How to make a logic probabilistic?

Pedro Baltazar

SQIG - IT, Lisbon - Portugal

pedro.baltazar@ist.utl.pt

CMU, CMACS Seminar - January 14th, 2010

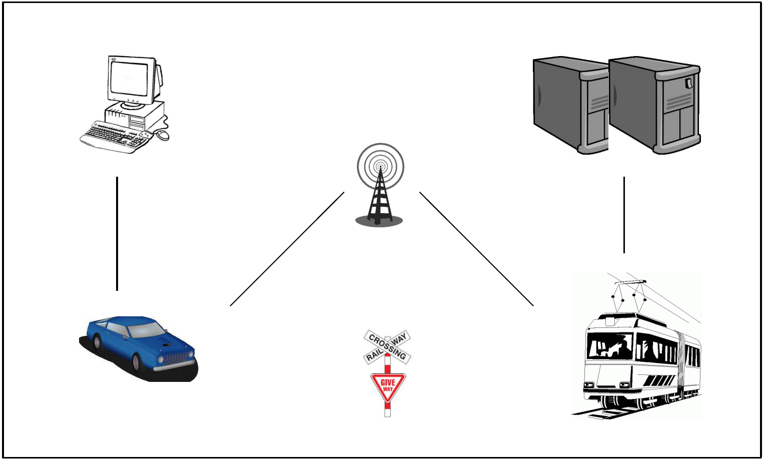
Sources:

- D. Henriques, M. Biscaia, P. Baltazar, and P. Mateus,
  Probabilistic quantified linear temporal logic: Model checking, SAT and complete Hilbert calculus.
  submitted for publication.

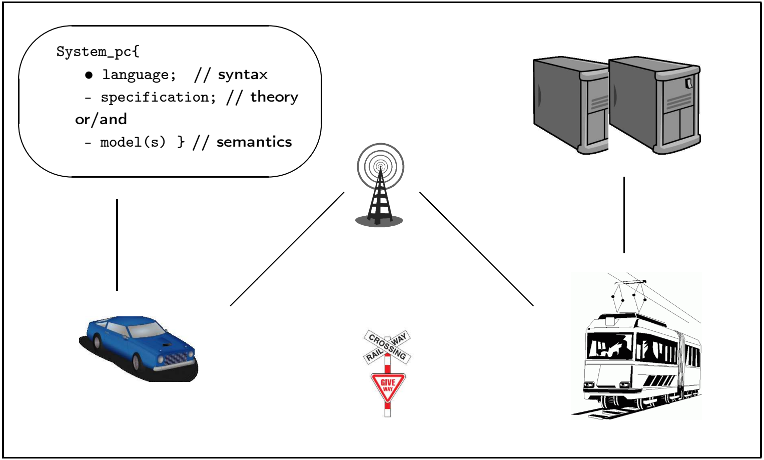
- P. Baltazar and P. Mateus.
  Temporalization of probabilistic propositional logic.
  LFCS 2009, LNCS, 2009.

- P. Baltazar, P. Mateus, R. Nagarajan, and N. Papanikolaou.
  Exogenous probabilistic computation tree logic.
  Electronic Notes in Theoretical Computer Science, 190(3) : 95–110, 2007.

