For me context is the key - from that comes the understanding of everything. - Kenneth Noland

One-pass Context-based Tableaux Systems for CTL and ECTL.

TIME 2020

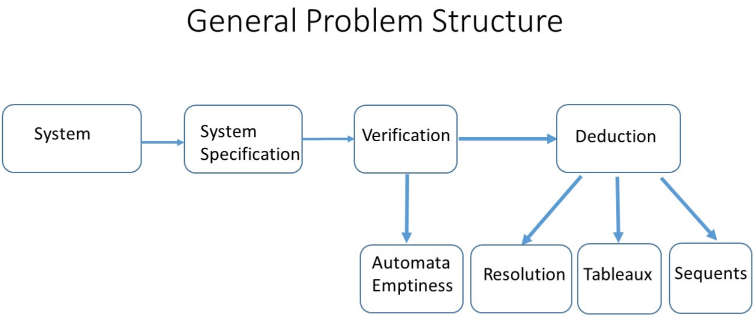
Alex Abuin
Alexander Bolotov
Montserrat Hermo
Paqui Lucio

25 September 2020

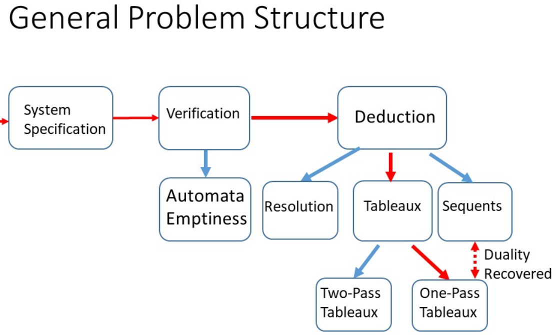
Structure of the Talk

- Motivation/General Problem Structure
- Branching-Time Logics Framework
- Tableaux: General Idea
- More motivation: 1 pass vs 2 pass Tableaux
- Our Approach - Context Based Tableaux
- Formalisms/Tableaux Rules for CTL
- Extension to ECTL
- Conclusions and Future Work

Motivation


General Problem Structure
System
System Specification
Verification
Deduction
Automata Emptiness
Resolution
Tableaux
Sequents

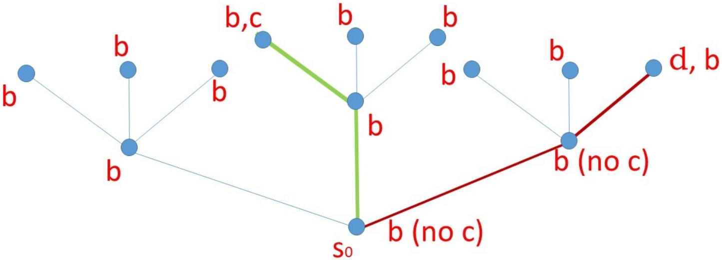
Motivation


General Problem Structure
System
System Specification
Verification
Deduction
Automata Emptiness
Resolution
Tableaux
Sequents
Duality Recovered
Two-Pass Tableaux
One-Pass Tableaux

Branching-Time Framework

- Temporal operators:
  - $\bigcirc$ - ‘at the next moment of time’, $\diamondsuit$ - ‘eventually’, $\square$ - ‘always’ and $\mathcal{U}$ - ‘until’
- paths quantifiers: A - ‘for all paths’; E - ‘there exists a path

$S_0 \models A\square b$ $\qquad$ $S_0 \models E\diamondsuit c$ $\qquad$ $S_0 \models E(\neg c\,\mathcal{U} d)$



Tableaux: very general idea for one and two pass methods

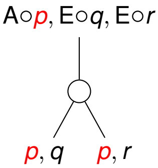
- Labelled graph
- Expansion rules: α rules generate a successor node $\dfrac{\phi \wedge \psi}{\phi, \psi}$
- β rules 'split' into alternative nodes $\dfrac{\phi \vee \psi}{\phi \mid \psi}$
- Use fixpoint properties of temporal operators:

  $\mathsf{A}\Box\phi = \phi \wedge \mathsf{A}\circ\mathsf{A}\Box\phi \qquad \mathsf{E}\phi\mathcal{U}\psi = \psi \vee (\phi \wedge \mathsf{E}\circ\mathsf{E}\phi\mathcal{U}\psi)$

$$\frac{\mathsf{A}\Box\phi}{\phi, \mathsf{A}\circ\mathsf{A}\Box\phi} \qquad \frac{\mathsf{E}\phi\mathcal{U}\psi}{\psi \mid \phi \wedge \mathsf{E}\circ\mathsf{E}\phi\mathcal{U}\psi}$$

Tableaux: building 'next' states

- A state is a node such that all its formulae are literals or of the form A○ϕ or E○ϕ
- This allows to build successor nodes to propagate ϕ.
- In our construction the tableaux graph is an AND-OR TREE:



Motivation: 2 pass vs 1 pass

- Two-pass tableaux:

1st pass: checks the validity of the given formula by creating a graph
2d pass: checks if all eventualities are satisfied.

