Weighted Unranked Tree Automata over Valuation Monoids

Doreen Heusel

(joint work with Manfred Droste, Heiko Vogler)

Universität Leipzig

15.01.2016
Weighted word automata

- alphabet $\Sigma = \{n, \ell\}$, word $w = n\ell\ell$
- automaton $A$:

\[ \begin{array}{c}
  \ell \mid n \quad \ell \mid n \\
  p_1 \quad p_2
\end{array} \]

weight of successful runs:

\[ \begin{array}{c}
  \text{wt}(r_1) = -1 \cdot 1 \cdot 1 = -1 \\
  \text{wt}(r_2) = 1 \cdot 1 \cdot 1 = 1 \\
  \text{wt}(r_3) = 1 \cdot 1 \cdot 1 = 1
\end{array} \]

$\|A\|(w) = \sum_{\text{succ. run } r \text{ on } w} \text{wt}(r) = -1 + 1 + 1$
Weighted word automata

- alphabet $\Sigma = \{n, \ell\}$, word $w = n\ell\ell$
- automaton $A$:

  ![Diagram of an automaton with two states $p_1$ and $p_2$ and transitions labeled with $n$, $\ell$, and $\ell$]

- successful runs on $w$:
  - $r_1 = p_1 \xrightarrow{n} p_2 \xrightarrow{\ell} p_2 \xrightarrow{\ell} p_2$
  - $r_2 = p_1 \xrightarrow{n} p_1 \xrightarrow{\ell} p_1 \xrightarrow{\ell} p_2$
  - $r_3 = p_1 \xrightarrow{n} p_1 \xrightarrow{\ell} p_1 \xrightarrow{\ell} p_2$

- weight of successful runs:
  - $\text{wt}(r_1) = -1 \cdot 1 \cdot 1 = -1$
  - $\text{wt}(r_2) = 1 \cdot 1 \cdot 1 = 1$
  - $\text{wt}(r_3) = 1 \cdot 1 \cdot 1 = 1$

- $\|A\|(w) = \sum_{\text{successful run } r \text{ on } w} \text{wt}(r) = -1 + 1 + 1$
Weighted word automata

- alphabet $\Sigma = \{n, \ell\}$, word $w = n\ell\ell$
- automaton $A$:

```
1,\ell \mid 1,n
```

- successful runs on $w$:
  - $r_1 = p_1 \xrightarrow{n} p_2 \xrightarrow{\ell} p_2 \xrightarrow{\ell} p_2$
  - $r_2 = p_1 \xrightarrow{n} p_1 \xrightarrow{\ell} p_2 \xrightarrow{\ell} p_2$
  - $r_3 = p_1 \xrightarrow{n} p_1 \xrightarrow{\ell} p_1 \xrightarrow{\ell} p_2$

\[
\|A\|(w) = \sum_{\text{succ. run } r \text{ on } w} \text{wt}(r) = -1 + 1 + 1 = 1
\]
Weighted word automata

- alphabet $\Sigma = \{n, \ell\}$, word $w = n\ell\ell$
- automaton $A$:

```
graph G {
  p1 [label=1,\ell \mid 1,n
  p2 [label=1,\ell \mid 1,n

  p1 -- 1,\ell [label=1,\ell \mid 1,n
  p1 -- -1,n
  p1 -- p2 [label=1,\ell
  p2 -- p2 [label=1,\ell
  p2 -- p1 [label=1,\ell
}
```

- successful runs on $w$:
  - $r_1 = p_1 \xrightarrow{n} p_2 \xrightarrow{\ell} p_2 \xrightarrow{\ell} p_2$
  - $r_2 = p_1 \xrightarrow{n} p_1 \xrightarrow{\ell} p_2 \xrightarrow{\ell} p_2$
  - $r_3 = p_1 \xrightarrow{n} p_1 \xrightarrow{\ell} p_1 \xrightarrow{\ell} p_2$

- weight of successful runs:
  - $\text{wt}(r_1) = -1 \cdot 1 \cdot 1 = -1$
  - $\text{wt}(r_2) = 1 \cdot 1 \cdot 1 = 1$
  - $\text{wt}(r_3) = 1 \cdot 1 \cdot 1 = 1$
Weighted word automata

- alphabet $\Sigma = \{n, \ell\}$, word $w = n\ell\ell$
- automaton $A$:

  ![Diagram of automaton $A$]

