
Summary of Paper

The logic RoCTL* (Robust CTL*) is a logic for reasoning about robustness in the presence of errors. For example, it can express that "for every error-free path, it is always the case that, even if one error is introduced, and if the path diverges, the divergent path still satisfies \phi". The syntax of RoCTL* is like CTL* except it has additional path quantifiers, specifically the Robustly quantifier R. Moreover, the semantics of RoCTL* arise from unfolding Kripke structures in which some transitions can be successful and others can have errors. The targets of unsuccessful transitions are marked by a special atom. It has been argued in earlier work that RoCTL* can naturally and succinctly reason about systems and situations/protocols in which failures may occur and yet the (original) specification is still satisfied.

It is known that satisfiability and model checking of RoCTL* are decidable, and that RoCTL* is expressively equivalent to CTL*, but non-elementarily more succinct than CTL*. The main open question (not solved in this paper) is whether RoCTL* has an elementary decision procedure. This motivates the study of fragments and logics related to RoCTL*. Indeed, one of the stated motivations is to find fragments of RoCTL* that one can prove have elementary (or efficient) decision procedures. To this end, the authors identify a CTL-like fragment called State-RoCTL* that has the same complexity (of satisfiability and model checking) as CTL. Another related logic is the bundled-variant RoBCTL*. This logic has the same syntax as RoCTL* but the semantics are different: paths are restricted to a given set of paths (that are suffix and fusion closed). The authors prove that satisfiability is decidable using an extension Reynolds' tableaux procedure for CTL*. They also identify syntactic fragments of RoBCTL* for which satisfiability is elementary. Finally, another CTL-like fragment Pair-RoCTL* is considered, which turns out to be much more like CTL* than CTL.

Evaluation

The paper is within the scope of I&C and readers of this journal will find it relevant.

The results seem novel, and the main results were presented in conferences (TIME and JELIA) and one of the authors' PhD Thesis.

The paper seems to have been hastily put together and requires much work to bring it into publishable form. In particular (see comments to

the authors, below)
- some notations can be improved to help the reader quickly grasp the nature of the objects,
- some important definitions are missing,
- the main objects, including RoCTL*, should be better motivated,
- the contributions are not spelled out clearly enough,
- some sections/paragraphs appear out of the blue which makes them hard to follow,
- some of the proofs require cleaning and more care, and all of the main proofs require carefully worded proof ideas to make them more digestible.

Due to the poor organisation of the proofs, I was not able to verify all of them line by line.
Having said that, the proof techniques all have to deal with the operator R which has the unusual property that R\phi is a path-formula, while R is a path quantifier. This leads, e.g., to an interesting variation

of the tableaux proof for CTL* of [8].

I suggest acceptance, subject to major revisions.

General comments to the authors

-- It would help the reader if the examples (of formulas, as well as the intended scenarios where the logics might be applied) were more fleshed out.

In particular, you should clearly explain (or give examples to illustrate) what you mean by "error" and "failure"? Are these the same notions? I suppose so, although "failure" suggests that system crashes (rather than, e.g., some sensor giving the incorrect reading, or some process failing to make an action in time).

Also, it seems you intend that R quantifies over paths which are deviations of the original path, even if no error was introduced, and which are error free from the point of deviation. It is not clear why this is a natural definition. This definition should be justified (by examples, by clearer explanations, or by citations).

Also, I think there should be a good discussion about what you mean by "Robust System". E.g., you say on page 5 that a robust systems often has an implicit fairness constraint, and then give an example. The paper would be greatly enhanced if this was part of a deeper discussion about the types of systems to which these logics are targeted.

