Weighted and counter systems

Marc Zeitoun

Joint with Benedikt, Paul, Benjamin

LaBRI, U. Bordeaux, LSV, ENS Cachan, CNRS, INRIA

Barbizon 2011

Let's begin with the end

Thank you!

(Pot de départ fin juin)

Recipe: How to ask for a survey

1. Ask kindly for a talk (don't be too demanding)

P. Would you like to give a talk for Barbizon?

M. Sure. I'll be short in time, may I reuse an already prepared talk?

P. Perfect!

Recipe: How to ask for a survey

1. Ask kindly for a talk (don't be too demanding)

P. Would you like to give a talk for Barbizon?

M. Sure. I'll be short in time, may I reuse an already prepared talk?

P. Perfect!

2. Wait 10 days

P. A survey would be *much* better, can you please give a title?

M. . . .

Counter and Weighted Systems

	Counter (mainly ∞)	Weighted (∞, Tempo, Mexico)
What for?	Represent runs of machines	Resources/functions
Motivation	MC for ∞ systems, complexity	Quantitative MC

Counter and Weighted Systems

	Counter (mainly ∞)	Weighted (∞, Tempo, Mexico)
What for?	Represent runs of machines	Resources/functions
Motivation	MC for ∞ systems, complexity	Quantitative MC
Typical questions	MC. Reach, cover, TL,...	MC. Comparison, bound.
Typical tools	Linear algebra, unfoldings, wqo	Automata, games, algebra

Counter and Weighted Systems

	Counter (mainly ∞)	Weighted (∞, Tempo, Mexico)
What for?	Represent runs of machines	Resources/functions
Motivation	MC for ∞ systems, complexity	Quantitative MC
Typical questions	MC. Reach, cover, TL,...	MC. Comparison, bound.
Typical tools	Linear algebra, unfoldings, wqo	Automata, games, algebra

This talk: presentation of weighted automata on words.

Two popular types of Weighted Automata

1. Probabilistic automata.
2. Min-+ automata.

Can be presented uniformly.

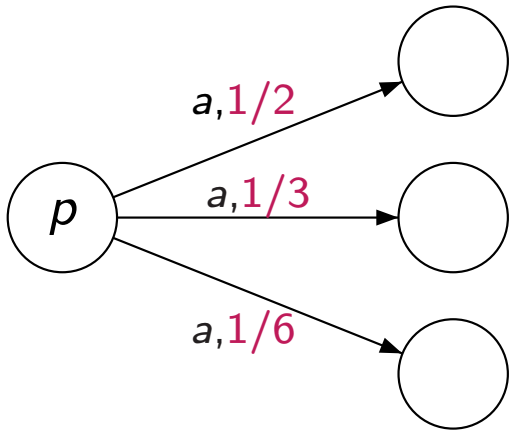
No uniform algorithms for typical questions.

Minsky machines vs. Probabilistic Automata

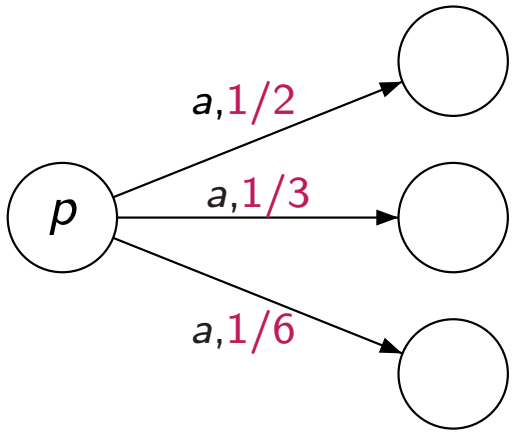
- Crux: Counters affect runs in Minsky machines, not in probabilistic automata.
- Runs are aggregated in probabilistic automata to compute a function.

	Minsky	Prob. Automaton
Guards	$= 0?$	None
Updates along transition	$+1, -1$	$\times \alpha$
Semantic of nondeterminism	Boolean	Sum

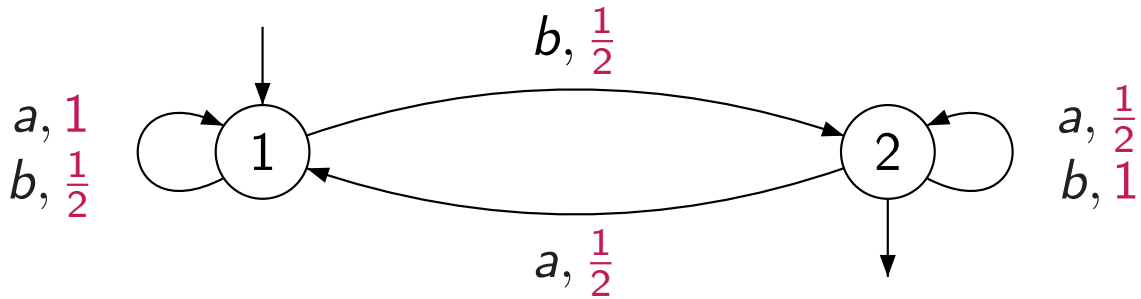
Probabilistic Automata



Probabilistic Automata



- Weight (probability) of a run: product of all transition weights.
  word: sum of weights of all successful runs.



- $[\![\mathcal{A}]\!](ba) = 1/4$
- Interpreting (a, b) as $(0, 1)$: $[\![\mathcal{A}]\!](a_1 \cdots a_n) = 0.a_n \cdots a_1$

3 Questions for Probabilistic Automata

1. Equality: $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$?

   Decidable (Schützenberger'61, Tzeng'92) Minimal automaton.

3 Questions for Probabilistic Automata

1. Equality: $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$?

   Decidable (Schützenberger'61, Tzeng'92) Minimal automaton.

2. Boundedness: $[\![\mathcal{A}]\!] < 1/2$?

   Undecidable (Paz'71) PCP

   From morphisms $f, g : \{0,1\}^* \to \{0,1\}^*$, compute $1/2[1 - (\overline{f(w)} - \overline{g(w)})]$

3 Questions for Probabilistic Automata

1. Equality: $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$?

   Decidable (Schützenberger'61, Tzeng'92) Minimal automaton.

2. Boundedness: $[\![\mathcal{A}]\!] < 1/2$?

   Undecidable (Paz'71) PCP

   From morphisms $f, g : \{0,1\}^* \to \{0,1\}^*$, compute $1/2[1 - (\overline{f(w)} - \overline{g(w)})]$

3. 0 value: are there words accepted with arbitrarily small probability?

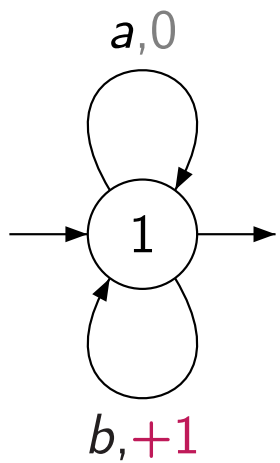
   Undecidable (Gimbert & Oualhadj'10)

Min-+ Automata

- Costs in $\mathbb{N} \cup \{\infty\}$.
- Weight (cost) of a run: sum of all transition weights.

  word: min of weights of all successful runs.

Weight of a word u = minimal cost for going from Init to Final reading u.



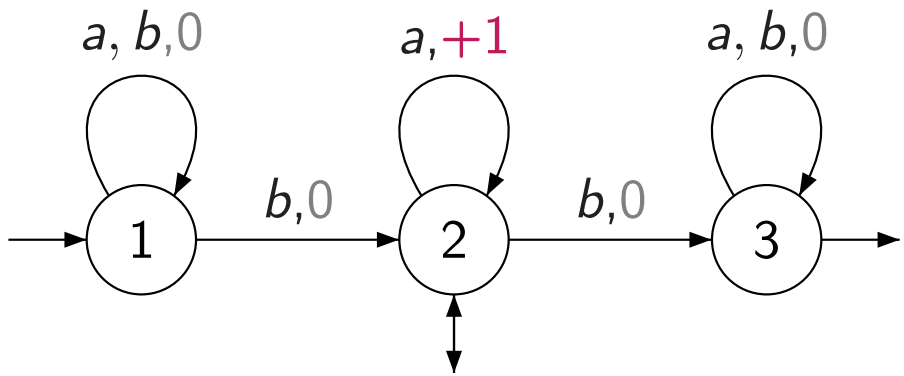
$$[\![\mathcal{A}]\!](u) = |u|_b$$

Min-+ Automata

- Costs in $\mathbb{N} \cup \{\infty\}$.
- Weight (cost) of a run: sum of all transition weights.

  word: min of weights of all successful runs.

Weight of a word u = minimal cost for going from Init to Final reading u.



$[\![\mathcal{B}]\!](aaaabaabaaa) = 2$

$[\![\mathcal{B}]\!](u)$: min length of block of a's

From Probabilistic Automata to min-+ Automata

1. Transform product into sum: apply $\log \rightsquigarrow$ high probabilities yield small distances

$$([0,1],+,\times) \xrightarrow{-\log} (\mathbb{R}_{\geq 0} \cup \infty, \oplus, +)$$
$$xy \rightsquigarrow -\log(xy) = -\log(x) + -\log(y)$$
$$x + y \rightsquigarrow -\log(x+y)$$

From Probabilistic Automata to min-+ Automata

1. Transform product into sum: apply $\log \rightsquigarrow$ high probabilities yield small distances
2. Approximate new sum unsing min: Viterbi's approximation.

$$([0,1],+,\times) \xrightarrow{-\log} (\mathbb{R}_{\geq 0} \cup \infty, \oplus, +)$$
$$xy \rightsquigarrow -\log(xy) = -\log(x) + -\log(y)$$
$$x + y \rightsquigarrow -\log(x+y)$$

$$\left| \log(x+y) - \min(-\log(x), -\log(y)) \right| \leq \log 2$$

From Probabilistic Automata to min-+ Automata

1. Transform product into sum: apply $\log \rightsquigarrow$ high probabilities yield small distances
2. Approximate new sum unsing min: Viterbi's approximation.

$$([0,1],+,\times) \xrightarrow{-\log} (\mathbb{R}_{\geq 0} \cup \infty, \oplus, +) \xrightarrow{\text{Viterbi}} (\mathbb{R}_{\geq 0} \cup \infty, \min, +).$$

$$xy \rightsquigarrow -\log(xy) = -\log(x) + -\log(y)$$

$$x + y \rightsquigarrow -\log(x+y)$$

$$\left| \log(x+y) - \min(-\log(x), -\log(y)) \right| \leq \log 2$$

The 3 Questions after log-transforming

Probabilistic		Min-+
$[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$	D	$[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$
$[\![\mathcal{A}]\!] > \alpha$	U	$[\![\mathcal{A}]\!] \leq k$
0-value	U	Is $[\![\mathcal{A}]\!]$ bounded?

- Limitedness (refinement of boundedness): Is $[\![\mathcal{A}]\!]$ bounded over its support?

The 3 Questions after log-transforming

Probabilistic		Min-+	
$[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$	D	$[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$	U
$[\![\mathcal{A}]\!] > \alpha$	U	$[\![\mathcal{A}]\!] \leq k$	D
0-value	U	Is $[\![\mathcal{A}]\!]$ bounded?	D

- Limitedness (refinement of boundedness): Is $[\![\mathcal{A}]\!]$ bounded over its support?