  - successful runs on $w$:
    - $r_1 = p_1 \xrightarrow{n} p_2 \xrightarrow{\ell} p_2 \xrightarrow{\ell} p_2$
    - $r_2 = p_1 \xrightarrow{n} p_1 \xrightarrow{\ell} p_2 \xrightarrow{\ell} p_2$
    - $r_3 = p_1 \xrightarrow{n} p_1 \xrightarrow{\ell} p_1 \xrightarrow{\ell} p_2$
  - weight of successful runs:
    - $wt(r_1) = -1 \cdot 1 \cdot 1 = -1$
    - $wt(r_2) = 1 \cdot 1 \cdot 1 = 1$
    - $wt(r_3) = 1 \cdot 1 \cdot 1 = 1$

  $\|A\|(w) = \sum_{\text{succ. run } r \text{ on } w} wt(r) = -1 + 1 + 1$
Weight structure: semirings

Example

\((\mathbb{Z}, +, \cdot, 0, 1)\)
Weight structure: semirings

Example

\((\mathbb{Z}, +, \cdot, 0, 1), (\mathbb{R} \cup \{-\infty\}, \max, +, -\infty, 0), (\{0, 1\}, \lor, \land, 0, 1), \ldots\)
Weight structure: semirings

Example

\[(\mathbb{Z}, +, \cdot, 0, 1), (\mathbb{R} \cup \{-\infty\}, \max, +, -\infty, 0), (\{0, 1\}, \lor, \land, 0, 1), \ldots\]

\[\mathcal{S} = (\mathcal{S}, +, \cdot, 0, 1)\]

- \((K, +, 0)\) and \((K, \cdot, 1)\) monoids, i.e.:
  - +, · associative
  - \(\forall x \in \mathcal{S}: 0 + x = x + 0 = x, 1 \cdot x = x \cdot 1 = 1\)

- 0 multiplicative zero

- · distributive over +
Droste, Vogler (2009):
- weighted unranked tree automata over semirings

Theorem

\[ S = [ \phi ] \]

for some syntactically restricted wMSO sentence \( \phi \) \( \Rightarrow \) \( S \) recognizable

\[ S = \| M \| \]

for some s-wuta \( M \) \( \Rightarrow \) \( S \) wMSO definable

\( K \) commutative:
\[ S = [ \phi ] \]

for some syntactically restricted wMSO sentence \( \phi \) \( \iff \) \( S \) recognizable
Droste, Vogler (2009):
- weighted unranked tree automata over semirings
  - e.g. used to analyze (fully structured) XML-documents
Droste, Vogler (2009):
- weighted unranked tree automata over semirings
  - e.g. used to analyze (fully structured) XML-documents
- weighted MSO logic for unranked trees
Droste, Vogler (2009):
- weighted unranked tree automata over semirings
  - e.g. used to analyze (fully structured) XML-documents
- weighted MSO logic for unranked trees

**Theorem**

\[ \mathbb{K} \text{ semiring, } S \text{ tree series over } \mathbb{K}, \text{ alphabet } \Sigma. \]

1. \[ S = [\varphi] \text{ for some syntactically restricted wMSO sentence } \varphi \]
   \[ \Rightarrow S \text{ recognizable} \]

2. \[ S = \|M\| \text{ for some s-wuta } M \]
   \[ \Rightarrow S \text{ wMSO definable} \]

3. \[ \mathbb{K} \text{ commutative:} \]
   \[ S = [\varphi] \text{ for some syntactically restricted wMSO sentence } \varphi \]
   \[ \Leftrightarrow S \text{ recognizable} \]
weighted unranked tree automata $\neq$ restricted weighted MSO
weighted unranked tree automata \(\neq\) restricted weighted MSO

open problem:

2. Find a subclass of wta which is expressively equivalent to the syntactically restricted MSO\((S, \Sigma)\)-logic.

3. Are there crucially different results if we replace our depth-first left-to-right order on trees
here: tree valuation monoids (Droste, G., Märker, Meinecke 2011) as weight structure
  - based on ideas of Chatterjee, Doyen, Henzinger (2008)
  - generalized to (word) valuation monoids by Droste, Meinecke (2010)
  - model e.g. average
here: tree valuation monoids (Droste, G., Märker, Meinecke 2011) as weight structure

- based on ideas of Chatterjee, Doyen, Henzinger (2008)
- generalized to (word) valuation monoids by Droste, Meinecke (2010)
- model e.g. average

Goals

- define weighted unranked tree automata
- weighted MSO logic over tree valuation monoids
here: tree valuation monoids (Droste, G., Märker, Meinecke 2011) as weight structure
- based on ideas of Chatterjee, Doyen, Henzinger (2008)
- generalized to (word) valuation monoids by Droste, Meinecke (2010)
- model e.g. average

