LINEAR TIME AND BRANCHING TIME SEMANTICS

FOR RECURSION WITH MERGE

J.W. de Bakker
Mathematical Centre, Kruislaan 413, 1098 SJ Amsterdam
Free University, Amsterdam

J.A. Bergstra
Mathematical Centre, Amsterdam

J.W. Klop
Mathematical Centre, Amsterdam

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

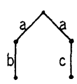
ABSTRACT

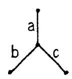
We consider two ways of assigning semantics to a class of statements built from a set of atomic actions (the 'alphabet'), by means of sequential composition, nondeterministic choice, recursion and merge (arbitrary interleaving). The first is linear time semantics (LT), stated in terms of trace theory; 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 prove 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. If the alphabet is finite, the trace set of a process is closed and trace is a continuous operation. Using trace, we then can carry over BT into LT.

1. INTRODUCTION

We study two ways of assigning meaning to a simple language L which has elementary actions (a,b,c,...), 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 parallellism (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) ∪ (a;c) and a;(b ∪ 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: 

and 

.

(Trees are not quite what we want, though. The statement $a \cup a$ should yield the object  rather than  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 ($\|$) introduces various questions. For traces, "$\|$" 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 "$\|$" operator.

Thirdly, in accordance with the emphasis which in the study of concurrency is put onto nonterminating computations, we want to include a mathematical 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 "$\|$" 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., $d(abc,abde) = 2^{-3}$, 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 . The framework for LT semantics is then taken as the complete partially ordered set of closed sets, with "$\supseteq$" (set containment) as the "$\sqsubseteq$" 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.

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. The proof of Lemma 4.4 is omitted here and can be found in [2].

2. LT SEMANTICS: MATHEMATICAL BACKGROUND AND SEMANTICAL EQUATIONS

Let A be an alphabet with elements a,b,... . (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,... be statement variables from a set *Stmv*, which we shall use in the formation of *recursive* or μ-statements. The syntax for the language L is given (in a self-explanatory BNF notation) in

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

2.1.1. EXAMPLES. $(a;b) \cup (a\|c)$, $\mu x[(a;\mu y[(b;y)\|x]) \cup c]$.

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 isn't 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 $X \to aX \mid b$.

(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 $\mu x[a\|x]$ contain unguarded occurrences of x. (In language theory, the equivalent notion is the "Greibach condition", as in Nivat [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) $\leqslant$ 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 $\leqslant$ abccb; abccb[3] = abc; abc[5] = abc; abc[0] is the empty word.)

(c) Let $x,y \in A^\infty$. The distance or *metric* $d: A^\infty \to [0,1]$ 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 write C for the collection $\mathcal{P}_c(A^\infty)$.

We define the order "$\sqsubseteq$" on C by putting $X \sqsubseteq Y$ iff $X \supseteq Y$ (with "$\supseteq$" set-containment).

2.3. LEMMA. d *is a metric on* A^∞, *and* C *is a complete partially ordered set with respect to* $\sqsubseteq$, *with* A^∞ *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 $\hat{d}$, $\tilde{d}$:
(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* $P_c(M)$.
(b) *If* (M,d) *is complete, then so is* $(P_c(M),\tilde{d})$. *Also, for* $\langle X_n\rangle_n$ *a Cauchy sequence in* $P_c(M)$, *we then have that* $\lim_n X_n = \{x \mid x_n \to x, \text{ with } x_n \in X_n\}$.

PROOF. See e.g. [6]. A complete proof of (b) is contained in [4]. □

2.6. COROLLARY. *The Hausdorff metric on* C *turns it into a complete metric space.* □

The Hausdorff metric on C will be written as d_L (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 C and its cpo structure:

2.7. PROPOSITION. *Let* $\langle X_n\rangle_n$ *be both a Cauchy sequence in* 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 $x \in \bigcap_n X_n$. □

We shall use C *with its cpo structure* as semantic domain for the trace semantics of L. (By Corollary 2.6, 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 C as model. (Technically, these two theorems are among the main results of the paper.) First we give the natural definitions of the basic operations on A^∞ and 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, $x\|y$ 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 $X\|Y = \bigcup\{x\|y \mid x \in X, y \in Y\}$. We will write also XY for $X\cdot Y$.