The 3 Questions after log-transforming

Probabilistic		Min-+	
$[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$	D	$[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$	U
$[\![\mathcal{A}]\!] > \alpha$	U	$[\![\mathcal{A}]\!] \leq k$	D
0-value	U	Is $[\![\mathcal{A}]\!]$ bounded?	D

- Limitedness (refinement of boundedness): Is $[\![\mathcal{A}]\!]$ bounded over its support?

Problems very sensitive to underlying semiring.

Min-+ Automata: Results

- Equality $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$: Undecidable! [Krob'94].
  New and simple proof by Th. Colcombet (Minsky machines).

- Given k, is $[\![\mathcal{A}]\!] \leq k$ decidable (easy).

- Limitedness: Is $[\![\mathcal{A}]\!]$ *bounded* on its support? Decidable
  Much work, eg [Hashigushi'81, Simon, Leung, Kirsten, Colcombet]

Min-+ Automata: Results

- Equality $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$: Undecidable! [Krob'94].
  New and simple proof by Th. Colcombet (Minsky machines).

- Given k, is $[\![\mathcal{A}]\!] \leq k$ decidable (easy).

- Limitedness: Is $[\![\mathcal{A}]\!]$ *bounded* on its support? Decidable
  Much work, eg [Hashigushi'81, Simon, Leung, Kirsten, Colcombet]

  Note: Boundedness in min-+ is more complicated than for VASS:

  VASS: $\exists B\ \forall w\ \forall \rho \in \mathrm{Runs}(w) : \quad \mathrm{Val}(\rho) \leqslant B.$

Min-+ Automata: Results

- Equality $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$: Undecidable! [Krob'94].
  New and simple proof by Th. Colcombet (Minsky machines).

- Given k, is $[\![\mathcal{A}]\!] \leq k$ decidable (easy).

- Limitedness: Is $[\![\mathcal{A}]\!]$ *bounded* on its support? Decidable
  Much work, eg [Hashigushi'81, Simon, Leung, Kirsten, Colcombet]

  Note: Boundedness in min-+ is more complicated than for VASS:

  VASS: $\exists B\ \forall w\ \forall \rho \in \mathrm{Runs}(w) : \quad \mathrm{Val}(\rho) \leqslant B.$

  min-+: $\exists B\ \forall w\ \exists \rho \in \mathrm{Runs}(w) : \quad \mathrm{Val}(\rho) \leqslant B.$

Limitedness: Motivation

Finite power property

- Given L regular,

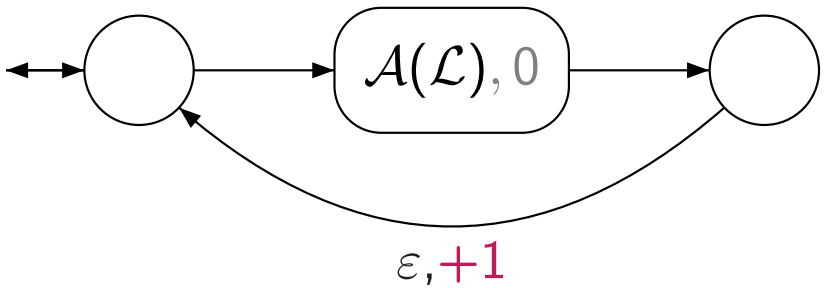
  Decide if there exists $n \geq 0$ such that $L^* = (\varepsilon + L)^n$.

Limitedness: Motivation

Finite power property

- Given L regular,

  Decide if there exists $n \geq 0$ such that $L^* = (\varepsilon + L)^n$.
- Thomson's construction for star:



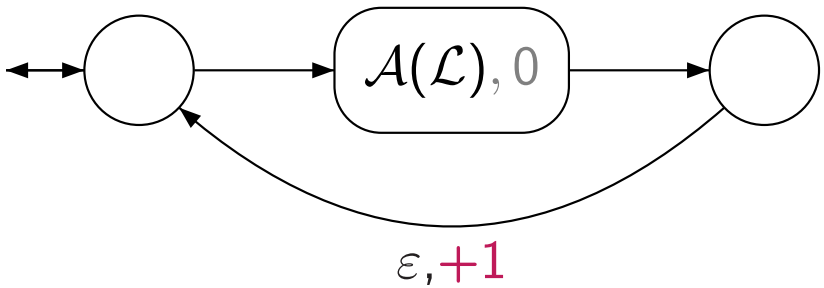
New automaton is limited iff. L has the finite power property.

Limitedness: Motivation

Finite power property

- Given L regular,

  Decide if there exists $n \geq 0$ such that $L^* = (\varepsilon + L)^n$.
- Thomson's construction for star:

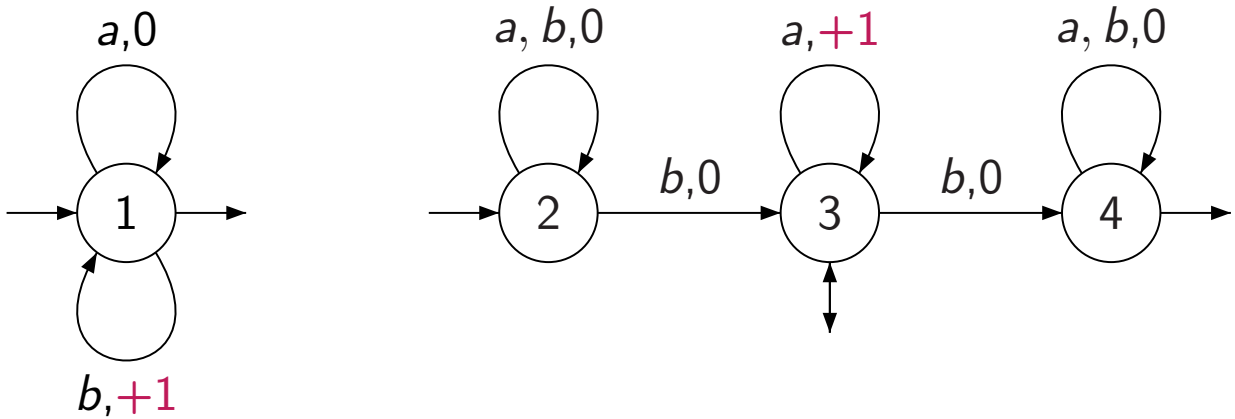


New automaton is limited iff. L has the finite power property.

- Generalization to multiple counters (Kirsten) $\rightsquigarrow$ star-height problem.
- Note: universality is a particular case of limitedness

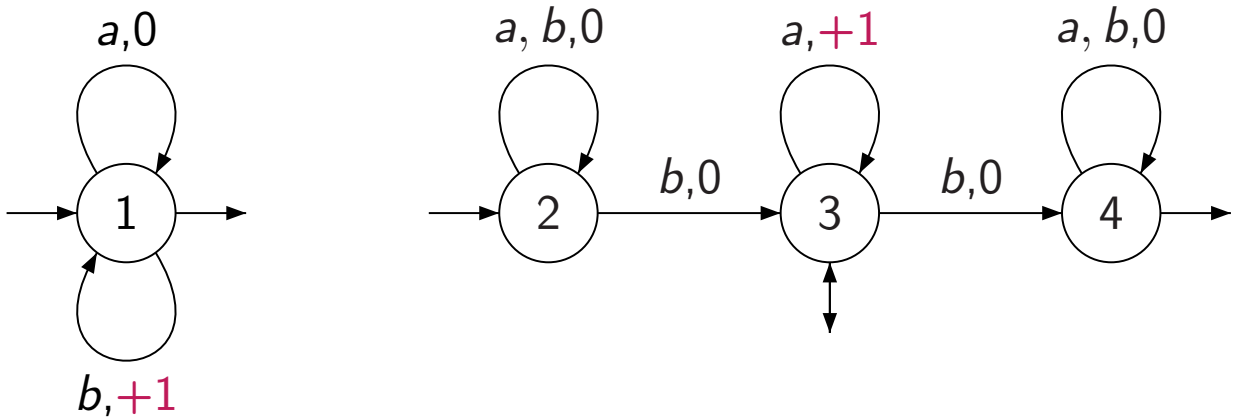
  ($\Longrightarrow$ limitedness is PSPACE-hard).

Limitedness is decidable



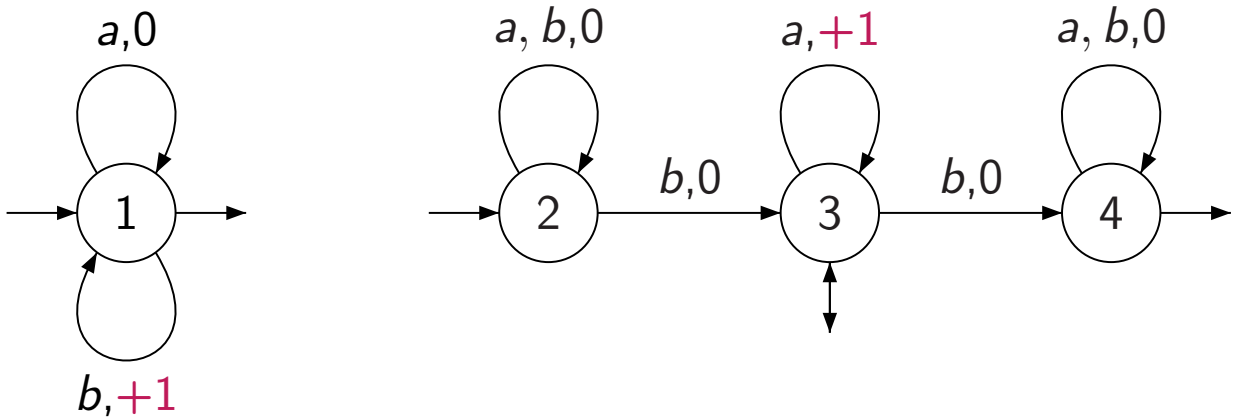
- Transformation monoid: $a = \begin{pmatrix} 0 & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty \\ \infty & \infty & 1 & \infty \\ \infty & \infty & \infty & 0 \end{pmatrix}$, $b = \begin{pmatrix} 1 & \infty & \infty & \infty \\ \infty & 0 & 0 & \infty \\ \infty & \infty & \infty & 0 \\ \infty & \infty & \infty & 0 \end{pmatrix}$
- Problem: the monoid generated is infinite.
- Solution: Abstract by projecting onto $\{0, 1, \infty\}$: $0 \mapsto 0$, $n \mapsto 1$, $\infty \mapsto \infty$.

Limitedness is decidable



- Loss of precision in the abstraction: $aa = a = \begin{pmatrix} 0 & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty \\ \infty & \infty & 1 & \infty \\ \infty & \infty & \infty & 0 \end{pmatrix}$.
- Add a new value ω meaning "unbounded".
- New operation $\#$ "iterate a large number of times": $a^{\#} = \begin{pmatrix} 0 & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty \\ \infty & \infty & \omega & \infty \\ \infty & \infty & \infty & 0 \end{pmatrix}$
- Think of $(a^{\#}b)^{\#}$ as $(a^n b)^n$ for large n.

