Model Checking Partial State Spaces with 3-Valued Temporal Logics (Extended Abstract)

Glenn Bruns and Patrice Godefroid

Bell Laboratories, Lucent Technologies
{grb,god}@bell-labs.com

Abstract. We address the problem of relating the result of model checking a partial state space of a system to the properties actually possessed by the system. We represent incomplete state spaces as partial Kripke structures, and give a 3-valued interpretation to modal logic formulas on these structures. The third truth value $\perp$ means "unknown whether true or false". We define a preorder on partial Kripke structures that reflects their degree of completeness. We then provide a logical characterization of this preorder. This characterization thus relates properties of less complete structures to properties of more complete structures. We present similar results for labeled transition systems and show a connection to intuitionistic modal logic. We also present a 3-valued CTL model checking algorithm, which returns $\perp$ only when the partial state space lacks information needed for a definite answer about the complete state space.

1 Introduction

The theory and engineering of model checking has led to tools that can analyze systems with millions of states. However, many systems we would like to analyze have state spaces that are still often orders of magnitude larger than these tools can handle. In this case a common approach is simply to explore just a part of the state space; unexplored states and transitions are then absent in the incomplete or "partial" state space.

In model checking a partial state space, the main issue is how answers obtained in checking the partial state space relate to properties of the full state space. Obviously, one cannot assume that all answers apply to the full state space. The partial state space may be lacking "bad" states one is interested in avoiding, or "good" states that one is interested in reaching. Naively, one could work only with simple safety properties such as "all reachable states satisfy proposition p", with the understanding that if the answer is false in the partial state space then it is also false in the complete state space. But this approach too strongly restricts the properties we can check.

Clearly a more systematic understanding is needed. We can work logically and ask: which class of properties will hold of the complete state space just if they hold of the partial state space? Or we can work operationally and ask: how should we describe the relationship between a partial state space and a more complete one? For example, we might consider that a partial state space is *simulated* by a more complete one. Then we know that box-free modal mu-calculus formulas that hold of the partial state space will hold of the complete state space [BBLS92]. A problem with using the simulation relation for this purpose is that it limits the kinds of properties we can check of the partial state space, and does not tell us that if a property fails to hold of a partial state space then it also fails to hold of more complete state spaces.

Our solution to this problem is to use models that capture explicitly the incompleteness of state spaces, and to use 3-valued logics to capture the possibility that we may not know whether a property is true or not in a partial state space. In this approach every formula of the logic can be checked of the partial state space. If the answer true or false is obtained then the answer also holds of the complete state space. If the answer $\perp$ (meaning "unknown") is obtained then the partial state space lacks information needed for a definite answer about the complete state space.

A state-based framework is adopted for most of the paper. In the next section we review Kripke structures and propositional modal logic. In Section 3 we define partial Kripke structures and the interpretation of modal logic on these structures. We also define a preorder on partial Kripke structures that reflects their degree of completeness and show that propositional modal logic (under our 3-valued interpretation) characterizes this preorder. In Section 4 we present a model checker for 3-valued CTL. In Section 5 we show how our results can be applied to the problem of model-checking partial state space. In Section 6 the main results of Section 3 are reworked in an action-based framework. In Section 7 we present our conclusions and discuss related work.

2 Kripke Structures and Modal Logic

Let P be a nonempty finite set of *atomic propositions*.

Definition 1. A Kripke structure M is a tuple $(S, L, \mathcal{R})$, where S is a set of *states*, $L : S \times P \rightarrow \{\mathit{true}, \mathit{false}\}$ is an *interpretation* that associates a truth value in $\{\mathit{true}, \mathit{false}\}$ with each atomic proposition in P for each state in S, and $\mathcal{R} \subseteq S \times S$ is a *transition* relation on S.

For technical convenience, we assume that a Kripke structure has no terminating state by requiring that $\mathcal{R}$ be *total*, i.e., that every state has an outgoing $\mathcal{R}$-transition. This assumption does not restrict the modeling power of the formalism, since we can model a terminated execution as repeating forever its last state by adding a self-loop to that state. Note that Kripke structures can be *nondeterministic*: a state can have more than one outgoing $\mathcal{R}$-transition. We also assume that the number of outgoing transitions from a state is finite.

Temporal logics are modal logics geared towards the description of the temporal ordering of events [Eme90]. Propositional modal logic (e.g, see [Var97]) is propositional logic extended with the modal operator $\Diamond$. Propositional modal logic can itself be extended with a fixpoint operator to form a modal fixpoint logic, also referred to as the propositional μ-calculus [Koz83]. This very expressive logic includes as fragments linear-time temporal logic (LTL) [MP92] and computation-tree logic (CTL) [CE81].

For the sake of simplicity, let us first consider propositional modal logic. More expressive logics will be discussed later. We now recall the syntax and semantics of propositional modal logic.

