Axiomatising Weighted Monadic Second-Order Logic on Finite Words

Antonis Achilleos and Mathias Ruggaard Pedersen

ICE-TCS, Department of Computer Science, Reykjavík University

NWPT, November 13, 2019

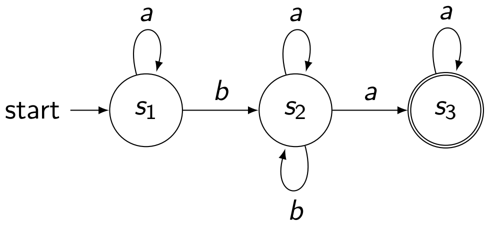
Agenda

1. Weighted automata
2. Weighted monadic second-order logic
3. Axiomatisation
4. Decision problems
5. Conclusion

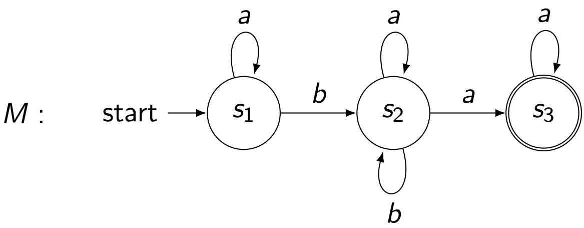
Weighted automata

Finite automata to weighted automata

M :

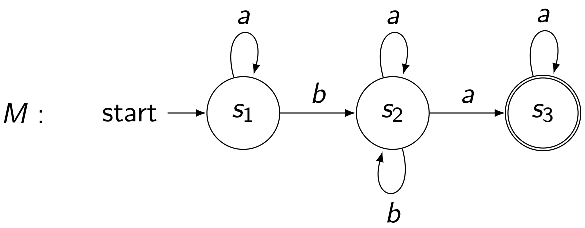


Finite automata to weighted automata



$$bbaa \in L(M)$$

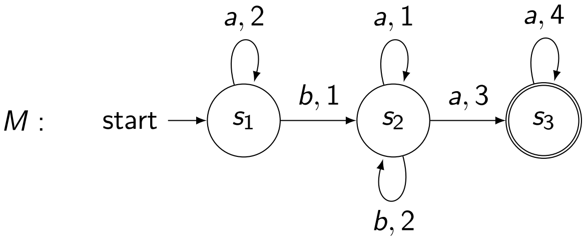
Finite automata to weighted automata



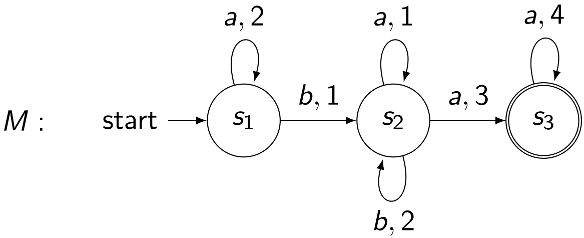
$$bbaa \in L(M)$$

$$abba \in L(M)$$

Finite automata to weighted automata

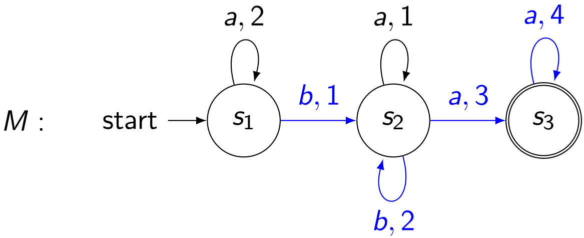


Finite automata to weighted automata



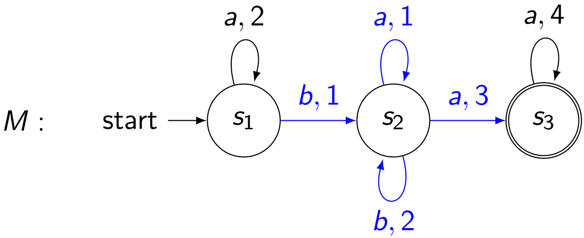
$$bbaa \mapsto (1 \times 2 \times 3 \times 4) + (1 \times 2 \times 1 \times 3)$$

Finite automata to weighted automata



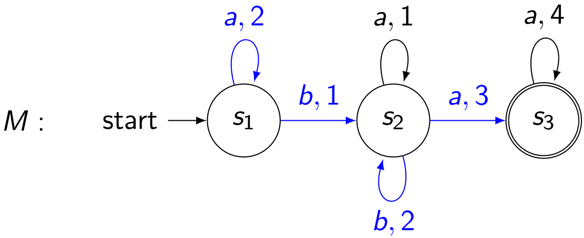
$$bbaa \mapsto (1 \times 2 \times 3 \times 4) + (1 \times 2 \times 1 \times 3)$$

