

# Weighted versus Probabilistic Logics

Benedikt Bollig and Paul Gastin

LSV, ENS Cachan, INRIA, CNRS, FRANCE

DLT, 30 June 2009

# Motivations

## Analysis of quantitative systems

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

## Models

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

All are special cases of Weighted Automata.

# Motivations

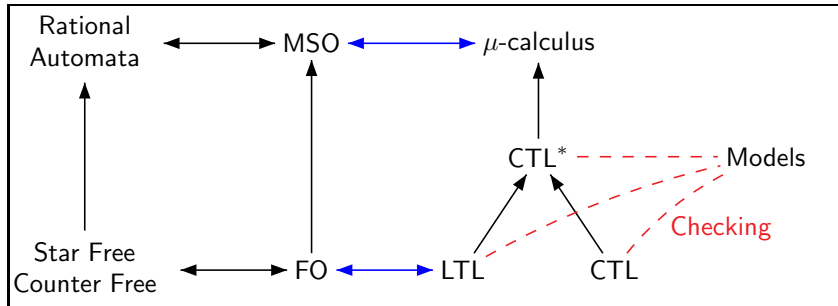
## Specification

- ▶ 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

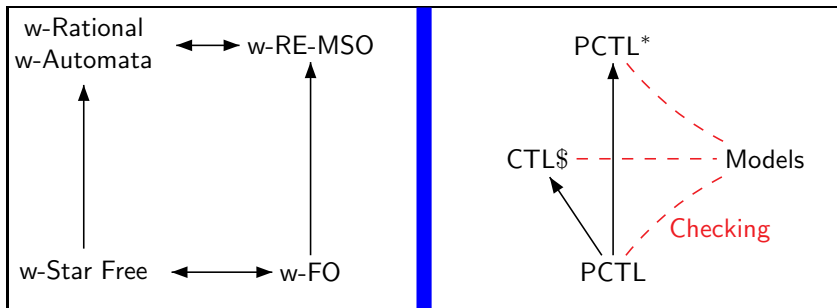
## Natural Problems

- ▶ Satisfiability
- ▶ Model Checking
- ▶ Expressivity

# Qualitative (Boolean) Picture



# Quantitative Picture



Our aim is to compare and unify these logics

# Plan

## ① Weighted Automata

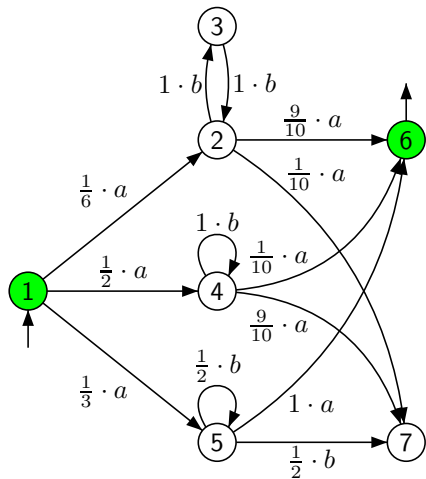
Weighted MSO Logic

Weighted CTL\* and PCTL\*

Weighted CTL\* versus weighted MSO

Conclusion and Open problems

# Weighted Automata by Examples



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

$$\pi_1 = 1 \xrightarrow{a} 4 \xrightarrow{b} 4 \cdots 4 \xrightarrow{b} 4 \xrightarrow{a} 6$$

$$\text{weight}(\pi_1) = \frac{1}{2} \cdot 1^n \cdot \frac{1}{10} = \frac{1}{20}$$

$$\pi_2 = 1 \xrightarrow{a} 5 \xrightarrow{b} 5 \cdots 5 \xrightarrow{b} 5 \xrightarrow{a} 6$$

$$\text{weight}(\pi_2) = \frac{1}{3} \cdot \left(\frac{1}{2}\right)^n \cdot 1 = \frac{1}{3 \cdot 2^n}$$