Definition 2. Given a nonempty finite set P of atomic propositions, formulas of *propositional modal logic* have the following abstract syntax, where p ranges over P:

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

Definition 3. The *satisfaction* of a formula ϕ of propositional modal logic in a state s of a Kripke structure $M = (S, L, \mathcal{R})$, written $(M, s) \models \phi$, is defined inductively as follows:

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

The derived modal operator $\Box$ is the dual of $\Diamond$, i.e., $\neg \Diamond \neg$. Thus, we have $(M, s) \models \Box\, \phi$ if $(M, t) \models \phi$ for all t such that $(s, t) \in \mathcal{R}$. When M is understood, we write $s \models \phi$ instead of $(M, s) \models \phi$.

Propositional modal logic can be used to define an equivalence relation on states of Kripke structures: two states are equivalent if they satisfy the same set of formulas of the logic. It is well known [HM85] that the equivalence relation induced in this way by propositional modal logic coincides with the notion of bisimulation relation [Mil89, Par81] (or more accurately, with that of zig-zag relation [vB84], since propositions are mentioned in the relation).

Definition 4. 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 $s_2' \in S_2$ such that $(s_2, s_2') \in \mathcal{R}_2$ and $(s_1', s_2') \in \mathcal{B}$, and
- if $(s_2, s_2') \in \mathcal{R}_2$, then there is some $s_1' \in S_1$ such that $(s_1, s_1') \in \mathcal{R}_1$ and $(s_1', s_2') \in \mathcal{B}$.

Two states s_1 and s_2 are *bisimilar*, denoted $s_1 \sim s_2$, if they are related by some bisimulation relation.

Theorem 5. *[HM85] 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*

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

Propositional modal logic is then called a *logical characterization* of $\sim$. This means that propositional modal logic cannot distinguish between bisimilar states, and that states satisfying exactly the same set of propositional modal logic formulas are bisimilar.

3 Partial Kripke Structures and 3-Valued Modal Logic

To model check partial state spaces, we need a way to model the absence of information about the missing parts of the full state space, both operationally (in terms of Kripke structures) and logically (in terms of modal logics). A natural approach to the operational modeling of incompleteness is to model an incomplete state space with a kind of partially-defined Kripke structure. We show that a compatible approach in the logical modeling of incompleteness is to interpret modal logic with a third truth value $\perp$, which is understood as "unknown". More precisely, we model partial state spaces as partial Kripke structures. We then define a 3-valued modal logic whose semantics is defined with respect to partial Kripke structures. We proceed by studying an equivalence relation and preorder implicitly defined by this logic. As before, let P be a nonempty finite set of atomic propositions.

Definition 6. A *partial Kripke structure* M is a tuple $(S, L, \mathcal{R})$, where S is a set of *states*, $L : S \times P \to \{\mathit{true}, \perp, \mathit{false}\}$ is an *interpretation* that associates a truth value in $\{\mathit{true}, \perp, \mathit{false}\}$ with each atomic proposition in P for each state in S, and $\mathcal{R} \subseteq S \times S$ is a *transition* relation on S.

In interpreting propositional modal logic on partial Kripke structures, we interpret the operators $\wedge$ and $\neg$ using Kleene's strongest regular 3-valued propositional logic [Kle87]. In this logic $\perp$ is understood as "unknown whether true or false". A simple way to define conjunction (resp. disjunction) in this logic is as the minimum (resp. maximum) of its arguments, under the order $\mathit{false} < \perp < \mathit{true}$. We write min and max for these functions, and extend them to sets in the obvious way, with $\min(\emptyset) = \mathit{true}$ and $\max(\emptyset) = \mathit{false}$. We define negation using the function neg that maps *true* to *false*, *false* to *true*, and $\perp$ to $\perp$. Notice that these functions give the usual meaning of the propositional operators when applied to values *true* and *false*.

We now consider a 3-valued propositional modal logic having the same syntax as propositional modal logic, and the following semantics.

Definition 7. The truth value of a formula ϕ of 3-valued propositional modal logic in a state s of a partial Kripke structure $M = (S, L, \mathcal{R})$, written $[(M, s) \models \phi]$, is defined inductively as follows:

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

We again define $\Box$ as the dual of $\Diamond$, so $[(M,s) \models \Box\phi] = \min(\{[(M,t) \models \phi] \mid (s,t) \in \mathcal{R}\})$. This semantics gives the usual meaning of the propositional and modal operators when applied to complete Kripke structures.

This 3-valued propositional modal logic can be used to define a preorder on partial Kripke structures that reflects their degree of completeness. Let $\leq$ be the ordering on truth values such that $\perp \leq \mathit{true}$, $\perp \leq \mathit{false}$, $x \leq x$ (for all $x \in \{true, \perp, false\}$), and $x \not\leq y$ otherwise. Note that the operators neg, min and max are monotonic with respect to $\leq$: if $x \leq x'$ and $y \leq y'$, we have $\mathrm{neg}(x) \leq \mathrm{neg}(x')$, $\min(x, y) \leq \min(x', y')$, and $\max(x, y) \leq \max(x', y')$. This property is important to prove the results that follow.