Finite automata to weighted automata



$$bbaa \mapsto (1 \times 2 \times 3 \times 4) + (1 \times 2 \times 1 \times 3)$$

Finite automata to weighted automata



$$bbaa \mapsto (1 \times 2 \times 3 \times 4) + (1 \times 2 \times 1 \times 3)$$

$$abba \mapsto (2 \times 1 \times 2 \times 3)$$

Semirings

Weights are taken from a semiring.

Definition

A semiring is a tuple $(X, +, \times, 0, 1)$ such that

- $(X, +, 0)$ is a commutative monoid,
- $(X, \times, 1)$ is a monoid,
- $\times$ distributes over $+$, and
- 0 is absorbing for $\times$.

Semirings

Weights are taken from a semiring.

Definition

A semiring is a tuple $(X, +, \times, 0, 1)$ such that

- $(X, +, 0)$ is a commutative monoid,
- $(X, \times, 1)$ is a monoid,
- $\times$ distributes over $+$, and
- 0 is absorbing for $\times$.

Examples:

- $(\mathbb{Z}, +, \times, 0, 1)$
- $(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$
- $(\mathbb{R} \cup \{+\infty\}, \min, +, +\infty, 0)$

Büchi-Elgot-Trakhtenbrot theorem

Theorem (Büchi, Elgot, Trakhtenbrot)

MSO and (non-weighted) automata are expressively equivalent.

Weighted monadic second-order logic

Weighted MSO and FO

$$\varphi ::= \top \mid P_a(x) \mid x \leq y \mid x \in X \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \forall x.\varphi \mid \forall X.\varphi \qquad \text{(MSO)}$$

$$\Psi ::= r \mid \varphi \,?\, \Psi_1 : \Psi_2 \qquad \text{(step-wMSO)}$$

$$\Phi ::= \mathbf{0} \mid \textstyle\prod_x \Psi \mid \varphi \,?\, \Phi_1 : \Phi_2 \mid \Phi_1 + \Phi_2 \mid \sum_x \Phi \mid \sum_X \Phi \qquad \text{(core-wMSO)}$$

Weighted MSO and FO

$$\varphi ::= \top \mid P_a(x) \mid x \leq y \mid \cancel{x \in X} \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \forall x.\varphi \mid \cancel{\forall X.\varphi} \qquad \text{(FO)}$$

$$\Psi ::= r \mid \varphi \,?\, \Psi_1 : \Psi_2 \qquad \text{(step-wFO)}$$

$$\Phi ::= \mathbf{0} \mid \prod_x \Psi \mid \varphi \,?\, \Phi_1 : \Phi_2 \mid \Phi_1 + \Phi_2 \mid \sum_x \Phi \mid \cancel{\sum_X \Phi} \qquad \text{(core-wFO)}$$

Semantics of weighted MSO

MSO: For a word w and a valuation σ

$w, \sigma \models \top$		always,
$w, \sigma \models P_a(x)$	iff	$w(\sigma(x)) = a$,
$w, \sigma \models x \leq y$	iff	$\sigma(x) \leq \sigma(y)$,
$w, \sigma \models x \in X$	iff	$\sigma(x) \in \sigma(X)$,
$w, \sigma \models \neg\varphi$	iff	$w, \sigma \not\models \varphi$,
$w, \sigma \models \varphi_1 \wedge \varphi_2$	iff	$w, \sigma \models \varphi_1$ and $w, \sigma \models \varphi_2$,
$w, \sigma \models \forall x.\varphi$	iff	$w, \sigma[x \mapsto i] \models \varphi$ for all $i \in \{1, \dots, \lvert w \rvert\}$
$w, \sigma \models \forall X.\varphi$	iff	$w, \sigma[X \mapsto I] \models \varphi$ for all $I \subseteq \{1, \dots, \lvert w \rvert\}$

Semantics of weighted MSO

step-wMSO: We define a function $[\![\cdot]\!]$ that assigns a weight to each pair of word and valuation

$$[\![r]\!](w,\sigma) = r$$

$$[\![\varphi\,?\,\Psi_1 : \Psi_2]\!](w,\sigma) = \begin{cases} [\![\Psi_1]\!](w,\sigma) \text{ if } w,\sigma \models \varphi \\ [\![\Psi_2]\!](w,\sigma) \text{ otherwise} \end{cases}$$

