Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

J.W. de Bakker, J.-J. Ch. Meyer, E.-R. Olderog

Infinite streams and finite observations in the semantics of uniform concurrency

(preliminary version)

Department of Computer Science Report CS-R8508 April

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

Infinite Streams and Finite Observations in the Semantics of Uniform Concurrency

(preliminary version)

J.W. de Bakker
Centre for Mathematics and Computer Science &
Free University, Amsterdam

J.-J. Ch. Meyer
Free University, Amsterdam

E.-R. Olderog
Christian-Albrechts-Universität Kiel

Two ways of assigning meaning to a language with uniform concurrency are presented and compared. The language has uninterpreted elementary actions from which statements are composed using sequential composition, nondeterministic choice, parallel composition with communication, and recursion. The first semantics uses infinite streams in the sense which is a refinement of the linear time semantics of De Bakker et al. The second semantics uses the finite observations of Hoare et al., situated "in between" the divergence and readiness semantics of Olderog & Hoare. It is shown that the two models are isomorphic and that this isomorphism induces an equivalence result between the two semantics.

1980 *Mathematics Subject Classification:* 68B10, 68C01
1982 CR Categories: D.3.1, F.3.2, F.3.3. 69D41, 69F32, 69F33
Key Words & Phrases: concurrency, denotational semantics, streams, uniform languages, observations, Smyth ordering, parallel composition, topological closedness.

Note:

1. The research of J.W. de Bakker is partially supported by ESPRIT project 415: Parallel Architectures and Languages
2. This report will be published in the Proceedings 12^th International Colloquium on Automata, Languages and Programming

Report CS-R8508
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

1. Introduction

Infinite streams of actions or states provide a natural and clear concept for describing the behaviour of non-terminating concurrent processes [Br, Ni]. The supporting mathematics, however, tends to get complicated even if some simplifying assumptions on the admissible sets of streams are possible [Br, BBKM]. On the other hand, finite traces of actions or more generally finite observations like ready or failure pairs typically require a rather simple mathematics to justify the semantic constructions [BHR, FLP, OH2]. However, these constructions often seem more "ad hoc" and less clear conceptually. Also, finite observations are in general less expressive than infinite streams, for example in the presence of fairness [He2, OH2].

Our paper now presents an interesting case where infinite streams and finite observations are equally expressive in the sense of an isomorphism. We establish our results for a core language $\mathfrak{L}$ of uniform or schematic concurrency [BMOZ] involving uninterpreted atomic actions, sequential composition, nondeterministic choice (local nondeterminism), parallel composition (merge) with communication and recursion. For $\mathfrak{L}$ we introduce two versions of (denotational)linear time semantics [BBKM].

The first semantics $\mathfrak{D}_{str}$ is based on finite and infinite streams of actions. $\mathfrak{D}_{str}$ refines the linear time semantics LT developed in [BBKM] in that it deals more satisfactorily with recursion. This is achieved by using a Smyth-like ordering on sets of streams. When developing the semantics $\mathfrak{D}_{str}$ we shall carefully motivate the conditions of flatness and topological closedness for our powerdomain of streams. In particular, topological closedness will be crucial for proving the continuity of the semantic operators. Unfortunately, these proofs are rather complicated [Me, BBKM].

The second semantics $\mathfrak{D}_{obs}$ fits into the specification-oriented approach to the semantics of concurrent processes [OH1/2]- a generalization of the specific failure semantics in [BHR]. The starting point for the approach is a simple correctness criterion for processes: a process P satisfies a specification S, denoted by $P \,\mathbf{sat}\, S$, if every observation we can make about P is allowed by S. An observation is a finitely representable information about the computational behaviour of processes. Examples of observations are (finite) traces, traces with divergence information, ready pairs and failure pairs leading to the (increasingly sophisticated) trace, divergence, readiness and failure semantics for concurrent processes [OH2]. Our specific observation semantics $\mathfrak{D}_{obs}$ for $\mathfrak{L}$ can be seen as "in between" the divergence and the readiness semantics of [OH2].

Our main result is that both approaches to the semantics of $\mathfrak{L}$ are isomorphic. This isomorphism has various benefits in the mutual understanding of both approaches:

- the concepts in $\mathfrak{D}_{str}$ have a natural translation into $\mathfrak{D}_{obs}$: for example, topological closedness in $\mathfrak{D}_{str}$ gets translated into prefix closedness in $\mathfrak{D}_{obs}$,
- through this translation the constructions for $\mathfrak{D}_{obs}$ become clear conceptually,
- most important perhaps, proofs of continuity of the semantic operators in $\mathfrak{D}_{str}$ now become very simple via the isomorphism to $\mathfrak{D}_{obs}$, involving only the notion of domain finite relations on the side of observations [OH1/2]. Thus through the idea of observation we can circumvent the technically difficult continuity proofs of [BBKM, Me].

Our paper is backed up by the reports [Me] and [OH2]; the linking isomorphism result will be proved fully in its final version.