Definition 8. 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 $s_2' \in S_2$ such that $(s_2, s_2') \in \mathcal{R}_2$ and $s_1' \preceq s_2'$, and
- if $(s_2, s_2') \in \mathcal{R}_2$, then there is some $s_1' \in S_1$ such that $(s_1, s_1') \in \mathcal{R}_1$ and $s_1' \preceq s_2'$.

Intuitively, $s_1 \preceq s_2$ means that s_1 and s_2 are "nearly bisimilar" except that the atomic propositions in state s_1 may be less defined than in state s_2. Obviously, $s_1 \sim s_2$ implies $s_1 \preceq s_2$.

The following theorem shows how the completeness preorder can be logically characterized with 3-valued propositional modal logic.

Theorem 9. *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*

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

Proof. Proofs of theorems are omitted in this extended abstract because of space constraints.

In other words, partial Kripke structures that are "more complete" with respect to $\preceq$ have more definite properties with respect to $\leq$, i.e., have more properties that are either *true* or *false*. Moreover, any formula ϕ of 3-valued propositional modal logic that evaluates to *true* or *false* on a partial Kripke structure has the same truth value when evaluated on every more complete structure.

Formulas that evaluate to $\perp$ on a partial Kripke structure must be evaluated on a more complete structure to get a definite answer. Obviously, any partial Kripke structure can be completed to obtain a traditional fully-defined Kripke structure, where ϕ always evaluates to either *true* or *false*. Some partial Kripke structures can only be completed to form Kripke structures that all satisfy the property ϕ, or to form Kripke structures that all violate ϕ (this is the case, for instance but not exclusively, when ϕ is a tautology or is unsatisfiable with a 2-valued interpretation). Some other partial Kripke structures can be completed to form Kripke structures that satisfy ϕ as well as Kripke structures that violate ϕ. Note that checking a formula ϕ on a partial Kripke structure may return $\perp$ even if ϕ is a tautology or is unsatisfiable in the 2-valued interpretation.

The following theorem states that 3-valued propositional modal logic logically characterizes the equivalence relation induced by the completeness preorder $\preceq$.

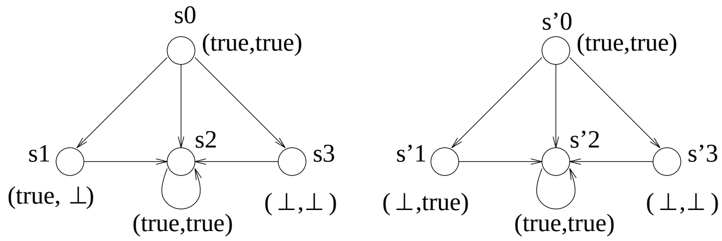
Theorem 10. *Let $M_1 = (S_1, L_1, \mathcal{R}_1)$ and $M_2 = (S_2, L_2, \mathcal{R}_2)$ be partial Kripke structures such that $s_1 \in S_1$ and $s_2 \in S_2$, and let Φ denote the set of all formulas of 3-valued propositional modal logic. Then*

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

The bisimulation relation of Definition 4 can be applied directly to partial Kripke structures. Two states s_1 and s_2 of partial Kripke structures are *bisimilar*, denoted $s_1 \sim s_2$, if they are related by some bisimulation relation. Since $\sim$ is a stronger relation than $\preceq$, $s_1 \sim s_2$ implies both $s_1 \preceq s_2$ and $s_2 \preceq s_1$, and that 3-valued propositional modal logic cannot distinguish between bisimilar states.

However, the converse is not true: $s_1 \preceq s_2$ and $s_2 \preceq s_1$ does not imply $s_1 \sim s_2$. This is illustrated by the example below. The existence of such an example proves that, in contrast with 2-valued propositional modal logic, 3-valued propositional modal logic is *not* a logical characterization of bisimulation as defined in Definition 4.

Example 1. Here is an example of two non-bisimilar states that cannot be distinguished by any formula of 3-valued propositional modal logic.



These two partial Kripke structures have two atomic propositions p and q, whose truth value is defined in each state as indicated in the figure by a pair of the form (p, q). We have the following relations:

- $s_2 \preceq s'_2$ and $s'_2 \preceq s_2$,
- $s_3 \preceq s'_3$ and $s'_3 \preceq s_3$,
- $s_1 \preceq s'_2$ and $s'_3 \preceq s_1$, $s'_1 \preceq s_2$ and $s_3 \preceq s'_1$,
- $s_0 \preceq s'_0$ and $s'_0 \preceq s_0$.

We have that $s_0 \preceq s'_0$ and $s'_0 \preceq s_0$, but $s_0 \not\sim s'_0$ since s_1 is not bisimilar to any state in the second partial Kripke structure.

