A Weighted MSO Logic with Storage Behaviour and its Büchi-Elgot-Trakhtenbrot Theorem

Heiko Vogler[1] Manfred Droste[2] Luisa Herrmann[1]

[1]Department of Theoretical Computer Science, Technische Universität Dresden

[2]Institut für Informatik, University of Leipzig

March 14, 2016

BET-Theorem [Büchi 61,62, Elgot 61, Trakhtenbrot 61]

Let $L \subseteq \Sigma^*$. The following are equivalent:

1. L is $\mathrm{MSO}(\Sigma)$-definable.
2. L is recognizable.

BET-Theorem [Büchi 61,62, Elgot 61, Trakhtenbrot 61]

Let $L \subseteq \Sigma^*$. The following are equivalent:

1. L is $\mathrm{MSO}(\Sigma)$-definable.
2. L is recognizable.

generalizations in 3 directions:

- form of models (timed words, trees, pictures,...)
- unweighted vs. weighted
- beyond recognizability

BET-Theorems for weighted languages

unweighted	string languages $L \subseteq \Sigma^*$
	[BET 61,62]
weight algebra K	weighted languages $r\colon \Sigma^* \to K$

weighted language = formal power series

BET-Theorems for weighted languages

unweighted	string languages $L \subseteq \Sigma^*$
	[BET 61,62]
weight algebra K	**weighted languages $r\colon \Sigma^* \to K$**
semirings	[Droste, Gastin 05,07,09]

weighted language = formal power series

BET-Theorems for weighted languages

unweighted	string languages $L \subseteq \Sigma^*$
	[BET 61,62]
weight algebra K	**weighted languages $r\colon \Sigma^* \to K$**
semirings	[Droste, Gastin 05,07,09]
bounded lattices	[Droste, Vogler 12]

weighted language = formal power series

BET-Theorems for weighted languages

unweighted	string languages $L \subseteq \Sigma^*$
	[BET 61,62]

weight algebra K	weighted languages $r\colon \Sigma^* \to K$
semirings	[Droste, Gastin 05,07,09]
bounded lattices	[Droste, Vogler 12]
valuation monoids	[Droste, Meinecke 12]

weighted language = formal power series

BET-Theorems beyond recognizability

BET for cf languages [Schwentick, Lautemann, Therien 94]

Let $L \subseteq \Sigma^*$. The following are equivalent:

1. L is definable by $\exists M.\varphi$
   where M is matching, φ is MSO formula.
2. L is a context-free language.

BET-Theorems beyond recognizability

BET for cf languages [Schwentick, Lautemann, Therien 94]

Let $L \subseteq \Sigma^*$. The following are equivalent:

1. L is definable by $\exists M.\varphi$
   where M is matching, φ is MSO formula.
2. L is a context-free language.

BET for indexed languages [Fratani, Voundy 15]

Let $L \subseteq \Sigma^*$. The following are equivalent:

1. L is definable by $\exists DM.\varphi$
   where DM is Dyck matching, φ is MSO formula.
2. L is a realtime indexed language.

Idea of our new logic

languages	automata	logic
regular languages	finite automata	φ
context-free languages	pushdown automata	$\exists M.\varphi$
indexed languages	nested stack automata	$\exists DM.\varphi$

Idea of our new logic

languages	automata	logic
regular languages	finite automata	φ
context-free languages	pushdown automata	$\exists M.\varphi$
indexed languages	nested stack automata	$\exists DM.\varphi$
	automata with storage S	

Idea of our new logic

languages	automata	logic
regular languages	finite automata	φ
context-free languages	pushdown automata	$\exists M.\varphi$
indexed languages	nested stack automata	$\exists DM.\varphi$
	automata with storage S	
	finitely encoded AFA	

AFA = abstract family of acceptors

Idea of our new logic

languages	automata	logic
regular languages	finite automata	φ
context-free languages	pushdown automata	$\exists M.\varphi$
indexed languages	nested stack automata	$\exists DM.\varphi$
	automata with storage S	
full principal AFL	finitely encoded AFA	

