DISCRETE EVENT SYSTEMS: ANALYSIS AND CONTROL. PROC OF WODES2000, PP 39-48, GHEN BELGIUM, AUGUST 2000.

SYMBOLIC ABSTRACTIONS OF AUTOMATA

S. Pinchinat

H. Marchand

IRISA, INRIA Rennes, Campus univ. de Beaulieu F-35042 RENNES, France

pinchina@irisa.fr, hmarchan@irisa.fr

Keywords: Intensional transition systems, polynomials, symbolic bisimulations, model reduction.

Abstract We describe the design of abstraction methods based on symbolic techniques: classical **abstraction by state fusion** has been considered. 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, ... We also introduce other kinds of abstraction, falling into the category of **abstraction by restriction**: in particular, we study the use of the controller synthesis methodology to achieve the restriction synthesis.

1. INTRODUCTION

Although many algorithms offer a wide range of techniques to analyze behavioral properties of systems, the state explosion phenomenon has caused their UN-usability of real systems. This observation has led to many kinds of proposals, approaches called modular (Clarke et al., 1989; Larsen, 1989; Clarke et al., 1994; Clarke and Kurshan, 1990), symbolic (Burch et al., ; McMillan, 1993) ... on the one hand, and techniques based on partial order reductions (Godefroid, 1990; Peled, 1994; Clarke and Kurshan, 1990) or abstractions (Bensalem et al., 1998; Cousot and Cousot, 2000) on the other hand. However, in many practical cases, these methods are not sufficient enough to perform effective verification. In this paper, we show that these techniques are not incompatible: we consider a wide family of abstraction methods (those based on state fusion) but in the symbolic philosophy, thus taking advantage of both methodologies : given a symbolic system description and a state fusion criterion (eg. an equivalence relation over states) also defined symbolically, our algorithm delivers a symbolic description of the reduced system. Moreover, we extend the approach for another kind of abstraction relying on behavioral restriction for which symbolic controller synthesis applies in a natural way.

2. INTENSIONAL LABELED TRANSITION SYSTEMS

Mathematical Framework: in the following, we write $\mathbb{Z}/_{p\mathbb{Z}}$ for the finite field $\{0, 1, \ldots, p-1\}$, with p prime. Let Z be a finite set of k distinct variables $Z_1, ..., Z_k$. We denote by $\mathbb{Z}/_{p\mathbb{Z}}[Z]$ the set of polynomials over variables $Z_1, ..., Z_k$ which coefficients range over $\mathbb{Z}/_{p\mathbb{Z}}$. We recall that $(\mathbb{Z}/_{p\mathbb{Z}}[Z], +, *)$ is a ring. Given a polynomial $P(Z) \in \mathbb{Z}/_{p\mathbb{Z}}[Z]$, we associate its set of solutions $Sol(P) \subseteq (\mathbb{Z}/_{p\mathbb{Z}})^k$:

$$Sol(P) \stackrel{\text{def}}{=} \{(z_1, ..., z_k) \in (\mathbb{Z}/_{p\mathbb{Z}})^k | P(z_1, ..., z_k) = 0\} \tag{1}$$

It is worthwhile noting that in $\mathbb{Z}/_{p\mathbb{Z}}[Z]$, $Z_1^p - Z_1, ..., Z_k^p - Z_k$ evaluate to zero. Then for any $P(Z) \in \mathbb{Z}/_{p\mathbb{Z}}[Z]$, one for instance has $Sol(P) = Sol(P + (Z_i^p - Z_i))$. We write $P_1 \equiv P_2$ whenever $Sol(P_1) = Sol(P_2)$. We then introduce the quotient ring of polynomial functions $A[Z] = \mathbb{Z}/_{p\mathbb{Z}}[Z]/_{<Z^p - Z>}$, where all polynomials $Z_i^p - Z_i$ are identified to zero, written for short $Z^p - Z$. $A[Z]$ can be regarded as the set of polynomial functions with coefficients in $\mathbb{Z}/_{p\mathbb{Z}}$ for which the degree in each variable is lower than $(p-1)$. It is also possible to define a representative of $Sol(P)$ (i.e. $[P]_{\equiv}$), called the *canonical generator*. We now mention some useful properties to manipulate polynomials, namely :

