

Effective behaviour algebras

Philippe Darondeau, Serge Yoccoz

► **To cite this version:**

Philippe Darondeau, Serge Yoccoz. Effective behaviour algebras. [Research Report] RR-0776, INRIA. 1987. inria-00075776

HAL Id: inria-00075776

<https://hal.inria.fr/inria-00075776>

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.

INRIA

**UNITÉ DE RECHERCHE
INRIA-RENNES**

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP 105
78153 Le Chesnay Cedex
France
Té (1) 39 63 55 11

Rapports de Recherche

N° 776

**EFFECTIVE BEHAVIOUR
ALGEBRAS**

**Philippe DARONDEAU
Serge YOCCOZ**

DECEMBRE 1987

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

EFFECTIVE BEHAVIOUR ALGEBRAS

Philippe DARONDEAU* (CNRS RENNES)

Serge YOCCOZ* (CNRS BORDEAUX)

* IRISA, Campus de Beaulieu, F35042 RENNES CEDEX

**UER de Mathématiques et Informatique,

Université de Bordeaux I,

230 cours de la Libération, F33400 TALENCE

Publication interne n° 383 - Novembre 87 - 18 pages.

abstract : This paper is an attempt to define effective behaviour algebras and effective (i.e. provable) behaviour equivalences. It offers thereby a defense and illustration of the thesis asserting the identity between effective infinitary objects and \sum_1^1 sets (and transformations). In this framework, an effective equivalence of indiscernibility is defined, and its logical and topological properties are investigated.

ALGÈBRES EFFECTIVES DE COMPORTEMENTS

Résumé : Cet article tente de définir des algèbres effectives de comportements, et des équivalences effectives (c.a.d. prouvables) sur ces algèbres. Il offre par là même une défense et une illustration de la thèse selon laquelle les objets infinitaires effectifs sont essentiellement les ensembles \sum_1^1 de $\mathbb{N} \rightarrow \mathbb{N}$ (et leurs transformations). Dans ce cadre, nous introduisons une équivalence effective d'indiscernabilité, et étudions ses propriétés logiques et topologiques.

I Introduction :

Recent work on behaviour algebras such as CCS, TCSP, ACP... has renewed the study of effective transitions systems (t.s for short) and their equivalences [1,2,3,4]. We feel that these equivalences are either too liberal or too strict with respect to infinite behaviours of t.s . Indeed, equivalences really accounting for infinite behaviours fail in the following sense: they have no complete proof systems in first order logic, even with recursive infinitary rules[5], such as complete induction . Our goal is to fill in this gap .

By an effective family of t.s [7], we mean a quintuple $(S, \psi_S, A, \psi_A, f)$ where $\psi_S : S \longrightarrow \mathbb{N}$, resp. $\psi_A : A \longrightarrow \mathbb{N}$ are bijective codings for states, resp. actions , and f is a recursive function enumerating triples $\langle \psi_S(s), \psi_A(a), \psi_S(s') \rangle$, standing for transitions

$$s \xrightarrow{a} s' . \quad \text{A transition system of}$$

the family is then given by an initial state s_0 .

By an effective algebra of t.s, we mean a family of t.s whose set of states S is an algebra of terms (the operators of the algebra are then induced by the corresponding operators on terms) . In such algebras, one may introduce computations of a t.s as sequences of transitions

$$(s_i \xrightarrow{a_i} s_{i+1}) , i \leq \gamma$$

between terms, where $\gamma \in \mathbb{N} \cup \{\omega\}$. Clearly, for any term of the algebra, identified with the corresponding transition system, the set of computations of that term may be represented as a Σ_1^1 set of $(\mathbb{N} \rightarrow \mathbb{N})$, i.e a set of the form [6]

$$\{f / (\exists g) (\forall i) R(f, g, i)\} \text{ where } R \text{ is recursive relation .}$$

Further, the operators of the algebra induce operators on Σ_1^1 sets, recursive on their Σ_1^1 codes. The same holds for the alternative behaviour algebra, obtained by erasing states from computations and still holds if some actions are considered as invisible and thus are erased from traces (e.g τ in CCS).

