Generalized Model Checking: Theory and Applications

Patrice Godefroid

Bell Laboratories, Lucent Technologies

References:
(BG99) with G. Bruns, CAV'99
(BG00) with G. Bruns, CONCUR'2000
(GJH01) with R. Jagadeesan and M. Huth, CONCUR'2001

Introduction: Model Checking

Reactive system:

- controls something;
- continually interacts with its environment.

Examples: telephone switch, airplane, ATM, power plant, pacemaker, etc.

Viewed as a FSM (automaton), called **state space**.

Behavior described in terms of sequences of states/transitions.

Language for temporal properties: **Temporal Logic**

- Example: $G(p \Rightarrow Fq)$ is an LTL formula.

Given a system M and a temporal-logic formula ϕ,
"does M satisfy ϕ?" = **Model Checking**

Currently (2001), there are many model-checking tools around, including in industry (hardware and software).

Reasoning about Partial State Spaces

Model Checking consists of two steps:

- Construct the state space M (e.g., Kripke structure) of the system.
- Check that $M \models \phi$; the answer is "true" or "false".

But what if M is only partially known?
(because too large, use of abstraction,...)

Partial state space M' = incomplete representation of state space.

What properties of M can one check by analyzing M'?

- some properties can be proved (true);
- some properties can be disproved (false);
- some properties cannot be determined! (unknown)

Overview

Reasoning about partial state spaces requires models and logics that can represent the third possibility [BG99].

- Models for partial state spaces: "Partial" Kripke structures (but also MTS, XTS, ...).
- Measure of completeness: completeness preorder $\preceq$.
- Logics L: 3-valued temporal logics ($true$, $false$ and $\perp$).
- Correspondence between $\preceq$ and L:
  Partial $\preceq$ Complete **iff** $\forall \phi \in L$:
  - Partial $\models \phi \Rightarrow$ Complete $\models \phi$ (1)
  - Partial $\not\models \phi \Rightarrow$ Complete $\not\models \phi$ (2)
- Model checking for 3-valued temporal logics.
- Better precision: Generalized Model Checking
  - Given M and ϕ, does there exists M_c such that $M \preceq M_c$ and $M_c \models \phi$?
  - Generalizes both model checking and satisfiability.
- Applications.
- Related Work.

Background: Kripke Struct., Modal Logic

Kripke structure $M = (S, L, \mathcal{R})$ with

- S is a set of states,
- $L : S \times P \to \{true, false\}$ is a labeling function,
- $\mathcal{R} : S \times S$ is a transition relation.

Propositional modal logic

Syntax:

$$\phi ::= p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \Diamond\,\phi$$

Semantics:

$$\begin{array}{lll}
(M, s) & \models p & \text{if } L(s, p) = true \\
(M, s) & \models \neg\phi & \text{if } (M, s) \not\models \phi \\
(M, s) & \models \phi_1 \wedge \phi_2 & \text{if } (M, s) \models \phi_1 \text{ and } (M, s) \models \phi_2 \\
(M, s) & \models \Diamond\,\phi & \text{if } (M, t) \models \phi \text{ for some } t \\
 & & \quad \text{such that } (s, t) \in \mathcal{R}
\end{array}$$

Example:

Background: Kripke Struct., Modal Logic

The equivalence relation induced by propositional modal logic coincides with the notion of bisimulation relation[HM85].

Definition: Let $M_1 = (S_1, L_1, \mathcal{R}_1)$ and $M_2 = (S_2, L_2, \mathcal{R}_2)$ be Kripke structures. A binary relation $\mathcal{B} \subseteq S_1 \times S_2$ is a **bisimulation relation** if $(s_1, s_2) \in \mathcal{B}$ implies:

- $\forall p \in P : L_1(s_1, p) = L_2(s_2, p)$,
- if $(s_1, s_1') \in \mathcal{R}_1$, then there is some $(s_2, s_2') \in \mathcal{R}_2$ such that $(s_1', s_2') \in \mathcal{B}$, and
- if $(s_2, s_2') \in \mathcal{R}_2$, then there is some $(s_1, s_1') \in \mathcal{R}_1$ such that $(s_1', s_2') \in \mathcal{B}$.

