Sublogics of a Branching Time Logic of Robustness

John M[c]Cabe-Dansted, Clare Dixon, Tim French and Mark Reynolds[1a,b,a,a]

[a] *Computer Science and Software Engineering (M002)*
The University of Western Australia
35 Stirling Highway Crawley WA 6009 Perth, Australia
{*john.mccabe-dansted,tim.french,mark.reynolds*}*@uwa.edu.au*
[b] *University of Liverpool*
Department of Computer Science
Ashton Street, Liverpool, L69 3BX, United Kingdom
CLDixon@liv.ac.uk

Abstract

In this paper we study sublogics of RoCTL*, a recently proposed logic for specifying robustness. RoCTL* allows specifying robustness in terms of properties that are robust to a certain number of failures. RoCTL* is an extension of the branching time logic CTL* which in turn extends CTL by removing the requirement that temporal operators be paired with path quantifiers. In this paper we consider three sublogics of RoCTL*. We present a tableau for RoBCTL*, a bundled variant of RoCTL* that allows fairness constraints to be placed on allowable paths. We then examine two CTL-like restrictions of CTL*. Pair-RoCTL* requires a temporal operator to be paired with a path quantifier; we show that Pair-RoCTL* is as hard to reason about as the full CTL*. State-RoCTL* is restricted to State formulas, and we show that we there is a linear truth preserving translation of State-RoCTL into CTL, allowing State-RoCTL to be reasoned about as efficiently as CTL.

Keywords: RoCTL*, Bundles, Tableau, CTL
2010 MSC: 03B70, 68Q17

1. Introduction

RoCTL* [1] is temporal logic for specifying *Robustness*. As with other simpler propositional temporal logics such as the popular linear LTL [2], it can be used to reason about the behaviour of finite state transition systems, and hardware or software systems that can be modelled so. Temporal connectives, such as $\bigcirc$ for next step (which indicates that its argument is true at the next step), and $\mathcal{U}$ for until (a bimodal operator that indicates that one argument is true until the other is satisfied), added to the classical propositional ones, allow us to specify and deduce properties of such systems as they operate over time.

[1] Corresponding Author, Ph: (+61 8) 6488 2281, Fax: (+61 8) 6488 1089

The branching logics CTL [3] and CTL* [4] add the ability to reason about alternative possible steps. RoCTL*, goes even further, and adds basic deontic operators to consider what the system should do, versus what it may do if some transitions goes wrong. The robustness operators, allow us to consider overall behaviour of the system if only a limited number of transitions go wrong.

Thus, RoCTL* has three path quantifiers: AllPaths (**A**), Obligatory (**O**), and Robustly (**R**). From these operators the duals ExistPath (**E**), Permissable (**P**) and Prone (**Я**) are defined. This extends the Full Computation Tree Logic (CTL*) whose only path quantifier is **A** (and its dual **E**).

As an example, consider the following specification in RoCTL* which uses the usual LTL temporal connective $\Box$ for always. The formula $\mathbf{OR}\Box(p \rightarrow Fq)$ says that it is obligatory that robustly always a p request is eventually followed by a q grant. This means that along every failure-free behaviour, even if there is one failure deviating from such a behaviour then any p event is eventually followed some time in the future by a q event. In earlier work [1, 5] we have showed that this language can comfortably and effectively express situations with conflicting resource requirements, temporary communication failures and contrary-to-duty obligations (what a system should do when it hasn't done what it should do).

The logic RoCTL* was introduced in [1] but more detailed technical investigation of the language and it computational properties was reported in [6]. That paper showed that the full RoCTL* is decidable. It provided translations of RoCTL* into Quantified Computation Tree Logic (QCTL*) [7] and Full Computation Tree Logic (CTL*), either of which can be used to decide RoCTL*. However, it was also shown that translating RoCTL* formulas into CTL* results in at least a singly exponentially blowup per alternation with the **R** operator. Thus, there is no elementary reduction of RoCTL* into tree automata or CTL*. Nor is there any known elementary decision procedure.

In Section 5 below we will argue that RoCTL* is not significantly harder than CTL* to decide, in practice. However, for many purposes, decision procedures even for CTL* are too computationally demanding and, since RoCTL* is a conservative extension of CTL*, it is clear that RoCTL* is at least as hard to decide as CTL*.

We clearly have a problem with advocating direct use of the full RoCTL* language for applications. This leads us to make the common move of considering restricted sub-languages of RoCTL* and wondering if some sufficiently expressive sub-languages may have easier decision problems. Thus the current paper considers sub-logics of RoCTL* from the point of view of being amenable to automated reasoning as well as their expressiveness.

We will investigate semantic and syntactic sublogics of RoCTL*. RoBCTL* is a so-called *bundled* variant of RoCTL* : *limit closure* is not valid in RoBCTL*, while it is valid in RoCTL*. Informally the limit closure property states that if we have an infinite sequence of states σ such that for all n the first n states form an allowed path, then the infinite sequence σ forms an allowed path. This property ensures that all paths through the structure are allowed in CTL*, while in BCTL* we are limited to some bundle of paths which may or may not

include all paths through the structure. We will also investigate two CTL-like restrictions of RoCTL*.

We explain the concept of a bundled variant of a branching time logic in the following two sections. However, the basic idea is that like the bundled variant, BCTL* [8] (also known as ∀LTFC [9]), of CTL*, not all paths through the structure need to be allowed to contribute to the semantics: the bundle is the set of paths that are allowed. The bundled variant has a set of valid formulas which is a subset of those of RoCTL*. We investigate RoBCTL* for two reasons. Firstly, because it is generally easier to find tableaux for bundled logics; for example, note that the tableau for BCTL* [8] was found before the tableaux for CTL* [10, 11] and that the latter were either much more complicated or relied on complex parity game solvers. Secondly, as shown in [12], bundled logics can expressive fairness constraints which are commonly needed for specifications. In fact, in a robust system there is often an implicit fairness constraint. For example, a normally functioning network will lose a packet with some probability. However, (almost surely) given enough retransmits the packet will eventually get through. The obvious way to specify this in CTL* or RoCTL* results in a contradiction while RoBCTL* allows models that disallow paths that do not satisfy fairness constraints.

In Section 5 we will present a tableau based decision procedure for the bundled variant RoBCTL* of RoCTL*, and show that under certain reasonable restrictions on the nesting of operators, the tableau is a little closer to being usable.

The other sub-languages we investigate are syntactical restrictions of RoCTL* using the original semantics. Even if an elementary decision procedure is found for RoCTL*, it is clear that it will be at least as complex as CTL*. Despite the expressive power of CTL*, the less expressive CTL is frequently used. This is because the decision problems for CTL are much easier. The CTL* decision problems are exponentially harder [13] than the CTL decision problems [14], the satisfiability of CTL* formulas is 2-EXPTIME complete while testing the satisfiability of CTL formulas can be performed in singly exponential time; model checking for CTL* is singly exponential in the length of the formula, but for CTL we can model check in polynomial time.

We will examine two CTL-like restrictions of RoCTL*. Arguably the most intuitive CTL-like restriction of RoCTL* is to require (like in CTL) that each of the LTL operators ($\bigcirc$ and $\mathcal{U}$) be paired with a path quantifier (**A**, **O**, or **R**). This restriction is called Pair-RoCTL. However, it will be shown in Section 6 that this restriction of RoCTL* has the same expressivity as the full RoCTL* and it will similarly be shown that Pair-RoCTL is as at least as hard to decide as CTL*. The difficulty in reasoning with Pair-RoCTL comes from the fact that $\mathbf{R}\phi$ is not a state formula. That is the truth of $\mathbf{R}\phi$ may depend on which future eventuates, not just the current state. Thus another restriction of RoCTL* called State-RoCTL will be examined. This restriction instead pairs the LTL operators with a non-empty sequence of path operators which must form a state formula. Hence $\mathbf{OR}(\phi\mathcal{U}\psi)$ is a State-RoCTL formula but $\mathbf{R}(\phi\mathcal{U}\psi)$ is not. It will be shown in Section 7 that we can use standard CTL decision procedures

to reason about State-RoCTL, although we need to inspect the internal state of the CTL model checker of [15] to achieve the same order of complexity as the original. Thus although Pair-RoCTL is an intuitive definition of a CTL-like restriction of RoCTL*, State-RoCTL is more CTL-like in complexity than Pair-RoCTL.

Every property that can be expressed in State-RoCTL can be expressed in CTL, yet State-RoCTL can naturally express interesting robustness properties. CTL is significantly less expressive than CTL* and RoCTL*. For example, CTL cannot express fairness, the property that if a process is ready to run infinitely often, then it will be chosen to run infinitely often [4]. Nevertheless State-RoCTL is expressive enough to capture some interesting properties of RoCTL*, such as direct alternations between $\mathbf{R}$ and $\mathbf{Я}$. Additionally, the truth preserving translation of State-RoCTL into CTL results in a formula that may be exponentially longer than the original. A linear translation will be given that is both satisfiability preserving and computationally efficient, providing efficient decision procedures for State-RoCTL, but this translation adds atoms and is not expressively equivalent.

With current technology State-RoCTL is tractable for larger problems than CTL*; we have given simple translations into CTL and there are a number of easily available and fast decision procedures for CTL. The resolution procedure CTL-RP [16] can solve the decision problems relating to (coordinated attack) problem presented in [1] in under 2 seconds. The Tableau Work Bench [17] also has little difficulty with translations of our examples into CTL, determining that the specification for feeding a cat in Example 15 is satisfiable almost instantaneously. We also note that dedicated CTL solvers tend to be more efficient [18] than the CTL* prover proposed by [11].

The main contributions of this paper are: (Section 5) a tableau for RoBCTL*, which terminates but can require a non-elementary amount of time; (Section 6) a proof that Pair-RoCTL is at least as hard to reason with as CTL*; and (Section 7) a polynomial reduction of State-RoCTL to CTL which demonstrates that out of the sublogics considered in this paper, State-RoCTL is the easiest sublogic of RoCTL* to reason about.

The core results in this paper have been presented in abridged form at conferences [19, 20]. For a more thorough discussion of RoCTL* and related logics, see the PhD thesis [5].

2. Motivation of Bundled Logics

RoBCTL* is a bundled variant of RoCTL*. Limit closure is not valid in RoBCTL* (or BCTL*), while it is valid in RoCTL*. We will also investigate CTL-like restrictions of RoCTL*.

We investigate this bundled variant for two reasons. Firstly, because it is easier to find tableaux for bundled logics. As such, when investigating whether a tableau would result in an elementary decision procedure for RoCTL*, RoBCTL* is an obvious starting point. Secondly, we already have presented decision procedures for RoCTL* in [6]. Since RoBCTL* can deal with systems

that are not limit closed, finding a decision procedure for RoBCTL* is more useful than a second decision procedure for RoCTL*. Further we note that in a robust system there is often an implicit fairness constraint. A normally functioning network will lose a packet with some probability. However, (almost surely) given enough retransmits the packet will eventually get through. The obvious way to specify this in CTL* or RoCTL* results in a contradiction while RoBCTL* allows models that disallow paths that do not satisfy fairness constraints.

In Section 5 we will present a tableau based decision procedure for the bundled variant RoBCTL* of RoCTL*. We find that this tableau based decision procedure is also non-elementary but we will see that this is only an issue when we have alternations with Robustly that are unbroken by AllPath or Obligatory operators; this is not a problem for the examples given in Section 4. Also, this means that this tableau is elementary (at worst 3-exponential) when dealing with the fragment of RoBCTL*/RoCTL* that excludes until.

Two features of the RoBCTL* tableau, not present in the BCTL* tableau, are the ability to deal with path quantifiers that do not result in state formulas and the ability to deal with eventualities that change over time. With QCTL* we implemented the Robustly operator by marking the current path with a variable (quantified atom). Attempts to extend the BCTL* tableau to allow marked paths, or bundles of deviating paths resulted in non-finite tableaux. We have instead used a successor function that adds enough formulas to the closure that we can handle deviations without having to distinguish paths in the tableau. For example, the successor of "Next ϕ" is "ϕ". This can make the closure set large, although the size of the closure set will be elementary if the number of alternations involving the Robustly operator are bounded.

We must also be able to deal with eventualities that change over time. In BCTL* the only eventuality is of the form "ϕ Until ψ", which remains unchanged until it is resolved by ψ occurring. In RoBCTL* we have "Eventually a path will deviate and along that path ϕ will be hold", at the next step this becomes "Eventually a path will deviate and the temporal successor of ϕ will hold".

3. RoBCTL*, RoCTL*, CTL* and CTL

In this section we define RoBCTL*, RoCTL*, CTL* and CTL. We first provide some basic definitions, starting with our set of variables.

The intuition behind RoCTL* is that there are two types of transitions: failure transitions and success transitions. A failure transition represents a transition that only occurs when the system exhibits some sort of failure, for example losing a packet. A success transition represents a transition that can occur when no such failure occurs. This was how RoCTL* was defined in French et al. [1]. In the paper we compare RoCTL* to CTL*. To do this we need to define RoCTL* on CTL* structures. As CTL* only has a single transition relation, we will instead use a special atom $\mathbf{v}$, used to indicate that the last transition was a failure. To convert an original RoCTL* structure to this new CTL* based RoCTL* structure, we can unroll the structure into a tree so that

every state has at most one possible parent, and add the **v** atom to the states reached by failure transitions. For a full proof of equivalence of these definitions see M^{c}Cabe-Dansted [5].

Definition 1. We let $\mathbb{V}$ be our set of variables. The set $\mathbb{V}$ contains a special variable **v**. A valuation g is a map from a set of worlds S to the power set of the variables.

The statement $p \in g(w)$ is to be read as "the variable p is true at world w".

The **v** atom will be used to define failing transitions. Informally it may be possible to enter a state labelled with **v**, but it is forbidden to do so; entering such a state will be considered a failure.

As is normal we say a binary relation is serial if every element has a successor.

Definition 2. We say that a binary relation R on S is serial (total) if for every a in S there exists b in S such that aRb.

While in some logics the truth of formulas depends solely on the current world, the truth of CTL* and BCTL* (and hence RoCTL* and RoBCTL*) may depend on which future eventuates. These futures are represented as infinitely long (full) paths through the structure. For this reason, we provide a formal definition of fullpaths.

Definition 3. We call an ω-sequence $\sigma = \langle w_0, w_1, \ldots \rangle$ of worlds a fullpath iff for all non-negative integers i we have $w_i R w_{i+1}$. For all i in $\mathbb{N}$ we define $\sigma_{\geq i}$ to be the fullpath $\langle w_i, w_{i+1}, \ldots \rangle$, we define σ_i to be w_i and we define $\sigma_{\leq i}$ to be the sequence $\langle w_0, w_1, \ldots, w_i \rangle$.

We now define allowable sets of fullpaths called bundles.

Definition 4. We say that a set of fullpaths Π is suffix closed iff for all $\pi \in \Pi$ and positive integers i we have $\pi_{\geq i} \in \Pi$. We say that a set of fullpaths Π is fusion closed iff for any pair of non-negative integers i, j and fullpaths $\sigma, \pi \in \Pi$ such that $\sigma_i = \pi_j$ we have $\sigma_{\leq i} \cdot \pi_{\geq j+1} \in \Pi$. We say a set of fullpaths B through (S, R) is a bundle of (S, R) iff B is suffix and fusion closed.

We now provide a definition of a structure.

Definition 5. A BCTL-structure $M = (S, R, g, B)$ is a 4-tuple containing a set of worlds S, a serial binary relation R on S, a valuation g on the set of worlds S and a bundle B on (S, R). We say M is a RoBCTL-structure if every world has an allowed successor, that is for all $x \in S$ there exists $(x, y) \in R$ such that $\mathbf{v} \notin g(y)$.

We now define the property of failure-freeness. This means that, in the future, no failing transitions are taken. Informally, a failure-free fullpath represents a perfect future. This is somewhat similar to a very simple traditional deontic logic called Standard Deontic Logic (SDL). SDL is the modal logic K with the addition of seriality on the Krikpe semantics, meaning that what is morally necessary (or obligatory) is also permissible. Whereas the Obligatory operator in Standard Deontic Logic quantifies over acceptable worlds, the Obligatory operator we will define quantifies over failure-free fullpaths.

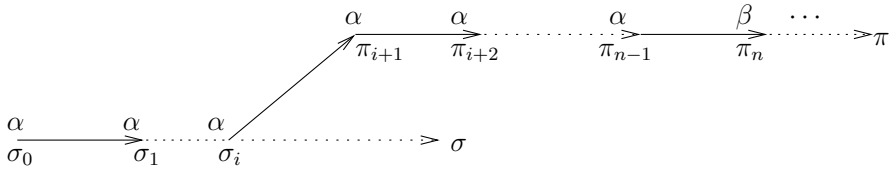

Figure 1: Example of Deviation π satisfying $\alpha \mathcal{U} \beta$

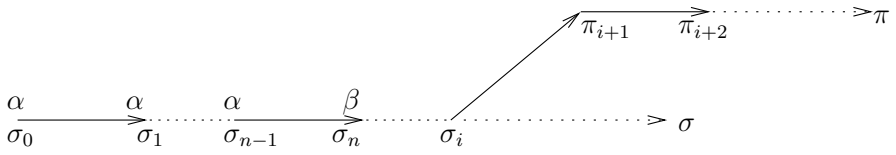

Figure 2: Path σ satisfying $\alpha \mathcal{U} \beta$, and π which deviates after β occurs

Definition 6. We say that a fullpath σ is failure-free iff for all $i > 0$ we have $\mathbf{v} \notin g(\sigma_i)$. We define $\delta^\omega(w)$ to be the set of all fullpaths starting with world w in B and $\delta^0(w)$ to be the set of all failure-free fullpaths starting with w in B. We call a BCTL-structure a RoBCTL-structure iff $\delta^0(w)$ is non-empty for every $w \in S$.

We choose the notation above as δ^ω includes paths with potentially an infinite number of failures and δ^0 only includes paths without any failures.

We will now define deviations. We will use the notation δ^+ as informally deviations represent the possibility of adding an additional failure to some step i along a path. After i we follow a different path, and we allow only a single failure not on the existing path so no failures occur after $i+1$. Deviations are intended to represent possible failures we may wish to be able to recover from, and if our system is robust to failures we also want it to be robust in the face of correct transitions. For this reason we allow the new transition added at step i to be a success as well as a failure.

Definition 7. For two fullpaths σ and π we say that π is an i-deviation from σ iff $\sigma_{\leq i} = \pi_{\leq i}$ and $\pi_{\geq i+1} \in \delta^0(\pi_{i+1})$. We say that π is a deviation from σ if there exists a non-negative integer i such that π is an i-deviation from σ. We define a function δ^+ from fullpaths to sets of fullpaths such that where σ and π are fullpaths, π is in $\delta^+(\sigma)$ iff π is a deviation from σ.

For an example of a formula being satisfied along a deviation, see Figure 1. In this example $\alpha \mathcal{U} \beta$ is satisfied along the deviation as α is satisfied along the original path up to σ_i, where the deviation occurs, and then $\alpha \mathcal{U} \beta$ continues to be satisfied along the deviation. More trivial examples are also possible. In 2 we see that we reach β before the deviation occurs, so $\alpha \mathcal{U} \beta$ is likewise satisfied along the original path.

We see that $\delta^0(\sigma_0) \subseteq \delta^+(\sigma) \subseteq \delta^\omega(\sigma_0)$. Where p varies over $\mathbb{V}$, we define RoCTL* (and equivalently RoBCTL*) formulas according to the following

abstract syntax

$$\phi := p \,|\, \neg\phi \,|\, (\phi \wedge \phi) \;|\; (\phi \mathcal{U} \phi) \,|\, \bigcirc\phi \,|\, \mathbf{A}\phi \,|\, \mathbf{O}\phi \,|\, \mathbf{R}\phi \;.$$

A formula that begins with $\mathbf{A}$, $\neg\mathbf{A}$, $\mathbf{O}$, $\neg\mathbf{O}$, p or $\neg p$ is called a state formula. To allow for alternative semantic interpretations of RoBCTL* and RoCTL* (such as in [21]), we do not consider a formula that explicitly contains $\mathbf{v}$ to be a RoBCTL* formula, although our results work equally well for such formulas. We call the logic without this restriction RoBCTL$^*_{\mathbf{v}}$. The $\neg$, $\wedge$, $\bigcirc$, $\mathcal{U}$ and $\mathbf{A}$ are the familiar "true", "not", "and", "next", "until" and "all paths" operators from CTL.

Definition 8. We say that a pair of formulas ϕ, ψ are equivalent ($\phi \equiv \psi$) iff for all structures M and paths σ through M:

$$M, \sigma \vDash \phi \iff M, \sigma \vDash \psi \;.$$

We now define the abbreviations in terms of the base operators as follows: $\bot \equiv (p \wedge \neg p)$, $\top \equiv \neg\bot$, $\phi \vee \psi \equiv \neg(\neg\phi \wedge \neg\psi)$, $\Diamond\phi \equiv (\top \mathcal{U} \phi)$, $\Box\phi \equiv \neg\Diamond\neg\phi$, $\phi \mathcal{W} \psi \equiv (\phi \mathcal{U} \psi) \vee \Box\phi$, $\mathbf{E}\phi = \neg\mathbf{A}\neg\phi$, $\phi \rightarrow \psi \equiv (\neg\phi \vee \psi)$ and $\phi \leftrightarrow \psi \equiv (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$ are defined as in CTL*. As with Standard Deontic Logic (SDL, also known as KD or D) Wright [22] logic, we define $\mathbf{P}\phi \equiv \neg\mathbf{O}\neg\phi$. Finally, we define the dual $\mathbf{Я}$ of $\mathbf{R}$ as the abbreviation $\mathbf{Я}\phi \equiv \neg\mathbf{R}\neg\phi$. We call the $\mathbf{O}$, $\mathbf{P}$, $\mathbf{R}$, $\mathbf{Я}$ operators Obligatory, Permissible, Robustly and Prone respectively.

