced; it is clear how the choice offered by the timeout has been resolved. If \( a \) occurs at time 2, then the subsequent behaviour is nondeterministically either \( P \) or \( R \). The event \( b \) is made available at time 2, but is immediately retracted by the timeout operator, hence it is possible only at that point in time. Even though \( b \) was offered deterministically by the first process, it has become point-nondeterministic through the introduction of the timeout operator. However, the point nondeterminism of \( c \) at time 2 is essential, since \( c \) is on offer for some interval leading up to that time.

We will dominate this process with an operator which resolves the nondeterministic choice in favour of the first process (provided it was offering the performed event for
some interval leading up to that time). Furthermore, the nonessential point nondeter-
minism is resolved by refusing to allow the performance of offending events; only
events which have been on offer for some interval up to the timeout are permitted,
since they are forced by Axiom (4).

\[
\mathcal{F}_D[P \xrightarrow{T} Q] = \{ (s, \mathbf{N}) \mid \text{begin}(s) < T \land (s, \mathbf{N}) \in \mathcal{F}_D[P] \} \\
\cup \{ (s, \mathbf{N}) \mid \text{begin}(s) > T \land (\langle \rangle, \mathbf{N} \ll T) \in \mathcal{F}_D[P] \land (s, \mathbf{N}) \rightarrow T \in \mathcal{F}_D[Q] \} \\
\cup \{ ((T, a) \leftarrow s, \mathbf{N}) \mid \exists \varepsilon \in (\langle \rangle, [T - \varepsilon, T] \times \{a\} \in \mathcal{F}_D[P] \\
\land (\langle \rangle, \mathbf{N} \ll T) \in \mathcal{F}_D[Q] \\
\lor (\langle \rangle, \mathbf{N} \ll T) \in \mathcal{F}_D[Q] \}
\]

External choice

The only point at which nondeterminism is introduced by the external choice
operator is at the moment the choice is resolved. If both alternatives of the choice are
able to perform the first event, then it is nondeterministic which alternative is in fact
chosen. We will dominate this choice with an operator $\boxplus$ which always chooses the
left-hand choice in such cases. For example, in the case where both arguments are
initially prepared to perform event $a$, the subsequent behaviour is nondeterministic
between the two alternatives for $\text{TCS}^{\text{SP}}$ external choice:

\[(a \rightarrow P) \boxplus (a \rightarrow Q) = a \rightarrow (P \cap Q)\]

We resolve the subsequent nondeterminism in favour of the left-hand alternative:

\[(a \rightarrow P) \boxplus (a \rightarrow Q) = a \rightarrow P\]

The semantics is given as follows:

\[
\mathcal{F}_D[P \boxplus Q] = \{ (s, \mathbf{N}) \mid s = \langle \rangle \land (\langle \rangle, \mathbf{N}) \in \mathcal{F}_D[P] \cap \mathcal{F}_D[Q] \\
\lor (s, \mathbf{N}) \in \mathcal{F}_D[P] \\
\land ((T, a) \leftarrow s) \in \mathcal{F}_D[Q] \\
\lor s = (\langle T, a \rangle) \leftarrow s' \\
\land ((t, a) \left\{ \} \in \mathcal{F}_D[Q] \setminus \mathcal{F}_D[P] \\
\land (s, \mathbf{N}) \in \mathcal{F}_D[Q] \\
\land (\langle \rangle, \mathbf{N} \ll t) \in \mathcal{F}_D[P] \} \]
Binary internal choice
This introduces an initial nondeterministic choice. We resolve this by always picking the left hand argument.

$$\mathcal{F}_D[\langle P \leftarrow Q \rangle] \equiv \mathcal{F}_D[P]$$

Arbitrary nondeterministic choice
We may consider each arbitrary nondeterministic choice as an indexed nondeterministic choice, with indexing set $I$. Arbitrary nondeterministic choice is considered as a family of indexed nondeterministic choices, one for each indexing set. Each member of the family is dominated by a corresponding operator which takes as its argument a set of processes indexed by $I$, and returns the $d(I)$th element.

$$\mathcal{F}_D[\bigcap_{i \in I} P_i] \equiv \mathcal{F}_D[P_{d(I)}]$$

Parallel composition
Although parallel composition preserves determinism for most processes, nonessential point nondeterminism may be introduced in the single case where one side of the combination performs an event at the instant the other side withdraws an event from offer. For example, consider the process

$$((a \rightarrow \text{STOP}) \triangleright \text{STOP})_{a \parallel b}((b \rightarrow \text{STOP}) \triangleright \text{STOP})$$

If the left-hand side performs event $a$ at time 1, then the event $b$ that is possible subsequently in the trace at the same time, is also refusible from that time onward. Hence it is not essential point nondeterminism, even though event $b$ must be possible at time 1 if the $a$ occurs either earlier or later, as this may be deduced from the fact that it is on offer for some interval since the last event. A symmetric argument applies to the event $a$ following the $b$ at time 1. Hence the maximal process above this one will allow both $a$ and $b$ at any times before time 1, but no more than one of them at time 1.

The semantics of the dominating parallel operator must therefore remove any nonessential point nondeterminism that would be introduced by the standard parallel operator.

$$\mathcal{F}_D[P_{A \parallel B} Q] = \{(s, N) | \exists N_P, N_Q : s = s' (A \cup B)$$

$$\land acc(P, Q, A, B, s)$$

$$\land (s \upharpoonright A, N_P) \in \mathcal{F}_D[P]$$

$$\land (s \upharpoonright B, N_Q) \in \mathcal{F}_D[Q]$$

$$\land N \upharpoonright (A \cup B) = (N_P \upharpoonright A) \cup (N_Q \upharpoonright B)\}$$
The predicate \( \text{acc} \) is defined as follows:

\[
\text{acc}(P, Q, A, B, s) = \forall u, t, a \in \text{acc}(\langle t, a \rangle) \leq s \land \text{end}(u) = t \Rightarrow \\
\quad a \in A \Rightarrow \forall \varepsilon \cup (u \in A, [t, t + \varepsilon) \times \{a\}) \not\in \mathcal{F}_D[P] \\
\quad a \in B \Rightarrow \forall \varepsilon \cup (u \in B, [t, t + \varepsilon) \times \{a\}) \not\in \mathcal{F}_D[Q]
\]