The main theorems of this section state that the operations $\cdot$, $\cup$, $\|$ 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* C, $X\cdot Y$, $X \cup Y$ *and* $X\|Y$ *are in* C.

PROOF. See Appendix. □

2.10. THEOREM. *Let* A *be finite. Then the operations* $\cdot$, $\cup$, $\|$ *from* $C \times C$ *to* C *are continuous in both their arguments.*

PROOF. See Appendix. □

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 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 $\Gamma = \mathcal{S}tmv \to C$, and let γ range over Γ. Let, as before, X range over 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 $[C \to C]$ stand for the collection of all continuous functions from C to C, and let, for Φ $[C \to C]$, $\mu\Phi$ denote its least fixed point. We have

2.11. DEFINITION. The semantic mapping $[\![\ ]\!]_L : \mathcal{L} \to (\Gamma \to C)$ is given by

$[\![a]\!]_L(\gamma) = \{a\}$, $[\![s_1;s_2]\!]_L(\gamma) = [\![s_1]\!]_L(\gamma) \cdot [\![s_2]\!]_L(\gamma)$

$[\![s_1 \cup s_2]\!]_L(\gamma) = [\![s_1]\!]_L(\gamma) \cup [\![s_2]\!]_L(\gamma)$, $[\![s_1 \| s_2]\!]_L(\gamma) = [\![s_1]\!]_L(\gamma) \| [\![s_2]\!]_L(\gamma)$

$[\![\mu x[s]]\!]_L(\gamma) = \mu\Phi_{s,\gamma}$ where $\Phi_{s,\gamma} = \lambda X. [\![s]\!]_L(\gamma\{X/x\})$.

This definition is justified by the following Lemma:

2.12. LEMMA. (i) $\lambda X_1 \ldots X_n. [\![s]\!]_L(\gamma\{X_i/x_i\}_{i=1}^n) \in [C \to [C \to \ldots \to [C \to C]\ldots]]$ (n factors C)
(ii) *The functions in (i) are monotonic.*

PROOF. (i) Routine (see, e.g., [1] Theorem 7.9), once Theorem 2.10 is available.
(ii) By a simple inductive proof. Or: note that C is also a complete lattice, and use the fact that in a complete lattice continuous functions are monotonic (see e.g. [1]). □

2.13. COROLLARY. $[\![\mu x[s]]\!]_L(\gamma) = \bigcap_n \Phi^n_{s,\gamma}(A^{\infty})$ *where* $\Phi_{s,\gamma}$ *is as in Definition 2.11.*

PROOF. By Definition 2.11, Lemma 2.12(i) and the Tarski-Knaster fixed point theorem. □

2.14. EXAMPLE. $[\![\mu x[(a;x) \cup b]]\!]_L(\gamma) = \mu[\lambda X. [\![(a;x) \cup b]\!]_L(\gamma\{X/x\})] = \mu[\lambda X.((a \cdot X) \cup b)] = \bigcap_n X_n$, where $X_0 = A^{\infty}$, and $X_{i+1} = (a \cdot X_i) \cup b$. Hence, $\bigcap_n X_n = a^* b \cup a^{\omega}$.

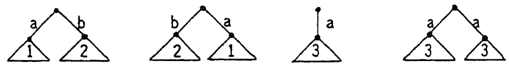
2.15. REMARK. For statements which have unguarded μ-terms, the semantics $[\![\ ]\!]_L$ may not be the most natural one. E.g. we have - for any γ - that $[\![\mu x[x]]\!]_L(\gamma) = A^{\infty}$ and $[\![\mu x[x;b]]\!]_L(\gamma) = A^{\omega}$. We shall return to this point in Section 4, where we are 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 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).

