Logics for Weighted Automata

Benjamin Monmege

Aix-Marseille Université, Laboratoire d'Informatique et Systèmes

Gala 2019 *IIT Bombay*

Software Verification

Software Verification

Critical Software

- communication systems
- e-commerce
- health databases
- energy production

Software Verification

Software Verification

Property to be verified

Critical Software

- communication systems
- e-commerce
- health databases
- energy production

TO BE VERIFIED

Software Verification

Property to be verified

Is the property verified or not by the software?

Critical Software
- communication systems
- e-commerce
- health databases
- energy production

TO BE VERIFIED

Software Verification

Property to be verified

- May an *error* state be reached?
- Is there a book written by X, rented by Y?
- Does this leader election protocol permit to elect the leader?

erified

ware?

are

systems

- e-commerce
- health databases
- energy production

TO BE VERIFIED

Software Verification

Property to be verified

- May an *error* state be reached?
- Is there a book written by X, rented by Y?
- Does this leader election protocol permit to elect the leader?

From **Boolean** to **Quantitative** Verification

- What is the probability for an *error* state to be reached?
- How many books, written by X, have been rented by Y?
- What is the maximal delay ensuring that this leader election protocol permits the election?

erified
ware?

are
systems
- e-commerce
- health databases
- energy production

TO BE VERIFIED

Formal Verification

Property to be verified

Is the property verified or not by the software?

Critical Software

- communication systems
- e-commerce
- health databases
- energy production

TO BE VERIFIED

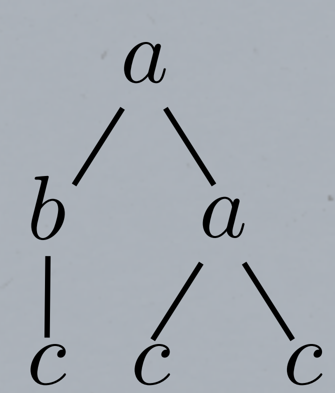
Formal Verification

Property to be verified

Is the property verified or not by the model?

Formal Model

$ababcaabb$



Critical Software

- communication systems
- e-commerce
- health databases
- energy production

TO BE VERIFIED

Formal Verification

Property to be verified
Formal Specification

$$(a+b)^\star c(ac)^+$$

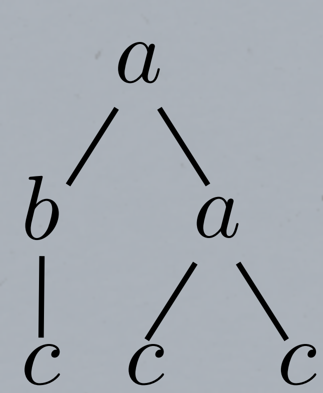
$$\forall x \forall y \,(x < y \Rightarrow \exists z \,(x < z < y))$$

$$\mathsf{F}\,\mathsf{G}\,(p\,\mathsf{U}\,q)$$

Is the property verified or not by the model?

Formal Model

$ababcaabb$



Critical Software

- communication systems
- e-commerce
- health databases
- energy production

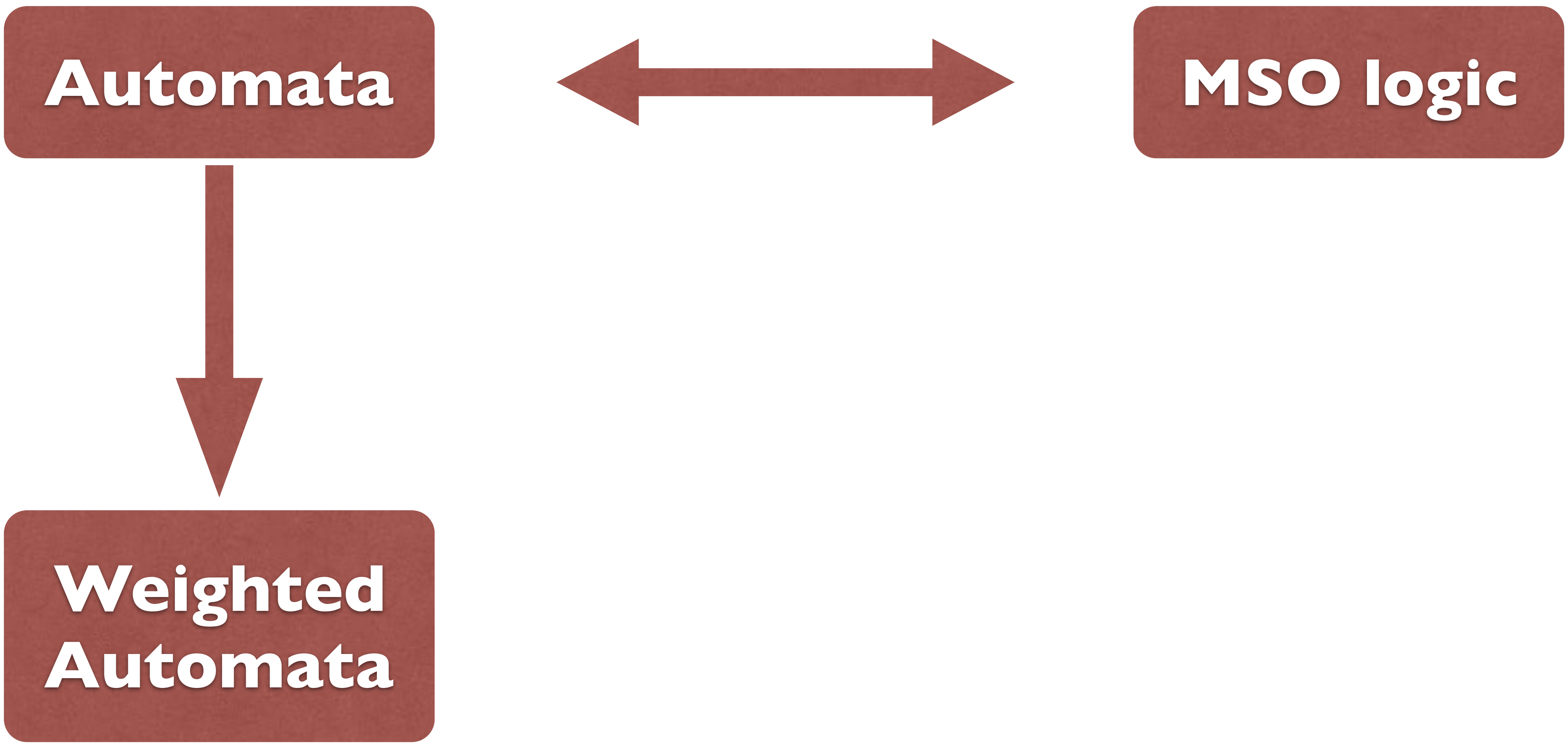
TO BE VERIFIED

Qualitative/Quantitative

- Qualitative, Boolean: [Büchi 60, Elgot 61, Trakhtenbrot 61]

Qualitative/Quantitative

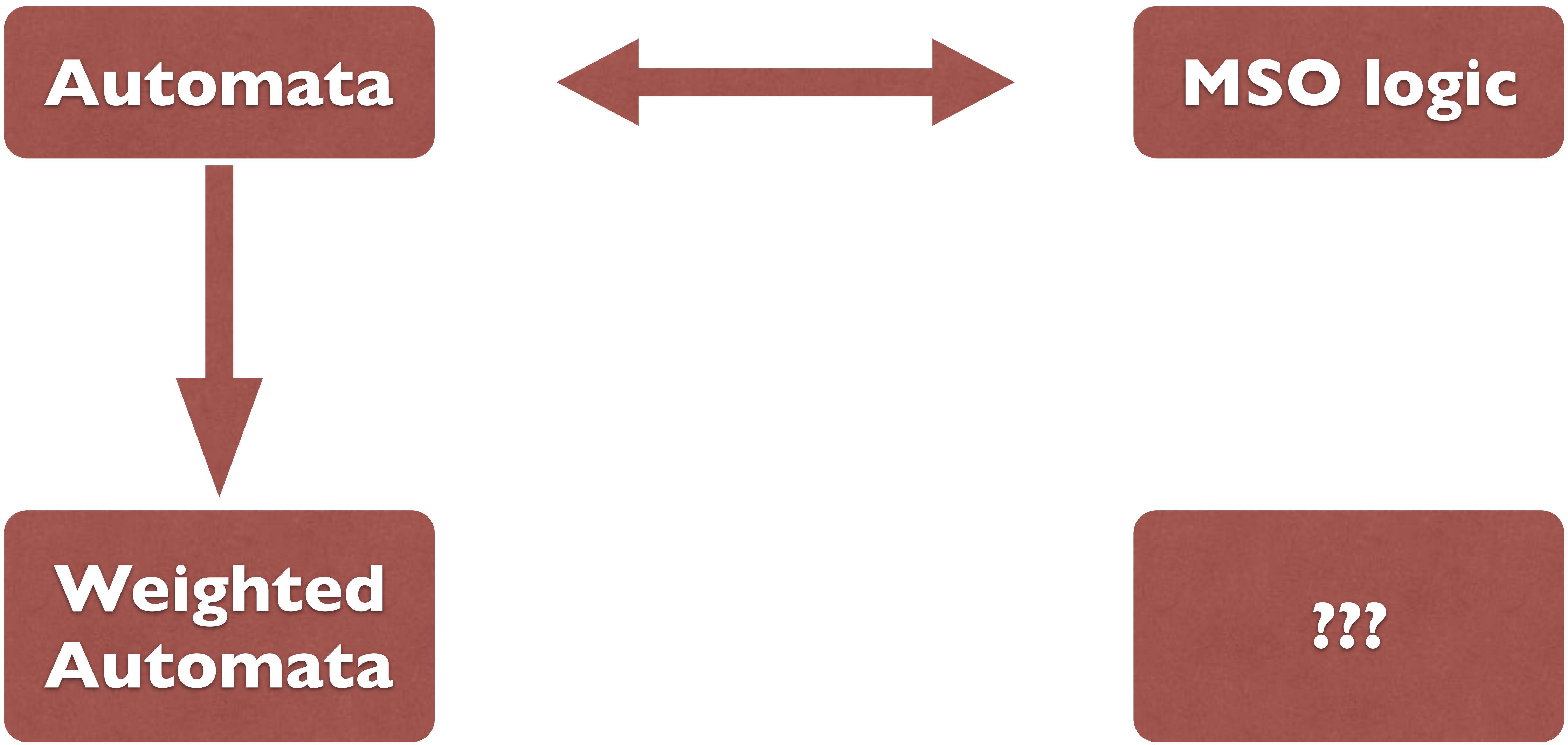
- Qualitative, Boolean: [Büchi 60, Elgot 61, Trakhtenbrot 61]



- Quantitative, weights

Qualitative/Quantitative

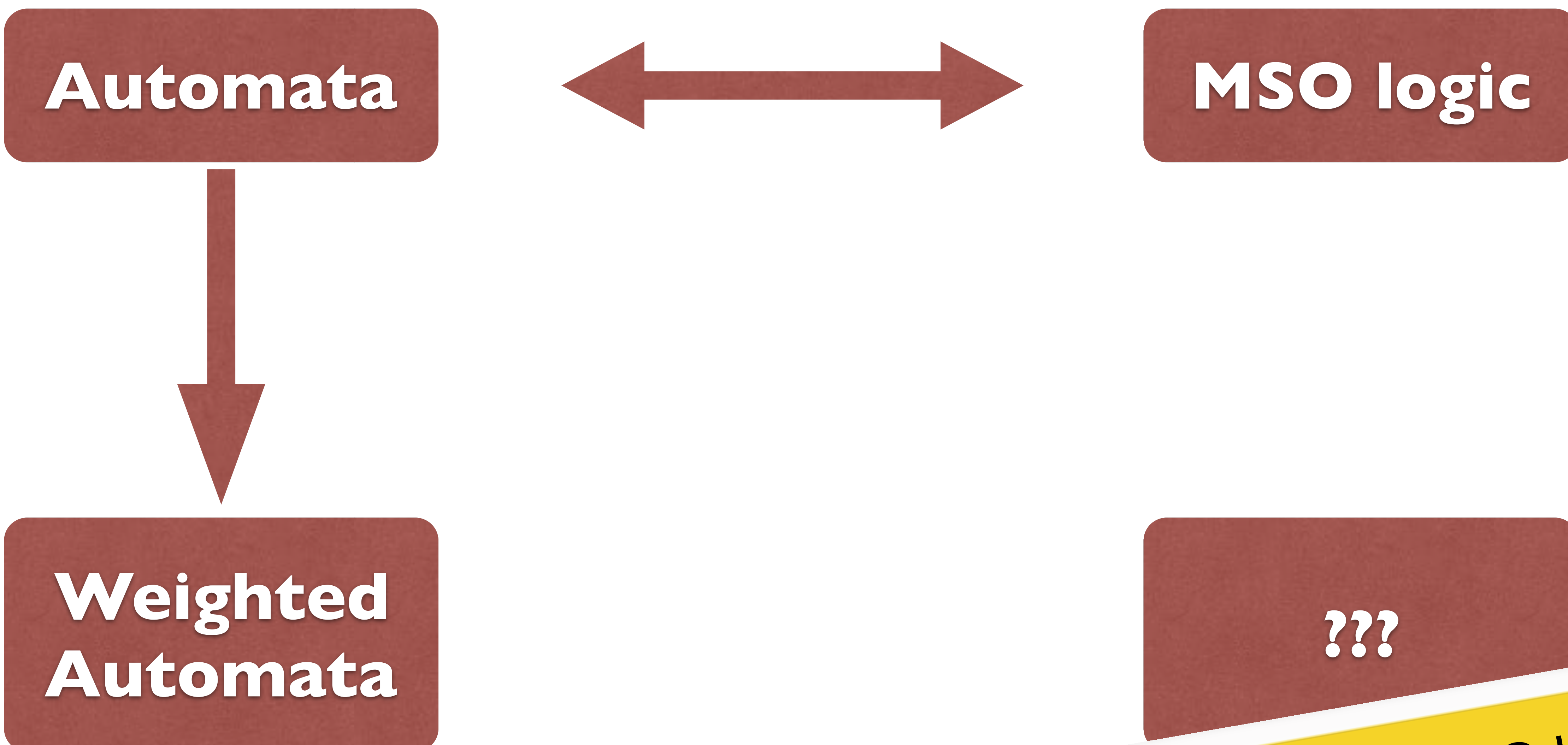
- Qualitative, Boolean: [Büchi 60, Elgot 61, Trakhtenbrot 61]



- Quantitative, weights

Qualitative/Quantitative

- Qualitative, Boolean: [Büchi 60, Elgot 61, Trakhtenbrot 61]

