View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Elsevier - Publisher Connector

Theoretical Computer Science 53 (1987) 67-97
North-Holland

PROBABILISTIC IANOV'S SCHEMES*

D. FRUTOS ESCRIG

Departamento de Informática y Automática, Facultad de Matemáticas, Universidad Complutense, 28040 Madrid, Spain

Abstract. We present probabilistic Ianov's schemes, studying their semantics and proving the equivalence between operational and denotational ones. We also study the equivalence of schemes relative to them; as usual all these equivalence problems are decidable, and we prove it giving the appropriate decision algorithms.

Introduction

Ianov's schemes are, without any doubt, the simplest control systems. Therefore, when we decided to study probabilistic programs, we began by studying probabilistic Ianov's schemes. But as probabilism and nondeterminism are closely related, we studied nondeterministic Ianov's schemes first, considering the three classical (nowadays!) ways to define a semantics of nondeterministic programs: (i) Hoare's or angelical semantics (forget about the infinite computations) [1, 6, 7], (ii) Plotkin's semantics (consider all the computations) [9], and (iii) Smyth's semantics (nontermination is a disaster) [11]. In [2, 4] you can find an exhaustive study of these schemes, whose principal results will be resumed in the first section of this paper.

Then, probabilistic schemes are obtained from nondeterministic ones in the following way: an arc leaving an OR-node is labelled by a rational number, namely, the probability that a computation reaching that OR-node continues to the other end node of the arc. So, we have to complicate our schemes, but this complication is not gratuitous, as we obtain some intuitive improvements—semantics telling what is 'probable' instead of 'possible'—and some technical improvements—having a 'natural' semantics. If we consider trace semantics, we find an equivalence between denotational and operational ones, which is not always true for nondeterministic schemes.

In Section 2 we will define our probabilistic schemes and their natural semantics (operational and denotational) and prove the equivalence. Section 3 is dedicated to the study of the equivalence of schemes relative to the defined semantics. In Section 4 we will define a Smyth-like semantics of probabilistic schemes which induces an equivalence between schemes that is decidable too, although proving this turns out to be rather difficult. Finally, in Section 5 we study trace semantics;

* This paper is an extended version of the one with the same title that was presented at CAAP-86, and that appeared in Lecture Notes of Computer Science 214, (Springer, Berlin, 1986).

0304-3975/87/$3.50 © 1987, Elsevier Science Publishers B.V. (North-Holland)

in this case we need a nontrivial notion of probabilistic powerdomain as the base domain is not a flat one.

You can find more details about the subject and, in particular, about the relation between nondeterministic and probabilistic schemes in [2].

1. Nondeterministic Ianov's schemes

Definition 1.1. S is a *nondeterministic Ianov's scheme* if $S=(P, A, G, r)$, where P, A, G, and r are defined as follows:

(i) P is a set of predicate symbols.

(ii) A is a set of action symbols.

(iii) G is a directed finite graph, whose nodes have labels from the set $A \cup P \cup \{\text{STOP, OR}\}$. If a node n is labelled by $a \in A$, there is exactly one arc leaving n; if a node n is labelled $p \in P$, there are two arcs, labelled TRUE and FALSE, going out from n; and STOP-nodes are the only ones from which no arcs leave.

(iv) r is a distinguished node in G, from which all other nodes in G can be reached. r is called the *root* of the scheme.

Definition 1.2. (a) An *interpretation* I is a 3-tuple $\langle D, \varphi, \psi\rangle$, where D is a set (I's domain), $\varphi : D \times P \to \{\text{TRUE, FALSE}\}$ and $\psi : D \times A \to D$. We write $\varphi_p(d)$ for $\varphi(d, p)$ and $\psi_a(d)$ for $\psi(d, a)$.

(b) We say that I is *free* when $D = A^*$ and $\psi_a(d) = d \cdot a$.

Definition 1.3. (a) If we have a scheme $S=(P, A, G, r)$ and $n \in G$, we can define the *subscheme* $S_n = (P, A, G_n, n)$, where G_n is the complete subgraph of G, whose nodes are those that can be reached from n.

(b) We can classify the schemes s by paying attention to their roots: If r is labelled by $p \in P$ and n_1, n_2 are nodes such that $(r, n_1), (r, n_2) \in G$, $l(r, n_1) = \text{TRUE}$, $l(r, n_2) = \text{FALSE}$, we will denote the scheme S by $p(S_{n_1}, S_{n_2})$; if r is labelled by $a \in A$ and $(r, n) \in G$, we will denote S by $a(S_n)$; if r is labelled by OR and $\{n_1, \ldots, n_k\} = \{n \in G \mid (r, n) \in G\}$ we will denote S by $\text{OR}(S_{n_1}, \ldots, S_{n_k})$; and if r is labelled by STOP, we will say that $S = \text{STOP}$.

Remark. In the remainder of the paper the label of $n \in G$ will be denoted by $l(n)$, and we will use l too, to denote the Boolean label on the arcs leaving a predicate node.

Definition 1.4. (a) A *configuration* (under I) is a pair (d, S) where $d \in D$, and S is a scheme in the class of which I is an interpretation, that is,

$$I = \langle D, \varphi, \psi\rangle, \qquad S = (P, A, G, r),$$

$$\varphi : D \times P' \to \{\text{TRUE, FALSE}\}, \qquad \psi : D \times A' \to D,$$

with $P \subseteq P'$ and $A \subseteq A'$.

(b) A *computation step of* S (under I) $(d_1, S_1) \to^I (d_2, S_2)$, is a pair of configurations (d_1, S_1) and (d_2, S_2) such that

$$S_1 = \text{STOP} \Rightarrow S_2 = \text{STOP} \wedge d_1 = d_2;$$

$$S_1 = a(S_1') \Rightarrow S_2 = S_1' \wedge d_2 = \psi_a(d_1);$$

$$S_1 = p(S_1', S_1'') \Rightarrow$$
$$((\varphi_p(d_1) = \text{TRUE} \Rightarrow S_2 = S_1')$$
$$\wedge (\varphi_p(d_1) = \text{FALSE} \Rightarrow S_2 = S_1'') \wedge d_2 = d_1);$$

$$S_1 = \text{OR}(S_1', \ldots, S_k') \Rightarrow (\exists\, i \in \{1, \ldots, k\}, S_2 = S_i') \wedge d_2 = d_1.$$

(c) A *computation* c *of* S, under I, from $d \in D$ $(c \in C_I(S, d))$ is a sequence of computation steps c: $(d_1, S_1) \to^I (d_2, S_2) \to^I \cdots$, where $d_1 = d$ and $S_1 = S$. We say that c stops $(c\downarrow)$ with result $r(c)$ if there is some $i \in \mathbb{N}$ such that $S_i = \text{STOP}$ and $d_i = r(c)$.

(d) A *partial computation* $c_p \in C_I^p(S, d)$ is a finite sequence of computation steps c_p: $(d_1, S_1) \to^I \cdots \to^I (d_n, S_n)$, where $S_1 = S$ and $d_1 = d$. We say that c_p stops with result $r(c_p) = d_n$, when $S_n = \text{STOP}$.

Definition 1.5. The *operational semantics of* S (under I) are given by

$$\mathscr{S}_A(S, d) = \{d' \in D \mid \exists\, c \in C_I(S, d),\ c\downarrow \wedge r(c) = d\};$$

$$\mathscr{S}_P(S, d) = \begin{cases} \mathscr{S}_A(S, d) \cup \{\perp\}, & \text{if } \exists\, c \in C_I(S, d),\ c\uparrow, \\ \mathscr{S}_A(S, d), & \text{otherwise;} \end{cases}$$

$$\mathscr{S}_{SM}(S, d) = \begin{cases} \perp, & \text{if } \exists\, c \in C_I(S, d),\ c\uparrow, \\ \mathscr{S}_A(S, d), & \text{otherwise.} \end{cases}$$

Here $c\uparrow$ denotes $\neg c\downarrow$.

Remark. We have not made explicit the dependence from I, as we do not want to overload the notation.

To define the denotational counterparts of these operational semantics, we have to introduce the corresponding powerdomains.

Definition 1.6. (a) The angelical powerdomain $\mathscr{P}_A(D)$ is given by the powerset 2^D ordered by set inclusion; (b) Plotkin's powerdomain $\mathscr{P}(D)$ is $\mathscr{P}_F(D) \cup \mathscr{P}_\perp(D)$, where $\mathscr{P}_F(D) = \{A \in 2^D \mid |A| < \infty\}$ and $\mathscr{P}_\perp(D) = \{A \mid A = \{\perp\} \cup B, B \in 2^D\}$, ordered by Egli–Milner's order relative to the flat one over D; (c) Smyth's powerdomain $\mathscr{P}_{SM}(D)$ is the flat domain $(\mathscr{P}_F(D))_\perp$.

Definition 1.7. The denotational semantics $\mathcal{S}^{\mathrm{d}}_X$ of a scheme S, where $X \in \{\mathrm{A}, \mathrm{P}, \mathrm{SM}\}$, are defined as usual by means of a system of equations, relating the semantics of S and any one of its subschemes S_n. The minimal solution of the system gives us the denotational semantics of these schemes. The system is the following:

$$S = \mathrm{STOP} \Rightarrow \mathcal{S}^{\mathrm{d}}_X[\![S]\!](d) = \{d\};$$

$$S = a(S_1) \Rightarrow \mathcal{S}^{\mathrm{d}}_X[\![S]\!](d) = \mathcal{S}^{\mathrm{d}}_X[\![S_1]\!](\psi_a(d));$$

$$S = p(S_1, S_2) \Rightarrow$$

$$\mathcal{S}^{\mathrm{d}}_X[\![S]\!](d) = \mathrm{COND}(\varphi_p, \mathcal{S}^{\mathrm{d}}_X[\![S_1]\!], \mathcal{S}^{\mathrm{d}}_X[\![S_2]\!])(d);$$

$$S = \mathrm{OR}(S_1, \ldots, S_k) \Rightarrow \mathcal{S}^{\mathrm{d}}_X[\![S]\!](d) = \bigcup_{i=1}^{k}{}_X \, \mathcal{S}^{\mathrm{d}}_X[\![S_i]\!](d),$$

where each operator $\bigcup_X$ is defined as follows:

$$\bigcup_{\mathrm{A}} = \bigcup_{\mathrm{P}} = \bigcup \quad \text{(set union)},$$

$$\bigcup_{\mathrm{SM}} A_i = \begin{cases} \perp, & \text{if } \exists\, i \in \{1, \ldots, k\},\ A_i = \perp, \\ \bigcup A_i, & \text{otherwise.} \end{cases}$$

Theorem 1.8. *The operational and denotational semantics* $\mathcal{S}_X$ and $\mathcal{S}^{\mathrm{d}}_X$ *are equivalent for each* $X \in \{\mathrm{A}, \mathrm{P}, \mathrm{SM}\}$.

To study the equivalence of schemes, we have to introduce a normal form:

Definition 1.9. We say that S_1 and S_2 are X-equivalent ($S_1 \sim_X S_2$) for $X \in \{\mathrm{A}, \mathrm{P}, \mathrm{SM}\}$, if for any interpretation of both and for all $d \in D$, we have $\mathcal{S}_X(S_1, d) = \mathcal{S}_X(S_2, d)$.

Proposition 1.10. $S_1 \sim_X S_2$ *iff for any free interpretation of both we have* $\mathcal{S}_X(S_1, \varepsilon) = \mathcal{S}_X(S_2, \varepsilon)$.

Notation 1.11. We will say that a computation c begins executing $\alpha \in A^*$, when α is a prefix of the chain of actions executed along c ($\alpha \in \mathrm{Pre}(c)$); if c stops, we will call $\mathrm{Act}(c)$ the chain of actions executed along c; and if c loops ($c\circlearrowright$), i.e. c does not stop and only executes a finite chain of actions, we will denote the chain of actions by $\mathrm{Lact}(c)$.

Definition 1.12. We say that S_1 and S_2 are *strongly equivalent* ($S_1 \simeq S_2$) iff for any interpretation I of both and for all $d \in D$, we have

(i) $\forall \alpha \in A^*\ (\exists c \in C_I(S_1, d), \alpha \in \mathrm{Pre}(c))$

$\Leftrightarrow (\exists c' \in C_I(S_2, d), \alpha \in \mathrm{Pre}(c'))$;

(ii) $\forall \alpha \in A^*\ (\exists c \in C_I(S_1, d), c\downarrow \wedge \mathrm{Act}(c) = \alpha)$

$\Leftrightarrow (\exists c' \in C_I(S_2, d), c'\downarrow \wedge \mathrm{Act}(c') = \alpha)$;

(iii) $\forall \alpha \in A^*\ (\exists c \in C_I(S_1, d), c\circlearrowright \wedge \mathrm{Lact}(c) = \alpha)$

$\Leftrightarrow (\exists c' \in C_I(S_2, d), c'\circlearrowright \wedge \mathrm{Lact}(c') = \alpha)$.

