stichting
mathematisch
centrum

MC

AFDELING INFORMATICA
(DEPARTMENT OF COMPUTER SCIENCE)

IW 238/83 OKTOBER

J.W. DE BAKKER & J.I. ZUCKER

COMPACTNESS IN SEMANTICS FOR MERGE AND FAIR MERGE

Preprint

kruislaan 413 1098 SJ amsterdam

Printed at the Mathematical Centre, Kruislaan 413, Amsterdam, The Netherlands.

The Mathematical Centre, founded 11 February 1946, is a non-profit institution for the promotion of pure and applied mathematics and computer science. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

1980 Mathematics subject classification: 68B10, 68C05

1982 CR-Categories: D.1.3, F.1.2, F.3.2, F.3.3.

Compactness in semantics for merge and fair merge[*)]

by

J.W. de Bakker & J.I. Zucker[**)]

ABSTRACT

An analysis of the role of compactness in defining the semantics of the merge and fair merge operations is provided. In a suitable context of hyperspaces (sets of subsets) a set is compact iff it is the limit of a sequence of finite sets; hence, compactness generalises bounded nondeterminacy. The merge operation is investigated in the setting of a simple language with elementary actions, sequential composition, nondeterministic choice and recursion. Metric topology is used as a framework to assign both a linear time and a branching time semantics to this language. It is then shown that the resulting meanings are compact trace sets and compact processes, respectively. This result complements previous work by De Bakker, Bergstra, Klop & Meyer. For the fair merge, an approach using scheduling through random choices is adopted - since a direct definition precludes the use of closed, let alone of compact sets. In the indirect approach, a compactness condition is used to show that the fair merge of two fair processes yields a fair process.

KEY WORDS & PHRASES: *concurrency, merge, recursion, compactness, denotational semantics, fair merge, metric topology, Hausdorff metric, trace theory, hyperspace*

[*)] This report will be submitted for publication elsewhere.
[**)] Department of Computer Science, State University of New York, 4226 Ridge Lea Road, Amherst, NY 14226, USA.

0. INTRODUCTION

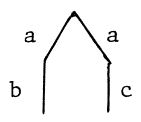
In the last few years we have seen a remarkable increase in the importance of topological tools in denotational semantics. Topology has always played a role in Scott's domain theory (for a recent example see Scott [25]; much information is contained in the comprehensive volume Gierz et al. [15]). An extension of its area of application was initiated by Nivat and his school (e.g. Arnold & Nivat [4,5], Nivat [21,22]) who use metric techniques, especially when dealing with the study of infinite words and infinite computations. Further recent evidence for our observation is provided by papers such as Arnold [3] or Smyth [26].

The present paper is devoted to two case studies concerning the role of *compactness* in semantics. We adopt the metric approach, continuing the above mentioned investigations of Nivat et al., and, furthermore, our own work as described in De Bakker & Zucker [8,9], De Bakker, Bergstra, Klop & Meyer [7], and De Bakker & Zucker [10]. More specifically, we take as starting point the latter two papers, and investigate the role of compactness in the development of linear time and branching time semantics for a language with recursion and merge, and of the definition of fair merge based on an appropriate alternation of random choices.

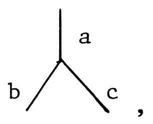
Before going into somewhat more detail about the aims and achievements of our paper, we make a few remarks on the role of metric topology and compactness in general. Classical denotational semantics - in particular when concerned with sequential programming - has relied primarily on order structures (lattices, complete partially ordered sets, etc.). As a consequence of the vigorous current interest in concurrency, new questions have arisen for which an approach solely in terms of order is not necessarily the most convenient one. Semantics of concurrency requires the preservation of intermediate stages of the computation in order to deal with phenomena such as interleaving, synchronization etc. In the simplest case they appear as *traces*, i.e. (possibly infinite) sequences of elementary actions. Two traces, e.g. abcd, abce, have no natural order relation, but a *distance* can be conveniently defined for them: We take 2^{-3}, or, in general, 2^{-n+1}, where $n \geq 1$ is the first position where the sequences differ. Distances can be defined as well for *sets* of sequences, and appropriate limit considerations can be based on well-known metric tools.

Compactness is a generalization of finiteness. In fact, it can be seen as a direct counterpart of the familiar property of *bounded nondeterminacy* in sequential denotational semantics (see, e.g., De Bakker [6]; Apt & Plotkin [2] discuss the effects of lifting the boundedness condition). More specifically, we shall develop a topological framework in which it is the case that a set is compact iff it is the limit of a sequence of finite sets. Compactness is a desirable property since it is preserved by various operations. For example, continuous mappings preserve compactness, a result which turns out to be quite fruitful below. In many situations, compactness is a direct consequence of the *finiteness* of the alphabet of elementary actions which underlies the model at hand. Our paper does not impose this finiteness condition; more effortis then needed to obtain certain compactness results.

