A tree-style one-pass tableau for an extension of ECTL+

Bolotov, A., Hermo, M. and Lucio, P.

A paper presented at the 25th Workshop on Automated Reasoning: Bridging the Gap between Theory and Practice, University of Cambridge, Apr 2018, University of Cambridge.

A tree-style one-pass tableau for a extension of ECTL$^+$.

Alexander Bolotov[1] Montserrat Hermo[2] Paqui Lucio[2]

[1] University of Westminster, W1W 6UW, London, UK.
[2] University of the Basque Country, 20018-San Sebastián, Spain.

Abstract: Only restricted versions of fairness are expressible in the well-known branching temporal logics ECTL and ECTL$^+$, while the full expressiveness of branching-time logic in CTL* makes this logic extremely challenging for the application of the tableau technique. Tree-shaped one-pass tableaux are well suited for the automation and are amenable for the implementation. We present here a sound and complete method of tree-style one-pass tableau for a sub-logic of CTL* which is more expressive than the logic ECTL$^+$allowing the formulation of some fairness constraints with 'until' operator. The provided example follows by an algorithm for constructing a systematic tableau that enables to prove completeness.

1 Introduction

For the specification of the reactive and distributed systems, or, most recently, autonomous systems, where the modelling of the possibilities 'branching' into the future is essential, the branching-time logics (BTL) give us an appropriate framework. The most used class of formalisms are 'CTL' (Computation Tree Logic) type logics: CTL itself, ECTL (Extended CTL) [2] that was defined to enable simple fairness constraints but not their Boolean combinations and ECTL$^+$ ([3]) which further extends ECTL allowing Boolean combinations of ECTL fairness constraints (but not permitting their nesting). The literature on fairness constraints, even in linear-time setting, lacks the analysis of their formulation with a 'stronger' temporal operator - $\mathcal{U}$ ('until') such as $\Box(A\,\mathcal{U}\,B)$ or $A\,\mathcal{U}\,\Box B$. Here we bridge this gap, providing an analysis of such complex fairness constraints with $\mathcal{U}$ (also allowing the nesting of temporal operators) in the branching-time setting weaker than CTL*. Thus, we consider the logic that extends ECTL$^+$ with the modalities $\Box\,\mathcal{U}$ and $\mathcal{U}\,\Box$. While the addition of the former does not increase the ECTL$^+$ expressiveness[1], $A\,\mathcal{U}\,(\Box B)$ cannot be expressed in the ECTL$^+$ language. The fairness constraint $\mathsf{A}(p\,\mathcal{U}\,\Box q)$ can be read as 'q is true along all paths of the computation except possibly their finite initial interval, where p is true'.

Lamport Notation / CTL-type name	Formulae expressible here but not above
$\mathcal{B}(\mathcal{U},\circ)$ / CTL	
$\mathcal{B}(\mathcal{U},\circ,\Box\Diamond)$/ ECTL	$\mathsf{E}(\Box\Diamond q)$
$\mathcal{B}^+(\mathcal{U},\circ,\Box\Diamond)$ / ECTL$^+$	$\mathsf{E}(\Box\Diamond q \wedge \Box\Diamond r)$
$\mathcal{B}^+(\mathcal{U},\circ,\mathcal{U}\,\Box)$	$\mathsf{A}((p\,\mathcal{U}\,\Box q) \wedge (s\,\mathcal{U}\,\Box\neg q))$
$\mathcal{B}^\star(\mathcal{U},\circ)$ / CTL*	$\mathsf{A}\Diamond(\circ p \wedge \mathsf{E}\circ\neg p)$

Figure 1: BTL classification.

In Figure 1 we fit our logic into the hierarchy of branching-time logics: '$\mathcal{B}$' is used for 'Branching', followed by the set of only allowed modalities as parameters; $\mathcal{B}^+$ indicates admissible Boolean combinations of the modalities and $\mathcal{B}^\star$ reflects 'no restrictions' in either concatenations of the modalities or Boolean combinations between them. We present a tree-style one-pass tableau for the logic $\mathcal{B}^+(\mathcal{U},\circ,\mathcal{U}\,\Box)$ continuing the analogous developments in linear-time case [4].

$\mathcal{B}^+(\mathcal{U},\circ,\mathcal{U}\,\Box)$ language is defined with linear-time temporal operators $\Box$ (always), $\circ$ (next time), and $\mathcal{U}$ (until), and path quantifiers - A (on all future paths) and E (on some future path). The state (σ) and path (π) formulae are defined below (state formulae are wff).