Goals

- define weighted unranked tree automata
- define weighted MSO logic
- study the relationship between automata and logic
here: tree valuation monoids (Droste, G., Märker, Meinecke 2011) as weight structure
- based on ideas of Chatterjee, Doyen, Henzinger (2008)
- generalized to (word) valuation monoids by Droste, Meinecke (2010)
- model e.g. average

Goals

- define weighted unranked tree automata
- define weighted MSO logic

- study the relationship between automata and logic
- solve Droste’s and Vogler’s equivalence problem
here: tree valuation monoids (Droste, G., Märker, Meinecke 2011) as weight structure
  - based on ideas of Chatterjee, Doyen, Henzinger (2008)
  - generalized to (word) valuation monoids by Droste, Meinecke (2010)
  - model e.g. average

Goals

- define weighted unranked tree automata weighted MSO logic over tree valuation monoids
- study the relationship between automata and logic
- solve Droste’s and Vogler’s equivalence problem
- study support of weighted unranked tree automata
Unranked Trees

- $U_\Sigma$: set of all trees over set $\Sigma$
$\mathbb{D} = (D, +, \text{Val}, \Diamond, \emptyset, 1)$:

- commutative monoid $(D, +, \emptyset)$
- $\text{Val}: U_D \rightarrow D$ with $\forall t \in T_D$:
  - $\text{pos}(t) = \{\varepsilon\} \rightarrow \text{Val}(t) = t(\varepsilon)$
  - $\emptyset \in \text{im}(t) \rightarrow \text{Val}(t) = \emptyset$
  - $\text{im}(t) = \{1\} \rightarrow \text{Val}(t) = 1$
- $\Diamond: D^2 \rightarrow D$; $\forall d \in D : \emptyset \Diamond d = d \Diamond \emptyset = \emptyset \land 1 \Diamond d = d \Diamond 1 = d$
(Product) Tree Valuation Monoid

\[ D = (D, +, \text{Val}, \diamond, 0, 1) : \]

- commutative monoid \((D, +, 0)\)
- \(\text{Val}: U_D \to D\) with \(\forall t \in T_D:\)
  - \(\text{pos}(t) = \{\varepsilon\} \to \text{Val}(t) = t(\varepsilon)\)
  - \(\emptyset \in \text{im}(t) \to \text{Val}(t) = 0\)
  - \(\text{im}(t) = \{1\} \to \text{Val}(t) = 1\)
- \(\diamond: D^2 \to D; \forall d \in D : 0 \diamond d = d \diamond 0 = 0 \land 1 \diamond d = d \diamond 1 = d\)

**Example (Average)**

\[ Q_{\text{avg}} = (Q \cup \{-\infty, \infty\}, \max, \text{avg}, \min, -\infty, \infty) \]

\[ t \in U_{Q_{\text{avg}} } \quad \rightsquigarrow \quad \text{avg}(t) = \frac{\sum_{u \in \text{pos}(t)} t(u)}{|\text{pos}(t)|} \]
Weighted Unranked Tree Automata (wuta)

Example over $Q_{av}$ and $\Sigma = \{a\}$:

- $M_{ex} = (Q, A, \gamma)$ with $Q = \{n, \ell\}$, $\gamma(n) = 0$, $\gamma(\ell) = 1$,

\[\begin{array}{c}
A_{n,a}: \\
\begin{array}{c}
p_1 \rightarrow 1,\ell | 0,n \\
p_1 \rightarrow 0,n \rightarrow p_2 \\
p_2 \rightarrow 1,\ell \\
p_2 \rightarrow 0,n \\
p_2 \rightarrow p \\
p \rightarrow \end{array}
\end{array}\]

$A_{\ell,a}$:
extended run \((q, s, l)\) on \(t_{ex}\):
extended run \((q, s, l)\) on \(t_{\text{ex}}\):
extended run \((q, s, l)\) on \(t_{ex}\):
extended run \((n, s, l)\) on \(t_{ex}\):
extended run \((n, s, l)\) on \(t_{ex}\):

\[
\begin{array}{c}
\text{extended run } \(n, s, l\) \\
on \text{ on } t_{ex}:
\end{array}
\]

