Looking for a Cheaper ROSA

Fernando L. Pelayo*, Fernando Cuartero, and Diego Cazorla

Departamento de Sistemas Informáticos
Escuela Superior de Ingenieria Informática
Universidad de Castilla-La Mancha
02071-Albacete, Spain
{FernandoL.Pelayo,Fernando.Cuartero,Diego.Cazorla}@uclm.es

Abstract. Process Algebras, PAs, are formalisms able to capture the behaviour of a computing system by, for example, giving the labelled transition system, LTS, where states are nodes and where all possible evolutions of the system are arcs; The drawing of the complete LTS is a NP-complete task, so that, the reaching of a particular 'desired' state is a problem which deserves some heuristic for improving the amount of resources to be carried out. In this line, Artificial Intelligence by means of Genetic Algorithms (GA's), provides metaheuristic techniques that have obtained good results in problems in which exhaustive techniques fail due to the size of the search space, as it is the exploration of a LTS.

In this paper, we try to avoid this problem, so only unfolding the most promising (for the task of reaching a 'goal' state) branches within the LTS.

Keywords: Process Algebra, Genetic Algorithm, Complexity.

1 Introduction

Artificial intelligence, AI, can be seen as the intelligence of machines and the branch of computer science that aims to create it. AI studies and designs intelligent agents, i.e., systems that perceive its environment and take actions that maximize its chances of success.

These agents can be categorized into several kinds according to the type of problems to solve or according to the strategies to follow. One of the typical problems to work in, is searching for a particular state among a lot of them. Genetic Algorithms, GAs [5,4], are strategies to be followed in order to solve AI problems, specially when the knowledge of the environment is not strong enough to easily guide the searching process. In fact, although they have been widely used to solve problems in the fields of combinatorial and numerical optimization, it is very rare to find them used dealing with the problem of improving the computational cost of analyzing via Process Algebras [10].

ROSA is a Markovian process algebra "functionally" close to PNAL [2]. Markovian time is added by means of the inclusion of actions whose duration is

* Research partially supported by projects TIN2009-14312 & CGL2007-66440-C04-03.

J. Cabestany, I. Rojas, and G. Joya (Eds.): IWANN 2011, Part II, LNCS 6692, pp. 380–387, 2011.

modelled by Exponentially distributed random variables of parameters $\lambda \in \mathbb{R}^+ - \{0\}$ and immediate actions, whose duration can be modelled by $Exp[\infty]$. There are some other differences between **ROSA** and PNAL as the order when solving the non-deterministic choices against the probabilistic ones or the inclusion of non-determinism when cooperating some type of actions.

ROSA [8] does not impose any syntactical restrictions on the components of a parallel operator, and thus, the specification labour becomes easier than in some other models.

The usefulness of **ROSA**, as well as of so many PAs is out of any doubt, but as exposed, the computational cost of the unfolding of the whole LTS is unbroachable from a practical perspective, so that we propose a way to only unfold the more promising states among the reachable (through a single transition) set of states from a given one. This, of course, would mean a saving on the computational cost of producing the LTS by the operational semantics of **ROSA**, in this sense we entitled this paper *Looking for a Cheaper ROSA*.

This paper is structured as follows: next 2 sections provide rough descriptions of the Markovian process Algebra **ROSA** and of a generic Genetic Algorithm, respectively. Then a topology structure over the set of **ROSA** processes is defined and that *promising function* which is claimed to make **ROSA** a "cheaper formalism" is finally presented.

2 The Markovian Process Algebra ROSA

Let $\Delta = \{a, b, c, \ldots\}$ be an ordered finite set of action types.

Let $Id = \{X, Y, Z, \ldots\}$ be a finite set of variables of process.

We will denote by the latest letters of the latin alphabet $r, s, t, \ldots$ probabilities.

We will denote by greek letters $\alpha, \beta, \gamma, \ldots$ time parameters for actions.

