Modelling infinitary behaviours of communicating systems

Philippe Darondeau, Boubakar Gamatié

INRIA

UNITÉ DE RECHERCHE
INRIA-RENNES

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France

Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 749

MODELLING INFINITARY BEHAVIOURS OF COMMUNICATING SYSTEMS

Philippe DARONDEAU
Boubakar GAMATIE

NOVEMBRE 1987

IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE
ET SYSTÈMES ALÉATOIRES

Campus Universitaire de Beaulieu
35042-RENNES CÉDEX
FRANCE
Téléphone: 99 36 20 00
Télex: UNIRISA 950 473 F
Télécopie: 99 38 38 32

MODELLING INFINITARY BEHAVIOURS OF COMMUNICATING SYSTEMS

Philippe Darondeau & Boubakar Gamatié

Publication Interne n° 375 - Octobre 1987 - 52 pages

Abstract

We present a series of three denotational models for CCS, where finite and infinite behaviours both are accounted for by means of greatest fixed points. The two extremes in the series are an operational model, where the meanings of programs are transition systems, and an observational model, where meanings are a combination of ready sets and extended communication traces. The operational model is derived through a systematic construction from structural operational specifications. The observational model is obtained both both both from the operational one in two steps, corresponding to successive applications of a closed and a non closed erasing morphism. The observational model is proved fully abstract for the equivalence p ~ q iff $\forall r$ L(p,r)=L(q,r), where L(p,r) is the set of the finite or infinite traces of possible interactions between programs p and r set in parallel.

MODELES DES COMPORTEMENTS INFINITAIRES DE SYSTEMES COMMUNICANTS

Résumé

Nous présentons une série de trois modèles dénotationnels pour CCS, dans lesquels comportements finis et infinis sont simultanément représentés au moyen de plus grands points fixes. Les deux bouts de la chaine sont un modèle opérationnel, dans lequel les significations des programmes sont des systèmes de transitions, et un modèle observationnel, dans lequel les significations sont des associations de ready sets et de traces de communication étendues. Le modèle opérationnel est obtenu par une construction systématique à partir des spécifications opérationnelles structurelles. Le modèle observationnel est dérivé du modèle opérationnel en deux étapes, correspondant à l' application d'un morphisme fermé et à celle d'un morphisme non fermé. Le modèle observationnel est fully abstract pour l'équivalence p ~ q ssi $\forall r$ L(p,r)=L(q,r), où L(p,r) est l'ensemble des traces finies ou infinies des interactions possibles entre les programmes p et r, mis en parallèle.

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE
(L. A. 227)
UNIVERSITÉ DE RENNES 1 I. N. S. A. DE RENNES

INSTITUT NATIONAL DE RECHERCHE
EN INFORMATIQUE ET EN AUTOMATIQUE
(LABORATOIRE DE RENNES)

TABLE OF CONTENTS

MODELLING INFINITARY BEHAVIOURS OF COMMUNICATING SYSTEMS

Ph. DARONDEAU B. GAMATIE
IRISA-INRIA
Campus Universitaire de Beaulieu
35042 Rennes cedex (France)

"Fixed point semantics in my opinion, is a general terminology, to say, the homomorphic image, in some semantic domain, of the operational semantics, or more exactly the set of computations. So you consider the unique, or the set of computations of a program, and then you extract some relevant information, and you want to characterize this relevant information without looking at the set of computations. But the problem is, what is the relevant information about the computation ?"

André ARNOLD, IFIP WG 2.2, 1983 (in [10], section 'discussion')

1. INTRODUCTION

This paper, evolved from [15], suggests a systematic method for deriving models of programming languages from their operational specifications, and applies it to produce an observational model for infinitary behaviours of communicating systems (in Milner's C.C.S). Let us explicitate the two objectives separately.

For us, the observational models of a programming language L are those models which are fully abstract [27] w.r.t. equivalences or preorders induced by L-defined experiments such as tests [18,19]. Of course, all the observational models are homomorphic images of the operational model which defines algebraically the meanings of L programs as transition systems, but arbitrary homomorphisms do not necessarily define observational models, for it

is not always possible to find out families of L-experiments which induce their canonical equivalence. Concerning Milner's Calculus of Communicating Systems [28], observational models based on tests with binary results have been constructed for the pure version of the calculus [18,25]. However, it has been established that infinite tests with infinitary results are strictly more discriminating than binary tests, and therefore induce different models [13,14]. There lays the main motivation for the present paper where we construct the observational model for the largest possible family of infinite CCS tests with infinitary results. Despite some disagreement about divergence, the resulting model may be seen as a development of the ideas presented in [20,21,22].

Let us now discuss the issue of systematic methods for producing semantics of programming languages. A striking feature of the denotational method is its close affinity to logics, specially patent in the fact that interpretation functions are homomorphisms between Σ-algebras of terms and Σ-algebras of meanings. Nevertheless, model morphisms have not been paid so far much attention in computer science. Joining some other authors [10,29] we would like to emphasize the essential role of model morphisms in any systematic method for assigning semantics to a programming language L : given a model of L for some equivalence on transition systems generated by programs, other series of models may be derived therefrom through morphisms which forget more and more information about the transition systems. The derivation is not always easy, for some trouble may arise with fixpoints as will be seen later on. Nevertheless, the main gap in the method is probably the lack of a general process for obtaining a primary model from the operational specifications of L. We contribute to fill in that gap by providing a rather general technique for translating structural operational specifications à la Plotkin [30] into denotational definitions of transition system meanings.

The remaining sections are organized as follows. The general problem of translating structural operational definitions into equivalent denotational definitions is addressed in section 2; an operational model of CCS is produced in section 3 along the resulting technique. Section 4 relates several forms of observational equivalences on CCS programs and fixes the task of constructing a fully abstract model for the equivalence: $p \sim q$ if and only if p and q have similar sets of finite or infinite interactions with any program r. Section 5 introduces an auxiliary model of CCS, located halfway between the operational model and the fully abstract model for $\sim$; that auxiliary model is obtained by

simply erasing intermediate program states from computations and replacing their endpoints by ready sets. In section 6, we extend the observational equivalence ~ into an observational preorder $\lesssim$ and connect $\lesssim$ to preorders in the auxiliary model. The fully abstract model for $\lesssim$ is drawn from the auxiliary model in section 7, and is proved correct in section 8. The concluding section 9 is followed by an appendix where the proofs of the starred propositions may be found.

2. FROM OPERATIONAL DEFINITIONS TO OPERATIONAL MODELS

In this section, we address the problem of translating structural operational definitions à la Plotkin [30] into equivalent denotational definitions. We provide a uniform translation, yielding models where the meanings of recursive programs are greatest fixed points in the lattice of sets of infinitary transition sequences. An alternative translation, where the meanings of recursive programs are fixed points of contracting mappings,has been studied in [2].

2.1 Assumptions on syntax

We assume given a set V of program variables, ranged over by x, and an arbitrary signature $\Sigma = \cup\{\Sigma_k / \ 0 \leq k \leq K\}$, where Σ_k is ranged over by the k-ary symbol ω_k. The set of recursive open terms (denoted TERM) has typical elements t with the following syntax :

$$t = x \mid \omega_i(t_1, \ldots t_i) \mid t \ \underline{wh} \ (x_1 = t_1, \ldots x_n = t_n).$$

In the above, $(x_1 = t_1, \ldots x_n = t_n)$ is a declaration and represents a function $D:V \rightarrow$ TERM with finite non empty domain $dom(D) = \{x_1, \ldots x_n\}$ and corresponding values $D(x_i) = t_i$. In (t $\underline{wh}$ D) the free occurences of the x_i are bound to the t_i in t and in the t_j's. Recursive terms are defined up to the α-conversion of their bounded variables, hence we assume that two distinct declarations have disjoint domains and that no conflict occurs within terms between free variables and bounded variables. The set of programs (denoted PROG) is the set of terms without free variables (i.e. closed terms). We let metavariables s,t,u,v denote both open terms and programs.

2.2 Assumptions on the operational definitions

We assume given an arbitrary set N of transition labels (ranged over by υ), with a distinguished element σ, called the null action, and a finite set A of schemes of axioms and rules of inference for logical transitions $u -_D^{\upsilon}\rightarrow v$ where $u,v \in$ TERM, $\upsilon \in$ N, and D is either a declaration or the undefined function Ω. We let LT denote the set of the logical transitions between terms.

With regard to recursive definitions, we suppose that A contains the following axiom and rules for declarations, where t,u resp. D,D' are metavariables ranging over TERM resp. over (V $\rightarrow$ TERM).

AXIOM

$\Rightarrow \quad x -_D^{\sigma}\rightarrow D(x)$ for D(x) defined — DEC1

RULES

$t -_D^{\upsilon}\rightarrow u \Rightarrow (t \text{ wh } D') -_D^{\upsilon}\rightarrow (u \text{ wh } D')$ — DEC2

$t -_D^{\upsilon}\rightarrow u \Rightarrow (t \text{ wh } D) -_{\Omega}^{\upsilon}\rightarrow (u \text{ wh } D)$ — DEC3

$t -_{\Omega}^{\upsilon}\rightarrow u \Rightarrow t -_D^{\upsilon}\rightarrow u$ — DEC4

With regard to operators in Σ, we set the requirement that all axioms and rules appear in the form

$$\frac{\{\ t_1 -_D^{\upsilon_1}\rightarrow t'_1, \ldots t_n -_D^{\upsilon_n}\rightarrow t'_n\ \}}{\omega(u_1,\ldots u_k) -_D^{\upsilon}\rightarrow \omega'(u'_1,\ldots u'_m)} \{\text{for } R(\upsilon_1,\ldots \upsilon_n, \upsilon)\}$$

where braces indicate optional parts and the following are satisfied:

- $\omega \in \Sigma_k$, $\omega' \in \Sigma_m$ and $m \leq k$, or $\omega' = id \notin \Sigma$ and $m = 1$
- the metavariables t_i (resp. t'_i resp. u_i resp. u'_i) stand for generic terms, and D is a fixed meta-variable ranging over (V $\rightarrow$ TERM)
- all the t_i (resp. t'_i resp. u_i resp. u'_i) are different
- $\{t_1 \ldots t_n\} \subseteq \{u_1 \ldots u_k\}$
- $\{t'_1 \ldots t'_n\} = \{u'_1 \ldots u'_m\} \setminus \{u_1 \ldots u_k\}$.

These constraints are reminiscent from [17]. Up to the identification between t and id(t), we consider that the quadruple $\omega(u_1 \ldots u_k) -_D{}^{\upsilon}\!\!\rightarrow u'_1$ is a logical transition as soon as $\omega(u_1 \ldots u_k) -_D{}^{\upsilon}\!\!\rightarrow id(u'_1)$ has been proved, and we call it a degenerated transition.

2.3 The operational domain D_{op}

Our goal is to construct a generic model $\mathcal{M}_{op}$ of programs s.t. $\mathcal{M}_{op}[\![t]\!] = \{\varepsilon_t\} \cup \{(t_i -^{\upsilon_i}\!\!\rightarrow t_{i+1})_{i<\gamma} \;/\; t_0 = t \;\&\; \forall i \; (t_i -\Omega^{\upsilon_i}\!\!\rightarrow t_{i+1}) \in LT\}$. The domain D_{op} of the model is a complete lattice, namely the powerset $(P(^{\infty}PT), \subseteq)$, where $^{\infty}PT$ is the set of the sequences ε_t or $(t_i -^{\upsilon_i}\!\!\rightarrow t_{i+1})_{i<\gamma}$ such that $t \in TERM$, $\gamma \leq \omega$ and $\forall i<\gamma$: $(t_i -^{\upsilon_i}\!\!\rightarrow t_{i+1}) \in TERM \times N \times TERM$. We let $^{*}PT$ be defined in the same way with $\gamma \neq \omega$. Be aware of the fact that the elements of the set $^{\infty}PT = TERM \times N \times TERM$, called pseudo-transitions, are not necessarily provable in A.

Up to the isomorphism sending θ_u to $\langle u, \theta_u \rangle$ when θ_u is ε_u or $(u -^{\upsilon}\!\!\rightarrow v)\theta_v$, $^{\infty}PT$ is a complete partial order for the product $\leq$ of the discrete and prefix orders; it is also a complete metric space under product of the discrete and ultrametric topologies [7]. Observe that limits in the Scott topology induced by $\leq$ coïncide with limits in the ultrametric topology. Henceforth, θ resp. Q range over $^{\infty}PT$ resp. $P(^{\infty}PT)$, and $Pref(\theta)$ resp. $CL(Q)$ mean the set of finite prefixes of θ resp. the topological closure of Q.

2.4 The generic model $\mathcal{M}_{op}$

The interpretation of the operational model $\mathcal{M}_{op}$ is a continuous algebra $(D_{op}, \Sigma \cup \{\sigma_x / x \in V\}, \subseteq)$, where the σ_x are implicit guard operators of arity 1 (one per variable x). All the operations $\omega_k : D_{op}{}^k \rightarrow D_{op}$ are the union additive extensions of homonym operations $(^{\infty}PT)^k \rightarrow D_{op}$. The implicit guard operators σ_x: $(^{\infty}PT) \rightarrow D_{op}$, are defined as $\sigma_x(\theta) = \{\varepsilon_x\} \cup \{(x -^{\upsilon}\!\!\rightarrow u)\theta' \;/\; \varepsilon_u \leq \theta' \leq \theta\}$.

As an exception to the standard rule of union additive extension, which is implicitly used throughout the paper, we set $\sigma_x(Q) = \{\varepsilon_x\}$ if Q is the empty set.

The interpretation $(D_{op}, \Sigma \cup \{\sigma_x / x \in V\}, \subseteq)$ induces a meaning function $\mathcal{M}_{op}$: TERM$\rightarrow$(ENV$\rightarrow D_{op}$) where ENV is the set of environments $e \in (V \rightarrow D_{op})$. In the inductive specification given hereafter, Fix $\mathbf{X}$ $\mathbf{F}$ denotes the greatest fixed point of $\mathbf{F}$ in $\mathbf{X}$, and the product operation $\times$: (V $\rightarrow$ TERM) x PT $\rightarrow$ PT is defined as $D \times \varepsilon_u = \varepsilon_u$ wh D , $D \times (t_i \overset{v}{\rightarrow} t_{i+1})_{i<\gamma} = (t_i$ wh $D \overset{v}{\rightarrow} t_{i+1}$ wh $D)_{i<\gamma}$. As usual, $e[\mathbf{X}/\mathbf{x}]$ stands for $e[X_1/x_1]\ldots[X_n/x_n]$, where $e[X/x]$ is e' such that $e'(x)=X$ and $e'(y)=e(y)$ for $x \neq y$.

THE INDUCTIVE SPECIFICATION OF $\mathcal{M}_{op}$

1. $\mathcal{M}_{op}[|x|](e) = \sigma_x(e(x))$
2. $\mathcal{M}_{op}[|\omega_k(t_1 \ldots t_k)|](e) = \omega_k(\mathcal{M}_{op}[|t_1|](e), \ldots \mathcal{M}_{op}[|t_k|](e))$
3. $\mathcal{M}_{op}[|t$ wh $(x_1=t_1, \ldots x_n=t_n)|](e) = D \times \mathcal{M}_{op}[|t|](e[\mathbf{V}/\mathbf{x}])$

   letting $\mathbf{V} = Fix_{\mathbf{X}}\ \mathbf{F}(e[\mathbf{X}/\mathbf{x}])$, $F_i(e) = \mathcal{M}_{op}[|t_i|](e)$,

   and $D=(x_1=t_1, \ldots x_n=t_n)$, $\mathbf{x}=(x_1, \ldots x_n)$, $\mathbf{X}=(X_1, \ldots X_n)$, $\mathbf{F}=(F_1, \ldots F_n)$.

Remark Since D_{op} is a complete lattice, the principle of Bekic' and Scott for simultaneous fixed points is valid [5].

In order to specialise the above specification, we suggest now a systematic construction of Σoperators ω_k: $(D_{op})^k \rightarrow D_{op}$ based on the axioms and rules in A.

2.5 Towards a finitary Σ-interpretation

We proceed first to the construction of finitary operations $\underline{\omega}_k$: $(^*PT)^k \to P(^*PT)$. In the sequel, $\underline{id}$ is the identity on $^*PT$, and $(u \xrightarrow{\upsilon} v)\bullet\theta$ is the set $\{\varepsilon_u,\ (u \xrightarrow{\upsilon} v)\} \cup \{(u \xrightarrow{\upsilon} v)\theta' / \ \varepsilon_v < \theta' \leq \theta\}$.

For ω in Σ, let $A[\omega]$ be the subset of axioms and rules in A of the form:

$$\gamma: \quad \frac{\{\ t_1 -_D \upsilon_1 \to t'_1\ ,\ldots\ t_n -_D \upsilon_n \to t'_n\ \}}{\omega(u_1,\ldots u_k) -_D \upsilon \to \omega'(u'_1,\ldots u'_m)} \qquad \{\text{for } R(\upsilon_1,\ldots\upsilon_n))\}$$

For γ in $A[\omega]$, let relations γ', γ'' on $\{1\ldots k\}\times\{1\ldots m\}$ and functions $f\gamma: \{1\ldots n\} \to \{1\ldots k\}$, $g\gamma: \{1\ldots n\} \to \{1\ldots m\}$ be defined as follows:

$\gamma'(i,j) \Leftrightarrow (\exists l,\ u_i = t_l \ \&\ t'_l = u'_j)$,

$\gamma''(i,j) \Leftrightarrow (u_i = u'_j)$,

$f\gamma(i)=j \ \Leftrightarrow \ t_i = u_j$,

$g\gamma(i)=j \ \Leftrightarrow \ t'_i = u'_j$.