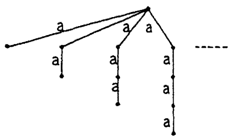
For an approach to uniform processes via projective limits, see [5]; and for an approach where processes are congruence classes of trees ('behaviours'), see Milner [8,9]. (See [2] for a comparison between the present uniform processes and Milner's behaviours.)

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

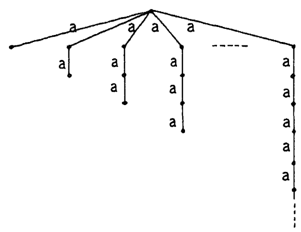


we have that the first and second, and the third and fourth represent the same process. Also, processes are closed objects: they contain all their limit points, in a sense to be made precise in a moment. E.g., the tree t_1 does not represent a process, but tree t_2 does, since it contains also the limit process "a^ω ".

t_1



t_2



Technically, processes are obtained as follows:

0. Start from alphabet A as before; moreover, a so-called <u>nil</u>-process p_0 is assumed.
1. Define P_n, $n=0,1,\ldots$, by $P_0 = \{p_0\}$, $P_{n+1} = \mathcal{P}(A \times P_n)$, where $\mathcal{P}(.)$ stands for the collection of all subsets of (.). Write $P_\omega = \bigcup_n P_n$.
2. Introduce a metric on P_n (by suitably combining Definition 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

3.1. <u>THEOREM</u>. $\mathcal{P} \cong \{p_0\} \cup \mathcal{P}_c(A \times \mathcal{P})$

where $\mathcal{P}_c(.)$ refers to the collection of all closed subsets of (.) - 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 $n \geq 0$, or $p = \lim_i p_i$, with $\langle p_i \rangle_i$ a Cauchy sequence of elements p_i in P_i.

3.2. <u>DEFINITION</u>. (a) $p \circ p_0 = p$, $p \circ X = \{p \circ x \mid x \in X\}$, $p \circ \langle a,q \rangle = \langle a, p \circ q \rangle$, $p \circ \lim_i q_i = \lim_i (p \circ q_i)$

(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) $p \| p_0 = p_0 \| p = p$, $X \| Y = \{x \| Y \mid x \in X\} \cup \{X \| y \mid y \in Y\}$,
$\langle a,p\rangle \| Y = \langle a, p\|Y\rangle$, $X \| \langle a,q\rangle = \langle a, X\|q\rangle$, $(\lim_i p_i) \| (\lim_j q_j) = \lim_k (p_k \| q_k)$.

3.3. LEMMA. *The above operations are well-defined and continuous in both arguments.*

This lemma is the counterpart of the results in the Appendix for the LT framework. For the proof - which does not require more effort than the LT case - see [4].

By way of preparation for the definition of the recursive case we need a classical result. A mapping $T:\mathcal{P} \to \mathcal{P}$ is called *contracting* whenever $d_B(T(p), T(p')) \leqslant c.d_B(p,p')$ with $0 \leqslant c < 1$. We will need Banach's fixed point theorem:

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

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 $\mathcal{P}_2 \cong \{p_0\} \cup P_c((A \cup \{\tau\}) \times \mathcal{P}_2)$. As before, we apply the familiar environment technique. Let $\Gamma = \mathcal{S}tmv \to \mathcal{P}_2$. We define the BT-semantics for $\mathcal{L}$ in

3.5. DEFINITION. The semantic mapping $[\![\ ]\!]_B : \mathcal{L} \to (\Gamma \to \mathcal{P}_2)$ is given by

$[\![a]\!]_B(\gamma) = \{\langle a,p_0\rangle\}$

$[\![s_1;s_2]\!]_B(\gamma) = [\![s_2]\!]_B(\gamma) \circ [\![s_1]\!]_B(\gamma)$

$[\![s_1 \cup s_2]\!]_B(\gamma) = [\![s_1]\!]_B(\gamma) \cup [\![s_2]\!]_B(\gamma)$