In the first part of the paper, we are concerned with a simple language L which features, besides elementary actions $\underline{a},\underline{b},\underline{c},\ldots$, fundamental concepts such as sequential composition $(S_1;S_2)$, recursion, nondeterministic choice $(S_1 \cup S_2)$, and merge $(S_1 \| S_2$, denoting arbitrary interleaving of the elementary actions of S_1 and $S_2)$. In [7] we have provided a detailed semantics for this language based on a combination of cpo- and metric techniques. We distinguish the so-called "linear time" (LT) and "branching time" (BT) semantics for L, adopting a terminology inspired by the model theory of temporal logic. The crucial difference between LT and BT is illustrated by the difference between the treatment of the two programs $(\underline{a};\underline{b}) \cup (\underline{a};\underline{b})$ and $\underline{a};\ (\underline{b} \cup \underline{c})$. In LT, both have as meaning the trace set $\{ab,ac\}$. In BT we obtain, respectively, the trees



and



,

thus preserving the difference in the moment of choice between the two cases. Technically, in BT we do not require left-distributivity of ";", over "$\cup$". Moreover, as a consequence of our wish to impose commutativity and idempotence of choice $(S_1 \cup S_2 = S_2 \cup S_1$, and $S \cup S = S)$ as a feature of our model, we cannot, in general, use trees. Instead, we need another notion, viz. that of *process* (first described in a metric setting in [8,9]). A process is like an unordered tree, but without repetitions in its successor sets. Also, processes are closed objects: they contain all their limit points, in a sense to be made precise below. In [7] we have used a cpo structure on trace sets for LT, and (closed) processes for BT. What we shall present below is a metric approach for both cases, based on compact trace sets for the first, and on compact processes for the second case. Besides a certain uniformity obtained in using the metric approach throughout, we also circumvent the restriction to a finite alphabet which was imposed at certain essential points in the development in [7] (in particular in theorem 2.10 of that paper). The results of part 1 can be summarized as follows: For each $S \in L$, its LT semantics $[\![S]\!]_L$ is a compact set, and its BT semantics $[\![S]\!]_B$ is a compact process. Moreover, there exists a continuous mapping *trace* which maps $[\![S]\!]_B$ to $[\![S]\!]_L$. An important technical role is played by a theorem of Michael [19] which can be paraphrased (in the context of hyperspaces) as "a compact union of compact sets is compact".

The second half of the paper is devoted to an analysis of *fair merge*. Consider, e.g., two sequences 0^ω and 1^ω (a^ω denotes an infinite sequence of a's). Their fair merge *excludes* all sequences $(0 \cup 1)^*(0^\omega \cup 1^\omega)$, i.e., sequences with, eventually, only zero's or ones. Hence, the resulting set cannot be closed (since it does not contain all limits of sequences of finite approximations), let alone compact. Thus, a *direct* approach based on compactness does not work. In [10] we have instead proposed an indirect approach based on "implementing" fairness through suitable alternation of random choices (continuing an idea of Plotkin [23]; random assignement is also used extensively in Apt & Olderog [1]). What we shall do below is to present a *proof* - not provided in [10] - that the fair merge of two fair processes (defined as in [10]) is again fair. In the argument an essential role is played, once more, by a compactness property of the processes involved.

The organisation of the paper is as follows: You are now reading section 0 which gives the introduction. In section 1 we briefly describe some of the topological notions and results which are necessary for the development below. Section 2 presents the announced result for LT semantics, and section 3 for BT semantics. Section 4, finally, is devoted to the fair merge.

Besides the already mentioned literature, we would like to refer to the papers by Golson and Rounds [16], and Rounds [24], which are also concerned with the use of metric topology in general, and the role of compactness in particular, in the semantics of concurrency. Processes in general have been studied extensively by Milner, e.g. [20]; the *algebra* of processes is pursued by Bergstra & Klop, e.g. [11,12].

1. TOPOLOGICAL PRELIMINARIES

We assume known the notions of metric space, Cauchy sequence (CS) in a metric space, isometry (distance-preserving bijection), limits and closed sets, completeness of a metric space, and the theorem stating that each metric space (M,d) can be completed to (i.e., isometrically embedded in) a complete metric space. Throughout our paper, we shall only consider distances d with values in the interval [0,1]. Explicit mentioning of the metric d is often omitted.

We first present the standard definitions of *continuous* and *contracting* functions:

DEFINITION 1.1.

a. Let M_1, M_2 be two metric spaces. A function $\phi: M_1 \rightarrow M_2$ is called continouos whenever, for each CS $<x_i>_{i=0}^{\infty}$ in M_1, we have that $<\phi(x_i)>_{i=0}^{\infty}$ is a CS in M_2, and $\phi(\lim_i x_i) = \lim_i \phi(x_i)$.
b. Let $\phi: M \rightarrow M$. We call ϕ contracting whenever, for each $x,y \in M$, $d(\phi(x),\phi(y)) \leq c*d(x,y)$, for some constant c with $0 \leq c < 1$.

A well-known classical result is Banach's fixed point theorem:

THEOREM 1.2. Let $\phi: M \rightarrow M$ be contracting. Then ϕ has a unique fixed point x satisfying $x = \lim_i \phi^i(x_0)$, for any $x_0 \in M$. □

Let (M,d) be a complete metric space. (It simplifies matters to assume completeness from now on; certain definitions or claims made below would, in fact, remain valid without this requirement.) For $X,Y \subseteq M$ we can define the so-called Hausdorff distance $\tilde{d}(X,Y)$:

DEFINITION 1.3. Let $x,y \in M$ and $X,Y \subseteq M$.
a. $\hat{d}(x,Y) = \inf_{y\in Y} d(x,y)$
b. $\tilde{d}(X,Y) = \max(\sup_{x\in X} \hat{d}(x,Y), \sup_{y\in Y} \hat{d}(y,X))$
(By convention, $\inf \emptyset = 1$, $\sup \emptyset = 0$.)