Then, s_1 and s_2 are **bisimilar**, denoted $s_1 \sim s_2$.

Theorem: Let $M_1 = (S_1, L_1, \mathcal{R}_1)$ and $M_2 = (S_2, L_2, \mathcal{R}_2)$ be Kripke structures such that $s_1 \in S_1$ and $s_2 \in S_2$, and let Φ denote the set of all formulas of propositional modal logic. Then

$$(\forall \phi \in \Phi : [(M_1, s_1) \models \phi] = [(M_2, s_2) \models \phi]) \text{ iff } s_1 \sim s_2.$$

Propositional modal logic is then called a **logical characterization** of $\sim$.

Models for Partial State Spaces

Partial Kripke structure $M = (S, L, \mathcal{R})$ with

- S is a set of states,
- $L : S \times P \rightarrow \{true, false, \perp\}$ is a labeling function,
- $\mathcal{R} : S \times S$ is a transition relation.

$\perp$ means "unknown whether true or false".

Example:

3-Valued Propositional Modal Logic

Syntax (unchanged): $\phi ::= p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \Diamond\, \phi$

Semantics:

$$\begin{array}{lcl}
[(M,s) \models p] & = & L(s,p) \\
[(M,s) \models \neg\phi] & = & \mathrm{comp}([(M,s) \models \phi]) \\
[(M,s) \models \phi_1 \wedge \phi_2] & = & \min([(M,s) \models \phi_1], [(M,s) \models \phi_2]) \\
[(M,s) \models \Diamond\, \phi] & = & \max(\{[(M,t) \models \phi] \mid (s,t) \in \mathcal{R}\})
\end{array}$$

- Negation maps $true$ to $false$, $false$ to $true$, and $\perp$ to $\perp$.
- Conjunction is defined as the minimum of its arguments, under the "truth" order $false < \perp < true$.
  Ex: $true \wedge \perp = \perp$, $false \wedge \perp = false$.
- Extension of Kleene's strongest regular 3-valued propositional logic.
- The usual meaning of the propositional operators is preserved when applied to values $true$ and $false$.

Example:

Partial Kripke Structures and 3-Valued Modal Logic

Let $\leq$ be the "information" ordering on truth values in which $\bot \leq true$ and $\bot \leq false$.

Definition: [BG99]
Let $M_1 = (S_1, L_1, \mathcal{R}_1)$ and $M_2 = (S_2, L_2, \mathcal{R}_2)$ be partial Kripke structures. The *completeness preorder* is the greatest relation $\preceq \subseteq S_1 \times S_2$ such that $s_1 \preceq s_2$ implies the following:

- $\forall p \in P : L_1(s_1, p) \leq L_2(s_2, p)$,
- if $(s_1, s_1') \in \mathcal{R}_1$, then there is some $(s_2, s_2') \in \mathcal{R}_2$ such that $s_1' \preceq s_2'$, and
- if $(s_2, s_2') \in \mathcal{R}_2$, then there is some $(s_1, s_1') \in \mathcal{R}_1$ $s_1' \preceq s_2'$.

Example:

Partial Kripke Structures and 3-Valued Modal Logic (Continued)

Theorem: [BG99]
Let M_1 and M_2 be partial Kripke structures such that $s_1 \in M_1$ and $s_2 \in M_2$, and let Φ denote the set of all formulas of 3-valued propositional modal logic. Then

$$s_1 \preceq s_2 \text{ iff } (\forall \phi \in \Phi : [(M_1, s_1) \models \phi] \leq [(M_2, s_2) \models \phi]).$$

Thus, partial Kripke structures that are "more complete" with respect to $\preceq$ have more definite properties with respect to $\leq$.