Conversely, we know effective algebras of t.s, namely MEIJE resp. CCS, in which any Σ_1^1 set of $(\mathbb{N} \rightarrow \mathbb{N})$ represents the behaviour (complete traces resp. visible traces) of a term, given from the code of the set by a recursive procedure (for MEIJE, this results mainly from the two theorems by de Simone quoted in [7]).

This comes as a confirmation of the following thesis :

- * Effective infinitary objects are (at most) Σ_1^1
- * Effective operations on effectived infinitary objects are Σ_1^1 transformations, i.e they are union additive extensions of relations whose graph is Σ_1^1 .

On behalf of this thesis, we exclude from effective behaviour

algebras the factor algebras (of t.s) obtained by greatest bisimulations [8]: not only classes of these equivalences are Π_2^1 sets of terms, but the quotient of an effective transition system by its greatest bisimulation is not always an effective t.s .

In the sequel, we study only linear behaviour of t.s : such behaviours are essentially the Σ_1^1 sets (of $(\mathbb{N} \rightarrow \mathbb{N})$) .

It appears from the last remark that the existence of complete proof systems for behavioural equivalences of t.s reduces to the corresponding problem for equivalences between Σ_1^1 sets .

Traditionally, proof systems used in Computer Science are defined within first-order logic with recursive infinitary rules (these rules are generally used for complete induction on \mathbb{N} - algorithmic logic- or on more structured domains - Scott's complete induction). We claim that this logical framework is powerful enough to formalize infinitary properties which are not finitely approximable and thus not inductive in the usual sense . Although no complete axiomatization for equality between Σ_1^1 sets can be obtained in this logical fragment [5], the problem is trivial in any fragment of second-order logic . We introduce in the next section a coarser equivalence between Σ_1^1 sets, which is even a congruence for Σ_1^1 transformations, and show that it has a complete proof system

in RL (first order logic with recursive infinitary rules) .
 Furthermore, we believe that this equivalence is the finest
 equivalence between Σ_1^1 sets provable in RL .

II Indiscernibility as a provable congruence on Σ_1^1 sets :

1) Σ_1^1 sets and transformations :

In the sequel, f, g, \dots range over $\mathcal{N} = (\mathbb{N} \rightarrow \mathbb{N})$; A, B, \dots over $P(\mathcal{N})$.
 Similarly, i, j, \dots range over \mathbb{N} ; a, b, \dots over $P(\mathbb{N})$.

A subset A of \mathcal{N} (resp. a subset a of \mathbb{N}) is Σ_1^1 iff there
 exists a recursive relation R such that

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

$$(\text{ resp. } a = \{k / (\exists g) (\forall i) R(k, g, i)\}) .$$

A Σ_1^1 transformation Φ on $P(\mathcal{N})$ is the union additive extension
 of a function $\varphi: \mathcal{N} \longrightarrow P(\mathcal{N})$ such that

$$\{(g, f) / g \in \varphi f\} \text{ is } \Sigma_1^1 .$$

It is easy to see that Σ_1^1 transformations preserve Σ_1^1 sets .

2) Σ_1^1 indiscernibility :

For $A, B \Sigma_1^1$ subsets of \mathcal{N} , we say that A and B are Σ_1^1
 indiscernible ($A \sim_{\Sigma_1^1} B$) iff

$$\text{for any } \Sigma_1^1 \text{ subset } C \text{ of } \mathcal{N} , A \cap C = \emptyset \iff B \cap C = \emptyset .$$

Clearly, $\sim_{\Sigma_1^1}$ is an equivalence, and moreover a congruence with
 respect to Σ_1^1 transformations . This follows from two immediate
 properties :

(i) The class Σ_1^1 is closed under intersection

(ii) For any Σ_1^1 transformation Φ ,

$\Phi^{-1}(A) = \{f / (\exists g \in A) g \in \Phi(f)\}$ is a Σ_1^1 transformation .

3) Recursive logic :

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 $A \Leftarrow A_1, \dots, A_n$ with A, A_1, \dots, A_n belonging to $L(\Omega)$.

* IR is a set of recursive infinitary rules, written $A \Leftarrow A_1, \dots, A_i, \dots$ each of which is associated with an effective procedure P such that $P(i) = A_i$ for all i in \mathbb{N} .