We define truth of a RoBCTL* formula ϕ on a fullpath $\sigma = \langle w_0, w_1, \ldots \rangle$ in a RoBCTL-structure M recursively as follows:

$$\begin{array}{lll}
M, \sigma \vDash \bigcirc\phi & \text{iff} & M, \sigma_{\geq 1} \qquad\qquad\qquad\quad \vDash \phi \\
M, \sigma \vDash \phi \mathcal{U} \psi & \text{iff} & \exists i \in \mathbb{N}, \text{ s.t. } M, \sigma_{\geq i} \qquad \vDash \psi \text{ and;} \\
 & & \forall j \in \mathbb{N}, j < i \implies M, \sigma_{\geq j} \quad \vDash \phi \\
M, \sigma \vDash \mathbf{A}\phi & \text{iff} & \forall \pi \in \delta^{\omega}(\sigma_0), M, \pi \qquad \vDash \phi \\
M, \sigma \vDash \mathbf{O}\phi & \text{iff} & \forall \pi \in \delta^{0}(\sigma_0), M, \pi \qquad \vDash \phi \\
M, \sigma \vDash \mathbf{R}\phi & \text{iff} & \forall \pi \in \delta^{+}(\sigma), M, \pi \qquad \vDash \phi \text{ and;} \\
 & & M, \sigma \qquad\qquad\qquad\quad \vDash \phi \;.
\end{array}$$

The definitions for p, $\neg$ and $\wedge$ are as we would expect from classical logic.

$$\begin{array}{lllll}
M, \sigma \vDash p & \text{iff} & p & \in g(\sigma_0) \\
M, \sigma \vDash \neg\phi & \text{iff} & M, \sigma & \nvDash \phi \\
M, \sigma \vDash \phi \wedge \psi & \text{iff} & M, \sigma & \vDash \phi \text{ and;} \\
 & & M, \sigma & \vDash \psi \;.
\end{array}$$

The intuition behind the $\mathbf{R}$ operator is that it quantifies over paths that could result if a single error was introduced; the deviations have at most one failure not on the original path, and they are identical to the original path until this failure occurs.

Definition 9. We say that a function τ from formulas to formulas is satisfiability preserving iff: $\exists M, \sigma$ s.t. $M, \sigma \vDash \tau(\phi) \iff \exists M', \sigma'$ s.t. $M', \sigma' \vDash \tau(\phi)$. We say that a function τ from formulas to formulas is truth preserving iff for all M, σ and ϕ it is the case that $M, \sigma \vDash \phi \iff M, \sigma \vDash \tau(\phi)$.

Note that the existence of a satisfiability preserving translation is usually not interesting unless it is efficient. For example, we could translate all satisfiable formulas into $\top$ and unsatisfiable formulas into $\bot$.

Given that traditional modal logics define truth at worlds, instead of over paths, many important properties of modal logics assume such a definition of truth. When dealing with those properties we can use the following definition of truth of RoBCTL* formulas at worlds.

Definition 10. A RoBCTL* formula is true at a world if it is true on some path leading from that world, or more formally:

$$M, w \vDash \phi \text{ iff } \exists \pi \text{ s.t. } \pi_0 = w : M, \pi \vDash \phi \ .$$

For the purposes of this paper it is most natural to define RoCTL* as RoBCTL* with an additional restriction of the models.

Definition 11. A RoCTL-structure (S, R, g, B) is a RoBCTL-structure where B contains every possible path through R. Likewise, a CTL-structure is a BCTL-structure where B contains every possible path through R.

Note that the original definition of RoCTL-structures in [6] did not have B, but was rather a 3-tuple (S, R, g). Since B is determined by R in a RoCTL-structure, we can consider (S, R, g) as an abbreviation of (S, R, g, B). This leaves our definition of RoCTL* equivalent to that in [6].

We define CTL* to be the syntactic restriction of RoCTL* without **O** or **R**.

Definition 12. Where p varies over $\mathbb{V}$, we define CTL* formulas according to the following abstract syntax:

$$\phi := p \mid \neg\phi \mid (\phi \wedge \phi) \mid (\phi \mathcal{U} \phi) \mid \bigcirc\phi \mid \mathbf{A}\phi \ .$$

CTL* was proposed as an extension to CTL, which in turn was proposed as an extension of UB Logic that added an until operator Emerson and Halpern [14]. However, here it is more convenient to define CTL as a syntactic restriction of CTL*, with the following syntax Emerson and Clarke [3]:

$$\phi ::= p \mid \neg\phi \mid (\phi \wedge \phi) \mid \mathbf{A}(\phi \mathcal{U} \psi) \mid \mathbf{E}\bigcirc\phi \mid \mathbf{E}(\phi \mathcal{U} \psi) \ .$$

In CTL, we treat $\mathbf{A}\bigcirc\phi$, $\mathbf{A}\Diamond\phi$ and $\mathbf{A}(\phi \mathcal{W} \psi)$ as abbreviations of $\neg\mathbf{E}\bigcirc\neg\phi$, $\mathbf{A}(\top \mathcal{U} \phi)$ and

$$\neg\mathbf{E}((\neg\psi) \mathcal{U} (\neg\phi \wedge \neg\psi)) \ ,$$

respectively. Likewise, in CTL, we treat $\mathbf{E}\Diamond\phi$ and $\mathbf{E}(\phi \mathcal{W} \psi)$ as abbreviations of $\mathbf{E}(\top \mathcal{U} \phi)$ and $\neg\mathbf{A}(\neg\psi \mathcal{U} (\neg\phi \wedge \neg\psi))$. Finally, we define $\mathbf{A}\Box\phi$ and $\mathbf{E}\Box\phi$ as abbreviations of $\neg\mathbf{E}\Diamond\neg\phi$ and $\neg\mathbf{A}\Diamond\neg\phi$.

The following lemma will not be used in any proofs. It is included to make the subtle distinction between bundled verses unbundled logics more clear and in particular when they are interchangeable.

Lemma 13. *Given a formula ϕ that does not contain $\mathcal{U}$ (or the abbreviations $\Diamond$ and $\Box$ that use $\mathcal{U}$), ϕ will be satisfiable in RoBCTL* iff it is satisfiable in RoBCTL*.*

Proof. It is clear that limit closure only has an effect on infinitely long paths. For a more detailed proof, see the Section 3.3.4 of the PhD thesis M^{c}Cabe-Dansted [5]. □

We now define some operators for comparing formulas.

Definition 14. For any pair of formulas (ϕ, ψ), we say that $\phi \sqsubseteq \psi$ iff ϕ is a subformula of ψ. We use $|\cdots|$ for length. As normal where S is a set $|S|$ is the number of elements of S. For a formula ϕ, we define $|\phi|$ to be the total number of occurrences of symbols in the representation of ϕ excluding parentheses. For example, $|(p \wedge p)|$ is three as p occurs twice and $\wedge$ occurs once.

4. Examples

In this section we present some motivating examples for RoCTL*. For more examples, see the paper that introduced RoCTL* M^{c}Cabe-Dansted et al. [23], and the thesis M^{c}Cabe-Dansted [5].

The following example is taken from M^{c}Cabe-Dansted et al. [23]. It has been adapted to only use formulas in the fragment State-RoCTL* of RoCTL* that will be introduced in Section 7, and is used here as an example of a RoCTL* model that can be expressed in that fragment.

Example 15. We have a cat that does not eat the hour after it has eaten. If the cat bowl is empty we might forget to fill it. We must ensure that the cat never goes hungry, even if we forget to fill the cat bowl one hour. At the beginning of the first hour, the cat bowl is full. We have the following atoms:

b "The cat bowl is full at the beginning of this hour"

d "This hour is feeding time"

We can translate the statements above into RoCTL* statements:

1. $\mathbf{A}\Box(d \rightarrow \mathbf{A}\bigcirc\neg d)$: If this hour is feeding time, the next is not.
2. $\mathbf{A}\Box((d \vee \neg b) \rightarrow \mathbf{E}\bigcirc\neg b)$: If it is feeding time or the cat bowl was empty, then the bowl could be empty at the next step.
3. $\mathbf{A}\Box((\neg d \wedge b) \rightarrow \mathbf{A}\bigcirc b)$: If the bowl is full and it is not feeding time, the bowl will be full at the beginning of the next hour.
4. $\mathbf{OR}\Box\,(d \rightarrow b)$: It is obligatory that, if at most one failure occurs, it is always the case that the bowl must be full at feeding time.
5. b: The cat bowl starts full.

In (2) above note that $\mathbf{E}\bigcirc\neg b$ would be equivalent $\mathbf{Я}\bigcirc\neg b$.

Example 16. We define a system that will attempt to reach a safe state (represented by s) and warn (w) the user if the system enters an unsafe state ($\neg s$). The intuition is that if an error (e) is sufficiently serious then the system should consider any state where the error could occur within a single step to be unsafe.

1. $\mathbf{A}\Box\mathbf{O}\bigcirc s$: The system should always ensure that the system reaches a safe state by the next step.
2. $\mathbf{A}\Box(s \rightarrow \bigcirc\neg e)$: If the system is in a safe state an error e will not occur at the next step.
3. $s \wedge \neg e$: The system starts in a safe state with no error.
4. $\mathbf{A}\Box(\neg s \rightarrow \bigcirc w)$: If the system is in an unsafe state, the system will warn the user at the next step.

We may prove that if an error e almost occurs (that is a single extra violation could cause e to become true) the system will finally warn the user, that is $\mathbf{O}(\text{Я}\Diamond e \rightarrow \Diamond w)$.

5. A Tableau for RoBCTL*

Here we define a tableau RoBCTL-TAB for deciding RoBCTL* and RoBCTL*$_{\mathbf{v}}$. This tableau is an extension of Reynolds' Reynolds [8] tableau for BCTL*. As with tableaux for other temporal logics (see Reynolds [8] for a fuller discussion), this tableau is not a tree. The traditional tree-based method for constructing a tableau is to start with a single node and then build up the tableau by adding leaves. This tableau technique begins by adding all possible nodes and then acts like a sculptor, removing nodes but never adding them. Once we have finished removing nodes we find a model of the formula, provided that it is satisfiable.

An issue with branching temporal logics is that formulas are evaluated over paths rather than at worlds. As with Reynolds [8], we handle this by using hues and colours. The hues correspond to paths, and are sets of formula in the closure set. The colours represent worlds and are sets of hues. The worlds are represented as the collection of paths that could start at that world. The nodes of the tableau are colours, the edges are added between any pair of nodes that satisfy a temporal successor relation.

Definition 17. We define logical operations on sets of formulas as follows:

$$(S * T) = \{\varepsilon : \exists_{\phi \in S}\exists_{\psi \in T} \text{ s.t. } \varepsilon = (\phi * \psi)\} \text{ where } * \in \{\mathcal{U}, \wedge\}$$
$$*S = \{\varepsilon : \exists_{\phi \in S} \text{ s.t. } \varepsilon = *\phi\} \text{ where } * \in \{\neg, \mathbf{R}, \mathbf{A}, \bigcirc, \mathbf{O}\}$$

We define the abbreviations $\text{Я}, \mathbf{E}, \mathbf{P}, \vee, \rightarrow, \leftrightarrow$ on sets similarly.

Definition 18. We define a formula translation function Ξ as follows: let $\Psi = [\psi_0, \psi_1, \ldots]$ be an enumeration of all RoBCTL* formulas such that shorter formulas appear before longer formulas. For any RoCTL* formula ϕ we let $\Xi(\phi)$ be ψ_i where i is the smallest non-negative integer such that ϕ is equivalent to ψ_i under classical logic taking all subformulas with non-classical operators of

highest precedence as atoms. Likewise we define Ξ on sets of formulas such that $\Xi(\Phi) = \{\Xi(\phi) : \phi \in \Phi\}$.

A trival example of use of Ξ is that $\Xi(p \wedge p) = p$.

Note that, if we were implementing tableau based procedure we may want to define $\Xi(\phi)$ differently. All that is required is that $\Xi(\phi) = \Xi(\psi)$ iff ϕ is equivalent to ψ under classical logic taking all subformulas with non-classical operators of highest precedence as atoms. For example, to improve human understanding of the tableau, we may wish to choose Ξ such that it maximises the number of cases where $\Xi(\phi) = \phi$. Alternatively we may optimise the tableau by choosing a Ξ that is easier to compute; however, note that as there are only a singly exponential number of formulas of a given length, and even the tableau for BCTL* is doubly exponential, the complexity of computing the Ξ from Definition 18 will not affect the complexity results relating to this tableau.

We will now define a set of formula translation functions. The intention is that for any formula ϕ, fullpath σ through some structure M, then where a is the set of state formulas true at σ_0 we have $M, \sigma \vDash \phi \iff M, \sigma_{\geq 1} \vDash \bigcirc_a^{-1}(\phi)$.

Definition 19. For any set of state formulas a, we define a formula translation function $\bigcirc_a^{-1}$.

$$
\begin{aligned}
\bigcirc_a^{-1}(\phi \mathcal{U} \psi) &= \left(\bigcirc_a^{-1}(\phi) \wedge (\phi \mathcal{U} \psi)\right) \vee \bigcirc_a^{-1}(\psi) \\
\bigcirc_a^{-1}(\neg\phi) &= \neg \bigcirc_a^{-1}(\phi) \\
\bigcirc_a^{-1}(\bigcirc\phi) &= \phi \\
\bigcirc_a^{-1}(\phi \wedge \psi) &= \bigcirc_a^{-1}(\phi) \wedge \bigcirc_a^{-1}(\psi) \\
\bigcirc_a^{-1}(p) &= \begin{cases} \bot & \text{if } p \notin a \\ \top & \text{if } p \in a \end{cases} \\
\bigcirc_a^{-1}(\mathbf{A}\phi) &= \begin{cases} \bot & \text{if } \mathbf{A}\phi \notin a \\ \top & \text{if } \mathbf{A}\phi \in a \end{cases} \\
\bigcirc_a^{-1}(\mathbf{O}\phi) &= \begin{cases} \bot & \text{if } \mathbf{O}\phi \notin a \\ \top & \text{if } \mathbf{O}\phi \in a \end{cases} \\
\bigcirc_a^{-1}(\mathbf{R}\phi) &= \begin{cases} \bot & \text{if } \mathbf{A}\bigcirc\mathbf{O}\Xi\left(\bigcirc_a^{-1}(\phi)\right) \notin a \\ \mathbf{R}\Xi\left(\bigcirc_a^{-1}(\phi)\right) & \text{otherwise} \end{cases}
\end{aligned}
$$

To determine the required closure set, we will now define $\bigcirc^{-1}$, $\bigcirc^{-i}$ and $\bigcirc^{\star}$ in terms of $\bigcirc_a^{-1}$.

Definition 20. We define a function $\bigcirc^{-1}$ from sets of formulas to sets of formulas as follows: given a set of formulas Φ, a formula ψ is a member of $\bigcirc^{-1}(\Phi)$ iff there exists $\phi \in \Phi$ and a set of state formulas a such that $\psi \in \bigcirc_a^{-1}(\phi)$.

Definition 21. We define $\bigcirc^{-i}$ recursively as $\bigcirc^{-i}(\Phi) = \bigcirc^{-1}\left(\bigcirc^{1-i}(\Phi)\right)$ and $\bigcirc^{0}(\Phi) = \Phi$. Let $\bigcirc^{\star}(\Phi)$ be the normalised closure of a set of formulas Φ under $\bigcirc^{-1}$. That is, $\phi \in \bigcirc^{\star}(\Phi)$ iff there exists a non-negative integer i such that $\phi \in \Xi\left(\bigcirc^{-i}(\Phi)\right)$.

For example, $\bigcirc^\star(\{\bigcirc\bigcirc p\}) = \{\bigcirc\bigcirc p, \bigcirc p, p, \bot, \top\}$. Although $\bigcirc^\star$ and $\bigcirc^{-1}$ are finite, they can become very large. See Section 5.2 for a detailed discussion of cardinality.

Definition 22. Let $\gamma = \bigcirc\Box\neg\mathbf{v}$ represent the statement "this path is failure-free"

Note that γ is not a RoBCTL* formula, because it contains $\mathbf{v}$; it is a RoBCTL$^*_{\mathbf{v}}$ formula.

Definition 23. The closure $\mathbf{cl}\phi$ of the formula ϕ is defined as the smallest set that satisfies the four following requirements:

1. $\mathbf{cl}\phi \supseteq \{\phi, \gamma\}$
2. For all $\psi \in \mathbf{cl}\phi$, if $\alpha \sqsubseteq \psi$ then $\delta \in \mathbf{cl}\phi$.
3. For all $\psi \in \mathbf{cl}\phi$, $\neg\psi \in \mathbf{cl}\phi$ or there exists α such that $\psi = \neg\alpha$ and $\alpha \in \mathbf{cl}\phi$.
4. If $\mathbf{R}\psi \in \mathbf{cl}\phi$ then $\mathbf{cl}\phi \supseteq \mathbf{R}\bigcirc^\star(\psi)$ and $\mathbf{cl}\phi \supseteq \mathbf{A}\bigcirc\mathbf{O}\bigcirc^\star(\psi)$.

Recall that we have defined logical operations on sets of formulas (Definition 17). Thus $\mathbf{A}\bigcirc\mathbf{O}\bigcirc^\star(\psi)$ represents the set of formulas that results when each element of $\bigcirc^\star(\psi)$ is prefixed with $\mathbf{A}\bigcirc\mathbf{O}$.

The requirement (4) above is required to ensure that the successor formulas from Definitions 20 and 19 are included in the closure set.

Definition 24 (MPC)**.** We say that $a \subseteq \mathbf{cl}\phi$ is Maximally Propositionally Consistent (MPC) iff for all $\alpha, \beta \in a$

(M1) if $\beta = \neg\alpha$ then $\beta \in a$ iff $\alpha \notin a$,

(M2) if $\alpha \wedge \beta \in \mathbf{cl}\phi$ then $(\alpha \wedge \beta) \in a \leftrightarrow (\alpha \in a$ and $\beta \in a)$

A hue is roughly speaking a set of formulae that could hold along a single fullpath. Note though that a hue is underspecified, as $\{\mathbf{A}\bigcirc p, \mathbf{E}\bigcirc\neg p, \ldots\}$ could be a hue even though $\mathbf{A}\bigcirc p$ and $\mathbf{E}\bigcirc\neg p$ are clearly not consistent. We will need many more relations and rules to eliminate other forms of inconsistency from the tableau.

Definition 25. [Hue] A set $a \subseteq \mathbf{cl}\phi$ is a hue for ϕ iff

(H1) a is MPC;

(H2) if $\alpha\mathcal{U}\beta \in a$ then $\alpha \in a$ or $\beta \in a$;

(H3) if $\neg(\alpha\mathcal{U}\beta) \in a$ then $\beta \notin a$; and

(H4) if $\mathbf{A}\alpha \in a$ or $\mathbf{R}\alpha \in a$ then $\alpha \in a$.

Note that we do not require that $\alpha \in a$ if $\mathbf{O}\alpha \in a$. As $\mathbf{O}$ is a deontic operator $\mathbf{O}\alpha \rightarrow \alpha$ is not valid.

Let H_ϕ be the set of hues of ϕ.

Definition 26. We define a function $\mathfrak{h}$ on paths such that

$$\mathfrak{h}(\pi) = \{\alpha : \alpha \in \mathbf{cl}\phi \text{ and } \pi \vDash \alpha\}$$

As H1–4 are simply properties that any set of formulas that hold along the same path must satisfy, it is clear that the following lemma holds.

Lemma 27. *From the semantics of RoBCTL*, we see that for each $\pi \in B$, $\mathfrak{h}(\pi)$ is a hue.*

Proof. (H1) Since the semantics of the $\wedge$ and $\neg$ operators in RoBCTL* come from classical logic, it is clear that $\mathfrak{h}(\pi)$ is MPC.

(H2) If $\alpha\mathcal{U}\beta \in \mathfrak{h}(\pi)$ then $\pi \vDash \alpha\mathcal{U}\beta$ and we see that either β is satisfied immediately and so $\pi \vDash \beta$ or $\pi \vDash \alpha$; hence $\alpha \in \mathfrak{h}(\pi)$ or $\beta \in \mathfrak{h}(\pi)$.

(H3) Likewise if $\neg(\alpha\mathcal{U}\beta) \in \mathfrak{h}(\pi)$ then we see that $\pi \nvDash \alpha\mathcal{U}\beta$ and so $\pi \nvDash \beta$ and so $\beta \notin \mathfrak{h}(\pi)$, demonstrating that H3 is satisfied.

(H4) If $\mathbf{A}\alpha \in \mathfrak{h}(\pi)$ then $\pi \vDash \mathbf{A}\alpha$ and so all paths starting at π_0, including π, satisfy α. Likewise if $\mathbf{R}\alpha \in \mathfrak{h}(\pi)$ then $\pi \vDash \mathbf{R}\alpha$ and so $\pi \vDash \alpha$. Either way $\alpha \in \mathfrak{h}(\pi)$. □

The following lemma motivates the definition of $\bigcirc_a^{-1}\phi$.