For $\gamma \in A[\omega]$, $\upsilon \in N$ and $\theta_1\ldots\theta_k \in {}^*PT$, let $\omega^{\gamma,\upsilon}(\theta_1\ldots\theta_k)$ stand for the conditional expression $[\text{cond},\ (\omega(v_1\ldots v_k) \xrightarrow{\upsilon} \omega'(v'_1\ldots v'_m))\bullet\omega'(\theta'_1\ldots\theta'_m),\ \emptyset]$ where cond stands for the conjunction of the following conditions 1 to 3 (for all i and j):

1 - $(\upsilon_1,\ldots\upsilon_n,\upsilon) \in R$,

2 - $\theta_{f\gamma(i)} = (v_{f\gamma(i)} \xrightarrow{\upsilon_i} v'_{g\gamma(i)})\ \theta'_{g\gamma(i)}$,

3 - $u'_i = u_j \Rightarrow \theta'_i = \theta_j \ \&\ \varepsilon_{v'_i} \leq \theta'_i \ \&\ \varepsilon_{vj} \leq \theta_j$.

The finitary operations $\underline{\omega}_k : (^*PT)^k \to P(^*PT)$ are specified by the inductive formula **F** :

$$\underline{\omega}_k\ (\theta_1\ldots\theta_k) = \{\varepsilon_t / \ t = \omega_k(t_1\ldots t_k) \ \&\ \varepsilon_{ti} \leq \theta_i\} \cup [\cup\ \{\omega_k^{\gamma,\upsilon}(\theta_1\ldots\theta_k) \ / \ \gamma \in A[\omega_k] \ \&\ \upsilon \in N\}]$$

The following properties are verified :

- $(\forall \theta_i)$, $\underline{\omega}_k(\theta_1 \ldots \theta_k)$ is prefix closed,
- $(\forall i,\ \theta_i \leq \theta'_i) \Rightarrow \omega_k(\theta_1 \ldots \theta_k) \subseteq \omega_k(\theta'_1 \ldots \theta'_k)$.

For $t \in$ TERM and $D \in (V \rightarrow \text{TERM})$, let $D^*[|t|]$ be the set of sequences ε_t or $(t_i -^{v_i} \rightarrow t_{i+1})_{i \leq n}$ s.t. $t = t_0$ and $\forall i\ (t_i -_D^{v_i} \rightarrow t_{i+1}) \in LT$. Next proposition shows that functions D^* act like morphisms between the Σ-algebra of terms and the algebra $(D_{op},\ (\underline{\omega}/\ \omega \in \Sigma))$.

proposition 2.1* $\quad \forall t_i \in \text{TERM},\ \forall \omega_k \in \Sigma_k,\ \forall D$:

$$D^*[|\omega_k(t_1 \ldots t_k)|] = \underline{\omega}_k(D^*[|t_1|], \ldots D^*[|t_k|])\ .$$

2.6 Towards an infinitary Σ-interpretation

We introduce here infinitary operations $\overline{\omega_k}$: $(^{\infty}PT)^k \rightarrow P(^{\infty}PT)$ consistent with formula F. These operations are defined as $\overline{\omega_k}(\theta_1 \ldots \theta_k) = CL(\underline{\omega}_k(Pref(\theta_1) \ldots Pref(\theta_k))$, and are extended along union additive extension into operators ω_k working on sets. Since $\overline{\omega_k}(\theta_1 \ldots \theta_k)$ is the set of limits of the increasing chains in $\underline{\omega}_k(Pref(\theta_1) \ldots Pref(\theta_k))$, the following arise directly from the properties of the $\underline{\omega}_k$:

- $(\forall \theta_i)\ \overline{\omega_k}(\theta_1 \ldots \theta_k)$ is closed and prefix closed,
- $(\forall i,\ \theta_i \leq \theta'_i) \Rightarrow \overline{\omega_k}(\theta_1 \ldots \theta_k) \subseteq \overline{\omega_k}(\theta'_1 \ldots \theta'_k)$.

The proof for the consistency with formula F is straightforward but lengthy.

Let us define the saturated subsets of $^{\infty}PT$ as the closed and prefix closed sets Q such that: $\quad (\theta_1 \in Q\ \&\ \theta_2\theta_3 \in Q) \Rightarrow (\theta_1\theta_3 \in {}^{\infty}PT \Rightarrow \theta_1\theta_3 \in Q)$.

Next proposition states the crucial property of the extended operations ω_k.

proposition 2.2* For saturated sets Q_i :

$$\omega_k(Q_1 \ldots Q_k) = CL(\underline{\omega}_k(Pref(Q_1) \ldots Pref(Q_k)))\ .$$

2.7 Full adequacy

The following theorem expresses the full adequacy of the model $\mathcal{M}_{op}$ w.r.t. the logical system A.

Theorem 1 $\forall\, t \in PROG,\ \forall e:\ \mathcal{M}_{op}[|t|](e) = \{\varepsilon_t\} \cup \{(t_i -^{\nu_i}\!\!\rightarrow t_{i+1})_{i<\gamma} /\ t_0 = t\ \&\ \gamma \leq \omega\ \&\ \forall i,\ (t_i -^{\Omega^{\nu_i}}\!\!\rightarrow t_{i+1}) \in LT\}$.

proof If we let $\Omega^{\infty}[|t|] = CL(\Omega^{*}[|t|])$, the above reads as $\mathcal{M}_{op}[|t|](e) = \Omega^{\infty}[|t|]$. In order to show that equality, let us define $\mathcal{M}^{*}_{op}$ in the same way as $\mathcal{M}_{op}$, but with least fixed points in place of greatest fixed points. Then $\mathcal{M}^{*}_{op}[|t|](e) = \Omega^{*}[|t|]$, by proposition 2.1 and Tarski's least fixed point theorem for ω-continuous functions [32] . In view of the next lemma, the expected result follows with the help of the Beki`c principle for simultaneous fixed points.

lemma Let $\mathbb{X} = \langle X_1 \ldots X_n \rangle$ and $\mathbb{F}(\mathbb{X}) = \langle F_1(\sigma_{x1}(X_1) \ldots \sigma_{xn}(X_n)), F_n(\sigma_{x1}(X_1) \ldots \sigma_{xn}(X_n)) \rangle$ where the F_i are finite expressions over Σ extended with left products $(D\ x)$. Let $\langle Q_1 \ldots Q_n \rangle$ be the least solution of $\mathbb{X} = \mathbb{F}(\mathbb{X})$ in $(D_{op})^n$, then $\langle CL(Q_1) \ldots CL(Q_n) \rangle$ is the greatest solution of $\mathbb{X} = \mathbb{F}(\mathbb{X})$ in $(D_{op})^n$.

proof Let us first observe that the Q_i are prefix closed subsets of $^{*}PT$, whence $\langle CL(Q_1) \ldots CL(Q_n) \rangle$ is a solution by virtue of proposition 2.2. Now, for any solution $\langle Q'_1 \ldots Q'_n \rangle$, $\langle Pref(Q'_1) \ldots Pref(Q'_n) \rangle$ is also a solution, and thus we are done if we can prove that $\mathbb{F}$ has a unique fixed point in $(P(^{*}PT))^n$. Let us observe that for any $\theta_i \in {}^{*}PT$ and $\omega_k \in \Sigma$, $\{\theta /\ |\theta| < n\ \&\ \theta \in \omega_k(\theta_1 \ldots \theta_k)\}$ is included in $\omega_k(\theta_1^{<n} \ldots \theta_k^{<n})$, letting $\theta^{<n}$ denote the longest prefix of θ with length $(|\theta^{<n}|) < n$. Taking into account the implicit guard operators σ_{xi}, the expected conclusion follows by Tarski's least fixed point theorem for continuous functions.

The general framework elaborated here is exploited in the next section where we produce an operational model of CCS.

3. A CALCULUS OF COMMUNICATING SYSTEMS

We consider in fact a slightly augmented version of Milner's pure calculus of communicating systems [18,25]. The extensions affect unguarded recursion and non deterministic choice. Since the calculus is widely known, we keep the description concise and refer the reader to the bibliography for more comprehensive presentations.

We let Λ denote the union of two disjoint sets of complementary actions Δ and $\eta(\Delta)$, linked together by a system of reciprocal bijections η (i.e., $\eta(\eta(\lambda))=\lambda$ for $\lambda\in\Lambda$). The set of actions is $M=\Lambda\cup\{\tau\}$, where $\tau\notin\Lambda$ is the internal action. The set of transition labels is $N=M\cup\{\sigma\}$. Throughout the paper, we let λ resp. μ resp. ν range over Λ resp. M resp. N. We call renaming functions the one-one partial functions ρ from N to N such that $\rho(\upsilon)=\upsilon$ almost everywhere, $\rho(\sigma)=\sigma$, $\rho(\tau)=\tau$, and $\rho(\eta(\lambda)) = \eta(\rho(\lambda))$ unless both members of the equality are undefined.

The signature Σ of our algebra of CCS terms is $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$ where the Σ_i (ranged over by ω_i) are the following sets of i-ary symbols :

Σ_0 = {NIL},

Σ_1 = {μ / $\mu\in$ M} $\cup$ {ρ /ρ is a renaming function},

Σ_2 = {|, +, $\oplus$}.

Operators |, +, $\oplus$ are the usual asynchronous composition, external choice and internal choice. The binary operators are infixed, the guarding operators (μ) are prefixed, the renaming operators (ρ) are postfixed.

The operational definition of Σ is given by the following axioms and rules (where D is a fixed metavariable for declarations).

AXIOMS

$\Rightarrow \quad \mu t \xrightarrow{\mu}_D t$

$\Rightarrow \quad t \oplus u \xrightarrow{\sigma}_D t \qquad\qquad \Rightarrow t \oplus u \xrightarrow{\sigma}_D u$

RULES

$t -_D^{\mu}\rightarrow t' \Rightarrow t+u -_D^{\mu}\rightarrow t' \quad , \quad u+t -_D^{\mu}\rightarrow t'$

$t -_D^{\mu}\rightarrow t' \Rightarrow t|u -_D^{\mu}\rightarrow t'|u \quad , \quad u|t -_D^{\mu}\rightarrow u|t'$

$(t -_D^{\lambda}\rightarrow t', u -_D^{\lambda'}\rightarrow u') \Rightarrow (u|t -_D^{\tau}\rightarrow u'|t'), (t|u -_D^{\tau}\rightarrow t'|u')$ for $\lambda' = \eta(\lambda)$

$t -_D^{\upsilon}\rightarrow t' \Rightarrow t\rho -_D^{\rho\upsilon}\rightarrow t'\rho$ for $\rho(\upsilon)$ defined

$t -_D^{\sigma}\rightarrow t' \Rightarrow (t\ \omega_2\ u) -_D^{\sigma}\rightarrow (t'\ \omega_2\ u) \quad , \quad (u\ \omega_2\ t) -_D^{\sigma}\rightarrow (u\ \omega_2 t')$

By adding to the above the general axioms and rules (DEC1-DEC4), one obtains an axiom system *A* for transitions between recursive open terms (e.g. t wh D). As regards CCS programs, let us observe that every program has a finite number of immediate derivatives and a finite sort, where sort(t) is the set of all the possible actions λ of all the derivatives of t.

Example: A sample proof of transition is the following :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{x -_{D'}^{\sigma}\rightarrow D'(x)}{x|y -_{D'}^{\sigma}\rightarrow D'(x)|y}}{(x|y)\,\underline{wh}\ D -_{D'}^{\sigma}\rightarrow (D'(x)|y)\,\underline{wh}\ D}}{((x|y)\,\underline{wh}\ D)\,\underline{wh}\ D' -_{\Omega}^{\sigma}\rightarrow ((D'(x)|y)\,\underline{wh}\ D)\,\underline{wh}\ D'}}{((x|y)\,\underline{wh}\ D)\,\underline{wh}\ D' -_{D''}^{\sigma}\rightarrow ((D'(x)|y)\,\underline{wh}\ D)\,\underline{wh}\ D'}}{(((x|y)\,\underline{wh}\ D)\,wh\ D')\,\underline{wh}\ D'' -_{\Omega}^{\sigma}\rightarrow (((D'(x)|y)\,\underline{wh}\ D)\,\underline{wh}\ D')\,\underline{wh}\ D''}$$

Remark Since all the unwinding steps of recursive programs are traced by null actions, infinite sequences of σ-transitions are a characteristic mark of static divergence.

In order to make explicit a specialized version of the fully adequate model $\mathcal{M}_{op}$, we may now apply formula F and draw from the above specific axioms and rules inductive definitions of specific operations $\omega_k : ({}^*PT)^k \rightarrow P({}^*PT)$. The

resulting relations are valid also for the infinitary operations $\overline{\overline{\omega_k}}:({}^{\infty}PT)^k \to P({}^{\infty}PT)$. For that reason, we omit the underlining or surlining of operator symbols in the statements given below. To keep notations reasonable, we let $\theta_s=(s-\upsilon\to t)\theta_t$ and $\theta_u=(u-\upsilon'\to v)\theta_v$, with $\varepsilon_t\le\theta_t$ and $\varepsilon_v\le\theta_v$. We recall the reader that $(u-\upsilon\to v)\bullet Q$ means $\emptyset$ if $Q=\emptyset$ or otherwise $\{\varepsilon_u, (u-\upsilon\to v)\}\cup \{(u-\upsilon\to v)\theta'/\varepsilon_v\le\theta'\le\theta\in Q\}$.

THE OPERATIONAL Σ INTERPRETATION

D1 $\quad nil = \{\varepsilon_{nil}\}$

D2 $\quad \mu(\theta_t) = (\mu t-\mu\to t)\bullet\theta_t$

D3 $\quad (\varepsilon_t)\rho = \{\varepsilon_{t\rho}\},$

$\quad (\theta_s)\rho = [\upsilon\notin dom(\rho),\ \{\varepsilon_{s\rho}\},\ (s\rho-\rho(\upsilon)\to t\rho)\bullet(\theta_t)\rho]$

D4 $\quad \varepsilon_t+\varepsilon_v = \{\varepsilon_{t+v}\}$

$\quad \theta_s+\varepsilon_v = [\upsilon\neq\sigma,\ (s+v-\upsilon\to t)\bullet\theta_t,\ (s+v-\upsilon\to t+v)\bullet(\theta_t+\varepsilon_v)],$

$\quad \varepsilon_t+\theta_u = [\upsilon'\neq\sigma,\ (t+u-\upsilon'\to v)\bullet\theta_v,\ (t+u-\upsilon'\to t+v)\bullet(\varepsilon_t+\theta_v)],$

$\quad \theta_s+\theta_u = [\upsilon\neq\sigma,\ (s+u-\upsilon\to t)\bullet\theta_t,\ (s+u-\upsilon\to t+u)\bullet(\theta_t+\theta_u)]\cup$
$\quad [\upsilon'\neq\sigma,\ (s+u-\upsilon'\to v)\bullet\theta_v,\ (s+u-\upsilon'\to s+v)\bullet(\theta_s+\theta_v)]$

D5 $\quad \varepsilon_t\oplus\varepsilon_v = \{\varepsilon_{t\oplus v},\ (t\oplus v-\sigma\to t),\ (t\oplus v-\sigma\to v)\}$

$\quad \theta_s\oplus\varepsilon_v = \{(s\oplus v-\sigma\to v)\}\cup(s\oplus v-\sigma\to s)\bullet\theta_s \cup$
$\quad [\upsilon=\sigma,\ (s\oplus v-\upsilon\to t\oplus v)\bullet(\theta_t\oplus\varepsilon_v),\ \emptyset]$

$\quad \varepsilon_t\oplus\theta_u = \{(t\oplus u-\sigma\to t)\}\cup(t\oplus u-\sigma\to u)\bullet\theta_u \cup$
$\quad [\upsilon'=\sigma,\ (t\oplus u-\upsilon'\to t\oplus v)\bullet(\varepsilon_t\oplus\theta_v),\ \emptyset]$

$\quad \theta_s+\theta_u = (s\oplus u-\sigma\to s)\bullet\theta_s\cup(s\oplus u-\sigma\to u)\bullet\theta_u \cup$
$\quad [\upsilon=\sigma,\ (s\oplus u-\upsilon\to t\oplus u)\bullet(\theta_t\oplus\theta_u),\ \emptyset]\cup$
$\quad [\upsilon'=\sigma,\ (s\oplus u-\upsilon'\to s\oplus v)\bullet(\theta_s\oplus\theta_v),\ \emptyset]$

D6 $\quad \varepsilon_t|\varepsilon_v = \{\varepsilon_{t|v}\},$

$\quad \theta_s|\varepsilon_v = ((s|v)-\upsilon\to(t|v))\bullet(\theta_t|\varepsilon_v),$

$\quad \varepsilon_t|\theta_u = ((t|u)-\upsilon'\to(t|v))\bullet(\varepsilon_t|\theta_v),$

$\quad \theta_s|\theta_u = ((s|u)-\upsilon\to(t|u))\bullet(\theta_t|\theta_u))\cup((s|u)-\upsilon'\to(s|v))\bullet(\theta_s|\theta_v))\cup$
$\quad [(\upsilon\in\Lambda\ \&\ \upsilon'=\eta(\upsilon)),\ ((s|u)-\tau\to(t|v))\bullet(\theta_t|\theta_v)),\ \emptyset]$

---

4. OBSERVATIONAL EQUIVALENCES OF CCS PROGRAMS

Let us fix some terminology before we introduce the concept of observational equivalences. A *computation* issued from program t is a finite or infinite sequence of logical transitions $\theta_t = (t_i \text{-} \upsilon_i \rightarrow t_{i+1})_{i<\gamma}$ originating from $t_0 = t$ (we drop from now on the symbol Ω of the empty declaration) . The *full trace* of θ_t is the finite or infinite word $ftr(\theta_t) = (\upsilon_i)_{i<\gamma}$. The *trace* of θ_t is the finite or infinite word $tr(\theta_t) = \Pi_\Lambda(ftr(\theta_t))$, where Π_Λ is the projection erasing the occurences of symbols σ and τ. A *silent* computation is a computation with trace ε (the empty word) . A *twin-computation* is a pair (θ_u, θ_v) of computations from which it is possible, by the exclusive use of the logical rules of parallel composition, to reconstruct some silent computation $\theta_{u|v}$ that cannot be extended by silent transitions (labeled σ or τ) and cannot be obtained from a strictly smaller pair $(\theta,\theta') \leq (\theta_u, \theta_v)$.

