Available online at www.sciencedirect.com
SCIENCE DIRECT®

Electronic Notes in Theoretical Computer Science

ELSEVIER Electronic Notes in Theoretical Computer Science 112 (2005) 61–76
www.elsevier.com/locate/entcs

Abstraction and Probabilities for Hybrid Logics

Michael Huth[1]

Department of Computing
Imperial College London
London, United Kingdom

Abstract

We suggest and develop mathematical foundations for quantitative versions of hybrid logics by means of two related themes: a relational abstraction technique for hybrid computation tree logic and hybrid Kripke structures as an extension of the model-checking framework for computation tree logic with the ability to name, bind, and retrieve states; and a syntax and semantics for hybrid probabilistic computation tree logic over hybrid extensions of labelled Markov chains for which the relational abstraction techniques of hybrid Kripke structures should be transferable.

Keywords: hybrid logic, model checking, probabilistic system, abstraction.

1 Introduction

Hybrid logics (see e.g. http://www.hylo.net enhance basic modal and temporal logics with the ability to bind names to unique states in models. This extension is an important ability in applications that have to track states or other objects across space or time. If we think of a hybrid logic as a temporal logic enriched with syntactic clauses for the look-up and binding of names, it is natural to ask whether established model-checking methodology can be adapted to, or retained, in this hybrid setting. Apart from the work by Franceschet & Rijke [10], surprisingly little attention has been given to the extension of model checking to hybrid temporal logics. We are also not aware

[1] Email: M.Huth@doc.imperial.ac.uk

doi:10.1016/j.entcs.2004.01.023

of any work on hybrid logics over quantitative or probabilistic models. Note that this paper only discusses *propositional* temporal logics.

This paper therefore provides a modest first step in this direction by developing two model-checking themes for a hybrid extension of computation tree logic [4] : the sound relational abstraction of qualitative models with respect to *all* properties of a hybrid computation tree logic; and the extension of probabilistic systems and probabilistic computation tree logic [12] with hybrid constructs. The connection between these themes is twofold:

(i) probabilities can be seen as a form of abstraction of qualitative information, reducing the determinism of a system [2] ; and

(ii) the techniques for relational abstraction of qualitative systems are expected to be transferable to probabilistic hybrid systems in future work.

2 Hybrid computation tree logic

We define a hybrid version of computation tree logic [4] and its models.

Definition 2.1 (i) *A* Kripke structure *with signature* Obs *is a tuple* $M = (\Sigma, R \subseteq \Sigma \times \Sigma, L\colon \mathsf{Obs} \to \mathbb{P}(\Sigma))$ *where* Obs *is a set of* atomic observables*.*

(ii) *A* hybrid *Kripke structure with signature* $\mathsf{Obs} = \mathsf{AP} + \mathsf{Nom}$ *is a tuple* $M = (\Sigma, R \subseteq \Sigma \times \Sigma, L\colon \mathsf{Obs} \to \mathbb{P}(\Sigma))$*, where* AP *and* Nom *are disjoint sets of* atomic propositions *and* nominals*, respectively, such that for all* $n \in \mathsf{Nom}$ *the set* $L(n)$ *contains exactly one element.*

(iii) *We write* (M, i) *to denote that state* i *of* M *is the* initial state *of* M*.*

A hybrid Kripke structure consists of a set of states Σ, a state transition relation R, and a labelling function L where, for each observable $o \in \mathsf{Obs}$, $L(o)$ denotes the set of states in Σ at which o holds; see Figure 1. These models are not merely Kripke structures due to the constraints on L: all nominals $n \in \mathsf{Nom}$ hold at *exactly one* state of the model, whereas atomic propositions $p \in \mathsf{AP}$ may hold at no, exactly one, or more than one state. In this paper, we present a hybrid extension of computation tree logic for specifications of properties as this prepares the ground for a hybrid extension of probabilistic computation tree logic [12], but Theorem 3.6 of this paper adapts to the full propositional mu-calculus [15].

For a signature $\mathsf{Obs} = \mathsf{AP} + \mathsf{Nom}$, an adequate fragment of computation tree logic is

$$\phi ::= \bot \mid o \mid \neg\phi \mid \phi \wedge \phi \mid \mathtt{EX}\,\phi \mid \mathtt{E}[\phi\,\mathtt{U}\,\phi] \mid \mathtt{AF}\,\phi \tag{1}$$

[2] At the same time, probabilities may be seen as concretizations of a “zero-one” nondeterminism.

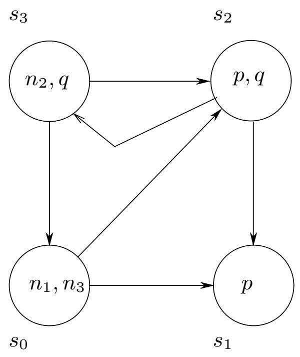

Fig. 1. A hybrid Kripke structure M with signature $\mathsf{Obs} = \{p, q\} + \{n_1, n_2, n_3\}$. A state s is tagged with o iff $s \in L(o)$. In that case, we also write $(M, s) \models o$.

where $o \in \mathsf{Obs}$. The temporal patterns $\mathtt{EX}\,\phi$, $\mathtt{E}[\phi_1\,\mathtt{U}\,\phi_2]$, and $\mathtt{AF}\,\phi$ express "At some next state ϕ," and "On some path ϕ_1 until ϕ_2," and "For all paths, eventually ϕ," respectively. Every hybrid Kripke structure M is also a Kripke structure if we "forget" the constraints on the labelling function. So the satisfaction relation $(M, s) \models \phi$ is the familiar one for Kripke structures (e.g. [7]). As usual, we write $\phi \vee \psi$ for $\neg(\neg\phi \wedge \neg\psi)$, and $\phi \to \psi$ for $\neg(\phi \wedge \neg\psi)$. Moving from Kripke structures to hybrid Kripke structures restricts the class of models and so changes the notions of satisfiability and validity. We discuss two standard examples from the literature.

Example 2.2 (i) *For the computation tree logic formula*

$$\mathtt{EX}\,(n \wedge p) \wedge \mathtt{EX}\,(n \wedge q) \to \mathtt{EX}\,(p \wedge q) \tag{2}$$

we may think of n, p, and q as atomic propositions that can be true at no, one, or more states. Then we can easily find a state in a Kripke structure where this formula is false. If we think of n as being a nominal in a hybrid Kripke structure, the formula is valid. For if the premise is true, then the unique successor state s named by n (i.e. $L(n) = \{s\}$) satisfies p and satisfies q, so there is a successor state satisfying $p \wedge q$.