Lemma 28. *If $\mathfrak{h}(\pi) = a$ and ϑ is a fullpath such that $\vartheta_0 = \pi_0$ then $\vartheta \vDash \phi$ iff $\vartheta_{\geq 1} \vDash \bigcirc_a^{-1}\phi$.*

Proof. For any formula ϕ, let L_ϕ be the statement: "for all structures, and paths ϑ through that structure, we have $\vartheta \vDash \phi$ iff $\vartheta_{\geq 1} \vDash \bigcirc_a^{-1}\phi$."

It is clear that L_ϕ is true when ϕ is a state formula or a formula of the form $\bigcirc\psi$. For some pair of formulas (ϕ, ψ), say that L_ϕ and L_ψ is true, then:

($\Longrightarrow$)

1. Say that $\vartheta \vDash (\phi\mathcal{U}\psi)$,
   (a) If $\vartheta \vDash \psi$ then $\vartheta_{\geq 1} \vDash \bigcirc_a^{-1}\psi$ and thus $\vartheta_{\geq 1} \vDash (\cdots) \vee \bigcirc_a^{-1}(\psi)$ so $\vartheta_{\geq 1} \vDash \bigcirc_a^{-1}(\phi\mathcal{U}\psi)$.
   (b) If $\vartheta \nvDash \psi$ then $\vartheta \vDash \phi$ and $\vartheta_{\geq 1} \vDash (\phi\mathcal{U}\psi)$. Hence $\vartheta_{\geq 1} \vDash \bigcirc_a^{-1}(\phi) \wedge (\phi\mathcal{U}\psi)$ and so $\vartheta_{\geq 1} \vDash \bigcirc_a^{-1}(\phi\mathcal{U}\psi)$.
2. Say that $\vartheta \vDash \neg\phi$. Then $\vartheta \nvDash \phi$ and so $\vartheta_{\geq 1} \nvDash \bigcirc_a^{-1}(\phi)$. Finally, $\vartheta_{\geq 1} \vDash \neg\bigcirc_a^{-1}(\phi)$.
3. Say that $\vartheta \vDash \phi \wedge \psi$. Then $\vartheta \vDash \phi$ and $\vartheta \vDash \psi$. Thus $\vartheta_{\geq 1} \vDash \bigcirc_a^{-1}\phi$ and $\vartheta_{\geq 1} \vDash \bigcirc_a^{-1}\psi$. Hence $\vartheta_{\geq 1} \vDash \bigcirc_a^{-1}(\phi) \wedge \bigcirc_a^{-1}(\psi) = \bigcirc_a^{-1}(\phi \wedge \psi)$
4. Say that $\vartheta \vDash \mathbf{R}\phi$;
   (a) thus $\sigma \vDash \phi$ for any deviation σ from ϑ. Note that as a deviation, $\sigma_0 = \vartheta_0$, and hence $\sigma_{\geq 1} \vDash \bigcirc_a^{-1}(\phi)$; additionally for any path σ with $\sigma_0 = \vartheta_0$, if $\sigma_{\geq 1}$ is failure-free then σ is a deviation from ϑ and so $\vartheta \vDash \mathbf{A}\bigcirc\mathbf{O}\bigcirc_a^{-1}(\phi)$. As Ξ is a normalisation function, it follows that $\vartheta \vDash \mathbf{A}\bigcirc\mathbf{O}\Xi\left(\bigcirc_a^{-1}(\phi)\right)$; and

(b) we will show that $\vartheta_{\geq 1} \vDash \mathbf{R}\Xi\left(\bigcirc_a^{-1}(\phi)\right)$. Say that σ' is a deviation from $\vartheta_{\geq 1}$. Then from fusion closure of the set of paths B, there exists a path σ such that $\sigma_{\geq 1} = \sigma'$ and $\sigma_0 = \vartheta_0$. This path σ is a deviation from ϑ, and so $\sigma \vDash \phi$ and thus $\sigma' \vDash \bigcirc_a^{-1}(\phi)$. Hence $\vartheta_{\geq 1} \vDash \mathbf{R}\bigcirc_a^{-1}(\phi)$.

($\Longleftarrow$)

1. Say that $\vartheta_{\geq 1} \vDash \bigcirc_a^{-1}(\phi \mathcal{U} \psi) = \left(\bigcirc_a^{-1}(\phi) \wedge (\phi \mathcal{U} \psi)\right) \vee \bigcirc_a^{-1}(\psi)$
   (a) If $\vartheta_{\geq 1} \vDash \bigcirc_a^{-1}(\phi) \wedge (\phi \mathcal{U} \psi)$ then $\vartheta \vDash \phi$ and $\vartheta_{\geq 1} \vDash (\phi \mathcal{U} \psi)$ so $\vartheta \vDash (\phi \mathcal{U} \psi)$.
   (b) If $\vartheta_{\geq 1} \vDash \bigcirc_a^{-1}(\psi)$ then $\vartheta \vDash \psi$ and so $\vartheta \vDash (\phi \mathcal{U} \psi)$.
2. Say that $\vartheta_{\geq 1} \vDash \neg\bigcirc_a^{-1}(\phi)$. Then $\vartheta_{\geq 1} \nvDash \bigcirc_a^{-1}(\phi)$ and so $\vartheta \nvDash \phi$. Thus $\vartheta \vDash \neg\phi$.
3. Say that $\vartheta_{\geq 1} \vDash \bigcirc_a^{-1}(\phi \wedge \psi)$. Then, from the definition of $\bigcirc_a^{-1}$ we have $\vartheta_{\geq 1} \vDash \bigcirc_a^{-1}\phi$ and $\vartheta_{\geq 1} \vDash \bigcirc_a^{-1}\psi$. Thus $\vartheta \vDash \phi$ and $\vartheta \vDash \psi$. Hence $\vartheta \vDash \phi \wedge \psi$
4. Say that $\vartheta_{\geq 1} \vDash \bigcirc_a^{-1}(\mathbf{R}\phi)$. Clearly $\vartheta_{\geq 1} \nvDash \bot$, and so $\bigcirc_a^{-1}(\mathbf{R}\phi) \neq \bot$. Thus, from Definition 19, we know that $\bigcirc_a^{-1}(\mathbf{R}\phi) = \mathbf{R}\bigcirc_a^{-1}(\phi)$ and that

$$\mathbf{A}\bigcirc\mathbf{O}\left(\bigcirc_a^{-1}(\phi)\right) \in a = \mathfrak{h}(\pi) \ .$$

   It follows that $\vartheta \vDash \mathbf{A}\bigcirc\mathbf{O}\left(\bigcirc_a^{-1}(\phi)\right)$. Say σ is a deviation from ϑ; we will show that $\sigma_{\geq 1} \vDash \bigcirc_a^{-1}(\phi)$, and so $\sigma \vDash \phi$. Since every deviation forces ϕ it follows that $\vartheta \vDash \mathbf{R}\phi$:
   (a) as $\vartheta \vDash \mathbf{A}\bigcirc\mathbf{O}\left(\bigcirc_a^{-1}(\phi)\right)$, if σ is a 0-deviation then $\sigma_{\geq 1} \vDash \bigcirc_a^{-1}(\phi)$;
   (b) if σ is an i-deviation where $i > 0$, then $\sigma_{\leq 1} = \vartheta_{\leq 1}$ and so $\sigma_{\geq 1}$ is an $(i-1)$-deviation from $\vartheta_{\geq 1}$. As $\vartheta_{\geq 1} \vDash \mathbf{R}\bigcirc_a^{-1}(\phi)$, it follows that $\sigma_{\geq 1} \vDash \bigcirc_a^{-1}(\phi)$.

By induction on the length of the formula we see that the lemma holds. □

We will now define a temporal successor relation r_X on hues, so called because it will satisfy $\mathfrak{h}(\pi)\, r_X\, \mathfrak{h}(\pi_{\geq 1})$ for all paths π. The definition below is similar to that found in Reynolds [8], but with the additional requirements (R5) and (R6).

Definition 29. [r_X] The temporal successor r_X relation on hues is defined as follows: for all hues a, b put (a, b) in r_X iff the following conditions are satisfied:

(R1) $\bigcirc\alpha \in a$ implies $\alpha \in b$

(R2) $\neg\bigcirc\alpha \in a$ implies $\alpha \notin b$

(R3) $\alpha\mathcal{U}\beta \in a$ and $\beta \notin a$ implies $\alpha\mathcal{U}\beta \in b$

(R4) $\neg(\alpha\mathcal{U}\beta) \in a$ and $\alpha \in a$ implies $\neg(\alpha\mathcal{U}\beta) \in b$

(R5) $\mathbf{R}\alpha \in a$ implies $\bigcirc_a^{-1}(\mathbf{R}\alpha) \in b$

(R6) $\neg\mathbf{R}\alpha \in a$ implies $\neg\bigcirc_a^{-1}(\mathbf{R}\alpha) \in b$

We will now define a relation on hues $r_{\mathbf{A}}$, which informally describes hues which might describe paths which start at the same state.

Definition 30 ($r_{\mathbf{A}}$)**.** For hues a, b, we put (a, b) in $r_{\mathbf{A}}$ iff the following conditions hold:

(A1) $\mathbf{A}\alpha \in a$ iff $\mathbf{A}\alpha \in b$ and $\mathbf{O}\alpha \in a$ iff $\mathbf{O}\alpha \in b$

(A2) For all $p \in \mathbb{V}$, we have $p \in a$ iff $p \in b$

Note that if $(a, b) \in r_{\mathbf{A}}$ then for all formulas ϕ we have $\bigcirc_a^{-1}(\phi) = \bigcirc_b^{-1}(\phi)$ (i.e. $\bigcirc_a^{-1} = \bigcirc_b^{-1}$). The $r_{\mathbf{A}}$ relation is used to specify which pairs of hues can exist in the same "colour"; a colour represents a set of hues for paths which could start at the same world.

Definition 31. A set of hues C is a colour of ϕ iff

(C1) for all $a, b \in C$ we have $(a, b) \in r_{\mathbf{A}}$; and

(C2) if $a \in C$ and $\neg\mathbf{A}\alpha \in a$ or $\neg\mathbf{R}\alpha \in a$ then there is $b \in C$ such that $\neg\alpha \in b$; and

(C3) if $a \in C$ and $\neg\mathbf{O}\alpha \in a$ then there is $b \in C$ such that $\neg\alpha \in b$ and $\gamma \in b$; and

(C4) there exists $a \in C$ such that $\gamma \in a$

Note that we cannot have both $\mathbf{A}\alpha$ and $\neg\mathbf{R}\alpha$ in a hue $a \in C$, as from C2 there would have to exist a hue $b \in C$ such that $\neg\alpha \in b$, from C1 and A1 we know that $\mathbf{A}\alpha \in b$. Finally from $H1$ we know that $\alpha \in b$ but since b is MPC this contradicts $\neg\alpha \in b$.

Let C_ϕ be the colours of ϕ. We define a successor relation on C_ϕ as follows:.

Definition 32 (R_X)**.** We define a temporal successor relation R_X on colours as follows: for all $C, D \in C_\phi$, put $(C, D) \in R_X$ iff for all $b \in D$ there exists $a \in C$ such that $(a, b) \in r_X$.

5.1. Pruning the Tableau

The temporal successor relations ensure that each step of the tableau is consistent. Note though that for any finite n, the formula $\Diamond\phi$ does not require that ϕ occur in any of the next n steps. Thus each time-step being consistent with the next is not enough to ensure that $\Diamond\phi$ is satisfied. We will use the term eventuality to informally describe formulas for which the consistency of each temporal step is not sufficient to ensure that the formula is really satisfied on the resulting model.

The following pruning procedure is based on the technique used in Reynolds [8]. However, to extend this technique to RoBCTL* a new type of eventuality must be considered: $\neg\mathbf{R}\psi$. This formula can be interpreted as "either $\neg\psi$ or there exists a path which eventually deviates, and $\neg\psi$ holds along that path."

It is necessary to explicitly handle this eventuality. Imagine a tableau with only one colour $C = \{\{\neg\mathbf{R}\Box p, \Box p, p, \neg\mathbf{v}, \top\}\}$, it is clear that $(C, C) \in R_X$ so without handling eventualities of the form $\neg\mathbf{R}\psi$ this tableau would be accepted. We need to ensure that eventually a path deviates on which $\Box p$ is false. The rules (1) and (2) below correspond to removal rules 1 and 2 from Reynolds [8], but the rule (3) is original.

Initially, we let the set S' of colours equal C_ϕ. We say that a 3-tuple (C, c, α) is an instance iff $C \in S'$, c is a hue, α is a formula and $\alpha \in c \in C$. We iteratively remove colours from S' according to the following rules until no more colours can be removed:

1. Remove C from S' if we cannot find successors for every hue in C. That is, we remove C from S' if there exists a hue c in C such that for every $D \in S'$,
   (a) $(C, D) \notin R_X$, or
   (b) for every $d \in D$, the pair $(c, d) \notin r_X$.
2. An instance $(C, c, \alpha\mathcal{U}\beta)$ is directly fulfilled iff $\beta \in c$. Initially, an instance is fulfilled iff it is directly fulfilled; we iteratively mark $(C, c, \alpha\mathcal{U}\beta)$ as fulfilled iff there exists a fulfilled instance $(D, d, \alpha\mathcal{U}\beta)$ such that $(C, D) \in R_X$ and $(c, d) \in r_X$. We finish when we can no longer mark instances as fulfilled. Finally, for all instances $(C, c, \alpha\mathcal{U}\beta)$ that are not fulfilled, we remove C from S'.
3. An instance $(C, c, \neg\mathbf{R}\alpha)$ is directly fulfilled iff $\mathbf{A}\bigcirc\mathbf{O}\Xi\left(\bigcirc_c^{-1}(\alpha)\right) \notin c$ or $\alpha \notin c$. Initially, an instance is fulfilled iff it is directly fulfilled; we iteratively mark $(C, c, \neg\mathbf{R}\alpha)$ as fulfilled iff there exists a fulfilled instance $(D, d, \neg\mathbf{R}\alpha')$ such that $(C, D) \in R_X$, $(c, d) \in r_X$ and $\mathbf{R}\alpha' = \bigcirc_c^{-1}(\mathbf{R}\alpha)$; we finish when we can no longer mark instances as fulfilled. Finally, for all instances $(C, c, \neg\mathbf{R}\alpha)$ that are not fulfilled, we remove C from S'.

Definition 33. We say that the tableau succeeds if there exists a hue h and colour C such that $\phi \in h \in C \in S'$ after the pruning is complete.

5.2. Cardinality of the Closure Set

The complexity of the tableau is doubly exponential with respect to $|\mathbf{cl}\phi|$. To see this note that the set of hues is a subset of $2^{\mathbf{cl}\phi}$ and so the set of colours is a subset of $2^{2^{\mathbf{cl}\phi}}$. Thus constructing all the colours can be done in $2^{2^{|\mathbf{cl}\phi|}}$ time. The time required to prune a colour is polynomial in the number of colours.

We see $|\mathbf{cl}\phi|$ is linear with respect to $|\phi| \max_{\psi \sqsubseteq \phi} |\bigcirc^\star(\{\psi\})|$, from Definition 23 of $\mathbf{cl}\phi$. Thus if $|\bigcirc^\star(\{\psi\})|$ is n-exponential then the overall complexity of the tableau is $(n+2)$-exponential. In this section we will discuss the size of $|\bigcirc^\star(\{\psi\})|$.

Theorem 34. *When we require that there are no more than m pairs of alternations between $\mathbf{R}$ and $\mathcal{U}$ that are not broken by an $\mathbf{A}$ (or $\mathbf{O}$) then $|\bigcirc^\star(\{\psi\})|$ is $3m$-exponential on the size of the formulas.*

We prove this by showing that we can build $|\bigcirc^\star(\{\psi\})|$ recursively: below we show that we can recurse through any number of $\mathbf{R}$ operators with a singly exponential blowup until we reach a $\mathcal{U}$ operator, and we can recurse through any number of $\mathcal{U}$ operators with a doubly exponential blowup until we reach a $\mathbf{R}$ operator.

Lemma 35. *Say that Φ is a set of formulas each starting with $\mathbf{R}$, and ψ is a formula constructed from x instances of state formulas, y instances of $\wedge, \mathcal{U}, \bigcirc, \neg$ operators (we exclude the $\mathbf{R}$ operator) and elements of Φ. Then $\bigcirc^\star(\{\psi\})$ is doubly exponential with respect to $x + y + \max_{\phi\in\Phi} \bigcirc^\star(\phi)$.*

Proof. Let $\mathbf{C}$ be a function such that $\mathbf{C}(\Phi)$ represents all normalised classical formulas with the elements of Φ as atoms. Note that a truth table on n atoms has 2^n rows, and hence there are 2^{2^n} equivalence classes on such formulas. It follows that $|\mathbf{C}(\Phi)| \leq 2^{2^{|\Phi|}}$. Let $\oint$ be a function from formulas to sets of formulas such that $\phi \in \oint(\psi)$ iff $\bigcirc\phi \sqsubseteq \psi$ or $\phi = (\phi_1 \mathcal{U} \phi_2) \sqsubseteq \psi$. For any set of formulas Φ, we define $\oint(\Phi)$ as $\bigcup_{\phi\in\Phi} \oint(\phi)$. We see that the following statement holds:

$$\bigcirc^\star(\{\psi\}) \subseteq \mathbf{C}\left(\oint(\psi) \cup \left(\bigcup_{\mathbf{R}\phi\in\Phi} \bigcirc^\star(\{\mathbf{R}\phi\})\right)\right)$$

It follows that $\bigcirc^\star(\{\psi\}) \in \mathcal{O}\left(2^{2^{(x+y)+\sum_{\mathbf{R}\phi\in\Phi} \bigcirc^\star(\{\mathbf{R}\phi\})}}\right)$. In other words, $\bigcirc^\star(\{\psi\})$ is doubly exponential with respect to $x + y + \max_{\phi\in\Phi} \bigcirc^\star(\{\phi\})$. □

Definition 36. When considering some formula ψ, let $\#(\phi)$ be the number of times that ϕ occurs in ψ without being part of a state formula (not nested inside an $\mathbf{O}$ or $\mathbf{A}$).

Lemma 37. *Say that Φ is a set of formulas, and ψ is a formula constructed from any number of instances of $\wedge$ and $\neg$ operators, x instances of state formulas, y instances of $\mathbf{R}$ operators, and z instances of $\bigcirc$ operators (we exclude the $\mathcal{U}$ operator) and elements of Φ. Then $|\bigcirc^\star(\{\psi\})|$ is singly exponential with respect to $x + y + z + \max_{\phi\in\Phi} \bigcirc^\star(\{\phi\})$.*

Proof. Consider the set $\bigcirc^{-i}(\{\psi\})$. By inspecting the definition of $\bigcirc^{-1}$ we see that we have two choices when we reach a state formula, $\top$ and $\bot$. When we reach a $\mathbf{R}$ operator, we have two choices, terminate with $\bot$ or continue to recurse. It is clear that for all $\phi \in \Phi$ and $j \in [0, \infty]$ it is the case that $\left|\bigcirc^{-j}(\{\phi\})\right| \leq |\bigcirc^\star(\{\phi\})|$. By taking the product of all these cases we get the following:

$$\left|\bigcirc^{-i}(\{\psi\})\right| \leq 2^x 2^y \prod_{\phi\in\Phi} |\bigcirc^\star(\{\phi\})|^{\#(\phi)}$$

Note that $\bigcirc^{-1}$ removes any $\bigcirc$ operator or state formula that is not inside an $\bigcirc$ or $\mathcal{U}$ operator. It is clear from induction that $\bigcirc^{-i}$ will have removed any

$\bigcirc$ operator or state formula that is not inside i $\bigcirc$ operators or a $\mathcal{U}$ operator. Since ψ does not contain any $\mathcal{U}$ operator that does not form part of a $\phi \in \Phi$, it is the case that for $i > z$ we no element of $\bigcirc^{-i}(\{\psi\})$ would contain an $\bigcirc$ operator that does not form part of an element of $|\bigcirc^{\star}(\{\psi\})|$, and have already replaced all state formulas which do not form part of an element of $|\bigcirc^{\star}(\{\psi\})|$ with either $\top$ or $\bot$. It follows that

$$\left|\bigcup_{i>z} \bigcirc^{-i}(\{\psi\})\right| \leq 2^x 2^y \prod_{\phi\in\Phi} |\bigcirc^{\star}(\{\phi\})|^{\#(\phi)}.$$

Since $\bigcirc^{\star}(\{\psi\}) = \bigcup_{i\geq 0} \Xi\bigcirc^{-i}(\{\psi\})$ and $\bigcirc^{-0}(\{\psi\}) = 1$,

$$|\bigcirc^{\star}(\{\psi\})| = \left|\bigcup_{i\geq 0} \bigcirc^{-i}(\{\psi\})\right| \leq 1 + (1+z)\, 2^x 2^y \prod_{\phi\in\Phi} |\bigcirc^{\star}(\{\phi\})|^{\#(\phi)}.$$

□

Corollary 38. *The tableau provides a 3-exponential decision procedure for the fragments of RoCTL* and RoBCTL* without Until.*

Proof. Clear from Lemmas 37 and 13. □

Definition 39. Let $\mathbf{R}^n$ be a sequence $\mathbf{R}\mathbf{R}\ldots\mathbf{R}$ of n instances of the $\mathbf{R}$ operator.

We see that also $\bigcirc^{\star}(\{\mathbf{R}^n\phi\}) = \{\mathbf{R}x\colon\ x \in \bigcirc^{\star}(\phi)\} \cup \{\bot\}$. The $\mathbf{R}^n$ operator is interesting as it represents the statement "even with n additional failures" and a significant factor in the design of the $\mathbf{R}$ was to provide a simple unimodal operator that could represent this statement. However, in the QCTL* based decision procedure for RoCTL* French et al. [21] the $\mathbf{R}^n$ operator involves a non-elementary blowup in the complexity. By comparison, in this tableau the complexity is independent of n in $\mathbf{R}^n$.