Example:

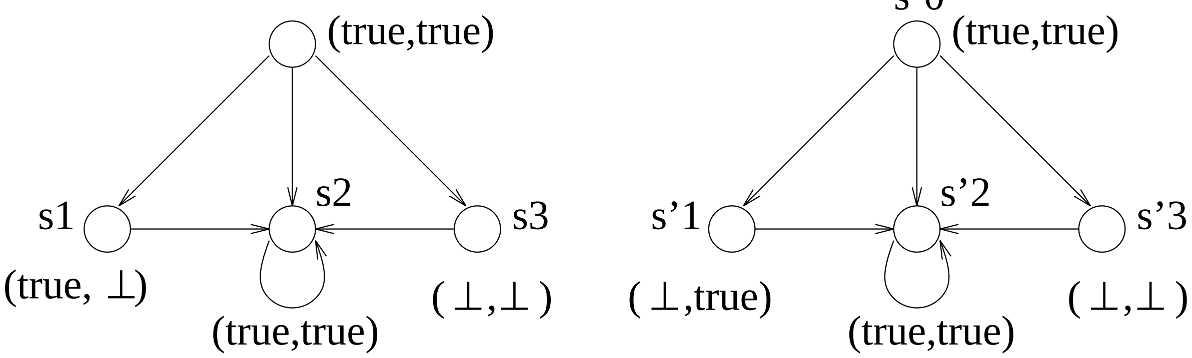
Partial Kripke Structures and 3-Valued Modal Logic (Continued)

Corollary: [BG99]
Let $M_1 = (S_1, L_1, \mathcal{R}_1)$ and $M_2 = (S_2, L_2, \mathcal{R}_2)$ be partial Kripke structures such that $s_1 \in S_1$ and $s_2 \in S_2$, and let Φ denote the set of all formulas of 3-valued propositional modal logic. Then

$$(\forall \phi \in \Phi : [(M_1, s_1) \models \phi] = [(M_2, s_2) \models \phi]) \text{ iff}$$
$$(s_1 \preceq s_2 \text{ and } s_2 \preceq s_1).$$

Note: $s_1 \sim s_2$ implies $s_1 \preceq s_2$ and $s_2 \preceq s_1$,
but $s_1 \preceq s_2$ and $s_2 \preceq s_1$ does not imply $s_1 \sim s_2$!

Example:
s_0 and s'_0 are not bisimilar, but cannot be distinguished by any formula of 3-valued propositional modal logic.



3-Valued Model Checking

Problem: Given a state s of a partial Kripke structure M and a formula ϕ, what is the value of $[(M, s) \models \phi]$?

Theorem: [BG00]
The model-checking problem for a 3-valued temporal logic can be reduced to two model-checking problems for the corresponding 2-valued logic.

Procedure:
STEP 1: complete M into two "extreme" complete Kripke structures, called the **optimistic** and **pessimistic** completions:

- Extend P to P' such that, for every $p \in P$ there exists a $\overline{p} \in P'$ such that $L(s, p) = \mathrm{comp}(L(s, \overline{p}))$ for all s in S.
- $M_o = (S, L_o, \mathcal{R})$ with

$$L_o(s, p) \stackrel{\mathrm{def}}{=} \begin{cases} true & \text{if } L(s, p) = \perp \\ L(s, p) & \text{otherwise} \end{cases}$$

- $M_p = (S, L_p, \mathcal{R})$ with

$$L_p(s, p) \stackrel{\mathrm{def}}{=} \begin{cases} false & \text{if } L(s, p) = \perp \\ L(s, p) & \text{otherwise} \end{cases}$$

3-Valued Model Checking (Continued)

STEP 2: transform ϕ to its positive form $T(\phi)$ with $T(\neg p) = \overline{p}$.