**Interleaving**

The interleaving operator introduces nondeterminism whenever both of its components wish to perform the same event. If the event is performed only once, then it is nondeterministic which process performed the event. For example, we have the equivalence

\[
\text{fork}(a \rightarrow P) || (a \rightarrow Q) = a \rightarrow (P || (a \rightarrow Q) \cap (a \rightarrow P) || Q)
\]

We follow the approach to external choice, and resolve the nondeterminism in favour of the left-hand component; hence our dominating operator will be a left-hand favouring interleave operator \( \text{fork} || \). If its arguments are deterministic, then any trace of \( P \text{fork} || Q \) can have arisen in only one way. We define the \( \text{unzip}_{P, Q} \) function to produce the contribution to the trace of \( P \) and \( Q \). Then \( P \text{fork} || Q \) will be able to refuse a set \( \mathbb{N} \) precisely when both of its components are able to do so, while performing their part of the trace. Thus, for example, we will obtain the equivalence

\[
(a \rightarrow P) \text{fork} || (a \rightarrow Q) = a \rightarrow (P \text{fork} || (a \rightarrow Q))
\]

The semantics is given as follows:

\[
\mathcal{F}_D[P \text{fork} || Q] = \{(s, N) \mid \exists u, w \cup (u, w) = \text{unzip}_{P, Q} s \\
\quad \land (u, N) \in \mathcal{F}_D[P] \\
\quad \land (w, N) \in \mathcal{F}_D[Q]\}
\]

The function \( \text{unzip}_{P, Q} : T\Sigma^*_\leq \rightarrow T\Sigma^*_\leq \times T\Sigma^*_\leq \) is defined as follows:

\[
\text{unzip}_{P, Q}(\langle \rangle) = (\langle \rangle, \langle \rangle)
\]

\[
\text{unzip}_{P, Q}(s \rightarrow \langle t, a \rangle, w) = (u \rightarrow \langle (t, a), w) \quad \text{if } (u \rightarrow \langle (t, a), w) \in \mathcal{F}_D[P] \\
\quad \land (\text{end}(w) < t \\
\quad \lor \exists \varepsilon \cup (u, [t, t + \varepsilon) \times \{a\}) \in \mathcal{F}_D[P])
\]

\[
=(u, w \rightarrow \langle (t, a) \rangle) \quad \text{if the above condition does not hold, and}
\]

\[
(w \rightarrow \langle (t, a) \rangle, w) \in \mathcal{F}_D[Q] \\
\quad \land (\text{end}(u) < t \\
\quad \lor \exists \varepsilon \cup (w, [t, t + \varepsilon) \times \{a\}) \in \mathcal{F}_D[P])
\]

else undefined

where \( \text{unzip}_{P, Q}(s) = (u, w) \).
Observe that
\[ unzip_{P,Q}(s) = (u, w) \implies s \sqcup u \sqcup w \]
The function \( unzip \) on an infinite trace is the pairwise limit of \( unzip \) on all the approximations to the trace.

**Hiding**

The hiding operator introduces much nondeterminism into the behaviour of a process. If the external choice between two events is internalised, then it is resolved nondeterministically. If only one side of the choice is internalised, then point nondeterminism may result at the instant the internal event is made available. Furthermore, originally distinct traces may become identical when a set of events is hidden; either trace may have occurred, and the process may nondeterministically be in either resulting state.

For example, consider the following process:

\[
(a \rightarrow P \land Wait 1 ; (b \rightarrow Q \land d \rightarrow a \rightarrow R \land e \rightarrow STOP)) \setminus \{d, e\}
\]

If the \( a \) event does not occur before time 1, then internal events \( d \) and \( e \), and an external \( b \) event are made possible. If neither \( a \) nor \( b \) occur at this point, one of the internal events will occur. Hence the event \( b \) is made point-nondeterministically available at time 1. If an \( a \) occurs at time 1 following event \( d \), then the visible trace will be simply \( \langle 1, a \rangle \), the same trace as would be produced if the \( a \) had pre-empted the hidden event \( d \) at that time. Hence the process following the observation of \( a \) at time 1 is nondeterministically either \( P \setminus \{d, e\} \) or \( R \setminus \{d, e\} \). Finally, if no event is observed up to and including time 1, then a nondeterministic internal choice is made between the \( d \) and the \( e \).

We will resolve the nondeterminism as follows: the nondeterministic choice between competing internal events will be resolved by the choice function \( c \). External events competing with internal events will be permitted only if they are offered for some interval leading up to the time of the possible internal event; hence nonessential point nondeterminism is excluded. Any external event not competing with internal events is permitted except for the case in which an external event follows a string of internal events at that time and it was possible without the internal events. Thus this resolution of nondeterminism would insist that \( a \) at time 1 is followed by \( P \); hence the event \( a \) at time 1 cannot be performed after the internal \( d \) at that time. The event \( b \) cannot occur at any time. If no external event occurs before or at time 1, then the choice function chooses which of events \( d \) and \( e \) to perform internally.

