A Clausal Resolution Method for Branching-Time Logic ECTL$^+$

Alexander Bolotov and Artie Basukoski
Harrow School of Computer Science,
University of Westminster, HA1 3TP, UK
A.Bolotov@wmin.ac.uk
A.Basukoski@wmin.ac.uk

Abstract

The branching-time temporal logic ECTL$^+$ is strictly more expressive than ECTL, in allowing Boolean combinations of fairness operators and single temporal operators. We show that the key elements of the resolution method previously defined for a weaker logic, ECTL, namely the clausal normal form, the concepts of step resolution and a temporal resolution, are applicable in this new framework of ECTL$^+$. We show that any ECTL$^+$ formula can be translated into its normal form using the technique previously defined for the logic ECTL. This enables us to apply to the generated set of clauses a resolution technique originally defined for the basic computation tree logic CTL.

1 Introduction

In [2, 3] a clausal resolution approach to CTL has been developed, extending the original definition of the method for the linear-time case [5]. In [1] a clausal resolution method has been developed to capture the extended computation tree logic, ECTL. In this work we show that adding a corresponding technique to cope with fairness constraints, an ECTL$^+$ formula can be translated to a set of clauses in the normal form, to which we then apply a clausal resolution technique initially defined for the logic CTL.

2 Syntax of ECTL$^+$

In the language of ECTL$^+$ we extend the language of linear-time temporal logic, which uses future time $\Box$ (always), $\Diamond$ (sometime), $\bigcirc$ (next time), $\mathcal{U}$ (until) and $\mathcal{W}$ (unless), by path quantifiers **A** (on all future paths) and **E** (on some future path). In the syntax of ECTL$^+$, similar to CTL and ECTL, we distinguish *state* (S) and *path* (P) formulae, such that well formed formulae are state formulae. These are inductively defined below (where C is a formula of classical propositional logic)

$S ::= C|S \wedge S|S \vee S|S \Rightarrow S|\neg S|\mathbf{A}P|\mathbf{E}P$
$P ::= \Box S|\Diamond S|\bigcirc S|S\,\mathcal{U}\,S|S\,\mathcal{W}\,S|\Box\Diamond S|\Diamond\Box S$
$P ::= P \wedge P|P \vee P|P \Rightarrow P|\neg P$

Examples of ECTL$^+$ formulae that are not expressible in a weaker logic ECTL, are $\mathbf{A}(\Diamond\Box A \wedge \Box\Diamond A)$, $\mathbf{E}(\Diamond\Box A \vee \bigcirc\neg A$ (where A is any ECTL$^+$ formula), which express the Boolean combination of fairness properties or temporal operators in the scope of a path quantifier.

3 Key ideas in the extension of the resolution method for ECTL^+

Since ECTL^+ incorporates the boolean combination of simple temporal operators as in CTL^+, as well as the fairness constraints of ECTL, additional transformation procedures needed to be implemented to bring an ECTL^+ formula into the required normal form SNF_{CTL}. For ECTL^+ formulae that are CTL^+ formulae we rely on the fact that CTL^+ is as expressive as CTL and that equivalences have been provided in [4] for the translation of CTL^+formulae into equivalent CTL formulae. For the case of nested temporal operators we invoke the method already developed for ECTL [1].

Managing embedded boolean combinations of temporal subformulae in ECTL^+.
The unique feature of ECTL^+ is its ability to express boolean combinations of simple temporal operators and fairness constraints. The major contribution of this paper is the introduction of these transformations and the relevant correctness proofs. We also bridge the gap in establishing the correctness of the transformation procedure for ECTL, namely we show that if the set of clauses generated for an ECTL formula is satisfiable then the original formula is satisfiable.

4 Conclusions and Future Work

We have described the extension of the clausal resolution method to the useful branching-time logic ECTL^+. With the proof that SNF_{CTL} can be served as the normal form for ECTL^+, the algorithm becomes fully functional for the latter. We believe that a number of techniques explored in this paper will be useful in developing the resolution method for the extensions of ECTL^+ to CTL^*. Taking into account these observations, we define a future task to *refine* this algorithm, and having analysed the *complexity* of the clausal resolution method for ECTL and ECTLplus, to develop corresponding prototype systems.

References

[1] A. Bolotov. Clausal resolution for extended computation tree logic ectl.

[2] A. Bolotov and C. Dixon. Resolution for Branching Time Temporal Logics: Applying the Temporal Resolution Rule. In *Proceedings of the 7th International Conference on Temporal Representation and Reasoning (TIME2000)*, pages 163–172, Cape Breton, Nova Scotia, Canada, 2000. IEEE Computer Society.

[3] A. Bolotov and M. Fisher. A Clausal Resolution Method for CTL Branching Time Temporal Logic. *Journal of Experimental and Theoretical Artificial Intelligence.*, 11:77–93, 1999.

[4] E. A. Emerson. Temporal and Modal Logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science: Volume B, Formal Models and Semantics.*, pages 996–1072. Elsevier, 1990.

[5] M. Fisher. A Resolution Method for Temporal Logic. In *Proc. of the XII International Joint Conference on Artificial Intelligence (IJCAI)*, pages 99–104, 1991.