2. The language $\mathfrak{L}$

Let A be a finite set of *actions,* with $a,b \in A$, $* : A \times A \xrightarrow{part} A$ be a partial binary operation on A called *communication function,* and *Pvar* be a set of *process variables,* with $x,y \in Pvar$. Then the set of *(concurrent) processes* $\mathfrak{L}$, with $P,Q \in \mathfrak{L}$, is given by the following BNF-syntax:

2.1. Definition.

$$P \ ::= a \,|\, P;Q \,|\, P \text{ or } Q \,|\, P \,\|\, Q \,|\, x \,|\, \mu x[P]$$

2.2. REMARKS. Every action $a \in A$ denotes a process, the one which finishes (terminates successfully) after performing a. $P;Q$ denotes *sequential composition* such that Q starts once P has finished. P **or** Q denotes *nondeterministic choice,* also known as *local nondeterminism* [FHLR]. $P \| Q$ denotes *communication merge* (cf. [BK]) where parallel composition is modelled by arbitrary interleaving plus communication between those actions a of P and b of Q for which $a*b$ is defined. For example, if only $b*c$ is defined, we will obtain the following equation in our semantics:

$$(a;b)\|c \;=\; a;b;c \textbf{ or } a;c;b; \textbf{ or } c;a;b \textbf{ or } a;(b*c).$$

Communication merge is inspired by [Mi2, BK, Wi], though we do not assume any algebraic property of $*$.

By varying the communication function $*$, we can express more familiar notions of parallel composition like shuffle (arbitrary merge) or merge with binary communication as in CCS[Mi].

Starting from actions $a \in A$, the operators ;, **or** , and $\|$ can only define concurrent processes P with finite semantic behaviour; infinite behaviours require processes P involving recursion, expressed here by the μ-construct [dB].

3. THE STREAM SEMANTICS $\mathcal{D}_{str}$

Let $\perp \notin A$. Then we define the set of *streams* $Str(A)$, with $u,v,w \in Str(A)$, as follows [Br]:

3.1. DEFINITION. $Str(A) = A^* \cup A^\omega \cup A^*.\{\perp\}$.

3.2. REMARKS. $Str(A)$ includes the set $A^\infty = A^* \cup A^\omega$ of finite and infinite words over A [Ni], called here *finished* and *infinite streams,* respectively, and additionally the set $A^*.\{\perp\}$ of *unfinished streams.* The linear time semantics LT of [BBKM] was entirely based on A^∞. The reason for including unfinished streams $u.\perp$ as well is that they allow a more satisfactory treatment of recursion (see Proposition 3.28).

Let ϵ denote the *empty* (finished) stream, $\leqslant$ the *prefix relation* and $<$ *the proper prefix relation* over streams, and $|u|$ the *length* of a stream u, with $|u| = \infty$ for infinite u*'s*. Additionally we use the following approximation relation $\sqsubseteq$:

3.3. DEFINITION. $u \sqsubseteq v$ iff the following holds
- if u is finished or infinite then $u = v$,
- if u is unfinished, i.e., of the form $u = u'\perp$, then $u' \leqslant v$.

3.4. EXAMPLES. $a \leqslant a\perp$, $a\perp \not\leqslant a$, $a\perp \not\leqslant ab$ but $a \not\sqsubseteq a\perp$, $a\perp \sqsubseteq a$, $a\perp \sqsubseteq ab$.

Consider for a moment an arbitrary cpo $(C, \sqsubseteq_c, \perp_c)$ and a subset $S \subseteq C$.

3.5. DEFINITION. S is called *flat* if $x \sqsubseteq_c y$ implies $x = y$ for all $x,y \in S$. If $C \setminus \{\perp_c\}$ is flat, the cpo $(C, \sqsubseteq_c, \perp_c)$ itself is called flat.

3.6. PROPOSITION. $(Str(A), \sqsubseteq, \perp)$ *is a non-flat cpo.*

To provide meaning to concurrent processes $P \in \mathcal{L}$ we need (certain) *sets* of streams. Let $\mathcal{P}(Str(A))$ denote the powerset of streams, with typical elements $X,Y \in \mathcal{P}(Str(A))$. Then we will use the following *Smyth relation* [Sm]:

3.7. DEFINITION. $X \sqsubseteq_S Y$ if $\forall v \in Y\ \exists u \in X$: $u \sqsubseteq v$.

3.8. REMARK. $X \supseteq Y$ implies $X \sqsubseteq_S Y$.

It is well-known that the Smyth relation $\sqsubseteq_S$ is not antisymmetric and thus not a partial order on non-flat domains like $\mathcal{P}(Str(A))$ [Ba, Br]. But the Smyth relation is a pre-order which generates an equivalence relation $\equiv_S$ on $\mathcal{P}(Str(A))$:

$$X \equiv_S Y \quad \text{iff} \quad X \sqsubseteq_S Y \quad \text{and} \quad Y \sqsubseteq_S X.$$

What are the sets identified by $\equiv_S$?

3.9. DEFINITION. $\min_S(X) = \{v \in X \mid \neg \exists u \in X \colon u \sqsubseteq v \wedge u \neq v\}$ is the set of minimal streams in X.
Then $X \equiv_S Y$ if and only if $\min_S(X) = \min_S(Y)$. Thus the sets $\min_S(X)$ form a system of representatives of the equivalence classes under $\equiv_S$. Note that $\min_S(X)$ is flat.

3.10. DEFINITION. $\mathcal{P}_f(Str(A))$ is the set of all flat subsets of $Str(A)$.

3.11. PROPOSITION. *$\mathcal{P}(Str(A))/\equiv_S$ is isomorphic to $\mathcal{P}_f(Str(A))$.*

3.12. PROPOSITION. *$(\mathcal{P}_f(Str(A)), \sqsubseteq_S, \{\perp\})$ is a cpo.*

The proof can be found in [Ba] (see [Me]). Next, we need some auxiliary operators on streams.

