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

Department of Computer Science Report CS-R8512 September

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

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. Furthermore, a definition of the hiding operation which is inspired by the infinite streams approach is presented. Finally, the continuity of this operation is proved in the framework of finite observations.

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 submitted for publication elsewhere. A preliminary version (Report CS-R 8508) appeared in the Proceedings 12[th] International Colloquium on Automata, Languages and Programming (W. Brauer, ed.), Lecture Notes in Computer Science 194, pp. 149-157, Springer, 1985.

Report CS-R8512
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 1, Ni]. The supporting mathematics, however, tends to get complicated even if some simplifying assumptions on the admissible sets of streams are possible [Br 1, 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. This isomorphism will have various benefits in the mutual understanding of both approaches. More specifically, we establish our results for a core language $\mathcal{L}$ of uniform or schematic concurrency [BMOZ] involving uninterpreted atomic actions, sequential composition, nondeterministic choice (local nondeterminism), parallel composition (merge) with communication and recursion. For $\mathcal{L}$ 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 important 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 for sequential and parallel composition. 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 [OH 1/2] - 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* **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. Important examples of observations include (finite) traces, traces with divergence information, ready pairs and failure pairs leading to the (increasingly sophisticated) trace semantics, divergence semantics, readiness semantics and failure semantics for concurrent processes [OH2].

Characteristic and uniform for specification-oriented semantics is a simple nondeterminism ordering (reverse set-inclusion) on sets of observations [BHR], simple closure conditions on sets of observations, and a very simple way of constructing continuous semantic operators. Our specific observation semantics $\mathfrak{D}_{obs}$ for $\mathcal{L}$ follows these construction principles and 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 $\mathcal{L}$ are isomorphic. In fact, we can view $\mathfrak{D}_{obs}$ as a special representation of $\mathfrak{D}_{str}$. 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 [OH 1/2]. Thus through the idea of observation we can circumvent the technically difficult continuity proofs of [BBKM, Me].

These results seem to indicate that the notion of finite observation is more successful here than that of inifinite streams. This is not any more the case for more ambitious language constructs like hiding. We show that for an extended language $\mathcal{L}^*$ with hiding the idea of infinite streams very well motivates a standard definition of hiding due to [BHR]. Furthermore, a proof of the continuity of hiding in the framework of finite observations is provided, using techniques described in [OH2]. (The proof of the continuity of hiding in stream semantics requires certain techniques not included in the present paper and will appear elsewhere.)

Thus infinite streams and finite observations provide us with valuable, complementary information about one and the same computational structure.

Our results on the individual semantics $\mathfrak{D}_{str}$ and $\mathfrak{D}_{obs}$ are backed up by the reports [Me] and [OH2]. The linking isomorphism result is proved fully in sections 5 and 7.

2. The Language $\mathcal{L}$

Let A be a finite set of *actions*, with $a,b \in A$, $\star : A \times A \overset{part}{\rightarrow} A$ 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* $\mathcal{L}$, with $P,Q \in \mathcal{L}$, is given by the following *BNF*-syntax:

2.1 Definition ($\mathcal{L}$)

$$P ::= a \mid P;Q \mid P \textbf{ or } Q \mid P \parallel Q \mid x \mid \mu x[P]$$

2.2 Remark. Every action $a \in A$ denotes a process, the one which *finishes (successfully terminates)* 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 $a \star b$ is defined. For example, if only $b \star c$ is defined, we will obtain the following equation in our semantics:

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

Communication merge is inspired by [Mi2, BK, Wi], though we do not assume any algebraic properties of $\star$ while defining its semantics. The use of a partial communication function $\star$ seems new; it avoids to give any default value like δ [BK] or 0 [Wi] to pairs $a \star b$ which should not communicate.

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].

By varying the communication function $\star$, we can express more familiar notions of parallel composition:

2.3 Example.

1. Shuffle / arbitrary merge. Take $\star$ as the totally undefined function.

2. CCS: binary communication. Let $C \subseteq A$ be a designated set of *communications*, with $c \in C$, let $\bar{\ }: C \rightarrow C$ be a bijection on C providing *matching* communications c and $\bar{c}$ such that $\bar{\bar{c}} = c$ holds, and let τ be a special symbol in $A \setminus C$ denoting the so-called *invisible action.*

We define

$$c \star \bar{c} = \tau$$

for all $c \in C$, and take $\star$ to be undefined otherwise. Thus e.g.

$$(\tau;c)\|\bar{c} = \tau;c;\bar{c} \textbf{ or } \tau;\bar{c};c \textbf{ or } \bar{c};\tau;c \textbf{ or } \tau;\tau$$

will hold in our semantics. This is the parallel composition of CCS [Mi 1], except that we may have more non-communicating actions in $A \setminus C$ than just τ.

3. The Stream Semantics $\mathfrak{D}_{str}$

Let $\perp \notin A$. Then we define the set of *streams* $Str(A)$, with $u,v,w \in Str(A)$, as follows [Br 1]:

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

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 $A^* \cdot \{\perp\}$ of *unfinished streams*. The linear time semantics *LT* of [BBKM] - given for an $\mathfrak{L}$ with arbitrary merge (cf. Example 2.3) - 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 3.32).

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.

3.3 Examples. $a \leqslant a\perp$, $a\perp \nleqslant a$, $a\perp \nleqslant ab$.

For streams u,v we use the following approximation relation $\sqsubseteq$:

3.4 Definition. $u \sqsubseteq v$ iff the following holds:

- if u is finished or infinite then $u = v$,
- if u is unfisnished, i.e. of the form $u = u'\perp$, then $u' \leqslant v$.

3.5 Examples. $a \not\sqsubseteq a\perp$, $a\perp \sqsubseteq a$, $a\perp \sqsubseteq ab$.

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

3.6 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.

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

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

3.8 Definition. $X \sqsubseteq_S Y$ iff $\forall v \in Y\ \exists u \in X : u \sqsubseteq v$.

3.9 Aside. This is "one half" of the *Egli-Milner relation* [Pl 1]:

$$X \sqsubseteq_{EM} Y \text{ iff } X \sqsubseteq_S Y \text{ and } \forall u \in X\ \exists v \in Y : u \sqsubseteq v.$$

3.10 Diagram.

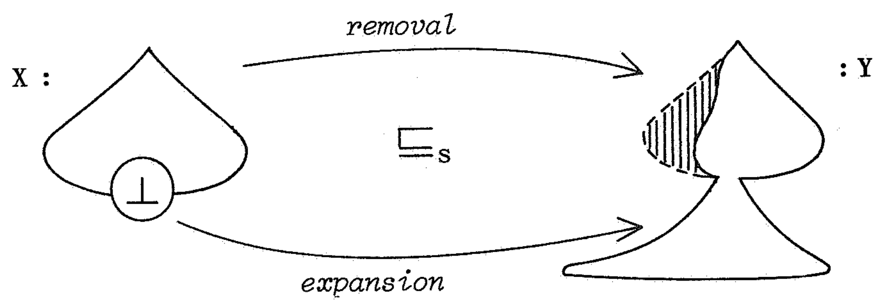

Figure. 1

Arbitrary streams $u \in X$ can be *removed* (this is not possible in the Egli-Milner relation); unfinished streams $u\bot \in X$ can be *expanded.*

3.11 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 1]. E.g. for $X = \{a\bot, ab\bot, abc\}$ and $Y = \{a\bot, abc\}$ both $X \sqsubseteq_S Y$ and $Y \sqsubseteq_S X$ hold. But the Smyth relation is a pre-order which generates an equivalence relation $\equiv_S$ on $\mathcal{P}(Str(A))$:

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

What are the sets identified by $\equiv_S$?