The idea of looking at observational equivalences induced by program defined experiments probably dates back to [11]. There, the author addressed some criticism to bisimulation equivalences and suggested to study another equivalence and congruence, with the intendto make semantics spring from syntax by internalizing experiments in the programming language. In the present framework, these alternative relations may be restated as follows:

Let the set of all the possible interactions between programs u and v be the language $L(u,v) = \{tr(\theta_v) \;/\; \exists \theta_u \; (\theta_u, \theta_v) \in Twincomp\}$, where *Twincomp* is the set of twin-computations; then program t is *observationally equivalent* to program u ($t \sim_1 u$) if and only if $L(t,v) = L(u,v)$ for every program v and program t is *observationally congruent* to program u ($t \approx_1 u$) if and only if $C[t] \sim_1 C[u]$ for every program context C[.].

A similar line of thought has been followed by De Nicola and Hennessy in [18], where they introduced tests as program defined experiments with binary results. Assuming a result function Result defined on computations and such that $\theta \rightarrow$ Result$(\theta) \in \{\perp, T\}$, the proposal of [18] amounts roughly to the

following : program t is *test equivalent* to program u ($t \sim_2 u$) if and only if $\forall$ v, $\{Result(\theta_v) / \exists \theta_t : (\theta_t,\theta_v) \in Twincomp\} = \{Result(\theta_v) / \exists \theta_u : (\theta_u,\theta_v) \in Twincomp\}$, and t is *test congruent* to u ($t \approx_2 u$) if and only if $C[t] \sim_2 C[u]$ for every program context C[.].

The above presented concepts may appear as weakenings of a stronger concept of observational equivalence and congruence, later introduced in [12] for a subset of CCS. The stronger relations are the following: program t is *observationally equivalent* to program u ($t \sim_3 u$) if and only if for every program v, $\{(\theta_v) / \exists \theta_t \ (\theta_t,\theta_v) \in Twincomp\} = \{(\theta_v) / \exists \theta_u \ (\theta_u,\theta_v) \in Twincomp\}$, and t is *observationally congruent* to u ($t \approx_3 u$) if and only if $C[t] \sim_3 C[u]$ for every context C[.].

Notice that both $\sim_1$ and $\sim_3$ are variants of the testing equivalence $\sim_2$, since they may be recovered by setting $Result(\theta)=tr(\theta)$ resp. $Result(\theta)=\theta$ in the definition of $\sim_2$. However, a main difference between on one hand $\sim_1$ and $\sim_3$ and on the other hand $\sim_2$, is the non finiteness of the domain of possible results for tests. That property makes $\approx_1$ and $\approx_3$ strictly more discriminative than $\approx_2$, or than the congruence induced by the following form $\sim'_2$ of $\sim_2$ (for which it was wrongly conjectured in [11] that $\approx'_2 = \approx_1$): $t \sim'_2 u$ iff $\forall \mu$, $L(t,\mu NIL)=L(u,\mu NIL)$. A simple case of disagreement between $\sim_1$ and $\sim_2$ is shown by the following pair of CCS programs t and u (where $\overline{\lambda}=\eta(\lambda)$), and function "$/\alpha$" is the renaming function acting as the identity on $N\setminus\{\alpha,\overline{\alpha}\}$)

$t = ((x|y)|w)/\alpha/\beta/\gamma$ wh ($x = (\alpha x \,|\, \beta NIL) + \alpha\gamma NIL$,
$y = \overline{\gamma} z$,
$z = \overline{\beta}(z|\, \delta NIL)$,
$w = \alpha w$)

$u = t + (x$ wh $(x = \delta x))$.

For these programs t and u : $(t \sim_2 u)$ but $\neg(t \sim_3 u)$ and $\neg(t \sim_1 u)$ since, for $v = x$ wh $(x = \overline{\delta} x)$, we get $L(t,v) = \overline{\delta}^* \neq L(u,v) = \overline{\delta}^* \cup \{\overline{\delta}^\omega\}$.
Incidentaly, this example shows that the set of traces of a CCS program is not necessarily closed in the usual ultrametric topology on words.

The above programs t and u turn out to be non equivalent w.r.t. $\sim_2$ under the assumption of fairness, which thus intensifies the separation power of tests [26]. However, less obvious cases of disagreement still arise between $\sim_1$ and $\sim_2$, due to the deep affinity of effective transition systems with Σ^1_1 parts of $\mathbb{N} \to \mathbb{N}$. The interested reader is referred to [13,14] for the exhibition of such cases.

As regards the equivalences $\sim_1$ and $\sim_3$, the strict inclusion $\sim_3 \subset \sim_1$ holds obviously, for the mapping tr is an erasing morphism. It will be shown with small effort in the paper that the induced congruences ($\approx_1$ and $\approx_3$) are nevertheless identical. After restriction to finite behaviours, those congruences may be axiomatized, as they coincide with $\approx_2$ and also with the congruences studied e.g. in [18,22,8]. In spite of that, we know from [16] that the search of complete proof systems for the unrestricted congruences $\approx_1$ and $\approx_3$ is pointless.

Now we are ready to formulate our objective, which is to construct a fully abstract model of CCS for a contextual preorder $\precsim_1$ generalizing the congruence $\approx_1$. For the ease of the task and clarity of the presentation, we introduce beforehand an intermediate stage between the operational model studied in the previous sections and the observational model at which we are aiming.

5. AN INTERMEDIATE MODEL

According to the terminology of [21,22], let us define the interface set of a program p as the set of (null or non null) actions that p can perform instantaneously, i.e. interface(p) = $\{\upsilon / \exists q,\ p \xrightarrow{\upsilon} q\}$. Let ϕ_{int} be the morphism which transforms sets of computations into sets of stateless computations in an elementwise way, by erasing the initial and intermediate states in all computations, and replacing the final state of finite computations by the interface set of the corresponding program. As far as we are concerned with the investigation of the congruence $\approx_1$, all we need to capture about program t is the information wanted for computing $L(C[t],u)$ for any context C[.] and

program u. Let us forget for the moment about contexts: we claim that enough information for computing $L(t,u)$ is available from $\phi_{int}(\mathcal{M}_{op}[|t|])$ and $\phi_{int}(\mathcal{M}_{op}[|u|])$. Suppose we have at hands a compositional model of programs, say $\mathcal{M}_{int}$, satisfying $\mathcal{M}_{int}[|t|] = \phi_{int}(\mathcal{M}_{op}[|t|])$, then $\mathcal{M}_{int}[|t|]$ yields enough information on t for computing L(C[t],u) for every context C[.] and program u, and the fully abstract model w.r.t. $\sim_1$ must therefore be a factor model of $\mathcal{M}_{int}$.

The goal of the present section is to derive $\mathcal{M}_{int}$ from $\mathcal{M}_{op}$ along the morphism ϕ_{int}. The research of the adequate morphism between $\mathcal{M}_{int}$ and the fully abstract model is left to a subsequent section.

5.1 The erasing morphism ϕ_{int}

In this alinea, we give the precise definition of ϕ_{int} and show the exact dependance between L(t,u) and $\phi_{int}(\mathcal{M}_{op}[|t|])$.

<u>definition</u> The set of *stateless computations*, denoted SC, is the set of all pairs $\langle w,R\rangle$ or $\langle W,\omega\rangle$ such that $w \in N^*$, R is a finite subset of N, and $W \in N^\omega$. The set SC^{fin} is the intersection of SC with $(N^* \times P(N))$.

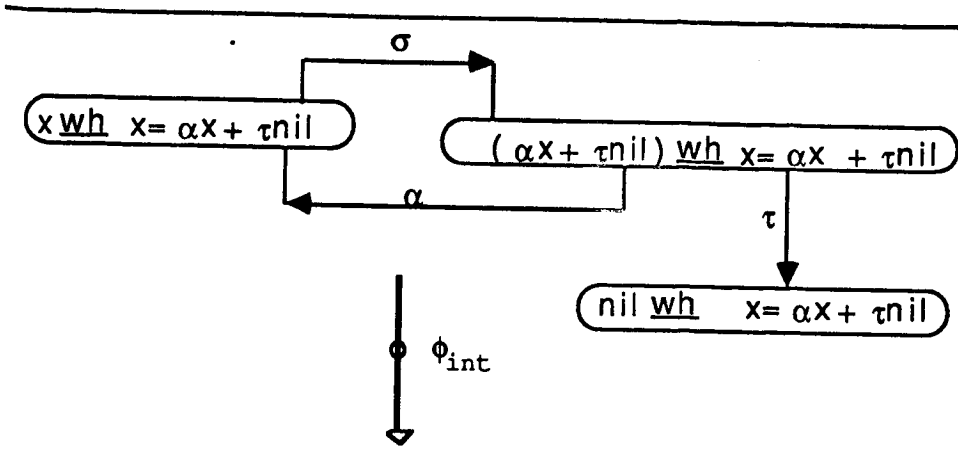
<u>definition</u> $\phi_{int} : P(^\infty LT) \to P(SC)$ is the function : $\phi_{int}(Q) = \{\phi_{int}(\theta) / \theta \in Q\}$, where $\phi_{int} : {}^\infty LT \to SC$ (and ${}^*LT \to SC^{fin}$) is the function defined as:

$\phi_{int}(\theta) = \langle ftr(\theta), \omega\rangle$ if θ is an infinite computation,

$\phi_{int}(\theta) = \langle \varepsilon, interface(t)\rangle$ if θ is the empty computation issued from t,

$\phi_{int}((t\text{-}^{\upsilon}\!\!\to u)\theta) = \langle \upsilon w, R\rangle$ if $\phi_{int}(\theta) = \langle w,R\rangle$.

<u>Example</u>: The set $\phi_{int}(\mathcal{M}op[|\, x \ \underline{wh}\ x=\alpha x+\tau nil\ |])$ is shown in the following figure.



$$\{<(\sigma\alpha)^{\omega}, \omega>\} \cup \{<(\sigma\alpha)^{*}, \{\sigma\}>\} \cup \{<(\sigma\alpha)^{*}\tau, \varnothing>\} \cup \{<(\sigma\alpha)^{*}\sigma, \{\alpha,\tau\}>\}$$

definition: Two stateless computations $<w_1,R_1>$ and $<w_2,R_2>$ are z-twin ($<w_1,R_1> \xleftarrow{z} <w_2,R_2>$) if and only if the following hold:

i) $\Pi_{\Lambda}(w_1) = \eta(z)$ & $\Pi_{\Lambda}(w_2) = z$

ii) $[R_1 \neq \omega \,\&\, R_2 \neq \omega] \rightarrow (R_1 \cup R_2) \cap \{\sigma,\tau\} = \varnothing$ & $R_1 \cap \eta(R_2) = \varnothing$.

Next statement follows easily:

proposition5.1: $L(t,u) = \{z \in \Lambda^{\infty} / \exists \theta_t \in \mathcal{M}_{op}[|t|],\ \exists \theta_u \in \mathcal{M}_{op}[|u|]:\ \phi_{int}(\theta_t) \xleftarrow{z} \phi_{int}(\theta_u)\}$

5.2 The intermediate domain D_{int}

The domain D_{int} of the model $\mathcal{M}_{int}$ is the powerset $P(SC)$ of the set of stateless computations, ordered by pure set inclusion. A set Q in D_{int} is said to be prefix closed, resp. closed, if the set of traces $\{w: <w,R> \in Q\}$ is prefix closed, resp. closed in the usual topology on words. For $Q \in D_{int}$, we let $Fin(Q)$ denote the set of the finite elements in Q , i.e. $Fin(Q) = Q \cap SC^{fin}$, and we let $CL(Q)$ denote the topological closure of Q, i.e. $CL(Q) = Q \cup \{<w,\omega> : w = lim(w_i)$ & $<w_i,R_i> \in Q\}$.

5.3 The intermediate model $\mathcal{M}_{int}$

The interpretation of the model $\mathcal{M}_{int}$ is a continuous algebra $(D_{int}, \Sigma \cup \{\sigma\}, \sqsubseteq)$, where Σ has been added an implicit guard operator σ. All the operations $\omega_k : {D_{int}}^k \to D_{int}$ are the union additive extensions of homonym operations $\omega_k : (SC)^k \to D_{int}$. The implicit guard operator σ: $SC \to D_{int}$ is defined by $\sigma\langle w,R\rangle = \{\langle\varepsilon,\sigma\rangle,\langle\sigma w,R\rangle\}$. As an exception to the standard rule of union additive extension, we set $\sigma Q = \{\langle\varepsilon,\sigma\rangle\}$ if Q is the empty set.

The intermediate meaning function $\mathcal{M}_{int}$: TERM $\to ((V \to D_{int}) \to D_{int})$ is inductively specified by the following statements, where Fix denotes greatest fixed points in the complete lattice $(D_{int})^n$.

---

1. $\mathcal{M}_{int}[\![x]\!](e) = \sigma(e(x))$
2. $\mathcal{M}_{int}[\![\omega_k(t_1,\ldots t_k)]\!](e) = \omega_k(\mathcal{M}_{int}[\![t_1]\!](e), \ldots \mathcal{M}_{int}[\![t_k]\!](e))$
3. $\mathcal{M}_{int}[\![t \text{ wh } (x_1 = t_1,\ldots x_n = t_n)]\!](e) = \mathcal{M}_{int}[\![t]\!](e[\mathbb{V} / \mathbb{x}])$

   letting $\mathbb{V} = Fix_{\mathbb{X}}\mathbb{F}(e[\mathbb{X}/\mathbb{x}])$, $F_i(e) = \mathcal{M}_{int}[\![t_i]\!](e)$ and

   $\mathbb{x} = (x_1, \ldots x_n)$, $\mathbb{X} = (X_1, \ldots X_n)$, $\mathbb{F} = (F_1, \ldots F_n)$.

The elementary operations $\omega_k : (SC)^k \to D_{int}$ are defined by six relations D'_1-D'_6 , where $\otimes$: $N^\infty \times N^\infty \to P(N^\infty)$ is the parallel composition operator defined as follows:

i) for w and w' finite :

a) $w \otimes \varepsilon = \{w\} = \varepsilon \otimes w$

b) $\nu w \otimes \nu' w' = \nu(w \otimes \nu' w') \cup \nu'(\nu w \otimes w') \cup [\nu = \lambda \ \& \ \nu' = \eta(\lambda) \ , \ \tau(w \otimes w'), \varnothing]$

ii) for w or w' infinite : $w \otimes w'$ is the set of words with infinitely many left factors in $(\ lf(w) \otimes lf(w')\ )$, where lf(w) means the finite left factors of w.

Examples: $\alpha^{\omega}\otimes\beta^{\omega}$ is the set of all infinite words on $\{\alpha,\beta\}$, and $\alpha^{\omega}\otimes(\bar{\alpha}^{\omega})$ is the set of all infinite words on $\{\alpha,\bar{\alpha},\tau\}$. $\alpha^{\omega}\otimes\beta$ is the set of all infinite words on $\{\alpha,\beta\}$ with at most one occurence of β. $\alpha\otimes\bar{\alpha}$ is the finite language $\{\alpha\bar{\alpha}, \bar{\alpha}\alpha, \tau\}$. $\sigma^{n}\otimes\sigma^{m}$ is the singleton set $\{\sigma^{n+m}\}$.

In the following, we let $<\{w, w\in L\},R>$ stand for $\{<w,R>/w\in L\}$ and we sometimes omit brackets for singleton sets.

