On Quantitative Logics and Weighted Automata

Paul Gastin

LSV, ENS Cachan, INRIA, CNRS, FRANCE

Leipzig, Wata 2010, May 3-7

Slides at http://www.lsv.ens-cachan.fr/~gastin/Talks/

Motivations

Analysis of quantitative systems

- Probabilistic Systems
- Minimization of costs
- Maximization of rewards
- Computation of reliability
- Optimization of energy consumption
- ...

Models (no time)

- Probabilistic automata (generative, reactive)
- Transition systems with costs or rewards
- ...

All are special cases of Weighted Automata.

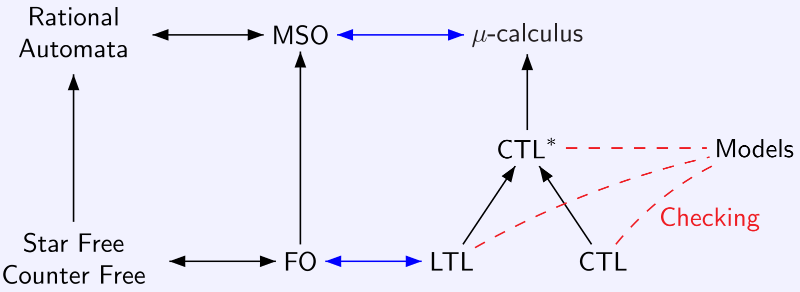
Motivations

Specifications should also be quantitative

Aim: introduce weighted MSO logic (wMSO) and study its properties

- Satisfiability
- Model Checking
- Expressivity

Qualitative (Boolean) Picture



We should extend this picture to the quantitative setting.

Plan

1. MSO Logic

2. Weighted MSO Logic

3. Weighted MSO versus Weighted Automata

4. Weighted CTL* and PCTL*

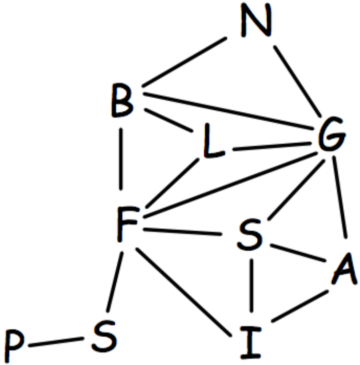
5. Conclusion and Open Problems

Structures

A structure s consists of

- $\mathrm{pos}(s)$ set of positions/vertices/nodes
- $\lambda_s : \mathrm{pos}(s) \to \Sigma$ labeling of positions
- Relations depending on the structure:
  - E edges in graphs
  - $<$ linear order for words
    $\lessdot$ successor relation for words
  - $\lessdot_1$ and $\lessdot_2$ two successor relations for binary trees
    $\leq = (\lessdot_1 \cup \lessdot_2)^*$ associated partial order

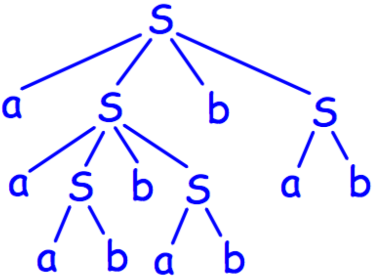
Graphs



Words

$w = a\ b\ a\ c\ a\ b\ c\ a\ c\ b$

Trees



MSO Logic

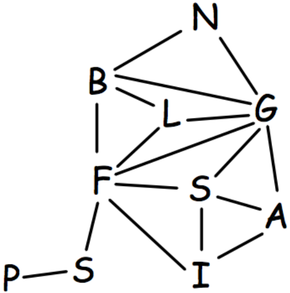
Ingredients

- Variables:
  - Positions (first-order) : $x, y, x_1, x_2, \ldots$
  - Sets of positions (predicates) : $X, Y, X_1, X_2, \ldots$
  - $x \in X$: atomic formula
- Quantifications: $\exists x$, $\exists X$, $\forall x$, $\forall X$
- Boolean connectives: $\vee$, $\wedge$, $\neg$, $\rightarrow$, $\longleftrightarrow$
- Labels: $P_a(x)$ for $a \in \Sigma$ (constant predicates)
- Relations of the structures: $x \mathrel{E} y$, $x < y$, $x \lessdot_1 y$, $\ldots$

MSO over Graphs

Definition: Syntax

$$\varphi ::= 0 \mid 1 \mid P_a(x) \mid x \in X \mid x\ E\ y \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi$$
$$\mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists X\, \varphi \mid \forall X\, \varphi$$



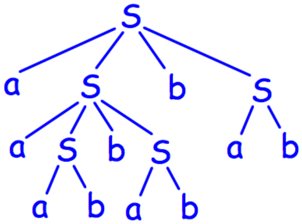
Example: 3-coloring

$$\exists R, B, G \qquad \forall x\,(x \in R \vee x \in B \vee x \in G)$$
$$\wedge\, \forall x, y\,(x\ E\ y \to \neg(x, y \in R \vee x, y \in B \vee x, y \in G))$$

MSO over Trees

Definition: Syntax

$$\varphi ::= 0 \mid 1 \mid P_a(x) \mid x \in X \mid x \le y \mid x \lessdot_i y \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi$$
$$\mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists X\, \varphi \mid \forall X\, \varphi$$



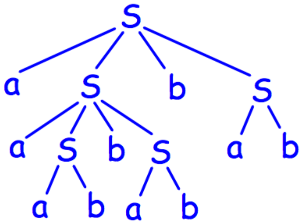
Example: Parse tree for $S \to aSbS + ab$

$$\begin{aligned}
\forall x \quad & \mathrm{root}(x) \longrightarrow P_S(x) \\
& \wedge\, \mathrm{leaf}(x) \longleftrightarrow (P_a(x) \vee P_b(x)) \\
& \wedge\, P_S(x) \longrightarrow \exists x_1, x_2\, (x \lessdot_1 x_1 \wedge P_a(x_1) \wedge x \lessdot_2 x_2 \wedge P_b(x_2)) \\
& \qquad\qquad \vee \exists x_1, x_2, x_3, x_4\, (x \lessdot_1 x_1 \wedge P_a(x_1) \wedge x \lessdot_2 x_2 \wedge P_S(x_2) \\
& \qquad\qquad\qquad \wedge\, x \lessdot_3 x_3 \wedge P_b(x_3) \wedge x \lessdot_4 x_4 \wedge P_S(x_4))
\end{aligned}$$

MSO over Ordered Unranked Trees

Definition: Syntax

$$\varphi ::= 0 \mid 1 \mid P_a(x) \mid x \in X \mid x \leq y \mid x <_s y \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi$$
$$\mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists X\, \varphi \mid \forall X\, \varphi$$



Example: Parse tree for $S \to aSbS + ab$

$$\begin{aligned}
\forall x \quad & \mathrm{root}(x) \longrightarrow P_S(x) \\
& \wedge\, \mathrm{leaf}(x) \longleftrightarrow (P_a(x) \vee P_b(x)) \\
& \wedge\, P_S(x) \longrightarrow \exists x_1, x_2\, (x \lessdot_1 x_1 \wedge P_a(x_1) \wedge x_1 \lessdot_s x_2 \wedge P_b(x_2)) \\
& \qquad \vee \exists x_1, x_2, x_3, x_4\, (x \lessdot_1 x_1 \wedge P_a(x_1) \wedge x_1 \lessdot_s x_2 \wedge P_S(x_2) \\
& \qquad\qquad \wedge\, x_2 \lessdot_s x_3 \wedge P_b(x_3) \wedge x_3 \lessdot_s x_4 \wedge P_S(x_4))
\end{aligned}$$

Some Questions about MSO Logic

Problems

- Expressivity
  Compare with other formalisms
- Satisfiability
  Given $\varphi \in \mathrm{MSO}(\Sigma, \lessdot_1, \lessdot_2)$ does there exist a binary tree t such that $t \models \varphi$?
- Model checking
  Given $\varphi \in \mathrm{MSO}(\Sigma, <)$ and a model M, does $M \models \varphi$?
  i.e., $w \models \varphi$ for all $w \in \mathcal{L}(M)$
- Complexity of the above decision problems

Solution: MSO = Automata

- Finite words: Elgot'61, Trakhtenbrot'61
- Infinite words: Büchi'60
- Infinite trees: Rabin'69

Effective translations between MSO and Automata.
Closure and decision properties of automata.

Free variables and assignments

Effective translation

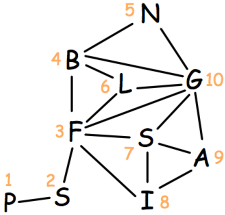
$$\varphi \in \mathsf{MSO}(\Sigma, \ldots) \qquad \longleftrightarrow \qquad \mathcal{A}$$

with $s \models \varphi$ iff $s \in \mathcal{L}(\mathcal{A})$ for all structures s.

The translation from MSO to Automata is by structural induction.
We need to deal with free variables.

Example: 3-coloring

$$\varphi(R, B, G) = \forall x\,(x \in R \vee x \in B \vee x \in G) \\ \wedge \forall x, y\,(x\;E\;y \rightarrow \neg(x, y \in R \vee x, y \in B \vee x, y \in G))$$



Choose sets $\sigma(R)$, $\sigma(B)$, $\sigma(G)$ of positions and evaluate φ with this assignment.

Assignments

Definition: Assignments

Let s be a structure.
Let $\mathcal{V}$ be a finite set of first-order and second-order variables.

A $(\mathcal{V}, s)$-assignment σ is a function mapping

- first-order variables in $\mathcal{V}$ to elements in $\mathrm{pos}(s)$ and
- second-order variables in $\mathcal{V}$ to subsets of $\mathrm{pos}(s)$.

For $i \in \mathrm{pos}(s)$, let $\sigma[x \mapsto i]$ be the $(\mathcal{V} \cup \{x\}, s)$ assignment mapping x to i and which coincides with σ otherwise.