3.12 DEFINITION. $\min_S(X) = \{v \in X \mid \nexists u \in X : u \sqsubseteq v \wedge u \neq v\}$ 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.

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

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

3.15 PROPOSITION. *$(\mathcal{P}_f(Str(A)), \sqsubseteq_s, \{\bot\})$ is a cpo.*

PROOF. The proof can be read off from [Ba] (see [Me]). □

In fact, Back [Ba] considers the more complicated case of the Egli-Milner relation which forces him to require additional closure properties for sets $X \subseteq Str(A)$ when proving the *cpo* property. We also have to introduce additional closure properties, but at a later stage when it comes to proving the continuity of various semantic operators.

To define these operators, we first 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 $\bot \cdot v = \bot$. More specifically, for $u\bot \in A^* \cdot \{\bot\}$ we set $u\bot \cdot v = u\bot$, and for $u \in A^\infty$, $v\bot \in A^* \cdot \{\bot\}$ we set $u \cdot (v\bot) = (uv) \cdot \bot$.

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

$$u \| v = u \mathbin{\|\!\_} v \cup v \mathbin{\|\!\_} u \cup u | v$$

where recursively $\bot \mathbin{\|\!\_} v = \{\bot\}$, $\epsilon \mathbin{\|\!\_} v = \{v\}$, $a \cdot u \mathbin{\|\!\_} v = a \cdot (u \| v)$ and $a | b = \{a * b\}, a | (bv) = (a * b) \cdot v$, $au | b = (a * b) \cdot u, au | bv = (a * b) \cdot (u \| v)$ provided $a * b$ is defined; in in all other cases $u | v = \emptyset$. This finite recursive definition of $\|$ using $\mathbin{\|\!\_}$ and $|$ is due to [BK].

To lift these definitions to flat sets of streams, we enforce flatness of the results by applying the operator $\min_S$ of Definition 3.12, and use the following notion of *n-th approximation* $u^{[n]}$, $n \geqslant 0$, for streams u : $u^{[n]} = u$ if $|u| < n$ and $u^{[n]} = u'\bot$ if $|u| \geqslant n$ and $u' \leqslant u$ with $|u'| = n$. Thus $u^{[0]} = \bot$ holds for all streams u. We extend this definition pointwise to subsets $X \subseteq Str(A)$ by putting

$X^{[n]} = \{u^{[n]} \mid u \in X\}$. Note that $X^{[n]} \sqsubseteq_S X^{[n+1]}$ holds.

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

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

Parallel composition

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

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

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

$$X \|^{Str} Y = \bigsqcup_{n=0}^{\infty}{}_s (X^{[n]} \|^{Str} Y^{[n]}).$$

3.16 THEOREM. *The semantic operators*

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

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

PROOF. The proof is given in [Me]. Showing monotonicity is not trivial for ; and $\|$ due to the complex definition of $\sqsubseteq_s$. □

To provide meaning to recursive processes too, we will have to show that the semantic operators op^{Str} are also continuous.

3.17 THEOREM. $\textbf{or}^{Str}$ *is continuous under* $\sqsubseteq_s$.

Unfortunately, the operators $;^{Str}$ and $\|^{Str}$ are not continuous on arbitrary flat sets of streams.

3.18 Counterexamples. Take $X = \{a^\omega\}$ where a^ω is the infinite stream of a's and

$$X_n = \{u \in \{a\}^* \mid \text{ where } |u| \geq n\},\ n \geq 0$$

Clearly $<X_n>_{n \geq 0}$ is a $\supseteq$-chain, and hence a $\sqsubseteq_s$-chain by Remark 3.11. Note that $\bigsqcup_{n=0}^{\infty}{}_s X_n = \varnothing$, whereas both $\bigsqcup_{n=0}^{\infty}{}_s (X_n ;^{Str} X) \neq \varnothing$ and $\bigsqcup_{n=0}^{\infty}{}_s (X_n \|^{Str} X) \neq \varnothing$. Thus

$$\bigsqcup_{n=0}^{\infty}{}_s (X_n\, op^{Str} X) \neq (\bigsqcup_{n=0}^{\infty}{}_s X_n)\, op^{Str} X$$

for both $op \in \{;, \|\}$.

To rescue the continuity of ; and $\|$, we will restrict ourselves to closed sets of streams.

3.19 DEFINITION [Ba]. $X \subseteq Str(A)$ is *closed* if for every infinitely often increasing chain $<u_n>_{n \geq 0}$ 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* with respect to the following metric topology on $Str(A)$.

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

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

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

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

3.22 PROPOSITION. *$(Str(A),d)$ is a 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.

3.23 PROPOSITION. *$X = \{a^n b a^\omega \mid n \geqslant 0\} \cup \{a^\omega\}$ is (topologically) closed, but $Y = \{a^n b a^\omega \mid n \geqslant 0\}$ 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].

3.25 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:

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

PROOF. The proof is rather involved and given in full detail in [Me]. □

Using the lemma, we can now establish the following results:

3.27 PROPOSITION. *$(\mathcal{P}_{ncf}(Str(A)), \sqsubseteq_s, \{\bot\})$ is a cpo.*

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

PROOF. This uses Lemma 3.26 and otherwise follows the proof of Theorems 2.9 and 2.10 in [BBKM]; in particular the case of $\|$ is involved. □

3.29 REMARK. Lemma 3.26 and Theorem 3.28 do not hold, in general, for *infinite* sets A of actions.

We can now define the denotational stream semantics $\mathcal{D}_{Str}$ for $\mathcal{L}$. We adopt the usual technique with environments to deal with (free) process variables. This 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 $\gamma\{X/x\}$ denote the environment which is like γ, except for its value in x which is now X. Let $[\mathcal{P}_{ncf}(Str(A)) \rightarrow_S \mathcal{P}_{ncf}(Str(A))]$ denote the collection of all $\sqsubseteq_S$-continuous functions from

$\mathcal{P}_{ncf}(Str(A))$ to $\mathcal{P}_{ncf}(Str(A))$, and let, for $\Phi \in [\mathcal{P}_{ncf}(Str(A)) \rightarrow \mathcal{P}_{ncf}(Str(A))]$, $\mu\Phi$ denote its least fixed point.