The INTERMEDIATE Σ INTERPRETATION

D'1 $\quad nil = \{<\varepsilon,\emptyset>\}$

D'2 $\quad \mu<w,R> = \{<\varepsilon,\mu>, <\mu w,R>\}$

D'3 $\quad <w,R>\rho =$ **if** $w\notin (dom(\rho)^{\infty})$ **then** $\emptyset$ **else if** $R=\omega$ **then** $<\rho(w), R>$ **else** $<\rho(w), R\cap dom(\rho)>$

D'4 $\quad$ for $u,u'\in\sigma^{\infty}$ and $v,v'\in MN^{\infty}\cup\{\varepsilon\}$,
$<uv,R> + <u'v',R'> =$

$$[\ (v\neq\varepsilon)\ ,\ <(u\otimes lf(u'))v,R>\ ,\emptyset]\ \cup$$
$$[\ (v'\neq\varepsilon)\ ,\ <(lf(u)\otimes u')v',R'>\ ,\emptyset]\ \cup$$
$$[\ (\ u=\sigma^{\omega}\vee u'=\sigma^{\omega})\ ,<\ (u\otimes u')\ ,\omega>\ ,\emptyset]\ \cup$$
$$[\ (\ u\neq\sigma^{\omega}\ \&\ u'\neq\sigma^{\omega}\ \&\ v=\varepsilon\ \&\ v'=\varepsilon)\ ,\ <\ (u\otimes u')\ ,\ R\cup R'\ >\ ,\ \emptyset]$$

D'5 $\quad$ for $u,u'\in\sigma^{\infty}$ and $v,v'\in MN^{\infty}\cup\{\varepsilon\}$,
$<uv,R> \oplus <u'v',R'> = <\varepsilon,\{\sigma\}>\cup$
$<(u\otimes(lf(u')\sigma))v,\ R\ >\cup<((lf(u)\sigma)\ \otimes\ u')v',R'>\cup$
$[(\ u=\sigma^{\omega}\vee\ u'=\sigma^{\omega})\ ,\ <\ (u\sigma\otimes u'\sigma)\ ,\ \omega>\ ,\ \emptyset]\ \cup$
$[(\ u\neq\sigma^{\omega}\ \&\ u'\neq\sigma^{\omega}\ \&\ v=\varepsilon\ \&\ v'=\varepsilon),<\ u\otimes u',\ \{\sigma\}\ >,\ \emptyset\ ]$

D'6 $\quad <w,R>|<w',R'> = \{<w'',R''>\ /\ w''\in\ (w\otimes w')\ \}$,
where $R'' =$ **if** $(\ R=\omega\vee R'=\omega\ )$ **then** ω **else**
if $(\exists\lambda:\lambda\in R\cap\eta(R'))$ **then** $(R\cup R'\cup\{\tau\})$
else $(R\cup R')$

More genuine definitions might have been expected by the reader: the above definitions of the operations ω_k are indeed strongly influenced by our future use of $\mathcal{M}_{int}$ as a springboard towards a more abstract model.

Examples

- $\langle\alpha,\{\beta\}\rangle \mid \langle\overline{\alpha},\{\overline{\beta}\}\rangle = \{\langle\alpha\overline{\alpha},\{\beta,\overline{\beta},\tau\}\rangle, \langle\overline{\alpha}\alpha,\{\beta,\overline{\beta},\tau\}\rangle, \langle\tau,\{\beta,\overline{\beta},\tau\}\rangle\}$
- $\langle\sigma^{\omega},\omega\rangle + \langle\tau,\{\alpha\}\rangle = \{\langle\sigma^{\omega},\omega\rangle\} \cup \langle\sigma^{*}\tau,\{\alpha\}\rangle$
- $\langle\sigma^{2},\{\alpha,\tau\}\rangle + \langle\sigma,\{\beta\}\rangle = \{\langle\sigma^{3},\{\alpha,\tau,\beta\}\rangle\}$
- $\langle\sigma\tau,\{\alpha\}\rangle + \langle\sigma^{2},\{\beta\}\rangle = \{\langle\sigma\tau,\{\alpha\}\rangle, \langle\sigma^{2}\tau,\{\alpha\}\rangle, \langle\sigma^{3}\tau,\{\alpha\}\rangle\}$
- $\langle\sigma^{\omega},\omega\rangle \oplus \langle\tau,\{\alpha\}\rangle = \{\langle\varepsilon,\{\sigma\}\rangle, \langle\sigma^{\omega},\omega\rangle\} \cup \langle\sigma^{*}\sigma\tau,\{\alpha\}\rangle\}$
- $\langle\sigma^{2},\{\alpha,\tau\}\rangle \oplus \langle\sigma,\{\beta\}\rangle = \{\langle\varepsilon,\{\sigma\}\rangle, \langle\sigma^{3},\{\sigma\}\rangle, \langle\sigma^{3},\{\alpha,\tau\}\rangle, \langle\sigma^{4},\{\alpha,\tau\}\rangle, \langle\sigma^{2},\{\beta\}\rangle, \langle\sigma^{3},\{\beta\}\rangle, \langle\sigma^{4},\{\beta\}\rangle\}$.

5.4 Full adequacy

The full adequacy of the model $\mathcal{M}_{int}$ may be stated as follows:

Theorem 2 $\forall t \in PROG : \mathcal{M}_{int}[|t|] = \phi_{int}(\mathcal{M}_{op}[|t|])$.

The proof for that theorem relies upon three propositions.

proposition 5.2 $\forall t \in PROG : \phi_{int}(\mathcal{M}_{op}[|t|])$ is closed and prefix closed, and $\phi_{int}(\mathcal{M}_{op}[|t|]) = CL(\phi_{int}(\mathcal{M}^{*}_{op}[|t|]))$ where $\mathcal{M}^{*}_{op}[|t|]) = Pref(\mathcal{M}_{op}[|t|])$.

proof indication Closedness follows from prefix closedness, since the transition system generated by t has the finite branching property and by Koenig's lemma.

proposition 5.3 For any prefix closed sets Q_i of sequences of transitions $(t_j - v_j \rightarrow t_{j+1})$ between programs:

$(\forall \omega_k \in \Sigma) : \phi_{int}(\omega_k(Q_1 \ldots Q_k)) = \omega_k(\phi_{int}(Q_1) \ldots \phi_{int}(Q_k))$.

proof This arises from a direct comparison between D1-D6 and D'1-D'6

proposition 5.4 a) For any prefix closed sets $Q_i \in D_{int}$ and for every $\omega_k \in \Sigma$, $\omega_k(Q_1 \ldots Q_k)$ is prefix closed, and $CL(\omega_k(Q_1 \ldots Q_k)) = \omega_k(CL(Q_1) \ldots CL(Q_k))$.
b) For any closed and prefix closed sets $Q_i \in D_{int}$ and for every $\omega_k \in \Sigma$, $\omega_k(Q_1 \ldots Q_k)$ is closed and prefix closed, and $Fin(\omega_k(Q_1 \ldots Q_k)) = \omega_k(Fin(Q_1) \ldots Fin(Q_k))$.

Coming back to theorem 2, let us define $\mathcal{M}^{*}_{int}$ in the same way as $\mathcal{M}_{int}$, but with least fixed points in place of greatest fixed points. By proposition 5.3 and by Tarski's least fixed point theorem for ω-continuous functions, $\mathcal{M}^{*}_{int}[|t|] = \phi_{int}(\mathcal{M}^{*}_{op}[|t|])$. In view of proposition 5.2, theorem 2 will be proved if we can show $\mathcal{M}_{int}[|t|] = CL(\mathcal{M}^{*}_{int}[|t|])$. Now, that equality follows from the next lemma with the help of the Bekic' principle for simultaneous fixed points .

lemma Let $\mathbb{X} = \langle X1 \ldots Xn \rangle$ and $\mathbb{F}(\mathbb{X}) = \langle F1(\sigma(X_1), \ldots \sigma(X_n)), \ldots, Fn(\sigma(X_1), \ldots \sigma(X_n)) \rangle$ where the F_i are finite expressions over Σ. Let $\langle Q_1 \ldots Q_n \rangle$ be the least solution of $\mathbb{X}=\mathbb{F}(\mathbb{X})$ in $(D_{int})^n$, then the Q_i are prefix closed and $\langle CL(Q_1) \ldots CL(Q_n) \rangle$ is the greatest solution of $\mathbb{X} = \mathbb{F}(\mathbb{X})$ in $(D_{int})^n$.

proof Let $\langle Y_1 \ldots Y_n \rangle$ be a solution. Clearly, $\langle Fin(Y_1) \ldots Fin(Y_n) \rangle$ is also a solution. Suppose $\langle w, \omega \rangle \in Y_i$. Owing to the definition of the implicit guard operators σ , there exists for every j some finite word $w_j < w$ s.t. $j \leq |w_j|$ and $\langle w_j, R_j \rangle \in \Pi_i \circ \mathbb{F}^j \langle \emptyset \ldots \emptyset \rangle$. Thus, $Y_i \subseteq Cl(Fin(Y_i))$. By proposition 5.4, we are done if we can prove that $\mathbb{F}$ has a unique, and furthermore prefix closed solution in $P(SC^{fin})^n$. We proceed exactly as in the proof given for the lemma in §2.7 .

Theorem 2 and proposition 5.1 show that $\mathcal{M}_{int}$ is the image of the operational model by an erasing morphism which preserves all the informations relevant to the congruence $\approx_1$. Nevertheless, some irrelevant informations are retained, e.g. different meanings are assigned to programs differing only by null actions. We undertake in the next section the research of a second erasing morphism, inducing the fully abstract model as a factor of $\mathcal{M}_{int}$. By the way, we generalize the equivalence $\sim_1$ and the associated congruence $\approx_1$ into corresponding preorders.

6. PREORDERS ON PROGRAMS AND INTERMEDIATE MEANINGS

We successively introduce preorders on programs (§6.1), connect them with preorders on intermediate meanings (§6.2), and prove the agreement between two variants of the observational congruence (§6.3) .

6.1 Preorders on programs

The idea under the observational equivalence $\sim_1$ is that equivalent programs t and u should interact in the same way with any other program v. Since $L(u,v)$ is the language formed of all the possible sequences of interactions between u and v, $t \sim_1 u$ iff $\forall v, L(t,v)=L(u,v)$ is a faithful encoding of that intuition.

A natural extension of the observational equivalence is the observational preorder introduced by the following

definition : $t \lesssim_1 u$ iff $\forall v, L(t,v) \subseteq L(u,v)$.

Here, the idea is that t is smaller than u if every interactive behaviour of t is also an interactive behaviour of u ; intuitively, if you have been put an order for u but have delivered t instead of u then no v can testify against you. But one may complaint about your trick when using u inside contexts, and especially sum contexts (e.g. $[.] + \alpha nil$). In order to avoid any kind of trouble and still optimize the effort, what is needed is that t be an *implementation* of u ($t \sqsubseteq_1 u$) in the following sense :

definition: $t \sqsubseteq_1 u$ iff $\forall C[.], \forall v, L(C[t],v) \subseteq L(C[u],v)$.

The implementation preorder is thus the contextual restriction of the observation preorder, and the equivalence it induces is the observational congruence of section 4.

Examples The following relations, as well as all the equivalence laws of [18] are valid:

- $u \sqsubseteq_1 u \oplus v$, $u \sqsubseteq_1 \tau u$, $\tau(u+v) \sqsubseteq_1 \tau u + \tau v$
- $u + \tau v \sqsubseteq_1 \tau(u+v) + \tau v$, $x \text{ wh } (x=x) \approx_1 x \text{ wh } (x = \tau x)$.
- $\tau u + \tau v \not\sqsubseteq_1 \tau(u+v)$,

  because $\varepsilon \in L(\tau\alpha nil + \tau\beta nil, \overline{\alpha} nil)$ and $\varepsilon \notin L(\tau(\alpha nil + \beta nil), \overline{\alpha} nil)$.
- $u \sim_1 \tau u$ but $u \not\approx_1 \tau u$,

  because $\varepsilon \in L(\alpha nil + \tau nil, \overline{\alpha} nil)$ and $\varepsilon \notin L(\alpha nil + nil, \overline{\alpha} nil)$.

Remark The above examples make it clear why only the maximal computations of (u|v) have been considered while defining $L(u,v)$.

6.2 Preorders on intermediate meanings.

A natural way to find out a fully abstract model for the observational preorder $\lesssim_1$ (i.e. a model for $\sqsubseteq_1$) is to transfer the preorder from programs to their stateless computations and then to analyse the resulting equivalence of sets of intermediate meanings : the canonical homomorphism will conduct us from the intermediate model to the fully abstract model. This program is undertaken in the next series of statements (proofs are postponed until the end of the section).

definition Relation $\lesssim$ is the preorder on stateless computations such that $\langle w,R\rangle \lesssim \langle w',R'\rangle$ if and only if $\langle w,R\rangle \leftarrow^z\rightarrow \langle w'',R''\rangle \Rightarrow \langle w',R'\rangle \leftarrow^z\rightarrow \langle w'',R''\rangle$ for any stateless computation $\langle w'',R''\rangle$. We let $\lesssim$ denote also the Hoare extension of $\lesssim$: $Q_1 \lesssim Q_2 \Leftrightarrow \forall\, q_1 \in Q_1,\ \exists\, q_2 \in Q_2,\ q_1 \lesssim q_2$.

Proposition 6.1 $\langle w,R\rangle \lesssim \langle w',R'\rangle$ if and only if the following conditions hold:

- $\Pi_\Lambda(w) = \Pi_\Lambda(w')$,
- $(R = \omega) \Rightarrow (R' = \omega)$,
- $(R,R' \neq \omega \ \&\ R \cap \{\sigma,\tau\} = \emptyset) \Rightarrow R' \subseteq R$.

Examples.

The following relations and their symmetrical are true :

$\langle \tau\alpha(\sigma\tau)^\omega, \omega\rangle \lesssim \langle \alpha\tau^\omega, \omega\rangle$,

$\langle \alpha\tau, \emptyset\rangle \lesssim \langle \tau\alpha, \emptyset\rangle$

$\langle \varepsilon, \{\tau,\alpha\}\rangle \lesssim \langle \tau, \{\tau,\alpha,\beta\},\rangle \lesssim \langle \varepsilon, \{\tau\}\rangle$.

The following relations are true, but their symmetrical are false :

$\langle \tau\alpha\tau\beta,\ \{\tau\}\rangle \lesssim \langle \alpha\tau\beta,\ \{\alpha,\beta\}\rangle \lesssim \langle \alpha\beta,\ \{\alpha\}\rangle$,

$\langle \alpha\beta,\ \{\alpha\}\rangle \lesssim \langle \alpha\beta,\ \emptyset\rangle \lesssim \langle \alpha\beta\tau^\omega,\ \omega\rangle$.

Proposition 6.2* $\forall u,v \in \mathrm{PROG}$: $(u \lesssim_1 v) \Leftarrow (\mathcal{M}_{\mathrm{int}}[\![u]\!] \lesssim \mathcal{M}_{\mathrm{int}}[\![v]\!]) \Leftarrow (u \sqsubseteq_1 v)$.

Indeed, propositions 6.1 and 6.2 suggest us a model for $\lesssim_1$ and also a way to derive it from $\mathcal{M}_{\mathrm{int}}$, but being aware of the gap between the observational preorder and the implementation preorder (due to sum contexts as

shown earlier), we have to fill in that gap and to guess the proper transposition of $\precsim_1$ to intermediate meanings. The idea is to take into account the initial internal actions of programs, whence the following definition.

Definition Relation $\sqsubseteq$ is the preorder on stateless computations such that $\langle w,R\rangle \sqsubseteq \langle w',R'\rangle$ iff the following conditions hold :

- $\langle w,R\rangle \lesssim \langle w',R'\rangle$
- $(\Pi_\Lambda(w)=\varepsilon \;\&\; R\neq\omega \;\&\; R'\neq\omega \;\&\; \{\sigma,\tau\}\cap(R\cup R')=\emptyset) \Rightarrow (\Pi_\tau(w)\neq\varepsilon \Rightarrow \Pi_\tau(w')\neq\varepsilon)$.

We let $\sqsubseteq$ denote also the Hoare extension of $\sqsubseteq$.

Examples.

$\langle\tau,\ \{\tau,\alpha\}\rangle \sqsubseteq \langle\varepsilon,\ \{\tau,\alpha,\beta\}\rangle \sqsubseteq \langle\varepsilon,\{\sigma\}\rangle$ and

$\langle\varepsilon,\ \{\alpha,\beta\}\rangle \sqsubseteq \langle\tau,\ \{\alpha\}\rangle \sqsubseteq \langle\sigma^\omega,\omega\rangle$ but

$\langle\tau,\ \{\alpha\}\rangle \not\sqsubseteq \langle\varepsilon,\emptyset\rangle$.

Next proposition shows that the attempted transposition is at least half successful.

Proposition 6.3 * $\forall u,v \in PROG : \; u \lesssim_1 v \;\Leftarrow\; \mathcal{M}_{int}[|u|] \sqsubseteq \mathcal{M}_{int}[|v|] \;\Leftarrow\; u \sqsubseteq_1 v$

One of the goals of the forthcoming section 8 is to establish the implication $\mathcal{M}_{int}[|u|] \sqsubseteq \mathcal{M}_{int}[|v|] \Rightarrow u \sqsubseteq_1 v$. Provided that it holds, and seeing the definition of $\sqsubseteq$,our implementation preorder $\sqsubseteq_1$ on programs turns out to be an infinitary variant of the preorder studied in [20,21,22]. Nevertheless, our implementation preorder $\sqsubseteq_1$ makes a total confusion between static divergence (due to unguarded recursive definitions) and dynamic divergence (due to infinite sequences of internal actions), and also a partial confusion between potential divergence and systematic divergence. We identify for instance (x wh x=τx) and ((x+τNIL) wh x=τx), although the former program may terminate steadily whereas the latter always diverges. These programs are not identified in [21,22]. The reason why our implementation preorder is not so fine is the absence of a sequential composition in our version of CCS. We prove in the next alinea that $\sqsubseteq_1$ captures precisely those distinctions which can be evidenced by CCS tests.