We have

LEMMA 1.4. Let $\mathcal{P}_c(M)$ be the collection of all *closed* subsets of M. Then $(\mathcal{P}_c(M),\tilde{d})$ is a metric space. Moreover, (if M is complete) $(\mathcal{P}_c(M),\tilde{d})$ is complete, and, for $\langle X_i\rangle_i$ a CS in $\mathcal{P}_c(M)$, we have that

$$\lim_i X_i = \{x \mid x = \lim_i x_i,\ x_i \in X_i,\ \langle x_i\rangle_i \text{ a CS in } M\}.$$

Proof. For the first part see, e.g., Engelking [14]. The second statement is due to Hahn [17].

Next, we introduce the important notion of *compactness*. Also, the definition of a set being *totally bounded* is given.

DEFINITION 1.5.
a. A subset X of M is called compact if each open cover of X has a finite subcover.
b. Let, for each $\varepsilon > 0$ and $x \in M$, $N_\varepsilon(x) \overset{\text{df.}}{=} \{y \mid d(x,y) < \varepsilon\}$. A subset X of M is called totally bounded if, for all $\varepsilon > 0$, there exists a finite set $F \subseteq M$ such that

$$X \subseteq \bigcup_{x\in F} N_\varepsilon(x).$$

The following theorem characterizes compactness in a number of different ways:

THEOREM 1.6. For any $X \subseteq M$, the following are equivalent:
a. X is compact
b. X is closed and totally bounded
c. X is the limit (in the Hausdorff metric) of a sequence of *finite* sets.

Proof. Standard topology (see [13] or [14]). □

The following properties of compact sets are important in the sequel:

THEOREM 1.7.
a. Each closed subset of a compact set is compact
b. The continuous image of a compact set is compact. I.e., if $\phi: M \to N$ is continuous, $X \subseteq M$ is compact, and $\hat{\phi}(X) \overset{\text{df.}}{=} \{\phi(x) \mid x \in X\}$, then $\hat{\phi}(X)$ is compact.
c. If $X \subseteq M$, $Y \subseteq N$, X and Y compact, then $X \times Y$ is compact in the product topology for $M \times N$.
d. If $\langle X_i\rangle_i$ is a CS of compact sets in M, and $X = \lim_i X_i$, then X is compact.
Proof. a,b,c. Standard.
d. For each i there is a CS of finite sets $\langle Y_{i,j}\rangle_j$ such that $X_i = \lim_j Y_{i,j}$. Then X is the limit of the diagonal sequence $\langle Y_{i,i}\rangle_i$, hence, X is compact. □

The next property of compact sets may be somewhat less well-known. It is due to Michael ([19]). Let, for (M,d) a complete metric space, $(\mathcal{P}_{comp}(M),\tilde{d})$ be the space of compact subsets of M, equipped with the Hausdorff metric. (By theorem 1.7c we know that $(\mathcal{P}_{comp}(M),\tilde{d})$ is complete.) We have

THEOREM 1.8. Let $X_i, i \in I$, be compact subsets of M, and let $\{X_i \mid i \in I\}$ be compact in $(\mathcal{P}_{comp}(M),\tilde{d})$. Then $X \overset{df.}{=} \bigcup \{X_i \mid i \in I\}$ is compact in (M,d).

Proof. See [19]. $\square$

We now specialize our considerations to spaces of sequences and sets of sequences. Let A be a, *possibly infinite,* alphabet with elements $a,b,c,\ldots$. Let A^* be the set of all finite sequences over A, let A^ω be the set of all infinite sequences over A, and let $A^\infty \overset{df.}{=} A^* \cup A^\omega$. Let $x,y,\ldots$ denote elements of A^∞. The prefix of $x \in A^\infty$ of length n will be denoted by $x[n]$ (with the convention that, e.g., $abc[5] = abc$; $abc[0]$ is the empty word). The distance $d(x,y)$ for $x,y \in A^\infty$ is defined by

$$\tilde{d}(x,y) = 2^{-\max\{n \mid x[n] = y[n]\}}$$

(with the convention that $2^{-\infty} = 0$).
Let $\mathcal{P}_c(A^\infty)$ denote the class of all closed subsets of A^∞. The distance d on A^∞ can be extended to the Hausdorff distance $\tilde{d}$ on $\mathcal{P}_c(A^\infty)$ in the manner described above. Alternatively, we might define

$$d(X,Y) = 2^{-\max\{n \mid X[n] = Y[n]\}},$$

where $X[n] = \{x[n] \mid x \in X\}$. We omit the straightforward proof that the two definitions of $\tilde{d}$ are equivalent.

It is known (e.g. Nivat [21]) that, in case A is finite, A^∞ and (hence) all its closed subsets are compact. This is no longer the case for an infinite alphabet A. Only in certain situations - of which we treat the case that we are concerned with subsets of A^∞ which are meanings of statements in some suitable language - can we again establish compactness. The next two sections are devoted to an exposition of this, and a similar, result.