3.30 DEFINITION. The semantic mapping

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

is given by:

$$\mathcal{D}_{Str}[\![a]\!](\gamma) = \{a\} \tag{i}$$

$$\mathcal{D}_{Str}[\![P;Q]\!](\gamma) = \mathcal{D}_{Str}[\![P]\!](\gamma);^{Str}\mathcal{D}_{Str}[\![Q]\!](\gamma) \tag{ii}$$

$$\mathcal{D}_{Str}[\![P \textbf{ or } Q]\!](\gamma) = \mathcal{D}_{Str}[\![P]\!](\gamma) \textbf{ or }^{Str}\mathcal{D}_{Str}[\![Q]\!](\gamma) \tag{iii}$$

$$\mathcal{D}_{Str}[\![P \| Q]\!](\gamma) = \mathcal{D}_{Str}[\![P]\!](\gamma)\|^{Str}\mathcal{D}_{Str}[\![Q]\!](\gamma) \tag{iv}$$

$$\mathcal{D}_{Str}[\![x]\!](\gamma) = \gamma(x) \tag{v}$$

$$\mathcal{D}_{Str}[\![\mu x[P]]\!](\gamma) = \mu\Phi_{P,\gamma} \tag{vi}$$

where $\Phi_{p,\gamma} = \lambda X \cdot \mathcal{D}_{Str}[\![P]\!](\gamma\{X/x\})$. Let us evaluate the stream semantics $\mathcal{D}_{Str}$ of recursive processes more precisely. 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 $Q ; (\cdots x \cdots)$. A process P is called *guarded* (cf [Mi 1] or [Ni], where *Greibach* replaces guarded) whenever, for each recursive subprocess $\mu y[Q]$ of P we have that Q is guarded in y.

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

3.32 PROPOSITION. *Consider a concurrent process P without free process variables. If P is not guarded,* $\mathcal{D}_{Str}[\![P]\!](\gamma) = \{\bot\}$ *holds.*

Thus all unguarded processes are identified in our semantics. This simple solution seems more attractive than the results computed by the linear time semantics LT in [BBKM]. In LT, for example, one obtains $LT[\![\mu x[x]]\!](\gamma) = A^{\infty}$, but surprisingly $LT[\![\mu x[x;b]]\!](\gamma) = A^{\omega}$ - without an intuitively clear explanation of these differences. (We remark, however, that for guarded processes P the equation $\mathcal{D}_{Str}[\![P]\!](\gamma) = LT[\![P]\!](\gamma)$ holds [Me, Ni].)

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

4.1 Background. Motivated by some of the construction principles in the failure semantics [BHR], a new approach to the semantics of concurrent processes has been developed in [OH 1/2]. It is called "specification-oriented" because it starts from the following simple concept of process correctness: a process P satisfies a specification S, abbreviated **Psat$S$**, 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. In [OH2] the feasibility of this idea has been demonstrated by treating the (increasingly sophisticated) examples of counter, trace, divergence, readiness and failure semantics, which support the specification of both safety and (certain) liveness properties.

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 most language operators (except hiding: cf. Section 8).

We shall not explain these general principles any further, but rather 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 $\mu x[x]$ (or hiding an infinite stream of actions: see Remark 8.6 later). 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 $Str(A)$. Instead we jump now directly to sets of observations. 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.

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

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

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

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

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

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 (see the Aside 4.7). This idea is familiar from Dijkstra's weakest precondition semantics where a diverging program will not achieve any postcondition [Pl 2].

Properties (i), (ii) are typical conditions on traces to be found in [BHR, FLP, OH 1/2]. Property (iii) is a new "linear version" of the extensibility condition in the readiness [OH2] or failure semantics [BHR] where the local branching structure of a process is recorded by requiring *more than one* extension of a trace to be present in H. 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 $Obs(A)$.

4.7 ASIDE. Following [OH2], a *specification* S is an arbitrary subset $S \subseteq Obs(A)$ whereas the saturated subsets $H \subseteq Obs(A)$ are called process specifications; they serve as semantics for processes via $\mathcal{D}_{Obs}$. The relation $P \,\mathbf{sat}\, S$ is then defined by $H \subseteq S$ with $H = \mathcal{D}_{Obs}[\![P]\!]$. A *reasonable* specification S will only talk about traces and terminated traces: $S \subseteq A^* \cup A^* \cdot \{\surd\}$. Then P **sat** S holds only if P never diverges.

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

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

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

PROOF. Clearly, $\supseteq$ is partial order on $\mathcal{P}_{sat}(Obs(A))$ with $Obs(A)$, which is saturated, as its least element. The only property we have to check is that for every chain

$$H_0 \supseteq H_1 \supseteq ... \supseteq H_n \supseteq ...$$

of saturated sets also $\bigcap_{n=0}^{\infty} H_n$ is saturated. This is clear for property (i) and the universal properties (ii) and (iv) of Definition 4.4. Only showing the existential property (iii) could be a problem. Fortunately it is not because A is finite. □

We see that proving the *cpo* property for $\mathcal{P}_{sat}(Obs(A))$ is much simpler than for $\mathcal{P}_{ncf}(Str(A))$:*cf.* Lemma 3.26. 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

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

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 stream 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

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

by defining

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

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

5.3 THEOREM. *Ψ is an isomorphism from the cpo $(P_{ncf}(Str(A)), \sqsubseteq_s, \{\perp\})$ onto the cpo*

$(P_{sat}(Obs(A)), \supseteq, Obs(A))$, i.e. Ψ is bijective, yields $\Psi(\{\bot\}) = Obs(A)$ and strongly preserves the partial orders:

$$X \sqsubseteq_s Y \text{ iff } \Psi(X) \supseteq \Psi(Y)$$

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

Proof. (1) Ψ *is well-defined*:
Let $X \in \mathcal{P}_{ncf}(Str(A))$. We have to check that $\Psi(X) \in \mathcal{P}_{sat}(Obs(A))$ holds.

(i) $\epsilon \in \Psi(X)$: since X is non-empty and (ii) holds.
(ii) $\Psi(X)$ is prefix closed: by definition of Ψ.
(iii) $\Psi(X)$ is extensible: by the definition of Ψ, all histories $h \in \Psi(X)$ not ending in $\surd$ have some extension $h\alpha$ with $\alpha \in A \cup \{\surd, \uparrow\}$ in $\Psi(X)$.
(iv) $\Psi(X)$ treats divergence as chaos: a history $h\uparrow \in \Psi(X)$ can originate only from a stream $h\bot \in X$ which, by the definition of Ψ, leads to $h \cdot Obs(A) \subseteq X$ as well.

(2) $\Psi(\{\bot\}) = Obs(A)$:
by definition of Ψ, *cf.* the argument for (1), (iv) above.

(3) Ψ *is surjective*:
Take some $H \in \mathcal{P}_{sat}(Obs(A))$. We have to present some $Y \in \mathcal{P}_{ncf}(Str(A))$ with $\Psi(Y) = H$. We define Y via the following auxiliary set $X \subseteq Obs(A)$:

$$\begin{aligned} X = &\{u \in A^* \mid u\surd \in H\} \\ &\cup \{v \in A^\omega \mid \forall h \in A^* : h \leqslant v \text{ implies } h \in H\} \\ &\cup \{u \in A^* \cdot \{\bot\} \mid u\uparrow \in H\}. \end{aligned}$$

To enforce flatness, we apply to X the operator $\min_s$ of Definition 3.12 yielding:

$$Y = \min_s(X).$$

We first show $Y \in \mathcal{P}_{ncf}(Str(A))$.

