HAL
open science

Proof systems for infinite behaviours

Philippe Darondeau, Serge Yoccoz

To cite this version:

Philippe Darondeau, Serge Yoccoz. Proof systems for infinite behaviours. [Research Report] RR-0632, INRIA. 1987. inria-00075921

HAL Id: inria-00075921

https://inria.hal.science/inria-00075921v1

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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° 632

PROOF SYSTEMS FOR INFINITE BEHAVIOURS

Philippe DARONDEAU
Serge YOCCOZ

Mars 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

Publication Interne n° 348
Février 1987
12 pages

PROOF SYSTEMS FOR INFINITE BEHAVIOURS

Philippe DARONDEAU *
Serge YOCCOZ **

Abstract :

We establish the following result, which holds in particular for the class Π of CCS programs. Let Π be a class of programs and let $\sim$ be an equivalence on Π. Suppose any equality between Σ^1_1 sets of N => {0,1} is Turing reducible to the equivalence of two programs in Π. Then a first order proof system with recursive infinitary rules cannot be complete for the equality in $\Pi/\sim$.

COMPORTEMENTS INFINIS ET SYSTEMES DE PREUVE

Résumé :

Nous établissons le résultat suivant, qui vaut en particulier pour la classe Π des programmes CCS. Soit Π une classe de programmes et soit $\sim$ une équivalence sur Π. Supposons que toute égalité d'ensembles Σ^1_1 de N =>{0,1} soit Turing réductible à l'équivalence de deux programmes de Π. Alors un système de preuve du premier ordre avec règles infinitaires récursives ne peut pas être complet pour l'égalité dans $\Pi/\sim$.

* IRISA - Campus de Beaulieu - 35042 RENNES CEDEX
** Université de Bordeaux I - UER Mathématiques et Informatique - 351 Cours de la Libération - 33405 TALENCE CEDEX.

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)

PROOF SYSTEMS FOR INFINITE BEHAVIOURS

In the area of communicating behaviours, proof systems have been constructed for two types of semantic equivalences : bisimulation equivalences [Milner, Hennessy-Plotkin, Stirling] and testing equivalences [De Nicola-Hennessy, Hennessy] . In particular, pure CCS has recently been given a model for the smallest testing equivalence, the one obtained from CCS tests resulting in the infinitary histories of their own computations [Darondeau- Gamatie] . Like bisimulation equivalences and unlike the other testing equivalences, the smallest testing equivalence $\sim$ does always separate processes that differ upon infinite sequences of external actions . This raises the question of proof systems for CCS/$\sim$. We prove in this paper the following result :

Theorem : No first order proof system, even in the presence of (recursive) infinitary rules, is both sound and complete for the equality in CCS/$\sim$.

In fact, we prove the more general result stated as follows :

Let Π be a class of programs and let $\sim$ be an equivalence on Π . Suppose any equality between Σ^1_1 sets of $\mathbf{N} \Rightarrow \{0,1\}$ is Turing reducible to the equivalence of two programs in Π . Then a first order proof system with recursive infinitary rules cannot be complete for the equality in $\Pi/\sim$.

The proof of the theorem is divided as follows :

Lemma 1 : The validity of any Π^1_2 formula of the arithmetic is Turing-reducible to the equality of two Σ^1_1 sets of functions from $\mathbf{N}$ to $\{0,1\}$.

Lemma 2 : Any equality between two Σ^1_1 sets in $\mathbf{N} \Rightarrow \{0,1\}$ is Turing-reducible to the equivalence of two processes p and q in CCS .

Lemma 3 : If a proof system Σ with recursive infinitary rules is given, the provability in Σ is a Σ^1_2 property (i.e is given by a Σ^1_2 formula) .

Lemma 4 : There is no effective procedure reducing the validity of a Π^1_2 formula to the validity of a Σ^1_2 formula .

It is clear that lemmas 1 through 4 prove the above theorem , and that lemmas 1,3 and 4 prove our generalized statement .

PAPIER RECUPERE ET RECYCLE

Proof of lemma 1 :

Let $\overline{F}$ be the subset of functions from $\mathbf{N}$ to $\{0,1\}$ with an infinite number of zeroes . Any $\overline{f} \in \overline{F}$ is the code of some f in F : $\overline{f}(i)$ is the digit with rank i+1 in the Gödel representation of the sequence f(0),f(1),... Indeed, the bijective relation

$\overline{\ }$ included in $F \times \overline{F}$ coincides with the set
$\{(f,\overline{f})/\ (\exists h)(\forall i)(\forall j)\ K(f,\overline{f},h,i,j)\}$ where K(f,g,h,i,j) is the recursive relation :

$[h(0) = f(0) + 1]$ & $[h(i+1) = h(i)+f(i+1)+2]$ & $[g(h(i)) = 0]$ & $[(j<h(0)) \Rightarrow g(j)=1]$ & $[(h(i)<j<h(i+1)) \Rightarrow g(j) = 1]$.