Property 1 *For all polynomials $P_1, P_2, P \in \mathbb{Z}/_{p\mathbb{Z}}[Z]$, $Sol(P_1) \subseteq Sol(P_2)$ whenever $(1 - P_1^{p-1}) * P_2 \equiv 0$. Moreover, by defining $P_1 \oplus P_2 \stackrel{def}{=} (P_1^{p-1} + P_2^{p-1})^{p-1}$, we have, $Sol(P1) \cap Sol(P_2) = Sol(P_1 \oplus P_2)$, $Sol(P_1) \cup Sol(P_2) = Sol(P_1 * P_2)$, and $(\mathbb{Z}/_{p\mathbb{Z}})^k \setminus Sol(P) = Sol(1 - P^{p-1})$.*

In the following, we shall use $P_1 \Rightarrow P_2$ to denote the set $\{z \in \mathbb{Z}/_{p\mathbb{Z}}{}^k | P_1(z) = 0 \Rightarrow P_2(z) = 0\}$. It is equal to $(1 - P_1^{p-1}) * P_2$.

Finally, we introduce the *existential/universal abstractions* (or quantifications) over polynomials w.r.t. some variables. Let $P \in \mathbb{Z}/_{p\mathbb{Z}}[Z]$, we shall write $\exists Z_i P$ for the polynomial $P|_{Z_i=0} * P|_{Z_i=1} * \ldots * P|_{Z_i=p-1}$, where $P|_{Z_i=v}$ is P obtained by instantiating any occurrence of variable Z_i by value v. Similarly, we define a dual variable abstraction over polynomials, based on universal quantification : $\forall Z_i P$ is computed as $P|_{Z_i=0} \oplus P|_{Z_i=1} \oplus \ldots \oplus P|_{Z_i=p-1}$ whose solutions are elements of the form $(z_1, ..., z_{i-1}, z_{i+1}, ..., z_k)$ s.t. $\forall z_i, (z_1, ..., z_{i-1}, z_i, z_{i+1}, ..., z_k) \in Sol(P)$.

Polynomial Implementation: it turns out that the best known implementation (for memory and computation performance) of polynomials over $\mathbb{Z}/_{p\mathbb{Z}}$ is based on their decomposition according to the Lagrange polynomials, leading to p-ary Decision Diagrams data structures (p-DD). A classical instance of this approach is the well-known Shannon decomposition for the case $p = 2$, with associated Binary Decision Diagrams (BDD) for boolean functions (Bryant, 1986). In our framework, we fall under the case $p = 3$. Polynomials are then encoded by Ternary Decision Diagrams (3-DD) (Marchand and Le Borgne, 1999) which are the actual implementation of polynomial in our formal calculus system SIGALI (Pinchinat et al., 1999).

Intensional transition system model (Kouchnarenko and Pinchinat, 1998): An (n, m)*-dimensional Intensional Labeled Transition System (or* (n, m)*-ILTS)* is a structure $S = (X, X', Y, I, \mathcal{T})$ where $X = \{X_1, ..., X_n\}$ and $X' = \{X'_1, ..., X'_n\}$ are two sets of *(source and target) states variables*, $Y = \{Y_1, ..., Y_m\}$ is a set of labels variables, $I \in \mathbb{Z}/_{p\mathbb{Z}}[X]$ characterizes initial states and $\mathcal{T}(X, Y, X') \in \mathbb{Z}/_{p\mathbb{Z}}[X, Y, X']$ describes the legal transitions. Given some source state $x \in (\mathbb{Z}/_{p\mathbb{Z}})^n$ and some target state $x' \in (\mathbb{Z}/_{p\mathbb{Z}})^n$, the set $Sol(\mathcal{T}(x, Y, x'))$ denotes all the possible labels of transitions from x to x'. We shall call $Ext(T)$ the corresponding "extensional" Labeled Transition System (LTS).

3. ABSTRACTION BY STATE FUSION

This section is devoted to the computation of a reduced system according to a fusion state criterion. First, we assume that the criterion is given by some symbolic canonical surjection. In this case, reducing the system is straightforward. Next, we explore other means such as equivalence relation between states, partitions, ...