Limitedness is decidable



- Algorithm: compute the $(\#,\cdot)$ monoid generated by letter matrices.
- If for some matrix M, $\min(\text{Init}, M, \text{Final}) = \omega$: witness of non-limitedness.

$$\text{Example: } (a^{\#}b)^{\#}a^{\#} = \begin{pmatrix} \omega & \infty & \infty & \infty \\ \infty & 0 & \omega & \omega \\ \infty & \infty & \infty & \omega \\ \infty & \infty & \infty & 0 \end{pmatrix}$$

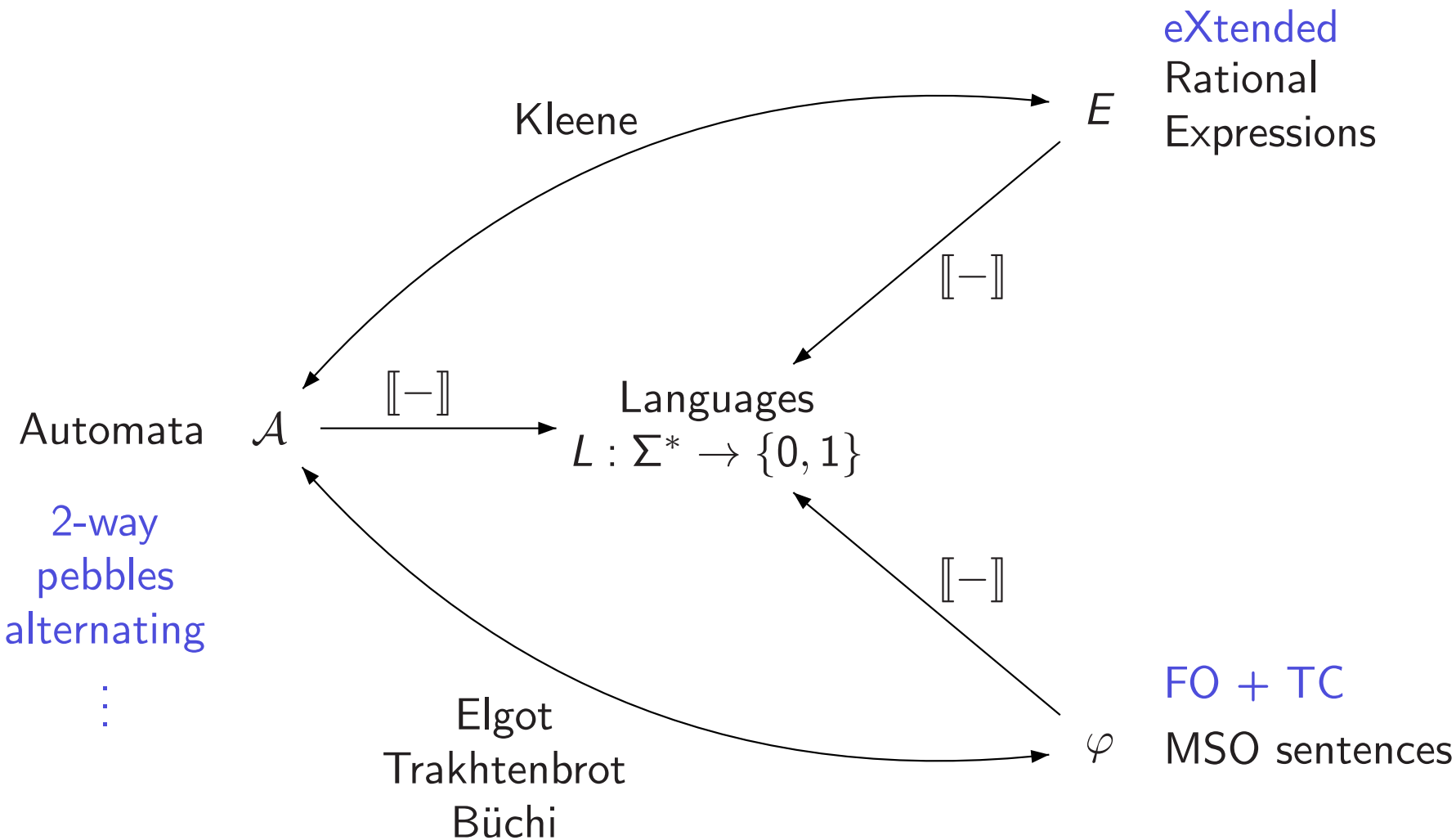
- Otherwise: this shows that "regular behaviors" à la $(a^n b)^n$ are limited.
  Difficult: show that all behaviors are also limited. Algebraic tools of I. Simon

Specifying quantitative properties

Develop high-level denotational formalism

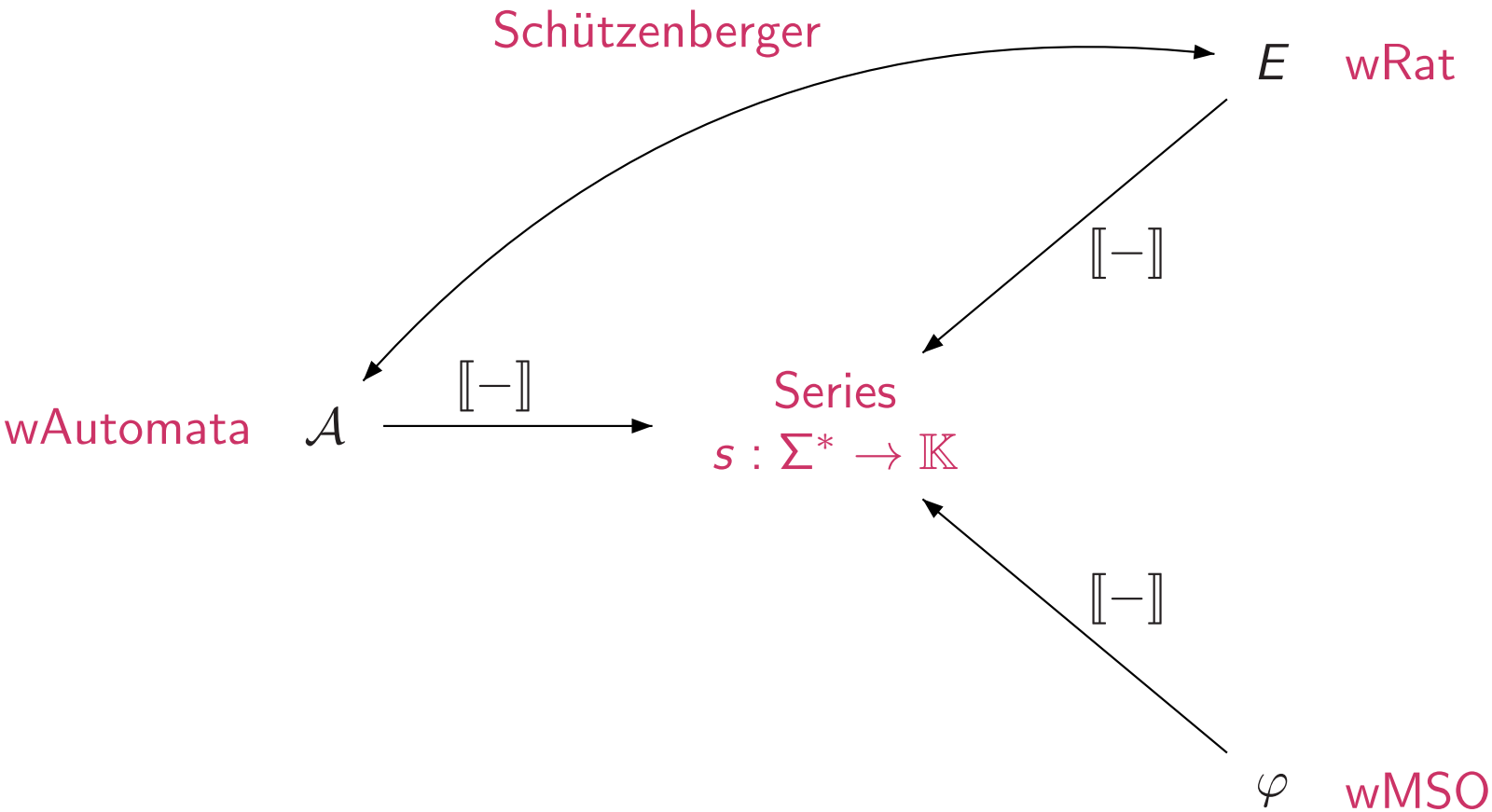
- to express easily quantitative properties on words/trees,
- should allow us to compute arithmetic expressions (possibly guarded by logical conditions written in a standard language (eg., FO or XPath),
- should have an equivalent operational model.

On words: what remains true for weights?



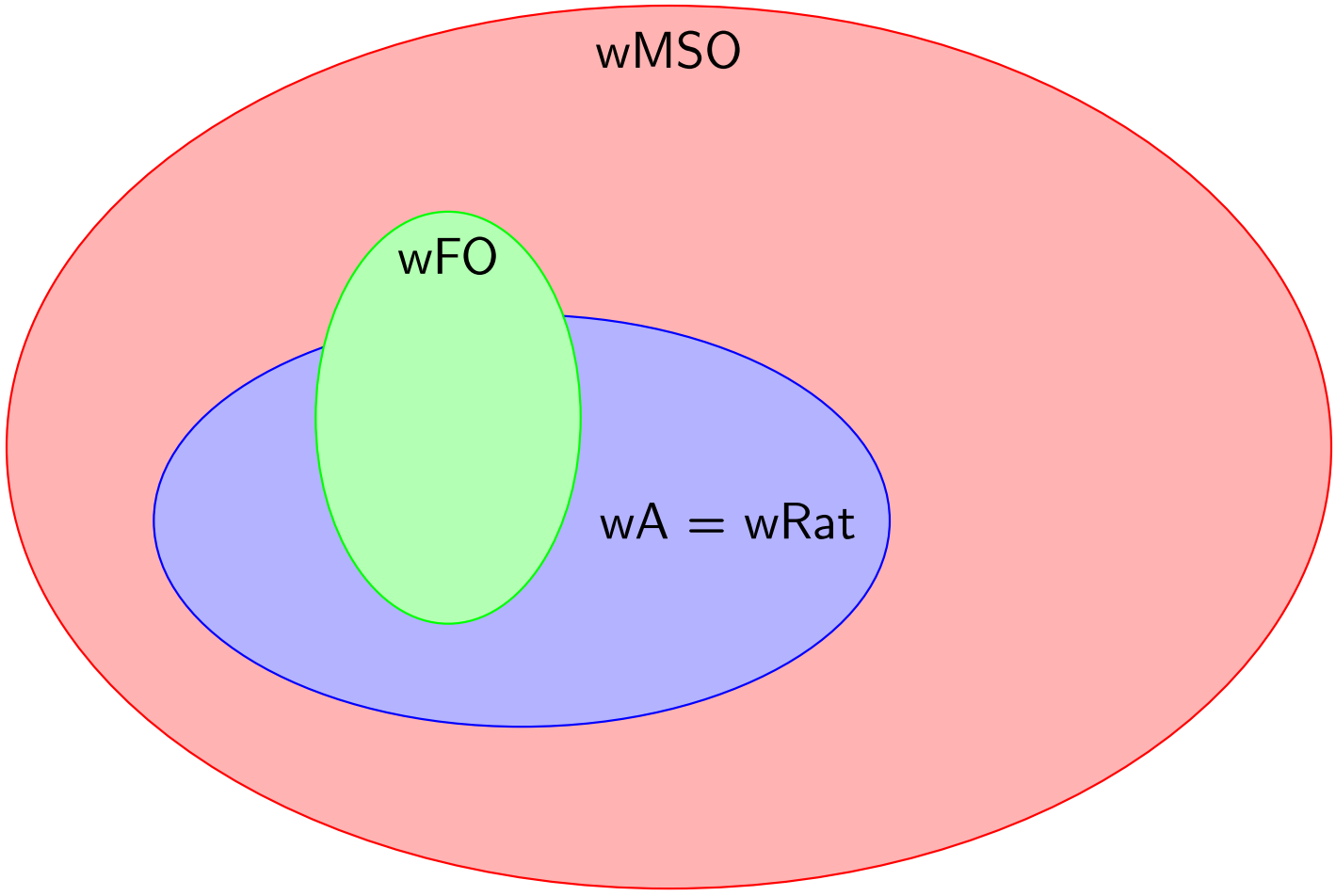
Boolean: $\mathbb{B} = (\{0,1\}, \vee, \wedge, 0, 1)$

On words: what remains true for weights?