4 A Model Checker for 3-Valued CTL

We have so far focused on modal propositional logic because it is a simple context in which to present our ideas. However, this logic cannot express even simple safety properties. In this section we present a 3-valued semantics for computation-tree logic (CTL) [CE81] as well as a model-checking algorithm. We consider CTL because it extends the expressiveness of modal propositional logic, the model-checking algorithm for the standard 2-valued interpretation is well known, and because it is expressive enough to specify many interesting properties.

Our algorithm is based on the algorithm of [CES86]. We focus here, as in [CES86], on formulas of the form $A(f_1 \mathcal{U} f_2)$, since the model checking of other CTL formulas is either similar or much simpler. Formula $A(f_1 \mathcal{U} f_2)$ holds of a state in a Kripke structure if, along all paths from the state, there exists a state in the path for which f_2 holds and for which f_1 holds of all previous states in the path.

The semantics of CTL is given as an inductive definition of the satisfaction relation $\models$ between a Kripke structure $M = (S, L, \mathcal{R})$ and a CTL formula. The clause of the definition of $\models$ for formula $A(f_1 \mathcal{U} f_2)$ reads

$$\begin{aligned} s_0 \models A(f_1 \mathcal{U} f_2) \ \text{ iff } \ & \text{for all paths } (s_0, s_1, \ldots), \\ & \exists i \geq 0 : s_i \models f_2 \wedge \forall j : 0 \leq j < i \Rightarrow s_j \models f_1 \end{aligned}$$

(Here the Kripke structure M is understood.) To see if a state satisfies $A(f_1 \mathcal{U} f_2)$, the procedure *au* of the CTL model checker of [CES86] works roughly as follows. It is called with a formula f of the form $A(f_1 \mathcal{U} f_2)$, a state s_0, and a result variable b. It is assumed that when *au* is called a state is *labeled* with f_1 just if it satisfies f_1, and similarly for f_2. A depth-first search of the states reachable from s_0 is then made, with states *marked* as they are visited. Initially, if s_0 is labeled with f_2, then s_0 is labeled with $A(f_1 \mathcal{U} f_2)$ and the procedure terminates with b set to *true*. Otherwise if s_0 is not labeled with f_1, then the procedure terminates with b set to *false*. Otherwise procedure *au* is called recursively on all successors of s_0. If a recursive call is made to a state that is already marked, the procedure terminates with b set to *false*.

For 3-valued CTL we define $A(f_1 \mathcal{U} f_2)$ as follows:

$$\begin{aligned} [s_0 \models A(f_1 \mathcal{U} f_2)] &= \min(\{[(s_0, s_1, \ldots) \models f_1 \mathcal{U} f_2] \mid (s_0, s_1, \ldots) \text{ a path}\}) \\ [(s_0, s_1, \ldots) \models f_1 \mathcal{U} f_2] &= \max(\{[(s_0, s_1, \ldots) \models f_1 \mathcal{U}_k f_2] \mid k \ \geq 0\}) \\ [(s_0, s_1, \ldots) \models f_1 \mathcal{U}_k f_2] &= \min(\min(\{[s_i \models f_1] \mid i < k\}), \{[s_k \models f_2]\}) \end{aligned}$$

The min operators in this definition correspond to conjunction and universal quantification in the definition for 2-valued CTL above, and similarly the max operator in this definition corresponds to existential quantification in the definition above. The two definitions agree on complete Kripke structures.

Consider the problem of adapting the procedure *au* of the CTL model checker of [CES86] to the 3-valued case. One way in which the algorithm becomes more complicated is in checking a state for the first time. Suppose our partial Kripke structure has only a single path, and that the value of formulas f_1 and f_2 for the first three states on the path are as follows:

$$s_0 : (f_1 = \textit{true}, f_2 = \perp), s_1 : (f_1 = \textit{true}, f_2 = \textit{false}), s_2 : (f_1 = \perp, f_2 = \textit{true})$$

We see that f_2 is $\perp$ at s_0, but we cannot conclude immediately that $A(f_1 \mathcal{U} f_2)$ is $\perp$ at s_0 because we may find (and do, in this example) that f_2 is *true* at a later state. However, if f_1 were $\perp$ or *false* at state s_1, then we could conclude that $A(f_1 \mathcal{U} f_2)$ is $\perp$ at s_0.

Determining the result when a cycle is detected also becomes more complicated. In the 2-valued algorithm we know that f_1 holds but f_2 does not along all states in a cycle. In the 3-valued case, for states in a cycle it may be that f_2 is *true* and f_2 is *false*, or f_2 is *true* and f_2 is $\perp$, or f_2 is $\perp$ and f_2 if *false*.

Figure 1 shows our modified version of the procedure *au* of [CES86]. The idea behind the algorithm is to check the partial Kripke structure twice. First check the structure under a *pessimistic* interpretation, in which the value $\perp$ is understood as *false*. If the result of the check is *true* then return *true* as the 3-valued result. Then check the structure under an *optimistic* interpretation, in which the value $\perp$ is understood as *true*. If the result of this check is *false* then return *false*. Otherwise return $\perp$.