Corollary 40. *When we consider only formulas without a $\mathcal{U}$ operator nested within a $\mathbf{R}$ operator nested within a $\mathcal{U}$ operator (unbroken by an $\mathbf{A}$ or $\mathbf{O}$), then $|\bigcirc^{\star}(\{\psi\})|$ is 4-exponential with respect to ψ and the complexity is at worst 6-exponential.*

Proof. Let Φ_1 be the set of formulas that do not contain any $\mathcal{U}$ operators. From Lemma 37 we see that $|\bigcirc^{\star}(\{\phi\})|$ is singly exponential in $|\phi|$ for $\phi \in \Phi_1$. Let Φ_2 be the set of formulas that consist only of elements of Φ_1 and operators other than $\mathcal{U}$. From Lemma 35 we see that $|\bigcirc^{\star}(\{\phi\})|$ is 3-exponential in $|\phi|$ for $\phi \in \Phi_2$. Let Φ_3 be the set of formulas that consist only of elements of Φ_2 and operators other than $\mathbf{R}$. From Lemma 35 we see that $|\bigcirc^{\star}(\{\phi\})|$ is 4-exponential in $|\phi|$ for $\phi \in \Phi_3$. We see that Φ_3 is precisely the set of formulas that do not contain a $\mathcal{U}$ operator nested within a $\mathbf{R}$ operator nested within a $\mathcal{U}$ operator, and furthermore that $\bigcirc^{\star}(\{\mathbf{A}\phi\}) = \bigcirc^{\star}(\{\mathbf{O}\phi\}) = \{\top, \bot\}$, and so these nestings would not pose a problem if broken by an $\mathbf{O}$ or $\mathbf{A}$. Finally since the tableau is 2-exponential in the size of the closure set the result follows. □

Most real world uses for Ro(B)CTL*, including the examples in Section 4, do not require multiple alternations between $\mathcal{U}$ and $\mathbf{R}$ and so they would be elementary to decide (at worst 6-exponential). Also clearly $|\bigcirc^\star(\{\psi\})|$ is $\mathcal{O}(1)$ for fixed ψ or for ψ of fixed length. As such $\mathbf{cl}\phi$ is linear on $|\phi|$ when the length of the subformulas with $\mathbf{R}$ as the operator of highest precedence is bounded, and complexity is 2-exponential like CTL*. This indicates that we can introduce some (fixed) RoBCTL* properties into a set of systems specified in BCTL* without affecting the overall complexity.

5.3. Soundness

RoBCTL*-TAB is sound, that is, if it succeeds on ϕ then ϕ is satisfiable in RoBCTL*

Say that RoBCTL*-TAB finishes with the set S' of colours. Then we define a RoBCTL-structure (S, R, g, B) as follows: the transition frame (S, R) is simply (S', R_X), and the valuation $g(C)$ of a world/colour C contains an atom p iff the hues in C contain p. We now define the set of bundled paths B.

We call an ω-sequence $\langle (c_0, h_0), (c_1, h_1), \ldots \rangle$ a thread through S' iff for all $i \geq 0$: each $c_i \in S'$, each $h_i \in c_i$, each $(c_i, c_{i+1}) \in R_X$, each $(h_i, h_{i+1}) \in r_X$. We say that this is a fulfilling thread iff for all $i \geq 0$

1. For all formulae of the form $(\alpha \mathcal{U} \beta)$ in h_i, there exists $j \geq i$ such that $\beta \in h_j$
2. For all formulae of the form $\neg\mathbf{R}\alpha$ in h_i, $\neg\alpha \in h_i$ or there exists $j \geq i$ such that

$$\bigcirc_{h_j}^{-1} \bigcirc_{h_{j-1}}^{-1} \bigcirc_{h_{j-2}}^{-1} \ldots \bigcirc_{h_i}^{-1} (\mathbf{R}\alpha) = \bot$$

We include a fullpath $\sigma = \langle c_0, c_1, \ldots \rangle$ in B iff there exists a fulfilling thread $\langle (c_0, h_0), (c_1, h_1), \ldots \rangle$, and we say that this thread justifies σ being in B.

Lemma 41. *Requirement (2) above is equivalent to the statement: there exists $j \geq i$ such that $\left(c_j, h_j, \neg \bigcirc_{h_{j-1}}^{-1} \bigcirc_{h_{j-2}}^{-1} \ldots \bigcirc_{h_i}^{-1} (\mathbf{R}\alpha)\right)$ is directly fulfilled.*

Proof. ($\Longrightarrow$) Say requirement (2) holds.

Case 1. $\neg\alpha \in h_i$. Then $(c_i, h_i, \neg(\mathbf{R}\alpha))$ is directly fulfilled.

Case 2. There exists $j \geq i$ such that

$$\bigcirc_{h_j}^{-1} \bigcirc_{h_{j-1}}^{-1} \bigcirc_{h_{j-2}}^{-1} \ldots \bigcirc_{h_i}^{-1} (\mathbf{R}\alpha) = \bot$$

Say Without Loss of Generality (WLOG) that j is as small as possible, i.e.

$$\bigcirc_{h_{j-1}}^{-1} \bigcirc_{h_{j-2}}^{-1} \ldots \bigcirc_{h_i}^{-1} (\mathbf{R}\alpha) \neq \bot$$

Then there exists β such that

$$\mathbf{R}\beta = \bigcirc_{h_{j-1}}^{-1} \bigcirc_{h_{j-2}}^{-1} \ldots \bigcirc_{h_i}^{-1} (\mathbf{R}\alpha)$$

and

$$\bigcirc_{h_j}^{-1}\mathbf{R}\beta = \bot$$

Thus $\mathbf{A}\bigcirc\mathbf{O}\Xi\left(\bigcirc_{h_j}^{-1}(\beta)\right) \notin h_j$ and so $(c_j, h_j, \neg\mathbf{R}\beta)$ is directly fulfilled. Hence

$$\left(c_j, h_j, \neg\bigcirc_{h_{j-1}}^{-1}\bigcirc_{h_{j-2}}^{-1}\ldots\bigcirc_{h_i}^{-1}(\mathbf{R}\alpha)\right)$$

is directly fulfilled.

($\Longleftarrow$) Say there exists $j \geq i$ such that $\left(c_j, h_j, \neg\bigcirc_{h_{j-1}}^{-1}\bigcirc_{h_{j-2}}^{-1}\ldots\bigcirc_{h_i}^{-1}(\mathbf{R}\alpha)\right)$ is directly fulfilled. Hence, there exists β such that $\mathbf{R}\beta = \bigcirc_{h_{j-1}}^{-1}\bigcirc_{h_{j-2}}^{-1}\ldots\bigcirc_{h_i}^{-1}(\mathbf{R}\alpha)$.

Case 1. $\mathbf{A}\bigcirc\mathbf{O}\Xi\left(\bigcirc_{h_j}^{-1}(\beta)\right) \notin h_j$. Then $\bigcirc_{h_j}^{-1}(\mathbf{R}\beta) = \bot$ and so

$$\bigcirc_{h_j}^{-1}\bigcirc_{h_{j-1}}^{-1}\bigcirc_{h_{j-2}}^{-1}\ldots\bigcirc_{h_i}^{-1}(\mathbf{R}\alpha) = \bot$$

Case 2. $\neg\beta \in h_j$. From Corollary 42 below, $\beta \in h_j$ iff $\alpha \in h_i$. Hence $\neg\alpha \in h_i$. □

Corollary 42. *Say that there exists β such that*

$$\mathbf{R}\beta = \bigcirc_{h_{j-1}}^{-1}\bigcirc_{h_{j-2}}^{-1}\ldots\bigcirc_{h_i}^{-1}(\mathbf{R}\alpha)\ .$$

Where $\mathbf{R}\alpha \in \mathbf{cl}\phi$. Then $\beta \in h_j$ iff $\alpha \in h_i$.

From the definition of $\mathbf{cl}\phi$ we see that if $\mathbf{R}\alpha \in \mathbf{cl}\phi \implies \mathbf{R}\beta \in \mathbf{cl}\phi$. Hence we see that the above corollary follows from Lemma 41 and induction. We may likewise use induction to show that $\sigma_{\geq i} \vDash \alpha$ iff $\sigma_{\geq j} \vDash \bigcirc_{h_{j-1}}^{-1}\bigcirc_{h_{j-2}}^{-1}\ldots\bigcirc_{h_i}^{-1}(\mathbf{R}\alpha)$.

We will show that B is a bundle by showing that B is suffix closed, fusion closed and non-empty.

Lemma 43. *B is suffix closed.*

Proof. Say $\mu = \langle(c_0, h_0), (c_1, h_1), \ldots\rangle$ justifies $\sigma \in B$. Clearly if there is an eventuality at σ_m which is fulfilled at σ_n then $n \geq m$. Thus for any suffix $\sigma_{\geq j}$ then if σ_n is not on $\sigma_{\geq j}$ then σ_m is not on $\sigma_{\geq j}$ either. Hence we see that for all $j \geq 0$, $\mu_{\geq j}$ justifies $\sigma_{\geq j} \in B$. □

Lemma 44. *B is fusion closed*

Say that σ, π are in B and $\sigma_0 = \pi_1$. We will show below that

$$\langle\pi_0, \sigma_0, \sigma_1, \ldots\rangle \in B\ .$$

The general case where $\sigma_0 = \pi_j$ follows from prefix closure and induction.

As $\sigma \in B$, there is a fulfilling thread $\mu = \langle(\sigma_0, h_1), (\sigma_1, h_2), \ldots\rangle$. As $(\pi_0, \pi_1) \in R_X$, we can choose h_0 from π_0 such that $(h_0, h_1) \in r_X$.

If $\alpha\mathcal{U}\beta \in h_0$, then $\beta \in h_0$ or $\alpha\mathcal{U}\beta \in h_1$. As μ is fulfilling, if $\alpha\mathcal{U}\beta \in h_1$ then there exists $j \geq 1$ such that $\beta \in h_j$.

If $\neg\mathbf{R}\alpha \in h_0$ then

1. $\neg\alpha \in h_0$ or $\neg\mathbf{A}\bigcirc\mathbf{O}\Xi\left(\bigcirc_a^{-1}(\alpha)\right) \in h_0$; or
2. $\neg\mathbf{R}\Xi\left(\bigcirc_{h_0}^{-1}(\alpha)\right) \in h_1$. (From (R6) and Definition 19)

If (1) then the eventuality $\neg\mathbf{R}\alpha$ is directly fulfilled. Otherwise, from Lemma 41, there exists ϕ and j such that

$$\begin{aligned}\mathbf{R}\phi =& \bigcirc_{h_{j-1}}^{-1}\bigcirc_{h_{j-2}}^{-1}\ldots\bigcirc_{h_1}^{-1}\left(\mathbf{R}\Xi\left(\bigcirc_{h_0}^{-1}(\alpha)\right)\right)\\ =& \bigcirc_{h_{j-1}}^{-1}\bigcirc_{h_{j-2}}^{-1}\ldots\bigcirc_{h_0}^{-1}(\mathbf{R}\alpha)\end{aligned}$$

and $\mathbf{A}\bigcirc\mathbf{O}\Xi\left(\bigcirc_{h_j}(\phi)\right) \notin h_j$ or $\phi \notin h_j$. Thus the eventuality $\neg\mathbf{R}\alpha$ is fulfilled by h_j.

Lemma 45. *If $a \in c \in S'$ then there is a fulfilling thread*

$$\mu = \langle (c_0, h_0), (c_1, h_1), \ldots\rangle$$

such that $h_0 = a$ and $c_0 = c$. Thus $\sigma = \langle c_0, c_1, c_2, \ldots\rangle \in B$.

Proof. As with Reynolds Reynolds [8] we iteratively satisfy the oldest eventuality first. Hence every eventuality is eventually fulfilled.

Say we have chosen the first n elements of μ and $0 \leq i \leq n$. We say that an eventuality $\alpha\mathcal{U}\beta \in h_i$ is unfulfilled iff for all $j \leq i \leq n$ the formula $\beta \notin h_j$. We say that an eventuality $\neg\mathbf{R}\phi \in h_i$ is unfulfilled iff for all $j \leq i \leq n$ the instance

$$\left(c_i, h_i, \neg\bigcirc_{h_{j-1}}^{-1}\bigcirc_{h_{j-2}}^{-1}\ldots\bigcirc_{h_i}^{-1}(\mathbf{R}\alpha)\right)$$

is not directly fulfilled.
For the lowest i such that there exists an unfulfilled eventuality in h_i we fulfil this eventuality as follows:

Case 1: If no such i exists, we choose (c_{n+1}, h_{n+1}) such that $(c_n, c_{n+1}) \in R_X$ and $(h_n, h_{n+1}) \in r_X$.

Case 2: If the eventuality is of the form $\alpha\mathcal{U}\beta$, then there must exist $\alpha\mathcal{U}\beta \in h_n$. Due to the pruning rule, for some j there must exist a sequence of instances

$$(c_n, h_n, \alpha\mathcal{U}\beta), (c_{n+1}, h_{n+1}, \alpha\mathcal{U}\beta), \ldots, (c_j, h_j, \alpha\mathcal{U}\beta)$$

such that the final instance is directly fulfilled ($\beta \in h_j$), and each other instance is fulfilled by the next instance in the chain. Having now chosen μ up to (c_j, h_j) with $\beta \in h_j$, the eventuality $\alpha\mathcal{U}\beta \in h_i$ is now fulfilled.

Case 3: If the eventuality is of the form $\neg\mathbf{R}\alpha$, then there must exist $\neg\mathbf{R}\phi \in h_n$ where

$$\mathbf{R}\phi = \bigcirc_{h_{n-1}}^{-1}\bigcirc_{h_{n-2}}^{-1}\ldots\bigcirc_{h_i}^{-1}(\mathbf{R}\alpha).$$

Due to the pruning rule, for some j there must exist a sequence of instances

$$(c_n, h_n, \neg\mathbf{R}\phi), \left(c_{n+1}, h_{n+1}, \neg\bigcirc_{h_n}^{-1}\mathbf{R}\phi\right), \ldots, \left(c_j, h_j, \neg\bigcirc_{h_{j-1}}^{-1}\bigcirc_{h_{j-2}}^{-1}\cdots\bigcirc_{h_n}^{-1}\mathbf{R}\phi\right)$$

such that the final instance is directly fulfilled, and each other instance is fulfilled by the next instance in the chain. Having now chosen μ up to (c_j, h_j), the eventuality $\neg\mathbf{R}\alpha \in h_i$ is now fulfilled, as

$$\neg\bigcirc^{-1}_{h_{j-1}}\bigcirc^{-1}_{h_{j-2}}\cdots\bigcirc^{-1}_{h_n}\mathbf{R}\phi = \neg\bigcirc^{-1}_{h_{j-1}}\bigcirc^{-1}_{h_{j-2}}\cdots\bigcirc^{-1}_{h_n}\bigcirc^{-1}_{h_{n-1}}\bigcirc^{-1}_{h_{n-2}}\ldots\bigcirc^{-1}_{h_i}(\mathbf{R}\alpha),$$

and

$$\left(c_j, h_j, \neg\bigcirc^{-1}_{h_{j-1}}\bigcirc^{-1}_{h_{j-2}}\cdots\bigcirc^{-1}_{h_i}(\mathbf{R}\alpha)\right),$$

is directly fulfilled. □

Lemma 46. *For all α in $\mathbf{cl}\phi$, for all threads $\mu = \langle(c_0, h_0), (c_1, h_1), \ldots\rangle$ justifying $\sigma = \langle c_0, c_1, \ldots\rangle$ we have*

$$(S, R, g, B), \sigma \vDash \alpha \text{ iff } \alpha \in h_0$$

Proof. We will prove this using induction. Let L_ψ be the statement: for all threads $\mu = \langle(c_0, h_0), (c_1, h_1), \ldots\rangle$ justifying $\sigma = \langle c_0, c_1, \ldots\rangle$ we have

$$(S, R, g, B), \sigma \vDash \psi \text{ iff } \psi \in h_0 \ .$$

First note that L_ψ holds by definition when ψ is an atom. Assume that L_ψ holds for all ψ in $\mathbf{cl}\phi$ where $|\psi| \leq n$ and the number of nested $\mathbf{R}$ operators in ψ is less than m. We can show that L_ψ holds for any ψ in $\mathbf{cl}\phi$ where $|\psi| \leq n+1$ and the number of nested $\mathbf{R}$ operators is less than $m+1$. We will show below the case where ψ is of the form $\mathbf{R}\beta$, the other forms are left as an exercise for the reader (or see Reynolds [8]).Should we expand this?

($\Longrightarrow$) Suppose by contradiction that $\mathbf{R}\beta \notin h_0$ so that $\neg\mathbf{R}\beta \in h_0$. If $\neg\beta \in h_0$ then $\sigma \vDash \neg\beta$, which gives us a contradiction. Thus there exists i such that $\bigcirc_{h_i}\bigcirc^{-1}_{h_{i-1}}\bigcirc^{-1}_{h_{i-2}}\ldots\bigcirc^{-1}_{h_)}(\mathbf{R}\beta) = \bot$. It is now easy to use Lemma 28 and induction to show that $\sigma \vDash \mathbf{R}\beta$ iff $\sigma \vDash \bot$, which again gives us a contradiction.

($\Longleftarrow$) Suppose by contradiction that $(S, R, g, B), \sigma \nvDash \mathbf{R}\beta$. If $\sigma \nvDash \beta$ then $\beta \notin h_0$, so for some i there must exist an i-deviation π from σ such that $\pi \nvDash \beta$. We can show that $\pi_{\geq i} \nvDash \bigcirc^{-1}_{\pi_{i-1}}\cdots\bigcirc^{-1}_{\pi_1}\bigcirc^{-1}_{\pi_0}(\beta)$ and so

$$\Xi\bigcirc^{-1}_{\pi_{i-1}}\cdots\Xi\bigcirc^{-1}_{\pi_1}\Xi\bigcirc^{-1}_{\pi_0}(\beta) \notin \pi_i.$$

Thus

$$\bigcirc^{-1}_{\pi_{i-1}}\cdots\bigcirc^{-1}_{\pi_1}\bigcirc^{-1}_{\pi_0}(\mathbf{R}\beta) = \mathbf{R}\Xi\bigcirc^{-1}_{\pi_{i-1}}\cdots\Xi\bigcirc^{-1}_{\pi_1}\Xi\bigcirc^{-1}_{\pi_0}(\beta) \notin \pi_i.$$

It follows that $\mathbf{R}\beta \notin \pi_0 = \sigma_0$. Should we expand this? □

5.4. Completeness

Lemma 47. *RoBCTL*-TAB is complete, that is, if ϕ is satisfiable in RoBCTL$_\mathbf{v}$ structure then RoBCTL*-TAB halts and succeeds on ϕ.*

The tableau is finite so RoBCTL*-TAB will halt.

Say that ϕ is satisfiable. Then there exists a RoBCTL$_{\mathbf{v}}$ structure (S, R, g, B) and path π^0 in B such that $\pi^0 \vDash \phi$. We will define a translation ρ from worlds to colours, and show that for each world w in S, the colour $\rho(w)$ will not be pruned from the tableau. Hence S' will be non-empty when RoBCTL*-TAB halts, and so RoBCTL*-TAB will succeed.

We will now define a function from worlds to colours. This definition uses the function from paths to hues defined in Definition 26.

Definition 48. We define a function ρ on worlds to sets of hues:

$$\rho(w) = \{\mathfrak{h}(\pi) \colon \pi \in B \text{ and } \pi_0 = w\} \ .$$

Lemma 49. *For each $w \in S$, $\rho(w)$ is a colour.*

Proof. Obvious, as C1–C4 are trivial consequences of the semantics of RoBCTL*. □

Case 1 and 2 below are similar to the corresponding proof in Reynolds [8]. Case 3 is original.

Lemma 50. *For each $w \in S$, $\rho(w)$ is never removed from S'.*

Proof. Suppose for contradiction that $\rho(w)$ is removed.

Case 1: $\rho(w)$ is removed from S' by rule 1 as we cannot find successors for every hue in $\rho(w)$. Then there exists a hue $c \in \rho(w)$ such that for every $D \in S'$, $(\rho(w), D) \notin R_X$, or for every $d \in D$, the pair $(c, d) \notin r_X$. Since $c \in \rho(w)$ there must exist a fullpath $\pi \in B$ such that $\pi_0 = w$. It is easy to verify that $(\rho(\pi_0), \rho(\pi_1)) \in R_X$, that $\mathfrak{h}(\pi_{\geq 1}) \in \rho(\pi_1)$ and that $(c, \mathfrak{h}(\pi_{\geq 1})) \in r_X$.

Case 2: $\rho(w)$ is removed from S' by rule 2 as it contains an unfulfillable eventuality of the form $\alpha \mathcal{U} \beta$.

Thus there is $a \in \rho(w)$ such that $\alpha \mathcal{U} \beta \in a$, but $\beta \notin a$ and there is no sequence of instances

$$\langle (c_0, h_0, \alpha \mathcal{U} \beta), (c_1, h_1, \alpha \mathcal{U} \beta), \ldots, (c_n, h_n, \alpha \mathcal{U} \beta) \rangle$$

with $n \geq 0$ such that $c_0 = \rho(w)$, $h_0 = a$, $\alpha_0 = \alpha$ each $h_i \in c_i \in S_0$, each $(c_i, c_{i+1}) \in R_X$, each $(h_i, h_{i+1}) \in r_X$ and with $\beta \in h_n$.