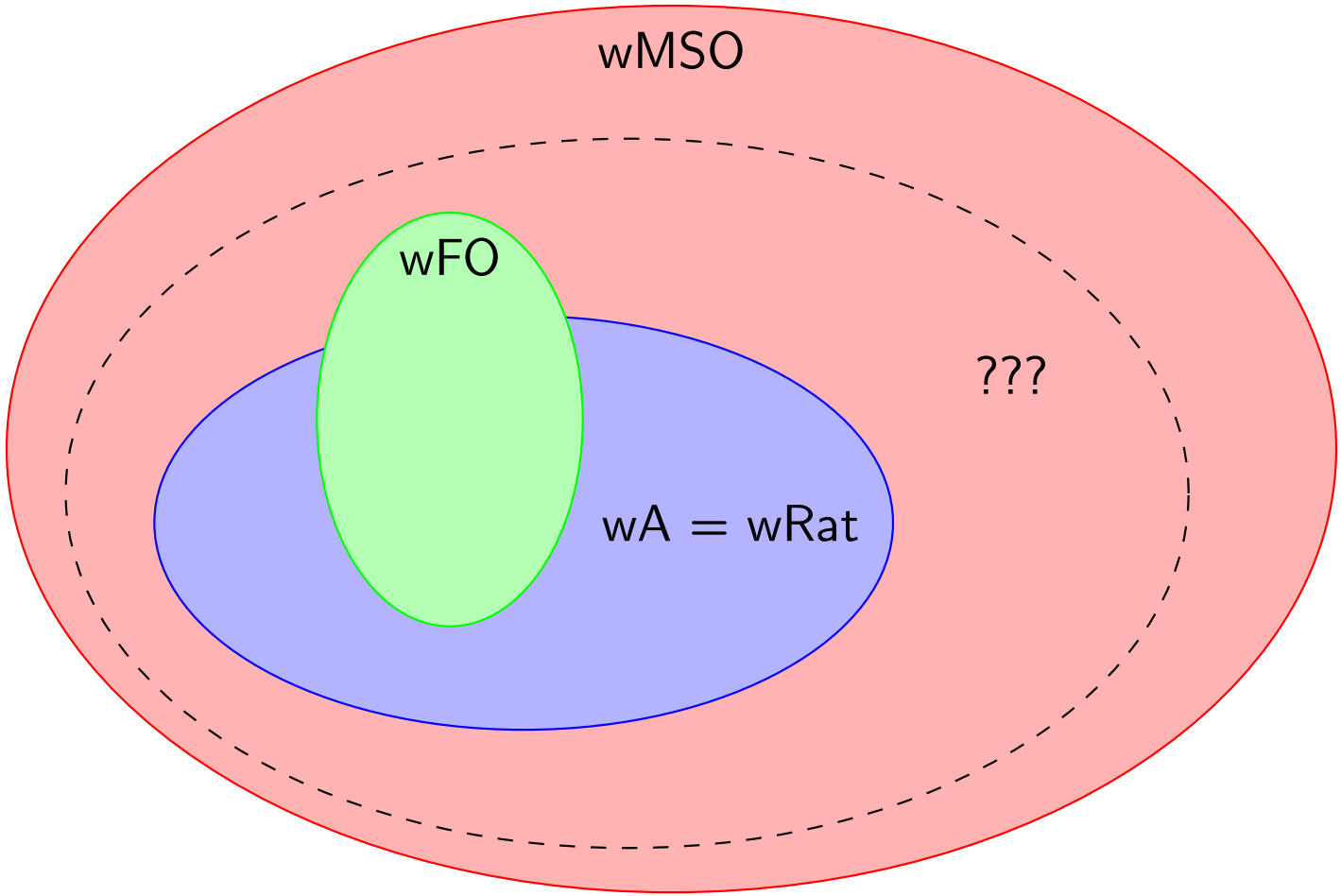


Quantitative: $\mathbb{K} = (K, +, \times, 0, 1)$

Expressiveness in the weighted setting



Expressiveness in the weighted setting



Find a robust class containing both wFO and wAutomata.

Weighted Expressions

Syntax of WE($\mathcal{L}$)

Fix $\mathcal{L}$ a logic (eg, MSO, FO($<$)).

$$E ::= \varphi \mid k \mid E \oplus E \mid E \otimes E \mid \bigoplus_x E \mid \bigotimes_x E$$

where $\varphi \in \mathcal{L}$, $k \in K$, x is a first-order variable.

Semantics

- An expression E without free variables defines a mapping $[\![\varphi]\!] : \Sigma^+ \to K$.
- For $\varphi \in \mathcal{L}$, we have $[\![\varphi]\!](w) \in \{0, 1\}$ (in the chosen semiring).

Weighted Expressions

Syntax of WE($\mathcal{L}$)

Fix $\mathcal{L}$ a logic (eg, MSO, FO($<$)).

$$E ::= \varphi \mid k \mid E \oplus E \mid E \otimes E \mid \bigoplus_x E \mid \bigotimes_x E$$

where $\varphi \in \mathcal{L}$, $k \in K$, x is a first-order variable.

Semantics

- An expression E without free variables defines a mapping $[\![\varphi]\!] : \Sigma^+ \to K$.
- For $\varphi \in \mathcal{L}$, we have $[\![\varphi]\!](w) \in \{0, 1\}$ (in the chosen semiring).
- $\bigoplus_x \varphi$ interpreted as a sum over all positions.
- $\bigotimes_x \varphi$ interpreted as a product over all positions.

Weighted expressions: examples

On $(\mathbb{N}, +, \times)$:

- $[\![\bigoplus_x a(x)]\!](u) = \sum_{i \in \mathrm{pos}(u)} [\![a(x)]\!](u, i) = |u|_a$ recognizable by wA

Weighted expressions: examples

On $(\mathbb{N}, +, \times)$:

- $[\![\bigoplus_x a(x)]\!](u) = \sum_{i \in \mathrm{pos}(u)} [\![a(x)]\!](u, i) = |u|_a$ recognizable by wA
- $[\![\bigotimes_y 2]\!](u) = \prod_{i \in \mathrm{pos}(u)} [\![2]\!](u, i) = 2^{|u|}$ recognizable by wA

Weighted expressions: examples

On $(\mathbb{N}, +, \times)$:

- $[\![\bigoplus_x a(x)]\!](u) = \sum_{i \in \mathrm{pos}(u)} [\![a(x)]\!](u, i) = |u|_a$ recognizable by wA
- $[\![\bigotimes_y 2]\!](u) = \prod_{i \in \mathrm{pos}(u)} [\![2]\!](u, i) = 2^{|u|}$ recognizable by wA
- $[\![\bigotimes_x \bigotimes_y 2]\!](u) = \prod_{i \in \mathrm{pos}(u)} [\![\bigotimes_y 2]\!](u, i) = (2^{|u|})^{|u|} = 2^{|u|^2}$ not recognizable

w-Automata are not closed under $\bigotimes$.

Capturing Weighted Automata

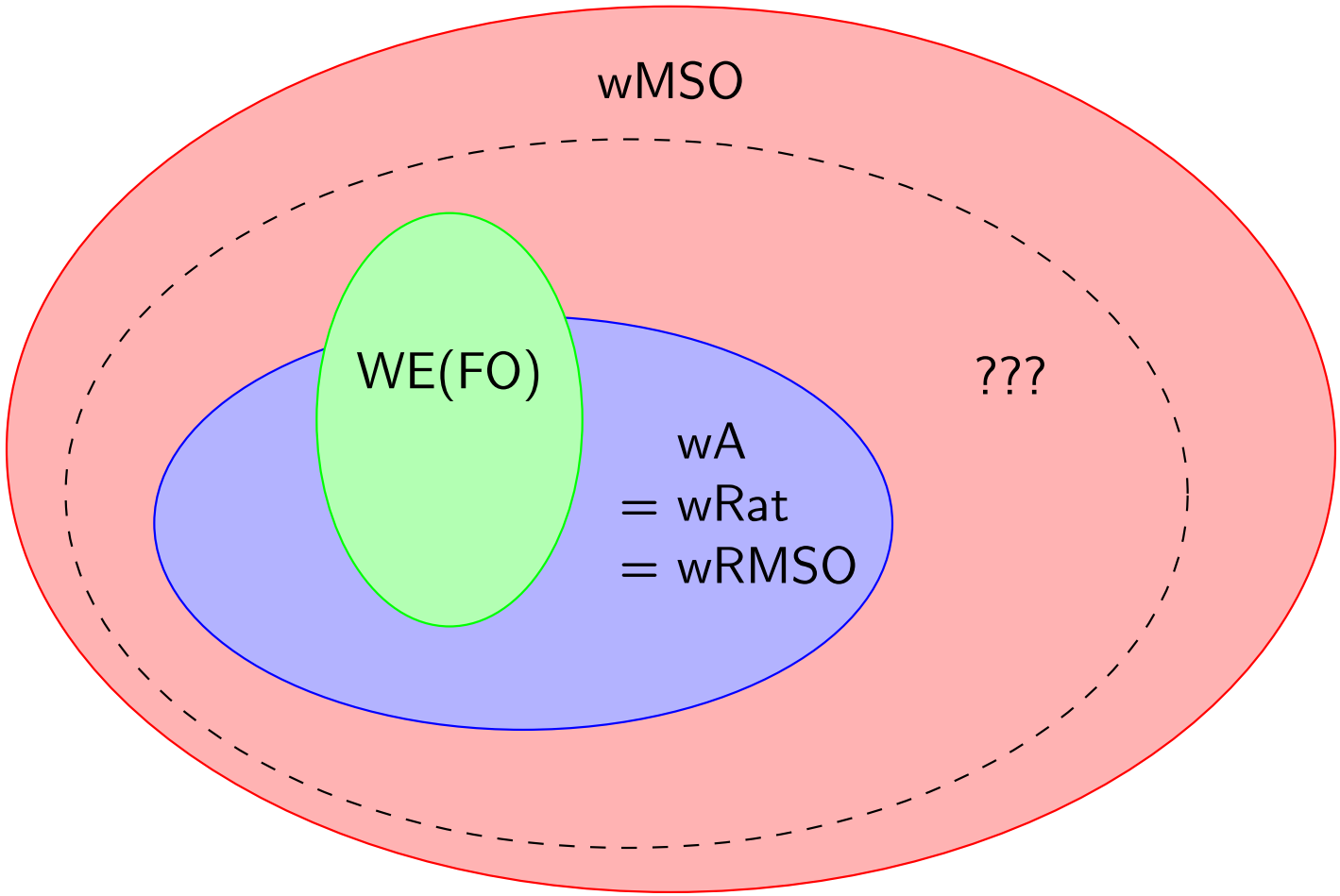
Theorem (Droste & Gastin'05)

$$\text{wAutomata} = \text{wRMSO}$$

wRMSO consists of weighted expressions with

- $\bigotimes_x$ restricted to $\bigvee \bigwedge$ of constants and boolean formulae.
- A new second order weighted operator, $\bigoplus_X$, restricted to boolean formulae.