CONCATENATION $u \cdot v$: For $u, v \in A^\infty = A^* \cup A^\omega$ the concatenation $u \cdot v$ is well-known from the theory of infinitary languages [Ni]. We extend this definition to arbitrary streams by imposing the equation $\perp \cdot v = \perp$.

COMMUNICATION MERGE $u \| v$: Here we consider only finite streams $u, v \in A^* \cup A^* \cdot \{\perp\}$. Then $u \| v$ is a *set* of (finite) streams defined by

$$u \| v \;=\; u \,⫡\, v \cup v \,⫡\, u \cup u \mid v$$

where recursively $\epsilon ⫡ v = \{v\}$, $\perp ⫡ v = \{\perp\}$, $a \cdot u ⫡ v = a \cdot (u \| v)$ and $au \mid bv = (a*b) \cdot (u \| v)$ provided $a*b$ is defined; in all other cases $u \mid v = \emptyset$. This finite recursive definition of $\|$ using $⫡$ and $\mid$ is due to [BK].
To lift these definitions to flat sets of streams, we use the operator $\min_S$ of Definition 3.9 and the following notion of *n-th approximation* $u^{[n]}$, $n \geqslant 0$, for streams u: $u^{[n]} = u$ if $|u| < n$ and $u^{[n]} = u'\perp$ if $|u| \geqslant n$ and $u' \leqslant u$ with $|u'| = n$. We extend this definition pointwise to subsets $X \subseteq Str(A)$ by putting $X^{[n]} = \{u^{[n]} \mid u \in X\}$. Now let $X, Y \in \mathcal{P}_f(Str(A))$.

SEQUENTIAL COMPOSITION

$$X ;^{str} Y \;=\; \min_S(\{u \cdot v \mid u \in X \text{ and } v \in Y\})$$

LOCAL NONDETERMINISM

$$X \textbf{ or } Y \;=\; \min_S(X \cup Y)$$

PARALLEL COMPOSITION
For $X, Y \subseteq A^* \cup A^*.\{\perp\}$ (involving only finite streams) we set

$$X \|^{str} Y \;=\; \min_S(\{w \in Str(A) \mid \exists u \in X, v \in Y \colon w \in u \| v\})$$

and for arbitrary flat $X, Y \subseteq Str(A)$ we work with semantic approximations:

$$X \|^{str} Y \;=\; \bigsqcup_{n=0}^{\infty}{}_S (X^{[n]} \|^{str} Y^{[n]}).$$

3.13. THEOREM. *The semantic operators*

$$op^{str} \colon \mathcal{P}_f(Str(A)) \times \mathcal{P}_f(Str(A)) \rightarrow \mathcal{P}_f(Str(A))$$

with op $\in\{;\ ,\ \mathbf{or},\ \|\}$ *are both well-defined and* $\sqsubseteq_S$*-monotonic.*

The proof is given in [Me]. Showing monotonicity is not trivial for ; and $\|$. To provide meaning to recursive processes too, we will have to show that the semantic operators op^{str} are also continuous.

3.14. THEOREM. $\mathbf{or}^{str}$ *is continuous under* $\sqsubseteq_S$.

Unfortunately, the operators $;^{str}$ and $\|^{str}$ are not continuous on arbitrary flat sets of streams. To rescue the continuity of ; and $\|$, we will restrict ourselves to closed sets of streams.

3.15. DEFINITION. **[Ba].** $X\subseteq Str(A)$ is *closed* if for every infinitely often increasing chain $<u_n>_{n\geqslant 0}$ of unfinished streams in $Str(A)$ the property

$$\forall n\geqslant 0\exists v_n\in X:\ u_n\sqsubseteq v_n$$

implies that the stream limit $\bigsqcup_{n=0}^{\infty} u_n\in X$.

At first sight this closedness property looks a bit technical, but it is not. We can show that it coincides with the clear concept of *topological closedness* w.r.t. the following metric topology on $Str(A)$.

3.16. DEFINITION. The *distance* d: $Str(A)\times Str(A)\rightarrow[0,1]$ is given by

$$d(u,v)\ =\ 2^{-\min\{n\,|\,u^{[n]}\neq v^{[n]}\}}$$

with the convention that $2^{-\infty}=0$.

3.17. EXAMPLES. $d(abc,aba)=2^{-3}, d(a^n,a^\omega)=2^{-n-1}$.

3.18. PROPOSITION. *$(Str(A),d)$ is a complete metric space.*

Thus, we can talk of *Cauchy sequences* $<u_n>_{n\geqslant 0}$ of streams, their *topological limits* and of *topologically closed* sets $X\subseteq Str(A)$, i.e. where every Cauchy sequence $<u_n>_{n\geqslant 0}$ with $u_n\in X$ has its topological limit (which exists in $Str(A)$) inside X.

3.19. PROPOSITION. *A subset $X\subseteq Str(A)$ is closed iff X is topologically closed.*

3.20. EXAMPLES. $X=\{a^n ba^\omega|n\geqslant 0\}\cup\{a^\omega\}$ is (topologically) closed, but $Y=\{a^n ba^\omega|n\geqslant 0\}$ is not.