2. LINEAR TIME SEMANTICS

We study the semantics of a simple language L, which features some standard sequential concepts (sequential composition, recursion) together with nondeterministic choice and merge (arbitrary interleaving). In the present section we present the LT semantics for L, in which we do not distinguish between, e.g., the meanings of $\underline{a};\underline{b} \cup \underline{a};\underline{c}$ and $\underline{a};(\underline{b}\cup\underline{c})$. In section 3 we shall deal with its BT semantics. Let $\underline{a},\underline{b},\ldots$ be elements of a, possibly infinite, set $\underline{A}$ of elementary actions. We assume that for each (syntactic) $\underline{a}$ we have a corresponding (semantic) a in the alphabet A. Let $\underline{x},\underline{y},\ldots$ be elements of a set of *statement variables* *Stmv*. The variables $\underline{x},\underline{y}$ will be used in the formation of *recursive* or μ-statements. The syntax for *L* is given, in a self-explanatory BNF notation, in

DEFINITION 2.1.

$$S ::= \underline{a} \mid \underline{x} \mid S_1;S_2 \mid S_1 \cup S_2 \mid S_1 \| S_2 \mid \mu\underline{x}[S].$$

Here $\underline{x}$ is required to occur only *guarded* (see remark below) in S.

Examples: $(\underline{a}\|\underline{b}) \cup (\underline{a}\|\underline{c})$, $\mu\underline{x}[(\underline{a};\underline{x}) \cup \underline{b}]$, $\mu\underline{x}[\underline{a};\mu\underline{y}[(\underline{b};\underline{y})\|\underline{x}] \cup \underline{c}]$.

Remarks.
1. Syntactic ambiguities should be remedied by using parentheses or conventions for the priorities of the operations.
2. (For the reader who is not familiar with the μ-notation) A term such as $\mu\underline{x}[(\underline{a};\underline{x}) \cup \underline{b}]$ has the same intended meaning as a *call* of the procedure declared (in an ALGOL-like language) by $P \Leftarrow (\underline{a};P) \cup \underline{b}$, or, alternatively, generates the same language of finite *and infinite* words as the grammar $X \to \underline{a}X \mid \underline{b}$.
3. In a term $\mu\underline{x}[S]$, occurrences of $\underline{x}$ in S may be "guarded", i.e. of the form $\ldots\underline{a};(\ldots\underline{x}\ldots)\ldots$, for some $\underline{a} \in \underline{A}$. We shall consider only terms $\mu\underline{x}[S]$ in which all occurrences of $\underline{x}$ are guarded. In [7] we have dealt, in a cpo setting, with the consequences of dropping this restriction. In language theory, the equivalent notion is the "Greibach condition", see e.g. [21].

We proceed with the development of the LT semantics for *L*. In this, we use a metric (rather than the cpo framework of [7]). Let, for brevity, *C* stand for $P_{comp}(A^\infty)$. We shall assign meaning to statements $S \in L$ as elements of *C*. Due to the presence of recursion, we employ an *environment* component in the defining equations, which serves to assign meaning to the free statement variables in S. Let $\Gamma \overset{df.}{=} Stmv \to C$, and let γ range over Γ.

We first discuss the definitions of the basic operations on $X,Y \subseteq A^\infty$. We use the obvious fact that, for each $x \in A^\infty$, $x = \lim_i x[i]$. We assume known the definitions of $x.y$ and $x\|y$ for $x,y \in A^*$ (see, e.g., [18]). We give

DEFINITION 2.2. Let $x,y \in A^\infty$, $X,Y \subseteq A^\infty$.

a. $x.y = \lim_i (x[i].y[i])$

   $x\|y = \lim_i (x[i]\|y[i])$

b. $X.Y = \{x.y \mid x \in X, y \in X\}$

   $X \cup Y$ is the set-theoretic union of X and Y

   $X\|Y = \bigcup \{x\|y \mid x \in X, y \in Y\}$.

Remark. Direct definitions - which avoid the use of CS and limits - for x.y and $x \| y$ with $x,y \in A^\omega$ are also possible. We omit the proof that definition 2.2a yields equivalent results.

We have the following lemma:

LEMMA 2.3.
a. The sequences $\langle x[i].y[i]\rangle_i$ and $\langle x[i] \| y[i]\rangle_i$ are CS of finite sets.
b. x.y and $x \| y$ are compact.

Proof. a. Left to the reader (who might consult Appendix B of [9] for very similar results).
b. For "." this is trivial; for $x \| y$ it follows from part a and theorem 1.7d. □

The three operations $.,\cup,\|$ are continuous:

LEMMA 2.4. The operations $.,\cup,\|$ are continuous mappings: $A^\infty \times A^\infty \to \mathcal{P}_{\text{comp}}(A^\infty)$.

Proof. Omitted (cf. for techniques of [9], Appendix B). □

We can now prove the central theorem of this section:

THEOREM 2.5. For X,Y compact subsets of A^∞, X.Y, $X\cup Y$ and $X \| Y$ are compact subsets of A^∞.