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

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

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

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

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

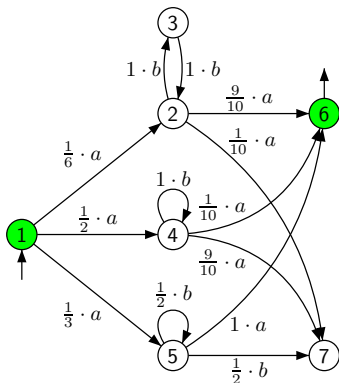


# Reactive Probabilistic Finite Automata

Definition: RPFA on  $\mathbb{Prob} = (\mathbb{R}_{\geq 0}, +, \cdot, 0, 1)$

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

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

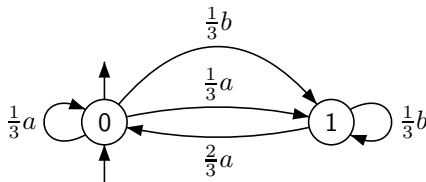


# Generative Probabilistic Finite Automata

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

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

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



# Plan

## Weighted Automata

### 2 Weighted MSO Logic

## Weighted CTL\* and PCTL\*

## Weighted CTL\* versus weighted MSO

## Conclusion and Open problems

# Weighted MSO

## Short history

Introduced by Droste & Gastin (ICALP'05)

Aim: Logical characterization of weighted automata.

Generalization of Elgot's and Büchi's theorems.

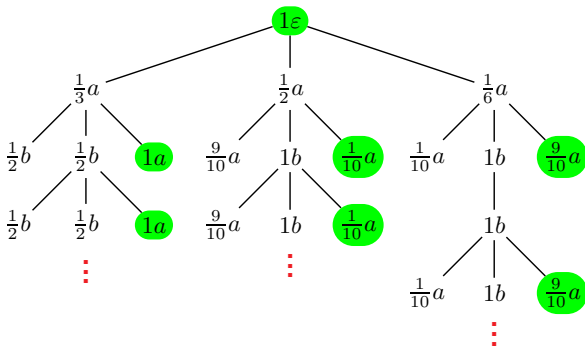
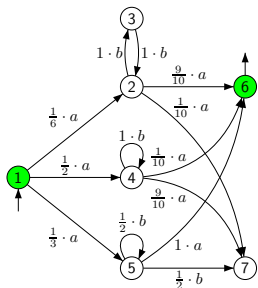
Extended to

- |                       |                                  |
|-----------------------|----------------------------------|
| ▶ Trees               | Droste & Vogler                  |
| ▶ Infinite words      | Droste & Kuske, Droste & Rahonis |
| ▶ Pictures            | Fischtner                        |
| ▶ Traces              | Meinecke                         |
| ▶ Distributed systems | Bollig & Meinecke                |
| ▶ ...                 |                                  |

No link with quantitative temporal logics such as PCTL or CTL\$.

Semantics on unfoldings of weighted automata

# Unfoldings of Weighted Automata



Definition: Weighted Tree

$$\begin{aligned}
 t: D^* &\rightarrow K \times \Sigma \\
 u &\rightarrow (\kappa_t(u), \ell_t(u))
 \end{aligned}$$

# Extended Weighted MSO

## Definition: Vocabulary

- ▶  $\mathcal{C}$  is a vocabulary of symbols  $\bowtie \in \mathcal{C}$  with  $\text{arity}(\bowtie) \in \mathbb{N}$ .
  - ▶  $\mathcal{C} = \{\vee, \wedge, \neg\}$
  - ▶  $\mathcal{C} = \{\wedge, \neg, <\}$
- ▶ Each symbol  $\bowtie \in \mathcal{C}$  is given a semantics  $\llbracket \bowtie \rrbracket : K^{\text{arity}(\bowtie)} \rightarrow K$ .
  - ▶  $\llbracket \vee \rrbracket = \oplus$
  - ▶  $\llbracket \wedge \rrbracket = \otimes$
  - ▶  $\llbracket \neg \rrbracket(k) = \begin{cases} \mathbf{1} & \text{if } k = \mathbf{0} \\ \mathbf{0} & \text{otherwise} \end{cases}$
  - ▶ Probabilistic:  $\llbracket \neg \rrbracket(k) = 1 - k$  or  $\llbracket \neg \rrbracket(k) = \max(0, 1 - k)$
  - ▶ Ordered semiring:  $\llbracket < \rrbracket : K^2 \rightarrow \{\mathbf{0}, \mathbf{1}\}$