-- Please discuss the relationship between your work and related works more fully.
E.g.,
- You suggest that in previous work on this logic

[1,5,etc], it is shown that RoCTL* is expressively equivalent to CTL* (and in fact a conservative extension of CTL*). It is worth pointing out what this means exactly. In particular, these logics are expressively equivalent over which set of models?
- The paper Faella, Napoli, Parente, 2010 "Graded ATL" deals with counting paths in a reactive setting, and the paper Aminof, Murano, Rubin, 2014: " Satisfiability and Model Checking of CTL* with Graded Path Modalities" deals with path quantifiers that can express certain forms of robustness that can not be expressed in CTL* (see also the references in this paper).
- You provide a tableau proof following Reynolds. Since the automata-theoretic approach to temporal-logics is well-established, you might discuss why you chose a tableau proof. Are there difficulties giving an automata-theoretic proof?

-- Some notations should be improved. For instance, delta^0, \delta^+, \delta^\omega also depend on B, and this should be made explicit (at least in the definitions). Other examples of poor notation are RoCTL*^S and State-RoCTL* --- please choose one.

-- There are many statements in the paper of the form "logic L is expressively equivalent to logic Q" or "logic L is an extension of logic Q". The problem is that the logics L and Q may be over different classes of models. E.g., CTL is over kripke-structures while RoCTL^S is over ROCTL^S structures. Thus you should clearly explain what you mean by "expressively equivalent" and "an extension of".

-- It seems that RoBCTL* is a semantic sublogic of RoCTL*. This should be stressed. What is the relationship between these logics wrt satisfiability? are they expressively equivalent?

Also, you should discuss the model-checking problem for RoBCTL* or explain why it was omitted. Does it make sense to talk about finite (or finitely-presentable) RoBCTL*-structures?

-- Please provide a table summarising the known results and your contributions. This makes it much easier for readers to grasp your contributions. An alternative is to add a bold subheading at the end of the introduction called "Our contributions" (i.e., before par 4 on page 4) and to make that paragraph much more precise (e.g., replace "at least as hard to reason about" by what you actually prove). Also, the relationship between the present work and the logics with v subscripted (e.g., RoCTL*_v) should be made more explicit, and mentioned in this summary.

-- The motivation for bundled logic (pages 4 and 5) could be better structured. E.g., you suggest that one reason for studying them is that it is easier to find Tableau proofs. Ok, but what are the advantages of Tableau proofs over, e.g., automata-theoretic proofs? Also, the last three paragraphs of that section (i.e., from "In section 5" to "will hold.") do not discuss the motivation for bundled logics, but rather they discuss how the tableau construction of the present paper differs from that in Reynolds' paper. This discussion should be labeled as such, and perhaps moved to a more appropriate place (perhaps in the same section as the Tableau proof). Also, since tableau's are an important part of this paper, it would be nice to have a gentler introduction to the proof, i.e., please give the general idea of the tableau (what is its shape, how is it labelled, when is it successful, etc.), and also, please give an example of a satisfiable formula and its tableau. Also, please give some intuitions/explanations/justifications for the definition of a bundle, i.e., why fusion- and suffix-closed are assumed.

Detailed comments to the authors.

Please stress in the preliminaries that indexing of paths starts at 0, i.e., w_0, w_1, \dots (instead of w_1, w_2, \dots).
Page 2:

- You write \box for always and then F for sometimes. Is there a good reason to mix these notations?

- It would be useful to the reader to say that more examples appear in Section 4.

- I would suggest adding an example scenario. In particular, what is "failure" intended to mean? (see comment about page 6 below).

- par 4: you state that full RoCTL* is decidable but, a) you have not described what full means, b) please be explicit if you mean that satisfiability is decidable or model-checking is decidable. Moreover, it is helpful to the reader to say a bit more about these translations, e.g., what do they preserve (truth? satisfiability?).

- par -1: the phrase "limit closure is (not) valid in a logic" is hard to understand. Can you rephrase it? What is "limit closure" a property of? (the set of runs that are considered by path quantifiers?)

Also, it would be helpful to state here that RoBCTL* is a semantic sublogic.

Also, please formally define "limit closure".