Proof. For "$\cup$" this is trivial. The proof for "." is simpler than that for "$\|$", which we now give. Let X,Y be compact subsets of A^∞. By theorem 1.7c, $X \times Y$ is compact in $A^\infty \times A^\infty$. By the continuity of "$\|$" as mapping: $A^\infty \times A^\infty \to \mathcal{P}_{\text{comp}}(A^\infty)$ (lemma 2.4) and theorem 1.7b, we have that $\hat{\|}(X,Y) \overset{\text{df.}}{=} \{x\|y \mid x \in X,\ y \in X\}$ is a compact *subset* of $\mathcal{P}_{\text{comp}}(A^\infty)$. Thus, $\hat{\|}(X,Y)$ is a compact subset of $\mathcal{P}_{\text{comp}}(A^\infty)$ consisting of "points" which are compact subsets of A^∞. We can therefore apply Michael's theorem and obtain that $\|(X,Y) = \bigcup \{x\|y \mid x \in X,\ y \in Y\}$ is a compact subset of A^∞. □

We are now sufficiently prepared for the main definition of this section:

DEFINITION 2.6. LT semantics for L.
The mapping $[\![\ ]\!]_L : L \to (\Gamma \to C)$ is defined by
$[\![\underline{a}]\!]_L(\gamma) = \{a\}$, $[\![\underline{x}]\!](\gamma) = \gamma(x)$,
$[\![S_1;S_2]\!]_L(\gamma) = [\![S_1]\!]_L(\gamma).[\![S_2]\!]_L(\gamma)$
$[\![S_1 \cup S_2]\!]_L(\gamma) = [\![S_1]\!]_L(\gamma) \cup [\![S_2]\!]_L(\gamma)$
$[\![S_1 \| S_2]\!]_L(\gamma) = [\![S_1]\!]_L(\gamma) \| [\![S_2]\!]_L(\gamma)$
$[\![\mu\underline{x}[S]]\!]_L(\gamma) = \lim_i X_i$, where X_0 is arbitrary, and $X_{i+1} = [\![S]\!]_L(\gamma\{X_i/\underline{x}\})$

(In the last formula, $\gamma\{X_i/\underline{x}\}$ denotes an environment which is like γ, but for its value in $\underline{x}$ which is set to X_i.)

We verify that this definition assigns a compact set as meaning to each $S \in L$:

THEOREM 2.7. For each $S \in L$ $[\![S]\!]_L(\gamma) \in \mathcal{C}$.

Proof. Induction on the structure of S. If $S \equiv \underline{a}$ or $S \equiv \underline{x}$, the result is clear. If $S \equiv S_1;S_2$, $S \equiv S_1 \cup S_2$ or $S \equiv S_1 \| S_2$, we use theorem 2.5. If $S \equiv \mu\underline{x}[S_1]$ we use the fact - the easy proof of which we leave to the reader - that, for $\underline{x}$ guarded in S_1, $\lambda X.[\![S_1]\!]_L(\gamma\{X/\underline{x}\})$ is a contracting mapping: $\mathcal{C} \to \mathcal{C}$. From this we obtain that $\langle X_i \rangle_i$ is a CS; an appeal to theorem 1.7d then yields the described result. $\square$

Thus, we have shown compactness of $[\![S]\!]_L(\gamma)$ independent of the finiteness of A. We also observe that, by Banach's theorem, for guarded $\mu\underline{x}[S]$ we have that its meaning equals the (unique) fixed point of $\lambda X.[\![S]\!](\gamma\{X/\underline{x}\})$, in accordance with the intended meaning of the recursive construct $\mu\underline{x}[S]$.

3. BRANCHING TIME SEMANTICS

We follow [7,8,9] in the design of a branching time semantic framework for the language L as introduced in section 2, with the replacement of "closed" by "compact" at certain crucial points as major difference.

Let A be any (finite or infinite) alphabet. Let p_0 denote the so-called <u>nil</u>-process - the role of which will become clear as we go along. The first definition introduces sets of finite *processes* over A, and associated metrics on these sets:

DEFINITION 3.1. For n = 0,1,... we define sets P_n and metrics d_n on P_n:

a. $P_0 = \{p_0\}$, $P_{n+1} = \{p_0\} \cup \mathcal{P}_{\text{finite}}(A \times P_n)$

b. $d_0(p,q) = 0$; d_{n+1} is defined as follows: let $p,q \in P_{n+1}$.
Either

(i) $p = q = p_0$. Then $d_{n+1}(p,q) = 0$

(ii) $p = p_0$, $q \neq p_0$ or vice versa. Then $d_{n+1}(p,q) = 1$.

(iii) $p = X \subseteq A \times P_n$, $q = Y \subseteq A \times P_n$. Then $d_{n+1}(p,q) = \tilde{d}_{n+1}(X,Y)$, where $\tilde{d}_{n+1}$ is the Hausdorff metric (definition 1.3) induced by the distance $\bar{d}_{n+1}$ between "points" $\langle a',p'\rangle$, $\langle a'',q'\rangle$ defined by

$$\bar{d}_{n+1}(\langle a',p'\rangle,\langle a'',q'\rangle) = 1, \text{ if } a' \neq a''$$
$$= \tfrac{1}{2}d_n(p',q'), \text{ if } a' = a''.$$

We now consider the set $P_\omega \overset{\text{df.}}{=} \bigcup_n P_n$ of all finite processes, together with the metric $d = \bigcup_n d_n$ (with the natural definition of $\bigcup_n d_n$). (P_ω,d) is a metric space which can be *completed* to a complete metric space, say (P,d). We can show that

THEOREM 3.2.