# Extended Weighted MSO

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

$$\begin{aligned} \varphi ::= & k \mid \kappa(x) \mid \bowtie(\varphi_1, \dots, \varphi_{\text{arity}(\bowtie)}) \\ & \mid P_a(x) \mid x \leq y \mid x \in X \mid \exists x.\varphi \mid \exists X.\varphi \mid \forall x.\varphi \mid \forall X.\varphi \end{aligned}$$

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

Definition: Semantics:  $\llbracket \varphi \rrbracket_{\mathcal{V}} : \text{Trees}(D, \mathbb{K}, \Sigma_{\mathcal{V}}) \rightarrow K$

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

Let  $t : D^* \rightarrow K \times \Sigma$  be a weighted tree and  $\sigma$  a  $(\mathcal{V}, t)$ -assignment.  
 $u \rightarrow (\kappa_t(u), \ell_t(u))$

$$\begin{aligned} \llbracket k \rrbracket_{\mathcal{V}}(t, \sigma) &= k \\ \llbracket \kappa(x) \rrbracket_{\mathcal{V}}(t, \sigma) &= \kappa_t(\sigma(x)) \\ \llbracket P_a(x) \rrbracket_{\mathcal{V}}(t, \sigma) &= \begin{cases} 1 & \text{if } \ell_t(\sigma(x)) = a \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

$$\llbracket \bowtie(\varphi_1, \dots, \varphi_r) \rrbracket_{\mathcal{V}}(t, \sigma) = \llbracket \bowtie \rrbracket(\llbracket \varphi_1 \rrbracket_{\mathcal{V}}(t, \sigma), \dots, \llbracket \varphi_r \rrbracket_{\mathcal{V}}(t, \sigma)) \quad \text{if } \text{arity}(\bowtie) = r$$

Recall that  $\llbracket \vee \rrbracket = \oplus$  and  $\llbracket \wedge \rrbracket = \otimes$

# Extended Weighted MSO

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

$$\begin{aligned} \varphi ::= & k \mid \kappa(x) \mid \bowtie(\varphi_1, \dots, \varphi_{\text{arity}(\bowtie)}) \\ & \mid P_a(x) \mid x \leq y \mid x \in X \mid \exists x.\varphi \mid \exists X.\varphi \mid \forall x.\varphi \mid \forall X.\varphi \end{aligned}$$

Definition: Semantics:  $\llbracket \varphi \rrbracket_{\mathcal{V}} : \text{Trees}(D, \mathbb{K}, \Sigma_{\mathcal{V}}) \rightarrow K$

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

Let  $t : D^* \rightarrow K \times \Sigma$  be a weighted tree and  $\sigma$  a  $(\mathcal{V}, t)$ -assignment.  
 $u \rightarrow (\kappa_t(u), \ell_t(u))$

$$\llbracket x \leq y \rrbracket_{\mathcal{V}}(t, \sigma) = \begin{cases} \mathbf{1} & \text{if } \sigma(x) \leq \sigma(y) \\ \mathbf{0} & \text{otherwise} \end{cases} \quad \begin{array}{l} \leq \text{ is the prefix} \\ \text{ordering on } \text{dom}(t) \end{array}$$

$$\llbracket x \in X \rrbracket_{\mathcal{V}}(t, \sigma) = \begin{cases} \mathbf{1} & \text{if } \sigma(x) \in \sigma(X) \\ \mathbf{0} & \text{otherwise} \end{cases}$$



