Weighted Traces, Their Logics, and an Extension to Weighted MSCs

Ingmar Meinecke
meinecke@informatik.uni-leipzig.de

Institute of Computer Science, Leipzig University
Germany

Workshop “Algebraic Theory of Automata and Logic”
Szeged 30.9.–1.10.2006

Outline

Introduction

quantitative aspects of sequential and distributed systems

- runtime
- multiplicities of certain patterns
- unsafe behavior $\curvearrowright$ probabilities

Mazurkiewicz traces as an important model of concurrency

- comprehensive theory for trace languages
- impact on other models like message sequence charts (MSCs)

weighted logics as a new formalism

- introduced for words by Droste & Gastin
- extended to trees, images, infinite words recently (Vogler, Droste, Mäurer, Rahonis)

Introduction

quantitative aspects of sequential and distributed systems

- runtime
- multiplicities of certain patterns
- unsafe behavior $\curvearrowright$ probabilities

Mazurkiewicz traces as an important model of concurrency

- comprehensive theory for trace languages
- impact on other models like message sequence charts (MSCs)

weighted logics as a new formalism

- introduced for words by Droste & Gastin
- extended to trees, images, infinite words recently (Vogler, Droste, Mäurer, Rahonis)

Introduction

quantitative aspects of sequential and distributed systems

- runtime
- multiplicities of certain patterns
- unsafe behavior $\curvearrowright$ probabilities

Mazurkiewicz traces as an important model of concurrency

- comprehensive theory for trace languages
- impact on other models like message sequence charts (MSCs)

weighted logics as a new formalism

- introduced for words by Droste & Gastin
- extended to trees, images, infinite words recently (Vogler, Droste, Mäurer, Rahonis)

Outline

Weight Structure = Semirings

general frame: execution of a system $\mapsto$ weight

Examples (weight structures)

- $(\mathbb{N}, +, \cdot, 0, 1)$ (counting)
- $(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$ (runtime)
- $([0,1], \max, \cdot, 0, 1)$ (probabilities)
- $\mathbb{B} = (\{0,1\}, \vee, \wedge, 0, 1)$ (Boolean algebra)
- $(\mathfrak{P}(\Sigma^*), \cup, \cdot, \emptyset, \{\varepsilon\})$ (transducer)
- $(R_M, \cup, \circ, \emptyset, \Delta)$ (binary relations on M)

$\mathbb{K} = (K, \oplus, \circ, \mathbb{0}, \mathbb{1})$ is a *semiring* if:

1. $(K, \oplus, \mathbb{0})$ commutative monoid, $(K, \circ, \mathbb{1})$ monoid,
2. $\circ$ distributes over $\oplus$, and $\mathbb{0} \circ k = k \circ \mathbb{0} = \mathbb{0}$ for all $k \in K$

Weight Structure = Semirings

general frame: execution of a system $\mapsto$ weight

Examples (weight structures)

- $(\mathbb{N}, +, \cdot, 0, 1)$ (counting)
- $(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$ (runtime)
- $([0,1], \max, \cdot, 0, 1)$ (probabilities)
- $\mathbb{B} = (\{0,1\}, \vee, \wedge, 0, 1)$ (Boolean algebra)
- $(\mathfrak{P}(\Sigma^*), \cup, \cdot, \emptyset, \{\varepsilon\})$ (transducer)
- $(R_M, \cup, \circ, \emptyset, \Delta)$ (binary relations on M)

$\mathbb{K} = (K, \oplus, \circ, \mathbb{0}, \mathbb{1})$ is a *semiring* if:

1. $(K, \oplus, \mathbb{0})$ commutative monoid, $(K, \circ, \mathbb{1})$ monoid,
2. $\circ$ distributes over $\oplus$, and $\mathbb{0} \circ k = k \circ \mathbb{0} = \mathbb{0}$ for all $k \in K$

Weight Structure = Semirings

general frame: execution of a system $\mapsto$ weight

Examples (weight structures)

- $(\mathbb{N}, +, \cdot, 0, 1)$ (counting)
- $(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$ (runtime)
- $([0,1], \max, \cdot, 0, 1)$ (probabilities)
- $\mathbb{B} = (\{0,1\}, \vee, \wedge, 0, 1)$ (Boolean algebra)
- $(\mathfrak{P}(\Sigma^*), \cup, \cdot, \emptyset, \{\varepsilon\})$ (transducer)
- $(R_M, \cup, \circ, \emptyset, \Delta)$ (binary relations on M)

$\mathbb{K} = (K, \oplus, \circ, \mathbb{0}, \mathbb{1})$ is a *semiring* if:

1. $(K, \oplus, \mathbb{0})$ commutative monoid, $(K, \circ, \mathbb{1})$ monoid,
2. $\circ$ distributes over $\oplus$, and $\mathbb{0} \circ k = k \circ \mathbb{0} = \mathbb{0}$ for all $k \in K$

What are Traces?

- global independency, no auto-concurrency
- alphabet Σ, irreflexive and symmetric *independence relation* $I \subseteq \Sigma \times \Sigma$
- interchange adjacent independent letters
- *trace* $=$ equivalence class of words
- *trace monoid* $\mathbb{M} = \mathbb{M}(\Sigma, I) = \Sigma^*/I$
- *canonical epimorphism* $\varphi : \Sigma^* \to \mathbb{M} : w \mapsto [w]$

Example

- for $I = a - b$ we have $t = [abcbbad] = [bacabbd]$
- *lexicographic normal form* of t for $a < b < c < d$ is $\mathrm{LNF}(t) = abcabbd$

What are Traces?

- global independency, no auto-concurrency
- alphabet Σ, irreflexive and symmetric *independence relation* $I \subseteq \Sigma \times \Sigma$
- interchange adjacent independent letters
- *trace* $=$ equivalence class of words
- *trace monoid* $\mathbb{M} = \mathbb{M}(\Sigma, I) = \Sigma^*/I$
- *canonical epimorphism* $\varphi : \Sigma^* \to \mathbb{M} : w \mapsto [w]$

Example

- for $I = a - b$ we have $t = [abcbbad] = [bacabbd]$
- *lexicographic normal form* of t for $a < b < c < d$ is $\mathrm{LNF}(t) = abcabbd$

Traces as Dependence Graphs

dependence graph (V, E, l) = acyclic graph with $l : V \to \Sigma$ such that $(l(x), l(y)) \in D \iff (x, y) \in E \cup E^{-1} \cup id_V$

$= abcabbd$

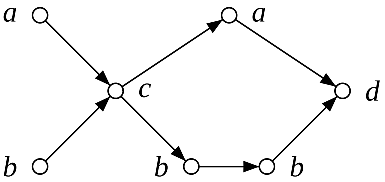
with $(a, b) \in I$

monoid of finite dependence graphs $\cong \mathbb{M}(\Sigma, D)$

$\curvearrowright$ graphical representation more appropriate for logic

Traces as Dependence Graphs

dependence graph (V, E, l) = acyclic graph with $l : V \to \Sigma$ such that $(l(x), l(y)) \in D \iff (x, y) \in E \cup E^{-1} \cup id_V$



