

LINEAR TIME AND BRANCHING TIME SEMANTICS FOR RECURSION WITH MERGE

J.W. DE BAKKER, J.A. BERGSTRA and J.W. KLOP

*Department of Computer Science, Centre for Mathematics and Computer Science, Kruislaan 413,
1098 SJ Amsterdam, The Netherlands*

J.-J.Ch. MEYER

*Department of Computer Science, Free University, De Boelelaan 1081, 1081 HV Amsterdam, The
Netherlands*

Abstract. We consider two ways of assigning semantics to a class of statements built from a set of atomic actions (the 'alphabet'), by means of sequential composition, nondeterministic choice, recursion and merge (arbitrary interleaving). The first is linear time semantics (LT), stated in terms of trace theory; the semantic domain is the collection of all closed sets of finite and infinite words. The second is branching time semantics (BT), as introduced by De Bakker and Zucker; here the semantic domain is the metric completion of the collection of finite processes. For LT we prov. the continuity of the operations (merge, sequential composition) in a direct, combinatorial way.

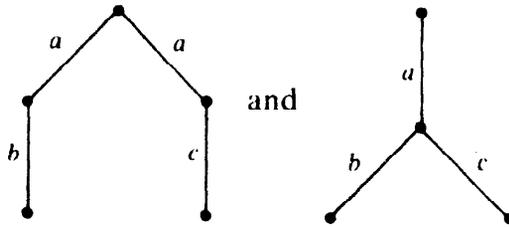
Next, a connection between LT and BT is established by means of the operation *trace* which assigns to a process its set of traces. We show that the trace set of a process is closed and that *trace* is continuous. This requires the compactness of the semantic domains, ensured by the finiteness of the alphabet. Using *trace*, we then can carry over BT into LT.

1. Introduction

We study two ways of assigning meaning to a simple language \mathcal{L} which has elementary actions (a, b, c, \dots), sequential composition, nondeterministic choice, recursion and merge (arbitrary interleaving) as its constituent concepts. This type of language may be seen as the core of various current approaches to parallelism (mostly to be extended with further concurrent concepts such as synchronization and communication, and often with simple iteration rather than full recursion), and it deserves in our opinion a full study of its associated semantics. There are a number of issues one encounters in developing a rigorous theory for this purpose.

Firstly, there is the issue of 'linear time' versus 'branching time', a terminology one finds, e.g., in investigations of the model theory of temporal logic. In fact, an important motivation for our investigation was to better understand this phenomenon. 'Linear time' is easy: it is nothing but trace theory. For example, in the linear time model both the statements $(a ; b) \cup (a ; c)$ and $a ; (b \cup c)$ obtain as associated meaning the so-called trace set $\{ab, ac\}$. 'Branching time' refers to an approach where one wants to distinguish between these two statements. Here for

the two statements we obtain as meaning the two trees



(Trees are not quite what we want, though. The statement $a \cup a$ should yield the object $a \uparrow$ rather than $a \vee a$ as its meaning, and there are further differences—to be explained below—between trees and the objects in the branching time universe.)

Secondly, the appearance of merge (\parallel) introduces various questions. For traces, " \parallel " is to be defined as the usual shuffle in the sense of language theory; for the branching time model a new definition is required. Also, various known results about context-free (or algebraic) languages, possibly with infinite words, have to be extended due to the addition of the " \parallel " operator.

Thirdly, in accordance with the emphasis which in the study of concurrency is put onto nonterminating computations, we want to include a mathematically rigorous treatment of finite *and infinite* actions specified by the programs in our language. For example, employing the μ -notation for recursion, we want as (linear time) meaning of $\mu x[a;x]$ the sequence a^ω (the infinite sequence of a 's), and for $\mu x[(a;x) \cup b]$ the set of sequences $(a^*b) \cup a^\omega$. The trace theory to be developed below is a continuation of the investigation of languages of infinite words by Nivat and his school [10–13]. The inclusion of the " \parallel " operation is responsible for further technical problems which—as far as we know—are not dealt with in their work in a way resembling our approach. (Also, in cases where Nivat addresses questions of semantics, these concern languages which are completely different from our \mathcal{L} .)

The development of the models for linear time and branching time semantics (from now on abbreviated to LT and BT) starts with a few tools from metric topology. For LT, not much more is used than the definition of *distance* between words. E.g., $d(abc, abde) = 2^{-3+1}$ where 3 is the index where the sequences exhibit their first difference. Next, a notion of *closed* set (closed with respect to d) is introduced. For example, the set a^* is not closed since it does not contain its limit point a^ω . The framework for LT semantics is then taken as the complete partially ordered set of closed sets, with " \supseteq " (set containment) as the " \sqsupseteq " ordering of the cpo. For BT we use the (mathematical) notion of *process* which is an element of a *domain* of processes obtained as solution of a domain equation by topological *completion* techniques. Domain equations have been studied extensively by Scott [15, 16] and, in a nondeterministic setting and using category theory, by Plotkin [14] and Smyth [17]. The theory of processes has been described elsewhere [3, 4], and is included here to facilitate comparison between the LT and BT semantics (and to make the paper more self-contained).

Section 2 is devoted to LT semantics, Section 3 to BT semantics, and Section 4 to the relationship between the two, and to some variations on the preceding definitions. Some of the proofs which support the mathematical theory are contained in Appendices A and B.

2. LT semantics: Mathematical background and semantic equations

Let A be an alphabet with elements a, b, \dots . (Most of the results below hold when A is finite or infinite. In a few cases, we require A to be finite.) Let x, y, \dots be statement variables from a set $\mathcal{S}tmv$, which we shall use in the formation of *recursive* or μ -statements. The syntax for the language \mathcal{L} is given (in a self-explanatory BNF notation) as follows.

2.1. Definition

$$S ::= a \mid S_1 ; S_2 \mid S_1 \cup S_2 \mid S_1 \parallel S_2 \mid x \mid \mu x[S].$$

2.1.1. Examples

$$(a ; b) \cup (a ; c), \quad (a \parallel b) \cup (a \parallel c), \quad \mu x[(a ; x) \cup b],$$

$$\mu x[a ; (b \parallel x)] \parallel \mu y[b ; (a \parallel y)], \quad \mu x[(a ; \mu y[(b ; y) \parallel x]) \cup c].$$

2.1.2. Remarks. (1) Syntactic ambiguities should be remedied by using parentheses or conventions for the priority of the operations.

(2) (For the reader who is not familiar with the μ -notation.) A term such as $\mu x[(a ; x) \cup b]$ has the same meaning as a *call* of the procedure declared (in an ALGOL-like language) by $P \Leftarrow (a ; P) \cup b$, or, alternatively, generates the same language (of finite *and infinite* words) as the grammar $X \rightarrow aX \mid b$.

(3) In a term $\mu x[S]$, x may occur ‘guarded’ in S , i.e., when S has the form $a ; (--x--)$: a recursive ‘call’ of x is guarded by at least one elementary action $a \in A$. Terms like $\mu x[x]$, $\mu x[x ; b]$ or $\mu x[a \parallel x]$ contain unguarded occurrences of x . (In language theory, the equivalent notion is the ‘Greibach condition’, as in [12].) Certain results below are—though mathematically correct—not necessarily semantically satisfactory for statements with unguarded variables.

We now turn to the development of the underlying semantic framework.

2.2. Definition. (a) $A^\infty = A^* \cup A^\omega$, where A^* is the set of all finite words over A , and A^ω the set of all infinite words.

(b) \leq denotes the usual prefix relation (a partial order) on A^∞ . The prefix of $x \in A^\infty$ of length n will be denoted by $x[n]$.

(Examples: $abc \leq abccb$; $abccb[3] = abc$; $abc[5] = abc$; $abc[0]$ is the empty word.)

