Compositional verification

Evgeny Filonenko
fzh285@wjpserver.cs.uni-sb.de

1 Introduction

Model checkers are useful tools for the verification of finite state systems. However, they traditionally suffer from the state explosion problem. This problem arises in systems which are composed of many parallel processes; in general, the size of the state space grows exponentially with the number of processes. By introducing symbolic representations for sets of states and transition relations, systems with very large state spaces can be verified.
Unfortunately, the symbolic procedures still have limits, and many realistic problems are not tractable due to their size. The obvious method for trying to avoid the state explosion problem is to use the natural decomposition of the system. The goal is to verify properties of individual components, and use them to deduce additional properties of the system. To be able to efficiently decide whether a property is true in all systems containing a given component, we eliminate the existential path quantifiers. Therefore we do compositional verification only concerning formulas of a restricted form of CTL*, which is called ACTL*.
The rest of this paper consists of 3 sections. In section 2 we give a short description of ACTL* logic Section 4 gives methods permitting using Kripke structures effectively to describe behaviour of automata. Also it will be shown how to reduce size of structure with help of preorders. They help to replace a large structure by a smaller structure which satisfies the same properties. Sometimes it is necessary for simplifying the process of verifying one or several properties of the original structure. Section 4 shows how to use structures to describe two or more automata working in parallel. Using several theorems from this section we can deduce, what kind of properties the whole system has consisting of several components. We will give an example of using such theorems at the end of this section.

2 ACTL* logic

In this paper we use the logic ACTL*, obtained by restricting the logic CTL*. In ACTL* a formula may only include the universal path quantifier over paths. To avoid implicit existential path quantifiers resulting from the use of negation, we

assume that the formulas are given in positive normal form, that is negations are only applied to atomic propositions.
The logic is the set of the following formulas:

- for every atomic proposition p, p and ¬p are state formulas
- the constants *true* and *false* are state formulas
- if φ and ψ are state formulas, then $\varphi \wedge \psi$ and $\varphi \vee \psi$ are state formulas
- if φ is a path formula, then A(φ) is a state formula
- if φ is a state formula, then φ is also a path formula
- if φ and ψ are path formulas, then $\varphi \wedge \psi$ and $\varphi \vee \psi$ are path formulas
- if φ and ψ are path formulas, then so are
  - Xφ
  - φ U ψ
  - φ R ψ

3 Preorders between structures

Structures are used to describe states, their relations and state properties in the state systems. In this section we will show how to avoid the state explosion problem by developing techniques that replace a large structure by a smaller structure which satisfies the same properties. We use Kripke structure for describing systems. It is convenient to include a set of initial states S_0 and a set of atomic propositions AP with every Kripke structure M. The following definition gives us a typical Kripke structure M:

Definition 1 *The Kripke structure M = (AP, S, R, S_0, L) is a tuple, where*

- *AP is a finite set of atomic propositions*
- *S is a finite set of states*
- $R \subseteq S \times S$ *is a transition relation*
- $S_0 \subseteq S$ *is a set of initial states*
- *L: $S \rightarrow 2^{AP}$ is a function that maps each state to the set of atomic propositions true in that state*

Further in this paper we shall often use the notion of a *path*. We define it now.

Definition 2 *A path in the Kripke structure M is an infinite sequence of states* $\pi = s_0 s_1 s_2 ...$ *such that for all* $i \in \mathbb{N}$*, transition relation* $R(s_i,\ s_{i+1})$ *holds.*

Since we have defined Kripke structure, it is now possible to construct an abstraction for such a structure, which we call *simulation.* Abstractions are used to hide details of the original structure, hence, they might have a smaller set of atomic propositions. We define abstraction with the help of a simulation relation:

Definition 3 *The relation* $H \subseteq S \times S'$ *is a simulation between M and M′ with their atomic propositions* $AP \supseteq AP'$ *if and only if for all s and* s'*, if* $H(s,\ s')$ *then the following conditions hold:*

1. $L(s) \cap AP' = L'(s')$
2. *For all* s_1 *with* $R(s,\ s_1)$*, there is a state* s'_1 *with the* $R'(s',\ s'_1)$ *and* $H(s_1,\ s'_1)$

The following lemma shows two main properties of simulation relations.