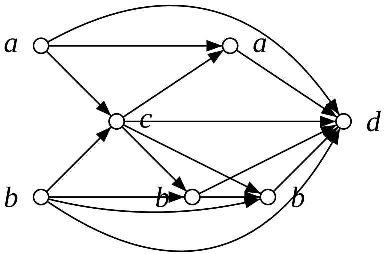
$= abcabbd$
with $(a, b) \in I$

monoid of finite dependence graphs $\cong \mathbb{M}(\Sigma, D)$

$\curvearrowright$ graphical representation more appropriate for logic

Traces as Dependence Graphs

dependence graph (V, E, l) = acyclic graph with $l : V \to \Sigma$ such that $(l(x), l(y)) \in D \iff (x, y) \in E \cup E^{-1} \cup id_V$



$= abcabbd$

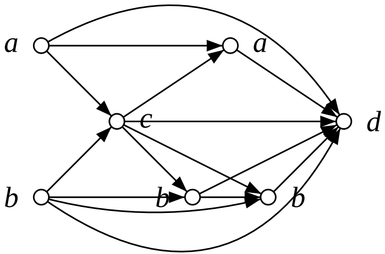
with $(a, b) \in I$

monoid of finite dependence graphs $\cong \mathbb{M}(\Sigma, D)$

$\curvearrowright$ graphical representation more appropriate for logic

Traces as Dependence Graphs

dependence graph (V, E, l) = acyclic graph with $l : V \to \Sigma$ such that $(l(x), l(y)) \in D \iff (x, y) \in E \cup E^{-1} \cup id_V$



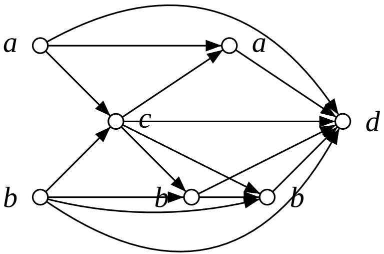
$= abcabbd$
with $(a, b) \in I$

monoid of finite dependence graphs $\cong \mathbb{M}(\Sigma, D)$

↷ graphical representation more appropriate for logic

Traces as Dependence Graphs

dependence graph (V, E, l) = acyclic graph with $l : V \to \Sigma$ such that $(l(x), l(y)) \in D \iff (x, y) \in E \cup E^{-1} \cup id_V$



$= abcabbd$
with $(a, b) \in I$

monoid of finite dependence graphs $\cong \mathbb{M}(\Sigma, D)$
$\curvearrowright$ graphical representation more appropriate for logic

Recognizable Trace Series

$\mathbb{K} = (K, \oplus, \circ, \mathbb{0}, \mathbb{1})$ *commutative* semiring, i.e., $\circ$ commutative

$S : \mathbb{M} \to \mathbb{K}$ *recognizable* if there are: a finite state set Q, a monoid homomorphism $\mu : \mathbb{M} \to \mathbb{K}^{Q\times Q}$, $\lambda \in \mathbb{K}^{1\times Q}$, $\gamma \in \mathbb{K}^{Q\times 1}$ such that $(S, t) = \lambda\, \mu(t)\, \gamma$.

semiring $\mathbb{N}$, $(a, b) \in I$,
S is the behavior of
a *weighted automaton with I-diamond-property*: $\mu(ab) = \mu(ba)$
$\implies (S, ab) = (S, ba) = 12$

Recognizable Trace Series

$\mathbb{K} = (K, \oplus, \circ, \mathbb{0}, \mathbb{1})$ *commutative* semiring, i.e., $\circ$ commutative

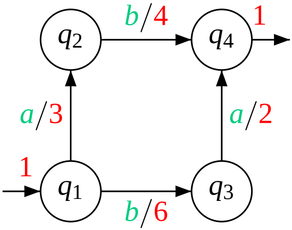
$S : \mathbb{M} \to \mathbb{K}$ *recognizable* if there are: a finite state set Q,
a monoid homomorphism $\mu : \mathbb{M} \to \mathbb{K}^{Q\times Q}$, $\lambda \in \mathbb{K}^{1\times Q}$, $\gamma \in \mathbb{K}^{Q\times 1}$
such that $(S, t) = \lambda\, \mu(t)\, \gamma$.

semiring $\mathbb{N}$, $(a, b) \in I$,
S is the behavior of
a *weighted automaton with*
I-diamond-property: $\mu(ab) = \mu(ba)$
$\implies (S, ab) = (S, ba) = 12$

Recognizable Trace Series

$\mathbb{K} = (K, \oplus, \circ, \mathbb{0}, \mathbb{1})$ *commutative* semiring, i.e., $\circ$ commutative

$S : \mathbb{M} \to \mathbb{K}$ *recognizable* if there are: a finite state set Q, a monoid homomorphism $\mu : \mathbb{M} \to \mathbb{K}^{Q\times Q}$, $\lambda \in \mathbb{K}^{1\times Q}$, $\gamma \in \mathbb{K}^{Q\times 1}$ such that $(S, t) = \lambda\,\mu(t)\,\gamma$.



semiring $\mathbb{N}$, $(a, b) \in I$,
S is the behavior of
a *weighted automaton with I-diamond-property*: $\mu(ab) = \mu(ba)$
$\implies (S, ab) = (S, ba) = 12$

Outline

The Intuition behind Weighted Logics

usual MSO-formula Φ: then Φ holds or not $\curvearrowright [\![\Phi]\!] \in \{0, 1\}$

a *quantitative semantics* $=$ number of verifications:

- $(x, y) \in E$ has one verification $\curvearrowright [\![(x, y) \in E]\!] = 1$
- $\Phi \vee \Psi$ has $[\![\Phi]\!] + [\![\Psi]\!]$ verifications
- $\Phi \wedge \Psi$ has $[\![\Phi]\!] \cdot [\![\Psi]\!]$ verifications
- $\exists x. \Phi$ has as many verifications as elements that verify Φ

What about negation?

- could be defined in Boolean algebras, but in general semirings not clear
- solution: *negate atomic formulas only* (and extend syntax by disjunction and universal quantification)

$\longrightarrow$ weighted logics introduced for words by Droste & Gastin (2005)

The Intuition behind Weighted Logics

usual MSO-formula Φ: then Φ holds or not $\curvearrowright [\![\Phi]\!] \in \{0, 1\}$

a *quantitative semantics* $=$ number of verifications:

- $(x, y) \in E$ has one verification $\curvearrowright [\![(x, y) \in E]\!] = 1$
- $\Phi \vee \Psi$ has $[\![\Phi]\!] + [\![\Psi]\!]$ verifications
- $\Phi \wedge \Psi$ has $[\![\Phi]\!] \cdot [\![\Psi]\!]$ verifications
- $\exists x.\ \Phi$ has as many verifications as elements that verify Φ

What about negation?

- could be defined in Boolean algebras, but in general semirings not clear
- solution: *negate atomic formulas only* (and extend syntax by disjunction and universal quantification)

$\longrightarrow$ weighted logics introduced for words by Droste & Gastin (2005)

The Intuition behind Weighted Logics

<u>usual MSO-formula Φ:</u> then Φ holds or not $\curvearrowright [\![\Phi]\!] \in \{0, 1\}$

a *quantitative semantics* $=$ number of verifications:

- $(x, y) \in E$ has one verification $\curvearrowright [\![(x, y) \in E]\!] = 1$
- $\Phi \vee \Psi$ has $[\![\Phi]\!] + [\![\Psi]\!]$ verifications
- $\Phi \wedge \Psi$ has $[\![\Phi]\!] \cdot [\![\Psi]\!]$ verifications
- $\exists x.\ \Phi$ has as many verifications as elements that verify Φ

What about negation?