\[
\begin{array}{c}
a n \\
(p_1, n, p_2) \\
a n \\
(p_2, n, p_2) \\
a n \\
(p_1, n, p_2) \\
a n \\
(p_2, n, p_2) \\
a n \\
(p_1, n, p_2) \\
a n \\
(p_2, n, p_2) \\
(a \ell) \\
(a \ell) \\
(a \ell) \\
(a \ell) \\
(a \ell) \\
(a \ell) \\
(a \ell) \\
(a \ell) \\
(a \ell) \\
(a \ell) \\
(a \ell)
\end{array}
\]
extended run \((n, s, l)\) on \(t_{ex}:\)
extended run \((n, s, l)\) on \(t_{\text{ex}}\):

```
(\(p_1, n, p_2\))   (\(p_2, n, p_2\))
  \(\text{a}\) \(\text{n}\)  \(\text{a}\) \(\text{n}\)
  \(\text{a}\) \(\text{l}\)  \(\text{a}\) \(\text{l}\)
(\(p_1, l, p_2\))   (\(p_2, n, p_2\))
  \(\text{a}\) \(\text{l}\)  \(\text{a}\) \(\text{l}\)
  \(\text{a}\) \(\text{l}\)
(\(p_1, l, p_2\))   (\(p_2, n, p_2\))
  \(\text{a}\) \(\text{n}\)  \(\text{a}\) \(\text{l}\)
  \(\text{a}\) \(\text{l}\)
```
extended run \((n, s, l)\) on \(t_{\text{ex}}\):
extended run \((n, s, l)\) on \(t_{ex}\):
extended run \((n, s, l)\) on \(t_{ex}:\)

![Diagram](image-url)
extended run \((n, s, l)\) on \(t_{\text{ex}}\):
extended run \((n, s, l)\) on \(t_{\text{ex}}\):
extended run \((n, s, l)\) on \(t_{\text{ex}}\):
extended run \((n, s, l)\) on \(t_{\text{ex}}\):
extended run \((n, s, l)\) on \(t_{ex}:\)

```
\[
\begin{array}{c}
\text{a } n \\
(p_1, n, p_2) \\
(p_1, l, p_2) \\
(p_1, l, p_2) \\
(p_1, l, p_2) \\
\text{a } n \\
(p_2, n, p_2) \\
(p_2, n, p_2) \\
(p_2, n, p_2) \\
(p_2, l, p_2) \\
\text{a } n \\
(p_1, n, p_2) \\
(p_2, l, p_2) \\
(p_2, l, p_2) \\
(p_2, l, p_2) \\
\end{array}
\]
```
extended run $(n, s, l)$ on $t_{ex}$:
extended run \((n, s, l)\) on \(t_{ex}\):

\[
\text{wt}(q, s, l)_u = \begin{cases} 
\gamma(q) & \text{if } u = \varepsilon, \\
\mu_A(s(u)) & \text{otherwise}
\end{cases}
\]
extended run \((n, s, l)\) on \(t_{ex}\):

\[
\text{wt}(q, s, l)_u = \begin{cases} 
\gamma(q) & \text{if } u = \varepsilon, \\
\mu_A(s(u)) & \text{otherwise}
\end{cases}
\]
extended run $(n, s, l)$ on $t_{ex}$:

\[
\begin{align*}
\text{wt}(q, s, l)_u &= \begin{cases} 
\gamma(q) & \text{if } u = \varepsilon, \\
\mu_{\mathcal{A}}(s(u)) & \text{otherwise}
\end{cases}
\end{align*}
\]
extended run \((n, s, l)\) on \(t_{\text{ex}}\):

\[
\begin{align*}
\text{wt}(q, s, l)_u &= \begin{cases} 
\gamma(q) & \text{if } u = \varepsilon, \\
\mu_A(s(u)) & \text{otherwise}
\end{cases}
\end{align*}
\]
extended run \((n, s, l)\) on \(t_{\text{ex}}\):

\[
\begin{align*}
\text{wt}(q, s, l)_u &= \begin{cases} 
\gamma(q) & \text{if } u = \varepsilon, \\
\mu_A(s(u)) & \text{otherwise}
\end{cases}
\end{align*}
\]
extended run \((n, s, l)\) on \(t_{ex}\):

\[
\begin{align*}
\text{wt}(q, s, l)_u &= \begin{cases} 
  \gamma(q) & \text{if } u = \varepsilon, \\
  \mu_A(s(u)) & \text{otherwise}
\end{cases}
\end{align*}
\]
extended run \((n, s, l)\) on \(t_{ex}\):

\[
\begin{align*}
\text{wt}(q, s, l)_u &= \begin{cases} 
\gamma(q) & \text{if } u = \varepsilon, \\
\mu_A(s(u)) & \text{otherwise}
\end{cases}
\end{align*}
\]
extended run \((n, s, l)\) on \(t_{ex}\):