6.3 An alternative definition of the implementation preorder

The idea under the equivalence $\sim_3$ introduced in section 4 is that, given a pair of equivalent programs t and u, any program v behaves exactly in the same way wether it interacts with t or with u. A contextual preorder $\lesssim_3$ generalizing that equivalence may be defined as follows:

$t \lesssim_3 u$ iff $\forall C[.], \forall v: \Theta(C[t],v) \subseteq \Theta(C[u],v)$

where $\Theta(p,q)$ is the set of computations equal to

$\{\theta_q \in \mathcal{M}_{op}[|q|] \ / \ (\theta_p,\theta_q) \in \text{Twincomp for some } \theta_p \in \mathcal{M}_{op}[|p|]\}$.

Let us show $\lesssim_3 = \lesssim_1$ (and thus $\approx_3 = \approx_1$).

Clearly, $\lesssim_3 \subseteq \lesssim_1$. In the reverse direction: $t \lesssim_1 u \Rightarrow C[t] \lesssim_1 C[u]$ for all contexts, and $p \lesssim_1 q \Rightarrow \Theta(p,v) \subseteq \Theta(q,v)$ for all v, by propositions 6.1 and 6.2. Hence, $\lesssim_1 \subseteq \lesssim_3$ and the two preorders coincide.

7. AN OBSERVATIONAL MODEL

There emerges more or less from proposition 6.3 that the canonical homomorphism ϕ induced by the equivalence $\approx$ on D_{int} (defined as $\lesssim \cap \lesssim^{-1}$) bridges the intermediate model $\mathcal{M}_{int}$ and another model $\mathcal{M}_{obs}$ fully abstract for the observational preorder $\lesssim_1$ (on programs). Following that suggestion, we construct in §7.1 and §7.2 an abstract domain D_{obs} and an associated interpretation for the Σ operators, so that ϕ appears as a continuous homomorphism between two continuous Σ algebras. In spite of that, we encounter in §7.3 some difficulties for representing the ϕ images of greatest fixed points. This leads us to suggest in §7.4 a polymorphic meaning function $\mathcal{M}$ combining the use of D_{int} and D_{obs} (for open, resp. closed terms). The full abstractness of $\mathcal{M}$ is studied in §7.5.

7.1 The abstract domain D_{obs}

Henceforth in the paper, $\perp$ is an abbreviation for $\{\sigma,\tau\}$ and the pairs $\langle w,\omega\rangle$ and $\langle w\sigma^{\omega},\omega\rangle$ are identified. Our first definition suggests representations for the equivalence classes of stateless computations.

definition ϕ_{abs}: SC $\rightarrow$ SC is the forgetful function s.t. :

- $\phi_{abs}\langle w,\omega\rangle = \langle\Pi_{\Lambda}(w),\omega\rangle$,
- $\phi_{abs}\langle w,R\rangle = \langle\Pi_{\Lambda}(w),\perp\rangle$ if $R\cap\{\sigma,\tau\}\neq\emptyset$,
- $\phi_{abs}\langle w,R\rangle = \langle\Pi_{\Lambda}(w),R\rangle$ if $R\subseteq\Lambda$ & $(\Pi_{\Lambda}(w)\neq\varepsilon \vee \Pi_{\tau}(w)=\varepsilon)$,
- $\phi_{abs}\langle w,R\rangle = \langle\tau,R\rangle$ if $R\subseteq\Lambda$ & $(\Pi_{\Lambda}(w)=\varepsilon \,\&\, \Pi_{\tau}(w)\neq\varepsilon)$.

Examples

- $\phi_{abs}\langle(\alpha\tau)^{\omega},\omega\rangle = \langle\alpha^{\omega},\omega\rangle$,
- $\phi_{abs}\langle\alpha\sigma\alpha,\{\sigma,\alpha\}\rangle = \langle\alpha\alpha,\perp\rangle$,
- $\phi_{abs}\langle\tau\alpha\tau\sigma,\{\alpha,\beta\}\rangle = \langle\alpha,\{\alpha,\beta\}\rangle$,
- $\phi_{abs}\langle\tau\sigma\tau\sigma,\{\alpha,\beta\}\rangle = \langle\tau,\{\alpha,\beta\}\rangle$.

proposition 7.1 For all stateless computations $\langle w,R\rangle$ and $\langle w',R'\rangle$,
$\langle w,R\rangle \precsim \langle w',R'\rangle \Leftrightarrow \phi_{abs}\langle w,R\rangle \precsim \phi_{abs}\langle w',R'\rangle$.

definition The poset of *abstract computations* $(AC,\precsim)$ is the restriction of $(SC,\precsim)$ to the set $AC = \phi_{abs}(SC)$.

An *abstract computation* is thus a pair of one of the forms $\langle W,\omega\rangle$ or $\langle w,R\rangle$ or $\langle w,\perp\rangle$ or $\langle\tau,R\rangle$ where $W\in\Lambda^{\infty}$, $w\in\Lambda^{*}$ and R is a finite subset of Λ. Further, $\precsim$ is the least order relation satisfying the following axioms, where $w\in\Lambda^{*}$ and $R''\subseteq R'\subseteq R$:

- $\langle w,\perp\rangle \precsim \langle w,R\rangle \precsim \langle w,R'\rangle \precsim \langle w,\omega\rangle$,
- $\langle\varepsilon,R'\rangle \precsim \langle\tau,R'\rangle \precsim \langle\tau,R''\rangle \precsim \langle\varepsilon,\omega\rangle$.

We introduce now a set D_{obs} of representations for the equivalence classes of sets of stateless computations.

definition $\phi : D_{int} \to D_{int}$ is the forgetful function such that

- $\phi(Q) = top\{\phi_{abs}(q) \ / \ q \in Q \}$,

where top : $P(AC) \to P(AC)$ is the function such that

- $top(Q) = \{q \in Q \ / \ \forall\, q' \in Q \ q \sqsubseteq q' \Rightarrow q' = q\}$

(remark that $\phi = \phi^2$).

proposition7.2 For all sets $Q, Q' \in D_{int}$, $Q \lesssim Q' \Leftrightarrow \phi(Q) \sqsubseteq \phi(Q')$.

definition The poset of *observations* $(D_{obs}, \sqsubseteq)$ is the restriction of $(D_{int}, \lesssim)$ to the set $D_{obs} = \phi(D_{int})$.

Thus, an observation is a set of pairwise incomparable stateless computations; observations are ordered along the Hoare extension of $\sqsubseteq$ on AC.

Next proposition shows that D_{obs} is indeed a domain.

proposition 7.3 $(D_{obs}, \sqsubseteq, \emptyset)$ is a complete upper semi-lattice, with least upper bounds of subsets given by $\bigsqcup D = top(\cup Q \ / Q \in D)$.

proof It is clearly enough to show : $(\forall Q' \in D)\ (Q' \sqsubseteq top(\cup\, Q \ / \ Q \in D))$; and this is true, since there cannot exist, in view of the axioms of $\sqsubseteq$, any strictly increasing chain of abstract computation in $(\cup\, Q \ / \ Q \in D)$.

Our intend is to use D_{obs} as the domain of a denotational model in which $\phi(\mathcal{M}_{int}[\![t]\!])$ is the meaning assigned to program t. Before we proceed in that direction, let us briefly refer the intended model to the readiness model of [29] [6]. We recall that a ready set is the set of actions a stable program can perform immediately. The set Ready(t) is equal to Interface(t) if that interface set does not include τ (or σ), and is undefined otherwise. When $R \neq \omega$, abstract computations $\langle w,R \rangle$ are indeed ready pairs, with trace w and ready set R. As regards the observational behaviour of a particular program t, the axiomatisation of (AC, $\sqsubseteq$) shows that a ready pair $\langle w,R \rangle$ can be neglected if t may diverge after w or may reach after w some stable state with a strictly smaller ready set $R' \subseteq R$. The axioms also show that all the infinite computations should be represented as such by corresponding traces.

7.2 An interpretation for $\Sigma \cup \{\sigma\}$ in D_{obs}

We introduce first elementary operations $\omega_k : AC^k \to P(AC)$ such that the following diagram commutes for all ω_k :

$$\begin{array}{ccc} SC^k & \xrightarrow{\omega_k} & P(SC) \\ \downarrow \phi_{abs}{}^k & & \downarrow \phi_{abs}{}^k \\ AC^k & \xrightarrow{\omega_k} & P(AC) \end{array}$$

Those operations on abstract computations are defined by the following relations D"1-D"6 (where some brackets are omitted for singleton sets).

D"1 $\quad nil = \{<\varepsilon, \emptyset>\}$,

D"2 $\quad \upsilon<w,R> = [\upsilon \in \Lambda\ ,\ <\varepsilon, \{\upsilon\}>,\ <\varepsilon, \perp>] \cup$

$\underline{if}\ (\upsilon = \sigma)\ \underline{then}\ <w,R>$

$\underline{else\ if}\ (R \in \{\perp, \omega\} \vee w \notin \{\varepsilon, \tau\})$

$\underline{then}\ [\upsilon = \tau\ ,\ <w,R>,\ <\upsilon w,R>]$

$\underline{else}\ [\upsilon = \tau,\ <\tau,R>,\ <\upsilon,R>]$

D"3 $\quad <w,R>\rho = \underline{if}\ w \notin (dom(\rho)^\infty)\ \underline{then}\ \emptyset\ \underline{else}$

$[R \in \{\tau, \omega\},\ <\rho(w),R>,\ <\rho(w),\ \rho(R \cap dom(\rho))>]$

D"4 $\quad <w,R> + <w',R'> = [(w \neq \varepsilon \vee R = \omega),\ <w,R>,\ \emptyset] \cup$

$[(w' \neq \varepsilon \vee R' = \omega),\ <w',R'>,\ \emptyset] \cup$

$\underline{if}\ (w = \varepsilon\ \&\ R \neq \omega\ \&\ w' = \varepsilon\ \&\ R' \neq \omega)$

$\underline{then}\ [(R = \perp \vee R' = \perp),\ <\varepsilon, \perp>,\ <\varepsilon,\ R \cup R'>]$,

D"5 $\quad <w,R> \oplus <w',R'> = \{<\varepsilon, \perp>,\ <w,R>,\ <w',R'>\}$,

D"6 $\quad <w,R> \mid <w',R'> = \underline{if}\ (R = \omega \vee R' = \omega)\ \underline{then}\ \{<w'', \omega> / \ w'' \in \Pi_\Lambda(w \otimes w')\}$

$\underline{else}\ \underline{if}\ (R = \perp \vee R' = \perp \vee (\exists \lambda, \lambda \in R \cap \eta(R'))$

$\underline{then}\ \{<w'', \perp> \ / \ w'' \in \Pi_\Lambda\ (w \otimes w')\}$

$\underline{else}\ \{<h(w''),\ R \cup R'> \ / \ w'' \in (w \otimes w')\}$

where $h(w'') = [\ \Pi_\Lambda(w'') \neq \varepsilon,\ \Pi_\Lambda(w''),\ [\Pi_\tau(w'') \neq \varepsilon,\ \tau,\ \varepsilon]\ ]$.

---

proposition 7.4 The operations $\omega_k : AC^K \to P(AC)$ are monotone with respect to $\sqsubseteq$ (on AC) and its Hoare extension (on $P(AC)$).

Examples (continued from section 5)

- $\langle\alpha,\{\beta\}\rangle \mid \langle\overline{\alpha},\{\overline{\beta}\}\rangle = \{\langle\alpha\overline{\alpha},\perp\rangle\ ,\ \langle\overline{\alpha}\alpha,\perp\rangle\ ,\ \langle\varepsilon,\perp\rangle\}$,
- $\langle\varepsilon,\omega\rangle + \langle\tau,\{\alpha\}\rangle = \{\langle\varepsilon,\omega\rangle,\ \langle\tau,\{\alpha\}\rangle\}$,
- $\langle\varepsilon,\perp\rangle + \langle\varepsilon,\{\beta\}\rangle = \{\langle\varepsilon,\perp\rangle\}$,
- $\langle\tau,\{\alpha\}\rangle + \langle\varepsilon,\{\beta\}\rangle = \{\langle\tau,\ \{\alpha\}\rangle\}$,
- $\langle\tau,\{\alpha\}\rangle + \langle\tau,\{\beta\}\rangle = \{\langle\tau,\ \{\alpha\}\rangle,\ \langle\tau,\{\beta\}\rangle\}$,
- $\langle\varepsilon,\omega\rangle \oplus \langle\tau,\{\alpha\}\rangle = \{\langle\varepsilon,\perp\rangle,\ \langle\varepsilon,\omega\rangle,\ \langle\tau,\{\alpha\}\rangle\}$
- $\langle\varepsilon,\perp\rangle \oplus \langle\varepsilon,\{\beta\}\rangle = \{\langle\varepsilon,\perp\rangle,\ \langle\varepsilon,\{\beta\}\rangle\}$.

We now lift the operations ω_k from AC to D_{obs} and show some general properties of the resulting structure.

definition $(D_{obs},\ \Sigma\cup\{\sigma\})$ is the algebra with carrier set D_{obs} and operations $\omega_k : (D_{obs})^k \to D_{obs}$ given as $\omega_k(Q_1 \ldots Q_k) = top(\cup\{\omega_k(q_1 \ldots q_k)\ /q_i \in Q_i\})$, with the exceptional case $\sigma\emptyset = \{\langle\varepsilon,\perp\rangle\}$.

proposition 7.5 $(D_{obs},\ \Sigma\cup\{\sigma\},\ \sqsubseteq\ )$ is a continuous algebra.

proof indication The continuity of the operators ω_k stems from relations such as $q \in \omega_2(\bigsqcup D, Q') \Rightarrow \exists\, Q \in D,\ q \in \omega_2(Q,Q'))$, which is an easy consequence of propositions 7.3 and 7.4.

proposition 7.6 Function ϕ is a continuous homomorphism between the continuous algebras $(D_{int},\ \Sigma\cup\{\sigma\},\ \subseteq)$ and $(D_{obs},\ \Sigma\cup\{\sigma\},\ \sqsubseteq\ )$.

proof This proposition is a direct corollary of propositions 7.2, 7.3 and 7.5.

In the sequel, $\mathcal{D}_{int}$ denotes the subset of D_{int} with elements Q such that either $\{w/\langle w,R\rangle \in Q\}$ or $\Omega\{w/\langle w,R\rangle \in Q\}$ is prefixed closed, letting $\Omega(\tau)=\varepsilon$, $\Omega(u\sigma^\omega)=u$ for $u \in \Lambda^*$, and $\Omega(w) = w$ in all the other cases. Thus Q and ϕQ are both in $\mathcal{D}_{int}$ if Q is a prefixed closed set in D_{int}.

7.3 Where fixed points are altered by morphisms.

Suppose we know a closure operation CL on D_{obs} making the following assertions true:

a. for any function $\mathbb{F}(X_1 \ldots X_n)=<F_1(\sigma X_1,\ldots\sigma X_n),\ldots F_n(\sigma X_1,\ldots\sigma X_n)>$, where the F_i are finite Σexpressions, the greatest solution of $\mathbb{X}=\mathbb{F}(\mathbb{X})$ in $(D_{obs}, \sqsubseteq)^n$ is the componentwise closure of its least solution;

b. for all t in PROG, $\phi(CL(Fin(\mathcal{M}_{int}[|t|])))$ is the closure of $\phi(Fin(\mathcal{M}_{int}[|t|]))$.

By simply changing the interpretation for the Σ operators in the definition of $\mathcal{M}_{int}$, the latter might be turned into a meaning function $\mathcal{M}_{obs}$ satisfying $\mathcal{M}_{obs}[|t|] = \phi(\mathcal{M}_{int}[|t|])$ for all t in PROG. This is indeed the method we have followed for deriving $\mathcal{M}_{int}$ from $\mathcal{M}_{op}$ through morphism ϕ_{int} .

Unfortunately, the method is not correct for deriving $\mathcal{M}_{obs}$ from $\mathcal{M}_{int}$: as we shall see, the relation $\phi(\text{Fix } F)=\text{Fix}(\phi F)$ is not valid in that framework, although a similar relation holds for least fixed points (by proposition 7.6).