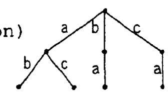
$[\![s_1 \| s_2]\!]_B(\gamma) = [\![s_1]\!]_B(\gamma) \| [\![s_2]\!]_B(\gamma)$

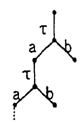
$[\![x]\!]_B(\gamma) = \gamma(x)$

$[\![\mu x[S]]\!]_B(\gamma) = \lim_i p_i$, where p_0 is the nil-process and $p_{i+1} = \{\langle\tau, [\![S]\!]_B(\gamma\{p_i/x\})\rangle\}$.

3.6. EXAMPLES. (For simplicity we omit γ.)

(1) $[\![a_1;a_2]\!]_B = \{\langle a_2,p_0\rangle\} \circ \{\langle a_1,p_0\rangle\} = \{\langle a_1, \{\langle a_2,p_0\rangle\}\rangle\}$

(2) $[\![a \| (b \cup c)]\!]_B$ = (in a natural picture representation) 

(3) $[\![\mu x[(a;x) \cup b]]\!]_B = \lim_i p_i$, where $p_{i+1} = \{\langle\tau, \{\langle a,p_i\rangle, \langle b,p_0\rangle\}\rangle\}$; = 

(4) $[\![\mu x[x]]\!]_B = [\![\mu x[x;b]]\!]_B = \{\langle\tau, \{\langle\tau, \{\langle\tau, \ldots\rangle\}\rangle\}\rangle\}$.

3.7. 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 $T = \lambda p.\{\langle\tau, [\![S]\!]_B(\gamma\{p/x\})\rangle\}$. Contractivity is enforced by the $\langle\tau,\ldots\rangle$ construct. Operationally, the $\langle\tau,\ldots\rangle$ 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. In order to establish a correspondence between LT and BT semantics, we will only consider processes whose terminating branches all terminate in p_0 and not in $\emptyset$, which according to the definition of processes is also possible. (Termination in $\emptyset$ is used in [3,4] to model failure, but in the present context this issue is not yet at stake.) That is, we adopt the natural restriction to the closure of

$$\mathcal{P}_{[\![\,]\!]} = \{ [\![S]\!]_B(\gamma) \mid S \text{ not containing free statement variables}, \gamma \in \mathit{Stmv} \to \mathcal{P} \}.$$

(Note that $\mathcal{P}_{[\![\,]\!]}$ itself is not yet 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:

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

For use in Theorem 4.7, we note that $\mathcal{P}^+ = \{ [\![S]\!]_B(\gamma) \mid \text{all } S, \gamma \in \mathit{Stmv} \to \mathcal{P}^+ \}$. .

4.1. DEFINITION. Let $p \in \mathcal{P}^+$. (1) A *path* π for p is a (finite or infinite) sequence $<a_1,p_1>, <a_2,p_2>, \ldots$ such that $<a_1,p_1> \in p$ and $<a_{i+1},p_{i+1}> \in p_i$, $i = 1,2,\ldots$

(2) (i) Let $\pi = <a_1,p_1>, <a_2,p_2>, \ldots$ be an infinite path of $p \in \mathcal{P}^+$. Then $a_1a_2\ldots \in A^\omega$ is called a *trace* of p.

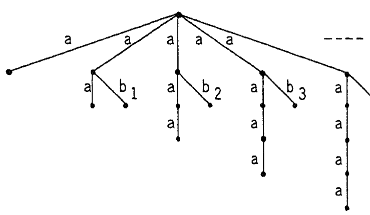
(ii) Let $\pi = <a_1,p_1>, \ldots, <a_n,p_0>$ be a finite path of $p \in \mathcal{P}^+$. Then $a_1a_2\ldots a_n \in A^*$ is a *trace* of p.