- One-pass tableaux:

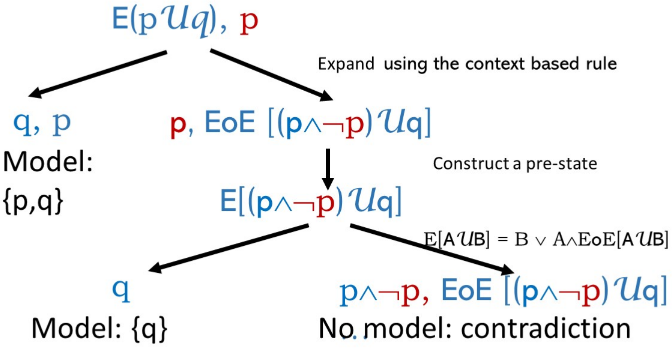
one pass: checks the validity of the input without auxiliary constructions,
need: mechanism to ensure this.

- Our mechanism to ensure a one pass - *CONTEXT* (of the selected eventuality)

Branching-Time Logics Framework: context

- CTL well formed formulae are state formulae - every temporal operator is preceded by a path quantifier:
  - $\mathsf{E}\Diamond p$ ✓
  - $\mathsf{E}\Box\Diamond p$ **X**
- Context
  - is a collection of state formulae accompanying the eventuality in the node.
  - forces eventualities to be fulfilled as soon as possible.
- In the node $\{\mathsf{E}(p\mathcal{U}q), p\}$ the context of $\mathsf{E}(p\mathcal{U}q)$ is p

CONTEXT forces fulfilment of eventualities



Branching-Time Logics Framework: CTL-type logics

State formulae

$$\sigma ::= \top \mid p \mid \sigma_1 \wedge \sigma_2 \mid \neg\sigma \mid \mathsf{E}\pi$$

Path formulae

$$\begin{array}{lll}
\pi_{\mathrm{CTL}^\star} & ::= & \sigma \mid \pi_1 \wedge \pi_2 \mid \neg\pi \mid \circ\pi \mid \pi\mathcal{U}\pi \\
\pi_{\mathrm{ECTL}^\#} & ::= & \sigma \mid \pi_1 \wedge \pi_2 \mid \neg\pi \mid \circ\sigma \mid \sigma\mathcal{U}(\sigma \wedge \Diamond\sigma) \\
 & & \quad \mid \Box(\sigma \vee \Box\sigma) \mid \sigma\mathcal{U}(\Box\sigma) \mid \Box(\sigma\mathcal{U}\sigma) \\
\pi_{\mathrm{ECTL}^+} & ::= & \sigma \mid \pi_1 \wedge \pi_2 \mid \neg\pi \mid \circ\sigma \mid \sigma\mathcal{U}\sigma \mid \Box\sigma \mid \Box\Diamond\sigma \mid \Diamond\Box\sigma. \\
\pi_{\mathrm{ECTL}} & ::= & \sigma \mid \neg\pi \mid \circ\sigma \mid \sigma\mathcal{U}\sigma \mid \Box\sigma \mid \Box\Diamond\sigma \mid \Diamond\Box\sigma. \\
\pi_{\mathrm{CTL}} & ::= & \sigma \mid \neg\pi \mid \circ\sigma \mid \sigma\mathcal{U}\sigma \mid \Box\sigma.
\end{array}$$

Development of One Pass Method for Branching Time

- Previous work - one pass tableaux for ECTL$^{\#}$
- This covers ECTL^{+}
- However, too complex to cover simpler logics - CTL and ECTL
- We made a ‘U-turn’ from ECTL# (and ECTL+) to bridge this gap