AFA = abstract family of acceptors
AFL = abstract family of languages

Idea of our new logic

languages	automata	logic
regular languages	finite automata	φ
context-free languages	pushdown automata	$\exists M.\varphi$
indexed languages	nested stack automata	$\exists DM.\varphi$
	automata with storage S	
full principal AFL	finitely encoded AFA	

generator language:
set B of behaviours of S

AFA = abstract family of acceptors
AFL = abstract family of languages

Idea of our new logic

languages	automata	logic
regular languages	finite automata	φ
context-free languages	pushdown automata	$\exists M.\varphi$
indexed languages	nested stack automata	$\exists DM.\varphi$
	automata with storage S	$\exists B.\varphi$
full principal AFL	finitely encoded AFA	

generator language:
set B of behaviours of S

AFA = abstract family of acceptors
AFL = abstract family of languages

Idea of our new logic

languages	automata	logic
regular languages	finite automata	φ
context-free languages	pushdown automata	$\exists M.\varphi$
indexed languages	nested stack automata	$\exists DM.\varphi$
	automata with storage S	$\exists B.\varphi$
full principal AFL	finitely encoded AFA	
weighted languages		
	weighted automata with storage	$\sum_{B}^{\mathrm{beh}} e$

generator language:
set B of behaviours of S

AFA = abstract family of acceptors
AFL = abstract family of languages

Goal

Theorem [Vogler, Droste, Herrmann LATA '16]

Let K be a unital valuation monoid, S a storage type, and $r\colon \Sigma^* \to K$. The following are equivalent:

1. r is definable by an expression over (S, Σ, K).
2. r is (S, Σ, K)-recognizable.

Goal

Theorem [Vogler, Droste, Herrmann LATA '16]

Let K be a unital valuation monoid, S a storage type, and $r\colon \Sigma^* \to K$. The following are equivalent:

1. r is definable by an expression over (S, Σ, K).
2. r is (S, Σ, K)-recognizable.

outline of the talk:

- weight structure: unital valuation monoid
- storage type
- weighted automata with storage
- weighted MSO expressions
- results

Unital Valuation Monoid

$(K, +, \mathrm{val}, 0, 1)$ [Droste, Meinecke 12]

- $(K, +, 0)$ commutative monoid
- $\mathrm{val}\colon K^* \to K$
  - $\mathrm{val}(a) = a$
  - $\mathrm{val}(\dots 0 \dots) = 0$
  - $\mathrm{val}(\dots 1 \dots) = \mathrm{val}(\dots\dots)$ ignore "1"s
  - $\mathrm{val}(\varepsilon) = 1$

Unital Valuation Monoid

$(K, +, \mathrm{val}, 0, 1)$ [Droste, Meinecke 12]

- $(K, +, 0)$ commutative monoid
- $\mathrm{val}\colon K^* \to K$
  - $\mathrm{val}(a) = a$
  - $\mathrm{val}(\dots 0 \dots) = 0$
  - $\mathrm{val}(\dots 1 \dots) = \mathrm{val}(\dots\dots)$ ignore "1"s
  - $\mathrm{val}(\varepsilon) = 1$

Examples

1) $\mathrm{Disc} = (\mathbb{R}_{\geq 0} \cup \{-\infty\}, \max, \mathrm{val}_{\mathrm{disc}}, -\infty, 0)$

$$\mathrm{val}_{\mathrm{disc}}(k_1 \dots k_n) = 0.5^0 \cdot k_1 + \dots + 0.5^{n-1} \cdot k_n$$

Unital Valuation Monoid

$(K, +, \mathrm{val}, 0, 1)$ [Droste, Meinecke 12]

- $(K, +, 0)$ commutative monoid
- $\mathrm{val}\colon K^* \to K$
  - $\mathrm{val}(a) = a$
  - $\mathrm{val}(\dots 0 \dots) = 0$
  - $\mathrm{val}(\dots 1 \dots) = \mathrm{val}(\dots\dots)$ ignore "1"s
  - $\mathrm{val}(\varepsilon) = 1$

Examples

1) $\mathrm{Disc} = (\mathbb{R}_{\geq 0} \cup \{-\infty\}, \max, \mathrm{val}_{\mathrm{disc}}, -\infty, 0)$

$$\mathrm{val}_{\mathrm{disc}}(k_1 \dots k_n) = 0.5^0 \cdot k_1 + \dots + 0.5^{n-1} \cdot k_n$$

2) strong bimonoid $(K, +, \cdot, 0, 1)$
   - $(K, +, 0)$ commutative monoid
   - $(K, \cdot, 1)$ monoid $\mathrm{val}(a_1 \dots a_n) = a_1 \cdot \dots \cdot a_n$
   - 0 absorbing for $\cdot$

e.g., semirings, bounded lattices

Storage type

[Scott 67, Engelfriet 86]

$S = (C, P, F, c_0)$

- C set (configurations)
- P set of functions $p\colon C \to \{\text{true}, \text{false}\}$ (predicates)
- F set of partial functions $f\colon C \to C$ (instructions)
- c_0 element of C (initial configuration)

Storage type

[Scott 67, Engelfriet 86]

$S = (C, P, F, c_0)$

- C set (configurations)
- P set of functions $p \colon C \to \{\text{true}, \text{false}\}$ (predicates)
- F set of partial functions $f \colon C \to C$ (instructions)
- c_0 element of C (initial configuration)