Semantics of weighted MSO

core-wMSO: We define a function $[\![\cdot]\!]$ that assigns a multiset of sequences of weights to each pair of word and valuation

$$[\![\mathbf{0}]\!](w,\sigma) = \emptyset$$

$$[\![\varphi \,?\, \Phi_1 : \Phi_2]\!](w,\sigma) = \begin{cases} [\![\Phi_1]\!](w,\sigma) & \text{if } w,\sigma \models \varphi \\ [\![\Phi_2]\!](w,\sigma) & \text{otherwise} \end{cases}$$

$$[\![\textstyle\prod_x \Psi]\!](w,\sigma) = \{\!| r_1 r_2 \dots r_{|w|} |\!\} \text{ where } r_i = [\![\Psi]\!](w,\sigma[x \mapsto i])$$

Semantics of weighted MSO

core-wMSO: We define a function $\llbracket \cdot \rrbracket$ that assigns a multiset of sequences of weights to each pair of word and valuation

$$\llbracket \mathbf{0} \rrbracket (w, \sigma) = \emptyset$$

$$\llbracket \varphi\, ?\, \Phi_1 : \Phi_2 \rrbracket (w, \sigma) = \begin{cases} \llbracket \Phi_1 \rrbracket (w, \sigma) \text{ if } w, \sigma \models \varphi \\ \llbracket \Phi_2 \rrbracket (w, \sigma) \text{ otherwise} \end{cases}$$

$$\llbracket \textstyle\prod_x \Psi \rrbracket (w, \sigma) = \{\!| r_1 r_2 \ldots r_{|w|} |\!\} \text{ where } r_i = \llbracket \Psi \rrbracket (w, \sigma[x \mapsto i])$$

$$\llbracket \Phi_1 + \Phi_2 \rrbracket (w, \sigma) = \llbracket \Phi_1 \rrbracket (w, \sigma) \uplus \llbracket \Phi_2 \rrbracket (w, \sigma)$$

$$\llbracket \textstyle\sum_x \Phi \rrbracket (w, \sigma) = \biguplus_{1 \leq i \leq |w|} \llbracket \Phi \rrbracket (w, \sigma[x \mapsto i])$$

$$\llbracket \textstyle\sum_X \Phi \rrbracket (w, \sigma) = \biguplus_{I \subseteq \{1, \ldots, |w|\}} \llbracket \Phi \rrbracket (w, \sigma[X \mapsto I])$$

Example

Count the maximum number of consecutive a's.

Example

Count the maximum number of consecutive a's.

$$\varphi = x \leq y \wedge \forall z.((x \leq z \wedge z \leq y) \rightarrow P_a(z))$$

$$\Psi = \varphi \,?\, 1 : 0 \quad \Phi' = \prod_y \Psi \quad \Phi = \sum_x \Phi' \quad w = abaa$$

$$(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$$

Example

Count the maximum number of consecutive a's.

$$\varphi = x \leq y \wedge \forall z.((x \leq z \wedge z \leq y) \rightarrow P_a(z))$$

$$\Psi = \varphi \,?\, 1 : 0 \quad \Phi' = \prod\nolimits_y \Psi \quad \Phi = \sum\nolimits_x \Phi' \quad w = abaa$$

$$(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$$

$$\begin{aligned} & \llbracket \Phi \rrbracket (w, \sigma) \\ = & \llbracket \Phi' \rrbracket (w, \sigma[x \mapsto 1]) \uplus \llbracket \Phi' \rrbracket (w, \sigma[x \mapsto 2]) \\ & \uplus \llbracket \Phi' \rrbracket (w, \sigma[x \mapsto 3]) \uplus \llbracket \Phi' \rrbracket (w, \sigma[x \mapsto 4]) \end{aligned}$$

Example

Count the maximum number of consecutive a's.

$$\varphi = x \leq y \wedge \forall z.((x \leq z \wedge z \leq y) \rightarrow P_a(z))$$

$$\Psi = \varphi\,?\,1 : 0 \quad \Phi' = \prod_y \Psi \quad \Phi = \sum_x \Phi' \quad w = abaa$$

$$(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$$

$$\begin{aligned} & \llbracket \Phi \rrbracket (w, \sigma) \\ = & \llbracket \Phi' \rrbracket (w, \sigma[x \mapsto 1]) \uplus \llbracket \Phi' \rrbracket (w, \sigma[x \mapsto 2]) \\ & \uplus \llbracket \Phi' \rrbracket (w, \sigma[x \mapsto 3]) \uplus \llbracket \Phi' \rrbracket (w, \sigma[x \mapsto 4]) \end{aligned}$$