We define a trace \( s \) to be acceptable in process \( P \), hiding set \( A \), if it is consistent with this resolution of nondeterminism; in this case we will write \( acc_{P,A}(s) \)

\[
F_P[P \land \langle A \rangle] = \{ (s \setminus A, N) \mid (s, N \cup [0, \infty) \times A) \in F_P[P] \land acc_{P,A}(s) \}
\]
The predicate $\text{acc}_{P,A}$ is defined as follows; the auxiliary set $S_{P,A}(u)(t)$ denotes the set of events from $A$ that $P$ may perform at time $t$ having performed the trace $u$.

$$\text{acc}_{P,A}(s) = \forall u, w, t, a \bullet s = s \cup \langle (t, a) \rangle \cup w$$ \iff \begin{align*}
(S_{P,A}(u)(t) \neq \{\}) \\
\wedge (a = c(S_{P,A}(u)(t))) \\
\vee (\neg \exists \varepsilon \bullet (u, [t - \varepsilon, t) \times \{a\}) \in D[P] \\
\wedge \text{end}(u) < t \\
\wedge a \notin A)) \\
\vee \\
(S_{P,A}(u)(t) = \{\}) \\
\wedge (\exists \varepsilon \bullet (u \ll [t, [t - \varepsilon, t) \times \{a\}) \in D[P] \wedge (u \uparrow t) \setminus A = \langle \rangle \\
\vee (u \uparrow t) \setminus A \neq \langle \rangle \\
\vee \text{end}(u) < t))
\end{align*}$$

where

$$S_{P,A}(u)(t) = \{b \in A \mid (u \cup \langle (t, b) \rangle, [0, t) \times A) \in D[P]\}$$

The clauses of $\text{acc}_{P,A}$ correspond to the following situations:

- Event $a$ is the best internal event possible at that time.
- Event $a$ is an external event competing with an internal event; in this case, we must have evidence that it is possible, which will consist of the fact that it was not refusible up to that time, and that no event has already occurred at that time.
- Event $a$ is an external event, not competing with an internal event. In that case, there are three possibilities:
  (a) It may follow a string of purely internal events at that time; then it must not have been forcible beforehand (otherwise we already have a way of obtaining the visible part of the trace).
  (b) It may follow a string of events at time $t$ that already includes an external event.
  (c) It may be the first event to occur at that time.

**Alphabet transformation**

The alphabet transformation operator introduces nondeterminism when the alphabet transform function $f$ is not one–one, since there may be many ways of performing a single event; in this situation, the resulting process following the performance of such an event will be the nondeterministic composition of all the possible subsequent processes. We resolve the nondeterminism by employing the choice function $c$ to decide between these possibilities. The function $\text{undo}_f$ applied to a trace $s$ produces the unique trace of process $P$ that could have produced the trace $s$ consistent with the
choice function \( c \); it is undefined if there was no way of producing \( s \) consistent with the choice function.

\[
\mathcal{F}_D\left[ f_t(P) \right] = \{ (s, \mathbb{N}) | (undo_f(s), f^{-1}(\mathbb{N})) \in \mathcal{F}_D \left[ P \right] \}
\]

The function \( undo_f : T\Sigma^\omega_{\leq} \rightarrow T\Sigma^\omega_{\leq} \) is defined as follows:

\[
undo_f(\langle \rangle) = \langle \rangle
\]

\[
undo_f(s \prec (t, a)) = undo_f(s) \prec \langle t, c(A) \rangle \quad \text{if } A \neq \{ \}
\]

\[
\text{undefined} \quad \text{if } A = \{ \}
\]

where

\[
A = \{ d | undo_f(s) \prec \langle t, d \rangle \in \text{traces}(P) \land f(d) = a \}
\]

\( undo_f \) applied to an infinite trace \( w \) is the limit of \( undo_f \) applied to the finite approximations of \( w \).

**The payoff**

**Lemma 4.18.** Each TCSP\(_D\) operator \( F \) dominates its corresponding TCSP operator \( f \). In other words, \( f \circ t \subseteq t \circ F \).

**Proof.** This follows from the definition of each operator. In each case, we obtain that

\[
\mathcal{F}_D \left[ P \right] = \{ (s, \mathbb{N}) \in \mathcal{F}_f \left[ P \right] | C(s, \mathbb{N}) \} \text{ for some condition } C. \quad \square
\]

**Lemma 4.19.** If a TCSP function \( f \) is a contraction mapping with respect to \( d \), then so is its corresponding TCSP\(_D\) function \( F \).

**Corollary 4.20.** If a TCSP function \( f \) is \( t \)-guarded for some \( t > 0 \) then the corresponding TCSP\(_D\) function \( F \) has a unique fixed point.

This follows from the fact that if a TCSP\(_D\) function \( F \) is \( t \)-guarded for \( t > 0 \), then it is a contraction mapping in the complete metric space \( T\Sigma^\omega_D \), and hence has a unique fixed point.

**Corollary 4.21.** If a TCSP function \( f \) is \( t \)-guarded for some \( t > 0 \) then it has a least fixed point in \( T\Sigma^\omega_1 \).

This follows from Lemma 4.18 and Corollary 4.20, which allow us to apply the results of Section 2.
Acknowledgment

The authors would like to thank Frank Oles for his contribution towards the material in Section 1. Mike Mislove is grateful to the Oxford PRG for their hospitality during a sabbatical visit in 1991, when this research was initiated; and Steve Schneider is grateful to Tulane University for their hospitality during a reciprocal visit in Autumn 1991. Mike Mislove and Bill Roscoe thank the ONR for support; and Steve Schneider is grateful to the SERC for their support under fellowship B91/RFH/312. This research also received support from the ESPRIT SPEC and REACT projects.

Appendix A. Well-definedness of $F_D$

The function $F_D$ is defined on sets of behaviours, and yields a set of behaviours. We must show that $F_D[P \ast Q] \in TM_D$ under the assumption that $F_D[P] \in TM_D$ and $F_D[Q] \in TM_D$. To do this, we establish that a set obtained by applying $F_D$ meets Axioms (1)-(4), is closed, is finitely variable, and meets predicate (M).

This must be established for each operator of the language. The proofs of most operators are routine and unenlightening, and are not presented here. The difficult cases are external choice, interleaving, and hiding, and these are given below.