We will not define here normal form schemes (see [2, 4]); you only need to know that it is very easy to associate to any normal form scheme S, a regular language $L(S)$ that characterizes the family of sets $\{\mathrm{Res}(c) \mid c \in C_I(S, d), c\downarrow\}$ for any interpretation I of S and any $d \in D$. Besides we have the following theorem:

Theorem 1.13. *For each Ianov's scheme S we can construct $\bar{S}$, which is in normal form and strongly equivalent to S.*

Theorem 1.14. $S_1 \sim_A S_2 \Leftrightarrow L(\bar{S}_1) = L(\bar{S}_2)$.

Corollary 1.15. *Angelical equivalence of Ianov's schemes is a decidable property.*

To decide Smyth's equivalence is more complicated: All the infinite computations c must be identified, because if such a c exists, the semantics will become $\perp$. But by means of a careful study we can identify the nodes of normal form schemes that witness the existence of such computations c, so that from $\bar{S}$ we can construct S^{SM} verifying the following theorem.

Theorem 1.16. (a) $S \sim_{SM} S^{SM}$
(b) $S_1 \sim_{SM} S_2 \Leftrightarrow S_1^{SM} \sim_A S_2^{SM}$.

Corollary 1.17. *Smyth's equivalence of Ianov's schemes is decidable.*

Finally for Plotkin's equivalence we have the following theorem.

Theorem 1.18. $S_1 \sim_P S_2 \Leftrightarrow (S_1 \sim_A S_2 \wedge S_1 \sim_{SM} S_2)$.

Corollary 1.19. *Plotkin's equivalence of Ianov's schemes is decidable.*

The last results in this section concern trace semantics of Ianov's schemes.

Definition 1.20. We define the (*free*) *trace semantics* of Ianov's schemes S (under a free interpretation I) as the function $\mathscr{S}_{SEQ}(S, \alpha)$ that associates to each $\alpha \in A^*$, the set of sequences of actions, eventually ended by STOP, executed by the elements of $C_I(S, \alpha)$.

Remark. We could define trace semantics under any arbitrary interpretation by taking sequences of elements from the domain D. But they would be more difficult to manipulate and no more general, in essence, as free interpretations are universal.

Obviously

$$\mathscr{S}_{SEQ}(S, \alpha) \subseteq S(A) = A^* \cup A^* \cdot \{\mathrm{STOP}\} \cup A^\infty,$$

so that to get a denotational counterpart of our operational semantics, we first have to consider $S(A)$ ordered by the prefix ordering and then $\mathcal{P}(S(A))$, where $\mathcal{P}$ denotes Plotkin's powerdomain. Recall that the elements of $\mathcal{P}(S(A))$ are the convex and closed closures of finitely generable subsets of $S(A)$. We will denote this closure of X by $\bar{X}$.

Once more the denotational semantics is given by a set of equations:

Definition 1.21. A *trace denotational semantics* $\mathcal{S}^{\mathrm{d}}_{\mathrm{SEQ}}[\![S]\!]$ of Ianov's schemes S is defined by

$$S = \mathrm{STOP} \Rightarrow \mathcal{S}^{\mathrm{d}}_{\mathrm{SEQ}}[\![S]\!](\alpha) = \{\mathrm{STOP}\};$$

$$S = p(S_1, S_2) \Rightarrow \mathcal{S}^{\mathrm{d}}_{\mathrm{SEQ}}[\![S]\!](\alpha) = \mathrm{COND}(\varphi_p, \mathcal{S}^{\mathrm{d}}_{\mathrm{SEQ}}[\![S_1]\!], \mathcal{S}^{\mathrm{d}}_{\mathrm{SEQ}}[\![S_2]\!])(\alpha);$$

$$S = a(S_1) \Rightarrow \mathcal{S}^{\mathrm{d}}_{\mathrm{SEQ}}[\![S]\!](\alpha) = a \cdot \mathcal{S}^{\mathrm{d}}_{\mathrm{SEQ}}[\![S_1]\!](\alpha \cdot a);$$

$$S = \mathrm{OR}(S_1, \ldots, S_k) \Rightarrow \mathcal{S}^{\mathrm{d}}_{\mathrm{SEQ}}[\![S]\!](\alpha) = \overline{\bigcup_{i=1}^{k} \mathcal{S}^{\mathrm{d}}_{\mathrm{SEQ}}[\![S_i]\!](\alpha)}.$$

And we have the following theorem.

Theorem 1.22. *For each Ianov's scheme S, and all* $\alpha \in A^*$, *we have*

$$\overline{\mathcal{S}_{\mathrm{SEQ}}(S, \alpha)} = \mathcal{S}^{\mathrm{d}}_{\mathrm{SEQ}}[\![S]\!](\alpha).$$

As $\mathcal{S}_{\mathrm{SEQ}}(S, \alpha)$ is not always convex, $\mathcal{S}_{\mathrm{SEQ}}$ and $\mathcal{S}^{\mathrm{d}}_{\mathrm{SEQ}}$ can be unequal; but nonconvexity is always due to the existence of loops in S, and these are syntactically detectable when S is in normal form. So we can define a new trace semantics $\mathcal{S}^{\mathrm{L}}_{\mathrm{SEQ}}$ that allows the new 'action' LOOP, whose meaning is "and then the computation loops". This semantics will give us subsets of

$$S^{\mathrm{L}}(A) = A^* \cdot \mathrm{LOOP} \cup A^* \cdot \mathrm{STOP} \cup A^{\infty},$$

whose prefix ordering is trivial. Then its denotational counterpart $\mathcal{S}^{\mathrm{dL}}_{\mathrm{SEQ}}$ verifies the following theorem.

Theorem 1.23. *For each Ianov's scheme S, and for all* $\alpha \in A^*$, *we have*

$$\mathcal{S}^{\mathrm{L}}_{\mathrm{SEQ}}(\bar{S}, \alpha) = \mathcal{S}^{\mathrm{dL}}_{\mathrm{SEQ}}[\![\bar{S}]\!](\alpha).$$

As $\mathcal{S}_{\mathrm{SEQ}}$ and $\mathcal{S}^{\mathrm{d}}_{\mathrm{SEQ}}$ are not equal, we have two new equivalence relations between Ianov's schemes: $\sim_{\mathrm{SEQ}}$ and $\sim^{\mathrm{d}}_{\mathrm{SEQ}}$. In fact, it is very easy to prove that they are not the same.

We have the following results:

Proposition 1.24. *By means of its normal form we can associate to each Ianov's scheme two more regular languages $L^{f}(S)$ and $L^{l}(S)$ that reflect its partial and looping computations, so that we have*

$$S_1 \sim_{\mathrm{SEQ}} S_2 \Leftrightarrow L(S_1) = L(S_2) \wedge L^{f}(S_1) = L^{f}(S_2) \wedge L^{l}(S_1) = L^{l}(S_2).$$

Corollary 1.25. *Strong equivalence is exactly SEQ-equivalence, and it is decidable.*

Proposition 1.26. *From $L^{f}(S)$ and $L(S)$ we can construct a new regular language $L^{c}(S)$ that reflects the convex closure of $\mathscr{S}_{\mathrm{SEQ}}$ to get $\mathscr{S}^{d}_{\mathrm{SEQ}}$, so that we have*

$$S_1 \sim^{d}_{\mathrm{SEQ}} S_2 \Leftrightarrow L(S_1) = L(S_2) \wedge L^{f}(S_1) = L^{f}(S_2) \wedge L^{c}(S_1) = L^{c}(S_2).$$

Corollary 1.27. *dSEQ-equivalence of Ianov's schemes is a decidable property.*

2. Probabilistic Ianov's schemes: Definitions and semantics

Definition 2.1. A *probabilistic* Ianov's scheme is a nondeterministic one, in which the arcs leaving an OR-node are labelled by probabilities (positive rational numbers), so that the sum over all the outgoing arcs is 1.

Intuitively, when a computation reaches an OR-node, it consults the value of a discrete random variable and chooses between the arcs leaving the node according to their probabilities.

Definition 2.2. (a) We will denote by S^{-} the nondeterministic scheme obtained from a probabilistic scheme S by deleting the probabilities.

(b) A *precomputation* c of S is a partial computation of S^{-}; each such c has a probability $\mathrm{prob}(c)$ that is defined as the product of the probabilities of the arcs leaving the OR-nodes chosen along c.

Example 2.3. Consider scheme S in Fig. 1 and a free interpretation I with $\varphi_p(\varepsilon) = \varphi_p(d) = \mathrm{FALSE}$, $\varphi_p(dd) = \mathrm{TRUE}$. We have amongst others the following precomputations:

$$c_1\colon\ (S_1, \varepsilon) \xrightarrow{1/2} (S_2, \varepsilon) \to (S_3, a),$$

$$\mathrm{prob}(c_1) = \tfrac{1}{2};$$

$$c_2\colon\ (S_1, \varepsilon) \xrightarrow{1/4} (S_1, \varepsilon) \xrightarrow{1/4} (S_4, \varepsilon) \to (S_6, \varepsilon) \xrightarrow{1/2} (S_8, \varepsilon) \to (S_{10}, c),$$

$$\mathrm{prob}(c_2) = \tfrac{1}{32}.$$

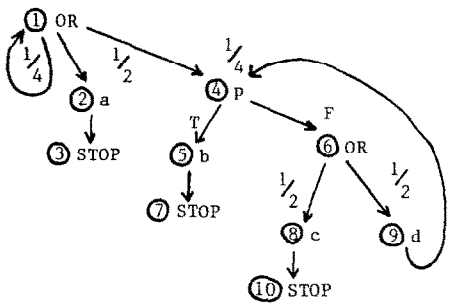

Fig. 1.

Definition 2.4. (a) We say that a precomputation $c: (S_1, d_1) \rightarrow^* (S_n, d_n)$ *stops*, if $S_n = \text{STOP}$ but $S_{n-1} \neq \text{STOP}$; d_n is its result ($\text{res}(c) = d_n$). We denote by $\text{Pc}_I(S, d)$ the set of precomputations of S (under I) from the input d and by $\text{Pcf}_I(S, d)$ its subset of stopping precomputations.

(b) Given an interpretation I of S, we define the *operational semantics of* S (under I) as the function $\mathcal{S}_{\text{PROB}}(S, d)$ that associates to each input $d \in D$ that probability distribution which assigns to each possible result $d' \in D$ the sum of the probabilities of the elements in $\text{Pcf}_I(S)$ with result d', and that associates to $\perp$ the complement to 1.

To define a denotational semantics, we need a probabilistic powerdomain.

Definition 2.5. $\mathcal{P}_{\text{PROB}}(D_\perp)$ is the domain whose elements are the discrete probability distributions over D, ordered by

$$p \sqsubseteq p' \Leftrightarrow \forall d \in D,\ p(d) \leq p'(d).$$

Proposition 2.6. $\mathcal{P}_{\text{PROB}}(D_\perp)$ *is a cpo.*

Proof. It is very easy to check that if $\{p_i \mid i \in I\}$ is a directed subset of $\mathcal{P}_{\text{PROB}}(D_\perp)$, its lub, p, is given by

$$p(d) = \sup_{i \in I} p_i(d), \quad \forall d \in D,$$

$$p(\perp) = \inf_{i \in I} p_i(\perp).$$

So that in general we have

$$p(x) = \lim_{i \in I} p_i(x), \quad \forall x \in D_\perp.$$

You can find in [2, 3] a detailed proof of a more general result. □

As usual, to define a denotational semantics we need a structural classification of schemes as is given in Definition 1.3(b), except for the case when r is labelled by OR. If r is labelled by OR, $\{(r, n_1), \ldots, (r, n_k)\}$ is the set of arcs leaving r, and each one of them is labelled by q_i, we denote S by $\text{OR}(q_1 : S_{n_1}, \ldots, q_k : S_{n_k})$.

On the other hand, by $(q_1 : x_1, \ldots, q_x : x_k)$ we will denote the finite distribution p concentrated in $\{x_1, \ldots, x_k\}$, such that $p(x_i) = q_i$, $1 \leqslant i \leqslant k$.