Example

$\mathrm{MON}(M) = (C, P, F, c_0)$ ($(M, \cdot, 1)$ monoid)

- $C = M, \quad c_0 = 1,$
- $P = \{\top?\} \cup \{1?\},$
- $F = M \qquad m(m') = m' \cdot m$

Weighted Automata with Storage

$S = (C, P, F, c_0)$ storage type
Σ alphabet
K unital valuation monoid

$A = (Q, Q_0, Q_f, T, \mathrm{wt})$ (S, Σ, K)-automaton

- Q finite set (states)
- $Q_0 \subseteq Q$ (initial states)
- $Q_f \subseteq Q$ (final states)
- T finite set (transitions)

$$\tau = (q_1, a, p, q_2, f)$$

$q_1, q_2 \in Q, \quad a \in \Sigma, \quad p \in P, \quad f \in F$

- $\mathrm{wt}\colon T \to K$ (weight assignment)

Weighted Automata with Storage

$S = (C, P, F, c_0)$ storage type
Σ alphabet
K unital valuation monoid

$A = (Q, Q_0, Q_f, T, \mathrm{wt})$ (S, Σ, K)-automaton

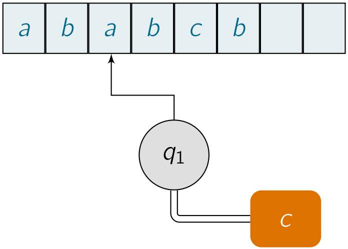
$$\tau = (q_1, a, p, q_2, f)$$

Weighted Automata with Storage

$S = (C, P, F, c_0)$ storage type
Σ alphabet
K unital valuation monoid

$A = (Q, Q_0, Q_f, T, \mathrm{wt})$ (S, Σ, K)-automaton

$$\tau = (q_1, a, p, q_2, f)$$

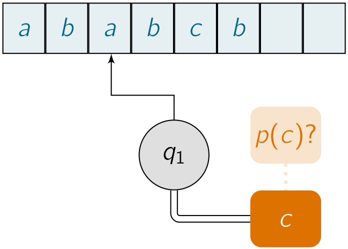


Weighted Automata with Storage

$S = (C, P, F, c_0)$ storage type
Σ alphabet
K unital valuation monoid

$A = (Q, Q_0, Q_f, T, \mathrm{wt})$ (S, Σ, K)-automaton

$$\tau = (q_1, a, p, q_2, f)$$

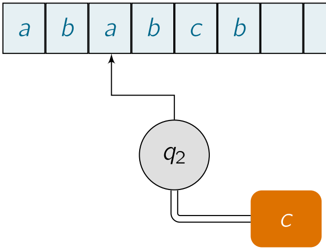


Weighted Automata with Storage

$S = (C, P, F, c_0)$ storage type
Σ alphabet
K unital valuation monoid

$A = (Q, Q_0, Q_f, T, \mathrm{wt})$ (S, Σ, K)-automaton

$\tau = (q_1, a, p, q_2, f)$

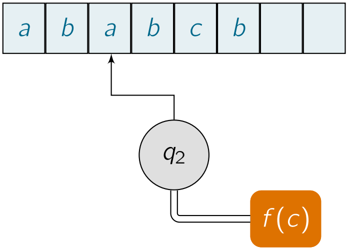


Weighted Automata with Storage

$S = (C, P, F, c_0)$ storage type
Σ alphabet
K unital valuation monoid

$A = (Q, Q_0, Q_f, T, \mathrm{wt})$ (S, Σ, K)-automaton

$\tau = (q_1, a, p, q_2, f)$

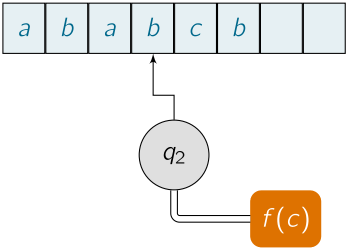


Weighted Automata with Storage

$S = (C, P, F, c_0)$ storage type
Σ alphabet
K unital valuation monoid

$A = (Q, Q_0, Q_f, T, \mathrm{wt})$ (S, Σ, K)-automaton

$$\tau = (q_1, a, p, q_2, f)$$



Weighted Automata with Storage

$w = a_1 \ldots a_n \in \Sigma^*$
$\theta = \tau_1 \ldots \tau_n, \quad \tau_i \in T$
$q_f \in Q_f$

computation:

$$(q_0, a_1 a_2 \ldots a_n, c_0) \vdash^{\tau_1} (q_1, a_2 \ldots a_n, c_1) \vdash^{\tau_2} \ldots \vdash^{\tau_n} (q_f, \varepsilon, c_n)$$