As $a \in \rho(w)$, there exists π in B such that $\pi_0 = w$ and $\mathfrak{h}(\pi) = a$. Hence $\pi \vDash \alpha \mathcal{U} \beta$ and so there exists a non-negative integer i such that $\pi_{\geq i} \vDash \beta$. Hence the instance $(\rho(\pi_i), \mathfrak{h}(\pi_{\geq i}), \alpha \mathcal{U} \beta)$ is directly fulfilled, and so such a sequence of instances does exist:

$$\langle (\rho(\pi_0), \mathfrak{h}(\pi), \alpha \mathcal{U} \beta), (\rho(\pi_1), \mathfrak{h}(\pi_{\geq 1}), \alpha \mathcal{U} \beta), \ldots, (\rho(\pi_i), \mathfrak{h}(\pi_{\geq i}), \alpha \mathcal{U} \beta) \rangle.$$

By contradiction, $\rho(w)$ is not removed.

Case 3: $\rho(w)$ is removed from S' by rule 3 as it contains an unfulfillable eventuality of the form $\neg \mathbf{R} \alpha$. Thus there is $a \in \rho(w)$ such that $\neg \mathbf{R} \alpha \in a$, but $\alpha \in a$ and there is no sequence of instances

$$\langle (c_0, h_0, \neg \mathbf{R} \alpha_0) (c_1, h_1, \neg \mathbf{R} \alpha_1), , \ldots, (c_n, h_n, \neg \mathbf{R} \alpha_n) \rangle$$

with $n \geq 0$ such that $c_0 = \rho(w)$, $h_0 = a$, $\alpha_0 = \alpha$ each $h_i \in c_i \in S_0$, each $(c_i, c_{i+1}) \in R_X$, each $(h_i, h_{i+1}) \in r_X$, each $\mathbf{R}\alpha_{i+1} = \bigcirc_{h_i}^{-1}(\mathbf{R}\alpha_i)$ and $(c_n, h_n, \neg\mathbf{R}\alpha_n)$ being directly fulfilled.

Since $a \in \rho(w)$ there exists π with $a = \mathfrak{h}(\pi), \pi \in B, \pi_0 = w$.

Now $\neg\mathbf{R}\alpha \in a = \mathfrak{h}(\pi)$ so $\pi \not\models \mathbf{R}\alpha$. If $\pi \not\models \alpha$ then $\neg\alpha \in a$ and $(c_0, h_0, \neg\mathbf{R}\alpha_0)$ is directly fulfilled. If $\pi \models \alpha$ then for some i there exists an i-deviation $\sigma \in B$ from π such that $\sigma \not\models \alpha$. We see that $\sigma_{\geq i+1} \not\models \alpha_{i+1}$ and $\sigma_{\geq i+1}$ is failure-free. As $\pi_i = \sigma_i$ and $\sigma_{\geq i} \not\models \alpha_i$ it follows that $\pi_{\geq i} \not\models \mathbf{A}\bigcirc\mathbf{O}\alpha_{i+1}$ and so $\mathbf{A}\bigcirc\mathbf{O}\Xi(\alpha_{i+1}) \notin \mathfrak{h}(\pi_{\geq i})$. Hence $(\rho(\pi_i), \mathfrak{h}(\pi_{\geq i}), \neg\mathbf{R}\alpha_i)$ is directly fulfilled. Thus such a sequence does exist:

$$\langle(\rho(\pi), \mathfrak{h}(\pi_{\geq 1}), \neg\mathbf{R}\alpha), (\rho(\pi_1), \mathfrak{h}(\pi_{\geq 1}), \neg\mathbf{R}\alpha_1), \ldots, (\rho(\pi_i), \mathfrak{h}(\pi_{\geq i}), \neg\mathbf{R}\alpha_i)\rangle$$

By contradiction, $\rho(w)$ is not removed. □

6. Pair-RoCTL

Recall that reasoning about CTL* is harder than CTL. For example, satisfiability checking for CTL* is double-exponentially complete [24, 25, 26, 13], whereas this problem for CTL can be decided in singly exponential time. For this reason it is of interest to find a CTL-like restriction of RoCTL*. One possibility is to require (as with CTL) that every temporal operator be paired with a path quantifier.

6.1. Definition of Pair-RoCTL

We define Pair-RoCTL or RoCTL$^{\mathrm{P}}$ as a fragment of RoCTL* that is CTL like in the sense that every path operator is paired with a path quantifier and vice-versa. Formally, the syntax of RoCTL$^{\mathrm{P}}$ is defined by the following syntax:

$$\begin{aligned} \phi &:= \phi \wedge \phi \mid \neg\phi \mid p \mid \mathbf{A}\psi \mid \mathbf{O}\psi \mid \mathbf{R}\psi \\ \psi &:= \phi \overline{\mathcal{U}} \phi \mid \phi \mathcal{U} \phi \mid \bigcirc\phi \end{aligned}$$

The Release operator $\overline{\mathcal{U}}$ above is defined such that $\phi \overline{\mathcal{U}} \psi \equiv \neg(\neg\phi \mathcal{U} \neg\psi)$. Note that the addition of $\overline{\mathcal{U}}$ to the BNF above is a convenience so that we do not have to include the three path operators $\mathbf{E}$, $\mathbf{P}$, and $\mathbf{Я}$ as $\mathbf{E}(\phi\mathcal{U}\psi)$, $\mathbf{P}(\phi\mathcal{U}\psi)$ and $\mathbf{Я}(\phi\mathcal{U}\psi)$ can be represented as $\neg\mathbf{A}(\neg\phi\overline{\mathcal{U}}\neg\psi)$, $\neg\mathbf{O}(\neg\phi\overline{\mathcal{U}}\neg\psi)$ and $\neg\mathbf{R}(\neg\phi\overline{\mathcal{U}}\neg\psi)$ respectively. We will present a translation τ from CTL* into RoCTL$^{\mathrm{P}}$. We note that the translation could be a lot simpler if we only translated LTL formulas, and it is a well known result that LTL is as hard to model check as CTL* Emerson and Lei [27]. However, satisfiability checking is considerably harder for CTL* than LTL, as CTL* is 2-EXPTIME hard Vardi and Stockmeyer [13] (and complete Emerson and Jutla [26]) for double exponential time while satisfiability checking LTL is complete for polynomial space Sistla and Clarke [28].

The intuition behind the following translation is that we force the CTL* formula to be evaluated over paths consisting solely of failures. These paths cannot be deviations as deviations have a failure-free suffix. Thus $\mathbf{Я}\phi$ will be satisfied iff ϕ is satisfied, and so we can pair each $\bigcirc$ or $\mathcal{U}$ operator with a $\mathbf{Я}$ without changing whether the formula is satisfied.

Requiring that $\Box\bigcirc\mathbf{v}$ hold along a path would ensure that the path is fully failing; however, $\mathbf{v}$ is a special atom which does not occur in RoCTL* formulas. We instead create an atom f such that f is only true when the last transition was a failure,[2] and the subset of the states where f is true form a CTL model. We then use f in place of $\mathbf{v}$.

When comparing RoCTL* and CTL* we will only consider $\text{RoCTL}_{\mathbf{v}}$ structures. This means that

1. we can check that we are being evaluated over a fully failing path (one that satisfies $\Box f$) by use of $\mathbf{Я}\Box f$; and
2. we can represent any linear temporal operator (e.g. $\Diamond$) in RoCTL^{P} by prefixing it with $\mathbf{Я}$ (e.g. we translate $\Diamond$ into $\mathbf{Я}\Diamond$)

We will now define a translation τ from CTL* to RoCTL^{P}. The intention is that $\tau(\phi)$ is satisfiable iff ϕ is satisfiable.

Definition 51. We define a function τ from CTL* formulas to RoCTL^{P} formulas as follows:

[[Tim suggests making this Itemised. However, the formula seem to big to itemise correctly and I would need to strip out the explanation.Let ϕ be a CTL* formula in Negation Normal Form (NNF). We will begin by defining κ_f,τ_y, prev, τ_{prev}, τ_1 and τ_p which we will use to define a translation function τ. The RoCTL^{P} formula κ_f ensures that the f atom remains false once it becomes false, that f is only true if the last transition was a failure transition and that the subset of the worlds S that satisfy f true is serial.

$$\begin{aligned}\kappa_f = \mathbf{A}\Box(\neg f \rightarrow \mathbf{A}\bigcirc\neg f)\wedge \\ \mathbf{A}\Box\mathbf{O}\bigcirc\neg f \wedge \mathbf{A}\Box(f \rightarrow \mathbf{E}\bigcirc f)\ .\end{aligned}$$

The RoCTL^{P} formula $\tau_y(\phi)$, defined below, is used to encode the state-formulas of an arbitrary CTL* formula ϕ into atoms. It ensures that each atom of the form $y_{\mathbf{A}\psi}$ is only true at those states that satisfy $\mathbf{A}\psi$. Likewise each atom of the form $y_{\mathbf{E}\psi}$ is only true at those states that satisfy $\mathbf{E}\psi$. Note that we do not require that $y_{\mathbf{A}\psi}$ be true at states that satisfy $\mathbf{A}\psi$. This is because $y_{\mathbf{A}\psi}$ occurs only positively in $\tau_1(\phi)$, which will be defined below, and so making $y_{\mathbf{A}\psi}$ false will never make $\tau_1(\phi)$ to be true. Thus a requirement that $\mathbf{A}\psi \implies y_{\mathbf{A}\psi}$ would

[2] If we are using $\text{RoCTL}_{\mathbf{v}}$ structures then this means $\bigcirc f$ will be true only where $\bigcirc\mathbf{v}$ is true. Note that as Pair-RoCTL is a restriction of RoCTL*, Pair-RoCTL formulas do not include the atom $\mathbf{v}$.

be redundant when testing for satisfiability.

$$\tau_y(\phi) = \bigwedge_{\mathbf{A}\psi \sqsubseteq \phi} \mathbf{A}\Box\left((y_{\mathbf{A}\psi} \wedge (\mathbf{\mathfrak{R}}\Box f)) \rightarrow \tau_1(\psi)\right) \wedge \bigwedge_{\mathbf{E}\psi \sqsubseteq \phi} \mathbf{A}\Box\left(y_{\mathbf{E}\psi} \rightarrow \mathbf{E}\bigcirc\mathrm{prev}\left(\tau_1\left(\psi\right)\right)\right) \ .$$

We now define the function prev from RoCTL$^{\mathrm{P}}$ formulas to RoCTL$^{\mathrm{P}}$ formulas with the intention that $\bigcirc\mathrm{prev}\left(\psi\right) \leftrightarrow \psi$ on all paths through our structure. By $[p/p']$ we denote replacing the atom p with p', and by $[\forall p \sqsubseteq \psi : p/p']$ we denote similarly replacing all atoms.

$$\mathrm{prev}\left(\psi\right) = \psi[\forall p \sqsubseteq \psi : p/p']$$

The translation above replaces each occurrence of p with p', relying on p' being true exactly when p was true at the last state. We define the function τ_{prev} below from RoCTL$^{\mathrm{P}}$ formulas to RoCTL$^{\mathrm{P}}$ formulas for the purpose of ensuring that this holds for each atom of the form p' that occurs in some formula ψ. We denote α is a subformula of β by $\alpha \sqsubseteq \beta$.

$$\tau_{\mathrm{prev}}\left(\psi\right) = \mathbf{A}\Box\left(\bigwedge_{p' \sqsubseteq \psi}\left(p \rightarrow \mathbf{A}\bigcirc p'\right) \wedge \bigwedge_{p' \sqsubseteq \psi}\left(\neg p \rightarrow \mathbf{A}\bigcirc \neg p'\right)\right) \ .$$

We can now define τ_p, which adds all the new atoms required by combining κ_f τ_y and τ_{prev}.

$$\tau_p(\phi) = \kappa_f \wedge \tau_y\left(\phi\right) \wedge \tau_{\mathrm{prev}}\left(\tau_y\left(\phi\right)\right) \ .$$

We now define the τ_1 translation. This is the core of the τ translation, but it depends on atoms introduced by the τ_p translation for correctness.

$$\begin{aligned}
\tau_1(\mathbf{A}\psi) &= y_{\mathbf{A}\psi} \wedge \mathbf{\mathfrak{R}}\Box f \\
\tau_1(\mathbf{E}\psi) &= y_{\mathbf{E}\psi} \wedge \mathbf{\mathfrak{R}}\Box f \\
\tau_1(\bigcirc\psi) &= (\mathbf{\mathfrak{R}}\bigcirc\tau_1(\psi)) \\
\tau_1(\psi * \phi) &= \mathbf{\mathfrak{R}}[\tau_1(\psi) * \tau_1(\phi)], * \in \mathcal{U}, \vec{\mathcal{U}} \\
\tau_1(p) &= (p \wedge \mathbf{\mathfrak{R}}\Box f) \\
\tau_1(\neg p) &= (\neg p \wedge \mathbf{\mathfrak{R}}\Box f)
\end{aligned}$$

Finally we define $\tau\left(\phi\right)$ itself as follows:

$$\tau\left(\phi\right) = \tau_1\left(\phi\right) \wedge \tau_p\left(\phi\right) \ .$$

6.2. Translating a CTL Model into a RoCTL^P Model*

Recall Definition 11 of CTL-structures. Given a CTL-structure $M = (S, R, g)$, we construct a RoCTL-structure $M^{\mathfrak{R}} = \left(S^{\mathfrak{R}}, R^{\mathfrak{R}}, g^{\mathfrak{R}}\right)$ from M as follows:

- we add a new "success" world s so that $S^{\mathfrak{R}} = S \cup \{s\}$
- the accessibility relation $R^{\mathfrak{R}}$ is the least relation that satisfies $R^{\mathfrak{R}} \supseteq R$ and $\langle w, s\rangle \in R^{\mathfrak{R}}$ for all $w \in S^{\mathfrak{R}}$
- the valuation $g^{\mathfrak{R}}$ satisfies the following:
  - for every atom p in the original formula ϕ and $w \in S$ we have $p \in g^{\mathfrak{R}}(w)$ iff $p \in g(w)$
  - the failure atom f and violation atom $\mathbf{v}$ is true at every world except the success world. Formally, $f \in g^{\mathfrak{R}}(w)$ iff $w \neq s$ and $\mathbf{v} \in g^{\mathfrak{R}}(w)$ iff $w \neq s$
  - for every atom of the form y_ψ in $\tau_y(\phi)$ and every world $w \in S$ it is the case that $y_\psi \in g^{\mathfrak{R}}(w)$ iff $M, w \vDash \psi$.

Note that we need the f atom as the $\mathbf{v}$ atom cannot explicitly appear in any RoCTL* or RoCTL$^{\mathrm{P}}$ formula.

We will now define a function h to add the p' atoms into the model.

Definition 52. We define a function h from RoCTL$_{\mathbf{v}}$ structures to RoCTL$_{\mathbf{v}}$ structures such that for any RoCTL$_{\mathbf{v}}$ structure $M = (S, R, g)$ we have $h(M) = \left(S^h, R^h, g^h\right)$ where:

1. $S^h = R$, so every world in $h(M)$ is of the form (w, v) where w and v are in S (that is worlds of M). Being in world (w, v) means roughly "we are currently at world v but were at world w previously"
2. for any pair of worlds (w, v) and (x, y) in S^h we have $(w, v)\, R^h\, (x, y)$ iff $x = v$ and $(x, y) \in R$
3. for all $p \in \mathbb{V}$, it is the case that $p \in g^h(\langle w, v\rangle) \iff p \in g(v)$ and $p' \in g^h(\langle w, v\rangle) \iff p \in g(w)$

We will use $\langle ?, w\rangle$ to represent $\langle v, w\rangle$ for some arbitrary v, when we do not care about truth values of the p' atoms at this world. For convenience we extend the definition of h such that $h(\sigma) = \langle ?, \sigma_0\rangle, \langle \sigma_0, \sigma_1\rangle, \langle \sigma_1, \sigma_2\rangle \ldots$, and $h(M, \sigma) = h(M), h(\sigma)$.

Lemma 53. *For all RoCTL$_{\mathbf{v}}$ structures M, fullpaths σ through M and RoCTL* formulas ϕ that do not contain atoms of the form p' we have $M, \sigma \vDash \phi \iff h(M, \sigma) \vDash \phi \iff h(M, \sigma) \vDash \bigcirc prev(\phi)$. If each atom p' is true exactly when p was true at the previous world (or $M, \sigma \vDash \tau_{prev}(\phi)$) for some formula ϕ then $h(M, \sigma) \vDash \phi \iff h(M, \sigma) \vDash \bigcirc prev(\phi)$.*

Proof. We have shown that RoCTL$^*_{\mathbf{v}}$ is bisimulation invariant in M$^{\mathrm{c}}$Cabe-Dansted et al. [6], and so it is clear that $M, \sigma \vDash \phi \iff h(M, \sigma) \vDash \phi$. It is easy to see that $h(M, \sigma) \vDash p \iff h(M, \sigma) \vDash \bigcirc p'$, or in other words that $h(M, \sigma) \vDash \psi \iff h(M, \sigma) \vDash \bigcirc \mathrm{prev}(\psi)$ for ψ of length 1. Thus it is clear from induction on the length of formula and the semantics of RoCTL* that $h(M, \sigma) \vDash \psi \iff h(M, \sigma) \vDash \bigcirc \mathrm{prev}(\psi)$ for ψ of any length. □

Definition 54. Given a model M for an RoCTL$^{\mathrm{P}}$ formula $\tau(\phi)$ we construct a model M^C for the CTL* formula ϕ as follows: remove all worlds where f is false.

6.3. Proof of Correctness

Without loss of generality we can assume that each structure has a world w_0 such that every other world is reachable from that world, so for example $M, w_0 \vDash \mathbf{A}\Box p$ means that p is true at all worlds in M.

When we use "$(\neg)$" in a sentence this indicates that the sentence remains true if all occurrences of $(\neg)$ is replaced with a $\neg$ or if all occurrences of $(\neg)$ are simply removed. This is similar to how the $\pm$ operator is frequently used.

Lemma 55. *For any structure M that satisfies $M, w_0 \vDash \mathbf{A}\Box\mathbf{O}\bigcirc(\neg f)$, it is the case that $M, \pi \vDash \mathbf{\Lambda}\Box f \implies M, \pi \vDash \Box f$ and $M, \pi \vDash \mathbf{\Lambda}\tau_1(\psi) \implies M, \pi \vDash \tau_1(\psi)$ for all paths π through M and formulas ψ.*

Proof. As $M, \pi \vDash \mathbf{\Lambda}\Box f$ either $M, \pi \vDash \Box f$ or there exists a deviation σ from π that satisfies $\Box f$. Any deviation σ from π has a failure-free suffix. That is there exists i such that $\sigma_{\geq i}$ is failure-free. As $M, w_0 \vDash \mathbf{A}\Box\mathbf{O}\bigcirc\neg f$, it is the case that $M, \sigma_{\geq i} \vDash \mathbf{O}\bigcirc\neg f$, and as $\sigma_{\geq i}$ is failure-free, $M, \sigma_{\geq i+1} \vDash \neg f$ and so $M, \sigma \nvDash \Box f$. Hence, by contradiction, $M, \pi \vDash \Box f$.

As any deviation π from σ has a failure-free suffix and $M, \sigma \nvDash \Box f$, we see that $M, \sigma \nvDash ((\neg)\, p \wedge \mathbf{\Lambda}\Box f)$ for any $p \in \mathbb{V}$. Thus we see that $M, \sigma \nvDash \tau_1(\psi)$ for any ψ of length 1. It is easy to see by recursion on length of ψ that $M, \sigma \nvDash \tau_1(\psi)$ for ψ of any length. It follows that $M, \pi \vDash \mathbf{\Lambda}\tau_1(\psi) \implies M, \pi \vDash \tau_1(\psi)$. □

For the next lemma, recall that s was the success world defined at the beginning of Section 6.2.

Lemma 56. *Say that $M^{\mathfrak{R}}, \sigma \vDash \tau_1(\psi)$ for some path σ through $M^{\mathfrak{R}}$, then $s \neq \sigma_i$ for any non-negative integer i.*

Proof. For any fullpath π such that $M^{\mathfrak{R}}, \pi \nvDash \mathbf{\Lambda}\Box f$ we see that for ψ of the form $\mathbf{A}\theta$, $\mathbf{E}\theta$, p or $\neg p$ it is the case that $M^{\mathfrak{R}}, \pi \nvDash \tau_1(\psi)$. If there exists an integer i such that $\sigma_i = s$ we see that $M^{\mathfrak{R}}, \sigma_j \nvDash \Box f$ for any $j \in \mathbb{N}$, and hence by Lemma 55 above we have $M^{\mathfrak{R}}, \sigma_j \nvDash \mathbf{\Lambda}\Box f$. By induction, we see that $M^{\mathfrak{R}}, \sigma \nvDash \tau_1(\psi)$, for any formulas ψ. □

Lemma 57. *For any CTL* formula ϕ, CTL-structure M and fullpath π through M we have $h\left(M^{\mathfrak{R}}, \pi\right) \vDash \tau(\phi)$ if $M, \pi \vDash \phi$.*

Proof. Let $H(\psi)$ be the statement: for any CTL-structure M and fullpath π through M we have $h\left(M^{\mathfrak{R}}, \pi\right) \vDash \tau(\psi)$ if $M, \pi \vDash \psi$. We will now show $H(\psi)$ is true for all ψ such that $|\psi| = 1$, that is all ψ that consist of a single atom.