Definition 2.7. The *denotational semantics* of a probabilistic Ianov's scheme S (under I) is the function $\mathscr{S}^{\mathrm{d}}_{\mathrm{PROB}}[\![S]\!] : D \to \mathscr{P}_{\mathrm{PROB}}(D_\perp)$ defined by:

$$S = \mathrm{STOP} \Rightarrow \mathscr{S}^{\mathrm{d}}_{\mathrm{PROB}}[\![S]\!](d) = (1 : d);$$

$$S = a(S_1) \Rightarrow \mathscr{S}^{\mathrm{d}}_{\mathrm{PROB}}[\![S]\!](d) = \mathscr{S}^{\mathrm{d}}_{\mathrm{PROB}}[\![S_1]\!](\psi_a(d));$$

$$S = p(S_1, S_2) \Rightarrow \mathscr{S}^{\mathrm{d}}_{\mathrm{PROB}}[\![S]\!](d) = \mathrm{COND}(\varphi_p, \mathscr{S}^{\mathrm{d}}_{\mathrm{PROB}}[\![S_1]\!], \mathscr{S}^{\mathrm{d}}_{\mathrm{PROB}}[\![S_2]\!])(d);$$

$$S = \mathrm{OR}(q_1 : S_1, \ldots, q_k : S_k) \Rightarrow \mathscr{S}^{\mathrm{d}}_{\mathrm{PROB}}[\![S]\!](d) = \sum_{i=1}^{k} q_i \cdot \mathscr{S}^{\mathrm{d}}_{\mathrm{PROB}}[\![S_i]\!](d).$$

To check that $\mathscr{S}^{\mathrm{d}}_{\mathrm{PROB}}$ is well defined, we only have to check the continuity of the function $\sum_{i=1}^{k} q_i \cdot p_i$. This is an immediate consequence of the fact that finite sums and limits commute.

Theorem 2.8. *Under any interpretation, we have for each probabilistic scheme S that $\mathscr{S}^{\mathrm{d}}_{\mathrm{PROB}}[\![S]\!](d) = \mathscr{S}_{\mathrm{PROB}}(S, d)$ for all $d \in D$.*

Proof. First of all we will see that our operational semantics verifies the equations defining our denotational semantics. Indeed, it is only nontrivial to check the last equation: If $S = \mathrm{OR}(q_1 : S_1, \ldots, q_k : S_k)$ and $c \in \mathrm{Pcf}_I(S, d)$, c must begin with a step $(S, d) \to^I_{q_i} (S_i, d)$ for some $i \in \{1, \ldots, k\}$; the rest of c will be an element of $\mathrm{Pcf}_I(S_i, d)$. So each result of some $c_i \in \mathrm{Pcf}_I(S_i, d)$ is also a result of some $c \in \mathrm{Pcf}_I(S, d)$ with $\mathrm{prob}(c) = q_i \cdot \mathrm{prob}(c_i)$ and, conversely, we have

$$\mathscr{S}_{\mathrm{PROB}}(S, d) = \sum_{i=1}^{k} q_i \cdot \mathscr{S}_{\mathrm{PROB}}(S_i, d).$$

Then, as the denotational semantics is defined as the least solution of the considered system, we have

$$\mathscr{S}^{\mathrm{d}}_{\mathrm{PROB}}[\![S]\!](d) \sqsubseteq \mathscr{S}_{\mathrm{PROB}}(S, d).$$

To prove the converse inclusion we will consider approximations to the operational semantics. $\mathscr{S}^n_{\mathrm{PROB}}(S, d)$ for $n \in \mathbb{N}$ are defined in the same way as $\mathscr{S}_{\mathrm{PROB}}(S, d)$ but allowing only precomputations of length no greater than n. It is clear that

$$\mathscr{S}_{\mathrm{PROB}}(S, d) = \bigsqcup_{n \in \mathbb{N}} \mathscr{S}^n_{\mathrm{PROB}}(S, d)$$

and we can check by induction on n that

$$\mathscr{S}^n_{\mathrm{PROB}}(S, d) \sqsubseteq \mathscr{S}^{\mathrm{d}}_{\mathrm{PROB}}[\![S]\!](d),$$

for each $n \in \mathbb{N}$. Indeed, if $n = 0$ and $S \neq \mathrm{STOP}$, we have

$$\mathscr{S}^0_{\mathrm{PROB}}(S, d) = (1 : \bot) \sqsubseteq \mathscr{S}^{\mathrm{d}}_{\mathrm{PROB}}[\![S]\!](d);$$

and if $S = \mathrm{STOP}$

$$\mathscr{S}^0_{\mathrm{PROB}}(S, d) = (1 : d) = \mathscr{S}^{\mathrm{d}}_{\mathrm{PROB}}[\![S]\!](d).$$

Now, we will suppose that the result is proved for $n \leqslant k$, and we will prove it for $n = k+1$. For instance, if $S = a(S_1)$, any $c \in \mathrm{Pcf}_I(S, d)$ will begin with the step $(S, d) \rightarrow^I (S_1, \psi_a(d))$, and continue with an element of $\mathrm{Pcf}_I(S_1, \psi_a(d))$. So

$$\mathscr{S}^{k+1}_{\mathrm{PROB}}(S, d) = \mathscr{S}^k_{\mathrm{PROB}}(S_1, \psi_a(d)).$$

As

$$\mathscr{S}^k_{\mathrm{PROB}}(S_1, \psi_a(d)) \sqsubseteq \mathscr{S}^{\mathrm{d}}_{\mathrm{PROB}}[\![S_1]\!](\psi_a(d))$$

by the induction hypothesis, we can conclude the desired inclusion from the definition of the denotational semantics. On the other hand, if $S = \mathrm{OR}(q_1 : S_1, \ldots, q_r : S_r)$, we have

$$\mathscr{S}^{k+1}_{\mathrm{PROB}}(S, d) = \sum_{i=1}^{r} q_i \cdot \mathscr{S}^k_{\mathrm{PROB}}(S_i, d)$$

$$\sqsubseteq \sum_{i=1}^{r} q_i \cdot \mathscr{S}^{\mathrm{d}}_{\mathrm{PROB}}[\![S_i]\!](d) = \mathscr{S}^{\mathrm{d}}_{\mathrm{PROB}}[\![S]\!](d).$$

The other cases are very similar. □

3. Equivalence of probabilistic Ianov's schemes

As in the nondeterministic case, we will manage regular languages, although now, of course, we need probabilistic languages. They will be a version of those defined by Paz [8].

Definition 3.1. (a) A *probabilistic finite automaton* is a tuple $\mathscr{A} = (S, I, s_0, F, A)$, where S is a finite set of states; I the input alphabet; $s_0 \in S$ the initial state; $F \subseteq S$ the set of acceptor states; and A the transition function, associating to any $x \in I$ a square matrix of rational nonnegative numbers, whose dimension is the cardinality of S and such that the sum of the elements in each row is 1.

(b) We will denote by $L(\mathscr{A})$, the *language accepted by* $\mathscr{A}$. $L(\mathscr{A})$ is the set of pairs $(q : \alpha)$, where $\alpha \in A^*$ and q is the sum of the probabilities of the computations of $\mathscr{A}$ that accept α.

(c) *Probabilistic regular languages* are the languages accepted by probabilistic finite automata.

Proposition 3.2. *The equivalence of probabilistic finite automata is a decidable property.*

Proof. See [2, 8]. □

We will also need a notion of strong equivalence.

Notation. (a) $\mathrm{Pca}_I(S, d)$ is the set of precomputations whose last step executes an action (the root of S_{n-1} is labelled by some $a \in A$); $\mathrm{Pca}_I(S, d)$ also contains the precomputation of length 0.

(b) For $c \in \mathrm{Pc}_I(S, d)$, $\mathrm{Act}(c)$ will denote the sequence of actions executed along c.

(c) We say that $c \in \mathrm{Pc}_I(S, d)$ loops, if there is neither an extension of c executing more actions, nor a prefix of c with that property. We will denote by $\mathrm{Pl}_I(S, d)$ the set of all loops.

Definition 3.3. We say that S_1 and S_2 are *strongly equivalent* ($S_1 \simeq S_2$) iff under any interpretation I of both and for all $d \in D$, we have for each $\alpha \in A^*$:

(i) $$\sum_{c \in A(S_1, I, d, \alpha)} \mathrm{prob}(c) = \sum_{c \in A(S_2, I, d, \alpha)} \mathrm{prob}(c),$$

where $A(S, I, d, \alpha) = \{c \in \mathrm{Pca}_I(S, d) \mid \mathrm{Act}(c) = \alpha\}$;

(ii) $$\sum_{c \in F(S_1, I, d, \alpha)} \mathrm{prob}(c) = \sum_{c \in F(S_2, I, d, \alpha)} \mathrm{prob}(c),$$

where $F(S, I, d, \alpha) = \{c \in \mathrm{Pcf}_I(S, d) \mid \mathrm{Act}(c) = \alpha\}$;

(iii) $$\sum_{c \in L(S_1, I, d, \alpha)} \mathrm{prob}(c) = \sum_{c \in L(S_2, I, d, \alpha)} \mathrm{prob}(c),$$

where $L(S, I, d, \alpha) = \{c \in \mathrm{Pl}_I(S, d) \mid \mathrm{Act}(c) = \alpha\}$.

Remark. It is easy to check that (iii) follows from (i) and (ii), because any precomputation executing a chain of actions will loop iff it neither executes a new action nor stops. Nevertheless, we will preserve the three conditions to keep the similarity to Definition 1.12. Note that there the three conditions are independent of each other, as we can infer the probability, but not the mere 'possibility', of a loop from those of precomputations. See also the following example.

Example 3.4. In Fig. 2 you can see a probabilistic scheme S such that under any I and for all $d \in D$, we have $\mathrm{Pl}_I(S, d) = \emptyset$, since $\sum_{c \in F(S, I, d, \alpha)} \mathrm{prob}(c) = 1$. Nevertheless, S has an infinite computation that loops.

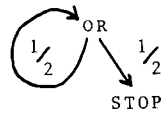

Fig. 2.

Definition 3.5. We say that two probabilistic schemes S_1 and S_2 are *semantically equivalent* ($S_1 \sim_{\text{PROB}} S_2$) iff under any interpretation I and for each $d \in D$, we have $\mathcal{S}_{\text{PROB}}(S_1, d) = \mathcal{S}_{\text{PROB}}(S_2, d)$.

Proposition 3.6. $S_1 \simeq S_2 \Rightarrow S_1 \sim_{\text{PROB}} S_2$.

Proof. As free interpretations are universal, we have that $S_1 \simeq S_2$ iff, under any free interpretation, we have $\mathcal{S}_{\text{PROB}}(S_1, \varepsilon) = \mathcal{S}_{\text{PROB}}(S_2, \varepsilon)$. This is trivially implied by Definition 3.3(ii). □

Now we introduce probabilistic schemes in normal form.

Definition 3.7. We say that a probabilistic Ianov's scheme is *in normal* form, when its graph is constituted by blocks as shown in Fig. 3. There $P = \{p_1, \ldots, p_n\}$ and each leaf h_{ij} represents either a STOP-node, or an action node from which an arc is leaving to the root of another (perhaps the same) block, or a LOOP-node representing any scheme that neither executes actions nor stops. Moreover the root of the scheme must be that of one of the blocks.

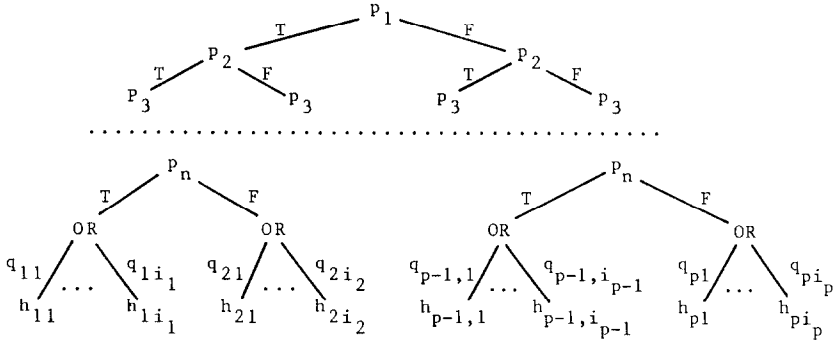

Fig. 3.

Theorem 3.8. *For each probabilistic Ianov's scheme S, we can construct an* $\bar{S}$ *in normal form, which is strongly equivalent to S.*

The proof of this theorem is based on the following lemma.

Lemma 3.9. *For each probabilistic Ianov's scheme S, we can construct an S' which is strongly equivalent to S and is in the class* IS^{p} *of inductive schemes that can be constructed by application of the following rules*:

(1) $\text{STOP} \in \text{IS}^{\text{p}}$.

(2) *If* $S \in \mathrm{IS}^{\mathrm{p}}$, *then for each* $a \in A$, $a(S) \in \mathrm{IS}^{\mathrm{p}}$. (*Formally, if* $S = (P, A, G, r)$, *then* $a(S) = (P, A, G', n)$ *where* G' *is obtained by adding a new node* n *labelled by* a *and connected to* r *by an arc.*)

(3) *If* $S_1, S_2 \in \mathrm{IS}^{\mathrm{p}}$, *then for each* $p \in P$, $p(S_1, S_2) \in \mathrm{IS}^{\mathrm{p}}$.