- could be defined in Boolean algebras, but in general semirings not clear
- solution: *negate atomic formulas only* (and extend syntax by disjunction and universal quantification)

$\longrightarrow$ weighted logics introduced for words by Droste & Gastin (2005)

The Intuition behind Weighted Logics

usual MSO-formula Φ: then Φ holds or not $\curvearrowright [\![\Phi]\!] \in \{0, 1\}$

a *quantitative semantics* $=$ number of verifications:

- $(x, y) \in E$ has one verification $\curvearrowright [\![(x, y) \in E]\!] = 1$
- $\Phi \vee \Psi$ has $[\![\Phi]\!] + [\![\Psi]\!]$ verifications
- $\Phi \wedge \Psi$ has $[\![\Phi]\!] \cdot [\![\Psi]\!]$ verifications
- $\exists x.\ \Phi$ has as many verifications as elements that verify Φ

What about negation?

- could be defined in Boolean algebras, but in general semirings not clear
- solution: *negate atomic formulas only* (and extend syntax by disjunction and universal quantification)

$\longrightarrow$ weighted logics introduced for words by Droste & Gastin (2005)

The Intuition behind Weighted Logics

usual MSO-formula Φ: then Φ holds or not $\curvearrowright [\![\Phi]\!] \in \{0,1\}$

a *quantitative semantics* $=$ number of verifications:

- $(x,y) \in E$ has one verification $\curvearrowright [\![(x,y) \in E]\!] = 1$
- $\Phi \vee \Psi$ has $[\![\Phi]\!] + [\![\Psi]\!]$ verifications
- $\Phi \wedge \Psi$ has $[\![\Phi]\!] \cdot [\![\Psi]\!]$ verifications
- $\exists x.\ \Phi$ has as many verifications as elements that verify Φ

What about negation?

- could be defined in Boolean algebras, but in general semirings not clear
- solution: *negate atomic formulas only* (and extend syntax by disjunction and universal quantification)

$\longrightarrow$ weighted logics introduced for words by Droste & Gastin (2005)

The Intuition behind Weighted Logics

usual MSO-formula Φ: then Φ holds or not $\curvearrowright [\![\Phi]\!] \in \{0,1\}$

a *quantitative semantics* $=$ number of verifications:

- $(x,y) \in E$ has one verification $\curvearrowright [\![(x,y) \in E]\!] = 1$
- $\Phi \vee \Psi$ has $[\![\Phi]\!] + [\![\Psi]\!]$ verifications
- $\Phi \wedge \Psi$ has $[\![\Phi]\!] \cdot [\![\Psi]\!]$ verifications
- $\exists x.\ \Phi$ has as many verifications as elements that verify Φ

What about negation?

- could be defined in Boolean algebras, but in general semirings not clear
- solution: *negate atomic formulas only* (and extend syntax by disjunction and universal quantification)

$\longrightarrow$ weighted logics introduced for words by Droste & Gastin (2005)

The Intuition behind Weighted Logics

usual MSO-formula Φ: then Φ holds or not $\curvearrowright [\![\Phi]\!] \in \{0, 1\}$

a *quantitative semantics* $=$ number of verifications:

- $(x, y) \in E$ has one verification $\curvearrowright [\![(x, y) \in E]\!] = 1$
- $\Phi \vee \Psi$ has $[\![\Phi]\!] + [\![\Psi]\!]$ verifications
- $\Phi \wedge \Psi$ has $[\![\Phi]\!] \cdot [\![\Psi]\!]$ verifications
- $\exists x. \, \Phi$ has as many verifications as elements that verify Φ

What about negation?

- could be defined in Boolean algebras, but in general semirings not clear
- solution: *negate atomic formulas only* (and extend syntax by disjunction and universal quantification)

$\longrightarrow$ weighted logics introduced for words by Droste & Gastin (2005)

The Intuition behind Weighted Logics

usual MSO-formula Φ: then Φ holds or not $\curvearrowright [\![\Phi]\!] \in \{0, 1\}$

a *quantitative semantics* $=$ number of verifications:

- $(x, y) \in E$ has one verification $\curvearrowright [\![(x, y) \in E]\!] = 1$
- $\Phi \vee \Psi$ has $[\![\Phi]\!] + [\![\Psi]\!]$ verifications
- $\Phi \wedge \Psi$ has $[\![\Phi]\!] \cdot [\![\Psi]\!]$ verifications
- $\exists x.\ \Phi$ has as many verifications as elements that verify Φ

What about negation?

- could be defined in Boolean algebras, but in general semirings not clear
- solution: *negate atomic formulas only* (and extend syntax by disjunction and universal quantification)

$\longrightarrow$ weighted logics introduced for words by Droste & Gastin (2005)

The Intuition behind Weighted Logics

usual MSO-formula Φ: then Φ holds or not $\curvearrowright [\![\Phi]\!] \in \{0, 1\}$

a *quantitative semantics* $=$ number of verifications:

- $(x, y) \in E$ has one verification $\curvearrowright [\![(x, y) \in E]\!] = 1$
- $\Phi \vee \Psi$ has $[\![\Phi]\!] + [\![\Psi]\!]$ verifications
- $\Phi \wedge \Psi$ has $[\![\Phi]\!] \cdot [\![\Psi]\!]$ verifications
- $\exists x.\ \Phi$ has as many verifications as elements that verify Φ

What about negation?

- could be defined in Boolean algebras, but in general semirings not clear
- solution: *negate atomic formulas only* (and extend syntax by disjunction and universal quantification)

$\longrightarrow$ weighted logics introduced for words by Droste & Gastin (2005)

The Intuition behind Weighted Logics

<u>usual MSO-formula Φ:</u> then Φ holds or not $\curvearrowright [\![\Phi]\!] \in \{0, 1\}$

a *quantitative semantics* $=$ number of verifications:

- $(x, y) \in E$ has one verification $\curvearrowright [\![(x, y) \in E]\!] = 1$
- $\Phi \vee \Psi$ has $[\![\Phi]\!] + [\![\Psi]\!]$ verifications
- $\Phi \wedge \Psi$ has $[\![\Phi]\!] \cdot [\![\Psi]\!]$ verifications
- $\exists x. \Phi$ has as many verifications as elements that verify Φ

What about negation?