Note that Y typically arises through a *fair merge* of $Y_1=\{a^\omega\}$ and $Y_2=\{b\}$. Hence notions like fairness or eventuality are not expressible using only (topologically) closed sets of streams [He2, Me].

3.21. DEFINITION. $\mathcal{P}_{ncf}(Str(A))$ is the set of all non-empty, closed and flat subsets of $Str(A)$.

The following lemma is crucial for the further development:

3.22. LEMMA. *If $<X_n>_{n\geqslant 0}$ is a $\sqsubseteq_S$-chain of sets $X_n\in\mathcal{P}_{ncf}(Str(A))$, then $\bigsqcup_{n=0}^{\infty}{}_S X_n\neq\varnothing$.*

The proof is rather involved [Me]. We can now establish the following results:

3.23. PROPOSITION. *$(\mathcal{P}_{ncf}(Str(A)),\ \sqsubseteq_S,\{\perp\})$ is a cpo.*

3.24. THEOREM. *The operators $;^{str}$ and $\|^{str}$, when restricted to $\mathcal{P}_{ncf}(Str(A))$, are continuous under* $\sqsubseteq_S$.

The proof uses Lemma 3.2.2 and otherwise follows [BBKM]; the case of $||$ is difficult.

3.25. REMARK. Lemma 3.22 and Theorem 3.24 do not hold, in general, for *infinite* sets A of actions.
We can now define the denotational stream semantics $\mathcal{D}_{Str}$ for $\mathcal{L}$. The set of environments is given by $\Gamma = Pvar \rightarrow \mathcal{P}_{ncf}(Str(A))$, with $\gamma \in \Gamma$. Let, as before, X,Y range over $\mathcal{P}_{ncf}(Str(A))$, and let $\gamma' = \gamma\{X/x\}$ be as γ but with $\gamma'(x) = X$. For a $\sqsubseteq_S$ -continuous function Φ from $\mathcal{P}_{ncf}(Str(A))$ to $\mathcal{P}_{ncf}(Str(A))$ let $\mu\Phi$ denote its least fixed point.

3.26. DEFINITION. The semantic mapping

$$\mathcal{D}_{Str}[\![\cdot]\!]: \mathcal{L} \rightarrow (\Gamma \rightarrow \mathcal{P}_{ncf}(Str(A)))$$

is given by:
(i) $\mathcal{D}_{Str}[\![a]\!](\gamma) = \{a\}$
(ii) $\mathcal{D}_{Str}[\![P;Q]\!](\gamma) = \mathcal{D}_{Str}[\![P]\!](\gamma);^{str}\mathcal{D}_{Str}[\![Q]\!](\gamma)$
(iii) $\mathcal{D}_{Str}[\![P \,\mathbf{or}\, Q]\!](\gamma) = \mathcal{D}_{Str}[\![P]\!](\gamma)\mathbf{or}^{str}\mathcal{D}_{Str}[\![Q]\!](\gamma)$
(iv) $\mathcal{D}_{Str}[\![P \,||\, Q]\!](\gamma) = \mathcal{D}_{Str}[\![P]\!](\gamma)||^{str}\mathcal{D}_{Str}[\![Q]\!](\gamma)$
(v) $\mathcal{D}_{Str}[\![x]\!](\gamma) = \gamma(x)$
(vi) $\mathcal{D}_{Str}[\![\mu x[P]]\!](\gamma) = \mu\Phi_{P,\gamma}$ where $\Phi_{P,\gamma} = \lambda X \cdot \mathcal{D}_{Str}[\![P]\!](\gamma\{X/x\})$.

A process $P \in \mathcal{L}$ is called *guarded in x* whenever all occurrences of x in P are within subprocesses of P of the form $Q;(\ldots x \ldots)$. A process P is called *guarded* (cf. [Mil] or [Ni], where *Greibach* replaces guarded) whenever, for each recursive subprocess $\mu y[Q]$ of P we have that Q is guarded in y.

3.27. EXAMPLES. $\mu x[a;x \,\mathbf{or}\, b]$ and $\mu x[a;(x \,||\, b)]$ are guarded; $\mu x[x]$, $\mu x[x;a \,\mathbf{or}\, b]$ and $\mu x[x \,||\, b]$ are not.

3.28. PROPOSITION. *In the semantics $\mathcal{D}_{Str}$ all unguarded processes P (without free process variables) are identified:* $\mathcal{D}_{Str}[\![P]\!](\gamma) = \{\perp\}$.

This solution seems more attractive than the results in the linear time semantics LT of [BBKM]. For example, $LT[\![\mu x[x]]\!](\gamma) = A^{\infty}$ but (surprisingly) $LT[\![\mu x[x;b]]\!](\gamma) = A^{\omega}$.

4. THE OBSERVATION SEMANTICS $\mathcal{D}_{obs}$

4.1. Background.

Motivated by the failure semantics of [BHR], a new approach to the semantics of concurrent processes has been developed in [OH1/2]. It is called "specification-oriented" because it starts from the following simple concept of process correctness: a process P satisfies a specification S, abbreviated P **sat** S, if every observation we can make about P is allowed by S. The idea is that by varying the structure of observations we can express various types of process semantics and process correctness in a uniform way.

The principles of specification-oriented semantics are:

- an observation is a finitely representable information about the operational behaviour of processes,
- therefore the set of possible observations about a process enjoys some natural closure properties with respect to certain predecessor and successor observations,
- sets of observations are ordered by the nondeterminism ordering (reverse set-inclusion) [BHR],
- this ordering leads to a simple mathematics, in particular a very simple continuity argument for the language operators.