3.1. SYSTEM REDUCTION W.R.T. A FUSION CRITERION

Given a symbolic canonical surjection (see Definition 1), we explain how one can define the associated quotient ILTS.

Definition 1 *Assume given an* (n, m)*-ILTS* $S = (X, X', Y, I, \mathcal{T})$ *where* $X = \{X_1, ..., X_n\}$. *We say that* ϕ *is a symbolic fusion criterion over* S *w.r.t. a set of "l" fresh variables* $Z = \{Z_1, ..., Z_l\}$ *whenever (1)* $l \leq n$, *(2)* $\phi \in \mathbb{Z}/_{p\mathbb{Z}}[X, Z]$, *and (3) for all* $x \in (\mathbb{Z}/_{p\mathbb{Z}})^n$, $\phi(x, Z) = 0$ *has a unique solution.* •

By Definition 1, $Sol(\phi) \subseteq (\mathbb{Z}/_{p\mathbb{Z}^n}) \times (\mathbb{Z}/_{p\mathbb{Z}^l})$, and because of (3), $Sol(\phi)$ defines a surjective mapping from $(\mathbb{Z}/_{p\mathbb{Z}})^n$ to $(\mathbb{Z}/_{p\mathbb{Z}})^l$. Now, we define a transition system whose states are obtained by gluing those x's mapped onto the same element of $(\mathbb{Z}/_{p\mathbb{Z}})^l$. The reduction of S w.r.t. ϕ is $S_\phi \stackrel{\text{def}}{=}$

$(Z, Z', Y, \mathcal{T}_\phi)$, defined by

$$\mathcal{T}_\phi(Z, Y, Z') = \exists X \exists X' (\phi(X, Z) \oplus \mathcal{T}(X, Y, X') \oplus \phi(X', Z')) \qquad (2)$$

Informally, $\mathcal{T}_\phi(z, y, z') = 0$ whenever there exists one x in the class encoded by its representative z and one x' in the class encoded by its representative z' such that $x \xrightarrow{y} x'$ holds in the original system S. Note that when initial states are taken into account, defined by some polynomial $I(X)$, the corresponding initial predicate $I'(Z)$ in the reduced model is obtained by $I'(Z) = \exists X\{\phi(X, Z) \oplus I(X)\}$.

In general, S_ϕ has more behaviors than S, unless the state fusion criterion ϕ is derived from a bisimulation equivalence, in which case behavioral properties are faithfully preserved by the reduction.

3.2. REDUCTION MODULO AN EQUIVALENCE RELATION

Assume given an (n, m)-ILTS $S = (X, X', Y, I, \mathcal{T})$ and an equivalence relation ρ over the states of $Ext(S)$, which is symbolically represented by some $R \in \mathbb{Z}/_{p\mathbb{Z}}[X, X_d]$. Here $X_d = \{X_{d_1}, ..., X_{d_n}\}$ is a copy of X.

We explain how to construct a corresponding state fusion criterion to apply previous section techniques. Assuming the number of R-classes is k, and $p^{r-1} < k \leq p^r$ (for some $r \geq 1$), we show how to compute a p-DD, say Φ, over variables $Z = \{Z_1, ..., Z_r\}$ and $X = \{X_1, ..., X_n\}$ denoting the state fusion criterion ϕ associated to R. To do so, we directly work on the data structures, namely the p-Decision Diagrams (p-DD). Intuitively, we start from the p-DD of $R(X, X_d) \subseteq (\mathbb{Z}/_{p\mathbb{Z}})^n \times (\mathbb{Z}/_{p\mathbb{Z}})^n$, with the particular reorder of the variables $X_i \prec X_{d_j}$, $\forall i, j$. Call θ this p-DD.