# Extended Weighted MSO

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

$$\begin{aligned} \varphi ::= & k \mid \kappa(x) \mid \bowtie(\varphi_1, \dots, \varphi_{\text{arity}(\bowtie)}) \\ & \mid P_a(x) \mid x \leq y \mid x \in X \mid \exists x.\varphi \mid \exists X.\varphi \mid \forall x.\varphi \mid \forall X.\varphi \end{aligned}$$

Definition: Semantics:  $\llbracket \varphi \rrbracket_{\mathcal{V}} : \text{Trees}(D, \mathbb{K}, \Sigma_{\mathcal{V}}) \rightarrow K$

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

Let  $t : D^* \rightarrow K \times \Sigma$  be a weighted tree and  $\sigma$  a  $(\mathcal{V}, t)$ -assignment.

$$u \rightarrow (\kappa_t(u), \ell_t(u))$$

$$\llbracket \exists x.\varphi \rrbracket_{\mathcal{V}}(t, \sigma) = \bigoplus_{u \in \text{dom}(t)} \llbracket \varphi \rrbracket_{\mathcal{V} \cup \{x\}}(t, \sigma[x \rightarrow u])$$

$$\llbracket \exists X.\varphi \rrbracket_{\mathcal{V}}(t, \sigma) = \bigoplus_{U \subseteq \text{dom}(t)} \llbracket \varphi \rrbracket_{\mathcal{V} \cup \{X\}}(t, \sigma[X \rightarrow U])$$

$$\llbracket \forall x.\varphi \rrbracket_{\mathcal{V}}(t, \sigma) = \bigotimes_{u \in \text{dom}(t)} \llbracket \varphi \rrbracket_{\mathcal{V} \cup \{x\}}(t, \sigma[x \rightarrow u])$$

$$\llbracket \forall X.\varphi \rrbracket_{\mathcal{V}}(t, \sigma) = \bigotimes_{U \subseteq \text{dom}(t)} \llbracket \varphi \rrbracket_{\mathcal{V} \cup \{X\}}(t, \sigma[X \rightarrow U])$$

# Examples and Macros

Example:

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

$$\llbracket \varphi_1 \rrbracket(t) = \bigoplus_{u \in \text{dom}(t)} P_b(u) \wedge (\kappa(u) > 0)$$

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

# Examples and Macros

Definition: Useful macro

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

If  $\varphi_1$  is boolean, we have

$$\llbracket \varphi_1 \xrightarrow{+} \varphi_2 \rrbracket_{\mathcal{V}}(t, \sigma) = \begin{cases} \llbracket \varphi_2 \rrbracket_{\mathcal{V}}(t, \sigma) & \text{if } \llbracket \varphi_1 \rrbracket_{\mathcal{V}}(t, \sigma) = \mathbf{1} \\ \mathbf{1} & \text{otherwise.} \end{cases}$$

Example:

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

$$\llbracket \varphi_1 \rrbracket(t) = \bigotimes_{u \in \text{dom}(t)} ((P_a(x) \wedge (\kappa(x) > 0)) \xrightarrow{+} \kappa(x))$$

multiplies the positive values of all  $a$ -labeled nodes.

# Examples and Macros

Definition: Macros for Boolean formulas

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

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

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

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

Example:

- ▶ Let  $\text{path}(x, X)$  be a boolean formula stating that  $X$  is a maximal path starting from node  $x$ ,
- ▶ The boolean formula  $\exists z.(z \in X \wedge P_b(z) \wedge \forall y.(x < y < z \xrightarrow{+} P_a(y)))$  checks if  $X$  satisfies  $a$  SU  $b$ ,
- ▶ 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

$$\exists X.(\text{path}(x, X) \wedge \exists z.(z \in X \wedge P_b(z) \wedge \forall y.(x < y < z \xrightarrow{+} P_a(y))) \wedge \xi(x, X))$$

# Original Weighted MSO

## Definition: Original Weighted MSO

Droste & Gastin

- ▶  $\mathcal{C} = \{\vee, \wedge\}$
- ▶ negations of atomic formulas
- ▶  $\kappa(x)$  is not allowed

sREMSO is a syntactic restriction of the existential fragment.