Let us now start with an example of a semantics- not treated in[OH2]- which fits into this framework. We use two distinct symbols $\surd, \uparrow \notin A$ to define the following set $Obs(A)$ of *observations,* with $h \in Obs(A)$:

4.2. DEFINITION. $Obs(A) = A^* \cup A^* \cdot \{\surd, \uparrow\}$.

4.3. REMARKS. Here observations are finite *traces* or *histories* over A and the extra symbols $\surd$ and $\uparrow$. representing *successful termination* [BHR] and *divergence* [OH2], respectively. Divergence $\uparrow$ stands for an infinite internal loop of a process generated by an unguarded recursion like $\mu x[x]$. Thus in spite of their finite representation, not all observations can be made effectively; a similar concession is also present in the concept of testing due to [dNH].

As for streams we let ϵ denote the *empty* history and $\leqslant$ the *prefix relation* between histories. Apart from $\leqslant$ we do not introduce any further relation on $Obs(A)$ which would correspond to $\sqsubseteq$ on $Str(A)$. Let $\mathcal{P}(Obs(A))$ denote the powerset of $Obs(A)$, with $H \in \mathcal{P}(Obs(A)))$.

4.4. DEFINITION. $H \subseteq Obs(A)$ is called *saturated* iff the following holds:

(i) H includes the *minimal observation*, i.e. $\epsilon \in H$,

(ii) H is *prefix closed*, i.e.

$$h \in H \quad \text{and} \quad h' \leqslant h \quad \text{imply} \quad h' \in H$$

(iii) H is *extensible*, i.e.

$$h \in H \setminus A^* \cdot \{\surd\} \quad \text{implies} \quad \exists \alpha \in A \cup \{\surd, \uparrow\}: h\alpha \in H$$

(iv) H treats divergence as *chaos*, i.e.

$$h\uparrow \in H \quad \text{and} \quad h' \in Obs(A) \quad \text{imply} \quad hh' \in H.$$

4.5. REMARKS. These closure properties are (partly) motivated by looking at saturated H's as the sets of possible observations about a concurrent process:

(i) As long as the process has not yet started, we only observe the empty history ϵ.

(ii) Whenever we have observed a history h, also all its prefixes h' are observable.

(iii) Only histories $h\surd$ indicate the successful termination of the observed process; for all other histories h some *extension* $\alpha \in A \cup \{\surd, \uparrow\}$ is certain to happen, but we do not know which one, by looking at h.

(iv) Identifying divergence $h\uparrow$ after a history h with the *chaotic closure* $h \cdot Obs(A)$ cannot be explained operationally, rather it originates from the desire to ban diverging processes from satisfying any reasonable specification. This idea is familiar from Dijkstra's weakest precondition semantics where a diverging program will not achieve any postcondition [Pl].

Properties (i), (ii) are typical conditions on traces to be found in [BHR, FLP, OH1/2]. Property (iii) is a new "linear version" of the extensibility condition in the readiness [OH2] or failure semantics [BHR]. Property (iv) is typical for a simple, but proper treatment of divergence [OH2]; without $\uparrow$ unsatisfactory results occur [BHR] akin to those in the LT semantics [BBKM] (cf. end of Section 3).

4.6. DEFINITION. $\mathcal{P}_{sat}(Obs(A))$ is the set of all saturated subsets of $Obs(A)$.

On $\mathcal{P}_{sat}(Obs(A))$ we introduce the following *nondeterminism order* $\sqsubseteq_N$ [BHR]:

4.7. DEFINITION. $H_1 \sqsubseteq_N H_2$ iff $H_1 \supseteq H_2$.

4.8. PROPOSITION. *$(\mathcal{P}_{sat}(Obs(A)), \supseteq, Obs(A))$ is a cpo.*

Proving the cpo property for $\mathcal{P}_{sat}(Obs(A))$ is much simpler than for $\mathcal{P}_{ncf}(Str(A))$: cf. Lemma 3.22. But what is the relationship between $\mathcal{P}_{ncf}(Str(A))$ and $\mathcal{P}_{sat}(Obs(A))$ anyway? This is the topic of the next section.

5. The isomorphism between streams and observations

We wish to relate the cpo's $(\mathcal{P}_{ncf}(Str(A)), \sqsubseteq_S, \{\perp\})$ and $(\mathcal{P}_{sat}(Obs(A)), \supseteq, Obs(A))$. To this end we define a mapping Ψ, first as

$$\Psi: Str(A) \to \mathcal{P}(Obs(A)).$$

For $u \in A^*$ and $v \in A^\omega$ let

$$\Psi(u) = \{h \in A^* \mid h \leqslant u\} \cup \{u\surd\}$$

$$\Psi(v) = \{h \in A^* \mid h \leqslant v\}$$

$$\Psi(u\perp) = \{h \in A^* \mid h \leqslant u\} \cup \{uh \mid h \in Obs(A)\}.$$

5.1. Remarks. A finished streams u is translated into the set of all its prefixes plus $u\surd$ with $\surd$ signalling successful termination of u, an infinite stream is translated into the set of all its *finite* prefixes, and an unfinished stream $u\perp$ is translated into the set of all prefixes of u plus the chaotic closure $u \cdot Obs(A)$ of divergence $u\uparrow$.