Property 2 *At the end of every path x (over variables X's) in θ, the remaining sub-p-DD in variables X_d'denotes the R-class of x.* $\diamond$

Therefore, a traversal of all paths x in θ leads us to compute "on the fly" the number of R-classes, namely k ; also, during this enumerative phase (in the worse case we explore the whole state space), we incrementally achieve the computation of Φ by introducing one by one the r variables Z_i when necessary. The idea of the algorithm (Pinchinat et al., 1999) is the following: from the root of θ (variable X or the leaf 0 if R is trivial), we recursively go down along a path until a variable X_{d_j} is reached. Call θ' the remaining sub-p-DD below X_{d_j} in θ. By Property 2, θ' is an R-class. Provided we know this R-class has not been encountered yet, we attach θ' to the structure Φ as follows: either an available hanging branch in Φ is available. In this case θ' is plugged at this available place. Otherwise, we introduce a fresh variable Z_i at the top of Φ and wait for the complete p-DD over variables $\{Z_1, ..., Z_{i-1}\}$ containing θ' to be achieved, then plug it as a second son of Z_i.

The reader can refer to Figure 3.2, for the case $p = 3$. In this example, already built 3-DDs are drawn as triangles with solid bold lines, whereas ones under construction are drawn in dashed bold lines. Assume we

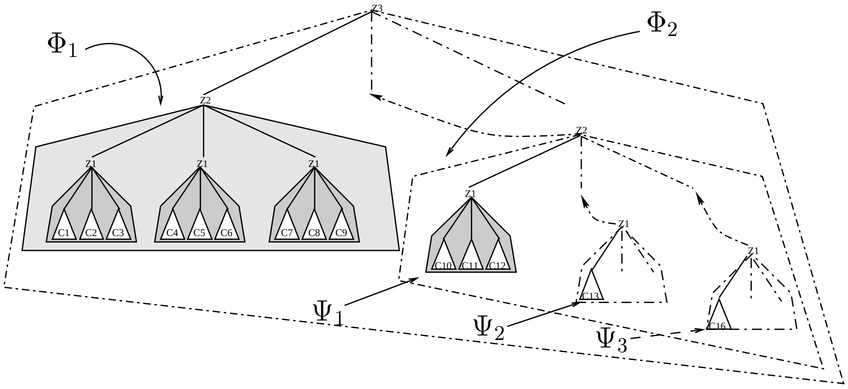

Figure 1 Construction of $\phi(X, Z)$

already have completed a 3-DD containing already treated classes, say from C_1 up-to C_9 ; call Φ_1 this structure. Suppose now that a new class C_{10} is encountered. Then a new variable Z_3 needs being introduced at the top Φ_1 and awaits for other sub-structures to be complete before plugging them underneath. For example, after structures Ψ_1 (containing C_{10} but also C_{11} and C_{12}), Ψ_2 and Ψ_3 have been achieved , Φ_2 can be completed and then plugged under Z_3. The final stage of the algorithm is to run a completion for the remaining non allocated branches (which exist when the number of classes k is not of the form p^r) by attaching them to the leaf **1**. In our example, suppose there is only 16 R-classes, i.e. the last class is C_{16}, then the remaining hanging branches of structure Ψ_3 will point to leaf **1**.

3.3. PARTICULAR CASES OF EQUIVALENCES: BISIMULATIONS

Bisimulation relations (Milner, 1989; Park, 1981; Van Glabbeek, 1993) have been shown to capture a nice notion of "the same behavior" : a bisimulation is an equivalence relation between states of a *labeled transition system* (LTS) which therefore enables to perform a particular state fusion abstraction. The behavioral properties of the abstract system coincide with the original one. We first recall that the classical strong bisimulation can be handled symbolically.

Definition 2 *(Park, 1981; Milner, 1989) Given two LTSs* $t_1 = (Q_1, \Sigma, \mathcal{I}_1, \rightarrow_1)$ *and* $t_2 = (Q_2, \Sigma, \mathcal{I}_2, \rightarrow_2)$, *a* **strong bisimulation** *between* t_1 *and* t_2 *is a binary relation* $\rho \subseteq Q_1 \times Q_2$ *s.t.* $(q_1, q_2) \in \rho$ *when-*

ever for all $\sigma \in \Sigma$, for all transition $q_1 \xrightarrow{\sigma}_1 q'_1$ there exists a state q'_2 s.t. $q_2 \xrightarrow{\sigma}_2 q'_2$ and $(q'_1, q'_2) \in \rho$. And vice-versa •

Since bisimulations are closed under arbitrary unions, there exists a greatest bisimulation between t_1 and t_2, written $\asymp$ in the following. Assume given two ILTSs $S_U = (U, U', Y, I_U, \mathcal{T}_U)$ and $S_V = (V, V', Y, I_V, \mathcal{T}_V)$. Algorithm 1 gives a symbolic computation of the greatest bisimulation between S_U and S_V.