External choice

The operator is defined as follows:

$$F_D[P \triangleleft Q] = \{(s, \Xi) \mid s = \langle \rangle \land (\langle \rangle, \Xi) \in F_D[P] \cap F_D[Q]$$

$$\land (\langle \rangle, \Xi \triangleleft begin(s)) \in F_D[Q]$$

$$\lor s = \langle (t, a) \rangle \bowtie s'$$

$$\land (\langle (t, a), \{ \} \rangle \in F_D[Q] \setminus F_D[P]$$

$$\land (s, \Xi) \in F_D[Q]$$

$$\land (\langle \rangle, \Xi \triangleleft t) \in F_D[P] \}$$

There are three cases to consider: if no events have yet been performed, then the choice is unresolved, and so $P$ and $Q$ must both refuse whatever their environment offers. If events have been performed, and $P$ was able to perform the first one (which follows from $(s, \Xi) \in F_D[P]$ when $s \neq \langle \rangle$), then we require only that $Q$ could refuse $\Xi$ before the choice was resolved, since we have that the choice was resolved in favour of $P$. The choice may be resolved in favour of $Q$ only when $P$ is unable to perform the event resolving the choice; in this case, $P$ must refuse $\Xi$ before time $t$.

We wish to show that $F_D[P \triangleleft Q]$ is contained in $TM_D$ whenever $F_D[P]$ and $F_D[Q]$ are. We must therefore show that $F_D[P \triangleleft Q]$ meets Axioms (1)-(4), is equal
to its closure, is finitely variable, and meets property \((M)\), under the assumption that its components do.

It follows immediately from the semantic definition that

- axioms (1)–(3) are met,
- it is equal to its closure,
- it is finitely variable.

**Axiom (4)**

To prove that axiom (4) holds consider a behaviour \((s, \mathbb{N})\).

If \(s = \langle \rangle\), then let \(\mathbb{N}_P\) and \(\mathbb{N}_Q\) be the (unique) total extensions of \((s, \mathbb{N})\) in \(P\) and \(Q\) respectively. It then follows that \(\mathbb{N}_P \cap \mathbb{N}_Q\) is a total extension for \((s, \mathbb{N})\) in \(P \leftrightarrow Q\).

If \(s \neq \langle \rangle\), then if \((s, \mathbb{N}) \in \mathcal{F}_D[P]\), let \(\mathbb{N}_P\) be the total extension for \((s, \mathbb{N})\) in \(P\), and let \(\mathbb{N}_Q\) be the total extension for \((s, \mathbb{N})\) in \(Q\). It follows that \(((\mathbb{N}_P \cap \mathbb{N}_Q) \trianglelefteq t) \cup (\mathbb{N}_P \triangleright t)\) is a total extension for \((s, \mathbb{N})\) in \(P \leftrightarrow Q\).

If \(s = \langle (t, a) \rangle \triangleleft s',\) and \((\langle (t, a), \{ \} \rangle \in \mathcal{F}_D[Q] \setminus \mathcal{F}_D[P]\), then let \(\mathbb{N}_P\) be the total extension for \((\langle \rangle, \mathbb{N} \trianglelefteq t)\) in \(P\), and let \(\mathbb{N}_Q\) be the total extension for \((s, \mathbb{N})\) in \(Q\). Then \(((\mathbb{N}_P \cap \mathbb{N}_Q) \trianglelefteq t) \cup (\mathbb{N}_Q \triangleright t)\) is a total extension for \((s, \mathbb{N})\) in \(P \leftrightarrow Q\).

This covers all possible cases, so we conclude that \(\mathcal{F}_D[P \leftrightarrow Q]\) meets axiom (4).

**Finite maximality**

Finally, we show that \(\mathcal{F}_D[P \leftrightarrow Q]\) meets predicate \((M)\). Let \((s, \mathbb{N}) \in \mathcal{F}_D[P \leftrightarrow Q]\), and consider \((t, a) \in \mathbb{N}\) with \((s \triangleright \langle (t, a), \{ \} \rangle) \in \mathcal{F}_D[P \leftrightarrow Q]\). We wish to establish that \(\exists t \bullet end(s) \leq t < t \land (t_1, t) \times \{a\} \cap \mathbb{N} = \{\}\) since \((M)\) holds for \(P\).

**Case 1:** If \(s = \langle \rangle\), then \((s, \mathbb{N}) \in \mathcal{F}_D[P] \cap \mathcal{F}_D[Q]\).

- If \((\langle (t, a), \{ \} \rangle) = \mathcal{F}_D[P]\), then \((s \triangleright \langle (t, a), \{ \} \rangle) \in \mathcal{F}_D[P]\), and also \((t, a) \in \mathbb{N}\), so \(\exists t \bullet end(s) \leq t < t \land (t_1, t) \times \{a\} \cap \mathbb{N} = \{\}\), since \((M)\) holds for \(P\).

- If \((\langle (t, a), \{ \} \rangle) \in \mathcal{F}_D[Q] \setminus \mathcal{F}_D[P]\), then \((s \triangleright \langle (t, a), \{ \} \rangle) \in \mathcal{F}_D[Q]\), and \((t, a) \in \mathbb{N}\), so \(\exists t \bullet end(s) \leq t < t \land (t_1, t) \times \{a\} \cap \mathbb{N} = \{\}\), since \((M)\) holds for \(Q\).

**Case 2:** If \(s \neq \langle \rangle\), then \(s = \langle (t_0, a_0) \rangle \triangleright s'\).

- If \((\langle (t_0, a_0), \{ \} \rangle) \in \mathcal{F}_D[P]\), then \((s, \mathbb{N}) \in \mathcal{F}_D[P]\), and \((s \triangleright \langle (t, a), \{ \} \rangle) \in \mathcal{F}_D[P]\), so the result follows since \((M)\) holds for \(P\).

- If \((\langle (t_0, a_0), \{ \} \rangle) \in \mathcal{F}_D[Q] \setminus \mathcal{F}_D[P]\), then the result follows since \((M)\) holds for \(Q\).

It therefore follows that \((M)\) holds for \(\mathcal{F}_D[P \leftrightarrow Q]\), as required.