(i) Y is flat: by the definition of $\min_s$.
(ii) Y is non-empty: It suffices to show that $X \neq \emptyset$.
To this end, we start from $\epsilon \in H$ and apply the extensibility condition of (the saturated) H to ϵ and its successive extensions as long as possible.
Case 1. This is possible only finitely often. Then we end up in some history $u\surd \in H$ and have $u \in X$.
Case 2. This is possible infinitely often. If we eventually hit some history $u\uparrow \in H$ then $u\bot \in X$. Otherwise all finite prefixes $h \leqslant v$ of some infinite stream $v \in A^\omega$ are in H. Then $v \in X$.
Thus indeed $X \neq \emptyset$.
(iii) Y is topologically closed: Take some $v \in A^\omega$ and suppose there is a Cauchy sequence $<u_n>_{n \geqslant 0}$ with $u_n \in Y$ and (*) $\forall m \geqslant 0 \exists n \geqslant 0 : d(u_n, v) \leqslant 2^{-m}$. We have to show $v \in Y$. By (*), u_n and v agree on their first m symbols:

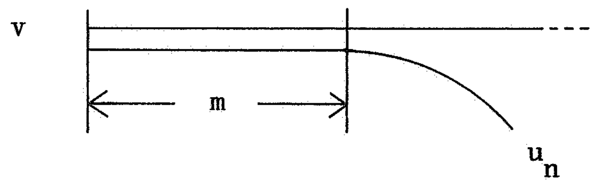

FIGURE 2

Since the original H is prefix closed, we obtain

$$\forall h \in A^* : h \leqslant v \text{ implies } h \in H.$$

Thus $v \in X$. But also $v \in Y$. Otherwise there is some $u \in A^*$ with $u < v$ and $u\bot \in Y$. By (*), there is some $n \geqslant O$ with $u < u_n$. Contradiction to $u_n \in Y$, which is flat by (i).
Next we show $\Psi(Y) = H$. It suffices to show $\Psi(X) = H$ since $\Psi(X) = \Psi(Y)$ holds by the construction of X and Y.

(i) $\Psi(X) \subseteq H$: by the definition of Ψ and X, and the prefix and chaotic closure of H.
(ii) $H \subseteq \Psi(X)$: Let $h \in H$. By applying the extensibility condition of H to h and its extensions as long as possible, we realize - similarly to the non-emptiness proof of Y above - that there exist some $u \in A^*$ or $v \in A^\omega$ with

$u \in X$ and $h \leqslant u\surd$, *or*

$v \in X$ and $h \leqslant v$, *or*

$u\bot \in X$ and $h \leqslant u\uparrow$.

By the definition of Ψ, we get $h \in \Psi(X)$.

(4) $X \sqsubseteq_s Y$ *implies* $\Psi(X) \supseteq \Psi(Y)$ *for all* $X, Y \in \mathcal{P}_{ncf}(Str(A))$:

By definition, $X \sqsubseteq_s Y$ iff $\forall v \in Y\ \exists u \in X : u \sqsubseteq v$. Take some $h \in \Psi(Y)$.

Case 1. $\exists u < h : u\bot \in X$. Then $h \in u \cdot Obs(A) \subseteq \Psi(X)$.

Case 2. $\forall u < h : u\bot \notin X$.
Then also $\forall v < h : v\bot \notin Y$ because, by the definition of $X \sqsubseteq_s Y$, $v\bot \in Y$ with $v < h$ could be generated only by some unfinished stream $u\bot$ with $u \leqslant v < h$ in X, which contradicts Case 2.
Consequently $\forall h' < h : h'\uparrow \notin \Psi(Y)$, and in particular $h \notin A^* \cdot \{\uparrow\}$.

Subcase 2.1. $h \in A^* \cdot \{\surd\}$
Then h is of the form $h = w\surd$. By the assumption of Case 2, we obtain $w \in Y$ and $w \in X$. But then also $h = w\surd \in \Psi(X)$.

Subcase 2.2. $h \notin A^* \cdot \{\surd\}$
Then $\exists w \in Y : h \leqslant w$. By the definition of $X \sqsubseteq_s Y$, $\exists v \in X : v \sqsubseteq w$. Due to Case 2 also $h \leqslant v$. Thus $h \in \Psi(X)$.

In all cases we have $h \in \Psi(X)$.

(5) $\Psi(X) \supseteq \Psi(Y)$ *implies* $X \sqsubseteq_s Y$ *for all* $X, Y \in \mathcal{P}_{ncf}(Str(A))$:

Take $v \in Y$. By the definition of $X \sqsubseteq_s Y$, we have to present some $u \in X$ with $u \sqsubseteq v$.
Case 1. $\exists u < v : u\bot \in X$.
Then $u\bot \sqsubseteq v$.

Case 2. $\forall u < v : u\bot \notin X$.
Then $v \notin A^* \cdot \{\bot\}$ and $\forall h < v : h\uparrow \notin \Psi(X)$.

Subcase 2.1. $v \in A^*$
Then $v\surd \in \Psi(Y)$ and thus $v\surd \in \Psi(X)$.
By Case 2, we get $v \in X$. Of course, $v \sqsubseteq v$.

Subcase 2.2. $v \in A^\omega$
Then $H = \{h \in A^* \mid h \leqslant v\} \subseteq \Psi(Y) \subseteq \Psi(X)$.
By Case 2, we find for each $h \in H$ of length $m \geqslant O$ some $u_m \in X$ with $h \leqslant u_m$.
In other words:

$$\forall m \geqslant O \exists u_m \in X : d(u_m, v) \leqslant 2^{-m}.$$

Thus $<u_m>_m \geqslant 0$ is a Cauchy sequence in X converging against v. By the topological closure of X, we get $v \in X$ and of course $v \sqsubseteq v$.

In all cases we found some $u \in X$ with $u \sqsubseteq v$.

(6) Ψ *is injective*:
Suppose $\Psi(X) = \Psi(Y)$ holds for $X, Y \in \mathcal{P}_{ncf}(Str(A))$.
By (5), we obtain $X \sqsubseteq_s Y$ and $Y \sqsubseteq_s X$, i.e. $X \equiv_s Y$. Since X and Y are flat, $X = Y$ follows (cf. Proposition 3.14). □

5.4 Corollary. The isomorphism ψ is (of course) continuous, i.e. for every $\sqsubseteq_s$-chain $<X_n>_{n \geqslant 0}$ we have

$$\psi(\bigsqcup_{n=0_s}^{\infty} X_n) = \bigcap_{n=0}^{\infty} \psi(X_n).$$

Proof. Consider a $\sqsubseteq_s$-chain $<X_n>_{n \geqslant 0}$. Since ψ is bijective, its inverse ψ^{-1} exists, and since ψ strongly preserves the partial orders, both ψ and ψ^{-1} are monotonic. Thus $<\psi(X_n)>_{n \geqslant 0}$ is a $\supseteq$-chain, and both

$$\bigcap_{n=0}^{\infty} \psi(X_n) \supseteq \psi(\bigsqcup_{n=0_s}^{\infty} X_n) \tag{1}$$

and

$$\bigsqcup_{n=0}^{\infty} \psi^{-1}(\psi(X_n)) \sqsubseteq_s \psi^{-1}(\bigcap_{n=0} \psi(X_n)). \tag{2}$$