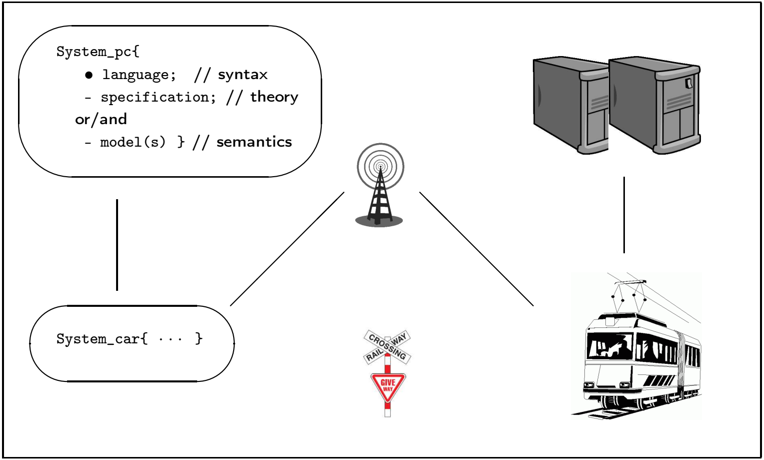
CPS : Cyber-Physical Systems



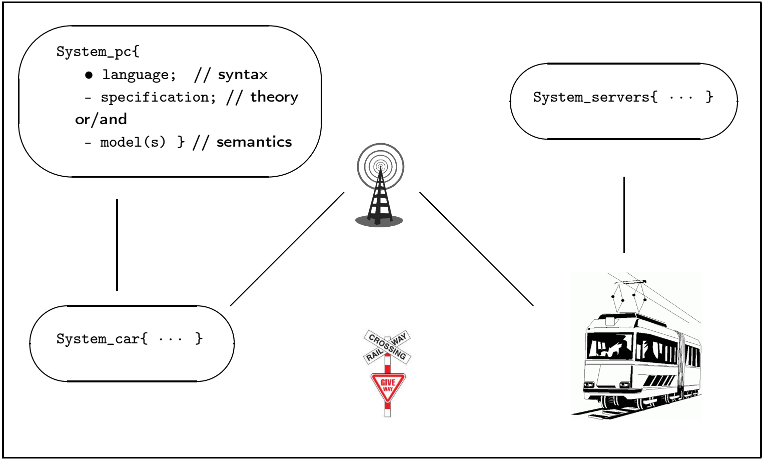
CPS : Cyber-Physical Systems

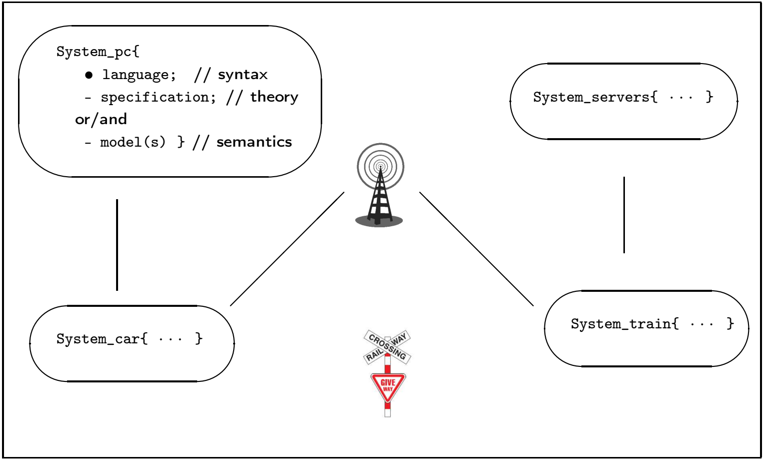


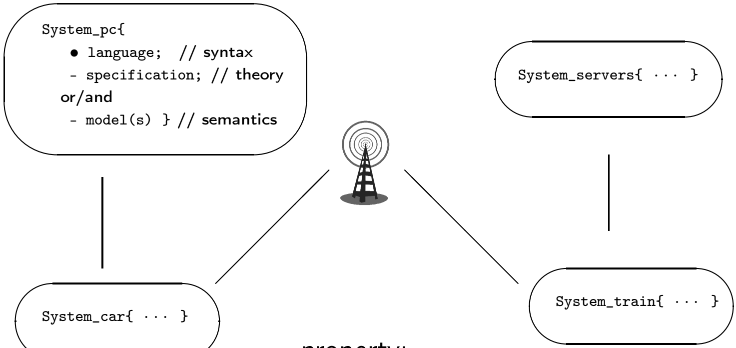
CPS : Cyber-Physical Systems



CPS : Cyber-Physical Systems

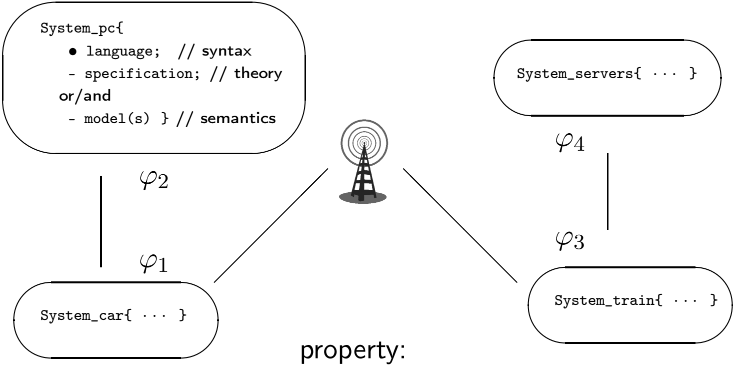



System_pc{
• language; // syntax
- specification; // theory
or/and
- model(s) } // semantics
System_servers{ · · · }
System_car{ · · · }
System_train{ · · · }
CROSSING
RAILWAY
GIVE WAY



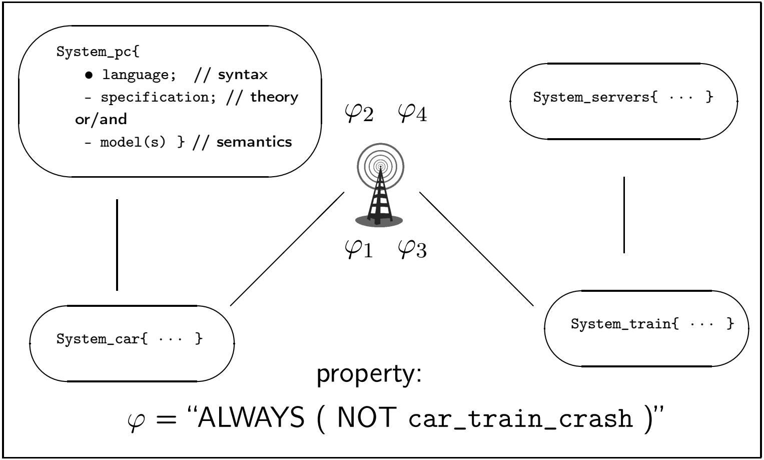
property:

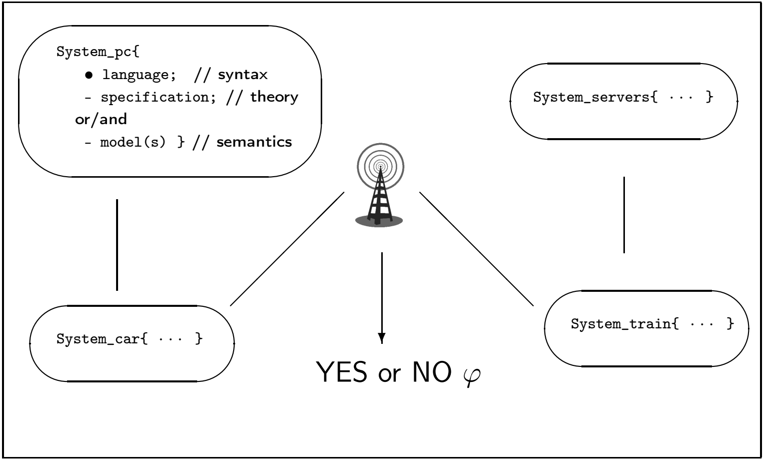
φ = "Always (NOT `car_train_crash`)"



property:

φ = "Always (NOT car_train_crash)"


System_pc{
• language; // syntax
- specification; // theory
or/and
- model(s) } // semantics
System_servers{ ··· }
φ2 φ4
φ1 φ3
System_car{ ··· }
System_train{ ··· }
property:
φ = "ALWAYS (NOT car_train_crash)"


System_pc{
• language; // syntax
- specification; // theory
or/and
- model(s) } // semantics
System_servers{ · · · }
System_car{ · · · }
YES or NO φ
System_train{ · · · }

(some) Logics in Verification

non-probabilistic	probabilistic

■ Propositional logic
■ Modal logic, CTL, LTL
■ First-order theories:
  ■ Presburger arithmetic
  ■ Pointer logic
  ⋮
■ Separation logic
■ Duration calculus
■ Metric temporal logic
■ Differential dynamic logic
⋮
	■ PCTL and PCTL*
	■ Continuous stochastic logic
	⋮
	

1. Exogenous Combination of Logics
2. Probabilization of Logics:
   - (generic) SAT
   - completeness
3. Examples:
   - EPPL - Probabilistic propositional logic
   - PTL - Probabilistic temporal logic
   - CTPL - Temporal EPPL

Definition (Satisfaction system)

Let $\mathcal{L}$ be a set of **formulas**, $\mathcal{M}$ a class of **models** and $\Vdash \subseteq \mathcal{M} \times \mathcal{L}$ a **satisfaction** relation.
The tuple $\mathscr{S} = \langle \mathcal{L}, \mathcal{M}, \Vdash \rangle$ is a **satisfaction system**.