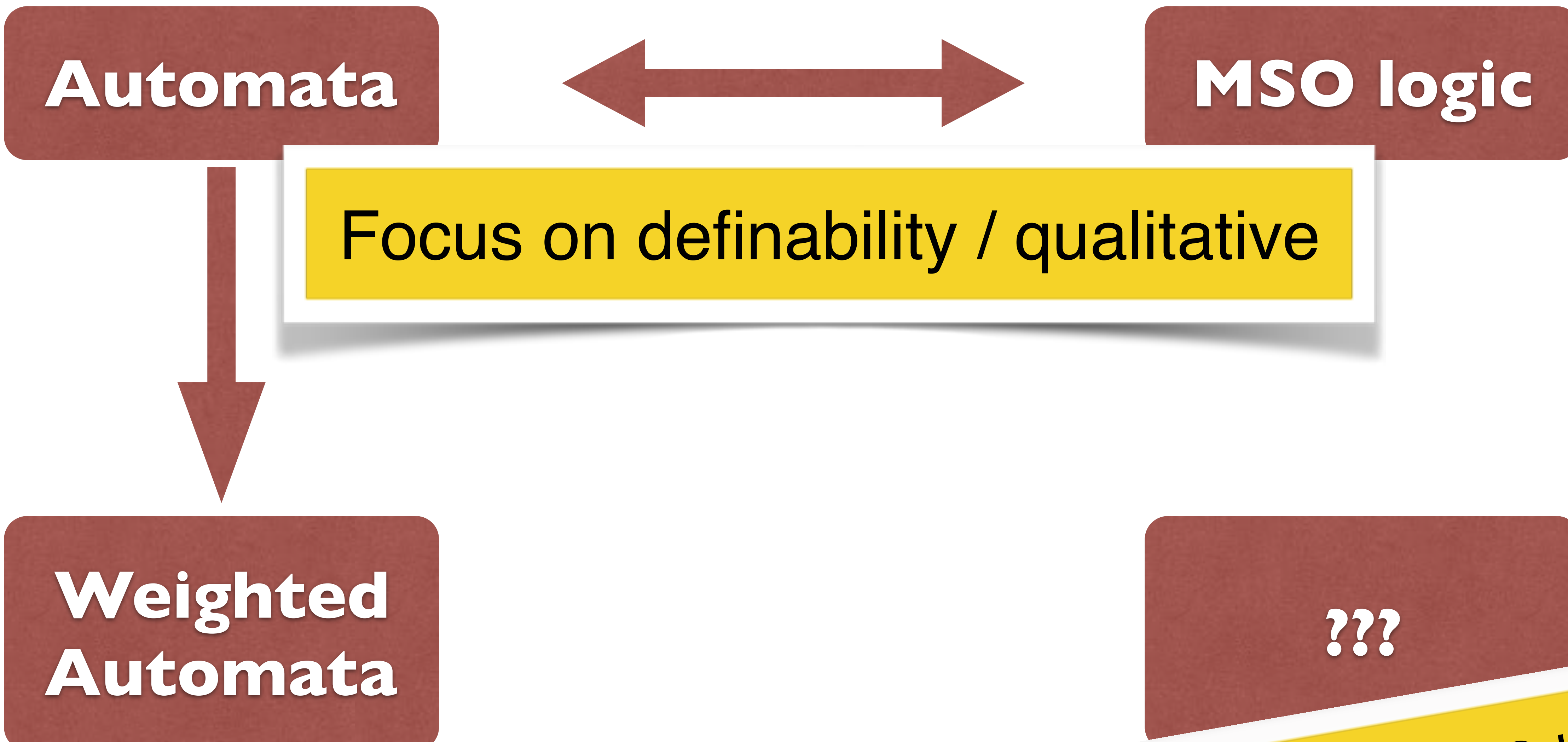


- Quantitative, weights

Find suitable weighted MSO logic

Qualitative/Quantitative

- Qualitative, Boolean: [Büchi 60, Elgot 61, Trakhtenbrot 61]

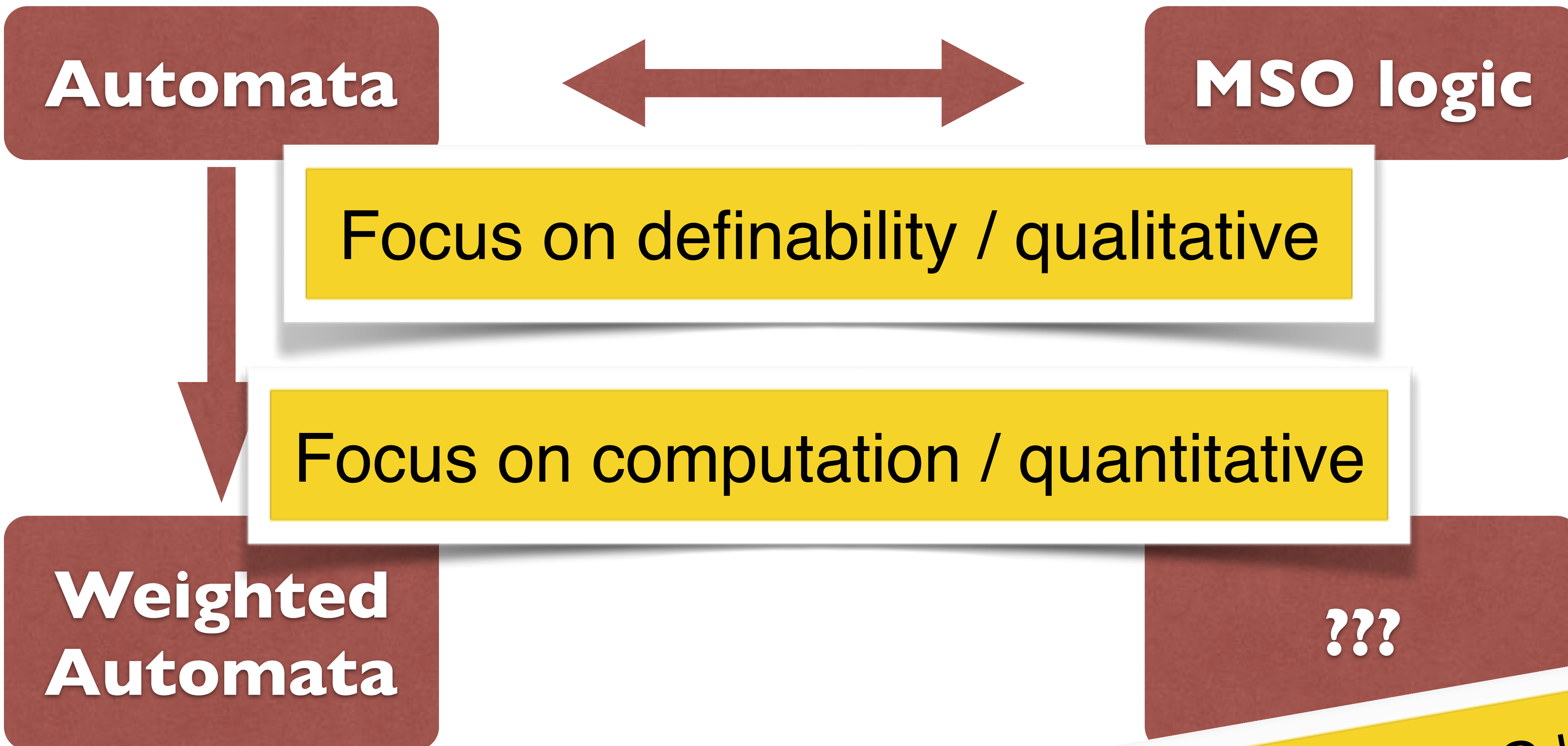


- Quantitative, weights

Find suitable weighted MSO logic

Qualitative/Quantitative

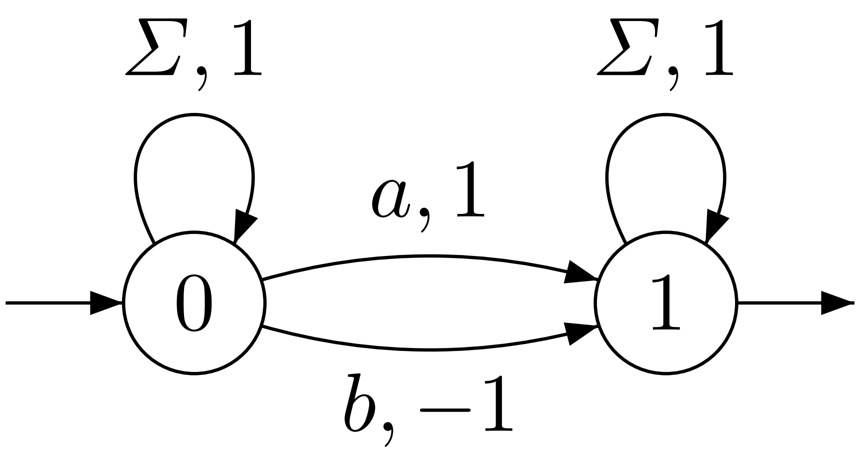
- Qualitative, Boolean: [Büchi 60, Elgot 61, Trakhtenbrot 61]

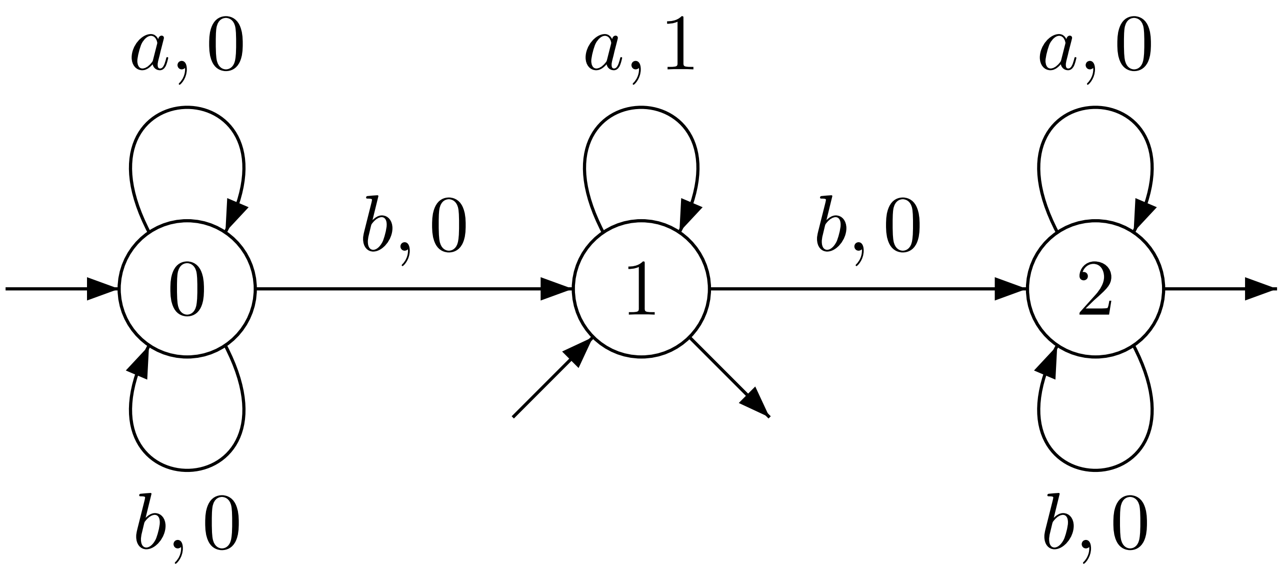


- Quantitative, weights

Find suitable weighted MSO logic

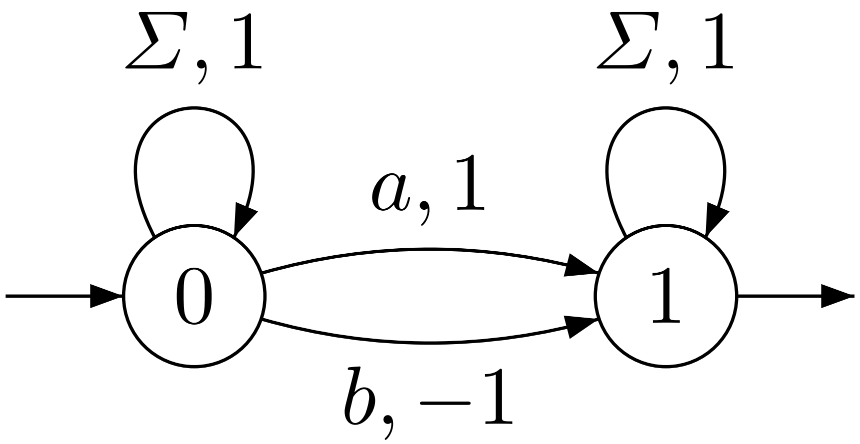
Weighted Automata

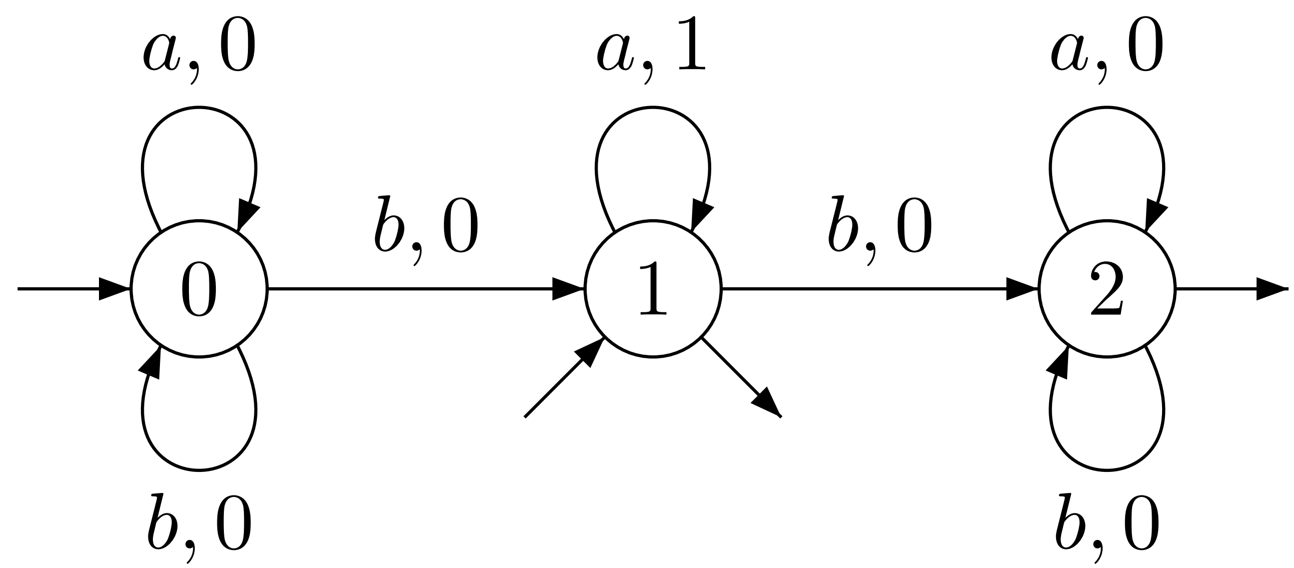




[Schützenberger 61]