Example

Count the maximum number of consecutive a's.

$$\varphi = x \leq y \wedge \forall z.((x \leq z \wedge z \leq y) \rightarrow P_a(z))$$

$$\Psi = \varphi \,?\, 1 : 0 \quad \Phi' = \prod_y \Psi \quad \Phi = \sum_x \Phi' \quad w = abaa$$

$$(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$$

$$\begin{aligned} & [\![\Phi]\!]\,(w,\sigma) \\ = & \{\!|\, [\![\Psi]\!]\,(w,\sigma[x \mapsto 1, y \mapsto 1])\, [\![\Psi]\!]\,(w,\sigma[x \mapsto 1, y \mapsto 2]) \\ & [\![\Psi]\!]\,(w,\sigma[x \mapsto 1, y \mapsto 3])\, [\![\Psi]\!]\,(w,\sigma[x \mapsto 1, y \mapsto 4]) |\!\} \\ & \uplus \left[\!\left[\Phi'\right]\!\right](w,\sigma[x \mapsto 2]) \uplus \left[\!\left[\Phi'\right]\!\right](w,\sigma[x \mapsto 3]) \uplus \left[\!\left[\Phi'\right]\!\right](w,\sigma[x \mapsto 4]) \end{aligned}$$

Example

Count the maximum number of consecutive a's.

$$\varphi = x \leq y \wedge \forall z.((x \leq z \wedge z \leq y) \rightarrow P_a(z))$$

$$\Psi = \varphi \,?\, 1 : 0 \quad \Phi' = \prod_y \Psi \quad \Phi = \sum_x \Phi' \quad w = abaa$$

$$(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$$

$$\llbracket \Phi \rrbracket (w, \sigma)$$
$$= \{\!|1000|\!\}$$
$$\uplus \llbracket \Phi' \rrbracket (w, \sigma[x \mapsto 2]) \uplus \llbracket \Phi' \rrbracket (w, \sigma[x \mapsto 3]) \uplus \llbracket \Phi' \rrbracket (w, \sigma[x \mapsto 4])$$

Example

Count the maximum number of consecutive a's.

$$\varphi = x \leq y \wedge \forall z.((x \leq z \wedge z \leq y) \rightarrow P_a(z))$$

$$\Psi = \varphi\,?\,1 : 0 \quad \Phi' = \prod_y \Psi \quad \Phi = \sum_x \Phi' \quad w = abaa$$

$$(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$$

$$\begin{aligned} & [\![\Phi]\!]\,(w, \sigma) \\ = & \{\!|1000|\!\} \\ & \uplus\ [\![\Phi']\!]\,(w, \sigma[x \mapsto 2]) \uplus [\![\Phi']\!]\,(w, \sigma[x \mapsto 3]) \uplus [\![\Phi']\!]\,(w, \sigma[x \mapsto 4]) \end{aligned}$$

Example

Count the maximum number of consecutive a's.

$$\varphi = x \leq y \wedge \forall z.((x \leq z \wedge z \leq y) \rightarrow P_a(z))$$

$$\Psi = \varphi\,?\,1 : 0 \quad \Phi' = \prod_y \Psi \quad \Phi = \sum_x \Phi' \quad w = abaa$$

$$(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$$

$$\begin{aligned} & [\![\Phi]\!]\,(w,\sigma) \\ = & \{\!|1000|\!\} \uplus \{\!|0000|\!\} \\ & \uplus [\![\Phi']\!]\,(w,\sigma[x \mapsto 3]) \uplus [\![\Phi']\!]\,(w,\sigma[x \mapsto 4]) \end{aligned}$$

Example

Count the maximum number of consecutive a's.

$$\varphi = x \leq y \wedge \forall z.((x \leq z \wedge z \leq y) \rightarrow P_a(z))$$

$$\Psi = \varphi\, ?\, 1 : 0 \quad \Phi' = \prod_y \Psi \quad \Phi = \sum_x \Phi' \quad w = abaa$$

$$(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$$

$$\begin{aligned} &[\![\Phi]\!]\,(w, \sigma) \\ &= \{\!|1000|\!\} \uplus \{\!|0000|\!\} \uplus \{\!|0011|\!\} \\ &\quad \uplus [\![\Phi']\!]\,(w, \sigma[x \mapsto 4]) \end{aligned}$$