STEP 3: evaluate $T(\phi)$ on M_o and M_p using traditional 2-valued model checking, and combine the results:

$$[(M,s) \models \phi] = \begin{cases} true & \text{if } (M_p, s) \models T(\phi) \\ false & \text{if } (M_o, s) \not\models T(\phi) \\ \perp & \text{otherwise} \end{cases}$$

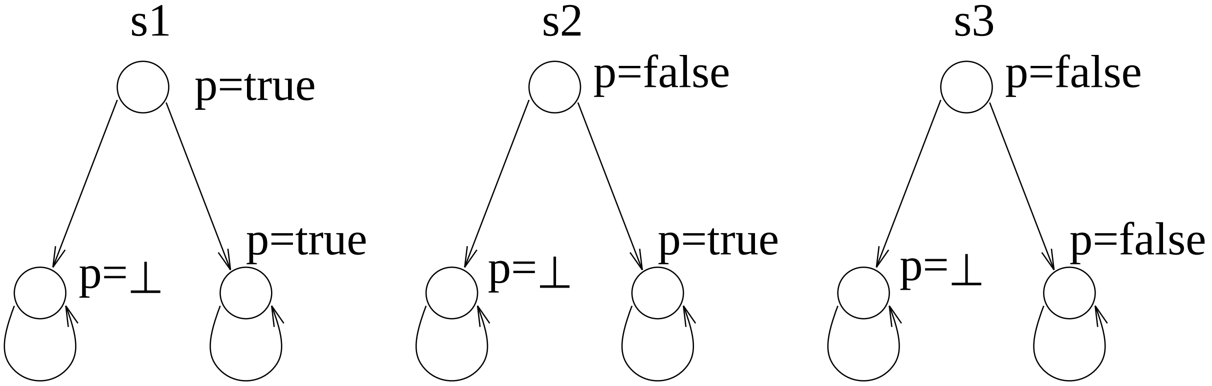
This can be done using existing model-checking tools!

Corollary: 3-valued model checking has the same complexity as traditional 2-valued model checking.

Examples

Application 1:
Generation of a partial Kripke structure from a partial state-space exploration such that, by construction, $s'_0 \preceq s_0$ [BG99].

Examples:



- $[s_1 \models A(true\,\mathcal{U}\,p)] = true$
- $[s_2 \models A(true\,\mathcal{U}\,p)] = \perp$
- $[s_3 \models A(true\,\mathcal{U}\,p)] = false$

New 3-Valued Semantics

Observation: One can argue that the previous semantics returns $\perp$ more often than it should...

Example: Consider $M = (\{s_0\}, L, \{(s_0, s_0)\})$ with $L(s_0, p) = \perp$ and $L(s_0, q) = true$.

$[(M, s_0) \models\ p \vee \neg p\ ] = \perp$

$[(M, s_0) \models\ q \wedge (p \vee \neg p)\ ] = \perp$

although these formulas are $true$ for all complete Kripke struct. more complete than M! $\rightsquigarrow$ **New 3-valued semantics**

Definition: The truth value of a temporal-logic formula ϕ in a state s of a partial Kripke structure M under the **thorough** interpretation, written $[(M, s) \models \phi]_t$, is:

$$[(M, s) \models \phi]_t = \begin{cases} true & \text{if } (M', s') \models \phi \text{ for all } (M', s') : s \preceq s' \\ false & \text{if } (M', s') \not\models \phi \text{ for all } (M', s') : s \preceq s' \\ \perp & \text{otherwise} \end{cases}$$

Is model checking more expensive with this semantics?

YES! Indeed, one needs to solve two

Generalized Model-Checking Problems

Generalized Model Checking

Definition: [BG00]
Given a state s of a partial Kripke structure M and a formula ϕ of a temporal logic L, does there exist a state s' of a complete Kripke structure M' such that $s \preceq s'$ and $(M', s') \models \phi$?

This **generalized model-checking problem** is thus a generalization of both satisfiability (all Kripke structures are potential solutions) and model checking (a single Kripke structure needs to be checked).

Theorem:
Let L denote the modal μ-calculus or any of its fragments. Then the satisfiability problem for L is reducible (in linear-time and logarithmic space) to the generalized model-checking problem for L.