It is easy to see that for every path σ through M we have $M^{\mathfrak{R}}, \sigma \vDash \mathbf{\Lambda}\Box f$ and so for every $p \in \mathbb{V}$, it is the case that $M, \sigma \vDash (\neg)\, p \implies M^{\mathfrak{R}}, \sigma \vDash ((\neg)\, p \wedge \mathbf{\Lambda}\Box f)$. Thus for all formulas ψ that consist of a single atom it is the case that $M, \sigma \vDash \psi \implies M^{\mathfrak{R}}, \sigma \vDash \tau_1(\psi)$. Say that $M, \sigma \vDash \psi \implies h\left(M^{\mathfrak{R}}, \sigma\right) \vDash \tau_1(\psi)$ for all ψ

of length less than some integer n. Now we assume that $H(\psi)$ is true for all ψ such that $|\psi| \leq n$ for some positive integer n, and will prove that $H(\psi)$ holds for any ψ such that $|\psi| = n+1$.

$\psi = \alpha \mathcal{U} \beta$: For all σ it is the case that $M, \sigma \vDash \alpha \implies M^{\mathfrak{R}}, \sigma \vDash \tau_1(\alpha)$ and $M, \sigma \vDash \beta \implies M^{\mathfrak{R}}, \sigma \vDash \tau_1(\beta)$. Say that $M, \sigma \vDash \alpha \mathcal{U} \beta$ then there exists an integer i such that $M, \sigma_{\geq i} \vDash \beta$ and for all integers j less than i we have $M, \sigma_{\geq j} \vDash \alpha$. Thus $M^{\mathfrak{R}}, \sigma_{\geq i} \vDash \tau_1(\beta)$ and for all j less than i we have $M^{\mathfrak{R}}, \sigma_{\geq j} \vDash \tau_1(\alpha)$. Hence $M^{\mathfrak{R}}, \sigma \vDash \tau_1(\alpha) \mathcal{U} \tau_1(\beta)$, so $M^{\mathfrak{R}}, \sigma \vDash \mathfrak{R}(\tau_1(\alpha) \mathcal{U} \tau_1(\beta)) = \tau_1(\psi)$.

$\psi = \alpha \overline{\mathcal{U}} \beta$: Say that $M, \sigma \vDash \alpha \overline{\mathcal{U}} \beta$. Then $M, \sigma \vDash \neg(\neg\alpha \mathcal{U} \neg\beta)$, and so for all $i \in \mathbb{N}$ either $M, \sigma_{\geq i} \nvDash \neg\beta$ or there exists $j < i$ such that $M, \sigma_{\geq j} \nvDash \neg\alpha$. Thus $M^{\mathfrak{R}}, \sigma_{\geq i} \vDash \tau_1(\beta)$ or there exists $j < i$ such that $M^{\mathfrak{R}}, \sigma_{\geq j} \vDash \tau_1(\alpha)$ and so $M^{\mathfrak{R}}, \sigma \vDash \neg(\tau_1(\neg\alpha) \mathcal{U} (\neg\beta))$. Hence $M^{\mathfrak{R}}, \sigma \vDash \mathfrak{R}\neg(\tau_1(\neg\alpha) \mathcal{U} (\neg\beta)) = \tau_1(\alpha \overline{\mathcal{U}} \beta)$

$\psi = \bigcirc\alpha$: if $M, \sigma \vDash \bigcirc\alpha$ then $M, \sigma_{\geq 1} \vDash \alpha$. Thus $M^{\mathfrak{R}}, \sigma_{\geq 1} \vDash \tau_1(\alpha)$, and so $M^{\mathfrak{R}}, \sigma \vDash \bigcirc\tau_1(\alpha) = \tau_1(\psi)$.

$\psi = \mathbf{E}\alpha$: By definition, $M, \sigma \vDash \mathbf{E}\alpha$ iff $M^{\mathfrak{R}}, \sigma \vDash y_{\mathbf{E}\alpha}$. As σ is a path through M, it does not contain the success state and so $M^{\mathfrak{R}}, \sigma \vDash \Box f$ and so if $M, \sigma \vDash \mathbf{E}\alpha$ then $M^{\mathfrak{R}}, \sigma \vDash y_{\mathbf{E}\alpha} \wedge \mathfrak{R}\Box f = \tau_1(\psi)$.

$\psi = \mathbf{A}\alpha$: As above.

It is now clear from induction that $M^{\mathfrak{R}}, \pi \vDash \tau_1(\phi)$. We see that case for $\mathbf{A}\alpha$ and $\mathbf{E}\alpha$ above are trivial. The complexity was taken outside τ_1 in the form:

$$\bigwedge_{\mathbf{A}\psi \sqsubseteq \phi} \mathbf{A}\Box((y_{\mathbf{A}\psi} \wedge (\mathfrak{R}\Box f)) \rightarrow \tau_1(\psi))$$

$$\bigwedge_{\mathbf{E}\psi \sqsubseteq \phi} \mathbf{A}\Box(y_{\mathbf{E}\psi} \rightarrow \mathbf{E}\bigcirc\mathrm{prev}(\tau_1(\psi)))$$

Now $y_{\mathbf{E}\psi}$ occurs exactly on those σ_0 where $M, \sigma \vDash \mathbf{E}\psi$. Thus $M^{\mathfrak{R}}, \sigma \vDash \mathbf{E}\tau_1(\psi)$ and $h\left(M^{\mathfrak{R}}, \sigma\right) \vDash \mathbf{E}\tau_1(\psi)$, thus $h\left(M^{\mathfrak{R}}, \sigma\right) \vDash \mathbf{E}\bigcirc\mathrm{prev}(\tau_1(\psi))$. Likewise $y_{\mathbf{A}\psi}$ occurs exactly on those σ_0 where $M, \sigma \vDash \mathbf{A}\psi$, that is, for every path σ' through M such that $\sigma'_0 = \sigma_0$ it is the case that $M, \sigma \vDash \psi$ and so $M^{\mathfrak{R}}, \sigma \vDash \tau_1(\psi)$ and $M^{\mathfrak{R}}, \sigma \vDash (y_{\mathbf{A}\phi} \wedge (\mathfrak{R}\Box f)) \rightarrow \tau_1(\psi)$. If the path σ is not through M, it contains the success state s, and so $\sigma \nvDash \mathfrak{R}\Box f$ and so again $M^{\mathfrak{R}}, \sigma \vDash (y_{\mathbf{A}\phi} \wedge (\mathfrak{R}\Box f)) \rightarrow \tau_1(\psi)$.

It is now easy to show that the way $M^{\mathfrak{R}}$ is constructed ensures that $h\left(M^{\mathfrak{R}}, \sigma\right) \vDash \tau_p(\phi)$, and since $h\left(M^{\mathfrak{R}}, \sigma\right) \vDash \tau_1(\phi)$, it follows that $h\left(M^{\mathfrak{R}}, \sigma\right) \vDash \tau(\phi)$.

□

Given any CTL-structure M we can construct the RoCTL$_{\mathbf{v}}$ structures $M^{\mathfrak{R}}$ and $h\left(M^{\mathfrak{R}}\right)$. From Lemma 57 above it is clear that $\tau(\phi)$ is satisfiable if ϕ

is satisfiable. We have not yet shown that for any RoCTL$_{\mathbf{v}}$ structure we can construct an equivalent CTL-structure. We will now deal with constructing a CTL-structure from an arbitrary RoCTL$_{\mathbf{v}}$ structure; from Lemma 58 below it is clear that ϕ is satisfiable, then $\tau(\phi)$ is satisfiable. By combining these we get Theorem 59, with which we will conclude this section.

Lemma 58. *For any RoCTL$_{\mathbf{v}}$ structure structure M, If $M, w_0 \vDash \tau(\phi)$ then $M^C, w_0 \vDash \phi$.*

Proof. Recall that M^C is the translation of M defined in Definition 54; that is the structure M with the worlds that do not satisfy f removed.

As $M \vDash \tau(\phi)$, clearly $M \vDash \tau_p(\phi)$. Since $M \vDash \tau_p(\phi)$ we see that $M, w_0 \vDash \mathbf{A}\Box\mathbf{O}\bigcirc(\neg f)$ from Lemma 55 and again it is the case that $M, \pi \vDash \mathbf{\mathfrak{R}}\Box f \implies M, \pi \vDash \Box f$ and $M, \pi \vDash \mathbf{\mathfrak{R}}\tau_1(\psi) \implies M, \pi \vDash \tau_1(\psi)$ for all CTL* formulas ψ and fullpaths π through M.

Say that for all paths σ through M and all CTL* formulas ψ with $|\psi| \leq n$ for some integer n, it is the case that $M, \sigma \vDash \tau_1(\psi) \implies M, \sigma \vDash \psi$. Now consider the case where $|\psi| = n+1$. Where ψ is of the form $(\neg)\, p$, $\alpha\overrightarrow{\mathcal{U}}\beta$, $\alpha\mathcal{U}\beta$ or $\bigcirc\alpha$ we see that $M, \sigma \vDash \tau_1(\psi) \implies M, \sigma \vDash \psi$ using the same arguments made in the previous lemma.

Now say ψ is of the form $\mathbf{A}\theta$, and say $M, \sigma \vDash \tau_1(\psi) = y_{\mathbf{A}\theta} \wedge \mathbf{\mathfrak{R}}\Box f$. From τ_y we know that $\mathbf{A}\Box((y_{\mathbf{A}\theta} \wedge (\mathbf{\mathfrak{R}}\Box f)) \rightarrow \tau_1(\theta))$. Consider a path π through M^C such that $\pi_0 = \sigma_0$, i.e. $\pi \in \delta^\omega(\sigma_0)$. We know that $M^C, \pi_0 \vDash \Box f$ as f is true at every world in M^C, so likewise $M, \pi \vDash \Box f$ and $M, \pi \vDash \mathbf{\mathfrak{R}}\Box f$. From $\tau_1(\phi)$ we know that $M, \pi \vDash y_{\mathbf{A}\theta}$ and from τ_y we know that $\mathbf{A}\Box((y_{\mathbf{A}\theta} \wedge (\mathbf{\mathfrak{R}}\Box f)) \rightarrow \tau_1(\theta))$, hence $M, \pi \vDash \tau_1(\theta)$. Since $|\theta| \leq n$ it follows that $M^C, \pi \vDash \theta$, for all $\pi \in \delta^\omega(\sigma_0)$. Thus $M^C, \sigma \vDash \mathbf{A}\theta$.

Say ψ is of the form $\mathbf{E}\theta$ and $M, \sigma \vDash \tau_1(\psi) = y_{\mathbf{E}\theta} \wedge \mathbf{\mathfrak{R}}\Box f$. As $M, \sigma \vDash y_{\mathbf{E}\theta}$, from τ_y we know that $M, \sigma \vDash \mathbf{E}\bigcirc\text{prev}(\tau_1(\theta))$. As $M, \sigma \vDash \tau_{\text{prev}}(\tau_y(\phi))$, we see that $M, \sigma \vDash \mathbf{E}(\tau_1(\theta))$. Finally, since $|\theta| \leq n$ we know that $M, \sigma \vDash \mathbf{E}\theta$. □

Theorem 59. *$\tau(\phi)$ is satisfiable in RoCTL^P* iff ϕ is satisfiable in CTL*.

6.4. Model Checking

Given a formula ϕ, and a model checking procedure for RoCTL$^{\text{P}}$ we can compute the RoCTL$_{\mathbf{v}}$ structure $M^{\mathfrak{R}}$ from the CTL-structure M as follows.

First we add the f atom as above. Then, for subformulas α of the form $\mathbf{A}\psi$ (or $\mathbf{E}\psi$) we perform the following, starting with the shortest subformula α. For each world t in $M^{\mathfrak{R}}$ we pick an arbitrary world $\langle s, t\rangle$ in $h\left(M^{\mathfrak{R}}\right)$ and model check $\mathbf{E}\bigcirc\text{prev}(\tau_1(\psi))$, if this formula holds at $\langle s, t\rangle$ we add y_ϕ to the valuation of the world t.

We now have the model $h\left(M^{\mathfrak{R}}\right)$, such that $h\left(M^{\mathfrak{R}}\right) \vDash \tau(\phi)$ iff $M \vDash \phi$. Thus where $m = |\phi|$ and n is the number of worlds in M we can easily reduce the problem of model checking a CTL* formula to mn model checking problems of RoCTL$^{\text{P}}$ formula of length $\mathcal{O}(m)$ on models with no more than n^2 worlds. As the model checking algorithm for CTL* is PSPACE-complete Clarke et al. [15], the model checking problem for RoCTL$^{\text{P}}$ is PSPACE-hard.

6.5. Expressivity

Previously we have given a translation from CTL* to Pair-RoCTL that preserves satisfiability. However, the translation was not expressively equivalent to the original, and a satisfiability preserving translation tells us nothing about the expressivity of the language. After all, we can easily construct a satisfiability preserving translation from CTL* to $\top$ and $\bot$. Here we will briefly define a translation that is truth preserving, but has a singly exponential blowup in the size of the resulting formulas.

We now present a translation function τ from CTL* formulas to Pair-RoCTL formulas such that for all structures M and paths σ it is the case that $M^{\mathfrak{R}}, \sigma \vDash \tau(\phi)$ iff $M, \sigma \vDash \phi$ is satisfiable; where $M^{\mathfrak{R}}$ is $\text{RoCTL}_{\mathbf{v}}$ structure generated from M as by adding a success world s and f, $\mathbf{v}$ atoms as in Section 6.2. Unlike Section 6.2 we need not at any atoms of the form $y_{\mathbf{A}\psi}$, $y_{\mathbf{E}\psi}$ or p' to $M^{\mathfrak{R}}$.

It is well known that for we can split a formula ϕ into state formulas and formulas that are true at the next step, that is into the form:

$$\bigvee_i \alpha_i \wedge \bigcirc \psi_i \ ,$$

where α is a list of state formulas and ψ is a list of formulas. Informally, we will define $\tau_1(\mathbf{A}\phi)$ and $\tau_1(\mathbf{E}\phi)$ in terms the above split as follows:

$$\begin{aligned} \tau_1(\mathbf{A}\phi) &= \bigvee_i \tau_1(\alpha_i) \wedge \mathbf{A}\bigcirc\tau_1(\psi_i) \\ \tau_1(\mathbf{E}\phi) &= \bigvee_i \tau_1(\alpha_i) \wedge \mathbf{E}\bigcirc\tau_1(\psi_i) \ . \end{aligned}$$

We will now define such a split so that we can provide a precise definition of this τ_1.

Recall Definition 19 of $\bigcirc_a^{-1}$, a set of formula translation functions which has the following property:

Lemma 60. *Let σ be an arbitrary path, ϕ an arbitrary formula and again let Φ be the set of all state subformulas of ϕ. Then $\sigma \vDash \phi$ iff $\sigma_{\geq 1} \vDash \bigcirc_a^{-1}(\phi)$.*

This lemma (and its proof) is very similar to Lemma 28.

Definition 61. Let Φ be the set of all state subformulas of ϕ. Then we define a function **split** from formulas to formulas such that for any formula ϕ

$$\textbf{split}(\phi) = \bigvee_{a \in 2^{\Phi}} \left(\left(\bigwedge_{\psi \in a} \psi \right) \wedge \left(\bigwedge_{\psi \in (\Phi - a)} \neg\psi \right) \wedge \bigcirc\bigcirc_a^{-1}(\phi) \right)$$

Lemma 62. *Let σ be an arbitrary path, ϕ an arbitrary formula. Then $\sigma \vDash \phi$ iff $\sigma \vDash \textbf{\textit{split}}(\phi)$.*

Proof. We see that exactly one clause of the form

$$\left(\bigwedge_{\psi\in a}\psi\right)\wedge\left(\bigwedge_{\psi\in(\Phi-a)}\neg\psi\right)$$

will hold along σ: the clause where for each $\psi\in\Phi$ we have $\psi\in a$ iff $M,\sigma\vDash\psi$. Thus $M,\sigma\vDash\mathbf{split}\left(\phi\right)$ iff $M,\sigma\vDash\bigcirc_a^{-1}\left(\phi\right)$ for this a, and so from Lemma 60 we know that $M,\sigma\vDash\phi$. □

The formula translation function τ_1 is defined as follows:

$$\begin{aligned}\tau_1\left(\mathbf{A}\phi\right) &= \bigvee_{a\in 2^\Phi}\left(\left(\bigwedge_{\psi\in a}\psi\right)\wedge\left(\bigwedge_{\psi\in(\Phi-a)}\neg\psi\right)\wedge\mathbf{A}\bigcirc\tau_1\left(\bigcirc_a^{-1}\left(\phi\right)\right)\right)\\ \tau_1\left(\mathbf{E}\phi\right) &= \bigvee_{a\in 2^\Phi}\left(\left(\bigwedge_{\psi\in a}\psi\right)\wedge\left(\bigwedge_{\psi\in(\Phi-a)}\neg\psi\right)\wedge\mathbf{E}\bigcirc\tau_1\left(\bigcirc_a^{-1}\left(\phi\right)\right)\right)\ ,\end{aligned}$$

for formulas of forms other than $\mathbf{A}\phi$ and $\mathbf{E}\phi$, we define τ_1 as in Definition 51. As this new τ_1 does not add any atoms except for f we do not need most of τ_p and so τ becomes:

$$\tau\left(\phi\right) = \tau_1\left(\phi\right)\wedge\kappa_f\ ,$$

where κ_f is defined the same as in Section 6.2.

Translating a CTL* model M to RoCTL$^{\mathrm{P}}$ is now trivial, as we only have to add the $f/\mathbf{v}$ atoms to the valuation of each world of M and then add a success world s at which f and $\mathbf{v}$ are false. Given a RoCTL$^{\mathrm{P}}$ model to translate into a CTL* model, we either reject the model as inconsistent if it does not satisfy κ_f, or remove all worlds where f is false. As the translation of the structure does not depend on the formula being translated, this gives us an expressivity result.

Theorem 63. *For any CTL-structure M and fullpath σ through M we have $M^{\mathfrak{R}},\sigma\vDash\tau\left(\phi\right)\iff M,\sigma\vDash\phi$.*

The proof of correctness of this new translation is similar to the previous translation and has been omitted.

7. State-RoCTL

In this section we will show that the decision problems for State-RoCTL (RoCTL$^{\mathrm{S}}$) are of similar complexity to the corresponding decision problems for CTL. We will do this by presenting a translation of State-RoCTL into CTL. Note that as we are now translating formulas into CTL we do not have to avoid using the $\mathbf{v}$ atom as CTL (and CTL*, RoCTL$^*_{\mathbf{v}}$) formulas can contain this atom. Here we find it convenient to define $\mathbf{E}\bigcirc$ and $\mathbf{P}\bigcirc$ as base operators, as this notation allows a simpler presentation of the proofs relating to RoCTL$^{\mathrm{S}}$.

Definition 64. We define RoCTL$^{\mathrm{S}}$ formulas as follows: using

$$\alpha := \mathbf{A}\theta \mid \mathbf{E}\theta \mid \mathbf{E}\bigcirc\alpha \mid \mathbf{O}\theta \mid \mathbf{P}\theta \mid \mathbf{P}\bigcirc\alpha \mid \alpha \wedge \alpha \mid \neg\alpha \mid p$$
$$\theta := \boldsymbol{\text{Я}}\theta \mid \mathbf{R}\theta \mid \alpha\mathcal{U}\alpha \ ,$$

formulas of the form θ will not be called RoCTL$^{\mathrm{S}}$ formulas, instead they will be called RoCTL$^{\mathrm{S}}$ path-formulas. In RoCTL$^{\mathrm{S}}$ the distinction between state and path formulas is important. We will use the symbols α and β to refer to state formulas in this section. The symbol θ will be used to refer to a path-formula, and the symbols ϕ and ψ will be used to refer to formulas that may be either state or path formulas. Additionally where we could write something like $M, \sigma_{\geq i} \vDash \alpha$ we will instead write $M, \sigma_i \vDash \alpha$ to remind ourselves that, as α is a state formula, the choice of the remainder of the path is irrelevant to the truth of α.

We define similar abbreviations to CTL, for example we use $\mathbf{O}\bigcirc\alpha$ as an abbreviation for $\neg\mathbf{P}\bigcirc\neg\alpha$. Additionally, note that the first transition of a deviation can be any success or failure transition leading away from the current node. As such it is clear that $\boldsymbol{\text{Я}}\bigcirc\alpha \leftrightarrow \mathbf{E}\bigcirc\alpha$ and $\mathbf{R}\bigcirc\alpha \leftrightarrow \mathbf{A}\bigcirc\alpha$ are valid for any state formula α. Thus we treat $\boldsymbol{\text{Я}}\bigcirc\alpha$ and $\mathbf{R}\bigcirc\alpha$ as abbreviations for $\mathbf{E}\bigcirc\alpha$ and $\mathbf{A}\bigcirc\alpha$ respectively. Note that these last two abbreviations are not valid for the full RoCTL* logic, $\boldsymbol{\text{Я}}\bigcirc\alpha$ is only equivalent to $\mathbf{E}\bigcirc\alpha$ because in RoCTL$^{\mathrm{S}}$ we know that α is a state-formula.

7.1. Expressivity

We will now define a translation from RoCTL$^{\mathrm{S}}$ to CTL. To understand how the translation to CTL works, consider the formula $\boldsymbol{\text{Я}}(\alpha\mathcal{U}\beta)$. If $M, \sigma \vDash \boldsymbol{\text{Я}}(\alpha\mathcal{U}\beta)$ then either $\sigma \vDash \alpha\mathcal{U}\beta$ or there exists a deviation from σ like π in Figure 1, for some integers i and n.