Example

Count the maximum number of consecutive a's.

$$\varphi = x \leq y \wedge \forall z.((x \leq z \wedge z \leq y) \rightarrow P_a(z))$$

$$\Psi = \varphi\,?\,1 : 0 \quad \Phi' = \prod_y \Psi \quad \Phi = \sum_x \Phi' \quad w = abaa$$

$$(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$$

$$\begin{aligned} &[\![\Phi]\!]\,(w, \sigma) \\ &= \{\!|1000|\!\} \uplus \{\!|0000|\!\} \uplus \{\!|0011|\!\} \uplus \{\!|0001|\!\} \end{aligned}$$

Example

Count the maximum number of consecutive a's.

$$\varphi = x \leq y \wedge \forall z.((x \leq z \wedge z \leq y) \rightarrow P_a(z))$$

$$\Psi = \varphi \,?\, 1 : 0 \quad \Phi' = \prod_y \Psi \quad \Phi = \sum_x \Phi' \quad w = abaa$$

$$(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$$

$$\begin{aligned} & [\![\Phi]\!](w, \sigma) \\ &= \{\!|1000|\!\} \uplus \{\!|0000|\!\} \uplus \{\!|0011|\!\} \uplus \{\!|0001|\!\} \\ &= \max\{1, 0, 2, 1\} = 2 \end{aligned}$$

Weighted MSO and weighted automata

Theorem (Droste and Gastin, TCS 2007)

Weighted MSO and and weighted automata are expressively equivalent.

Weighted MSO and weighted automata

Theorem (Droste and Gastin, TCS 2007)

Weighted MSO and and weighted automata are expressively equivalent.

Theorem (Droste and Gastin, MFCS 2019)

Weighted FO and aperiodic polynomially ambiguous weighted automata are expressively equivalent.

Axiomatisation

MSO

Theorem (Gheerbrant and ten Cate, LMCS 2012)

MSO on finite words has a complete axiomatisation.

step-wMSO

(S1):	$\Gamma \vdash r_1 \approx r_2$	if $r_1 = r_2$
(S2):	$\Gamma \vdash \Psi_1 \approx \Psi_2$ implies $\Gamma \cup \{\varphi\} \vdash \Psi_1 \approx \Psi_2$	$\forall \varphi \in$ MSO
(S3):	$\Gamma \vdash \Psi \approx \varphi\, ?\, \Psi : \Psi$	
(S4):	$\Gamma \vdash \neg\varphi\, ?\, \Psi_1 : \Psi_2 \approx \varphi\, ?\, \Psi_2 : \Psi_1$	
(S5):	$\Gamma \vdash \varphi\, ?\, \Psi_1 : \Psi_2 \approx \Psi_1$	if $\Gamma \vdash \varphi \leftrightarrow \top$
(S6):	if $\Gamma \cup \{\varphi\} \vdash \Psi_1 \approx \Psi$ and $\Gamma \cup \{\neg\varphi\} \vdash \Psi_2 \approx \Psi$, then $\Gamma \vdash \varphi\, ?\, \Psi_1 : \Psi_2 \approx \Psi$	

Table: Axioms for step-wMSO.

step-wMSO

Theorem (Completeness)

$\Gamma \vdash \Psi_1 \approx \Psi_2$ *if and only if* $[\![\Psi_1]\!](w, \sigma) = [\![\Psi_2]\!](w, \sigma)$ *for all* (w, σ) *such that* $(w, \sigma) \models \Gamma$.

step-wMSO

Theorem (Completeness)

$\Gamma \vdash \Psi_1 \approx \Psi_2$ *if and only if* $[\![\Psi_1]\!](w, \sigma) = [\![\Psi_2]\!](w, \sigma)$ *for all* (w, σ) *such that* $(w, \sigma) \models \Gamma$.