Definition (Satisfaction system)

Let $\mathcal{L}$ be a set of **formulas**, $\mathcal{M}$ a class of **models** and $\Vdash \subseteq \mathcal{M} \times \mathcal{L}$ a **satisfaction** relation.
The tuple $\mathscr{S} = \langle \mathcal{L}, \mathcal{M}, \Vdash \rangle$ is a **satisfaction system**.

Definition (Morphism and weak morphism)

A **morphism** $h : \mathscr{S} \to \mathscr{S}'$ is a pair $\langle \overline{h}, \underline{h} \rangle$, with

$$\overline{h} : \mathcal{L} \to \mathcal{L}' \quad \text{and} \quad \underline{h} : \mathcal{M}' \to 2^{\mathcal{M}}$$

morphism: for all $m \in \underline{h}(m')$, $m \Vdash \varphi$ iff $m' \Vdash' \overline{h}(\varphi)$

Definition (Satisfaction system)

Let $\mathcal{L}$ be a set of **formulas**, $\mathcal{M}$ a class of **models** and $\Vdash \subseteq \mathcal{M} \times \mathcal{L}$ a **satisfaction** relation.
The tuple $\mathscr{S} = \langle \mathcal{L}, \mathcal{M}, \Vdash \rangle$ is a **satisfaction system**.

Definition (Morphism and weak morphism)

A **morphism** $h : \mathscr{S} \to \mathscr{S}'$ is a pair $\langle \overline{h}, \underline{h} \rangle$, with

$$\overline{h} : \mathcal{L} \to \mathcal{L}' \quad \text{and} \quad \underline{h} : \mathcal{M}' \to 2^{\mathcal{M}}$$

morphism: for all $m \in \underline{h}(m')$, $m \Vdash \varphi$ iff $m' \Vdash' \overline{h}(\varphi)$
weak morphism: exists $m \in \underline{h}(m')$, $m \Vdash \varphi$ iff $m' \Vdash' \overline{h}(\varphi)$

for all $\varphi \in \mathcal{L}$ and for all $m' \in \mathcal{M}_h \stackrel{def}{=} \{m' \in \mathcal{M}' : \underline{h}(m') \neq \emptyset\}$.

Definition ((Weak) equivalent systems)

$\mathscr{S}$ and $\mathscr{S}'$ are (resp. weak) equivalent if there are (resp. weak) total morphisms $h : \mathscr{S} \to \mathscr{S}'$ and $h' : \mathscr{S}' \to \mathscr{S}$ such that

$$\varphi \dashv\vDash' \overline{h}'(\overline{h}(\varphi)) \quad \text{and} \quad \psi \dashv\vDash \overline{h}(\overline{h}'(\psi)), \quad \text{for } \varphi \in \mathcal{L}, \psi \in \mathcal{L}'.$$

Denoted by

- equivalent, $\mathscr{S}_1 \approxeq_S \mathscr{S}_2$
- weak equivalent, $\mathscr{S}_1 \approxeq_S^w \mathscr{S}_2$

Definition ((Weak) equivalent systems)

$\mathscr{S}$ and $\mathscr{S}'$ are (resp. weak) equivalent if there are (resp. weak) total morphisms $h : \mathscr{S} \to \mathscr{S}'$ and $h' : \mathscr{S}' \to \mathscr{S}$ such that

$$\varphi \mathrel{=\!\!|\!\!\models'} \overline{h}'(\overline{h}(\varphi)) \quad \text{and} \quad \psi \mathrel{=\!\!|\!\!\models} \overline{h}(\overline{h}'(\psi)), \quad \text{for } \varphi \in \mathcal{L}, \psi \in \mathcal{L}'.$$

Denoted by

- equivalent, $\mathscr{S}_1 \approxeq_S \mathscr{S}_2$
- weak equivalent, $\mathscr{S}_1 \approxeq_S^w \mathscr{S}_2$

Proposition ($\langle \mathcal{L}, \mathcal{M}_1, \Vdash_1 \rangle \approxeq_S \langle \mathcal{L}, \mathcal{M}_2, \Vdash_2 \rangle$)

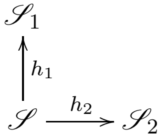
$\Gamma \models_1 \varphi$ *iff* $\Gamma \models_2 \varphi$.

Proposition ($\langle \mathcal{L}, \mathcal{M}_1, \Vdash_1 \rangle \approxeq_S^w \langle \mathcal{L}, \mathcal{M}_2, \Vdash_2 \rangle$)

$\models_1 \varphi$ *iff* $\models_2 \varphi$.

Exogenous Combination of Logics

Let $h_1 : \mathscr{S} \to \mathscr{S}_1$ and $h_2 : \mathscr{S} \to \mathscr{S}_2$ be morphisms.



Exogenous Combination of Logics

Let $h_1 : \mathscr{S} \to \mathscr{S}_1$ and $h_2 : \mathscr{S} \to \mathscr{S}_2$ be morphisms.

$$\begin{array}{ccc} \mathscr{S}_1 & & \\ \uparrow h_1 & & \\ \mathscr{S} & \xrightarrow{h_2} & \mathscr{S}_2 \end{array}$$

Idea: $\mathscr{S}_1 \otimes \mathscr{S}_2 = \langle \mathcal{L}_1 \otimes \mathcal{L}_2, \mathcal{M}', \Vdash' \rangle$, with $\mathcal{M}' \subseteq \mathcal{M}_1 \times \mathcal{M}_2$

Example (Parametrization)

$$\mathscr{S}_{(h_1 \Rightarrow h_2)} = \langle \mathcal{L}_1, \mathcal{M}_{(h_1 \Rightarrow h_2)}, \Vdash_1 \rangle,$$

where $\mathcal{M}_{(h_1 \Rightarrow h_2)} = \{m \in \mathcal{M}_{h_1} : \underline{h}_1(m) \subseteq \underline{h}_2(\mathcal{M}_2)\}$.

