Covering Step Graph Preserving Failure Semantics

François Vernadat, François Michel
e-mail : {vernadat,fmichel}@laas.fr

LAAS-CNRS
7 avenue du Colonel Roche F-31077 Toulouse cedex - France

Abstract: Within the framework of concurrent systems, several verification approaches require as a preliminary step the complete derivation of the state space. Partial-order methods are efficient for reducing the state explosion due to the modeling of parallelism by interleaving.

In the case of persistent or sleep sets, only a subset of enable transitions is examined, the derived graph is then a subgraph of the whole graph. The resulting sub-graph may be used for verifying absence of deadlock or more specific properties.

The covering step graph (CSG) approach visits all the transitions, but some independent events are put together to build a single transition step, the firing of this transition step is then atomic.

In a CSG, steps of independent transitions are substituted as much as possible to the subgraph which would result from the firing of the independent transitions. The potential benefit of such a substitution may be exponential with respect to the number of “merged” independent transitions.

This paper investigates the on-the-fly derivation of covering step graphs preserving failure semantics. Testing Equivalence and CSP semantics are considered.

Keywords: concurrent systems, state space exploration, partial-order, failure semantics, verification methods.

1 Introduction

The state space derivation represents the preliminary step of several verification methods for concurrent systems. This approach is made attractive by the existence of efficient and automatic verification techniques, such as bisimulation and model-checking. The combinational explosion is the main limitation of this approach.

Many studies are currently in progress for reducing this problem: on-the-fly bisimulation [FM 90], symbolic marking graph derivation through symmetrical folding [Jen 87], so-called partial order techniques which attempt to avoid the combinational explosion resulting from the concurrency interpretation by means of interleaving: persistent sets [Val 89], sleep sets [GW 91]. An other direction, followed in [Esp 93, McMil 95], avoids the state explosion by using model checker directly on the system description (unfolding of Petri Nets).

The partial order techniques (see [WG 93] for a general survey) represent the framework of the approach developed in this paper. The basic principle of these approaches consists in considering a single specific path among all the sequences
which possess the same Mazurkiewicz's trace [Maz 87]. The aim is to obtain a sub-graph of the initial one where transition interleaving will be as reduced as possible. The set of firable transitions from each reached state is limited by means of “sleep set” or “persistent set” (both methods can be combined). Figure 1 depicts the results of these methods upon the graph describing the interleaving of ab and cd.

The approach used here, visits all the transitions, but some independent events are put together to build a single transition step, the firing of this transition step is then atomic. The resulting graph is referred as covering step graph (CSG). The potential benefit of such a substitution may be exponential with respect to the number of “merged” independent transitions. In the case of Milner's scheduler [Mil 85], the gain is exponential: for n sites, the standard LTS consists of $n \times 2^n$ states and $(n^2 + n) \times 2^{n-1}$ transitions, while the CSG consists of $n + 1$ states and $n + 1$ transitions.

The CSG analysis allows the verification of global reachability properties, such as deadlock, but also more elaborated behavioral properties such as observational equivalence.

This paper follows the work initiated in [VAM 96]. The framework of failure semantics is now considered. First, a definition of such covering structures preserving failure semantics (testing equivalence [Bri 88] and CSP semantics [OH 86]) is proposed. In a second step, the problem of on-the-fly derivation is investigated. A general algorithm and associated sufficient conditions are presented, for such derivation. A specific instantiation of the general algorithm is then proposed and a first evaluation is made.

Section 2 presents the main concepts of failure semantics. General covering step graphs are presented in section 3. Section 4 adapts the general definition of CSG to failure semantics. Section 5 shows how to modify a standard enumeration algorithm in order to obtain a covering step graph preserving failure semantics. A preliminary assessment of the proposed approach is developed in section 6.