BTL Logics	$\mathsf{E}\Box\Diamond q$	$\mathsf{E}(\Box\Diamond q \wedge \Diamond\Box\neg q)$	$\mathsf{A}((p\mathcal{U}\Box q) \vee (s\mathcal{U}\Box\neg r))$	$\mathsf{A}\Diamond(\circ p \wedge \mathsf{E}\circ\neg p)$	One-pass Tableaux
$\mathcal{B}(\mathcal{U},\circ)$ (CTL)	X	X	X	X	This paper
$\mathcal{B}(\mathcal{U},\circ,\Box\Diamond)$ (ECTL)	$\checkmark$	X	X	X	This paper
$\mathcal{B}^{+}(\mathcal{U},\circ,\Box\Diamond)$ (ECTL^{+})	$\checkmark$	$\checkmark$	X	X	$\checkmark$
$\mathcal{B}^{+}(\mathcal{U},\circ,\mathcal{U}\Box)$ (ECTL$^{\#}$)	$\checkmark$	$\checkmark$	$\checkmark$	X	$\checkmark$
$\mathcal{B}^{*}(\mathcal{U},\circ)$ (CTL*)	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	X

One-pass Tableau Method for CTL - α- and β-Rules

- Standard but preserve context

$$(\wedge)\ \frac{\Sigma, \sigma_1 \wedge \sigma_2}{\Sigma, \sigma_1, \sigma_2} \qquad (\mathsf{Q}\Box)\ \frac{\Sigma, \mathsf{Q}\Box\sigma}{\Sigma, \sigma, \mathsf{Q}\circ\mathsf{Q}\Box\sigma}$$

$$(\vee)\ \frac{\Sigma, \sigma_1 \vee \sigma_2}{\Sigma, \sigma_1 \ \mid\ \Sigma, \sigma_2} \qquad (\mathsf{Q}\mathcal{U})\ \frac{\Sigma, \mathsf{Q}(\sigma_1 \mathcal{U} \sigma_2)}{\Sigma, \sigma_2 \ \mid\ \Sigma, \sigma_1, \mathsf{Q}\circ\mathsf{Q}(\sigma_1 \mathcal{U} \sigma_2)}$$

$$(\mathsf{Q}\mathcal{R})\ \frac{\Sigma, \mathsf{Q}(\sigma_1 \mathcal{R} \sigma_2)}{\Sigma, \sigma_1, \sigma_2 \ \mid\ \Sigma, \sigma_2, \mathsf{Q}\circ\mathsf{Q}(\sigma_1 \mathcal{R} \sigma_2)} \qquad (\mathsf{Q}\Diamond)\ \frac{\Sigma, \mathsf{Q}\Diamond\sigma}{\Sigma, \sigma \ \mid\ \Sigma, \mathsf{Q}\circ\mathsf{Q}\Diamond\sigma}$$

One-pass Tableau Method for CTL: Next-state Rule

- Responsible for obtaining AND-OR tree

$$(\mathrm{Q}\circ)\ \frac{\Sigma, \mathsf{A}\circ\sigma_1, \ldots, \mathsf{A}\circ\sigma_\ell, \mathsf{E}\circ\sigma'_1, \ldots, \mathsf{E}\circ\sigma'_k,}{\sigma_1, \ldots, \sigma_\ell, \sigma'_1 \ \& \ \ldots \ \& \ \sigma_1, \ldots, \sigma_\ell, \sigma'_k}$$

where Σ is a set of literals.

$$\mathsf{A}\circ\mathsf{A}(\mathbf{F}\,\mathcal{R}\,\neg q), \mathsf{E}\circ\mathsf{E}(p\,\mathcal{U}\,q) \wedge \mathsf{E}\circ\neg q$$

$(\wedge)$

$$\mathsf{A}\circ\mathsf{A}(\mathbf{F}\,\mathcal{R}\,\neg q), \mathsf{E}\circ\mathsf{E}(p\,\mathcal{U}\,q), \mathsf{E}\circ\neg q$$

$(\mathrm{Q}\circ)$

$$\mathsf{A}(\mathbf{F}\,\mathcal{R}\,\neg q), \mathsf{E}(p\,\mathcal{U}\,q) \qquad \mathsf{A}(\mathbf{F}\,\mathcal{R}\,\neg q), \neg q$$

