Review 1

The paper aims at presenting an algorithm that realizes forgetting for CTL formulas, that is, removing propositional atoms from CTL formulas while preserving satisfaction up to removed atoms; satisfaction up to removed atoms is formalized by a notion of V-bisimulation that ignores atoms from V. To achieve this, the authors devise an intricate procedure that builds upon the resolution method for CTL from [Zhang, Hustadt and Dixon, 2014]. The procedure first transforms the input formula to a normal form (outside of standard CTL syntax) that allows for resolution. The set of clauses that is obtained from the resolution process then is transformed, removing clauses that mention variables from V; finally, the resulting set of clauses is transformed back to a CTL formula that no longer uses variables from V.

While the principle idea seems to be interesting and worth pursuing, and the presented method might be appropriate to solve the problem, the paper has several grave problems:

-- The paper is riddled with mistakes, misspellings and imprecise formulations, in some places up to the point of incomprehensibility. E.g. "satisfaction" and "satisfiability" are mixed up in the standard definition of CTL semantics and the very central notion F_CTL(phi,V) is left undefined. For a small selection of other examples, see the detailed comments below.

-- Some proofs in the paper are not convincing and raise doubts in the correctness of the results.

a) some of the results (in particular those about V-bisimulations) are imported from unpublished work of the authors. Though these results are not very deep or technically involved, this seems to be bad practice.

b) many of the proofs that are contained in the paper are too sketchy and short to be convincing, and lack intuition. More explanation on the various constructions should be provided. E.g. in the proof of Proposition 6, it appears crucial to me, to reason why all clauses that contain atoms from V can be removed from the set of clauses at this point. The proof sketch however just states that a \Gamma-bisimilar model can be constructed by "adding or deleting some atoms". The longer proof in the appendix does not provide more intuition on this point either.

Since the paper fails to argue in a comprehensible manner why the procedure is correct, and provides next to no intuition why the various intricate constructions are necessary and lead to correct results, I am not convinced that the compound method is correct.

On the positive side, the paper provides two example formulas for which the procedure seems to yield correct final results; however, the examples on their own do not sufficiently clarify the ideas behind the construction.

-- The paper is not self-contained. It is true that there are space restrictions, but the authors are modifying (and heavily relying on) a resolution method for CTL from another paper without introducing or explaining it at all; for instance, Example 6 consists of a list of clauses obtained by named resolution rules from [Zhang, Hustadt and Dixon, 2014] that are not shown or explained in the current paper. This makes it very difficult to understand the example, and, more gravely, the central ideas of the paper, without having the detailed definitions of the cited paper in mind.

-- The practical use of the introduced method and its complexity theoretic consequences remain hazy. According to the introduction of the paper, the objective of forgetting is to keep formulas more manageable, but the authors do not give an upper bound on the size of the formula that is constructed by their procedure; since it is obtained as a conjunction of clauses, it seems possible that the blow-up could be exponential.

Due to the large amount of mistakes and the unclear presentation, I come to the conclusion that the paper in its current form does not satisfy the high standards of this conference and hence recommend not to accept it for publication.

Detailed comments:

- The paper contains many misspellings, orthographic mistakes and inaccuracies, sometimes up to 20 mistakes on a single page (I mention just a few below). It would probably help to use a spell-checker.

- Two different symbols are used to denote the empty set.

- Definition syntax CTL, p. 2: having brackets as explicit operators (without semantics) is unusual.

- Definition semantics CTL, p. 2: as mentioned above, the defined relation is the satisfaction relation, not the satisfiability relation.

- Definition semantics CTL, last paragraph, p. 2: the sentence "The V-irrelevant of a set of formulas can be defined similarly." is incomprehensible.

- Section 2.3, definition of semantics of SNF_CTL^g, p. 3: The semantics of e.g. the operator AF is defined "analogous to that of CTL given previously", but it was defined by negation previously, that is, as "not EG not". It is somewhat unclear if the dual of EG from standard CTL or the dual of EG_ind is meant to be used here.

- Definition 2., p. 4: should the ampersand be a dot?

- Definition 3., p. 4 (et passim): a mixture of mathematical quantifiers and textual quantifiers is used ("\forall j\notin I there is..").