(ii) *Using nominals, one also gets a richer correspondence theory between formula and properties of the transition relation. The formula $n \to \neg\mathtt{EX}\,n$, interpreted over nominals and Kripke frames*[3] *only, expresses that the transition relation R is irreflexive; it is known that this property cannot be expressed within modal logic over Kripke frames.*

The analysis of hybrid models benefits from enhancing computation tree logic with standard hybrid operators, which we present here for a branching-

[3] A Kripke frame $F = (\Sigma, R)$ is like a Kripke structure $M = (\Sigma, R, L)$ except that we are not in control of choosing the labelling function L, so $(\Sigma, R) \models \phi$ iff *for all* L, $(\Sigma, R, L) \models \phi$.

time logic *CTL*. Let *CTL*(@) be the extension of computation tree logic with the satisfaction operator @

$$\phi ::= \bot \mid o \mid \neg\phi \mid @_n\,\phi \mid \phi \wedge \phi \mid \mathtt{EX}\,\phi \mid \mathtt{E}[\phi\,\mathtt{U}\,\phi] \mid \mathtt{AF}\,\phi \tag{3}$$

where $o \in \mathsf{Obs}$ and $n \in \mathsf{Nom}$. The intended meaning of $@_n\,\phi$ is to "jump" to the unique state $s' \in L(n)$ and evaluate ϕ in that state:

$$(M, s) \models @_n\,\phi \qquad \text{iff} \qquad (M, s') \models \phi \text{ for } L(n) = \{s'\}\,. \tag{4}$$

Note that $(M, s) \models @_n\,\phi$ either holds in all states of M or in none. This operator is self-dual: $@_n\,\phi$ and $\neg @_n\,\neg\phi$ are semantically equivalent over hybrid Kripke structures.

In a hybrid Kripke structure, the labelling function L binds all nominals to a unique state. Viewing nominals as parameters, we can bind them to unique states for the evaluation of formulas. Consider *CTL*(↓) which adds to computation tree logic the operator $\downarrow n.\phi$, whose semantics requires tagging $\models$ with the labelling function L of the underlying hybrid Kripke structure. For computation tree logic or *CTL*(@), the evaluation of $(M, s) \models_L \phi$ does not change L. For *CTL*(↓) the labelling function L changes for the evaluation of clauses of the form $\downarrow n.\phi$.

Definition 2.3 *Let $L[n \mapsto s]$ be the labelling function with $L[n \mapsto s](o) = L(o)$ for all $o \in \mathsf{Obs}$ with $o \neq n$ and $L[n \mapsto s](n) = \{s\}$. Then we set*

$$(M, s) \models_L \downarrow n.\phi \qquad \textit{iff} \qquad (M, s) \models_{L[n \mapsto s]} \phi\,. \tag{5}$$

We conclude that model checks for *CTL*(↓) over the hybrid model M are checks $(M, s) \models_L \phi$ with the initial labelling function of M, but where the evaluation of checks for sub-formulas of the form $\downarrow n.\psi$ updates L statically.

In hybrid logic, the binder $\downarrow n.\phi$ allows one to express that a state s belongs to a cycle (a property *not* expressible in temporal logic) by checking

$$(M, s) \models_L \downarrow n.\mathtt{E}[\neg\bot\,\mathtt{U}\,n]\,. \tag{6}$$

If we think of the labelling algorithm for model checking as an abstract machine, then $@_n\,\phi$ corresponds to a lookup of "location" n with a continuation that jumps to that located state and evaluates ϕ at that location, whereas $\downarrow n.\phi$ stores the current location at n and continues with the evaluation of ϕ at the current state.

Finally, consider *CTL*(∃) which adds a binder for locations that seems contrary to the locality principle inherent in Kripke's satisfaction relation $\models$:

$$(M, s) \models_L \exists n.\phi \qquad \text{iff} \qquad \text{for some } s' \in \Sigma\text{: } (M, s) \models_{L[n \mapsto s']} \phi\,. \tag{7}$$

The lack of locality of this operator means that no purely bottom-up labelling algorithm for model checking is available. For example, the check $(M, s) \models \exists n.@_n\,\mathtt{E}[\neg\bot\,\mathtt{U}\,n]$ holds iff the model M contains *some* cycle, not necessarily through s; similar problems emerge in a bottom-up evaluation of $\downarrow n.\phi$. In the sequel, we write $CTL(@, \downarrow)$ etc for extensions of CTL with all listed operators.

Example 2.4 *In the hybrid Kripke structure in Figure 1, the check* $(M, s_0) \models_L @_{n_2} \mathtt{EX}\, \neg p$ *holds since* $(M, s_3) \models_L \mathtt{EX}\, \neg p$. *The check* $(M, s_0) \models_L \downarrow n_2 . \neg \mathtt{EX}\, n_2$ *holds since* $(M, s_0) \models_{L[n_2 \mapsto s_0]} \neg \mathtt{EX}\, n_2$, *which holds as* $(s_0, s_0) \notin R$. *The check* $(M, s_0) \models_L \exists n_1 . @_{n_1} \neg p \wedge \mathtt{EX}\, p$ *holds as, e.g.,* $(M, s_0) \models_{L[n_1 \mapsto s_3]} @_{n_1} \neg p \wedge \mathtt{EX}\, p$.

3 Relational abstraction of hybrid models

The state-explosion problem of model checking, that the size of the state space of a model is typically exponential in the number of atomic propositions, poses a significant challenge to the application of model checking to realistic and scalable problems [7]. This is exacerbated by the fact that the addition of the operators $\downarrow$ or $\exists$ to computation tree logic make the model checking problem PSPACE-complete, although the addition of nominals and @ alone does not change the linear complexity of checks in the size of the model [10].

Abstraction is seen as a key technique for mitigating the effect of state-space explosions. Its standard approach [6] abstracts a model via a "safe simulation" such that formulas of linear-time temporal logic or the universal fragment of computation tree logic ("for all paths") which are true in the abstract model are also true in the concrete one. Counter-examples of the abstract model, however, often are spurious in that they do not reflect genuine bugs in the concrete model.

Three-valued model checking [8,2] abstracts concrete models by a "mix" of safe and live simulations such that verifications ("the property holds") *and* refutations ("the property does not hold") of properties on the abstract model apply to the concrete one as well, for temporal logics with unrestricted use of path quantifiers or negation. The price being paid here is that model checks may have a third result value "unknown" which does not reveal anything about the abstract [4] or concrete model. But we gain the ability to freely combine such path quantifiers, e.g. as in the verification of a safety property on an abstraction (obtained by the "addition" of paths) by appeal to fairness assumptions (obtained by the "removal" of paths).