Weighted Automata with Storage

$w = a_1 \dots a_n \in \Sigma^*$
$\theta = \tau_1 \dots \tau_n, \quad \tau_i \in T$
$q_f \in Q_f$

computation:

$$(q_0, a_1 a_2 \dots a_n, c_0) \vdash^{\tau_1} (q_1, a_2 \dots a_n, c_1) \vdash^{\tau_2} \dots \vdash^{\tau_n} (q_f, \varepsilon, c_n)$$

$\Theta(w)$

Weighted Automata with Storage

$w = a_1 \dots a_n \in \Sigma^*$
$\theta = \tau_1 \dots \tau_n, \quad \tau_i \in T$
$q_f \in Q_f$

computation:

$$(q_0, a_1 a_2 \dots a_n, c_0) \vdash^{\tau_1} (q_1, a_2 \dots a_n, c_1) \vdash^{\tau_2} \dots \vdash^{\tau_n} (q_f, \varepsilon, c_n)$$

$\Theta(w)$

weighted language recognized by A: $\|A\| \colon \Sigma^* \to K$

$$\|A\|(w) = \sum_{\theta \in \Theta(w)} \mathrm{wt}(\theta)$$

$$\mathrm{wt}(\tau_1 \dots \tau_n) = \mathrm{val}(\mathrm{wt}(\tau_1) \dots \mathrm{wt}(\tau_n))$$

Weighted Automata with Storage

$\Sigma = \{a, b, \#\}$
$\mathrm{MON}(\mathbb{Z})$, $(\mathbb{Z}, +, 0)$
Disc

$r\colon \Sigma^* \to \mathrm{Disc}$

$\mathrm{supp}(r) = \{w\# \mid w \in \{a, b\}^*, |w|_a = |w|_b\}$

Weighted Automata with Storage

$\Sigma = \{a, b, \#\}$
$\mathrm{MON}(\mathbb{Z})$, $(\mathbb{Z}, +, 0)$
Disc

$r \colon \Sigma^* \to \mathrm{Disc}$

$\mathrm{supp}(r) = \{w\# \mid w \in \{a, b\}^*, |w|_a = |w|_b\}$

$$a\ b\ b\ a\ \# \quad \in \quad \mathrm{supp}(r)$$

Weighted Automata with Storage

$\Sigma = \{a, b, \#\}$
$\mathrm{MON}(\mathbb{Z}),\ (\mathbb{Z}, +, 0)$
Disc

$r\colon \Sigma^* \to \mathrm{Disc}$

$\mathrm{supp}(r) = \{w\# \mid w \in \{a, b\}^*, |w|_a = |w|_b\}$

$$r(a\ b\ b\ a\ \#)$$

Weighted Automata with Storage

$\Sigma = \{a, b, \#\}$
$\mathrm{MON}(\mathbb{Z}),\ (\mathbb{Z}, +, 0)$
Disc

$r\colon \Sigma^* \to \mathrm{Disc}$

$\mathrm{supp}(r) = \{w\# \mid w \in \{a, b\}^*, |w|_a = |w|_b\}$

$r(a\ b\ b\ a\ \#)$

a	2
b	1
$\#$	0

Weighted Automata with Storage

$\Sigma = \{a, b, \#\}$
$\mathrm{MON}(\mathbb{Z}),\ (\mathbb{Z}, +, 0)$
Disc

$r\colon \Sigma^* \to \mathrm{Disc}$

$\mathrm{supp}(r) = \{w\# \mid w \in \{a, b\}^*, |w|_a = |w|_b\}$

$r(a\ b\ b\ a\ \#)$ 2 1 1 2 0

a	2
b	1
$\#$	0

Weighted Automata with Storage

$\Sigma = \{a, b, \#\}$
$\mathrm{MON}(\mathbb{Z}),\ (\mathbb{Z}, +, 0)$
Disc

$r\colon \Sigma^* \to \mathrm{Disc}$

$\mathrm{supp}(r) = \{w\# \mid w \in \{a, b\}^*, |w|_a = |w|_b\}$

$$r(a\ b\ b\ a\ \#) = \mathrm{val}_{\mathrm{disc}}(2\ 1\ 1\ 2\ 0)$$

a	2
b	1
$\#$	0

Weighted Automata with Storage

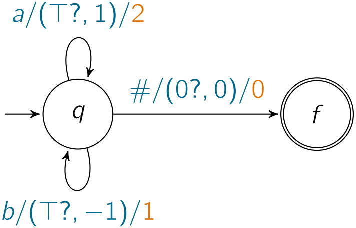
$\Sigma = \{a, b, \#\}$
$\mathrm{MON}(\mathbb{Z}),\ (\mathbb{Z}, +, 0)$
Disc