**Dominating interleaving**

The semantic equation for this operator is given as follows:

\[
\mathcal{F}_D[P \downarrow Q] = \{ (s, \mathbb{N}) \mid \exists u, w \bullet (u, w) = unzip_{P, Q} s \\
\land (u, \mathbb{N}) \in \mathcal{F}_D[P] \\
\land (u, \mathbb{N}) \in \mathcal{F}_D[Q] \}
\]

\[
unzip_{P, Q}(\langle \rangle) = (\langle \rangle, \langle \rangle)
\]
unzip_{P,Q}(s \prec \langle (t, a) \rangle) = (u \prec \langle (t, a) \rangle, w)$ if $(u \prec \langle (t, a) \rangle, \{ \}) \in \mathcal{F}_D[P]$

\[ \land (\text{end}(w) < t) \land (\text{end}(u) < t) \land \exists \varepsilon \cdot (u, [t, t+\varepsilon] \times \{a\}) \in \mathcal{F}_D[P] \]

$\neg (u, w \prec \langle (t, a) \rangle)$ if the above condition does not hold, and

\[ (w \prec \langle (t, a) \rangle, \langle \rangle) \in \mathcal{F}_D[Q] \land (\text{end}(u) < t) \land (\text{end}(w) < t) \land \exists \varepsilon \cdot (w, [t, t+\varepsilon] \times \{a\}) \in \mathcal{F}_D[P] \]

else undefined

where $(u, w) = unzip_{P,Q}(s)$. The function unzip on an infinite trace is the pairwise limit of unzip on all the approximations to the trace.

We must prove that $\mathcal{F}_D[P + || Q]$ lies in $TM_D$, under the assumption that $\mathcal{F}_D[P]$ and $\mathcal{F}_D[Q]$ are both contained in $TM_D$.

We first observe that

- axioms (1)–(3) are met
- $\mathcal{F}_D[P + || Q]$ is equal to its closure, since every approximation to an infinite behaviour must have been generated the same way.

**Axiom (4)**

To establish axiom (4), consider $(s, \mathcal{N}) \in \mathcal{F}_D[P + || Q]$. Let $(u, w) = unzip_{P,Q}(s)$; this must be defined, since $(s, \mathcal{N})$ is in the semantic set $\mathcal{F}_D[P + || Q]$. We therefore obtain that $(u, \mathcal{N}) \in \mathcal{F}_D[P]$ and $(w, \mathcal{N}) \in \mathcal{F}_D[Q]$. These may be extended to total behaviours $(u, \mathcal{N}_P)$ and $(w, \mathcal{N}_Q)$ for $P$ and $Q$, respectively. Let $\mathcal{N}' = \mathcal{N}_P \cap \mathcal{N}_Q$. We have $\mathcal{N} \subseteq \mathcal{N}'$, and $(s, \mathcal{N}') \in \mathcal{F}_D[P + || Q]$. We show that (C1) and (C2) hold for $\mathcal{N}'$.

Consider $(t, a) \notin \mathcal{N}'$. Then either $(t, a) \notin \mathcal{N}_P$ or $(t, a) \notin \mathcal{N}_Q$.

(1) If $(t, a) \notin \mathcal{N}_P$, then $(u < t \prec \langle (t, a) \rangle, \mathcal{N}_P \prec \langle t \rangle) \in \mathcal{F}_D[P]$, and also $(w < t, \mathcal{N}_Q \prec \langle t \rangle) \in \mathcal{F}_D[Q]$. Now $(u < t, w < t) = unzip_{P,Q}(s < t)$; also, the unique total refusal for $u < t$ in $P$ must agree with $\mathcal{N}_P$ up to time $\text{begin}(u > t)$, and hence will not contain $(t, a)$; thus no refusal of $P$ whose corresponding trace is $u < t$ can contain $(t, a)$. This yields $\neg \exists \varepsilon \cdot (u, [t, t+\varepsilon] \times \{a\}) \in \mathcal{F}_D[P]$. Thus by the definition of unzip we obtain that $unzip_{P,Q}(s < t \prec \langle (t, a) \rangle) = (u < t \prec \langle (t, a) \rangle, w < t)$, and so it follows from the semantic equation for $\oplus$ that $(s < t \prec \langle (t, a) \rangle, \mathcal{N}' \prec \langle t \rangle) \in \mathcal{F}_D[P + || Q]$.

(2) If $(t, a) \in \mathcal{N}_P$, then $(t, a) \notin \mathcal{N}_Q$.

(i) if $(u < t \prec \langle (t, a) \rangle)$ is traces$(P)$ and $\text{end}(w) < t$, then $unzip_{P,Q}(s < t \prec \langle (t, a) \rangle) = (u < t \prec \langle (t, a) \rangle, w < t)$, and so $(s < t \prec \langle (t, a) \rangle, \mathcal{N}' \prec \langle t \rangle) \in \mathcal{F}_D[P + || Q]$. Otherwise
(ii) \( \text{unzip}_{P, Q}(s \triangleleft t \triangleright \langle (t, a) \rangle) = (u \triangleleft t, w \triangleleft t \triangleright \langle (t, a) \rangle) \), since we have (by reasoning entirely similar to the previous case) that \( (w \triangleleft t \triangleright \langle (t, a) \rangle) \in \text{traces}(Q) \), and
\[
\neg \exists \epsilon (w \triangleleft [t, t + \epsilon) \times \{a\}) \in \mathcal{F}_D[P].
\]
Again we obtain \( (s \triangleleft t \triangleright \langle (t, a) \rangle, \ N' \triangleleft t) \in \mathcal{F}_D[P \leftrightarrow Q] \).

Thus condition (C1) holds.

We now establish condition (C2). Assume \( t > 0 \) and \( \neg \exists \epsilon > 0 \bullet ((t - \epsilon, t) \times \{a\}) \subseteq \mathbb{N}' \).
Then \( \text{unzip}_{P, Q}(s \triangleleft [t, t + \epsilon) \times \{a\}) \in \mathbb{N}' \), \( (u \triangleleft [t, t + \epsilon) \times \{a\}) \in \mathcal{F}_D[P] \), and so \( (s \triangleleft t \triangleright \langle (t, a) \rangle, \ N' \triangleleft t) \in \mathcal{F}_D[P \leftrightarrow Q] \).