Thus, generalized model checking is as hard as satisfiability.

Is it harder?

Branching-Time Temporal Logics

Theorem: (CTL) Given a state s_0 of partial Kripke structure $M = (S, L, \mathcal{R})$ and a CTL formula ϕ, one can construct an alternating Büchi word automaton $A_{(M,s_0),\phi}$ over a 1-letter alphabet with at most $O(|S| \cdot 2^{O(|\phi|)})$ states such that

$$(\exists (M', s_0') : s_0 \preceq s_0' \text{ and } (M', s_0') \models \phi) \text{ iff } \mathcal{L}(A_{(M,s_0),\phi}) \neq \emptyset.$$

Corollary: if such a M' exists, there exists one with at most $|S| \cdot 2^{O(|\phi|)}$ states.

Theorem: The generalized model-checking problem for a state s_0 of a partial Kripke structure $M = (S, L, \mathcal{R})$ and a CTL formula ϕ can be decided in time $O(|S|^2 \cdot 2^{O(|\phi|)})$.

Theorem: The generalized model-checking problem for CTL is EXPTIME-complete.

See [BG00] for other branching-time temporal logics.

Theorem: (Summary) Let L denote propositional logic, propositional modal logic, CTL, or any branching-time logic including CTL (such as CTL* or the modal μ-calculus). The generalized model-checking problem for the logic L has the same complexity as the satisfiability problem for L.

Linear-Time Temporal Logics

Theorem: (LTL)
Given a state s_0 of partial Kripke structure $M = (S, L, \mathcal{R})$ and an LTL formula ϕ, one can construct an alternating Büchi word automaton $A_{(M,s_0),\phi}$ over a 1-letter alphabet with at most $O(|S| \cdot 2^{|\phi|})$ states such that

$$(\exists(M', s_0') : s_0 \preceq s_0' \text{ and } (M', s_0') \models \phi) \text{ iff } \mathcal{L}(A_{(M,s_0),\phi}) \neq \emptyset.$$

Theorem:
The generalized model-checking problem for a state s_0 of a partial Kripke structure $M = (S, L, R)$ and an LTL formula ϕ can be decided in time $O(|S|^2 \cdot 2^{2|\phi|})$.

Theorem:
The generalized model-checking problem for linear-time temporal logic is EXPTIME-complete.

For LTL, generalized model checking is thus **harder** than satisfiability and model checking!

(both of these problems are PSPACE-complete for LTL)

Note: similar phenomenon for "realizability" and "synthesis" for LTL specifications [ALW89,PR89].

Summary

Model Checking: (3-valued semantics)

- MC can be reduced to two 2-valued MC problems.
- MC has the same complexity as 2-valued MC.

Generalized Model Checking: (thorough 3-val. sem.)

- For BTL, GMC has the same complexity as satisfiablity.
- For LTL, GMC is harder than satisfiablity and MC.

Logic	**MC**	**SAT**	**GMC**
PL	Linear	NP	NP
PML	Linear	PSPACE	PSPACE
CTL	Linear	EXPTIME	EXPTIME
μ-calculus	NP$\cap$co-NP	EXPTIME	EXPTIME
LTL	PSPACE	PSPACE	EXPTIME

Another Model: Modal Transition Systems

Modal Transition Systems [LT88] were originally introduced as part of a top-down process-refinement framework.

Labeled Transition System $M = (S, L, \mathcal{R})$ with S is a set of states, L is a set of action symbols (labels), $\mathcal{R} : S \times L \times S$ is a transition relation.

Modal Transition System $M = (M^{\text{must}}, M^{\text{may}})$ where $M^{\text{must}} = (S, L, \mathcal{R}_{\text{must}})$ and $M^{\text{may}} = (S, L, \mathcal{R}_{\text{may}})$ are LTSs such that $\mathcal{R}_{\text{must}} \subseteq \mathcal{R}_{\text{may}}$.