Weighted Automata

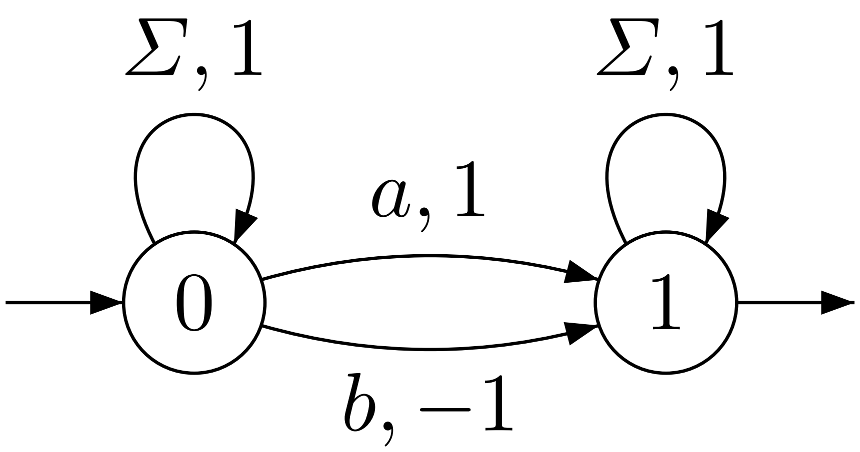


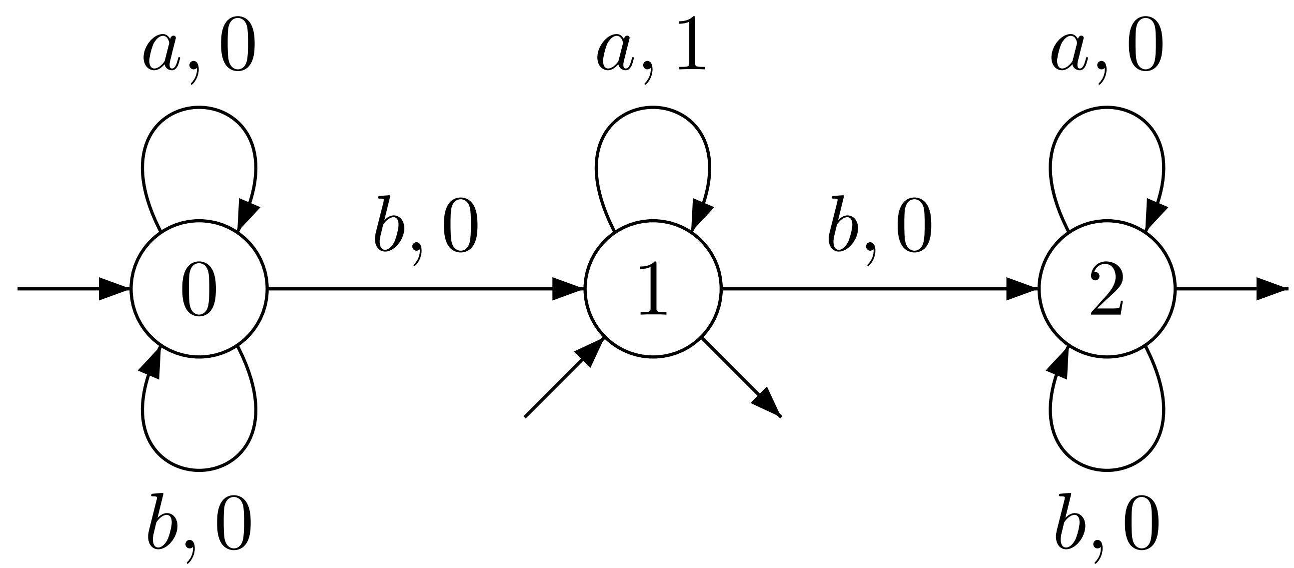


$$(\mathbf{Z}, +, \times, 0, 1)$$

[Schützenberger 61]

Weighted Automata





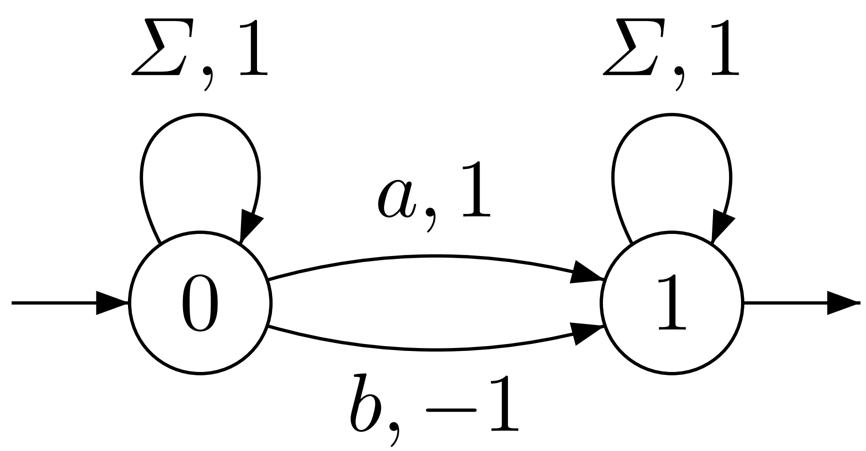
$$(\mathbf{Z}, +, \times, 0, 1)$$

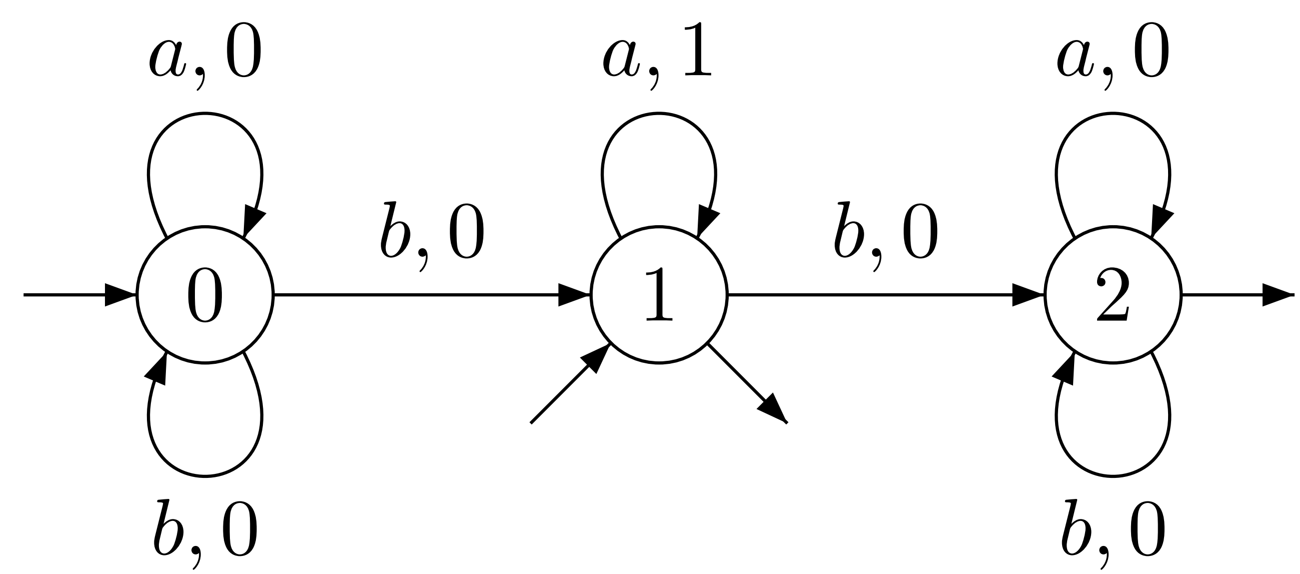
$$0 \xrightarrow{a,1} 1 \xrightarrow{b,1} 1 \xrightarrow{a,1} 1$$

1

[Schützenberger 61]

Weighted Automata





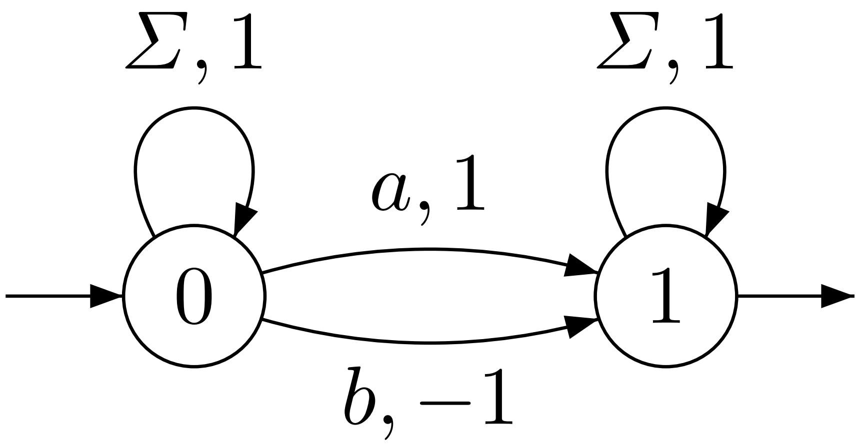
$$(\mathbf{Z}, +, \times, 0, 1)$$

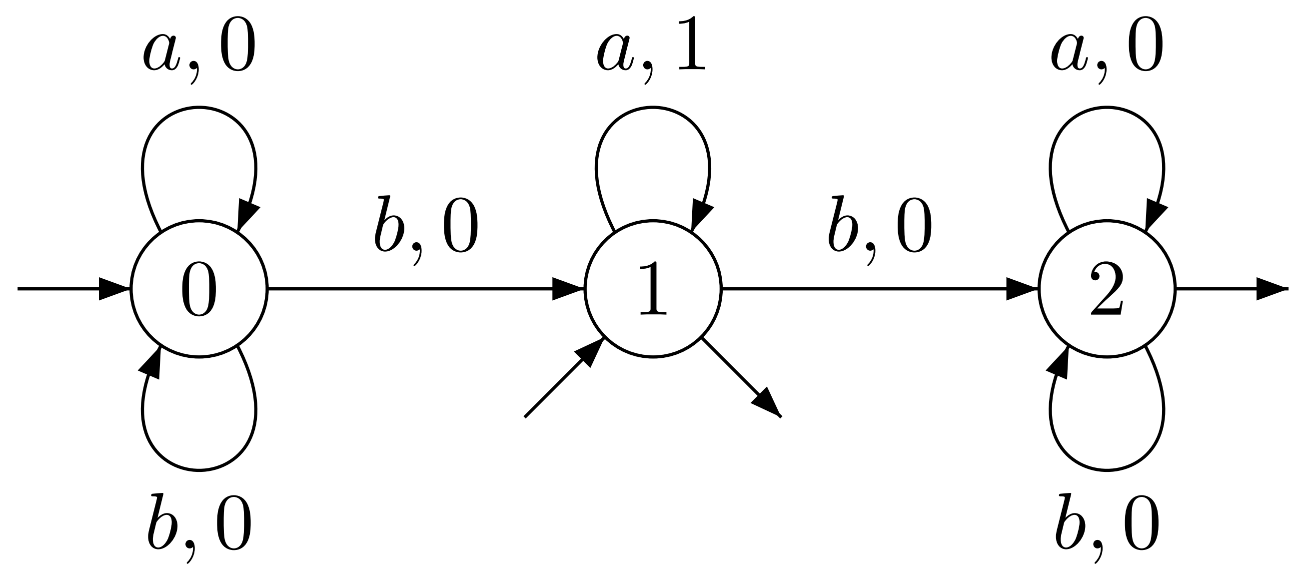
$0 \xrightarrow{a,1} 1 \xrightarrow{b,1} 1 \xrightarrow{a,1} 1$ — 1

$0 \xrightarrow{a,1} 0 \xrightarrow{b,-1} 1 \xrightarrow{a,1} 1$ — -1

[Schützenberger 61]

Weighted Automata





$$(\mathbf{Z}, +, \times, 0, 1)$$

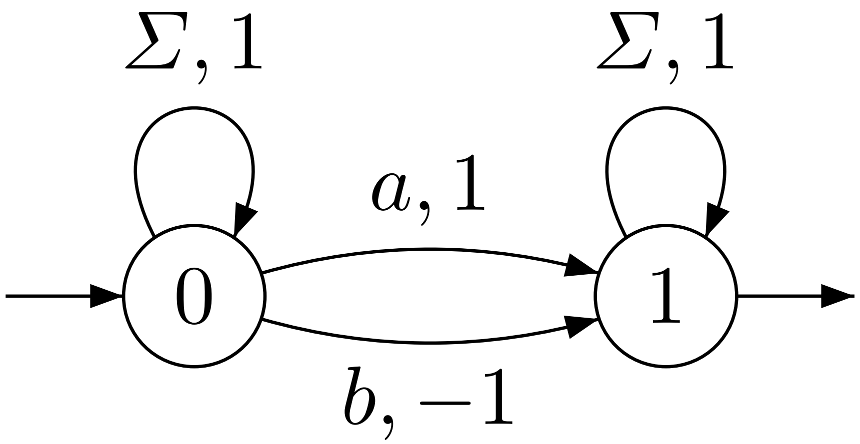
$0 \xrightarrow{a,1} 1 \xrightarrow{b,1} 1 \xrightarrow{a,1} 1$ — 1

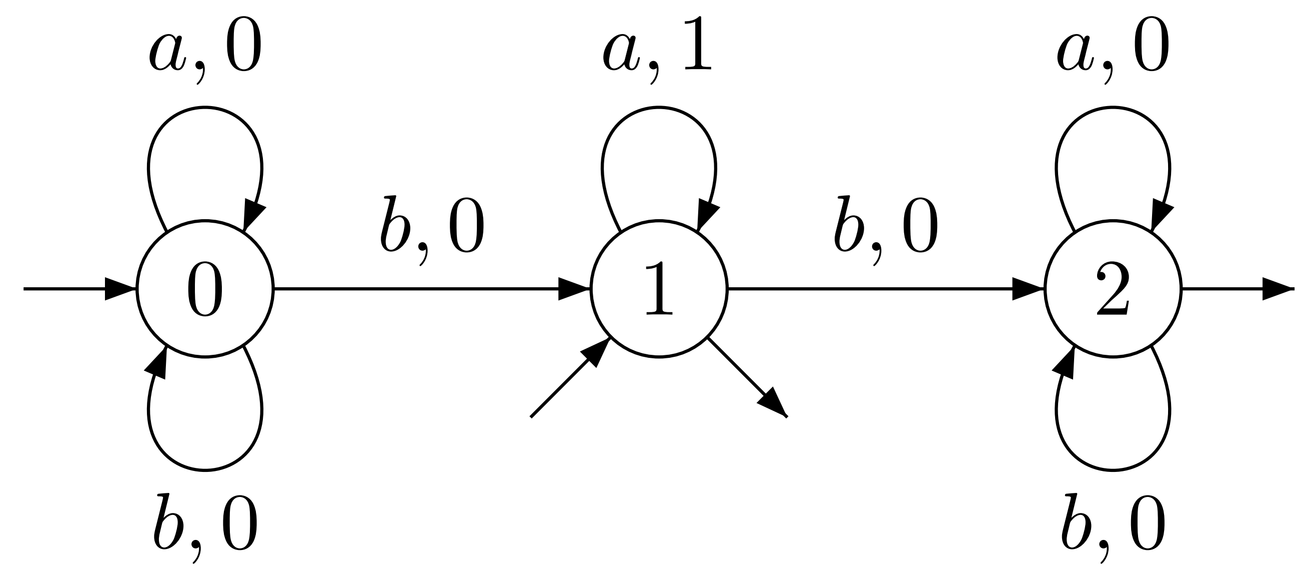
$0 \xrightarrow{a,1} 0 \xrightarrow{b,-1} 1 \xrightarrow{a,1} 1$ — -1