\[
\begin{align*}
\text{wt}(q, s, l)_u &= \begin{cases} 
\gamma(q) & \text{if } u = \varepsilon, \\
\mu_A(s(u)) & \text{otherwise}
\end{cases}
\end{align*}
\]
extended run \((n, s, l)\) on \(t_{\text{ex}}\):

\[
\text{avg}
\]

\[
wt(q, s, l)_u = \begin{cases} 
\gamma(q) & \text{if } u = \varepsilon, \\
\mu_A(s(u)) & \text{otherwise}
\end{cases}
\]
extended run \((n, s, l)\) on \(t_{\text{ex}}\):

\[
\text{avg} \left( \begin{array}{c}
\text{wt}(q, s, l)_u = \begin{cases} 
\gamma(q) & \text{if } u = \varepsilon, \\
\mu_A(s(u)) & \text{otherwise}
\end{cases}
\end{array} \right)
\]
Remark

Wuta over tv-monoids subsume wuta over commutative semirings.
Remark

Wuta over tv-monoids subsume wuta over commutative semirings

- \( \text{wuta } \mathcal{M} = (Q, A, \gamma) \) over commutative semiring \( \mathbb{K} = (K, +, \circ, 0, 1) \):
  - has initial, final weights

Lemma

Let \( \mathbb{K} \) be a commutative semiring.

\[ \{ \text{s-wuta over } \mathbb{K} \} = \{ \text{s-wuta with initial, final weights in } \{0, 1\} \} \]
Remark

Wuta over tv-monoids subsume wuta over **commutative** semirings

- **wuta** \( M = (Q, A, \gamma) \) over commutative semiring \( K = (K, +, \circ, 0, 1) \):
  - has initial, final weights

Lemma

\( K \) **commutative semiring**.

\( \{s\text{-}wuta \text{ over } K\} = \{s\text{-}wuta \text{ with initial, final weights in } \{0, 1\}\} \)

- multiplies weights of a run

\[ \rightarrow \text{tv-monoid } K_{tv} = (K, +, Val, 0) \text{ with } Val(t) = \prod_{u \in pos(t)} t(u) \]
Remark

Wuta over tv-monoids subsume wuta over *commutative* semirings

- wuta $M = (Q, A, \gamma)$ over commutative semiring $K = (K, +, \circ, 0, 1)$:
  - has initial, final weights

Lemma

$K$ *commutative* semiring.

$\{s\text{-}wuta \text{ over } K\} = \{s\text{-}wuta \text{ with initial, final weights in } \{0, 1\}\}$

- multiplies weights of a run

$\rightarrow$ tv-monoid $K_{tv} = (K, +, \text{Val}, 0)$ with $\text{Val}(t) = \prod_{u \in \text{pos}(t)} t(u)$

- $\|M\|_K = \|M\|_{K_{tv}}$
Remark

Wuta over tv-monoids subsume wuta over commutative semirings

- \( \text{wuta } M = (Q, A, \gamma) \) over commutative semiring \( K = (K, +, \circ, 0, 1) \):
  - has initial, final weights

Lemma

\( K \) commutative semiring.
\{s-wuta over \( K \}\} = \{s-wuta with initial, final weights in \{0, 1\}\}

- multiplies weights of a run
  \( \rightarrow \) tv-monoid \( K_{tv} = (K, +, \text{Val}, \emptyset) \) with \( \text{Val}(t) = \prod_{u \in \text{pos}(t)} t(u) \)

- \( \|M\|_K = \|M\|_{K_{tv}} \)

Remark

wuta restricted to semirings are strict subclass of s-wuta
Weighted MSO Logic for Unranked Trees over tv-Monoids

Definition (Syntax of wMSO over $\mathbb{D}$)

- **boolean formulas:**
  \[ \beta ::= \text{label}_a(x) | \text{desc}(x, y) | x \leq y | x \sqsubseteq y | x \in X | \neg \beta | \beta \land \beta | \forall x \beta | \forall X \beta \]

- **weighted MSO formulas:**
  \[ \varphi ::= d | \beta | \varphi \lor \varphi | \varphi \land \varphi | \exists x \varphi | \forall x \varphi | \exists X \varphi \]
Definition (Syntax of wMSO over $\mathbb{D}$)

- **boolean formulas:**
  $$\beta ::= \text{label}_a(x) | \text{desc}(x, y) | x \leq y | x \sqsubseteq y | x \in X | \neg \beta | \beta \land \beta | \forall x \beta | \forall X \beta$$