Proofs in Σ may be defined inductively as finite-path trees whose terminal roots are the concluding step of proof .

A formula φ is provable in Σ if :

* φ is an instance of a provable formula ψ ,

* φ belongs to AX ,

* $\varphi = \sigma(A)$ and there are substitutions $\sigma_1, \dots, \sigma_n ; \tau_1, \dots, \tau_n$ such that for all i , $\sigma = \sigma_i \tau_i$ and $\tau_i(A_i)$ is provable in Σ and

$A \in A_1, \dots, A_n$ belongs to R ,

* $\varphi = \sigma(A)$ and there are substitutions σ_i, τ_i ($i \in \mathbb{N}$) such that for all i , $\sigma = \sigma_i \tau_i$ and $\tau_i(A_i)$ is provable in Σ and $A \in A_1, \dots, A_i, \dots$ belongs to IR .

4) A theorem :

Theorem : There exists a complete proof system for \sim_2 in RL .

5) Complements [6]:

* Kleene normal form theorem for Σ_1^1 sets . For $k \geq 0$ and $l \geq 0$, there exist recursive predicates

$$T'_{k+1,1}(z, f_1, \dots, f_k, g, x_1, \dots, x_l, w)$$

where z, x_1, \dots, x_l, w are integer variables, such that for any Σ_1^1 set R , $R \subseteq \mathcal{N}^k \times \mathbb{N}^l$ there is a z such that

$$R = \{ \langle f_1, \dots, f_k, x_1, \dots, x_l \rangle / (\exists g) (\forall w) \text{ not } T'_{k+1,1}(z, f, g, x, w) \}$$

z is then called a Σ_1^1 index for R .

* Π_1^1, Δ_1^1 .

A subset A of \mathcal{N} (resp. a of \mathbb{N}) is a Π_1^1 set iff $\mathcal{N} - A$ is a Σ_1^1 set (resp. $\mathbb{N} - a$ is Σ_1^1 set) .

A subset A of \mathcal{N} (resp. a of \mathbb{N}) is a Δ_1^1 set iff it is both a Σ_1^1 and a Π_1^1 set .

* Π_1^1 -completeness .

A set A is one-one reducible to a set B ($A \leq_1 B$) if there is a one-one recursive function f such that

$$(\forall x) [x \in A \iff fx \in B] .$$

A set A is Π_1^1 complete if and only if $B \leq_1 A$ for every Π_1^1 set B .

The set T of finite path trees is Π_1^1 complete (and thus the same holds for proof trees in RL) .

6) Hints for the proof of the theorem :

Notations : $\Sigma_f(x)$ is the Σ_1^1 subset of \mathcal{N} of Σ_1^1 index x .

$\Sigma_N(x)$ is the Σ_1^1 subset of \mathcal{N} of Σ_1^1 index x .

Lemma 1 : Problems $\Sigma_f(x) \sim \Sigma_f(y)$ and $\Sigma_N(x') = \Sigma_N(y')$ are effectively inter-reducible .

Notation : For any theory Θ , $RL(\Theta)$ stands for the following assertion :

" There is a complete proof system for Θ in RL "

Lemma 2 : $RL(\Sigma_N(x) = \Sigma_N(y))$ is equivalent to

$RL((\forall f)(\exists j)T'_{1,1}(z,f,j)) \& RL((\exists f)(\forall j)\text{not } T'_{1,1}(z,f,j))$

(The recursive infinitary rule

$$\Sigma_N(x) = \Sigma_N(y) \Leftarrow (i \in \Sigma_N(x) \iff i \in \Sigma_N(y))_{i \in \mathcal{N}}$$

is clearly valid and complete) .