In fact, the assertions a and b cannot be satisfied together, as is evidenced by two remarks:

a'. the least solution of $X=\sigma X$ in D_{obs} is $< \varepsilon,\perp >$ whereas the greatest solution is $\{<w,\omega>/w\in\Lambda^\infty\}$, thus $CL\{< \varepsilon,\perp >\} = < \Lambda^\infty,\omega>$ if a is valid;

b'. by setting t=x wh (x=x) in b, we get the equality between $<\varepsilon,\omega>$ and $CL<\varepsilon,\perp>$ and thus the assertions a and b cannot hold together, for $<\varepsilon,\omega> \neq <\Lambda^\infty,\omega>$

In a more general register, the inequality between $<\varepsilon,\omega>$ and $<\Lambda^\infty,\omega>$ entails the invalidity of the relation $\phi(\text{Fix } F)=\text{Fix}(\phi F)$, because $<\varepsilon,\omega> = \phi(\text{Fix}_X(\sigma X))$ and $<\Lambda^\infty,\omega> = \text{Fix}_X(\phi(\sigma X))$.

7.4 A polymorphic meaning function $\mathcal{M}$

An alternative method for extracting from $\mathcal{M}_{int}$ a meaning function $\mathcal{M}_{obs}$ (hopefully) satisfying $(\forall t\in PROG)\ \mathcal{M}_{obs}(t) = \phi\mathcal{M}_{int}(t)$ is to modify the definition of $\mathcal{M}_{int}$ in the following way:

- everywhere in the definition, substitute $\Phi\mathcal{M}_{int}(t)$ for $\mathcal{M}_{int}(t)$ provided that t is a closed term.

Since D_{obs} is a part of D_{int}, the above substitutions make sense. The result of the construction is a meaning function $\mathcal{M}$: TERM$\rightarrow((V\rightarrow D_{int})\rightarrow D_{int})$, which assigns constant values $\mathcal{M}[|t|] : (V\rightarrow D_{int})\rightarrow D_{obs}$ to closed terms t. As for $\mathcal{M}_{obs}$, it is the restriction of $\mathcal{M}$ to the set PROG of closed terms.

The above ideas are put in practice in the following statements, where $\mathcal{M}$ is given an inductive and polymorphic definition. Constant functions are identified with their constant values, $\omega_k{}^{obs}$ stands for $\omega_k: {D_{obs}}^k \rightarrow D_{obs}$, $\omega_k{}^{int}$ stands for $\omega_k: {D_{int}}^k \rightarrow D_{int}$, $e \in (V\rightarrow D_{int})$, and Fix denotes greatest fixed points in $(D_{int},\subseteq)^n$.

THE INDUCTIVE SPECIFICATION OF $\mathcal{M}$

1. t is a variable $x \in V$

   $\mathcal{M}[|t|](e) = \sigma^{int}(e(x))$

2. t is a closed term $\omega_k(t_1 \ldots t_k)$

   $\mathcal{M}[|t|] = \omega_k{}^{obs}(\mathcal{M}[|t_1|], \ldots \mathcal{M}[|t_k|])$

3. t is an open term $\omega_k(t_1 \ldots t_k)$ with free variables

   $\mathcal{M}[|t|](e) = \omega_k{}^{int}(\mathcal{M}[|t_1|](e), \ldots \mathcal{M}[|t_k|](e))$

4. t is an open term t'wh D with free variables and D is $(x_1{=}t_1, \ldots, x_n{=}t_n)$:

   $\mathcal{M}[|t|](e) = \mathcal{M}[|t'|](e[\mathbb{V}/\mathbb{x}])$

   letting $\mathbb{V} = Fix_{\mathbb{X}}\, \mathbb{F}(e[\mathbb{X}/\mathbb{x}])$, $F_i(e) = \mathcal{M}[|t_i|](e)$

   and $\mathbb{x} = (x_1 \ldots x_n)$, $\mathbb{X} = (X_1 \ldots X_n)$, $\mathbb{F} = (F_1 \ldots F_n)$.

5. t is closed term t' wh D

   $\mathcal{M}[|t|] = \phi\mathcal{M}[|t'|](e[\mathbb{V}/\mathbb{x}])$

   with $\mathbb{V}$ defined as above .

The full adequacy of the meaning function $\mathcal{M}$ is stated by the following theorem.

theorem 3 $\forall t \in PROG : \mathcal{M}[|t|] = \phi\mathcal{M}_{int}[|t|]$.

The above is an immediate consequence of proposition 7.6 and the following assertion HOM, the proof of which is slightly deferred.

HOM : For any set $U \in \mathcal{D}_{int}$ and for any finite Σ expressions F_i, let $\mathbf{F}(U)(X_1 \ldots X_n) = \langle F_1(U, \sigma X_1 \ldots \sigma X_n), \ldots F_n(U, \sigma X_1 \ldots \sigma X_n)\rangle$ then

$\phi Fix_{\mathbf{X}} \mathbf{F}(U) = \phi Fix_{\mathbf{X}} F(\phi U)$.

The general derivation method which was suggested here works indeed as soon as the equality $\phi\, Fix_{\mathbf{X}} \mathbf{F}(\mathcal{M}_{int}[|u|]) = \phi Fix_{\mathbf{X}} F(\phi\mathcal{M}_{int}[|u|])$ is valid for closed terms.

remark The Beki'c principle for simultaneous fixed points is valid in $\mathcal{M}$ (this follows clearly from HOM) .

7.5 Full abstraction

Theorem 3 states the full adequacy of $\mathcal{M}$ (w.r.t. ϕ), but it does not say anything about the full abstractness of $\mathcal{M}$ (w.r.t. $\lesssim_1$). We recall from [27] that full abstraction means the logical equivalence :

- $\forall u, v \in PROG$: $\mathcal{M}[|u|] \sqsubseteq \mathcal{M}[|v|] \Leftrightarrow (\forall C[.])\ C[u] \lesssim_1 C[v]$.

The full abstractness of $\mathcal{M}$ is claimed by the following statement:

Theorem 4 $\forall u, v \in PROG$: $\mathcal{M}[|u|] \sqsubseteq \mathcal{M}[|v|] \Leftrightarrow u \precsim_1 v$.

From propositions 6.3, 7.2 and theorem 3, we know that the right to left implication is valid. There remains to establish the reverse relation. In view of proposition 6.2, the left to right implication is an immediate consequence of the following claim:

- for any programs u,v and for any program context C[.]:

$$\mathcal{M}[|u|] \sqsubseteq \mathcal{M}[|v|] \Rightarrow \mathcal{M}[|C[u]|] \sqsubseteq \mathcal{M}[|C[v]|].$$

By proposition 7.4 and theorem 3, that assertion follows from the next property MON with the help of the Beki'c principle for simultaneous fixed points.

MON For any sets $U,V \in \mathcal{D}_{int}$ and for any finite Σ expressions F_i, let $\mathbf{F}(U)(X_1 \ldots X_n) = \langle F_1(U,\sigma X_1 \ldots \sigma X_n), \ldots F_n(U,\sigma X_1 \ldots \sigma X_n)\rangle$ then

$$\phi U \sqsubseteq \phi V \Rightarrow \phi \mathrm{Fix}_{\mathbf{X}} \mathbf{F}(U) \sqsubseteq \phi \mathrm{Fix}_{\mathbf{X}} F(V) .$$

The goal of section 8 is to prove the fundamental property MON, which obviously entails HOM.

8. ABOUT INFINITE Σ TREES.

The fundamental relation MON, which is the cornerstone of our fully abstract model, may be equivalently expressed as:

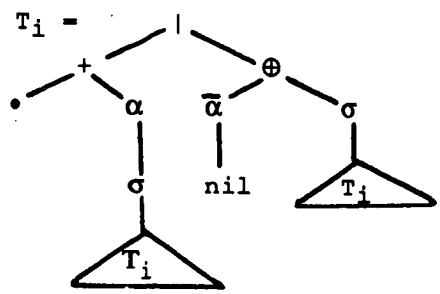
$\forall U,V \in \mathcal{D}_{int}$: $U \sqsubseteq V \Rightarrow \mathrm{Fix}_{\mathbf{X}} \mathbf{F}(U) \sqsubseteq \mathrm{Fix}_{\mathbf{X}} F(V)$.

That relation is not immediate : it states the monotony w.r.t. preorder $\sqsubseteq$ of functions defined by means of greatest fixed points w.r.t. inclusion, and $\subseteq$ is strictly included in $\sqsubseteq$. Our first step towards a proof of MON is to state, for that property of fixed points and (extended) operators ω_k on $\mathcal{D}_{int}$ (= $P(SC)$), an equivalent formulation in terms of infinite trees and (elementary) operators ω_k on SC. This is accomplished in §8.1 to §8.4. At this stage, we observe that the resulting assertion concerns only the trace component of stateless computations, i.e. the first projection of pairs $\langle w,R\rangle$. We construct in §8.5 an interpretation for the Σ operators in a simplified domain of trace languages, and finally rephrase our assertion in that framework in §8.6. The proof for the final assertion is given in appendix.

In all the section, we let fix $\mathbf{F}(U)$, resp. Fix $\mathbf{F}(U)$, denote the least, resp. the greatest, fixed point in $(X_1 \ldots X_n)$ of $\mathbf{F}(U)(X_1 \ldots X_n)$, where $\mathbf{F}(U)(X_1 \ldots X_n) = \langle F_1(U, \sigma X_1 \ldots \sigma X_n), \ldots F_n(U, \sigma X_1 \ldots \sigma X_n) \rangle$ and functions F_i are induced by finite term contexts $t_i[.]$ with free variables $(x_1 \ldots x_n)$.

8.1 Unfolding trees

Terms $t[\bullet]$ in the free Σ algebra over generators $\{\bullet, x_1, \ldots x_n\}$ may be isomorphically identified with finite trees . For $1 \leq i \leq n$ let T_i be the unfolding tree that arises from $t_i[\bullet]$ through the continuous expansion of the variables x_j, $1 \leq j \leq n$, into corresponding trees $\sigma(t_j[\bullet])$. Then T_i is an infinitary rational tree on the (extended) signature $\Sigma \cup \{\sigma\}$ and the (reduced) set of variables $\{\bullet\}$, and every infinite branch of T_i has infinitely many σ-nodes. A typical example is the following tree T_i, obtained from $t_i[\bullet] = (\bullet + \alpha(x_i)) \mid (\overline{\alpha}(nil) \oplus x_i)$



We assume in the sequel the usual representation of trees by partial functions from $\mathbb{N}^*$ to $\Sigma \cup \{\sigma\}$.

8.2 SC-assignments for Σ-trees.

Given an unfolding tree T and a set U in $\mathcal{D}_{int}$, we let U^T denote the set of the U-assignments of T, defined as the partial functions f:dom(T)→SC that assign computations to nodes of T in agreement with the following rules:

- ε (the root node) ∈ dom(f) ;
- if s ∈ dom(f) and T(s) = • then f(s) ∈ U ;
- if s ∈ dom(f) and T(s) = nil then f(s) = <ε,∅> ;
- if s ∈ dom(f) and T(s) = ω_k, k > 0, then

either $f(s) \in \omega_k(f(s.1),\ldots f(s.k))$, or $\omega_k = \sigma$ and $f(s) = \langle \varepsilon, \{\sigma\}\rangle$.

A partial assignment $f \in U^T$ is said to be *finite* if dom(f) is finite, *infinite* if dom(f) is infinite. A finite assignment f is *finitary* if ∀s ∈ dom(f): $f(s) = \langle w,R\rangle \Rightarrow R \neq \omega$. An infinite assignment $f \in U^T$ is *productive* if $f(\varepsilon) \neq g(\varepsilon)$ for every finite assignment $g \in U^T$. Clearly, for any productive assignment f, $f(\varepsilon) = \langle w,R\rangle \Rightarrow R = \omega$.

8.3 Fixed points as sets of assignments

Since σ(∅) is the singleton set < ε,{σ} >, the following equality appears as an immediate consequence of Tarski's least fixed point theorem:

$\Pi_i(\text{fix } \mathbf{F}(U)) = \{ f(\varepsilon) \;/\; f \in U^{T_i} \;\&\; f \text{ is finite}\}$.

Now for any fixed $f \in U^{T_i}$, let $V_j = \{ f(s) \;/ s \in S_{ij} \}$, where S_{ij} is the set of the nodes $s \in \text{dom}(T_i)$ which represent occurences of x_j, then $\langle V_1 \ldots V_n\rangle$ is componentwise included in $\mathbf{F}(U)$ and hence in Fix $\mathbf{F}(U)$ which is the greatest post fixed point of $\mathbf{F}(U)$. Hence we can claim:

$\Pi_i(\text{ Fix } \mathbf{F}(U)) = \{ f(\varepsilon) \;/\; f \in U^{T_i}\}$.

From the above, there emerges:

Π_i(Fix $\mathbf{F}(U)$ - fix $\mathbf{F}(U)$) = { f(ε) / f ∈ $U^T{}_i$ & f is productive}

---

where "-" is the componentwise difference between vectors.

8.4 Towards a proof of relation MON

By Tarski's least fixed point theorem, the following variant of relation MON holds, because ϕ is a continuous homomorphism between two continuous Σ-algebras: $\forall\ U,V \in \mathcal{D}_{int}$: $\phi U \sqsubseteq \phi V \Rightarrow \phi$ fix $\mathbf{F}(U) \sqsubseteq \phi$ fix $\mathbf{F}(V)$.
The above may be equivalently expressed as :

$\forall\ U,V \in \mathcal{D}_{int}$: $U \sqsubseteq V \Rightarrow$ fix $\mathbf{F}(U) \sqsubseteq$ fix $\mathbf{F}(V)$.

Relation MON will therefore follow if we can show:

$\forall\ U,V \in \mathcal{D}_{int}$: $U \sqsubseteq V \Rightarrow$ Fix $\mathbf{F}(U)$ - fix $\mathbf{F}(U) \sqsubseteq$ fix $\mathbf{F}(V)$.

In view of §8.3, this last implication is entailed by the following claim:

Claim1 For any set $U \in \mathcal{D}_{int}$, there exists a class Γ of U-assignments for unfolding trees T with the following properties:
- all the productive U-assignments are in Γ,
- ϕ(f(ε) / f ∈ (Γ ∩ U^T)) = { <w,ω> / w ∈ $K_T(U)$ },
- $K_T(U)$ is an ⊆-increasing function of Π_Λ(w / <w,R> ∈ U) .

8.5 Trace assignments for Σ trees

It may be observed that the above claim has nothing to do with the second projection of stateless computations, and may thus be rephrased in terms of traces. Fortunately, the defining relations which have been stated in §5.3 for operations ω_k : $SC^k \rightarrow P(SC)$ induce corresponding operations ω_k: $(N^\infty)^k \rightarrow P(N^\infty)$ s.t. for all stateless computations, $<w_1,R_1>,\ldots<w_k,R_k>$: $\omega_k(w_1 \ldots w_k)$ = { w / ∃R $<w,R> \in \omega k(<w_1,R_1>,\ldots<w_k,R_k>)$ } . Using these derived operators on traces,

let us weaken the previous definitions given for SC-assignments into corresponding definitions for trace assignments, by simply dropping the second projection of pairs <w,R> . Our previous claim may be rephrased into the following.

Claim2 For any set $U \subseteq N^{\infty}$ s.t. either U or $\Omega(U)$ -see §7.2- is prefix closed, there exists a class Γ of U-assignments for unfolding trees T with the following properties:

- all the productive U-assignments are in Γ,
- $\Pi_{\Lambda}\{ f(\varepsilon) \ / \ f \in (\Gamma \cap U^{T}) \} = K_{T} (\Pi_{\Lambda}(U))$,
- K_{T} is increasing for $\subseteq$.

As we shall see now, the class Γ of the claim may even be reduced to a class of productive assignments, to the cost of a slight modification of the operations ω_{k} on traces.

8.6 Subscripted actions

The modification amounts to supply actions with subscripts which identify agents in unfolding trees and are used to indicate the logical origin of the actions.

Modified operations

Let $\mathbb{N} = N \cup \{ \upsilon_{s} \ / \ v \in N \ \& \ s \in \mathbb{N}^{+} \} \cup \{ \tau_{<s,s'>} \ / \ s,s' \in \mathbb{N}^{+} \}$.