(4) *For all* $k \in \mathbb{N}$, *if* $S_1, \ldots, S_k \in \mathrm{IS}^{\mathrm{p}}$, $q_1, \ldots, q_k \in \mathbb{Q}^+$, and $\sum_{i=1}^{k} q_i = 1$, *then*

$$\mathrm{OR}(q_1 : S_1, \ldots, q_k : S_k) \in \mathrm{IS}^{\mathrm{p}}.$$

(5) *Finally, to allow backing arcs, when* $S = (P, A, G, r) \in \mathrm{IS}^{\mathrm{p}}$ *and* n *is a* STOP *node such that there is only one arc* (n', n) *arriving at* n, *we have* $S^{-n} = (P, A, G^{-n}, r) \in \mathrm{IS}^{\mathrm{p}}$ *where* G^{-n} *is obtained from* G *by removing* n, *and replacing the arc* (n', n) *by* (n', r).

Proof. The proof is by induction on the number k of nodes of S that are not labelled by STOP. If $k = 0$, then $S = \mathrm{STOP}$, and we can take $S' = \mathrm{STOP}$. If $k = t + 1$, we check if some of the arcs of scheme S arrive at its root. If so, we add a new STOP-node for such an arc and replace the arc by a new one reaching the added STOP-node. When all arcs of the scheme S reaching its root are replaced, we classify S according to the label of its root. For instance, if $S = p(S_1, S_2)$, it is possible that the subgraphs G_1 and G_2 defining S_1 and S_2 are not disjoint. In that case, we split their common nodes, obtaining a graph G^* that has two disjoint subgraphs G_1^* and G_2^*, so that the induced scheme $S^* = p(S_1^*, S_2^*)$ verifies $S \simeq S^*$. But it is clear that S_1^* and S_2^* have less nodes not labelled by STOP than S has, so that, by the induction hypothesis, we can construct $S_1', S_2' \in \mathrm{IS}^{\mathrm{p}}$, $S_1' \simeq S_1^*$, $S_2' \simeq S_2^*$. Now it is clear that $S' = p(S_1', S_2')$ is an inductive scheme and $S' \simeq S$.

We can reason in a similar way when $S = a(S_1)$ or $S = \mathrm{OR}(q_1 : S_1, \ldots, q_k : S_k)$.

Finally, when our original scheme S had some arc reaching its root, we have modified S getting S_1. Then we have constructed its inductive form S_1'. If we observe how this is done, we see that S_1' will have several copies of each one of the STOP-nodes that were added to get S_1, and we only have to remove those copies and redirect the arcs reaching them to the root of S_1'. So we get S', inductive scheme by rule (5) in the definition, and clearly $S \simeq S'$. □

Remark. A more formal treatment of the considered copies of the introduced STOP-nodes would be needed to get a completely formal proof. But we have avoided this, as this treatment is very similar to the development done in the proof of Theorem 3.8. Therefore, when we construct $\bar{S}$, we can suppose that $S \in \mathrm{IS}^{\mathrm{p}}$.

3.1. Construction of $\bar{S}$

$\bar{S}$ is constructed by structural recurrence, considering the way in which S is constructed as an element of IS^{p}. However, we cannot do a straightforward inductive construction, but we have to define, at the same time, a function f that associates the STOP-nodes in S to those in $\bar{S}$ such that for any STOP-node n in S we have for all $\alpha \in A^*$ and under any free interpretation I:

$$\sum_{c \in C(n, \alpha)} \mathrm{prob}(c) = \sum_{c \in \bar{C}(n, \alpha)} \mathrm{prob}(c),$$

where

$$C(n, \alpha) = \{c \in \mathrm{Pcf}_I(S, \varepsilon) \mid c \text{ stops over } n,\ \mathrm{Act}(c) = \alpha\}$$

and

$$\bar{C}(n, \alpha) = \{c \in \mathrm{Pcf}_I(\bar{S}, \varepsilon) \mid c \text{ stops over some } n' \in f^{-1}(n),\ \mathrm{Act}(c) = \alpha\}.$$

(1) If $S = \mathrm{STOP}$, $\bar{S}$ is the scheme shown in Fig. 4, and f associates to any STOP-node in $\bar{S}$, the only one in S. Note that $\bar{S}$ is defined for a fixed P.

(2) If $S = a(S_1)$, then $\bar{S}$ is the scheme shown in Fig. 5, where each $\bar{S}_1$ stands for a copy of the normal form of the subscheme S_1. As each STOP-node in $\bar{S}$ is in a copy of $\bar{S}_1$, f will associate to each of them the same node as f_1 (defined by the induction hypothesis when constructing $\bar{S}_1$) did, when they were considered as STOP-nodes of S.

(3) The case $S = \mathrm{OR}(q_1 : S_1, \ldots, q_k : S_k)$ is a trivial generalization of the previous one, and so it does not need a more extended comment.

(4) When $S = p(S_1, S_2)$, we consider $\overline{S_1}$ and $\overline{S_2}$ and set under each OR-node in the root block of $\bar{S}$ the same subscheme as that beginning in the corresponding node of $\overline{S_1}$ (resp. $\overline{S_2}$) if the OR-node is in a branch including an arc labelled by TRUE (respectively FALSE) leaving the node labelled by p_i, where $p = p_i$. The function f is defined by considering f_1 and f_2, and taking into account that S_1 and S_2 are subschemes of S.

(5) Finally, if S is obtained from S_1 by replacing (n', n) by (n', r), we will construct $\bar{S}$ by introducing some modifications in $\bar{S}_1$. We take the set N of nodes in $\bar{S}_1$ from

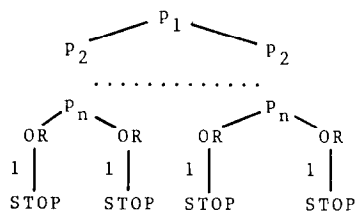

Fig. 4.

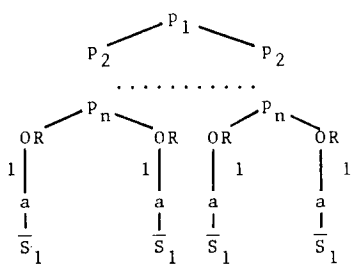

Fig. 5.

which an arc, labelled by q say, is leaving to a node in $f_1^{-1}(n)$. We replace such an arc by a new one reaching a new node that we have to add to $\bar{S}_1$. Let $n'' \in N$. We will consider the test that takes us to n'' (that is, the tuple of Boolean values of the arcs of the path from the root of n'''s block to n''). Then we take for n''' the OR-node in the root block of $\bar{S}_1$ that can be reached following that test and for $\{n_j | j \in J\}$ the set of nodes such that $(n''', n_j) \in \bar{S}_1$ with $q_j = l(n''', n_j)$. If, for all $j \in J$, $n_j \in f_1^{-1}(n'')$, then we add only one node, labelled by LOOP, that will be accessible by an arc with probability q. Otherwise, let

$$t = \sum_{f_1(n_j)=n} q_j.$$

We will consider the rest of the n_j and, for each n_j labelled by $a_j \in A$, we add a node labelled by a_j; for each node labelled by STOP, we add a new node labelled STOP; and the same for the nodes labelled by LOOP. All these arcs will be reached from n'' by an arc with probability $q \cdot q_j/(1-t)$. Finally, to define f, we observe that there are two kinds of STOP-nodes in $\bar{S}$: those that come from $\bar{S}_1$ and those added by our construction. For the former we define f by extending f_1 and, if m was added when we studied n_j, we take $f(m) = f_1(n_j)$.

Proof of Theorem 3.8. We have to show by structural induction, that $S \simeq \bar{S}$ and that the defined f verifies the condition imposed on it.

(1) If $S = \text{STOP}$, everything is obvious.

(2) If $S = a(S_1)$, we have $S_1 \simeq \bar{S}_1$ and f_1 verifies the imposed condition. To see that $S \simeq \bar{S}$, we only have to observe that for $c \in \text{Pc}_I(\bar{S}, \varepsilon)$ with $|c| \geqslant n+2$ we have $c = c_1 \cdot c_2$, where $\text{Act}(c_1) = a$, $\text{prob}(c_1) = 1$ and $c_2 \in \text{Pc}_I(\bar{S}_1, a)$. The same is true for all the computations c of S. For the same reason, if m is a STOP-node of $\bar{S}$, we consider the test that takes us to the copy of $\bar{S}_1$ in which m lies and, if, under (free) I, ε does not pass it, then it is clear that

$$\sum_{c \in C(n,\alpha)} \text{prob}(c) = 0 = \sum_{c \in \bar{C}(n,\alpha)} \text{prob}(c).$$

Otherwise, when $\alpha \neq a \cdot \beta$ for any $\beta \in A^*$,

$$\sum_{c \in C(n,\alpha)} \text{prob}(c) = 0 = \sum_{c \in \bar{C}(n,\alpha)} \text{prob}(c),$$

too. Finally, if $\alpha = a \cdot \beta$ (denoting by C_1 and $\bar{C}_1$ the corresponding sets for S_1 and $\bar{S}_1$), we have

$$\begin{aligned} \sum_{c \in C(n,\alpha)} \text{prob}(c) &= \sum_{c \in C_1(n,\beta)} \text{prob}(c) \\ &= \sum_{c \in \bar{C}_1(n,\beta)} \text{prob}(c) = \sum_{c \in \bar{C}(n,\alpha)} \text{prob}(c). \end{aligned}$$

(3) The cases $S = p(S_1, S_2)$ and $S = \text{OR}(q_1 : S_1, \ldots, q_k : S_k)$ are similar.

(4) Finally, if S is obtained from S' by replacing the arc (n', n) by (n', r), we have $S' \simeq S'$ and f' verifies the imposed condition. To prove that $S \simeq \bar{S}$, we use an

intermediate scheme S^* that is (informally) defined by Fig. 6 and observe the modifications of $\overline{S'}$ to get $\bar{S}$.

Since

$$\sum_{v=0}^{\infty} t^v \cdot q_{ij} = q_{ij}/(1-t),$$

we have $\bar{S} \simeq S^*$, because once we reached the new OR-node of S^*, we can turn around in the loop any number of times (v with probability t^v) before reaching m_j. So we only have to prove $S \simeq S^*$. Taking $S_1 = S$ and $S_2 = S^*$, to check Definition 3.3(i) (and similarly Definition 3.3(ii)) we introduce the set $A^v(S_1, I, d, \alpha)$, defined in the same way as $A(S_1, I, d, \alpha)$ but considering only precomputations of S that go exactly v times along the arc (n', r), and the set $A^v(S_2, I, d, \alpha)$, defined as $A(S_2, I, d, \alpha)$ but considering only the precomputations that cross exactly v times the OR-nodes added to $\overline{S'}$ to get S^*. Then we can prove

$$\sum_{c \in A^v(S_1, I, d, \alpha)} \text{prob}(c) = \sum_{c' \in A^v(S_2, I, d, \alpha)} \text{prob}(c'),$$

by induction on v.

If $v = 0$, the elements of $A^0(S_1, I, d, \alpha)$ are in correspondence with those of $A(S', \underline{I}, d, \alpha)$ and the elements of $A^0(S_2, I, d, \alpha)$ are in correspondence with those of $A(\overline{S'}, I, d, \alpha)$, but we know by hypothesis of the structural induction, that

$$\sum_{c \in A(S', I, d, \alpha)} \text{prob}(c) = \sum_{c \in A(\overline{S'}, I, d, \alpha)} \text{prob}(c),$$

and from this our result follows.

If $v > 0$, each $c \in A^v(S_1, I, d, \alpha)$ can be broken down in $c_1 \in \text{Pcf}(S', d)$, stopping over n with $\text{Act}(c_1) = \beta$, and $c_2 \in A^{v-1}(S_1, I, \overline{\psi_\beta}(d), \gamma)$ where $\overline{\psi_\beta}(d) = \psi_{\beta_m} \circ \cdots \circ \psi_{\beta_1}(d)$ and $\alpha = \beta\gamma$. Analogously $c' \in A^v(S_2, I, d, \alpha)$ can be divided in $c'_1 \in \text{Pcf}(S', d)$, stopping over some element of $(f')^{-1}(n)$ with $\text{Act}(c'_1) = \beta'$, and $c'_2 \in A^{v-1}(S_2, I, \overline{\psi_{\beta'}}(d), \gamma')$ where $\alpha = \beta'\gamma'$. Then, by the hypothesis of our structural induction and the induction hypothesis on v, we have

$$\sum_{c \in C'(n,\beta)} \text{prob}(c) = \sum_{c' \in \bar{C}'(n,\beta)} \text{prob}(c')$$

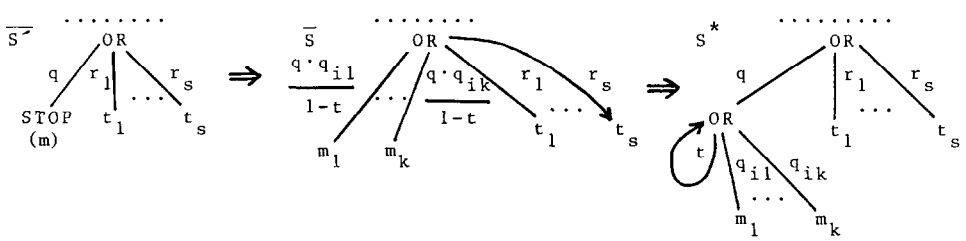

Fig. 6.

and

$$\sum_{c \in A^{v-1}(S_1, I, \bar{\psi}_\beta(d), \gamma)} \text{prob}(c) = \sum_{c' \in A^{v-1}(S_2, I, \bar{\psi}_\beta(d), \gamma)} \text{prob}(c').$$

From these facts, the desired result follows without any additional difficulty.