Lemma 1 *The simulation relation* $\preceq$ *is a preorder on the set of structures, i.e. it is*

- *reflexive:* $M \preceq M$
- *transitive: if* $M \preceq M'$ *and* $M' \preceq M''$ *then* $M \preceq M''$

Another property of the simulation relation is that if it associates a state s with a state s′ then every successor of s has a corresponding successor of s'.

Definition 4 *A Kripke structure M′ simulates M if there exist relation* $H \subseteq S \times S'$*, such that for all initial states* s_0 *in M there exists an initial state* s'_0 *in M′ for which* $H(s_0,\ s'_0)$ *holds.*

We denote the fact that M′ simulates M with M $\preceq$ M′.
An example of simulating structures is shown in figure 1. Here AP = AP′ = {a, b, c, d}. In order to show that M simulates M′ we must show that there is a simulation relation H′(s', s). To show it we choose a simulation relation that associates both states s'_1 and s'_2 in M′ with the state s_1 in M. Each of the other states in M′ is associated with all of the states in M that have the same label. So, for this case H′ = $\{(s'_1,s_1),(s'_1,s_2),(s'_0,s_0),(s'_3,s_3),(s'_4,s_4)\}$.
To see that M′ simulates M, we must show that there is a simulation relation H(s, s'). We choose the relation that associates both states s_1 and s_2 in M with state s'_1 im M′. All of the other states of M are associated with states in M′ that have the same label. So, H = $\{(s_1,s'_1),(s_1,s'_2),(s_0,s'_0),(s_3,s'_3),(s_4,s'_4)\}$.

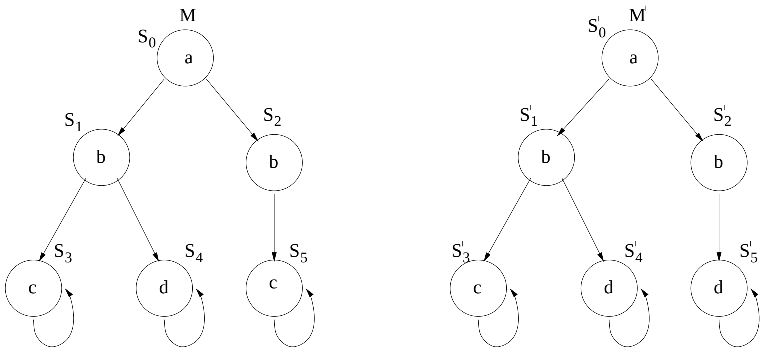

Figure 1 : Two structures simulating each other

To introduce the next lemma we need to know the notion of corresponding paths.

Definition 5 *Paths* $\pi = s_0 s_1 \ldots$ *in structure M and* $\pi' = s'_0 s'_1 \ldots$ *in structure M′ correspond if and only if for for all i, simulation relation* $H(s_i,\ s'_i)$ *holds.*

The following lemma uses the notion of corresponding paths. It shows, that if two systems have states which satisfy simulation relation, then these systems will behave in the same manner following paths starting from such states.

Lemma 2 *Assume that s and s′ are states such that* $H(s,\ s')$*. Then for every path* π *starting from s there is a corresponding path* π' *starting from s′.*

The notion of a simulation relation can be extended for *fair* Kripke structures with fairness constraint. A fair Kripke structure has a set of fairness constraints as an additional component. Each fairness constraints is a subset of the atomic propositions. A fair path in Kripke structure is a path such that for each fairness constraint at least one of its states appears infinitely often on the path. Formally, we make the following definitions:

Definition 6 *The fair Kripke structure* $M = (AP,\ S,\ R,\ S_0,\ L,\ F)$ *is a tuple, where*

- *AP is a finite set of atomic propositions*
- *S is finite set of states*
- $R \subseteq S \times S$ *is a transition relation*
- $S_0 \subseteq S$ *is a set of initial states*
- $L\colon S \rightarrow 2^{AP}$ *is a function that maps each state to the set of atomic propositions true in that state*
- $F \subseteq 2^S$ *is a set of fairness constraints*

Using fairness constraints we can check if path in the Kripke structure is $fair$. Let *inf(*π*)* $\subseteq$ S return states which appear infinitely often in the path π.

Definition 7 *A path* π *in a Kripke structure is fair if and only if for all* P *in* F, $\inf(\pi) \cap P \neq \emptyset$.