Completeness preorder: Given two MTSs M_1 (abstract) and M_2 (concrete), we have $M_1 \preceq M_2$ if there exists a relation $\rho \subseteq S_{M_1} \times S_{M_2}$ such that (i) ρ is a simulation from M_1^{must} to M_2^{must} and (ii) ρ is a simulation from M_2^{may} to M_1^{may}.

PKS can be translated to MTS, and vice-versa ("must be true" means "true", "may not be true" means "false", otherwise "$\perp$").

Our results for PKS (logical characterization with negation, algorithms and complexity bounds for MC and GMC) carry over to MTS.

Applications

Application 1: partial state-space exploration [BG99]

Application 2: automatic abstraction [GHJ01]

IDEA: given a concrete system C, if $C \models \phi$ cannot be decided, generate a (smaller) abstraction A and check $A \models \phi$ instead.

Example: predicate abstraction

- Let $\psi_1, \ldots, \psi_n$ be n predicates on variables of C.
- Abstract states are vectors of n bits b_i.
- A concrete state c is abstracted by an abstract state

$$[c] = (b_1, \ldots, b_n) \text{ iff } \forall 1 \leq i \leq n : b_i = \psi_i(c).$$

State of the art: A is a traditional "state space" and

$$(c_1 \to c_2) \Rightarrow ([c_1] \to [c_2]).$$

In other words, A **simulates** C.

- If ϕ is a $\forall$-property, $A \models \phi$ implies $C \models \phi$,
- but $A \not\models \phi$ does not imply anything about C!

Automatic Abstraction Revisited

Observation: A is really a partial state space!

A can be represented by a modal transition system [GHJ01].

Abstraction relation:

1. $(c_1 \rightarrow c_2) \Rightarrow ([c_1] \rightarrow_{\text{may}} [c_2])$
2. $(\forall c_i \in [a] : \exists c_i \rightarrow c_j \wedge c_j \in [a']) \Rightarrow ([a] \rightarrow_{\text{must}} [a'])$

By construction, $A \preceq C$.

Computing an MTS A using (1)+(2) can be done at the same computational cost (same complexity) as computing a "conservative" abstraction (simulation) using (1) alone: (2) can be built by dualizing all the steps necessary to build (1).

This is shown for predicate and cartesian abstraction in [GHJ01].

Automatic Abstraction Revisited (Continued)

Having $A \preceq C$ yields three **improvements**:

1. Any temporal-logic formula ϕ can now be checked.
2. Verification results are now both complete and sound:
   - $A \models \phi$ implies $C \models \phi$
   - $A \not\models \phi$ implies $C \not\models \phi$
3. Generalized model checking can now improve precision of verification results for a "same" cost in $|A|$.
   - Traditional "focussing" (adding predicates) improves precision but may involve an exponential blow-up in $|A|$.
   - GMC improves precision on a property-driven basis and in polynomial time in $|A|$!

Current work: quadratic to linear, practical examples, relationship to abstract interpretation, ...

Other Related Work

"Mixed transition systems" [Dams et al.,94]

- Intuitively, a mixed transition system for a LTS is an abstract MTS without the constraint $\mathcal{R}_{\text{must}} \subseteq \mathcal{R}_{\text{may}}$.
- Hence, restricted support of negation, alternations in ϕ.
- Dams' work is very similar in spirit (reasoning about abstractions of concrete programs for model checking).

Conservative abstraction for the full mu-calculus: [SS99]

"Extended transition systems" [Mil81]

- XTS = LTS + "divergence predicate"
- In [BG99], we define 3-valued Hennessy-Milner Logic, and show that 3-valued HML logically characterizes the "divergence preorder" [Mil81,Wal90].
- Close correspondence with Plotkin's intuitionistic modal logic (inspired our reduction from 3-val MC to 2-val MC).

3-Valued logic for program analysis: [SRW99] shape graphs, first-order 3-valued logic, "focussing",... (inspired the beginning of this work but technical details fairly different)