In this section we work with a hybrid Kripke structure $M = (\Sigma, R, L)$ with signature $\mathsf{Obs} = \mathsf{AP} + \mathsf{Nom}$, a set of designated abstract states $\hat{\Sigma}$, and a relation $\rho \subseteq \Sigma \times \hat{\Sigma}$ where $s\rho t$ specifies that state t abstracts s (and, equivalently, that s is a concrete instance of t). We wish to define a hybrid model $\hat{M} = (\hat{\Sigma}, \hat{R}, \hat{L})$

[4] In Bruns & Godefroid's *generalized* model checking [3] "unknown" reveals that some concretizations of the abstraction do, and some don't, satisfy the property.

such that ρ is, by construction, a witness to the fact that $\hat{M}$ abstracts M. For that, we assume that ρ is *left-total* and *right-total* (respectively):

(8) $\forall s \in \Sigma\, \exists t \in \hat{\Sigma}\colon s\rho t$
$\forall t \in \hat{\Sigma}\, \exists s \in \Sigma\colon s\rho t$

A practically relevant example is $\hat{\Sigma}$ being the set of classes of some partition on Σ, and $s\rho t$ stating $s \in t$. Such partitions could be induced by a finite set of formulas (e.g. boolean guards from program code) on the concrete state space. The abstract structure $\hat{M}$ should satisfy that all verifications, $(\hat{M}, t) \models \phi$, and all refutations, $(\hat{M}, t) \models \neg\phi$, of ϕ in the abstract model apply in the abstracted model (M, s) as well:

(9) $\forall \phi \in CTL(@, \downarrow, \exists)\, \forall (s,t) \in \Sigma \times \hat{\Sigma}\colon s\rho t \;\&\; (\hat{M}, t) \models \phi \;\Rightarrow\; (M, s) \models \phi\,.$

The abstraction introduced by the relation ρ relaxes the constraints of hybrid logic and presents them in a 3-valued version.

Definition 3.1 *A* 3-valued *hybrid Kripke structure with signature* $\mathsf{Obs} = \mathsf{AP}+\mathsf{Nom}$ *is a tuple* $M = (\Sigma, R^a, R^c, L^a, L^c)$ *where* (Σ, R^a, L^a) *and* (Σ, R^c, L^c) *are Kripke structures with signature* Obs *subject to the following constraints:*

(i) $R^a \subseteq R^c$*;*
(ii) *for all* $o \in \mathsf{Obs}$, $L^a(o) \subseteq L^c(o)$*; and*
(iii) *for all* $n \in \mathsf{Nom}$, $L^a(n)$ *contains at most one element; if so* $L^a(n) = L^c(n)$*.*

The intuition about R^a and L^a, already expressed for labelled transition systems by Larsen & Thomsen in [17], is that they represent "must"-information ("definite," "necessarily so" etc), whereas $R^c \setminus R^a$ and $L^c(o) \setminus L^a(o)$ denote "may"-information ("possibly," "could be so" etc). If $L^a(n)$ is non-empty, we force $L^a(n) = L^c(n)$ since no element of $L^c(n) \setminus L^a(n)$ can have a refinement different from the one for $s \in L^a(n)$, in the sense of Definition 3.4 below.

This interpretation of "may"- and "must"-information confirms that we can view a hybrid Kripke structure $M = (\Sigma, R, L)$ as the 3-valued hybrid Kripke structure (Σ, R, R, L, L). Therefore, we may define abstractions on 3-valued hybrid Kripke structure in general, allowing for an incremental abstract-and-refine methodology of 3-valued model checking as in [11].

Definition 3.2 *For a 3-valued hybrid Kripke structure* $A = (\Sigma, R^a, R^c, L^a, L^c)$ *with signature* $\mathsf{Obs} = \mathsf{AP} + \mathsf{Nom}$*, a set* $\hat{\Sigma}$*, and a left-total and right-total relation* $\rho \subseteq \Sigma \times \hat{\Sigma}$ *we define a tuple* $\hat{A} = (\hat{\Sigma}, \hat{R}^a, \hat{R}^c, \hat{L}^a, \hat{L}^c)$*:*

- $(t, t') \in \hat{R}^a$ *iff for all* $s\rho t$ *there is some* $s'\rho t'$ *with* $(s, s') \in R^a$*;*
- $(t, t') \in \hat{R}^c$ *iff for some* $(s, s') \in R^c$ *we have* $s\rho t$ *and* $s'\rho t'$*;*

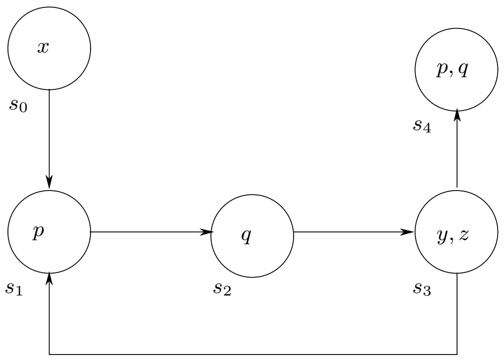

Fig. 2. A shape graph. Nodes are cells in a heap. The set of nominals consists of those program identifiers x, y, and z that do point to a cell in the heap. As no identifier can point to more than one cell at a time we have a hybrid heap model.

- *$t \in \hat{L}^a(o)$ iff for all $s\rho t$ we have $s \in L^a(o)$; and*
- *$t \in \hat{L}^c(o)$ iff for some $s\rho t$ we have $s \in L^c(o)$.*

Example 3.3 *The 3-valued hybrid Kripke structure $\hat{A}$ of Figure 3 is obtained in this manner from the hybrid Kripke structure A in Figure 2. To see this, we set*

$$\rho = \{(s_0, t_0), (s_1, t_1), (s_2, t_2), (s_3, t_2), (s_4, t_2)\} .$$

Then ρ is left-total and right-total. In $\hat{A}$, the two transitions (s_0, s_1) and (s_1, s_2) are modelled as solid lines since t_i is only related to s_i for $i = 1, 2$; for the same reason, their labels x and p are preserved as "must"-information in t_0 and t_1, respectively. There is a dashed line from t_2 to t_1 because (i) there is a transition (s_3, s_1), $s_3\rho t_2$, and $s_1\rho t_1$; and (ii) $s_2\rho t_2$ but there is no transition out of s_2 to some s with $s\rho t_1$. Similarly, we account for the dashed transition from t_2 back to itself. No labels at t_2 are "must"-information and all but x are "may"-information. For example, for y this is so since s_3 satisfies y but s_4 doesn't and both are related to t_2 via ρ.

This example suggests that hybrid models and logics can express shape graphs [19]. Note the definitions of $\hat{L}^a$ and $\hat{L}^c$ for nominals: If A is a hybrid Kripke structure, then $\hat{L}^a(n) = \{t\}$ iff $\{s \in \Sigma \mid s\rho t\} = L^a(n)$; and $\hat{L}^c(n)$ contains all those t for which $s_o\rho t$ where $s_0 \in L^c(n)$. So if we want to verify ϕ on an abstraction and $N \subseteq \mathsf{Nom}$ is the set of nominals occurring under an even number of negations in ϕ, then $s_n \in L(n)$ has to be abstracted as a singleton for each $n \in N$. This won't corrupt the reduction of the size of the state space as N will be very small compared to Σ, e.g. 7 versus 10^7.