We can also apply the similarity between $\bar{S}$ and S^*, to prove that f verifies the imposed condition. Therefore, we define C^* as $\bar{C}$, but using S^* instead of $\bar{S}$, and obviously

$$\sum_{c \in \bar{C}(n,\alpha)} \text{prob}(c) = \sum_{c \in C^*(n,\alpha)} \text{prob}(c).$$

We can prove

$$\sum_{c \in C^{*v}(n,\alpha)} \text{prob}(c) = \sum_{c \in C^v(n,\alpha)} \text{prob}(c),$$

by following a method very similar to the one used in the previous part of this long proof. We define $C^v(n, \alpha)$ in the same way as $C(n, \alpha)$ but allowing only precomputations crossing v times the arc (n', r); and we define $C^{*v}(n, \alpha)$ as $C^*(n, \alpha)$ but considering only the precomputations that go v times through the OR-nodes added to $\bar{S}$ to get S^*. By induction on v we can prove that

$$\sum_{c \in C^v(n,\alpha)} \text{prob}(c) = \sum_{c \in C^{*v}(n,\alpha)} \text{prob}(c). \qquad \square$$

Example 3.10. In Fig. 7 you can see a probabilistic scheme S and its corresponding normal form $\bar{S}$. The reader is urged to follow the steps to get $\bar{S}$.

Using the normal form we will associate a probabilistic automaton to each probabilistic scheme.

Definition 3.11. Given a probabilistic scheme, we define $\mathfrak{A}(S)$ as the *probabilistic automaton* whose states are the nodes of $\bar{S}$, excluding its OR-nodes; its alphabet is

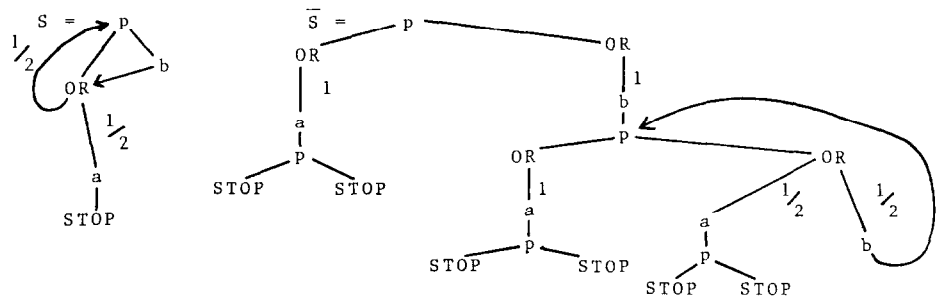

Fig. 7.

$P \cup \bar{P} \cap A$, where $P = \{p_1, \ldots, p_n\}$; s_0 is the root of $\bar{S}$; F is the set of STOP-nodes; and the transition function $f: P \cup \bar{P} \cup A \to (N \times N \to \mathbb{Q})$ is defined by

$f(p)(n, n') = 1$,

if $p \neq p_n$, $l(n) = p$, $(n, n') \in G_{\bar{S}}$, $l(n, n') = \text{TRUE}$;

$f(\bar{p})(n, n') = 1$,

if $p \neq p_n$, $l(n) = p$, $(n, n') \in G_{\bar{S}}$, $l(n, n') = \text{FALSE}$;

$f(p_n)(n, n') = l(n'', n')$,

if $l(n) = p_n$, $(n, n''), (n'', n') \in G_{\bar{S}}$, $l(n, n'') = \text{TRUE}$;

$f(\overline{p_n})(n, n') = l(n'', n')$,

if $l(n) = p_n$, $(n, n''), (n'', n') \in G_{\bar{S}}$, $l(n, n'') = \text{FALSE}$;

$f(a)(n, n') = 1$

if $l(n) = a$, $(n, n') \in G_{\bar{S}}$;

$f(x)(n, n') = 0$ otherwise.

We will denote by $L^{\text{PROB}}(S)$ the probabilistic language accepted by $\mathfrak{A}(S)$.

Remark. If S does not have predicate nodes, action nodes will play the role of p_n in this definition, and we would have to add an artificial root node.

Proposition 3.12. (a) $L^{\text{PROB}}(S)$ *accepts only pairs* (α, q) *where* $q \in \mathbb{Q} \cap (0, 1]$, *and* $\alpha = t^1 a^1 \ldots t^m a^m t^{m+1}$ *with* $a^i \in A$, $t^i = r_1^i \ldots r_n^i$, *where* $r_j^i \in \{p_j, \overline{p_j}\}$.

(b) $(\alpha, q) \in L^{\text{PROB}}(S)$ *iff under any interpretation* I *such that* $\varphi_{p_j}(a^1 \ldots a^t) = q_j^{t+1}$ *for all* $t \in \{0, \ldots, m\}$, $j \in \{1, \ldots, n\}$ we have

$$\sum_{c \in F(S, I, \varepsilon, a^1 \cdots a^t)} \text{prob}(c) = q.$$

Proof. Both are immediate consequences of the way in which $\mathfrak{A}(S)$ was defined. □

Corollary 3.13. *If* $S_1 \sim_{\text{PROB}} S_2$, *then* $L^{\text{PROB}}(S_1) = L^{\text{PROB}}(S_2)$. *So* $L^{\text{PROB}}(S)$ *is well defined, as it does not depend on the normal form chosen to define* $\mathfrak{A}(S)$.

Theorem 3.14. *Let* S_1 *and* S_2 *be two probabilistic schemes. Then we have*

$$S_1 \sim_{\text{PROB}} S_2 \Leftrightarrow L^{\text{PROB}}(S_1) = L^{\text{PROB}}(S_2).$$

Corollary 3.15. *The semantical equivalence of probabilistic Ianov's schemes is decidable.*

Proof. If we have to compare S_1 and S_2, we only have to construct their normal forms, and from them the automata $\mathfrak{A}(S_1)$ and $\mathfrak{A}(S_2)$. Finally, we can apply Proposition 3.2. □

4. Alternative treatments of nontermination in probabilistic schemes

The semantics of probabilistic schemes that we have just defined gives us the probability of any possible output and also the probability of a nonstop computation. Therefore we can classify it as a Plotkin-like semantics. Our next question would be whether we can define some reasonable angelical-like or Smyth-like semantics. The first one would give us the probability of any possible output and, then, by complement to 1, also the probability of nonstop computations. So this new semantics would be equivalent to the one earlier mentioned. On the other hand, a Smyth's semantics, defined by considering nontermination with positive probability as a disaster, would be considerably different. In the following we will study it.

Definition 4.1. (a) We will take as semantical domain $\mathscr{P}_p^{\mathrm{SM}}(D) = \mathscr{P}_p(D) \cup \{\mathrm{NT}\}$, with NT (nontermination) as its bottom element, and the order induced by the one in $\mathscr{P}_p(D_\perp)$, that is to say the trivial one: if $p_1, p_2 \in \mathscr{P}_p(D)$ and $p_1 \sqsubseteq p_2$, then $p_1 = p_2$.

(b) Given a probabilistic scheme S and an interpretation I, we define the Smyth-like semantics of S as the function $\mathscr{S}_{\text{PR-SM}}(S, d)$ given by

$$\mathscr{S}_{\text{PR-SM}}(S, d) = \begin{cases} \mathscr{S}_{\text{PROB}}(S, d), & \text{if } \sum_{d' \in D} \mathscr{S}_{\text{PROB}}(S, d)(d') = 1, \\ \text{NT}, & \text{otherwise.} \end{cases}$$

Unfortunately this semantics is not continuous relative to the natural ordering over the set of normal form schemes that is obtained by taking LOOP as bottom. This can be seen in the following example.

Example 4.2. Let S_n, $n \in \mathbb{N}$, and S be as in Fig. 8. Then, we have $(S_n)_n \to S$, but $\mathscr{S}_{\text{PR-SM}}(S_n, \varepsilon) = \text{NT}$ and $\mathscr{S}_{\text{PR-SM}}(S, \varepsilon) = ((\frac{1}{2})^n : a^{n-1} \mid n \in \mathbb{N}) \neq \text{NT}$.

Therefore, we cannot define a denotational semantics equivalent to our Smyth's semantics. And we could think that the equivalence induced by it is not decidable, arguing that in finite time we can only approximate the semantics of S by that of some S_n. Fortunately this intuitive idea is not correct, as this new equivalence between probabilistic schemes is decidable, too. Next we will prove it.

To detect nontermination with positive probability, we have to introduce the notion of 'sets of nodes with positive probability of nontermination'; these will be

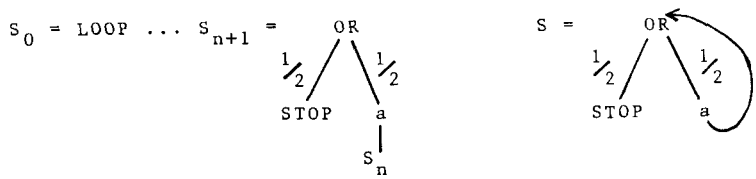

Fig. 8.

the sets of nodes such that, if the precomputations that begin by executing some chain of actions α lead us to each one of their nodes, then their extensions will not terminate with some positive probability.

Definition 4.3. We will define *sets of nodes with positive probability of nontermination* (snppnt) as those sets of OR-nodes $\{n_1, \ldots, n_k\}$ of a normal form probabilistic scheme such that, under any (free) interpretation I, there is some $i \in \{1, \ldots, k\}$ such that a precomputation in $\mathrm{Pc}_I(S_{n_i}, \varepsilon)$ does not stop with positive probability.

These sets do not depend on the probabilities over the arcs of S, but only on its nondeterministic structure. So we will introduce a similar definition for nondeterministic schemes.

Definition 4.4. We will define *sets of nodes with partial computation without completion* (snpcwc) as those sets of OR-nodes $\{n_1, \ldots, n_k\}$ of a nondeterministic scheme in normal form such that, under any (free) interpretation I, there is some $i \in \{1, \ldots, k\}$ and $c \in C_I^{\mathrm{p}}(S_{n_i}, \varepsilon)$ that cannot be extended to a final computation.

Theorem 4.5. *The snpcwc's of an nf (normal form) nondeterministic scheme can be detected by an algorithm*

The proof needs some previous lemmas.