$r \colon \Sigma^* \to \mathrm{Disc}$

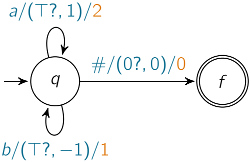
$\mathrm{supp}(r) = \{w\# \mid w \in \{a, b\}^*, |w|_a = |w|_b\}$

$$r(a\ b\ b\ a\ \#) = \mathrm{val}_{\mathrm{disc}}(2\ 1\ 1\ 2\ 0)$$

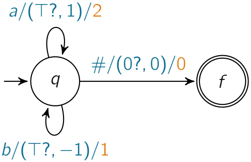
a	2
b	1
$\#$	0



Storage Behaviour



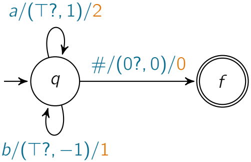
Storage Behaviour



computation

$$(q, ab\#, 0) \vdash^{\tau_1} (q, b\#, 1) \vdash^{\tau_2} (q, \#, 0) \vdash^{\tau_3} (f, \varepsilon, 0)$$

Storage Behaviour



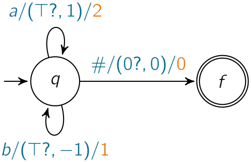
computation

$$(q, ab\#, 0) \vdash^{\tau_1} (q, b\#, 1) \vdash^{\tau_2} (q, \#, 0) \vdash^{\tau_3} (f, \varepsilon, 0)$$

used transitions

$$(q, a, \top?, q, 1) \;\; (q, b, \top?, q, -1) \;\; (q, \#, 0?, f, 0)$$

Storage Behaviour



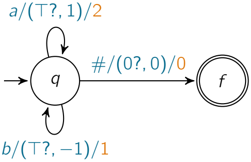
computation

$$(q, ab\#, 0) \vdash^{\tau_1} (q, b\#, 1) \vdash^{\tau_2} (q, \#, 0) \vdash^{\tau_3} (f, \varepsilon, 0)$$

used transitions

$$(q, a, \top?, q, 1) \quad (q, b, \top?, q, -1) \quad (q, \#, 0?, f, 0)$$

Storage Behaviour



computation

$$(q, ab\#, 0) \vdash^{\tau_1} (q, b\#, 1) \vdash^{\tau_2} (q, \#, 0) \vdash^{\tau_3} (f, \varepsilon, 0)$$

used transitions

$$(q, a, \top?, q, 1) \quad (q, b, \top?, q, -1) \quad (q, \#, 0?, f, 0)$$

$\Omega \subseteq_{\text{fin}} P \times F$

storage behaviour

$$(\top?, 1)\,(\top?, -1)\,(0?, 0) \in \mathrm{B}(\Omega)$$

Weighted MSO Logic with Storage

Weighted MSO Logic with Storage

variable assignment σ:

	a	b	b	a
f.o. variables		x_1		x_2
s.o. variables	X_1	X_1		X_2
s.o. behaviour variable B	$(\top?, 1)$	$(\top?, -1)$	$(\top?, -1)$	$(\top?, 1)$

Weighted MSO Logic with Storage

$\Omega \subseteq_{\text{fin}} P \times F$

variable assignment σ:

	a	b	b	a
f.o. variables		x_1		x_2
s.o. variables	X_1	X_1		X_2
s.o. behaviour variable B	$(\top?, 1)$	$(\top?, -1)$	$(\top?, -1)$	$(\top?, 1)$
		$\in \Sigma_{\mathcal{V}}$		

Weighted MSO Logic with Storage

$\Omega \subseteq_{\text{fin}} P \times F$

unweighted MSO

$$\psi ::= \mathrm{P}_a(x) \mid \mathrm{next}(x, y) \mid x \in X \mid B(x) = (p, f)$$

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x.\varphi \mid \exists X.\varphi \mid \forall x.\varphi \mid \forall X.\varphi$$

Weighted MSO Logic with Storage

$\Omega \subseteq_{\text{fin}} P \times F$

unweighted MSO

$$\psi ::= \mathrm{P}_a(x) \mid \mathrm{next}(x, y) \mid x \in X \mid B(x) = (p, f)$$

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x.\varphi \mid \exists X.\varphi \mid \forall x.\varphi \mid \forall X.\varphi$$

$$(u, \sigma) \models (B(x) = (p, f)) \text{ iff } \sigma(B)_{\sigma(x)} = (p, f)$$

variable assignment σ:

a	b	b	a
	x_1		x_2
X_1	X_1		X_2
$(\top?, 1)$	$(\top?, -1)$	$(\top?, -1)$	$(\top?, 1)$

