Proving Correctness of Labeled Transition Systems by Semantic Tableaux

Wolfgang May

Institut für Informatik

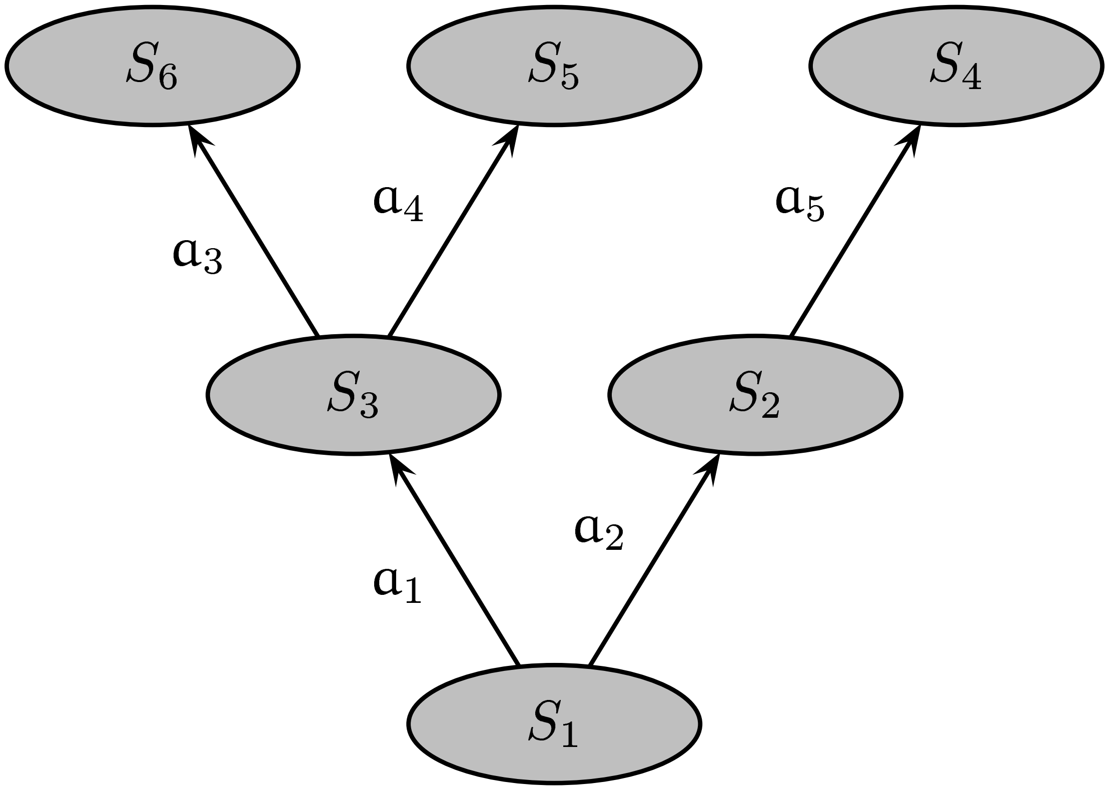
Universität Freiburg

Germany

May 13, 1997

Process Verification

States + Actions



Kripke-Structures

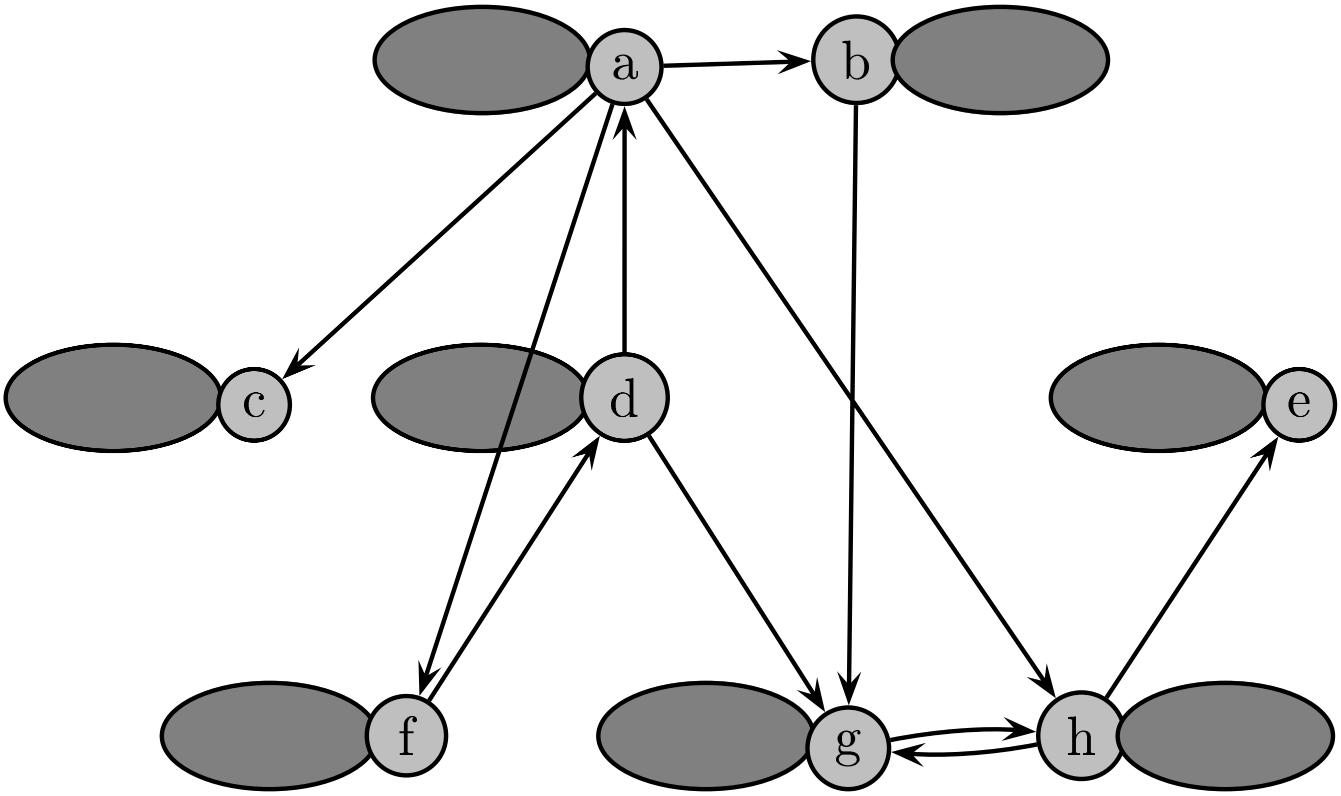
$$\mathbf{K} = (\mathbf{G}, \mathbf{R}, \mathbf{M})$$

G: A set of states (names)

$\mathbf{R} \subset \mathbf{G} \times \mathbf{G}$: An accessibility relation

M: $\mathbf{G} \rightarrow$ {First-order interpretations over a signature Σ}

Path: sequence $(g_0, g_1, \ldots)$.



Review: First-order CTL

- Every first-order formula is a CTL-state formula.

- Let F and G CTL-state formulas, P a CTL-path formula, x a variable. Then

- $\neg F$, $F \wedge G$, and $F \vee G$ are CTL-state formulas.
- $\forall x : F$ and $\exists x : F$ are CTL-state formulas.

Modalities:

- $\circ F$, $\Box F$, $\Diamond F$, and $(F \;\mathsf{until}\; G)$ are CTL-path formulas.
- $\neg P$ is a CTL-path formula.

Path quantifiers:

- $\mathsf{A}P$ and $\mathsf{E}P$ are CTL-state formulas.

- Every CTL-state formula is a CTL-formula.

Labeled Kripke-Structures

"Actions as First-Class Citizens"