For $s \in \mathbb{N}^{*}$ and $\omega_{k} \in \Sigma$, we introduce operations $\sigma_{s} : \mathbb{N}^{\infty} \rightarrow P (\mathbb{N}^{\infty})$: $\sigma_{s}(w) = \{\varepsilon, \sigma_{s} w\}$ and new operations $\omega_{k} : (\mathbb{N}^{\infty})^{k} \rightarrow P(\mathbb{N}^{\infty})$, defined by variants of relations D'1-D'6 where the second projections are dropped and the following changes are made:

- u resp. u',v,v' are replaced in the statement of conditions by δu resp. $\delta u', \delta v, \delta v'$ where δ denotes the subscript erasing morphism (from $\mathbb{N}^{\infty}$ to N^{∞}),
- the parallel composition $\otimes$ is adapted in such a way that the following holds for $v,v' \in \mathbb{N}$ and $s,s' \neq \varepsilon$:

$\upsilon w \otimes \upsilon' w' = (\upsilon(w \otimes \upsilon' w') \cup \upsilon'(\upsilon w \otimes w')) \cup$

$\underline{\text{if}}\ \upsilon = \lambda\ \&\ \upsilon' = \eta(\lambda)\ \underline{\text{then}}\ \tau(w \otimes w')\ \underline{\text{else}}$

$\underline{\text{if}}\ \upsilon = \lambda_s\ \&\ \upsilon' = \eta(\lambda)\ \underline{\text{then}}\ \tau_s(w \otimes w')\ \underline{\text{else}}$

$\underline{\text{if}}\ \upsilon = \lambda\ \&\ \upsilon' = \eta(\lambda)_{s'}\ \underline{\text{then}}\ \tau_{s'}(w \otimes w')\ \underline{\text{else}}$

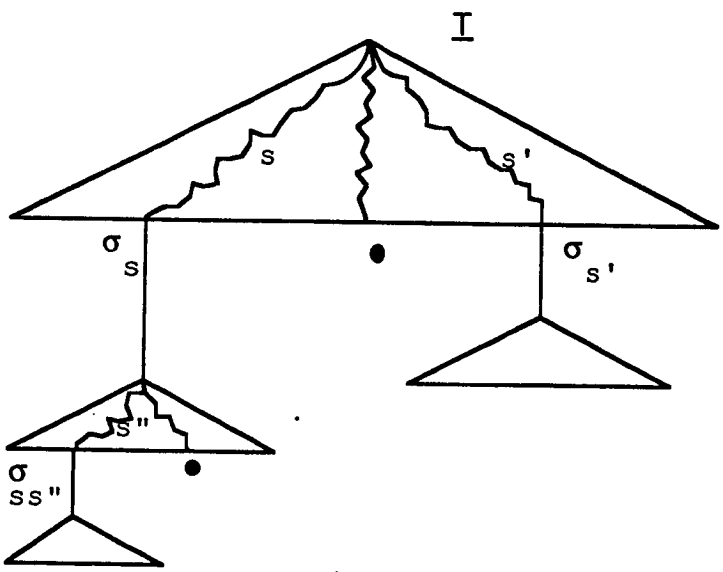
$\underline{\text{if}}\ \upsilon = \lambda_s\ \&\ \upsilon' = \eta(\lambda)_{s'}\ \underline{\text{then}}\ \tau_{<s,s'>}(w \otimes w')$.

Modified trees

Let $\underline{\Sigma} = \Sigma \cup \{\sigma_s / s \in \mathbb{N}^+\}$. For any unfolding tree T, we define an associated tree $\underline{T}$ (on $\underline{\Sigma} \cup \{\bullet\}$) by stating the following relations:

- dom $(\underline{T})$ = dom(T),
- $\underline{T}(s) = \sigma_s$ if $T(s) = \sigma$, $\underline{T}(s) = T(s)$ otherwise.

Thus, $\underline{T}$ may be decomposed into finite subtrees according to the following diagram:



Modified assignments

Let $\psi_s : N^\infty \to \underline{N}^\infty$ be the morphism $\psi_s(\lambda) = \lambda_s$ and $\psi_s(\upsilon) = \upsilon$ if $\upsilon \in \{\sigma,\tau\}$. For any set $U \in P(N^\infty)$, we denote by $U^{\underline{T}}$ the set of the U-assignments of $\underline{T}$, defined as the partial functions $f: \text{dom}(\underline{T}) \to \underline{N}^\infty$ satisfying the following conditions of coherence:

- ε (the root node) $\in$ dom(f) ;

- if $s \in dom(f)$ and $\mathbb{T}(s) = \bullet$ then $f(s) = \psi_s(u)$ for some $u \in U$,

- if $s \in dom(f)$ and $\mathbb{T}(s) = nil$ then $f(s) = \varepsilon$,

- if $s \in dom(f)$ and $\mathbb{T}(s) = \omega_k$, $k > 0$, then

either $f(s) \in \omega_k(f(s.1),\ldots f(s.k))$ or $\omega_k = \sigma_s$ and $f(s) = \varepsilon$.

A definition of Γ

For $U \in P(N^\infty)$, we state $\Gamma = \{\delta_\circ f \ / \ f \in \Gamma\}$ where Γ is the family of all the productive U-assignments for unfolding trees $\mathbb{T}$. Note that an assignment $f \in U^{\mathbb{T}}$ is productive if and only if $f(\varepsilon)$ contains an infinite number of subscripted symbols σ_s (each of which occurs only once). Our previous claim 2 follows now from the next proposition, where we let $\psi: N^\infty \to N^\infty$ be the morphism that erases the non subscripted symbols σ,τ and leaves all the other symbols invariant.

proposition 8.1* $\forall U, V \in P(N^\infty)$: if U or $\Omega(U)$ and V or $\Omega(V)$ are prefix closed, and if $\Pi_\Lambda(U) = \Pi_\Lambda(V)$, then for any productive assignment $f \in U^{\mathbb{T}}$, there is a productive assignment $g \in V^{\mathbb{T}}$ s.t. $\psi(f(\varepsilon)) = \psi(g(\varepsilon))$.

9. CONCLUSION

The outcome of the study is manyfold. We have shown that definitions of programming languages along Plotkin's method of structural operational semantics may also be taken as specifications of initial denotational models, where the meanings of programs are sets of computations. We have determined which informations should be kept and which informations should be erased from the initial model of CCS for obtaining a model of the equivalence $p \sim_1 q$ if and only if p and q have similar sets of maximal interactions with any program r. We have suggested a way for turning the difficulties encountered in deriving the abstract model through a non closed erasing morphism, namely a polymorphic definition of meanings in which open, resp. closed terms are dealt with differently.

Let us stress the limitations and open perspectives of our work, in relation to some other approaches to concurrency.

A rough separation may be observed between two families of denotational models for concurrency, namely the family of purely order theoretic models and the family of mixed space and order theoretic models. Models of the first class use classical ω-algebraic domains [9,18,24,25,29]. In the second class of models, the order relation is usually the (reverse) inclusion between closed subsets in some metric completion of languages, trees or tree-languages [3,23,31], and the considered c.p.o.'s are ω-algebraic because all those metric spaces have denumerable bases.

For the sake of simplicity, we have presented a translation of structural operational semantics into ordered models, but the topological treatment has indeed been left behind: our (initial) semantic operators are set extensions of profinite operators on words, and they preserve distances in the usual ultrametric topology . A clearly topological approach of the problem has been taken in [2], where an alternative translation of structural operational semantics into models of the second category is defined and applied to the full version of CCS with value passing.

In the two categories of denotational models discussed above, it is usually assumed that each element or open set of the denumerable base represents a property that may be tested under finite experiments [1], i.e. under experiments that either succeed in finite time or fail [18]. Thus, identical meanings are assigned to some pairs of CCS programs which differ by their respective sets of ω-traces. A possible escape to that criticism is to restrict oneself to a proper subset of CCS, in which the assumption of 'uniform concurrency' is valid [4]. Behaviours are then closed sets in the usual ultrametric topology, and infinite stream semantics resp. finite observation semantics are isomorphic [3]. In the present paper, where the full version of pure CCS was considered, we were not enrightened to set the assumption of uniform concurrency. So, we could not expect a countable base for abstract meanings, and had no hope to get closed meanings for programs in the usual ultrametric topology. The investigation of alternative topologies is an open way of research .

The artifice we have suggested for turning the non closedness of the erasing morphism ϕ (namely the polymorphic definition of the abstract meaning function) is valid in every other situation where $\phi\,(\,Fix_X\,F(U)(X)) = \phi(Fix_X F(\Phi U)(X))$ for all parametric functions $F(U)$ in the abstract

interpretation. In order to raise things to a more methodological level, there remains to investigate to what extent the validity of the above relation depends on the profinite properties of the operations.

Acknowledgment This paper owes much to an illuminating remark of E.R. Olderog.

APPENDIX

Indications of proof for proposition 2.1

The inclusion $\supseteq$ follows by induction on the length of transition sequences (simultaneously for all operators). The reverse inclusion is more difficult to prove. Given $\theta \in D^*[\![\omega_k(t_1 \ldots t_k)]\!]$, we must construct $\theta_i \in D^*[\![t_i]\!]$ s.t. $\theta \in \underline{\omega}_k(\theta_1, \ldots \theta_k)$. This is trivial for sequences θ of length zero (i.e. $\theta = \varepsilon_t$). Suppose $\theta = (\omega_k(u_1 \ldots u_k) -_D^{\upsilon}\rightarrow \omega'_m(u'_1 \ldots u'_m))\theta'$, where the logical transition before θ' is a non degenerated transition proved along some rule $\gamma \in A[\omega_k]$. Suppose $\theta' \in \underline{\omega}'_m(\theta'_1 \ldots \theta'_m)$ for some $\theta'_j \in D^*[\![u'_j]\!]$. Then one obtains the θ_i from the θ'_j by prefixing some of them with the transitions used as premisses for applying rule γ. But some sequences θ'_j may be left unchanged (up to a change of their index), and zero- length sequences θ_i may also appear in this induction step.

Proof of proposition 2.2

The inclusion $\subseteq$ follows from the definitions. We establish the reverse inclusion, i.e. $\theta \in CL(\underline{\omega}_k(Pref(Q_1), \ldots Pref(Q_k))) \Rightarrow \theta \in \omega_k(Q_1 \ldots Q_k)$.

If θ is a finite sequence, then $\theta \in \underline{\omega}_k(\theta_1 \ldots \theta_k)$ for some finite $\theta_i \in Pref(Q_i)$, and hence $\theta \in \omega_k(Q_1 \ldots Q_k)$ by the assumption of saturatedness. In the remaining part of the proof, we assume $\theta = (u_i -^{\upsilon_i}\rightarrow u_{i+1})_{i<\omega} \in CL(\underline{\omega}_k(Pref(Q_1) \ldots Pref(Q_k)))$, and show $\theta \in \omega_k(Q_1 \ldots Q_k)$ by recursively solving the general problem $P(\theta, \omega_k, Q_1 \ldots Q_k)$ stated as follows :

- for $j = 1...k$ and $n \in \mathbb{N}$, construct k decreasing n-indexed chains of non empty subsets $Q^n{}_j \subseteq Q_j$ such that either $Q^n{}_j$ is utltimately constant or $\cap_n Q^n{}_j$ is a singleton set, and θ belongs to $\overline{\omega_k}(\theta_1...\theta_k)$ for arbitrary θ_j in $\cap_n Q^n{}_j$.

Let $u_o = \omega_k(t_1...t_k)$, whence $(\forall i)\ \varepsilon_{ti} \in Pref(Q_i)$, and let $\theta^{<n} = (u_i -^{\upsilon_i}\!\!\rightarrow u_{i+1})_{i<n}$, whence $(\forall n)\ \theta^{<n} \in \underline{\omega}_k(Pref(Q_1)...Pref(Q_k))$. We proceed by cases.

<u>case a</u> For some axiom of the form : $\omega_k(s_1...s_k) -_D^{\upsilon}\!\!\rightarrow id(s_p)$ <u>for</u> $R(\upsilon)$; $R(\upsilon_o)$ holds and there is an infinite number of m s.t. $\theta^{<m} = (u_o -^{\upsilon_0}\!\!\rightarrow t_p)\theta^m{}_p$ for some $\theta^m{}_p \in Pref(Q_p)$.

For all n, we set $Q^n{}_j = \{\varepsilon_{tj}\}$ for $j \neq p$ and $Q^n{}_p = \{(u_{i+1} -^{\upsilon_{i+1}}\!\!\rightarrow u_{i+2})_{i<\omega}\}$.

<u>case b</u> For some operational rule of the form :

$$\frac{s_p -_D^{\upsilon'}\!\!\rightarrow s'_p}{\omega_k(s_1...s_k) -_D^{\upsilon}\!\!\rightarrow id(s'_p)} \ \underline{\text{for}}\ R(\upsilon',\upsilon),$$

the following conditions hold for infinitely many m :

- $R(\upsilon'_m, \upsilon_o)$,
- $(t_p -^{\upsilon'_m}\!\!\rightarrow u_1)\ (u_{i+1} -^{\upsilon_{i+1}}\!\!\rightarrow u_{i+2})_{i<m} \in Pref(Q_p)$.

For all n, we set $Q^n{}_j = \{\varepsilon_{tj}\}$ for $j \neq p$ and

$Q^n{}_p = \{(t_p -^{\upsilon'_m}\!\!\rightarrow u_1)(u_{i+1} -^{\upsilon_{i+1}}\!\!\rightarrow u_{i+2})_{i<\omega}\}$ for some (fixed) m.

<u>case c</u> For some operational rule γ of the form

$$\gamma: \frac{s_1 -_D^{\upsilon'_1}\!\!\rightarrow s'_1, \ldots\ s_p -_D^{\upsilon'_p}\!\!\rightarrow s'_p}{\omega_k(y_1...y_k) -_D^{\upsilon}\!\!\rightarrow \omega_l(y'_1...y'_l)} \ \underline{\text{for}}\ R(\upsilon'_1,...\upsilon'_p,\upsilon),$$

the following conditions hold for infinitely many m :

- $\theta^{<m} \in \omega_k{}^{\gamma,\upsilon_0}(\theta^m{}_1, \ldots \theta^m{}_k)$ for some $\theta^m{}_i \in Pref(Q_i)$

- $\theta^m{}_i = (t_i\text{-}\upsilon^m{}_z \rightarrow t'_i)\theta'^m{}_i$ for some $\upsilon^m{}_z$ if $i = f\gamma(z)$

- $R(\upsilon^m{}_1, \ldots \upsilon^m{}_p, \upsilon_o)$.

Observe here that the t'_i are independant of m !

Now, we set $Q^o{}_j = Q_j$ for all j, and we specify $Q_i{}^{n+1}$ in terms of auxiliary sets $Q'^n{}_j$ which are the solution of a recurrent problem . More precisely, we set :

$Q_i{}^{n+1} = \{\varepsilon_{ti}\}$ if $i \notin dom(\gamma') \cup dom(\gamma'')$,

$Q_i{}^{n+1} = Q'^n{}_j$ if $\gamma''(i,j)$,

$Q_i{}^{n+1} = (t_i\text{-}\upsilon^m{}_z \rightarrow t'_i)Q'^n{}_j$ for some (fixed) m if $\gamma'(i,j)$ and $i = f\gamma(z)$.

The auxiliary sets $Q'^n{}_j$ are in turn defined as the solutions of $P(\theta', \omega_1, Q'_1 \ldots Q'_1)$ with parameters as follows :

- $\theta' = (u_{i+1}\text{-}\upsilon_{i+1} \rightarrow u_{i+2})_{i<\omega}$,

- $Q'_j = Q_i$ if $\gamma''(i,j)$,

- $Q'_j = \{\theta'' \ / \ \exists v \ ((t_i\text{-}\upsilon \rightarrow t'_i)\theta'' \in Q_i) \ \}$ if $\gamma'(i,j)$ and $i = f\gamma(z)$.

This completes the resolution of problem P, since $Q_i{}^{n+1} \subseteq Q_i$ follows recursively from the assumption of saturatedness (which is inherited by the $Q'^n{}_j$). Hence the proposition is proved .

proof of proposition 6.2

The lefmost implication is a direct consequence of propositions 5.1 and 6.1 . The proof for the rightmost implication is a weakening of the proof for proposition 6.3 .

proof of proposition 6.3

Assuming $\phi_{int}(\theta_u) = \langle w, R \rangle$ for some fixed $\theta_u \in \mathcal{M}_{op}[|u|]$, we proceed by case analysis on $\langle w, R \rangle$, to show that $u \sqsubseteq_1 v$ implies the existence of some $\theta_v \in \mathcal{M}_{op}[|v|]$ satisfying $\phi_{int}(\theta_u) \sqsubseteq \phi_{int}(\theta_v)$. In each of the five cases encountered, z stands for $\Pi_\Lambda(\eta(w))$, and $u \sqsubseteq_1 v$ is supposed to hold.

case 1. $R \neq \omega$, $R \cap \{\sigma,\tau\} = \emptyset$, $\Pi_\Lambda(w) \neq \varepsilon$, $\Pi_\tau(w) = \varepsilon$:

if $z = \lambda_1 \ldots \lambda_n$ and $(sort(v) \setminus R) = \eta\{\alpha_1 \ldots \alpha_m\}$, let $t = \lambda_1(\ldots\lambda_n(\alpha_1 nil + \ldots + \alpha_m nil)\ldots)$, then $z \in L(u,t) \Rightarrow z \in L(v,t)$, hence there is some θ_v in $\mathcal{M}_{op}[|v|]$ s.t. the following holds of $\langle w', R' \rangle = \phi_{int}(\theta_v)$:

$\Pi_\Lambda(\eta(w')) = z$ & $((R' = \omega) \vee (R' \cap \eta\{\sigma,\tau,\alpha_1,\ldots\alpha_m\} = \emptyset)$.

Suppose $R' \neq \omega$ and $R' \cap \{\sigma,\tau\} = \emptyset$ then, since R' is included in the sort of v, the assertions $R' \cap \eta\{\alpha_1 \ldots \alpha_m\} = \emptyset$ and $R' \subseteq R$ are equivalent. Therefore, $\langle w,R \rangle \lesssim \langle w',R' \rangle$ in all situations.

case 2. $R \neq \omega$, $R \cap \{\sigma,\tau\} = \emptyset$, $\Pi_\Lambda(w) = \varepsilon$, $\Pi_\tau(w) \neq \varepsilon$:

if $\eta(\alpha_o) \notin sort(u) \cup sort(v)$ and $(sort(v) \setminus R) = \eta\{\alpha_1 \ldots \alpha_m\}$, let $C[.] = (. + \eta\alpha_o(nil))$ and $t = \alpha_o nil + \ldots + \alpha_m nil$; then $\varepsilon \in L(C[u],t) \Rightarrow \varepsilon \in L(C[v],t)$.