- Section 4., top of p. 5: F_CTL(phi,V) is used here but never defined. I assume it is supposed to be "the" result of forgetting V from phi. However, as mentioned below Definition 2., this is only defined up to some equivalence. Hence it should be clarified, what F_CTL(phi,V) is and what precisely the algorithm is meant to compute. Leaving out this central definition makes it impossible to verify the proof of the main theorem (Theorem 2) in detail.

- Section 4.1., p. 5: The transformation of the input formula to normal form uses a function "simp" which is not explained (and apparently not used later on).

- Algorithm 1, p. 5: Why are the sets T_\phi and V' the only ones that are initialized? Also, T_\phi is initialized once again in Algorithm 2.

- Proof sketch for Proposition 3, p. 5: The first line of the second paragraph is not a sentence. The sketch also considers only the base case of the induction, which does not provide intuition to the reader. It would help to sketch at least one inductive step here, that is, the application of some rule.

- Example 2, p. 5: I assume it should be V'={x,y,z,w} instead of V'={x,y,x,w}. The role of the additional atom w (which also occurs in Example 7) remains unexplained.

- Section 4.2., p. 5: give at least some intuition on the set Res (e.g.: what kind of clauses are contained in this set and why are they the relevant clauses?)

- Algorithm 3, p. 6: the notions of positively and negatively appearing atoms are used before being defined (negatively appearing atoms are undefined altogether).

- Example 3, p. 6: line (7) seems to be obtained by SRES6 from 5. and 8., not from 5. and 7. Similarly, line (8) seems to be obtained by SRES6 from 6. and 8., not from 5. and 8.

- Definition 4, item (i), p. 6: I assume "one and on one atom" should be "one and only one atom".

- Definition 4, item (iv), p. 6: it is assumed that i\in{1,..,n}, but i is not used afterwards. Also I think it should be
"..there are \alpha\supset p_1, ..,\alpha\supset p_1 \in \Gamma .."
instead of
".. there is \alpha\supset p_1, ..,\alpha\supset p_1 \in \Gamma ..".
Furthermore, there is a capitalization mistake.

- Example 4, p. 6: give reasons why y can be instantiated to q\wedge AX(q\vee f \vee m). Also, this is not a sentence: "And z can be instantiated by r."

- Proof of Proposition 5, p. 7: It is unclear what the first sentence is supposed to mean.

- Proposition 6, p. 7: "stretch" -> "sketch".

- Example 5, p. 7: Why is the third clause on the left (the one that mentions p) not removed at this point?

- Proposition 8, p. 8, the meaning of the name "NI-BRemain" of the proposition is unclear.

- Definition of T_CTL, p. 8: C' is not quantified.

- Proposition 10, p. 9: For the bound to be meaningful, it is necessary to provide some bound on the number n' of variables introduced in the transformation. Also there is a bracket missing in O((m+1)2^{4(n+n')}, and since O-notation is used, it is not necessary to mention the constant factor +1. The proof of Proposition 10 in the appendix refers to lines 19-31 of some algorithm in the paper (I assume Algorithm 3), but the numbering of lines in the paper is different.

- It would probably aid comprehensibility of the paper to move Section 5

(Related Work) more towards the beginning.

- The bibliography contains various capitalization mistakes (e.g. "ctl" vs.
"CTL" and "kripke" vs. "Kripke").

Review 2

I have very low confidence in the area, so my review is concerning the generality of the paper.
The paper appear identify correctly the problem and the previous work. Them define the formal preliminaries and an algorithms to perform forgetting in CTL. By the end they prove that the algorithm proposed is sound (I have not checked carefully the proof, but seems correct).

Review 3

This paper introduces a resolution calculus for computing the result of forgetting in computational tree logic (CTL). The paper uses the resolution calculus by Zhang, Hustadt and Dixon to resolve away symbols to be forgotten; then additional propositions (introduced by the normal form transformation) and indices (introduced by applications of the resolution rules) get eliminated. The authors show that their algorithm is sound and terminating in exponential time.
It would appear that all ideas in this paper have already been introduced in the literature, so authors' contribution is to 'make them work' in the context of CTL. The paper is reasonably well-written and so can be accepted for publication.
For a firmer accept, I would expect to see an evaluation. I could not find any references to the completeness of the procedure or whether forgetting always exists in CTL. Either a theoretical analysis of the cases where the algorithm always computes the result of forgetting or an experimental assessment of the success rate would definitely make the paper stronger.