- **weighted MSO formulas:**
  $$\varphi ::= d | \beta | \varphi \lor \varphi | \varphi \land \varphi | \exists x \varphi | \forall x \varphi | \exists X \varphi$$

Semantics of $\varphi$: maps every (tree $t$, assignment $\sigma$) to a $d \in \mathbb{D}$

- defined inductively:

<table>
<thead>
<tr>
<th>true</th>
<th>false</th>
<th>d</th>
<th>$\lor$</th>
<th>$\exists$</th>
<th>$\forall$</th>
<th>$\land$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>d</td>
<td>$+$</td>
<td>$\sum$</td>
<td>Val</td>
<td>$\diamond$</td>
</tr>
</tbody>
</table>
example wMSO formula over $Q_{\text{avg}}$ and $\Sigma = \{a\}$

- wMSO formula $\varphi_{\text{ex}}(x)$ with $\llbracket \varphi_{\text{ex}}(x) \rrbracket(t, \sigma) = \begin{cases} 1 & \text{if } \sigma(x) \text{ is leaf} \\ 0 & \text{otherwise} \end{cases}$
example wMSO formula over $\mathbb{Q}_{avg}$ and $\Sigma = \{a\}$

- wMSO formula $\varphi_{ex}(x)$ with
  $$\llbracket \varphi_{ex}(x) \rrbracket(t, \sigma) = \begin{cases} 
1 & \text{if } \sigma(x) \text{ is leaf} \\
0 & \text{otherwise}
\end{cases}$$

$$\llbracket \forall x \varphi_{ex}(x) \rrbracket(t) = \begin{bmatrix} 
0 \\
1 \end{bmatrix} \quad \text{avg} \quad = \frac{2}{4}$$
example wMSO formula over $\mathbb{Q}_{\text{avg}}$ and $\Sigma = \{a\}$

- wMSO formula $\varphi_{\text{ex}}(x)$ with $\llbracket \varphi_{\text{ex}}(x) \rrbracket(t, \sigma) = \begin{cases} 1 & \text{if } \sigma(x) \text{ is leaf} \\ 0 & \text{otherwise} \end{cases}$

\[
\llbracket \forall x \varphi_{\text{ex}}(x) \rrbracket(t) = \frac{\sum_{x} \text{avg}(\llbracket \varphi_{\text{ex}}(x) \rrbracket(t))}{|\Sigma|}
\]

\[
\llbracket \forall x \varphi_{\text{ex}}(x) \rrbracket(t) = \| M_{\text{ex}} \| (t)
\]
Weighted Tree Automata and Weighted MSO Logic

words: restriction of ∀-quantifier and conjunction

_assoc restrictions on the underlying ptv-monoid

conditionally commutative tv-semirings or cctv-semirings

- ◊ associative
- ◊ is distributive over +, Val

∀ t, t' ∈ U_D :

\[ \text{pos}(t) = \text{pos}(t') \land \text{im}(t), \text{im}(t') \text{ commute} \]

\[ \rightarrow \text{Val}(t) \diamond \text{Val}(t') = \text{Val}((t_u \diamond t'_u)_{u \in \text{pos}(t)}) \]
Weighted Tree Automata and Weighted MSO Logic

words: restriction of $\forall$-quantifier and conjunction
\( \bowtie \) restrictions on the underlying ptv-monoid

conditionally commutative tv-semirings or cctv-semirings

- \( \Diamond \) associative
- \( \Diamond \) is distributive over \(+, Val\)
- \( \forall t, t' \in U_D \) :
  
  \[ \text{pos}(t) = \text{pos}(t') \land \text{im}(t), \text{im}(t') \text{ commute} \]
  
  \[ \rightarrow \text{Val}(t) \Diamond \text{Val}(t') = \text{Val}((t_u \Diamond t'_u)_{u \in \text{pos}(t)}) \]

Example

\( Q_{\text{avg}+} = (Q \cup \{-\infty, \infty\}, \text{max, avg, min, } -\infty, \infty) \) is a cctv-semiring
Theorem

cctv-semiring, \( S : U_{\Sigma} \rightarrow \mathbb{D} \) tree series. TFAE:

1. \( S = \|\mathcal{M}\| \) for a wuta \( \mathcal{M} \)