$(abba, \sigma) \models (B(x_2) = (\top?, 1))$

Weighted MSO Logic with Storage

$\Omega \subseteq_{\mathrm{fin}} P \times F$

unweighted MSO

$$\psi ::= \mathrm{P}_a(x) \mid \mathrm{next}(x, y) \mid x \in X \mid B(x) = (p, f)$$

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x.\varphi \mid \exists X.\varphi \mid \forall x.\varphi \mid \forall X.\varphi$$

Weighted MSO Logic with Storage

$\Omega \subseteq_{\mathrm{fin}} P \times F$

unweighted MSO

$$\psi ::= \mathrm{P}_a(x) \mid \mathrm{next}(x, y) \mid x \in X \mid B(x) = (p, f)$$

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x.\varphi \mid \exists X.\varphi \mid \forall x.\varphi \mid \forall X.\varphi$$

weighted B-expressions $\mathrm{BExp}(S, \Sigma, K)$ [Fülöp, Stüber, Vogler 10]

$$e ::= \mathrm{Val}_\kappa \mid e + e \mid \varphi \triangleright e \mid \sum_x e \mid \sum_X e$$

Weighted MSO Logic with Storage

$\Omega \subseteq_{\text{fin}} P \times F$

unweighted MSO

$$\psi ::= \mathrm{P}_a(x) \mid \mathrm{next}(x, y) \mid x \in X \mid B(x) = (p, f)$$

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x.\varphi \mid \exists X.\varphi \mid \forall x.\varphi \mid \forall X.\varphi$$

weighted B-expressions $\mathrm{BExp}(S, \Sigma, K)$ [Fülöp, Stüber, Vogler 10]

$$e ::= \mathrm{Val}_\kappa \mid e + e \mid \varphi \triangleright e \mid \sum_x e \mid \sum_X e$$

$[\![e]\!]_\mathcal{V} \colon \Sigma_\mathcal{V}{}^* \to K$ such that

Weighted MSO Logic with Storage

$\Omega \subseteq_{\text{fin}} P \times F$

unweighted MSO

$$\psi ::= \mathrm{P}_a(x) \mid \mathrm{next}(x, y) \mid x \in X \mid B(x) = (p, f)$$

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x.\varphi \mid \exists X.\varphi \mid \forall x.\varphi \mid \forall X.\varphi$$

weighted B-expressions $\mathrm{BExp}(S, \Sigma, K)$ [Fülöp, Stüber, Vogler 10]

$$e ::= \mathrm{Val}_\kappa \mid e + e \mid \varphi \rhd e \mid \sum_x e \mid \sum_X e$$

$[\![e]\!]_\mathcal{V} \colon {\Sigma_\mathcal{V}}^* \to K$ such that

$\kappa \colon \Sigma_\mathcal{V} \to K$

$$[\![\mathrm{Val}_\kappa]\!]_\mathcal{V}(w) = \mathrm{val}(\kappa(w))$$

Weighted MSO Logic with Storage

$\Omega \subseteq_{\text{fin}} P \times F$

unweighted MSO

$$\psi ::= \mathrm{P}_a(x) \mid \mathrm{next}(x, y) \mid x \in X \mid B(x) = (p, f)$$

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x.\varphi \mid \exists X.\varphi \mid \forall x.\varphi \mid \forall X.\varphi$$

weighted B-expressions $\mathrm{BExp}(S, \Sigma, K)$ [Fülöp, Stüber, Vogler 10]

$$e ::= \mathrm{Val}_\kappa \mid e + e \mid \varphi \rhd e \mid \sum_x e \mid \sum_X e$$

$[\![e]\!]_\mathcal{V} \colon \Sigma_\mathcal{V}{}^* \to K$ such that

$$[\![\varphi \rhd e]\!]_\mathcal{V}(w) = \begin{cases} [\![e]\!]_\mathcal{V}(w) & \text{if } w \in \mathcal{L}_\mathcal{V}(\varphi) \\ 0 & \text{otherwise} \end{cases}$$

Weighted MSO Logic with Storage

$\Omega \subseteq_{\text{fin}} P \times F$

unweighted MSO

$$\psi ::= \mathrm{P}_a(x) \mid \mathrm{next}(x, y) \mid x \in X \mid B(x) = (p, f)$$