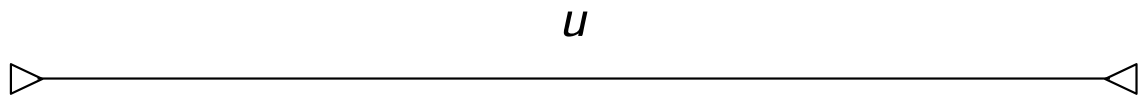
Extending instead of Restricting ?



Aim: robust class extending both WE(FO) and wAutomata.

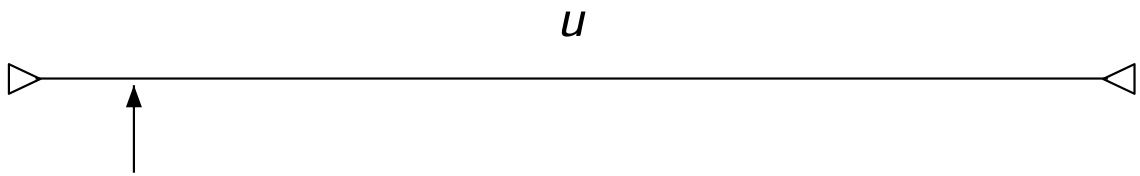
(2-way) Pebble weighted automata

- Automaton with 2-way mechanism and pebbles $\{1, \ldots, r\}$.



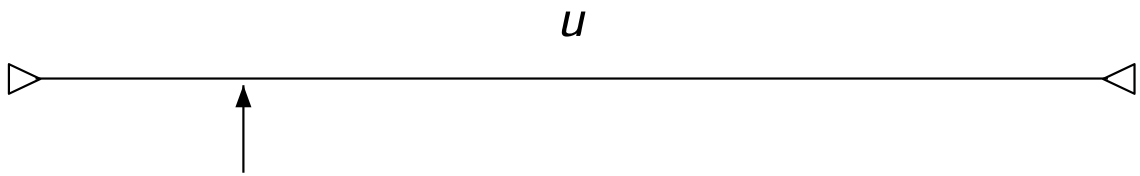
(2-way) Pebble weighted automata

- Automaton with 2-way mechanism and pebbles $\{1, \ldots, r\}$.



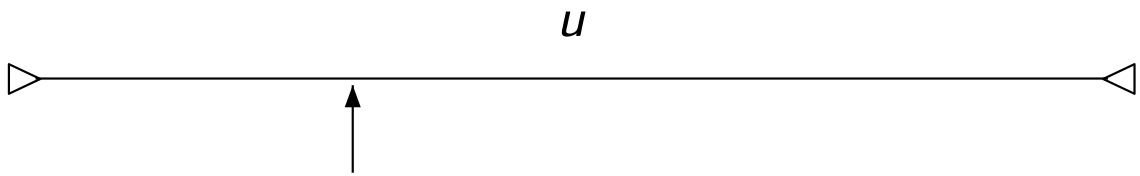
(2-way) Pebble weighted automata

- Automaton with 2-way mechanism and pebbles $\{1, \ldots, r\}$.



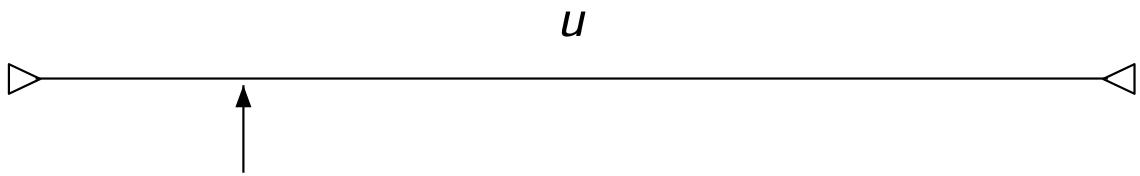
(2-way) Pebble weighted automata

- Automaton with 2-way mechanism and pebbles $\{1, \ldots, r\}$.



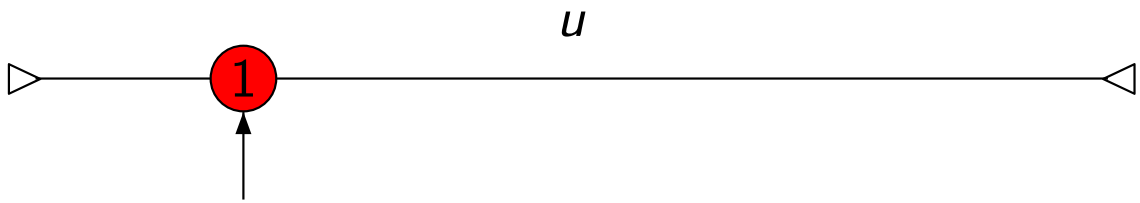
(2-way) Pebble weighted automata

- Automaton with 2-way mechanism and pebbles $\{1, \ldots, r\}$.



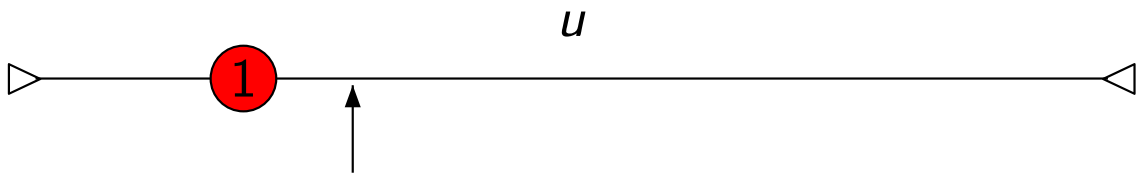
(2-way) Pebble weighted automata

- Automaton with 2-way mechanism and pebbles $\{1, \ldots, r\}$.



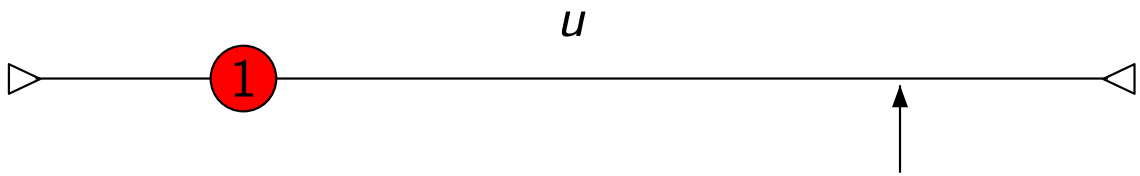
(2-way) Pebble weighted automata

- Automaton with 2-way mechanism and pebbles $\{1, \ldots, r\}$.



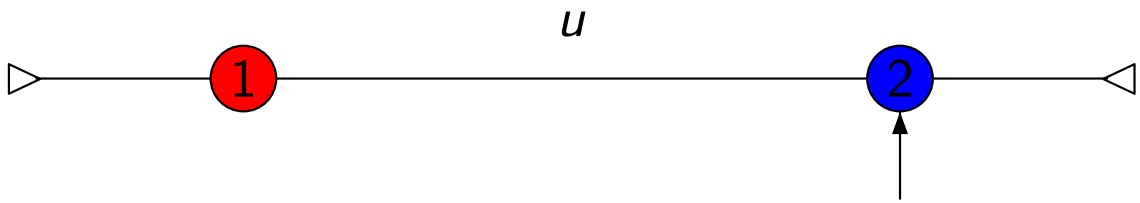
(2-way) Pebble weighted automata

- Automaton with 2-way mechanism and pebbles $\{1, \ldots, r\}$.



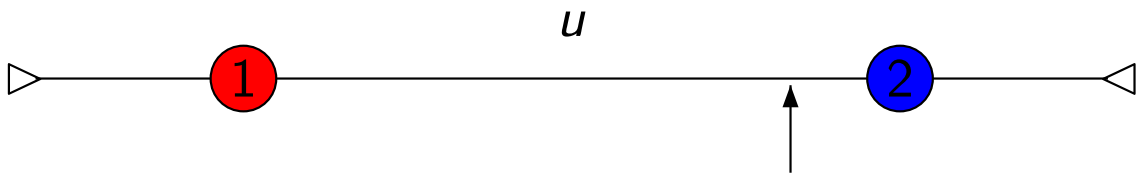
(2-way) Pebble weighted automata

- Automaton with 2-way mechanism and pebbles $\{1, \ldots, r\}$.



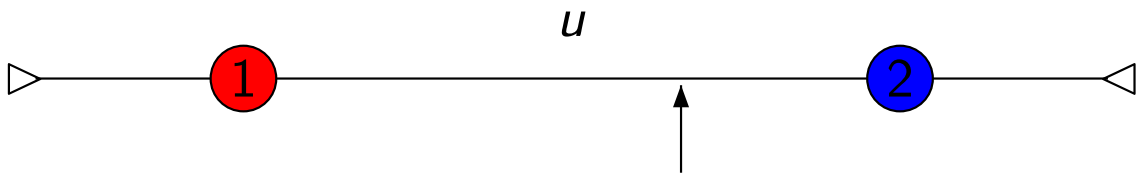
(2-way) Pebble weighted automata

- Automaton with 2-way mechanism and pebbles $\{1, \ldots, r\}$.



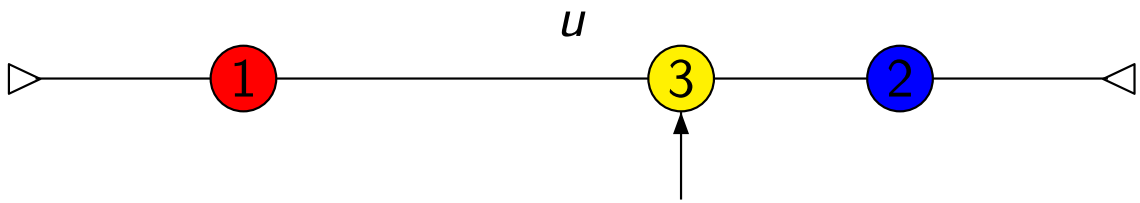
(2-way) Pebble weighted automata

- Automaton with 2-way mechanism and pebbles $\{1, \ldots, r\}$.