Before we can show that the abstraction $\hat{A}$ of A in Definition 3.2 secures (9) we need to present the satisfaction relation $\models$ for a 3-valued hybrid

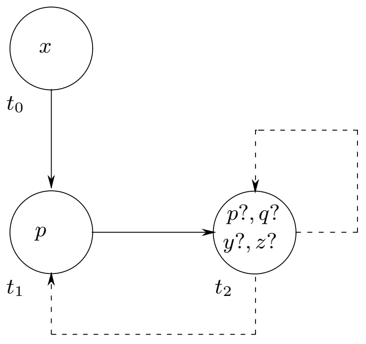

Fig. 3. A 3-valued hybrid Kripke structure that is an abstraction of the hybrid Kripke structure from Figure 2. Solid lines and observables o comprise R^a and $L^a(o)$, respectively. Dashed lines and observables $o?$ comprise $R^c \setminus R^a$ and $L^c(o) \setminus L^a(o)$, respectively.

Kripke structure M in two modes, "a" (**a**sserted) and "c" (**c**onsistent), where $(M, s) \models^a$ and $(M, s) \models^c$ denote "ϕ must hold at state s in M" and "ϕ may hold at state s in M," respectively. If $M = (\Sigma, R, L)$ is a hybrid Kripke structure, then (Σ, R, R, L, L) is a 3-valued hybrid Kripke structure such that $\models^a$ and $\models^c$ are equal and so define $\models$ formally for M. We also define abstraction and refinement formally.

Definition 3.4 *Let* $A = (\Sigma, R^a, R^c, L^a, L^c)$ *and* $\hat{A} = (\hat{\Sigma}, \hat{R}^a, \hat{R}^c, \hat{L}^a, \hat{L}^c)$ *be two 3-valued hybrid Kripke structures with signature* $\mathsf{Obs} = \mathsf{AP} + \mathsf{Nom}$.

(i) *A relation* $Q \subseteq \Sigma \times \hat{\Sigma}$ *is a* refinement *iff* $(s, t) \in Q$ *implies*
  (a) *for all* $(t, t') \in \hat{R}^a$, *there is some* $(s, s') \in R^a$ *with* $(s', t') \in Q$;
  (b) *for all* $(s, s') \in R^c$, *there is some* $(t, t') \in \hat{R}^c$ *with* $(s', t') \in Q$;
  (c) *for all* $o \in \mathsf{Obs}$, $t \in \hat{L}^a(o)$ *implies* $s \in L^a(o)$; *and*
  (d) *for all* $o \in \mathsf{Obs}$, $s \in L^c(o)$ *implies* $t \in \hat{L}^c(o)$.

(ii) *We say that* $(\hat{A}, t)$ abstracts (is refined by) (A, s) *iff there is a refinement* Q *with* $(s, t) \in Q$.

(iii) *For* $s \in \Sigma$ *and* $n \in \mathsf{Nom}$, *the labelling* $L[n \mapsto^a s]$ *is the pair of labelling functions* $(L[n \mapsto^a s]^a, L[n \mapsto^a s]^c)$, *which equals* (L^a, L^c) *except at* n, *where* $L[n \mapsto^a s]^a(n) = L[n \mapsto^a s]^c(n) = \{s\}$; *the labelling* $L[n \mapsto^c s]$ *is the pair of labelling functions* $(L[n \mapsto^c s]^a, L[n \mapsto^c s]^c)$, *which equals* (L^a, L^c) *except at* n, *where* $L[n \mapsto^c s]^c(n) = L^c(n) \cup \{s\}$ *and* $L[n \mapsto^a s]^a(n) = \{\}$.

(iv) *We define* $\models^a_L$ *and* $\models^c_L$ *for 3-valued hybrid Kripke structures, where* $m \in \{a, c\}$, $\neg a = c$, *and* $\neg c = a$:
- $(A, s) \not\models^m_L \bot$;
- $(A, s) \models^m_L o$ *iff* $s \in L^m(o)$;
- $(A, s) \models^m_L \neg\phi$ *iff* $(A, s) \not\models^{\neg m}_L \phi$;
- $(A, s) \models^m_L @_n \phi$ *iff there is some* $s' \in L^m(n)$ *with* $(A, s') \models^m_L \phi$;

- *$(A, s) \models^m_L \downarrow n.\phi$ iff $(A, s) \models^m_{L[n \mapsto^m s]} \phi$;*
- *$(A, s) \models^m_L \exists n.\phi$ iff there is some s' with $(A, s) \models^m_{L[n \mapsto^m s']} \phi$;*
- *$(A, s) \models^m_L \phi_1 \wedge \phi_2$ iff $((A, s) \models^m_L \phi_1$ and $(A, s) \models^m_L \phi_2)$;*
- *$(A, s) \models^m_L \mathtt{EX}\,\phi$ iff there is some $(s, s') \in R^m$ such that $(A, s') \models^m_L \phi$;*
- *$(A, s) \models^m_L \mathtt{E}[\phi_1\,\mathtt{U}\,\phi_2]$ iff there is some $0 \leq j$ with $s = s_0$ such that for all $k \in \{0, 1, \ldots, j-1\}$ we have $(s_k, s_{k+1}) \in R^m$ and $(A, s_k) \models^m_L \phi_1$, and $(A, s_j) \models^m_L \phi_2$; and*
- *$(A, s) \models^m_L \mathtt{AF}\,\phi$ iff there is no infinite sequence $(s_i)_{i \geq 0}$ with $s = s_0$ such that, for all $k \geq 0$, $(s_k, s_{k+1}) \in R^{\neg m}$ and $(A, s_k) \not\models^m_L \phi$.*

Remark 3.5 *The ability to jump to arbitrary states in which to continue the evaluation of model checks means that (9) cannot be secured by just showing that the abstract state t indeed abstracts the concrete one s. Sound abstraction becomes a* global property *in that we need left-total and right-total refinement relations, which are thankfully closed under composition and subsume all state space partitions.*

The effect of $\hat{L}[n \mapsto^a t]$ in $\hat{A}$ is a "must"-bind of n to t; and the effect of $L[n \mapsto^c s]$ is a "may"-bind of n to s. Both actions constrain the un-abstracted "may-" and "must-" bindings of n in A conservatively.