- could be defined in Boolean algebras, but in general semirings not clear
- solution: *negate atomic formulas only* (and extend syntax by disjunction and universal quantification)

$\longrightarrow$ weighted logics introduced for words by Droste & Gastin (2005)

Weighted Logics for Traces

commutative $\mathbb{K} = (K, \oplus, \circ, \mathbb{0}, \mathbb{1})$, dependence graphs (V, E, l)

$$\Phi ::= k \mid P_a(x) \mid E(x,y) \mid x \in X \mid \neg P_a(x) \mid \neg E(x,y) \mid \neg x \in X \mid$$
$$\Phi \vee \Psi \mid \Phi \wedge \Psi \mid \exists x.\Phi \mid \exists X.\Phi \mid \forall x.\Phi \mid \forall X.\Phi$$

and *semantics* $[\![\Phi]\!]_{\mathcal{V}} : \mathbb{M}_{\mathcal{V}}(\Sigma, D) \to \mathbb{K}$ (assignment $\sigma : \mathcal{V} \to V$)

- $[\![k]\!]_{\mathcal{V}}(t,\sigma) = k$
- $[\![E(x,y)]\!]_{\mathcal{V}}(t,\sigma) = \begin{cases} \mathbb{1} & \text{if } (\sigma(x),\sigma(y)) \in E, \\ \mathbb{0} & \text{otherwise} \end{cases}$
- $[\![\neg\Phi]\!]_{\mathcal{V}}(t,\sigma) = \begin{cases} \mathbb{1} & \text{if } [\![\Phi]\!]_{\mathcal{V}}(t,\sigma) = \mathbb{0}, \\ \mathbb{0} & \text{if } [\![\Phi]\!]_{\mathcal{V}}(t,\sigma) = \mathbb{1} \end{cases}$
- $[\![\Phi \wedge \Psi]\!]_{\mathcal{V}}(t,\sigma) = [\![\Phi]\!]_{\mathcal{V}}(t,\sigma) \circ [\![\Psi]\!]_{\mathcal{V}}(t,\sigma)$
- $[\![\exists x.\Phi]\!]_{\mathcal{V}}(t,\sigma) = \bigoplus_{v \in V} [\![\Phi]\!]_{\mathcal{V}\cup\{x\}}(t, \sigma[x \to v])$

Weighted Logics for Traces

commutative $\mathbb{K} = (K, \oplus, \circ, \mathbb{0}, \mathbb{1})$, dependence graphs (V, E, l)

$$\Phi ::= k \mid P_a(x) \mid E(x,y) \mid x \in X \mid \neg P_a(x) \mid \neg E(x,y) \mid \neg x \in X \mid$$
$$\Phi \vee \Psi \mid \Phi \wedge \Psi \mid \exists x.\Phi \mid \exists X.\Phi \mid \forall x.\Phi \mid \forall X.\Phi$$

and *semantics* $[\![\Phi]\!]_{\mathcal{V}} : \mathbb{M}_{\mathcal{V}}(\Sigma, D) \to \mathbb{K}$ (assignment $\sigma : \mathcal{V} \to V$)

- $[\![k]\!]_{\mathcal{V}}(t, \sigma) = k$
- $[\![E(x,y)]\!]_{\mathcal{V}}(t, \sigma) = \begin{cases} \mathbb{1} & \text{if } (\sigma(x), \sigma(y)) \in E, \\ \mathbb{0} & \text{otherwise} \end{cases}$
- $[\![\neg\Phi]\!]_{\mathcal{V}}(t, \sigma) = \begin{cases} \mathbb{1} & \text{if } [\![\Phi]\!]_{\mathcal{V}}(t, \sigma) = \mathbb{0}, \\ \mathbb{0} & \text{if } [\![\Phi]\!]_{\mathcal{V}}(t, \sigma) = \mathbb{1} \end{cases}$
- $[\![\Phi \wedge \Psi]\!]_{\mathcal{V}}(t, \sigma) = [\![\Phi]\!]_{\mathcal{V}}(t, \sigma) \circ [\![\Psi]\!]_{\mathcal{V}}(t, \sigma)$
- $[\![\exists x.\Phi]\!]_{\mathcal{V}}(t, \sigma) = \bigoplus_{v \in V} [\![\Phi]\!]_{\mathcal{V} \cup \{x\}}(t, \sigma[x \to v])$

Weighted Logics for Traces

commutative $\mathbb{K} = (K, \oplus, \circ, \mathbb{0}, \mathbb{1})$, dependence graphs (V, E, l)

$$\Phi ::= k \mid P_a(x) \mid E(x,y) \mid x \in X \mid \neg P_a(x) \mid \neg E(x,y) \mid \neg x \in X \mid \\ \Phi \vee \Psi \mid \Phi \wedge \Psi \mid \exists x.\Phi \mid \exists X.\Phi \mid \forall x.\Phi \mid \forall X.\Phi$$

and *semantics* $[\![\Phi]\!]_{\mathcal{V}} : \mathbb{M}_{\mathcal{V}}(\Sigma, D) \to \mathbb{K}$ (assignment $\sigma : \mathcal{V} \to V$)

- $[\![k]\!]_{\mathcal{V}}(t,\sigma) = k$
- $[\![E(x,y)]\!]_{\mathcal{V}}(t,\sigma) = \begin{cases} \mathbb{1} & \text{if } (\sigma(x),\sigma(y)) \in E, \\ \mathbb{0} & \text{otherwise} \end{cases}$
- $[\![\neg\Phi]\!]_{\mathcal{V}}(t,\sigma) = \begin{cases} \mathbb{1} & \text{if } [\![\Phi]\!]_{\mathcal{V}}(t,\sigma) = \mathbb{0}, \\ \mathbb{0} & \text{if } [\![\Phi]\!]_{\mathcal{V}}(t,\sigma) = \mathbb{1} \end{cases}$
- $[\![\Phi \wedge \Psi]\!]_{\mathcal{V}}(t,\sigma) = [\![\Phi]\!]_{\mathcal{V}}(t,\sigma) \circ [\![\Psi]\!]_{\mathcal{V}}(t,\sigma)$
- $[\![\exists x.\Phi]\!]_{\mathcal{V}}(t,\sigma) = \bigoplus_{v \in V} [\![\Phi]\!]_{\mathcal{V}\cup\{x\}}(t, \sigma[x \to v])$

Recognizability = Definability

<u>even for words:</u> general wMSO-formulas exceed recognizability

define *class* RMSO with Φ *restricted*, if

- no occurence of $\forall X.\Psi$ and
- $\forall x.\Psi$ only with $[\![\,\Psi\,]\!] = \sum_{i=1}^{n} k_i \mathbf{1}_{L_i}$ a *definable step function* (L_i definable languages)

Theorem

For $\mathbb{M}(\Sigma, D)$ trace monoid & $\mathbb{K}$ commutative semiring:
$S : \mathbb{M}(\Sigma, D) \to \mathbb{K}$ *recognizable* $\iff$ $S = [\![\,\Phi\,]\!]$ *for restricted* Φ.