Applying ψ^{-1} to (1) yields

$$\psi^{-1}(\bigcap_{n=0}^{\infty} \psi(X_n)) \sqsubseteq_s \psi^{-1}(\psi(\bigsqcup_{n=0_s}^{\infty} X_n)). \tag{3}$$

Connecting and simplifying (2) and (4) yields

$$\bigsqcup_{n=0_s}^{\infty} X_n \sqsubseteq_s \psi^{-1}(\bigcap_{n=0}^{\infty} \psi(X_n)) \sqsubseteq_s \bigsqcup_{n=0_s}^{\infty} X_n.$$

Hence

$$\psi^{-1}(\bigcap_{n=0}^{\infty} \psi(X_n)) = \bigsqcup_{n=0_s}^{\infty} X_n. \tag{4}$$

Applying ψ to (4) yields after simplification

$$\bigcap_{n=0}^{\infty} (X_n) = \psi(\bigsqcup_{n=0}^{\infty} X_n),$$

the desired continuity property of ψ. □

5.5 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 *special representation* of the general construction $\mathcal{P}_{ncf}(Str(A))$. This provides us with a new mutual understanding of the closure properties in both domains: *topological closedness* on streams corresponds to taking *all finite prefixes* as observations, *flatness* of sets 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 $h \notin A^* \cdot \{\surd\}$ 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 $\mathcal{P}_{sat}(Obs(A))$.

Some parts of the isomorphism Ψ look familiar from the well-known (trivial) isomorphism for the case of discrete *cpo's* of the form $A_\perp = A \cup \{\perp\}$ (with $x \sqsubseteq y$ iff $x = \perp$ *or* $x = y$ for $x,y \in A_\perp$),

used to justify Dijkstra's weakest precondition semantics for nondeterministic state transformers [Pl2]. There the Smyth ordering $\sqsubseteq_s$ on sets $X, Y \subseteq A_\perp$, defined by

$$X \sqsubseteq_s Y \text{ iff } \forall y \in Y \exists x \in X : x \sqsubseteq y$$

as in Definition 3.8, can easily be shown isomorphic to the superset ordering $\supseteq$ if $\perp$ is replaced by its chaotic closure $A_\perp$ [Pl2]. But this technique is of course too simple for the set $Str(A)$ of finite and infinite streams.

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

Let us now continue with the development of the observation semantics $\mathcal{D}_{Obs}$. To define the semantic operators for $\mathcal{D}_{obs}$ we could well provide indirect definitions by using the previous isomorphism. But it will be illuminating to discuss direct definitions first. This is so because the ordering $\supseteq$ on sets of observations allows a very simple, uniform proof of (monotonicity and) continuity for the operators ;, $\cup$ and $\|$ in $\mathcal{L}$.

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.

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

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.*

6.2 REMARK. If R is not domain finite, op_R is not continuous in general (see Section 8).
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:

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

Well-definedness of $;^{Obs}$ does not follow from a simple, general principle; this has to be checked separately:

6.3 PROPOSITION. *The operator $;^{Obs}$ is well-defined, i.e. preserves the saturation property of its arguments.*

But monotonicty and continuity of $;^{Obs}$ follow from Lemma 6.1. To see this we take $\mathbf{X} = Obs(A) \times Obs(A)$ and $\mathbf{Y} = Obs(A)$. Next we look for a domain finite relation $R \subseteq \mathbf{X} \times \mathbf{Y}$ such that

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

R can be read off from $;^{Obs}$ as follows: $(h_1,h_2)\ R\ h$ 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 - together with the fact that for sets $X_i, Y_i \in \mathcal{P}_{sat}(Obs(A))$

$$X_1 \times Y_1 \supseteq X_2 \times Y_2 \text{ iff } X_1 \supseteq X_2 \text{ and } Y_1 \supseteq Y_2$$

holds - imply:

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

6.5 CAUTION. Note that there are *many* relations $R \subseteq \mathbf{X} \times \mathbf{Y}$ satisfying (*) above. But not every such relation is domain finite. For example, by omitting the condition $h_2 = \epsilon$ in clause (i) of R, we lose the domain finiteness of R so that Lemma 6.1 does not apply.

The discussion of the remaining semantic operators will be more brief. *Local nondeterminism* is modelled by set-theoretic union

$$H_1 \ \mathbf{or}^{Obs}\ H_2 = H_1 \cup H_2$$

which is well-defined and (by a straightforward application of Lemma 6.1) monotonic and continuous under $\supseteq$. *Parallel composition* is defined by

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

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

$$h_1 \| h_2 = h_1 ⫦ h_2 \cup h_2 ⫦ h_1 \cup h_1 \mid h_2$$

with $\epsilon ⫦ \epsilon = \{\epsilon\}$, $ah_1 ⫦ h_2 = a \cdot (h_1 \| h_2)$, $\surd ⫦ h = \{h\}$, $\uparrow ⫦ \epsilon = Obs(A)$ and with $ah_1 \mid bh_2 = (a*b) \cdot (h_1 \| h_2)$ provided $a*b$ is defined; in all other cases $h_1 ⫦ h_2 = \varnothing$ and $h_1 | h_2 = \varnothing$.

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

Proof. Again, well-definedness has to be checked separately. But monotonicity and continuity follow from lemma 6.1 by taking as domain finite relation that R with $(h_1,h_2)\ R\ h$ iff $h \in h_1 \| h_2$. □

6.7 **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

$$X \|^{Str} Y = \bigsqcup_{n=o}^{\infty}{}_{s} (X^{[n]} \|^{Str} Y^{[n]})$$

in the general case.

We can now define the denotational observation semantics $\mathcal{D}_{Obs}$ for $\mathcal{L}$. Again we use environments $\gamma \in \Gamma$, but now with respect to $\Gamma = Pvar \rightarrow \mathcal{P}_{sat}(Obs(A))$.

6.8 **Definition.** The semantic mapping

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

is given by

$$\mathcal{D}_{Obs}[\![a]\!](\gamma) = \{\epsilon, a, a\surd\} \qquad \text{(i)}$$

$$\mathcal{D}_{Obs}[\![P;Q]\!](\gamma) = \mathcal{D}_{Obs}[\![P]\!](\gamma) ;^{Obs} \mathcal{D}_{Obs}[\![Q]\!](\gamma) \qquad \text{(ii)}$$

$$\mathcal{D}_{Obs}[\![P \text{ or } Q]\!](\gamma) = \mathcal{D}_{Obs}[\![P]\!](\gamma) \text{ or }^{Obs} \mathcal{D}_{obs}[\![Q]\!](\gamma) \qquad \text{(iii)}$$

$$\mathcal{D}_{Obs}[\![P \| Q]\!](\gamma) = \mathcal{D}_{Obs}[\![P]\!](\gamma) \|^{Obs} \mathcal{D}_{Obs}[\![Q]\!](\gamma) \qquad \text{(iv)}$$

$$\mathcal{D}_{Obs}[\![x]\!](\gamma) = \gamma(x) \qquad \text{(v)}$$

$$\mathcal{D}_{Obs}[\![\mu x[P]]\!](\gamma) = \mu \Phi_{P,\gamma} \qquad \text{(vi)}$$

where $\Phi_{P,\gamma} = \lambda H \cdot \mathcal{D}_{Obs}[\![P]\!](\gamma\{H/x\})$.