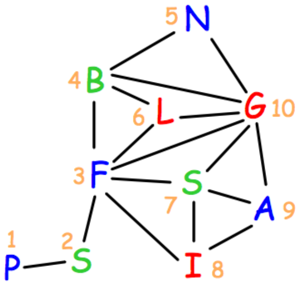
For $I \subseteq \mathrm{pos}(s)$, we define $\sigma[X \mapsto I]$ similarly.

Assignments

Example: 3-coloring

$$\exists R, B, G \qquad \forall x\,(x \in R \vee x \in B \vee x \in G)$$
$$\wedge\, \forall x, y\,(x\ E\ y \rightarrow \neg(x, y \in R \vee x, y \in B \vee x, y \in G))$$

$$\sigma = [B \mapsto \{1, 3, 5, 9\}, G \mapsto \{2, 4, 7\}, R \mapsto \{6, 8, 10\}, x \mapsto 6, y \mapsto 10]$$
$$g, \sigma \not\models (x\ E\ y \rightarrow \neg(x, y \in R \vee x, y \in B \vee x, y \in G))$$



Assignments as Extended labeling

Example:

$$(abaa, x \mapsto 2, X \mapsto \{1,3\}) \text{ is encoded by } \begin{array}{cccc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ a & b & a & a \end{array}$$

Definition: Encoding

Let $\mathcal{V}$ be a finite set of first-order and second-order variables.

Define $\Sigma_{\mathcal{V}} = \Sigma \times \{0,1\}^{\mathcal{V}}$.

Let s be a structure over Σ (i.e., $\lambda_s : \mathrm{pos}(s) \to \Sigma$) and σ be a $(\mathcal{V}, s)$-assignment.

(s, σ) is encoded as the structure s' over $\Sigma_{\mathcal{V}}$ with the extended labeling $\lambda_{s'}$ defined for $i \in \mathrm{pos}(s)$ by $\lambda_{s'}(i) = (\lambda_s(i), \tau)$ with

$$\begin{array}{rcl} \tau(x) = 1 & \text{iff} & i = \sigma(x) \\ \tau(X) = 1 & \text{iff} & i \in \sigma(X) \end{array}$$

A structure s' over $\Sigma_{\mathcal{V}}$ will be written as a pair (s, σ).
Note that σ is not necessarily a valid $(\mathcal{V}, s)$-assignment.

Semantics

3 equivalent definitions

Let $\varphi \in \mathsf{MSO}(\Sigma, \dots)$ be a formula.

Let $\mathcal{V} \supseteq \mathrm{Free}(\varphi)$ be a finite set of first-order and second-order variables.

Let (s, σ) be a structure over $\Sigma_{\mathcal{V}}$.

1. Classical: if σ is a valid $(\mathcal{V}, s)$-assignment
$$s, \sigma \models \varphi$$
2. Language:
$$\mathcal{L}_{\mathcal{V}}(\varphi) = \{(s, \sigma) \mid \sigma \text{ is a valid } (\mathcal{V}, s)\text{-assignment and } s, \sigma \models \varphi\}$$
3. Characteristic function of $\mathcal{L}_{\mathcal{V}}(\varphi)$:
$$[\![\varphi]\!]_{\mathcal{V}}(s, \sigma) = \begin{cases} 1 & \text{if } \sigma \text{ is a valid } (\mathcal{V}, s)\text{-assignment and } s, \sigma \models \varphi \\ 0 & \text{otherwise.} \end{cases}$$

Plan

MSO Logic

2 Weighted MSO Logic

Weighted MSO versus Weighted Automata

Weighted CTL^* **and** PCTL^*

Conclusion and Open Problems

Weighted MSO Logic (wMSO)

Quantitative semantics

Let $\mathcal{A}$ be a wA and s a struture.
The semantics $[\![\mathcal{A}]\!](s)$ is a value from some semiring: $\mathbb{B}$, $\mathbb{N}$, $\mathbb{R}$, ...

Let φ be a wMSO formula, s a structure and σ a $(\mathcal{V}, s)$- assignment.
The semantics $[\![\varphi]\!]_{\mathcal{V}}(s, \sigma)$ should also be a value from the semiring.

History of equivalences between (restricted) wMSO and wA

Generalizations of Elgot's, Trakhtenbrot's, Büchi's, Rabin's theorems.

- Finite words — Droste & Gastin, ICALP'95
- Trees — Droste & Vogler, TCS'06
- Infinite words — Droste & Rahonis, CIAA'07
- Pictures — Fischtner, STACS'06
- Traces — Meinecke, CSR'06
- Distributed systems — Bollig & Meinecke, LFCS'07
- ...

Semirings

Definition: Semiring

- $\mathbb{K} = (K, \oplus, \otimes, \mathbf{0}, \mathbf{1})$
- $(K, \oplus, \mathbf{0})$ is a commutative monoid,
- $(K, \otimes, \mathbf{1})$ is a monoid,
- multiplication distributes over addition, and $\mathbf{0}$ is absorbant.

Examples:

- Boolean: $\mathbb{B} = (\{\mathbf{0}, \mathbf{1}\}, \vee, \wedge, \mathbf{0}, \mathbf{1})$
- Natural: $(\mathbb{N}, +, \cdot, 0, 1)$
- Tropical: $(\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$
- Probabilistic: $\mathbb{P}\text{rob} = (\mathbb{R}_{\geq 0}, +, \cdot, 0, 1)$
- Reliability: $([0, 1], \max, \cdot, 0, 1)$

Syntax of Weighted MSO Logics (wMSO)

Definition: Syntax for words $\mathrm{wMSO}(\mathbb{K}, \Sigma, \leq)$

$$\varphi ::= k \mid x \leq y \mid P_a(x) \mid x \in X \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \\ \mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists X\, \varphi \mid \forall X\, \varphi$$

with $k \in \mathbb{K}$ and $a \in \Sigma$

Definition: Syntax for (ranked) trees $\mathrm{wMSO}(\mathbb{K}, \Sigma, \leq, \lessdot_1, \lessdot_2)$

$$\varphi ::= k \mid x \leq y \mid x \lessdot_i y \mid P_a(x) \mid x \in X \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \\ \mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists X\, \varphi \mid \forall X\, \varphi$$

with $k \in \mathbb{K}$ and $a \in \Sigma$

Definition: Syntax for (ordered) unranked trees $\mathrm{wMSO}(\mathbb{K}, \Sigma, \leq, <_s)$

$$\varphi ::= k \mid x \leq y \mid x <_s y \mid P_a(x) \mid x \in X \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \\ \mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists X\, \varphi \mid \forall X\, \varphi$$

with $k \in \mathbb{K}$ and $a \in \Sigma$

Semantics of wMSO

Constants $k \in \mathbb{K}$

$$[\![k]\!]_{\mathcal{V}}(s,\sigma) = \begin{cases} k & \text{if } \sigma \text{ is a valid } (\mathcal{V}, s)\text{-assignment} \\ \mathbf{0} & \text{otherwise} \end{cases}$$

Atomic formulas: we use the boolean semantics

$$[\![P_a(x)]\!]_{\mathcal{V}}(s,\sigma) = \begin{cases} \mathbf{1} & \text{if } \lambda_s(\sigma(x)) = a \\ \mathbf{0} & \text{otherwise} \end{cases} \qquad [\![x \in X]\!]_{\mathcal{V}}(s,\sigma) = \begin{cases} \mathbf{1} & \text{if } \sigma(x) \in \sigma(X) \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$[\![x \leq y]\!]_{\mathcal{V}}(s,\sigma) = \begin{cases} \mathbf{1} & \text{if } \sigma(x) \leq \sigma(y) \\ \mathbf{0} & \text{otherwise} \end{cases} \qquad [\![x \lessdot_i y]\!]_{\mathcal{V}}(s,\sigma) = \begin{cases} \mathbf{1} & \text{if } \sigma(x) \lessdot_i \sigma(y) \\ \mathbf{0} & \text{otherwise} \end{cases}$$

Negation: extends the boolean semantics

$$[\![\neg\varphi]\!]_{\mathcal{V}}(s,\sigma) = \begin{cases} \mathbf{1} & \text{if } [\![\varphi]\!]_{\mathcal{V}}(s,\sigma) = \mathbf{0} \text{ and } \sigma \text{ is a valid } (\mathcal{V}, s)\text{-assignment} \\ \mathbf{0} & \text{otherwise} \end{cases}$$

Semantics of wMSO

Disjunction and existential quantifications are sums

$$[\![\varphi_1 \vee \varphi_2]\!]_{\mathcal{V}}(s,\sigma) = [\![\varphi_1]\!]_{\mathcal{V}}(s,\sigma) \oplus [\![\varphi_2]\!]_{\mathcal{V}}(s,\sigma)$$

$$[\![\exists x\, \varphi]\!]_{\mathcal{V}}(s,\sigma) = \bigoplus_{i \in \mathrm{dom}(s)} [\![\varphi]\!]_{\mathcal{V}\cup\{x\}}(s,\sigma[x \to i])$$

$$[\![\exists X\, \varphi]\!]_{\mathcal{V}}(s,\sigma) = \bigoplus_{I \subseteq \mathrm{dom}(s)} [\![\varphi]\!]_{\mathcal{V}\cup\{X\}}(s,\sigma[X \to I])$$

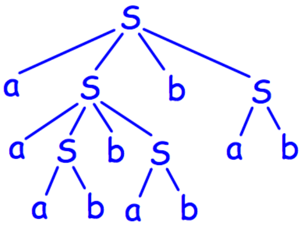
Conjunction and universal quantifications are products

$$[\![\varphi_1 \wedge \varphi_2]\!]_{\mathcal{V}}(s,\sigma) = [\![\varphi_1]\!]_{\mathcal{V}}(s,\sigma) \otimes [\![\varphi_2]\!]_{\mathcal{V}}(s,\sigma)$$

$$[\![\forall x\, \varphi]\!]_{\mathcal{V}}(s,\sigma) = \bigotimes_{i \in \mathrm{dom}(s)} [\![\varphi]\!]_{\mathcal{V}\cup\{x\}}(s,\sigma[x \to i])$$

$$[\![\forall X\, \varphi]\!]_{\mathcal{V}}(s,\sigma) = \bigotimes_{I \subseteq \mathrm{dom}(s)} [\![\varphi]\!]_{\mathcal{V}\cup\{X\}}(s,\sigma[X \to I])$$

Examples



Compute overt $\mathbb{N}$ the semantics of

- $[\![\exists x\, P_S(x)]\!](t)$
- $[\![\exists x\, (P_a(x) \vee (P_b(x) \wedge 2))]\!](t)$
- $[\![\exists x\, (P_S(x) \wedge \neg \exists y\, (x < y \wedge P_S(y)))]\!](t)$
- $[\![\exists x\, (P_S(x) \wedge \exists y\, (x < y \wedge P_S(y)))]\!](t)$

Can we compute

- number of nodes for the rule $S \to aSbS$
- the value $2^{|t|_a} \cdot 3^{|t|_b}$
- the number of leaves of odd depth

Boolean fragment of wMSO

Definition: syntax of bMSO$(\Sigma, \leq, <_i)$

$$\varphi ::= \mathbf{0} \mid \mathbf{1} \mid x \leq y \mid x <_i y \mid P_a(x) \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

with $a \in \Sigma$.