Let φ be the Π^1_2 formula $(\forall f)(\exists g)(Qw)\ R(f,g,w)$, where w stands for $w_1,\ldots,w_k$ and Qw is an alternation of first order quantifiers on $w_1, \ldots, w_k$. Then φ is valid iff

$$\{\overline{f}/\ (\exists g)(Qw)\ R(f,g,w)\} = F$$
$$\text{iff } \{\overline{f}/(\exists f)(\exists h)(\exists g)(Qw)(\forall i)(\forall j)[R(f,g,w)\ \&\ K(f,\overline{f},h,i,j)]\} = \overline{F}$$

But this is clealy an equality between two Σ^1_1 sets, and so there is an effective reduction of φ to the equality of two Σ^1_1 sets in $\mathbf{N} \Rightarrow \{0,1\}$.

Proof of lemma 2 :

For any CCS program t , call $L^\omega(t)$ the set of infinite words $w = \mu_0\mu_1 \ldots$ which are the labels of some infinite sequence of transitions

$$t_0 \xrightarrow{\mu_0} t_1 \xrightarrow{\mu_1} \ldots \quad \text{from} \quad t_0 = t\ .$$

Hence, each μ_i either is the internal action τ or else belongs to the set of external actions , say Λ . For w, w' belonging to $(\Lambda \cup \{\tau\})^\omega$ let $w \sim^\omega w'$ iff there are infinite factorizations $w = w_1w_2\ldots$, $w' = w'_1w'_2\ldots$ such that $w_i \sim^* w'_i$ for all i, where $\sim^*$ is the congruence on $(\Lambda \cup \{\tau\})^*$ generated from the axioms $\lambda\sim^*\tau\lambda$ and $\lambda\sim^*\lambda\tau$ ($\lambda \in \Lambda$). Up to the equivalence $\sim^\omega$, $L^\omega(t)$ may be reduced to the normalized set $[L^\omega(t)]$ equal to

$$\{\ w \in \Lambda^\omega \cup \Lambda^*\tau^\omega/\ \exists\ w' \in L^\omega(t),\ w \sim^\omega w'\}.$$

Now suppose that U and V are Σ^1_1 subsets of $\{0,1\}^\omega$ (i.e $\mathbb{N} \to \{0,1\}$) for some pair of actions 0,1 in Λ . Following the construction given in [Darondeau], CCS programs u and v satisfying $[L^\omega(u)] = (\{0,1\}^*\tau^\omega \cup U)$ and $[L^\omega(v)] = (\{0,1\}^*\tau^\omega \cup V)$ may be obtained effectively uniformly from any Σ^1_1 indexes of U and V (Hints of the construction are given in Appendix). Furthermore, U=V iff $u \sim v$ for the smallest testing equivalence of [Darondeau-Gamatie], so the proof is complete .

Proof of lemma 3 :

We need first some definitions and clarifications .
We consider Ω, a denumerable and possibly multisorted algebra of terms with variables, and $L(\Omega)$, a first order language constructed in the usual way from Ω and some finite family of relational symbols .

A proof system Σ with (recursive) infinitary rules is a triple $\langle Ax, R, IR\rangle$ where :

- Ax is a finite consistent set of formulas from $L(\Omega)$ (the axioms) .
- R is a set of finitary inference rules, written $A \Leftarrow A_1, \ldots, A_i$ with A, A_i belonging to $L(\Omega)$.
- IR is a set of recursive infinitary rules, written $A \Leftarrow A_1, \ldots, A_i, \ldots$ each of which is associated with an effective procedure P such that $P(i) = A_i$ for all i in $\mathbf{N}$.

A formula φ is provable in Σ if :

- φ is an instance of a provable formula θ ,
- φ belongs to Ax ,
- $\varphi = \sigma(A)$ and there are substitutions $\sigma_1, \ldots, \sigma_n$; $\tau_1, \ldots, \tau_n$ such that for all i, $\sigma = \sigma_i . \tau_i$ and $\tau_i(A_i)$ is provable in Σ, and $A \Leftarrow A_1, \ldots, A_i$ belongs to R ,
- $\varphi = \sigma(A)$ and there are substitutions σ_i, τ_i ($i \in \mathbf{N}$) such that for all i, $\sigma = \sigma_i . \tau_i$ and $\tau_i(A_i)$ is provable in Σ, and $A \Leftarrow A_1, \ldots, A_i, \ldots$ belongs to IR .