(c) Let $x, y \in A^\infty$. The distance or *metric* $d : A^\infty \rightarrow [0, 1]$ is defined by

$$d(x, y) = \begin{cases} 2^{-\min\{n \mid x[n] \neq y[n]\} + 1} & \text{if } (\exists n)(x[n] \neq y[n]), \\ 0 & \text{otherwise (i.e., if } x = y). \end{cases}$$

(d) $\mathcal{P}_c(A^\infty)$ denotes the collection of all *closed* subsets of A^∞ . Here ‘closed’ refers to the metric d , i.e., $X \in \mathcal{P}_c(A^\infty)$ whenever each Cauchy sequence $\langle x_n \rangle_n$ has a limit in X . (By definition, the elements of a Cauchy sequence have arbitrarily small distances for sufficiently large index.) In the sequel we shall use—for brevity— \mathcal{C} for the collection $\mathcal{P}_c(A^\infty)$.

We define the order “ \sqsubseteq ” on \mathcal{C} by putting $X \sqsubseteq Y$ iff $X \supseteq Y$ (with “ \supseteq ” set-containment).

2.2.1. Notation. Often our notation will not reflect the difference between x and $\{x\}$, for $x \in A^\lambda$. Thus we may write $a^*b \cup a^\omega$ where really $a^*\{b\} \cup \{a^\omega\}$ is meant.

2.3. Lemma. d is a metric on A^∞ , and \mathcal{C} is a complete partially ordered set with respect to \sqsubseteq , with A^λ as bottom element and with $\bigsqcup_n X_n = \bigcap_n X_n$, for $\langle X_n \rangle_n$ a \sqsubseteq -chain.

For later use (in Section 4) we introduce one further definition with a theorem and a corollary.

2.4. Definition (Hausdorff distance). For any metric space (M, d) , $x, y \in M$ and $X, Y \subseteq M$ we define distances \hat{d}, \tilde{d} as follows:

- (a) $\hat{d}(x, Y) = \inf\{d(x, y) \mid y \in Y\}$, where $\inf \emptyset = 1$,
- (b) $\tilde{d}(X, Y) = \max(\sup\{\hat{d}(x, Y) \mid x \in X\}, \sup\{\hat{d}(y, X) \mid y \in Y\})$ where $\sup \emptyset = 0$.

2.5. Theorem. (a) \tilde{d} is a metric for $\mathcal{P}_c(M)$.

(b) If (M, d) is complete, then so is $(\mathcal{P}_c(M), \tilde{d})$. Also, for $\langle X_n \rangle_n$ a Cauchy sequence in $\mathcal{P}_c(M)$, we then have that $\lim_n X_n = \{x \mid x_n \rightarrow x, \text{ with } x_n \in X_n\}$.

Proof (see, e.g., [6]). A complete proof of (b) is contained in [4]. □

2.6. Corollary. The Hausdorff metric on \mathcal{C} turns it into a complete metric space.

The Hausdorff metric on \mathcal{C} will be written as d_H (to be contrasted with the Hausdorff metric d_B on \mathcal{P} , in Section 3).

In Section 4 we will need the following connection between the metric on \mathcal{C} and its cpo structure.

2.7. Proposition. Let $\langle X_n \rangle_n$ be both a Cauchy sequence in \mathcal{C} and a \sqsubseteq -chain. Then

$$\bigsqcup_n X_n = \lim_n X_n.$$

Proof. By Theorem 2.5 we must prove that

$$\bigcap_n X_n = \{x \mid x = \lim_n x_n, \text{ for some } x_n \in X_n\}.$$

Here (\subseteq) is trivial.

(\supseteq): let $x = \lim_n x_n$ for some sequence $\langle x_n \rangle_n$ such that $x_n \in X_n$. Since $X_n \subseteq X_0$ for all n , we have $x_n \in X_0$. Since X_0 is closed, $x \in X_0$. Likewise $x = \lim_n x_{n+1}$ is an element of X_1 , etc. Hence $x \in \bigcap_n X_n$. \square

We shall use \mathcal{C} with its cpo structure as semantic domain for the trace semantics of \mathcal{L} . (By Corollary 2.6, \mathcal{C} is also a complete metric space. However, contrary to the situation for BT semantics, we find the cpo structure more convenient for the LT semantics.) We need two theorems to support \mathcal{C} as model. (Technically, these two theorems are among the main results of this paper.) First we give the natural definitions of the basic operations on A^∞ and \mathcal{C} .

2.8. Definition. (a) For $x, y \in A^\infty$, $x \cdot y$ (mostly written as xy) is the usual *concatenation* of sequences (including the convention that $xy = x$ for $x \in A^\omega$). Further, $x \parallel y$ is the set of all *shuffles* of x with y (extending to the infinite case the classical definition of the shuffle of two finite words).

(b) $X \cup Y$ is the set-theoretic union of X and Y ; $X \cdot Y = \{x \cdot y \mid x \in X, y \in Y\}$, and $X \parallel Y = \bigcup \{x \parallel y \mid x \in X, y \in Y\}$. We will also write XY for $X \cdot Y$.

The main theorems of this section state that the operations \cdot, \cup, \parallel preserve closedness and are *continuous* (in the usual cpo sense) in both their arguments. (But note the proviso in Theorem 2.10).

2.9. Theorem. For X, Y in \mathcal{C} , $X \cdot Y, X \cup Y$ and $X \parallel Y$ are in \mathcal{C} .

The proof will be given in Appendix A.

2.10. Theorem. Let A be finite. Then the operations \cdot, \cup, \parallel from $\mathcal{C} \times \mathcal{C}$ to \mathcal{C} are continuous in both their arguments.

The proof will be given in Appendix A.

2.10.1. Remark. The finiteness condition on A ensures *compactness* of A^∞ (as observed in [12]). We then have that each sequence in A^∞ has a convergent subsequence. It is readily seen that this implies that, for each \sqsubseteq -chain $\langle X_n \rangle_n$ such that $X_n \neq \emptyset$ for all n , we have that $\bigcap_n X_n \neq \emptyset$, and this fact is needed in the proof of Theorem 2.10. We do not know whether this fact can also be enforced by weaker conditions than the finiteness of A . A possibility circumventing the need for it would be to define

$$X \cdot \emptyset = \emptyset \cdot X = X \parallel \emptyset = \emptyset \parallel X = X.$$

However, this has certain semantic disadvantages which become manifest, e.g., when elementary actions are articulated to assignments and tests (assuming that a test which has the value *false* in some state, delivers the empty state set as a result).

We proceed with the definition of the linear time semantics for \mathcal{L} . We adopt the usual technique with environments to deal with (free) statement variables. Let $\Gamma = \mathcal{P}t_{mv} \rightarrow \mathcal{C}$, and let γ range over Γ . Let, as before, X range over \mathcal{C} , and let $\gamma\{X/x\}$ stand for the environment which is like γ , but for its value in x which is now X . Let $[\mathcal{C} \rightarrow \mathcal{C}]$ stand for the collection of all continuous functions from \mathcal{C} to \mathcal{C} , and let, for $\Phi \in [\mathcal{C} \rightarrow \mathcal{C}]$, $\mu\Phi$ denote its least fixed point. We have the following definition.

2.11. Definition. The semantic mapping $\llbracket _ \rrbracket_L : \mathcal{L} \rightarrow (\Gamma \rightarrow \mathcal{C})$ is given by

$$\llbracket a \rrbracket_L(\gamma) = \{a\}, \quad \llbracket S_1; S_2 \rrbracket_L(\gamma) = \llbracket S_1 \rrbracket_L(\gamma) \cdot \llbracket S_2 \rrbracket_L(\gamma),$$

$$\llbracket S_1 \cup S_2 \rrbracket_L(\gamma) = \llbracket S_1 \rrbracket_L(\gamma) \cup \llbracket S_2 \rrbracket_L(\gamma), \quad \llbracket S_1 \parallel S_2 \rrbracket_L(\gamma) = \llbracket S_1 \rrbracket_L(\gamma) \parallel \llbracket S_2 \rrbracket_L(\gamma)$$

and

$$\llbracket \mu x[S] \rrbracket_L(\gamma) = \mu\Phi_{S,\gamma} \quad \text{where } \Phi_{S,\gamma} = \lambda X. \llbracket S \rrbracket_L(\gamma\{X/x\}).$$

This definition is justified by the following lemma.