Note: We may use any kind of Boolean logic to condition on. The above result holds for any such logic which has a complete axiomatisation.

core-wMSO

(C1):	$\Gamma \vdash \Phi + 0 \approx \Phi$	
(C2):	$\Gamma \vdash \prod_x \Psi_1 \approx \prod_x \Psi_2$	if $\Gamma \vdash \Psi_1 \approx \Psi_2$
(C3):	$\Gamma \vdash \varphi_1 \,?\, \Phi_1 : \Phi_2 + \varphi_2 \,?\, \Phi_1' : \Phi_2' \approx$ $\varphi_1 \wedge \varphi_2 \,?\, \Phi_1 + \Phi_2 :$ $(\varphi_1 \wedge \neg\varphi_2 \,?\, \Phi_1 + \Phi_2' :$ $(\neg\varphi_1 \wedge \varphi_2 \,?\, \Phi_1' + \Phi_2 : \Phi_1' + \Phi_2'))$	
(C4):	$\Gamma \vdash \sum_X \varphi \,?\, \Phi_1 : \Phi_2 \approx \varphi \,?\, \sum_X \Phi_1 : \sum_X \Phi_2$	if $X \notin \mathtt{var}(\varphi)$
(C5):	$\Gamma \vdash \Phi_1 \approx \Phi_2$ implies $\Gamma \vdash \sum_X \Phi_1 \approx \sum_X \Phi_2$	

Table: Axioms for core-wMSO.

core-wMSO

Conjecture

Neither core-wFO nor core-wMSO has a complete, recursive axiomatisation.

Decision problems

Model checking

Classically: Given M and φ, do we have $M \models \varphi$?

Model checking

Classically: Given M and φ, do we have $M \models \varphi$?

Weighted: Given (w, σ), Φ, and r, do we have $[\![\Phi]\!](w, \sigma) = r$?
step-wMSO: Decidable using model checking of MSO (or FO), which is PSPACE-complete
core-wMSO: Decidable, complexity unclear

Satisfiability

Clasically: Given φ, does there exist M such that $M \models \varphi$?

Satisfiability

Clasically: Given φ, does there exist M such that $M \models \varphi$?

Weighted 1: Given (w, σ) and Φ, does there exist r such that $\llbracket \Phi \rrbracket (w, \sigma) = r$?
Since $\llbracket \cdot \rrbracket$ is a total function, this is trivial

Satisfiability

Clasically: Given φ, does there exist M such that $M \models \varphi$?

Weighted 1: Given (w, σ) and Φ, does there exist r such that $[\![\Phi]\!](w, \sigma) = r$?
Since $[\![\cdot]\!]$ is a total function, this is trivial

Weighted 2: Given Φ and r, does there exist (w, σ) such that $[\![\Phi]\!](w, \sigma) = r$?
step-wMSO: Decidable using satisfiability of MSO (or FO), which has non-elementary complexity
core-wMSO: ??? Conjecture: Decidable

Satisfiability

Weighted 3: Given Φ_1 and Φ_2, does there exist (w, σ) such that $[\![\Phi_1]\!](w, \sigma) = [\![\Phi_2]\!](w, \sigma)$?

step-wMSO: Decidable using satisfiability

core-wMSO: ??? Conjecture: Undecidable, even for FO

Validity

Classically: Given φ, do we have $M \models \varphi$ for all M?

Validity

Classically: Given φ, do we have $M \models \varphi$ for all M?

Weighted 1: Given Φ, does there exist r such that $[\![\Phi]\!](w, \sigma) = r$ for all (w, σ)?

step-wMSO: Decidable using validity of MSO (or FO), which has non-elementary complexity

core-wMSO: Does not make sense (except $\mathbf{0}$)

Validity

Classically: Given φ, do we have $M \models \varphi$ for all M?

Weighted 1: Given Φ, does there exist r such that $[\![\Phi]\!](w,\sigma) = r$ for all (w,σ)?
step-wMSO: Decidable using validity of MSO (or FO), which has non-elementary complexity
core-wMSO: Does not make sense (except $\mathbf{0}$)

Weighted 2: Given Φ_1 and Φ_2, do we have $[\![\Phi_1]\!](w,\sigma) = [\![\Phi_2]\!](w,\sigma)$ for all (w,σ)?
step-wMSO: Decidable using validity
core-wMSO: ??? Conjecture: Undecidable, even for FO

Summary

- We have given a complete axiomatisation of the step layer of weighted MSO.
- We are currently working on the problem of giving a complete axiomatisation for the core layer of weighted MSO. Our current conjecture is that no such axiomatisation exists.
- We have investigated decision problems for weighted MSO that extend classical decision problems for logics such as model checking, satisfiability, and validity.
- For these decision problems, we have decidability results for the step layer, but the core layer is still unclear.

Open problems

- Complete axiomatisation of core-wMSO?
- If no such axiomatisation exists, can we get one for fragments of core-wMSO or core-wFO?
- Establish tight lower and upper bounds on complexity for the decision problems.