Proofs in Σ may be defined inductively as finite-path trees whose terminal roots are the concluding steps of proof .
According to the division of Σ, there are three kinds of proof constructors :

- for all A in Ax and for all substitutions σ, $(A, \sigma A, \sigma)$ is a constant proof for the formula σA .
- for any r in R , say $A \Leftarrow A_1, \ldots, A_n$, and for any substitutions $\sigma, \tau_1, \ldots, \tau_n$ satisfying $\sigma = \sigma_i \tau_i$, $((\tau_1, \ldots, \tau_n), r, \sigma A, \sigma)$ is an n-ary proof constructor, which yields a proof of σA whenever it is applied to some n-tuple $T_1, \ldots, T_n$ of proofs for formulas $\tau_1 A_1, \ldots, \tau_n A_n$.
- for any r in IR , say $A \Leftarrow A_1, \ldots, A_i, \ldots$ and for any substitutions σ, τ_i ($i \in \mathbf{N}$) verifying $\sigma = \sigma_i . \tau_i$, $((\tau_i)_{i \in \mathbf{N}}, r, \sigma A, \sigma)$ is an infinitary proof constructor, which yields a proof of σA whenever it is appied to some sequence $(T_i)_{i \in \mathbf{N}}$ of proofs for formulas $\tau_i A_i$.

It is clear that a formula φ of $L(\Omega)$ is provable in Σ if and only if it has a proof in Σ, i.e iff there is a corresponding proof tree that assesses φ .

The *general idea of the demonstration* is to show that the existence of a proof tree, for a given formula φ of $L(\Omega)$; is expressed by a Σ^1_2 formula of the arithmetic . For that purpose, we proceed in three steps :

- we slightly change the representation of proof trees by shifting substitutions τ_i one level down in the trees, in such a way that τ_i appears in the labelling of the node that proves $\tau_i A_i$.
- we complete then the finite-path trees into uniformly ω-branching trees, by adding wherever necessary dummy nodes labelled with \$, the empty proof . This amounts to make a partial function $T : \mathbf{N}^* \to PC \setminus \$$ into a total function $T : \mathbf{N}^* \to PC$, where PC is the set of proof constructors .
- finally, we bijectively encode the uniformly ω-branching proof trees into functions from $\mathbf{N}$ to $\mathbf{N}$, and show that the resulting set of functions PT is Π^1_1 . The set PT is made out of those functions which are codes for trees in which every branch is information finite and every node represents a correct application of Σ (since the encoding is recursive at each node, this last property is uniformly expressed by a recursive predicate) .

In view of the bijective connection between PT and the set of finite-path trees over PC, the proof of the lemma follows from the Π^1_1 characterization of PT .

We put down here our *conventions of notation* . We suppose that we are given the following recursive encodings into $\mathbf{N}$:

- a numbering N of the axioms and rules of Σ
- a one to one encoding f_1 of the finite formulas of $L(\Omega)$
- a one to one encoding f_2 of the finitary rules in R such that if r is $A=A_1,\ldots A_n$ then $f_2(r)=\langle f_1(A),\ldots,f_1(A_n)\rangle$, $\langle . \rangle$ being the usual coding from $\mathbf{N}^*$ to $\mathbf{N}$ with $lg\langle k_1,\ldots,k_n\rangle = n$.
- a one to one encoding f_3 of the infinitary rules in IR given by $f_3(A=A_1,\ldots,A_i,\ldots) = \langle f_1(A),z\rangle$, z being the index of the Turing machine enabling us to recover A_i from i .
- a one to one encoding f_4 of the finite substitutions over the algebra Ω .

In the sequel, << is the relation given by

$$i \ll j \text{ iff } (\exists k)(\, f_4^{-1}(i) = f_4^{-1}(k).f_4^{-1}(j)) \ .$$

Since all the cnsidered substitutions are finite, this relation is recursive .