Lemma 4.6. *Given an nf nondeterministic scheme S, and a set of roots of some blocks in S, $\{r_1, \ldots, r_k\}$, if there is some free interpretation I such that for all $i \in \{1, \ldots, k\}$, any $c_i \in C_I^{\mathrm{p}}(S_{r_i}, \varepsilon)$ can be extended to some $c_i' \in C_I(S_{r_i}, \varepsilon)$ that stops, then for each i we can find I_i with the same property as I, and such that there is some $c \in C_{I_i}(S_{r_i}, \varepsilon)$ that stops, and whose length, measured by the number of blocks that it traverses, is less or equal than some constant M that only depends on the number of blocks constituting S.*

Proof. Let B the number of blocks in S. We take as M_k the number of k-tuples that can be made up by subsets of the set $\{1, \ldots, B+2\}$ (The elements of this set represent the blocks of S and the states LOOP and STOP.) We consider the tree of computations of $(S_{r_1}, \ldots, S_{r_k})$, so that each node of the tree corresponds to some $\alpha \in A^*$ and is labelled by the tuple of sets of blocks that can be reached from r_i by some $c \in C_I(S_{r_i}, \varepsilon)$ executing α, and including LOOP or STOP if there is some prefix of α and some $c \in C_I(S_{r_i}, \varepsilon)$ executing it, looping or stopping afterwards. Then if I verifies the hypothesis, we will have for each i some $c \in C_I(S_{r_i}, \varepsilon)$ that stops. We will take $\alpha = \mathrm{Act}(c)$ and consider the node of the tree associated to α. If it is in a level below M_k, we are done, and we can take $I_i = I$. Otherwise, in the branch of the tree that takes us to the considered node, there will be some nodes labelled in the same way. If α_1, α_2 are the sequences associated to two of them, we will change

I, taking $\varphi'(\alpha_1\beta) = \varphi(\alpha_2\beta)$, for all $\beta \in A^*$, and φ' equal to φ for the rest of the chains. It is clear that I' verifies the same property as I, but now we have a node associated to a final computation $c \in C_{I'}(S_{r_i}, \varepsilon)$ in a lower level. Iterating this process, we will take c to a level below M_k, obtaining the desired interpretation. □

Corollary 4.7. *Under the same hypothesis as in Lemma* 4.6, *there will be some (free) interpretation I' with the same property as I, and such that for each $i \in \{1, \ldots, k\}$ there is some $c_i \in C_{I'}(S_{r_i}, \varepsilon)$ that stops and has length less than $k \cdot M$.*

Proof. First we apply Lemma 4.6 to r_1, obtaining I_1 such that some $c_1 \in C_{I_1}(S_{r_1}, \varepsilon)$ stops, having length $s_1 \leq M$. We will use the same tree as constructed for the proof of Lemma 4.6 pruning it at level s_1. Next we consider r_2 and check if any second components of the labels over the nodes of that level contain STOP. If so, we will have some $c_2 \in C_{I_1}(S_{r_2}, \varepsilon)$ that stops and has length less than s_1. We take $I_2 = I_1$ and go ahead with r_3. Otherwise, we will take some of the nodes with a nonempty second component, say $\{r'_1, \ldots, r'_l\}$. Obviously, we can apply Lemma 4.6 to this set, getting I'_2 such that there is some $c'_2 \in C_{I'_2}(S_{r'_1}, \varepsilon)$ that stops and has length $s_2 \leq M$. Then we take $\alpha_1 = \mathrm{Act}(c_1)$ and define I_2 by $\varphi_2(\alpha_1\beta) = \varphi'_2(\beta)$ for all $\beta \in A^*$, and φ_2 equal to φ_1 for the rest of the sequences. So c_1 remains in $C_{I_2}(S_{r_1}, \varepsilon)$ and c'_2 turns on some $c_2 \in C_{I_2}(S_{r_2}, \varepsilon)$ that stops, having length $s_1 + s_2 \leq 2 \cdot M$. Finally, we prune the new tree (under I_2) at level $s_1 + s_2$ and consider r_3. Step by step we define $I_3, \ldots, I_k$. It is clear that the last one verifies the conditions of the searched I'. □

Lemma 4.8. *If $N = \{n_1, \ldots, n_k\}$ is not an snpcwc of S, we can find an interpretation I'' with some periodicity properties (to be explained in the proof) such that for each $i \in \{1, \ldots, k\}$ any $c \in C^{\mathrm{p}}_{I''}(S_{n_i}, \varepsilon)$ can be extended to a final computation, whose length is not greater than that of c plus $2 \cdot B \cdot M$, with B and M as in Lemma* 4.6.

Proof. If N is not an snpcwc of S, there is some I such that any $c \in C^{\mathrm{p}}_I(S_{n_i}, \varepsilon)$ can be extended to a final computation. If we consider for each $a \in A$ the set R_a of roots of blocks in S that can be reached from some n_i by an arc labelled by a, then R_a verifies the hypothesis of Corollary 4.7. So we can construct a forest (finite set of trees) in the following way:

- There will be a tree for each $a \in A$, such that $R_a \neq \emptyset$, with its root labelled R_a.
- By applying Corollary 4.7 to one of these sets, R say, we will have an interpretation I_R such that for each $r \in R$ there is some $c \in C_{I_R}(S_r, \varepsilon)$ that stops with length less than $B \cdot M$. We will consider for each $\alpha \in A^*$ the sets R_α of blocks of S that can be reached by some $c \in C^{\mathrm{p}}_{I_R}(S_r, \varepsilon)$ executing α with length $B \cdot M$. Clearly, if $R_\alpha \neq \emptyset$, it verifies the hypothesis of Corollary 4.7, and we will add to the tree whose root is R a node below it labelled by R_α.
- We add, in the same way, new nodes below any of the previously introduced ones and stop the growth of each branch when it has two equally labelled nodes.

To define I'' over α we take $\alpha = a \cdot \alpha_1 \ldots \alpha_t \cdot \alpha_{t+1}$, where $|\alpha_j| = B \cdot M$ for all $j \in \{1, \ldots, t\}$ and $|\alpha_{t+1}| < B \cdot M$. We check if there is a sequence of nodes in our forest, $m_0, \ldots, m_t$, such that (1) m_0 is labelled by R_a and, for any $j \in \{0, \ldots, t-1\}$, either m_{j+1} is a son of m_j, or m_j is a leaf of some tree and m_{j+1} is a son of the other node in the same branch as m_j, labelled as m_j; and (2) α_{j+1} is the sequence of actions that made m_{j+1} appear below its father. If so, we define $\varphi''_p(\alpha) = \varphi'_p(\alpha_{t+1})$, where I' is the interpretation associated, by applying Corollary 4.7, to the label of m_t in the second step of the construction of the forest. Otherwise, we can define I'' over α in an arbitrary way. Let us check that I'' so defined verifies the lemma. Indeed, if $c \in C^p_{I''}(S_{n_i}, \varepsilon)$, $\mathrm{Act}(c) = \alpha$. We break down α as before and it is clear that c ends with some $c_1 \in C^p_{I'}(S_{n'}, \varepsilon)$ with n' in the label of m_t. Then either c_1 has an extension that stops with length less than $M \cdot B$, or it has an extension of this length that does not stop. In the first case, we are done. In the second case, we can extend the sequence $m_1, \ldots, m_t$ by a new node m_{t+1}, so that the extended sequence c_2 has the same properties as the original and c_2 ends over some n'' in the label of m_{t+1}. If I''' is the interpretation associated to m_{t+1}, we have some $c_3 \in C_{I'''}(S_{n''}, \varepsilon)$ that stops with length less than $B \cdot M$. As we have defined I'', we can extend c_2 by c_3, getting the desired extension of c. □

Now we can give the algorithm to accomplish Theorem 4.5.

Algorithm. We will consider the restrictions of free interpretations to sequences of length at most $B \cdot M$ (there are finitely many) and we will use these restrictions to construct a forest as the one defined in the proof of Lemma 4.8. We will accomplish this construction by the following backtracking procedure:

- We expand each node of the forest that is not a leaf, as defined in the aforementioned proof, by considering an interpretation I' such that (i) if $\{r_1, \ldots, r_k\}$ is the label of the considered node, we have for each $i \in \{1, \ldots, k\}$ some $c_i \in C_{I'}(S_{r_i}, \varepsilon)$ that stops with length less than $B \cdot M$; and (ii) none of the sons induced when applying I' contains LOOP in its label.

- Any node that is not a leaf and can neither be expanded as explained before will be considered as a failure node of the process, so that it will bring about a backtracking step of our construction.

If, finally, we get a forest as the one in Lemma 4.8, then N is not an snpcwc, and conversely.

Proof of Theorem 4.5. Obviously our algorithm is effective. On the other hand, if N is not an snpcwc, Lemma 4.8 tells us that some forest as constructed by the algorithm exists and, since the search is exhaustive, it will be found. Conversely, if there exists a forest as the one in the proof of the lemma, we can define the associated interpretation I'' that bears witness to N not being an snpcwc. □

Now we will relate snppnt's and snpcwc's.

Proposition 4.9. *The snppnt's of an (nf) probabilistic scheme S are exactly the snpcwc's of the associated nondeterministic scheme* S^-.

Proof. (a) Let N be an snpcwc of S^-. Then, under any I, we have an $n \in N$ and a $c \in C_I^p(S_n^-, \varepsilon)$ that cannot be extended to a final computation. But c, having some positive probability, can be considered as an element of $\mathrm{Pc}_I(S_n, \varepsilon)$ and, therefore, we have a computation of S_n, under I, that does not stop with at least the same probability. So N is an snppnt of S. (b) Conversely, if N is not an snpcwc of S^-, we have some interpretation I'' as defined in Lemma 4.8, so that each $c \in C_{I''}^p(S_n, \varepsilon)$ has an extension to a final computation whose length is less than $2 \cdot B \cdot M$ plus the length of c. This extension will be executed, when c is finished, with a probability greater than some fixed $\varepsilon > 0$ (we can take ε_0 equal to the least probability over the arc's in S and ε as $\varepsilon_0^{2 \cdot B \cdot m}$). Thus, a computation in $\mathrm{Pc}_{I''}(S_n, \varepsilon)$ will stop having length less than $2 \cdot B \cdot M$ with probability greater than ε; and if its length is greater than $k \cdot (2 \cdot B \cdot M)$, it will stop with length less than $(k+1) \cdot (2 \cdot B \cdot M)$ with probability greater than ε. So, a computation in $\mathrm{Pc}_{I''}(S_n, \varepsilon)$ will have length greater than $k \cdot (2 \cdot B \cdot M)$ with probability $(1-\varepsilon)^k$ or less. Thus it will not stop, with length greater than anything, with probability $(1-\varepsilon)^\infty = 0$. So I'' proves that N is not an snppnt of S. □

By knowing the snppnt's of S we will associate a probabilistic automaton to it.

Proposition 4.10. *Given an nf nondeterministic scheme S, we can construct a finite automaton* $\mathfrak{A}^d(S)$ *that tells us, given a word* $t^1 a^1 \dots t^k a^k t^{k+1}$, *which is the set of* OR*-nodes in S that can be reached by some* $c \in C_I^p(S, \varepsilon)$, $\mathrm{Act}(c) = \alpha$, *after passing the test* t^{k+1}, *under I, such that* $\varphi_{p_i}(a^1 \dots a^j) = q_i^{j+1}$ *for all* $j \in \{0, \dots, k\}$, $i \in \{1, \dots, n\}$, *where* $t^j = q_1^j \dots q_n^j$.

Proof. Very similar to the classical proof to get a finite automaton equivalent to a given nondeterministic automaton. □

Definition 4.11. We will associate to each (nf) probabilistic scheme S, a *probabilistic automaton* $\mathfrak{A}^{SM}(S)$ simulating $\mathfrak{A}(S)$ and $\mathfrak{A}^d(S^-)$ in parallel, so that, if the input begins by some prefix such that $\mathfrak{A}^d(S^-)$ gives for it an snppnt of S, then $\mathfrak{A}^{SM}(S)$ rejects the input; otherwise $\mathfrak{A}^{SM}(S)$ behaves as $\mathfrak{A}(S)$ having the same acceptor states. We will denote by $L^{PR\text{-}SM}(S)$ the language accepted by $\mathfrak{A}^{SM}(S)$.

Theorem 4.12. *Given nf probabilistic schemes* S_1, S_2, *we have*

$$S_1 \sim_{PR}^{SM} S_2 \Leftrightarrow L^{PR\text{-}SM}(S_1) = L^{PR\text{-}SM}(S_2).$$

Proof. ($\Rightarrow$) If $L^{\text{PR-SM}}(S_1) \neq L^{\text{PR-SM}}(S_2)$, we have an interpretation I and a sequence $\alpha \in A^*$ such that, if w is the associated word $w = t^1 a^1 \ldots t^k a^k t^{k+1}$, we have

$$(q_1 : w) \in L^{\text{PR-SM}}(S_1), \qquad (q_2 : w) \in L^{\text{PR-SM}}(S_2)$$

with $q_1 \neq q_2$ ($q_1 > q_2$ say). Then in S_1, once we have executed α and passed a final test of predicates, we arrive to a set of nodes N that is not an snppnt of S_1. The same is true for each prefix $\beta = a^1 \ldots a^l$ of α. So that, by changing I over the sequences $a^1 \ldots a^l b\gamma$ ($\forall t, \forall b \neq a^{l+1}$, and $\forall \gamma \in A^*$) and over $\alpha\gamma$ ($\forall \gamma \in A^*$), by applying the definition of snppnt, we can get I' such that under I', the computations of S_1 stop with probability 1. So $\mathcal{S}_{\text{SM-PR}}(S_1, \varepsilon) \neq \text{NT}$ and, as I' is equal to I over α and its prefixes, we also have

$$\mathcal{S}_{\text{SM-PR}}(S_1, \varepsilon)(\alpha) = q_1.$$

But for the same reason we would have

$$\mathcal{S}_{\text{SM-PR}}(S_2, \varepsilon) = \text{NT} \quad \text{or} \quad \mathcal{S}_{\text{SM-PR}}(S_2, \varepsilon)(\alpha) = q_2.$$

In any case,

$$\mathcal{S}_{\text{SM-PR}}(S_1, \varepsilon) \neq \mathcal{S}_{\text{SM-PR}}(S_2, \varepsilon),$$

so that $S_1 \not\simeq_{\text{PR}}^{\text{SM}} S_2$.