## Theorem: Droste & Gastin

- ▶ From any  $w$ -Aut  $\mathcal{A}$  we can construct a formula  $\varphi$  in sREMSO s.t.  $\llbracket \varphi \rrbracket = \llbracket \mathcal{A} \rrbracket$ ,
- ▶ From any formula  $\varphi$  in sREMSO we can construct a  $w$ -Aut  $\mathcal{A}$  s.t.  $\llbracket \varphi \rrbracket = \llbracket \mathcal{A} \rrbracket$ .

## Definition: Satisfiability

A formula  $\varphi$  is satisfiable if  $\llbracket \varphi \rrbracket(t) \neq \mathbf{0}$  for some  $t$ .

## Corollary: Satisfiability

The satisfiability problem is decidable for sREMSO.

# Extended Weighted MSO

## Proposition: Satisfiability

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

## Proof:

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

By [DG],  $\exists \varphi \in \text{sREMSO}(\mathbb{P}\text{rob}, \Sigma, \{\vee, \wedge, \neg\})$  such that  $\llbracket \varphi \rrbracket(w) = \llbracket \mathcal{A} \rrbracket(w)$  for all unweighted words  $w \in \Sigma^*$ .

Since  $\varphi$  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  $\llbracket p < \varphi \rrbracket(w) \neq 0$  iff  $\llbracket \mathcal{A} \rrbracket(w) > p$ .

Hence,  $p < \varphi$  is satisfiable iff the automaton  $\mathcal{A}$  with threshold  $p$  accepts a nonempty language. By Rabin's Theorem, this is undecidable.

# Plan

Weighted Automata

Weighted MSO Logic

3 Weighted CTL\* and PCTL\*

Weighted CTL\* versus weighted MSO

Conclusion and Open problems

# Weighted CTL\*

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

**Boolean** path formulas:  $\psi ::= \varphi \mid \psi \wedge \psi \mid \neg\psi \mid \psi \text{ SU } \psi$

**Quantitative** state formulas:  $\varphi ::= k \mid \kappa \mid p \mid \bowtie(\varphi_1, \dots, \varphi_{\text{arity}(\bowtie)}) \mid \mu(\psi)$

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

Definition: Semantics for Boolean path formulas

$t: D^* \rightarrow K \times \Sigma$       weighted tree,  $w$  branch of  $t$ ,  $u$  node on  $w$ .  
 $u \rightarrow (\kappa_t(u), \ell_t(u))$

$t, w, u \models \varphi$       if  $\llbracket \varphi \rrbracket(t, u) \neq \mathbf{0}$

$t, w, u \models \psi_1 \wedge \psi_2$  if  $t, w, u \models \psi_1$  and  $t, w, u \models \psi_2$

$t, w, u \models \neg\psi$       if  $t, w, u \not\models \psi$

$t, w, u \models \psi_1 \text{ SU } \psi_2$  if  $\exists u < v \leq w : (t, w, v \models \psi_2 \text{ and } \forall u < v' < v : t, w, v' \models \psi_1)$



# Weighted CTL\*

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

**Boolean** path formulas:  $\psi ::= \varphi \mid \psi \wedge \psi \mid \neg\psi \mid \psi \text{ SU } \psi$

**Quantitative** state formulas:  $\varphi ::= k \mid \kappa \mid p \mid \bowtie(\varphi_1, \dots, \varphi_{\text{arity}(\bowtie)}) \mid \mu(\psi)$