Lemma 3 : $RL((\exists f)(\forall j)\text{not } T'_{1,1}(z,f,j))$

(Introduce the infinitary rule

$$(\exists f)(\forall j)\text{not } T'_{1,1}(z,f,j) \Leftarrow (\text{not } T'_{0,i+1}(h(x,i),y_0,\dots,y_i,i))_{i \in \mathcal{N}}$$

where $h(x,i)$ is the recursive function such that

$$T'_{1,1}(x,f,i) \iff T'_{0,i+1}(h(x,i),y_0,\dots,y_i,i) \text{ if } f(j) = y_j .$$

Lemma 4 : $RL((\forall f)(\exists j)T'_{1,1}(x,f,j))$

(Let $FPT(\varphi_z)$ mean : the recursive function φ_z of index z is the characteristic function of a finite path tree . Then

(i) there is a recursive function p such that

$$(\forall f)(\exists j)T'_{1,1}(x,f,j) \iff FPT(\varphi_{p(x)})$$

(ii) $FPT(\varphi_z)$ is provable with the help of the recursive infinitary rule :

$$FPT(\varphi_z) \Leftarrow (\varphi_z(i) = 0 \text{ or } FPT(\varphi_{h(z,i)}))_{i \in \mathbb{N}}$$

where $\varphi_{h(z,y_0)}(k) = \varphi_z(l)$ for $k = \langle y_1, \dots, y_n \rangle$ and $l = \langle y_0, \dots, y_n \rangle$

III Effective behaviour algebras :

* Summary :

On behalf of our thesis on effective infinitary objects and transformations (cf. Section I) an effective behaviour algebra must satisfy :

(i) the behaviour of a term is a Σ_1^1 subset of \mathcal{N}

(ii) the operators induced by contexts are Σ_1^1

transformations .

So any effective algebra factors through \sim_z into another effective behaviour algebra whose equality is provable - we conjecture that \sim_z is the finest equivalence with this nice property .

We delineate in the remaining of this section two generic behaviour algebras which are effective according to our thesis (and cover all models currently developed for linear behaviour of concurrent systems) :

* the first model is generic for initial linear models (sets of infinitary transitions sequences)

* the second model is generic for the usual abstraction

therefrom (e.g sets of infinitary traces) .

(a) A generic framework for operational models :

Programming languages under concern are usual term algebras with additional combinators for recursion . Among the latter, one may think of the the rec combinator used in most process algebras - e.g $\text{rec}(x).t(x)$ - or more generally of the let-rec combinator of ML .

A possible way to define effective transition systems whose states are terms of the programming language, is to follow the well known method of structured operational semantics [9] due to G.Plotkin . As far as all the side conditions of the inference rules are partial recursive (which is certainly reasonable) , this operational definition results in one global transition system per programming language, but it is clear that one may also consider the result as an effective algebra of transition systems (defined in the introduction) . Since transition systems are faithfully represented by corresponding sets of (infinitary) transition sequences, the above algebra of transition systems induces an effective poweralgebra of transition sequences .

Let the infinitary sequences of labelled transitions between terms be represented as usual in \mathcal{N} ; then, the elements of the power algebra become Σ_1^1 subsets of \mathcal{N} and the operators of the poweralgebra are recursive on their Σ_1^1 indexes . But even

better, we know from recent work that these operators are union additive extensions of operators $\varphi : \mathcal{N}^n \longrightarrow P(\mathcal{N})$ whose 'graph' $(\langle \langle x, y \rangle / y(\varphi x) \rangle)$ is Σ_1^1 - so they are Σ_1^1 transformations .

Now, as regards combinators for recursion, we have shown that they lead naturally to greatest fixpoints with respect to inclusion if the recursive definition satisfies the Greibach condition . But $\lambda U. \text{gfp}_x F(U, X)$ is a Σ_1^1 transformation if $F(U, X)$ is a Σ_1^1 transformation of U and X . To sum up, all contexts in the programming language induce Σ_1^1 transformations in the corresponding poweralgebra, which is thus an effective behaviour algebra .