Remark: Boolean and Quantitative semantics coincide on bMSO

$$[\![\neg\varphi]\!]_{\mathcal{V}}(s,\sigma) = \begin{cases} \mathbf{1} & \text{if } [\![\varphi]\!]_{\mathcal{V}}(s,\sigma) = \mathbf{0} \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$[\![\varphi_1 \wedge \varphi_2]\!]_{\mathcal{V}}(s,\sigma) = [\![\varphi_1]\!]_{\mathcal{V}}(s,\sigma) \otimes [\![\varphi_2]\!]_{\mathcal{V}}(s,\sigma)$$

$$[\![\forall x\, \varphi]\!]_{\mathcal{V}}(s,\sigma) = \bigotimes_{i \in \mathrm{dom}(s)} [\![\varphi]\!]_{\mathcal{V}\cup\{x\}}(s, \sigma[x \to i])$$

$$[\![\forall X\, \varphi]\!]_{\mathcal{V}}(s,\sigma) = \bigotimes_{I \subseteq \mathrm{dom}(s)} [\![\varphi]\!]_{\mathcal{V}\cup\{X\}}(s, \sigma[X \to I])$$

Boolean fragment of wMSO

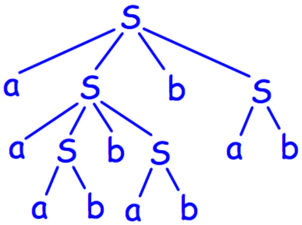
Definition: Macros for disjunction and existential quantifications

$$\varphi_1 \underline{\vee} \varphi_2 \stackrel{\text{def}}{=} \neg(\neg\varphi_1 \wedge \neg\varphi_2)$$

$$\underline{\exists}\, x\, \varphi \stackrel{\text{def}}{=} \neg\forall x\, \neg\varphi$$

$$\underline{\exists}\, X\, \varphi \stackrel{\text{def}}{=} \neg\forall X\, \neg\varphi$$

Hence, we can easily define boolean formulas for all MSO properties.



Number of nodes for the rule $S \to aSbS$

$$\exists x\, (P_S(x) \wedge \underline{\exists}\, y\, (x < y \wedge P_S(y)))$$

Useful Macro

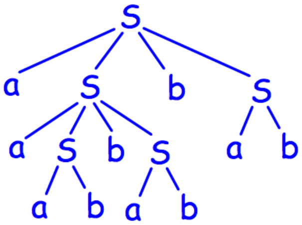
Definition: Useful macro

$$\varphi_1 \xrightarrow{+} \varphi_2 \stackrel{\text{def}}{=} \neg\varphi_1 \vee (\varphi_1 \wedge \varphi_2)$$

If φ_1 is boolean (i.e., if $[\![\varphi_1]\!]$ takes values in $\{0,1\}$), we have

$$[\![\varphi_1 \xrightarrow{+} \varphi_2]\!]_{\mathcal{V}}(s,\sigma) = \begin{cases} [\![\varphi_2]\!]_{\mathcal{V}}(s,\sigma) & \text{if } [\![\varphi_1]\!]_{\mathcal{V}}(s,\sigma) = \mathbf{1} \\ \mathbf{1} & \text{otherwise.} \end{cases}$$

If φ_1, φ_2 are boolean, then $\varphi_1 \xrightarrow{+} \varphi_2$ is the usual boolean implication.



$$\forall x\,((P_a(x) \xrightarrow{+} 2) \wedge (P_b(x) \xrightarrow{+} 3))$$

Step formulas

Definition: Syntax of bMSO-step

$$\varphi ::= k \mid \alpha \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi$$

with $k \in \mathbb{K}$ and $\alpha \in$ bMSO

Proposition: Normal form for bMSO-step

For every bMSO-step formula φ, one can construct an equivalent formula

$$\psi = \bigvee_{\text{finite}} k_n \wedge \varphi_n$$

with $\varphi_n \in$ bMSO and $k_n \in K$.

Proof:

Let $\alpha_1, \ldots, \alpha_p$ be the bMSO formulas in φ.

For $I \subseteq \{1, \ldots, p\}$, define $\varphi_I = \varphi[\alpha_i/1$ if $i \in I,\ 0$ otherwise$]$.

Then, $[\![\varphi_I]\!] = k_I$ is constant.

Define $\psi_I = \bigwedge_{i \in I} \alpha_i \wedge \bigwedge_{i \notin I} \neg\alpha_i$, a bMSO formula.

Then, φ is equivalent to $\psi = \bigvee_{I \subseteq \{1,\ldots,p\}} k_I \wedge \psi_I$.

First-order fragments

Definition: Weighted first-order (wFO)

$$\varphi ::= k \mid x \leq y \mid x <_i y \mid P_a(x) \mid x \in X \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi$$

with $k \in \mathbb{K}$ and $a \in \Sigma$.

Definition: Boolean first-order (bFO)

$$\varphi ::= 0 \mid 1 \mid x \leq y \mid x <_i y \mid P_a(x) \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi$$

with $a \in \Sigma$.

We can use the macros $\underline{\vee}$, $\xrightarrow{+}$ and $\underline{\exists}\, x$.

Definition: bFO-step

$$\varphi ::= k \mid \alpha \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi$$

with $k \in \mathbb{K}$ and $\alpha \in$ bFO

Plan

MSO Logic

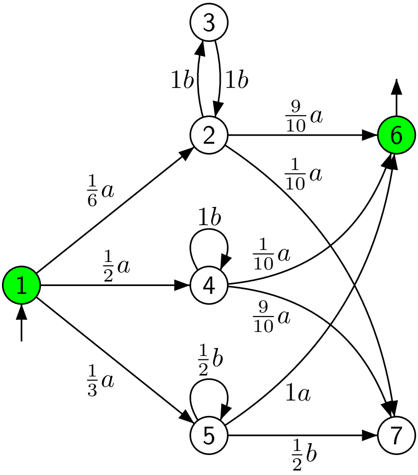
Weighted MSO Logic

3. Weighted MSO versus Weighted Automata
   - Weighted Automata
   - From Weighted MSO to Weighted Automata
   - From Weighted Automata to Weighted MSO
   - Transitive closure

Weighted CTL* and PCTL*

Conclusion and Open Problems

Weighted Automata by Example



Several paths for $v = ab^n a$:

$\pi_1 = 1 \xrightarrow{a} 4 \xrightarrow{b} 4 \cdots 4 \xrightarrow{b} 4 \xrightarrow{a} 6$
$\text{weight}(\pi_1) = \frac{1}{2} \cdot 1^n \cdot \frac{1}{10} = \frac{1}{20}$

$\pi_2 = 1 \xrightarrow{a} 5 \xrightarrow{b} 5 \cdots 5 \xrightarrow{b} 5 \xrightarrow{a} 6$
$\text{weight}(\pi_2) = \frac{1}{3} \cdot (\frac{1}{2})^n \cdot 1 = \frac{1}{3 \cdot 2^n}$

If n is even:
$\pi_3 = 1 \xrightarrow{a} 2 \xrightarrow{b} 3 \xrightarrow{b} 2 \cdots 2 \xrightarrow{a} 6$
$\text{weight}(\pi_3) = \frac{1}{6} \cdot 1^n \cdot \frac{9}{10} = \frac{3}{20}$

Probabilistic: $\mathbb{P}\text{rob} = (\mathbb{R}_{\geq 0}, +, \cdot, 0, 1)$

$$[\![\mathcal{A}]\!](v) = \begin{cases} \frac{1}{20} + \frac{1}{3 \cdot 2^n} & \text{if } n \text{ is odd} \\ \frac{1}{5} + \frac{1}{3 \cdot 2^n} & \text{if } n \text{ is even} \end{cases}$$

Reliability: $([0,1], \max, \cdot, 0, 1)$

$$[\![\mathcal{A}]\!](v) = \begin{cases} \max(\frac{1}{20}, \frac{1}{3 \cdot 2^n}) & \text{if } n \text{ is odd} \\ \max(\frac{3}{20}, \frac{1}{3 \cdot 2^n}) & \text{if } n \text{ is even} \end{cases}$$

Weighted Automata formally

Definition: Weighted Automaton

A *weighted automaton* over $\mathbb{K}$ and Σ is a quadruple $\mathcal{A} = (Q, \lambda, \mu, \gamma)$ where

- Q is the nonempty finite set of *states*,
- $\mu : \Sigma \to K^{Q \times Q}$ is the *transition weight function*,
- and $\lambda, \gamma \in K^Q$ provide weights for entering and leaving a state, respectively.