To any complete ω-branching labelled tree $T[.]: \mathbf{N}^* \to \mathbf{N}$ we associate the total function $T(.) : \mathbf{N} \to \mathbf{N}$ given by

$$T[i_1 \ldots i_n] = T(\langle i_1, \ldots i_n \rangle) \text{ for any word } i_1 \ldots i_n$$

in $\mathbf{N}^*$. Finally, for any k in $\mathbf{N}$, we let S_k denote the recursive relation given by :

$$j \ S_k \ i \quad \text{iff} \quad ((j = \langle i_1, \ldots, i_n \rangle) \Rightarrow (i = \langle i_1, \ldots, i_n, k \rangle)) .$$

Before we give the *formal specification* of PT, we indicate the way to encode the elementary steps of proof . In order to reflect the shifting down of the substitutions τ_i, we consider slightly modified proofs $\tau(P)$, where the substitution τ is applied to the result of the proof P .

- An elementary proof $\tau(A, \sigma A, \sigma)$ is coded by the number $\langle 1, i_0, i_1, i_2, i_3 \rangle$ where $i_0 = f_4(\tau)$, $i_1 = N(A)$, $i_2 = f_1(\sigma A)$ and $i_3 = f_4(\sigma)$.
- An n-ary inference $\tau((\tau_1, \ldots, \tau_n), r, \sigma A, \sigma)$ is coded by the number $\langle 2, i_0, i_1, i_2, i_3 \rangle$ where $i_0 = f_4(\tau)$, $i_1 = N(r)$, $i_2 = f_1(\sigma A)$, and $i_3 = f_4(\sigma)$.
- An infinitary inference $\tau((\tau_i)_{i \in N}, r, \sigma A, \sigma)$ is coded by the number $\langle 3, i_0, i_1, i_2, i_3 \rangle$ where $i_0 = f_4(\tau)$, $i_1 = N(r)$, $i_2 = f_1(\sigma A)$ and $i_3 = f_4(\sigma)$.
- The empty proof \$ has the null code $\langle 0 \rangle = 0$.

By way of definition, a function f from $\mathbf{N}$ to $\mathbf{N}$ is a member of the set PT (read "proof trees") if and only if it satisfies the properties of 'local correctness' (LC(f)) and 'well foundedness' (WF(f)) specified hereafter .

$$WF(f) \leftrightarrow (\forall g)([(\forall i)(\exists j) g(i+1) S_j g(i)] \to [(\exists i) f(g(i)) = \langle 0 \rangle])$$

$$LC(f) \leftrightarrow (\forall i)[LC_0(f,i) \text{ or } LC_1(f,i) \text{ or } LC_2(f,i) \text{ or } LC_3(f,i)]$$

$$LC_0(f,i) \leftrightarrow f(i) = \langle 0 \rangle \ \& \ (\forall j)(\forall k)[j S_k i \to f(j) = \langle 0 \rangle]$$

$$\begin{aligned} LC_1(f,i) \leftrightarrow\ & (\exists i_0)(\exists i_1)(\exists i_2)(\exists i_3)[f(i) = \langle 1, i_0, i_1, i_2, i_3 \rangle \ \& \\ & i_2 = f_1([f_4^{-1}(i_3)](N^{-1}(i_1)))] \end{aligned}$$

$$\begin{aligned} LC_2(f,i) \leftrightarrow\ & (\exists i_0)(\exists i_1)(\exists i_2)(\exists i_3)[f(i) = \langle 2, i_0, i_1, i_2, i_3 \rangle \ \& \\ & i_2 = f_1([f_4^{-1}(i_3)](f_1^{-1}.\Pi_1.f_2.N^{-1}(i_1))) \ \& \\ & (\forall n)[lg(f_2(N^{-1}(i_1))) = n+1 \to Q_2(f,n,i_1,i_3)] \end{aligned}$$

$$\begin{aligned} Q_2(f,n,i_1,i_3) \leftrightarrow\ & (\forall j)(\forall k)[j S_k i \to ([k \geq n \ \& \ f(j) = \langle 0 \rangle] \text{ or } [k < n \\ & \& \ lg(f(j)) = 5 \ \& \ \Pi_2(f(j)) \ll i_3 \ \& \\ & f_1^{-1}.\Pi_4.f(j) = [f_4^{-1}.\Pi_2.f(j)](f_1^{-1}.\Pi_{K+1}.f_2.N^{-1}(i_1))])] \end{aligned}$$

$$LC_5(f,i) \leftrightarrow (\exists i_0)(\exists i_1)(\exists i_2)(\exists i_3)[f(i) = \langle 3, i_0, i_1, i_2, i_3\rangle \;\&\; i_2 = f_1([f_4^{-1}(i_3)](f_1^{-1}.\Pi_1.f_5.N^{-1}(i_1))) \;\&\; lg(f_5(N^{-1}(i_1))) = 2 \;\&\; Q_5(f, \Pi_2.f_5.N^{-1}(i_1), i_3)]$$

$$Q_5(f,z,i_3) \leftrightarrow (\forall j)(\forall k)[jS_k i \rightarrow f_1^{-1}.\Pi_4.f(j) = [f_4^{-1}.\Pi_2.f(j)](\varphi_z(k))]$$
where φ_z is the function computed by the Turing machine of index z .