$$\mathbf{K} = (\mathbf{G}, \mathbf{A}, \mathbf{R}, \mathbf{M})$$

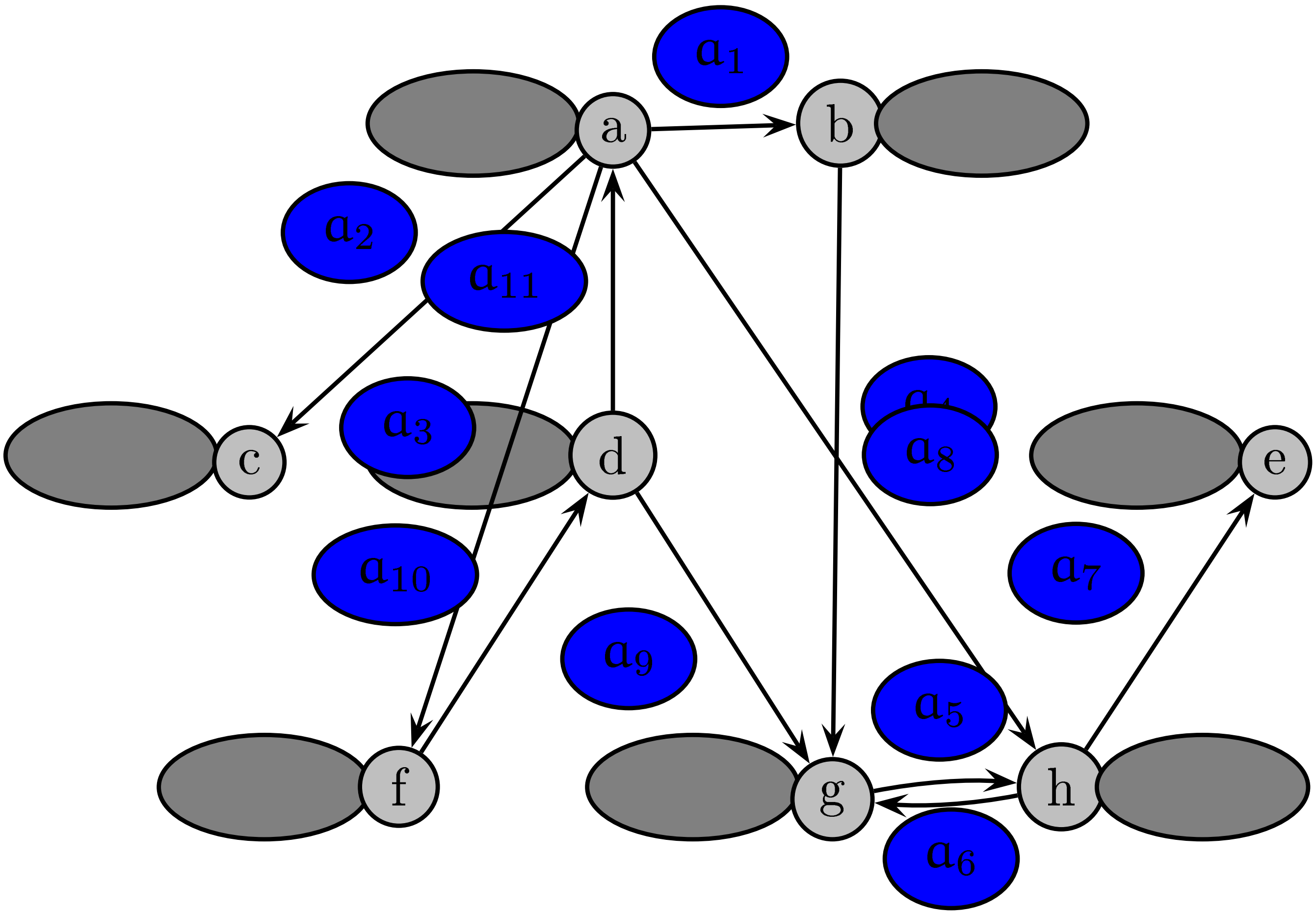
G: A set of states (names),

A: A set of actions,

$\mathbf{R} \subset \mathbf{G} \times 2^{\mathbf{A}} \times \mathbf{G}$: An accessibility relation,

M: $\mathbf{G} \rightarrow$ {First-order interpretations over a signature Σ}

Path: sequence $(g_0, \mathfrak{a}_1, g_1, \mathfrak{a}_2, g_2 \ldots)$.