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

4.2. EXAMPLES. $\mathit{trace}(\{<a,\{<b,p_0>\}>,<a,\{<c,p_0>\}>\}) = \{ab,ac\}$,

$\mathit{trace}(\{<a,\{<a,\ldots>\}>\}) = \{a^\omega\}$, $\mathit{trace}([\![\mu x[(a;x) \cup b]]\!]_B(\gamma)) = (\tau a)^\omega \cup (\tau a)^*\tau b$.

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 $A = \{a\} \cup \{b_i \mid i \geq 1\}$:



4.3. EXAMPLE. Consider $p \in \mathcal{P}^+$ as given by the tree i.e.: $p = \{<a,p_i> \mid i \geq 0\}$ where p_0 is the nil-process and for $n > 0$: $p_n = \{<b_n,p_0>,<a,q_{n-1}>\}$,

$q_n = \{<a,<a,<a,\ldots,<a,p_0>\}>\ldots>\}$ (n times a).

Then $\mathit{trace}(p) = \{a^n | n \geq 1\} \cup \{aa_m | m \geq 1\}$, which is not closed as it lacks a^ω.

However, with the additional assumption that A is finite, we have (by a nontrivial proof which is omitted here and can be found in [2]) that $\mathit{trace}(p)$ is closed indeed. In fact we have:

4.4. LEMMA. *Let* A *be finite. Then: (i)* $\mathit{trace}(p) \in C$,
(ii) trace *is continuous (with respect to the Hausdorff metrics in* $\mathcal{P}^+$ *and* C*).* □

We will also need the following fact, whose proof is routine and omitted here:

4.5. PROPOSITION. $\mathit{trace}: \mathcal{P}^+ \rightarrow C$ *is an homomorphism (with respect to the operations* $\cdot$, $\cup$, $||$ *on* $\mathcal{P}^+$ *and* C*).* □

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

4.6. DEFINITION. The universal process for $\mathcal{P}^+$, called p_u, is the (unique) solution of the equation $p = \{<a,p> \mid a \in A\} \cup \{<a,p_0> \mid a \in A\}$.
(Note that $\mathit{trace}(p_u) = A^\infty$.)

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

(¶) $\qquad \mathit{trace}([\![S]\!]_B) = [\![S]\!]_L$.

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

$$\mathit{trace}([\![\mu x[x]]\!]_B = \mathit{trace}(\{<\tau,\{<\tau,\ldots>\}>\}) = \{\tau^\omega\} \neq A^\infty = [\![\mu x[x]]\!]_L.$$

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 3.7, which in turn was necessary to allow us to apply Banach's fixed point theorem 3.4. However, another approach may also be adopted which will lead to a positive answer to the question (¶). It is convenient to treat separately 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 $T' = \lambda p.[\![S]\!]_B(\gamma\{p/x\})$, T' is now contracting for each S, and $\lim_{i \geq 1} T'(p_i)$, 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

4.7. THEOREM. *Assume statement* S *is closed and involves only guarded* μ*-terms. Let* $[\![S]\!]_L$ *be as before, and let* $[\![S]\!]_B$ *be as in Definition 3.5, except that in the clause for* $\mu x[S]$*, we replace* p_0 *by* p_u *and define*

$$p_{i+1} = [\![S]\!]_B(\gamma\{p_i/x\}).$$

Then:

$$\mathit{trace}([\![S]\!]_B) = [\![S]\!]_L.$$

PROOF. 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 $\gamma \in \mathit{Stmv} \to \mathcal{P}^+$:

$$[\![S']\!]_L(\mathit{trace} \circ \gamma) = \mathit{trace}([\![S']\!]_B(\gamma)).$$

Case (i). $S' \not\equiv \mu x[S]$. Now the result follows easily by the induction hypothesis and and the homomorphism properties of *trace*.