Thus (C2) is satisfied.

Finite variability

Finite variability follows straightforwardly: if \( \{s_i\} \) is an increasing chain of traces of \( P \leftrightarrow Q \), then \( \{\text{unzip}_{P, Q}(s_i)\} \) produces two increasing chains \( \{u_i\} \) and \( \{w_i\} \). Both \( P \) and \( Q \) are finitely variable, so both of these chains must have limits which are traces; hence the sequence to which \( \{s_i\} \) tends must also be a trace.

Finite maximality

Finally, we must prove that \( P \leftrightarrow Q \) meets predicate (M). Let \( (s, \mathbb{N}) \in \mathcal{F}_D[P \leftrightarrow Q] \), and consider \( (t, a) \in \mathbb{N} \) with \( (s \triangleleft \langle (t, a) \rangle, \{\} \) \in \mathcal{F}_D[P \leftrightarrow Q] \).

Now, if \( \text{unzip}_{P, Q}(s \triangleleft \langle (t, a) \rangle) = (u \triangleleft \langle (t, a) \rangle, w) \), then \( u \triangleleft \langle (t, a) \rangle \in \text{traces}(P) \), and either \( \text{end}(w) < t \) or \( \neg \exists \epsilon > 0 \bullet (u, [t, t + \epsilon) \times \{a\}) \in \mathcal{F}_D[P] \).

Hiding

The semantics is given as follows:

\[
\mathcal{F}_D[P\setminus A] = \{(s\setminus A, \mathbb{N}) | (s, \mathbb{N} \cup [0, \infty) \times A) \in \mathcal{F}_D[P] \land \text{acc}_{P, A}(s)\}
\]

\[
\text{acc}_{P, A}(s) = \forall u, w, t, a \bullet s = u \triangleleft \langle (t, a) \rangle \triangleright w \Rightarrow
\]

\[
(S, A) \neq \{(\}
\]

\[
\land (a = c(S, A)(u)(t))
\]
\[ \forall (\neg \exists e \bullet (u, (t-e, t) \times \{a\}) \in \mathcal{F}_D[P] \]
\[ \land \ \text{end}(u) < t \]
\[ \land \ a \notin A) \]
\[ \lor \]
\[ (S_{F,A}(u)(t) = \{ \} \]
\[ \land (\exists e \bullet (u < t, [t-e, t) \times \{a\}) \in \mathcal{F}_D[P] \land (u^t) \setminus A = \langle \rangle \]
\[ \lor (u^t) \setminus A \neq \langle \rangle \]
\[ \lor \ \text{end}(u) < t) )) \]

where

\[ S_{F,A}(u)(t) = \{ b \in A \mid (u \sim \langle t, b \rangle, [0, t) \times A) \in \mathcal{F}_D[P] \} \]

We prove that, if \( \mathcal{F}_D[P] \) is in \( TM_D \), then so is \( \mathcal{F}_D[P \setminus \{A\}] \).

It follows straightforwardly that

- Axiom (3) holds.
- The set is equal to its own closure; this will follow from the fact (proved below) that visible traces may arise in exactly one way, so any infinite trace will be possible if its approximations are, since the traces giving rise to the approximations will have a limit, which gives rise to the infinite trace.

**Axiom (1)**

To show that \( (\langle \rangle, \{ \}) \in \mathcal{F}_D[P \setminus \{A\}] \), we must find some timed trace \( w \) such that \( (u, [0, \infty) \times A) \in \mathcal{F}_D[P] \) and \( acc_{F,A}(u) \).

Define

\[ s_0 = \langle \rangle \]
\[ N_0 = \{ \} \]
\[ N_n' = \text{the total extension of } (s_n, N_n) \]
\[ t_n = \sup \{ t \mid [0, t) \times A \subseteq N_n' \} \]
\[ a_n = c(A \setminus \sigma(N_n'^{\uparrow t_n})) \text{ if } t_n < \infty \]
\[ s_{n+1} = s_n \sim \langle (t_n, a_n) \rangle \text{ if } t_n < \infty \]
\[ = s_n \text{ otherwise} \]
\[ N_{n+1} = [0, t_n) \times A \]

Then the \( (s_n, N_n) \) are chain under \( \leq \) in \( \mathcal{F}_D[P] \), and so there is some sequence of timed events \( u \) such that every \( s_n \) is a prefix of \( u \). Furthermore, we must have \( acc_{F,A}(u) \), and
If \( \#u < \infty \), then there is some \( s_n = s_{n+1}(=u) \) with \( t_n = \infty \), and thus \( (s_n, [0, \infty) \times A) \in \mathcal{F}_D[P] \), and \( s_n \wedge A = \langle \rangle \); so \( \langle \rangle, \{ t \} \in \mathcal{F}_D[P \wedge A] \). Otherwise \( \#u = \infty \).

Since \( u \) is a timed trace \( ((t_0, a_0), (t_1, a_1), \ldots) \) we must have that \( t_n \to \infty \) because \( \mathcal{F}_D[P] \) is finitely variable. But then every failure of the form \( (u, [0, \infty) \times A) \in t \) is a behaviour of \( \mathcal{F}_D[P] \). Since \( \mathcal{F}_D[P] \) is closed, we must have that \( (u, [0, \infty) \times A) \in \mathcal{F}_D[P] \), and therefore that \( (u \wedge A, \{ t \}) \in \mathcal{F}_D[P \wedge A] \) as required.

**Axiom (2)**

Given \((s \rightarrow s', \mathbb{N}) \in \mathcal{F}_D[P \wedge A]\), we wish to show that \((s, \mathbb{N} \ll \text{begin}(s')) \in \mathcal{F}_D[P \wedge A]\).