7. The Isomorphism between Streams and Observations: continued

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

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

$$\Psi(X\ op^{Str}\ Y) = \Psi(X) op^{Obs} \Psi(Y)$$

holds.

Proof. Every $X \in \mathcal{P}_{ncf}(Str(A))$ can be approximated by

$$X = \bigsqcup_{n=0}^{\infty}{}_{S}\ X^{[n]}$$

where the $X^{[n]}$, $n \geqslant 0$, are defined as in Section 3. By Theorems 3.17, 3.28, 5.3 and Propositions 6.4, 6.6, the operators op^{Str}, Ψ, op^{Obs} with $op \in \{;, \mathbf{or}, \|\}$ are all continuous. Thus it suffices to prove

$$\Psi(X op^{Str} Y) = \Psi(X) op^{Obs} \Psi(Y) \qquad (1)$$

only for sets $X, Y \in \mathcal{P}_{ncf}(Str(A))$ without infinite streams $u \in A^{\omega}$. Noticing that

$$\Psi(min_s(Z)) = \Psi(Z) \tag{2}$$

holds for all $Z \subseteq Str(A)$, we can simplify the proof of (1) further.

Case 1. op = or

$$\Psi(X \mathbf{or}^{Str} Y) = \Psi(min_s(X \cup Y)) = \Psi(X \cup Y) =$$
$$\Psi(X) \cup \Psi(Y) = \Psi(X) \mathbf{or}^{Obs} \Psi(Y).$$

Case 2. op = ;
By the definition of $;^{Str}$ and fact (2) it suffices to show that

$$\Psi(u \cdot v) = \Psi(u);^{Obs}\Psi(v) \tag{3}$$

holds "pointwise" for all $u, v \in A^* \cup A^* \cdot \{\perp\}$.

Subcase 2.1. $u \in A^*$
We show the inclusion "$\subseteq$" of (3). Let $h \in \Psi(u \cdot v)$. If $h \leqslant u$ then $h \in \Psi(u)$ and h does not contain $\surd$. hence $h \in \Psi(u);^{Obs}\Psi(v)$. If $u < h$, e.g. $h = u \cdot h'$, then $u \surd \in \Psi(u)$ and $h' \in \Psi(v)$. Hence also $h \in \Psi(u);^{Obs}\Psi(v)$.
The proof of inclusion "$\supseteq$" requires a similar case analysis.

Subcase 2.2. $u = u' \perp \in A^* \cdot \{\perp\}$
$\Psi(u \cdot v) = \Psi(u) = \{h \mid h \leqslant u'\} \cup \{u'h \mid h \in Obs(A)\}$
$= \Psi(u);^{Obs}\Psi(v)$ [since $u' \uparrow \in \Psi(u)$].

Case 3. op = ||
Again, it suffices to show the "pointwise" equation

$$\Psi(u \| v) = \Psi(u) \|^{Obs} \Psi(v). \tag{4}$$

But this case is more involved because of the recursive definitions of $\|$ and its auxiliary operators $\Vert\!\!\_$ and $|$. For sets $H_1, H_2 \subseteq Obs(A)$ and $op \in \{\Vert\!\!\_, \mid\}$ let

$$H_1 op^{Obs} H_2 = \{h \mid \exists h_1 \in H_1, h_2 \in H_2 : h \in h_1 op h_2\}$$

analogously to the definition of $H_1 \|^{Obs} H_2$ in Section 6. Furtheron, for $op \in \{\|, \Vert\!\!\_, \mid\}$ and $n \geqslant 0$ let

$$op_n$$

abbreviate the following assertion:

$$\forall u, v \in A^* \cup A^* \cdot \{\perp\} : \mid u \mid + \mid v \mid = n \Rightarrow$$
$$\Psi(u \; op \; v) = \Psi(u) op^{Obs} \Psi(v).$$

For example, $\|_n$ asserts that equation (4) holds for all streams u, v of length $\mid u \mid + \mid v \mid = n$.
To prove (4), we will show by induction on n that

$$\|_n \wedge \Vert\!\!\_{\,n} \wedge \mid_n \tag{5}$$

holds for all $n \geqslant 0$. To this end, we show

$$\Vert\!\!\_{\,0} \wedge \mid_0 \tag{6}$$

$$\forall_n \geqslant 0 : \Vert\!\!\_{\,n} \wedge \mid_n \Rightarrow \|_n \tag{7}$$

$$\forall_n \geqslant 0 : \|_n \Rightarrow \Vert\!\!\_{\,n+1} \wedge \mid_{n+1}. \tag{8}$$

Ad (6). Clearly, $|u| + |v| = 0$ implies $u=v=\epsilon$.
By the definitions, we have

$$\Psi(\epsilon \lfloor\!\lfloor \epsilon) = \Psi(\{\epsilon\}) = \{\epsilon, \surd\} =$$
$$\{\epsilon, \surd\} \lfloor\!\lfloor^{Obs} \{\epsilon, \surd\} = \Psi(\epsilon) \lfloor\!\lfloor^{Obs} \Psi(\epsilon)$$

and

$$\Psi(\epsilon \mid \epsilon) = \varnothing = \Psi(\epsilon) \mid^{Obs} \Psi(\epsilon).$$

Ad (7). Assume $\lfloor\!\lfloor_n$ and $|_n$. To show $\|_n$ we take u,v with $|u| + |v| = n$:

$$\begin{aligned}\Psi(u\|v) &= \Psi(u \lfloor\!\lfloor v \cup v \lfloor\!\lfloor u \cup u \mid v)[\text{def}\cdot u\|v] \\ &= \Psi(u \lfloor\!\lfloor v) \cup \Psi(v \lfloor\!\lfloor u) \cup \Psi(u \mid v)\ [\text{def}.\Psi] \\ &= (\Psi(u) \lfloor\!\lfloor^{Obs} \Psi(v)) \cup (\Psi(v) \lfloor\!\lfloor^{Obs} \Psi(u)) \\ &\quad \cup (\Psi(u) \mid^{Obs} \Psi(v))[\text{ assumption }] \\ &= \Psi(u) \|^{Obs} \Psi(v)[\text{ def. } \|^{Obs}\ ].\end{aligned}$$

Ad (8). Assume $\|_n$. We show $\lfloor\!\lfloor_{n+1}$. Take u,v with $|u| + |v| = n$.

Subcase 3.1. $u = \epsilon$

$$\begin{aligned}\Psi(\epsilon \lfloor\!\lfloor v) &= \Psi(\{v\}) = \{\surd\} \lfloor\!\lfloor^{Obs} \Psi(\{v\}) \\ &= \Psi(\epsilon) \lfloor\!\lfloor^{Obs} \Psi(v).\end{aligned}$$

Subcase 3.2. $u = a \cdot u'$

$$\begin{aligned}\Psi(au' \lfloor\!\lfloor v) &= \Psi(a \cdot (u'\|v)) \\ &= \Psi(a);^{Obs} \Psi(u'\|v)[\text{ def.}\Psi \text{ and } ;^{Obs}\ ] \\ &= \Psi(a);^{Obs} (\Psi(u')\|^{Obs} \Psi(v))[\text{assumption}] \\ &= \Psi(au') \lfloor\!\lfloor^{Obs} \Psi(v)[\text{def.} \lfloor\!\lfloor^{Obs}].\end{aligned}$$

The proof of $|_{n+1}$ is similar. □

Theorem 3.1, Corollary 5.4 and Theorem 7.1 yield:

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

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

holds.