Terms of **ROSA** are defined by the following BNF expression:

$$P ::= \mathbf{0} \mid X \mid a.P \mid \langle a, \lambda \rangle.P \mid P \oplus P \mid P + P \mid P \oplus_r P \mid P||_A P \mid recX : P$$

where $\lambda \in \mathbb{R}^+ - \{0\}$, $A \subseteq \Delta$, $a \in \Delta$, $X \in Id$, . is concatenation, $\oplus$, $+$ and $\oplus_r$ are internal, external and probabilistic choices, $r \in [0, 1]$, $||$ is parallel, *rec* stands for recursion and P is a process of **ROSA**.

The Algebra induced by this expression makes up the set of **ROSA** processes.

A detailed description of the operational semantics and the performance evaluation algorithm of **ROSA** can be found in [9], where with the aim of making **ROSA** a more usable formalism, some steps have been done in the line of fully automatize its analyzing skills.

3 A Basic Genetic Algorithm

Although there are different types of GA's, they all share the following three processes: selection, reproduction and evaluation. The algorithm repeats these processes cyclically until a stop condition is reached. In [7], the authors have developed a first approximation to the problem we are dealing with, including:

– A generic description of a basic GA
– A formal definition of the reproduction operators
– A **ROSA** specification of the referred GA
– A complete performance study of this GA

In this paper we are concerned with the proper definition of the evaluation (of population in GAs) process. So that, we propose a metric on the states space, to be taken as basis for the selection (of the more promising individuals to conform the new population in GAs) process and therefore preventing to generate all branches of the LTS of **ROSA**.

4 Towards a Cheaper ROSA

Our main goal is to improve **ROSA** to be able to solve problems in a cheaper way, even automatically as in the line followed by [9]. In order to do this, our next step towards a Genetic Process Algebra is to define a function that, given a final state, associates to each state/process a measure of how promising is such state as being path to reach the final one. This function will be named *promising function, p-f* (we hope that the definition of the definitive fitness function could take this as basis).

We adopt the Means-End policy which tries to minimize the distance between the present state and the final one. In order to do that, following the reference [11], given P and Q a pair of **ROSA** processes, our metric takes as basis the Bayre metric and is defined as follows

$$d(P,Q) = \frac{1}{2^{l(P \sqcap Q)}} - \frac{1}{2^n}$$

– $l(P)$ is the length of the process P and is defined inductively over the syntactic structure of **ROSA** processes, as follows

$$\begin{array}{lcl} \mathbf{l} : \{\mathbf{ROSA}\ procs.\} & \longrightarrow & \mathbb{N} \\ \mathbf{0} & \mapsto & 0 \\ X & \mapsto & 1 \\ a.P & \mapsto & 2 + l(P) \\ \langle a, \lambda \rangle.P & \mapsto & 2 + l(P) \\ P \oplus Q & \mapsto & l(P) + 1 + l(Q) \\ P + Q & \mapsto & l(P) + 1 + l(Q) \\ P \oplus_r Q & \mapsto & l(P) + 1 + l(Q) \\ P||_A Q & \mapsto & l(P) + 1 + l(Q) \\ recX : P & \mapsto & 2 + l(P) \\ (P) & \mapsto & 2 + l(P) \end{array}$$

– $n = max\{l(P), l(Q)\}$
– $P \sqcap Q$ is the longest common initial part of processes P and Q

Theorem 1. *The function d so defined is a metric.*

Proof. "d is a metric over $\{$**ROSA** *processes*$\} \Leftrightarrow d$ holds $(1) \wedge (2) \wedge (3)$" where:

1. $\forall P, Q \in \{\textbf{ROSA}\ processes\}.d(P,Q) = 0 \Leftrightarrow P = Q$
2. $\forall P, Q \in \{\textbf{ROSA}\ processes\}.d(P,Q) = d(Q,P)$
3. $\forall P, Q, T \in \{\textbf{ROSA}\ processes\}.d(P,Q) \leq d(P,T) + d(T,Q)$