Definition (probabilization + globalization)

The **probabilization + globalization operator** transforms $\langle \mathcal{L}, \mathcal{M}, \Vdash \rangle$ into the system $\mathscr{S}^{(p+g)} = \langle \mathcal{L}^{(p+g)}, \mathcal{M}^{(p+g)}, \Vdash^{(p+g)} \rangle$:

- $\mathcal{L}^{(p+g)}$ is (with $\beta \in \mathcal{L}$ and $r \in Alg(\mathbb{R})$)

$$t ::= r \ [\!] \ \textstyle\int \beta \ [\!] \ (t + t) \ [\!] \ (t.t)$$

$$\varphi ::= [\beta] \ [\!] \ (t < t) \ [\!] \ (\sim\varphi) \ [\!] \ (\varphi \sqsupset \varphi);$$

Definition (probabilization + globalization)

The **probabilization + globalization operator** transforms $\langle \mathcal{L}, \mathcal{M}, \Vdash \rangle$ into the system $\mathscr{S}^{(p+g)} = \langle \mathcal{L}^{(p+g)}, \mathcal{M}^{(p+g)}, \Vdash^{(p+g)} \rangle$:

- $\mathcal{L}^{(p+g)}$ is (with $\beta \in \mathcal{L}$ and $r \in Alg(\mathbb{R})$)

$$t ::= r \;[\!]\; \textstyle\int\beta \;[\!]\; (t + t) \;[\!]\; (t.t)$$
$$\varphi ::= [\beta] \;[\!]\; (t < t) \;[\!]\; (\sim\varphi) \;[\!]\; (\varphi \sqsupset \varphi);$$

- $\mathcal{M}^{(p+g)}$ is the class of all $m = \langle S, \mathcal{F}, \mathbf{P}, V \rangle$, where $\langle S, \mathcal{F}, \mathbf{P} \rangle$ is a probability space, and $V : S \to \mathcal{M}$ is a *measurable valuation*, *i.e.* $V^{-1}[\beta] \stackrel{def}{=} \{s \in S : V(s) \Vdash \beta\} \in \mathcal{F}$;

Definition (probabilization + globalization)

The **probabilization + globalization operator** transforms $\langle \mathcal{L}, \mathcal{M}, \Vdash \rangle$ into the system $\mathscr{S}^{(p+g)} = \langle \mathcal{L}^{(p+g)}, \mathcal{M}^{(p+g)}, \Vdash^{(p+g)} \rangle$:

- $\mathcal{L}^{(p+g)}$ is (with $\beta \in \mathcal{L}$ and $r \in Alg(\mathbb{R})$)

$$t ::= r \;[\!]\; \textstyle\int\beta \;[\!]\; (t+t) \;[\!]\; (t.t)$$
$$\varphi ::= [\beta] \;[\!]\; (t < t) \;[\!]\; (\sim\varphi) \;[\!]\; (\varphi \sqsupset \varphi);$$

- $\mathcal{M}^{(p+g)}$ is the class of all $m = \langle S, \mathcal{F}, \mathbf{P}, V \rangle$, where $\langle S, \mathcal{F}, \mathbf{P} \rangle$ is a probability space, and $V : S \to \mathcal{M}$ is a *measurable valuation*, *i.e.* $V^{-1}[\beta] \overset{def}{=} \{s \in S : V(s) \Vdash \beta\} \in \mathcal{F}$;
- the satisfaction relation $\Vdash^{(p+g)}$ is given by
  - $[\![\textstyle\int\beta]\!]_m = \mathbf{P}(V^{-1}[\beta])$
  - $m \Vdash^{(p+g)} [\beta]$ iff $V(S) \Vdash \beta$;

  $(\ldots)$

weak morphism $h_p : \mathscr{S}^p \to \mathscr{S}_{\mathsf{RCF}}(\{x_\beta : \beta \in \mathcal{L}\} \cup X_{alg} \cup X)$

- $\Delta^p_{\mathscr{S}}$ - probabilistic (sub)theory of $\mathscr{S}$ in RCF

weak morphism $h_p : \mathscr{S}^p \to \mathscr{S}_{\mathsf{RCF}}(\{x_\beta : \beta \in \mathcal{L}\} \cup X_{alg} \cup X)$

- $\Delta^p_{\mathscr{S}}$ - probabilistic (sub)theory of $\mathscr{S}$ in RCF
- finite $\Delta^\Sigma_\varphi \subseteq \mathcal{L}_{\mathsf{RCF}}$, such that $\Delta^p_{\mathscr{S}} \vDash_{\mathsf{RCF}} \varphi$ iff $\Delta^\varphi_\Sigma \vDash_{\mathsf{RCF}} \varphi$

weak morphism $h_p : \mathscr{S}^p \to \mathscr{S}_{\mathsf{RCF}}(\{x_\beta : \beta \in \mathcal{L}\} \cup X_{alg} \cup X)$

- $\Delta^p_{\mathscr{S}}$ - probabilistic (sub)theory of $\mathscr{S}$ in RCF
- finite $\Delta^{\Sigma}_{\varphi} \subseteq \mathcal{L}_{\mathsf{RCF}}$, such that $\Delta^p_{\mathscr{S}} \vDash_{\mathsf{RCF}} \varphi$ iff $\Delta^{\varphi}_{\Sigma} \vDash_{\mathsf{RCF}} \varphi$

weak morphism $h_p : \mathscr{S}^p \to \mathscr{S}_{\mathsf{RCF}}(\{x_\beta : \beta \in \mathcal{L}\} \cup X_{alg} \cup X)$

- $\Delta^p_{\mathscr{S}}$ - probabilistic (sub)theory of $\mathscr{S}$ in RCF
- finite $\Delta^\Sigma_\varphi \subseteq \mathcal{L}_{\mathsf{RCF}}$, such that $\Delta^p_{\mathscr{S}} \vDash_{\mathsf{RCF}} \varphi$ iff $\Delta^\varphi_\Sigma \vDash_{\mathsf{RCF}} \varphi$

Proposition (Transference of SAT)

φ has a model in $\mathcal{M}^p$ iff $\overline{h}_p(\varphi) \wedge \Delta^\Sigma_\varphi$ has a model in $\mathbb{R}^X$.

weak morphism $h_p : \mathscr{S}^p \to \mathscr{S}_{\mathsf{RCF}}(\{x_\beta : \beta \in \mathcal{L}\} \cup X_{alg} \cup X)$

- $\Delta^p_{\mathscr{S}}$ - probabilistic (sub)theory of $\mathscr{S}$ in RCF
- finite $\Delta^\Sigma_\varphi \subseteq \mathcal{L}_{\mathsf{RCF}}$, such that $\Delta^p_{\mathscr{S}} \vDash_{\mathsf{RCF}} \varphi$ iff $\Delta^\varphi_\Sigma \vDash_{\mathsf{RCF}} \varphi$

Proposition (Transference of SAT)

φ has a model in $\mathcal{M}^p$ iff $\overline{h}_p(\varphi) \wedge \Delta^\Sigma_\varphi$ has a model in $\mathbb{R}^X$.