Definition: Semantics $\quad [\![\mathcal{A}]\!] : \Sigma^* \to K$

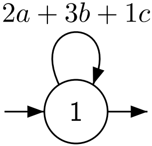
The *weight* of a path $\pi : q_0 \xrightarrow{a_1} q_1 \longrightarrow \ldots \longrightarrow q_{n-1} \xrightarrow{a_n} q_n$ in $\mathcal{A}$ is

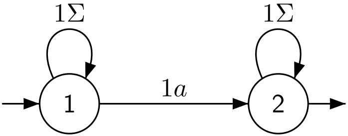
$$\mathrm{weight}(\pi) \stackrel{\mathrm{def}}{=} \mu(a_1)_{q_0,q_1} \cdots \mu(a_n)_{q_{n-1},q_n}.$$

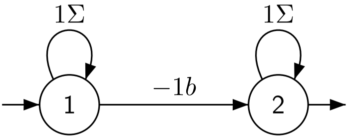
The semantics of $\mathcal{A}$ is defined by

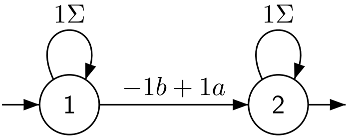
$$[\![\mathcal{A}]\!](w) = \sum_{\text{path } \pi = p \xrightarrow{w} q} \lambda(p) \cdot \mathrm{weight}(\pi) \cdot \gamma(q)$$

Weighted Automata: Examples









Weighted Automata: semantics revisited

Definition:

The transition weight function $\mu : \Sigma \to K^{Q \times Q}$ is extended to a morphism

$$\mu : \Sigma^* \to K^{Q \times Q}$$

Proposition: Semantics

For $w \in \Sigma^*$ and $p, q \in Q$, we have $\mu(w)_{p,q} = \sum_{\text{path } \pi = p \xrightarrow{w} q} \mathrm{weight}(\pi)$.

Hence, $[\![\mathcal{A}]\!](w) = \sum_{\text{path } \pi = p \xrightarrow{w} q} \lambda(p) \cdot \mathrm{weight}(\pi) \cdot \gamma(q) = \lambda \cdot \mu(w) \cdot \gamma$.

From wMSO to wA: constants

Valid assignments and constant formulas

Note that $[\![\varphi]\!]_{\mathcal{V}}(s,\sigma) = \mathbf{0}$ if σ is not a valid $(\mathcal{V}, s)$-assignment.

Consider a deterministic and complete finite automaton (over words, trees, ...) accepting the pairs (s,σ) over $\Sigma_{\mathcal{V}}$ such that σ is a valid $(\mathcal{V}, s)$-assignment.

Put weight $\mathbf{1}$ on all transitions as well as on final states.

Put weight k on the initial state.

We get an automaton $\mathcal{A}(k,\mathcal{V})$ is equivalent to the constant formula k:

$$[\![\mathcal{A}(k,\mathcal{V})]\!] = [\![k]\!]_{\mathcal{V}}.$$

From wMSO to wA: Boolean Connectives

- Disjunction is sum: Take disjoint union of $\mathcal{A}_1$ and $\mathcal{A}_2$.
- Conjunction is product: Take synchronized product of $\mathcal{A}_1$ and $\mathcal{A}_2$.

  If $p_1 \xrightarrow{k_1 a} q_1$ in $\mathcal{A}_1$ and $p_2 \xrightarrow{k_2 a} q_2$ in $\mathcal{A}_2$ then

  $$(p_1, p_2) \xrightarrow{(k_1 k_2) a} (q_1, q_2) \qquad \text{in } \mathcal{A}_1 \otimes \mathcal{A}_2$$

- Negation: restricted to bMSO-step formulas

Existential Quantifications

$$[\![\exists X\,\varphi]\!]_{\mathcal{V}}(s,\sigma) = \bigoplus_{I\subseteq \mathrm{dom}(s)} [\![\varphi]\!]_{\mathcal{V}\cup\{X\}}(s,\sigma[X\to I])$$

Proof:

Let $\mathcal{A} = \mathcal{A}(\varphi, \mathcal{V}\cup\{X\}) = (Q,\lambda,\mu,\gamma)$

Then, $\mathcal{A}' = \mathcal{A}(\exists X\,\varphi, \mathcal{V}) = (Q,\lambda,\mu',\gamma)$ with

$$\mu'(a,\tau) = \mu(a,\tau[X\mapsto 0]) \oplus \mu(a,\tau[X\mapsto 1])$$

Fix $w = a_1\cdots a_n$ and a path $\pi = p_0, p_1, \ldots, p_n \in Q^*$. For each $i \in \mathrm{pos}(w)$,

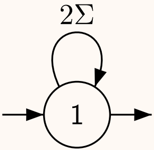
$p_{i-1} \xrightarrow{k_i^0(a_i,0)} p_i$ and $p_{i-1} \xrightarrow{k_i^1(a_i,1)} p_i$ are grouped in $p_{i-1} \xrightarrow{(k_i^0\oplus k_i^1)a_i} p_i$.

Then,

$$\begin{aligned}\mathrm{weight}_{\mathcal{A}'}(\pi,w) &= (k_1^0\oplus k_1^1)(k_2^0\oplus k_2^1)\cdots(k_n^0\oplus k_n^1)\\ &= \bigoplus_{I\subseteq \mathrm{pos}(w)} \mathrm{weight}_{\mathcal{A}}(\pi,w,[X\mapsto I])\end{aligned}$$

Universal Quantifications

Example: $[\![\forall x\, 2]\!]$ is recognizable



We have $[\![\forall x\, 2]\!](w) = \prod_{1 \leq i \leq |w|} [\![2]\!](w, x \mapsto i) = 2^{|w|}$.

Example: $[\![\forall y \forall x\, 2]\!]$ is not recognizable when $\mathbb{K} = (\mathbb{N}, +, \cdot, 0, 1)$

We have $[\![\forall y \forall x\, 2]\!](w) = \prod_{1 \leq i \leq |w|} [\![\forall x\, 2]\!](w, y \mapsto i) = 2^{|w|^2}$.

Let $\mathcal{A} = (Q, \lambda, \mu, \gamma)$ and $M = \max\{|\lambda_p|, |\gamma_p|, |\mu(a)_{p,q}| \mid p, q \in Q, a \in A\}$.

Then, for any $w \in A^*$, we have $[\![\mathcal{A}]\!](w) \leq |Q|^{|w|+1} \cdot M^{|w|+2} = 2^{\mathcal{O}(|w|)}$.

Therefore, $[\![\forall y \forall x\, 2]\!]$ is not recognizable.

Universal Quantifications

Example: $[\![\forall X\ 2]\!]$ is not recognizable when $\mathbb{K} = (\mathbb{N}, +, \cdot, 0, 1)$

We have $[\![\forall X\ 2]\!](w) = \prod_{I \subseteq \{1,\ldots,|w|\}} [\![2]\!](w, X \mapsto I) = 2^{2^{|w|}}$.

Remark:

The same counter-examples hold for

- the tropical semiring $(\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$
- the arctical semiring $(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$

From wMSO to wA: $\forall x\, \varphi$

Proof: Consider $\forall x\, \varphi$ with $\varphi = \bigvee_{1 \leq j \leq n} k_j \wedge \varphi_j$ a bMSO-step formula.

Let $\mathcal{W} = \mathrm{Free}(\varphi)$ and $\mathcal{V} = \mathrm{Free}(\forall x\, \varphi) = \mathcal{W} \setminus \{x\}$.

We assume that the languages $L_j = L_{\mathcal{W}}(\varphi_j)$ $(1 \leq j \leq n)$ form a partition of $A^*_{\mathcal{W}}$.

Let $(w, \sigma) \in A^*_{\mathcal{V}}$. $\forall i \in \mathrm{pos}(w)$, $\exists! \nu(i) \in \{1, \ldots, n\}$ such that $(w, \sigma[x \to i]) \in L_{\nu(i)}$.

We have $$[\![\forall x\, \varphi]\!](w, \sigma) = \prod_{1 \leq i \leq |w|} [\![\varphi]\!](w, \sigma[x \to i]) = \prod_{1 \leq i \leq |w|} k_{\nu(i)}.$$

The map $\nu : \mathrm{pos}(w) \to \{1, \ldots, n\}$ is encoded with an extended labeling.

Let $\widetilde{A} = A \times \{1, \ldots, n\}$. A word in $(\widetilde{A}_{\mathcal{V}})^*$ will be written (w, ν, σ) where $(w, \sigma) \in A^*_{\mathcal{V}}$ and $\nu \in \{1, \ldots, n\}^{|w|}$ is interpreted as a map $\nu : \mathrm{pos}(w) \to \{1, \ldots, n\}$.

$\widetilde{L} = \{(w, \nu, \sigma) \in (\widetilde{A}_{\mathcal{V}})^* \mid \quad \nu(i) = j \quad \text{iff} \quad (w, \sigma[x \to i]) \in L_j\}$ is recognizable.

Then, $\mathcal{A}(\forall x\, \varphi, \mathcal{V})$ running on (w, σ) guesses ν, checks that its guess is correct with the (deterministic) automaton for $\widetilde{L}$, and computes $\prod_{1 \leq i \leq |w|} k_{\nu(i)}$.

Restricted Weighted MSO Logic

Definition: $\mathcal{L}$-step where $\mathcal{L}$ being bFO or bMSO

$$\alpha ::= k \mid \beta \mid \neg\alpha \mid \alpha \vee \alpha \mid \alpha \wedge \alpha$$

with $k \in \mathbb{K}$ and $\beta \in \mathcal{L}$

Definition: $\exists\forall(\mathcal{L}$-step$)$ where $\mathcal{L}$ being bFO or bMSO

Formulas of the form $\exists X \forall x\, \varphi(x, X)$ where φ is an $\mathcal{L}$-step formula.