Two remarks are enough to complete the demonstration :

- The set PT of proof trees is in Π_1^1 , for it is expressed as $\{f/LC(f) \;\&\; WF(f)\}$, LC(f) and WF(f) being respectively a Π_2^0 and a Π_1^1 formula (of f) .
- The set of Σ-provable formulas of $L(\Omega)$, expressed as $\{\varphi/(\exists f)[f \in PT \;\&\; f(0) \neq \langle 0\rangle \;\&\; \varphi = \Pi_3.f(0)]\}$, is therefore in Σ_2^1 .

□

Proof of lemma 4 :

We need some definitions, which we borrow from [Rogers] (we give, for further information, the place where they are introduced in that book).

D_x is the finite set in **N** of canonical index x (p.70) . Let φ_z be the z-th recursive function in the Gödel numbering . Then $W_z = \mathrm{dom}\varphi_z$ (p.61) .

For $X \subset$ **N** , relativized functions φ_z^X may also be introduced . The motivation for the definition is as follows .

Let P'_z be the oracle machine with index z and varying oracle X (p.130) . Any branch of the computation diagram of P'_z determines two sets D' and D" in **N** defined by

D' = {w/ an affirmative answer to "is w in X?" is used on the branch }

D" = {w/ a negative answer to "is w in X?" is used on the branch }

By the **s-m-n** theorem we know that there is a recursive function h such that :

$W_{h(z)}$ = { $\langle x,y,u,v\rangle$/ a terminating branch exists in the diagram for P'_z with input x, for which $D' = D_u$ and $D'' = D_v$ and for which the output is y }

Furthermore $W_{h(z)}$ is regular in the following sense :

- for any z , W_z is *regular* (p.131-132) if and only if it contains only consistent quadruples $\langle x,y,u,v\rangle$ and no distinct compatible quadruples $\langle x,y,u,v\rangle$ and $\langle x,y',u',v'\rangle$;
- $\langle x,y,u,v\rangle$ is consistent if $D_u \cap D_v \neq \emptyset$;
- $\langle x,y,u,v\rangle$ and $\langle x,y',u',v'\rangle$ are compatible if

$$D_u \cap D_{v'} = D_{u'} \cap D_v = \emptyset$$

φ_z^X could then be defined in an obvious way from $W_{h(z)}$, but the following property (theorem II,p.132) suggests a more abstract definition :
there is a total recursive function ρ such that for all z, $W_{\rho(z)}$ is regular and if W_z is regular then $W_z = W_{\rho(z)}$.
Finally we set

$$\varphi_z^X = \{\langle x,y\rangle / \ (\exists u)(\exists v)\langle x,y,u,v\rangle \in W_{\rho(z)} \ \& \ D_u \subset X \ \& \ D_v \subset \overline{X}\}$$

If we use more than one oracle (say two since it will the case here) we define

$\varphi_z^{X,Y}$ as $\varphi_z^{X\,join\,Y}$, where Xjoiny is a coded sum of X and Y .

We say that $\varphi_z^{X,Y}(0)$ is convergent by the i-th step in the enumeration of $W_{\rho(z)}$ if the i-th element of $W_{\rho(z)}$ in the canonical enumeration of $W_{\rho(z)}$ is <u>some quadruple</u> $\langle 0,y,u,v\rangle$ satisfying $D_u \subset X\,join\,Y$ and $D_v \subset \overline{X\,join\,Y}$.

Then $T_{2,0} = \{ \langle X,Y,z,i\rangle / \ \varphi_z^{X,Y}$ is convergent by the i-th step in the enumeration of $W_{\rho(z)} \}$

and

$$T_{2',0} = \{ \langle f,g,z,i\rangle / \ \langle \tau(f),\tau(g),z,i\rangle \text{ belongs to } T_{2,0} \}$$

letting $\tau(f)$ denote the graph of f (p.337,348) .

