Emmanuel Filiot

Nicolas Mazzocchi

Jean-François Raskin

Université libre de Bruxelles

FCT 2017 – Bordeaux

Decidable Weighted Expressions with Presburger Combinators

Boolean vs Quantitative Languages

$$L : \Sigma^* \to \{0, 1\}$$

Classical decision problems

Emptiness	$\exists u. f(u) \geq 1$
Universality	$\forall u. f(u) \geq 1$
Inclusion	$\forall u. f(u) \geq g(u)$
Equivalence	$\forall u. f(u) = g(u)$

Boolean vs Quantitative Languages

$$L : \Sigma^* \to \cancel{\{0, 1\}}\ \mathbb{Z} \cup \{-\infty\}$$

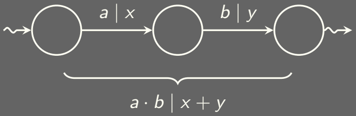
Classical quantitative decision problems

Emptiness	$\exists u. f(u) \geq \cancel{1}\ \nu$	for some threshold ν
Universality	$\forall u. f(u) \geq \cancel{1}\ \nu$	for some threshold ν
Inclusion	$\forall u. f(u) \geq g(u)$	
Equivalence	$\forall u. f(u) = g(u)$	

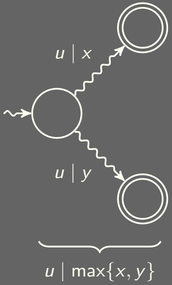
Classical Model: Weighted Automata

(max,+) WA

Transition sequence



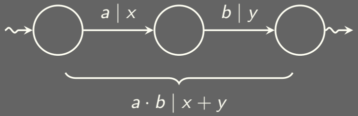
Non-determinism



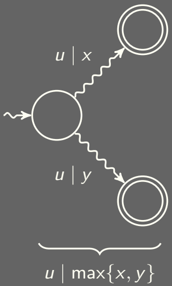
Classical Model: Weighted Automata

(max,+) WA

Transition sequence



Non-determinism



Undecidability [Krob 1994]

Quantitative language-inclusion is undecidable for (max,+) WA

- Even for linearly ambiguous automata [Colcombet 2010]

Decidable Formalisms: Restriction

Finitely ambiguous (max,+) WA

Define functions of the form,

$$u \mapsto \max\{\mathcal{A}_1(u), \ldots, \mathcal{A}_k(u)\}$$

$\mathcal{A}_i$: Unambiguous WA

- ☺ Quantitative decision problems are DECIDABLE [Filiot et al. 2012]
- ☺ Closed under *max* and *sum*
- ☹ Limited expressive power (*min*, *minus*, ...)

Decidable Formalisms: New model

Mean-payoff expressions [Chatterjee et al. 2010]

$$E ::= \mathcal{A} \mid \max(E, E) \mid \min(E, E) \mid E + E \mid -E$$

$\mathcal{A}$: Deterministic WA

- ☺ Quantitative decision problems are PSPACE-COMPLETE [Velner 2012]
- ☺ Closed under *max*, *min*, *sum* and *minus*
- ☹ Determinism (define Lipschitz continuous functions)
- ☹ Does **not** contain all finitely ambiguous (max,+) WA
- ☹ Monolithism (apply on the whole word)

Contributions

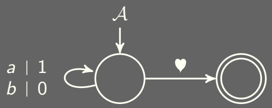
❶ Simple expressions

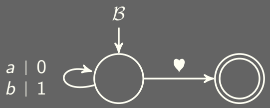
$$E ::= \mathcal{A} \mid \phi(E, E)$$

$\mathcal{A}$: Unambiguous WA

ϕ : $\exists$FO$[\leq, +, 0, 1]$ formula defining function with arity two

Example





$$E = \max(\mathcal{A} - \mathcal{B}, \mathcal{B} - \mathcal{A})$$

$$u \mapsto |\mathcal{A}(u) - \mathcal{B}(u)|$$

Contributions

❶ Simple expressions

$$E ::= \mathcal{A} \mid \phi(E, E)$$

$\mathcal{A}$: Unambiguous WA
ϕ : $\exists$FO$[\leq, +, 0, 1]$ formula defining function with arity two