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

Definition: Semantics for quantitative state formulas

$t : D^* \rightarrow K \times \Sigma$       weighted tree,  $u$  node of  $t$ ,  $\Sigma = 2^{Prop}$ .  
 $u \rightarrow (\kappa_t(u), \ell_t(u))$

$$\llbracket k \rrbracket(t, u) = k$$

$$\llbracket \kappa \rrbracket(t, u) = \kappa_t(u)$$

$$\llbracket p \rrbracket(t, u) = \begin{cases} 1 & \text{if } p \in \ell_t(u) \\ 0 & \text{otherwise} \end{cases}$$

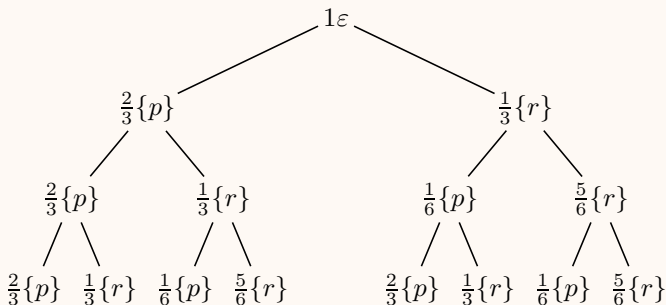
$$\llbracket \bowtie(\varphi_1, \dots, \varphi_r) \rrbracket(t, u) = \llbracket \bowtie \rrbracket(\llbracket \varphi_1 \rrbracket(t, u), \dots, \llbracket \varphi_r \rrbracket(t, u)) \quad \text{if } \text{arity}(\bowtie) = r$$

$$\llbracket \mu(\psi) \rrbracket(t, u) = \bigoplus_{w \in \text{Branches}(t) \mid t, w, u \models \psi} \bigotimes_{v \mid u < v \leq w} \kappa_t(v)$$

# Example for $\mu(\psi)$

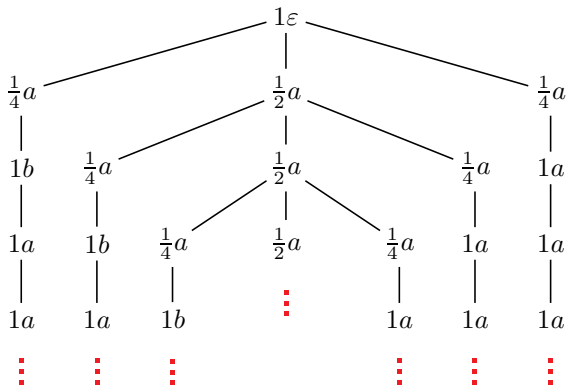
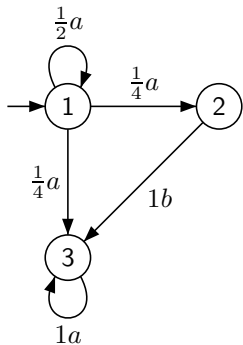
Example:

$$\llbracket \mu(\psi) \rrbracket(t, u) = \bigoplus_{w \in \text{Branches}(t) \mid t, w, u \models \psi} \bigotimes_{v \mid u < v \leq w} \kappa_t(v)$$



$$\llbracket \mu(p \text{ SU } r) \rrbracket(t) = \frac{2}{3} \cdot \frac{2}{3} \cdot \frac{1}{3} + \frac{2}{3} \cdot \frac{1}{3} \cdot \left( \frac{1}{6} + \frac{5}{6} \right) + \frac{1}{3} \cdot (1) = \frac{19}{27}$$

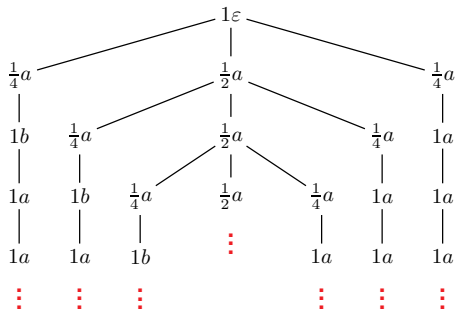
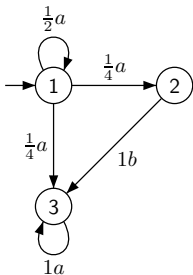
# Unfoldings are infinite (regular) trees