The interesting case is

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

Some notation: $\mathit{trace} \circ \gamma = \gamma'$. Further, we employ again the notation of Definition 2.11: $\Phi_{S,\gamma'} = \lambda X.[\![S]\!]_L(\gamma'\{X/x\})$. Finally, p_n is defined as in the statement of the theorem.

First we prove

CLAIM 1. $\mathit{trace}(p_n) = \Phi^n_{S,\gamma'}(A^\infty)$.

Proof of Claim 1. $\mathit{trace}(p_n) = \mathit{trace}([\![S]\!]_B(\gamma\{p_{n-1}/x\}))$ = (by the induction hypothesis) $[\![S]\!]_L(\gamma'\{\mathit{trace}(p_{n-1})/x\}) = (\lambda X.[\![S]\!]_L(\gamma'\{X/x\}))(\mathit{trace}(p_{n-1})) = \Phi_{S,\gamma'}(\mathit{trace}(p_{n-1}))$.
Hence $\mathit{trace}(p_n) = \Phi^n_{S,\gamma'}(\mathit{trace}(p_u)) = \Phi_{S,\gamma'}(A^\infty)$.

CLAIM 2. $\bigcap_n \Phi^n_{S,\gamma'}(A^\infty) = \lim_n \Phi^n_{S,\gamma'}(A^\infty)$.

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* (Lemma 4.4), $\{\mathit{trace}(p_n)\}$ is therefore also a Cauchy sequence. So by Claim 1, $\{\Phi^n_{S,\gamma'}(A^\infty)\}$ is a Cauchy sequence.

Furthermore, the $\Phi^n_{S,\gamma'}$ are monotonic (Lemma 2.12(ii)). Since A^∞ is the maximal element of $\mathcal{C}$, the sequence $\{\Phi^n_{S,\gamma'}(A^\infty)\}$ is therefore decreasing (w.r.t. $\subseteq$). Now Claim 2 follows by Proposition 2.7.

Now we have: $[\![S']\!]_L(\mathit{trace} \circ \gamma) \equiv [\![\mu x[S]]\!]_L(\gamma')$ = (Coroll.2.13)

$\bigcap_n \Phi^n_{S,\gamma'}(A^\infty)$ = (Claim 2) $\lim_n \Phi^n_{S,\gamma'}(A^\infty)$ = (Claim 1)

$\lim_n \mathit{trace}(p_n)$ = (Lemma 4.4) $\mathit{trace}(\lim_n p_n)$ = (definition in the

present theorem) $\mathit{trace}[\![\mu x[S]]\!]_B(\gamma) \equiv \mathit{trace}[\![S']\!]_B(\gamma)$.

□

We continue with

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

Firstly, we can maintain the definition of $[\![S]\!]_L$, and use the revised definition of $[\![S]\!]_B$ as stated in Theorem 4.7. 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, $T'(p_u)$, $T'^2(p_u)$,... converges to a fixed point of T'. However, this fact has indeed - with some effort, and for arbitrary initial q - been established in Bergstra & Klop [5]. Thus, we can base our revised definition on their theorem, and again obtain - by the same reasoning as in the proof of Theorem 4.7 - that (¶) holds.

Secondly, we may also keep the definition of $[\![S]\!]_B$ as in Definition 3.5, and revise that of $[\![S]\!]_L$. We then replace the last clause of Definition 2.11 by

$$[\![\mu x[S]]\!]_L(\gamma) = \mu[\lambda X.\,[\![\tau;S]\!]_L(\gamma\{X/x\})].$$

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 4.7 applies again.

APPENDIX: Well-definedness and continuity of the operations $\cdot$, $\cup$, $||$ 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.