2.12. Lemma. (i) $\lambda X_1 \dots \lambda X_n. \llbracket S \rrbracket_L(\gamma\{X_i/x_i\}_{i=1}^n) \in [\mathcal{C} \rightarrow [\mathcal{C} \rightarrow \dots \rightarrow [\mathcal{C} \rightarrow \mathcal{C}] \dots]]$
(n factors \mathcal{C})

(ii) *The functions in (i) are monotonic.*

Proof. (i) Proving (i) is a routine matter (see, e.g., [1, Theorem 7.9]) once Theorem 2.10 is available.

(ii) Follows by a simple inductive proof. Alternatively: note that \mathcal{C} is also a complete lattice, and use the fact that in a complete lattice continuous functions are monotonic (see, e.g., [1]). \square

2.13. Corollary. $\llbracket \mu x[S] \rrbracket_L(\gamma) = \bigcap_n \Phi_{S,\gamma}^n(A^x)$ where $\Phi_{S,\gamma}$ is as in Definition 2.11.

Proof. The proof follows by Definition 2.11, Lemma 2.12(i) and the Tarski-Knaster fixed point theorem. \square

2.14. Examples

$$\begin{aligned} (1) \quad \llbracket \mu x[(a;x) \cup b] \rrbracket_L(\gamma) &= \mu[\lambda X. \llbracket (a;x) \cup b \rrbracket_L(\gamma\{X/x\})] \\ &= \mu[\lambda X. ((a \cdot X) \cup b)] = \bigcap_n X_n, \end{aligned}$$

where $X_0 = A^x$, and $X_{i+1} = (a \cdot X_i) \cup b$. Hence, $\bigcap_n X_n = a^*b \cup a^w$.

(2) The 'fixed point property':

$$\llbracket \mu x[S] \rrbracket_L(\gamma) = \llbracket S \rrbracket_L(\gamma \{ \llbracket \mu x[S] \rrbracket_L(\gamma) / x \})$$

or

$$\llbracket \mu x[S] \rrbracket_L(\gamma) = \llbracket S\{\mu x[S]/x\} \rrbracket_L(\gamma),$$

where in the second equation $\{\cdot/\cdot\}$ denotes syntactical substitution.

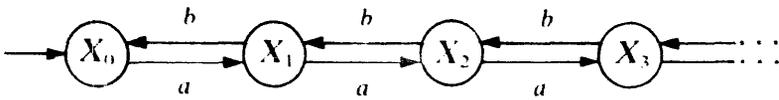
(3) Let S be $\mu x[a;(b \parallel x)]$. Then $\llbracket S \rrbracket_L(\gamma)$ is, remarkably, a nonregular set of traces. Using the fixed point property (2) and the following property of $\llbracket \cdot \rrbracket$, which is easily derived from the definition of $\llbracket \cdot \rrbracket$:

$$\bigcup a_i X_i \parallel \bigcup b_j Y_j = \bigcup a_i (X_i \parallel \bigcup b_j Y_j) \cup \bigcup b_j (Y_j \parallel \bigcup a_i X_i)$$

one computes (abbreviating $X_0 = \llbracket S \rrbracket_L(\gamma)$, $X_{n+1} = X_n \parallel b$):

$$X_0 = aX_1, \quad X_{n+1} = aX_{n+2} \cup bX_n \quad (n \geq 0),$$

from which it is easy to see that X_0 is the set of maximal traces in the labelled and directed graph



Alternatively, $X_0 = \{x \in A^\omega \mid (\forall n) |x[n]|_a \geq |x[n]|_b\}$, where $|x[n]|_a$ is the number of occurrences of a in $x[n]$, the prefix of x of length n .

(4) Let S be as in (3), $S' \equiv \mu x[b;(a \parallel x)]$ and $T \equiv \mu x[(a;x) \cup (b;x)]$. Then a similar computation as in (3) shows that

$$\llbracket S \parallel S' \rrbracket_L(\gamma) = \llbracket T \rrbracket_L(\gamma) = \{a, b\}^\omega.$$

2.15. Remark. For statements which have unguarded μ -terms, the semantics $\llbracket \cdot \rrbracket_L$ may not be the most natural one. E.g., we have—for any γ —

$$\llbracket \mu x[x] \rrbracket_L(\gamma) = A^\omega \quad \text{and} \quad \llbracket \mu x[x;b] \rrbracket_L(\gamma) = A^\omega.$$

We shall return to this point in Section 4, where we are in a position to compare both LT and BT semantics for such unguarded μ -terms.

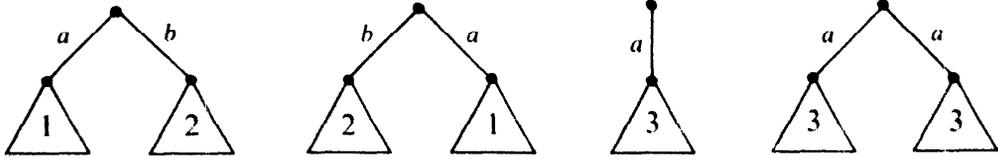
3. BT semantics: Mathematical background and semantic equations

The *branching time* semantics for \mathcal{L} is based on the theory of processes as sketched in [3] and described more fully in [4]. We briefly recall the main facts from this theory (in the terminology of [3, 4] referring only to uniform processes).

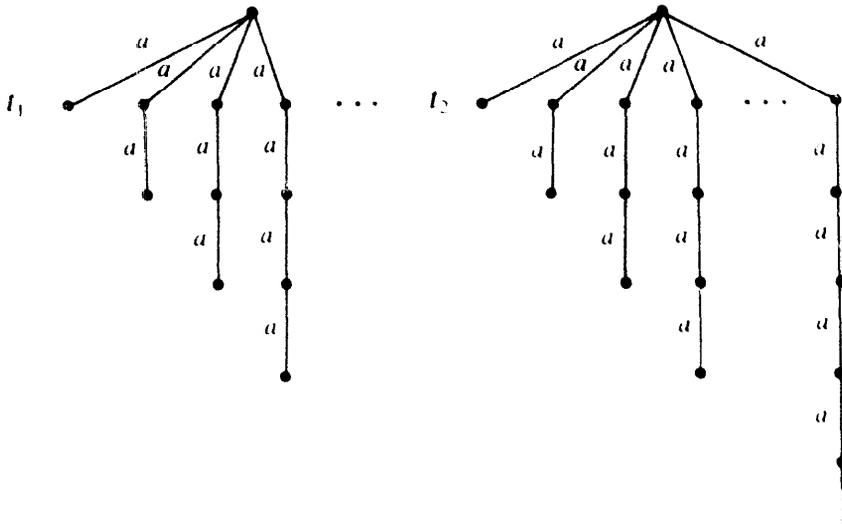
For an approach to uniform processes via projective limits, see [5]; and for an approach where processes are congruence classes of trees ('behaviours'), see [8, 9].

(Cf. also our Remark 4.3 for a brief comparison between the present uniform processes and Milner's behaviours.)

Here, processes are objects which are best compared to labelled unordered trees without repetitions in successor sets. Considering the examples



we have that the first and second, and the third and fourth represent the same process. Also, processes are closed objects: they contain all their limit points, in a sense to be made precise in a moment. E.g., the tree t_1 does not represent a process, but tree t_2 does, since it contains also the limit process "a $^\omega$ ". (Cf. Remark 4.3 where it is explained in what sense a tree represents a process.)



Technically, processes are obtained as follows:

Step 0. Start from the alphabet A as before; in addition, a so-called *nil*-process p_0 is assumed.

Step 1. Define P_n , $n = 0, 1, \dots$, by $P_0 = \{p_0\}$, $P_{n+1} = P(A \times P_n)$, where $P(\cdot)$ stands for the collection of all subsets of (\cdot) . Write $P_\omega = \bigcup_n P_n$.

Step 2. Introduce a metric on P_n (by suitably combining Definitions 2.2(c) and 2.4) and take \mathcal{P} as the *completion* of P_ω . Let d_B be the metric on \mathcal{P} .