2. \( S = \llbracket \varphi \rrbracket \) for a \( \forall \)- and commutatively \( \land \)-restricted wMSO sentence \( \varphi \)
Theorem

\[ \mathbb{D} \text{ cctv-semiring, } S : U_{\Sigma} \to \mathbb{D} \text{ tree series. } \text{TFAE:} \]

1. \( S = ||M|| \) for a wuta \( M \)
2. \( S = [\varphi] \) for a \( \forall \)- and commutatively \( \land \)-restricted wMSO sentence \( \varphi \)

- solves Droste’s and Vogler’s equivalence problem
Theorem

\[ \text{cctv-semiring, } S: U_\Sigma \rightarrow \mathbb{D} \ \text{tree series.} \ \text{TFAE:} \]

1. \( S = \|M\| \) for a wuta \( M \)
2. \( S = \llbracket \varphi \rrbracket \) for a \( \forall \)- and commutatively \( \land \)-restricted wMSO sentence \( \varphi \)

- solves Droste’s and Vogler’s equivalence problem
- generalization of respective results for:
  - s-wuta and wMSO over commutative semirings (Droste, Vogler)
Theorem

**D** cctv-semiring, \( S : U_\Sigma \to \mathbb{D} \) tree series. TFAE:

1. \( S = \| M \| \) for a wuta \( M \)
2. \( S = \left[ \varphi \right] \) for a \( \forall \)- and commutatively \( \land \)-restricted wMSO sentence \( \varphi \)

- solves Droste’s and Vogler’s equivalence problem
- generalization of respective results for:
  - s-wuta and wMSO over commutative semirings (Droste, Vogler)
  - wta and wMSO for ranked trees over tv-monoids (Droste et al.)
Support of wuta

Satisfiability Problem

∃ tree $t$ with $\llbracket \varphi \rrbracket(t) \neq 0$ for given formula $\varphi$?
Support of wuta

**Satisfiability Problem**

\[ \exists\ \text{tree } t \text{ with } \llbracket \varphi \rrbracket(t) \neq 0 \text{ for given formula } \varphi? \]

⇔

**Problem**

Is \( \text{supp}(A) \neq \emptyset \) for a given wuta \( A \)?

**Definition (Support)**

A weighted automaton.

\[ \text{supp}(A) = \{ t \in U_{\Sigma} \mid \| A \|(t) \neq 0 \} \]
since the 80’s: support of probabilistic automata has been studied e.g. in books by Paz, Bukharaev
since the 80’s: support of probabilistic automata has been studied e.g. in books by Paz, Bukharaev

∃ weighted automata with non-recognizable support, e.g.:

$$\exists \begin{pmatrix} 1, \ell \mid 1, n \\ 1, \ell \mid 1, n \end{pmatrix}$$

in semiring $((\mathbb{Z}, +, \cdot, 0, 1))$

- $|w|_\ell - |w|_n = \|\mathcal{A}\|(w)$ recognizable, but
- $supp(\mathcal{A}) = \{w \in \{\ell, n\}^* | |w|_\ell \neq |w|_n\}$ not recognizable
since the 80’s: support of probabilistic automata has been studied e.g. in books by Paz, Bukharaev

∃ weighted automata with non-recognizable support, e.g.:

in semiring \((\mathbb{Z}, +, \cdot, 0, 1)\)

- \(|w|_\ell - |w|_n = \|A\|_\ell(w)\) recognizable, but
- \(\text{supp}(A) = \{w \in \{\ell, n\}^* \mid |w|_\ell \neq |w|_n\}\) not recognizable

effective construction of support automata for weighted automata over commutative, zero-sum free semirings (Kirsten)
Theorem

$\mathcal{M}$ wuta over zero-sum free, commutative semiring.

$\text{supp}(\mathcal{M}) = \{ t \in U_\Sigma \mid \|\mathcal{M}\|(t) \neq \emptyset \}$ is recognizable.
Theorem

\( \mathcal{M} \) wuta over zero-sum free, commutative semiring.

\[ \text{supp}(\mathcal{M}) = \{ t \in U_\Sigma \mid \|\mathcal{M}\|(t) \neq \emptyset \} \] is recognizable.

Proof:

\[ \mathcal{M} \]
Construction of Support Automaton $M_s$ for wuta $M$

**Theorem**

$M$ wuta over zero-sum free, commutative semiring.

$\text{supp}(M) = \{ t \in U_\Sigma \mid \|M\|(t) \neq \emptyset \}$ is recognizable.