Page 3: "need to be allowed" --> "are allowed".
"subset of those of RoCTL*" ---> "subset of those allowed in RoCTL*"
"expressive fairness" --> "express fairness"

The sentence "The obvious way ..." must be rewritten. What is this way? What is the contradiction?

Also, you seem to imply that bundled logics can easily handle probabilistic reasoning. If so, this is a point worth stressing!

Definition 6:
-- You write "w in B" (twice). At first I thought you meant that $w \in B$ (which would have been a typo). Now I see that you mean that the full path is in B. Please rewrite to make this clear.

On pages 2 and 3 you refer to logics that are "amenable to automated reasoning" and "a little closer to being usable". Please be explicit: will the reader see decision procedures with (relatively) low complexity? Or will the reader see other arguments hat these logics are usable (e.g., decision procedures that are easy to implement).

Page 3:
par -2: You state that CTL is frequently used. Please be explicit. Used in tools? in papers about verification?

Satisfiability of CTL is EXPTIME-COMPLETE (not just EXPTIME).

You write "model checking CTL* is singly exponential in the length of the formula". What about the size of the Kripke structure?

par -1:why do you not call Pair-RoCTL simply RoCTL?

Page 4: par 1: The phrase "CTL model checker of [15]" sounds like it refers to a tool that does model checking, rather than an algorithm.

par 2: "not expressively equivalent" --> "not truth preserving" (?)

par 3: "With current technology State-RoCTL is tractable for larger problems than CTL*" does not make sense to me. Please rewrite/rephrase/be more explicit. What does "larger problems" mean? Larger in size? Larger in scope? Larger in quantity?

par -2: It would not hurt to remind the reader what "bundled variant" means, and what "limit closure" refers to.

Page 6:
The sentence "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." needs fleshing out, since it is confusing as it is written. In what sense is it forbidden to enter a state labeled v? What forbids this? Also,

what is "failure" supposed to mean here? Does it mean that the system crashes? or that some sensor gives the incorrect reading but the system still runs?

Page 7:
-- line -2: please say how sigma is quantified.

-- Definition 7: please stress that if pi is a deviation of sigma then pi has only finitely many failures.

Page 7 and 8: You write that you define RoCTL* and, in parenthesis that you define RoBCTL*, but then give just a single definition. What then is the definition of RoBCTL*? I assume you mean that the syntax is the same but the semantic differs. If this is the case, please say so here.

Page 8: Please rephrase the definition/description of RoBCTL*_v. I understand it to mean that this logic does allow formulas to explicitly contain v.

Also, what is the relationship between the logics in [21] and RoBCTL*_v? Also, how is v useful/used if the logic can't talk about it?

[Style] It seems you should decouple this comment about "v" from the last sentence, i.e., "The \neg, \wedge, ... from CTL" otherwise one might think (as I did) that this last sentence is specific to RoBCTL*_v.

Page 8:
-- Definition of Syntax. Why is (\phi \wedge \psi) not a state formula?
-- The style of the semantics are a little unusual. Is there a good reason not to mimic the style of the usual definition of CTL*, i.e., define state formulas \phi and path formulas \psi, and the define M,w \models \phi and define M,\pi \models \psi?

-- Please stress that, unlike O\phi and A\phi, the formula R\phi is a path formula (not a state

formula).
The difficulty is one of notation: since \A and \O and \R all use the same font, it is natural for the reader to assume that they are the same types of objects, i.e., path quantifiers \X such that \X \phi is a state formula.

Definition 9: I think you meant to write \phi instead of \tau(\phi) in the second line of the dfn.

Page 10: Lemma 13 has a typo. I guess one of the RoBCTL*s should be RoCTL*.

-- Definition 14: This definition could be tightened a bit, namely,a) "the representation of", b) you give three sentences describing three overloadings of the notation |x| (please make this clearer).

-- if it is not possible to define State-RoCTL* before using it, at least give an informal description here.

-- what does "RoCTL* model" refer to? a kripke structure?