We can then show the following theorem.

3.1. Theorem. $\mathcal{P} \cong \{p_0\} \cup P_c(A \times \mathcal{P})$.

(Here $P_c(\cdot)$ refers to the collection of all closed subsets of (\cdot) —with respect to d_B —and \cong denotes isometry.)

The next definition gives the main operations upon processes. We distinguish the cases $p = p_0$, $p = X \subseteq \mathcal{P}(A \times P_n)$ for some $n > 0$, or $p = \lim_i p_i$, with $\langle p_i \rangle_i$ a Cauchy sequence of elements p_i in P .

3.2. Definition

$$(a) \quad p \circ p_0 = p, \quad p \circ X = \{p \circ x \mid x \in X\}, \quad p \circ \langle a, q \rangle = \langle a, p \circ q \rangle,$$

$$p \circ \lim_i q_i = \lim_i (p \circ q_i),$$

(b) $p \cup p_0 = p_0 \cup p = p$, and, for $p, q \neq p_0$, $p \cup q$ is the set-theoretic union of p and q ,

$$(c) \quad p \parallel p_0 = p_0 \parallel p = p, \quad X \parallel Y = \{x \parallel Y \mid x \in X\} \cup \{X \parallel y \mid y \in Y\},$$

$$\langle a, p \rangle \parallel Y = \langle a, p \parallel Y \rangle, \quad X \parallel \langle a, q \rangle = \langle a, X \parallel q \rangle,$$

$$(\lim_i p_i) \parallel (\lim_j q_j) = \lim_k (p_k \parallel q_k).$$

3.3. Lemma. *The above operations are all well defined and continuous in both their arguments.*

This lemma is the counterpart of the results in Appendix A for the LT framework. The proof of the lemma—which does not require more effort than the LT case—is contained in [4].

By way of preparation for the definition of the recursive case we need a classical result. A mapping $T: \mathcal{P} \rightarrow \mathcal{P}$ is called *contracting* whenever $d_B(T(p), T(p')) \leq c \cdot d_B(p, p')$, with $0 \leq c < 1$. We have the following theorem.

3.4. Theorem. *If T is continuous and contracting, then for each $q \in \mathcal{P}$, the sequence $q, T(q), T^2(q), \dots$ is a Cauchy sequence converging to the unique fixed point of T .*

Proof. This is Banach's fixed point theorem. \square

3.5. Remark. Let $\sim: \mathcal{P} \rightarrow \{p_0\} \cup \mathcal{P}_c(A \times \mathcal{P})$ be the isometry whose existence was mentioned in Theorem 3.1. Then it is not hard to show that one can construct (via Cauchy sequences of approximations) elements p satisfying 'recursive definitions' such as

$$\tilde{p} = \{\langle a, p \rangle\} \quad \text{or} \quad \tilde{p} = \{\langle a, p \rangle \mid a \in A\} \cup \{\langle a, p_0 \rangle \mid a \in A\}.$$

(Moreover, the solutions to these equations are unique.)

Par abus de langage, we will omit reference to \sim henceforth and simply write $p = \{\langle a, p \rangle\}$ etc. Without this convention, an equation $p = \{\langle a, p \rangle\}$ could not have a solution, by the Axiom of Foundation of axiomatic set theory (ZF).

In the same vein we will speak about an infinite path $\langle a_1, p_1 \rangle, \langle a_2, p_2 \rangle, \dots$ such that $\langle a_{n+1}, p_{n+1} \rangle \in p_n$ for all $n > 1$. Here one should also read:

$$\langle a_{n+1}, p_{n+1} \rangle \in \tilde{p}_n.$$

Summing up, we can deal with '=' and '∈' in the usual way, without being bothered by the Axiom of Foundation.

As final preparatory step for the semantic definition we extend the alphabet A with a special so-called unobservable action τ and take as process domain the domain \mathcal{P}_2 given by

$$\mathcal{P}_2 \cong \{p_0\} \cup \mathcal{P}_c((A \cup \{\tau\}) \times \mathcal{P}_2).$$

As before, we apply the familiar environment technique. Let $\Gamma = \mathcal{P}tmv \rightarrow \mathcal{P}_2$. We define the BT-semantics for \mathcal{L} as follows.

3.6. Definition. The semantic mapping $\llbracket _ \rrbracket_B : \mathcal{L} \rightarrow (\Gamma \rightarrow \mathcal{P}_2)$ is given by

$$\llbracket a \rrbracket_B(\gamma) = \{\langle a, p_0 \rangle\},$$

$$\llbracket S_1 ; S_2 \rrbracket_B(\gamma) = \llbracket S_2 \rrbracket_B(\gamma) \circ \llbracket S_1 \rrbracket_B(\gamma),$$

$$\llbracket S_1 \cup S_2 \rrbracket_B(\gamma) = \llbracket S_1 \rrbracket_B(\gamma) \cup \llbracket S_2 \rrbracket_B(\gamma),$$

$$\llbracket S_1 \parallel S_2 \rrbracket_B(\gamma) = \llbracket S_1 \rrbracket_B(\gamma) \parallel \llbracket S_2 \rrbracket_B(\gamma),$$

$$\llbracket x \rrbracket_B(\gamma) = \gamma(x),$$

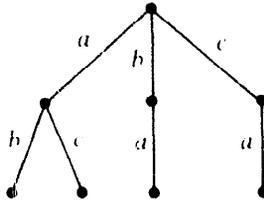
$$\llbracket \mu x[S] \rrbracket_B(\gamma) = \lim_i p_i, \quad \text{where } p_0 \text{ is the nil-process and}$$

$$p_{i+1} = \{\langle \tau, \llbracket S \rrbracket_B(\gamma\{p_i/x\}) \rangle\}.$$

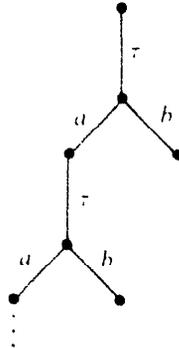
3.7. Examples. (For simplicity we omit γ .)

$$(1) \llbracket a_1 ; a_2 \rrbracket_B = \{\langle a_2, p_0 \rangle\} \circ \{\langle a_1, p_0 \rangle\} = \{\langle a_1, \{\langle a_2, p_0 \rangle\} \rangle\},$$

$$(2) \llbracket a \parallel (b \cup c) \rrbracket_B = (\text{in a natural picture representation})$$



(3) $\llbracket \mu x[(a;x) \cup b] \rrbracket_B = \lim_i p_i$, where $p_{i+1} = \{\langle \tau, \{\langle a, p_i \rangle, \langle b, p_0 \rangle\} \rangle\}$. In a picture we have



$$(4) \llbracket \mu x[x] \rrbracket_B = \llbracket \mu x[x;b] \rrbracket_B = \{\langle \tau, \{\langle \tau, \{\langle \tau, \dots \rangle\} \rangle\} \rangle\}.$$

3.8. Remark. The central clause is the definition of recursion $\mu x[S]$. We have solved this by introducing for each S an associated *contracting* mapping $T = \lambda p. \langle \tau, \llbracket S \rrbracket_B(\gamma\{p/x\}) \rangle$. Contractivity is enforced by the $\langle \tau, \dots \rangle$ construct. Operationally, the $\langle \tau, \dots \rangle$ action corresponds to the action of procedure entrance, which does not involve any ‘observable’ action in A . For such T , $\lim_i T^i(p_0)$ is its unique fixed point. (p_0 is only chosen for definiteness; other choices would of course yield the same result.) We shall return to the motivation for adopting this strategy in the next section.

4. LT and BT compared

In this section we compare the two semantics presented in Sections 2 and 3. More specifically, we discuss the relationship between LT and BT both for statements with *guarded* μ -terms only, and for statements with any form of recursion.

The main result of the section is stated in terms of the notion of *trace set* of a process. Roughly, the trace set of process p is the set of branches (terminating or infinite) obtained by viewing p as a labelled tree. Here we meet the following problem.