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x.\varphi \mid \exists X.\varphi \mid \forall x.\varphi \mid \forall X.\varphi$$

weighted B-expressions $\mathrm{BExp}(S, \Sigma, K)$

$$e ::= \mathrm{Val}_\kappa \mid e + e \mid \varphi \rhd e \mid \sum_x e \mid \sum_X e$$

weighted expressions $\mathrm{Exp}(S, \Sigma, K)$

$$\sum_B^{\mathrm{beh}} e$$

Weighted MSO Logic with Storage

$\Omega \subseteq_{\text{fin}} P \times F$

weighted expressions $\mathrm{Exp}(S, \Sigma, K)$

$$\sum_B^{\mathrm{beh}} e$$

Weighted MSO Logic with Storage

weighted expressions $\mathrm{Exp}(S, \Sigma, K)$

$$\textstyle\sum_B^{\text{beh}} e$$

$[\![\textstyle\sum_B^{\text{beh}} e]\!] \colon \Sigma^* \to K$

$$[\![\sum_B^{\text{beh}} e]\!](u) = \sum_{b \in \mathrm{B}(\Omega)} [\![e]\!]_{\{B\}}\big(u, [B \mapsto b]\big)$$

Weighted MSO Logic with Storage

weighted expressions $\mathrm{Exp}(S, \Sigma, K)$

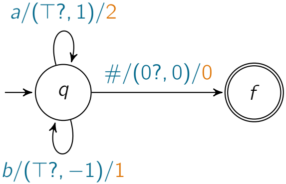
$$\sum\nolimits_B^{\text{beh}} e$$

$[\![\sum\nolimits_B^{\text{beh}} e]\!] \colon \Sigma^* \to K$

$$[\![\sum\nolimits_B^{\text{beh}} e]\!](u) = \sum_{b \in \mathrm{B}(\Omega)} [\![e]\!]_{\{B\}}\big(u, [B \mapsto b]\big)$$

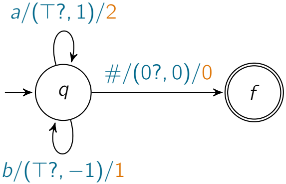
$r \colon \Sigma^* \to K$ is definable by an expression over (S, Σ, K) if there is $e \in \mathrm{Exp}(S, \Sigma, K)$ such that $r = [\![e]\!]$.

Example



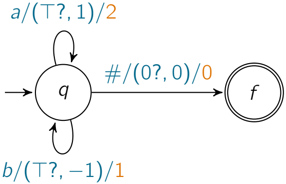
Example

$$\Omega: \quad \omega_1 = (\top?, 1)$$
$$\omega_2 = (\top?, -1)$$
$$\omega_3 = (0?, 0)$$



Example

$$\Omega: \; \omega_1 = (\top?, 1)$$
$$\omega_2 = (\top?, -1)$$
$$\omega_3 = (0?, 0)$$



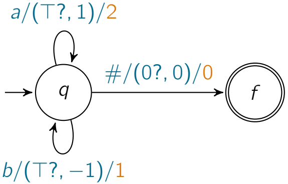
$$e = \sum\nolimits_B^{\mathrm{beh}}((\varphi_{\mathrm{last}} \wedge \varphi_{\mathrm{symb}}) \triangleright \mathrm{Val}_\kappa)$$

Example

$\Omega: \ \omega_1 = (\top?, 1)$

$\omega_2 = (\top?, -1)$

$\omega_3 = (0?, 0)$

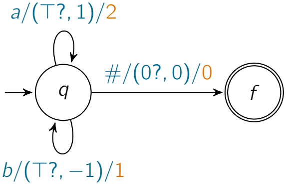


$$e = \sum_B^{\mathrm{beh}}((\varphi_{\mathrm{last}} \wedge \varphi_{\mathrm{symb}}) \triangleright \mathrm{Val}_\kappa)$$

	a	b	$\#$
B:	$(\top?, 1)$	$(\top?, -1)$	$(0?, 0)$

Example

$$\Omega: \ \omega_1 = (\top?, 1)$$
$$\omega_2 = (\top?, -1)$$
$$\omega_3 = (0?, 0)$$



$$e = \sum_B^{\mathrm{beh}} ((\varphi_{\mathrm{last}} \wedge \varphi_{\mathrm{symb}}) \triangleright \mathrm{Val}_\kappa)$$

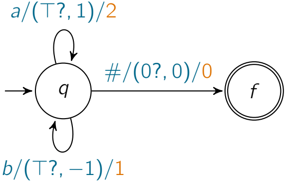
	a	b	$\#$
B:	$(\top?, 1)$	$(\top?, -1)$	$(0?, 0)$

Example

$\Omega\colon\ \omega_1 = (\top?, 1)$

$\omega_2 = (\top?, -1)$

$\omega_3 = (0?, 0)$



$$e = \sum_B^{\mathrm{beh}}((\varphi_{\mathrm{last}} \wedge \varphi_{\mathrm{symb}}) \triangleright \mathrm{Val}_\kappa)$$