-- Example 15, item 2: "was empty" --> "is empty". In the description you imply that a failure refers to forgetting to fill the bowl. In the formulas there is no mention of this. What do errors refer to then?
	What can be said about the set of formulas 1 through 5?

- In both the examples I feel that it would be clearer if the logics explicitly expressed something about v. Otherwise, you should give a model M of the formulas that has v on some of its states.

Page 11, dfn 17: why are the subscripts used here?

dfn 18: why is the enumeration needed here? you can define \xi without it.

dfn 19: why do you use the notation O for next as well as in O^-1 and O^*? What is the intuitive meaning of O^-1 for each of the more complicated formulas in the definition (i.e. Until and \R). What is the relationship between O^-1 and the past operator "Yesterday"?

Pages 12,13: You use * in two seemingly different ways. One in S*T and the other in O*. This would not be a problem except that the paragraph after dfn 23 seems to mix them up.

Page 13: dfn 22: please explain why you use O here.

Page 14: It would be easier to follow if Lemma 28 were placed closer to definition 23.
Also, what does (\dots) inside the proof refer to?

Page 16:

line 1: rephrase. perhaps, replace the first occurrence of "describes" by "relates" and remove "might".

I don't understand the relevance/point of the par after dfn 31.

Section 5.1. You talk about the tableau, but it has not yet been formally defined.

Page 17: You refer to a tableau as being "accepted", but this has not yet been formally defined.
In dfn 33 you use the word "succeeds", although i gather this means the same thing as "accepted".

Page 18: Lemma 35. Extra space before bold C in proof.

Page 19: Please formalise the hypothesis of Cor 40.

Page 20: That there are any "Real world uses" requires some justification.

Section 5.3: RoBCTL*-TAB is not defined. Also, please state the first sentence as a proposition or lemma, or simply say that the rest of the subsection will prove this fact.

Please give an intuition for the definition of thread, esp. item 2.

Page 21, lemma 44. missing period.

Page 23, lemma 47: you have not defined what it means for a tableau to halt.
Also, why is there a v in RoBCTL_v here?

Page 25:
Section 6.1: Why do you introduce yet another notation for pair-RoCTL?

Remove "like" from "like in the sense".

par -1: The paragraph starting "We note that the translation..." needs rewriting. First, what sort of translation? (cf dfn 9).

Second, I couldn't get the point of the paragraph. A translation from LTL to RoCTL^P would not be much simpler than one from CTL*? I suppose it depends on what type of translation.

Page 26:
[[Tim suggests...

What does it mean for a "subset of states to form a CTL model"?

Page 34: section 7.1. It seems you mean to join pars 1 and 2.

par 2: You write that "\beta does not occur on \pi_j for j \leq i". However I do not understand the reasoning.

Page 35:
Definition 66: Please remove the ambiguity formed by having "(or \reverseR)" in parenthesis. In particular, is \phi < \psi if phi and psi have the same number of \reverseR and |phi| < |\psi| but \psi has less R operators than phi?

Page 36, lemma 68: The sentence starting "Hence there exists an integer i..." seems to be missing a case, i.e., \sigma_j \not \models \beta for all j.

Also a little later you say that if \sigma \models \alpha \until \beta and if \sigma_i \not \models

\beta then \simga_i \models \alpha. However this reasoning is not correct unless i is the smallest integer k such that \sigma_k \not \models \beta.

Page 39: Lemma 69: you forgot to say that \psi is a state-formula.

Page 42:
"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."... I agree ;);)

Again, limit-closed is used and not defined earlier.

"via reductions into" --> "by reducing to"

"This tableau"... Which tableau?

"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." Please rewrite more clearly/be more explicit. What type of translation is exhibited?

What does it mean for CTL* to be "allowed to access the special violation atom"?

Par -2 talks about Pair-RoCTL. But then at the last sentence it suddenly talks about State-RoCTL. This is a bit jarring. Perhaps rewrite to make the connections clearer.

Par -1: You call the translation "efficient" and then "linear". Is this on purpose?

What "trivial modifications"?