Our algorithm merges these two checks so that they can be run at the same time. Procedure *au* now takes an additional argument $mode \subseteq \{p, o\}$ and returns a pair (v_p, v_o). If *mode* contains constant p (resp. o) then value v_p (v_o) is the result of the pessimistic search. If *mode* does not contain p (resp. o) then v_p (v_o) is *false*. When $mode = \{p, o\}$ we interpret the returned pairs (*false*, *false*), (*false*, *true*), and (*true*, *true*) in a 3-valued sense as *false*, $\perp$, and *true*, respectively. Notice that an optimistic check must give *true* if a pessimistic one does, so result (*true*, *false*) is impossible.

In our algorithm states have distinct optimistic and pessimistic labelings. Function $labeled(s, f, i)$ returns *true* just if state s has label $i \in \{p, o\}$ for formula f. Function $add\_label\_i(s, f, i)$ gives state s label i for formula f. Function $label(s, f, mode)$ returns a pair (v_p, v_o) where v_i is *false* if $i \notin mode$ and $labeled(s, f, i)$ otherwise. The operator $\vee$ on pairs of truth values is defined by $(x, y) \vee (u, w) = (x \vee u, y \vee w)$. The order $<$ on pairs of truth values is defined by (*false*, *false*) $<$ (*false*, *true*) $<$ (*true*, *false*) $<$ (*true*, *true*).

States have distinct optimistic and pessimistic markings as well. Function $marked(s)$ returns the set of interpretations for which state s is marked. Function $add\_mark(s, A)$, where $A \subseteq \{p, o\}$, marks s with each interpretation in A.

The proof of correctness of our model-checking algorithm is omitted in this extended abstract. Our modified procedure, like the original one, requires time $O(\mathrm{card}(S) + \mathrm{card}(\mathcal{R}))$, and thus the overall complexity of the resulting 3-valued CTL model-checking algorithm is still $O(\mathrm{length}(\phi) \times (\mathrm{card}(S) + \mathrm{card}(\mathcal{R})))$.

5 Applications

We now discuss how to exploit the results of the previous sections in practice. Consider a (possibly infinite) complete state space modeled as a Kripke structure $M = (S, L, \mathcal{R})$. Imagine that this state space is so large that only part of it can be explored. We now present a simple construction to define a partial Kripke structure $M' = (S', L', \mathcal{R}')$ representing only the explored states and transitions of this state space.

Let $S_E \subseteq S$ be the set of explored states and $\mathcal{R}_E \subseteq \mathcal{R}$ be the set of explored transitions. For this application, we can assign the value $\perp$ to all the atomic propositions in each unexplored state $s \in S \setminus S_E$. Since unexplored states are indistinguishable with this model, a single state $s_\perp$ of M' is enough to model all of them. For every unexplored transition $(s, t) \in \mathcal{R} \setminus \mathcal{R}_E$ such that $s \in S_E$,

```
procedure au(f,s,b,mode)
begin
    parent_mode := mode;
    mode := mode - marked(s);

    add_mark(s,mode);
    temp_mode := mode;
    for all i in temp_mode do
        begin
            if labeled(s,f2,i) then
                begin add_label_i(s,f,i); mode := mode - {i} end;
            else if ¬labeled(s,f1,i) then
                mode := mode - {i}
        end;
    if mode = ∅ then
        begin b := label(s,f,parent_mode); return end;

    push(s,ST);
    min := (true, true);
    for all s1 ∈ successors(s) do
        begin
            au(f,s1,b1,mode);
            if b1 < min then min := b1;
            if min = (false, false) then break
        end;

    pop(ST);
    b := min ∨ label(s,f,parent_mode - mode);
    add_label(s,f,b);
    return
end
```

Fig. 1. Procedure *au* of model-checking algorithm for 3-valued CTL

we add a transition $(s, s_\perp)$ in $\mathcal{R}'$ to model that we do not know where this unexplored transition leads. To preserve our assumption that $\mathcal{R}'$ is total, we also assume there is a transition $(s_\perp, s_\perp)$ in $\mathcal{R}'$. However, this is the only outgoing $\mathcal{R}'$-transition of $s_\perp$, modeling that unexplored states cannot lead back to explored states. In summary, we have the following:

- $S' = S_E \cup \{s_\perp\}$
- $L'(s,p) = \begin{cases} L(s,p) & \text{if } s \in S_E \\ \perp & \text{if } s = s_\perp \end{cases}$
- $\mathcal{R}' = \mathcal{R}_E \cup \{(s, s_\perp) \mid s \in S_E \text{ and } (s,t) \in \mathcal{R} \setminus \mathcal{R}_E\} \cup \{(s_\perp, s_\perp)\}$

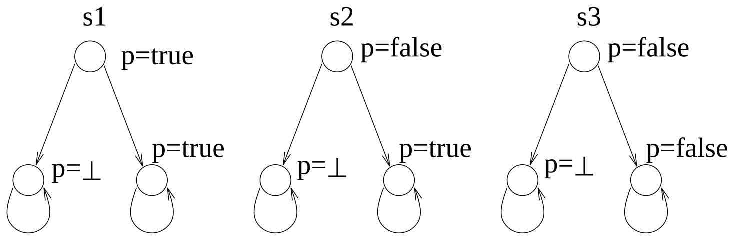
Let us assume that M has initial state s_0, and that s_0 is explored and denoted by s_0' in S'. It is easy to prove the following.