If \((s \rightarrow s', \mathbb{N}) \in \mathcal{F}_D[P \wedge A]\), then it follows that there is some trace \( w_0 \) such that \( w_0 \wedge A = s \wedge s' \land (w_0, \mathbb{N} \cup [0, \infty) \times A) \in \mathcal{F}_D[P] \). Then there are \( w \) and \( w' \) such that \( w \wedge w' = w_0 \), \( w \wedge A = s \), \( w' \wedge A = s' \), and \( \text{begin}(s') = \text{begin}(w') \). By axiom (2) for \( \mathcal{F}_D[P] \), we obtain that \((w, (\mathbb{N} \cup [0, \infty) \times A) \ll \text{begin}(w')) \in \mathcal{F}_D[P]\).

Define
\[
\begin{align*}
S_0 & = w \\
\mathbb{N}_0 & = (\mathbb{N} \ll \text{begin}(s')) \cup [0, \text{begin}(s')) \times A \\
\mathbb{N}_n & = \text{the total extension of } (s_n, \mathbb{N}) \\
t_n & = \sup \{ t \mid [0, t) \times A \subseteq \mathbb{N}' \} \\
a_n & = c(A \setminus \sigma(\mathbb{N}' \cup [0, t])) \text{ if } t < \infty \\
s_{n+1} & = s_n - (t_n, a_n) \quad \text{if } t < \infty \\
& = s_n \quad \text{otherwise}
\end{align*}
\]

Then by an argument entirely similar to that used in the proof for the previous axiom, we have a trace \( u \) as the limit of the \( s_n \), such that \((u, \mathbb{N} \ll \text{begin}(s')) \cup [0, \infty) \times A) \in \mathcal{F}_D[P] \), yielding that \((s, \mathbb{N} \ll s') \in \mathcal{F}_D[P \wedge A] \) (since \( u \wedge A = s \), and \( \text{acc}_{P, A}(u) \)), as required.

**Axiom (4)**

Consider \((s, \mathbb{N}) \in \mathcal{F}_D[P \wedge A]\). Then there is some trace \( w \) such that \( w \wedge A = s \) and \( (w, \mathbb{N} \cup [0, \infty) \times A) \in \mathcal{F}_D[P] \) and \( \text{acc}_{P, A}(w) \). Therefore there is a total extension \( (w', \mathbb{N}') \) in \( P \) such that \( \mathbb{N} \cup [0, \infty) \times A \subseteq \mathbb{N}' \).

We prove that \( \mathbb{N}' \) meets (C1) and (C2) for \( P \wedge A \).

Consider \((t, b) \notin \mathbb{N}' \) (so \( b \notin A \)). Then \((w' \ll t \rightarrow \langle t, b \rangle, \mathbb{N}' \ll \text{begin } (w')) \in \mathcal{F}_D[P]. \) Observe that \( S_{P, A}(w' \ll t)(t) = \{ \} \). We must consider the following possibilities:

1. If \( \exists e \cdot (w' \ll t, ([t-e, t) \times \{ b \}) \in \mathcal{F}_D[P] \) \((w' \uparrow t) \wedge A = \langle \rangle \), then \( \text{acc}_{P, A}(w' \ll t \rightarrow \langle t, b \rangle) \), and \((w' \ll \langle t, b \rangle, \mathbb{N}' \ll \text{begin } (t, b)) \in \mathcal{F}_D[P] \), so we have that \((w' \ll t \rightarrow \langle t, b \rangle, \mathbb{N}' \ll \text{begin } (t, b)) \in \mathcal{F}_D[P \wedge A] \) i.e. \((s \ll \langle t, b \rangle, \mathbb{N}' \ll \text{begin } (t, b)) \in \mathcal{F}_D[P \wedge A] \).
If \( w' \uparrow t \notin A \), then \( acc_{P,A}(w' \triangleleft t \prec (t,b)) \), and we proceed as in the previous case.

If \( w' \uparrow t \notin \prec \) (i.e. \( \text{end}(w' \triangleleft t) < t \)), then again \( acc_{P,A}(w' \triangleleft t \prec (t,b)) \), and we proceed as in the first case.

If none of the above cases apply, then \( w' \uparrow t \notin \prec \) and \( \text{end}(w' \triangleleft t) = t \), in this case \( SP,A(w' \triangleleft t) \notin \prec \), and \( SP,A(t) \notin \prec \), as required.

Thus \((C1)\) holds.

To establish \((C2)\) consider \( t > 0 \land \exists \varepsilon \cdot (t-\varepsilon,t) \times \{b\} \subseteq \varepsilon' \). Then we have that \( (w' \triangleleft t \prec (t,b)), \varepsilon' \triangleleft t \in \varepsilon D_{[P]} \). If \( SP,A(w' \triangleleft t) \notin \prec \), then \( \text{end}(w' \triangleleft t) < t \), and we have that \( acc_{P,A}(w' \triangleleft t \prec (t,b)) \). On the other hand, if \( SP,A(w' \triangleleft t) \notin \prec \), then \( \exists \varepsilon \cdot (w' \triangleleft t, [t-\varepsilon,t) \times \{b\}) \in \varepsilon D_{[P]} \) and \( b \notin A \), and \( \text{end}(w' \triangleleft t) < t \) we again obtain that \( acc_{P,A}(w' \triangleleft t \prec (t,b)) \). In either case, it follows that \( (s \triangleleft t \prec (t,b)), \varepsilon' \triangleleft t \in \varepsilon D_{[P \setminus A]} \), establishing \((C2)\). Hence axiom \((4)\) holds.

Finite variability

To establish finite variability, we prove that any trace of \( P \setminus A \) may be produced in exactly one way from the behaviours of \( P \). We establish this by proving that if \( s,s' \in \text{traces}(P \setminus A) \) and \( s < s' \), and we have traces \( w,w' \in \text{traces}(P) \) with \( acc_{P,A}(w) \) and \( acc_{P,A}(w') \) and \( \text{last}(w) \notin A \) and \( \text{last}(w') \notin A \) (to remove trailing hidden events from non-empty traces) and \( w \setminus A = s \) and \( w' \setminus A = s' \), then \( w \equiv w' \). So we will assume the antecedent, and try to establish that \( w \equiv w' \).