Transitions

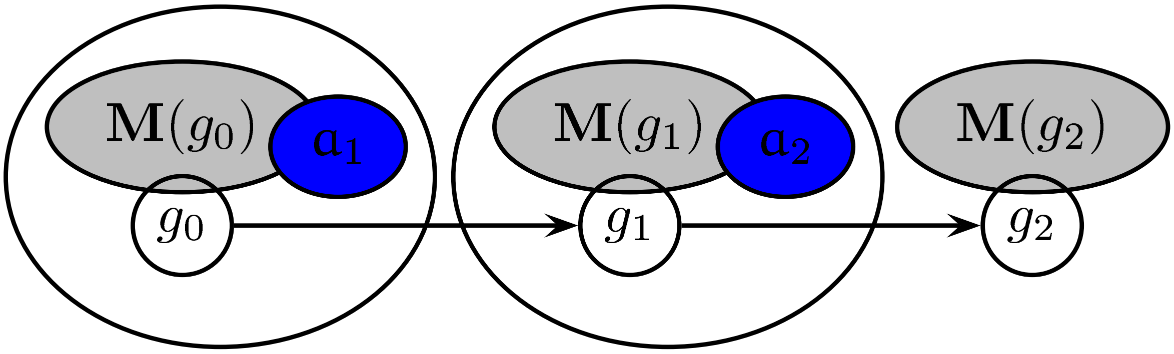
- Signature Σ_A of action symbols,
- For an n-ary action symbol a and $u_1, \ldots, u_n \in \mathbf{U}$, $a(u_1, \ldots, u_n)$ is an action.

Action symbols take the role of predicate symbols:

- T is an action formula.
- For $a \in \Sigma_A$ and terms $t_1, \ldots, t_n \in \mathsf{Term}_\Sigma$, $a(t_1, \ldots, t_n)$ is an action formula.
- For action formulas A and B, and a variable x, $\neg\mathrm{A}$, $\mathrm{A} \vee \mathrm{B}$, and $\exists x : \mathrm{A}$ are action formulas.

Transition labels $\mathfrak{a} \rightsquigarrow$ interpretations of Σ_A:
$b(u_1, \ldots, u_n) \in \mathfrak{a} \rightsquigarrow \mathfrak{a}(b(u_1, \ldots, u_n)) = \mathsf{true}$

Interpretation of Action Formulas



- $(\mathfrak{a}, \mathbf{I}, \chi) \models \top$.
- For every action symbol b, $t_1, \ldots, t_n \in \mathsf{Term}_\Sigma$ and $\mathfrak{a} \in 2^{\mathbf{A}}$:

$$(\mathfrak{a}, \mathbf{I}, \chi) \models b(t_1, \ldots, t_n) \;:\Leftrightarrow\; (\mathbf{I}(t_1, \chi), \ldots, \mathbf{I}(t_n, \chi)) \in \mathfrak{a}(b)\ .$$

- $\wedge$, $\vee$, $\neg$, $\exists$, and $\forall$ as usual.

MCTL: Coupling Actions and States

"!" **Conjunction:** Some transition satisfies A and the reached state satisfies F.

"?" **Implication:** If some transition satisfies A, then the reached state satisfies F.

- Let F, G be MCTL-state formulas, A, B action formulas. Then
  $!ⓐF$, $?ⓐF$, $!\boxed{A}F$, $?\boxed{A}F$, $!\Diamond_A F$, $?\Diamond_A F$, $!(A, F)\mathsf{U}(B, G)$, and $?(A, F)\mathsf{U}(B, G)$ are MCTL-path formulas.

○ ("Nexttime"):

$!ⓐF$: The transition to the next state satisfies A and in the next state F holds.

$?ⓐF$: If the transition to the next state satisfies A, then F holds in the next state.

◇ ("Sometimes"):

$!\Diamond_A F$: There exists a transition in the future which satisfies A and in the state reached by this transition, F holds.

$?\Diamond_A F$: If there exists a first transition satisfying A, then in the state reached by this transition, F holds.

Formal Semantics

For $p = (g_0, \mathfrak{a}_1, g_1, \mathfrak{a}_2, g_2, \ldots)$ labeled path and χ a variable assignment:

$$
\begin{array}{lcl}
(p,\chi) \models \,!\textcircled{\scriptsize A}\, F & :\Leftrightarrow & (\mathfrak{a}_1, \mathbf{M}(g_0), \chi) \models \mathrm{A} \text{ and } (g_1,\chi) \models F. \\
(p,\chi) \models \,?\textcircled{\scriptsize A}\, F & :\Leftrightarrow & \text{if } (\mathfrak{a}_1, \mathbf{M}(g_0), \chi) \models \mathrm{A}, \text{ then } (g_1,\chi) \models F. \\
(p,\chi) \models \,!\boxed{\scriptstyle A}\, F & :\Leftrightarrow & \text{for all } i\text{: } (\mathfrak{a}_{i+1}, \mathbf{M}(g_i), \chi) \models \mathrm{A} \text{ and } (g_{i+1},\chi) \models F. \\
(p,\chi) \models \,?\boxed{\scriptstyle A}\, F & :\Leftrightarrow & \text{for all } i\text{: } \text{if } (\mathfrak{a}_{i+1}, \mathbf{M}(g_i), \chi) \models \mathrm{A}, \text{ then } (g_{i+1},\chi) \models F. \\
(p,\chi) \models \,!\Diamond_{\mathrm{A}} F & :\Leftrightarrow & \text{there is an } i \text{ s.t. } (\mathfrak{a}_{i+1}, \mathbf{M}(g_i), \chi) \models \mathrm{A} \text{ and } (g_{i+1},\chi) \models F. \\
(p,\chi) \models \,?\Diamond_{\mathrm{A}} F & :\Leftrightarrow & \text{if there is a least } i \text{ s.t. } (\mathfrak{a}_{i+1}, \mathbf{M}(g_i), \chi) \models \mathrm{A}, \\
 & & \quad \text{then } (g_{i+1},\chi) \models F \text{ holds.}
\end{array}
$$

$(p, \chi) \models {!}(\mathrm{A}, F)\mathsf{U}(\mathrm{B}, G) \;\; :\Leftrightarrow$

there is a least i s.t. $(\mathfrak{a}_{i+1}, \mathbf{M}(g_i), \chi) \models \mathrm{B}$ and $(\mathbf{M}(g_{i+1}), \chi) \models G$,

and for all $0 \leq j < i$, $(\mathfrak{a}_{j+1}, \mathbf{M}(g_j), \chi) \models \mathrm{A}$ and $(\mathbf{M}(g_{j+1}), \chi) \models F$.

$(p, \chi) \models {?}(\mathrm{A}, F)\mathsf{U}(\mathrm{B}, G) \;\; :\Leftrightarrow$

there is a least i s.t. $(\mathfrak{a}_{i+1}, \mathbf{M}(g_i), \chi) \models \mathrm{B}$ and $(\mathbf{M}(g_{i+1}), \chi) \models G$,

and for all $0 \leq j < i$, if $(\mathfrak{a}_{j+1}, \mathbf{M}(g_j), \chi) \models \mathrm{A}$ then $(\mathbf{M}(g_{j+1}), \chi) \models F$.

Example

- $\mathbf{M}(g) \models \mathsf{E}!\ \llparenthesis a(t_1, \ldots, t_n) \rrparenthesis\ \mathsf{true}$:
  $a(\mathbf{M}(g)(t_1), \ldots, \mathbf{M}(g)(t_n))$ is executable in g.
- $\mathbf{M}(g) \models \mathsf{A}?\ \llparenthesis a(t_1, \ldots, t_n) \rrparenthesis\ \mathsf{false}$
  $a(\mathbf{M}(g)(t_1), \ldots, \mathbf{M}(g)(t_n))$ is not executable in g.
- $\mathbf{M}(g) \models \mathsf{A}?\langle a(t_1, \ldots, t_n)\rangle\, F$: if $a(\mathbf{M}(g)(t_1), \ldots, \mathbf{M}(g)(t_n))$ is performed in some future transition, a state satisfying F is reached.

Tableaux Representation

Entities to be described:

- Elements of the universe. A new constant resp. function symbol is introduced by a δ-rule when an $\exists$-quantor is processed.
- States: States are introduced when required by an existence formula (e.g. $\langle A \rangle F$ or Ⓐ F).
- Transitions.
- Paths: Paths are named when required by an existence formula of the kind $\mathsf{E}P$.