Definition: Restricted wMSO (wRMSO)

$$\varphi ::= k \mid x \leq y \mid x <_i y \mid P_a(x) \mid x \in X \mid \neg\alpha \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x\, \varphi \mid \forall x\, \alpha \mid \exists X\, \varphi$$

with $k \in \mathbb{K}$, $a \in \Sigma$ and $\alpha \in$ bMSO-step.

Theorem: Expressivity (DG ICALP'95, BGMZ ICALP'10)

Let f be a series over $\mathbb{K}$. The following are equivalent:

1. f is recognizable by a wA.
2. f is definable in $\exists\forall$(bFO-step).
3. f is definable in wRMSO.

Decidability

Proposition: The translation from wRMSO to wA is effective.

Corollary:

Satisfiability is decidable for wRMSO whenever emptiness is decidable for wA.

Equivalence is decidable for wRMSO whenever equivalence is decidable for wA.

wA to $\exists\forall$(bFO-step)

Proof: Let $\mathcal{A} = (Q, \lambda, \mu, \gamma)$

For $d \geq 1$ and $p, q \in Q$, define the bFO-step formula

$$\psi^d_{p,q}(x) = \bigvee_{v_1 \cdots v_d \in \Sigma^d} \Big(\mu(v_1 \cdots v_d)_{p,q} \wedge \bigwedge_{1 \leq j \leq d} P_{v_j}(x + j - 1) \Big)$$

For every word w and $i \in \mathrm{pos}(w)$ with $i + d - 1 \in \mathrm{pos}(w)$, we have

$$[\![\psi^d_{p,q}(x)]\!](w, i) = \mu(w[i, i + d - 1])_{p,q}$$

Semantics of $\mathcal{A}$ with macro-paths

For $w \in \Sigma^+$ and $k = \left\lfloor \frac{|w|}{d} \right\rfloor$ we have

$$\mu(w)_{p,q} = \sum_{q_1, q_2, \ldots, q_k \in Q} \mu(w[1, d])_{p,q_1} \cdot \mu(w[d + 1, 2d])_{q_1,q_2} \cdots \mu(w[kd + 1, |w|])_{q_k,q}$$

wA to $\exists\forall$(bFO-step)

Proof: $\mathcal{A} = (Q, \lambda, \mu, \gamma)$ is equivalent to $\exists X\, \forall x\, \varphi$

Assume $Q = \{1, \ldots, n\}$ and let $d = 2n + 1$. Assume also that 1 is the initial state.
A set X consisting of positions $x_0 < x_0 + q_0 < x_1 < x_1 + q_1 < x_2 < \ldots$
with $x_\ell = d\ell + 1$ and $1 \le q_\ell \le n$ encodes the macro-path of $\mathcal{A}$

$$q_0 \xrightarrow{w[1,d]} q_1 \xrightarrow{w[d+1,2d]} q_2 \quad \cdots \quad q_k \xrightarrow{w[kd+1,|w|]} ?$$

$$\varphi(x, X) = \quad \Big[\mathsf{last} > n \wedge \{1,2\} \subseteq X \wedge (\varphi_{\mathsf{far}} \vee \varphi_{\mathsf{near}})\Big]$$
$$\vee \Big[\mathsf{last} \le n \wedge X = \emptyset \wedge \big[x = \mathsf{first} \xrightarrow{+} \bigvee_{q \in Q} \psi_{1,q}^{\mathsf{last}}(1) \wedge \gamma(q)\big]\Big]$$

$$\varphi_{\mathsf{far}}(x, X) = (\mathsf{last} \ge x + d + n) \wedge \Big((x \in X \wedge X \cap\,]x, x+n] \ne \emptyset) \xrightarrow{+}$$
$$\bigvee_{p,q \in Q} X \cap\,]x, x+d+n] = \{x+p, x+d, x+d+q\} \wedge \psi_{p,q}^{d}(x)\Big)$$

$$\varphi_{\mathsf{near}}(x, X) = (\mathsf{last} < x + d + n) \wedge \Big((x \in X \wedge X \cap\,]x, x+n] \ne \emptyset) \xrightarrow{+}$$
$$\bigvee_{p,q \in Q} (X \cap\,]x, \mathsf{last}] = \{x+p\}) \wedge \psi_{p,q}^{\mathsf{last}-x+1}(x) \wedge \gamma(q)\Big)$$

Forward Transitive Closure

Definition: Forward Transitive Closure (BGMZ ICALP'10)

Let $\varphi(x, y)$ be a wMSO formula with $x, y \in \mathrm{Free}(\varphi)$. We define

$$\varphi^1(x, y) \stackrel{\text{def}}{=} (x \leq y) \wedge \varphi(x, y),$$

$$\varphi^2(x, y) \stackrel{\text{def}}{=} \exists z (x < z < y \wedge \varphi(x, z) \wedge \varphi(z, y))$$

More generally, for $n \geq 1$ we define

$$\varphi^{n+1}(x, y) \stackrel{\text{def}}{=} \exists z_1, \ldots, z_n \left[x = z_0 < z_1 < \cdots < z_n < z_{n+1} = y \wedge \bigwedge_{0 \leq i \leq n} \varphi(z_i, z_{i+1}) \right]$$

We now define a transitive closure operator $\mathsf{TC}^{<}_{xy}\varphi$ by $\quad \mathsf{TC}^{<}_{xy}\varphi = \bigvee_{n \geq 1} \varphi^n$.

Remark: the infinite disjunction above is well defined.

Forward Transitive Closure

Example:

Let $\psi = \mathsf{TC}^{<}_{x,y}(2 \wedge y = x+2)$ over the semiring $\mathbb{K} = \mathbb{N}$. We have

$$[\![\psi]\!](u, \mathsf{first}, \mathsf{last}) = \begin{cases} 2^n & \text{if } |u| = 2n+1 \text{ with } n \geq 1 \\ 0 & \text{otherwise.} \end{cases}$$

Remark: the support of ψ is not bFO-definable.

Forward Transitive Closure

Example: Modulo can be expressed with $\mathsf{TC}^<$

For $1 \leq m \leq \ell$: $$x \equiv_\ell m \stackrel{\text{def}}{=} x = m \vee \mathsf{TC}^<_{yz}(z = y + \ell)(m, x)$$

Forward Transitive Closure

Example: Computing big values with $\mathsf{TC}^<$

Let $\mathbb{K} = \mathbb{N}$ and $\varphi(x,y) \stackrel{\text{def}}{=} (y = x+1) \wedge \forall z\, 2 \wedge (x = 1 \xrightarrow{+} \forall z\, 2)$.

Then, $[\![\mathsf{TC}^<_{xy}\varphi]\!](w, \mathsf{first}, \mathsf{last}) = 2^{|w|^2}$

Recognizable series are not closed under $\mathsf{TC}^<$.

First-oder and Transitive Closure

Definition: FO+TC$^<$

$$\varphi ::= k \mid P_a(x) \mid x \leq y \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi \mid \mathsf{TC}^<_{xy}\varphi$$

where $k \in \mathbb{K}$, $a \in \Sigma$.

Definition: Bounded Transitive Closure

N-$\mathsf{TC}^<_{xy}\varphi$ is defined as $\mathsf{TC}^<_{xy}\varphi$ but jumps are limited by N.

Definition: FO+BTC$^<$

$$\varphi ::= k \mid P_a(x) \mid x \leq y \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi \mid N\text{-}\mathsf{TC}^<_{xy}\varphi$$

where $k \in \mathbb{K}$, $a \in \Sigma$ and $N > 0$.

Theorem: Bollig, G., Monmege, Zeitoun ICALP'10

FO+BTC$^<$ has the same expressive power as weighted automata with pebbles.

Fragments for Weighted Automata

Definition:

For $\mathcal{L} \subseteq$ bMSO, the fragment $\mathsf{TC}^{<}(\mathcal{L}\text{-step})$ consists of formulas of the form

$$\mathsf{TC}^{<}_{xy}\varphi$$

where $\varphi(x, y)$ is an $\mathcal{L}$-step formula with $\mathrm{Free}(\varphi) = \{x, y\}$.

Theorem: Expressivity of Transitive Closure (BGMZ ICALP'10)

Let f be a series over $\mathbb{K}$. The following are equivalent:

1. f is recognizable by a wA.
2. f is definable in $\mathsf{TC}^{<}((\mathsf{bFO} + \mathsf{mod})\text{-step})$.
3. f is definable in $\mathsf{TC}^{<}(\mathsf{bMSO}\text{-step})$.
4. f is definable in $\exists\forall(\mathsf{bFO}\text{-step})$.
5. f is definable in $\exists\forall(\mathsf{bMSO}\text{-step})$.
6. f is definable in wRMSO.

Expressivity of Transitive Closure

Proof: $\mathsf{TC}^{<}(\text{bMSO-step}) \subseteq \exists\forall(\text{bMSO-step})$

Let $\varphi(y,z) = \bigvee_{i \in I} k_i \wedge \varphi_i(y,z)$ with $\varphi_i \in$ bMSO.

We define a bMSO-step formula $\psi(x,X,y,z)$ such that

$$[\mathsf{TC}^{<}_{yz}(\varphi)](y,z) = \exists X\, \forall x\, \psi(x,X,y,z)$$

The quantification $\exists X$ guesses the intermediary positions: $y < y_1 < \cdots < y_n < z$
The quantification $\forall x$ computes the product for this guessed sequence.