We define a $fair$ simulation relation between two structures with help of $fair$ *paths*. Here we apply simulation relation to any two states taken from the fair paths of both Kripke structures M and M$'$.

Definition 8 *Let* M *and* M' *be two fair structures. Assume that* $AP \supseteq AP'$. *The relation* $H \subseteq S \times S'$ *is a* $fair$ *simulation relation between* M *and* M' *if and only if for all* s *and* s', *if* $H(s,\ s')$ *then the following conditions hold:*

1. $L(s) \cap AP' = L'(s')$
2. *For every fair path* $\pi = s_0 s_1 \ldots$ *from* $s = s_0$ *in* M, *there is a fair path* $\pi' = s'_0 s'_1 \ldots$ *from* $s' = s'_0$ *in* M' *such that for for all* i, *the simulation relation* $H(s_i,\ s'_i)$ *holds.*

Like before the notion M $\preceq_F$ M$'$ means that there exist a fair simulation relation H such that for for all initial states $s_0 \in S_0$ in M there is $s'_0 \in$ M$'$ such that the simulation relation H(s_0, s'_0) holds.
The next theorem gives us a means of deducing, whether Kripke structure M satisfies some ACTL* formula f or not. We lean on the fact, that Kripke structure M$'$ fairly simulates Kripke structure M and knowledge, that Kripke structure M$'$ satisfy formula f.

Theorem 1 *If* $M \preceq_F M'$, *then for every* $ACTL^*$ *formula* f *interpreted over fair paths,* M' *fairly satisfies formula* f *(denoted* $M' \models_F f$*) implies* $M \models_F f$.

Proof:
Since the structure M$'$ fairly simulates structure M we can conclude, that for all states s in M there is a state s' in M$'$, such that simulation H(s, s') holds and for every fair path π in M there is fair path π' in M$'$, such that simulation H(π, π') holds. Therefore to prove our theorem we must prove the following two conditions:

1. for every ACTL* (state) formula φ (with all atomic proposition in A$'$), if $s' \models \varphi$ then $s \models \varphi$; and
2. for every ACTL* path formula φ (with all atomic proposition in A$'$), if $\pi' \models \varphi$, then $\pi \models \varphi$.

The proof proceeds by induction on the structure of the formula.

1. If $\varphi = true$ or $\varphi = false$, the result is trivial. If $\varphi = p$, an atomic proposition, then $s' \models \varphi$ if and only if $p \in \mathrm{L}'(s')$. By the definition of simulation relation, $\mathrm{L}(s) \cap \mathrm{A}' = \mathrm{L}'(\mathrm{s}')$, and therefore $p \in \mathrm{L}(s)$ if and only if $p \in \mathrm{L}'(s)$. Thus $s \models \varphi$. The case where $\varphi = \neg p$ is similar.

2. If $\varphi = \varphi_1 \wedge \varphi_2$, then $s' \models \varphi_1$ and $s' \models \varphi_2$. The induction hypothesis implies $s \models \varphi_1$ and $s \models \varphi_2$; hence s $\models \varphi$. The case where $\varphi = \varphi_1 \vee \varphi_2$ is similar.

3. If $\varphi = \mathrm{A}\varphi_1$, then $s' \models \varphi$ if and only if for every fair path π from s, $\pi \models \varphi_1$. Let π be any fair path from s. By the definition of simulation relation, there exist a fair path π' from s' such that $\mathrm{H}(\pi, \pi')$ holds. If $s' \models \varphi$, then $\pi' = \models \varphi_1$ for any π' from s'. The induction hypothesis then implies $\pi \models \varphi_1$, and hence $s \models \varphi$.

4. If φ is a path formula consisting of only a state formula and $\pi' \models \varphi$, then the initial state s' of π' satisfies φ. By the induction hypothesis, s $\models \varphi$, and since s is the initial state of π, $\pi \models \varphi$.