$0 \xrightarrow{a,1} 0 \xrightarrow{b,1} 0 \xrightarrow{a,1} 1$ — 1

[Schützenberger 61]

Weighted Automata





$$(\mathbf{Z}, +, \times, 0, 1)$$

$$0 \xrightarrow{a,1} 1 \xrightarrow{b,1} 1 \xrightarrow{a,1} 1$$ 1

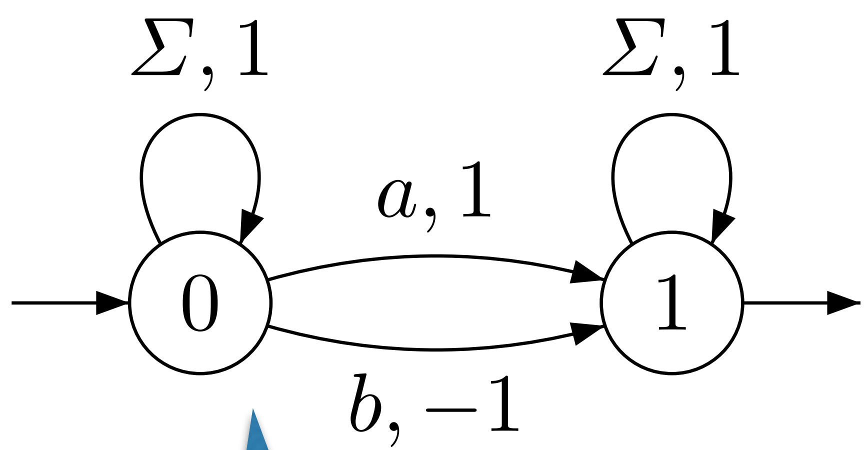
$$0 \xrightarrow{a,1} 0 \xrightarrow{b,-1} 1 \xrightarrow{a,1} 1$$ -1

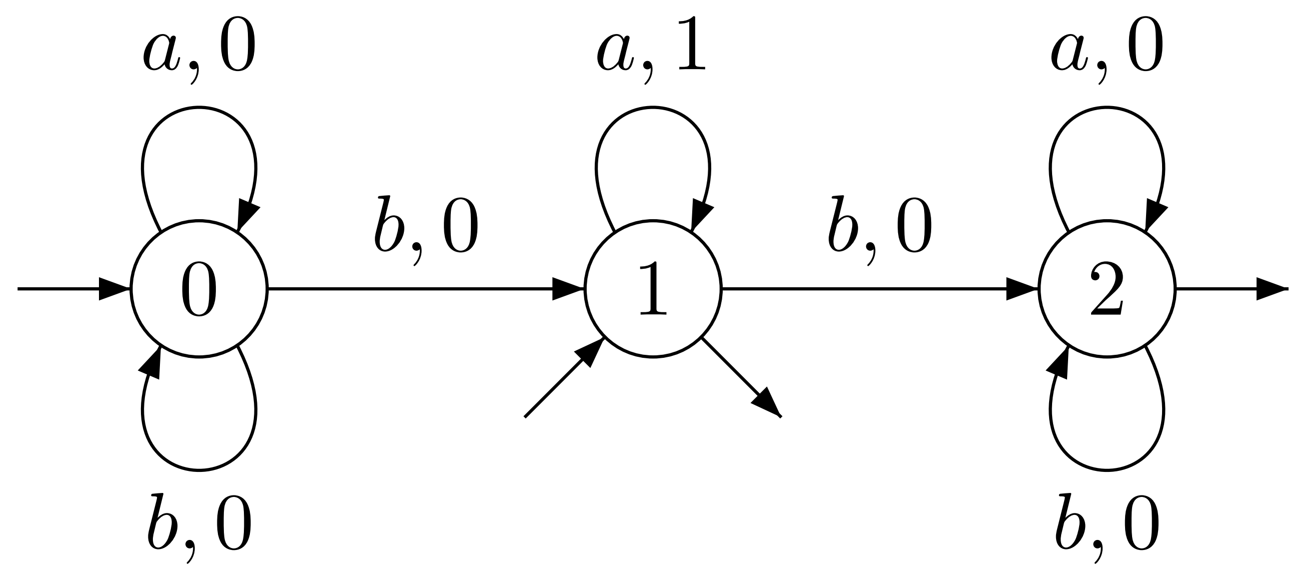
$$0 \xrightarrow{a,1} 0 \xrightarrow{b,1} 0 \xrightarrow{a,1} 1$$ 1

Semantics of *aba*: 1+(-1)+1 = 1

[Schützenberger 61]

Weighted Automata





$\#_a(w) - \#_b(w)$

$$(\mathbf{Z}, +, \times, 0, 1)$$

$$0 \xrightarrow{a,1} 1 \xrightarrow{b,1} 1 \xrightarrow{a,1} 1$$

1

$$0 \xrightarrow{a,1} 0 \xrightarrow{b,-1} 1 \xrightarrow{a,1} 1$$

-1

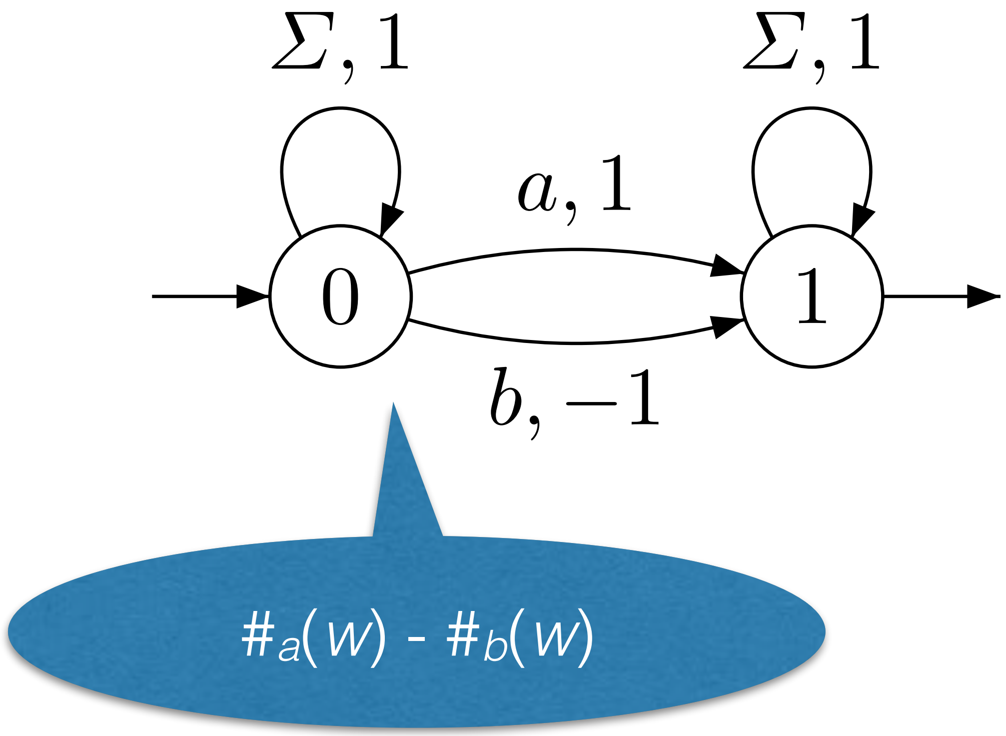
$$0 \xrightarrow{a,1} 0 \xrightarrow{b,1} 0 \xrightarrow{a,1} 1$$

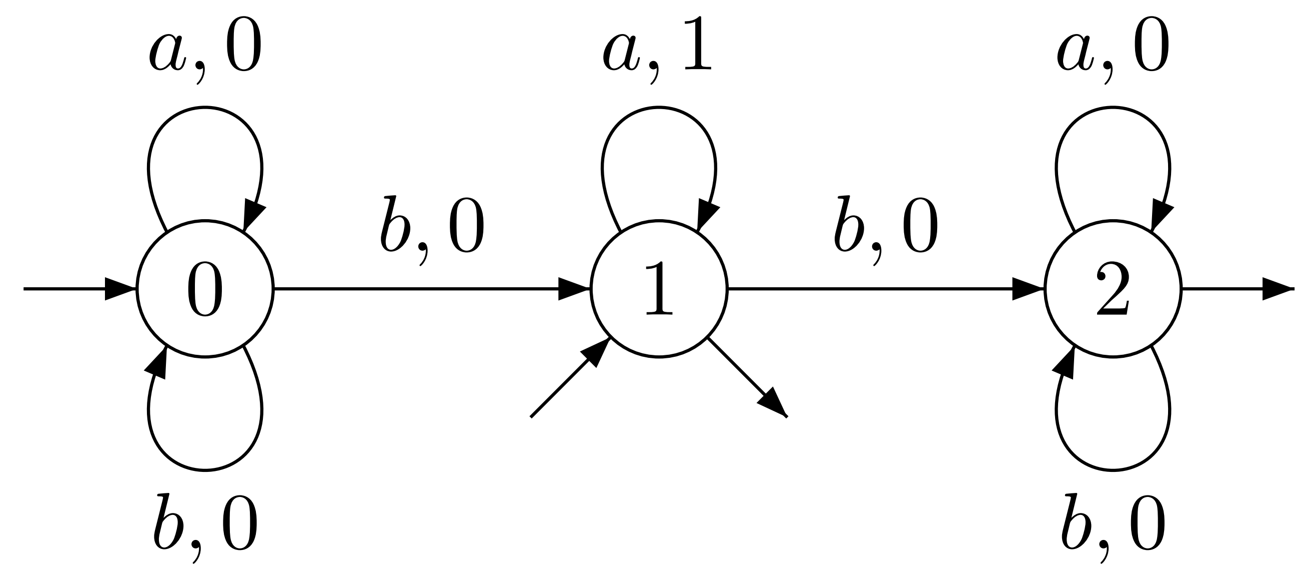
1

Semantics of *aba*: 1+(-1)+1 = 1

[Schützenberger 61]

Weighted Automata





$$(\mathbf{Z}, +, \times, 0, 1)$$

$$(\mathbf{Z} \cup \{-\infty\}, \max, +, -\infty, 0)$$

$$0 \xrightarrow{a,1} 1 \xrightarrow{b,1} 1 \xrightarrow{a,1} 1$$ 1

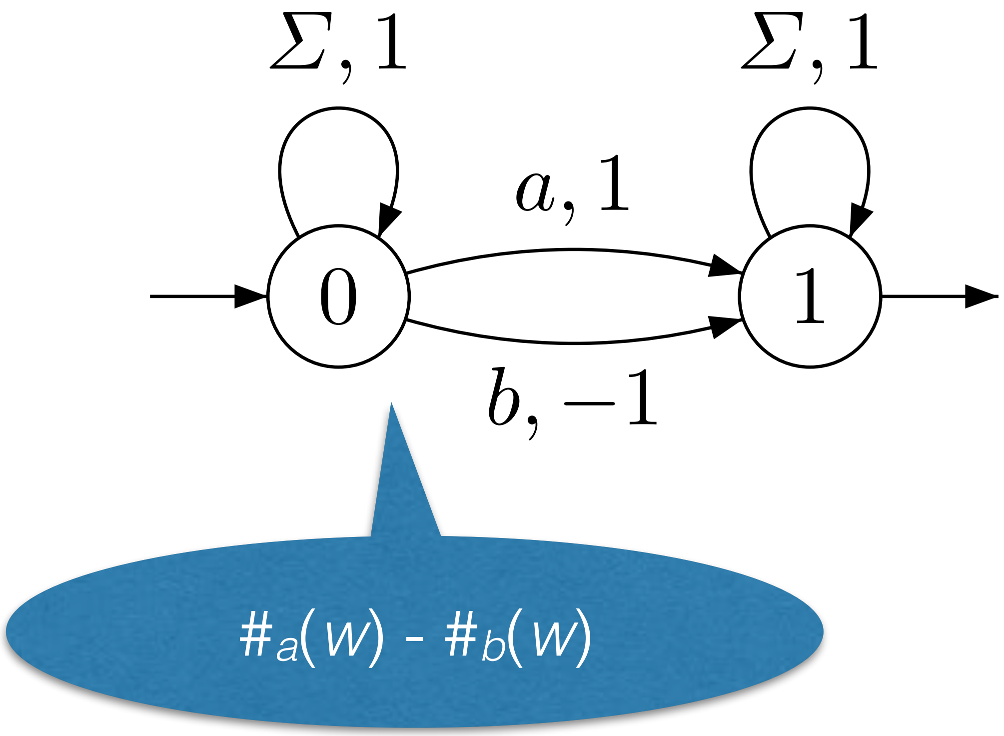
$$0 \xrightarrow{a,1} 0 \xrightarrow{b,-1} 1 \xrightarrow{a,1} 1$$ -1

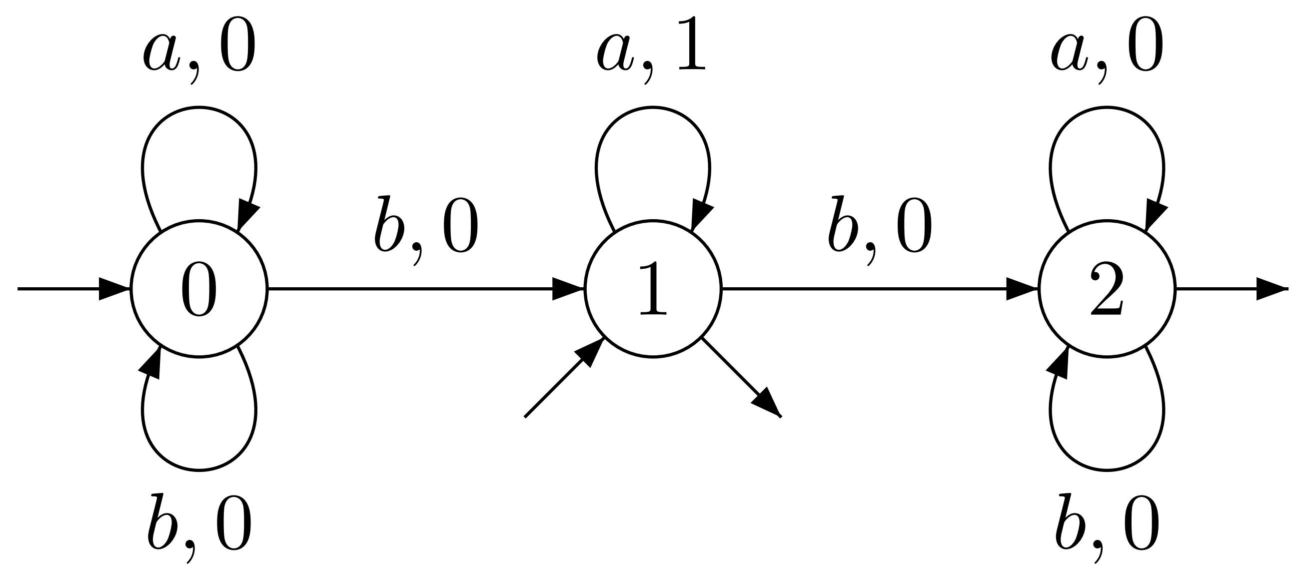
$$0 \xrightarrow{a,1} 0 \xrightarrow{b,1} 0 \xrightarrow{a,1} 1$$ 1