1. $d(P,Q) = 0 \Leftrightarrow \frac{1}{2^{l(P \sqcap Q)}} - \frac{1}{2^n} = 0 \Leftrightarrow l(P \sqcap Q) = n \Leftrightarrow P = Q$
2. $d(P,Q) = d(Q,P) \Leftrightarrow \frac{1}{2^{l(P \sqcap Q)}} - \frac{1}{2^n} = \frac{1}{2^{l(Q \sqcap P)}} - \frac{1}{2^n} \Leftrightarrow$
   $\Leftrightarrow P \ \sqcap \ Q = Q \ \sqcap \ P \Leftrightarrow \sqcap\ symmetry$
3. $d(P,Q) \leq d(P,T) + d(T,Q) \Leftrightarrow d(P,T) + d(T,Q) - d(P,Q) \geq 0 \Leftrightarrow$
   $\Leftrightarrow \frac{1}{2^{l(P \sqcap T)}} - \frac{1}{2^m} + \frac{1}{2^{l(T \sqcap Q)}} - \frac{1}{2^o} - \frac{1}{2^{l(P \sqcap Q)}} + \frac{1}{2^n} \geq 0 \Leftrightarrow$
   $\Leftrightarrow (\frac{1}{2^n} - \frac{1}{2^m} - \frac{1}{2^o}) + (\frac{1}{2^{l(P \sqcap T)}} + \frac{1}{2^{l(T \sqcap Q)}} - \frac{1}{2^{l(P \sqcap Q)}}) \geq 0$
   where:
   - $m = max\{l(P), l(T)\}$
   - $o = max\{l(T), l(Q)\}$
   - $n = max\{l(P), l(Q)\}$

   where either one can be less than the other two (**A**), or all the same (**B**)
   A : **Let's assume** $n < (m = o)$

$$n < m \Leftrightarrow n \leq m - 1 \Leftrightarrow 2^n \leq 2^{m-1} \Leftrightarrow \frac{1}{2^n} \geq \frac{1}{2^{m-1}} \Leftrightarrow$$

$$\Leftrightarrow \frac{1}{2^n} - \frac{1}{2^{m-1}} \geq 0 \Leftrightarrow \frac{1}{2^n} - \frac{1}{2^m} - \frac{1}{2^m} \geq 0 \Leftrightarrow \frac{1}{2^n} - \frac{1}{2^m} - \frac{1}{2^o} \geq 0$$

$$\sqcap\ transitivity \Leftrightarrow l(P \sqcap Q) \geq min\{l(P \sqcap T), l(T \sqcap Q)\} \Leftrightarrow$$

$$\Leftrightarrow (l(P \sqcap Q) \geq l(P \sqcap T)) \vee (l(P \sqcap Q) \geq l(T \sqcap Q)) \Leftrightarrow$$

$$\Leftrightarrow (2^{l(P \sqcap Q)} \geq 2^{l(P \sqcap T)}) \vee (2^{l(P \sqcap Q)} \geq 2^{l(T \sqcap Q)}) \Leftrightarrow$$

$$\Leftrightarrow (\frac{1}{2^{l(P \sqcap Q)}} \leq \frac{1}{2^{l(P \sqcap T)}}) \vee (\frac{1}{2^{l(P \sqcap Q)}} \leq \frac{1}{2^{l(T \sqcap Q)}}) \Leftrightarrow$$

$$\Leftrightarrow (\frac{1}{2^{l(P \sqcap T)}} - \frac{1}{2^{l(P \sqcap Q)}} \geq 0) \vee (\frac{1}{2^{l(T \sqcap Q)}} - \frac{1}{2^{l(P \sqcap Q)}} \geq 0) \Rightarrow$$

$$\Rightarrow \frac{1}{2^{l(P \sqcap T)}} + \frac{1}{2^{l(T \sqcap Q)}} - \frac{1}{2^{l(P \sqcap Q)}} \geq 0$$

   The case where $m < (n = o)$ or equivalently $o < (m = n)$, has a very similar proof.
   B : The proof is also valid here

Once it has been checked, some considerations must be made mainly over the property (1) $\forall P, Q \in \{\textbf{ROSA}\ processes\}.d(P,Q) = 0 \Leftrightarrow P = Q$.