5. The cases for the conjunction and disjunction of path formulas are similar to case 2.

6.
   - If $\varphi = \mathrm{X}\varphi_1$, then $\pi' \models \varphi$ implies $\pi'^1 \models \varphi$ (π'^1 is suffix of π). Now since we have simulation relation $\mathrm{H}(\pi, \pi')$ for paths π and π', we also have simulation relation $\mathrm{H}(\pi'^1, \pi^1)$. Then the induction hypothesis implies $\pi^1 \models \varphi_1$. Thus $\pi \models \varphi$.
   - If $\varphi = \varphi_1$ U φ_2, then $\pi' \models \varphi$ implies, that there exists place n in the path π' such that $\pi'^n \models \varphi_2$ and for all i < n, $\pi'^i \models \varphi_1$. Since we have simulation relation $\mathrm{H}(\pi, \pi')$, we have simulation relation $\mathrm{H}(\pi^j, \pi'^j)$ for any j. Applying the induction hypothesis, $\pi^n \models \varphi_2$ and $\pi^i \models \varphi_1$ for all i < n. Hence $\pi \models \varphi$.
   - The case where $\varphi = \varphi_1$ R φ_2 is similar to the previous two cases. □

4 Compositional reasoning

The method of compositional reasoning consists of three parts. First we decompose specifications into properties that describe the behavior of small parts of the system. Second we prove, that each part satisfies its local property. Third, we can deduce properties of the whole system from the local properties. Compositional reasoning uses the assume-guarantee paradigm, a technique, that verifies each component process separately. An example of the assume-guarantee paradigm is as follows:
Suppose that there are two processes M and M′. Since the behavior of process M depends on the behavior of process M′, the user specifies a set of assumptions

that must be satisfied by M′ in order to guarantee the correctness of process M. Since the behavior of process M′ also depends on the behavior of process M, the user specifies a set of assumptions that must be satisfied by M in order to guarantee the correctness of process M′. By combining the set of assumed and guaranteed properties of M and M′ in an appropriate manner, it is possible to establish the correctness of the entire system M ‖ M′ without constructing the global state-transition graph.

4.1 Composition of structures

Let $\mathrm{M} = (\mathrm{AP}, \mathrm{S}, \mathrm{R}, \mathrm{S}_0, \mathrm{L}, \mathrm{F})$ and $\mathrm{M}' = (\mathrm{AP}', \mathrm{S}', \mathrm{R}', \mathrm{S}'_0, \mathrm{L}', \mathrm{F}')$ be two structures. The *parallel composition* of M and M′, denoted by M ‖ M′, is the structure M″ defined by:

1. $\mathrm{S}'' = \{\ (\mathrm{s},\mathrm{s}') \mid \mathrm{L}(\mathrm{s}) \cap \mathrm{AP}' = \mathrm{L}'(\mathrm{s}') \cap \mathrm{AP}\ \}$
2. $\mathrm{S}'_0 = (\mathrm{S}_0 \times \mathrm{S}'_0) \cap \mathrm{S}''$
3. $\mathrm{AP}'' = \mathrm{AP} \cup \mathrm{AP}'$
4. $\mathrm{L}''((\mathrm{s}, \mathrm{s}')) = \mathrm{L}(\mathrm{s}) \cup \mathrm{L}'(\mathrm{s}')$
5. $\mathrm{R}''((\mathrm{s}, \mathrm{s}'), (\mathrm{t}, \mathrm{t}'))$ if and only if $\mathrm{R}(\mathrm{s}, \mathrm{t})$ and $\mathrm{R}'(\mathrm{s}', \mathrm{t}')$
6. $\mathrm{F}'' = \{\ (\mathrm{P} \times \mathrm{S}') \cap \mathrm{S}'' \mid \mathrm{P} \in \mathrm{F}\ \} \cup \{\ (\mathrm{S} \times \mathrm{P}') \cap \mathrm{S}'' \mid \mathrm{P}' \in \mathrm{F}'\ \}$

The composed model behaves in a synchronous way, meaning that with each transition of the composition we have a joint transition of the two components. States of the composition are pairs of component states that agree on the common atomic propositions (via the intersection $\mathrm{L}(\mathrm{s}) \cap \mathrm{AP}' = \mathrm{L}'(\mathrm{s}') \cap \mathrm{AP}$). The fairness constraint is designed to insure the *fair path property*, which states that a path in M ‖ M′ is fair if and only if its restriction to each component results in a fair path. The first set of pairs in the constraint

$$\{(P \times S') \cap S'' \mid P \in F\}$$

insures that the restriction of a path in M″ to its component in S is a fair path in M. The second set of pairs

$$\{(S \times P') \cap S'' \mid P' \in F'\}$$

insures that the restriction of the path to its component in S′ is a fair path in M′. It is necessary to intersect each $\mathrm{P} \times \mathrm{S}'$ and $\mathrm{S} \times \mathrm{P}'$ with S″ as they may contain pairs that are not states in M″.
The next three theorems deal with the connection between parallel composition and the fair simulation preorder $\preceq_F$.