Theorem (SAT complexity lower-bound)

The SAT problem for $\mathscr{S}^p$ is at least PSPACE and obtaining a witness is at least EXPSPACE.

Proposition (Transference of weak completeness)

The axiomatization $\mathbb{AX}^p_{\mathscr{S}} \stackrel{def}{=} h_p^{-1}(\mathbb{AX}_{\mathsf{RCF}} + \Delta^p_{\mathscr{S}})$ is a sound and weakly complete axiomatization for $\mathscr{S}^p$.

Let $\varphi \in \mathcal{L}^{(p+g)}$

- $bf(\varphi) = \{\beta_1, \ldots, \beta_k\}$ - base formulas in φ

Let $\varphi \in \mathcal{L}^{(p+g)}$

- $bf(\varphi) = \{\beta_1, \ldots, \beta_k\}$ - base formulas in φ
- $atb(\varphi) = \{(\wedge_{i \in A} \beta_i) \wedge (\wedge_{i \notin A} \neg \beta_i) : A \in 2^k\}$ - atomic fml. for φ

Let $\varphi \in \mathcal{L}^{(p+g)}$

- $bf(\varphi) = \{\beta_1, \ldots, \beta_k\}$ - base formulas in φ
- $atb(\varphi) = \{(\wedge_{i \in A} \beta_i) \wedge (\wedge_{i \notin A} \neg\beta_i) : A \in 2^k\}$ - atomic fml. for φ
- $\Gamma_{\varphi,N}$ is the set of all $\beta \in atb(\varphi)$ such that $\models^g (\varphi \sqsupset [\neg\beta])$

Let $\varphi \in \mathcal{L}^{(p+g)}$

- $bf(\varphi) = \{\beta_1, \ldots, \beta_k\}$ - base formulas in φ
- $atb(\varphi) = \{(\wedge_{i \in A} \beta_i) \wedge (\wedge_{i \notin A} \neg \beta_i) : A \in 2^k\}$ - atomic fml. for φ
- $\Gamma_{\varphi,N}$ is the set of all $\beta \in atb(\varphi)$ such that $\vDash^g (\varphi \sqsupset [\neg \beta])$
- let $\psi_g = (\sqcap_{\beta \in \Gamma_{\varphi,N}} [\neg \beta])$ and $\psi_p = (\sqcap_{\beta \in \Gamma_{\varphi,N}} (\int \beta = 0))$

Let $\varphi \in \mathcal{L}^{(p+g)}$

- $bf(\varphi) = \{\beta_1, \ldots, \beta_k\}$ - base formulas in φ
- $atb(\varphi) = \{(\wedge_{i \in A} \beta_i) \wedge (\wedge_{i \notin A} \neg \beta_i) : A \in 2^k\}$ - atomic fml. for φ
- $\Gamma_{\varphi,N}$ is the set of all $\beta \in atb(\varphi)$ such that $\vDash^g (\varphi \sqsupset [\neg \beta])$
- let $\psi_g = (\sqcap_{\beta \in \Gamma_{\varphi,N}} [\neg \beta])$ and $\psi_p = (\sqcap_{\beta \in \Gamma_{\varphi,N}} (\int \beta = 0))$

Let $\varphi \in \mathcal{L}^{(p+g)}$

- $bf(\varphi) = \{\beta_1, \ldots, \beta_k\}$ - base formulas in φ
- $atb(\varphi) = \{(\wedge_{i \in A}\beta_i) \wedge (\wedge_{i \notin A}\neg\beta_i) : A \in 2^k\}$ - atomic fml. for φ
- $\Gamma_{\varphi,N}$ is the set of all $\beta \in atb(\varphi)$ such that $\vDash^g (\varphi \sqsupset [\neg\beta])$
- let $\psi_g = (\sqcap_{\beta \in \Gamma_{\varphi,N}}[\neg\beta])$ and $\psi_p = (\sqcap_{\beta \in \Gamma_{\varphi,N}}(\int\beta = 0))$

Let $\varphi^g \in \mathcal{L}^g$ and $\varphi^p \in \mathcal{L}^p$.

Proposition

A formula $(\varphi^g \sqcap \varphi^p)$ is satisfiable iff φ^g and $(\varphi^p \sqcap \psi_p)$ are satisfiable.

Let $\varphi \in \mathcal{L}^{(p+g)}$

- $bf(\varphi) = \{\beta_1, \ldots, \beta_k\}$ - base formulas in φ
- $atb(\varphi) = \{(\wedge_{i \in A} \beta_i) \wedge (\wedge_{i \notin A} \neg \beta_i) : A \in 2^k\}$ - atomic fml. for φ
- $\Gamma_{\varphi,N}$ is the set of all $\beta \in atb(\varphi)$ such that $\vDash^g (\varphi \sqsupset [\neg \beta])$
- let $\psi_g = (\sqcap_{\beta \in \Gamma_{\varphi,N}} [\neg \beta])$ and $\psi_p = (\sqcap_{\beta \in \Gamma_{\varphi,N}} (\int \beta = 0))$

Let $\varphi^g \in \mathcal{L}^g$ and $\varphi^p \in \mathcal{L}^p$.

Proposition

A formula $(\varphi^g \sqcap \varphi^p)$ is satisfiable iff φ^g and $(\varphi^p \sqcap \psi_p)$ are satisfiable.

Theorem (Transference of SAT)

If the SAT problem is solvable in $\mathscr{S}$, then it is solvable in $\mathscr{S}^{(p+g)}$.

Schema axiom: **IN** $([\beta] \sqsupset (\int\beta = 1))$

Schema axiom: **IN** $([\beta] \sqsupset (\int\beta = 1))$

Theorem (Transference of weak completeness)

If $\mathscr{S}$ has a weakly complete axiomatization $\mathbb{AX}_{\mathscr{S}}$, then

$$\mathbb{AX}_{\mathscr{S}}^{(p+g)} \stackrel{def}{=} \mathbb{AX}_{\mathscr{S}}^{p} + \mathbb{AX}_{\mathscr{S}}^{g} + \mathbf{IN}$$

is a weakly complete for $\mathscr{S}^{(p+g)}$.

Theorem (small-model theorem)

Every φ satisfiable has a model (probability dist.) of $2 \times size(\varphi)$.