We extend Ψ pointwise to a mapping

$$\Psi: \mathcal{P}(Str(A)) \to \mathcal{P}(Obs(A))$$

by

$$\Psi(X) = \bigcup_{w \in X} \Psi(w).$$

5.2. Examples. $\Psi(\{ab\}) = \{\epsilon, a, ab, ab\surd\}$, $\Psi(\{a^\omega\}) = \{a^n \mid n \geqslant 0\}$, $\Psi(\{\perp\}) = Obs(A)$.

5.3. Theorem. *Ψ is an isomorphism from the cpo $(\mathcal{P}_{ncf}(Str(A)), \sqsubseteq_S, \{\perp\})$ onto the cpo $(\mathcal{P}_{sat}(Obs(A)), \supseteq, Obs(A))$, i.e. Ψ is bijective, yields $\Psi(\{\perp\}) = Obs(A)$ and strongly preserves the partial orders:*

$$X \sqsubseteq_S Y \quad \textit{iff} \quad \Psi(X) \supseteq \Psi(Y)$$

for all $X, Y \in \mathcal{P}_{ncf}(Str(A))$.

5.4. Remarks. $\mathcal{P}_{ncf}(Str(A))$ has been constructed through a chain of clear domain-theoretical notions: streams, sets of streams, Smyth relation, flatness, continuity, topological closure, non-emptiness. The introduction of $\mathcal{P}_{sat}(Obs(A))$ with its saturation property may seem more ad hoc. But the theorem now tells us that $\mathcal{P}_{sat}(Obs(A))$ can in fact be viewed as a *special representation of the general construction* $\mathcal{P}_{ncf}(Str(A))$.

This provides us with a new mutual understanding of the closedness properties in both domains: *topological closedness* on streams corresponds to taking *all finite prefixes* as observations, *flatness* of set of streams corresponds to the *chaotic closedness* on observations, *non-emptiness* of sets of streams does *not* simply correspond to the fact that saturated sets of observations include ϵ, but that in addition they are *extensible.*

Whereas the non-emptiness of (lubs of) sets of streams is a global property, the extensibility of observations is a local property where every observation $h \notin A^*\{\uparrow\}$ can be locally extended by another $\alpha \in A \cup \{\surd, \uparrow\}$. This issue of "global" vs. "local" hints at why it is more difficult to prove the cpo property for $\mathcal{P}_{ncf}(Str(A))$ than for $\mathcal{P}_{sat}(Obs(A))$.

6. THE OBSERVATION SEMANTICS $\mathcal{D}_{obs}$: CONTINUED

Let us now continue with the observation semantics $\mathcal{D}_{obs}$. For the operators of $\mathcal{D}_{obs}$ we could well provide indirect definitions using the previous isomorphism. But it will be more illuminating to discuss direct definitions because the ordering $\supseteq$ on sets of observations allows a very simple, uniform proof of (monotonicity and) continuity.

In fact, this uniform argument can be explained independently of the specific structure of observations. Consider two sets $\mathbb{X},\mathbb{Y}$ and a relation $R \subseteq \mathbb{X}\times\mathbb{Y}$. Then R induces an operator

$$op_R : \mathcal{P}(\mathbb{X})\rightarrow\mathcal{P}(\mathbb{Y})$$

on the subsets of $\mathbb{X}$ by taking for every $X\subseteq\mathbb{X}$ the *pointwise image* of X under R, i.e.

$$op_R(X) = \{y\in\mathbb{Y} | \exists x\in\mathbb{X}: xRy\}.$$

6.1. LEMMA [OH2]. *The operator* op_R *is* $\supseteq$*-monotonic. Moreover, if* R *is domain finite, i.e. if for every* $y\in\mathbb{Y}$ *there exist only finitely many* $x\in\mathbb{X}$ *with* xRy, op_R *is also* $\supseteq$*-continuous.*

Let us demonstrate the use of the lemma in the case of *sequential composition.* First we define the corresponding semantic operator

$$;^{obs} : \mathcal{P}_{sat}(Obs(A))\times\mathcal{P}_{sat}(Obs(A))\rightarrow\mathcal{P}_{sat}(Obs(A))$$

as follows:

$$\begin{aligned} H_1;^{obs}H_2 = \; & \{h_1 | h_1\in H_1 \text{ and } h_1 \text{ does not contain } \surd\} \\ & \cup\{h_1h_2 | h_1\surd\in H_1 \text{ and } h_2\in H_2\} \\ & \cup\{h_1h | h_1\uparrow\in H_1 \text{ and } h\in Obs(A)\} \end{aligned}$$

Well-definedness of $;^{obs}$ has to be checked separately. But monotonicity and continuity of $;^{obs}$ follow from the general Lemma 6.1. Taking $\mathbb{X}=Obs(A)\times Obs(A)$ and $\mathbb{Y}=Obs(A)$ we look for a domain finite relation $R\subseteq\mathbb{X}\times\mathbb{Y}$ such that

$$(*) \quad ;^{obs} = op_R\upharpoonright\mathcal{P}_{sat}(Obs(A))\times\mathcal{P}_{sat}(Obs(A)).$$

R can be read off from $;^{obs}$ as follows:

$(h_1,h_2)Rh$ iff

(i) h_1 does not contain $\surd$, $h_2=\epsilon$ and $h=h_1$, or

(ii) h_1 ends in $\surd$ and $h=(h_1\setminus\surd)\cdot h_2$, or

(iii) h_1 ends in $\uparrow$, $h_2=\epsilon$ and $h\in(h_1\setminus\uparrow)\cdot Obs(h)$