Assume that \( w < w' \). Let \( w_0 \) be their longest common prefix. Then \( w = w_0 \circ w''' \) with \( w'' < \prec \), and \( w' = w_0 \circ w'''' \).

If \( w''' = \prec \) then \( w \setminus A \neq w' \setminus A \), which contradicts \( s < s' \). Hence \( w'''' < \prec \).

Let \( (t,a) = \text{head}(w') \), and let \( (t',a') = \text{head}(w'') \). Then \( a \neq a' \lor t \neq t' \). There are four possibilities for \( a, a' \):

(i) \( a \in A \) and \( a' \notin A \). If \( t < t' \) then \( (w_0 \circ (t', a')) \setminus [(0,t') \times A] \notin \varepsilon D_{[P]} \), yielding a contradiction. The assumption that \( t > t' \) similarly yields a contradiction.

Hence \( t = t' \). Then \( SP,A(w_0)(t) \notin \prec \), and so \( a = a' = c(SP,A(w_0))(t) \), yielding a contradiction.

(ii) \( a \notin A \) and \( a' \notin A \). Then \( w \setminus A \neq w' \setminus A \), yielding a contradiction.

(iii) \( a \in A \) and \( a' \notin A \). If \( t < t' \), then \( (w_0 \circ (t', a')) \setminus [(0,t') \times A] \notin \varepsilon D_{[P]} \), yielding a contradiction. If \( t > t' \), then \( w \setminus A \neq w' \setminus A \), also yielding a contradiction. Hence \( t = t' \).

We have that \( SP,A(w_0)(t) \notin \prec \), thus \( \exists \varepsilon \cdot (w_0, [(t-\varepsilon,t) \times \{a'\}] \in \varepsilon D_{[P]} \) \land \( \text{end}(w_0) < t \). The next visible action in \( w \) must be \( a' \), since \( w \setminus A \neq w' \setminus A \), and \( w \) has no trailing \( A \)’s. Thus \( w = w_0 \circ (t,a) \prec (t,a') \prec (t,a) \prec w_0' \), for some \( w_0 \) with \( w_0 \uparrow t = w_0' \). If \( acc_{P,A}(w_0 \circ (t,a) \prec (t,a') \prec (t,a) \prec w_0') \), then either \( SP,A(w_0 \circ (t,a) \prec (t,a) \prec w_0') \notin \prec \) and \( \text{end}(w_0 \circ (t,a) \prec (t,a) \prec w_0') < t \), which yields a contradiction, or else \( SP,A(w_0 \circ (t,a) \prec (t,a) \prec w_0') = \prec \). In this case, either \( \exists \varepsilon \cdot (w_0, [(t-\varepsilon,t) \times \{b\}] \in \varepsilon D_{[P]} \), which is not the case, or else \( w_0 \circ (t,a) \prec (t,a) \prec w_0') \uparrow t \setminus A \notin \prec \), which is not the case, or else
end \( w_0 \sim \langle (t, a) \rangle \sim w_0' \) \(< t \), which is also not the case. Hence a contradiction has resulted, and so this case is not possible.

(iv) \( a \not\in A \) and \( a' \in A \). An entirely similar argument to the previous case yields a contradiction again.

Therefore the assumption that \( w \not\subseteq w' \) is false, yielding the required result, from which finite variability follows.

**Finite maximality**

Finally, we wish to show that (M) holds of \( \mathcal{F}_D[P \setminus A] \). Consider \( (s, N) \in \mathcal{F}_D[P \setminus A] \), and \( (t, b) \in \mathbb{N} \) such that \( (s \sim \langle (t, b) \rangle, \{ \}) \in \mathcal{F}_D[P \setminus A] \). Observe that \( b \not\in A \). Let \( w \) be such that \( w[A] = s \) and \( acc_{P, A}(w) \sim \langle (t, b) \rangle \).

If \( S_{P, A}(w)(t) \neq \{ \} \), then \( \exists \alpha \in (w[A, [t-\delta, t) \times \{ b \}) \in \mathcal{F}_D[P] \), and \( end(w) < t \), and so \( \exists t_1 \leadsto end(w) \leq t_1 < t \land (t_1, t) \times \{ b \} \cap \mathbb{N} = \{ \} \), since (M) holds for \( P \). The result follows directly, since \( end(s) \leq end(w) \).

If \( S_{P, A}(w)(t) = \{ \} \), then it follows from \( (s, N) \in \mathcal{F}_D[P \setminus A] \) that there is some \( w' \) such that \( w'[A] = s \) and \( (w' \cup [0, \infty) \times A) \in \mathcal{F}_D[P] \) and \( acc_{P, A}(w') \). Then by the result used above for establishing finite variability, we have that \( w \subseteq w' \). Now since \( S_{P, A}(w)(t) = \{ \} \), we have that \( begin(w'[w \sim \langle t \rangle] > t) \), so \( w \not\subseteq t = w' \not\subseteq t \). Thus \( (w[N \cup (begin(w'[w \sim \langle t \rangle] > t)] \in \mathcal{F}_D[P] \). Also \( (w \sim \langle (t, b) \rangle, \{ \}) \in \mathcal{F}_D[P] \), and \( (t, b) \in \mathbb{N} \not\subseteq (begin(w'[w \sim \langle t \rangle] > t) \). Thus by (M) for \( P \) we obtain \( \exists t_1 \leadsto end(w) \leq t_1 < t \land (t_1, t) \times \{ b \} \cap \mathbb{N} \subseteq (begin(w'[w \sim \langle t \rangle] > t) = \{ \} \). The result follows again, since \( end(s) \leq end(w) \). Thus \( P \) meets (M) implies \( P \setminus A \) meets (M), as required.

**References**