Theorem (SAT complexity lower-bound)

The SAT problem for $\mathscr{S}^{(p+g)}$ is at least PSPACE and obtaining a witness is at least EXPSPACE.

Algorithm 1: $Sat_{\mathscr{S}}^{(p+g)}(\varphi)$

Input: formula $\varphi \in \mathcal{L}^{(p+g)}$

Output: $m = \langle M, \mathbf{P} \rangle$ ($m \Vdash^{(p+g)} \varphi$) or $\emptyset$ (No Model)

foreach $\varphi_i = (\varphi_{i,g} \sqcap \varphi_{i,p})$ *molecule of* φ **do**

    foreach $\Gamma \subseteq atb(\varphi)$ *of size* $\leq 2 \times Size(\varphi)$ **do**

        $M = \emptyset$;

        foreach $\beta \in \Gamma$ **do**

            $m_\beta \longleftarrow Sat_{\mathscr{S}}(\beta)$; $M = M \cup \{m_\beta\}$;

        end

        if $M \neq \emptyset$ *and* $M \Vdash^g \varphi_{i,g}$ **then**

            $\phi \longleftarrow \overline{h}_p(\varphi_{i,p} \sqcap \psi_{i,p})$;

            $\delta \longleftarrow \phi \wedge \Delta_\phi^\Sigma(\Gamma)$;

            $\eta \longleftarrow Sat_{\mathsf{RCF}}(\delta)$;

            if $\eta \neq \emptyset$ **then** **return** $m = \langle M, \mathbf{P}_\eta \rangle$;

        end

    end

end

return $\emptyset$ (No Model);

Let Λ be a countable set of propositional symbols.

Definition (EPPL)

$\mathscr{S}_{\mathsf{EPPL}}(\Lambda) = \langle \mathcal{L}_{\mathsf{EPPL}}(\Lambda), \mathcal{M}_{\mathsf{EPPL}}, \Vdash_{\mathsf{EPPL}} \rangle$:

- set of **formulas** $\mathcal{L}_{\mathsf{EPPL}}(\Lambda)$ is

$$\begin{aligned} \beta &::= \alpha \mathrel{[\!]} (\neg\beta) \mathrel{[\!]} (\beta \Rightarrow \beta) \\ t &::= r \mathrel{[\!]} \textstyle\int\beta \mathrel{[\!]} (t+t) \mathrel{[\!]} (t.t) \\ \varphi &::= [\beta] \mathrel{[\!]} (t < t) \mathrel{[\!]} (\sim\varphi) \mathrel{[\!]} (\varphi \sqsupset \varphi) \end{aligned}$$

  with $\alpha \in \Lambda$ and $r \in Alg(\mathbb{R})$;

Let $\{X_\alpha : \Omega \to 2\}_{\alpha\in\Lambda}$ be a stochastic process over $\langle \Omega, \mathcal{F}, \mathbf{P} \rangle$.

- $X_{(\neg\beta)} = 1 - X_\beta$;
- $X_{(\beta_1 \Rightarrow \beta_2)} = max\{1 - X_{\beta_1}, X_{\beta_2}\}$.

Definition (EPPL (cont.))

- the class of **models** $\mathcal{M}_{\mathsf{EPPL}}$ are the tuples $m = \langle S, \mathcal{F}, \mathbf{P}, \mathbf{X} \rangle$ such that $\mathbf{X} := \{X_\alpha : S \to 2\}_{\alpha \in \Lambda}$ is a stochastic process over $\langle S, \mathcal{F}, \mathbf{P} \rangle$;
- the **satisfaction** relation $\Vdash_{\mathsf{EPPL}}$ is defined by:
  - $[\![r]\!]_m = r$;
  - $[\![\int \beta]\!]_m = \mathbf{P}(X_\beta = 1)$
  - $[\![t_1 + t_2]\!]_m = [\![t_1]\!]_m + [\![t_2]\!]_m$;
  - $[\![t_1 . t_2]\!]_m = [\![t_1]\!]_m . [\![t_2]\!]_m$;

  - $m \Vdash_{\mathsf{EPPL}} [\beta]$ iff $X_\beta(s) = 1$ for all $s \in S$;
  - $m \Vdash_{\mathsf{EPPL}} (t_1 < t_2)$ iff $[\![t_1]\!]_m < [\![t_2]\!]_m$;
  - $m \Vdash_{\mathsf{EPPL}} (\sim\varphi)$ iff $m \nVdash_{\mathsf{EPPL}} \varphi$;
  - $m \Vdash_{\mathsf{EPPL}} (\varphi_1 \sqsupset \varphi_2)$ iff $m \nVdash_{\mathsf{EPPL}} \varphi_1$ or $m \Vdash_{\mathsf{EPPL}} \varphi_2$,

  for $m \in \mathcal{M}_{\mathsf{EPPL}}$ and $\varphi \in \mathcal{L}_{\mathsf{EPPL}}(\Lambda)$.

Theorem (equivalence)

$\mathscr{S}_{\textsf{EPPL}}(\Lambda) \approxeq_S \mathscr{S}^{(p+g)}_{\textsf{CPL}}(\Lambda)$.

Corollary (weak completeness)

The axiomatization $\mathbb{AX}^{(p+g)}_{\textsf{CPL}}$ *is weakly complete and sound for the satisfaction system* $\mathscr{S}_{\textsf{EPPL}}(\Lambda)$.

Theorem (SAT complexity)

The SAT problem for EPPL is PSPACE, and providing a witness (a model) is EXPSPACE.

Theorem (model-checking complexity)

It takes $O(|\varphi| \times |S|)$ *time to decide if an EPPL model* $m = \langle S, \mathbf{P}, \mathbf{X} \rangle$ *satisfies* φ.