Representation

- Tableau calculus with prefixes for states:

$$\boxed{a : F}$$

- Prefixes denoting transitions:

$$\boxed{p[a \circ b] : \mathrm{A}}$$

- Explicit description of paths:

$$\boxed{p : [a, \circ, b, \mathcal{F}, c, \ldots, \hat{\infty}]}$$

Naming of Entities

Elements of the universe:

δ-rule: $$\frac{\exists x : F}{F[\hat{f}(\mathsf{free}(T))/x]} \quad ; \hat{f} \text{ new.}$$

Elements of the Kripke-frame: States and paths:

Idea: *prefix symbols* ($\hat{\Gamma} := \{\hat{\alpha}, \hat{\beta}, \hat{\gamma}, \ldots\}$)
path symbols ($\hat{\Lambda} := \{\hat{\lambda}, \hat{\kappa}, \ldots\}$):

- prefixes (state descriptors): $\hat{\alpha}(t_1, \ldots, t_n)$.
  newly introduced states: $\hat{\alpha}(\mathsf{free}(T))$.
- path descriptors: $\hat{\lambda}(t_1, \ldots, t_n)$.
  newly introduced paths: $\hat{\lambda}(\mathsf{free}(T))$.
  $\leadsto$ Path formulas bound to paths:
  λP (cf. $\mathsf{A}P$, $\mathsf{E}P$).
- Arguments of prefixes and path descriptors contain only function symbols which are interpreted state independently.

Technicalities

Interpretation of prefix- and path symbols by $\Omega = (\phi, \pi, \psi)$.

Evaluation of prefixes and path terms: similarly to a first-order interpretation $\mathbf{I} = (I, \mathbf{U})$ with mapping and "universe":

$$\Phi = (\phi, \mathbf{P}(\mathbf{K})) \quad ,$$
$$\Pi = (\pi, \mathbb{N} \cup \{\infty\}) \quad ,$$
$$\Psi = (\psi, \mathbf{G}) \quad .$$

$\phi : \hat{\Lambda} \to \mathbf{U}^n \to Paths(\mathbf{K})$ maps every n-ary $\hat{\lambda} \in \hat{\Lambda}$ to a function $\phi(\hat{\lambda}) : \mathbf{U}^n \to Paths(\mathbf{K})$ resp. $\phi(\hat{\lambda}) : \mathbf{U}^n \to \mathbb{N} \to (2^{\mathbf{A}} \times \mathbf{G})$,

$\pi : \hat{\Lambda} \times (\hat{\Gamma} \cup \{\hat{\infty}\}) \to (\mathbf{U}^n \times \mathbf{U}^m) \to \mathbb{N} \cup \{\infty\}$ is an (in general not total) mapping of pairs of n-ary $\hat{\lambda} \in \hat{\Lambda}$ and m-ary $\hat{\gamma} \in \hat{\Gamma}$ to functions $\pi(\hat{\lambda}, \hat{\gamma}) : \mathbf{U}^n \times \mathbf{U}^m \to \mathbb{N} \cup \{\infty\}$ with $\pi(\lambda, \gamma) = \infty \;\Leftrightarrow\; \gamma = \hat{\infty}$.

$\psi : \hat{\Gamma} \to \mathbf{U}^n \to \mathbf{G}$ maps every n-ary $\hat{\gamma} \in \hat{\Gamma}$ to a function $\psi(\hat{\gamma}) : \mathbf{U}^n \to \mathbf{G}$.

Validity

- path information formulas:

$$I = \lambda : [\gamma_0, L_0, \gamma_1, L_1, \ldots, \gamma_n, L_n, \hat{\infty}]$$

consistent with Ω for a variable assignment χ if

- every $\hat{\gamma}$ occurs in I at most once, and
- for all i

$$\Pi(\lambda, \gamma_0, \chi) = 0 \;,\; \Pi(\lambda, \gamma_i, \chi) < \Pi(\lambda, \gamma_{i+1}, \chi) \;,$$
$$\text{and} \quad \Psi(\gamma_i, \chi) = \Phi(\lambda, \chi, \Pi(\lambda, \gamma_i, \chi)) \quad .$$

$(\mathbf{K}, \Omega, \chi) \models\!\!\!= I \quad :\Leftrightarrow$
I is consistent with Ω for χ, and for all $0 \leq i \leq n$:
$L_i = \circ : \Pi(\lambda, \gamma_{i+1}, \chi) = \Pi(\lambda, \gamma_i, \chi){+}1 \quad ,$
$L_i \neq \circ :$ for all j s.t. $\Pi(\lambda, \gamma_i, \chi) < j < \Pi(\lambda, \gamma_{i+1}, \chi)$:

$$(\Phi(\lambda, \chi, j), \chi) \models L_i \;,$$

- Prefixed formulas:

$$(\mathbf{K}, \Omega, \chi) \models\!\!\!= \gamma : F \quad :\Leftrightarrow \; (\Phi(\lambda, \chi)|_{\Pi(\lambda,\gamma,\chi)}, \chi) \models P$$
$$\Leftrightarrow \; (\Psi(\gamma, \chi), \chi) \models F$$

$$(\mathbf{K}, \Omega, \chi) \models\!\!\!= \lambda[\gamma \circ \delta] \!: \mathrm{A} \quad :\Leftrightarrow$$
$$(\Phi(\lambda, \chi)|_{\Pi(\lambda,\gamma,\chi)}, \chi) \models \,!\text{Ⓐ}\,\mathsf{true}$$

- Path formulas bound to paths:

$$(\mathbf{K}, \Omega, \chi) \models\!\!\!= \gamma : \lambda P \quad :\Leftrightarrow \; (\Phi(\lambda, \chi)|_{\Pi(\lambda,\gamma,\chi)}, \chi) \models P$$

The Tableau Calculus $\mathcal{TL}$

Initialization $\boxed{\dfrac{}{\hat{0} : \neg F}}$.

States: First-Order Tableau with prefixes

S–α: $\dfrac{\gamma : F \wedge G}{\begin{array}{c}\gamma : F \\ \gamma : G\end{array}}$

S–β: $\dfrac{\gamma : F \vee G}{\gamma : F \mid \gamma : G}$

S–γ: $\dfrac{\gamma : \forall x\, F}{\gamma : F[X/x]}$ with X a new variable.

S–δ: $\dfrac{\gamma : \exists x\, F}{\gamma : F[f(\mathsf{free}(T))/x]}$ with f a new function symbol and T the current branch.

Closure by states, S–$\perp$:

$$\frac{\begin{array}{c}\gamma : A \\ \gamma : B \\ \sigma(A) = \neg\sigma(B)\end{array}}{\begin{array}{c}\perp \\ \text{apply } \sigma_\gamma \text{ to the whole tableau.}\end{array}}$$

State Independent Terms

$\sigma_\gamma(X)$ is obtained from $\sigma(X)$ by replacing every function symbol $f \in \Sigma \backslash \Sigma^c$ by its localized symbol f_γ,

$$(M(g))(f_\gamma) := (M(\Psi(\gamma, \chi)))(f) \quad .$$

Transitions

Analogous:

T–α: $\dfrac{\lambda[\alpha{\circ}\beta] : \mathrm{A} \wedge \mathrm{B}}{\begin{array}{c}\lambda[\alpha{\circ}\beta] : \mathrm{A} \\ \lambda[\alpha{\circ}\beta] : \mathrm{B}\end{array}}$

T–β: $\dfrac{\lambda[\alpha{\circ}\beta] : \mathrm{A} \vee \mathrm{B}}{\lambda[\alpha{\circ}\beta] : \mathrm{A} \;\Big|\; \lambda[\alpha{\circ}\beta] : \mathrm{B}}$

T–γ: $\dfrac{\lambda[\alpha{\circ}\beta] : \forall x\, \mathrm{A}}{\lambda[\alpha{\circ}\beta] : \mathrm{A}[X/x]}$ with X a new variable.

T–δ: $\dfrac{\lambda[\alpha{\circ}\beta] : \exists x\, \mathrm{A}}{\lambda[\alpha{\circ}\beta] : \mathrm{A}[f(\mathsf{free}(T))/x]}$ f and T as for S–δ.

Dissolving Path Quantifiers

- $\gamma : \mathsf{E}P$:

$$\frac{\gamma : \mathsf{E}P}{\begin{array}{c}\hat{\kappa}(\mathsf{free}(T)) : [\gamma, \mathsf{true}, \hat{\infty}] \\ \gamma : \hat{\kappa}(\mathsf{free}(T))P\end{array}}$$	$\hat{\kappa}$ is a new path symbol, T the current branch

- $\gamma : \mathsf{A}P$:
  dissolved once along every path going through γ

- $\gamma : \lambda P$ dissolved along λ.