Theorem 3.6 (i) *Let $A = (\Sigma, R^a, R^c, L^a, L^c)$ and $\hat{A} = (\hat{\Sigma}, \hat{R}^a, \hat{R}^c, \hat{L}^a, \hat{L}^c)$ be two 3-valued hybrid Kripke structures with signature $\mathsf{Obs} = \mathsf{AP} + \mathsf{Nom}$ and let $Q \subseteq \Sigma \times \hat{\Sigma}$ be a left-total and right-total refinement such that $(s, t) \in Q$. For all formulas $\phi \in CTL(@, \downarrow, \exists)$ we have that $(\hat{A}, t) \models^a_{\hat{L}} \phi$ implies $(A, s) \models^a_L \phi$; and $(A, s) \models^c_L \phi$ implies $(\hat{A}, t) \models^c_{\hat{L}} \phi$.*

(ii) *Let A be a 3-valued hybrid Kripke structure and $\hat{A}$ defined from A as in Definition 3.2 for a left-total and right-total ρ. Then $\hat{A}$ is a 3-valued hybrid Kripke structure and for all $s\rho t$, $(\hat{A}, t)$ abstracts (A, s). In particular, item (1) applies.*

Proof. Item (2) follows from item (1) by construction. We prove item (1) by structural induction on ϕ. We focus on the clauses o, $@_n\,\phi$, $\downarrow n.\phi$, and $\exists n.\phi$ as the proofs for the remaining clauses are standard (see e.g. [8,2,14]).

- Consider o.
  - · Let $(\hat{A}, t) \models^a_{\hat{L}} o$. Then $t \in \hat{L}^a(o)$. From $(s, t) \in Q$ we infer $s \in L^a(o)$ which implies $(A, s) \models^a_L o$.
  - · Let $(A, s) \models^c_L o$. Then $s \in L^c(o)$. From $(s, t) \in Q$ we infer $t \in \hat{L}^c(o)$ which implies $(\hat{A}, t) \models^c_{\hat{L}} o$.
- Consider $@_n\,\phi$.
  - · Let $(\hat{A}, t) \models^a_{\hat{L}} @_n\,\phi$. Then there is some t' with $t' \in \hat{L}^a(n)$ and $(\hat{A}, t') \models^a_{\hat{L}} \phi$.

Since Q is right-total, there is some s' with $(s', t') \in Q$ and so $t' \in \hat{L}^a(n)$ implies $s' \in L^a(n)$. By induction, $(s', t') \in Q$ and $(\hat{A}, t') \models^a_{\hat{L}} \phi$ imply $(A, s') \models^a_L \phi$. But then $s' \in L^a(n)$ implies $(A, s) \models^a_L @_n \phi$.

· Let $(A, s) \models^c_L @_n \phi$. Then there is some s' with $s' \in L^c(n)$ and $(A, s') \models^c_L \phi$. Since Q is left-total, there is some t' with $(s', t') \in Q$ and so $s' \in L^c(n)$ implies $t' \in \hat{L}^c(n)$. By induction, $(s', t') \in Q$ and $(A, s') \models^c_L \phi$ imply $(\hat{A}, t') \models^c_{\hat{L}} \phi$. But then $t' \in \hat{L}^c(n)$ implies $(\hat{A}, t) \models^c_{\hat{L}} @_n \phi$.

- Consider $\downarrow n.\phi$.
  · Let $(\hat{A}, t) \models^a_{\hat{L}} \downarrow n.\phi$. Then $(\hat{A}, t) \models^a_{\hat{L}[n \mapsto^a t]} \phi$. If we replace $\hat{L}$ with $\hat{L}[n \mapsto^a t]$ and L with $L[n \mapsto^a s]$ in $\hat{A}$ and A (respectively), then the assumptions of item (1) still hold. By induction, $(\hat{A}, t) \models^a_{\hat{L}[n \mapsto^a t]} \phi$ therefore implies $(A, s) \models^a_{L[n \mapsto^a s]} \phi$ and so $(A, s) \models^a_L \downarrow n.\phi$.
  · Let $(A, s) \models^c_L \downarrow n.\phi$. Then $(A, s) \models^c_{L[n \mapsto^c s]} \phi$. If we replace $\hat{L}$ with $\hat{L}[n \mapsto^c t]$ and L with $L[n \mapsto^c s]$ in $\hat{A}$ and A (respectively), then the assumptions of item (1) still hold. By induction, $(A, s) \models^c_{L[n \mapsto^c s]} \phi$ therefore implies $(\hat{A}, t) \models^c_{\hat{L}[n \mapsto^c t]} \phi$ and so $(\hat{A}, t) \models^c_{\hat{L}} \downarrow n.\phi$.
- Consider $\exists n.\phi$.
  · Let $(\hat{A}, t) \models^a_{\hat{L}} \exists n.\phi$. Then there is some t' with $(\hat{A}, t) \models^a_{\hat{L}[n \mapsto^a t']} \phi$. Since Q is right-total, there is some s' with $(s', t') \in Q$. If we replace $\hat{L}$ with $\hat{L}[n \mapsto^a t']$ and L with $L[n \mapsto^a s']$ in $\hat{A}$ and A (respectively), then the assumptions of item (1) still hold. By induction, $(\hat{A}, t) \models^a_{\hat{L}[n \mapsto^a t']} \phi$ therefore implies $(A, s) \models^a_{L[n \mapsto^a s']} \phi$ and so $(A, s) \models^a_L \exists n.\phi$.
  · Let $(A, s) \models^c_L \exists n.\phi$. Then there is some s' with $(A, s) \models^c_{L[n \mapsto^c s']} \phi$. Since Q is left-total, there is some t' with $(s', t') \in Q$. If we replace $\hat{L}$ with $\hat{L}[n \mapsto^c t']$ and L with $L[n \mapsto^c s']$ in $\hat{A}$ and A (respectively), then the assumptions of item (1) still hold. By induction, $(A, s) \models^c_{L[n \mapsto^c s']} \phi$ therefore implies $(\hat{A}, t) \models^c_{\hat{L}[n \mapsto^c t']} \phi$ and so $(\hat{A}, t) \models^c_{\hat{L}} \exists n.\phi$.

Example 3.7 *Let us re-consider the hybrid Kripke structure A of Figure 2 and its abstraction $\hat{A}$ of Figure 3.*

(i) *We have $(\hat{A}, t_2) \models^a_{\hat{L}} @_x \mathtt{E}[x \vee p \,\mathtt{U}\, \neg x]$ since we have $(\hat{A}, t_0) \models^a_{\hat{L}} \mathtt{E}[x \vee p \,\mathtt{U}\, \neg x]$, where the latter is witnessed by the $\hat{R}^a$-path $t_0 \to t_1 \to t_2$. Since $s_4 \rho t_2$, Theorem 3.6 entails that $(A, s_4) \models_L @_x \mathtt{E}[x \vee p \,\mathtt{U}\, \neg x]$.*

(ii) *Finally, we have $(\hat{A}, t_2) \models^c_{\hat{L}} \mathtt{E}[y \,\mathtt{U}\, \neg p]$ since $t_2 \in \hat{L}^c(p) \setminus \hat{L}^a(p)$, but don't have $(A, s_4) \models^c_L \mathtt{E}[y \,\mathtt{U}\, \neg p]$ despite the fact that $s_4 \rho t_2$. The direction of transfer of model-checking results is therefore mode-dependent as stated in Theorem 3.6.*

4 Hybrid labelled Markov chains

Hybrid logics enrich temporal logics and their models with the ability to name and therefore track states in a model. For Kripke structures and computation tree logic, this enrichment requires a multiplicity constraint on the labelling function (which had to be relaxed in abstraction-based model checking) and the addition of standard hybrid operators to computation tree logic. In moving from qualitative hybrid logics to quantitative and probabilistic ones, several questions emerge:

(i) How do or should hybrid operators generalize to a quantitative or probabilistic setting?