THEOREM 2.9, 2.10. (i) $X,Y\in\mathcal{C} \Longrightarrow X||Y\in\mathcal{C}$, (ii) $X,Y\in\mathcal{C} \Longrightarrow X\cdot Y\in\mathcal{C}$,

(iii) *Let* A *be finite. Let* $X_n, Y_m \in \mathcal{C}$ $(n,m\geq 0)$ *be such that* $X_0\supseteq X_1\supseteq\cdots$ *and* $Y_0\supseteq Y_1\supseteq\cdots$. *Then:*

$$\Big(\bigcap_{n\geq 0} X_n\Big)\,||\,\Big(\bigcap_{m\geq 0} Y_m\Big) = \bigcap_{k\geq 0} (X_k||Y_k).$$

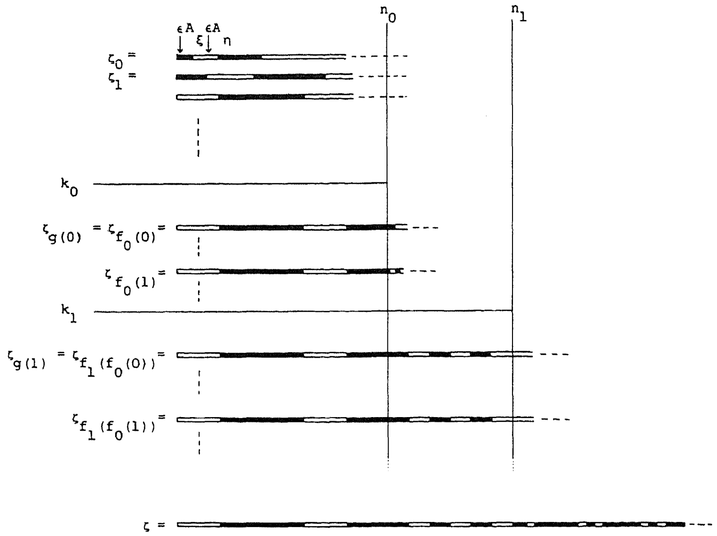
(iv) *As (iii) with* $||$ *replaced by* $\cdot$.

PROOF. The proofs of (i),...,(iv) all start with a Cauchy sequence $\{z_i\mid i\geq 0\}$, where the z_i are elements of $X||Y$, $X\cdot Y$, $\bigcap_{k\geq 0}(X_k||Y_k)$, $\bigcap_{k\geq 0} X_k\cdot Y_k$, respectively. Since we will need to specify which parts from z_i originate from X (resp. X_k) and which from Y (resp. Y_k), we introduce two disjoint copies A_ξ and A_η of the alphabet A. Intuitively, A_ξ and A_η are colored copies of A, say 'blue' resp. 'red'. The sequence $\{z_i\}$ is then colored, i.e. lifted to a sequence $\{\zeta_i\}$ where $\zeta_i\in(A_\xi\cup A_\eta)^\infty = B^\infty$ and $h(\zeta_i) = z_i$; h is the 'decoloring homomorphism' whose precise definition is left to the reader.

The sequence $\{\zeta_i\}$ is however in general no longer a Cauchy sequence in $\mathcal{P}_c(B^\infty)$. But it contains a subsequence $\{\zeta_{g(i)}\}$ which is a Cauchy sequence. The (colored) limit ξ of this subsequence is then used to prove the result.

More precisely:

Proof of (i). Let $\{z_i | i \geq 0\}$ be a Cauchy sequence such that $z_i \in X||Y$ $(i \geq 0)$. So $z_i \in x_i || y_i$ for some $x_i \in X$, $y_i \in Y$. Lifting to the alphabet B we find colored versions ζ_i, ξ_i, η_i such that $\xi_i \in A_\xi^\infty$, $\eta_i \in A_\eta^\infty$ and $\zeta_i \in \xi_i || \eta_i$.