Remember that a finite path of process p terminates in p_0 or in \emptyset . Semantically, the latter case signals unsuccessful termination. Now there are two possibilities:

- (i) because in \mathcal{C} there is no way of signalling unsuccessful termination, we may decide to exclude paths ending in \emptyset from the trace set of p , or
- (ii) \mathcal{C} will be enriched with a *fail* symbol which may be appended to the end of a finite word over A .

The disadvantage of (ii) is technical: all the operations on \mathcal{C} have to take the *fail* possibility into account. (Although we are not prepared to do so here, it seems quite well possible to extend LT semantics in this way.)

The disadvantage of (i) is essential: the operation ‘trace’, which is defined below, would not be continuous. (For, consider $q = \lim_n q_n$ where $q_0 = \langle \langle a, \emptyset \rangle \rangle$, $q_{n+1} = \langle \langle a, q_n \rangle \rangle$. Then the trace set of q_n is \emptyset , but that of q is $\{a^\omega\}$.)

We will adopt the following solution: in the present case of ‘uniform’ processes, i.e., processes where the $a \in A$ are atomic actions and not further specified, the issue of unsuccessful termination is not yet at stake. In fact, a process p which is the denotation of an expression, $p = \llbracket S \rrbracket_B(\gamma)$, has no branches ending in \emptyset . Therefore we decide, in order to establish a correspondence between LT and BT semantics, to adopt the natural restriction to the closure of

$$\mathcal{P}_{\perp} = \{ \llbracket S \rrbracket_B(\gamma) \mid \text{closed } S, \gamma \in \mathcal{S}lmv \rightarrow \mathcal{P} \}.$$

(Note that \mathcal{P}_{\perp} itself is not closed.) We will write \mathcal{P}^+ for this closure. Obviously, \mathcal{P}^+ is a complete metric subspace of \mathcal{P} . An alternative characterization of \mathcal{P}^+ is

$$\mathcal{P}^+ = \{ p \in \mathcal{P} \mid \text{all terminating paths of } p \text{ end in } p_0 \}.$$

For use in Theorem 4.9 we note that

$$\mathcal{P}^+ = \{\llbracket S \rrbracket_B(\gamma) \mid \text{all } S, \gamma \in \mathcal{L}mv \rightarrow \mathcal{P}^+\}.$$

4.1. Definition. Let $p \in \mathcal{P}^+$.

(1) A *path* π for p is a (finite or infinite) sequence

$$\langle a_1, p_1 \rangle, \langle a_2, p_2 \rangle, \dots$$

such that $\langle a_i, p_i \rangle \in p$ and $\langle a_{i+1}, p_{i+1} \rangle \in p_i, i = 1, 2, \dots$

(2) (i) Let $\pi = \langle a_1, p_1 \rangle, \langle a_2, p_2 \rangle, \dots$ be an infinite path of $p \in \mathcal{P}^+$. Then $a_1 a_2 \dots \in A^\omega$ is called a *trace* of p .

(ii) Let $\pi = \langle a_1, p_1 \rangle, \dots, \langle a_n, p_n \rangle$ be a finite path of $p \in \mathcal{P}^+$. Then $a_1 a_2 \dots a_n \in A^*$ is a *trace* of p .

(3) $trace(p)$ is the set of traces of p .

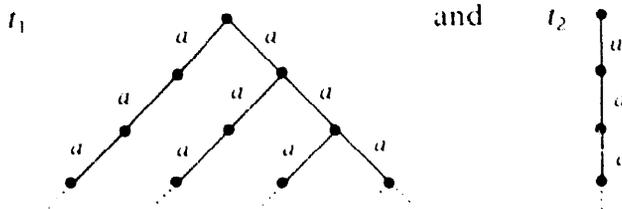
4.2. Examples

$$trace(\{\langle a, \{\langle b, p_0 \rangle\} \rangle, \langle a, \{\langle c, p_0 \rangle\} \rangle\}) = \{ab, ac\},$$

$$trace(\{\langle a, \{\langle a, \dots \rangle\} \rangle\}) = \{a^\omega\},$$

$$trace(\llbracket \mu x[(a;x) \cup b] \rrbracket_B(\gamma)) = (\tau a)^\omega \cup (\tau a)^* \tau b.$$

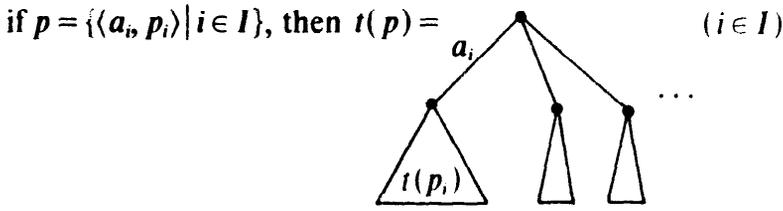
4.3. Remark. Clearly, the definition of traces of a process p suggests viewing p as a labelled tree $t(p)$, having the traces as branches. This view is not without difficulties, however: e.g., the labelled trees



should be identified as their approximations coincide. In [9], this consideration has given rise to the notion of a process (a 'behaviour') as the equivalence class of labelled trees (in fact, charts) modulo a congruence called 'bisimulation' (\approx). E.g., $t_1 \approx t_2$. A bisimulation is a certain relation R between the nodes of t_1, t_2 , where $Dom(R)$ is the set of nodes of t_1 and $Range(R)$ is the set of nodes of t_2 .

Now one can prove (we will not do so here) the following. If the bisimulation R is a *function*, write $t_1 > t_2$. (In the example this is the case.) We observe that an equivalence class of trees contains a unique $<$ -minimal tree. This is precisely the

tree which one can associate to a process p by the following definition:



provided the $\langle a_i, p_i \rangle$ are pairwise different.
 (I.e., the set $\{\langle a_i, p_i \rangle \mid i \in I\}$ contains no duplications.)

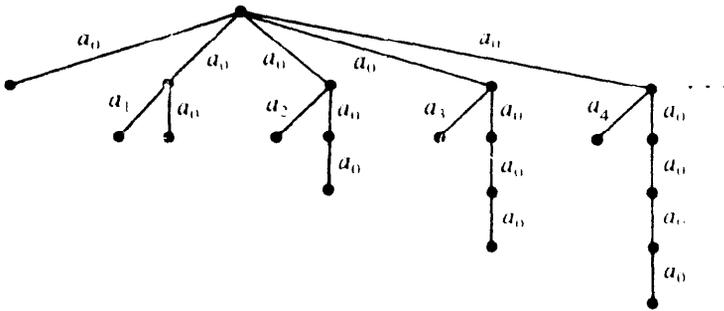
Moreover, if this proviso is dropped, then p can be developed in many different trees; in fact, to every tree which is bisimulation congruent with $t(p)$.

Example. If $p = \{\langle a, p \rangle\}$, then $t(p) = t_2$ above; also one can develop p to t_1 above, using at some points the representation $p = \{\langle a, p \rangle, \langle a, p \rangle\}$.

Note, finally, that congruent trees have the same set of branches.

Now we would like to assert that *trace* is an operation from \mathcal{P}^+ to \mathcal{C} , i.e., for $p \in \mathcal{P}^+$, $trace(p)$ is a closed set. Surprisingly, this need not to be the case if A is infinite: say $A = \{a_i \mid i \geq 0\}$.

4.4. Example. Consider $p \in \mathcal{P}^+$ as given by the tree



i.e., $p = \{\langle a_0, p_i \rangle \mid i \geq 0\}$ where p_0 is the *nil*-process, and, for $n > 0$,

$$p_n = \{\langle a_n, p_0 \rangle, \langle a_0, q_{n-1} \rangle\},$$

$$q_n = \{\langle a_0, \langle a_0, \langle a_0, \dots, \langle a_0, p_0 \rangle \rangle \dots \rangle\} \quad (n \text{ times } a_0).$$

Then $trace(p) = \{a_0^n \mid n \geq 1\} \cup \{a_0 a_m \mid m \geq 1\}$, which is not closed as it lacks a_0^ω .