Here $h_1\setminus\surd$ and $h_1\setminus\uparrow$ result from h_1 by removing from h_1 the symbols $\surd$ or $\uparrow$, respectively. Clearly, this R is domain finite. Thus Lemma 6.1 implies:

6.2. PROPOSITION. *The operator* $;^{obs}$ *is monotonic and continuous under* $\supseteq$.

The discussion of the remaining operators will be brief. *Local nondeterminism* is just set-theoretic union

$$H_1 \mathbf{or}^{obs} H_2 = H_1\cup H_2.$$

$\mathbf{or}^{obs}$ is well-defined and (by Lemma 6.1) monotonic and continuous under $\supseteq$. *Parallel composition* is defined by

$$H_1\|^{obs}H_2 = \{h | \exists h_1\in H_1,\, h_2\in H_2:\, h\in h_1\|h_2\}$$

where $h_1\|h_2$ is a set of observations given, similarly to the stream definition in Section 3, by

$$h_1\|h_2 = h_1 \mathbin{\|\!\!\llcorner} h_2 \cup h_2 \mathbin{\|\!\!\llcorner} h_1 \cup h_1|h_2$$

with $\epsilon \mathbin{\|\!\!\lfloor} \epsilon = \{\epsilon\}$, $ah_1 \mathbin{\|\!\!\lfloor} h_2 = a\cdot(h_1 \| h_2)$, $\surd \mathbin{\|\!\!\lfloor} h_2 = \{h_2\}$, $\uparrow \mathbin{\|\!\!\lfloor} \epsilon = Obs(A)$ and with $ah_1 | bh_2 = (a*b)\cdot(h_1 \| h_2)$ provided $a*b$ is defined; in all other cases $h_1 \mathbin{\|\!\!\lfloor} h_2 = \emptyset$ and $h_1 | h_2 = \emptyset$. Lemma 6.1 yields:

6.3. PROPOSITION. *The operator* $\|^{obs}$ *is well-defined, monotonic and continuous under* $\supseteq$.

6.4. REMARKS. In the observation semantics the continuity proof for the operators $;^{obs}$, $\mathbf{or}^{obs}$, $\|^{obs}$ could be reduced to a simple test on domain finiteness. In the stream semantics the operators $;^{str}$ and $\|^{str}$ will *fail* such a test. For example, the infinite stream a^ω can originate from infinitely many pairs of streams u,v in the sense of both $u\cdot v = a^\omega$ and $u \| v = a^\omega$. Thus finite observations are crucial here.

Another advantage of finite observations is that we can define the operators, in particular $\|^{obs}$, without reference to any semantic approximation of its arguments - unlike the stream operator $\|^{str}$ where we put

$$X \|^{str} Y = \bigsqcup_{n=0}^{\infty}{}_S (X^{[n]} \|^{str} Y^{[n]})$$

in the general case.

We can now define the denotational observation semantics $\mathfrak{D}_{obs}$ for $\mathfrak{L}$. Again we use environments $\gamma \in \Gamma$, but now w.r.t. $\Gamma = Pvar \rightarrow \mathfrak{P}_{sat}(Obs(A))$.

6.5. DEFINITION. The semantic mapping

$$\mathfrak{D}_{obs}[\![\cdot]\!]: \mathfrak{L} \rightarrow (\Gamma \rightarrow \mathfrak{P}_{sat}(Obs(A)))$$

is given by

(i) $\mathfrak{D}_{obs}[\![a]\!](\gamma) = \{\epsilon, a, a\surd\}$

(ii) $\mathfrak{D}_{obs}[\![P;Q]\!](\gamma) = \mathfrak{D}_{obs}[\![P]\!](\gamma) ;^{obs} \mathfrak{D}_{obs}[\![Q]\!](\gamma)$

(iii) $\mathfrak{D}_{obs}[\![P \,\mathbf{or}\, Q]\!](\gamma) = \mathfrak{D}_{obs}[\![P]\!](\gamma) \mathbf{or}^{obs} \mathfrak{D}_{obs}[\![Q]\!](\gamma)$

(iv) $\mathfrak{D}_{obs}[\![P \| Q]\!](\gamma) = \mathfrak{D}_{obs}[\![P]\!](\gamma) \|^{obs} \mathfrak{D}_{obs}[\![Q]\!](\gamma)$

(v) $\mathfrak{D}_{obs}[\![x]\!](\gamma) = \gamma(x)$

(vi) $\mathfrak{D}_{obs}[\![\mu x[P]]\!](\gamma) = \mu\Phi_{P,\gamma}$ where $\Phi_{P,\gamma} = \lambda H.\, \mathfrak{D}_{obs}[\![P]\!](\gamma\{H/x\})$.

7. THE ISOMORPHISM BETWEEN STREAMS AND OBSERVATIONS: CONTINUED

Here we wish to link the stream semantics $\mathfrak{D}_{Str}$ with the observation semantics $\mathfrak{D}_{obs}$. Recall that Ψ is the cpo isomorphism from $\mathfrak{P}_{ncf}(Str(A))$ onto $\mathfrak{P}_{sat}(Obs(A))$.