Semantics of *aba*: 1+(-1)+1 = 1

[Schützenberger 61]

Weighted Automata





$(\mathbf{Z}, +, \times, 0, 1)$

$(\mathbf{Z} \cup \{-\infty\}, \max, +, -\infty, 0)$

$0 \xrightarrow{a,1} 1 \xrightarrow{b,1} 1 \xrightarrow{a,1} 1$ — 1

$0 \xrightarrow{a,1} 0 \xrightarrow{b,-1} 1 \xrightarrow{a,1} 1$ — -1

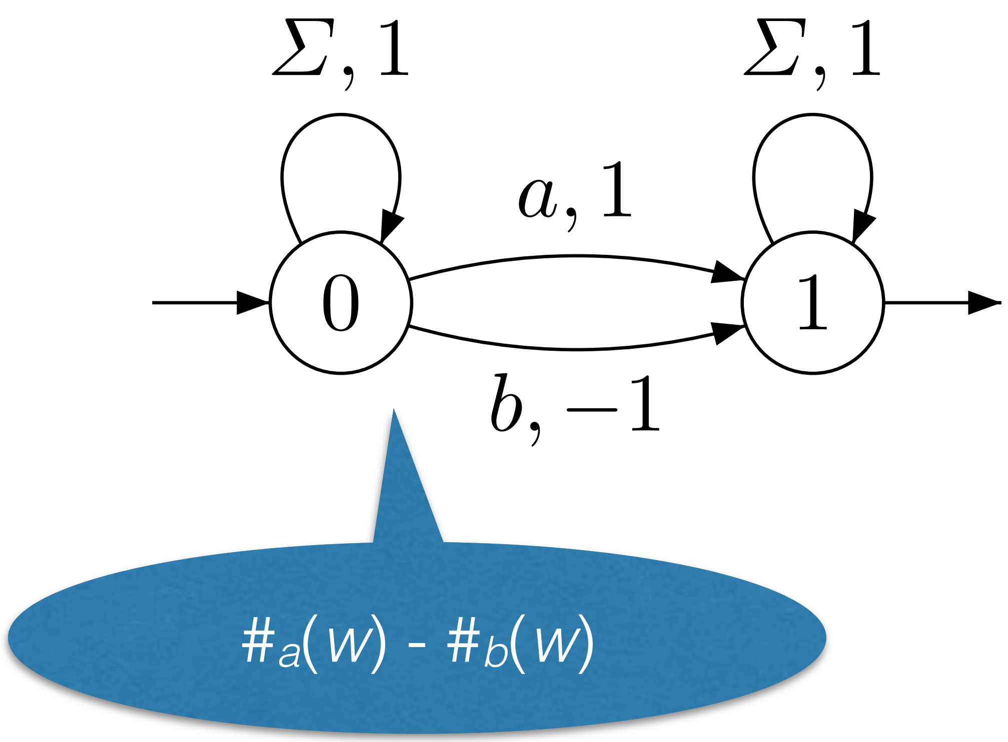
$0 \xrightarrow{a,1} 0 \xrightarrow{b,1} 0 \xrightarrow{a,1} 1$ — 1

$1 \xrightarrow{a,1} 1 \xrightarrow{a,1} 1 \xrightarrow{b,0} 2 \xrightarrow{a,0} 2$ — 2

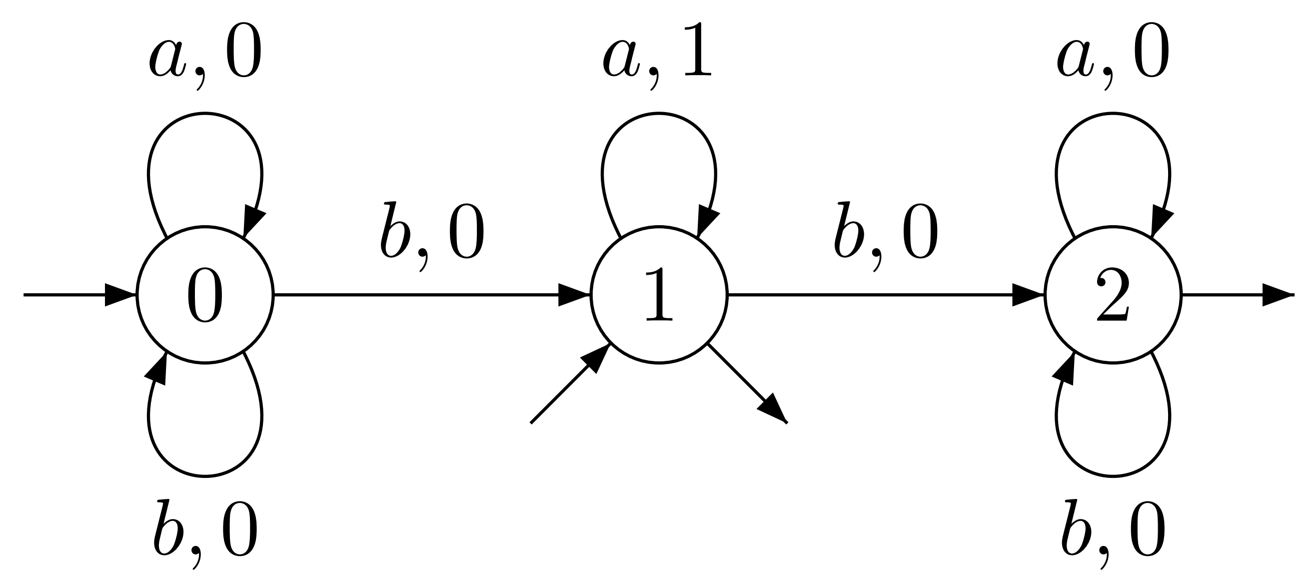
Semantics of *aba*: 1+(-1)+1 = 1

[Schützenberger 61]

Weighted Automata



$(\mathbf{Z}, +, \times, 0, 1)$



$(\mathbf{Z} \cup \{-\infty\}, \max, +, -\infty, 0)$

$0 \xrightarrow{a,1} 1 \xrightarrow{b,1} 1 \xrightarrow{a,1} 1$ — 1

$0 \xrightarrow{a,1} 0 \xrightarrow{b,-1} 1 \xrightarrow{a,1} 1$ — -1

$0 \xrightarrow{a,1} 0 \xrightarrow{b,1} 0 \xrightarrow{a,1} 1$ — 1

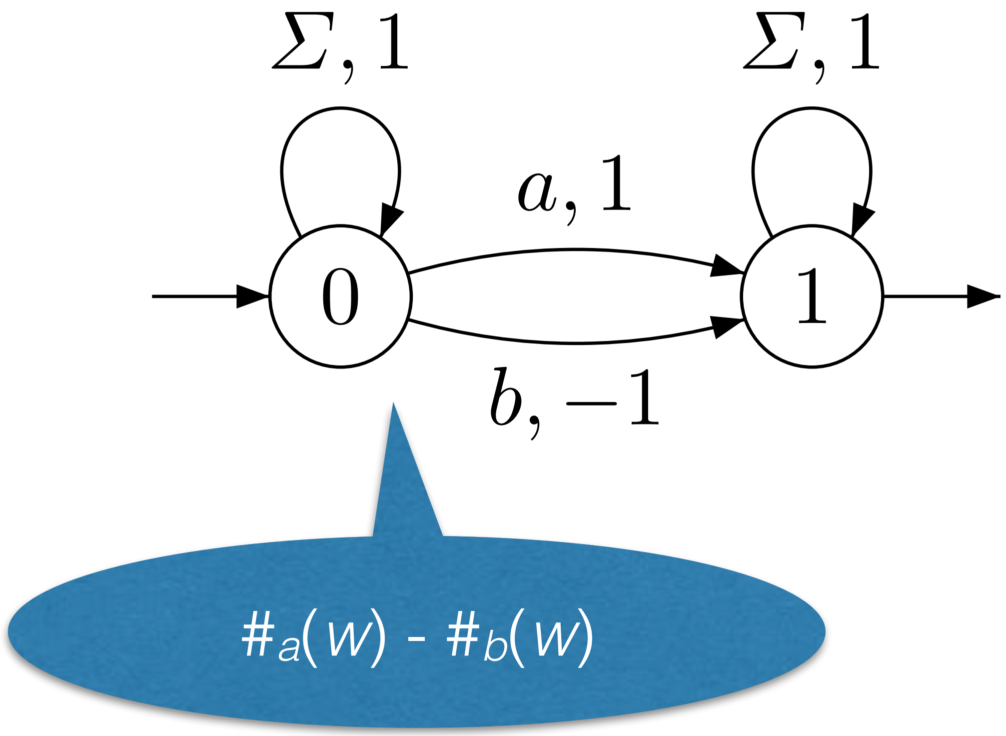
$1 \xrightarrow{a,1} 1 \xrightarrow{a,1} 1 \xrightarrow{b,0} 2 \xrightarrow{a,0} 2$ — 2

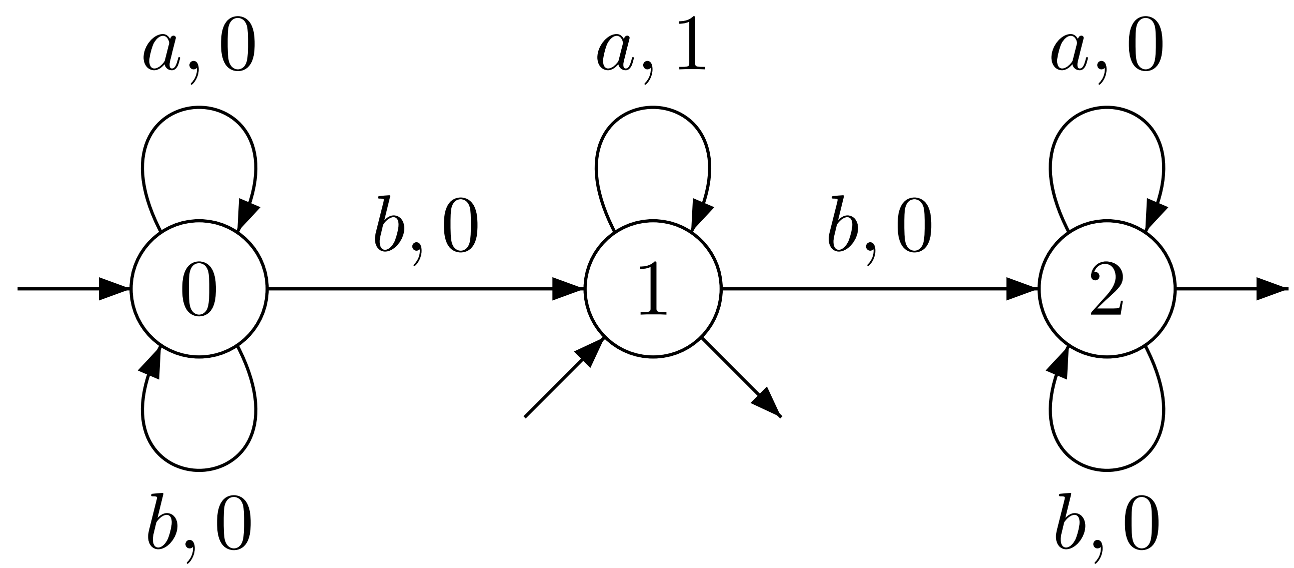
$0 \xrightarrow{a,0} 0 \xrightarrow{a,0} 0 \xrightarrow{b,0} 1 \xrightarrow{a,1} 1$ — 1

Semantics of *aba*: 1+(-1)+1 = 1

[Schützenberger 61]

Weighted Automata





$(\mathbf{Z}, +, \times, 0, 1)$

$(\mathbf{Z} \cup \{-\infty\}, \max, +, -\infty, 0)$

$0 \xrightarrow{a,1} 1 \xrightarrow{b,1} 1 \xrightarrow{a,1} 1$ — 1

$0 \xrightarrow{a,1} 0 \xrightarrow{b,-1} 1 \xrightarrow{a,1} 1$ — -1

$0 \xrightarrow{a,1} 0 \xrightarrow{b,1} 0 \xrightarrow{a,1} 1$ — 1

$1 \xrightarrow{a,1} 1 \xrightarrow{a,1} 1 \xrightarrow{b,0} 2 \xrightarrow{a,0} 2$ — 2

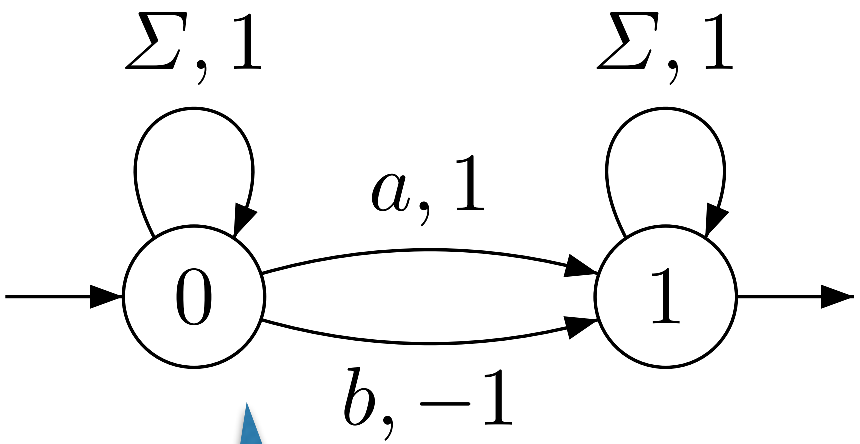
$0 \xrightarrow{a,0} 0 \xrightarrow{a,0} 0 \xrightarrow{b,0} 1 \xrightarrow{a,1} 1$ — 1

Semantics of *aba*: 1+(-1)+1 = 1

Semantics of *aaba*: max(2,1) = 2

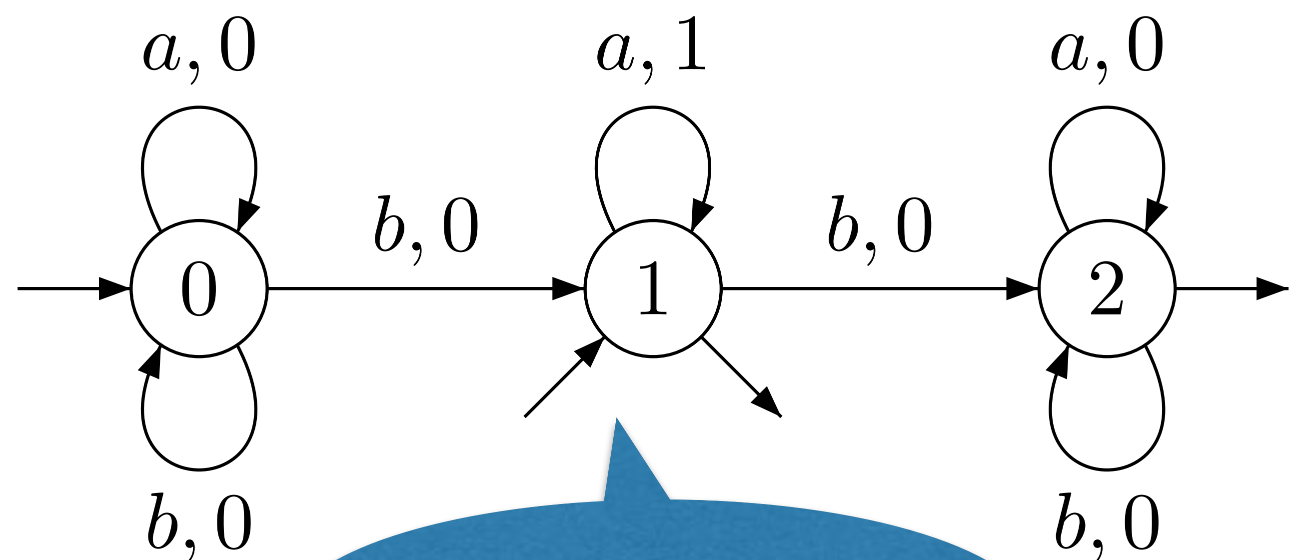
[Schützenberger 61]

Weighted Automata



$\#_a(w)$ - $\#_b(w)$

$$(\mathbf{Z}, +, \times, 0, 1)$$



max size of *a*'s blocks

$$(\mathbf{Z} \cup \{-\infty\}, \max, +, -\infty, 0)$$

$$0 \xrightarrow{a,1} 1 \xrightarrow{b,1} 1 \xrightarrow{a,1} 1 \quad 1$$

$$0 \xrightarrow{a,1} 0 \xrightarrow{b,-1} 1 \xrightarrow{a,1} 1 \quad -1$$

$$0 \xrightarrow{a,1} 0 \xrightarrow{b,1} 0 \xrightarrow{a,1} 1 \quad 1$$

$$1 \xrightarrow{a,1} 1 \xrightarrow{a,1} 1 \xrightarrow{b,0} 2 \xrightarrow{a,0} 2 \quad 2$$

$$0 \xrightarrow{a,0} 0 \xrightarrow{a,0} 0 \xrightarrow{b,0} 1 \xrightarrow{a,1} 1 \quad 1$$

Semantics of *aba*: 1+(-1)+1 = 1

Semantics of *aaba*: max(2,1) = 2

[Schützenberger 61]

How to Specify Quantitative Properties?