However, with the additional assumption that A is finite, we have (by a nontrivial proof) that $trace(p)$ is closed indeed. In fact we have the following lemma.

4.5. Lemma. *Let A be finite. Then:*

- (i) $trace(p) \in \mathcal{C}$,
- (ii) *trace* is continuous (with respect to the Hausdorff metrics in \mathcal{P}^+ and \mathcal{C}).

The proof will be given in Appendix B.

We will also need the following fact, the proof of which is routine and omitted here.

4.6. Proposition. $trace: \mathcal{P}^+ \rightarrow \mathcal{C}$ is an homomorphism (with respect to the operations \cdot, \cup, \parallel on \mathcal{P}^+ and \mathcal{C}).

4.7. Remark. A corollary of Lemma 4.5(i) and Proposition 4.6, together with the obvious surjectivity of $trace$, is that

$$X, Y \in \mathcal{C} \Rightarrow X \parallel Y \in \mathcal{C}.$$

(For, given $X, Y \in \mathcal{C}$, take p, q such that $trace(p) = X$ and $trace(q) = Y$. Then $X \parallel Y = trace(p) \parallel trace(q) = trace(p \parallel q) \in \mathcal{C}$.)

However, we have preferred to give also a direct combinatorial proof of this fact in Appendix A.

We also need the notion of *universal process* for \mathcal{P}^+ .

4.8. Definition. The universal process for \mathcal{P}^+ , called p_u , is the (unique) solution of the equation

$$p = \{\langle a, p \rangle \mid a \in A\} \cup \{\langle a, p_0 \rangle \mid a \in A\}.$$

Note that $trace(p_u) = A^*$.

In the following, it will be convenient to restrict ourselves to *closed* statements, i.e., statements without free statement variables. Now the natural question which suggests itself concerning the relationship between LT and BT is whether, for each closed S —omitting γ which is then superfluous—we have that

$$trace(\llbracket S \rrbracket_B) = \llbracket S \rrbracket_L. \quad (*)$$

Taken as it stands, the answer to the question is “no”. For example, taking $S \equiv \mu x[x]$ we have that

$$trace(\llbracket \mu x[x] \rrbracket_B) = trace(\{\langle \tau, \{\langle \tau, \dots \rangle\} \rangle\}) = \{\tau^\omega\} \neq A^* = \llbracket \mu x[x] \rrbracket_L.$$

This discrepancy is not an essential phenomenon, but due to the special role of the unobservable action τ for BT semantics. Remember that τ was introduced to enforce contractivity of the mapping T as defined in Remark 3.8, which in turn was necessary to allow us to apply Banach’s fixed point theorem. However, another approach may also be adopted which will lead to a positive answer to the question (*). It is convenient to separately treat the cases where

- (i) S has only guarded μ -terms, and
- (ii) S may have unguarded μ -terms.

Case (i) (only guarded μ -terms). In this case the ‘ τ -trick’ for BT is in fact superfluous. Taking $T' = \lambda p. \llbracket S \rrbracket_B(\gamma\{p/x\})$, T' is now contracting for each S , and $\lim_{i \rightarrow \infty} T'(p_i)$, with p_1 arbitrary, $p_{i+1} = T'(p_i)$, converges to the unique fixed point of

T' independent of the initial p_1 —which we may therefore choose as p_u to facilitate the proof of the following theorem.

4.9. Theorem. *Assume statement S is closed and involves only guarded μ -terms. Let $\llbracket S \rrbracket_L$ be as before, and let $\llbracket S \rrbracket_B$ be as in Definition 3.6, except that in the clause for $\mu x[S]$, we replace p_0 by p_u and define*

$$P_{i+1} = \llbracket S \rrbracket_B(\gamma\{p_i/x\}).$$

Then,

$$\text{trace}(\llbracket S \rrbracket_B) = \llbracket S \rrbracket_L.$$

Proof. (The proof uses all the results of this paper except the present theorem.)

We will prove the following stronger fact, necessary for the induction on the structure of statements S' (which now need not to be closed): for every S' containing only guarded μ -terms, and for every $\gamma \in \mathcal{S}tmv \rightarrow \mathcal{P}^+$:

$$\llbracket S' \rrbracket_L(\text{trace} \circ \gamma) = \text{trace}(\llbracket S' \rrbracket_B(\gamma)).$$

Case (i). $S' \neq \mu x[S]$. Now the result easily follows by the induction hypothesis and the homomorphism properties of trace .

The interesting case is the following:

Case (ii). $S' \equiv \mu x[S]$.

Some notation is needed first: $\text{trace} \circ \gamma = \gamma'$. Further, we employ again the notation of Definition 2.11: $\Phi_{S,\gamma} = \lambda X. \llbracket S \rrbracket_L(\gamma'\{X/x\})$. Finally, p_n is defined as in the statement of the theorem.

First we prove the following:

Claim 1. $\text{trace}(p_n) = \Phi_{S,\gamma}^n(A^\infty)$.

Proof of Claim 1

$$\begin{aligned} \text{trace}(p_n) &= \text{trace}(\llbracket S \rrbracket_B(\gamma\{p_{n-1}/x\})) \\ &= (\text{by the induction hypothesis}) \llbracket S \rrbracket_L(\gamma'\{\text{trace}(p_{n-1})/x\}) \\ &= (\lambda X. \llbracket S \rrbracket_L(\gamma'\{X/x\}))(\text{trace}(p_{n-1})) \equiv \Phi_{S,\gamma}(\text{trace}(p_{n-1})). \end{aligned}$$

Hence,

$$\text{trace}(p_n) = \Phi_{S,\gamma}^n(\text{trace}(p_u)) = \Phi_{S,\gamma}^n(A^\infty).$$

Claim 2. $\bigcap_n \Phi_{S,\gamma}^n(A^\infty) = \lim_n \Phi_{S,\gamma}^n(A^\infty)$.

Proof of Claim 2. By the fact that only guarded μ -terms are considered, $\{p_n\}$ is a Cauchy sequence. By the continuity of trace (Corollary B.7), $\{\text{trace}(p_n)\}$ is therefore also a Cauchy sequence. So, by Claim 1, $\{\Phi_{S,\gamma}^n(A^\infty)\}$ is a Cauchy sequence.

Furthermore, the $\Phi_{S,\gamma}^n$ are monotonic (Lemma 2.12(ii)). Since A^∞ is the maximal element of \mathcal{C} , the sequence $\{\Phi_{S,\gamma}^n(A^\infty)\}$ is therefore decreasing (w.r.t. \subseteq). Now Claim 2 follows by Proposition 2.7.

Now we have

$$\begin{aligned}
& \llbracket S' \rrbracket_L(\text{trace} \circ \gamma) \\
& \equiv \llbracket \mu x[S] \rrbracket_L(\gamma') && \text{(by Corollary 2.13)} \\
& = \bigcap_n \Phi_{S,\gamma'}^n(A^\infty) && \text{(by Claim 2)} \\
& = \lim_n \Phi_{S,\gamma'}^n(A^\infty) && \text{(by Claim 1)} \\
& = \lim_n \text{trace}(p_n) && \text{(by Corollary B.7)} \\
& = \text{trace}(\lim_n p_n) && \text{(by the definition in the present theorem)} \\
& = \text{trace} \llbracket \mu x[S] \rrbracket_B(\gamma) \\
& \equiv \text{trace} \llbracket S' \rrbracket_B(\gamma). \quad \square
\end{aligned}$$

We continue with the second case.

Case (ii). S involves at least one unguarded μ -term. Now two ways of achieving (*) are available.

Firstly, we can maintain the definition of $\llbracket S \rrbracket_L$, and use the revised definition of $\llbracket S \rrbracket_B$ as stated in Theorem 4.9. The crucial difference is that the mapping T' is now no longer contracting in general, and we cannot use Banach's fixed point theorem to show that the sequence $p_u, T'(p_u), T'^2(p_u), \dots$ converges to a fixed point of T' . However, this fact has indeed—with some effort, and for arbitrary initial q —been established in [5]. Thus, we can base our revised definition on their theorem, and again obtain—by the same reasoning as in the proof of Theorem 4.9—that (*) holds.