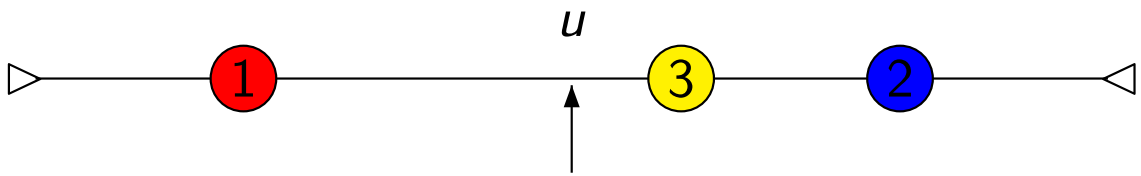
(2-way) Pebble weighted automata

- Automaton with 2-way mechanism and pebbles $\{1, \ldots, r\}$.



(2-way) Pebble weighted automata

- Automaton with 2-way mechanism and pebbles $\{1, \ldots, r\}$.

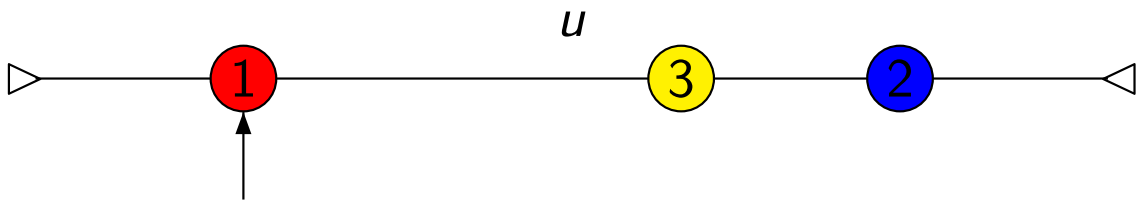


- Applicable transitions depend on current (state,letter,pebbles).

$$(p, a, k, \text{Pebbles}, \text{D}, q), \text{ where } \text{D} \in \{\leftarrow, \rightarrow, \text{lift}, \text{drop}\}.$$

(2-way) Pebble weighted automata

- Automaton with 2-way mechanism and pebbles $\{1, \ldots, r\}$.



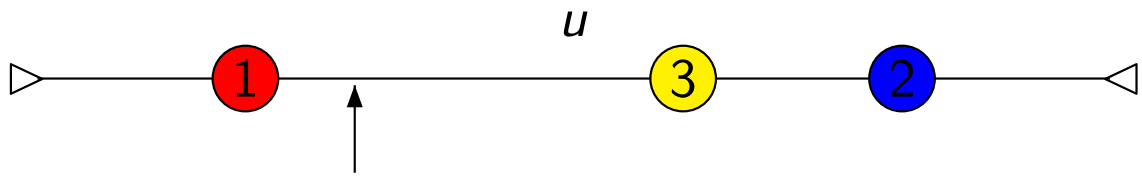
- Applicable transitions depend on current (state,letter,pebbles).

$$(p, a, k, \text{Pebbles}, \text{D}, q), \text{ where } \text{D} \in \{\leftarrow, \rightarrow, \text{lift}, \text{drop}\}.$$

- Stack policy: only the most recently dropped pebble may be lifted

(2-way) Pebble weighted automata

- Automaton with 2-way mechanism and pebbles $\{1, \ldots, r\}$.



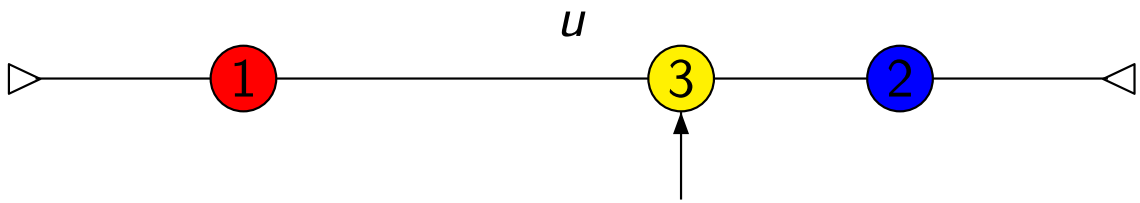
- Applicable transitions depend on current (state,letter,pebbles).

$$(p, a, k, \text{Pebbles}, \text{D}, q), \text{ where } \text{D} \in \{\leftarrow, \rightarrow, \text{lift}, \text{drop}\}.$$

- Stack policy: only the most recently dropped pebble may be lifted
- Weak policy: pebble may be lifted only when the head scans its position.

(2-way) Pebble weighted automata

- Automaton with 2-way mechanism and pebbles $\{1, \ldots, r\}$.



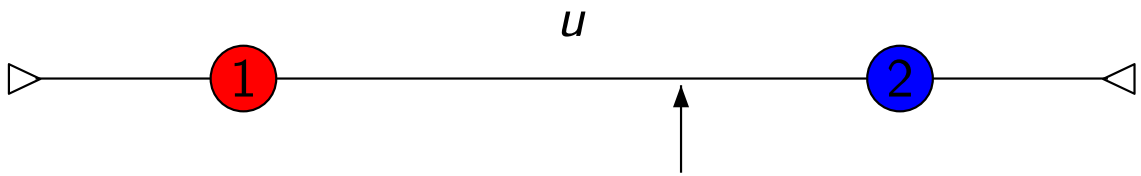
- Applicable transitions depend on current (state,letter,pebbles).

$$(p, a, k, \text{Pebbles}, \text{D}, q), \text{ where } \text{D} \in \{\leftarrow, \rightarrow, \text{lift}, \text{drop}\}.$$

- Stack policy: only the most recently dropped pebble may be lifted
- Weak policy: pebble may be lifted only when the head scans its position.

(2-way) Pebble weighted automata

- Automaton with 2-way mechanism and pebbles $\{1, \ldots, r\}$.



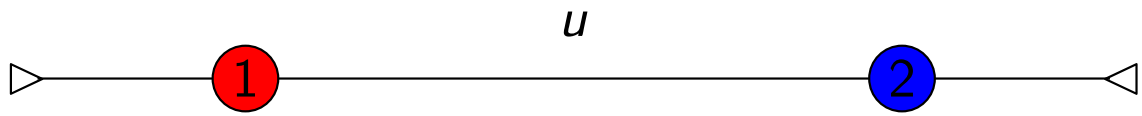
- Applicable transitions depend on current (state,letter,pebbles).

$$(p, a, k, \text{Pebbles}, \text{D}, q), \text{ where } \text{D} \in \{\leftarrow, \rightarrow, \text{lift}, \text{drop}\}.$$

- Stack policy: only the most recently dropped pebble may be lifted
- Weak policy: pebble may be lifted only when the head scans its position.

(2-way) Pebble weighted automata

- Automaton with 2-way mechanism and pebbles $\{1, \ldots, r\}$.



- Applicable transitions depend on current (state,letter,pebbles).

$$(p, a, k, \text{Pebbles}, \text{D}, q), \text{ where } \text{D} \in \{\leftarrow, \rightarrow, \text{lift}, \text{drop}\}.$$

- Stack policy: only the most recently dropped pebble may be lifted
- Weak policy: pebble may be lifted only when the head scans its position.
- Note. For Boolean word automata, this does not add expressive power.

Pebble weighted automata: semantics

Recall from the classical setting:

- Value of a word: sum of all weights of runs on this word.

$$[\![\mathcal{A}]\!](w) = \sum_{\rho \text{ run of } \mathcal{A} \text{ on } w} \text{weight}(\rho)$$

- No longer well defined for 2-way pebble automata
  (can loop $\Rightarrow$ can have arbitrarily large runs on a given word).

Pebble weighted automata: semantics

Recall from the classical setting:

- Value of a word: sum of all weights of runs on this word.

$$[\![\mathcal{A}]\!](w) = \sum_{\rho \text{ run of } \mathcal{A} \text{ on } w} \text{weight}(\rho)$$

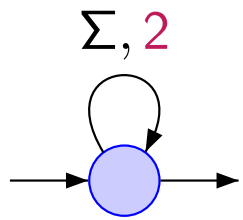
- No longer well defined for 2-way pebble automata (can loop $\Rightarrow$ can have arbitrarily large runs on a given word).
- Value of a word: sum of all weights of simple runs on this word.

$$[\![\mathcal{A}]\!](w) = \sum_{\rho \text{ simple run of } \mathcal{A} \text{ on } w} \text{weight}(\rho)$$

(Other solution would be to restrict to suitable semirings)

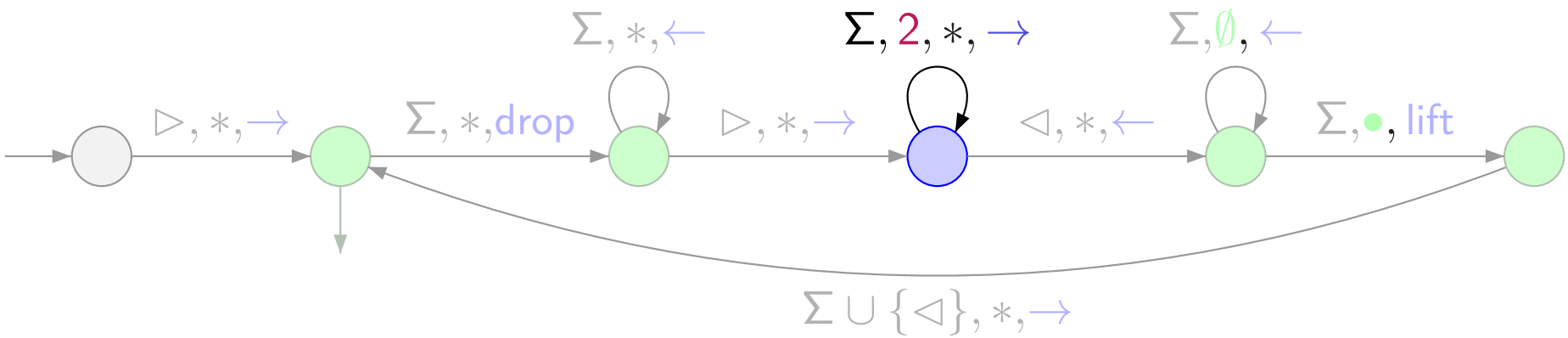
Why pebbles?

- Pebbles can be used to encode a first-order variable (its position).