Proof. *By induction on the structure of P. In particular, dealing with the case* $P = \mu x[Q]$ *uses continuity of* $\mathfrak{D}_{str}$, $\mathfrak{D}_{obs}$ *as well as strictness of* Ψ, *that is* $\Psi(\{\bot\}) = Obs(A)$. □

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

8. THE LANGUAGE $\mathcal{L}^*$ WITH HIDING

In this section we continue with the observation semantics. We wish to extend the previous language $\mathcal{L}$ to a language $\mathcal{L}^*$, again with $P,Q \in \mathcal{L}^*$, which includes a *hiding operator* $P \setminus b$ for every $b \in A$ [BHR]:

8.1 DEFINITION ($\mathcal{L}^*$)

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

8.2 REMARKS. Hiding an action b in a concurrent process P means that b is removed from the visible semantic behaviour of P. For example, the equation

$$(a;b;c) \setminus b = a;c \tag{1}$$

will hold in our semantics. Operationally, we imagine that the hidden action b occurs autonomously or *invisibly* after action a has finished. Once the hidden b has finished, action c can be performed. Thus a hidden action corresponds to the idea of an *ϵ-move* in automata theory (cf. e.g. [HU]).

With this operational idea of hiding in mind, equation (1) indicates that we abstract away from any notion of real time [Br2] in our semantics. Under a real time assumption, $(a;b;c) \setminus b$ would differ from $a;c$ in that it takes one time unit more to execute.

How to capture this informal idea of hiding within the observation semantics $\mathfrak{D}_{Obs}$? We first try to define a corresponding semantic operator

$$\cdot \setminus b^{Obs} : \mathcal{P}_{sat}(Obs(A)) \rightarrow \mathcal{P}_{sat}(Obs(A))$$

by putting for $H \in \mathcal{P}_{sat}(Obs(A))$

$$H \setminus b^{Obs} = \{h \setminus b \mid h \in H\} \tag{2}$$

where $h \setminus b$ results from h by removing every occurrence of b in h.

8.3 EXAMPLES. $(abc) \setminus b = ac$, $(abbb) \setminus b = a$.

Definition (2) seems natural, but unfortunately it is wrong for two reasons. First, it is not well-defined.

8.4 COUNTEREXAMPLE. Take

$$H_b = \{b^n \mid n \geq O\} \in \mathcal{P}_{sat}(Obs(A)).$$

Then

$$H_b \setminus b^{Obs} = \{\epsilon\} \notin \mathcal{P}_{sat}(Obs(A))$$

as $\{\epsilon\}$ violates the extension property (iii) of saturated sets of observations. □

Secondly, definition (2) is not $\supseteq$-continuous.

Note that the obvious *hiding relation* $R_b \subseteq Obs(A) \times Obs(A)$ with

$$hR_bh' \text{ iff } h' = h \setminus b$$

is not domain finite, so Lemma 6.1 is not applicable to definition (2). In fact, we have the

8.5 COUNTEREXAMPLE. Taking

$$H_n = \{\epsilon, \surd, b, ..., b^n\} \cup \{b^n \cdot h \mid h \in Obs(A)\},$$

$n \geq 0$, yields a chain

$$H_0 \supseteq H_1 \supseteq ... \supseteq H_n \supseteq ...$$

in $\mathcal{P}_{sat}(Obs(A))$, but

$$(\bigcap_{n=0}^{\infty} H_n) \setminus b^{Obs} = \{\epsilon, \surd\}$$

$$\neq (A\setminus\{b\})^* \cup (A\setminus\{b\})^* \cdot \{\surd,\uparrow\} = \bigcap_{n=0}^{\infty} (H_n \setminus b^{Obs}). \quad \square$$

Inside the observation semantics $\mathcal{D}_{Obs}$ it is not obvious how to correct definition (2) such that these problems are solved. At this moment it is very helpful to recall the isomorphism Ψ between streams and observations. Through Ψ we will get an insight how to modify (2).

To settle the first counterexample, take

$$X_b = \{b^\omega\}$$

in $\mathcal{P}_{ncf}(Str(A))$. Then $\Psi(X_b) = H_b$. Note that X_b can be approximated as $X_b = \bigsqcup_{n=0}^{\infty}{}_S X_b^{[n]}$ with

$$X_b^{[n]} = \{b^n \perp\}.$$

Now, if we apply the suggested definition (2) of hiding - denoted here by $\cdot \setminus b^{Str}$ instead of $\cdot \setminus b^{Obs}$ - to each approximation $X_b^{[n]}$ individually, we obtain

$$X_b^{[n]} \setminus b^{Str} = \{\perp\}.$$

Finally, setting $X_b \setminus b^{Str} = \bigsqcup_{n=0}^{\infty}{}_s (X_b^{[n]} \setminus b^{Str})$ yields

$$X_b \setminus b^{Str} = \{\perp\}.$$

Going back to the observation semantics, we thus expect

$$H_b \setminus b^{Str} = \Psi(X_b \setminus b^{Str}) = Obs(A),$$

the least element in $\mathcal{P}_{sat}(Obs(A))$.

8.6 Remarks. Through a purely mathematical argument in the stream semantics we realize that hiding an infinite stream of b's, e.g. $(\mu x[b;x])\setminus b$, should be semantically identical to unguarded recursions, e.g. $\mu x[x]$. This identification has a very clear intuitive interpretation as well: both $(\mu x[b;x]\setminus b$ and $\mu x[x]$ can be seen as instances of *divergence* where operationally an infinite internal loop is pursued (cf. Remark 4.3).

Note that in general the infinite stream of hidden b's need not start at the very beginning, but may occur only after some "ordinary" actions have been finished, e.g. in $P = a;a;(\mu x[b;x])\setminus b$. Then, using an analogous detour via streams as above, we see that P will be identified with $a;a;\mu x[x]$ and not with $\mu x[x]$ as divergence happens only after the a's.

We shall not state the full definition of hiding in the stream semantics here, but rather take our stream analysis of hiding as a motivation for the following correction of definition (2) in the observation semantics:

$$H\setminus b^{Obs} = \{h\setminus b \mid h\in H\}$$
$$\cup\{(h\setminus b)h' \mid \forall n\geqslant O : hb^n \in H \text{ and } h'\in Obs(A)\}.$$

This is the typical form of the hiding definition known from [BHR].

8.7 Proposition. *The operator* $\cdot\setminus b^{Obs}$ *is well-defined and* $\supseteq$*-monotonic.*

But what about $\supseteq$-continuity of $\cdot\setminus b^{Obs}$? Lemma 6.1 is not applicable as $\cdot\setminus b^{Obs}$ is not the pointwise image of the hiding relation R_b, which is not domain finite anyway. Instead of giving a direct proof for continuity, we apply here a general theorem of [OH 1/2] analyzing the typical structure of hiding operators.