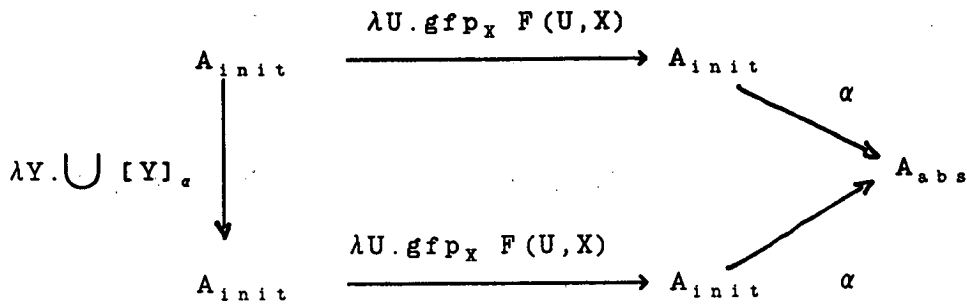
The same holds, independently of structural operational semantics, as long as the operational specification of the operators on terms (resp. of the combinators for recursion) gives rise to Σ_1^1 transformations (resp. to greatest fixpoints with respect to inclusion) .

(b) A generic framework for abstraction :

We consider here behaviour algebras derived from the aboved initial algebras through morphisms of poweralgebras, which forget irrelevant details from computations . More precisely, we assume abstraction morphisms α of the form $\alpha = \beta \circ \gamma$, where γ is any Σ_1^1 transformation (e.g erasing of states, or extraction if infinitely repeated states...) and β is a closure operator with respect to some recursive order on $\gamma(\mathcal{N})$, chosen among the

following family which reflects the three versions of powerdomains : downwards closure, convex closure, upwards closure .

Beside being morphisms of algebras, we ask our abstraction morphisms to satisfy the requirement expressed by the commutation of the diagram



- where . A_{init} is the behaviour algebra defined in II a)
- . A_{abs} is $\alpha(A_{init})$ and
- . $\lambda Y. \cup [Y]_{\alpha}$ is the operator $\lambda Y. \alpha^{-1}(\alpha(Y))$

The intuition behind the diagram is the following : given a recursively defined program $C[u]$ with subprogram u , you may as often as you wish, during an execution of $C[u]$, substitute without damage successive versions u_n of u such that $u_n \sim_{\alpha} u$, where \sim_{α} is the obvious equivalence (dynamic adaptation of concurrent programs) . Any morphism α satisfying the above yields an effective behaviour algebra A_{abs} , because the derived interpretations for recursive contexts are then Σ_1^1 transformations .

Examples : 1) γ may be the erasure of states, and β may be closure under prefix-ordering .

2) γ may also erase some actions considered as invisible (e.g τ in CCS or ACP_τ)

3) See [10] for a more complex abstraction yielding a model of CCS in which

$u = v$ iff u and v have identical sets of infinitary interactions with any program t set in parallel.

IV Topological and metric properties of indiscernibility :

a) The closure operator induced by \sim_Σ :

For A a Σ_1^1 subset of \mathcal{N} , let A^\wedge be the the union of all Σ_1^1 subsets B of \mathcal{N} such that $A \sim_\Sigma B$.

The following characterizations of A^\wedge are an easy consequence of a result by Louveau [11]:

$$A^\wedge = \bigcap \{C \mid \Pi_1^1 \text{ subset of } \mathcal{N} / \underline{ACC}\} = \bigcap \{C \mid \Delta_1^1 \text{ subset of } \mathcal{N} / \underline{ACC}\}$$

and A^\wedge is the largest Σ_1^1 set in the equivalence class of A .

Besides, the closure operator $(.)^\wedge$ satisfies de Morgan's laws .

b) Generalizing \sim_Σ into \sim_Δ :

For $A, B \Sigma_1^1$ subsets of \mathcal{N} , let $A \sim_\Delta B$ iff for any Δ_1^1 subset C of \mathcal{N} , $A \cap C = \emptyset \iff B \cap C = \emptyset$.

The above characterization shows that $A \sim_\Sigma B \iff A \sim_\Delta B$.

We consider from now on generalized definitions of relations \sim_Σ and \sim_Δ for arbitrary sets A and B . In this generalized

framework where the last equivalence is no longer valid, it appears that \sim_Δ has nicer topological properties than \sim_Σ .

For A in $P(\mathcal{N})$, let \bar{A} be the intersection of all Δ_1^1 subsets C of \mathcal{N} such that $A \subseteq C$. Then $(\bar{\cdot})$ is a closure operator inducing a topology T_Δ on \mathcal{N} , $A \sim_\Delta B \iff \bar{A} = \bar{B}$, and \bar{A} is the largest Borel set in the equivalence class of A .

c) The topology T_Δ :

T_Δ has a clopen basis, namely the family of Δ_1^1 subsets of \mathcal{N} (but all clopen sets are not Δ_1^1) .

