A Subset of Lotos with the Computational Power of Place/Transition-Nets

Michel Barbeau, Gregor v. Bochmann

Abstract

In this paper, we define a subset of Lotos that can be modelled by finite Place/Transition-nets (P/T-nets). That means that specifications in that Lotos subset can be translated into finite P/T-nets and validated using P/T-net verification techniques. An important aspect of our work is that we show that conversely P/T-nets can be simulated in our Lotos subset. It means that the constraints we put on Lotos in order to obtain finite nets are minimally restrictive. We may also conclude that our Lotos subset and P/T-nets have equivalent computational power. To the best of our knowledge, no such bidirectional translation scheme has been published before.

Topics:

Relationships between net theory and other approaches.

1. Introduction

In this paper, we define a subset of Basic Lotos [Bolo 87, ISO 88] that can be modelled by finite Place/Transition-nets (P/T-nets). That means that specifications in that Lotos subset can be represented and translated into finite P/T-nets and validated using P/T-net verification techniques. An important aspect of our work is that we show that conversely P/T-nets can be simulated in our Lotos subset. It means that the constraints we put on Lotos in order to obtain finite nets are minimally restrictive. We may also conclude that our Lotos subset and P/T-nets have equivalent computational power. To the best of our knowledge, no such bidirectional translation scheme has been published before.

The problem of modelling process-oriented languages, and more specifically CCS and CSP like languages, by Petri nets has been tackled by several authors. Cindio et al. [Cind 83], Degano et al. [Dega 88], Glabbeek [Glab 87], Goltz [Golt 84a, 84b, 88], Nielsen [Niel 86], Olderog [Olde 91] and Taubner [Taub 89] considered CCS or CSP, or both. Lotos has been worked by Marchena and Leon [Marc 89], and Garavel and Sifakis [Gara 90]. The approaches may be...
categorized based on the following criteria: i) style of definition, ii) finiteness of the representation, and iii) distinction of concurrency and nondeterminism.

One of two definition styles may be adopted, namely denotational or operational. A denotational style is used in: [Cind 83], [Gara 90], [Glab 87], [Golt 84a, 84b, 88], [Niel 86], [Marc 89] and [Taub 89], whereas an operational style, à la Plotkin, is used in: [Dega 88], [Olde 91] and in the present paper. In opposition to the operational approach, the denotational style is constructive. It means that the definition yields directly to a procedure for translating terms of the process-oriented language to Petri nets. However, we shown in [Barb 91a, b] that thanks to our operational definition an important P/T-net verification method can be adapted to Lotos without even translating the latter to the former.

Another important matter is whether or not the Petri net representation of the process-oriented language is finite. It is well known that an unbounded number of Petri net places and transitions is required to represent a process-oriented language when recursion is combined with parallel composition, sequential composition, hiding and disabling operators. This difficulty means that it is impossible to transfer to the process-oriented language several important verification techniques elaborated for Petri nets, since they require finite nets. Note that in our mind, finite nets does not mean finite state systems. Finite representations can be obtained by restricting the process-oriented language or using high-level Petri net models. Finite representations for subsets of CCS are proposed in [Golt 88], using P/T-nets, and in [Taub 89], using Predicate/Transition-nets, which is a high-level model. Finite extended Petri nets are generated from Lotos, with the finite control property, in [Gara 90], this work is also interesting because the data part of Lotos is also handled. In this paper we define a subset of Basic Lotos, with syntactical constraints, that can be modelled by finite P/T-nets.

Non distinction of concurrency and nondeterminism means that Lotos expressions such as a; stop]||b; stop and a; b; stop]||b; a; stop have the same semantic interpretation. Distinction of concurrency and nondeterminism allows accurate representation of behaviors by partial orders. It is a representation that shows just natural dependencies between actions. Multi-sets of actions are possible in a single transition. This has an impact on treatment of fairness problems [Reis 84]. Our Place/Transition-net semantics is less attractive, than definitions described in Refs. [Dega 88], [Golt 88], [Niel 86] and [Olde 91], with respect to distinction of concurrency and nondeterminism.

An important feature in our approach is that we show that P/T-nets can be simulated in our Lotos subset. Other authors have proposed simulations of Petri nets in languages such as Prolog, Azema et al. [Azem 84], or Meije, Boudol et al. [Boud 85]. These simulations are not in languages that have been shown translatable into finite Peo nets. The goal of Azema et al. is to use Prolog as a simulation tool for Petri nets whereas the aim of Boudol et al. is to provide a textual representation for Petri nets. Translation into Lotos of another graphical representation for behaviors, called Process-Gate Network, is described [Bolo 90].

In Section 2, we introduce the P/T-net model. Our Basic Lotos subset that can be translated into finite P/T-nets is called PLotos and is defined in Section