Algorithm 2: $SAT(\varphi)$

Input: formula $\varphi \in \mathcal{L}^{(p+g)}(\Lambda)$

Output: $m = \langle M, \mathbf{P} \rangle$ $(m \Vdash^{(p+g)}_{\mathsf{CPL}} \varphi)$ or $\emptyset$ (No Model)

foreach $\varphi_i = (\varphi_{i,g} \sqcap \varphi_{i,p})$ *molecule of* φ **do**

  foreach $M \subseteq 2^{\Lambda(\varphi)}$ *of* $size(M) \leq 2 \times Size(\varphi_i)$ **do**

    if $M \Vdash^{g} \varphi_{i,g}$ **then**

      $\phi \longleftarrow \overline{h}_p(\varphi_{i,p} \sqcap \psi_{i,p})$;

      $\psi \longleftarrow \phi \wedge \Delta^{\Sigma}_{\phi}(M)$;

      $\eta \longleftarrow Sat_{\mathsf{RCF}}(\psi)$;

      if $\eta \neq \emptyset$ **then** **return** $m = \langle M, \mathbf{P}_\eta \rangle$;

    end

  end

end

return $\emptyset$ *(No Model)*;

$\mathbb{AX}_{\mathsf{EPPL}}$ is

G1 $\vdash_{\mathsf{EPPL}} [\beta]$ for all valid $\beta \in \mathcal{L}_{\mathsf{CPL}}(\Lambda)$;

G2 $\vdash_{\mathsf{EPPL}} ([\beta_1 \Rightarrow \beta_2] \sqsupset ([\beta_1] \sqsupset [\beta_2]))$;

IN $\vdash_{\mathsf{EPPL}} ([\beta] \sqsupset (\int\beta = 1))$;

EqN $\vdash_{\mathsf{EPPL}} (\int\neg\beta = 1 - \int\beta)$;

EqP $\vdash_{\mathsf{EPPL}} (\int\beta \geq 0)$;

EqA $\vdash_{\mathsf{EPPL}} (\int(\beta_1 \vee \beta_2) = \int\beta_1 + \int\beta_2 - \int(\beta_1 \wedge \beta_2))$;

RCF $\vdash_{\mathsf{EPPL}} \varphi$

if $\overline{h}_p(\varphi) \wedge (\wedge_{r \in alg(\varphi)} \varphi_r(x_r))$ is a valid formula in the real closed fields - RCF;

MP $\varphi_1, (\varphi_1 \sqsupset \varphi_2) \vdash_{\mathsf{EPPL}} \varphi_2$.

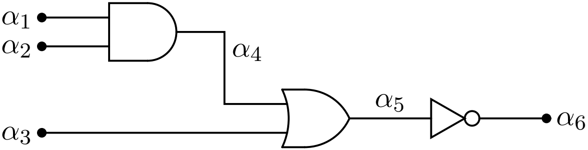


Figure: AND-OR-INVERTER (AOI21)

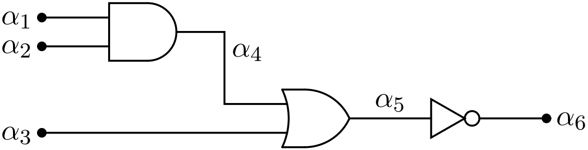


Figure: AND-OR-INVERTER (AOI21)

implementation:

$$(\int(\alpha_4 \Leftrightarrow \alpha_1 \wedge \alpha_2) > 0.97) \sqcap (\int(\alpha_5 \Leftrightarrow \alpha_3 \vee \alpha_4) > 0.99) \sqcap [(\alpha_6 \Leftrightarrow \neg \alpha_5)]$$

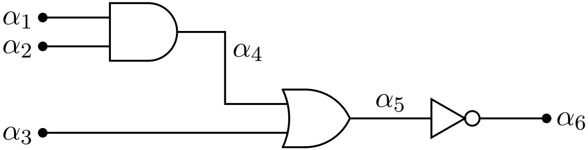


Figure: AND-OR-INVERTER (AOI21)

implementation:

$$(\textstyle\int(\alpha_4 \Leftrightarrow \alpha_1 \wedge \alpha_2) > 0.97) \sqcap (\int(\alpha_5 \Leftrightarrow \alpha_3 \vee \alpha_4) > 0.99) \sqcap [(\alpha_6 \Leftrightarrow \neg\alpha_5)]$$

specification:

$$(\textstyle\int \alpha_6 \Leftrightarrow \neg(\alpha_3 \vee (\alpha_1 \wedge \alpha_2)) \geq 0.98)$$

1) x = rand(); 2) y = rand(); 3) y = x ∨ y; 4) if (x) { 5) x = ¬ x; 6) else 7) x = x ∨ y; }	$\varphi_P = (\int \alpha_{x1} = 0.5) \sqcap (\int \alpha_{y1} = 0.5) \sqcap$ $\sqcap [\alpha_{y2} \Leftrightarrow \alpha_{x1} \vee \alpha_{y1}] \sqcap [\alpha_{x3} \Leftrightarrow \neg \alpha_{x2}] \sqcap$ $\sqcap [\alpha_{x4} \Leftrightarrow (\alpha_{x2} \vee \alpha_{y2})] \sqcap$ $\sqcap [\alpha_{x5} \Leftrightarrow (\alpha_{x2} ? \alpha_{x3} : \alpha_{x4})]$

Table: Translation to EPPL formula

$$\varphi_{saf} = ((\int \alpha_{x1} \leq 0.5) \sqcap (\int \alpha_{x2} \leq 0.5) \sqcap \ldots \sqcap (\int \alpha_{x5} \leq 0.5))$$

Program	EPPL formula

`1) x = rand();`
`2) y = rand();`
`3) y = x ∨ y;`
`4) if (x) {`
`5)    x = ¬ x;`
`6)    else`
`7)    x = x ∨ y; }`
	$\varphi_P = (\int \alpha_{x1} = 0.5) \sqcap (\int \alpha_{y1} = 0.5) \sqcap$
	$\sqcap [\alpha_{y2} \Leftrightarrow \alpha_{x1} \vee \alpha_{y1}] \sqcap [\alpha_{x3} \Leftrightarrow \neg \alpha_{x2}] \sqcap$
	$\sqcap [\alpha_{x4} \Leftrightarrow (\alpha_{x2} \vee \alpha_{y2})] \sqcap$
	$\sqcap [\alpha_{x5} \Leftrightarrow (\alpha_{x2} ? \alpha_{x3} : \alpha_{x4})]$

Table: Translation to EPPL formula

$$\varphi_{saf} = ((\int \alpha_{x1} \leq 0.5) \sqcap (\int \alpha_{x2} \leq 0.5) \sqcap \ldots \sqcap (\int \alpha_{x5} \leq 0.5))$$

$$SAT((\varphi_P \sqcap \sim \varphi_{saf})) = ?$$

Let Λ be a countable set of propositional symbols.