Algorithm 1:

1 Define the polynomial $\mathcal{R}_0(U, V) \stackrel{\text{def}}{=} 0$.

2 Compute until stabilization $(\mathcal{R}_j(U, V))_j$ defined by:

$\mathcal{R}_{j+1}(U, V)$ is the canonical generator of the $\equiv$-class of

$$\begin{cases} \mathcal{R}_j(U,V) \oplus \ \forall U' \forall Y [(\mathcal{T}_U(U,Y,U') \Rightarrow \exists V' (\mathcal{T}_V(V,Y,V') \oplus \mathcal{R}_j(U',V'))] \\ \qquad\qquad \oplus \ \forall V' \forall Y [\mathcal{T}_V(V,Y,V') \Rightarrow \exists U' (\mathcal{T}_U(U,Y,U') \oplus \mathcal{R}_j(U',V'))] \end{cases}$$

3 Call $\mathcal{R}(U, V)$ the result.

Theorem 1 *Algorithm 1 terminates and at the end, $R(u, v) = 0$ if and only if $u \asymp v$ in the extensional worlds $Ext(S_U)$ and $Ext(S_V)$.* ◇

Any other variant of bisimulation (eg. weak/delay/branching) can be considered likewise (see (Pinchinat et al., 1999) for more details).

3.4. OTHER KINDS OF STATE FUSION CRITERION

In this section, we explore other means to express the state fusion criterion. For instance, the fusion criterion can be specified by a state partition, or characterized by a set of logical propositions attached to states. In such cases, the abstract model computation can be simplified: let $\{P_1, ...P_k\}$ be a set of polynomials in $A[X_1, ...X_n]$ such that: $Sol(P_i) \cap Sol(P_j) = \emptyset$ and $\bigcup_{i \in \{1..k\}} Sol(P_i) = (\mathbb{Z}/_{p\mathbb{Z}})^n$. This set of polynomials is a symbolic representation of a state partition. Compared to Section 3.2, we somehow already have done most of the work since each polynomial denotes an equivalence class, as the C_i's are handled Section 3.2. Also because the number k of classes is known in advance, the construction of P-DD ϕ is immediate.

From a practical point of view, the state partition is often derived from a set of logical propositions over states (eg. state variable values). Let $\Pi = \{\Pi_1, ...\Pi_l\}$ denote propositions over states : $\Pi_i(x) = 0$ iff. Π_i is *true* in state x. From Π, a partition can be derived so that techniques above apply: for each Set I and J with $I \cup J = [1..l]$, $I \cap J = \emptyset$, we define a polynomial $\alpha_{IJ}(X) = \bigoplus_{i \in I} \Pi_i(X) \oplus \bigoplus_{j \in J} (1 - \Pi_j^{p-1}(X))$. Finally, as 2^l corresponds to the maximal number of equivalent classes, we will need at most $\lceil l * log_p(2) \rceil$ state variables Z_i to define the abstract model.

4. 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 events or a set of states.

Restriction methods are useful to prove "incorrectness" of systems : indeed, when a less general system is shown to violate some safety property, so does the more general one.

A first and standard approach consists in modifying the structure of the automata by removing either states or transitions that are labeled by some fixed events. The symbolic counterpart of these techniques is immediate : let $O(X)$ (resp. $A(Y)$) denote the set of states (resp. events) that are meant to be kept in the system, the symbolic restricted system is then defined by $T'(X,Y,X') = O(X) \oplus \mathcal{T}(X,Y,X') \oplus O(X')$ (resp. $T'(X,Y,X') = \mathcal{T}(X,Y,X') \oplus A(Y)$).

Also, more general restrictions can be considered : in particular on the basis of an acceptance criterion for the remaining behaviors in the restricted system, e.g. expressed in a temporal logic such as the propositional linear time temporal logic PLTL (Manna and Pnueli, 1992).