(ii) Is the use of model-checking back-ends and their data-structures (e.g. MTBBDs [5,1] and Kronecker Representation [18,9]) affected by the addition of hybrid operators, and if so how?

(iii) Do relational abstraction techniques for qualitative hybrid models transfer smoothly to the quantitative or probabilistic setting?

(iv) What is the complexity for model checking hybrid extensions of labelled Markov chains over hybrid extensions of probabilistic computation tree logic? It is worse than the one for the non-hybrid setting?

In this paper, we focus on the first question and only in the setting of finite-state labelled Markov chains and probabilistic computation tree logic without "bounded Until," e.g. as used in [1].

Definition 4.1 (i) *A* (finite-state) labelled Markov chain *with signature* Obs *is a tuple* $M = (\Sigma, R\colon \Sigma \times \Sigma \to [0,1], L\colon \mathsf{Obs} \to \mathbb{P}(\Sigma))$ *where* Σ *is finite;* Obs *is a set of atomic observables; for all* $s \in \Sigma$, $\sum_{s' \in \Sigma} R(s,s') = 1$*; and* Σ *and* L *have the same interpretation as for Kripke structures.*

(ii) *A* hybrid *(finite-state) labelled Markov chain with signature* $\mathsf{Obs} = \mathsf{AP} + \mathsf{Nom}$ *is a tuple* $M = (\Sigma, R\colon \Sigma \times \Sigma \to [0,1], L\colon (\mathsf{AP} \to \mathbb{P}(\Sigma)) + (\mathsf{Nom} \times \Sigma \to [0,1]))$ *where* $(\Sigma, R, L_{|\mathsf{AP}})$ *is a finite-state labelled Markov chain*[5] *;* AP *and* Nom *are disjoint sets of atomic propositions and nominals, respectively; and for all* $n \in \mathsf{Nom}$, $\sum_{s \in \Sigma} L(n,s) = 1$.

In a hybrid labelled Markov chain, the labelling function L has a sum type: as in the case of labelled Markov chains, $L(a)$ denotes those states of Σ in which atomic observable $a \in \mathsf{AP}$ holds; whereas $\lambda s.L(n,s)$ is the probability distribution of the nominal n in the state space Σ; see Figure 4 for a version of the hybrid Kripke structure from Figure 1 as a hybrid labelled Markov chain.

[5] We write $L_{|\mathsf{AP}}$ to denote the restriction of L to type $\mathsf{AP} \to \mathbb{P}(\Sigma)$.

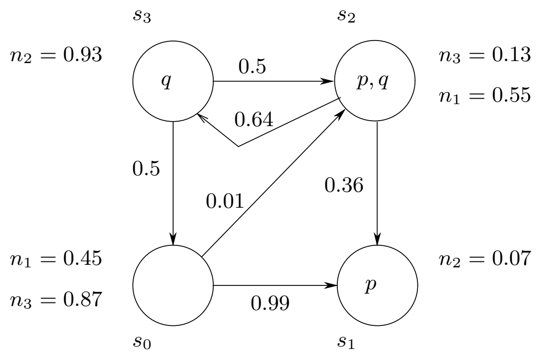

Fig. 4. A hybrid labelled Markov chain with signature $\mathsf{Obs} = \{p, q\} + \{n_1, n_2, n_3\}$. Probabilities of nominals are depicted next to the respective states. For example, $L(n_3, s_0) = 0.87$ and $L(n_3, s_3) = 0$.

We treat nominals probabilistically as the function $\lambda s.L(n, s)$ is a probability distribution over the set of states for each nominal $n \in \mathsf{Nom}$. Such a type is of interest as it models *probabilistic uncertainty of an observable agent's whereabouts*. But it also allows us to retain the original intent of hybrid logics by choosing $\lambda s.L(n, s)$ to be a point distribution $\delta_{s'}$ which assigns 1 to s' and 0 to all other states. Alternatively, one could choose other quantitative measures (risks, costs etc) so that $\sum_{s\in\Sigma} L(n, s)$ is no longer 1. The unifying point of such choices is that information about nominals is often uncertain or incomplete.

Now we discuss a suitable hybrid probabilistic temporal logic. The probabilistic computation tree logic (without "bounded Until")

$$\phi ::= \bot \mid a \mid \neg\phi \mid \phi \wedge \phi \mid [\mathrm{X}\phi]_{\sqsupseteq p} \mid [\phi\, \mathrm{U}\, \phi]_{\sqsupseteq p} \tag{10}$$

is due to Hansson & Jonsson [12] where $a \in \mathsf{AP}$, $p \in [0, 1]$, and $\sqsupseteq\ \in \{\geq, >\}$. Below we extend the familiar semantics of probabilistic computation tree logic over labelled Markov chains to our hybrid setting. This interpretation suggests probabilistic variants of the hybrid operators $@_n\, \phi$, $\downarrow n.\phi$, and $\exists n.\phi$:

Should the check $(M, s) \models_L @_n^{\sqsupseteq p}.\phi$ hold if $\sum\{L(n, s') \mid (M, s') \models_L \phi\} \sqsupseteq p$? Such an interpretation is similar to the one for $[\mathrm{X}\phi]_{\sqsupseteq p}$, expect that the state "transition" probabilities are governed by the probability distribution $\lambda s.L(n, s)$ instead of the probability distribution $\lambda s'.R(s, s')$, and is therefore computable with standard techniques from symbolic model checking of Markov chains, e.g. as implemented in the PRISM model checker [16]. Yet this interpretation is at odds with the role of conditional probabilities: Since $L(n, s')$ is the probability of n's being at state s', we wish to sum up all such weights for which the continuation check is true at s' under the assumption that “nominal n resides at state s',” so we have to set

(11) $(M, s) \models_L @_n^{\sqsupseteq p}.\phi$ iff $\sum\{L(n, s') \mid (M, s') \models_{L[n\mapsto\delta_{s'}]} \phi\} \sqsupseteq p\,.$

Unlike in the qualitative case, checks of $@_n^{\sqsupseteq p}\phi$ statically change the labelling function for the check of sub-formulas. Although this requires adaptations of existing algorithms for probabilistic model checking, the good news is that the continuation resolves the labelling information for n to a qualitative observable as found in a labelled Markov chain.

The qualitative check $(M, s) \models_L \downarrow n.\phi$ holds iff $(M, s) \models_{L[n\mapsto\delta_s]} \phi$ holds. Given that, we may as well assign probability distributions other than point distributions to the continuation of a probabilistic check:

(12) $(M, s) \models_L \downarrow(n, \delta).\phi$ iff $(M, s) \models_{L[n\mapsto\delta]} \phi\,.$

The qualitative check $(M, s) \models_L \exists n.\phi$ holds iff for some $s' \in \Sigma$ the check $(M, s) \models_{L[n\mapsto\delta_{s'}]} \phi$ holds. If we set $\Delta' = \{\delta_{s'} \mid s' \in \Sigma\}$, this is an instance of a general probabilistic check

(13) $(M, s) \models_L \exists(n, \Delta').\phi$ iff for some $\delta \in \Delta'$: $(M, s) \models_{L[n\mapsto\delta]} \phi\,.$

For this operator $\exists(n, \Delta')$ we may have to restrict the range of Δ' in order to make it computable or even feasibly so. We judge such extensions of probabilistic computation tree logic to be of potentially great use. For example, the idea of using probability distributions to model the presence of agents suggests applications in security.

This generality of probabilistic hybrid operators may not honor the original intent of hybrid temporal patterns. For example,

(14) $(M, s) \models_L \downarrow(n, \delta).[\neg\bot\ \mathrm{U}\ (n \wedge \psi)]_{\geq .9999}$

checks whether the node named by n is on a *probabilistic* cycle (on which ψ is true at least once) with probability at least .9999 *only* if the probability distribution δ does not smear the location of such a node, i.e. only if δ is of the form $\delta_{s'}$ for some $s' \in \Sigma$. Using point distributions, probabilistic hybrid logics are therefore able to express a kind of *probabilistic recurrence* of probabilistic trace sets.

5 Hybrid Probabilistic Computation Tree Logic

We summarize our discussion into a proposal for a hybrid probabilistic computation tree logic:

Definition 5.1 *Let* $\mathsf{Obs} = \mathsf{AP}+\mathsf{Nom}$ *be a signature for hybrid labelled Markov chains and* Δ *a class of discrete probability distributions subsuming all point distributions. Then* hybrid probabilistic computation tree logic, without “bounded Until,” *over* Obs *and* Δ *is defined by*

(15) $$\begin{aligned}\phi ::= \bot \mid a \mid \neg\phi \mid \phi\wedge\phi \mid [\mathrm{X}\phi]_{\sqsupseteq p} \mid [\phi\,\mathrm{U}\,\phi]_{\sqsupseteq p}\\ n^{\sqsupseteq p} \mid @_n^{\sqsupseteq p}.\phi \mid {\downarrow}(n,\delta).\phi \mid \exists(n,\Delta').\phi\end{aligned}$$

where $a \in \mathsf{AP}$, $n \in \mathsf{Nom}$, $p \in [0,1]$, $\sqsupseteq\ \in \{\geq, >\}$, $\delta \in \Delta$, *and* $\Delta' \subseteq \Delta$.

The qualitative operators ${\downarrow}n.\phi$ and $\exists n.\phi$ are derived in that $(M,s) \models_L {\downarrow}n.\phi$ is interpreted as $(M,s) \models_L {\downarrow}(n,\delta_s).\phi$; and $(M,s) \models_L \exists n.\phi$ as $(M,s) \models_L \exists(n,\{\delta_s \mid s \in \Sigma\}).\phi$.

Let $M = (\Sigma, R, L)$ be a hybrid finite-state labelled Markov chain with signature Obs. We define $(M,s) \models_L \phi$ for all ϕ of hybrid probabilistic computation tree logic. Given $s \in \Sigma$, let $\mathit{Path}(s)$ be the set of infinite paths in M beginning in s, where transitions $s \to s'$ occur iff $R(s,s') > 0$. Given ϕ, ϕ_1, and ϕ_2 of hybrid probabilistic computation tree logic and some $\pi \in \mathit{Path}(s)$ we define

- $\pi \models_L X\,\phi$ iff $(M,s') \models_L \phi$, where $\pi = s \to s' \to \ldots$;
- $\pi \models_L \phi_1\; U\; \phi_2$ iff there is some $k \geq 0$ such that the first $k-1$ states s_i of π satisfy $(M,s_i) \models_L \phi_1$ and the kth state s_k satisfies $(M,s_k) \models_L \phi_2$.

So we define $\models_L$ over certain path formulas and all state formulas of hybrid probabilistic computation tree logic by mutual induction, as done for probabilistic computation tree logic [12]. The semantics for $\bot$, a, negation, and conjunction is defined as for Kripke structures. The semantics for the path formulas and hybrid operators is

- $(M,s) \models_L [\mathrm{X}\phi]_{\sqsupseteq p}$ iff the probability of the set of those $\pi \in \mathit{Path}(s)$ with $\pi \models_L X\,\phi$ is $\sqsupseteq p$;
- $(M,s) \models_L [\phi_1\,\mathrm{U}\,\phi_2]_{\sqsupseteq p}$ iff the probability of the set of those $\pi \in \mathit{Path}(s)$ with $\pi \models_L \phi_1\; U\; \phi_2$ is $\sqsupseteq p$;
- $(M,s) \models_L n^{\sqsupseteq p}$ iff $L(n,s) \sqsupseteq p$;
- $(M,s) \models_L @_n^{\sqsupseteq p}.\phi$ iff $\sum\{L(n,s') \mid (M,s') \models_{L[n\mapsto\delta_{s'}]} \phi\} \sqsupseteq p$;
- $(M,s) \models_L {\downarrow}(n,\delta).\phi$ iff $(M,s) \models_{L[n\mapsto\delta]} \phi$; and
- $(M,s) \models_L \exists(n,\Delta').\phi$ iff for some $\delta \in \Delta'$, we have $(M,s) \models_{L[n\mapsto\delta]} \phi$.

Note that $\models_L$ is well defined for all hybrid finite-state labelled Markov chains since all path formulas over predicates (the sets of states for which a particular formula of hybrid probabilistic computation tree logic is true) give rise to measurable path sets [20] through the standard construction of measures on sequence spaces. The semantics for path formulas given above is as for probabilistic computation tree logic.