($\Leftarrow$) If $S_1 \not\simeq_{\text{PR}}^{\text{SM}} S_2$, we will have some I such that under I

$$\mathcal{S}_{\text{SM-PR}}(S_1, \varepsilon) \neq \mathcal{S}_{\text{SM-PR}}(S_2, \varepsilon).$$

But, then, one of them must be NT, since otherwise $L^{\text{PR-SM}}(S_1)$ and $L^{\text{PR-SM}}(S_2)$ would reflect the unequal values of these semantics. We will suppose

$$\mathcal{S}_{\text{SM-PR}}(S_1, \varepsilon) = \text{NT} \neq \mathcal{S}_{\text{SM-PR}}(S_2, \varepsilon).$$

But, then, $L^{\text{PR-SM}}(S_2)$ reflects the values of $\mathcal{S}_{\text{SM-PR}}(S_2, \varepsilon)$ over the sequences in A^* and their sum must be 1. This cannot be true for S_1, since then a computation of S_1 would stop with probability 1. This would contradict $\mathcal{S}_{\text{SM-PR}}(S_1, \varepsilon)$ being equal to NT. □

Corollary 4.13. *SM-equivalence between probabilistic Ianov's schemes is decidable.*

Proof. You only have to apply Theorems 3.8 and 4.5, Definition 4.11, Theorem 4.12 and Proposition 3.2. □

5. Trace semantics of probabilistic Ianov's schemes

First of all we have to introduce a new probabilistic powerdomain as now our original domain, $S(A)$ with its prefix ordering, is not a flat domain.

Definition 5.1. (a) *Scott's topology over* $S(A)$ is the one that has as basis the family

$$\{U_\alpha \mid \alpha \in A^* \cup A^* \,.\, \mathrm{STOP}\},$$

where $U_\alpha = \{\beta \in S(A) \mid \alpha \sqsubseteq \beta\}$.

(b) *Cantor's topology over* $S(A)$ is the one that has as subbasis the family

$$\{U_\alpha \mid \alpha \in A^* \cup A^* \,.\, \mathrm{STOP}\} \cup \{N_\alpha \mid \alpha \in A^* \cup A^* \,.\, \mathrm{STOP}\},$$

where $N_\alpha = S(A) - U_\alpha$.

(c) *Borel's* σ*-algebra over* $S(A)$, $\mathscr{B}(S(A))$, is the one induced by Scott's (eq. Cantor's) topology.

(d) *Probability distributions over* $S(A)$ are the *probability measures* over $\mathscr{B}(S(a))$.

Definition 5.2. The *operational trace semantics* of a probabilistic Ianov's scheme S is given by the function $\mathscr{S}_{\text{PR-SEQ}}(S, \alpha)$ associating to each $\alpha \in A^*$ the function that gives for each $\beta \in A^* \cup A^* \,.\, \mathrm{STOP}$ the probability with which a computation of $\mathrm{Pc}_I(S, \alpha)$ begins executing β; that is, the sum of the probabilities of the precomputations that execute β, executing its last action in its last step.

Definition 5.3. From $\mathscr{S}_{\text{PR-SEQ}}$ we can define $\mathscr{S}'_{\text{PR-SEQ}}$ that associates to each pair (S, α), a function defined over the basis of Cantor's topology over $S(A)$ as:

$$\mathscr{S}'_{\text{PR-SEQ}}(S, \alpha)(X) = \sum_{i=1}^{n} \mathscr{S}_{\text{PR-SEQ}}(S, \alpha)(\gamma_i) - \sum_{j=1}^{m} \mathscr{S}_{\text{PR-SEQ}}(S, \alpha)(\eta_j),$$

where

$$X = \{\beta \in S(A) \mid (\beta \sqsupseteq \gamma_1 \vee \cdots \vee \beta \sqsupseteq \gamma_n) \wedge \beta \not\sqsupseteq \eta_1 \wedge \cdots \wedge \beta \not\sqsupseteq \eta_m\},$$

with $\{\gamma_i\}$ and $\{\eta_j\}$ two sets of incomparable elements, such that

$$\forall j \in \{1, \ldots, m\} \quad \exists i \in \{1, \ldots, n\} \quad \gamma_i \sqsubseteq \eta_j,$$

but

$$\forall j \in \{1, \ldots, m\} \quad \neg \exists i \in \{1, \ldots, n\} \quad \gamma_i \sqsupseteq \eta_j.$$

Remark. It is not difficult to check that any X in the basis of Cantor's topology over $S(A)$ can be defined as above.

Proposition 5.4. $\mathscr{S}'_{\text{PR-SEQ}}$ *is well defined and finitely additive.*

Proof. We will denote by $X_{\{\gamma_i\}\{\eta_j\}}$ the set

$$X = \{\beta \in S(A) \mid (\beta \sqsupseteq \gamma_1 \vee \cdots \vee \beta \sqsupseteq \gamma_n) \wedge \beta \not\sqsupseteq \eta_1 \wedge \cdots \wedge \beta \not\sqsupseteq \eta_m\}.$$

We have to check that

$$X_{\{\gamma_i\}\{\eta_j\}} = X_{\{\gamma'_i\}\{\eta'_j\}} \Rightarrow \{\gamma_i\} = \{\gamma'_i\} \wedge \{\eta_j\} = \{\eta'_j\}$$

for we use the fact that the γ_i are minimal elements in X and the hypothesis, in Definition 5.3, about the families $\{\gamma_i\}$ and $\{\eta_j\}$. So $\mathcal{S}'_{\text{PR-SEQ}}$ is well defined. To check additivity we take $X = X_{\{\gamma_i\}\{\eta_j\}}$ and $X' = X_{\{\gamma'_i\}\{\eta'_j\}}$ so that $X \cap X' = \emptyset$ and $X \cup X' = X''$ where $X'' = X_{\{\gamma''_i\}\{\eta''_j\}}$; the general case would be similar. It is easy to check that X can be broken down into X_i's given by

$$X_i = \{\beta \sqsupseteq \gamma_i \wedge \forall j\ (\eta_j \sqsupseteq \gamma_i \Rightarrow \beta \not\sqsupseteq \eta_j)\},$$

so that the X_i's are pairwise disjoint, and the η_j's get classified since for each one of them there is exactly one γ_i such that $\gamma_i \sqsubseteq \eta_j$. Now, we have

$$\mathcal{S}'_{\text{PR-SEQ}}(S, \alpha)(X) = \sum_{i=1}^{n} \mathcal{S}'_{\text{PR-SEQ}}(S, \alpha)(X_i).$$

To prove the proposition, we can suppose that $\{\gamma_i\}$ and $\{\gamma'_i\}$ are unitary sets. Then, we will be in one of two cases: (i) γ_1 and γ'_1 are incomparable; (ii) $\gamma_1 \sqsubseteq \gamma'_1$ (or conversely). In (i) η_j and η'_j are incomparable, so that

$$X'' = X_{\{\gamma_1, \gamma'_1\}, \{\eta_j\} \cup \{\eta'_j\}}$$

and, clearly,

$$\mathcal{S}'_{\text{PR-SEQ}}(S, \alpha)(X'') = \mathcal{S}'_{\text{PR-SEQ}}(S, \alpha)(X) + \mathcal{S}'_{\text{PR-SEQ}}(S, \alpha)(X').$$

In (ii), as sets in the basis of the topology are convex, we would have a η_k equal to γ'_1, so that

$$X'' = X_{\{\gamma_1\}, \{\eta_1, \ldots, \eta_{k-1}, \eta_{k+1}, \ldots, \eta_m\} \cup \{\eta'_j\}},$$

then, to compute $\mathcal{S}'_{\text{PR-SEQ}}(S, \alpha)(X'')$ we have to add and subtract $\mathcal{S}_{\text{PR-SEQ}}(S, \alpha)(\eta_k)$ to the sum of $\mathcal{S}'_{\text{PR-SEQ}}(S, \alpha)(X)$ and $\mathcal{S}'_{\text{PR-SEQ}}(S, \alpha)(X')$. □

Corollary 5.5. *We can extend $\mathcal{S}'_{\text{PR-SEQ}}$ to $\mathcal{B}(S(A))$ getting a measure $\mathcal{S}''_{\text{PR-SEQ}}$.*

Proof. It is obvious that the sets in the basis of Cantor's topology are clopen and, as this topological space is compact (see [9]), there will be no infinite family of nonempty and disjoint sets from that basis, whose union is a new set of it. So $\mathcal{S}'_{\text{PR-SEQ}}$ is a measure over that basis. This basis is a semiring. So $\mathcal{S}_{\text{PR-SEQ}}$ can be extended (see [2] for details) to the σ-algebra generated by Cantor's topology, that is $\mathcal{B}(S(A))$. □

Remark. We have defined an 'operational' semantics of every scheme. We have quoted the word 'operational', since, strictly speaking, we can only expect that $\mathcal{S}_{\text{PR-SEQ}}$ and $\mathcal{S}'_{\text{PR-SEQ}}$ are operational, as $\mathcal{B}(S(A))$ is not a denumerable set. And we would even have some problems, when we try to compute $\mathcal{S}_{\text{PR-SEQ}}(S, \alpha)(\beta)$ directly, as an infinite sum would be sometimes required. However, this sum becomes a finite one, when S is in normal form. We can suppose this, as we can effectively obtain the normal form of every scheme.

Next we will define a denotational semantics which equivalent to $\mathscr{S}_{\text{PR-SEQ}}$.

Definition 5.6. The *domain of probability distributions* over $S(A)$, $\mathscr{P}_{\text{p}}(S(A))$, has as elements the probability measures over $\mathscr{B}(S(A))$, ordered by $\sqsubseteq$, which is defined as follows

$$m \sqsubseteq m' \text{ iff } \forall \alpha \in A^* \cup A^* \cdot \text{STOP}$$
$$m(U_\alpha) \leqslant m'(U_\alpha).$$

Proposition 5.7. *$\mathscr{P}_{\text{p}}(S(A))$ is a cpo.*

Proof. First of all $\sqsubseteq$ is an ordering: If we have $m, m' \in \mathscr{P}_{\text{p}}(S(A))$ such that $m(U_\alpha) = m'(U_\alpha)$ for all $\alpha \in A^* \cup A^* \cdot \text{STOP}$, then for $X = X_{\{\gamma_i\}\{\eta_j\}}$ we have

$$m(X) = \sum m(U_{\gamma_i}) - \sum m(U_{\eta_j})$$
$$= \sum m'(U_{\gamma_i}) - \sum m'(U_{\eta_j}) = m'(X).$$

So m and m', restricted to the basis of Cantor's topology, are equal and, as this basis is a semiring generating the σ-algebra $\mathscr{B}(S(A))$, the extension algorithm used in the proof of Corollary 5.5 tells us that $m = m'$. If $\{m_i \mid i \in I\}$ is a directed subset of $\mathscr{P}_{\text{p}}(S(A))$, we can take $m(U_\alpha) = \sup_{i \in I} m_i(U_\alpha)$ for all $\alpha \in A^* \cup A^* \cdot \text{STOP}$, extending it to $\mathscr{B}(S(A))$, as in Definition 5.3 and Corollary 5.5, to obtain a measure since it is clear that m is finitely additive. Obviously, m is the lub of $\{m_i \mid i \in I\}$ in $\mathscr{P}_{\text{p}}(S(A))$.