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 to 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 $\{\zeta_i | i \geq k_0\}$ 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 $g(j) = (f_j \circ \ldots \circ f_1 \circ f_0)(0)$, which evidently is a Cauchy sequence in $\mathcal{P}_c(B^\infty)$. Call the limit ζ. Then ζ can be decomposed (by projections to A_ξ, resp. A_η) into ξ, η such that $\zeta \in \xi || \eta$. Decoloring, we have $z \in x || y$. Since z is the limit of $\{z_i\}$, we are through if $x \in X$ and $y \in Y$. This follows easily because X,Y are closed.

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

Proof of (iii). ($\subseteq$) is trivial. ($\supseteq$): take $z \in \bigcap(X_i \| Y_i)$, so for all i: $z \in x_i \| y_i$ 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 $\zeta_i \in \xi_i \| \eta_i$. Construct ζ, ξ, η such that $\zeta \in \xi \| \eta$ 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 $x \in X_1$, and so on.

The finiteness condition on A is used to ensure that $\bigcap X_n \neq \emptyset$ and $\bigcap Y_m \neq \emptyset$. The non-emptiness 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 resp. ξ such that $h(\eta) = y \in \bigcap Y_m$ resp. $h(\xi) = x \in \bigcap X_n$, to be able to write $\zeta \in \xi \| \eta$ and $z \in x \| y$.

Proof of (iv): mutatis mutandis identical to that of (iii). □

REFERENCES

[1] DE BAKKER, J.W., Mathematical Theory of Program Correctness, Prentice-Hall International, 1980.

[2] DE BAKKER, J.W., J.A. BERGSTRA, J.W. KLOP & J.-J.CH. MEYER, Linear time and branching time semantics for recursion with merge. Report IW 211/82, Mathematical Centre, Amsterdam 1982.

[3] DE BAKKER, J.W. & J.I. ZUCKER, Denotational semantics of concurrency, Proc. 14th ACM Symp. on Theory of Computing, pp.153-158, 1982.

[4] DE BAKKER, J.W. & J.I. ZUCKER, Processes and the denotational semantics of concurrency, Report IW 209/82, Mathematisch centrum, Amsterdam 1982.

[5] BERGSTRA, J.A. & J.W. KLOP, Fixed point semantics in process algebras, Report IW 206/82, Mathematisch Centrum, Amsterdam 1982.

[6] ENGELKING, R., General Topology, Polish Scientific Publishers, 1977.

[7] FRANCEZ, N., D.J. LEHMANN & A. PNUELI, Linear history semantics for distributed languages, Proc. 21st Symp. Foundations of Computer Science, IEEE 1980, pp.143-151.

[8] MILNER, R., A Calculus for Communicating Systems, Springer LNCS 92, 1980.

[9] MILNER, R., A complete inference system for a class of regular behaviours, Preprint, Edinburgh 1982.

[10] NIVAT, M., Mots infinis engendrés par une grammaire algébrique, RAIRO Informatique Théorique Vol.11 (1977) pp.311-327.

[11] NIVAT, M., Sur les ensembles des mots infinis engendrés par une grammaire algébrique, RAIRO Informatique Théorique Vol.12 (1978) pp.259-278.

[12] NIVAT, M., Infinite words, infinite trees, infinite computations, Foundations of Computer Science III.2 (J.W. de Bakker & J. van Leeuwen,eds.) pp.3-52, Mathematical Centre Tracts 109, 1979.

[13] NIVAT, M., Synchronization of concurrent processes, Formal Language Theory (R.V. Book, ed.),pp.429-454, Academic Press, 1980.

[14] PLOTKIN, G.D., A power domain construction, SIAM J. on Comp., 5 (1976), pp.452-487.

[15] SCOTT, D.S., Data types as lattices, SIAM J. on Comp., 5 (1976), pp.522-587.

[16] SCOTT, D.S., Domains for denotational semantics, Proc. 9th ICALP (M. Nielsen & E.M. Schmidt, eds.), pp.577-613, Springer LNCS 140, 1982.

[17] SMYTH, M.B., Power domains, J. Comp. Syst. sciences, 16 (1978), pp.23-36.