Definition (PTL)

The probabilistic temporal logic (PTL) over Λ, is the system $\mathscr{S}_{\mathsf{PTL}}(\Lambda) = \langle \mathcal{L}_{\mathsf{PTL}}(\Lambda), \mathcal{M}_{\mathsf{PTL}}, \Vdash_{\mathsf{PTL}} \rangle$ where $\mathcal{L}_{\mathsf{PTL}}(\Lambda)$ is

$$\beta ::= \alpha \mathrel{[\!]} (\neg\beta) \mathrel{[\!]} (\beta \Rightarrow \beta) \mathrel{[\!]} (\mathsf{X}\beta) \mathrel{[\!]} (\beta \mathsf{U} \beta)$$
$$t ::= r \mathrel{[\!]} (\textstyle\int\beta) \mathrel{[\!]} (t + t) \mathrel{[\!]} (t.t)$$
$$\varphi ::= [\beta] \mathrel{[\!]} (t \leq t) \mathrel{[\!]} (\sim\varphi) \mathrel{[\!]} (\varphi \sqsupset \varphi)$$

with $\alpha \in \Lambda$, and $r \in alg(\mathbb{R})$;

$\{X_\alpha : S \to 2\}_{\alpha\in\Lambda}$ is extended to a stochastic process over $\langle S^\omega, \mathcal{F}, \mathbf{P}\rangle$ (sequence space of a Markov chain).

- $X_{(\mathsf{X}\beta)}(\pi) = X_\beta(\pi^{(1)})$
- $X_{(\beta_1 \mathsf{U} \beta_2)}(\pi) = X_{\beta_2}(\pi) + X_{(\neg\beta_2)}(\pi).X_{\beta_1}(\pi).X_{(\beta_1 \mathsf{U} \beta_2)}(\pi^{(1)})$

Definition (PTL (cont.))

- $\mathcal{M}_{\mathsf{PTL}}$ is the class of tuples $m = \langle S, P, \mu, V\rangle$ where $\langle S, P, \mu\rangle$ is a Markov chain and $V : S \to 2^{\Lambda}$;

- $\Vdash_{\mathsf{PTL}}$ is defined by
  - $[\![r]\!]_m = r$;
  - $[\![\int\beta]\!]_m = \mathbf{P}(X_\beta = 1)$;
  - $[\![t_1 + t_2]\!]_m = [\![t_1]\!]_m + [\![t_2]\!]_m$;
  - $[\![t_1.t_2]\!]_m = [\![t_1]\!]_m.[\![t_2]\!]_m$;

  - $m \Vdash_{\mathsf{PTL}} [\beta]$ iff $K_m \Vdash_{\mathsf{LTL}} \beta$;
  - $m \Vdash_{\mathsf{PTL}} (t_1 < t_2)$ iff $[\![t_1]\!]_m < [\![t_2]\!]_m$;
  - $m \Vdash_{\mathsf{PTL}} (\sim\varphi)$ iff $m \not\Vdash_{\mathsf{PTL}} \varphi$;
  - $m \Vdash_{\mathsf{PTL}} (\varphi_1 \sqsupset \varphi_2)$ iff $m \not\Vdash_{\mathsf{PTL}} \varphi_1$ or $m \Vdash_{\mathsf{PTL}} \varphi_2$,

  for $m \in \mathcal{M}_{\mathsf{PTL}}$ and $\varphi \in \mathcal{L}_{\mathsf{PTL}}(\Lambda)$.

Proposition (Exogenous weak equivalent)

$\mathscr{S}_{\mathsf{PTL}}(\Lambda) \approxeq^{w}_{S} \mathscr{S}^{(p+g)}_{\mathsf{LTL}}(\Lambda)$.

Corollary (Transference of weak completeness)

The axiomatization

$$\mathbb{AX}^{(p+g)}_{\mathsf{LTL}} \stackrel{def}{=} \mathbb{AX}^{g}_{\mathsf{LTL}} + \mathbb{AX}^{p}_{\mathsf{LTL}} + \mathbf{IN}$$

is a sound and weakly complete axiomatization for $\mathscr{S}_{\mathsf{PTL}}(\Lambda)$.

Theorem (Transference of SAT)

The SAT problem for PTL is PSPACE and obtaining a witness (model) is EXPSPACE.

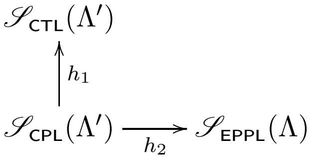
Definition (CTPL)

Consider the system

$$\mathscr{S}_{\mathsf{CTPL}}(\Lambda) = \langle \mathcal{L}_{\mathsf{CTPL}}(\Lambda), \mathcal{M}_{\mathsf{CTPL}}, \Vdash_{\mathsf{CTPL}} \rangle,$$

- $\mathcal{L}_{\mathsf{CTPL}}(\Lambda)$ is
  - $\varphi := \beta \;[\!]\; (\neg\varphi) \;[\!]\; (\varphi \Rightarrow \varphi) \;[\!]\; (\mathsf{AX}\varphi) \;[\!]\; (\mathsf{A}(\varphi \mathsf{U} \varphi)) \;[\!]\; (\mathsf{AG}\varphi)$

  with $\beta \in \mathcal{L}_{\mathsf{EPPL}}(\Lambda)$;
- $\mathcal{M}_{\mathsf{CTPL}}$ is the class of tuples $m = \langle S, R, V : S \to \mathcal{M}_{\mathsf{EPPL}} \rangle$, where $\langle S, R \rangle$ is a Kripke frame;
- $\Vdash_{\mathsf{CTPL}}$ is defined by
  - $m, s \Vdash_{\mathsf{CTPL}} \beta$ iff $V(s) \Vdash_{\mathsf{EPPL}} \beta$;
  - ... *(as in CTL)*



Proposition (Equivalence)

$\mathscr{S}_{(h_1 \Rightarrow h_2)} \approxeq_S \mathscr{S}_{\mathit{CTPL}}(\Lambda)$.

Theorem (Transference of weak completeness)

The axiomatization $\mathbb{AX}_{\mathit{CTL}} + h_1(h_2^{-1}(\mathbb{AX}_{\mathit{EPPL}}))$ *is weakly complete and sound for* $\mathscr{S}_{\mathit{CTPL}}(\Lambda)$.

Theorem (SAT complexity)

The satisfaction problem for CTPL is 2EXPTIME.

Future Work:

- study **exogenous combination** as a generic tool to analyze heterogeneous systems (cyber-physical systems):
  - automatic methods to combine systems;
  - generalize Nelson-Oppen combination procedure;
  - reuse of SAT and model-checking procedures (tools).

- investigate Craig's **interpolation** on probabilistic logics;

- developed **non-Hilbert calculus** for probabilistic logics (to applied in verification by rewriting)