A *domain finite well-founded structure* $(W, \rightarrow)$ consists of a set W and a domain finite relation $\rightarrow \subseteq W$ such that there exists no infinite chain

$$\ldots \rightarrow x_2 \rightarrow x_1 \rightarrow x_0$$

of elements $x_i \in W$. By a *grounded chain* of length $n \geqslant 0$ for $x \in W$ we mean a chain

$$x_0 \rightarrow \ldots \rightarrow x_n = x$$

such that x_0 is minimal with respect to $\rightarrow$ in W. To every element $x \in W$ we assign a *level* $\|x\|$ defined by

$$\|x\| = \min\{n \geqslant 0 \mid \exists \text{grounded chain of length } n \text{ for } x\}.$$

A subset $X \subseteq W$ is called *generable with respect to* $\rightarrow$ if for every element $x \in X$ there exists a grounded chain

$$x_0 \rightarrow \ldots \rightarrow x_n = x$$

inside X, i.e. with $\{x_0, \ldots, x_n\} \subseteq X$. By $\mathcal{P}_g(W)$ we denote the set of all subsets $X \subseteq W$ which are generable with respect to $\rightarrow$.

8.8 EXAMPLE. $(Obs(A), \rightarrow)$ with $\rightarrow \subseteq Obs(A) \times Obs(A)$ defined by

$$h \rightarrow h' \text{ iff } \exists \alpha \in A \cup \{\surd, \uparrow\} : h \cdot \alpha = h'$$

is a domain finite well-founded structure. Note that $\overset{*}{\rightarrow}$, the reflexive, transitive closure of $\rightarrow$, is the prefix relation $\leqslant$ on $Obs(A)$ and $\|h\|$ is the length of h. Here a set $X \subseteq Obs(A)$ is generable with respect to $\rightarrow$ iff it is prefix closed - one of the conditions for saturated sets of observations in Definition 4.4.

Let now $(\mathbb{X}_i, \rightarrow_i)$, $i = 1,2$, be domain finite well-founded structures and R some relation $R \subseteq \mathbb{X}_1 \times \mathbb{X}_2$. We consider now operators

$$op : \mathcal{P}_g(\mathbb{X}_1) \rightarrow \mathcal{P}(\mathbb{X}_2).$$

Besides the known pointwise image operator $op_R : \mathcal{P}_g(\mathbb{X}_1) \rightarrow \mathcal{P}(\mathbb{X}_2)$ of Section 6 (here restricted to $\mathcal{P}_g(\mathbb{X}_1)$) we consider an additional operator

$$op_R^{\infty} : \mathcal{P}_g(\mathbb{X}_1) \rightarrow \mathcal{P}(\mathbb{X}_2)$$

defined by

$$op_R^{\infty}(X) = \{y' \in \mathbb{X}_2 \mid \exists y \in \mathbb{X}_2 \overset{\infty}{\exists} x \in X : (xRy \text{ and } y \rightarrow_2^* y')\}$$

where $\overset{\infty}{\exists}$ means "there exist infinitely many" and where $\rightarrow_2^*$ denotes the reflexive, transitive closure of $\rightarrow_2$. We will study the combined operator

$$op_R \cup op_R^{\infty} : \mathcal{P}_g(\mathbb{X}_1) \rightarrow \mathcal{P}(\mathbb{X}_2).$$

To this end, we need the following concepts.

8.9 DEFINITION. $R \subseteq \mathbb{X}_1 \times \mathbb{X}_2$ is *level finite* if for every $y \in \mathbb{X}_2$ and $l \geqslant O$ there exist only finitely many $x \in \mathbb{X}_1$ with

$$xRy \text{ and } \|x\|_1 = l.$$

R is called *downward consistent* if

$$\rightarrow_1^* \circ R \subseteq R \circ \rightarrow_2^*$$

holds, with $\circ$ denoting the relational product.

8.10 THEOREM [OH2]. *The operator* $op_R \cup op_R^\infty : \mathcal{P}_g(\mathbb{X}_1) \to \mathcal{P}(\mathbb{X}_2)$ *is* $\supseteq$*-monotonic. Moreover, if R is level finite and downward consistent,* op_R^∞ *is also* $\supseteq$*-continuous.*

To apply the theorem to the particular hiding operator $\cdot \backslash b^{Obs}$, we take $(\mathbb{X}_i, \to_i) = (Obs(A), \to)$, $i=1,2$ and $R = R_b \subseteq Obs(A) \times Obs(A)$ as the hiding relation with

$$hR_bh' \text{ iff } h' = h \backslash b.$$

Then R_b is indeed level finite and downward consistent. Since

$$\cdot \backslash b^{Obs} = (op_{R_b} \cup op_{R_b^\infty}) \restriction \mathcal{P}_{sat}(Obs(A))$$

holds, we obtain:

8.11 COROLLARY. *The operator* $\cdot \backslash b^{Obs}$ *is* $\supseteq$*-continuous.*

8.12 REMARK. One of us (J.J.Ch.M.) has obtained a proof of the continuity of hiding in the stream semantics framework. The proof requires various additional results and will be published separately.

These preparations allow us to extend the observation semantics $\mathcal{D}_{Obs}$ to $\mathcal{L}^*$ by adding the following clause to Definition 6.8:

8.13 DEFINITION.

$$\mathcal{D}_{Obs}[\![P \backslash b]\!](\gamma) = \mathcal{D}_{Obs}[\![P]\!](\gamma) \backslash b^{Obs}.$$

We see that the level finite hiding operator $P \backslash b$ is more difficult to deal with than the domain finite operators $P;Q$, P **or** Q and $P \| Q$, but the stream domain helped us to understand its definition.

9. CONCLUDING REMARKS

We have not included any notion of *global* nondeterminism like $+$ [Mil] or $\Box$ [BHR] nor any notion of *deadlock* like *stop* [BHR] or δ [BK] in $\mathcal{L}$ or $\mathcal{L}^*$. 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, Theoretical Computer Science **34** (1984), pp. 134-156.

[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, 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.

[Br] S.D. BROOKES, On the relationship of CCS and CSP, in: J.Diaz (Ed.), Proc. 10th

ICALP, LNCS 154 (Springer, Berlin, 1983) 83-96.

[BHR] S.D. Brookes, C.A.R. Hoare, A.W. Roscoe, A theory of communicating sequential processes, J. ACM 31 (1984) 560-599.

[Br1] 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.

[Br2] M. Broy, Applicative real time programming, in. R.E.A. Mason (Ed.), IFIP Information Processing 83 (North Holland, Amsterdam, 1983) 259-264.

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

[FLP] 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.

[HU] J.E. Hopcroft, J.D. Ullman, Formal Languages and their Relation to Automata (Addison-Wesley, Reading, Mass., (1969).

[Me] J.-J.Ch. Meyer, Fixed points and the arbitrary and fair merge of a fairly simple class of processes, Tech. Reports IR-89/IR-92, Free University, Amsterdam, 1984.

[Mi1] R. Milner, A calculus of comunicating 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, Theoretical Computer Science 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 comunicating 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 comunicating processes, Tech. Monograph PRG-37, Progr. Research Group, Oxford Univ., 1984 (to appear in Acta Informatica).

[Pl1] G.D. Plotkin, A powerdomain construction, SIAM J. Comput. 5 (1976) 452-487.

[Pl2] 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, Theoretical Computer Science 34 (1984), 33-82.