$B:$

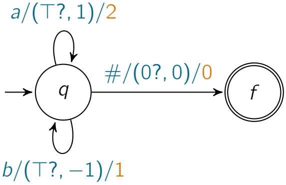
a	b	$\#$
$(\top?, 1)$	$(\top?, -1)$	$(0?, 0)$

Example

Ω: $\omega_1 = (\top?, 1)$

$\omega_2 = (\top?, -1)$

$\omega_3 = (0?, 0)$

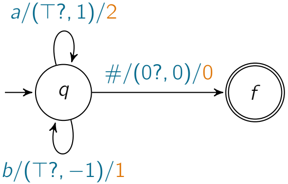


$$e = \sum_B^{\mathrm{beh}} ((\varphi_{\mathrm{last}} \wedge \varphi_{\mathrm{symb}}) \triangleright \mathrm{Val}_\kappa)$$

	a	b	$\#$	
B:	$(\top?, 1)$	$(\top?, -1)$	$(0?, 0)$	
B:	$(\top?, -1)$	$(\top?, 1)$	$(0?, 0)$	$\not\models \varphi_{\mathrm{symb}}$

Example

$$\Omega: \ \omega_1 = (\top?, 1)$$
$$\omega_2 = (\top?, -1)$$
$$\omega_3 = (0?, 0)$$



$$e = \sum_B^{\mathrm{beh}} ((\varphi_{\mathrm{last}} \wedge \varphi_{\mathrm{symb}}) \triangleright \mathrm{Val}_\kappa)$$

	a	b	$\#$
B:	$(\top?, 1)$	$(\top?, -1)$	$(0?, 0)$
κ:	2	1	0

BET-Theorem

Theorem [Vogler, Droste, Herrmann 16]

Let K be a unital valuation monoid, S a storage type, and $r\colon \Sigma^* \to K$. The following are equivalent:

1. r is definable by an expression over (S, Σ, K).
2. r is (S, Σ, K)-recognizable.

BET-Theorem

Theorem [Vogler, Droste, Herrmann 16]

Let K be a unital valuation monoid, S a storage type, and $r\colon \Sigma^* \to K$. The following are equivalent:

1. r is definable by an expression over (S, Σ, K).
2. r is (S, Σ, K)-recognizable.

Theorem [Vogler, Droste, Herrmann 16]

For each $\varphi \in \mathrm{Exp}(\mathrm{P}^n, \Sigma, K)$ the satisfiability problem is decidable if K is a zero-sum-free commutative strong bimonoid with a decidable zero-generation problem.

BET-Theorem

Theorem [Vogler, Droste, Herrmann 16]

Let K be a unital valuation monoid, S a storage type, and $r\colon \Sigma^* \to K$. The following are equivalent:

1. r is definable by an expression over (S, Σ, K).
2. r is (S, Σ, K)-recognizable.

Theorem [Vogler, Droste, Herrmann 16]

For each $\varphi \in \mathrm{Exp}(\mathrm{P}^n, \Sigma, K)$ the satisfiability problem is decidable if K is a zero-sum-free commutative strong bimonoid with a decidable zero-generation problem.

Thank you!

References I

- J. Büchi (1960). Weak second-order arithmetic and finite automata. *Zeitschr. für math. Logik und Grundl. der Mathem. 6*, pp. 66–92.
- J. Büchi (1962). On a decision method in restricted second-order arithmetics. *Proc. of 1960 International Congress for Logic, Methodology and Philosophy of Science*, pp. 1–11.
- C. Elgot (1961). Decision problems of finite automata design and related arithmetics. *Trans. Am. Math. Soc. 98*, pp. 21–52.
- B. Trakhtenbrot (1961). Finite automata and logic of monadic predicates. *Doklady Akademii Nauk SSSR 149*, pp. 326–329. in Russian
- M. Droste and P. Gastin (2007). Weighted automata and weighted logics. *Droste,M., Kuich, W., Vogler, H. (eds.) Handbook of Weighted Automata.*
- Z. Fülöp and T. Stüber and H. Vogler, H (2012). A Büchi-like theorem for weighted tree automata over multioperator monoids. *Theory Comput. Syst.50(2)*, pp. 241–278.
- C. Lautemann and T. Schwentick and D. Therien (1995). Logics for context-free languages. *LNCS 933*, pp. 205–261.

References II

- S. Fratani and E.M. Voundy (2015). Dyck-based characterizations of indexed languages. published on arXiv http://arxiv.org/abs/1409.6112, March 13, 2015.
- D. Scott (1967). Some definitional suggestions for automata theory. *Journal of Computer and System Sciences 1.2*, pp. 187–212.
- J. Engelfriet (1986). Context-Free Grammars with Storage. *CoRR abs/1408.0683*