Adapt technique from trace languages:
Translate edge relation E to $<$ for words and vice versa!
(Ebinger/Muschholl)
Then use result for words! (Droste/Gastin)

Recognizability = Definability

<u>even for words:</u> general wMSO-formulas exceed recognizability

define *class* RMSO with Φ *restricted*, if

- no occurence of $\forall X.\Psi$ and
- $\forall x.\Psi$ only with $[\![\Psi]\!] = \sum_{i=1}^{n} k_i \mathbf{1}_{L_i}$ a *definable step function* (L_i definable languages)

Theorem

For $\mathbb{M}(\Sigma, D)$ *trace monoid &* $\mathbb{K}$ *commutative semiring:*
$S : \mathbb{M}(\Sigma, D) \to \mathbb{K}$ *recognizable* $\iff$ $S = [\![\Phi]\!]$ *for restricted* Φ*.*

Adapt technique from trace languages:
Translate edge relation E to $<$ for words and vice versa!
(Ebinger/Muscholl)
Then use result for words! (Droste/Gastin)

Recognizability = Definability

even for words: general wMSO-formulas exceed recognizability

define *class* RMSO with Φ *restricted*, if

- no occurence of $\forall X.\Psi$ and
- $\forall x.\Psi$ only with $[\![\Psi]\!] = \sum_{i=1}^{n} k_i \mathbf{1}_{L_i}$ a *definable step function* (L_i definable languages)

Theorem

For $\mathbb{M}(\Sigma, D)$ trace monoid & $\mathbb{K}$ commutative semiring:
$S : \mathbb{M}(\Sigma, D) \to \mathbb{K}$ recognizable $\iff$ $S = [\![\Phi]\!]$ for restricted Φ.

Adapt technique from trace languages:
Translate edge relation E to $<$ for words and vice versa! (Ebinger/Muscholl)
Then use result for words! (Droste/Gastin)

Recognizability = Definability

even for words: general wMSO-formulas exceed recognizability

define *class* RMSO with Φ *restricted*, if

- no occurence of $\forall X.\Psi$ and
- $\forall x.\Psi$ only with $[\![\Psi]\!] = \sum_{i=1}^{n} k_i \mathbf{1}_{L_i}$ a *definable step function* (L_i definable languages)

Theorem

For $\mathbb{M}(\Sigma, D)$ *trace monoid &* $\mathbb{K}$ *commutative semiring:*
$S : \mathbb{M}(\Sigma, D) \to \mathbb{K}$ *recognizable* $\iff$ $S = [\![\Phi]\!]$ *for restricted* Φ.

Adapt technique from trace languages:
Translate edge relation E to $<$ for words and vice versa!
(Ebinger/Muscholl)
Then use result for words! (Droste/Gastin)

Translation Lemma

Lemma

For $T : \mathbb{M} \to \mathbb{K}$, $\varphi : \Sigma^* \to \mathbb{M}$ *canonical epimorphism, and* $(\varphi^{-1}(T), w) := (T, \varphi(w))$ *for* $w \in \Sigma^*$ *are equivalent:*

1. T *definable in* RMSO,
2. $S = \varphi^{-1}(T) : \Sigma^* \to \mathbb{K}$ RMSO*-definable,*
3. $S' = \varphi^{-1}(T)_{|\,\mathrm{LNF}} : \Sigma^* \to \mathbb{K}$ RMSO*-definable.*

Proof idea of $(3) \implies (1)$.

Let S' be defined by RMSO-formula Φ. Replace $x < y$ in Φ by new FO-formula $\mathrm{lex}(x, y)$ for traces with
$(t, \sigma) \models \mathrm{lex}(x, y) \iff \sigma(x) < \sigma(y)$ in $\mathrm{LNF}(t)$
and make $\mathrm{lex}(x, y)$ *unambiguous*!

▸ proof details

Translation Lemma

Lemma

For $T : \mathbb{M} \to \mathbb{K}$, $\varphi : \Sigma^* \to \mathbb{M}$ *canonical epimorphism, and* $(\varphi^{-1}(T), w) := (T, \varphi(w))$ *for* $w \in \Sigma^*$ *are equivalent:*

1. T *definable in* RMSO*,*
2. $S = \varphi^{-1}(T) : \Sigma^* \to \mathbb{K}$ RMSO*-definable,*
3. $S' = \varphi^{-1}(T)_{|\,\mathrm{LNF}} : \Sigma^* \to \mathbb{K}$ RMSO*-definable.*

Proof idea of $(3) \implies (1)$.

Let S' be defined by RMSO-formula Φ. Replace $x < y$ in Φ by new FO-formula $\mathrm{lex}(x,y)$ for traces with
$(t,\sigma) \models \mathrm{lex}(x,y) \iff \sigma(x) < \sigma(y)$ in $\mathrm{LNF}(t)$
and make $\mathrm{lex}(x,y)$ *unambiguous*!

proof details

Example: Height of a Trace

$\mathbb{K} = (\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$ and $H : \mathbb{M} \to \mathbb{K} : t \mapsto \mathrm{height}(t)$

$\mathrm{chain}(X) = \forall x, y \in X.\big(x = y \vee (x, y) \in E^+ \vee (y, x) \in E^+\big)$
is an FO-formula (E^+ FO over traces).

$\implies$ $\exists$ RFO-formula $\widehat{\mathrm{chain}(X)}$ defining $\mathbf{1}_{L(\mathrm{chain}(X))}$

$\mathrm{card}(X) = \forall x.\big((x \in X \longrightarrow 1) \wedge (\neg x \in X \longrightarrow 0)\big)$
has semantics $|X|$ over $\mathbb{K}$.

$\implies$ H defined by $\Phi = \exists X.\widehat{\mathrm{chain}(X)} \wedge \mathrm{card}(X)$

$\implies$ $H : \mathbb{M} \to \mathbb{K}$ recognizable

Example: Height of a Trace

$\mathbb{K} = (\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$ and $H : \mathbb{M} \to \mathbb{K} : t \mapsto \text{height}(t)$

$\text{chain}(X) = \forall x, y \in X.\big(x = y \vee (x, y) \in E^+ \vee (y, x) \in E^+\big)$
is an FO-formula (E^+ FO over traces).

$\implies$ $\exists$ RFO-formula $\widehat{\text{chain}(X)}$ defining $\mathbf{1}_{L(\text{chain}(X))}$

$\text{card}(X) = \forall x.\big((x \in X \longrightarrow 1) \wedge (\neg x \in X \longrightarrow 0)\big)$
has semantics $|X|$ over $\mathbb{K}$.

$\implies$ H defined by $\Phi = \exists X.\widehat{\text{chain}(X)} \wedge \text{card}(X)$

$\implies$ $H : \mathbb{M} \to \mathbb{K}$ recognizable

Example: Height of a Trace

$\mathbb{K} = (\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$ and $H : \mathbb{M} \to \mathbb{K} : t \mapsto \text{height}(t)$

$\text{chain}(X) = \forall x, y \in X. \big(x = y \vee (x, y) \in E^+ \vee (y, x) \in E^+\big)$
is an FO-formula (E^+ FO over traces).

$\Longrightarrow$ $\exists$ RFO-formula $\widehat{\text{chain}(X)}$ defining $\mathbf{1}_{L(\text{chain}(X))}$

$\text{card}(X) = \forall x. \big((x \in X \longrightarrow 1) \wedge (\neg x \in X \longrightarrow 0)\big)$
has semantics $|X|$ over $\mathbb{K}$.

$\Longrightarrow$ H defined by $\Phi = \exists X. \widehat{\text{chain}(X)} \wedge \text{card}(X)$

$\Longrightarrow$ $H : \mathbb{M} \to \mathbb{K}$ recognizable

Example: Height of a Trace

$\mathbb{K} = (\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$ and $H : \mathbb{M} \to \mathbb{K} : t \mapsto \mathrm{height}(t)$

$\mathrm{chain}(X) = \forall x, y \in X. \big(x = y \vee (x,y) \in E^+ \vee (y,x) \in E^+\big)$
is an FO-formula (E^+ FO over traces).