Secondly, we may also keep the definition of $\llbracket S \rrbracket_B$ as in Definition 3.6, and revise that of $\llbracket S \rrbracket_L$. We then replace the last clause of Definition 2.11 by

$$\llbracket \mu x[S] \rrbracket_L(\gamma) = \mu[\lambda X. \llbracket \tau; S \rrbracket_L(\gamma\{X/x\})].$$

All this amounts to the idea of replacing, both for LT and for BT, $\mu x[S]$ by $\mu x[\tau; S]$, thus ensuring that all statements have only guarded terms, so that Theorem 4.9 applies again.

Appendix A: Well-definedness and continuity of the operations \cdot, \cup, \parallel on \mathcal{C}

We will now give the proofs of Theorem 2.9 and 2.10. For both theorems the case of " \cup " is trivial; this leaves us with the following four propositions, which we will treat together since their proofs have a common structure.

Theorems 2.9 and 2.10. (i) $X, Y \in \mathcal{C} \Rightarrow X \parallel Y \in \mathcal{C}$.

(ii) $X, Y \in \mathcal{C} \Rightarrow XY \in \mathcal{C}$.

(iii) Let A be finite. Let $X_n, Y_m \in \mathcal{C} (n, m \geq 0)$ be such that $X_0 \supseteq X_1 \supseteq \dots$ and

Consider $n = n_0$. Since $\{z_i\}$ is a Cauchy sequence, there is a k_0 such that the prefixes $z_i[n_0]$ are constant for $i \geq k_0$, namely equal to $z_{k_0}[n_0]$. This need not be the case for $\zeta_i[n_0]$. However, since there are only finitely many colorings of $z_{k_0}[n_0]$, there is (by the pigeon-hole principle) a subsequence $\{\zeta_{f_0(i)}\}$ of $\{\zeta_i \mid i \geq k_0\}$ such that the prefixes $\zeta_{f_0(i)}[n_0]$ are constant for all i . (Here f_0 is some monotonic function from \mathbb{N} to \mathbb{N} .)

Now consider $n_1 > n_0$. From the sequence $\{\zeta_{f_0(i)}\}$ we can in the same way extract a subsequence $\{\zeta_{f_1(f_0(i))}\}$ whose n_1 -prefixes are constant. Continuing this procedure we find a sequence $\{\zeta_{g(j)}\}$ where g is a monotonic function such that $g(j) = (f_j \circ \dots \circ f_0)(0)$, which evidently is a Cauchy sequence in $P_c(B^\infty)$. Call the limit ζ . Then ζ can be decomposed (by projections to A_ξ , respectively A_η) into ξ, η such that $\zeta \in \xi \parallel \eta$. Decoloring, we have $z \in x \parallel y$. Since z is the limit of $\{z_i\}$, we are through if $x \in X$ and $y \in Y$. This easily follows because X, Y are closed.

Proof of (ii). The proof is almost identical to that of (i): we only have to replace $X \parallel Y$ by XY , and $z_i \in x_i \parallel y_i$ by $z_i = x_i y_i$ etc. (In the diagram of the proof of (i): the 'blue' parts precede the 'red' parts, instead of being mixed.)

Proof of (iii). (\subseteq) is trivial.

(\supseteq): Take $z \in \bigcap (X_i \parallel Y_i)$, so, for all i , $z \in x_i \parallel y_i$ for some $x_i \in X_i$ and $y_i \in Y_i$. Again, find colored versions ζ_i, ξ_i, η_i such that $\zeta_i \in B^\infty, \xi_i \in A_\xi^\infty, \eta_i \in A_\eta^\infty, h(\zeta_i) = z, h(\xi_i) = x_i, h(\eta_i) = y_i$ and $\zeta_i \in \xi_i \parallel \eta_i$. Construct ζ, ξ, η such that $\zeta \in \xi \parallel \eta$ as in (i).

Let $h(\xi) = x$ and $h(\eta) = y$. It remains to show that $x \in \bigcap X_n$ and $y \in \bigcap Y_m$. This follows because for each prefix x' of x there is a p such that $x' \leq x_p \in X_p \subseteq X_0$. Since X_0 is closed, it follows that $x \in X_0$; likewise $x \in X_1$, etc.

The finiteness condition on A is used to ensure that $\bigcap X_n \neq \emptyset$ and $\bigcap Y_m \neq \emptyset$. The nonemptiness of these intersections is needed in the case that $\zeta \in A_\xi^\infty$ or $\zeta \in A_\eta^\infty$ (i.e., ζ is entirely 'blue' or 'red'). In that case we need to pick an arbitrary η respectively ξ such that $h(\eta) = y \in \bigcap Y_m$ respectively $h(\xi) = x \in \bigcap X_m$, to be able to write $\zeta \in \xi \parallel \eta$ and $z \in x \parallel y$.

Proof of (iv). This proof is again mutatis mutandis identical to that of (iii). (Here we only need the nonemptiness of $\bigcap Y_m$.) \square

Appendix B. The operation $trace: \mathcal{P}^+ \rightarrow \mathcal{C}$

We will now prove Lemma 4.5, stating that, for finite A , the operation $trace: \mathcal{P}^+ \rightarrow \mathcal{C}$ is well defined and continuous. In order to do so, we need some preliminary facts which also have some independent interest. First we recall a definition from [3, 4].

B.0. Definition. (i) Let $p \in P_\omega$ (the set of processes of finite depth). Define $p[n]$ ($n \geq 0$) as follows:

- (1) if $p = p_0$, then $p[n] = \hat{p}_0$ ($n \geq 0$),
- (2) otherwise $p[0] = p_0, p[n+1] = \{\langle a, q[n] \rangle \mid \langle a, q \rangle \in p\}$.

(ii) Let $p \in \mathcal{P} \setminus P_\omega$. Then $p = \lim_i p_i$, for a Cauchy sequence $\{p_i\}_i$ with $p_i \in P_i$. Now $p[n] = \lim_i p_i[n]$.

B.1. Proposition. (i) Let d_B be the Hausdorff metric on \mathcal{P} . Let δ_B be the metric on \mathcal{P} defined by

$$\delta_B(p, q) = \begin{cases} 2^{-\min\{n | p[n] \neq q[n]\} + 1} & \text{if } (\exists n)(p[n] \neq q[n]), \\ 0 & \text{otherwise (in which case } p = q). \end{cases}$$

Then $d_B(p, q) = \delta_B(p, q)$ for all p, q .

(ii) Let d_L be the Hausdorff metric on \mathcal{C} . Let δ_L be the metric on \mathcal{C} defined analogous to δ_B . (For $X \in \mathcal{C}$, $X[n] = \{x[n] | x \in X\}$.)

Then $d_L(p, q) = \delta_L(X, Y)$ for all X, Y .

Proof. The proof is a mere routine matter. \square

B.2. Proposition. Let q_0, q_1, \dots be a Cauchy sequence in \mathcal{P}^+ with limit q . Suppose $x_i \in \text{trace}(q_i)$, $i \geq 0$, and x_0, x_1, \dots is a Cauchy sequence (in A^∞) with limit x .

Then $\text{trace}(q)$ contains a Cauchy sequence x'_0, x'_1, \dots with limit x .

Proof. The proof is immediate, via the metric δ_B (the n -prefixes of the traces of q_k can be made to coincide with those of q , for arbitrary n , by taking k large enough, i.e., such that $\delta_B(q_k, q) < 2^{-n}$). \square

B.3. Remark. The preceding proposition can be rephrased as follows:

$$\lim_n \overline{\text{trace}(q_n)} \subseteq \overline{\text{trace}(\lim_n q_n)}.$$

(Here, the overbar denotes the closure operator.)