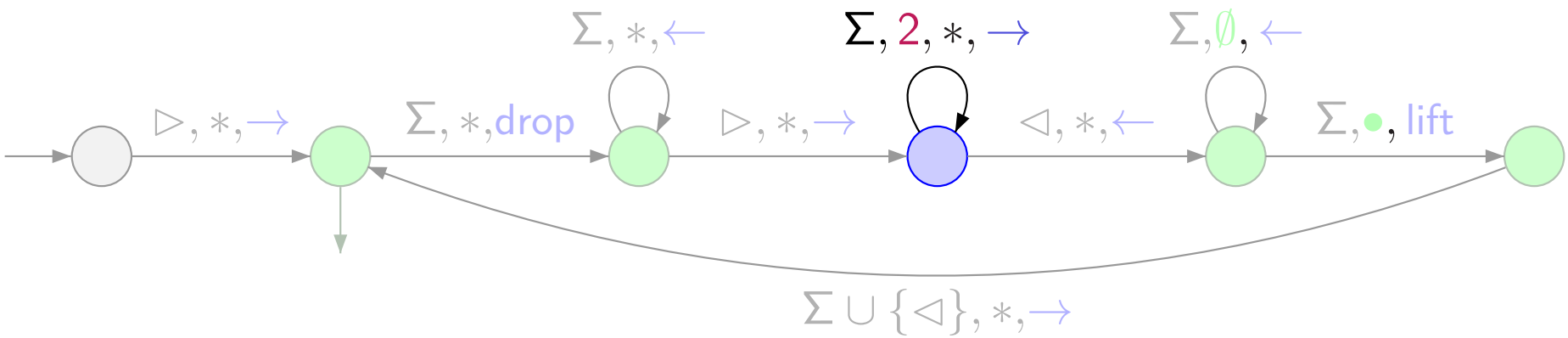
Why pebbles?

- Pebbles can be used to encode a first-order variable (its position).



Why pebbles?

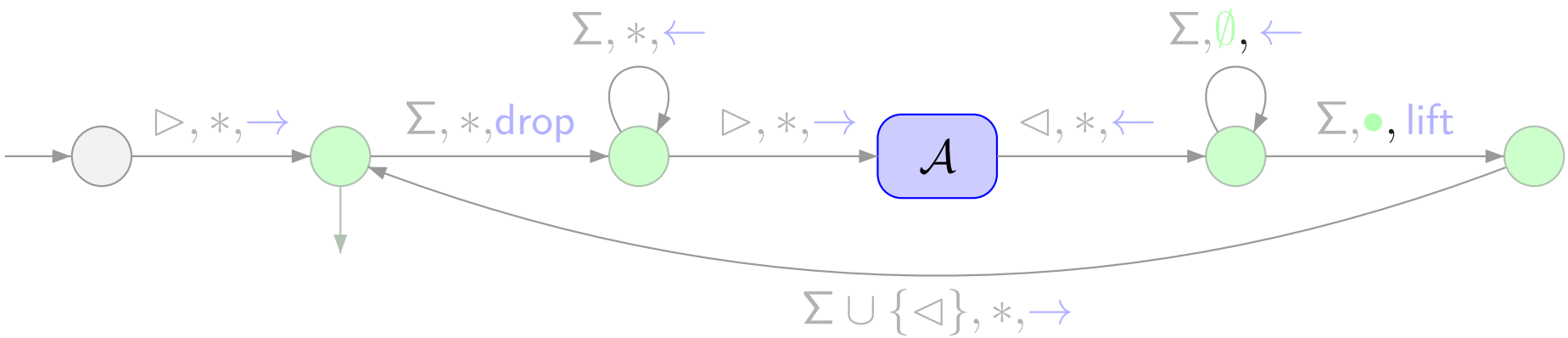
- Pebbles can be used to encode a first-order variable (its position).



- Computes $2^{|u|^2}$: pebbles add expressive power.

Why pebbles?

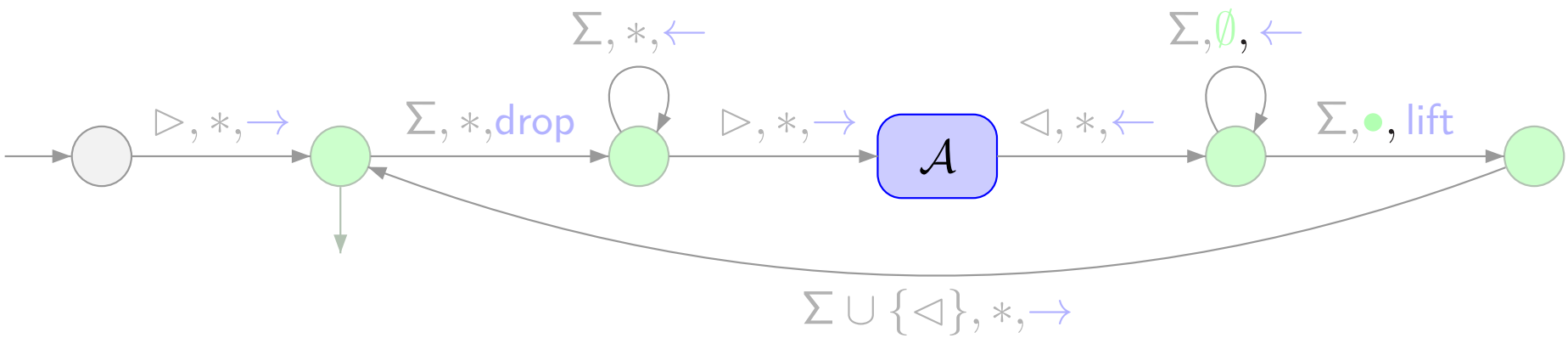
- Pebbles can be used to encode a first-order variable (its position).



- Computes $2^{|u|^2}$: pebbles add expressive power.
- Very same idea: pebble weighted automata are closed under $\bigoplus_x$.

Why pebbles?

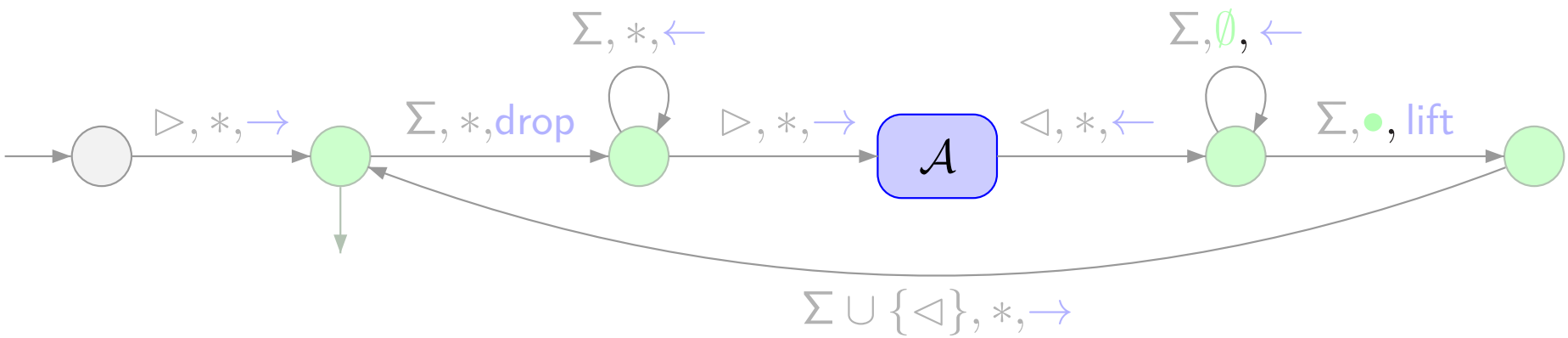
- Pebbles can be used to encode a first-order variable (its position).



- Computes $2^{|u|^2}$: pebbles add expressive power.
- Very same idea: pebble weighted automata are closed under $\bigoplus_x$.
  by dropping nondeterministically the pebble instead.

Why pebbles?

- Pebbles can be used to encode a first-order variable (its position).



- Computes $2^{|u|^2}$: pebbles add expressive power.
- Very same idea: pebble weighted automata are closed under $\bigoplus_x$.
  by dropping nondeterministically the pebble instead.
- Summary: pebble wA easily bring closure under $\bigotimes_x$ and $\bigoplus_x$.

1-way pebble weighted automata

The construction for closure under $\bigoplus_x$ and $\bigotimes_x$ uses specific automata.

- After a drop, go to the end of the word and reset.
- No other use of $\leftarrow$ move.

1-way pebble automata with ℓ-resets

A 1-way pebble automaton is a 2-way pebble automaton st.

- no $\leftarrow$ move. Replaced by new move: reset.
- no lift can be immediately followed by a drop,
- each time a pebble is dropped, it gets a credit for ℓ resets (recursively).

Similar to 1-way automata used by Neven, Schwentick, Vianu 04 for data words.

Closure properties of 1 way pwA

- 1-way pebble automata are nonlooping (pebble are "progressing").
- Semantics well-defined as $[\![\mathcal{A}]\!](w) = \sum_{\rho \text{ run of } \mathcal{A} \text{ on } w} \text{weight}(\rho)$.
- Still closed under $\bigoplus_x$ and $\bigotimes_x$.
- Closed under $\oplus$ and $\otimes$.

Closure properties of 1 way pwA

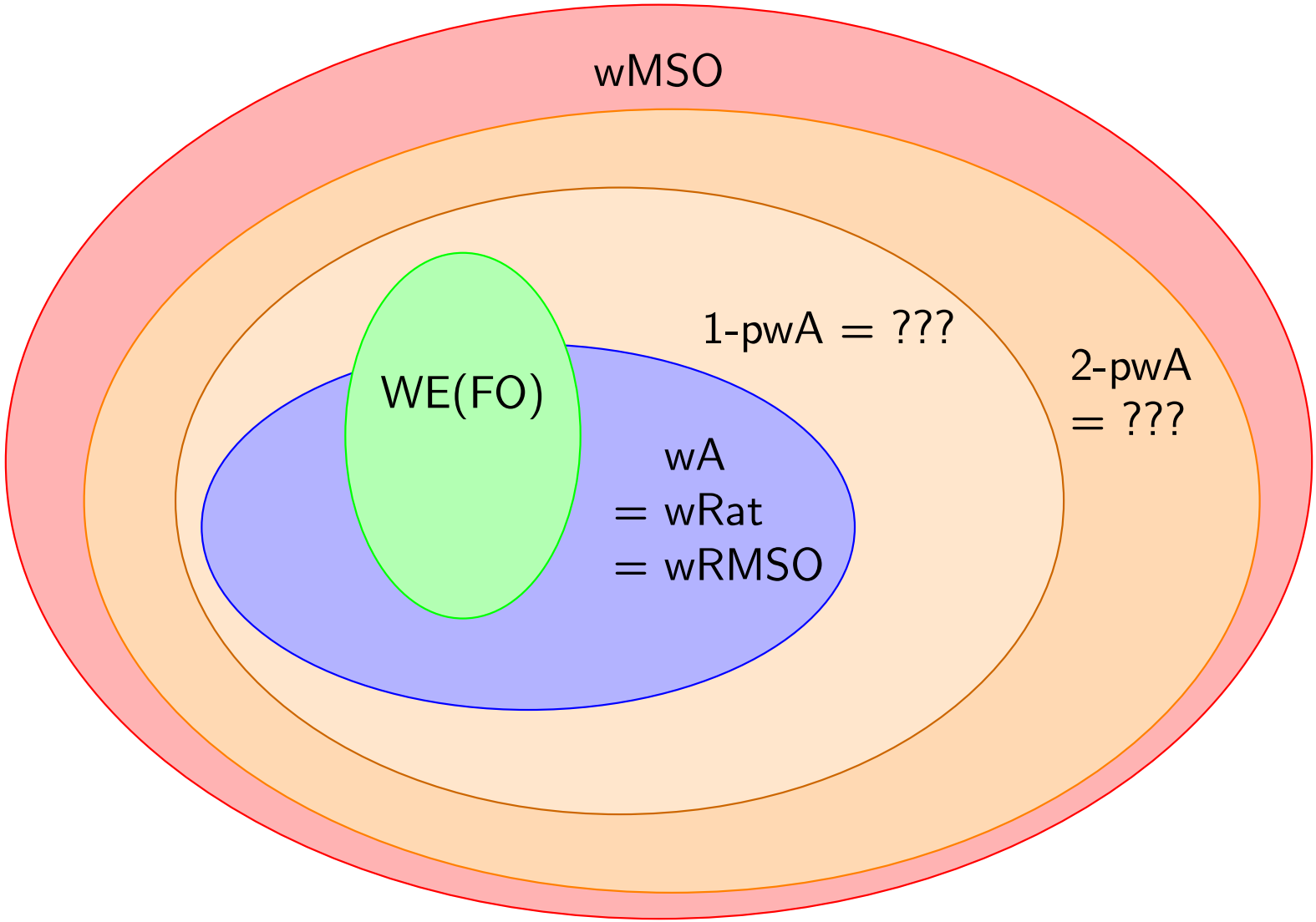
- 1-way pebble automata are nonlooping (pebble are "progressing").
- Semantics well-defined as $[\![\mathcal{A}]\!](w) = \sum_{\rho \text{ run of } \mathcal{A} \text{ on } w} \text{weight}(\rho)$.
- Still closed under $\bigoplus_x$ and $\bigotimes_x$.
- Closed under $\oplus$ and $\otimes$.