$$P = \{p_0\} \cup \mathcal{P}_{\text{comp}}(A \times P).$$

Proof. This is as in [9], but for the modification that

(i) We use compact sets instead of closed sets throughout

(ii) We use the theorem that a CS of compact sets has a compact limit (rather than Hahn's theorem that a CS of closed sets has a closed limit, which was fundamental in [9]). $\square$

Next, we define the three fundamental operations $\circ$, $\cup$, $\|$ for processes. Apart from a cosmetic change in the definition of "$\circ$", these definitions are as in [7,8,9]. Throughout, we distinguish the finite case ($p,q \in P_\omega$) and infinite case ($p \in P\backslash P_\omega$ or $q \in P\backslash P_\omega$). Moreover, we (implicitly) use induction on the *degree* of the processes concerned, where, for $p \in P_\omega$, *degree* (p) is given by: *degree* $(p_0) = 0$, and, for $p \neq p_0$, *degree* $(p) = n$ iff $p \in P_n\backslash P_{n-1}$.

DEFINITION 3.3. Let X,Y range over sets of finite processes.

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

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. $p_0 \| p = p \| p_0 = p$, and, for $p,q \neq p_0$, we put

$X \| Y = \{x \| Y \mid x \in X\} \cup \{X \| y \mid y \in Y\}$,

$\langle a,q\rangle \| Y = \langle a, q \| Y\rangle$, $X \| \langle a,p\rangle = \langle a, X \| p\rangle$

$(\lim_i p_i) \| (\lim_j q_j) = \lim_i (p_i \| q_i)$.

Using a combination of the techniques of Appendix B of [9] and the compactness properties of section 1 (but note that we do not need Michael's theorem here), we can justify the above definitions, and prove that the three operations $\circ, \cup, \|$ are continuous. It is now straightforward to define the branching time semantics for L. Let P be as in theorem 3.2., and let $\Gamma = Stmv \to P$.

DEFINITION 3.4. The valuation $[\![.]\!]_B : L \to (\Gamma \to P)$ is given by

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

$[\![S_1;S_2]\!]_B(\gamma) = [\![S_1]\!]_B(\gamma) \circ [\![S_2]\!]_B(\gamma)$, and similarly for $\cup, \|$

$[\![\mu\underline{x}[S]]\!]_B(\gamma) = \lim_i p_i$, where p_1 is arbitrary, and $p_{i+1} = [\![S]\!]_B(\gamma\{p_i/\underline{x}\})$.

The proof that, for each $S \in L$, $[\![S]\!]_B(\gamma) \in P$ can now be given exactly as that of theorem 2.7.

Finally, consider the mapping *trace* studied in [7]. We define *trace*: $P \to \mathcal{C}$ by

DEFINITION 3.5.

trace $(p_0) = \{\varepsilon\}$, *trace* $(X) = \cup \{$*trace* $(x) \mid x \in X\}$

trace $(\langle a,p\rangle) = a.$ *trace* (p),

trace $(\lim_i p_i) = \lim_i ($*trace* $(p_i))$.

In [7] it was shown that, for each $S \in L$ without free statement variables and each γ, (*): *trace* $([\![S]\!]_B(\gamma)) = [\![S]\!]_L(\gamma)$, provided the underlying alphabet is finite (and using the observation that $[\![S]\!]_B(\gamma) \neq \emptyset$ for each such S). Inspection of the proof of (*)

shows that it can be taken over, but with an appeal to the finiteness of A (used in [7] to establish that *trace* (p) is a closed set) replaced by suitable use of the above compactness results. Details are omitted.

4. FAIR MERGE

This section is devoted to a study of fair merge of processes. A similar analysis can be made for the fair merge of trace sets; we leave this to the interested reader. As remarked above, a *direct* definition of fair merge in terms of closed (let alone compact) sets seems not possible. Therefore, we use an indirect approach in which we model fair merge in terms of a scheduling mechanism which employs a sequence of random choices determining successively the (finite) number of times the left- and right operand of the fair merge operation should be chosen. (See also [10] for further explanation of this idea.) The indirect definition uses an *extended* domain of processes, viz. P solving the equation

$$P = \{p_0\} \cup \mathcal{P}_{\text{closed}}((A \cup \mathbb{N}) \times P)$$

where $\mathbb{N}$ is the set of natural numbers. Note that we have $\mathcal{P}_{\text{closed}}(.)$ rather than $\mathcal{P}_{\text{compact}}(.)$ in this equation. We shall use $B \overset{\text{df.}}{=} A \cup \mathbb{N}$, and b to range over B. Our first definition introduces some terminology:

DEFINITION 4.1.

a. $\Sigma_k\, p_k \overset{\text{df.}}{=} \{<k,p_k> \mid k \in \mathbb{N}\}$
A process $\Sigma_k\, p_k$ is called a "sum process".

b. A *basic* process is one of the form $\{<a_i,p_i>\}_{i\in I}$

c. A *path* for a process p is a (finite or infinite) sequence (*): $<b_1,p_1>, <b_2,p_2>,\ldots$, such that $<b_1,p_1> \in p$, and $<b_{i+1},p_{i+1}> \in p_i$, $i = 1,2,\ldots$. We say that path (*) *passes through* p_i, $i = 1,2,\ldots$.

d. An "action" b is called "enabled" in a path (*) whenever, for some i and q, $<b,q> \in p_i$. b *occurs* is (*) whenever, for some i, $b = b_i$.

c. A path (*) is called *fair* whenever, for all $a \in A$, if a is infinitely often enabled in (*), it infinitely often occurs in (*). A process is called fair whenever all its paths are fair. (Only actions in A are taken into account in the definition of fairness.)

f. Process q is a node of p - also called a subprocess of p - if there is a path from p which passes through q.

g. We call a process p *normal* if each node of p is either a basic node, or a sum node, or p_0.

h. p is called *pure* (or hereditarily basic) if each node of p is a basic node or p_0.

i. p is called hereditarily compact if each basic node of p is compact (as a subset of P).