Since both P and Q are just **ROSA**-syntactical expressions denoting processes, some distinctions on these syntactical expressions can affect processes with the same meaning, i.e., two syntactical-different processes not always represent two different processes in terms of their behaviours, let us see some examples:

Example 1. Let P and Q be a pair of **ROSA** processes, we need that processes $P \oplus Q$ and $Q \oplus P$ have distance 0, because in whatever interpretation of the semantics of processes, they should be equivalent. The same could be said about the processes $P + Q$ and $Q + P$, so this commutative property should be preserved. Moreover, the weighted commutative property of $\oplus_r$ should be also fulfilled, thus $P \oplus_r Q$, has to be equivalent to $Q \oplus_{1-r} P$, or more precisely the distance between them must be 0 in a correct definition of distance. □

Example 2. Furthermore, the definition of distance should also respect the associativity of the processes so that, given P, Q and R three **ROSA** *processes* we want that $d((P \oplus Q) \oplus R, P \oplus (Q \oplus R)) = 0$. In this line the associativity of $+$ and the weighted associativity of $\oplus_r$ has to be preserved. □

Example 3. Also, there are some cases in which distributive property must be satisfied. For instance, let us take P, Q and R as **ROSA** *processes*, then we want that $d((P \oplus Q) + R, (P + R) \oplus (Q + R)) = 0$.

Distributive is a difficult property to be studied and guaranteed, thus, we will follow the results presented in [2], and the corresponding distributive laws. □

Example 4. Finally, we want that derivative operators could be removed, and then, the equivalent expression without them should have distance 0 with the previous one. For instance we want that $d(a.0||_{\emptyset}b.0, a.b.0 + b.a.0) = 0$. □

In fact, we want that in an appropriate semantics, two equivalent processes would have distance 0 between them. The main objective of this paper is not the study of a theoretical semantics, such as denotational or axiomatic semantics. Of course, with the basis of our operational semantics, we could define a notion of bisimulation ([1,3]), and take this equivalence as the basis. But this is a considerable amount of effort, and this work have been already done. In fact, in [2] a **Proof System** is defined, and it is demonstrated the equivalence of a denotational semantics and a set of axioms and inference rules, in the sense that this system is sound and complete. That is, if two processes have the same denotational semantics, then, it can be proved by using the proof system that they are equivalent, and on the contrary, if the equivalence may be proved in the proof system, then, the processes have the same denotational semantics.

In order to solve all the cases shown in the above examples we need to introduce normal forms for **ROSA** processes.

Normal Forms

In the line of [2], we can define normal forms in a very natural way. They consist in a generalized probabilistic choice at the top, followed by a generalized internal choice between a set of states, which is followed by a generalized prefixed external choice between the actions (timed and immediate) in this set, whose continuations are also in normal form.

Definition 1. *(Normal forms)*

- *Process* ***0*** *is in normal form.*
- *If* $\mathcal{A}_i$ *is a convex set of sets of* $\Delta \times (0, +\infty) \cup \{\infty\}$ *and for every* $a \in \mathbf{Type}(A_j)$ *(see [8]) where* $A_j \in \mathcal{A}_i$ *there is a normal form* $nf(P_{A_j,a})$*, then*

$$\bigotimes_i [q_i] \bigoplus_{A_j \in \mathcal{A}_i} \mathop{+\!\!\!/}_{a \in \mathbf{Type}(A_j)} \langle a, \lambda_{A_j} \rangle . nf(P_{A_j,a})$$

is a normal form.