$\sigma ::= L \mid \sigma_1 \wedge \sigma_2 \mid \sigma_1 \vee \sigma_2 \mid \mathsf{A}\pi \mid \mathsf{E}\pi$

$\pi ::= \pi_1 \wedge \pi_2 \mid \pi_1 \vee \pi_2 \mid \circ\sigma \mid \sigma\,\mathcal{U}\,\sigma \mid \sigma\,\mathcal{U}\,(\Box\sigma) \mid \Box\sigma \mid \Box(\sigma\,\mathcal{U}\,\sigma)$

2 The tableau method

While our tableaux are AND-OR trees with nodes labelled by sets of state formulae, the only rule which introduces AND-nodes is the next-state rule:

$$\frac{\Sigma, \mathsf{A}\circ\Phi_1, \ldots, \mathsf{A}\circ\Phi_n, \mathsf{E}\circ\Psi_1, \ldots, \mathsf{E}\circ\Psi_m}{\mathsf{A}\Phi_1, \ldots, \mathsf{A}\Phi_n, \mathsf{E}\Psi_1 \;\&\; \ldots \;\&\; \mathsf{A}\Phi_1, \ldots, \mathsf{A}\Phi_n, \mathsf{E}\Psi_m}$$

Figure 2: NEXT-STATE RULE. (Σ is a (possibly empty) set of literals; Φ_i, Ψ_i are non-empty sets of formulae.)

The next-state rule labels a branch that splits into m branches, at the node labelled by the premise of this rule. The conclusion of this rule uses the $\&$ symbol to reflect to generation of m AND-successor nodes. We also apply α- and β-rules: an α-rule expands a branch at the node labelled by its premise, with a node labelled by the conclusion; a β-rule splits a branch by two or three OR-nodes labelled by the formulae in its conclusion (separated by $|$).

We handle inputs in a new, branching-time, setting in 'analytic" way, extending similar construction for linear-time logic [4]. This extension is possible due to the definition of the 'context' in which eventualities are to be fulfilled in this new setting. Our β^+-rules are characteristic (and crucial!) for our construction. They tackle difficult cases of formulae in $\mathcal{B}^+(\mathcal{U},\circ,\mathcal{U}\,\Box)$. The β^+-rules, similarly to

[1] $\Box(A\,\mathcal{U}\,B)$ can be expressed in ECTL$^+$ by $\Box(A \vee B) \wedge \Box\Diamond B$.

$$\frac{\Sigma, \mathsf{E}(\Box(\sigma_1 \,\mathcal{U}\, \sigma_2) \wedge \Pi)}{\Sigma, \mathsf{E}((\sigma_1 \,\mathcal{U}\, \sigma_2) \wedge \circ\Box(\sigma_1 \,\mathcal{U}\, \sigma_2) \wedge \Pi)}$$

Figure 3: α-RULE ($\mathsf{E}\Box\,\mathcal{U}$). (Σ is a (possibly empty) set of state-formulae and Π is a (possibly empty) conjunction of path-formulae.)

$$\frac{\Sigma, \mathsf{A}((\Box\sigma) \vee \Pi)}{\Sigma, \sigma, \mathsf{A}((\circ\Box\sigma) \vee \Pi) \mid \Sigma, \mathsf{A}\Pi}$$

Figure 4: β-RULE ($\mathsf{A}\Box\sigma$). (Σ, σ is a set of state-formulae and Π is a (possibly empty) disjunction of path-formulae.)

β-rules, split a branch into two or three branches; these are the only rules that utilise 'context' to force the soonest satisfaction of the eventualities. The context is given by the sets of state (Σ) and path (Π) formulae. While, the outer-context was already used in the linear-time tableaux [4], for branching-time, the new concept of the 'inner-context' is introduced. Figure 5 shows a β^+ rule that handles a disjunction of formulae, including $\mathcal{U}$ in the scope of the A quantifier.

$$\frac{\Sigma, \mathsf{A}((\sigma \,\mathcal{U}\, \sigma') \vee \Pi)}{\Sigma, \sigma' \mid \Sigma, \sigma', \mathsf{A}(\circ((\sigma \wedge (\neg\Sigma \vee \varphi_\Pi)) \,\mathcal{U}\, \sigma') \vee \Pi) \mid \Sigma, \mathsf{A}\Pi}$$

Figure 5: β^+-RULE $(\mathsf{A}\,\mathcal{U}\,\sigma)^+$. (Σ, σ, σ' is a set of state-formulae and Π is a (possibly empty) disjunction of path-formulae, φ_Π is the state-formula introduced by Def. 1.)