We are now ready to give the proof of the lemma, which we split into two claims :

<u>*Claim 1*</u> : Let us suppose that there is an effective procedure reducing any Π_2^1 formula φ to a Σ_2^1 formula θ . Then there is a recursive function h such that :

(1) $(\forall f)(\exists g)(\forall i)\ \text{not}[T_{2',0}\langle f,g,z,i\rangle] \leftrightarrow (\exists f)(\forall g)(\exists i)T_{2',0}\langle f,g,h(z),i\rangle$

<u>*Claim 2*</u> : There is no recursive function h such that (1) holds .

Proof of claim 2 :

Since ρ is a total function , $\rho_{\circ}h$ is also total . So, by the recursion theorem, we know that for some n in **N**, $\varphi_n = \varphi_{\rho(h(n))}$. So we have $W_n = W_{\rho(h(n))}$ and then W_n is regular, so

$$W_n = W_{\rho(n)} = W_{\rho(h(n))} .$$

It should be clear now that for any z,f,g, $W_{\rho(z)}$ is completely determining $\varphi_n^{\tau(f),\tau(g)}$.

So we have , if h exists :

$(\exists f)(\forall g)(\exists i)$ $\varphi_n^{\tau(f),\tau(g)}(0)$ is fixed a value by the i-th element in the enumeration of $W_{\rho(n)}$

$\Longleftrightarrow$

$(\forall f)(\exists g)(\forall i)$ the i-th element in the enumeration of $W_{\rho(n)}$ gives no value to $\varphi_n^{\tau(f),\tau(g)}(0)$

which is an obvious contradiction .

Proof of claim 1 :

Let φ be the formula $(\forall f)(\exists g)(\forall i)$ not $T_{2',0}\langle f,g,z,i\rangle$. Since $T_{2',0}$ is a recursive relation (of two function variables and two number variables), φ is in Π_2^1 form . If there is a uniform procedure reducing φ to θ in Σ_2^1, there is also a uniform procedure reducing φ to the normal Σ_2^1 form

$(\exists f)(\forall g)(\exists i)$ $T_{2',0}\langle f,g,z',i\rangle$ of θ (p.376) .

Hence, there is an effective (i.e recursive) function h such that (1) is verified . □

Some conclusions

The negative result which has been established in the paper extends far beyond the case of CCS, since it depends only on the following abilities of the language for specifying behaviours :

- have the full power of Turing machines
- allow arbitrary choice in binary alternatives
- offer at least two actions for external signalling .

Our theorem indicates clearly that second order proof systems should be considered if one asks for a thorough treatment of infinite behaviours (although research in this direction points to the fact that there is no non-trivial second-order proof system in this case) . But it might also indicate that the testing equivalence we have considered is too thin , for it reflects hardly effective distinctions between Σ^1_1 sets of behaviours (e.g in the case of Σ^1_1 sets A and B such that $A \subset B$ and B-A is a singleton set not in Σ^1_1) .
This issue will be examined in a future paper .

Appendix

We recall that a Σ^1_1 set A can be canonically represented as

$$A = \{f \ / \ (\exists g)(\forall i) R(f[i], g[i], i)\}$$

where R is a recursive relation and

$$f[n] = \langle f(0), \ldots f(n) \rangle \quad .$$

Owing to that property, one may approximate f (and simutaneously g) by succesive guesses of f(i) (and g(i)) and check at each step the recursive relation R . A failure of this process will lead to the divergence of the CCS program and thus produces a word belonging to $\{0,1\}^*\tau^\omega$.

Since it is easy to define a CCS process which generates $\{0,1\}^*\tau^\omega$, it is now enough to join these two procedures by prefixing them by an arbitrary choice .

References :

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.291-317 .

Darondeau Ph., Gamatie B. *A fully observational model for infinite behaviours of communicating systems* to appear in the proceedings of CAAP 87 (Springer-Verlag) .

De Nicola R. , Hennessy M. *Testing equivalences for processes* TCS 34 (84) pp.83-133 .

Hennessy M. *Acceptance trees* J.ACM 32 (85) pp.896-928 .

Hennessy M., Plotkin G. *A Term model for CCS* Springer Verlag LNCS 88 (80) pp.261-274 .

Milner R. *A Calculus for Communicating Systems* Springer Verlag LNCS 92 (80) .

Rogers H. *Theory of Recursive Functions and effective Computability* Mac-Graw-Hill (67) .

Stirling C. *Modal Logics for Communicating Systems* University of Edinburgh , Department of Computer Science, Report CSR 193-85 (85) .

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