Notice that immediate action a is denoted in normal form as $\langle a, \infty \rangle$ and $\otimes_i [q_i], i \in \{1, \ldots, n\}$ represents the n-extension of $\oplus_r$ in this way:

- $P \oplus_r Q$ will be represented by $[r]P \otimes [1-r]Q$
- $P \oplus_r (Q \oplus_s T)$ will be represented by $[r]P \otimes [(1-r) * s]Q \otimes [(1-r) * (1-s)]T$

As usual, normal forms are unique modulo associativity and commutativity. Nevertheless we need to impose more restrictions in order to have one and only one normal form for every process, i.e., we want that two processes such as $a+b$ and $b+a$ have the same normal form, for instance, $a+b$.

We need then to impose some restrictions related to the order in which actions, sets and probabilities appear in the normal form. These restrictions are the following:

- At external choice level, actions must appear in alphabetical order.
- At internal choice level, sets must appear in the induced lexicographic order.
- At probabilistic level, probabilities must appear in decreasing order. If two would have the same probability then the lexicographic order of their already ordered internal choice level processes will determinate.

Let us see an example. The longest common initial part of the following two processes is 0:

$$((\langle d,1\rangle.0 + \langle a,2\rangle.0) \oplus (\langle b,1\rangle.0 + \langle a,3\rangle.0)) \oplus_{0.3} (\langle f,\infty\rangle.0 \oplus \langle e,1\rangle.0)$$
$$(\langle e,1\rangle.0 \oplus \langle f,\infty\rangle.0) \oplus_{0.7} ((\langle a,3\rangle.0 + \langle b,1\rangle.0) \oplus (\langle d,1\rangle.0 + \langle a,2\rangle.0))$$

Nevertheless both processes share the same *ordered* normal form:

$$[0.7]\langle e,1\rangle.0 \oplus \langle f,\infty\rangle.0 \otimes [0.3]\langle a,3\rangle.0 + \langle b,1\rangle.0 \oplus \langle a,2\rangle.0 + \langle d,1\rangle.0$$

so their distance must be 0.

Notation: The *ordered* normal form of process P will be denoted by $\|P\|$

We assume as equal **ROSA** processes, every pair of them which have the same corresponding ordered normal forms:

$\forall P, Q \in \{\mathbf{ROSA}\ \mathit{processes}\}. P = Q \Leftrightarrow \|P\| = \|Q\|$

It is a sound assumption since in [2] an equivalent proof of the soundness of the pure functional behaviour of **ROSA** can be found, and in [6] a complete Proof System for Timed Observations is presented.

In this section we will omit the treatment of recursion, because it implies an important mathematical apparatus so requiring a considerable amount of space, and the result does not justify this effort. This is due to the fact that for defining correctly a normal form for infinite processes, we need a power domain, as well as an order relation, so that, an infinite process would be the limit of a chain of ascending finite processes, each of them, an approximation of this limit. In order to guarantee the existence of this limit, we need both to introduce a fixed point theory, and to proof that every operator is continuous.

Since we think that this considerable work is not interesting in our study, we leave for a future work the completion of this operator, and we address the interested reader to the paper [2], where it is defined the semantics for infinite processes in a similar syntax to **ROSA**. Thus, from now on, operator $recX : P$ is not considered.

Once the notion of *ordered normal form* is defined it is time to provide the metric which solve all the problems previously stated.

Definition 2. *Given a pair of **ROSA** processes P and Q the distance between them is $D(P,Q)$*

$$D(P,Q) = \frac{1}{2^{l(\|P\| \sqcap \|Q\|)}} - \frac{1}{2^N}$$

where:

- $l(\|P\|)$ *is the length of the process* $\|P\|$ *and is defined inductively over the syntactic structure of ordered normal forms of **ROSA** processes, as follows*

$$\begin{array}{lll} l : \|\{\textbf{\textit{ROSA}}\ procs.\}\| & \longrightarrow & \mathbb{N} \\ \textbf{\textit{0}} & \mapsto & 0 \\ a.P & \mapsto & 2 + l(P) \\ \langle a, \lambda \rangle.P & \mapsto & 2 + l(P) \\ +_{a \in A_j} P_a & \mapsto & m - 1 + \sum_{a \in A_j} l(P_a) (m = |A_j|) \\ \bigoplus_{A_j \in \mathcal{A}_i} P_j & \mapsto & k - 1 + \sum_{A_j \in \mathcal{A}_i} l(P_j) (k = |\mathcal{A}_i|) \\ \bigotimes_{i \in \{1 \ldots n\}} [q_i] P_i & \mapsto & 2n - 1 + \sum_{i \in \{1 \ldots n\}} l(P_i) \\ (P) & \mapsto & 2 + l(P) \end{array}$$