# Infinite sums and products

Example:

$$\llbracket \mu(a^*ba^\omega) \rrbracket(t, \varepsilon) = \bigoplus_{w \text{ left branch}} \bigotimes_{v \mid \varepsilon < v \leq w} \kappa_t(v) = \sum_{n \geq 0} \frac{1}{2^n} \cdot \frac{1}{4} \cdot 1^\omega = \frac{1}{2}$$

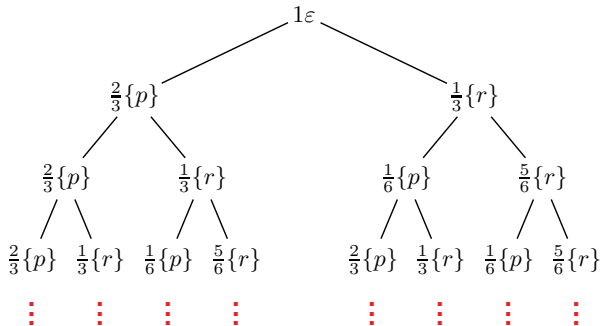
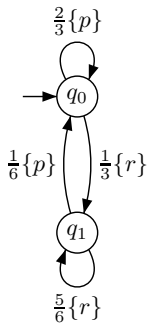


# We need infinite sums and products

## 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  $\llbracket \mu(\psi) \rrbracket$ , we use the probability measure on the sequence space.
- ▶ We get  $\llbracket \mu(p \text{ SU } r) \rrbracket(t, \varepsilon) = \sum_{n \geq 0} \left(\frac{2}{3}\right)^n \cdot \frac{1}{3} = 1$ .

# PCTL\* is a boolean fragment of wCTL\*

Definition: Probabilistic computation tree logic PCTL\* de Alfaró '98

The syntax of PCTL\* is given by:

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

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

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

Recall: Syntax of wCTL\*( $\mathbb{P}rob, Prop, \{\neg, \wedge, \geq\}$ )

**Boolean** path formulas:  $\psi ::= \varphi \mid \psi \wedge \psi \mid \neg\psi \mid \psi \text{ SU } \psi$

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

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

Remark: PCTL\* is a boolean fragment of wCTL\*

State formulas are restricted:

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

# wCTL is a fragment of wCTL\*

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

Only quantitative state formulas:

$$\varphi ::= k \mid \kappa \mid p \mid \bowtie(\varphi_1, \dots, \varphi_{\text{arity}(\bowtie)}) \mid \mu(\varphi \text{ SU}^{\leq n} \varphi)$$

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

Quantitative state formulas:  $\varphi ::= k \mid \kappa \mid p \mid \bowtie(\varphi_1, \dots, \varphi_{\text{arity}(\bowtie)}) \mid \mu(\psi)$

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  $\psi ::= \varphi \text{ SU}^{\leq n} \varphi$



# PCTL is a fragment of wCTL

Definition: Probabilistic CTL

Hansson & Jonsson '94

Only Boolean state formulas:

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

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

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

Only quantitative state formulas:

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

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

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

# Plan

Weighted Automata

Weighted MSO Logic

Weighted CTL\* and PCTL\*

4 Weighted CTL\* versus weighted MSO

Conclusion and Open problems

# wCTL\* is a fragment of wMSO

Definition: Translation of boolean path formulas

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

Implicitly,  $\psi$  has two free variables, the path (set of nodes) and the current node.

We associate with each path formula  $\psi \in \text{wCTL}^*(\mathbb{K}, \text{Prop}, \mathcal{C})$  a boolean MSO formula  $\underline{\psi}(x, X) \in \text{bMSO}(\mathbb{K}, \Sigma, \mathcal{C})$ .