Common Schema:

$$\frac{\begin{array}{c}\text{prefixed formula} \\ \text{path information formula}\end{array}}{\begin{array}{c}\text{prefixed formulas} \\ \text{path information formulas}\end{array}}$$

Next State

$$
\begin{array}{c|c}
\multicolumn{2}{c}{\alpha : \mathsf{A}?\textcircled{A}\, F} \\
\multicolumn{2}{c}{\lambda : [\ldots, \alpha, \circ, \beta, \ldots]} \\
\hline
\lambda[\alpha\circ\beta]{:}\neg \mathrm{A} & \beta : F \\
 & \lambda[\alpha\circ\beta]{:}\mathrm{A}
\end{array}
\qquad
\begin{array}{c}
\alpha : \mathsf{A}!\textcircled{A}\, F \\
\lambda : [\ldots, \alpha, \circ, \beta, \ldots] \\
\hline
\beta : F \\
\lambda[\alpha\circ\beta] : \mathrm{A}
\end{array}
$$

$$
\begin{array}{c|c}
\multicolumn{2}{c}{\alpha : \mathsf{A}!\textcircled{A}\, F} \\
\multicolumn{2}{c}{\lambda : [\ldots, \alpha, L, \beta, \ldots],\ \ L \neq \circ} \\
\hline
\lambda : [\ldots, \alpha, \circ, \hat{\gamma}(\mathsf{free}(T)), L, \beta, \ldots] & \text{if } \beta \neq \hat{\infty}\text{:} \\
\hat{\gamma}(\mathsf{free}(T)) : L & \lambda : [\ldots, \alpha, \circ, \beta, \ldots] \\
\hat{\gamma}(\mathsf{free}(T)) : F & \beta : F \\
\lambda[\alpha\circ\hat{\gamma}(\mathsf{free}(T))] : \mathrm{A} & \lambda[\alpha\circ\beta] : \mathrm{A}
\end{array}
$$

Sometimes

$$\alpha : \mathsf{A}?\Diamond_{\mathrm{A}} F$$

$$\lambda : [\ldots, \alpha, L, \beta, \ldots],\ L \neq \circ$$

$$\alpha : \lambda!\textcircled{\scriptsize A}\, F$$

$$\lambda : [\ldots, \alpha, L \cup \{\mathsf{A}?\Diamond_{\mathrm{A}} F, \lambda!\llparenthesis\neg\mathrm{A}\rrparenthesis\ \mathsf{true}\},$$

$$\hat{\gamma}(\mathsf{free}(T)), L, \beta, \ldots]$$

$$\alpha : \lambda!\llparenthesis\neg\mathrm{A}\rrparenthesis\ \mathsf{true}$$

$$\hat{\gamma}(\mathsf{free}(T)) : L$$

$$\hat{\gamma}(\mathsf{free}(T)) : \lambda!\textcircled{\scriptsize A}\, F$$

---

if $\beta \neq \hat{\infty}$:

$$\lambda : [\ldots, \alpha, L \cup \{\mathsf{A}?\Diamond_{\mathrm{A}} F, \lambda!\llparenthesis\neg\mathrm{A}\rrparenthesis\ \mathsf{true}\}, \hat{\infty}]$$

$$\alpha : \lambda!\llparenthesis\neg\mathrm{A}\rrparenthesis\ \mathsf{true}$$

$$\beta : \mathsf{A}?\Diamond_{\mathrm{A}} F$$

if $\beta = \hat{\infty}$:

$$\lambda : [\ldots, \alpha, L \cup \{\mathsf{A}?\Diamond_{\mathrm{A}} F, \lambda!\llparenthesis\neg\mathrm{A}\rrparenthesis\ \mathsf{true}\}, \beta, \ldots]$$

$$\alpha : \lambda!\llparenthesis\neg\mathrm{A}\rrparenthesis\ \mathsf{true}$$

Results

- The calculus is correct.
- First-order MCTL is not compact (since FO-CTL is not)
- Any calculus for first-order MCTL is *not* complete.
- This calculus is *not* complete.
- This calculus is complete modulo inductive properties.

- Integrate data-oriented and action-oriented aspects.

$\leadsto$ Verification of processes given as transition systems.

- Further work: Integration of a tableau calculus for some process definition language / process algebra.

$\Box$

Previous Related Paper:

W. May and P. H. Schmitt.
A Tableau Calculus for First-Order Branching Time Logic.
Intl. Conf. on Formal and Applied Practical Reasoning, FAPR'96,
LNCS 1085, pages 399–413. Springer, 1996.