One-pass Tableaux for Computation Tree Logic

Pietro Abate and Rajeev Goré and Florian Widmann
Computer Sciences Laboratory
The Australian National University
Australia

Overview

Syntax and Semantics of Computation Tree Logic (CTL)

Known Decision Procedures

The Description Logic Experience

Tableau Rules of Our Calculus for CTL

Example

Complexity

Syntax and Semantics of Computational Tree Logic

Atoms: $p ::= p_0 \mid p_1 \mid p_2 \cdots$

Fml: $\varphi ::= p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid E(\varphi\, U\, \varphi) \mid A(\varphi\, U\, \varphi) \mid A(\varphi\, B\, \varphi) \mid E(\varphi\, B\, \varphi)$

Transition Frame: pair (W, R) where W is a non-empty set of worlds and R is a total binary relation over W $(\forall w \in W.\, \exists v \in W.\, w\, R\, v)$.

Transition Sequence: σ in (W, R) is an infinite sequence $\sigma_0, \sigma_1, \sigma_2, \ldots$ of worlds in W such that $\sigma_i\, R\, \sigma_{i+1}$ for all $i \in N$.

w-sequence: σ in (W, R) is a transition sequence in (W, R) with $\sigma_0 = w$.

$\mathcal{B}(w)$: for $w \in W$ is the set of all w-sequences in (W, R)

Kripke Semantics

Model: $M = (W, R, L)$ is a transition frame (W, R) and a labelling function $L : W \to 2^{\mathsf{AP}}$ which associates with each world $w \in W$ a set $L(w)$ of propositional variables true at world w

$M, w \Vdash p$	iff	$p \in L(w)$, for $p \in \mathsf{AP}$
$M, w \Vdash \neg\psi$	iff	$M, w \nVdash \psi$
$M, w \Vdash \varphi \wedge \psi$	iff	$M, w \Vdash \varphi \;\&\; M, w \Vdash \psi$
$M, w \Vdash \varphi \vee \psi$	iff	$M, w \Vdash \varphi$ or $M, w \Vdash \psi$
$M, w \Vdash EX\varphi$	iff	$\exists v \in W.\, w\,R\,v \;\&\; M, v \Vdash \varphi$
$M, w \Vdash AX\varphi$	iff	$\forall v \in W.\, w\,R\,v \;\Rightarrow\; M, v \Vdash \varphi$
$M, w \Vdash E(\varphi\,U\,\psi)$	iff	$\exists \sigma \in \mathcal{B}(w).\, \exists i \in N.\, [M, \sigma_i \Vdash \psi \;\&\; \forall j < i.\, M, \sigma_j \Vdash \varphi]$
$M, w \Vdash A(\varphi\,U\,\psi)$	iff	$\forall \sigma \in \mathcal{B}(w).\, \exists i \in N.\, [M, \sigma_i \Vdash \psi \;\&\; \forall j < i.\, M, \sigma_j \Vdash \varphi]$
$M, w \Vdash E(\varphi\,B\,\psi)$	iff	$\exists \sigma \in \mathcal{B}(w).\, \forall i \in N.\, [M, \sigma_i \Vdash \psi \;\Rightarrow\; \exists j < i.\, M, \sigma_j \Vdash \varphi]$
$M, w \Vdash A(\varphi\,B\,\psi)$	iff	$\forall \sigma \in \mathcal{B}(w).\, \forall i \in N.\, [M, \sigma_i \Vdash \psi \;\Rightarrow\; \exists j < i.\, M, \sigma_j \Vdash \varphi]$

Smullyan's $\alpha-$ and $\beta-$notation

α	α_1	α_2
$\varphi \wedge \psi$	φ	ψ
$E(\varphi\, B\, \psi)$	$\sim\psi$	$\varphi \vee EX\, E(\varphi\, B\, \psi)$
$A(\varphi\, B\, \psi)$	$\sim\psi$	$\varphi \vee AX\, A(\varphi\, B\, \psi)$

β	β_1	β_2
$\varphi \vee \psi$	φ	ψ
$E(\varphi\, U\, \psi)$	ψ	$\varphi \wedge EX\, E(\varphi\, U\, \psi)$
$A(\varphi\, U\, \psi)$	ψ	$\varphi \wedge AX\, A(\varphi\, U\, \psi)$

Proposition: the formulae of the form $\alpha \leftrightarrow \alpha_1 \wedge \alpha_2$ and $\beta \leftrightarrow \beta_1 \vee \beta_2$ are valid

Eventualities: formulae ψ in $E(\varphi\, U\, \psi)$ and $A(\varphi\, U\, \psi)$

Known Decision Procedures

Complexity: satisfiability problem is EXPTime-complete

Ladner-like: consider all consistent subsets of the Fisher-Ladner closure, insert edges representing R as appropriate and then prune resulting graph

Two-pass Tableaux: construct a cyclic graph by applying tableau rules, then make multiple passes of this graph pruning inconsistent nodes and nodes containing unfulfilled eventualities (Ben-Ari, Manna, Emerson, Clarke)

Resolution Based: puts formulae into SNF and then uses multiple resolution rules specially designed rules (Fisher, Dixon, Peim, Basukoski, Bolotov)

Automata-Based: build a Büchi automaton and check whether the language accepted by the automaton is empty (Vardi, Wolper)

Advantage: meets the worst-case EXPTime complexity

Disadvantage: average-case and best-case is also EXPTime

The Description Logic Experience

Description Logics: very expressive extensions of multi-modal K

Complexity: EXPTime-complete logics

Tableaux Decision Procedures: suboptimal but “practical”

Optimisations: are crucial for efficiency

Ease of Implementation: is crucial for effective optimisations

Average Case: complexity is significantly better than EXPTime

Worst-case: complexity is sometimes worse than EXPTime but worst-case rarely arises in real-world examples

Properties of Our Tableaux for CTL

One-pass: we build a single-rooted finite tree with ancestor loops

Cut-free: there is no cut rule

Omega Rule Fee: there is no “finitised omega” rule

Worst-case Complexity: is 2EXPTime (for an EXPTime-complete problem)

Average-case Complexity: is much better because there is no initial phase which builds an exponential-sized data structure

Easy to Implement: involves simple operations like set-comparison, set-intersection, min and max on integers,

Easy to Optimise: using the techniques from description logics

Naive Implementation: `twb.rsise.anu.edu.au/demo`