How to Specify Quantitative Properties?

Weighted Monadic Second Order Logic [Droste&Gastin 05] generalized to trees [Droste&Vogler 06], infinite words [Droste&Rahonis 07], nested words [Mathissen 10] or pictures [Fichtner 11]

How to Specify Quantitative Properties?

Weighted Monadic Second Order Logic [Droste&Gastin 05] generalized to trees [Droste&Vogler 06], infinite words [Droste&Rahonis 07], nested words [Mathissen 10] or pictures [Fichtner 11]

Weighted Regular Expressions over finite words [Kleene 56, Schützenberger 61]

How to Specify Quantitative Properties?

Weighted Monadic Second Order Logic [Droste&Gastin 05] generalized to trees [Droste&Vogler 06], infinite words [Droste&Rahonis 07], nested words [Mathissen 10] or pictures [Fichtner 11]

Weighted Regular Expressions over finite words [Kleene 56, Schützenberger 61]

Weighted Temporal Logics:
PCTL [Hansson&Jonsson 94], WLTL [Mandrali 12]

How to Specify Quantitative Properties?

Weighted Monadic Second Order Logic [Droste&Gastin 05]
generalized to trees [Droste&Vogler 06], infinite words [Droste&Rahonis 07], nested words [Mathissen 10] or pictures [Fichtner 11]

Weighted Regular Expressions over finite words
[Kleene 56, Schützenberger 61]

Weighted Temporal Logics:
PCTL [Hansson&Jonsson 94], WLTL [Mandrali 12]

- Core weighted logic for weighted automata
- Enhancing the logic to handle more properties: FO vs pebbles
- Deciding weighted FO logic
- A special case: the transducers

Weighted MSO

$$\varphi ::= s \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg P_a(x) \mid \neg(x \leqslant y) \mid \neg(x \in X)$$
$$\mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists X\, \varphi \mid \forall X\, \varphi$$

[Droste&Gastin 2005]

Weighted MSO

$$\varphi ::= s \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg P_a(x) \mid \neg(x \leqslant y) \mid \neg(x \in X)$$
$$\mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists X\, \varphi \mid \forall X\, \varphi$$

Negation restricted to atomic formulae

[Droste&Gastin 2005]

Weighted MSO

$$\varphi ::= s \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg P_a(x) \mid \neg(x \leqslant y) \mid \neg(x \in X)$$
$$\mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists X\, \varphi \mid \forall X\, \varphi$$

Arbitrary constants from a semiring

Negation restricted to atomic formulae

[Droste&Gastin 2005]

Weighted MSO

$$\varphi ::= s \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg P_a(x) \mid \neg(x \leqslant y) \mid \neg(x \in X)$$
$$\mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists X\, \varphi \mid \forall X\, \varphi$$

- Semantics in a semiring $\mathbb{S} = (S, +, \times, 0, 1)$
  - Atomic formulae: 0, 1
  - disjunction, existential quantifications: sum
  - conjunction, universal quantifications: product
- Inspired from the boolean semiring $\mathbb{B} = (\{0,1\}, \vee, \wedge, 0, 1)$

[Droste&Gastin 2005]

Weighted MSO

$$\varphi ::= s \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg P_a(x) \mid \neg(x \leqslant y) \mid \neg(x \in X)$$
$$\mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists X\, \varphi \mid \forall X\, \varphi$$

- Examples

$$\varphi_1 = \exists x\, P_a(x)$$

$$[\![\varphi_1]\!](w) = |w|_a$$

[Droste&Gastin 2005]

Weighted MSO

$$\varphi ::= s \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg P_a(x) \mid \neg(x \leqslant y) \mid \neg(x \in X)$$
$$\mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists X\, \varphi \mid \forall X\, \varphi$$

- Examples

$$\varphi_1 = \exists x\, P_a(x) \qquad \varphi_2 = \forall x\, \exists y\, (y \leqslant x \wedge P_a(y))$$

$$[\![\varphi_1]\!](w) = |w|_a \qquad [\![\varphi_2]\!](abaab) = 1 \times 1 \times 2 \times 3 \times 3$$

$$[\![\varphi_2]\!](a^n) = n!$$

[Droste&Gastin 2005]

Weighted MSO

$$\varphi ::= s \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg P_a(x) \mid \neg(x \leqslant y) \mid \neg(x \in X)$$
$$\mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists X\, \varphi \mid \forall X\, \varphi$$

- Examples

$$\varphi_1 = \exists x\, P_a(x)$$

$$\varphi_2 = \forall x\, \exists y\, (y \leqslant x \wedge P_a(y))$$

$$[\![\varphi_1]\!](w) = |w|_a$$

$$[\![\varphi_2]\!](abaab) = 1 \times 1 \times 2 \times 3 \times 3$$

$$[\![\varphi_2]\!](a^n) = n!$$

Too big to be computed by a weighted automaton

[Droste&Gastin 2005]

Weighted MSO

$$\varphi ::= s \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg P_a(x) \mid \neg(x \leqslant y) \mid \neg(x \in X)$$
$$\mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \neg$$

Need to restrict weighted MSO

- Examples

$$\varphi_1 = \exists x\, P_a(x) \qquad \varphi_2 = \forall x\, \exists y\, (y \leqslant x \wedge P_a(y))$$

$$[\![\varphi_1]\!](w) = |w|_a \qquad [\![\varphi_2]\!](abaab) = 1 \times 1 \times 2 \times 3 \times 3$$

$$[\![\varphi_2]\!](a^n) = n!$$

Too big to be computed by a weighted automaton

[Droste&Gastin 2005]

Weighted MSO

$$\varphi ::= s \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg P_a(x) \mid \neg(x \leqslant y) \mid \neg(x \in X)$$
$$\mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists X\, \varphi \mid \forall X\, \varphi$$

Theorem: weighted automata = restricted wMSO

[Droste&Gastin 2005]

Weighted MSO

$$\varphi ::= s \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg P_a(x) \mid \neg(x \leqslant y) \mid \neg(x \in X)$$
$$\mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists X\, \varphi \mid \forall X\, \varphi$$

Theorem: weighted automata = restricted wMSO

[Droste&Gastin 2005]

Weighted MSO

$$\varphi ::= s \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg P_a(x) \mid \neg(x \leqslant y) \mid \neg(x \in X)$$
$$\mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists X\, \varphi \mid \forall X\, \varphi$$

φ almost boolean

Theorem: weighted automata = restricted wMSO

[Droste&Gastin 2005]

Weighted MSO

$$\varphi ::= s \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg P_a(x) \mid \neg(x \leqslant y) \mid \neg(x \in X)$$
$$\mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists X\, \varphi \mid \forall X\, \varphi$$

commutativity

φ almost boolean

Theorem: weighted automata = restricted wMSO

[Droste&Gastin 2005]

Core weighted MSO logic

- Boolean fragment

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

[Gastin&Monmege 2015]

Core weighted MSO logic

- Boolean fragment

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

- Step formulae $\quad \Psi ::= s \mid \varphi\,?\,\Psi : \Psi$

[Gastin&Monmege 2015]

Core weighted MSO logic

- Boolean fragment

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

- Step formulae $\Psi ::= s \mid \varphi \,?\, \Psi : \Psi$

if … then … else …

[Gastin&Monmege 2015]

Core weighted MSO logic

- Boolean fragment

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

- Step formulae

$$\Psi ::= s \mid \varphi\,?\,\Psi : \Psi$$

if ... then ... else ...

$$P_a(x)\,?\,1 : 0$$

[Gastin&Monmege 2015]

Core weighted MSO logic

- Boolean fragment

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

- Step formulae

$$\Psi ::= s \mid \varphi \,?\, \Psi : \Psi$$

if … then … else …

$$P_a(x) \,?\, 1 : 0$$

$$P_a(x) \,?\, 1 : (P_b(x) \,?\, -1 : 0)$$

[Gastin&Monmege 2015]

Core weighted MSO logic

- Boolean fragment

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

- Step formulae $\Psi ::= s \mid \varphi\,?\,\Psi : \Psi$

if … then … else …

$$P_a(x)\,?\,1 : 0$$

$$P_a(x)\,?\,1 : (P_b(x)\,?\,-1 : 0)$$

$$x \in X_1\,?\,s_1 : (x \in X_2\,?\,s_2 : \cdots (x \in X_{n-1}\,?\,s_{n-1} : s_n) \cdots)$$

[Gastin&Monmege 2015]

Core weighted MSO logic

- Boolean fragment

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

- Step formulae

$$\Psi ::= s \mid \varphi \,?\, \Psi : \Psi$$

if … then … else …

$$P_a(x) \,?\, 1 : 0$$

$$P_a(x) \,?\, 1 : (P_b(x) \,?\, -1 : 0)$$

$$x \in X_1 \,?\, s_1 : (x \in X_2 \,?\, s_2 : \cdots (x \in X_{n-1} \,?\, s_{n-1} : s_n) \cdots)$$

$$[\![\Psi]\!](w, \sigma) = s$$

some value occurring in Ψ

[Gastin&Monmege 2015]

Core weighted MSO logic

- Boolean fragment

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

- Step formulae $\quad \Psi ::= s \mid \varphi\,?\,\Psi : \Psi$

$P_a(x)\,?\,1 : 0$

...

$P_a(x)$

A step formula takes finitely many values

For each value, the pre-image is MSO-definable

$x \in X$ ... $X_2\,?\,s_2 : \cdots (x \in X_{n-1}\,?\,s_{n-1} : s_n)\cdots)$

$$[\![\Psi]\!](w, \sigma) = s$$

some value occurring in Ψ

Core weighted MSO logic

- Boolean fragment

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

- Step formulae $\quad \Psi ::= s \mid \varphi \,?\, \Psi : \Psi$

- core wMSO $\quad \Phi ::= \mathbf{0} \mid \varphi \,?\, \Phi : \Phi \mid \Phi + \Phi \mid \sum_x \Phi \mid \sum_X \Phi \mid \prod_x \Psi$

Core weighted MSO logic

- Boolean fragment

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

- Step formulae $\Psi ::= s \mid \varphi\, ?\, \Psi : \Psi$

- core wMSO $\Phi ::= \mathbf{0} \mid \varphi\, ?\, \Phi : \Phi \mid \Phi + \Phi \mid \sum_x \Phi \mid \sum_X \Phi \mid \prod_x \Psi$

no constants

Core weighted MSO logic

- Boolean fragment

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

- Step formulae $\quad \Psi ::= s \mid \varphi \,?\, \Psi : \Psi$

- core wMSO $\quad \Phi ::= \mathbf{0} \mid \varphi \,?\, \Phi : \Phi \mid \Phi + \Phi \mid \sum_x \Phi \mid \sum_X \Phi \mid \prod_x \Psi$

no constants

if … then … else …

Core weighted MSO logic

- Boolean fragment

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

- Step formulae $\quad \Psi ::= s \mid \varphi\, ?\, \Psi : \Psi$

- core wMSO $\quad \Phi ::= \mathbf{0} \mid \varphi\, ?\, \Phi : \Phi \mid \Phi + \Phi \mid \sum_x \Phi \mid \sum_X \Phi \mid \prod_x \Psi$

no constants

if … then … else …

Assigns a value from Ψ to each position

Core weighted MSO logic

- Boolean fragment

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

- Step formulae $\quad \Psi ::= s \mid \varphi \,?\, \Psi : \Psi$

- core wMSO $\quad \Phi ::= \mathbf{0} \mid \varphi \,?\, \Phi : \Phi \mid \Phi + \Phi \mid \sum_x \Phi \mid \sum_X \Phi \mid \prod_x \Psi$

no constants

if … then … else …

Assigns a value from Ψ to each position

$$\{\!|\prod_x \Psi|\!\}(w,\sigma) = \{\!\{(\llbracket\Psi\rrbracket(w,\sigma[x \mapsto i]))_i\}\!\} \in \mathbb{N}\langle R^\star\rangle$$

singleton multiset

Core weighted MSO logic

- Boolean fragment

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

- Step formulae $\quad \Psi ::= s \mid \varphi \,?\, \Psi : \Psi$

- core wMSO $\quad \Phi ::= \mathbf{0} \mid \varphi \,?\, \Phi : \Phi \mid \Phi + \Phi \mid \sum_x \Phi \mid \sum_X \Phi \mid \prod_x \Psi$

- Semantics
  - $\{\!|\, \mathbf{0} \,|\!\}(w,\sigma) = \varnothing$
  - sums over multisets

$$\{\!|\, \prod_x \Psi \,|\!\}(w,\sigma) = \{\!\{ (\llbracket \Psi \rrbracket (w, \sigma[x \mapsto i]))_i \}\!\} \in \mathbb{N}\langle R^\star \rangle$$