$\Longrightarrow$ $\exists$ RFO-formula $\widehat{\mathrm{chain}(X)}$ defining $\mathbf{1}_{L(\mathrm{chain}(X))}$

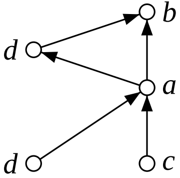
$\mathrm{card}(X) = \forall x. \big((x \in X \longrightarrow 1) \wedge (\neg x \in X \longrightarrow 0)\big)$
has semantics $|X|$ over $\mathbb{K}$.

$\Longrightarrow$ H defined by $\Phi = \exists X. \widehat{\mathrm{chain}(X)} \wedge \mathrm{card}(X)$

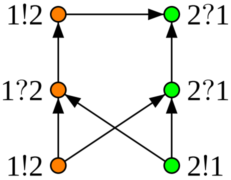
$\Longrightarrow$ $H : \mathbb{M} \to \mathbb{K}$ recognizable

Outline

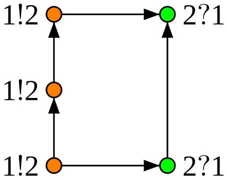
Dags over Distributed Alphabets



trace



MSC



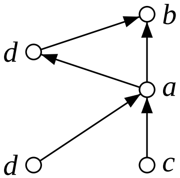
lossy MSC

Unifying frame: *distributed alphabet* $\widetilde{\Sigma} = \bigcup_{i \in Ag} \Sigma_i$ and dependence $D_{\widetilde{\Sigma}} = \{(a,b) \mid a \,\&\, b$ share a process$\}$;

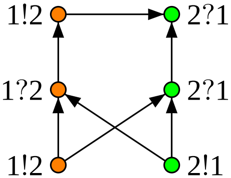
directed acyclic graph $(V, \lhd, l)$ with $l : V \to \widetilde{\Sigma}$ is a $\widetilde{\Sigma}$*-dag* if

- $\forall i \in Ag : l^{-1}(\Sigma_i)$ totally ordered
- $\forall (u,v),(u',v') \in \lhd$ with $l(u) D_{\widetilde{\Sigma}} l(u')$ and $l(v) D_{\widetilde{\Sigma}} l(v')$: $u \leq u' \iff v \leq v'$ (FIFO-property)

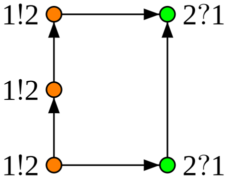
Dags over Distributed Alphabets



trace



MSC



lossy MSC

Unifying frame: *distributed alphabet* $\widetilde{\Sigma} = \bigcup_{i \in Ag} \Sigma_i$ and dependence $D_{\widetilde{\Sigma}} = \{(a,b) \mid a \,\&\, b$ share a process$\}$;

directed acyclic graph $(V, \lhd, l)$ with $l : V \to \widetilde{\Sigma}$ is a *$\widetilde{\Sigma}$-dag* if

- $\forall i \in Ag : l^{-1}(\Sigma_i)$ totally ordered
- $\forall (u,v), (u',v') \in \lhd$ with $l(u) D_{\widetilde{\Sigma}} l(u')$ and $l(v) D_{\widetilde{\Sigma}} l(v')$: $u \leq u' \iff v \leq v'$ (FIFO-property)

Weighted ACAs over $\widetilde{\Sigma}$-dags

weight structure = commutative semiring $\mathbb{K}$

$\Sigma_1 = \{a\}, \Sigma_2 = \{b, d\}, \Sigma_3 = \{c, e\}$

perform an action, new state and weight depending on immediate past,

determine immediate future by a type function $(a, p_1) \to \{b\}$,

leaving the system with global weight

Multiply weights of a run!
$\to \mathbf{wgt}(G) = 72$ in $\mathbb{N}$

Weighted ACAs over $\widetilde{\Sigma}$-dags

weight structure = commutative semiring $\mathbb{K}$

$\Sigma_1 = \{a\}, \Sigma_2 = \{b, d\}, \Sigma_3 = \{c, e\}$

perform an action, new state and weight depending on immediate past,

determine immediate future by a type function $(a, p_1) \to \{b\}$,

leaving the system with global weight

Multiply weights of a run!
$\to \mathbf{wgt}(G) = 72$ in $\mathbb{N}$

$a/q_1/1$

Weighted ACAs over $\widetilde{\Sigma}$-dags

weight structure = commutative semiring $\mathbb{K}$

$\Sigma_1 = \{a\}, \Sigma_2 = \{b, d\}, \Sigma_3 = \{c, e\}$

perform an action, new state and weight depending on immediate past,

determine immediate future by a type function $(a, p_1) \to \{b\}$,

leaving the system with global weight

Multiply weights of a run!

$\to \mathbf{wgt}(G) = 72$ in $\mathbb{N}$

$a/q_1/1$ $d/q_2/1$

Weighted ACAs over $\widetilde{\Sigma}$-dags

weight structure = commutative semiring $\mathbb{K}$

$\Sigma_1 = \{a\}, \Sigma_2 = \{b, d\}, \Sigma_3 = \{c, e\}$

perform an action, new state and weight depending on immediate past,

determine immediate future by a type function $(a, p_1) \to \{b\}$,

leaving the system with global weight

Multiply weights of a run!

$\to \mathbf{wgt}(G) = 72$ in $\mathbb{N}$

$a/q_1/1$ $d/q_2/1$ $e/q_3/2$

Weighted ACAs over $\widetilde{\Sigma}$-dags

weight structure = commutative semiring $\mathbb{K}$

$\Sigma_1 = \{a\}, \Sigma_2 = \{b, d\}, \Sigma_3 = \{c, e\}$

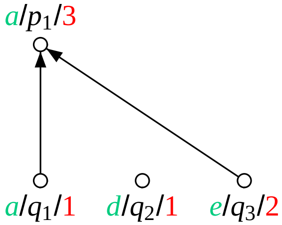
perform an action, new state and weight depending on immediate past,

determine immediate future by a type function $(a, p_1) \to \{b\}$,

leaving the system with global weight

Multiply weights of a run!

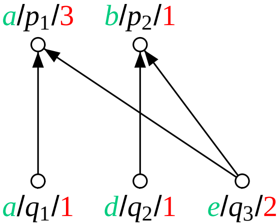
$\to \mathbf{wgt}(G) = 72$ in $\mathbb{N}$



Weighted ACAs over $\widetilde{\Sigma}$-dags

weight structure = commutative semiring $\mathbb{K}$

$\Sigma_1 = \{a\}, \Sigma_2 = \{b, d\}, \Sigma_3 = \{c, e\}$



perform an action, new state and weight depending on immediate past,

determine immediate future by a type function $(a, p_1) \to \{b\}$,

leaving the system with global weight

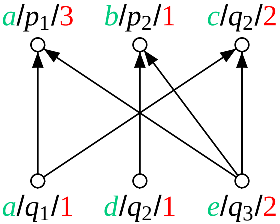
Multiply weights of a run!