Corollary

1-way pebble weighted automata capture WE(MSO).

For the converse, need to enrich the weighted expressions

Pebble weighted Automata vs WE(MSO)



Characterization of 1-way and 2-way pebble wA in terms of expressions?

Weighted transitive closure: TC and BTC

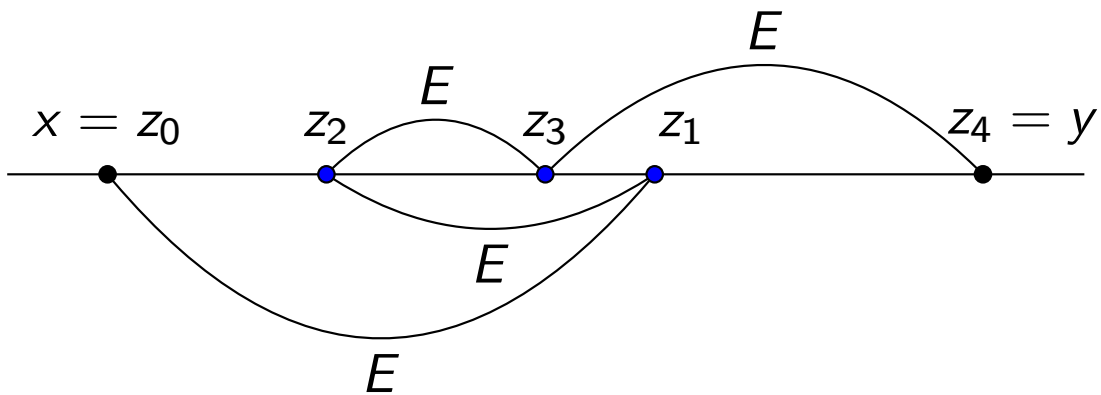
- Boolean setting: MSO $\equiv$ FO($<$) + Transitive closure.

Weighted transitive closure: TC and BTC

- Boolean setting: MSO $\equiv$ FO($<$) + Transitive closure.
- Weighted: for $E(x, y)$ with (at least) two first order free variables, let

$$E^n(x, y) = \bigoplus_{x=z_0, z_1, \ldots, z_n=y} \left(\bigotimes_{1 \leq \ell \leq n} E(z_{\ell-1}, z_\ell) \right)$$

where the sum ranges over seq. of pairwise distinct positions $z_0, \ldots, z_n$.



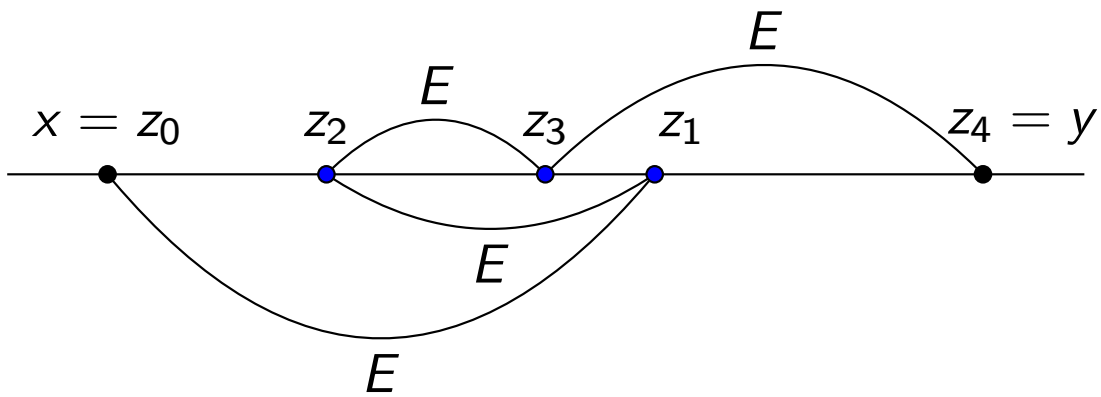
- The transitive closure operator is defined by $\mathrm{TC}_{xy} E = \bigoplus_{n \geq 1} E^n$.

Weighted transitive closure: TC and BTC

- Boolean setting: MSO $\equiv$ FO($<$) + Transitive closure.
- Weighted: for $E(x, y)$ with (at least) two first order free variables, let

$$E^n(x, y) = \bigoplus_{x=z_0, z_1, \ldots, z_n=y} \left(\bigotimes_{1 \leq \ell \leq n} E(z_{\ell-1}, z_\ell) \right)$$

where the sum ranges over seq. of pairwise distinct positions $z_0, \ldots, z_n$.

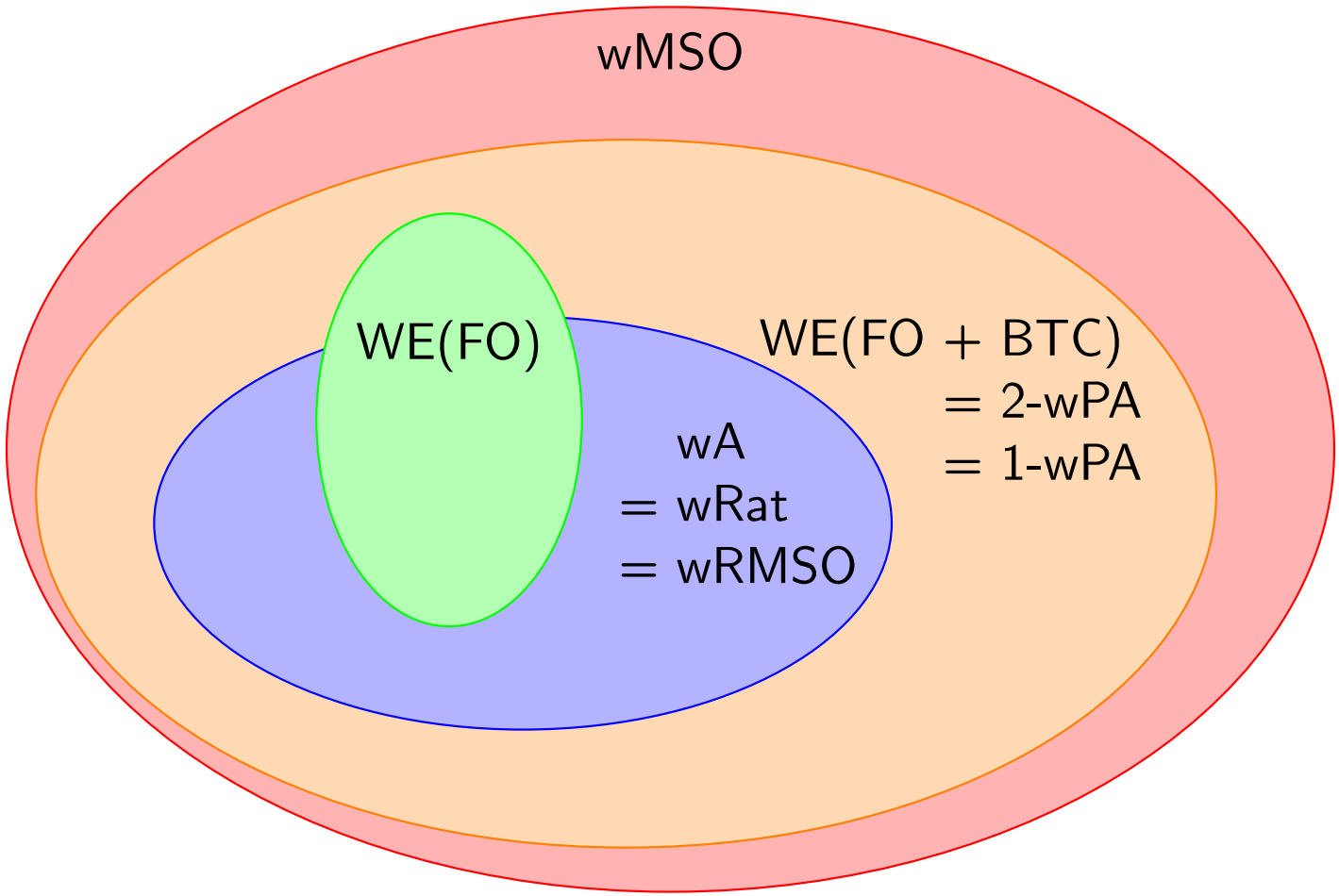


- The transitive closure operator is defined by $\mathrm{TC}_{xy} E = \bigoplus_{n \geq 1} E^n$.
- Bounded transitive closure BTC: $N\text{-}\mathrm{TC}_{xy} E = \mathrm{TC}_{xy}(E \wedge |x - y| \leq N)$

Expressiveness

Theorem: WE(FO + BTC) = 2-way pebble wA = 1-way pebble wA.

(On commutative semirings)



Related and Further Work

- Other nice framework in min-+: cost functions, S/B-automata
  [Th. Colcombet, M. Bojańczyk]

$$\text{Logics} \longleftrightarrow \text{Automata}$$

- Lack of results for proving non-expressiveness.
- Unbounded steps in transitive closure?
- Weak pebbles vs. strong pebbles?
- Extensions to other structures: Trees ($\rightsquigarrow$ Benjamin's talk)