**Proof:** Högberg, Maletti, Vogler (2009)

- $M$ weighted ranked tree automaton
- weighted ranked tree automaton simulating $M$
Construction of Support Automaton $M_s$ for wuta $M$

Theorem

$M$ wuta over zero-sum free, commutative semiring.

$$\text{supp}(M) = \{ t \in U_\Sigma \mid \|M\|(t) \neq \emptyset \}$$ is recognizable.

Proof:

Högberg, Maletti, Vogler (2009)

- $M$
- Weighted ranked tree automaton simulating $M$
- Ranked tree automaton recognizing support

Kirsten’s ideas for support automaton (2009)
Construction of Support Automaton $M_s$ for wuta $M$

**Theorem**

$M$ wuta over zero-sum free, commutative semiring.

$\text{supp}(M) = \{ t \in U_\Sigma \mid \|M\|(t) \neq \emptyset \}$ is recognizable.

**Proof:**

Högberg, Maletti, Vogler (2009)

- $M$
- Weighted ranked tree automaton simulating $M$
- Kirsten’s ideas for support automaton (2009)
- Unranked tree automaton recognizing support
- Ranked tree automaton recognizing support
Bollig, Gastin, Monmege, Zeitoun (ICALP 2010):

Let's consider the support of a series recognized by a pwA. For positive semirings, the latter problem, in turn, can be reduced to the decidable emptiness problem for classical pebble automata over the Boolean semiring. We leave it as an open problem to determine for which semirings the satisfiability problem is decidable.

**Unbounded transitive closure.** We do not know if allowing unbounded steps in the transitive closure leads beyond the power of (weak) pebble automata.
Bollig, Gastin, Monmege, Zeitoun (ICALP 2010):

the support of a series recognized by a pwA. For positive semirings, the latter problem, in turn, can be reduced to the decidable emptiness problem for classical pebble automata over the Boolean semiring. We leave it as an open problem to determine for which semirings the satisfiability problem is decidable.

Unbounded transitive closure. We do not know if allowing unbounded steps in the transitive closure leads beyond the power of (weak) pebble automata.

Is \( \text{supp}(\mathcal{M}) \neq \emptyset \) for nested weighted automaton \( \mathcal{M} \)?
Bollig, Gastin, Monmege, Zeitoun (ICALP 2010):

the support of a series recognized by a pwA. For positive semirings, the latter problem, in turn, can be reduced to the decidable emptiness problem for classical pebble automata over the Boolean semiring. We leave it as an open problem to determine for which semirings the satisfiability problem is decidable.

Unbounded transitive closure. We do not know if allowing unbounded steps in the transitive closure leads beyond the power of (weak) pebble automata. Unbounded transitive closure is harmful.

\[ \iff \]

Is \( \text{supp}(\mathcal{M}) \neq \emptyset \) for nested weighted automaton \( \mathcal{M} \)?

**Lemma (Droste, Heusel)**

*Wuta subsume nested weighted automata over commutative semirings*
Bollig, Gastin, Monmege, Zeitoun (ICALP 2010):

the support of a series recognized by a pwA. For positive semirings, the latter problem, in turn, can be reduced to the decidable emptiness problem for classical pebble automata over the Boolean semiring. We leave it as an open problem to determine for which semirings the satisfiability problem is decidable.

Unbounded transitive closure. We do not know if allowing unbounded steps in the transitive closure leads beyond the power of (weak) pebble automata.

Is $\text{supp}(M) \neq \emptyset$ for nested weighted automaton $M$?

Lemma (Droste, Heusel)

Wuta subsume nested weighted automata over commutative semirings

Corollary

Non-emptyness problem decidable
for nested weighted automata over zero-sum free, commutative semiring
Theorem

\[ \mathbb{D} \text{ cctv-semiring, } S : U_\Sigma \rightarrow \mathbb{D} \text{ tree series.} \]

\[ S = \|M\| \text{ for a wuta } M \Leftrightarrow S = \llbracket \varphi \rrbracket \text{ for a } \forall- \text{ and } \wedge- \text{restricted wMSO sentence } \varphi \]

→ solves Droste’s and Vogler’s equivalence problem

→ generalizes of respective results for:
  - s-wuta and wMSO over commutative semirings (Droste, Vogler)
  - wta and wMSO for ranked trees over tv-monoids (Droste et al.)
  - found model for unranked trees for average, discounting, semirings
  - four further equivalent results for other versions of the logic

Theorem

\( M \) wuta over zero-sum free, commutative semiring.

\[ \text{supp}(M) = \{ t \in U_\Sigma \mid \|M\|(t) \neq \emptyset \} \text{ is recognizable.} \]

→ partially solves satisfiability problem of Bollig et al.