Multisets of weight structures

- A run generates a sequence of weights $\mathrm{wgt}(\rho) = s_1 s_2 \cdots s_n$
- Abstract semantics $\{\!|\mathcal{A}|\!\}(w) = \{\{\mathrm{wgt}(\rho) \mid \rho \text{ run on } w\}\}$

multiset

Multisets of weight structures

- A run generates a sequence of weights $\mathrm{wgt}(\rho) = s_1 s_2 \cdots s_n$
- Abstract semantics $\{\!|\mathcal{A}|\!\}(w) = \{\{\mathrm{wgt}(\rho) \mid \rho \text{ run on } w\}\}$

$$\{\!|\mathcal{A}|\!\} \colon \Sigma^\star \to \mathbb{N}\langle R^\star \rangle$$

multiset

weights of A

Multisets of weight structures

- A run generates a sequence of weights $\mathrm{wgt}(\rho) = s_1 s_2 \cdots s_n$
- Abstract semantics $\{\!|\mathcal{A}|\!\}(w) = \{\{\mathrm{wgt}(\rho) \mid \rho \text{ run on } w\}\}$

$$\{\!|\mathcal{A}|\!\} \colon \Sigma^\star \to \mathbb{N}\langle R^\star \rangle$$

multiset

weights of A

- Aggregation $\mathsf{aggr} \colon \mathbb{N}\langle R^\star \rangle \to S$

Multisets of weight structures

Semiring: sum-product

$$\mathsf{aggr}_{\mathrm{sp}}(A) = \sum \prod A = \sum_{r_1 \cdots r_n \in A} r_1 \times \cdots \times r_n$$

$$\{\!|\mathcal{A}|\!\} \colon \Sigma^\star \to \mathbb{N}\langle R^\star \rangle$$

multiset

weights of A

- Aggregation $\qquad \mathsf{aggr} \colon \mathbb{N}\langle R^\star \rangle \to S$

[Droste&Perevoshchikov 2014]

Multisets of weight structures

Semiring: sum-product

$$\mathsf{aggr}_{\mathtt{sp}}(A) = \sum \prod A = \sum_{r_1 \cdots r_n \in A} r_1 \times \cdots \times r_n$$

Valuation monoid: sum-valuation

$$\mathsf{aggr}_{\mathtt{sv}}(A) = \sum \mathsf{Val}(A) = \sum_{r_1 \cdots r_n \in A} \mathsf{Val}(r_1 \cdots r_n)$$

weights of A

- Aggregation $\mathsf{aggr} \colon \mathbb{N}\langle R^\star \rangle \to S$

[Droste&Perevoshchikov 2014]

Multisets of weight structures

Semiring: sum-product

$$\text{aggr}_{\texttt{sp}}(A) = \sum \prod A = \sum_{r_1 \cdots r_n \in A} r_1 \times \cdots \times r_n$$

Valuation monoid: sum-valuation

$$\text{aggr}_{\texttt{sv}}(A) = \sum \mathsf{Val}(A) = \sum_{r_1 \cdots r_n \in A} \mathsf{Val}(r_1 \cdots r_n)$$

weights of A

Average value
Discounted value…

- Aggregation $\rightarrow S$

[Droste&Perevoshchikov 2014]

Multisets of weight structures

- A run generates a sequence of weights $\mathrm{wgt}(\rho) = s_1 s_2 \cdots s_n$
- Abstract semantics $\{\!|\mathcal{A}|\!\}(w) = \{\{\mathrm{wgt}(\rho) \mid \rho \text{ run on } w\}\}$

  $\{\!|\mathcal{A}|\!\} \colon \Sigma^\star \to \mathbb{N}\langle R^\star \rangle$

  multiset

  weights of A

- Aggregation $\mathsf{aggr} \colon \mathbb{N}\langle R^\star \rangle \to S$
- Concrete semantics $[\![\mathcal{A}]\!] = \mathsf{aggr} \circ \{\!|\mathcal{A}|\!\} \colon \Sigma^\star \to S$

Core weighted MSO logic

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

$$\Psi ::= s \mid \varphi\,?\,\Psi : \Psi$$

$$\Phi ::= \mathbf{0} \mid \varphi\,?\,\Phi : \Phi \mid \Phi + \Phi \mid \textstyle\sum_x \Phi \mid \sum_X \Phi \mid \prod_x \Psi$$

Theorem: weighted automata = core wMSO

[Gastin&Monmege 2015]

Core weighted MSO logic

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

$$\Psi ::= s \mid \varphi\, ?\, \Psi : \Psi$$

$$\Phi ::= \mathbf{0} \mid \varphi\, ?\, \Phi : \Phi \mid \Phi + \Phi \mid \textstyle\sum_x \Phi \mid \sum_X \Phi \mid \prod_x \Psi$$

Theorem: weighted automata = core wMSO

- Abstract semantics $\{\!|\, - \,|\!\}: \Sigma^\star \to \mathbb{N}\langle R^\star \rangle$

[Gastin&Monmege 2015]

Core weighted MSO logic

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

$$\Psi ::= s \mid \varphi\,?\,\Psi : \Psi$$

$$\Phi ::= \mathbf{0} \mid \varphi\,?\,\Phi : \Phi \mid \Phi + \Phi \mid \textstyle\sum_x \Phi \mid \sum_X \Phi \mid \prod_x \Psi$$

Theorem: weighted automata = core wMSO

- Abstract semantics $\{\!|\, - \,|\!\}: \Sigma^\star \to \mathbb{N}\langle R^\star \rangle$
- Concrete semantics $[\![-]\!] = \mathsf{aggr} \circ \{\!|\, - \,|\!\}: \Sigma^\star \to S$

[Gastin&Monmege 2015]

Core weighted MSO logic

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

$$\Psi ::= s \mid \varphi\,?\,\Psi : \Psi$$

$$\Phi ::= \mathbf{0} \mid \varphi\,?\,\Phi : \Phi \mid \Phi + \Phi \mid \textstyle\sum_x \Phi \mid \sum_X \Phi \mid \prod_x \Psi$$

Theorem: weighted automata = core wMSO

- Abstract semantics ...
- Concrete sema... $\Sigma^\star \to S$

Easy constructive proofs
preservation of the constants
no restriction on core wMSO
no hypotheses on weights

[Gastin&Monmege 2015]

Extensions

More general models than words: trees, nested words, labelled graphs, infinite words…

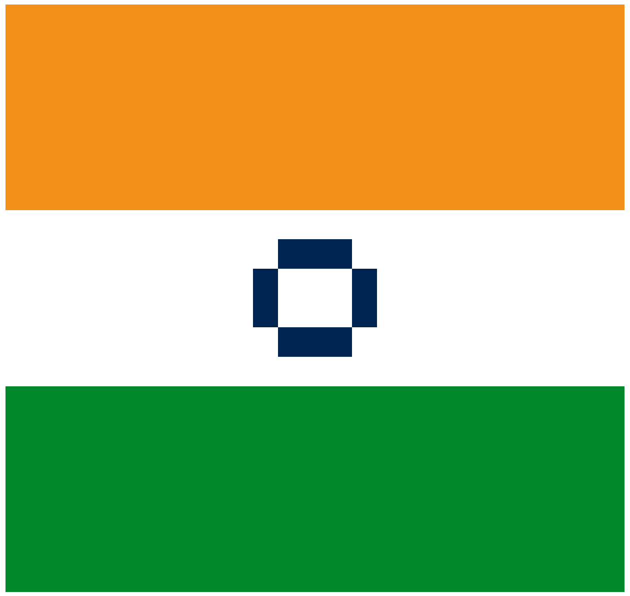
Other logics: other manageable fragment of wMSO formula than core wMSO

More powerful automata: finding equivalent fragments of wMSO

Logical Specifications: Query Examples

Logical Specifications: Query Examples

Is there a line of green pixels?



Logical Specifications: Query Examples

Is there a line of green pixels?

How many lines of green pixels are there?

Logical Specifications: Query Examples

Is there a line of green pixels?

How many lines of green pixels are there?

What is the size of the picture?

Logical Specifications: Query Examples

Is there a line of green pixels?

How many lines of green pixels are there?

What is the size of the picture?

What is the average lightness?

Logical Specifications: Query Examples

Is there a line of green pixels?

How many lines of green pixels are there?

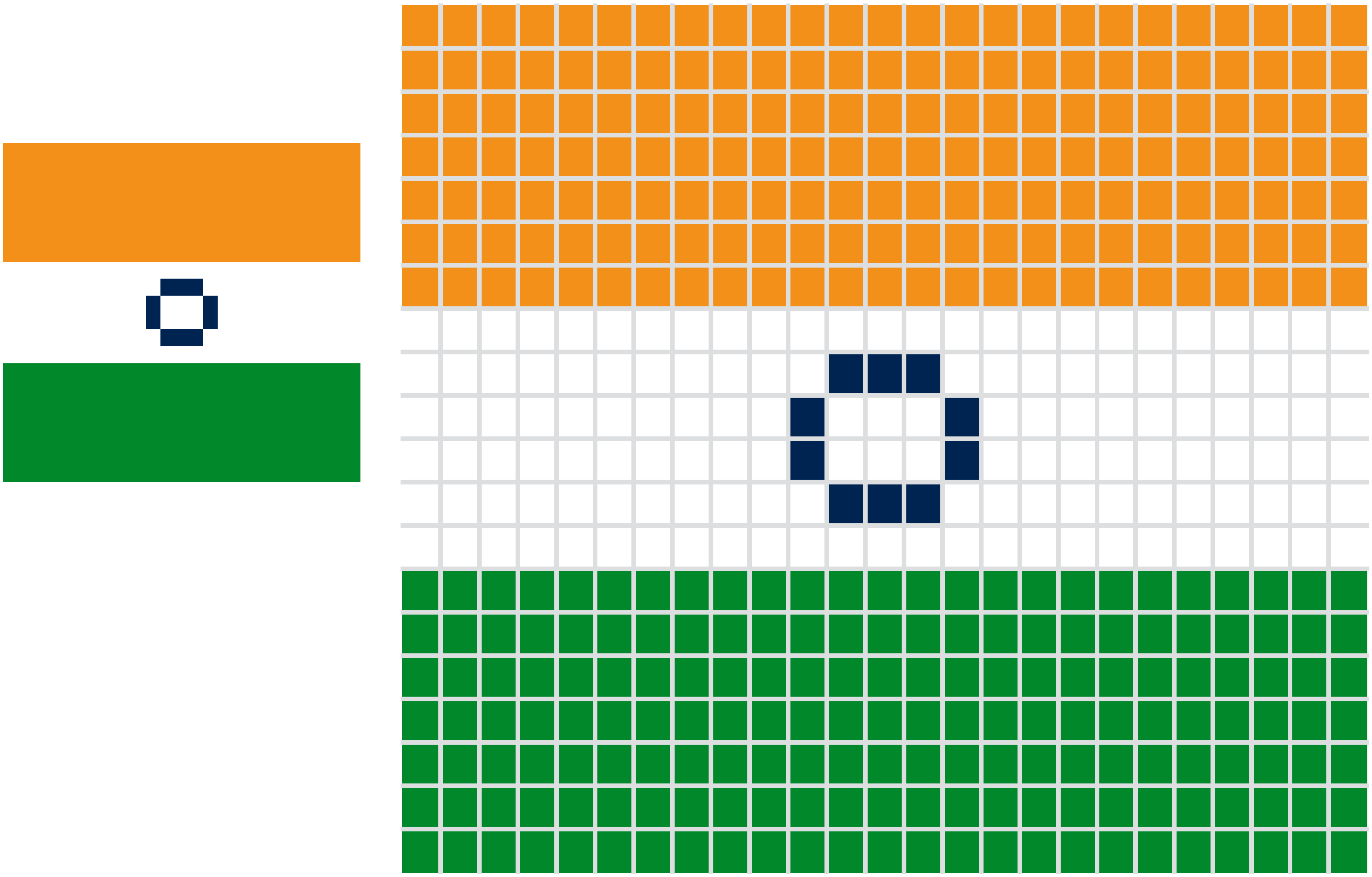
What is the size of the picture?

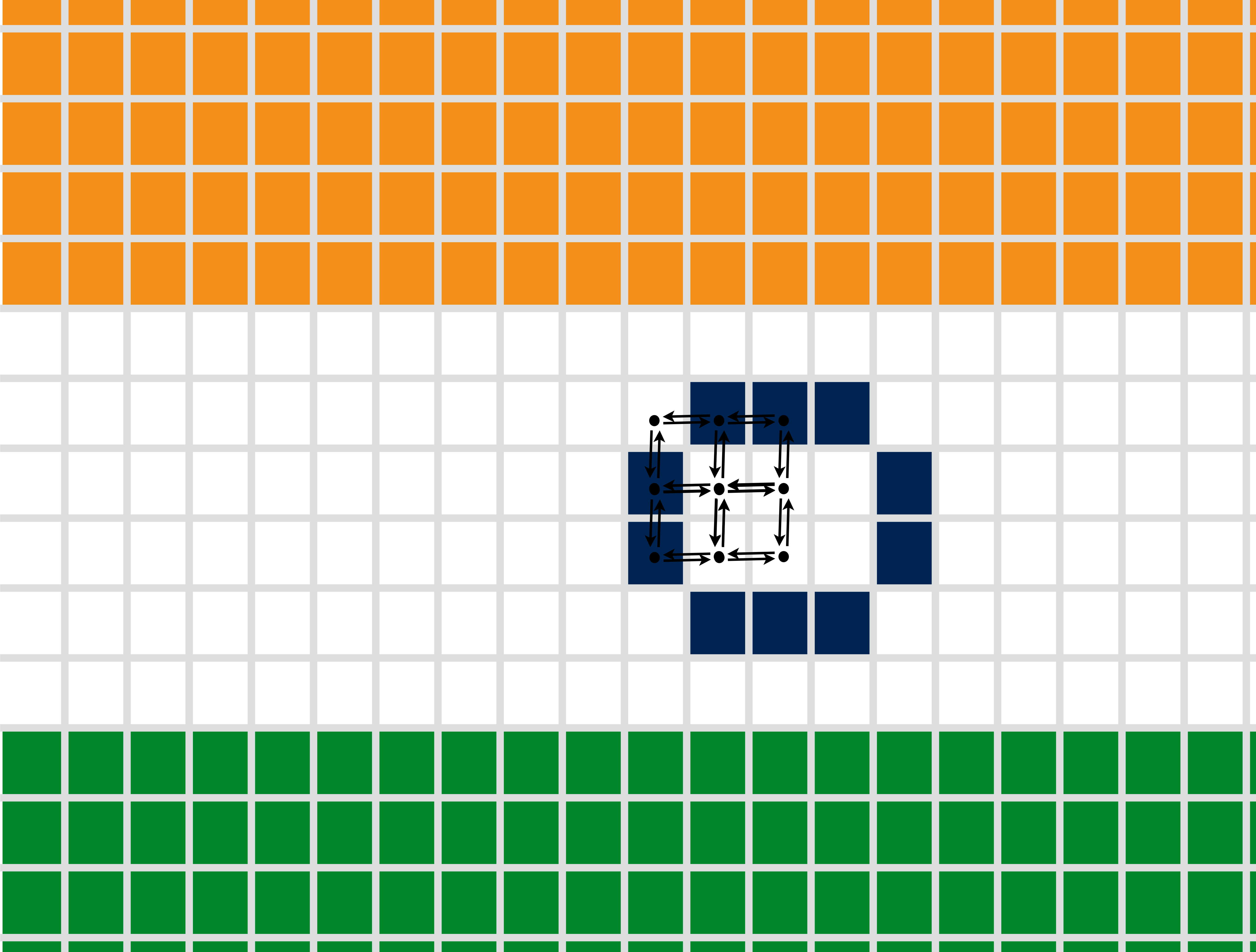
What is the average lightness?