Note that β does not occur on π_j for $j \leq i$ as then σ would also satisfy $\alpha\mathcal{U}\beta$. And so (as in Figure 1) we see that $M, \pi_{\geq i+1} \vDash \alpha\mathcal{U}\beta$. Since π is an i-deviation $\pi_{\geq i+1}$ is failure-free and so $\pi_{i+1} \vDash \mathbf{P}(\alpha\mathcal{U}\beta)$, thus the state formula $\mathbf{E}\bigcirc\mathbf{P}(\alpha\mathcal{U}\beta)$ holds at σ_i. Thus along the path σ the state-formula α holds until $\mathbf{E}\bigcirc\mathbf{P}(\alpha\mathcal{U}\beta)$ holds together with α. The translation will recursively push $\boldsymbol{\text{Я}}$ operators inside the $\mathcal{U}$ operator by replacing $\boldsymbol{\text{Я}}(\alpha\mathcal{U}\beta)$ with

$$\alpha\mathcal{U}(\beta \vee (\alpha \wedge \mathbf{E}\bigcirc\mathbf{P}(\alpha\mathcal{U}\beta))) \ .$$

The $\mathbf{R}$ operator is the dual of the $\boldsymbol{\text{Я}}$, so it will be handled similarly. We now formally define the translation.

Definition 65. We now define a translation function τ from RoCTL$^{\mathrm{S}}$ formulas and path-formulas to CTL-formulas and path-formulas respectively. For any atom p we let $\tau(p) = p$. For any RoCTL$^{\mathrm{S}}$ formula α we define τ such that the following equalities hold:

$$\tau(\mathbf{E}\bigcirc\alpha) = \mathbf{E}\bigcirc\tau(\alpha)$$
$$\tau(\mathbf{P}\bigcirc\alpha) = \mathbf{E}\bigcirc(\tau(\alpha) \wedge \neg\mathbf{v})$$
$$\tau(\neg\alpha) = \neg\tau(\alpha) \ .$$

Likewise, for any pair α, β of RoCTL$^{\mathrm{S}}$ formulas:

$$\begin{aligned}
\tau(\alpha \wedge \beta) &= \tau(\alpha) \wedge \tau(\beta) \\
\tau(\alpha \mathcal{U} \beta) &= \tau(\alpha)\, \mathcal{U} \tau(\beta) \\
\tau(\mathbf{R}(\alpha \mathcal{U} \beta)) &= \tau(\alpha \wedge \mathbf{A}\bigcirc\mathbf{O}(\alpha \mathcal{U} \beta))\, \mathcal{U} \tau(\beta) \\
\tau(\mathbf{Я}(\alpha \mathcal{U} \beta)) &= \tau(\alpha)\, \mathcal{U} \tau(\beta \vee (\alpha \wedge \mathbf{E}\bigcirc\mathbf{P}(\alpha \mathcal{U} \beta))) \ .
\end{aligned}$$

For any RoCTL$^{\mathrm{S}}$ path-formula θ:

$$\begin{aligned}
\tau(\mathbf{A}\theta) &= \mathbf{A}\tau(\theta) \\
\tau(\mathbf{E}\theta) &= \mathbf{E}\tau(\theta) \ .
\end{aligned}$$

The operators $\mathbf{O}$ and $\mathbf{P}$ are not in CTL so they will have to be handled differently to $\mathbf{A}$ and $\mathbf{E}$. First we will consider the case where $\alpha \mathcal{U} \beta$ is nested directly inside the $\mathbf{O}$ or $\mathbf{P}$ operators:

$$\begin{aligned}
&\tau(\mathbf{O}(\alpha \mathcal{U} \beta)) = \tau(\beta) \vee (\tau(\alpha) \wedge \mathbf{A}\bigcirc\mathbf{A}(\tau(\alpha)\, \mathcal{U}\, (\tau(\beta) \vee \mathbf{v}))) \\
&\tau(\mathbf{P}(\alpha \mathcal{U} \beta)) = \tau(\beta) \vee (\tau(\alpha) \wedge \mathbf{E}\bigcirc\mathbf{E}((\tau(\alpha) \wedge \neg\mathbf{v})\, \mathcal{U}\, (\tau(\beta) \wedge \neg\mathbf{v}))) \ .
\end{aligned}$$

We handle the case where $\mathbf{R}$ or $\mathbf{Я}$ occur inside $\mathbf{O}$ or $\mathbf{P}$ by specifying that for all path-formulas θ not of the form $\alpha \mathcal{U} \beta$:

$$\begin{aligned}
\tau(\mathbf{O}\theta) &= \tau(\mathbf{O}\tau(\theta)) \\
\tau(\mathbf{P}\theta) &= \tau(\mathbf{P}\tau(\theta)) \ ,
\end{aligned}$$

and similarly for $\mathbf{R}$ and $\mathbf{Я}$:

$$\begin{aligned}
\tau(\mathbf{R}\theta) &= \tau(\mathbf{R}\tau(\theta)) \\
\tau(\mathbf{Я}\theta) &= \tau(\mathbf{Я}\tau(\theta)) \ .
\end{aligned}$$

We see above that $\tau(\theta)$ is of the form $\alpha \mathcal{U} \beta$ and so the above definition does not require infinite recursion. For example, $\tau(\mathbf{RRR}(p \mathcal{U} q)) = \tau(\mathbf{R}\tau(\mathbf{R}\tau(\mathbf{R}(p \mathcal{U} q))))$, and we see that we can expand this from the inside out using the definition of $\tau(\mathbf{R}(\alpha \mathcal{U} \beta))$ above.

To help verify that τ is well-defined (not circular), we will define a partial ordering $<$ which is to be read as "is simpler than". We will then use this to show that the recursive definition of τ is not circular as at each step of the recursion we consider simpler formulas.

Definition 66. We define a partial ordering $<$ on RoCTL* formulas such that $\phi < \psi$ if ϕ has less $\mathbf{R}$ (or $\mathbf{Я}$) operators than ψ, and $\phi < \psi$ if ϕ and ψ have the same number of $\mathbf{R}$ operators and $|\phi| < |\psi|$.

We now show that τ is not circular, and well-defined.

Lemma 67. *The formula $\tau(\phi)$ is defined for all ϕ in the domain of τ.*

Proof. We see that for all ϕ in the domain, that is all ϕ that are either RoCTL$^{\mathrm{S}}$ formulas or path-formulas, $\tau(\phi)$ occurs in the Left-Hand Side (LHS) of the above definition. We also see that where $\tau(\phi)$ occurs on the LHS and $\tau(\psi)$ occurs on the RHS, $\psi < \phi$. Thus by induction on the number of $\mathbf{R}$ and induction on the length of the formulas we see that $\tau(\phi)$ is finite in length for all ϕ in the domain. □

Recall that $\tau(\phi)$ is a CTL formula. The following lemma demonstrates that RoCTL$^{\mathrm{S}}$ is expressively equivalent to CTL.

Lemma 68. *For all RoCTL$_{\mathbf{v}}$ structures M, fullpaths σ through M and RoCTL^{S} formulas ϕ, we have $M, \sigma \vDash \phi \iff M, \sigma \vDash \tau(\phi)$.*

Proof. For contradiction, say that ϕ is the simplest formula that provides a counter example to this lemma, that is there exists no counter example ψ such that $\psi < \phi$. In the proof of this lemma, we will use the notation $\pi_{\leq i} \cdot \sigma$ to represent the concatenation $\pi_0, \pi_1, \ldots, \pi_i, \sigma_0, \sigma_1, \ldots$ of $\pi_{\leq i}$ and σ.

We now show that ϕ does not provide a counter example for any RoCTL$_{\mathbf{v}}$ structure M and fullpath σ through M.

Case 1. Say that ϕ is of the form $\mathbf{R}(\alpha \mathcal{U} \beta)$.

($\Longrightarrow$) Say that $M, \sigma \vDash \mathbf{R}(\alpha \mathcal{U} \beta)$ and $M, \sigma \nvDash \tau(\mathbf{R}(\alpha \mathcal{U} \beta))$. Thus

$$M, \sigma \nvDash \tau(\alpha \wedge \mathbf{A}\bigcirc\mathbf{O}(\alpha \mathcal{U} \beta)) \, \mathcal{U} \tau(\beta) \ ,$$

and since ϕ is the simplest counter example and $\alpha \wedge \mathbf{A}\bigcirc\mathbf{O}(\alpha \mathcal{U} \beta), \beta < \phi$ we get:

$$M, \sigma \nvDash (\alpha \wedge \mathbf{A}\bigcirc\mathbf{O}(\alpha \mathcal{U} \beta)) \, \mathcal{U} \beta \ .$$

Hence there exists an integer i such that for all $j \leq i$ we have $M, \sigma_j \nvDash \beta$, and

$$M, \sigma_i \nvDash (\alpha \wedge \mathbf{A}\bigcirc\mathbf{O}(\alpha \mathcal{U} \beta)) \ .$$

Since $M, \sigma \vDash \mathbf{R}(\alpha \mathcal{U} \beta)$, we know $M, \sigma \vDash \alpha \mathcal{U} \beta$, and since $M, \sigma_i \nvDash \beta$ we see that $M, \sigma_i \vDash \alpha$ and hence

$$M, \sigma_i \nvDash \mathbf{A}\bigcirc\mathbf{O}(\alpha \mathcal{U} \beta) \ .$$

The sequence of operators $\mathbf{A}\bigcirc\mathbf{O}$ quantifies over exactly those fullpaths that are failure-free after the first step, or in other words 0-deviations. Thus there exists an i-deviation π from σ such that $M, \pi_{\geq i} \nvDash \bigcirc \alpha \mathcal{U} \beta$ and equivalently $M, \pi_{\geq i+1} \nvDash \alpha \mathcal{U} \beta$. Recall that $M, \sigma_j \nvDash \beta$ for all $j \leq i$; since β is a state-formula and $\sigma_{\leq i} = \pi_{\leq i}$ it follows that $M, \pi_j \nvDash \beta$. Combining this with the fact that $M, \pi_{\geq i+1} \nvDash \alpha \mathcal{U} \beta$ we find that $M, \pi \nvDash \alpha \mathcal{U} \beta$. Since π is a deviation from σ we find that $M, \sigma \nvDash \mathbf{R}(\alpha \mathcal{U} \beta)$, which contradicts our original assumption.

We now consider the reverse part of the case where ϕ is of the form $\mathbf{R}(\alpha \mathcal{U} \beta)$.

($\Longleftarrow$) Say that $M, \sigma \nvDash \mathbf{R}(\alpha \mathcal{U} \beta)$ and $M, \sigma \vDash \tau(\mathbf{R}(\alpha \mathcal{U} \beta))$. Since $M, \sigma \nvDash \mathbf{R}(\alpha \mathcal{U} \beta)$ either $M, \sigma \nvDash \alpha \mathcal{U} \beta$ or there exists a deviation π from σ such that $M, \pi \nvDash \alpha \mathcal{U} \beta$. If $M, \sigma \nvDash \alpha \mathcal{U} \beta$ then clearly $M, \sigma \nvDash (\alpha \wedge \mathbf{A}\bigcirc\mathbf{O}(\alpha \mathcal{U} \beta)) \, \mathcal{U} \beta$, so by contradiction there must exist an i-deviation π from σ such that $M, \pi \nvDash \alpha \mathcal{U} \beta$.

Since $M, \sigma \vDash (\alpha \wedge \mathbf{A}\bigcirc\mathbf{O}(\alpha \mathcal{U} \beta))\, \mathcal{U} \beta$ we see that there exists n such that $M, \sigma_n \vDash \beta$ and for all $m < n$ it is the case that

$$M, \sigma_m \vDash (\alpha \wedge \mathbf{A}\bigcirc\mathbf{O}(\alpha \mathcal{U} \beta)) \ .$$

Say that $n \leq i$. Since $M, \sigma_m \vDash \alpha$ for all $m < n$ and $M, \sigma_n \vDash \beta$, we can see that as $\pi_{\leq i} = \sigma_{\leq i}$ it must also be the case that $M, \pi \vDash (\alpha \mathcal{U} \beta)$, as in Figure 2.

However, recall that we chose the path π such that $M, \pi \nvDash \alpha \mathcal{U} \beta$. By contradiction we know that $n > i$.

Since $n > i$ we know that for all $j \leq i$ it is the case that $M, \sigma_j \vDash \alpha$ and $\sigma_j = \pi_j$. From this and the fact that $M, \pi \nvDash \alpha \mathcal{U} \beta$ it follows that $M, \pi_{\geq i+1} \nvDash \alpha \mathcal{U} \beta$. Since $\pi_{\geq i+1}$ is failure-free we see that $M, \pi_{i+1} \nvDash \mathbf{O}(\alpha \mathcal{U} \beta)$ and

$$M, \pi_i \nvDash \mathbf{A}\bigcirc\mathbf{O}(\alpha \mathcal{U} \beta) \ .$$

Since $\pi_i = \sigma_i$ we also have $M, \sigma_i \nvDash \mathbf{A}\bigcirc\mathbf{O}(\alpha \mathcal{U} \beta)$. However, recall that

$$M, \sigma_m \vDash (\alpha \wedge \mathbf{A}\bigcirc\mathbf{O}(\alpha \mathcal{U} \beta)) \ ,$$

for all $m < n$. By contradiction we see that the smallest counter-example ϕ cannot be of the form $\mathbf{R}(\alpha \mathcal{U} \beta)$. The proof for the case where ϕ is of the form $\mathbf{Я}(\alpha \mathcal{U} \beta)$ is similar.

Case 2. Say that ϕ is of the form $\mathbf{Я}(\alpha \mathcal{U} \beta)$ then recall that

$$\tau(\mathbf{Я}(\alpha \mathcal{U} \beta)) = \tau(\alpha)\, \mathcal{U} \tau(\beta \vee (\alpha \wedge \mathbf{E}\bigcirc\mathbf{P}(\alpha \mathcal{U} \beta))) \ .$$

Say that $M, \sigma \vDash \mathbf{Я}(\alpha \mathcal{U} \beta)$ and $M, \sigma \nvDash \tau(\mathbf{Я}(\alpha \mathcal{U} \beta))$. Thus

$$M, \sigma \nvDash \alpha \mathcal{U} (\beta \vee (\alpha \wedge \mathbf{E}\bigcirc\mathbf{P}(\alpha \mathcal{U} \beta))) \ ,$$

and clearly $M, \sigma \nvDash \alpha \mathcal{U} \beta$. Hence there exists an i-deviation π from σ such that $M, \pi \vDash \alpha \mathcal{U} \beta$, for some $i \in \mathbb{N}$. Since $M, \sigma \nvDash \alpha \mathcal{U} \beta$ we see that $M, \sigma_j \nvDash \beta$ for any $j \leq i$. Thus $M, \pi_{\geq i+1} \vDash \alpha \mathcal{U} \beta$, and since $M, \pi_{\geq i+1}$ is failure-free we know that $M, \pi_{i+1} \vDash \mathbf{P}(\alpha \mathcal{U} \beta)$. From this we know that $M, \sigma_i \vDash \mathbf{E}\bigcirc\mathbf{P}(\alpha \mathcal{U} \beta)$. Since $M, \pi \vDash \alpha \mathcal{U} \beta$ and β is not satisfied before π deviates from σ we know that for each $j \leq i$ we have $M, \sigma_j \vDash \alpha$. Hence $M, \sigma \vDash (\alpha \mathcal{U} (\alpha \wedge \mathbf{E}\bigcirc\mathbf{P}(\alpha \mathcal{U} \beta)))$ and so $M, \sigma \vDash \tau(\mathbf{Я}(\alpha \mathcal{U} \beta))$.

Case 3. Say that ϕ is of the form $\mathbf{P}\bigcirc\alpha$.

($\Longrightarrow$) Say that $M, \sigma \vDash \mathbf{P}\bigcirc\alpha$. Since $M, \sigma \vDash \mathbf{P}\bigcirc\alpha$ we see there exists a failure-free fullpath π such that $\pi_0 = \sigma_0$ and $M, \pi \vDash \bigcirc\alpha$ so $M, \pi_{\geq 1} \vDash \alpha$. Since π is failure-free we see that $M, \pi_{\geq 1} \vDash \neg\mathbf{v}$, and hence $M, \pi_{\geq 1} \vDash \alpha \wedge \neg\mathbf{v}$. As $M, \pi \vDash \bigcirc(\alpha \wedge \neg\mathbf{v})$ and $\pi_0 = \sigma_0$ we see that $M, \pi \vDash \mathbf{E}\bigcirc(\alpha \wedge \neg\mathbf{v})$. This is precisely $\tau(\mathbf{P}\bigcirc\alpha)$.

($\Longleftarrow$) Say that $M, \sigma \vDash \tau(\mathbf{P}\bigcirc\alpha) = \mathbf{E}\bigcirc(\alpha \wedge \neg\mathbf{v})$. Then there exists π such that $\pi_0 = \sigma_0$ and $M, \pi \vDash \bigcirc(\alpha \wedge \neg\mathbf{v})$. Thus $M, \pi_{\geq 1} \vDash (\alpha \wedge \neg\mathbf{v})$. We can construct a failure-free fullpath ρ starting at π_1, We note that as α is a state formula and $M, \pi_{\geq 1} \vDash \alpha$ the fullpath ρ also satisfies α. Since ρ is failure-free and satisfies $\neg\mathbf{v}$, we see that $\sigma_0 \cdot \rho$ is failure-free. Note that $M, \sigma_0 \cdot \rho \vDash \bigcirc\alpha$. It finally follows that $M, \sigma \vDash \mathbf{P}\bigcirc\alpha$.

Case 4. Say that ϕ is of the form $\mathbf{O}(\alpha \mathcal{U} \beta)$.

($\Longrightarrow$) Say that $M, \sigma \vDash \mathbf{O}(\alpha \mathcal{U} \beta)$. If $M, \sigma \vDash \beta$ then, as ϕ is the simplest counter-example and β is simpler, $M, \sigma \vDash \tau(\beta)$ and $M, \sigma \vDash \tau(\mathbf{O}(\alpha \mathcal{U} \beta))$.

If $M, \sigma \nvDash \beta$ then clearly $M, \sigma \vDash \alpha$ and $M, \sigma \vDash \tau(\alpha)$. For contradiction, consider a fullpath π such that $\pi_0 = \sigma_0$ and $M, \pi_{\geq 1} \nvDash \alpha \mathcal{U}(\beta \vee \mathbf{v})$; we see that there exists $i \in \mathbb{N}$ such that $M, \pi_{\geq j} \nvDash \tau(\beta) \vee \mathbf{v}$ for all $j \leq i$ and $M, \pi_{\geq i} \nvDash \tau(\alpha) \vee \tau(\beta) \vee \mathbf{v}$. We see that we can construct a failure-free path ρ such that $\rho_0 = \pi_i$. We see that $\pi_{\leq i-1} \cdot \rho$ is failure-free, and that $M, \pi_{\leq i-1} \cdot \rho \nvDash \alpha \mathcal{U} \beta$. This implies that $M, \sigma \nvDash \mathbf{O}(\alpha \mathcal{U} \beta)$, and so by contradiction we know that $M, \pi_{\geq 1} \vDash \tau(\alpha) \mathcal{U}(\tau(\beta) \vee \mathbf{v})$ for all paths π starting at σ_0. It is then easy to show that $M, \sigma \vDash \mathbf{A} \bigcirc \mathbf{A}(\alpha \mathcal{U}(\beta \vee \mathbf{v}))$ and since $M, \sigma \vDash \alpha$ we see that $M, \sigma \vDash \tau(\mathbf{O}(\alpha \mathcal{U} \beta))$.

($\Longleftarrow$) Say that $M, \sigma \nvDash \mathbf{O}(\alpha \mathcal{U} \beta)$ and $M, \sigma \vDash \tau(\mathbf{O}(\alpha \mathcal{U} \beta))$. We see that if $M, \sigma \vDash \tau(\beta)$ then $M, \sigma \vDash \beta$ and since β is a state formula, $M, \sigma \vDash \mathbf{O}(\alpha \mathcal{U} \beta)$. Say that instead

$$M, \sigma \vDash \tau(\alpha) \wedge \mathbf{A} \bigcirc \mathbf{A}(\tau(\alpha) \mathcal{U}(\tau(\beta) \vee \mathbf{v})) \ .$$

Then we see that for every path π starting at σ_0 we have

$$M, \pi \vDash \tau(\alpha) \wedge \bigcirc(\tau(\alpha) \mathcal{U}(\tau(\beta) \vee \mathbf{v})) \ ,$$

and $\alpha \wedge \bigcirc(\alpha \mathcal{U}(\beta \vee \mathbf{v}))$. Let π be failure-free, then we have $M, \pi \vDash \bigcirc \Box \neg \mathbf{v}$ and so $M, \pi \vDash \alpha \wedge \bigcirc(\alpha \mathcal{U} \beta)$. We see that $M, \pi \vDash \alpha \mathcal{U} \beta$ for all failure-free full paths π starting at σ_0 and so $M, \sigma \vDash \mathbf{O}(\alpha \mathcal{U} \beta)$.

Case 5. The case where ϕ is of the form $\mathbf{P}(\alpha \mathcal{U} \beta)$ is similar to $\mathbf{O}(\alpha \mathcal{U} \beta)$.