Define $\xi(x,X) = \bigvee_{i \in I} k_i \wedge \overline{\exists}\, y(x < y \wedge X \cap \,]x,y] = \{y\} \wedge \varphi_i(x,y))$

We have $[\![\xi]\!](u,j,J) = \begin{cases} [\![\varphi]\!](u,j,\mathrm{next}(j,J)) & \text{if } j < \max(J) \\ \mathbf{0} & \text{otherwise.} \end{cases}$

$$\psi(x,X,y,z) = (y = z \wedge X = \emptyset \wedge (x = z \xrightarrow{+} \varphi(y,z))) \vee$$
$$(y \neq z \wedge \{y,z\} \subseteq X \subseteq [y,z] \wedge ((x \in X \wedge x < z) \xrightarrow{+} \xi(x,X)))$$

Plan

MSO Logic

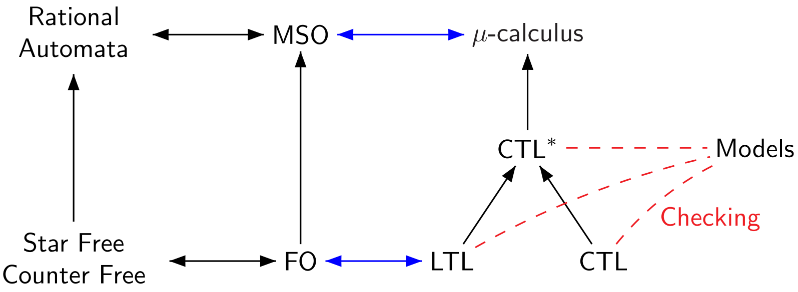
Weighted MSO Logic

Weighted MSO versus Weighted Automata

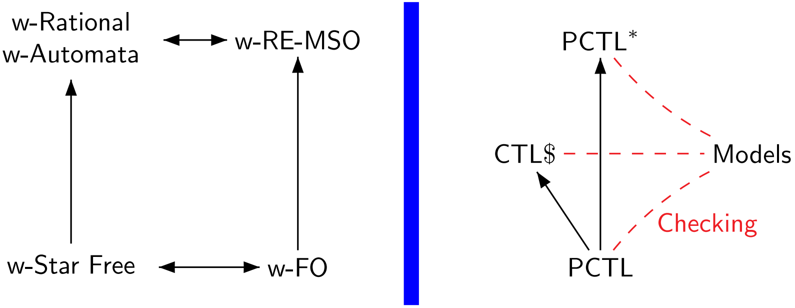
4. Weighted CTL* and PCTL*
   - Probabilistic Automata
   - Extended Weighted MSO Logic
   - Weighted CTL*
   - Weighted CTL* versus Weighted MSO

Conclusion and Open Problems

Qualitative (Boolean) Picture



Quantitative Picture



Our aim is to compare and unify these logics
Bollig & G. DLT'09

Quantitative Logics

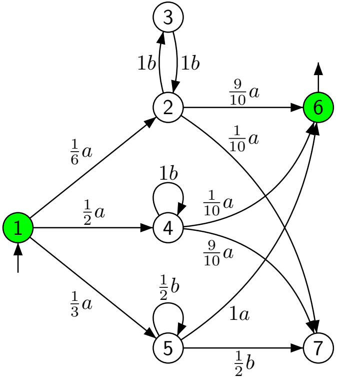
- PCTL: Probabilistic CTL — Hansson & Jonsson, '94
- PCTL*: Probabilistic CTL* — de Alfaro, '98
- CTL$: Valued CTL — Buchholz & Kemper, '03, '09
- wMSO: Weighted MSO — Droste & Gastin, '05, '07, '09

Reactive Probabilistic Finite Automata

Definition: rPFA on $\mathbb{P}\mathrm{rob} = (\mathbb{R}_{\geq 0}, +, \cdot, 0, 1)$

A *reactive probabilistic finite automaton (*rPFA*)* is a weighted automaton $\mathcal{A} = (Q, q_0, \mu, F)$ over $\mathbb{P}\mathrm{rob}$ such that, for all $q \in Q$ and $a \in \Sigma$,

$$\sum_{q' \in Q} \mu(q, a, q') \in \{0, 1\}$$

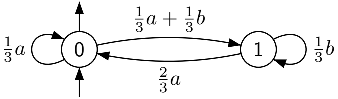


Generative Probabilistic Finite Automata

Definition: GPFA on $\mathbb{P}\text{rob} = (\mathbb{R}_{\geq 0}, +, \cdot, 0, 1)$

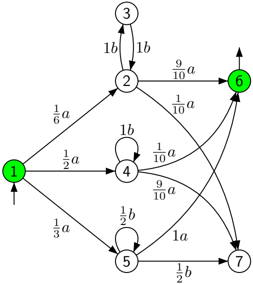
A *generative probabilistic finite automaton (*GPFA*)* is a weighted automaton $\mathcal{A} = (Q, q_0, \mu, F)$ over $\mathbb{P}\text{rob}$ such that, for all $q \in Q$,

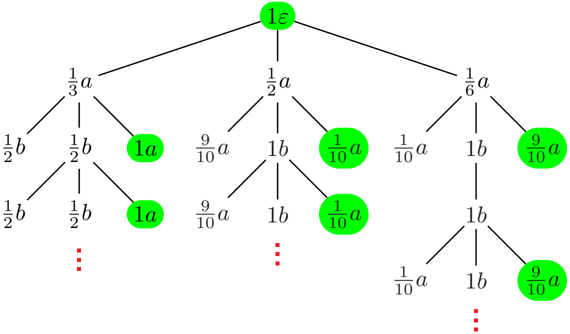
$$\sum_{(a,q') \in \Sigma \times Q} \mu(q, a, q') \in \{0, 1\}$$



Weighted Trees

Semantics of weighted MSO is on weighted trees which are unfoldings of weighted automata





Definition: Weighted Trees: $Trees(D, \mathbb{K}, \Sigma)$

$$\begin{array}{rccc} t: & D^* & \rightharpoonup & K \times \Sigma \\ & u & \to & (\kappa_t(u), \ell_t(u)) \end{array}$$

Extended Weighted MSO

Definition: Syntax of wMSO$(\mathbb{K}, \Sigma, \mathcal{C})$

$$\varphi ::= k \mid \kappa(x) \mid \bowtie(\varphi_1, \ldots, \varphi_{\mathrm{arity}(\bowtie)}) \mid P_a(x) \mid x \in X \mid x \leq y \mid x \lessdot_i y$$
$$\mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x\, \varphi \mid \exists X\, \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

where $k \in K$, $a \in \Sigma$, x, y are first-order variables, X is a set variable and $\bowtie \in \mathcal{C}$.

- $\mathcal{C}$ is a vocabulary of symbols $\bowtie \in \mathcal{C}$ with $\mathrm{arity}(\bowtie) \in \mathbb{N}$.
  - $\mathcal{C} = \{\prec\}$
- Each symbol $\bowtie \in \mathcal{C}$ is given a semantics $[\![\bowtie]\!] : K^{\mathrm{arity}(\bowtie)} \rightharpoonup K$.
  - Ordered semiring: $[\![\prec]\!] : K^2 \to \{\mathbf{0}, \mathbf{1}\}$

Definition: Semantics: $[\![\varphi]\!]_{\mathcal{V}} : \mathit{Trees}(D, \mathbb{K}, \Sigma_{\mathcal{V}}) \rightharpoonup K$

Let $\begin{array}{rccc} t: & D^* & \rightharpoonup & K \times \Sigma \\ & u & \to & (\kappa_t(u), \ell_t(u)) \end{array}$ be a weighted tree and σ a $(\mathcal{V}, t)$-assignment.

$$[\![\kappa(x)]\!]_{\mathcal{V}}(t, \sigma) = \kappa_t(\sigma(x))$$

$$[\![\bowtie(\varphi_1, \ldots, \varphi_r)]\!]_{\mathcal{V}}(t, \sigma) = [\![\bowtie]\!]([\![\varphi_1]\!]_{\mathcal{V}}(t, \sigma), \ldots, [\![\varphi_r]\!]_{\mathcal{V}}(t, \sigma)) \qquad \text{if } \mathrm{arity}(\bowtie) = r$$

Examples

Example:

Let $\varphi_1 = \exists x\,(P_b(x) \wedge (\kappa(x) > 0))$.

$$[\![\varphi_1]\!](t) = \bigoplus_{u \in \mathrm{dom}(t)} (\ell_t(u) = b) \otimes (\kappa_t(u) > 0)$$

is the number of nodes labeled b and having a positive weight.

Example:

Let $\varphi_2 = \forall x\,((P_a(x) \wedge (\kappa(x) > 0)) \xrightarrow{+} \kappa(x))$.

$$[\![\varphi_2]\!](t) = \bigotimes_{u \in \mathrm{dom}(t)} ((P_a(u) \wedge (\kappa_t(u) > 0)) \xrightarrow{+} \kappa_t(u))$$

multiplies the positive values of a-labeled nodes.

Examples

Example:

- Let $\mathrm{path}(x, X)$ be a boolean formula stating that X is a maximal path starting from node x,
- The following boolean formula checks if X satisfies a SU b,
  $\psi(x, X) = \exists z\,(z \in X \wedge x < z \wedge P_b(z) \wedge \forall y\,(x < y < z \xrightarrow{+} P_a(y)))$
- The quantitative formula $\xi(x, X) = \forall y\,((y \in X \wedge x < y) \xrightarrow{+} \kappa(y))$ computes the weight of path X, i.e., the product of weights of nodes in $X \setminus \{x\}$.

Then, we compute the sum of weights of paths from x satisfying a SU b with

$$\exists X\,(\mathrm{path}(x, X) \wedge \psi(x, X) \wedge \xi(x, X))$$

Extended Weighted MSO

Proposition: Satisfiability

The satisfiability problem for $\mathrm{wMSO}(\mathbb{P}\mathrm{rob}, \Sigma, \{<\})$ is undecidable.

Proof:

Let $\mathcal{A} = (Q, q_0, \mu, F)$ be a reactive probabilistic finite automaton over Σ.

By [DG], $\exists \varphi \in \mathrm{wRMSO}(\mathbb{P}\mathrm{rob}, \Sigma)$ such that $[\![\varphi]\!](w) = [\![\mathcal{A}]\!](w)$ for all unweighted words $w \in \Sigma^*$.