Theorem 2 *For all M and M′, M ∥ M′ $\preceq_F$ M*

Proof: Let S″ be the set of states of M ∥ M′. Define a relation H by

$$H = \{((s, s'), s) \mid (s, s') \in S''\}.$$

Now we need to show, that relation H is a simulation relation between states (s, s') and s. So, we must prove that H satisfies the two requirements from the definition of the simulation relation.

1. - If (s_0, s'_0) is an initial state of M ∥ M′, then $s_0 \in S_0$.
   - The label of (s, s') is L(s) ∪ L′(s'), and we can see that (L(s) ∪ L′(s')) ∩ AP = L(s). This is the first requirement of simulation relation.
2. If (s_0, s'_0), (s_1, s'_1),... is a fair path in M ∥ M′, then by the fair path property, $s_0, s_1, \ldots$ is a fair path in Kripke structure M. Therefore by the definition of H, we have H((s_i, s'_i), s'_i) for every i. Hence, H is a simulation relation and M ∥ M′ $\preceq_F$ M. □

The following theorem permits a component of a system to be replaced by an abstraction of that component. Thus, in order to show that some property is true in the system M ∥ M″, we can replace M by an abstraction M′ and then verify that the property holds in M′ ∥ M″. Checking M $\preceq_F$ M′ insures that M′ is indeed an abstraction of M.

Theorem 3 *For all M, M′ and M″, if M $\preceq_F$ M′ then M ∥ M″ $\preceq_F$ M′ ∥ M″*

Proof: Let H_0 be a simulation relation between M and M′. Define H_1 by

$$H_1 = \{((s, s''), (s', s'')) \mid H_0(s, s')\}.$$

1. We show that H_1 is a simulation relation. The first requirement for simulation relations is (L(s) ∪ L″(s″)) ∩ (AP′ ∪ AP″) = L′(s′) ∪ L″(s″).

$$\begin{aligned}(L(s) \cup L''(s'')) \cap (AP' \cup AP'') &= (L(s) \cap AP') \cup (L(s) \cap AP'') \\ &\quad \cup (L''(s'') \cap (AP' \cup AP'')) \\ &= L'(s') \cup (L''(s'') \cap AP') \cup L''(s'') \\ &= L'(s') \cup L''(s'')\end{aligned} \tag{1}$$

2. The second requirement is that for every fair path π = (s_0, s''_0), (s_1, s''_1), ... in M ∥M″, there is a fair path π' = (s'_0, s''_0), (s'_1, s''_1),... in M′ ∥ M″ such that for all i ≥ 0, H_1((s_i, s''_i), (s'_i, s''_i)).

- Let $(s_0, s''_0)(s_1, s''_1)\ldots$ be a fair path in M $\|$ M$''$ from (s, s$''$) = (s_0, s''_0), then from the definition of compositional structure we have that for all i $\geq$ 0, L(s_i) $\cap$ AP$''$ = L$''$(s''_i) $\cap$ AP.
- From the *fair path property* we can conclude that $\pi = s_0 s_1 \ldots$ is a fair path in M and $\pi'' = s''_0 s''_1 \ldots$ is a fair path in M$''$.
- From our assumption, that H_0(s, s$'$) is a simulation relation between M and M$'$, there is a fair path $\pi' = s'_0 s'_1 \ldots$ in M$'$.
- Now we can construct the path (s'_0, s''_0), (s'_1, s''_1),... from two fair paths π' and π''. We only need to show, that all states in constructed path are in M$'$ $\|$ M$''$. To show this we prove that the equation L$'$(s'_i $\cap$ AP$''$) = L$''$(s''_i $\cap$ AP$'$) holds for all i $\geq$ 0:

$$\begin{aligned} L'(s'_i \cap AP'' &= (L(s_i) \cap AP') \cap AP'' \\ &= (L(s_i) \cap AP'') \cap AP' \\ &\quad \text{(from def. of composed structure)} \\ &= (L''(s''_i) \cap AP') \cap AP' \\ &= L''(s''_i) \cap AP' (as AP' \subseteq AP) \end{aligned} \tag{2}$$

  Since our equation is true, each (s'_i, s''_i) is a state in M$'$ $\|$ M$''$, therefore (s'_0, s''_0), (s'_1, s''_1),... is a fair path in M$'$ $\|$ M$''$ and H_1((s_i, s''_i), (s'_i, s''_i)) holds for all i $\geq$ 0. □

Theorem 4 *For all M, M $\preceq_F$ M $\|$ M*

Proof: Define the relation H = {s, (s, s) | s $\in$ S}. We show that H is a simulation relation.

