Project 22703: SYRF

Deliverable 3.1.3

Symbolic abstraction of automata

Third Year

We contribute this year work performed by two distinct sites (INRIA Meije, and INRIA Ep-Atr). Work at INRIA Meije focused on *Compositional Reduction of Boolean Equation Systems*. It was in fact the major contribution of this site to the whole project this year. Work at INRIA Ep-Atr focused both on automata abstractions using Implicit Labelled Transition Systems and synbolic techniques, and on controller synthesis and the Supervisory Control problem, again solved with symbolic techniques. We describe all these activities conducted this year in the sequel.

1 INRIA Meije contribution

Approach

Large program development is handled everywhere by modular division. In the case of Esterel this module structure is flattened prior to compilation by *expanding* the submodule instances into their actual bodies. On the other hand due to the nature of the language the compilation process induces complex translations and interpretations, into a software representation form of *synchronous* circuits at *gate level*, or even further into explicit *Mealy FSM* descriptions. Important steps in the compilation process are *constructive causality* analysis, logical and sequential *optimization*, and sometimes *symbolic reachable state space construction*. The algorithms at work can be very expensive when applied globally, an extreme example being those which include symbolic RSS construction.

Modular compilation is therefore a highly desirable goal. Still, formal semantics must be preserved to the point of allowing processed submodules to be linked back instead of their former counterpart sinto larger modules. The purpose is to relax complexity bottlenecks, as induced for instance by global netlist optimization or symbolic reachable state space computation. The main difficulty is that submodule transformations should be semantically sound and preserve whatever is needed to link submodules back into a larger program. In our case this means clear separation between initial instant and other behaviours, so that optimization of the (code of) later instants has to be done in the context of whatever states the initial instant may lead to. This information in fact proved utterly important to reach some significant amount of reduction, and make the modular optimization efficient *both* in intermediate size and in time complexity.

Results achieved this year

We developped translators from SC format (the proprietary representation format in Esterel for netlists descriptions as Boolean Equations Systems) to Blif format (an entry point to powerful hardware logical synthesis tools), and back. The translators must be carefully designed to allow proper split between *surface* (as we shall call the part of the equation system dealing with behaviour at the initial instant) and *depth* (the code for other instants), and later reassembly, respecting the formal semantics and associating properly the external action calls.

We studied independent logical optimization of both surface and depth. It very soon appeared that optimization of registers in the depth part (the only part which may contain them) was rather disappointing, specially if compared with optimisation of submodules in isolation, without preserving the information meant for reconstruction and immersion into larger contexts. After inspection it became clear that the surface was in fact providing much valuable information on the possible configurations of state registers after initialisation, which helped greatly reduce the search space and equate more system variables. Thus we designed specific techniques to handle this surface information as contraint ("care set" in hardware logical synthesis jargon) to help in the reduction of the depth part of the Boolean Equation System.

Third, we implemented a SC linker which, provided a number of optimized subcomponents in SC format and a surrounding context, builds back a single SC description, possibly duplicating some of the subcomponents' surface as demanded by the preemption mechanisms.

The verification method is described in [2], together with a set of experiments on compositional optimizations.

2 INRIA Ep-Atr contributions

2.1 Abstractions

Approach

Last year, we presented *Implicit Labeled Transition Systems* (ILTS) as intermediate models for discrete event systems. We have studied operations of parallel composition and event hiding, as well as an equivalence criterion based on strong bisimulation semantics, namely the equivalence called *symbolic bisimulation.* We also proposed an algorithm that provides the reduced system according to a bisimulation equivalence relation. The aim of the work is to rely on intensional descriptions of the systems for symbolic abstractions purposes, based on behavioral equivalences. The intensional approach we propose has the main advantage to remain at an interesting level of abstraction to handle sets (of states, of events) in which algorithms and properties can entirely be expressed, thus providing us with a high level of expressiveness. This abstract level avoids us to bother with a particular choice of sets implementation, such as Binary Decision Diagrams (BDDs), even if all operations are actually based on this representation for sets. Moreover, the intensional formalism is completely compatible with the symbolic techniques, since intensionally described sets can afterwards be implemented by any kind of methods, e.g. decision diagrams. However, other tools with various implementations for sets might be considered using the same high level framework (e.g. Mathematica, Maple).

Achieved Results

This year, we achieved the design of abstraction methods based on symbolic techniques: classical **abstraction by state fusion** has been considered (with associated equivalence relation computation). We also introduced other kinds of abstraction, falling into the category of **abstraction by restriction**: in particular, we studied the use of the controller synthesis methodology to achieve the restriction synthesis [9].

Abstraction by state fusion The principles of abstraction by state fusion is classical : super states are obtained by fusionning concrete states and transitions are naturally lifted to super states. For this kind of abstraction, compositionality is trivial. In this setting, we propose a family of algorithms to build the abstracted system, in the symbolic/intensional philosophy. More precisely, we present a general method to abstract automata on the basis of a *state fusion criterion*, derived from e.g. equivalence relations (such as bisimulation), partitions, ...

