CTL-RP: A Computational Tree Logic Resolution Prover *

Lan Zhang⋆ Ullrich Hustadt⋆ Clare Dixon⋆

⋆ Department of Computer Science, University of Liverpool
Liverpool, L69 3BX, UK
{Lan.Zhang,U.Hustadt,CLDixon}@liverpool.ac.uk

1 Introduction

Temporal logic is considered an important tool in many different areas of Artificial Intelligence and Computer Science, including the specification and verification of concurrent and distributed systems. Computational Tree Logic CTL (Clarke and Emerson, 1982) is a branching-time temporal logic. Here we present the first resolution theorem prover for CTL, CTL-RP, which implements the sound and complete clausal resolution calculus $\mathsf{R}^{\succ,S}_{\mathrm{CTL}}$ (Zhang et al., 2008) based on an earlier calculus by Bolotov (2000). The calculus $\mathsf{R}^{\succ,S}_{\mathrm{CTL}}$ is designed in order to allow the use of classical first-order resolution techniques to emulate the rules of the calculus. We take advantage of this approach in the development of our prover CTL-RP which uses the first-order theorem prover SPASS (Weidenbach et al., 2007).

2 Normal form for CTL $\mathrm{SNF}^{\mathrm{g}}_{\mathrm{CTL}}$ and clausal resolution calculus $\mathsf{R}^{\succ,S}_{\mathrm{CTL}}$

The calculus $\mathsf{R}^{\succ,S}_{\mathrm{CTL}}$ operates on formulae in a clausal normal form called Separated Normal Form with Global Clauses for CTL, denoted by $\mathrm{SNF}^{\mathrm{g}}_{\mathrm{CTL}}$. The language of $\mathrm{SNF}^{\mathrm{g}}_{\mathrm{CTL}}$ clauses is defined over an extension of CTL in which we label certain formulae with an index ind taken from a countably infinite index set Ind and it consists of formulae of the following form.

$\mathbf{A}\Box(\mathbf{start} \Rightarrow \bigvee_{j=1}^{k} m_j)$	(initial clause)	$\mathbf{A}\Box(\bigwedge_{i=1}^{n} l_i \Rightarrow \mathbf{E}\bigcirc \bigvee_{j=1}^{k} m_{j\,\langle ind\rangle})$	(**E**-step clause)
$\mathbf{A}\Box(\mathbf{true} \Rightarrow \bigvee_{j=1}^{k} m_j)$	(global clause)	$\mathbf{A}\Box(\bigwedge_{i=1}^{n} l_i \Rightarrow \mathbf{A}\Diamond l)$	(**A**-sometime clause)
$\mathbf{A}\Box(\bigwedge_{i=1}^{n} l_i \Rightarrow \mathbf{A}\bigcirc \bigvee_{j=1}^{k} m_j)$	(**A**-step clause)	$\mathbf{A}\Box(\bigwedge_{i=1}^{n} l_i \Rightarrow \mathbf{E}\Diamond l_{\langle LC(ind)\rangle})$	(**E**-sometime clause)

where **start** is a propositional constant, l_i $(1 \leq i \leq n)$, m_j $(1 \leq j \leq k)$ and l are literals, that is atomic propositions or their negation, ind is an element of Ind. The symbols ind and $LC(ind)$ represent indices and limit closure of indices, respectively. As all clauses are of the form $\mathbf{A}\Box(P \Rightarrow D)$ we often simply write $P \Rightarrow D$ instead.

We have defined a set of transformation rules which allows us to transform an arbitrary CTL formula into an equi-satisfiable set of $\mathrm{SNF}^{\mathrm{g}}_{\mathrm{CTL}}$ clauses, a complete description of which can be found in (Zhang et al., 2008). The transformation rules are similar to those in (Bolotov, 2000), but modified to allow for global clauses.

$\mathsf{R}^{\succ,S}_{\mathrm{CTL}}$ consists of two types of resolution rules, *step* resolution rules (SRES1 to SRES8) and *eventuality* resolution rules (ERES1 and ERES2). Motivated by refinements of propositional and first-order resolution, we restrict the applicability of step resolution rules by means of an atom ordering $\succ$ and a selection function S, which helps to prune the search space dramatically. Due to lack of space, we only present two of the step resolution rules and one of the eventuality resolution rules. In the following l is a literal, P and Q are conjunctions of literals, and C and D are disjunctions of literals.

SRES2 $$\frac{P \Rightarrow \mathbf{E}\bigcirc(C \vee l)_{\langle ind\rangle},\ Q \Rightarrow \mathbf{A}\bigcirc(D \vee \neg l)}{P \wedge Q \Rightarrow \mathbf{E}\bigcirc(C \vee D)_{\langle ind\rangle}}$$

SRES3 $$\frac{P \Rightarrow \mathbf{E}\bigcirc(C \vee l)_{\langle ind\rangle},\ Q \Rightarrow \mathbf{E}\bigcirc(D \vee \neg l)_{\langle ind\rangle}}{P \wedge Q \Rightarrow \mathbf{E}\bigcirc(C \vee D)_{\langle ind\rangle}}$$

ERES1 $$\frac{P^{\dagger} \Rightarrow \mathbf{E}\bigcirc\mathbf{E}\Box l,\ Q \Rightarrow \mathbf{A}\Diamond\neg l}{Q \Rightarrow \mathbf{A}(\neg P^{\dagger}\, \mathcal{W}\, \neg l)}$$

where $P^{\dagger} \Rightarrow \mathbf{E}\bigcirc\mathbf{E}\Box l$ represents a set of $\mathrm{SNF}^{\mathrm{g}}_{\mathrm{CTL}}$ clauses which together imply $P^{\dagger} \Rightarrow \mathbf{E}\bigcirc\mathbf{E}\Box l$.