- ☺ Quantitative decision problems are PSPACE-COMPLETE
- ☺ Closed under Presburger definable functions
- ☺ Contain all finitely ambiguous (max,+) WA
- ☹ Monolithism (apply on the whole word)

Contributions

❷ Iterable expressions

$$E ::= \mathcal{A} \mid \phi(E, E) \mid E^{\circledast}$$

- Unique decomposition required
- Sum arbitrarily many factors

Examples

$E^{\circledast}$	$u_1 \heartsuit\, u_2 \heartsuit \ldots u_n \heartsuit \mapsto \sum_{i=1}^{n} E(u_i)$
$\phi(E^{\circledast}, F^{\circledast})$	$u \mapsto \phi\left(\sum_{i=1}^{n} E(u_i)\ ,\ \sum_{j=1}^{m} F(v_j)\right)$

Undecidability

Theorem

Quantitative decision problems are UNDECIDABLE *for iterable expressions*

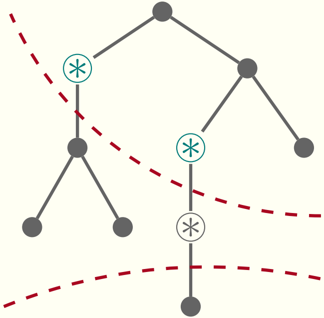
Proof by reduction from the 2-counter machine halting problem

$$\begin{array}{ll} \textit{Run} & \ldots (q_1, (x \mapsto c_1, y \mapsto d_1)) \, (q_2, (x \mapsto c_2, y \mapsto d_2)) \ldots \\ \textit{Input} & \cdots \vdash q_1 a^{c_1} b^{d_1} \lhd \rhd q_2 a^{c_2} b^{d_2} \dashv\vdash q_2 a^{c_2} b^{d_2} \lhd \ldots \end{array}$$

- regular constraints: initial, final, transitions, $(\vdash Qa^* b^* \rhd\lhd Qa^* b^* \dashv)^*$
- copy: E on factors $\rhd \cdots \lhd$ return 0 if correct and < 0 otherwise
- incr. / decr.: F on factors $\vdash \cdots \dashv$ return 0 if correct < 0 otherwise
- decide: $E^{\circledast} + F^{\circledast} \geq 0$

✓

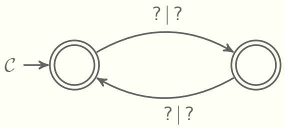
Synchronisation of expressions



Theorem

- *Synchronisation property is decidable in* PTIME
- *Synchronised iterable-expression are* DECIDABLE

Weighted Chop Automata

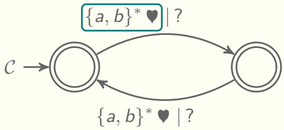


New model

- Generalise unambiguous WA
- Recursive definition

Weighted Chop Automata

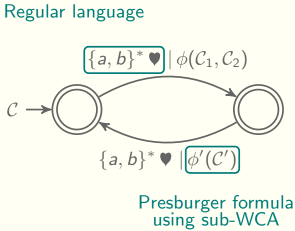
Regular language



New model

- Generalise unambiguous WA
- Recursive definition

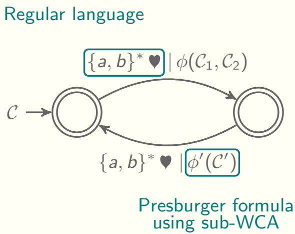
Weighted Chop Automata



New model

- Generalise unambiguous WA
- Recursive definition

Weighted Chop Automata



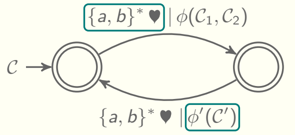
New model

- Generalise unambiguous WA
- Recursive definition

Example

input : $aa\heartsuit\, ba\heartsuit$
dec : $(aa\heartsuit, \phi(\mathcal{C}_1, \mathcal{C}_2))(ba\heartsuit, \phi'(\mathcal{C}'))$
val : $\phi\left(\mathcal{C}_1(aa\heartsuit), \mathcal{C}_2(aa\heartsuit)\right) + \phi'(\mathcal{C}'(ba\heartsuit)$