Theorem 11. *Let Kripke structure $M = (S, L, \mathcal{R})$ with initial state s_0 represent a complete state space, and let $M' = (S', L', \mathcal{R}')$ be a partial Kripke structure built from M by the construction above. Then*

$$s'_0 \preceq s_0.$$

Theorems 9 and 11 together guarantee that any formula ϕ of 3-valued propositional modal logic that evaluates to *true* or *false* on a partial state space defined with the construction above has the same truth value when evaluated on the corresponding complete state space.

Example 2. Consider the three following partial Kripke structures with a single atomic proposition p, whose truth value is defined in each state as indicated in the figure.



The formula $A(true\,\mathcal{U}\,p)$ of 3-valued CTL has a different truth value in each of the top states of these partial Kripke structures: $[s_1 \models A(true\,\mathcal{U}\,p)] = true$, $[s_2 \models A(true\,\mathcal{U}\,p)] = \perp$, and $[s_3 \models A(true\,\mathcal{U}\,p)] = false$.

An important application of our framework is thus to make it possible to cope with missing parts of the state space during model checking and still obtain a definite answer when this is possible. In the case of CTL properties, the algorithm of the previous section captures exactly when this is possible and how do it, for any CTL formula and any partial Kripke structure.

Other possible applications for our framework include the evaluation of heuristics for guiding the search and pruning state spaces (one can determine which heuristics more often give definite answers for which properties), and the analysis of systems containing state variables whose values cannot be read (and hence are unknown) at some points during the execution of the system.

6 An Action-Based Approach to Partial State Spaces

In this section we revisit the main results of Section 3 in an *action-based* framework, where system behavior is modeled as a labeled transition system rather than a Kripke structure. Here the focus is how a system responds to events (or *actions*), which are modeled as transition labels, rather than on the propositions that hold of system states.

To capture the incompleteness of a state space, a transition system can be labeled with a divergence predicate $\uparrow$ [Mil81]. If the divergence predicate holds for label a at state p then intuitively some of the a transitions from p in the full state space may be missing at p in the partial state space. Note that this predicate takes a value of either *true* or *false* for each state and label, while the atomic propositions of partial Kripke structures are 3-valued.

Definition 12. An *extended transition system* is a structure $(S, A, \{\xrightarrow{a} \mid a \in A\}, \uparrow)$ where S is a set of states, A is a set of labels, $\{\xrightarrow{a} \mid a \in A\}$ is a family of transition relations, and $\uparrow \subseteq S \times A$ is a *divergence* relation.

We write $p \xrightarrow{a} q$ if $(p, q) \in \xrightarrow{a}$. We write $p \uparrow a$ if $(p, a) \in \uparrow$ and say that p *diverges* for a. Also, we write $p \downarrow a$ if not $p \uparrow a$ and say that p *converges* for a.

The degree to which a state space is complete is modeled here by the *divergence preorder* [Mil81, Wal90] (also known as the *partial bisimulation preorder*), which is a generalization of the simulation and bisimulation relations. If a pair (p, q) is in this relation, then q must be able to match every transition of p. Furthermore, if p is convergent for label a, then p must be able to match every transition of q.

Definition 13. The *divergence preorder* $\sqsubseteq$ is the greatest binary relation on states of an extended transition system such that $p \sqsubseteq q$ implies:

- whenever $p \xrightarrow{a} p'$ there exists a q' such that $q \xrightarrow{a} q'$ and $p' \sqsubseteq q'$, and
- if $p \downarrow a$, then $q \downarrow a$ and whenever $q \xrightarrow{a} q'$ there exists a p' such that $p \xrightarrow{a} p'$ and $p' \sqsubseteq q'$.

This preorder differs from the completeness preorder of Section 3 because the divergence predicate specifically captures the possibility that transitions are missing at a state, while in partial Kripke structures an atomic proposition with value $\perp$ may not represent this possibility.

Hennessy-Milner Logic (HML) [HM85] is a propositional modal logic for labeled transition systems. Formulas of HML have the following abstract syntax:

$$\phi ::= \mathtt{tt} \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \langle a \rangle \phi$$

where a ranges over A. We use the standard propositional abbreviations, including $\mathtt{ff}$ and $\vee$, plus the derived modal operator $[a]$, defined by $[a]\phi = \neg\langle a \rangle \neg\phi$.