1. If $s_0 \in S_0$, then by defenition of composition (s_0, s_0) - is an initial state of M $\|$ M. Trivially we have L(s, s) = L(s).

2. If $s_0 s_1 \ldots$ is a fair path in M, then (s_0, s_0),(s_1, s_1),... is a fair path in M $\|$ M. Therefore by the definition of H, we have H(s_i, (s_i, s_i)) for all i $\geq$ 0. Hence H is a fair simulation relation and M $\preceq_F$ M $\|$ M. □

4.2 Verification example

In the following example we use the notion of *maximal model for the formula f* (tableau construction). We give a definition for this maximal model first.

Definition 9 *The maximal model M_{max} for the formula f under the relation $\preceq_F$ is a Kripke Structure M_{max}, which has only the states, in which one or several subformulas obtained from f are true (maximum number of such states), and from each such state there is a fair path, that satisfy formula f. This model has property:*

$$M \models_F f \text{ if and only if } M \preceq_F M_{max}$$

With the *tableau method* we can construct *maximal model* T_f for formula f. Proofs on ACTL* tableau construction can be found in [1].
Now we want to show, having some assumptions as preconditions, that Kripke structure M $\|$ M′ constructed from two models M and M′, satisfies formula f. To prove this we use all theorems, which we have proved above and tableau construction.
Assume that A, M and M′ represent finite state models, g and f represent fair ACTL* formulas.
We consider the situation, when we have Kripke structure A, which is the simple model of structure M. It simulates Kripke structure M and can have less number of states. It is easier to verify some properties for small strucures. Our assumptions are:

- It is easy to obtain A from M
- It easy to prove that $\mathrm{A} \parallel \mathrm{M}' \models_F \mathrm{g}$
- It easy to prove that $\mathrm{T}_g \parallel \mathrm{M}' \models_F \mathrm{g}$

From the assumptions above we have the following hypotheses:

- $\mathrm{M} \preceq_F \mathrm{A}$
- $\mathrm{A} \parallel \mathrm{M}' \models_F \mathrm{g}$
- $\mathrm{T}_g \parallel \mathrm{M}' \models_F \mathrm{g}$

Using the assume-guarantee technique, we show that $\mathrm{M} \parallel \mathrm{M}' \models_F f$ holds from the above assumptions.

Proof:

1. $M \preceq_F A$	Hypotheses
2. $M \parallel M' \preceq_F A \parallel M'$	Line 1 and Th.4 (if M $\preceq_F$ M′, then M $\parallel$ M″ $\preceq_F$ M′ $\parallel$ M″)
3. $A \parallel M' \models_F g$	Hypotheses
4. $A \parallel M' \preceq_F T_g$	Line 3 and def.9 (M′ $\models_F$ f iff M′ $\preceq_F$ T_f)
5. $M \parallel M' \preceq_F T_g$	Lines 2, 4 and transitivity of $\preceq_F$
6. $M \parallel M \parallel M' \preceq_F T_g \parallel M$	Line 5 and Theorem 4
7. $T_g \parallel M \models_F f$	Hypotheses
8. $M \parallel M \parallel M' \preceq_F f$	Lines 6, 7 and Theorem 1
9. $M \preceq_F M \parallel M$	Theorem 5
10. $M \parallel M' \preceq_F M \parallel M \parallel M'$	Line 9 and Theorem 4
11. $M \parallel M' \models_F f$	Lines 8, 10 and Theorem 1 $\square$

Literature

[1] O.Grumberg, D.E.Long: Model Checking and Modular Verification (1991)
[2] E.M.Clarke, D.E.Long, K.L.McMillan: Compositional Model Checking (1989)
[3] E.M.Clarke, O.Grumberg, A.Peled: Model Checking, The MIT Press, Cambridge, Massachusetts, London, England (1999)