Only two possibilities are left : either, for some $\theta_v \in \mathcal{M}_{op}[|v|]$, $\phi_{int}(\theta_v) = \langle w',R' \rangle$ where $\Pi_\Lambda(w') = \varepsilon$ & $R' = \omega$, or for some $\theta_v \in \mathcal{M}_{op}[|v|]$, $\phi_{int}(\theta_v) = \langle w',R' \rangle$ where $\Pi_\Lambda(w') = \varepsilon$ & $\Pi_\tau(w') \neq \varepsilon$ & $R' \neq \omega$ & $R' \cap \{\sigma,\tau\} = \emptyset$ & $R' \cap \eta\{\alpha_1 \ldots \alpha_m\} = \emptyset$ But then $\langle w,R \rangle \lesssim \langle w',R' \rangle$ in all situations for the same reasons as above.

case 3. $R \neq \omega$, $R \cap \{\sigma,\tau\} \neq \emptyset$:

there, θ_u can be extended by silent transitions; choose θ'_u maximal in the set of silent extensions of θ_u, substitute θ'_u for θ_u and iterate the proof : case 3 cannot occur again, and $\theta_u \lesssim \theta'_u$.

case 4. $R = \omega$, $z \in \Lambda^*$:

if $z = \lambda_1 \ldots \lambda_n$ and $\alpha,\beta \notin sort(u) \cup sort(v)$; let $C[.] = (.|\ \lambda_1(\ldots\lambda_n(\alpha(\beta nil))\ldots))$ and $t = \eta\alpha(\eta\beta(nil))$, then $\eta\alpha \in L(C[u],t) \Rightarrow \eta\alpha \in L(C[v],t)$ and there must exist $\theta_v \in \mathcal{M}_{op}[|v|]$ s.t. $\phi_{int}(\theta_v) = \langle w',\omega \rangle$ and $\Pi_\Lambda(\eta(w')) = z$.

case 5. $\eta(z) \in \{\lambda_1 \ldots \lambda_n\}^\omega$:

let $t = x$ wh $(x = \lambda_1 x + \ldots + \lambda_n x)$, then $\eta(z) \in L(u,t) \Rightarrow \eta(z) \in L(v,t)$ and there must exist $\theta_v \in \mathcal{M}_{op}[|v|]$ s.t. $\phi_{int}(\theta_v) = \langle w',\omega \rangle$ and $\Pi_\Lambda(\eta(w')) = z$.

No other case can occur, thus the proof is complete.

Proof of proposition 8.1

The proof is rather complex and calls for a lot of auxiliary definitions and facts. The presentation is in three parts. In a first part, we equip sets of assignments with both an ordered and a metric structures. In a second part, we state two properties of the operators ω_k concerning the non subscripted symbols σ,τ erased by ψ . The main body of the proof appears in the third part.

a.An ordered and a metric structures on assignments

definition $\mathbb{N}^\infty$ is ordered by $\leq$: $u \leq v \Leftrightarrow \exists w\ v=uw$.

fact1 Let $\omega_k \in \Sigma_k$, $z_1 \ldots z_k \in \mathbb{N}^\infty$ and $z \in \omega_k(z_1 \ldots z_k)$ then for any $z' \leq z$ with length at most n, there exist $z'_1 \leq z_1 \ldots z'_k \leq z_k$ with length at most n s.t. $z' \in \omega_k(z'_1 \ldots z'_k)$.

definition The order $\leq$ on $U^{\mathbb{T}}$ is the relation such that $h \leq h'$ iff $dom(h) \subseteq dom(h')$ & $\forall s \in dom(h)\ h(s) \leq h'(s)$.

fact2 Let $(f_n)_n$ be an increasing sequence of finitary assignments in $U^{\mathbb{T}}$, where $U = V \cup Pref(V)$ and $\lim_n f_n(s) \in V$ for all $s \in \cup_n dom(f_n)$, then $(f_n)_n$ has a least upper bound $f \in U^{\mathbb{T}}$, and indeed $f \in V^{\mathbb{T}}$ (this property comes mainly from the continuity of $\otimes$).

definition The distance d on $U^{\mathbb{T}}$ is $d(f,g) = 2^{-n}$ where n is the least integer such that either dom(f) and dom(g) differ by some string s of length n , or f(s) and g(s) differ by some prefix of length n, for some s of length less than or equal to n.

fact3 Let $(f_n)_n$ be a Cauchy sequence of assignments in $U^{\mathbb{T}}$, where $U = V \cup Pref(V)$ and $\lim_n f_n(s) \in V$ for all $s \in \cup_n dom(f_n)$, then $(f_n)_n$ converges to an assignment $f \in U^{\mathbb{T}}$, and indeed $f \in V^{\mathbb{T}}$
(this property follows as a corollary from facts 1 and 2).

b.Two properties of the operators ω_k w.r.t. silent actions

<u>definition</u> Relation $\leftarrow$ on $\mathbb{N}^\infty$ is the preorder s.t. $z \leftarrow z'$ if and only if $\psi(z) = \psi(z')$ and for any left factor $w \leq z$ in $\{ \sigma_s : s \in \mathbf{N}^* \}^*$, $\psi(w) = \psi(w')$ for some left factor $w' \leq z'$ in $\{ \sigma_s : s \mathbf{N}^* \}^*$.

<u>definition</u> Relation $\leftarrow$ on finitary assignments is the preorder such that $f \leftarrow g$ if and only if $dom(f) = dom(g)$ and $\forall s \in dom(f) : f(s) \leftarrow g(s)$.

<u>fact4</u> Let ω_k in Σ_k and let $z_i, z"_i \in \mathbb{N}^\infty$ be s.t. $z_i \leftarrow z"_i$ for $i=1...k$, then for all $z \in \omega_k(z_1...z_k)$ there exist $z', z'_i \in \mathbb{N}^\infty$ s.t. $z \leftarrow z'$, $z' \in \omega_k(z'_1...z'_k)$ and $z'_i \leq z"_i$ for all i .

<u>fact5</u> Let $\omega_k \in \Sigma_k$ and for $i=1...k$, let $w_i \in \mathbb{N}^\infty$ be s.t $w \in \omega_k(w_1...w_k)$, then for any $z \in \{\sigma,\tau\}^\infty$ and $i \in \{1...k\}$, the set $\omega_k(w_1...w_i z...w_k)$ contains either w or wz or $w\sigma z$.

c.The main body of the proof

Let U and V denote sets of (non subscripted) action sequences in N^∞, s.t. U or $\Omega(U)$ and V or $\Omega(V)$ are prefix closed and $\Pi_\Lambda(U) = \Pi_\Lambda(V)$. Let $f \in U^T$ be a productive U-assignement of T for some unfolding tree T. We must construct $g \in V^T$ such that $\psi(f(\varepsilon)) = \psi(g(\varepsilon))$.

§ Define $PU = U \cup Pref(U)$, and let Q denote the following predicate on assignments: $Q(h) \equiv \forall s \in dom(h)\ \max(|h(s)|, \#\{s' \leq s \ / \ h(s') = \bullet\}) \leq |h(\varepsilon)|$.
Since f is productive, $f(\varepsilon)$ may always be written in the unending form: $w_0\sigma_{s0}w_1\sigma_{s1}\cdots w_n\sigma_{sn}\cdots$, where $w_i \in (\bar{N} - \{\sigma_s \ / \ s \neq \varepsilon\})^*$.
From fact1, $f(\varepsilon) = \lim_n (f'_n(\varepsilon))$ for some sequence of finitary assignments f'_n in PU^T satisfying: $f'_n \leq f$ & $Q(f'_n)$ & $f'_n(\varepsilon) = w_0\sigma_{s0}w_1\sigma_{s1}\cdots w_n\sigma_{sn}$.

Again from fact1, each f'_n is the upper bound f_n^n of a finite chain of finitary assignments f_n^j in $PU^{\mathbb{T}}$ s.t.: $f_n^j \leq f'_n$ & $Q(f_n^j)$ & $f_n^j(\varepsilon) = w_0\sigma_{s0}w_1\sigma_{s1}\cdots w_j\sigma_{sj}$. Since T is an unfolding tree, the set of all the possible f_n^j is finite for fixed j. Thus, by Koenig's lemma, there is an increasing chain of finitary assignments f_n in $PU^{\mathbb{T}}$ satisfying: $f_n \leq f$ & $Q(f_n)$ & $f_n(\varepsilon) = w_0\sigma_{s0}w_1\sigma_{s1}\cdots w_n\sigma_{sn}$.
We set the following definitions for the remaining parts of the proof:
$S = \{ s \in dom(T) / T(s) = \bullet \}$,
$u^s = f(s)$ for $s \in S \cap dom(f)$ and $u_n^s = f_n(s)$ for $s \in S \cap dom(f_n)$.

§ Define $PV = V \cup Pref(V)$. For each $s \in S$, fix $v^s \in V$ s.t. $\Pi_\Lambda(u^s) = \Pi_\Lambda(v^s)$, anf then for all $n \in \mathbb{N}$, define v_n^s as the shortest left factor of v s.t. $\Pi_\Lambda(u_n^s) = \Pi_\Lambda(v_n^s)$. From facts 1 and 4, there exists for each n a finitary assignment g'_n in $PV^{\mathbb{T}}$ satisfying: $dom(g'_n) \subseteq dom(f_n)$, $g'_n(s) \leq v_n^s$ for all $s \in S$, and $g'_n(\varepsilon) \leftarrow w_0\sigma_{s0}w_1\sigma_{s1}\cdots w_n\sigma_{sn}$.
Again from fact1, each g'_n is the upper bound g_n^n of a finite chain of finitary assignments g_n^j in $PV^{\mathbb{T}}$ satisfying:
$Q(g_n^j)$ & $g_n^j(\varepsilon) \leftarrow w_0\sigma_{s0}w_1\sigma_{s1}\cdots w_j\sigma_{sj}$ & $g_n^j(s) \leq v_{j+1}^s$ for all $s \in S$.
Now, the set of all the possible g_n^j is finite for fixed j, and thus by Koenig's lemma, there is an increasing chain of finitary assignments g_n in $PV^{\mathbb{T}}$ satisfying: $g_n(\varepsilon) \leftarrow w_0\sigma_{s0}w_1\sigma_{s1}\cdots w_n\sigma_{sn}$ for all n.
From fact2, that sequence has a least upper bound g in $PV^{\mathbb{T}}$. For obvious reasons, the corresponding assignment satisfies $\psi(f(\varepsilon)) = \psi(g(\varepsilon))$, but g does not necessarily belong to $V^{\mathbb{T}}$.

§ If V is prefix closed, then $g \in V^{\mathbb{T}}$ and the proposition is shown. We consider now the converse case where $\Omega(V)$ is prefix closed. From that property and for any $s \in S$: $g(s) \in V$ or $g(s)z^s \in V$ for some $z^s \in \{\tau, \sigma^\omega\}$.
Let s_n denote the n-th element in the lexicographic ordering of S. By iterating the following process for constructing h_n from h_{n-1}, one may construct from $h_0 = g$ a sequence of assignments $h_n \in PV^{\mathbb{T}}$:

- given h_{n-1}, take $h_n(s) = h_{n-1}(s)$ for $s \not\leq s_n$, $h_n(s_n) = h_{n-1}(s_n)z^{sn}$, and form the $h_n(s')$ from the $h_{n-1}(s')$, $s'<s_n$, by appending to the latter the appropriate words in the set $\sigma^* \cup \sigma^*\tau \cup \sigma^\omega$.

The success of each step in the construction is guaranteed by fact 4. The resulting sequence $(h_n)_n$ is clearly a Cauchy sequence of assignments in PV^T . From fact 3, that sequence converges to a V-assignment $h \in V^T$.

For obvious reasons, $\psi(h(\varepsilon)) = \psi(g(\varepsilon))$, thus $\psi(h(\varepsilon)) = \psi(f(\varepsilon))$ and the proposition is shown.

BIBLIOGRAPHY

[1] Abramsky S.
Experiments, Power domains, and Fully Abstract Models for Applicative Multiprogramming , FCT 83,
Borgholm, Springer-Verlag LNCS 158 (83) pp. 1 - 13

[2] Badouel E.
Première étape vers une construction systématique de sémantiques comportementales,
Mémoire de DEA, Université de Rennes, June 87

[3] de Bakker J. W., Meyer J.J. Ch., Olderog E.R.
Infinite Streams and Finite Observations in the Semantics of Uniform Concurrency
ICALP 85, Naflion
Springer-Verlag LNCS 194 (85) pp. 149 - 157

[4] de Bakker J.W., Meyer J.J.Ch., Olderog E.R., Zucker J.I.
Transition Systems, Infinitary Languages and the Semantics of Uniform Concurrency
17th ACM STOC, Providence (85) pp. 252-262

[5] Beki'c H.
Definable Operations in General Algebras, and the Theory of Automata and Flowcharts
IBM Laboratory Vienna (69)
also in Programming Languages and their Definition
Selected papers of H. Beki'c -Springer-Verlag LNCS 177(84)

[6] Bergstra J. A. , Klop J.W., Olderog E.R.
Readies and Failures in the Algebra of Communicating Processes
Report CSR-8523, CWI, Amsterdam (85)

[7] Boasson L., Nivat M.
Adherences of Languages
JCSS 20 (80) pp. 285 - 309

[8] Brookes S., Hoare C.A.R., Roscoe A. W.
A Theory of Communicating Sequential Processes
JACM 31,3 (84) pp. 560-599

[9] Brookes S., Roscoe A. W.
An Improved Failures Model for Communicating Processes
Seminar on Concurrency, Brookes Roscoe Winskel (eds.)
Springer-Verlag LNCS 197 (85) pp. 281 - 305

[10] Broy M.
Fixed Point Theory for Communication and Concurrency,
Formal Description of Programming concepts-II (IFIP 83)
D. Bjorner (ed.) North Holland Publishing Company

[11] Darondeau Ph.
Quelques problèmes relatifs à la composition parallèle
Congrès AFCET 81, Gif sur Yvette, Editions Hommes et Techniques (Paris)

[12] Darondeau Ph.
About Fair Asynchrony
TCS 37 (85) pp. 305 - 336

[13] Darondeau Ph.
Separating and Testing
STACS 86, Orsay
Springer-Verlag LNCS 210 (86) pp. 203 - 212

[14] Darondeau Ph.
Une critique de la notion de test de processus fondée sur la non séparabilité de certaines classes de langages
Theoretical Informatics and Applications 20,3 (86) pp.291-318

[15] Darondeau Ph., Gamatie B.
A Fully Observational Model for Infinite Behaviours of Communicating Systems
TAPSOFT-CAAP 87 , Pisa
Springer-Verlag LNCS 249 (87) pp. 153 - 168

[16] Darondeau Ph, Yoccoz S.
Proof Systems for Infinite Behaviours
INRIA report 632 (March 87)

[17] De Simone R.
Calculabilité et expressivité dans l' algèbre de processus parallèles MEIJE ,
Thèse de 3e cycle, Université Paris 7 (84)

[18] De Nicola R., Hennessy M.
Testing Equivalences for Processes
TCS 34 (84) pp. 83 - 113

[19] De Nicola R.
Testing Equivalences and Fully Abstract Models for Communicating Processes.
Ph. D Thesis, University of Edinburgh (85)

[20] Gamatie B.
Observational Congruences of Non Deterministic and Communicating Finite Processes
RR - 254 - IRISA Rennes (85)

[21] Gamatie B.
Safe Implementation Equivalence for Asynchronous Non Deterministic Processes
MFCS 86, Bratislava
Springer-Verlag LNCS 233 (87) pp. 360-369

[22] Gamatie B.
Towards Specification and Proof of Asynchronous Systems
STACS 86, Orsay
Springer-Verlag LNCS 210 (86) pp. 203 - 212

[23] Golson W. Rounds W. C.
Connections between two Theories of Concurrency : Metric Spaces and Synchronization Trees
Information and Control 57 (83) pp. 102 - 124

[24] Hennessy M. Plotkin G.
A term model for CCS
Springer-Verlag LNCS 88 (80) pp. 261 - 274

[25] Hennessy M.
Acceptance trees
JACM 32 (85) pp. 896 - 928

[26] Hennessy M.
An Algebraic Theory of Fair Asynchronous Communicating Processes
TCS 49,2-3(87) pp.121-143

[27] Milner R.
Fully Abstract Models of Typed lambda-calculi
TCS 4 (77) pp. 1 - 23

[28] Milner R.
A Calculus of Communicating Systems
Springer - Verlag LNCS 92 (80)

[29] Olderog E. R. , Hoare C.A.R.
Specification Oriented Semantics for Communicating Processes
Acta Informatica 23,1 (86) pp.9-86

[30] Plotkin G.
A Structural Approach to Operational Semantics
Rept. DAIMI - FN - 19, Univ. of Aarhus,
Computer Science Department (81)

[31] Rounds W. C.
On the Relationships between Scott Domains, Synchronization Trees and Metric Spaces
Information and Control 66 (85) pp. 6 - 28

[32] Tarski A.
A Lattice Theoretic Fixpoint Theorem and its Applications.
Pacific Journal of Mathematics 5 (55) pp. 285 - 309.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique