Decidability of Quantifed Propositional Branching Time Logics

Tim French

Murdoch University, Murdoch, Perth, W.A. 6150,
Australia.
t.french@murdoch.edu.au

Abstract. We extend the branching temporal logics CTL and CTL* with quantified propositions and consider various semantic interpretations for the quantification. The use of quantificiation greatly increases the expressive power of the logics allowing us to represent, for example, tree-automata. We also show that some interpretations of quantification allow us to represent non-propositional properties of Kripke frames, such as the branching degree of trees. However this expressive power may also make the satisfiability problem for the logic undecidable. We give a proof of one such case, and also examine decidability in the less expressive semantics.

1 Introduction

Temporal logic has been particularly useful in reasoning about properties of systems. In particular, the branching temporal logics CTL* [4] and CTL [2] (a syntactic restriction of CTL*) have been used to verify the properties of non-deterministic and concurrent programs.

It is our goal to extend these results to a logic that allows propositional quantification. In the linear case PLTL [16] has been augmented with quantified propositions to get the significantly more expressive QPTL [18]. In [9] QPTL was shown to be able to reason about ω-automata, and prove the existence of refinement mappings [1]. A refinement mapping shows one system specification S_1 implements some other system specification S_2 by encoding the specifications in temporal logic where variables not common to S_1 and S_2 are quantified out. Propositional quantification has been shown to be related to logics of knowledge [8]. Finding decidable semantics for quantified propositional branching time logics is an important step to finding suitable semantics for the more expressive epistemic logics.

Previously branching temporal logics with quantified propositions have been examined in [11], [4], and [13], though only existential quantification was considered, and the structures were limited to trees. More powerful semantics for propositional quantification have been studied in intuitionistic propositional logic [19],

M. Brooks, D. Corbett, and M. Stumptner (Eds.): AI 2001, LNAI 2256, pp. 165–176, 2001.

[15]. In this paper we give three separate semantic interpretations for full propositional quantification. We show that one of these, the *Kripke semantics*, is highly undecidable. While the other two are decidable, the *tree semantics* can be used to reason about purely structural properties of the model (like the branching degree of trees). This can complicate the notion of refinement, so we introduce the *amorphous semantics* to overcome this, and sketch a decision process.

2 The Base Semantics, CTL*

We first describe the syntax and semantics of CTL*. The language CTL* consists of an infinite set of atomic variables $\mathcal{V} = \{x_0, x_1, ...\}$, the boolean operations $\neg, \vee$, the temporal operators $\bigcirc$, $\Box$, U (**next**, **generally** and **until** respectively) and the path quantifier E. The formulae of CTL* are defined by the following abstract syntax, where x varies over $\mathcal{V}$:

$$\alpha ::= x \mid \neg\alpha \mid \alpha_1 \vee \alpha_2 \mid \bigcirc\alpha \mid \Box\alpha \mid \alpha_1 \mathrm{U} \alpha_2 \mid \mathrm{E}\alpha \tag{1}$$

Definition 1. *A state formula of CTL* is a formula where every temporal operator ($\bigcirc$, $\Box$ or U) is in the scope of a path quantifier (E).*

The abbreviations $\wedge, \rightarrow, \leftrightarrow$ are defined as usual, and we define $\Diamond\alpha$ (future) to be $\neg\Box\neg\alpha$, and Aα to be $\neg$E$\neg\alpha$. We also consider the formulas $\top, \bot$ (respectively "true" and "false") to be abbreviations for, respectively $x_0 \vee \neg x_0$ and $\neg(x_0 \vee \neg x_0)$. To give the semantics for CTL* we define *$\mathcal{V}$-labeled Kripke frames*:

Definition 2. *A Kripke frame is a tuple (S, R) where*

1. *S is a nonempty set of states, or moments.*
2. *$R \subseteq S^2$ is a total binary relation.*

A $\mathcal{V}$-labeled Kripke frame is a Kripke frame with a valuation $\pi : S \longrightarrow \wp(\mathcal{V})$.

Let $M = (S, R, \pi)$ be a $\mathcal{V}$-labelled Kripke frame. A *path* b in M is an ω-sequence of states $b = (b_0, b_1, ...)$ such that for all i, $(b_i, b_{i+1}) \in R$ and we let $b_{\geq i} = (b_i, b_{i+1}, ...)$. We interpret a formula α of CTL* with respect to a $\mathcal{V}$-labelled Kripke frame M and a path b in M. We write $M, b \models \alpha$ where:

$$M, b \models x \Longleftrightarrow x \in \pi(b_0) \tag{2}$$

$$M, b \models \neg\alpha \Longleftrightarrow M, b \not\models \alpha \tag{3}$$

$$M, b \models \alpha \vee \beta \Longleftrightarrow M, b \models \alpha \text{ or } M, b \models \beta \tag{4}$$

$$M, b \models \bigcirc\alpha \Longleftrightarrow M, b_{\geq 1} \models \alpha \tag{5}$$

$$M, b \models \Box\alpha \Longleftrightarrow \forall i \geq 0,\ M, b_{\geq i} \models \alpha \tag{6}$$

$$M, b \models \alpha \mathrm{U} \beta \Longleftrightarrow \exists i \geq 0,\ M, b_{\geq i} \models \beta \text{ and } M, b_{\geq j} \models \alpha \text{ for all } j < i \tag{7}$$

$$M, b \models \mathrm{E}\alpha \Longleftrightarrow \text{ there is some path } b' \text{ s.t. } b'_0 = b_0 \text{ and } M, b' \models \alpha. \tag{8}$$

From the semantics given above we can see that the evaluation of a state formula only depends on the initial state of the path b, rather than the whole path. We restrict our attention to state formulas and define a *model* to be the tuple (S, R, π, s) (or (M, s)) where $s \in S$. A state formula α is *satisfied* by (M, s) (denoted $(M, s) \models \alpha$) if there is some path b in M with $b_0 = s$ and $M, b \models \alpha$. If for all models (M, s) we have $(M, s) \models \alpha$, then we say α is a *validity*.

The language CTL is a syntactic restriction of CTL*. Particularly, CTL requires that every temporal operator is paired with a path quantifier. To define CTL we only need to modify the abstract syntax (1) as follows:

$$\alpha ::= x \mid \neg\alpha \mid \alpha_1 \vee \alpha_2 \mid \mathrm{E}\bigcirc\alpha \mid \mathrm{E}\,\Box\alpha \mid \mathrm{A}\,\Box\alpha \mid \mathrm{E}(\alpha_1 \mathrm{U} \alpha_2) \mid \mathrm{A}(\alpha_1 \mathrm{U} \alpha_2)$$

The logic CTL is less expressive than CTL*, though it is frequently preferred as the model checking complexity is less than that for CTL*. We will show that this difference disappears when propositional quantification is considered.

3 Syntax and Semantics for Propositional Quantification

We add the operator $\exists$ to the languages defined above and define QCTL* to be the language consisting of the formulae defined by the abstract syntax above, where the following inductive step is included:

If α is a formula and $x \in \mathcal{V}$ then $\exists x\alpha$ is a formula.

The set of formulae of QCTL* is closed under complementation, and we let $\forall x\alpha$ be an abbreviation for $\neg\exists x\neg\alpha$. The logic QCTL is similarly defined to extend CTL.

The semantic interpretation of propositional quantification will rely on the following definition.

Definition 3. *Given some model $(M, s_0) = (S, R, \pi, s_0)$ and some $x \in \mathcal{L}$, an x-variant of (M, s_0) is some model $(M', s_0) = (S, R, \pi', s_0)$ where $\pi'(s)\backslash\{x\} = \pi(s)\backslash\{x\}$ for all $s \in S$.*

To complete the interpretation for formulae of QCTL* we augment the interpretation above (2-8) with

$$M, b \models \exists x\alpha \iff \text{There is some } x\text{-variant } M' \text{ of } M \text{ such that } M', b \models \alpha. \quad (9)$$

Since we will consider other possible interpretations below we refer to the set of semantics defined above as the *Kripke semantics.* We will show that QCTL* becomes highly undecidable over such a general semantic interpretation. There are two possible ways to overcome this.

3.1 Tree Semantics

The undecidability of the Kripke semantics results from the logic being able to specify structural properties of the model. By restricting the structures to trees, QCTL* becomes decidable. We give the following definitions and note the semantics for CTL* (2-8) can be restricted to $\mathcal{V}$-labelled trees without change.

Given some relation $R \subseteq S^2$ the *transitive closure* of R is $<_R \subseteq S^2$ where $(s,t) \in <_R$ if and only if for some $n > 0$ there exists $s_0, ...s_n \in S$ with $s_0 = s$, $s_n = t$ and for $i < n$, $(s_i, s_{i+1}) \in R$.

Definition 4. *A $\mathcal{V}$-labelled tree, (S, R, π, s) is a model that satisfies the following conditions:*

1. *S is a (countably) infinite set of nodes.*
2. *$<_R$ is irreflexive.*
3. *The past of $t \in S$, $\{s \in S | s <_R t\}$ is linerarly ordered by $<_R$.*
4. *Each maximally ordered subset of S is order-isomorphic to $\mathbb{N}$.*

We refer to this restriction as the *tree semantics*. Given any model, $(M, s) = (S, R, \pi, s_0)$ we can generate a $\mathcal{V}$-labelled tree $(M^T, s) = (S', R', \pi', s_0)$ (the *unwinding* of (M, s)) where:

– $S' \subseteq S^*$, $s_0 \in S'$, and for any word $w \in S^*$ and any $s \in S$ with $ws \in S'$ then $(s, t) \in R$ if and only if $wst \in S'$.
– $R' = \{(w, ws) \mid w \in S', ws \in S', s \in S\}$.
– $\pi'(s_0) = \pi(s_0)$ and $\pi'(ws) = \pi(s)$ for $s \in S$.

Lemma 1. *CTL* is insensitive to unwinding. That is $(M, s) \models \alpha$ if and only if $(M^T, s) \models \alpha$.*

This was proven in [3]. However QCTL* is not insensitive to unwinding, as was shown in [11]. For example consider a model consisting of a single state. For all possible valuations a proposition would be always true or always false, which is clearly not the case in tree semantics. While QCTL* becomes decidable in the tree semantics, it can still define purely structural properties of the model. Particularly, QCTL (and hence QCTL*) can define the number of successors a state has. For every $i \in \mathbb{N}$ we can define the state formula B_i such that $M, s \models B_i$ if and only if s has exactly i successors. For example $B_2 = \exists y B_2(y)$ where

$$B_2(y) = ((\mathrm{E}\bigcirc y \wedge \mathrm{E}\bigcirc \neg y) \wedge \forall x(\mathrm{E}\bigcirc(y \wedge x) \wedge \mathrm{E}\bigcirc(\neg y \wedge x) \rightarrow \mathrm{A}\bigcirc x)), \quad (10)$$

If there were only one successor of s, then $\mathrm{E}\bigcirc y$ and $\mathrm{E}\bigcirc\neg y$ could not both be true. If there were more than two successors, we could have x true at exactly two of the successors, and y true at exactly one of these. Then the left side of the implication would be satisfied, but $\mathrm{A}\bigcirc x$ would not.

3.2 Amorphous Semantics

The second way to avoid the undecidability of QCTL* is to give a different interpretation of propositional quantificiation. This will remove the ability of QCTL* to define any of the structural properties of the underlying model. The new interpretation requires the following definition:

Definition 5. *Given $X \subseteq \mathcal{V}$, the models $(M, s_0) = (S, R, \pi, s_0)$ and $(M', s') = (S', R', \pi', s'_0)$ are X-bisimilar if there exists some relation $B \subseteq S \times S'$ with $(s_0, s'_0) \in B$ and for all $(s, s') \in B$:*

1. *$\pi(s)\backslash X = \pi'(s')\backslash X$.*
2. *For all $t \in S$ such that $(s, t) \in R$, there exists $t' \in S'$ with $(s', t') \in R'$ such that $(t, t') \in B$.*
3. *For all $t' \in S'$ such that $(s', t') \in R'$, there exists $t \in S$ with $(s, t) \in R$ such that $(t, t') \in B$.*

The pairs (M, b) and (M', b') are X-bisimilar if there exists such a relation B for (M, b_0) and (M', b'_0) with $(b_i, b'_i) \in B$ for all $i \geq 0$. We write $\{x\}$-bisimilar as x-bisimilar, and $\emptyset$-bisimilar as bisimilar.

This definition is based on the notion of the bisimilarity of synchronization trees [14] and a similar notion of quantification has been considered in the case of intuitionistic propositional logic [19]. The amorphous semantics replace the interpretation of quantification (9) with

$$M, b \models \exists x\alpha \Leftrightarrow \text{ there is } (M', b'),\ x\text{-bisimilar to } (M, b),\ \text{with } M', b' \models \alpha. \quad (11)$$

The amorphous semantics allow us to disregard the purely structural properties of the model (for example, the formula B_2 (10) becomes unsatisfiable). This is particularly useful for proving the refinement of concurrent specifications, since the specifications do not have to be considered over identical structures. We give the following lemma without proof, though it is not hard to show.

Lemma 2. *1. X-bisimilarity is an equivalence relation.*
2. (M, s) and (M^T, s) are bisimilar.
3. QCTL* is insensitive to unwinding in the amorphous semantics.

4 Definability

Before addressing the decidability of QCTL* we simplify the syntax we must use. Particularly we show that in the case of the tree and the amorphous semantics, QCTL* is definable in a restriction of QCTL that does not include the U operators. We first claim that the U operator is definable by the other operators of QCTL*. The construction is taken from [9], and its soundness can easily be shown.

Given some pair (M, b) where $b = (b_0, b_1, ...)$ is a path in the model (M, b_0), and formulas α and β of QCTL* which do not contain the variable x_1, let $Until(\alpha, \beta) = \exists x_1(\Diamond \beta \wedge (x_1 \wedge \Box(x_1 \rightarrow (\beta \vee (\alpha \wedge \bigcirc x_1)))))$.

Lemma 3. *In the tree and amorphous semantics* $(M, b) \models \alpha U \beta \Leftrightarrow (M, b) \models \exists x_1 Until(\alpha, \beta)$.

We now show how every temporal operator can be paired with a path quantifier. By the above lemma we do not have to address the U operator.

Let Eα be any formula of QCTL* such that any subformula of α containing a branch quantifier is a formula of QCTL. For some $y \in \mathcal{V}$ that is not a variable of α we let $\alpha'(y)$ be the formula that results when all subformulas $\Box \beta$ of α, which are not directly preceded by a path quantifier are replaced with A$\Box(y \rightarrow \beta)$, and likewise for the $\bigcirc$ operator. Let

$$\alpha^* = \exists z(z \wedge \mathrm{A}\Box(z \leftrightarrow \mathrm{E}\bigcirc z) \wedge \forall y(y \wedge \mathrm{A}\Box(y \leftrightarrow (\mathrm{E}\bigcirc y \wedge z)) \rightarrow \alpha'(y))). \quad (12)$$

Lemma 4. $(M, s) \models E\alpha \Leftrightarrow (M, s) \models \alpha^*$*, in the amorphous and tree semantics.*

Proof. (Tree Semantics) The formula (12) restricts the evaluation of $\alpha'(y)$ to models where y is true on a set of branches, and the formula A$\Box(y \rightarrow \beta)$ restricts the interpretation only to paths where y is true. If we can restrict y to a single path the interpretation becomes equivalent to $\Box \beta$. Suppose $(M, s) \models$ Eα. We can choose z to be true only on a single path for which α is true. Then y can only be true along this path and the result follows. Conversely if $(M, s) \models \alpha^*$ then since we are considering all possible interpretations of y, then we must consider some interpretation which has y true on a single path, and Eα must be true. The $\bigcirc$ operator can be treated in the same way and the result follows. The proof for the amorphous semantics is similar, though the structure must be unwound first.

Corollary 1. *Given the amorphous semantics or the tree semantics, QCTL* is definable in QCTL.*

Proof. Given some formula α we first remove all occurences of U by using Lemma 3. The result can then be shown by induction over the complexity of formulas working from the inside out, and using the Lemma 4 to convert each branch quantified subformula to a formula of QCTL.

5 Decidability

We have defined three possible sets of semantics for QCTL*. The satisfiability problem for QCTL* will be shown to be highly undecidable for the Kripke semantics, while it is decidable for the tree and the amorphous semantics.

5.1 Undecidability

The consequence of the expressive power of the Kripke semantics QCTL is that the satisfiablity problem becomes undecidable. In fact Kremer [15] has shown that intuitionistic propositional logic with quantified propositions over Kripke structures is recursively isomorphic to full second-order logic. It is belived that QCTL can be shown to be just as powerful, though here we simply show that QCTL (and hence QCTL*) is not recursively enumerable. This is done by encoding the following tiling for $(\mathbb{N}, <) \times (\mathbb{N}, <)$ in QCTL.

We are given a finite set $\Gamma = \{\gamma_i | i = 1, ..., m\}$ of tiles. Each tile γ_i has four coloured sides: left, right, top and bottom, written γ_i^l, γ_i^r, γ_i^t, and γ_i^b. Each side can be one of n colours c_j for $j = 1, ..., n$. Given any set of these tiles, we would like to know if we can cover the plane $\mathbb{N} \times \mathbb{N}$ with these tiles such that adjacent sides share the same colour. Formally, given some finite set of tiles Γ we would like to decide if there exists a function $\lambda : \mathbb{N} \times \mathbb{N} \longrightarrow \Gamma$ such that for all $(x, y) \in \mathbb{N} \times \mathbb{N}$

1. $\lambda(x, y)^r = \lambda(x + 1, y)^l$
2. $\lambda(x, y)^t = \lambda(x, y + 1)^b$

where $\lambda(x, y)^t$ is the colour of the top side of the tile on (x, y), and likewise for the other sides. Finally we require that there is some specific tile γ_j that occurs infinitely often in the bottom row (i.e. $\lambda(x, 0) = \gamma_j$ for infinitely many x. In [7] this problem was shown to be highly undecidable, or Σ_1^1.

Theorem 1. *Given the Kripke semantics, the satisfiability problem for QCTL is highly undecidable.*

Given the set of tiles Γ we give a formula, $Tile^\Gamma$, of QCTL that is satisfiable if and only if the above tiling problem is satisfiable. To specify that some tile γ_j occurs infinitely often in the bottom row, we let γ_0 be a copy of γ_j, and suppose that γ_0 occurs only in the bottom row. This is clearly equivalent to the above problem.

We start by giving a formula that specifies the underlying Kripke structure to be a grid (i.e. a structure similar to a binary tree, but with the branches rejoining). To make a grid we have to specify that the two successors of any state have a common successor. This is done with the following formula:

$$G(y,z) = (y \wedge \mathrm{E}\bigcirc(y \wedge \mathrm{A}\bigcirc z)) \rightarrow \mathrm{E}\bigcirc(\neg y \wedge \mathrm{E}\bigcirc(z \wedge \neg y)) \tag{13}$$

$$S(y) = \mathrm{A}\,\square(B_2(y) \wedge \forall z(G(y,z) \wedge G(\neg y, z)) \tag{14}$$

The formula $B_2(y)$ (10) specifies the branching degree to be two, and that y is true for exactly one successor of any state. This formula uses the universally quantified variable z to ensure that the two successors of any state share a successor, since any interpretation that makes z true for all the successors of the first successor of some point, makes z true for at least one successor of the second successor of that point.

To encode the tiling we let each tile γ_i be represented by the variable t_i and define the formula $T_i = t_i \wedge \neg \bigvee_{j \neq i} t_j$. Rather than explicitly encoding the colours we just place restrictions on which tiles can succeed other tiles:

$$C^{\Gamma}(y) = \bigvee_{i=0}^{m} \left(T_i \wedge y \wedge \mathrm{E}\bigcirc \left(y \wedge \bigvee_{\gamma_j^l = \gamma_i^r}^{j<m} T_j \right) \wedge \mathrm{E}\bigcirc \left(\neg y \wedge \bigvee_{\gamma_j^b = \gamma_i^t}^{j<m} T_j \right) \right) \tag{15}$$

$$B^{\Gamma}(y) = \mathrm{A}\,\square(\neg y \rightarrow \mathrm{A}\,\square \neg T_0) \wedge \mathrm{E}\,\square \mathrm{E} \diamond T_0 \tag{16}$$

$$\Lambda^{\Gamma}(y) = B^{\Gamma}(y) \wedge \mathrm{A}\,\square(C^{\Gamma}(y) \vee C^{\Gamma}(\neg y)). \tag{17}$$

The first formula specifies the way tiles can fit together. The variable y is used to define rows and columns since the value of y is fixed along a row, and alternates along a column. The second formula specifies that the variable t_0 occurs only, and infinitely often, on the bottom row, since the only path where $\neg y$ is always true is the bottom row. We define the formula

$$Tile^{\Gamma} = y \wedge S(y) \wedge \Lambda^{\Gamma}(y). \tag{18}$$

Lemma 5. *The formula $Tile^{\Gamma}$ is satisfiable if and only if Γ can tile $(\mathbb{N}, <) \times (\mathbb{N}, <)$, with γ_0 occuring infinitely often on the bottom row.*

Proof. $(\longrightarrow)$ Suppose that $Tile^{\Gamma}$ is satisfied by some model $M = (S, R, \pi, s_0)$. We define the function $\mu : \mathbb{N} \times \mathbb{N} \longrightarrow S$ recursively such that $\mu((0,0)) = s_0$, and $\mu((a,b)) = t$ where

$$\mu(a-1,b) = s,\ (s,t) \in R \text{ and } y \in \pi(s) \leftrightarrow y \in \pi(t) \tag{19}$$

$$\text{or} \quad \mu(a,b-1) = s,\ (s,t) \in R \text{ and } y \in \pi(s) \leftrightarrow y \notin \pi(t) \tag{20}$$

The function μ will be surjective and well defined since every $s \in S$ has exactly two successors, by $B_2(y)$ and y will always be true on exactly one of these successors. Therefore there is always exactly one successor of s satisfying (19), and exactly one successor satisfying (20). We can then define the tiling function as $\lambda((a,b)) = \gamma_i \iff t_i \in \mu((a,b))$. The use of T_i in $\Lambda^{\Gamma}(y)$ ensures that every point (a, b) is assigned exactly one tile, and the function $B^{\Gamma}(y)$, along with the definition of μ ensures that the tile γ_0 occurs infinitely often on the bottom row. All that is left to show is that the sides of the tiles match up. To do this suppose at some state $\mu(a, b)$, y and t_i are true. Then by the definition of μ, at $\mu(a+1, b)$ y is true and by $C^{\Gamma}(y)$ there is some j such that t_j is true where $\gamma_i^l = \gamma_j^r$. Similarly at $\mu(a, b+1)$ y is not true and there is some k with t_k true where $\gamma_i^t = \gamma_k^b$. By the formula $S(y)$, $\mu(a+1, b)$ and $\mu(a, b+1)$ have a common successor where y is not true. By the definition of μ this state must be $\mu(a+1, b+1)$, and suppose t_ℓ is true at this state. The formula $C^{\Gamma}(y)$ ensures $\gamma_k^t = \gamma_\ell^b$ and similarly $C^{\Gamma}(\neg y)$ ensures $\gamma_j^l = \gamma_\ell^r$. Likewise we can show the case for $\neg y$, so the sides of any four adjacent tiles match up, and by applying a similar argument recursively we can see that the generated tiling is sound.

$(\longleftarrow)$ Given that λ is a tiling for Γ of $(\mathbb{N}, <) \times (\mathbb{N}, <)$ we can construct the model $M = (S, R, \pi, s_0)$ where $S = \mathbb{N} \times \mathbb{N}$, $R = \{((a, b), (c, d)) | c = a + 1 \text{ or } d = b + 1\}$, and $s_0 = (0, 0)$. We define $y \in \pi((a, b))$ iff b is even, and $t_i \in \pi(a, b)$ iff $\lambda(a, b) = \gamma_i$. It is straightforward to show that $(M, s_0) \models \mathit{Tile}^{\Gamma}$.

This proof demonstrates the extensive expressive power of QCTL*. In fact the only formulae that were used were from QCTL, and there was only one propositional quantifier used. It is possible to give a similar proof where no path quantifiers are used, and hence QPTL [18] is undecidable when repeated states are allowed.

5.2 Decidability

The tree semantics are decidable. We do not go into details as the proofs are complicated [5]. The decidability can be shown by expressive equivalence with the language tree of automata [17], though this approach requires careful treatment as the branching degree of the model may not be fixed. In [4] the equivalence between the Rabin tree automata over binary trees and existentially quantified CTL* is shown. This does not relect the full the expressive ability of QCTL* over tree semantics since it does not allow for a varying branching degree.

In [6] the decidability of a similar logic over trees was proven. In this proof it was shown that the formulas of the language could be transformed so the satisfaction of any formula could be decided over binary trees. This approach is also applicable in the case of QCTL*.

To show QCTL* is decidable in the amorphous semantics we extend a method introduced in [9]. We define a new kind of tree automaton refered to as an *amorphous automaton*[1] such that for any formula α of QCTL* an amorphous

[1] Amorphous tree automaton were defined in [12] to act on trees of varying branching degree. As they are a generalization of the construction presented here we will retain the name.

automaton A_α can be constructed that will accept exactly the models of α. The decidability of QCTL* then reduces to the emptiness problem for the automaton, which in turn reduces to the satisfiability problem for CTL*. An amorphous automaton A is given by the tuple $(\Sigma, Q, q_0, \delta, C)$ where

1. $\Sigma = \wp(var(\alpha))$ is an alphabet, where $var(\alpha)$ is the set of the variables occuring in α.
2. $Q = \{q_0, ...q_n\}$ is a set of automaton states, (we refer to states of a model as moments from now on, to avoid confusion).
3. $q_0 \in Q$ is the initial state.
4. $\delta : Q \times \Sigma \longrightarrow \wp(\wp(Q))$ is the transition fuction.
5. $C = \{(L_i, U_i) \mid L_i, U_i \subseteq Q\}_{i=1}^k$ is the Rabin acceptence condition.

We define a run of the automaton A over some model (M, s_0) to be a Q-labelled tree (T, R^T, λ, t_0) along with some function $\mu : T \to M$ such that:

1. $\lambda : T \longrightarrow Q$, (i.e. each node is marked with a single automaton state).
2. $\mu(t_0) = s_0$ and $\lambda(t_0) = q_0$.
3. If $\mu(t) = s$ and $(s, s') \in R$ then there is some $t' \in T$ such that $(t, t') \in R^t$ and $\mu(t') = s'$.
4. If $\mu(t) = s$ and $(t, t') \in R^T$ then there is some $s' \in S$ such that $(s, s') \in S^t$ and $\mu(t') = s'$.
5. There is some set $a \in \delta(\lambda(t), \pi(\mu(t)))$ such that $a = \{\lambda(t') | (t, t') \in R^T\}$, where π' is the projection of π onto the variables of α (i.e. $\pi' : S \to \Sigma$).

Let $r = (T, R^t, \lambda, t_o, \mu)$ be some run of the automaton A over a model (M, s). We say r is an accepting run if for every path b of r there is some $i \leq k$ such that some state $\ell \in L_i$ occurs infinitely often along b and every state $u \in U_i$ occurs only finitely often along b.

For each automaton A we can define a characteristic formula χ_A in QCTL*, such that $M, s \models \chi_A$ if and only if A accepts (M, s). To do this we use a set of variables $P = \{p_0, ...p_n\}$ to represent the automaton states, and define the formula $at\_q_i = p_i \wedge \neg \bigvee_{j \neq i} p_j$ for each state $q_i \in Q$ and $in\_Q' = \bigvee_{q \in Q'} at\_q_i$ and for each subset $Q' \subseteq Q$. For each element $\sigma \in \Sigma$ we define the formula $\overline{\sigma} = \bigwedge\{x | x \in \sigma\} \wedge \bigwedge\{\neg x | x \in var(\alpha) \backslash \sigma\}$. The formula χ_A is given by

$$run = \bigvee_{\sigma \in \Sigma} \bigvee_{q \in Q} \left(at\_q \wedge \overline{\sigma} \wedge \bigvee_{a \in \delta(q,\sigma)} \left(\mathrm{A} \bigcirc in\_a \wedge \bigwedge_{q' \in a} \mathrm{E} \bigcirc at\_a' \right) \right) \qquad (21)$$

$$acc = \mathrm{A} \bigvee_{i=1}^{k} (\Box \Diamond in\_L_i \wedge \Diamond \Box \neg in\_U_i) \qquad (22)$$

$$\chi_A = \exists p_0 ... \exists p_n \, (at\_q_0 \wedge \mathrm{A} \Box run \wedge acc) \, . \qquad (23)$$

Lemma 6. *An amorphous automaton A accepts some model (M, s) if and only if $M, s \models \chi_A$.*

This can be seen by comparision of the semantic defintions (2-8), (9) with the definition of the amorphous automata. Conversely for every formula α of QCTL* there is an automaton A_α that accepts exactly the models that satisfy α. To construct the automaton A_α we first convert α into a formula α' of QCTL, using Lemmas 3 and 4. A_α can then be constructed by induction over complexity of α'. We will give all the constructions for all operators except $\neg$. The complementation construction is double exponential, though it could possibly be optimized, and it has similarities with Klarlund's construction [10]. We let $A_\beta = (\Sigma, Q, q_0, \delta, C)$ and $A_\gamma = (\Sigma, Q^*, q_0^*, \delta^*, C^*)$.

1. Propositions. For $x \in \mathcal{V}$ we define $A = (\Sigma, \{q_0, q_1, q_2\}, q_0, \delta, \{(\{q_1\}, \emptyset)\})$, where for all $\sigma \in \Sigma$ $\delta(q, \sigma) = \{\{q_1\}\}$ if $q = q_0$ and $x \in \sigma$, or $q = q_1$, and $\delta(q, \sigma) = \{\{q_2\}\}$ otherwise.
2. $\beta \vee \gamma$. We define $A = (\Sigma, Q \cup Q^*, q', \delta', C \cup C^*)$, where for all $\sigma \in \Sigma$ $\delta'(q', \sigma) = \delta(q_0, \sigma) \cup \delta(q_0^*, \sigma)$, for all $q \in Q$ $\delta'(q, \sigma) = \delta(q, \sigma)$ and for all $q \in Q^*$ $\delta('q, \sigma) = \delta^*(q, \sigma)$.
3. $\mathrm{E}\bigcirc\beta$. We define $A = (\Sigma, Q \cup \{q_0', q_1'\}, q_0', \delta', C \cup \{(\{q_1\}, \emptyset)\})$, where for all $\sigma \in \Sigma$ $\delta'(q_0', \sigma) = \{\{q_0, q_1'\}\}$, $\delta'(q_1', \sigma) = \{\{q_1'\}\}$ and $\delta'(q, \sigma) = \delta(q, \sigma)$ for $q \in Q$. The construction for $\mathrm{E}\Diamond$ is similar.
4. $\mathrm{A}\Diamond\beta$. We define $A = (\Sigma, Q \cup \{q_0'\}, q_0', \delta', C)$, where for all $\sigma \in \Sigma$ $\delta'(q_0', \sigma) = \{\{q_0, q_0'\}, \{q_0\}, \{q_0'\}\}$ and $\delta'(q, \sigma) = \delta(q, \sigma)$ for $q \in Q$.
5. $\exists x \beta$. We define $A = (\Sigma, Q, q_0, \delta', C)$, where for all $\sigma \in \Sigma$ and all $q \in Q$ $\delta'(q, \sigma) = \delta(q, \sigma \backslash \{x\}) \cup \delta(q, \sigma \cup \{x\})$.

The complementation procedure and proofs of soundness for the above constructions will be given in [5]. To prove the decidability of the satisfiability problem for QCTL* in the amorphous semantics we have to show that we can decide whether or not an amorphous automaton A_α accepts the empty language (i.e. α is unsatisfiable). Rather than constructing such a decision process it is enough to note that α is equivalent to χ_{A_α}. Since χ_{A_α} is an existentially quantified formula of QCTL* we can reason that χ_{A_α} (and hence α) is satisfiable if and only if the unquantified part (i.e. $(at\_q_0 \wedge \mathrm{A}\,\Box run \wedge acc)$ from (21)) is satisfiable in CTL*. Since the CTL* is decidable the decidability of QCTL* follows and we are done.

6 Conclusion

We have defined three sets of semantics for QCTL*: Kripke semantics, tree semantics and amorphous semantics. While the Kripke semantics have been shown to be highly undecidable, there is good reason for further investigation. We have shown that QCTL* can reason about the structure of a model, but we do not yet know to what extent. For example, there is a formula of QCTL that is satisfied in the Kripke semantics by exactly the models that are trees. The variety of structures that are expressible may have applications in the theory of modal logic, or natural language processing.

In the case of the decidable semantics the complexity of the decision processes require further investigation. While there are some well known results in the case of the tree semantics, the amorphous semantics are relatively new and such issues are yet to be examined. The definition of amorphous automata is also of interest in its own right. The expressive power, complexity and applications are all yet to be fully explored.

References

1. M. Abadi and L. Lamport. The existence of refinement mappings. *Theorectical Computer Science*, 82(2):253–284, May 1991.
2. E. Clarke and E. Emerson. Synthesis of synchronization skeletons for branching time temporal logic. In *Proc. IBM Workshop on Logic of Programs, Yorktown Heights, NY*, pages 52–71. Springer, Berlin, 1981.
3. E. Emerson. Alternative semantics for temporal logics. *TCS*, 26, 1983.
4. E. Emerson and A. Sistla. Deciding full branching time logic. *Information and Control*, 61:175 – 201, 1984.
5. T. French. *The theory of branching time logics with quantified propositions.* PhD thesis, Murdoch University, In preparation.
6. Y. Gurevich and S. Shelah. The decision problem for branching time logic. *J. of Symbolic Logic*, 50:668–681, 1985.
7. D. Harel. Effective transformations on infinite trees, with applications to high undecidability, dominoes, and fairness. *J. A.C.M.*, 33(1):224–248, 1986.
8. R. van der Meyden K. Engelhardt and Y. Moses. Knowledge and the logic of local propositions. In *Conf. on Theoretical Aspects of Rationality and Knowledge*, 1998.
9. Yonit Kesten and Amir Pnueli. A complete proof systems for qptl. In *Proceedings, Tenth Annual IEEE Symposium on Logic in Computer Science*, pages 2–12, 1995.
10. N. Klarlund. Progress measures, immediate determinacy, and a subset construction for tree automata. *Annals of Pure and Applied Logic*, 69:243–268, 1994.
11. O. Kupferman. Augmenting branching temporal logics with existential quantification over atomic propositions. In *Computer Aided Verification, Proc. 7th Int. Conference*, pages 325–338, Liege, 1995. Springer-Verlag.
12. O. Kupferman and O. Grumberg. Branching time temporal logic and amorphous tree automata. In *Proceedings of the Fourth Conference on Concurrency Theory*, pages 262–277, Hildesheim, 1993. Springer-Verlag.
13. O. Kupferman and A. Pnueli. Once and for all. In *Proceedings of the Tenth IEEE Symposium on Logic in Computer Science*, San Diego, 1995.
14. R. Milner. A calculus of communicating systems. *Lecture Notes in Computer Science*, 92, 1980.
15. P.Kremer. On the complexity of propositional quantification in intuitionistic logic. *J. of Symbolic Logic*, 62(2):529–544, 1997.
16. A. Pnueli. The temporal logic of programs. In *Proceedings of the Eighteenth Symposium on Foundations of Computer Science*, pages 46–57, 1977.
17. M. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. AMS*, 141:1–35, 1969.
18. A. P. Sistla. *Theoretical Issues in the Design and Verification of Distributed Systems.* PhD thesis, Harvard University, 1983.
19. Albert Visser. Bisimulations, model descriptions and propositional quantifiers. Manuscript, see http://www.citeseer.nj.nec.com/visser96bisimulation.html.