Weighted Chop Automata



New model

- Generalise unambiguous WA
- Recursive definition

Example

input : $aa\heartsuit\, ba\heartsuit$
dec : $(aa\heartsuit, \phi(\mathcal{C}_1, \mathcal{C}_2))(ba\heartsuit, \phi'(\mathcal{C}'))$
val : $\phi\,(\mathcal{C}_1(aa\heartsuit), \mathcal{C}_2(aa\heartsuit)) + \phi'(\mathcal{C}'(ba\heartsuit)$

Operators for expressiveness equivalence

$$\left.\begin{array}{l} E \odot F : u_1 u_2 \mapsto E(u_1) + F(u_2) \\ E \rhd F : u \mapsto \textbf{if } u \in \mathrm{dom}(E) \textbf{ then } E(u) \textbf{ else } F(u) \end{array}\right\} \text{[Alur 2014]}$$

Synchronisation of WCA

Definition of $C_1 \parallel C_2$

Given C_1, C_2 two k-WCA, for all $u \in \mathrm{dom}(C_1) \cap \mathrm{dom}(C_2)$ such that

$$\mathrm{dec}_{C_1}(u) = (u_1, \phi_1)(u_2, \phi_2) \ldots (u_n, \phi_n)$$
$$\mathrm{dec}_{C_2}(u) = (v_1, \psi_1)(v_2, \psi_2) \ldots (v_m, \psi_m)$$

then $n = m$, $u_i = v_i$ and $\phi_i \parallel \psi_i$

Proposition

The product is well defined for synchronised k-WCA

$$p \xrightarrow{L|\phi} q \quad , \quad p' \xrightarrow{L'|\phi'} q' \rightsquigarrow (p, p') \xrightarrow{L \cap L' | (\phi, \phi')} (q, q')$$

Decidability

Theorem

The range of a synchronised WCA is semilinear

Proof by structural induction on the WCA.

- Base case: Simple expressions have a semilinear range
- Induction step: Assume $p \xrightarrow{L_{p,q} | \phi(C_1,\ldots,C_n)} q$ have semilinear range $S_{p,q}$
  1. Morphism μ from $(Q \times Q)^*$ to the monoid of semilinear sets:
     $\mu((p,q)) = S_{p,q}$
     If L regular then $\mu(L)$ is semilinear
     Take L as the set of accepting run, regular since all $L_{p,q}$ are
  2. $\prod_{i=1}^{n} C_i$ has a semilinear range $S_C \subseteq \mathbb{Z}^n$ by induction on n
     $\phi(x_1,\ldots,x_n)$ has a semilinear range $S_\phi \subseteq \mathbb{Z}^n$
     Take $S_\phi \cap S_C$

✓

Synchronised translation

Theorem

A quantitative language realised by a synchronised iterable-expression can be realised by a synchronised WCA

Proof by structural induction of the expression

- A: obvious
- $\phi(E_1, E_2)$: construct C_1, C_2 by induction such that $C_1 \parallel C_2$ then define C as $q_0 \xrightarrow{\mathrm{dom}(C_1) \cap \mathrm{dom}(C_2) | \phi(C_1, C_2)} q_1$
- $(A, E^{\circledast})$:
  1. chop A into smaller WA $A_{s,t}$ restricted to $\mathrm{dom}(E)$
  2. construct all $C_{s,t}$ and C by induction
  3. define C_A as $p \xrightarrow{L(A_{p,q}) | \phi_{id}(C_{p,q})} q$ (same set of states than A)
  4. define $C^{\circledast}$ with the trivial loop $q_0 \xrightarrow{\mathrm{dom}(E) | \phi_{id}(C)} q_0$

✓

Conclusion

Summary

Simple expressions: PSPACE-COMPLETE
Sum-iterable expressions: UNDECIDABLE
Synchronised sum-iterable expressions: DECIDABLE

Perspective

Iterate other operations (*max*, Presburger definable functions, . . .)

$$\underbrace{\begin{array}{cccc} r\,r\,r\,r\,g & r\,r\,r\,r\,r\,r\,r\,g & \ldots & r\,r\,r\,r\,r\,g \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 4 & 7 & r^n g \mapsto n & 5 \end{array}}_{\text{iterate max}}$$

Thanks!