The approach has the advantage to deliver the abstracted model still in an intensional style, whereas in classical approaches symbolic models might become explicit when submitted to reduction. However, the state fusion computation might still have some explicit features when derived from equivalence relations : we cannot yet avoid the equivalence classes enumeration, during which the reduced model is built "on the fly". Our choice of equivalence relations mainly focuses on the computation of greatest bisimulations (strong, but also weak/delay/branching), to ensure behavioral properties preservation. However, any kind of relations *larger* than the bisimulations can be considered (in particular, partitions, general relations,..., provided the user takes care of the correction of its abstraction).

Abstraction by restriction This abstraction aims to simplify the model by disallowing some behaviors. A naive approach would consist in modifying the structure of the automata by removing either a set of states or a set of events. The corresponding set of states could possibly be induced by a property (e.g. invariant property,...). Also, observing automata can be composed with the original one in order to get more subtle restrictions of the behaviors. Both techniques have been considered, and for the latter one, we have explored particular kind of restrictions : as in the control theory world , the status of events is twofold. Some events, called *uncontrollable*, cannot be prevented from occurring (e.g. think of an open system in an environment), whereas the others, though *controllable* can be disabled. In this new framework, the abstraction by restriction needs being refined, and falls into the controller synthesis issues by adding constraints to the initial system (by means of new boolean/polynomial equations), called the *controller*.

The prototyped functionalities enabling the above mentioned approaches are available in the Sigali tool box, the formal calculus system of the Signal environment [9], [6]. These new functionalities of Sigali are still under improvement process but have already been experimented for the case of significant Signal programs. In particular, some experiments have been conducted on EdF partner case study : the *power transformer station controller*.

Unfortunately the Syrf Steam Boiler case, although designed in Signal and studied in our group in the Task 3.4, turns out to be far too much numerical to be relevant for boolean automata issues.

2.2 Deviations : "Supervisory Control Problem"

Approach

As explained in the previous section, Supervisory Control Problem turn s out to be very closely related to abstraction issues ; as pointed out by the Syrf reviewers : "...controller can be seen as a natural and important extension...". Therefore, we decided to keep investigating significantly this theory, e.g. by merging symbolic bisimulation techniques and controller synthesis issues.

Control theory of discrete event systems allows to use constructive methods, that ensure, *a priori*, required properties of the system behavior. Therefore, the validation phase reduces to properties not guaranteed by the control.

Achieved Results

There exist different theories for the control of discrete event systems since the 80's [10], [3]. Usually, the starting point of these theories is: given a model for the system and the control objectives, a controller must be derived by various means such that the resulting behavior of the closed loop system meets the control objectives.

- This year, we have been interested in the controller synthesis problem within the *Ramadge and Wohnam Approach*. Symbolic bisimulation techniques (inherited from original Task 3.1.3. achievements) have been considered for solving the Supervisory Control Problem, in the same spirit as [1]. They solve the Supervisory Control Problem by exploiting the link between a given bisimulation and the controllability of languages. We improved this work by providing symbolic version of the algorithms. See [7], [9] for more details ([7] is a submission version of Section 5 in [9]).

- Meanwhile, many efforts for dissemination of the *Supervisory Control Problem using polynomial methods*, in the Signal Environment have been done [8],[6],[4],[5]. For a theoretical point of view, see [4] and its corresponding paper [6]. For an applicative point of view, see [8],[5]. In [8], we relate the experience of the controller synthesis methodology applied to the incremental specification of a power transformer station Controller (See Task 1 for more details).

References

[1] G. Barret and S. Lafortune. Bisimulation, the supervisory control problem and strong model matching for finite state machines. *Journal of Discrete Event Dynamic Systems*, 8(4):337–429, December 1998.

[2] A. Bouali, L. Henry-Greard, R. de Simone. Compositional Optimization and Analysis of Esterel programs. internal Draft report.

[3] L.E. Holloway, B.H. Krogh, and A. Giua. A survey of Petri net methods for controlled discrete event systems. *Discrete Event Dynamic Systems: Theory and Application*, 7:151–190, 1997.

[4] H. Marchand and M. Le Borgne. Supervisory control problem within the polynomial dynamical system framework. *Special Issue of the IEEE Transactions on Systems, Man and Cybernetics: Discrete Systems and Control.*, 2000. Submitted.

[5] H. Marchand, P. Bournai, M. Le Borgne, and P. Le Guernic. Incremental design od automatic discrete-event controllers in the signal environment. *Discrete Event Dynamical Systems.*, 1999. Submitted.

[6] H. Marchand and M. Le Borgne. The supervisory control problem of discrete event systems using polynomial methods. Technical Report 1271, Irisa, October 1999.

[7] H. Marchand and Pinchinat S. Supervisory control problem using symbolic techniques. In *2000 American Control Conference.*, 1999. Submitted.

[8] H. Marchand and M. Samaan. On the incremental design of a power transformer station controller using controller synthesis methodology. In *World Congress on Formal Methods (FM'99), Volume 1709 of LNCS*, pages 1605–1624, Toulouse, France, October 1999.

[9] S. Pinchinat, H. Marchand, and M. Le Borgne. Symbolic abstractions of automata and their application to the supervisory control problem research. Technical Report 1279, Irisa, November 1999.

[10] P. J. Ramadge and W. M. Wonham. The control of discrete event systems. *Proceedings of the IEEE; Special issue on Dynamics of Discrete Event Systems*, 77(1):81–98, 1989.