We give the following 3-valued interpretation of HML formulas. The truth value of an HML formula ϕ for a state p, written $[p \models \phi]$, is defined inductively as follows:

$$\begin{aligned}
[p \models \mathtt{tt}] &= \mathit{true} \\
[p \models \neg\phi] &= \mathrm{neg}([p \models \phi]) \\
[p \models \phi_1 \wedge \phi_2] &= \min([p \models \phi_1], [p \models \phi_2]) \\
[p \models \langle a \rangle \phi] &= \begin{cases} \max(\{[p' \models \phi] \mid p \xrightarrow{a} p'\}) & \text{if } p \downarrow a \\ \max(\{[p' \models \phi] \mid p \xrightarrow{a} p'\} \cup \{\perp\}) & \text{otherwise} \end{cases}
\end{aligned}$$

HML, under our 3-valued interpretation, characterizes the divergence preorder on extended transition systems.

Theorem 14. *Let* $(S, A, \{\xrightarrow{a} \mid a \in A\}, \uparrow)$ *be an extended transition system such that the set* $\{p' \mid p \xrightarrow{a} p'\}$ *is finite for all* p *in* S *and* a *in* A. *Let* p *and* q *be states in* S. *Then*

$$(\forall \phi : [p \models \phi] \ \leq \ [q \models \phi]) \textit{ iff } p \sqsubseteq q.$$

Our 3-valued interpretation of HML has a close connection to the intuitionistic interpretation by Plotkin. In [Sti87] Plotkin's interpretation is presented and it is shown that the logic characterizes the divergence preorder. A positive form of HML is used there, with syntax

$$\phi ::= \texttt{tt} \mid \texttt{ff} \mid \phi_1 \ \wedge \ \phi_2 \mid \phi_1 \ \vee \ \phi_2 \mid \langle a \rangle \phi \mid [a]\phi$$

Negation is not present, but the complementary forms of $\texttt{tt}$, $\wedge$, and $\langle a \rangle$ are included. The intuitionistic semantics of this logic is like that of standard 2-valued HML, except for the $[a]$ operator. The intuitionistic semantics of the two modal operators are:

$$p \models_I \langle a \rangle \phi \text{ if } \exists p' : p \xrightarrow{a} p' \text{ and } p' \models_I \phi$$
$$p \models_I [a]\phi \text{ if } p \downarrow a \text{ and } \forall p' : p \xrightarrow{a} p' \Rightarrow p' \models_I \phi$$

These two operators are no longer duals, unlike the standard 2-valued interpretation and our 3-valued interpretation. For example, if process p has no a transitions and $p \uparrow a$ then $p \not\models [a]\,\texttt{ff}$ and $p \not\models \langle a \rangle\, \texttt{tt}$.

The precise connection between this interpretation and our 3-valued interpretation is as follows. We define the syntactic complement $\mathrm{comp}(\phi)$ of a positive HML formula ϕ as follows: $\mathrm{comp}(\texttt{tt}) = \texttt{ff}$, $\mathrm{comp}(\texttt{ff}) = \texttt{tt}$, $\mathrm{comp}(\phi_1 \wedge \phi_2) = \mathrm{comp}(\phi_1) \vee \mathrm{comp}(\phi_2)$, $\mathrm{comp}(\phi_1 \vee \phi_2) = \mathrm{comp}(\phi_1) \wedge \mathrm{comp}(\phi_2)$, $\mathrm{comp}(\langle a \rangle \phi) = [a]\mathrm{comp}(\phi)$, and $\mathrm{comp}([a]\phi) = \langle a \rangle \mathrm{comp}(\phi)$. Then our 3-valued interpretation gives the result $\perp$ for p and ϕ just if both ϕ and $\mathrm{comp}(\phi)$ fail to hold for p.

Theorem 15. *Let* ϕ *be a formula of positive HML and let* p *be a state of an extended transition system. Then the following all hold:*

1. $[p \models \phi] = \textit{true}$ *iff* $p \models_I \phi$
2. $[p \models \phi] = \textit{false}$ *iff* $p \models_I \textit{comp}(\phi)$
3. $[p \models \phi] = \perp$ *iff* $p \not\models_I \phi$ *and* $p \not\models_I \textit{comp}(\phi)$

In [Sti87] the divergence preorder is characterized by intuitionistic HML as follows. Let p and q be processes in extended transition systems. Then $p \sqsubseteq q$ just if, for all ϕ of positive HML, $p \models_I \phi \Rightarrow q \models_I \phi$. From Theorem 15 the equivalent condition in 3-valued HML is $([p \models \phi] = \textit{true}) \Rightarrow ([p \models \phi] = \textit{true})$. Clearly $\forall \phi : ([p \models \phi] = \textit{true}) \Rightarrow ([q \models \phi] = \textit{true})$ is equivalent to the condition $\forall \phi : [p \models \phi] \leq [q \models \phi]$ used in Theorem 14 above. Thus, in this action-based framework, we could have defined 3-valued HML in terms of intuitionistic HML, and then derived our characterization result from the characterization result of [Sti87].