Since φ does not use $\kappa(x)$, considering weighted or unweighted words or trees does not make any difference.

Now, for $p \in [0,1]$ and $w \in \Sigma^*$ we have $[\![p < \varphi]\!](w) \neq 0$ iff $[\![\mathcal{A}]\!](w) > p$.

Hence, $p < \varphi$ is satisfiable iff the automaton $\mathcal{A}$ with threshold p accepts a nonempty language.

By , A. Paz (1971) this is undecidable.

Weighted CTL*

Definition: Syntax of wCTL$^*(\mathbb{K}, Prop, \mathcal{C})$

Boolean path formulas: $\psi ::= \varphi \mid \psi \wedge \psi \mid \neg\psi \mid \psi \, \mathsf{SU} \, \psi$

Quantitative state formulas:

$$\varphi ::= k \mid \kappa \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \bowtie(\varphi_1, \ldots, \varphi_{\mathrm{arity}(\bowtie)}) \mid \mu(\psi)$$

where $p \in Prop$, $k \in K$, $\bowtie \in \mathcal{C}$.

Definition: Semantics for Boolean path formulas with $\Sigma = 2^{Prop}$

$$\begin{array}{rccl} t: & D^* & \rightharpoonup & K \times \Sigma \\ & u & \to & (\kappa_t(u), \ell_t(u)) \end{array}$$

weighted tree, w branch of t, u node on w.

$t, w, u \models \varphi$	if $[\![\varphi]\!](t, u) \neq \mathbf{0}$
$t, w, u \models \psi_1 \wedge \psi_2$	if $t, w, u \models \psi_1$ and $t, w, u \models \psi_2$
$t, w, u \models \neg\psi$	if $t, w, u \not\models \psi$
$t, w, u \models \psi_1 \, \mathsf{SU} \, \psi_2$	if $\exists u < v \leq w : (t, w, v \models \psi_2$ and $\forall u < v' < v : t, w, v' \models \psi_1)$

Weighted CTL*

Definition: Syntax of wCTL*$(\mathbb{K}, \mathit{Prop}, \mathcal{C})$

Boolean path formulas: $\psi ::= \varphi \mid \psi \wedge \psi \mid \neg\psi \mid \psi \,\mathsf{SU}\, \psi$

Quantitative state formulas:

$$\varphi ::= k \mid \kappa \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \bowtie(\varphi_1, \ldots, \varphi_{\mathrm{arity}(\bowtie)}) \mid \mu(\psi)$$

where $p \in \mathit{Prop}$, $k \in K$, $\bowtie \in \mathcal{C}$.

Definition: Semantics for quantitative state formulas with $\Sigma = 2^{\mathit{Prop}}$

$$\begin{array}{cccc} t: & D^* & \rightharpoonup & K \times \Sigma \\ & u & \to & (\kappa_t(u), \ell_t(u)) \end{array}$$

weighted tree, u node of t.

$$[\![\kappa]\!](t,u) = \kappa_t(u) \qquad [\![p]\!](t,u) = \begin{cases} \mathbf{1} & \text{if } p \in \ell_t(u) \\ \mathbf{0} & \text{otherwise} \end{cases}$$

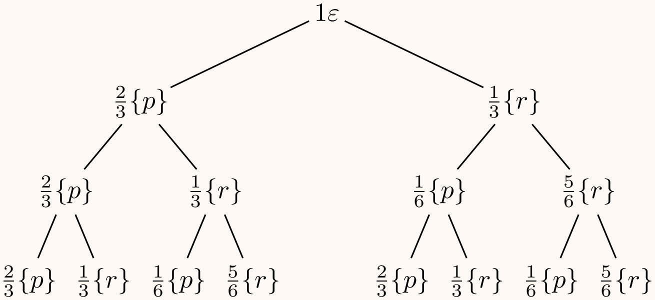
$$[\![\bowtie(\varphi_1, \ldots, \varphi_r)]\!](t,u) = [\![\bowtie]\!]([\![\varphi_1]\!](t,u), \ldots, [\![\varphi_r]\!](t,u)) \qquad \text{if arity}(\bowtie) = r$$

$$[\![\mu(\psi)]\!](t,u) = \bigoplus_{w \in \mathrm{Branches}(t) \mid t,w,u \models \psi} \ \bigotimes_{v \mid u < v \le w} \kappa_t(v)$$

Example for $\mu(\psi)$ on a finite tree

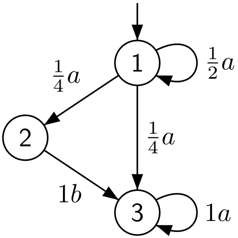
Example:

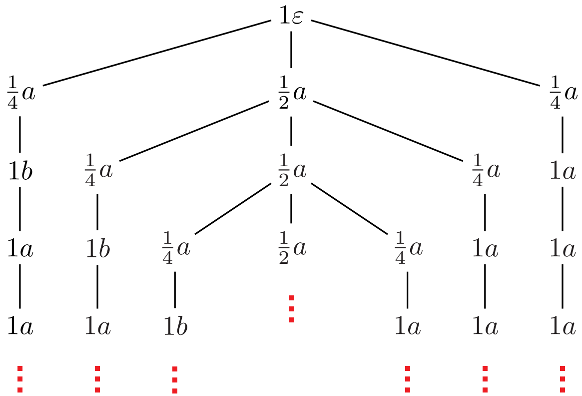
$$[\![\mu(\psi)]\!](t,u) = \bigoplus_{w \in \mathrm{Branches}(t) \,|\, t,w,u \models \psi} \; \bigotimes_{v \,|\, u < v \leq w} \kappa_t(v)$$



$$[\![\mu(p \,\mathsf{SU}\, r)]\!](t) = \frac{2}{3} \cdot \frac{2}{3} \cdot \frac{1}{3} + \frac{2}{3} \cdot \frac{1}{3} \cdot \left(\frac{1}{6} + \frac{5}{6}\right) + \frac{1}{3} \cdot (1) = \frac{19}{27}$$

Unfoldings are infinite (regular) trees





Example:

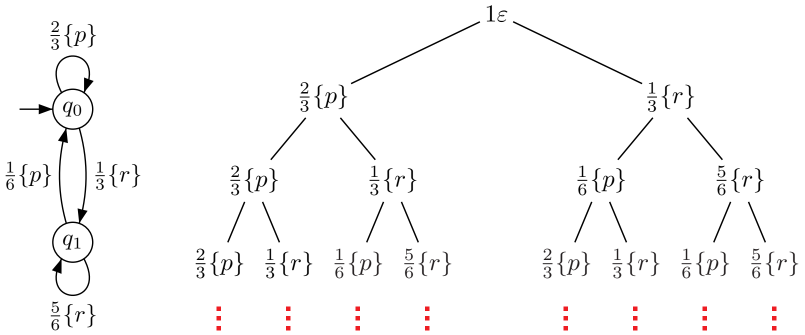
$$[\![\mu(\mathsf{F}\,b)]\!](t,\varepsilon) = \bigoplus_{w \text{ left branch}} \bigotimes_{v \mid \varepsilon < v \leq w} \kappa_t(v) = \sum_{n \geq 0} \frac{1}{2^n} \cdot \frac{1}{4} \cdot 1 = \frac{1}{2}$$

Infinite sums and products

Some well-defined infinite sums or products

- $\bigoplus_{i\in I} k_i$ is well defined if $|\{i \in I \mid k_i \neq 0\}| < \infty$,
- $\bigotimes_{i\in I} k_i$ is well defined if $|\{i \in I \mid k_i \neq 1\}| < \infty$,
- $\bigotimes_{i\in I} k_i$ is well defined if $k_i = 0$ for some $i \in I$,
- $\sum_{i\geq 0} \frac{1}{2^i}$

Unfoldings of gPFA



Probability measure

- The weight of each branch is an infinite product which converges to 0.
- The sum of the weights of all branches starting from any node should be 1.
- To define $[\![\mu(\psi)]\!]$, we use the probability measure on the sequence space.
- We get $[\![\mu(p \,\mathsf{SU}\, r)]\!](t, \varepsilon) = \sum_{n \geq 0} \left(\frac{2}{3}\right)^n \cdot \frac{1}{3} = 1.$

Probability measure

Definition: Let $\mathcal{A} = (Q, \mu)$ be a GPFA over $\mathbb{P}\mathrm{rob} = (\mathbb{R}_{\geq 0}, +, \cdot, 0, 1)$

- Let t^q be the tree unfolding of $\mathcal{A}$ starting from q,
- $D = \Sigma \times Q$ is the set of directions of t^q,
- For $u = (a_1, q_1)(a_2, q_2) \cdots (a_n, q_n) \in D^*$ and $q_0 = q$, we let

$$\mathrm{prob}^q(uD^\omega) = \prod_{i=1}^{n} \mu(q_{i-1}, a_i, q_i) = \prod_{v \in \mathrm{Pref}(u)} \kappa_{t^q}(u) \ .$$

- If ψ is a boolean path formula, then

$$\mathcal{L}_u^q(\psi) = \{w \in D^\omega \mid t^q, uw, u \models \psi\}$$

is regular, hence measurable (Vardi '85) and we define

$$[\![\mu(\psi)]\!](t^q, u) = \mathrm{prob}^{\mathsf{last}(q,u)}(\mathcal{L}_u^q(\psi))$$

PCTL* is a boolean fragment of wCTL*

Definition: Probabilistic computation tree logic PCTL* de Alfaro '98

The syntax of PCTL* is given by:

Boolean path formulas: $\psi ::= \varphi \mid \psi \wedge \psi \mid \neg\psi \mid \psi \, \mathsf{SU}^{\leq n} \, \psi$

Boolean state formulas: $\varphi ::= 0 \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mu(\psi) \geq k \mid \mu(\psi) > k$

where $n \in \mathbb{N} \cup \{\infty\}$, $p \in \mathit{Prop}$, $k \in [0,1]$.