7.1. THEOREM. *For every language operator* $op \in \{;\, , \mathbf{or}\, , \|\}$ *of* $\mathfrak{L}$ *and all* $X, Y \in \mathfrak{P}_{ncf}(Str(A))$

$$\Psi(X\ op^{str}\ Y) = \Psi(X) op^{obs} \Psi(Y)$$

holds.

7.2. COROLLARY. *For every concurrent process* $P \in \mathfrak{L}$ *and environment* $\gamma \in \mathfrak{P}var \rightarrow \mathfrak{P}_{ncf}(Str(A))$

$$\Psi(\mathfrak{D}_{Str}[\![P]\!](\gamma)) = \mathfrak{D}_{obs}[\![P]\!](\Psi \circ \gamma)$$

holds.

Together with theorem 5.3 the corollary says that the denotational *semantics* $\mathfrak{D}_{Str}$ and $\mathfrak{D}_{obs}$ are *isomorphic.*

8. Concluding remarks

We have not included any notion of *global* nondeterminism like + [Mi1] or □ [BHR] nor any notion of *deadlock* like *stop* [BHR] or δ [BK] in $\mathcal{L}$. This restriction allows us to work with a linear time approach in the form of streams or linear histories. It is a topic for further research to investigate whether our results can be extended to non-linear approaches like failure [BHR] or branching time semantics [BZ].

References

[Ba] R.J.R. Back, *A Continuous Semantics for Unbounded Nondeterminism,* Theoret. Comp. Sci. 23 (1983) 187-210.

[dB] J.W. de Bakker, *Mathematical Theory of Program Correctness* (Prentice Hall International, London, 1980).

[BBKM] J.W. de Bakker, J.A. Bergstra, J.W. Klop, J.-J. Ch. Meyer, *Linear Time and Branching Time Semantics for Recursion with Merge,* TCS 34 (1984) 135-156.

[BZ] J.W. de Bakker, J.I. Zucker, *Processes and the Denotational Semantics of Concurrency,* Inform. and Control 54 (1982) 70-120.

[BMOZ] J.W. de Bakker, J.-J. Ch. Meyer, E.-R. Olderog, J.I. Zucker, *Transition Systems, Infinitary Languages and the Semantics of Uniform Concurrency,* to appear in: Proc. 17th ACM STOC, Providence, R.I., 1985.

[BK] J.A. Bergstra, J.W. Klop, *Process Algebra for Synchronous Communication,* Information and Control, 60 (1984), pp. 109-137.

[BHR] S.D. Brookes, C.A.R. Hoare, A.W. Roscoe, *A Theory of Communicating Sequential Processes,* J. ACM 31 (1984) 560-599.

[Br] M. Broy, *Fixed Point Theory for Communication and Concurrency,* in: D. Bjørner (Ed.), Formal Description of Programming Concepts II (North-Holland, Amsterdam, 1983) 125-146.

[FHLR] N. Francez, C.A.R. Hoare, D.J. Lehmann, W.P. de Roever, *Semantics of Nondeterminism, Concurrency and Communicating,* JCSS 19 (1979) 290-308.

[FL] N. Francez, D.J. Lehmann, A. Pnueli, *A Linear History Semantics for Languages for Distributed Programming,* Theoret. Comp. Sci. 32 (1984) 25-46.

[He1] M.C.B. Hennessy, *Synchronous and Asynchronous Experiments on Processes,* Report CSR-125-82, Dept. of Comp. Sci., Univ. of Edinburgh, 1982.

[He2] M.C.B. Hennessy, *An Algebraic Theory of Fair Asynchronous Communicating Processes,* Manuscript, Dept. of Comp. Sci., Univ. of Edinburgh, 1984.

[Me] J.-J. Ch. Meyer, *Fixed Points and the Arbitrary and Fair Merge of a Fairly Simple Class of Processes,* Tech. Report IR-89/IR-92, Free University, Amsterdam, 1984.

[Mi1 R. Milner, *A Calculus of Communicating Systems,* LNCS 92 (Springer, Berlin, 1980).

[Mi2] R. Milner, *Calculi for Synchrony and Asynchrony,* Theoret. Comp. Sci. 25 (1983) 267-310.

[dNH] R. de Nicola, M.C.B. Hennessy, *Testing Equivalences for Processes,* TCS 34 (1984) 83-134.

[Ni] M. Nivat, *Infinite Words, Infinite Trees, Infinite Computations,* Foundations of Computer Science III. 2, Mathematical Centre Tracts 109 (1979) 3-52.

[OH1] E.-R. Olderog, C.A.R. Hoare, *Specification-oriented Semantics for Communicating Processes,* in: J. Diaz (Ed.), Proc. 10th ICALP, LNCS 154 (Springer, Berlin, 1983) 561-572.

[OH2] E.-R. Olderog, C.A.R. Hoare, *Specification-oriented Semantics for Communicating Processes,* Tech. Monograph PRG-37, Prog. Research Group, Oxford Univ., 1984 (to appear in Acta Informatica).

[Pl] G.D. PLOTKIN, *Dijkstra's Predicate Transformers and Smyth's Power Domains,* in: D. Bjørner (Ed.), Abstract Software Specification, LNCS 86 (Springer, Berlin, 1980) 527-553.

[Sm] M.B. SMYTH, *Power Domains,* JCSS 16 (1978) 23-26.

[Wi] G. WINSKEL, *Synchronisation Trees,* TCS 34 (1984) 33-82.