Definition 5.8. The denotational trace semantics of probabilistic schemes is given by the function $\mathscr{S}^{\text{d}}_{\text{PR-SEQ}}$, defined by means of the following set of equations:

$$\mathscr{S}^{\text{d}}_{\text{PR-SEQ}}[\![\text{STOP}]\!](\alpha) = (1 : \text{STOP});$$

$$\mathscr{S}^{\text{d}}_{\text{PR-SEQ}}[\![p(S_1, S_2)]\!](\alpha) = \text{COND}(\varphi_p, \mathscr{S}^{\text{d}}_{\text{PR-SEQ}}[\![S_1]\!], \mathscr{S}^{\text{d}}_{\text{PR-SEQ}}[\![S_2]\!])(\alpha);$$

$$\mathscr{S}^{\text{d}}_{\text{PR-SEQ}}[\![a(S_1)]\!](\alpha) = a \cdot \mathscr{S}^{\text{d}}_{\text{PR-SEQ}}[\![S_1]\!](\alpha \cdot a);$$

$$\mathscr{S}^{\text{d}}_{\text{PR-SEQ}}[\![\text{OR}(q_1 : S_1, \ldots, q_k : S_k)]\!](\alpha) = \sum_{i=1}^{k} q_i \cdot \mathscr{S}^{\text{d}}_{\text{PR-SEQ}}[\![S_i]\!](\alpha).$$

Here $a \cdot m$ is the measure defined by $a \cdot m(X) = m(X - a)$, with

$$X - a = \{\beta \in S(A) \mid a \cdot \beta \in X\}.$$

Proposition 5.9. *$\mathscr{S}^{\text{d}}_{\text{PR-SEQ}}$ is well defined.*

Proof. We have to check that the operators $a \cdot m$ and $\sum_i q_i \cdot m_i$ are continuous. But these are immediate consequences of the way the order between probability distributions is defined and the results of Proposition 5.7. □

Theorem 5.10. *For each probabilistic Ianov's scheme S, under any free interpretation I, and for all* $\alpha \in A^*$, *we have*

$$\mathscr{S}''_{\text{PR-SEQ}}(S, \alpha) = \mathscr{S}^{\text{d}}_{\text{PR-SEQ}}[\![S]\!](\alpha).$$

Proof. To prove that two probability distributions are the same, it is sufficient to check that their values over the sets U_β are equal. So we will prove that

$$\mathscr{S}_{\text{PR-SEQ}}(S, \alpha)(\beta) = \mathscr{S}^{\text{d}}_{\text{PR-SEQ}}[\![S]\!](\alpha)(U_\beta).$$

First, we will prove that, if we consider the set of equations that defines $\mathscr{S}^{\text{d}}_{\text{PR-SEQ}}[\![S]\!]$ and, for any subscheme S' of S, we have a function $f_{S'}$ associating to every $\alpha \in A^*$ a 'measure' over the family $B = \{U_\beta \mid \beta \in A^* \cup A^* \cdot \text{STOP}\}$ (that is to say, the restriction to B of a measure over $\mathscr{B}(S(A))$) so that these functions verify the mentioned set of equations, then their extensions, $f^*_{S'}$, to $\mathscr{B}(S(A))$ verify the system, too. The check is very simple and we will only show the third condition:

$$f^*_{a(S_1)}(\alpha) = a \cdot f^*_{S_1}(\alpha \cdot a).$$

But

$$f^*_{a(S_1)}(\alpha)(U_\beta) = f_{a(S_1)}(\alpha)(U_\beta),$$

and

$$a \cdot f^*_{S_1}(\alpha \cdot a)(U_\beta) = f^*_{S_1}(\alpha \cdot a)(U_\beta - a).$$

We will distinguish two cases: $\beta = a \cdot \beta'$ and $\beta = b \cdot \beta'$ with $b \neq a$. In the first we have

$$\begin{aligned} f^*_{S_1}(\alpha \cdot a)(U_\beta - a) &= f^*_{S_1}(\alpha \cdot a)(U_{\beta'}) \\ &= f_{S_1}(\alpha \cdot a)(U_{\beta'}), \end{aligned}$$

and

$$\begin{aligned} f_{a(S_1)}(\alpha)(U_\beta) &= a \cdot f_{S_1}(\alpha \cdot a)(U_\beta) \\ &= f_{S_1}(\alpha \cdot a)(U_{\beta'}). \end{aligned}$$

On the other hand, if $\beta = b \cdot \beta'$, then

$$\begin{aligned} a \cdot f_{S_1}(\alpha \cdot a)(U_\beta) &= 0 \\ &= f^*_{S_1}(\alpha \cdot a)(\phi) \\ &= f^*_{S_1}(\alpha \cdot a)(U_\beta - a). \end{aligned}$$

But it is clear that the function $\mathscr{S}_{\text{PR-SEQ}}(S', \alpha)$ verifies the set of equations that defines $\mathscr{S}^{\text{d}}_{\text{PR-SEQ}}[\![S']\!](\alpha)$, so that $\mathscr{S}''_{\text{PR-SEQ}}(S', \alpha)$ verifies it, too. Thus, we have

$$\mathscr{S}_{\text{PR-SEQ}}(S, \alpha)(\beta) \geqslant \mathscr{S}^{\text{d}}_{\text{PR-SEQ}}[\![S]\!](\alpha)(U_\beta)$$

for all $\beta \in A^* \cup A^* \cdot \text{STOP}$.

The converse inequalities will be proved by introducing the functions $\mathscr{S}^n_{\text{PR-SEQ}}(S, \alpha)$ approximating $\mathscr{S}_{\text{PR-SEQ}}(S, \alpha)$ and defined in the same way, but considering only precomputations of length n at most. It is clear that

$$\mathscr{S}_{\text{PR-SEQ}}(S, \alpha)(\beta) = \sup_{n \in \mathbb{N}} \mathscr{S}^n_{\text{PR-SEQ}}(S, \alpha)(\beta)$$

and we only have to prove that

$$\mathscr{S}^n_{\text{PR-SEQ}}(S, \alpha)(\beta) \leqslant \mathscr{S}^{\text{d}}_{\text{PR-SEQ}}[\![S]\!](\alpha)(U_\beta)$$

for all $n \in \mathbb{N}$. This is proved by an easy induction over n. For instance, if $S = \text{OR}(q_1 : S_1, \ldots, q_k : S_k)$, we have

$$\begin{aligned}
\mathscr{S}^{n+1}_{\text{PR-SEQ}}(S, \alpha)(\beta) &= \sum_{i=1}^{k} q_i \cdot \mathscr{S}^n_{\text{PR-SEQ}}(S, \alpha)(\beta) \\
&\leqslant \sum_{i=1}^{k} q_i \cdot \mathscr{S}^{\text{d}}_{\text{PR-SEQ}}[\![S_i]\!](\alpha)(U_\beta) \\
&= \mathscr{S}^{\text{d}}_{\text{PR-SEQ}}[\![S]\!](\alpha)(U_\beta). \qquad \square
\end{aligned}$$

Finally we will study the equivalence of probabilistic schemes relative to trace semantics. We will introduce, as in Proposition 1.23, a new probabilistic language associated to every probabilistic scheme.

Definition 5.11. Let S be a probabilistic Ianov's scheme. We define the *probabilistic regular language* $L^{\text{f}}(S)$ as the language accepted by the automaton $\mathfrak{A}^{\text{f}}(S)$, that is defined as $\mathfrak{A}(S)$, but taking as F the set of all the action states in $\mathfrak{A}(S)$.

Proposition 5.12. *Let* $\alpha = t^1 a^1 \ldots t^m a^m t^{m+1}$ *with* $a^i \in A$, $t^i = r^i_1 \ldots r^i_n$, *where* $P = \{p_1, \ldots, p_n\}$ *and* $r^i_j \in \{p_j, \overline{p_j}\}$. *Then, if I is a free interpretation such that*

$$\forall i \in \{0, \ldots, m\} \quad \forall j \in \{1, \ldots, n\} \quad \varphi_{p_j}(a^1 \ldots a^i) = r^i_j,$$

we have that $(q : \alpha) \in L^{\text{f}}(S)$ *iff S under I executes a computation that begins executing the sequence of actions* $a^1 \ldots a^m$ *with probability q.*

Proof. As $\mathfrak{A}^{\text{f}}(S)$ is defined, it simulates the precomputations of $\bar{S}$ under arbitrary free interpretations. And as S and $\bar{S}$ are strongly equivalent, they execute a precomputation that begins executing some given chain of actions with the same probability.

Definition 5.13. Let S_1 and S_2 be two probabilistic Ianov's schemes. We say that they are *trace equivalent* ($S_1 \sim^{\text{PR}}_{\text{SEQ}} S_2$) iff under any free interpretation we have

$$\mathscr{S}^{\text{d}}_{\text{PR-SEQ}}[\![S_1]\!] = \mathscr{S}^{\text{d}}_{\text{PR-SEQ}}[\![S_2]\!].$$

Theorem 5.14. $S_1 \sim^{\mathrm{PR}}_{\mathrm{SEQ}} S_2 \Leftrightarrow L(S_1) = L(S_2) \wedge L^{\mathrm{f}}(S_1) = L^{\mathrm{f}}(S_2)$.

Proof. ($\Rightarrow$): Let $\beta \in A^*$, we have

$$\mathscr{S}_{\text{PR-SEQ}}(S_1, \varepsilon)(\beta) = \mathscr{S}_{\text{PR-SEQ}}(S_2, \varepsilon)(\beta)$$

under any free interpretation. So that, if $\alpha = t^1 a^1 \dots t^m a^m t^{m+1}$, we can take I such that

$$\forall i \in \{0, \dots, m\} \quad \forall j \in \{1, \dots, n\} \quad \varphi_{p_j}(a^1 \dots a^i) = r^i_j.$$

Then we have

$$(q : \alpha) \in L^{\mathrm{f}}(S_1) \Leftrightarrow (q : \alpha) L^{\mathrm{f}}(S_2).$$

To prove that $L(S_1) = L(S_2)$, we can argue as before but considering $\beta \in A^* \cdot \mathrm{STOP}$.

($\Leftarrow$): We only have to reverse the reasoning in the proof of ($\Rightarrow$). □

Corollary 5.15. *Trace equivalence of probabilistic Ianov's schemes is decidable.*

Remark. In the nondeterministic case, we need another language, $L^{\mathrm{l}}(S)$, characterizing looping computations, to achieve Proposition 1.23. We could define such a language for probabilistic schemes, too. But as our remark after Definition 3.3 shows, we would have

$$L(S_1) = L(S_2) \wedge L^{\mathrm{f}}(S_1) = L^{\mathrm{f}}(S_2) \Rightarrow L^{\mathrm{l}}(S_1) = L^{\mathrm{l}}(S_2),$$

that is because such a language is superfluous in the probabilistic case.

Corollary 5.16. *Trace equivalence and strong equivalence of probabilistic Ianov's schemes, are the same property.*

6. Conclusion

We have seen that all the results about the semantics of nondeterministic Ianov's schemes in Section 1 (see also [2, 4]) can be translated to probabilistic schemes. We have even obtained some results that are not true in the nondeterministic case (for instance, Theorem 5.10 and the result in the remark after Corollary 5.15). Nevertheless there is an important result about nondeterministic schemes that cannot be translated to the probabilistic case. Roughly it says that any nondeterministic scheme admits a 'determined' normal form, $\bar{\bar{S}}$, verifying that, if two partial computations of $\bar{\bar{S}}$ execute the same chain of actions, executing the last one in its last step, then they must be equal. This property makes the proof of Theorem 1.15 easier since we do not need to work with sets of nodes, as in the proof of Theorem 4.12, to obtain the result of Corollary 4.13. Unfortunately, there are some probabilistic schemes that do not admit a 'determined' normal form, as was proved in [2].

Therefore, the use of sets of nodes in the aforementioned proof is mandatory. If S_1 and S_2 were in dnf, we could ask ourselves if it is easier to decide if they are Smyth equivalent. The answer is yes; you can find the corresponding decision algorithm in [2]. Moreover, we have found an algorithm to construct the dnf of a scheme, if one exists; so that 'to have a dnf' is a decidable property. This construction will appear in a forthcoming paper. On the other hand, the proof of the algorithm to decide Smyth's equivalence of dnf schemes is interesting by itself. It is a rather beautiful proof, at least in the biased eyes of this author. It is based on the use of the so-called 'static interpretations', which are defined by choosing a fixed answer for each predicate node in the scheme, so that the scheme becomes a new one without predicate nodes.

Another interesting subject is the complexity of all the given algorithms (see also [5, 10]).

Acknowledgment

Finally, I want to thank K. Indermark, who directed my Ph.D. Thesis and suggested the subject of this paper, and M. Nivat, who encouraged me to write this extended version after reading the preliminary version at CAAP-86.

References

[1] J. Engelfriet, Simple program schemes and formal languages, in: Lecture Notes in Computer Science **20** (Springer, Berlin, 1974).

[2] D. Frutos Escrig, Algunas cuestiones relacionadas con la semántica de construcciones probabilísticas, Tesis Doctoral, Fac. C. Matemáticas, Univ. Complutense, Madrid, July 1985.

[3] D. Frutos Escrig, Some probabilistic powerdomains in the category SFP, in: *Proceedings STACS-86*, Lecture Notes in Computer Science **210** (Springer, Berlin, 1986).

[4] D. Frutos Escrig and K. Indermark, Equivalence relations of nondeterministic Ianov's schemes, Schriften zur Informatik und Angewanten Mathematik, RWTH Aachen, 1987.

[5] S. Hart, M. Sharir and A. Pnueli, Termination of probabilistic concurrent programs, *ACM TOPLAS* **5** (3) (1983) 356–380.

[6] Y. Ianov, The logical schemes of algorithms, *Problemy Kibernet.* **1** (1960) 82–140.

[7] K. Indermark, On a class of schematic languages, in: R. Aguilar, ed., *Formal Languages and Programming* (North-Holland, Amsterdam, 1976) 1–13.

[8] A. Paz, *Introduction to Probabilistic Automata* (Academic Press, New York, 1971).

[9] G. Plotkin, A powerdomain construction, *SIAM J. Comput.* **5** (3) (1976) 452–487.

[10] L. E. Rosier and Hsu-Chun Yen, On the complexity of deciding fair termination of probabilistic concurrent finite-state programs, in: *Proceedings ICALP-86,* Lecture Notes in Computer Science **226** (Springer, Berlin, 1986).

[11] M. B. Smyth, Powerdomains, in: Lecture Notes in Computer Science **45** (Springer, Berlin, 1978) 537–543.