Remarks.

1. Note that, in clause i, we impose the compactness requirement only for basic nodes.
2. The theory below will be developed for hereditarily compact, normal processes (HCN processes, for short).
3. The set HCN is closed in P.
4. If A is finite then each pure process is heriditarily compact, and hence in HCN.

Fair merge will be defined for all normal processes, and it takes normal processes to normal processes. But to show that fairness is preserved by fair merge, we must also assume hereditary compactness. That is, as we will show, fair merge takes fair HCN processes to fair HCN processes.

DEFINITION 4.2 (fair merge). For p,q finite we define, by induction on *degree* (p) + *degree* (q), their fair merge $p\|_f q$, using a number of auxiliary operations $p\|_x q$, for $x = f;L;R;L,k;R,k$.

a. $p\|_x p_0 = p_0\|_x p = p$.
   Otherwise, assume $p,q \neq p_0$.
b. $p\|_f q = \{<2k, p\|_{L,k}q>\}_{k\in\mathbb{N}} \cup \{<2k+1, p\|_{R,k} q>\}_{k\in\mathbb{N}}$.
c. $p\|_{L,k+1} q = \{<b, p'\|_{L,k} q> \mid <b,p'> \in p\}$.
d. $p\|_{L,0} q = \{<b, p'\|_R q> \mid <b,p'> \in p\}$.
e. $p\|_R q = \Sigma_k (p\|_{R,k} q)$,

and the symmetric cases for c,d and e.

Remark. In order to extend the definitions to infinite processes, we must show that these operations are continuous.

LEMMA 4.3. $d(\Sigma_k p_k, \Sigma_k q_k) = \frac{1}{2}*\sup d(p_k,q_k)$.

Proof. Clear. □

LEMMA 4.4. For finite p,p',q,

$$d(p\|_x q, p'\|_x q) \leq d(p,p').$$

Proof. This is proved simultaneously for all x, by induction on $n \overset{\text{df.}}{=} \max(\textit{degree}\ (p), \textit{degree}\ (p')) + \textit{degree}\ (q)$. For fixed n, prove for x in the following order: L,0; L,k+1; L; R,0; R,k+1; R; f, and use the results of [9], Appendix B. □

LEMMA 4.5. For finite p,p',q,q'

$$d(p\|_x q, p'\|_x q') \leq \max(d(p,p'),d(q,q')).$$

Proof. This follows from lemma 4.4 together with the symmetric result, and the strong triangle inequality (since d is in fact an ultrametric). □

Now we are justified in defining:

DEFINITION 4.6. For p,q infinite, $p = \lim_i p_i$, $q = \lim_j q_j$, p_i and q_j finite, we put

$$p \|_x q = \lim_i (p_i \|_x q_i).$$

The proofs of the following lemma's are now direct (and omitted):

LEMMA 4.7. The statement of lemma 4.5 also holds for infinite processes. Hence, the operations "$\|_x$" are jointly continuous in both arguments.

LEMMA 4.8. The fair merge takes normal processes to normal processes.

LEMMA 4.9. Clauses a,b and e in the definition of fair merge (definition 4.2) hold also for infinite processes, but clauses c,d are changed (for infinite processes) to

c'. $p\|_{L,k+1}q = Cl\{\ldots\}$

d'. $p\|_{L,0}q = Cl\{\ldots\}$

i.e. the closures of the sets on the right-hand side above.

LEMMA 4.10. If p,q are in HCN then

a. all the clauses in the definition of fair merge hold for p and q

b. $p\|_f q$ is in HCN.

Proof. a. Consider clause c or d of definition 4.2. Let X be the set on the right-hand side of the definition. If p is a sum node, then it is clear that X is closed, since any two points in it are a distance 1 apart. If p is a basic node then it is compact, hence X, being a continuous image of p, is compact and hence closed.

b. Clear. □

The reader should observe the essential role played by the compactness requirement in the proof of part a.

We can now prove the main theorem of this section:

THEOREM 4.11. Let p,q be fair HCN processes. Then $p\|_f q$ is fair.

Proof. Let $path_0$ be any infinite path in $p\|_f q$. By lemma 4.10 it can be seen that $path_0$ can be (uniquely) represented as the "fair shuffle" of two paths $path_1$ in p and $path_2$ in q. Fairness for $path_0$ now follows by a simple argument: For suppose that a basic action a is enabled infinitely often in $path_0$: Then clearly a is enabled infinitely often in $path_1$ or $path_2$. Suppose without lack of generality a is enabled infinitely often in $path_1$. Because p is fair, a occurs infinitely often in $path_1$, and hence in $path_0$. □

Remarks.