Here, we somehow overstep this approach by considering the general framework of open synchronous systems : the events of the model are composed of two subparts. One part denotes the stimulus from the outside world and the other part the response of the system. According to this framework, there is no meaning to restrict the possible stimuli, whereas the response can be disallowed, in the same spirit of the pioneer proposal of (Ramadge and Wonham, 1989), and developed by (Marchand and Le Borgne, 1999) for the case of synchronous systems with symbolic techniques. Following these lines, we obtain the acceptable behaviors set computation, i.e. the restriction, by composing the original system with additional equation constraints, called a *controller*.

4.1. THE FRAMEWORK

We consider an "open" ILTS to be a model of the form $S = (X, X', Y, K, \mathcal{T}(X,Y,K,X'), I)$ which meaning is a LTS as before but with transitions like $x \overset{(y,k)}{\to} x'$, but where events are split into a pair (y,k) : component y, still called here the event is furnished by the environment of the system, and k is the response of the system.

We shall say that k is an admissible response in situation (x,y), or for short that k is admissible in (x,y) whenever (x,y,k) is a solution of the polynomial $Q(X,Y,K) \overset{\text{def}}{=} \exists X' \mathcal{T}(X,Y,K,X')$.

Given an open ILTS S, we shall consider restriction specifications, also called "control objectives" in (Marchand and Le Borgne, 1999) that are of two following sorts :

1 the acceptable behaviors of the system are such that all encountered states belong to a given set E ; we call this restriction specification "the invariance of E";

2 the acceptable behaviors of the system are such that along any execution, it is always possible to reach a given set of states E ; we call this restriction specification "the global reachability of E".

Several possible formalisms can be proposed to rigorously express these specifications, e.g. "Alternating Time Logic" of (Alur et al., 1998), "μ-calculus" of (Kozen, 1983), ...but this is out of the scope of this paper.

4.2. THE RESTRICTION PRINCIPLE

The restriction consists in keeping suitable possible responses of the system in a given situation (x, y) to select the "good" possible extensions of the current behavior. The proposed method performs a static computation of the possible responses by delivering two constraints $C_0(X)$ and $C(X, Y, K)$ interpreted as follows : $C_0(X)$ is a polynomial which denotes a set of suitable initial states, and $C(X, Y, K)$ a set of suitable k's for a given situation (x, y). The restricted system is then simply obtained as $S' = (X, X', Y, K, \mathcal{T}(X, Y, K, X') \oplus C(X, Y, K), I \oplus C_0(X))$.

4.3. C_0 AND C COMPUTATION ALGORITHMS

Assume given an open ITLS $S' = (X, X', Y, K, \mathcal{T}(X, Y, K, X'), I)$ and a set of states G, represented by a polynomial, say $G(X)$. It is possible to compute symbolically the set of state for which a response can be chosen to reach G in one step (whatever the stimulus y is). Write $\mathbf{Pre}_K(G)$ this set. Its polynomial representation can be computed by

$$\mathbf{Pre}_K(G) \stackrel{\text{def}}{=} \forall Y((\exists KQ)(X, Y, K) \Rightarrow \exists K \exists X' \mathcal{T}(X, Y, K, X') \oplus G(X'))$$

Now the computation of C_0 and C for "the invariance of E" case can be obtained by

(1) computing the sequence of polynomials

$$E_{i+1}(X) = E_i(X) \oplus \mathbf{Pre}_K(E_i)(X) \text{ init } E(X),$$

until stabilization to get say $\mathbf{Pre}^*_K(E)(X)$. The stabilization is inevitable since the sequence decreases and the set of states is finite. Note that $Sol(\mathbf{Pre}^*_K(E)(X)) \subseteq E$.

(2) defining $C_0(X) = \mathbf{Pre}^*_K(E)(X)$ and $C = \forall X'(\mathcal{T}(X, Y, K, X') \Rightarrow \mathbf{Pre}^*_K(E)(X))$

(3) analyzing the result as follows : if $C_0(X) \oplus I(X) = 0$ has a solution, then (C_0, C) composed with S achieves the restriction objectives, otherwise any behavior of S eventually exits E.