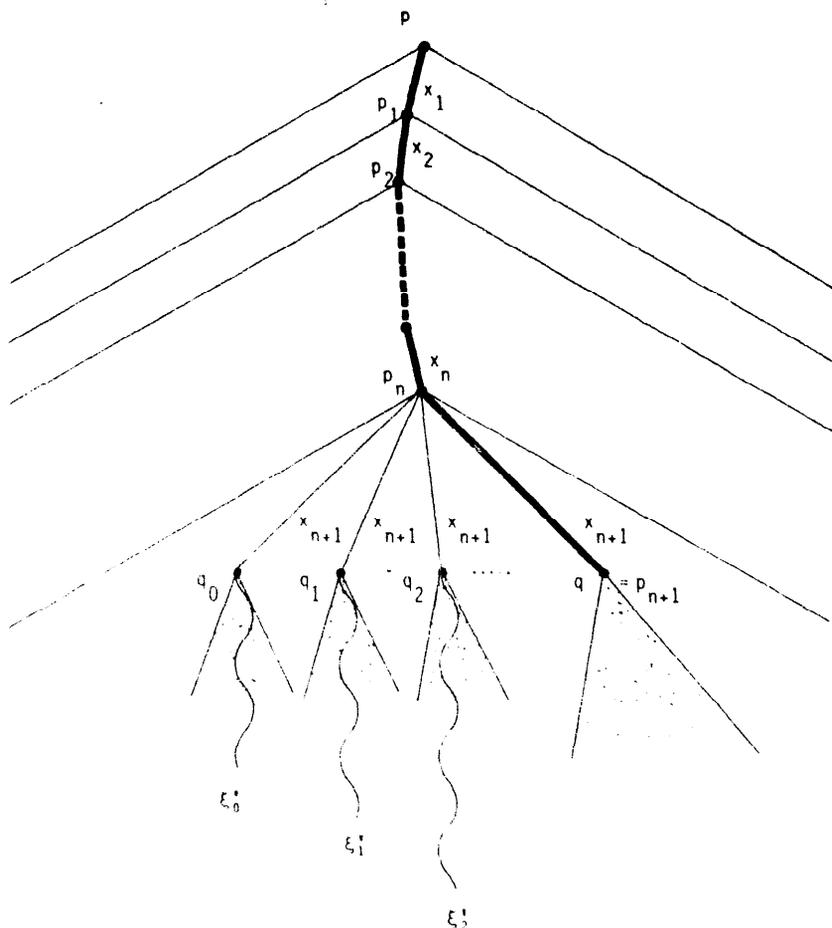
B.4. Lemma. Let A be finite. Then $\text{trace}(p)$ is closed for all $p \in \mathcal{P}^+$.

Proof. Suppose $x \in \overline{\text{trace}(p)}$. We have to prove $x \in \text{trace}(p)$. First we introduce the notation $x\langle n \rangle$ to denote the result of removing the prefix $x[n]$ from x . So $x = x[n]x\langle n \rangle$. Further, write $x = x_1 x_2 \dots$.

Now we define by induction on n a path $\langle x_1, p_1 \rangle, \langle x_2, p_2 \rangle, \langle x_3, p_3 \rangle, \dots$ in p . The result is (by Definition 4.1) that $x \in \text{trace}(p)$.

Basis. Start with p and the hypothesis $x \in \overline{\text{trace}(p)}$.

Induction step. Suppose p_n is defined. The induction hypothesis is $x\langle n \rangle \in \overline{\text{trace}(p_n)}$.



From the induction hypothesis we have

$$x(n) = \lim_m \xi_m$$

for some converging sequence $\{\xi_m\}$ in $trace(p_n)$. (ξ_m depends also on n , but we will not reflect this in our notation.) Likewise, by removing the first symbol:

$$x(n+1) = \lim_m \xi'_m,$$

where $\xi'_m = \xi_m \langle 1 \rangle$.

Let q_m be such that $\xi'_m \in trace(q_m)$. Since A is finite, \mathcal{P}^+ is compact and so there is a converging subsequence $\{q_{f(m)}\}$ of $\{q_m\}$. (Here f is some monotonic function from \mathbb{N} to \mathbb{N} .) Let q be its limit. Since p_n is closed, we have $\langle x_{n+1}, q \rangle \in p_n$. Now p_n will be q .

Finally it follows by Proposition B.2 that

$$x(n+1) \in \overline{trace(p_{n+1})},$$

which is the induction hypothesis for p_{n+1} . \square

In order to prove the continuity of $trace$, we observe the following.

B.5. Proposition. For all $p, q \in \mathcal{P}^+$:

$$\delta_B(p, q) \geq \delta_L(\text{trace}(p), \text{trace}(q)).$$

Proof. The proof easily follows by the simple nature of the δ -metrics. \square

This fact is exploited via the following general lemma.

B.6. Lemma. Let (M_i, d_i) be complete metric spaces ($i = 1, 2$). Let $f: M_1 \rightarrow M_2$ be a function such that $d_1(x, y) \geq d_2(f(x), f(y))$. Then f is continuous, i.e., $\lim_n f(x_n) = f(\lim_n x_n)$.

Proof. Obvious. \square

B.7. Corollary. The operation $\text{trace}: \mathcal{P}^+ \rightarrow \mathcal{C}$ is continuous, i.e.,

$$\lim_n \text{trace}(p_n) = \text{trace} \lim_n (p_n)$$

for $\{p_n\}_n$ a Cauchy sequence in \mathcal{P}^+ .

References

- [1] J.W. de Bakker, *Mathematical Theory of Program Correctness* (Prentice-Hall, Englewood Cliffs, NJ, 1980).
- [2] J.W. de Bakker, J.A. Bergstra, J.W. Klop and J.-J.Ch. Meyer, Linear time and branching time semantics for recursion with merge, in: J. Díaz, ed., *Proc. ICALP 83*, Barcelona, Lecture Notes in Computer Science **154** (Springer, Berlin, 1983).
- [3] J.W. de Bakker and J.I. Zucker, Denotational semantics of concurrency, *Proc. 14th ACM Symp. on Theory of Computing* (1982) pp. 153–158.
- [4] J.W. de Bakker and J.I. Zucker, Processes and the denotational semantics of concurrency, *Inform. Control* **54** (1/2) (1982) 70–120.
- [5] J.A. Bergstra and J.W. Klop, Fixed point semantics in process algebras, Rept. IW 206/82, Mathematisch Centrum, Amsterdam, 1982.
- [6] R. Engelking, *General Topology* (Polish Scientific Publishers, 1977).
- [7] N. Francez, D.J. Lehmann and A. Pnueli, Linear history semantics for distributed languages, *Proc. 21st Symp. on Foundations of Computer Science* (IEEE, 1980) pp. 143–151.
- [8] R. Milner, *A Calculus for Communicating Systems*, Lecture Notes in Computer Science **92** (Springer, Berlin, 1980).
- [9] R. Milner, A complete inference system for a class of regular behaviours, *J. Comput. System Sci.* **28** (3) (1984) 439–466.
- [10] M. Nivat, Mots infinis engendrés par une grammaire algébrique, *RAIRO Informatique Théorique* **11** (1977) 311–327.
- [11] M. Nivat, Sur les ensembles des mots infinis engendrés par une grammaire algébrique, *RAIRO Informatique Théorique* **12** (1978) 259–278.
- [12] M. Nivat, Infinite words, infinite trees, infinite computations, in: J.W. de Bakker and J. van Leeuwen, eds., *Foundations of Computer Science III.2*, Mathematical Centre Tracts **109** (1979) pp. 3–52.
- [13] M. Nivat, Synchronization of concurrent processes, in: R.V. Book, ed., *Formal Language Theory* (Academic Press, New York, 1980) pp. 429–454.

- [14] G.D. Plotkin, A power domain construction, *SIAM J. Comput.* **5** (1976) 452-487.
- [15] D.S. Scott, Data types as lattices, *SIAM J. Comput.* **5** (1976) 522-587.
- [16] D.S. Scott, Domains for denotational semantics, in: M. Nielsen and E.M. Schmidt, eds., *Proc. 9th ICALP*, Lecture Notes in Computer Science **140** (Springer, Berlin, 1982) pp. 577-613.
- [17] M.B. Smyth, Power domains, *J. Comput. Systems Sci.* **16** (1978) 23-36.