Definition 1 *Let Π be a disjunction of path-formulae of the three forms $\Box\sigma$, $\sigma \,\mathcal{U}\, \Box\sigma'$ and $\Box(\sigma \,\mathcal{U}\, \sigma')$ where σ and σ' are state-formulae. The formula φ_Π to be the following disjunction of state-formulae:*

$$\bigvee_{\Box\sigma \in \Pi} \sigma \;\vee \bigvee_{\sigma \,\mathcal{U}\, \Box\sigma' \in \Pi} \sigma' \;\vee \bigvee_{\Box(\sigma \,\mathcal{U}\, \sigma') \in \Pi} \mathsf{E}(\mathrm{T} \,\mathcal{U}\, \sigma').$$

3 Examples

Our example of an open tableau illustrates the use of the inner context (see Fig. 6). This tableaux is constructed by a systematic algorithm that keeps (along a branch) exactly one –if there is some– marked eventuality forcing its fulfilment. In Fig. 6, the semicolon inside the A-quantifier stands for disjunction, the marked eventuality is in black-boxes and (Q$\circ$) is the next-state rule. Note that Π consists of a path-formula $\Box p$, so φ_Π is just p. Since the marked eventuality is $\mathrm{T}\,\mathcal{U}\,\neg p$ and the outer-context Σ is empty, the subformula $(\sigma \wedge (\neg\Sigma \vee \varphi_\Pi))$ in the conclusion of the rule in Fig. 5 is just p. It is notable that this inner-context p enables the central branch to loop, given a model of the initial formula that –in this branch– does not force the eventuality, but satisfies the other disjunct in the A-quantifier: $\Box p$.

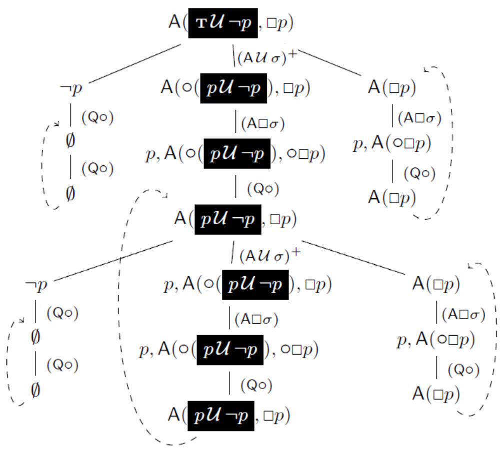

Figure 6: Open Tableau

4 Conclusion

We presented a tree-style one pass tableaux method for a new logic in the family of BTL – $\mathcal{B}^+(\mathcal{U}, \circ, \mathcal{U}\,\Box)$ – which extends the expressiveness of ECTL^+ fairness by a new class of fairness constraints utilising the $\mathcal{U}$ operator. The full details and the correctness proof are given in [1]. The tableaux rules that invoke the inner-context, enabled us to handle a particularly difficult class of formulae: A-disjunctive formulae with eventualities. The proof of correctness of β^+-rules is based on identifying relevant state-formulae inside specific path-modalities. This opens the prospect to study more expressive logics (eg $\mathsf{CTL}^\star$) by identifying subformulae that do not affect the 'context' which allows to simplify given structures. The presented technique, being the extension of a similar one for the linear-time setting, is amenable for implementation. In the refinement and implementation of our new algorithm we will rely on similar techniques used in the implementation of its linear-time analogue.

References

[1] A. Bolotov, M. Hermo, and P. Lucio. Extending fairness expressibility of ECTL+: a tree-style one-pass tableau approach, http://www.sc.ehu.es/jiwlucap/techreport18.pdf. Technical report, February 2018.

[2] E. A. Emerson and J. Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *Journal of Computer and System Sciences*, 30(1):1 – 24, 1985.

[3] E. A. Emerson and J. Y. Halpern. Sometimes and not never revisited: On branching versus linear time temporal logic. *J. ACM*, 33(1):151–178, January 1986.

[4] J. Gaintzarain, M. Hermo, P. Lucio, M. Navarro, and F. Orejas. Dual systems of tableaux and sequents for PLTL. *Journal of Logic and Algebraic Programming*, 78(8):701–722, 2009.