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

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

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

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

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

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

# wCTL\* is a fragment of wMSO

Definition: Translation of quantitative state formulas

$$\varphi ::= k \mid \kappa \mid p \mid \bowtie(\varphi_1, \dots, \varphi_{\text{arity}(\bowtie)}) \mid \mu(\psi)$$

Here,  $\varphi$  only has an implicit free variables, the current node.

We associate with each state formula  $\varphi \in \text{wCTL}^*(\mathbb{K}, \text{Prop}, \mathcal{C})$  a weighted MSO formula  $\overline{\varphi}(x) \in \text{bMSO}(\mathbb{K}, \Sigma, \mathcal{C})$ .

$$\llbracket \mu(\psi) \rrbracket(t, u) = \bigoplus_{w \in \text{Branches}(t) \mid t, w, u \models \psi} \bigotimes_{v \mid u < v \leq w} \kappa_t(v)$$

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

$$\text{path}(x, X) = x \in X$$

$$\wedge \forall z. (z \in X \xrightarrow{+} (z = x \vee \exists y. (y \in X \wedge y < z)))$$

$$\wedge \neg \exists y, z, z' \in X. (y < z \wedge y < z' \wedge z \neq z')$$

$$\wedge \forall y. ((y \in X \wedge \exists z. (y < z)) \xrightarrow{+} \exists z. (z \in X \wedge y < z))$$

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

# wCTL\* is a fragment of wMSO

## Theorem:

The translation is correct for finite trees on all semirings.

# wCTL is a fragment of wMSO on 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**.

**Theorem:**

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

Recall the syntax of wCTL:

$$\varphi ::= k \mid \kappa \mid p \mid \bowtie(\varphi_1, \dots, \varphi_{\text{arity}(\bowtie)}) \mid \mu(\varphi \text{ SU}^{\leq n} \varphi)$$

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

# wCTL is a fragment of wMSO on gPFA

Definition: Translation of  $\mu(\varphi_1 \text{ SU}^{\leq n} \varphi_2)$

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

$$\text{path}^{\leq \infty}(x, X) = x \in X$$

$$\wedge \forall z. (z \in X \xrightarrow{+} (z = x \vee \exists y. (y \in X \wedge y \leq z)))$$

$$\wedge \neg \exists y, z, z' \in X. (y \leq z \wedge y \leq z' \wedge z \neq z')$$

if  $n \in \mathbb{N}$ ,  $\text{path}^{\leq n}(x, X) = \text{path}^{\leq \infty}(x, X) \wedge \neg \exists x_0 \dots \exists x_n.$

$$(x_0 \in X \wedge \dots \wedge x_n \in X \wedge x < x_0 < x_1 < \dots < x_n)$$

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

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

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

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

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

# wCTL is a fragment of wMSO on gPFA

## Theorem:

The translation is correct for infinite trees which are unfoldings of GPFA.

## Corollary:

PCTL is a fragment of wMSO on GPFA



# Plan

Weighted Automata

Weighted MSO Logic

Weighted CTL\* and PCTL\*

Weighted CTL\* versus weighted MSO

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  $GPF\mathcal{A}$ .
- ▶ 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  $\mu$ -calculus which contains wCTL and compare its expressivity with wMSO.  
Weighted  $\mu$ -calculus on words Meinecke, DLT'09  
Weighted  $\mu$ -calculus for quantitative games Fischer, Grädel & Kaiser '08

# Open problems

## Quantitative bisimulation

- ▶ Probabilistic bisimulation Larsen & Skou, '91  
It is **not quantitative**, it defines a boolean relation on states.
- ▶ Generalized to weighted automata and CTL\$ Buchholz & Kemper '09  
But still not quantitative.
- ▶ We need to study **bisimulation distances** expressing **how close** two states are.  
See Fahrenberg, Larsen & Thrane '09