$\to \mathbf{wgt}(G) = 72$ in $\mathbb{N}$

Weighted ACAs over $\widetilde{\Sigma}$-dags

weight structure = commutative semiring $\mathbb{K}$

$\Sigma_1 = \{a\}, \Sigma_2 = \{b, d\}, \Sigma_3 = \{c, e\}$



perform an action, new state and weight depending on immediate past,

determine immediate future by a type function $(a, p_1) \to \{b\}$,

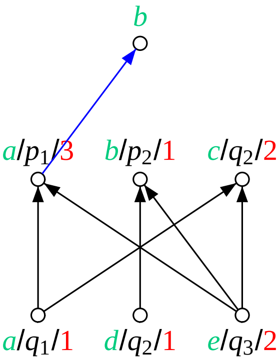
leaving the system with global weight

Multiply weights of a run!
$\to \mathbf{wgt}(G) = 72$ in $\mathbb{N}$

Weighted ACAs over $\widetilde{\Sigma}$-dags

weight structure = commutative semiring $\mathbb{K}$

$\Sigma_1 = \{a\}, \Sigma_2 = \{b, d\}, \Sigma_3 = \{c, e\}$



perform an action, new state and weight depending on immediate past,

determine immediate future by a type function $(a, p_1) \to \{b\}$,

leaving the system with global weight

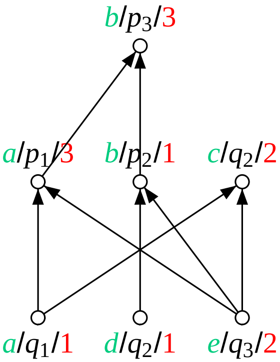
Multiply weights of a run!

$\to \mathbf{wgt}(G) = 72$ in $\mathbb{N}$

Weighted ACAs over $\widetilde{\Sigma}$-dags

weight structure = commutative semiring $\mathbb{K}$

$\Sigma_1 = \{a\}, \Sigma_2 = \{b,d\}, \Sigma_3 = \{c,e\}$



perform an action, new state and weight depending on immediate past,

determine immediate future by a type function $(a, p_1) \to \{b\}$,

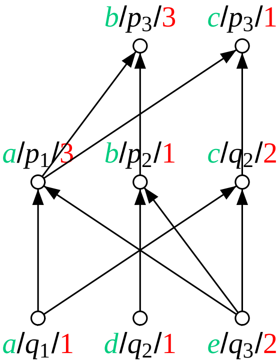
leaving the system with global weight

Multiply weights of a run!
$\to \mathbf{wgt}(G) = 72$ in $\mathbb{N}$

Weighted ACAs over $\widetilde{\Sigma}$-dags

weight structure = commutative semiring $\mathbb{K}$

$\Sigma_1 = \{a\}, \Sigma_2 = \{b, d\}, \Sigma_3 = \{c, e\}$



perform an action, new state and weight depending on immediate past,

determine immediate future by a type function $(a, p_1) \to \{b\}$,

leaving the system with global weight

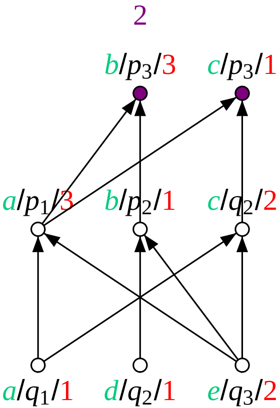
Multiply weights of a run!

$\to$ $\mathbf{wgt}(G) = 72$ in $\mathbb{N}$

Weighted ACAs over $\widetilde{\Sigma}$-dags

weight structure = commutative semiring $\mathbb{K}$

$\Sigma_1 = \{a\}, \Sigma_2 = \{b, d\}, \Sigma_3 = \{c, e\}$



perform an action, new state and weight depending on immediate past,

determine immediate future by a type function $(a, p_1) \to \{b\}$,

leaving the system with global weight

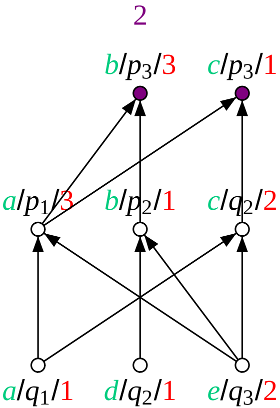
Multiply weights of a run!

$\to \mathbf{wgt}(G) = 72$ in $\mathbb{N}$

Weighted ACAs over $\widetilde{\Sigma}$-dags

weight structure = commutative semiring $\mathbb{K}$

$\Sigma_1 = \{a\}, \Sigma_2 = \{b, d\}, \Sigma_3 = \{c, e\}$



perform an action, new state and weight depending on immediate past,

determine immediate future by a type function $(a, p_1) \to \{b\}$,

leaving the system with global weight

Multiply weights of a run!

$\to \mathbf{wgt}(G) = 72$ in $\mathbb{N}$

Logical Characterization

Define a *reduced weighted MSO-logic* (RMSO) similar to those for traces.

Theorem (B. Bollig & I.M. 2006)

For a commutative semiring $\mathbb{K}$ and $S : \mathbb{DAG}(\widetilde{\Sigma}) \to \mathbb{K}$ are equivalent:

1. $S = \|\mathcal{A}\|$ *for some wACA with types* $\mathcal{A}$,
2. S *is RMSO-definable.*

(direct proof with rather tricky constructions)

generalizations & applications for:

traces, message sequence charts (MSCs), probabilistic lossy-channel systems, probabilistic asynchronous automata

Logical Characterization

Define a *reduced weighted MSO-logic* (RMSO) similar to those for traces.

Theorem (B. Bollig & I.M. 2006)

For a commutative semiring $\mathbb{K}$ *and* $S : \mathbb{DAG}(\widetilde{\Sigma}) \to \mathbb{K}$ *are equivalent:*

1. $S = \|\mathcal{A}\|$ *for some wACA with types* $\mathcal{A}$*,*
2. S *is RMSO-definable.*

(direct proof with rather tricky constructions)

generalizations & applications for:
traces, message sequence charts (MSCs), probabilistic lossy-channel systems, probabilistic asynchronous automata

Summary

- gave logical characterization of recognizable trace series
- avoided to repeat the whole proof, used a translation to word series and an unambiguity result instead
- moreover, a characterization of the FO-fragment ▸ FO
- more general unifying frame by wACA for $\widetilde{\Sigma}$-dags

Outlook

- What class is defined by wMSO?
- find other weighted logics (temporal logics)
- case studies and practical relevance of quantitative aspects of concurrency
  - message sequence charts (international telecommunication standard)
  - probabilistic lossy-channel systems

Summary

- gave logical characterization of recognizable trace series
- avoided to repeat the whole proof, used a translation to word series and an unambiguity result instead
- moreover, a characterization of the FO-fragment ▸ FO
- more general unifying frame by wACA for $\widetilde{\Sigma}$-dags

Outlook

- What class is defined by wMSO?
- find other weighted logics (temporal logics)
- case studies and practical relevance of quantitative aspects of concurrency
  - message sequence charts (international telecommunication standard)
  - probabilistic lossy-channel systems

Translation Lemma – Proof Details

Proof (cont.)