An analogous procedure can apply for the case of *the global reachability* E by changing step **(1)** into *compute the sequence of polynomials*

$$E_{i+1}(X) = E_i(X) * \mathbf{Pre}_K(E_i)(X) \text{ init } E(X)$$

5. CONCLUSION

This paper shows how abstraction techniques can be supported symbolically, thus taking advantage of two well established approaches to the state explosion problem. Abstraction by state fusion is fully detailed, two examples of abstraction by restriction are shown to be closely related to controller synthesis issues, also other restriction specifications such as attractivity, persistence, recurrence, ... can be dealt similarly. We refer to (Marchand and Le Borgne, 1999) and to a forthcoming report.

The methods rely on intensional models for the systems, that for this article are taken to be dynamical equational systems over a finite field. Actually, results can be generalized to an enlarged class of models : the class of first order representable ones.

References

Alur, R., Henzinger, T. A., and Kupferman, O. (1998). Alternating-time temporal logic. *Lecture Notes in Computer Science*, 1536:23–60.

Bensalem, S., Lakhnech, Y., and Owre, S. (1998). Computing abstractions of infinite state systems compositionally and automatically. In *Conference on Computer Aided Verification CAV'98*, LNCS 1427, pages 319–331.

Bryant, R. (1986). Graph-based algorithms for boolean function manipulations. *IEEE Transaction on Computers*, C-45(8):677–691.

Burch, J., Clarke, E., McMillan, K., Dill, D., and Hwang, L. Symbolic model checking: 10^{20} states and beyond. *Information and Computation*, 98(2):142–170.

Clarke, E., Grumberg, O., and Long, D. (1994). Model checking and abstraction. *ACM Transactions on Programming Languages and Systems*, 16(5):1512–1542.

Clarke, E. and Kurshan, R., editors (1990). *Proc. of the 2nd Work. on Computer-Aided Verification*, LNCS 531. Springer-Verlag.

Clarke, E., Long, D., and Mc Millan, K. (1989). A language for compositional specification and verification of finite state hardware controllers. In *Proc. of the 9th Int. Symp. on Computer Hardware Description Languages and Their Applications*, pages 281–295.

Cousot, P. and Cousot, R. (2000). Temporal abstract interpretation. In *Conference Record of the 27th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Programming Languages*, pages 12–25, New York, U.S.A. ACM Press.

Godefroid, P. (1990). Using partial orders to improve automatic verification methods. In *Proc. of the 2nd Work. on Computer-Aided Verification*, LNCS 531, pages 176–185. Springer-Verlag.

Kouchnarenko, O. and Pinchinat, S. (1998). Intensional approachs for symbolic methods. *Electronic Notes in TCS*, 18. http://www.elsevier.nl/locate/entcs/volume18.html.

Kozen, D. (1983). Results on the propositional μ-calculus. *Theoretical Computer Science*, 27(3):333–354.

Larsen, K. G. (1989). Modal specifications. In *Proc. Workshop Automatic Verification Methods for Finite State Systems, Grenoble, LNCS 407*, pages 232–246. Springer-Verlag.

Manna, Z. and Pnueli, A. (1992). *The Temporal Logic of Reactive and Concurrent Systems*, volume I: Specification. Springer-Verlag.

Marchand, H. and Le Borgne, M. (1999). The supervisory control problem of discrete event systems using polynomial methods. Research Report 1271, Irisa.

McMillan, K. (1993). *Symbolic Model Checking: An Approach to the state explosion problem.* Kluwer Academic.

Milner, R. (1989). A complete axiomatisation for observational congruence of finite-state behaviours. *SIAM J. Comput.*, 81(2):227–247.

Park, D. (1981). Concurrency and automata on infinite sequences. In *Proc. 5th GI Conf. on Th. Comp. Sci., LNCS 104*, pages 167–183. Springer-Verlag.

Peled, D. (1994). Combining partial order reductions with on-the-fly model-checking. In *Proc. of Worshop on Computer Aided Verification CAV'94*, LNCS 818, pages 377–390.

Pinchinat, S., Marchand, H., and Le Borgne, M. (1999). Symbolic abstractions of automata and their application to the supervisory control problem. Research Report 1279, IRISA.

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

Van Glabbeek, R. J. (1993). The linear time–branching time spectrum I-I: The semantics of sequential systems with silent moves (extended abstract). In *CONCUR '93*, volume 715 of *LNCS*, pages 66–81, Hildesheim, Germany. Springer-Verlag.