Recall: Syntax of wCTL$^*(\mathbb{P}\mathrm{rob}, \mathit{Prop}, \{\geq\})$

Boolean path formulas: $\psi ::= \varphi \mid \psi \wedge \psi \mid \neg\psi \mid \psi \, \mathsf{SU} \, \psi$

Quantitative state formulas: $\varphi ::= k \mid \kappa \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \geq \varphi \mid \mu(\psi)$

where $p \in \mathit{Prop}$, $k \in \mathbb{R}$.

Remark: PCTL* is a boolean fragment of wCTL$^*(\mathbb{P}\mathrm{rob}, \mathit{Prop}, \{\geq\})$

State formulas are restricted:

- do not use κ,
- use $\geq$ and $\mu(\psi)$ only in comparisons of the form: $(\mu(\psi) \geq k)$ or $\neg(k \geq \mu(\psi))$

wCTL **is a fragment of** wCTL*

Definition: Syntax of wCTL$(\mathbb{K}, Prop, \mathcal{C})$

Only quantitative state formulas:

$$\varphi ::= k \mid \kappa \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \bowtie(\varphi_1, \ldots, \varphi_{\mathrm{arity}(\bowtie)}) \mid \mu(\varphi \,\mathsf{SU}^{\leq n}\, \varphi)$$

where $p \in Prop$, $k \in K$, $\bowtie \in \mathcal{C}$, $n \in \mathbb{N} \cup \{\infty\}$.

Recall: Syntax of wCTL$^*(\mathbb{K}, Prop, \mathcal{C})$

Boolean path formulas: $\psi ::= \varphi \mid \psi \wedge \psi \mid \neg\psi \mid \psi \,\mathsf{SU}\, \psi$

Quantitative state formulas:

$$\varphi ::= k \mid \kappa \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \bowtie(\varphi_1, \ldots, \varphi_{\mathrm{arity}(\bowtie)}) \mid \mu(\psi)$$

where $p \in Prop$, $k \in K$, $\bowtie \in \mathcal{C}$.

Remark: wCTL is a fragment of wCTL$^*(\mathbb{K}, Prop, \mathcal{C})$

Boolean path formulas are restricted to $\psi ::= \varphi \,\mathsf{SU}^{\leq n}\, \varphi$

PCTL is a fragment of wCTL

Definition: Probabilistic CTL Hansson & Jonsson '94

Only Boolean state formulas:

$$\varphi ::= 0 \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mu(\varphi \,\mathsf{SU}^{\leq n}\, \varphi) \geq k \mid \mu(\varphi \,\mathsf{SU}^{\leq n}\, \varphi) > k$$

where $n \in \mathbb{N} \cup \{\infty\}$, $p \in \mathit{Prop}$, $k \in [0,1]$.

Recall: Syntax of wCTL$(\mathbb{P}\mathsf{rob}, \mathit{Prop}, \{\geq\})$

Only quantitative state formulas:

$$\varphi ::= k \mid \kappa \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \geq \varphi \mid \mu(\varphi \,\mathsf{SU}^{\leq n}\, \varphi)$$

where $p \in \mathit{Prop}$, $k \in [0,1]$, $n \in \mathbb{N} \cup \{\infty\}$.

Remark: PCTL is a fragment of wCTL$(\mathbb{P}\mathsf{rob}, \mathit{Prop}, \{\geq\})$

wCTL^* **is a fragment of** wMSO

Theorem: Bollig & G. DLT'09

wCTL^* is a fragment of wMSO for finite trees and arbitrary semirings.

Proof: Translation of boolean path formulas

$$\psi ::= \varphi \mid \psi \wedge \psi \mid \neg\psi \mid \psi \,\mathsf{SU}\, \psi$$

Implicitely, ψ has two free variables, the path (set of nodes) and the current node.

We build a boolean MSO formula $\underline{\psi}(x, X) \in \mathsf{bMSO}(\mathbb{K}, \Sigma, \mathcal{C})$.

$$\underline{\varphi}(x, X) = (\overline{\varphi}(x) \neq \mathbf{0})$$

$$\underline{\psi_1 \wedge \psi_2}(x, X) = \underline{\psi_1}(x, X) \wedge \underline{\psi_2}(x, X)$$

$$\underline{\neg\psi}(x, X) = \neg\underline{\psi}(x, X)$$

$$\underline{\psi_1 \,\mathsf{SU}\, \psi_2}(x, X) = \underline{\exists}\, z\, (z \in X \wedge x < z \wedge \underline{\psi_2}(z, X) \wedge \forall y\, ((x < y < z) \xrightarrow{+} \underline{\psi_1}(y, X)))$$

We assume that the interpretation of X is indeed a path.

We use $\underline{\exists}$, $\underline{\vee}$ and $\xrightarrow{+}$ to get boolean formulas.

wCTL* **is a fragment of** wMSO

Proof: Translation of quantitative state formulas

$$\varphi ::= k \mid \kappa \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \bowtie(\varphi_1, \ldots, \varphi_{\mathrm{arity}(\bowtie)}) \mid \mu(\psi)$$

Here, φ only has an implicit free variable, the current node.

We build a weighted MSO formula $\overline{\varphi}(x) \in \mathsf{wMSO}(\mathbb{K}, \Sigma, \mathcal{C})$.

$$[\![\mu(\psi)]\!](t,u) = \bigoplus_{w \in \mathrm{Branches}(t) \mid t,w,u \models \psi} \quad \bigotimes_{v \mid u < v \leq w} \kappa_t(v)$$

$$\overline{\mu(\psi)}(x) = \exists X\,(\mathrm{path}(x,X) \wedge \underline{\psi}(x,X) \wedge \xi(x,X))$$

$$\begin{aligned}
\mathrm{path}(x,X) = {} & x \in X \\
& \wedge \forall z\,(z \in X \xrightarrow{+} (z = x \mathbin{\underline{\vee}} \underline{\exists} y\,(y \in X \wedge y \lessdot z))) \\
& \wedge \neg\,\underline{\exists} y, z, z' \in X\,(y \lessdot z \wedge y \lessdot z' \wedge z \neq z') \\
& \wedge \forall y\,(\ (y \in X \wedge \underline{\exists} z\,(y < z)) \xrightarrow{+} \underline{\exists} z\,(z \in X \wedge y < z)\ )
\end{aligned}$$

$$\xi(x,X) = \forall y\,((y \in X \wedge x < y) \xrightarrow{+} \kappa(y))$$

wCTL **is a fragment of** wMSO **on gPFA**

Theorem: Bollig & G. DLT'09

wCTL is a fragment of wMSO on probabilistic systems (GPFA).

Unfoldings of probabilistic systems (GPFA) are infinite.

The translation of $\overline{\mu(\psi)}(x)$ given above does not work.

We need to be careful with the induced infinite sums and products.

wCTL **is a fragment of** wMSO **on gPFA**

Proof: Translation of $\mu(\varphi_1 \,\mathsf{SU}^{\leq n}\, \varphi_2)$

$$\overline{\mu(\varphi_1 \,\mathsf{SU}^{\leq n}\, \varphi_2)}(x) = \exists X\, (\mathrm{path}^{\leq n}(x, X) \wedge \underline{\psi}(x, X) \wedge \xi(x, X))$$

$$\begin{aligned}
\mathrm{path}^{\leq \infty}(x, X) = {} & x \in X \\
& \wedge \forall z\, (z \in X \xrightarrow{+} (z = x \,\underline{\vee}\, \underline{\exists} y\, (y \in X \wedge y \lessdot z))) \\
& \wedge \neg \underline{\exists} y, z, z' \in X\, (y \lessdot z \wedge y \lessdot z' \wedge z \neq z')
\end{aligned}$$

$$\mathrm{path}^{\leq n}(x, X) = \mathrm{path}^{\leq \infty}(x, X) \wedge \neg \underline{\exists} x_1, \ldots, x_n \in X\, (x < x_1 < \cdots < x_n)$$

$$\psi = (\varphi_1 \wedge \neg\varphi_2) \,\mathsf{SU}\, (\varphi_2 \wedge \neg(\mathbf{0} \,\mathsf{SU}\, \mathbf{1}))$$

$$\xi(x, X) = \forall y\, ((y \in X \wedge x < y) \xrightarrow{+} \kappa(y))$$

$\mathrm{path}^{\leq n}(x, X) \wedge \underline{\psi}(x, X)$ is a boolean formula which holds if and only if X is a minimal path satisfying $\varphi_1 \,\mathsf{SU}^{\leq n}\, \varphi_2$.

$\xi(x, X)$ computes the probability of this finite path.

$\exists X$ computes the sum of the probability of such paths.

Plan

MSO Logic

Weighted MSO Logic

Weighted MSO versus Weighted Automata

Weighted CTL* **and** PCTL*

5 Conclusion and Open Problems

Conclusion

- There is a very rich theory for probabilistic systems.
  - Various logics for specification
  - Efficient algorithms for model checking
  - and much more (probabilistic bisimulation, ...)
- Analysis of other quantitative properties is more and more important.
  Reliability, energy consumption, ...
- We should develop a strong theory for analysis of various quantitative aspects

  Building upon existing theory of weighted automata
  and the large experience in analysing probabilistic systems.

Open problems

Problems on wMSO

- Identify fragments for which satisfiability and model checking are decidable.
- Compare expressivity of wCTL* (or PCTL*) and wMSO on GPFA.
- Compare expressivity of wCTL* (or PCTL*) and wMSO on RPFA.
- Extend the comparison to other semirings.
  E.g. the Expectation semiring — Eisner '01
  Useful to compute expected rewards.
- Find a weighted μ-calculus which contains wCTL and compare its expressivity with wMSO.
  Weighted μ-calculus on words — Meinecke, DLT'09
  Weighted μ-calculus for quantitative games — Fischer, Grädel & Kaiser '08