One-pass Tableau Method for CTL: CONTEXT

- β^+-Rule – manages CONTEXT

$$(\mathsf{Q}\mathcal{U})^+ \quad \frac{\Sigma, \mathsf{Q}(\sigma_1 \mathcal{U} \sigma_2)}{\Sigma, \sigma_2 \;\mid\; \Sigma, \sigma_1, \mathsf{Q}{\circ}\mathsf{Q}((\sigma_1 \wedge \sim\Sigma')\mathcal{U}\sigma_2)}$$

where

$\Sigma' = \Sigma \setminus \{(\mathsf{A}\circ)^i \mathsf{A}\Box\sigma \mid i \geq 0 \text{ and } (\mathsf{A}\circ)^i \mathsf{A}\Box\sigma \in \Sigma\}$ and

$(\mathsf{A}\circ)^i$ stands for i times $\mathsf{A}\circ$.

$$\mathsf{A}(\mathbf{F}\,\mathcal{R}\,\neg q),\ \mathsf{E}(p\,\mathcal{U}\,q)$$

$(\mathsf{E}\mathcal{U})^+$

$$\mathsf{A}(\mathbf{F}\,\mathcal{R}\,\neg q),\ q \qquad\qquad \mathsf{A}(\mathbf{F}\,\mathcal{R}\,\neg q),\ p,\ \mathsf{E}{\circ}\,\mathsf{E}((p \wedge \mathsf{E}(\mathbf{T}\,\mathcal{U}\,q))\,\mathcal{U}\,q)$$

Tableaux Algorithm $\tau(\sigma)$

procedure SYSTEMATIC_TABLEAU(Σ_0) ▷ where Σ_0: set of CTL state formulae
  if Σ_0 is not uniform **then** T := Uniform_Tableau(Σ_0)
  while T has at least one expandable node **do**
    ▷ Invariant: Any expandable node of T is labelled by an uniform set
    Choose any node ℓ in T such that $\tau(\ell)$ is expandable ▷ $\tau(\ell)$ is uniform
    if there is no eventuality in $\tau(\ell)$ **then** $T := T[\ell \leftarrow$Uniform_Tableau($\tau(\ell)$)]
    else
      Eventuality_Selection($\tau(\ell)$)
      Apply_β^+-rule($\tau(\ell)$)
      Let ℓ_1, ℓ_2 the two children of ℓ
      for i = 1 .. 2 **do**
        if ℓ_i is expandable and $\tau(\ell_i)$ is not uniform **then**
          $T := T[\ell_i \leftarrow$Uniform_Tableau($\tau(\ell_i)$)]

Tableau Extension to ECTL

Two additional rules

$$(\mathsf{Q}\Box\Diamond)\ \frac{\Sigma, \mathsf{Q}\Box\Diamond\sigma}{\Sigma, \mathsf{Q}\Diamond\sigma, \mathsf{Q}\circ\mathsf{Q}\Box\Diamond\sigma} \qquad (\mathsf{Q}\Diamond\Box)\ \frac{\Sigma, \mathsf{Q}\Diamond\Box\sigma}{\Sigma, \mathsf{Q}\Box\sigma \ \mid\ \Sigma, \mathsf{Q}\Diamond\sigma, \mathsf{Q}\circ\mathsf{Q}\Diamond\Box\sigma}$$

that are based on these logical equivalences:

$$\mathsf{E}\Box\Diamond\sigma \equiv \mathsf{E}\Diamond\sigma \wedge \mathsf{E}\circ\mathsf{E}\Box\Diamond\sigma \qquad \mathsf{E}\Diamond\Box\sigma \equiv \mathsf{E}\Box\sigma \vee (\mathsf{E}\Diamond\sigma \wedge \mathsf{E}\circ\mathsf{E}\Diamond\Box\sigma)$$

$$\mathsf{A}\Box\Diamond\sigma \equiv \mathsf{A}\Diamond\sigma \wedge \mathsf{A}\circ\mathsf{A}\Box\Diamond\sigma \qquad \mathsf{A}\Diamond\Box\sigma \equiv \mathsf{A}\Box\sigma \vee (\mathsf{A}\Diamond\sigma \wedge \mathsf{A}\circ\mathsf{A}\Diamond\Box\sigma)$$

Conclusions and Future Work

- Context-based tableau extends naturally from linear-time to branching-time.
  - CTL tableau is much simpler than $\text{ECTL}^{\#}$ one, where two types of context (outer and inner) are used.
- Context-based tableau keeps the duality with sequent calculus.
  - Application in certified model checking: sequent proofs as certificates.
- Intuitive tableau method for CTL and ECTL.
  - Well suited for the automation and amenable for the implementation.