- $N = max\{l(\|P\|), l(\|Q\|)\}$

Promising Function, $\mathbf{p} - \mathbf{f}$, gives higher values to the more promising states to be followed for reaching S_F:

$$\begin{array}{lll} \mathbf{p} - \mathbf{f} : \{\textbf{ROSA}\ procs.\} & \longrightarrow & (0, 1] \\ P & \mapsto & 1 - D(P, S_F) \end{array}$$

Finally, our proposal is, given an initial state S_0 and a final one S_F, to apply all the rules of the operational semantics of **ROSA** to S_0 so generating a set of processes, and only follow on, with the state of this set that maximizes $p - f$ (associated to S_F). Therefore, the computational cost of the LTS, is moved from exponential to polynomial, so making it cheaper.

5 Conclusions and Future Work

In this paper we have provided the set of **ROSA** processes with a metric structure which allows to define a *promising function* $p - f$ for the sake of (computationally) improving the searching for 'a goal node' by means of this heuristic. This *promising function* establishes the first step towards a Genetic Process Algebra definition, since a slight variation of it, could be a $fitness$ function.

Our future work in this line is also concerned with the translation of the former operational semantics rules of **ROSA** to those rules which capture the same behaviour but over the domain of Ordered Normal Form processes.

References

1. Bloom, B.: Ready Simulation, Bisimulation, and the Semantics of CCS-like Languages. PhD thesis, Department of Electrical Engineering and Computer Science, MIT (1989)
2. Cazorla, D., Cuartero, F., Valero, V., Pelayo, F.L., Pardo, J.J.: Algebriac Theory of Probabilistic and Nondeterministic Processes. Journal of Logic and Algebraic Programming 55(1-2), 57–103 (2003)
3. Cleaveland, R., Hennessy, M.: Testing equivalence as a bisimulation equivalence. Formal Aspects of Computing 5, 1–20 (1993)
4. Goldberg, D.E.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading (1989)
5. Holland, J.H.: Adaptation in natural and artificial systems. The University of Michigan Press, Ann Arbor (1975)
6. Ortega-Mallén, Y., de Frutos-Escrig, D.: A complete proof system for timed observations. In: TAPSOFT 1991, vol. 1, pp. 412–440 (1991)
7. Pelayo, F.L., Cuartero, F., Ossa, L., Pelayo, M.L., Guirao, J.L.G.: Towards the evolutionary process algebra. In: 8th IEEE Int. Conf. on Cognitive Informatics, pp. 69–76. IEEE Computer Society Press, Los Alamitos (2009)
8. Pelayo, F.L., Cuartero, F., Valero, V., Cazorla, D.: Analysis of the MPEG-2 encoding algorithm with ROSA. Electronic Notes on Theoretical Computer Science 80(1), 185–202 (2003), `http://www.elsevier.nl/locate/entcs/volume80.html`
9. Pelayo, F.L., Pelayo, M.L., Guirao, J.G.: Generating the syntactic and semantics graphs for a markovian process algebra. Journal of Computational and Applied Mathematics 204, 38–47 (2007)
10. Godefroid, P., Khurshid, S.: Exploring very large state spaces using genetic algorithms. In: Procedings of the Conference on Tools and Algorithms for Construction and Analysis of Systems, Grenoble, France, pp. 266–280 (2002)
11. Rodriguez-Lopez, J., Romaguera, S., Valero, O.: Denotational semantics for programming languages, balanced quasi-metrics and fixed points. International Journal of Computer Mathematics 85(3), 623–630 (2008)