(\mathcal{N}, T_Δ) is metrizable, but it is neither compact nor locally compact .

The family of isolated points is exactly the family of Δ_1^1 singleton sets.

The latter family is not dense, whereas the Π_1^1 singleton sets form a dense family .

d) Continuous functions :

Although we have no full characterization of continuous functions for T_Δ , we know two important classes of continuous functions, namely :

(i) functions whose graph is Σ_1^1 (thus including Σ_1^1 transformations) ;

(ii) functions whose graph is the union of a Σ_1^1 -indexed family of Δ_1^1 sets (where indexes are Δ_1^1 codes) .

In general, any continuous function is also a closed mapping, since $A \sim_{\Delta} B$ implies $fA \sim_{\Delta} fB$ for continuous f . As a consequence, \sim_{Δ} is a congruence with respect to set-extensions of T_{Δ} -continuous functions.

e) A metric for T_{Δ} :

Let x, y be in \mathcal{N} . An ultrametric distance $d(x, y)$ compatible with T_{Δ} is defined by

$$d(x, y) = 2^{-n} \text{ where } n \text{ is the smallest } \Sigma_1^1 \text{ index of a } \Delta_1^1 \text{ set } C \text{ such that } x \in C \iff y \notin C .$$

$$= 0 \text{ if there is no such separating set (} x=y \text{ in this case) .}$$

But no distance compatible with T_{Δ} makes \mathcal{N} into a complete metric space, so it makes sense to find completions of (\mathcal{N}, d) .

V Directions for further research :

A first subject of investigation is the logic RL, both from model theoretic and proof theoretic points of view. In particular, the problem of whether the indiscernibility of behaviours is liable to RL proofs in a language built around terms of behaviour algebras, is worth consideration. Another concern about indiscernibility is to show that it is really the finest equivalence (on Σ_1^1) provable in RL. We conjecture that it is also the finest testable equivalence in the sense of effective binary tests (extending De Nicola and Hennesy's tests) . Since the separating sets on which relies

indiscernibility are Δ_1^1 sets, one may imagine a strong connection between tests and Δ_1^1 behaviours. A natural question is then the operational meaning of Δ_1^1 sets. A related topic is the search for general (not necessarily profinite) specification techniques for Σ_1^1 transformations, and more generally for T_Δ -continuous functions. This supposes of course that we obtain beforehand a full characterization of continuous functions. A last topic, with possible implications in formal language theory, is the (metric) completion of (N, T_Δ) . We have some reasons to believe that bi-infinite words come here into play.

Bibliography

- [1] Milner, R. A Calculus of Communicating Systems,
- 2] De Nicola, R. Hennessy, M. Testing equivalences for processes
TCS 34 (1984)
- [3] Brook, S.D. Hoare C.A.R. Roscoe A.W A theory of
communicating sequential processes JACM 31 (1984)
- [4] Bergstra, J.A Klop, J.W Algebra of communicating
processes with abstraction TCS 37 (1985)
- [5] Darondeau, P. Yoccoz, S. Proof systems for infinite behaviours
INRIA report n°632, to appear in
Information and Computation
- [6] Rogers, H.: Theory of recursive functions and effective

computability MacGraw Hill (1967)

- [7] Boudol, G. Notes on algebraic calculi of processes
in Logics and Models of concurrent systems,
K.Apt ed. Springer Verlag (NATO SAI Series 1985)
- [8] Park, D. Concurrency and automata on infinite sequences
Springer Verlag LNCS 84 (1980)
- [9] Plotkin, G. A structural approach to operational semantics
Daimi report fn 19 University of Aarhus (1981)
- [10] Darondeau, P. Gamatie, B. A fully observational model for
infinite behaviours of communicating systems .
Springer Verlag LNCS 249 (1987)
- [11] Louveau, A. Relations d'équivalence co-analytiques
Séminaire Choquet d'Initiation à l'Analyse
16° année (1976-77) I.H.P.

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