We develop a new completeness proof with a different approach from (Bolotov, 2000). The proof also shows that some eventuality resolution rules in (Bolotov, 2000), which are the most costly rules of the calculus, are redundant. The inference rules of $\mathsf{R}^{\succ,S}_{\mathrm{CTL}}$ can be used to decide the satisfiability of a given set N of $\mathrm{SNF}^{\mathrm{g}}_{\mathrm{CTL}}$ clauses by computing the saturation N' of N using at most an exponential number of inference steps; N is unsatisfiable iff N' contains a clause $\mathbf{true} \Rightarrow \mathbf{false}$ or $\mathbf{start} \Rightarrow \mathbf{false}$. This gives a complexity optimal EXPTIME decision procedure for CTL.

*This work is supported by EPSRC grant EP/D060451/1.

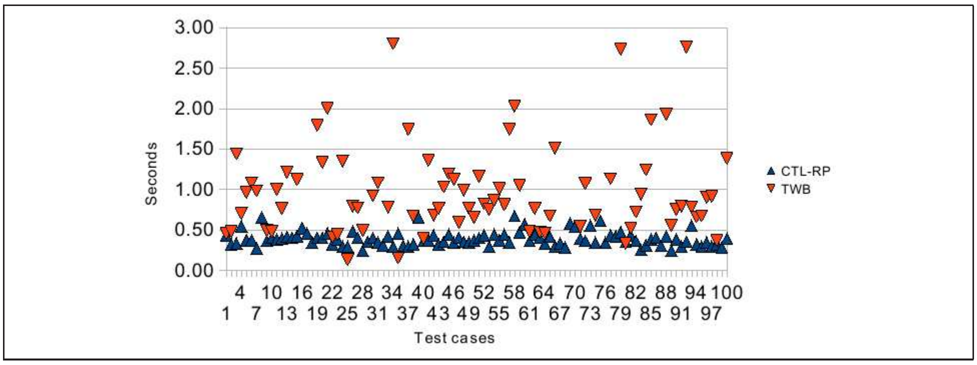

Figure 1: Performance on a set of benchmark formulae

3 CTL-RP

In order to obtain an efficient CTL theorem prover and to reuse existing state-of-the-art first-order resolution theorem provers, we adopt an approach analogous to that used in (Hustadt and Konev, 2004) to implement a resolution calculus for PLTL to implement the calculus $\mathsf{R}^{\succ,S}_{\mathrm{CTL}}$ and the associated decision procedure for CTL. A formal description of the approach and related proofs are presented in detail in (Zhang et al., 2008).

In our implementation of $\mathsf{R}^{\succ,S}_{\mathrm{CTL}}$, we first transform all $\mathrm{SNF}^{\mathrm{g}}_{\mathrm{CTL}}$ clauses except **A**- and **E**-sometime clauses into first-order clauses. Then we are able to use first-order ordered resolution with selection to emulate step resolution. For this part of the implementation we are using the theorem prover SPASS. **A**- and **E**-sometime clauses cannot be translated to first-order logic. Therefore, we continue to use the eventuality resolution rules ERES1 and ERES2 for inferences with **A**- and **E**-sometime clauses, respectively, and use the loop search algorithm presented in (Bolotov and Dixon, 2000) to find suitable premises for these rules. We utilise first-order ordered resolution with selection to perform the most costly task of "looking for merged clauses" in the loop search algorithm and we compute the results of applications of the eventuality resolution rules in the form of first-order clauses.

Besides CTL-RP, there is only one other CTL theorem prover we know of, namely a CTL module for the Tableau Workbench (TWB) (Abate and Goré, 2003). We have created several sets of benchmark formulae that we have used to compare CTL-RP version 00.09 with TWB version 3.4. The comparison was performed on a Linux PC with an Intel Core 2 CPU@2.13 GHz and 3G main memory, using the Fedora 9 operating system. In Figure 1, we show the experimental results on one of those sets of benchmark formulae. This set of benchmark formulae consists of one hundred formulae such that each formula specifies a randomly generated state transition system. The graph in Figure 1 indicates the CPU time in seconds required by TWB and CTL-RP to establish the satisfiability or unsatisfiability of each benchmark formula in the set of benchmark formulae. CTL-RP shows a much more stable performance on these benchmarks than TWB.

References

P. Abate and R. Goré. The Tableaux Workbench. In *Proc. TABLEAUX'03*, pages 230–236. Springer, 2003.

A. Bolotov. *Clausal Resolution for Branching-Time Temporal Logic*. PhD thesis, Manchester Metropolitan University, 2000.

A. Bolotov and C. Dixon. Resolution for Branching Time Temporal Logics: Applying the Temporal Resolution Rule. In *Proc. TIME'00*, pages 163–172. IEEE, 2000.

E. M. Clarke and E. A. Emerson. Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic. In *Logic of Programs, Workshop*, volume 131 of *LNCS*, pages 52–71. Springer, 1982.

U. Hustadt and B. Konev. TRP++: A Temporal Resolution Prover. In *Collegium Logicum*, pages 65–79. Kurt Gödel Society, 2004.

C. Weidenbach, R. A. Schmidt, T. Hillenbrand, R. Rusev, and D. Topic. System description: Spass version 3.0. In *Proc. CADE-21*, volume 4603 of *LNCS*, pages 514–520, 2007.

L. Zhang, U. Hustadt, and C. Dixon. First-order Resolution for CTL. Technical Report ULCS-08-010, Department of Computer Science, University of Liverpool, 2008.