1. Note the role of the closedness property in the first claim of this proof: Without the closedness of the X on the right-hand size of clauses c,d (see proof of lemma 4.10) we would have to take into account the possibility that nodes added by the closure

operation are involved in the formation of $path_0$ rather than $path_0$ being the (direct) fair shuffle of paths in p and q.

2. The reader may wonder whether the above argument would also work with the ordinary merge as defined in section 3, in which case we would not have to concern ourselves with sum nodes at all. In fact, it would not work. The reason is, roughly, as follows. Let p and q be two heriditarily compact pure processes, and let $path_0$ be a path in $p \| q$. Again, by lemma 4.10, $path_0$ can be uniquely represented as a shuffle of two paths, $path_1$ in p and $path_2$ in q. Suppose that a is infinitely often enabled in $path_0$ and suppose, for definiteness, that a occurs only in nodes of p (not of q). It does not follow that a is enabled infinitely often in $path_1$. This is because $path_0$ may be an "unfair" shuffle of $path_1$ and $path_2$ and, from a certain point onwards, may involve $path_2$ only, in which case a, although infinitely often enabled in $path_0$, never gets a chance to be enabled in $path_1$ again past this point.

3. Certain HCN processes are clearly "degenerate" from our point of view, namely those containing infinite paths which, from some point on, contain only sum nodes. We could exclude such processes explicitly from consideration, since we are only really concerned with processes which arise from finitely many applications of the fair merge operation to pure heriditarily compact processes. However the set of these processes is not closed in P. (Whether this is an important consideration is not so clear.)

4. A topic for further research is the combination of the techniques of this section with those of section 3, allowing the definition of the semantics of *L* extended with the fair merge operation.

REFERENCES

[1] APT, K.R. & E.R. OLDEROG, *Proof rules dealing with fairness*, Proc. Logic of Programs 1981 (D. Kozen, ed.), 1-9, LNCS 131, Springer, 1982.

[2] APT, K.R. & G.D. PLOTKIN, *A Cook's tour of countable nondeterminism*, Proc. 8th ICALP (S. Even & O. Kariv, eds), 479-494, LNCS 115, Springer, 1981.

[3] ARNOLD, A., *Topological characterizations of infinite behaviours of transition systems*, Proc. 10th ICALP (J. Diaz, ed.), 28-38, LNCS 154, Springer, 1983.

[4] ARNOLD, A. & M. NIVAT, *Metric interpretations of infinite trees and semantics of nondeterministic recursive programs*, Theoretical Computer Science 11, 181-206, 1980.

[5] ARNOLD, A. M. NIVAT, *The metric space of infinite trees, algebraic and topological properties*, Fund. Inform. III, 4, 445-476, 1980.

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

[7] DE BAKKER, J.W., J.A. BERGSTRA, J.W. KLOP & J.-J.Ch. MEYER, *Linear time and branching time semantics for recursion with merge*, Proc. 10th ICALP (J. Diaz, ed.), 39-51, LNCS 154, Springer, 1983.

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

[9] DE BAKKER, J.W. & J.I. ZUCKER, *Processes and the denotational semantics of concurrency*, Information and Control, 54, 70-120, 1982.

[10] DE BAKKER, J.W. & J.I. ZUCKER, *Processes and a fair semantics for the ADA rendez-vous*, Proc. 10th ICALP (J. Diaz, ed.), 52-66, LNCS 154, Springer, 1983.

[11] BERGSTRA, J.A. & J.W. KLOP, *Process algebra for communication and mutual exclusion*, Department of Computer Science Technical Report IW 218/83, Mathematisch Centrum, 1983.

[12] BERGSTRA, J.A. & J.W. KLOP, *The algebra of recursively defined processes and the algebra of regular processes*, Department of Computer Science Report IW 235/83, Mathematisch Centrum, 1983.

[13] DUGUNDJI, J., *Topology*, Allen and Bacon, 1966.

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

[15] GIERZ, G. et al., *A Compendium of Continuous Lattices*, Springer, 1980.

[16] GOLSON, W. & W.C. ROUNDS, *Connections between two theories of concurrency - metric spaces and synchronization trees*, University of Michigan Computing Research Laboratory Technical Report CRL TR 3-83, 1983.

[17] HAHN, H., *Reelle Funktionen*, Chelsea, 1948.

[18] HOPCROFT, J.E. & J.D. ULLMAN, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, 1979.

[19] MICHAEL, E., *Topologies on spaces of subsets*, Trans. AMS 71, 152-182, 1951.

[20] MILNER, R., *A Calculus of Communicating Systems*, LNCS 92, Springer, 1980.

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

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

[23] PLOTKIN, G.D., *A power domain for countable nondeterminism*, Proc. 9th ICALP (M. Nielsen & E.M. Schmidt, eds), 418-428, LNCS 140, Springer, 1982.

[24] ROUNDS, W.C., *On the relationships between Scott domains, synchronization trees, and metric spaces*, preprint, Department of Computer and Communication Sciences, University of Michigan, 1983.

[25] SCOTT, D.S., *Domain equations,* Proc. 6th IBM Symp. on Mathematical Foundations of Computer Science: Logical Aspects of Programs, 105-256, IBM Japan, 1981.

[26] SMYTH, M.B., *Power domains and predicate transformers, a topological view,* Proc. 10th ICALP (J. Diaz, ed.), 662-675, LNCS 154, Springer, 1983.

69D13
69F12
69F32
69F33