$\mathrm{lex}(x,y)$ is an FO-formula.
(for dependence graphs transitive closure is FO-definable)

Critical:

- weighted semantics of $\mathrm{lex}(x,y)$ should be $\mathbb{1}$ or $\mathbb{0}$
- $\exists$ RFO-formula $\widehat{\mathrm{lex}(x,y)}$ with weighted semantics $\mathbf{1}_{L(\mathrm{lex}(x,y))}$?
  $\curvearrowright$ unambiguity ▸ unambiguity

Then we proceed and obtain an RMSO-formula $\tilde{\Phi}$ with

$$[\![\, \tilde{\Phi} \,]\!](t,\sigma) = [\![\, \Phi \,]\!](\mathrm{LNF}(t),\sigma).$$

$\Longrightarrow$ $\tilde{\Phi}$ defines $T : \mathbb{M} \to \mathbb{K}$. ◂ return

Translation Lemma – Proof Details

Proof (cont.)

$\mathrm{lex}(x,y)$ is an FO-formula.
(for dependence graphs transitive closure is FO-definable)

Critical:

- weighted semantics of $\mathrm{lex}(x,y)$ should be $\mathbb{1}$ or $\mathbb{0}$
- $\exists$ RFO-formula $\widehat{\mathrm{lex}(x,y)}$ with weighted semantics $\mathbf{1}_{L(\mathrm{lex}(x,y))}$?
  $\curvearrowright$ unambiguity ▸ unambiguity

Then we proceed and obtain an RMSO-formula $\tilde{\Phi}$ with

$$[\![\tilde{\Phi}]\!](t,\sigma) = [\![\Phi]\!](\mathrm{LNF}(t),\sigma).$$

$\Longrightarrow$ $\tilde{\Phi}$ defines $T : \mathbb{M} \to \mathbb{K}$.

◂ return

Translation Lemma – Proof Details

Proof (cont.)

$\mathrm{lex}(x,y)$ is an FO-formula.
(for dependence graphs transitive closure is FO-definable)

Critical:

- weighted semantics of $\mathrm{lex}(x,y)$ should be $\mathbb{1}$ or $\mathbb{0}$
- $\exists$ RFO-formula $\widehat{\mathrm{lex}(x,y)}$ with weighted semantics $\mathbf{1}_{L(\mathrm{lex}(x,y))}$?
  $\curvearrowright$ unambiguity ▸ unambiguity

Then we proceed and obtain an RMSO-formula $\tilde{\Phi}$ with

$$[\![\tilde{\Phi}]\!](t,\sigma) = [\![\Phi]\!](\mathrm{LNF}(t),\sigma).$$

$\Longrightarrow$ $\tilde{\Phi}$ defines $T : \mathbb{M} \to \mathbb{K}$. ◂ return

Unambiguity of FO-languages

$\mathcal{C}$ = class of finite relational structures
Let $\mathcal{C}$ have a *simply definable linear order*, i.e., $\exists$ propositional $\Omega(x, y)$ defining a linear order on the elements of every $t \in \mathcal{C}$.

Lemma

Let $L = L(\Phi) \subseteq \mathcal{C}$ for $\Phi \in \mathrm{FO}$. Then both $\mathbf{1}_L$ and $\mathbf{1}_{\overline{L}}$ are definable in RFO.

Corollary

L FO-definable trace language $\implies$ $\mathbf{1}_L$ RFO-definable.

$$\Omega(x,y) = \bigvee_{(a,b)\in\prec} \Big(P_a(x) \wedge P_b(y)\Big) \vee \bigvee_{a\in\Sigma} \Big(P_a(x) \wedge P_a(y) \wedge \neg E(y,x)\Big)$$

◂ main proof

Unambiguity of FO-languages

$\mathcal{C}$ = class of finite relational structures
Let $\mathcal{C}$ have a *simply definable linear order*, i.e., $\exists$ propositional $\Omega(x,y)$ defining a linear order on the elements of every $t \in \mathcal{C}$.

Lemma

Let $L = L(\Phi) \subseteq \mathcal{C}$ for $\Phi \in \mathrm{FO}$. Then both $\mathbf{1}_L$ and $\mathbf{1}_{\overline{L}}$ are definable in RFO.

Corollary

L FO-definable trace language $\Longrightarrow$ $\mathbf{1}_L$ RFO-definable.

$$\Omega(x,y) = \bigvee_{(a,b)\in\prec} \Big(P_a(x) \wedge P_b(y)\Big) \vee \bigvee_{a\in\Sigma} \Big(P_a(x) \wedge P_a(y) \wedge \neg E(y,x)\Big)$$

main proof

Unambiguity of FO-languages

$\mathcal{C}$ = class of finite relational structures

Let $\mathcal{C}$ have a *simply definable linear order*, i.e., $\exists$ propositional $\Omega(x,y)$ defining a linear order on the elements of every $t \in \mathcal{C}$.

Lemma

Let $L = L(\Phi) \subseteq \mathcal{C}$ for $\Phi \in \mathrm{FO}$. Then both $\mathbf{1}_L$ and $\mathbf{1}_{\overline{L}}$ are definable in RFO.

Corollary

L FO-definable trace language $\implies$ $\mathbf{1}_L$ RFO-definable.

$$\Omega(x,y) = \bigvee_{(a,b)\in\prec} \Big(P_a(x) \wedge P_b(y)\Big) \vee \bigvee_{a\in\Sigma} \Big(P_a(x) \wedge P_a(y) \wedge \neg E(y,x)\Big)$$

main proof

FO-definable Trace Series

Theorem

$\mathbb{K}$ *commutative & weakly bi-aperiodic semiring.*
For $T : \mathbb{M} \to \mathbb{K}$ *are equivalent:*

1. T *is* RFO*-definable,*
2. T *is* FO*-definable,*
3. T *is aperiodic,*
4. T *is weakly aperiodic.*

$\mathbb{K}$ *weakly bi-aperiodic* if $(K, \oplus, \mathbb{0})$ and $(K, \circ, \mathbb{1})$ weakly aperiodic,

$S = (\lambda, \mu, \gamma)$ *aperiodic* if $\mu(M)$ aperiodic,

$S = (\lambda, \mu, \gamma)$ *weakly aperiodic* if
$\exists n \geq 0 \; \forall u, v, w \in M \; (S, uv^n w) = (S, uv^{n+1} w)$

summary

FO-definable Trace Series

Theorem

$\mathbb{K}$ commutative & weakly bi-aperiodic semiring.
For $T : \mathbb{M} \to \mathbb{K}$ are equivalent:

1. *T is RFO-definable,*
2. *T is FO-definable,*
3. *T is aperiodic,*
4. *T is weakly aperiodic.*

$\mathbb{K}$ *weakly bi-aperiodic* if $(K, \oplus, \mathbb{0})$ and $(K, \circ, \mathbb{1})$ weakly aperiodic,

$S = (\lambda, \mu, \gamma)$ *aperiodic* if $\mu(M)$ aperiodic,

$S = (\lambda, \mu, \gamma)$ *weakly aperiodic* if
$\exists n \geq 0\ \forall u, v, w \in M\ (S, uv^n w) = (S, uv^{n+1} w)$

summary