An advantage of 3-valued modal logic over intuitionistic modal logic is that it more naturally captures the problem of model-checking partial state spaces. For example, a 3-valued modal logic leads directly to a model checker that, given a state and a formula, returns *true*, *false*, or $\perp$. In contrast, a model checker directly based on intuitionistic modal logic would return either *true* or *false*. The value $\perp$ could only be inferred from the results of multiple checks.

7 Conclusions

We developed a simple framework for reasoning about partially-known behaviors of a system. We showed that the use of 3-valued temporal logics nicely models the absence of information about unknown parts of the state space of a system. We then precisely determined, both operationally and logically, the relationship between a partial state space and a more complete one. We also presented a model-checking algorithm for 3-valued CTL. This model checker can check any CTL formula on any partial state space, and returns either a definite answer of *true* or *false* concerning the full state space, or $\perp$ ("I don't know") if the partial state space lacks information needed for a definite answer.

We also compared our results on partial Kripke structures with existing work on extended transition systems. In the latter framework, we showed that Hennessy-Milner Logic with our 3-valued interpretation provides an alternative characterization of the divergence preorder in addition to the intuitionistic interpretation of Plotkin. Further work on divergence preorders and logics to characterize them can be found in [Sti87, Wal90]. Verification techniques based on the divergence preorder are described in [Wal90, CS90]. In all this work logical formulas are interpreted normally in the 2-valued sense. To our knowledge none of the work on 3-valued modal logics (e.g., [Seg67, Mor89, Fit92]) shows how these logics can be used to characterize relations like our completeness preorder.

The model-checking framework developed in this paper could be extended so it can be performed "symbolically" following the ideas of [BCM+90]. This would require the use of data structures and algorithms for representing and manipulating 3-valued formulas, such as Ternary Decision Diagrams [Sas97].

Acknowledgments

We thank Michael Benedikt and the anonymous referees for helpful comments on this paper.

References

[BBLS92] S. Bensalem, A. Bouajjani, C. Loiseaux, and J. Sifakis. Property preserving simulations. In *Proceedings of CAV '92, LNCS 663*, pages 260–273, 1992.

[BCM+90] J.R. Burch, E.M. Clarke, K.L. McMillan, D.L. Dill, and L.J. Hwang. Symbolic model checking: 10^{20} states and beyond. In *Proceedings of the 5th Symposium on Logic in Computer Science*, pages 428–439, Philadelphia, June 1990.

[CE81] E. M. Clarke and E. A. Emerson. Design and Synthesis of Synchronization Skeletons using Branching-Time Temporal Logic. In D. Kozen, editor, *Proceedings of the Workshop on Logic of Programs,* Yorktown Heights, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer-Verlag, 1981.

[CES86] E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, January 1986.

[CS90] Rance Cleaveland and Bernhard Steffen. When is "partial" adequate? A logic-based proof technique using partial specifications. In *Proceedings of the 5th Annual Symposium on Logic in Computer Science.* IEEE Computer Society Press, 1990.

[Eme90] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science.* Elsevier/MIT Press, Amsterdam/Cambridge, 1990.

[Fit92] Melvin Fitting. Many-valued modal logics II. *Fundamenta Informaticae*, 17:55–73, 1992.

[HM85] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161, 1985.

[Kle87] Stephen Cole Kleene. *Introduction to Metamathematics.* North Holland, 1987.

[Koz83] D. Kozen. Results on the Propositional Mu-Calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[Mil81] R. Milner. A Modal Characterization of Observable Machine Behavior. In *Proc. CAAP'81*, volume 112 of *Lecture Notes in Computer Science*, pages 25–34. Springer-Verlag, 1981.

[Mil89] R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.

[Mor89] Osamu Morikawa. Some modal logics based on a three-valued logic. *Notre Dame Journal of Formal Logic*, 30(1):130–137, 1989.

[MP92] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification.* Springer-Verlag, 1992.

[Par81] D. M. R. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, 5^{th} *GI Conference*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, 1981.

[Sas97] T. Sasao. Ternary Decision Diagrams – A Survey. In *Proc. IEEE International Symposium on Multiple-Valued Logic*, pages 241–250, Nova Scotia, May 1997.

[Seg67] Krister Segerberg. Some modal logics based on a three-valued logic. *Theoria*, 33:53–71, 1967.

[Sti87] Colin Stirling. Modal logics for communicating systems. *Theoretical Computer Science*, 49:331–347, 1987.

[Var97] M.Y. Vardi. Why is modal logic so robustly decidable? In *Proceedings of DIMACS Workshop on Descriptive Complexity and Finite Models.* AMS, 1997.

[vB84] J. van Bentham. Correspondence theory. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic Vol. II.* Reidel, 1984.

[Wal90] D. J. Walker. Bisimulation and divergence. *Information and Computation*, 85(2):202–241, 1990.

This article was processed using the LaTeX macro package with LLNCS style