Say that ϕ is of the form $\neg \alpha$. By definition $\tau(\neg \alpha) = \neg \tau(\alpha)$. RoCTL$^{\mathrm{S}}$ is a syntactic restriction of RoCTL*, so since τ leaves the $\neg$ unchanged we see that

$$M, \sigma \vDash \neg \alpha \iff M, \sigma \vDash \tau(\neg \alpha) \ .$$

Where ϕ is of the form $\mathbf{E} \bigcirc \alpha$, $\alpha \wedge \beta$, $\alpha \mathcal{U} \beta$, $\mathbf{O} \theta$, $\mathbf{P} \theta$, $\mathbf{R} \theta$, $\mathbf{Я} \theta$, $\mathbf{A} \theta$, or $\mathbf{E} \theta$, we likewise see that, as RoCTL$^{\mathrm{S}}$ is a syntactic restriction, we have $M, \sigma \vDash \phi \iff M, \sigma \vDash \tau(\phi)$. □

7.2. Complexity

We have shown that every statement in RoCTL$^{\mathrm{S}}$ can be expressed in CTL when the CTL logic formulas are allowed to reference the special atom $\mathbf{v}$. The translation into CTL isn't linear. For example, consider

$$\tau(\mathbf{Я}(\alpha \mathcal{U} \beta)) = \tau(\alpha) \mathcal{U} \tau(\beta \vee (\alpha \wedge \mathbf{E} \bigcirc \mathbf{P}(\alpha \mathcal{U} \beta))) \ .$$

See that α and β occur twice on the Right Hand Sided (RHS). Thus the length of the translated formula can double with each $\mathbf{Я}$. However, since α and β are state formulas, it is well known by implementers of decision procedures that we can replace α and β with atoms p_α and p_β. Elsewhere a clause is added requiring that

$$\mathbf{A} \Box(p_\alpha \leftrightarrow \tau(\alpha) \wedge p_\beta \leftrightarrow \tau(\beta)) \ .$$

A similar trick can be used with model checkers, where the atom p^α is added to states where the model checker determines that α holds. We will discuss the details of how this can be achieved for RoCTL$^{\mathrm{S}}$. Since we are translating the RoCTL$^{\mathrm{S}}$ formula into CTL*, we will explicitly state this property for CTL* as a lemma.

Lemma 69. *Say that in some CTL-structure M there exists an atom p_ψ such that for all fullpaths σ we have $M, \sigma \vDash p_\psi$ iff $M, \sigma \vDash \psi$. Then for any CTL* formula ϕ we have $M, \sigma \vDash \phi[\psi/p_\psi] \iff M, \sigma \vDash \phi$.*

It is trivial to prove this lemma inductively from the semantics of CTL*.

Corollary 70. *For any CTL* formula ϕ, and state-formula ψ, it is the case that ϕ is satisfiable iff $\hat{\phi}$ is satisfiable where $\hat{\phi} = \phi[\psi/p_\psi] \wedge \mathbf{A}\Box(p_\psi \leftrightarrow \psi)$.*

Proof. Say that $\hat{\phi}$ is satisfied on some structure M and fullpath π. We can say WLOG that every world in the structure M is accessible from π_0. Since $M, \sigma \vDash \mathbf{A}\Box(p_\psi \leftrightarrow \psi)$, we know that $M, \sigma \vDash p_\psi \iff M, \sigma \vDash \psi$ for all fullpaths σ through M. Thus from the previous lemma and the fact that $M, \pi \vDash \phi[\psi/p_\psi]$ we know that $M, \pi \vDash \phi$. Hence ϕ is satisfiable.

Say that ϕ is satisfied on some structure M and fullpath σ. Since ψ is a state-formula we can add p_ψ to M to produce M' such that p_ψ is true on exactly those states where ψ is true. Hence $M', \sigma \vDash \mathbf{A}\Box(p_\psi \leftrightarrow \psi)$ and from the previous lemma $M', \sigma \vDash \mathbf{A}\Box(p_\psi \leftrightarrow \psi)$. □

Theorem 71. *There exists a polynomial-time computable satisfiability preserving linear translation $\tau_{\mathbf{sat}}$ from RoCTL^S formulas into CTL formulas.*

Proof. Recall that

$$\begin{aligned}\tau(\mathbf{R}(\alpha\mathcal{U}\beta)) &= \tau(\alpha \wedge \mathbf{A}\bigcirc\mathbf{O}(\alpha\mathcal{U}\beta))\,\mathcal{U}\tau(\beta)\\ \tau(\mathbf{Я}(\alpha\mathcal{U}\beta)) &= \tau(\alpha)\,\mathcal{U}\tau(\beta \vee (\alpha \wedge \mathbf{E}\bigcirc\mathbf{P}(\alpha\mathcal{U}\beta)))\ .\end{aligned}$$

We define a translation function τ' similarly to τ with the exception that

$$\begin{aligned}\tau'(\mathbf{R}(\alpha\mathcal{U}\beta)) &= \tau'\left(p_{\tau'(\alpha)} \wedge \mathbf{A}\bigcirc\mathbf{O}\left(p_{\tau'(\alpha)}\mathcal{U}p_{\tau'(\beta)}\right)\right)\mathcal{U}p_{\tau'(\beta)}\\ \tau'(\mathbf{Я}(\alpha\mathcal{U}\beta)) &= \tau'\left(p_{\tau'(\alpha)}\right)\mathcal{U}\tau'\left(p_{\tau'(\beta)} \vee \left(p_{\tau'(\alpha)} \wedge \mathbf{E}\bigcirc\mathbf{P}\left(p_{\tau'(\alpha)}\mathcal{U}p_{\tau'(\beta)}\right)\right)\right)\ .\end{aligned}$$

Below, recall that $\sqsubseteq$ is read as "subformula of". We then define a translation function

$$\tau''(\alpha) = \tau'(\alpha) \wedge \bigwedge_{\beta \text{ s.t. } p_\beta \sqsubseteq (\tau'(\alpha))} \mathbf{A}\Box(p_\beta \leftrightarrow \beta)\ .$$

Recall that we need the clause $\mathbf{A}\Box\mathbf{E}\bigcirc\neg\mathbf{v}$ to ensure that the translated formula is only satisfied on RoCTL$_{\mathbf{v}}$ structure, so we let

$$\tau_{\mathbf{sat}} = \tau''(\alpha) \wedge \mathbf{A}\Box\mathbf{E}\bigcirc\neg\mathbf{v}\ .$$

From the previous corollary we see that $\tau''(\alpha)$ is satisfiable iff $\tau(\alpha)$ (and α) is satisfiable. We will now show that for any State-RoCTL* formula α it is the case that $|\tau''(\alpha)| \leq 45\,|\alpha|$, and hence that the translation $\tau''(\alpha)$ is linear. Other authors often use slightly different set of primitive operators, such as using $\vee$ as a primitive operator instead of $\wedge$. In such variations it may not be the case that $|\tau''(\alpha)| \leq 45\,|\alpha|$. Nevertheless translating between such minor syntactic changes is linear so, the linearity of the translation τ'' is preserved even if the precise factor 45 is not.

For compatibility with the definition of the length of formulas, in this proof we will not include (and) in the count of symbols in a formula. We can enumerate the symbols of α, and modify τ' such that τ' preserves this enumeration. Below we give an example of assigning an integer label i to the operators on the RHS from a label on the LHS. These labels do not have any semantic meaning, they are only added to assist in counting the number of operators used. Note also that we have expanded the $\vee$ operator into two the base operators $\neg$ and $\wedge$. We define $\tau'\left(\mathbf{\Re}^i\left(\alpha \mathcal{U}^j \beta\right)\right)$ to be equal to:

$$\tau'\left(p^i_{\tau'(\alpha)}\right)\mathcal{U}^i\tau\left(\neg^i\left(\neg^i p^i_{\tau'(\beta)} \wedge^i \neg^i\left(p^i_{\tau'(\alpha)} \wedge^i \mathbf{E}^i \bigcirc^i \mathbf{P}^i\left(p^i_{\tau'(\alpha)}\mathcal{U}^i p^i_{\tau'(\beta)}\right)\right)\right)\right) .$$

Likewise for the other lines defining τ' we define τ' such that the new operators introduced on the RHS have the same label as the operator of the highest precedence on the LHS. We see that in the line above, there are 15 symbols labelled with i for the one operator $\mathbf{\Re}$ labelled with i on the LHS. It is easy to see from induction that i occurs at most 15 times in $\tau''(\alpha)$ for each time it occurs in α. Note that τ'' contains $\leftrightarrow$ operators, which we define as abbreviations rather than as primitive operators. Since these are not nested, expanding each occurrence of $\cdots \wedge \mathbf{A}\Box(p_\beta \leftrightarrow \beta)$ into

$$\cdots \wedge \mathbf{A}\Box(\neg(\neg(p_\beta \wedge \beta) \wedge \neg(\neg p_\beta \wedge \neg\beta))) ,$$

at most doubles the number of i-labelled symbols. Hence for each i-labelled symbol in α there are at most 30 i-labelled symbols in $\tau''(\alpha)$. We see in the equation fragment above (excluding the portion represented as $\cdots$) has 15 unlabelled symbols. Thus for each symbol in α there are at most 45 symbols in $\tau''(\alpha)$. Thus $|\tau''(\alpha)| \leq 45\,|\alpha|$. □

We see that the translation function $\tau_{\mathbf{sat}}$ is itself computationally simple. Since the decision problem for CTL is in EXPTIME Emerson and Halpern [14], we get the following theorem.

Theorem 72. *The satisfiability decision problem for RoCTL^S is in EXPTIME.*

We will now show that, as with CTL Clarke et al. [15], we can model check RoCTL$^{\mathrm{S}}$ in $\mathcal{O}((|S| + |R|) \cdot |\psi|)$ time.

Theorem 73. *Given a RoCTL^S structure $M = (S, R, g)$, and formula ψ the RoCTL^S model checking problem can be decided in time of order $\mathcal{O}((|S| + |R|) \cdot |\psi|)$.*

Proof. It is easy to see that model-checking procedure is polynomial; for each atom $p_\beta \sqsubseteq \tau''(\psi)$ and world $w \in S$ we can use the CTL model checking procedure to model check β at the world w and add p_β recursively to the model if $M, w \vDash \beta$. We then see that

$$M, w \vDash \tau'(\psi) \iff M, w \vDash \tau'(\psi) \iff M, w \vDash \psi \ .$$

Hence model checking the RoCTL$^{\mathrm{S}}$ formula ψ will result in at most $|S| \cdot |\psi|$ calls to the $\mathcal{O}((|S| + |R|) \cdot |\psi|)$ model checking procedure. We will now refine this procedure.

The model checking procedure for CTL in Clarke et al. [15] marks each state with the subformulas of β that hold at each state when checking whether $M, w \vDash \beta$ holds for some formula β, structure M and world w. This is done so that each formula of length n can be easily model checked once each state has been marked with the subformulas of length $n-1$. After performing the CTL model checking procedure of Clarke et al. [15] we can inspect this marking to determine at which worlds of M the formula β holds at, and we do not need to perform the procedure for each world in S. As such we need to call the CTL model-checking procedure of Clarke et al. [15] at most $|\psi|$ times. This refinement has removed a factor of $|S|$. We will now show that algorithm achieves a complexity of $\mathcal{O}((|S| + |R|) \cdot |\psi|)$ despite needing to call the CTL model-checker multiple times. The basis of this proof is the fact that we only need to model check short formulas, which are in total shorter than $\tau''(\psi)$.

Let Φ be the set of formulas that we send to the CTL model-checker, that is

$$\Phi = \{\tau'(\psi)\} \cup \{\beta : p_\beta \sqsubseteq \tau'(\psi)\} \ .$$

Hence the time required is

$$\mathcal{O}\left(\sum_{\phi \in \Phi} ((|S| + |R|) \cdot |\phi|)\right) ,$$

which simplifies to

$$\mathcal{O}\left((|S| + |R|) \cdot \sum_{\phi \in \Phi} |\phi|\right) \ .$$

Note that each member of Φ is a subformula of $\tau''(\psi)$. Since each member of Φ comes from a separate part of $\tau''(\psi)$ we see that $\sum_{\phi \in \Phi} |\phi| \leq |\tau''(\psi)| \leq 45\,|\psi|$. As such the complexity is $\mathcal{O}((|S| + |R|) \cdot |\psi|)$. □

It is well known that model-checking CTL formulas is P-hard Schnoebelen [29]. As RoCTL$^{\mathrm{S}}$ is an extension of CTL it is clear that it is also P-hard. Together with the previous theorem, this gives us the following corollary.

Corollary 74. *Model checking RoCTL^S formulas is P-complete.*

8. Conclusion

I think you need a related work section before this. I think you should at least compare your work with other logics for robustness, perhaps other tableau for branching-time logics. There might be other related work you could mention also.

We discussed three sub-logics of RoCTL*: RoBCTL*, Pair-RoCTL and State-RoCTL. We presented a tableau based decision procedure for RoBCTL*, which allows us to reason about robustness in systems that need not be limit closed. The complexity of this decision procedure may be non-elementary; however, we discussed why this may not be a problem for most applications. For example, these examples had the property that they never had $\mathcal{U}$, $\mathbf{R}$, $\mathcal{U}$ nested in order, which is sufficient to ensure that the decision procedure for RoBCTL* terminates in an elementary amount of time. In M^{c}Cabe-Dansted et al. [23] it was shown that RoCTL* could be decided via reductions into QCTL* and tree automata, and further that no such reductions could provide an elementary decision procedure for RoCTL*. This tableau was first presented in M^{c}Cabe-Dansted [19], and no optimisation has been found to provide an elementary decision procedure for RoBCTL* or RoCTL*. This suggests the possibility that no elementary decision procedure for RoCTL* will be found. For this reason we also investigated syntactic restrictions of RoCTL*.

We have shown that although the syntax of Pair-RoCTL is an intuitive definition of a CTL-like restriction of RoCTL*, the properties of Pair-RoCTL are closer to CTL*. We have shown that every property that can be expressed in CTL* can be expressed in Pair-RoCTL, with a minor translation on the structures. Combining this with the result M^{c}Cabe-Dansted et al. [23] that every property that can be expressed in RoCTL* can be expressed in CTL* when CTL* is allowed to access the special violation atom indicates that Pair-RoCTL, CTL* and RoCTL* all have similar expressivity. Additionally, we have shown that the decision problems of Pair-RoCTL are in complexity classes that are at least as hard as those of CTL*, for example, satisfiability checking Pair-RoCTL is 2-EXPTIME hard. Determining whether the decision problems for Pair-RoCTL are as hard as those for the full RoCTL* remains an open problem. Nevertheless it is clear that State-RoCTL is a better choice where tractable decision problems are required.

State-RoCTL provides a highly decidable logic. State-RoCTL* has an efficient satisfiability-preserving translation into CTL. This allows existing implementations of CTL decision procedures to be used to decide State-RoCTL. Even existing model checking procedures for CTL can be used on the linear translation with only trivial modifications. Although having CTL-like complexity, State-RoCTL can naturally express non-trivial RoCTL* formulas, such as formulas that have Prone nested directly within Robustly.

9. Acknowledgements

This Project is supported by the Australian Government's International Science Linkages program and the Australian Research Council.

References

[1] T. French, J. C. M^{c}Cabe-Dansted, M. Reynolds, Axioms for Obligation and Robustness with Temporal Logic, J. Appl. Log. Accepted to appear in Deontic Logic Corner of JLC.

[2] A. Pnueli, The Temporal Logic of Programs, in: Proceedings of the FOCS, IEEE, 46–57, 1977.

[3] E. A. Emerson, E. M. Clarke, Using branching time temporal logic to synthesize synchronization skeletons, Sci. Comput. Programming 2 (3) (1982) 241–266, doi:10.1016/0167-6423(83)90017-5.

[4] E. A. Emerson, J. Y. Halpern, "Sometimes" and "not never" revisited: on branching versus linear time temporal logic, J. ACM 33 (1) (1986) 151–178.

[5] J. C. M^{c}Cabe-Dansted, A Temporal Logic of Robustness, Ph.D. thesis, The University of Western Australia, 2011.

[6] J. C. M^{c}Cabe-Dansted, T. French, M. Reynolds, S. Pinchinat, On the Expressivity of RoCTL*, in: C. Lutz, J.-F. Raskin (Eds.), Proceedings of the 16th International Symposium on Temporal Representation and Reasoning (TIME), TIME '09, IEEE Computer Society, ISBN 978-0-7695-3727-6, 37–44, see also journal paper [23], 2009.

[7] O. Kupferman, Augmenting branching temporal logics with existential quantification over atomic propositions, in: Proceedings of the CAV, Springer-Verlag, Liege, 325–338, 1995.

[8] M. Reynolds, A Tableau for Bundled CTL*, J. Log. & Comput. 17 (1) (2007) 117–132.

[9] C. Stirling, Modal and temporal logics, Oxford University Press, Inc., New York, NY, USA, ISBN 0-19-853761-1, 477–563, 1993.

[10] M. Reynolds, A tableau-based decision procedure for CTL*, J. Form. Asp. Comput. 23 (2011) 1–41.

[11] O. Friedmann, M. Latte, M. Lange, A Decision Procedure for CTL* Based on Tableaux and Automata, in: J. Giesl, R. Hähnle (Eds.), 5th International Joint Conference on Automated Reasoning (IJCAR), vol. 6173 of *LNCS*, Springer-Verlag, 331–345, 2010.

[12] J. C. M^{c}Cabe-Dansted, M. Reynolds, Fairness with EXPTIME Bundled CTL Tableau, to be presented at TIME 2014, see also expanded version `http://www.csse.uwa.edu.au/~john/papers/ExpFair.pdf`, 2014.

[13] M. Y. Vardi, L. J. Stockmeyer, Improved upper and lower bounds for modal logics of programs, in: Proceedings of the 17th annual ACM symposium on Theory of computing (STOC), STOC '85, ACM, New York, NY, USA, ISBN 0-89791-151-2, 240–251, 1985.

[14] E. A. Emerson, J. Y. Halpern, Decision Procedures and Expressiveness in the Temporal Logic of Branching Time, in: Proceedings of the STOC, ACM, 169–180, 1982.

[15] E. M. Clarke, E. A. Emerson, A. P. Sistla, Automatic verification of finite-state concurrent systems using temporal logic specifications, ACM Trans. Program. Lang. Syst. 8 (2) (1986) 244–263, ISSN 0164-0925.

[16] L. Zhang, U. Hustadt, C. Dixon, A resolution calculus for the branching-time temporal logic CTL, ACM Trans. Comput. Log. 15 (1) (2014) 10.

[17] P. Abate, R. Goré, The Tableau Workbench, Electron. Notes Theor. Comput. Sci. 231 (2009) 55–67, ISSN 1571-0661.

[18] R. Goré, J. Thomson, F. Widmann, An Experimental Comparison of Theorem Provers for CTL, in: C. Combi, M. Leucker, F. Wolter (Eds.), Proceedings of the TIME, IEEE, ISBN 978-1-4577-1242-5, 49–56, 2011.

[19] J. C. M^{c}Cabe-Dansted, A Tableau for RoBCTL*, in: S. Hölldobler, C. Lutz, H. Wansing (Eds.), Proceedings of the 11th European Conference on Logics in Artificial Intelligence (JELIA), vol. 5293 of *JELIA '08*, Springer-Verlag, ISBN 978-3-540-87802-5, 298–310, see also expanded version: `http://www.csse.uwa.edu.au/~john/papers/Da08Tableau.pdf`, 2008.

[20] J. C. M^{c}Cabe-Dansted, C. Dixon, CTL-Like Fragments of a Temporal Logic of Robustness, in: N. Markey, J. Wijsen (Eds.), Proceedings of the 17th International Symposium on Temporal Representation and Reasoning (TIME), TIME '10, IEEE, ISSN 1530-1311, 11–18, 2010.

[21] T. French, J. C. M^{c}Cabe-Dansted, M. Reynolds, A Temporal Logic of Robustness, in: B. Konev, F. Wolter (Eds.), Proceedings of the 6th International Symposium on Frontiers of Combining Systems (FroCoS), vol. 4720 of *FroCoS '07*, Springer-Verlag, ISBN 978-3-540-74620-1, 193–205, 2007.

[22] G. H. V. Wright, Deontic logic, Mind (1951) 1–15.

[23] J. C. M^{c}Cabe-Dansted, T. French, M. Reynolds, S. Pinchinat, Specifying Robustness, arXiv:1309.4416, Under Review.

[24] E. A. Emerson, A. P. Sistla, Deciding Branching Time Logic: A Triple Exponential Decision Procedure for CTL*, in: E. M. Clarke, D. Kozen (Eds.), Proceedings of the Logic of Programs, vol. 164 of *LNCS*, Springer-Verlag, ISBN 3-540-12896-4, 176–192, 1983.

[25] E. A. Emerson, A. P. Sistla, Deciding branching time logic, in: Proceedings of the 16th annual ACM symposium on Theory on computing (STOC), STOC '84, ACM Press, New York, NY, USA, ISBN 0-89791-133-4, 14–24, 1984.

[26] E. A. Emerson, C. S. Jutla, The Complexity of Tree Automata and Logics of Programs, SIAM J. Comput. 29 (1) (2000) 132–158, ISSN 0097-5397.

[27] E. A. Emerson, C.-L. Lei, Modalities for model checking (extended abstract): branching time strikes back, in: Proceedings of the 12^{th} ACM SIGACT-SIGPLAN symposium on Principles of programming languages, POPL '85, ACM, New York, NY, USA, ISBN 0-89791-147-4, 84–96, 1985.

[28] A. P. Sistla, E. M. Clarke, The complexity of propositional linear temporal logics, J. ACM 32 (3) (1985) 733–749, ISSN 0004-5411.

[29] P. Schnoebelen, The complexity of temporal logic model checking, Advances Modal Log. 4 (2003) 437–459.