Example 5.2 *To illustrate our semantics of hybrid probabilistic computation tree logic, we check* $(M,s_3) \models_L @_{n_3}^{>0.1}[\neg\bot\,\mathrm{U}\,n_3]_{\geq 0.01}$. *For that, we need to de-*

termine for which s with $L(n_3, s) > 0$ we have $(M, s) \models_{L[s \mapsto \delta_s]} [\neg \bot \,\mathrm{U}\, n_3]_{\geq 0.01}$; and then sum up all those $L(n_3, s)$ and check whether that sum is > 0.1. Only s_0 and s_2 are relevant here.

- *At state s_0 in M with labelling function $L[n_3 \mapsto \delta_{s_0}]$ the probability that s_0 is on a cycle is $0.01 \cdot 0.64 \cdot 0.5 \cdot (\sum_{i=0}^{\infty}(0.64 \cdot 0.5)^i) = 0.00948529\ldots$ which is not ≥ 0.01 and so $(M, s_0) \models_{L[s \mapsto \delta_{s_0}]} [\neg \bot \,\mathrm{U}\, n_3]_{\geq 0.01}$ does not hold, meaning that $L(s_0, n_3) = 0.87$ does not contribute to that sum.*
- *At state s_2 in M with labelling function $L[n_3 \mapsto \delta_{s_2}]$ the probability that s_2 is on a cycle is $0.64 \cdot 0.5 + 0.64 \cdot 0.5 \cdot 0.01 = 0.3232$ which is ≥ 0.01 and so $(M, s_2) \models_{L[s \mapsto \delta_{s_2}]} [\neg \bot \,\mathrm{U}\, n_3]_{\geq 0.01}$ holds, meaning that $L(s_2, n_3) = 0.13$ is the only contributor to that sum.*

Since $0.13 > 0.1$, we conclude that $(M, s_3) \models_L @_{n_3}^{>0.1}[\neg \bot \,\mathrm{U}\, n_3]_{\geq 0.01}$ holds.

6 Conclusions

We presented propositional hybrid logics as established enhancements of propositional temporal logics with the ability to name and re-bind specific states. We then provided a sound relational abstraction technique for hybrid Kripke structures and a hybrid version of computation tree logic. We further motivated and discussed a definition of hybrid labelled Markov chains and a syntax and semantics of probabilistic computation tree logic in this hybrid setting. Our abstraction techniques for hybrid Kripke structures should be transferable to hybrid labelled Markov chains and *quantitative* hybrid models, perhaps along the lines of [13].

Acknowledgments

Our exposition of hybrid logics was heavily inspired by the introductory article at http://www.hylo.net/ and comments made by Chris Hankin, Sebastian Nanz, and Herbert Wiklicky. We thank the anonymous referees, Joost-Pieter Katoen, and Prakash Panangaden for constructive comments that helped shape the final version of this paper.

References

[1] C. Baier, E. M. Clarke, V. Hartonas-Garmhausen, M. Kwiatkowska, and M. Ryan. Symbolic Model Checking for Probabilistic Processes. In *Proc. ICALP'97*, volume 1256 of *Lecture Notes in Computer Science*, pages 430–440, 1997.

[2] G. Bruns and P. Godefroid. Model Checking Partial State Spaces with 3-Valued Temporal Logics. In *Proc. of the 11th Conference on Computer Aided Verification*, volume 1633 of *Lecture Notes in Computer Science*, pages 274–287. Springer Verlag, July 1999.

[3] G. Bruns and P. Godefroid. Generalized Model Checking: Reasoning about Partial State Spaces. In *Proc. of CONCUR'2000 (11th International Conference on Concurrency Theory)*, volume 1877 of *Lecture Notes in Computer Science*, pages 168–182. Springer Verlag, August 2000.

[4] E. M. Clarke and E. A. Emerson. Synthesis of synchronization skeletons for branching time temporal logic. In D. Kozen, editor, *Logic of Programs Workshop*, number 131 in LNCS. Springer Verlag, 1981.

[5] E. M. Clarke, M. Fujita, and X. Zhao. *Representations of discrete functions*, chapter Multi-terminal binary decision diagrams and hybrid decision diagrams, pages 93–108. Kluwer academic publishers, 1996.

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

[7] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking.* The MIT Press, January 2000.

[8] D. Dams. *Abstract interpretation and partition refinement for model checking.* PhD thesis, Technische Universiteit Eindhoven, The Netherlands, 1996.

[9] L. de Alfaro, M. Kwiatkowska, G. Norman, D. Parker, and R. Segala. Symbolic Model Checking of Probablistic Processes using MTBBDs and the Kronecker Representation. In *Tools and Algorithms for the Construction and Analysis of Systems: 6th International Conference, TACAS 2000*, volume 1785 of *Lecture Notes in Computer Science*, pages 395–410, Berlin, Germany, March/April 2000. Springer Verlag.

[10] M. Franceschet and M. de Rijke. Model Checking for Hybrid Logics. In *Proc. of the Workshop on Methods for Modalities*, 2003.

[11] P. Godefroid and R. Jagadeesan. Automatic Abstraction Using Generalized Model Checking. In E. Brinksma and K. G. Larsen, editors, *Proc. 14th Int'l Conference on Computer Aided Verification (CAV 2002)*, volume 2404 of *Lecture Notes in Computer Science*, pages 137–150, Copenhagen, Denmark, July 2002. Springer Verlag.

[12] H. A. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.

[13] M. Huth. Possibilistic and Probabilistic Abstraction-Based Model Checking. In H. Hermanns and R. Segala, editors, *Process Algebra and Probabilistic Methods, Performance Modeling and Verification, Second Joint International Workshop PAPM-PROBMIV 2002*, volume 2399 of *Lecture Notes in Computer Science*, pages 115–134, Copenhagen, Denmark, July 25-26 2002. Springer.

[14] M. Huth, R. Jagadeesan, and D. A. Schmidt. Modal transition systems: a foundation for three-valued program analysis. In Sands D., editor, *Proc. of the European Symposium on Programming (ESOP'2001)*, pages 155–169. Springer Verlag, April 2001.

[15] D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[16] M. Kwiatkowska. Model Checking for Probability and Time: From Theory to Practice. Invited paper in *Proc. LICS'03*, pages 351-360, IEEE Computer Society Press, 2003.

[17] K. G. Larsen and B. Thomsen. A Modal Process Logic. In *Proc. of LICS'88*, pages 203–210. IEEE Computer Society Press, 1988.

[18] B. Plateau. On the Stochastic Structure of Parallelism and Synchronization Models for Distributed Algorithms. In *Proc. of the 1985 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pages 147–154, Austin, Texas, May 1985. ACM Press.

[19] M. Sagiv, T. Reps, and R. Wilhelm. Parametric Shape Analysis via 3-Valued Logic. In *Proc. of the 26th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 105–118, January 20-22, San Antonio, Texas 1999.

[20] M. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. 26th IEEE Symp. on Foundations of Computer Science*, pages 327–338, Portland, Oregon, October 1985.