What is the size of the biggest monochromatic rectangle?

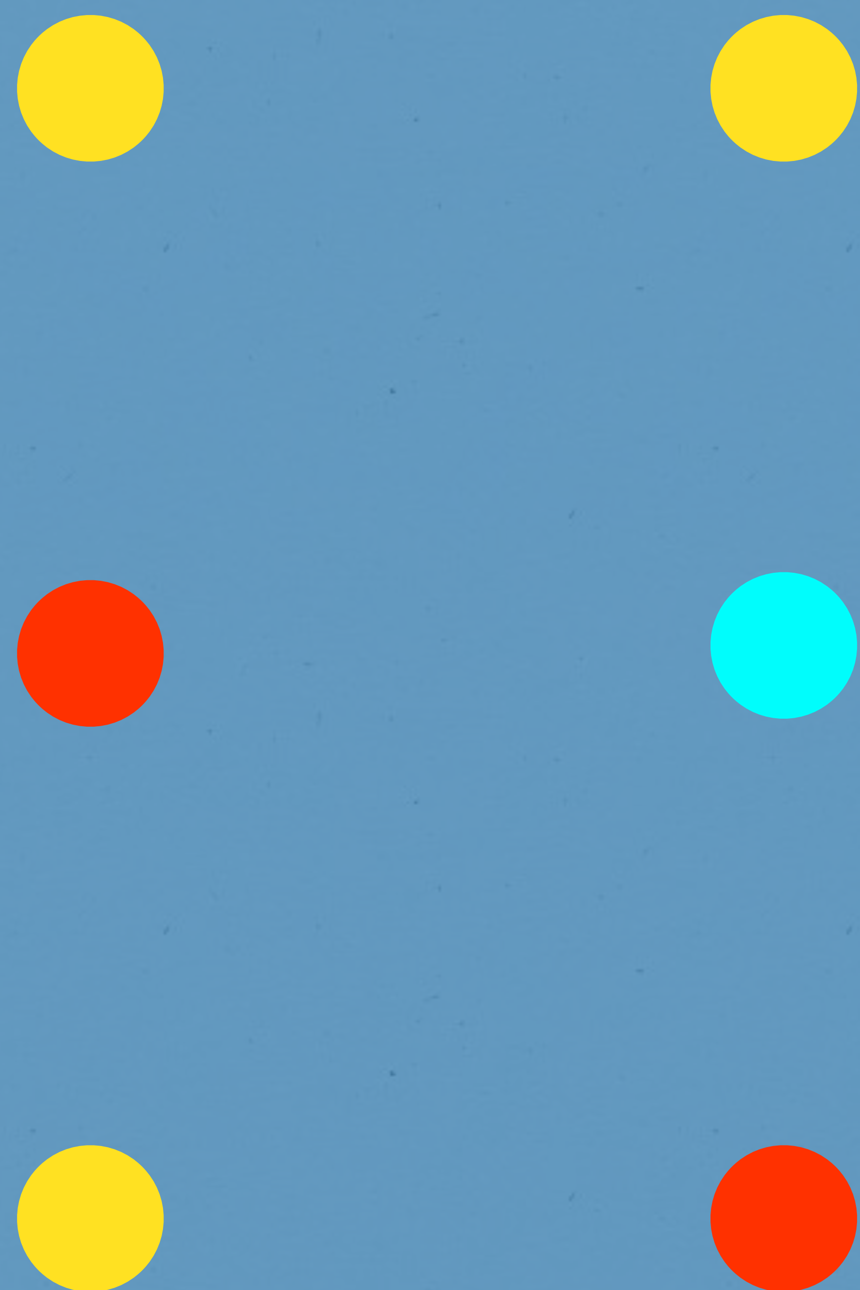
Modelling a picture as a graph

Modelling a picture as a graph



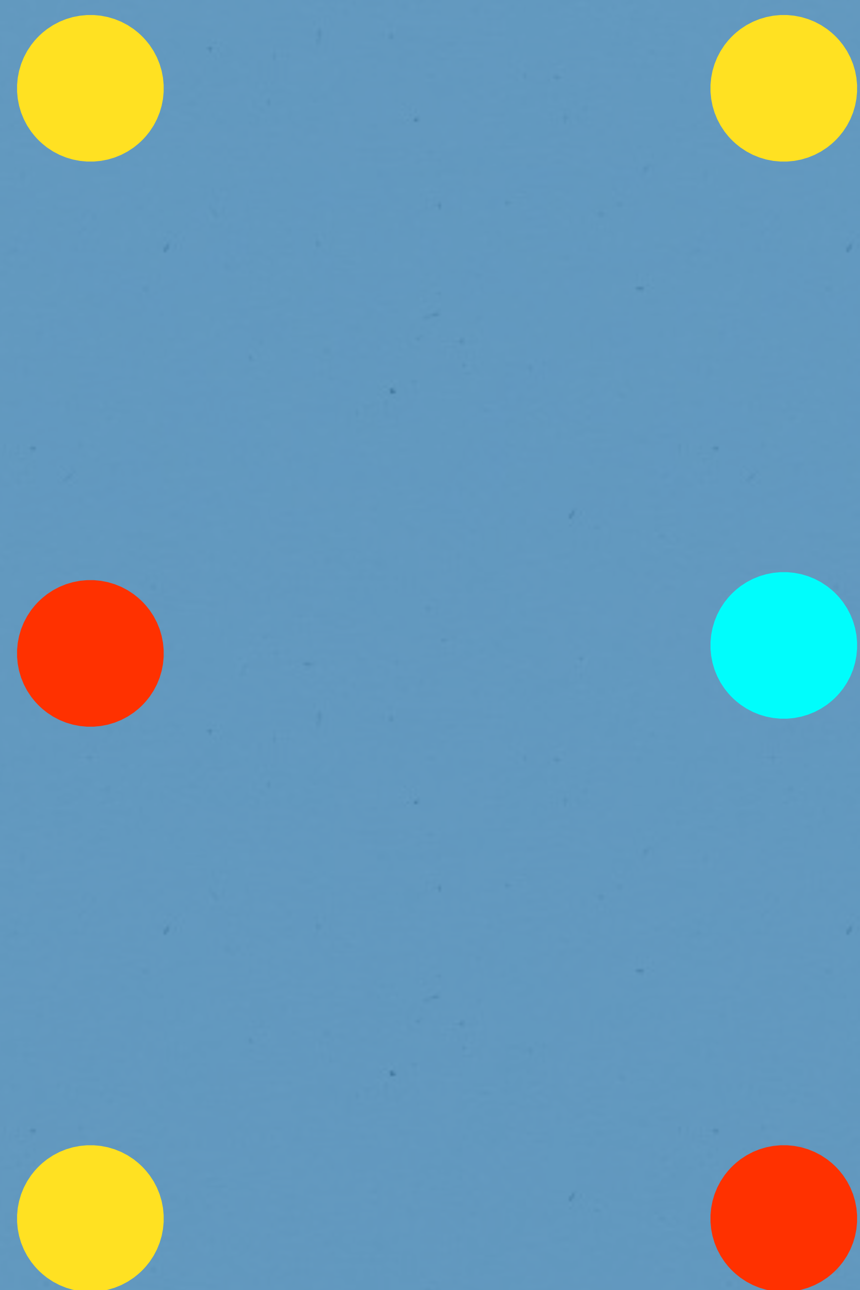


$$G = (V, (E_d)_{d \in D}, \lambda)$$



$$G = (V, (E_d)_{d \in D}, \lambda)$$

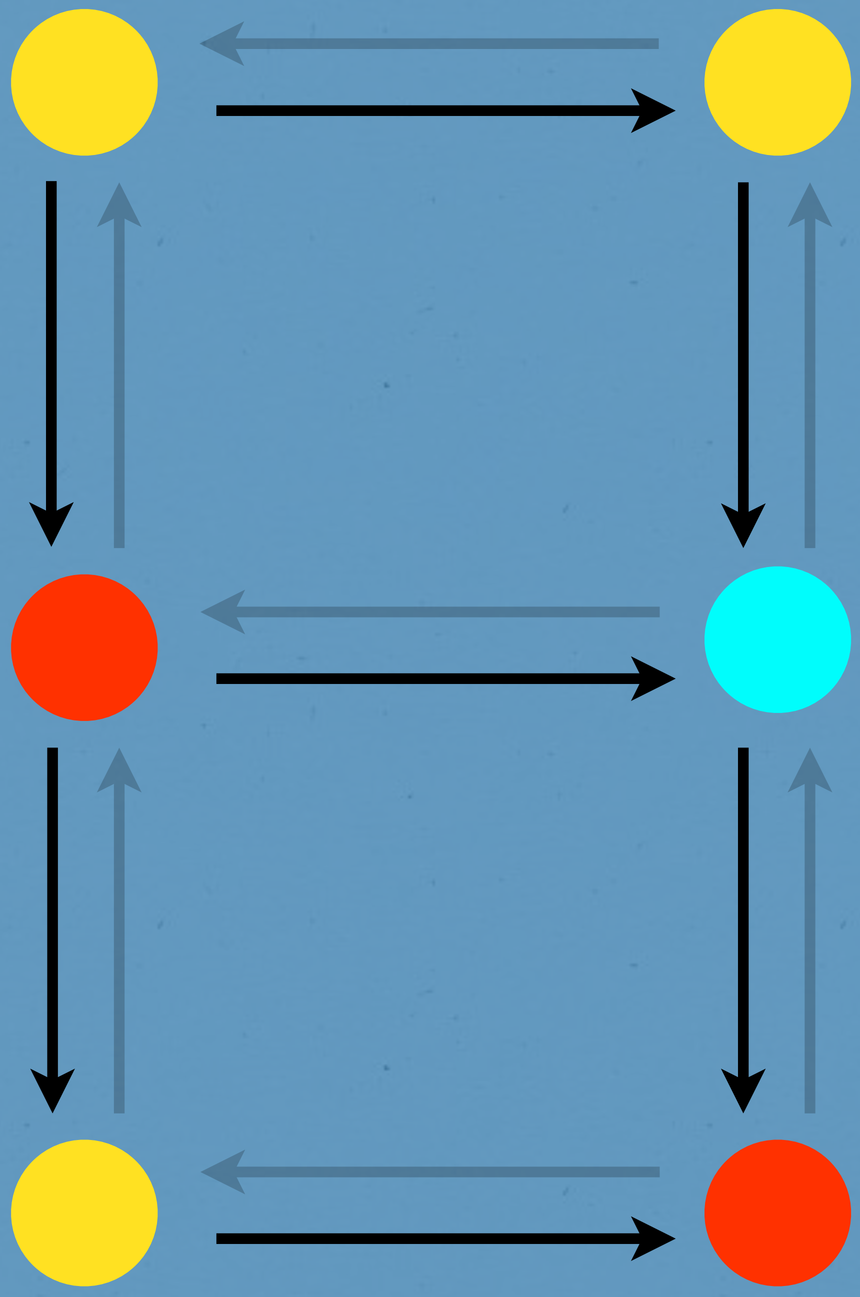
V	set of vertices
λ	labels of vertices



$$G = (V, (E_d)_{d \in D}, \lambda)$$

V	set of vertices
λ	labels of vertices
D	set of directions

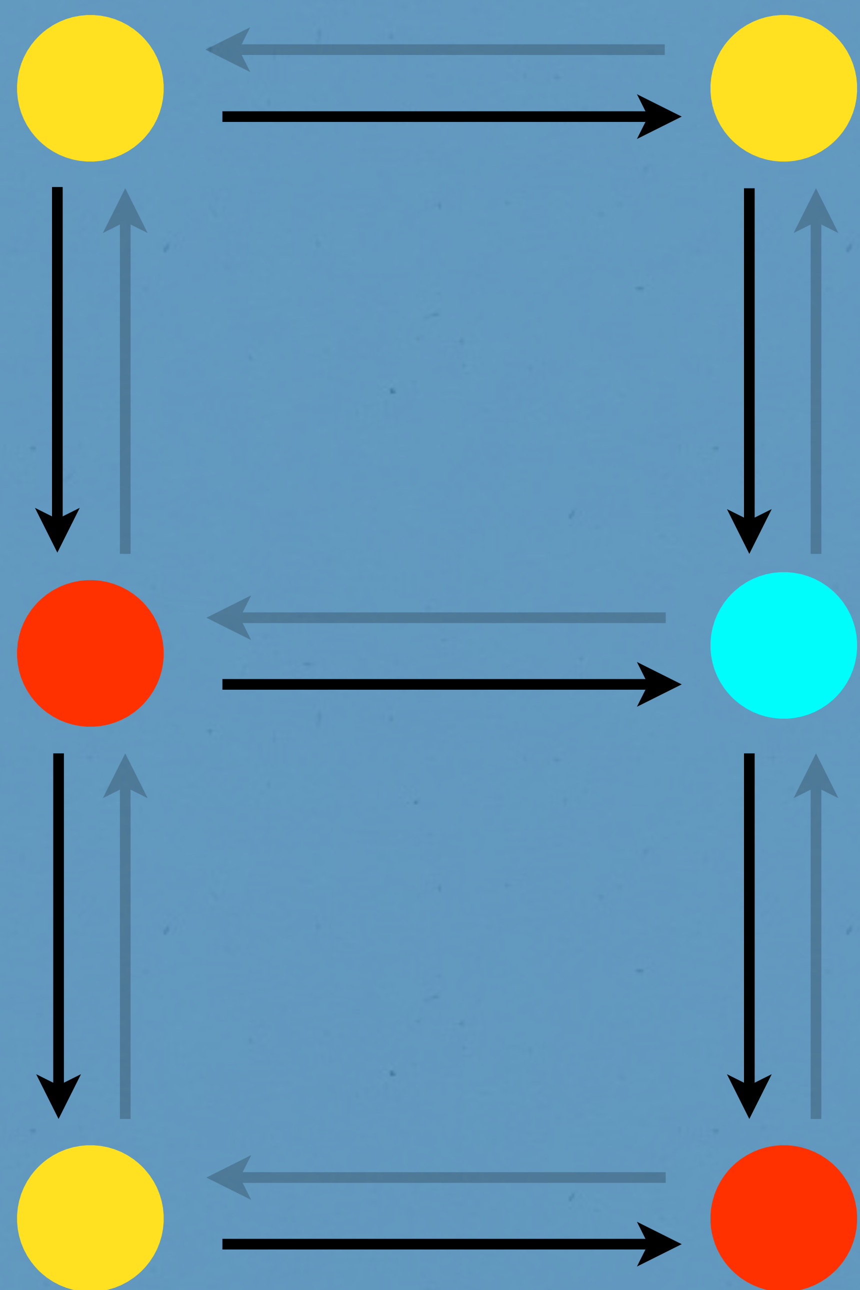
$$D = \{\rightarrow, \downarrow\} \cup \{\leftarrow, \uparrow\}$$



$$G = (V, (E_d)_{d \in D}, \lambda)$$

V	set of vertices
λ	labels of vertices
D	set of directions
E_d	set of *d*-edges

$$D = \{\rightarrow, \downarrow\} \cup \{\leftarrow, \uparrow\}$$



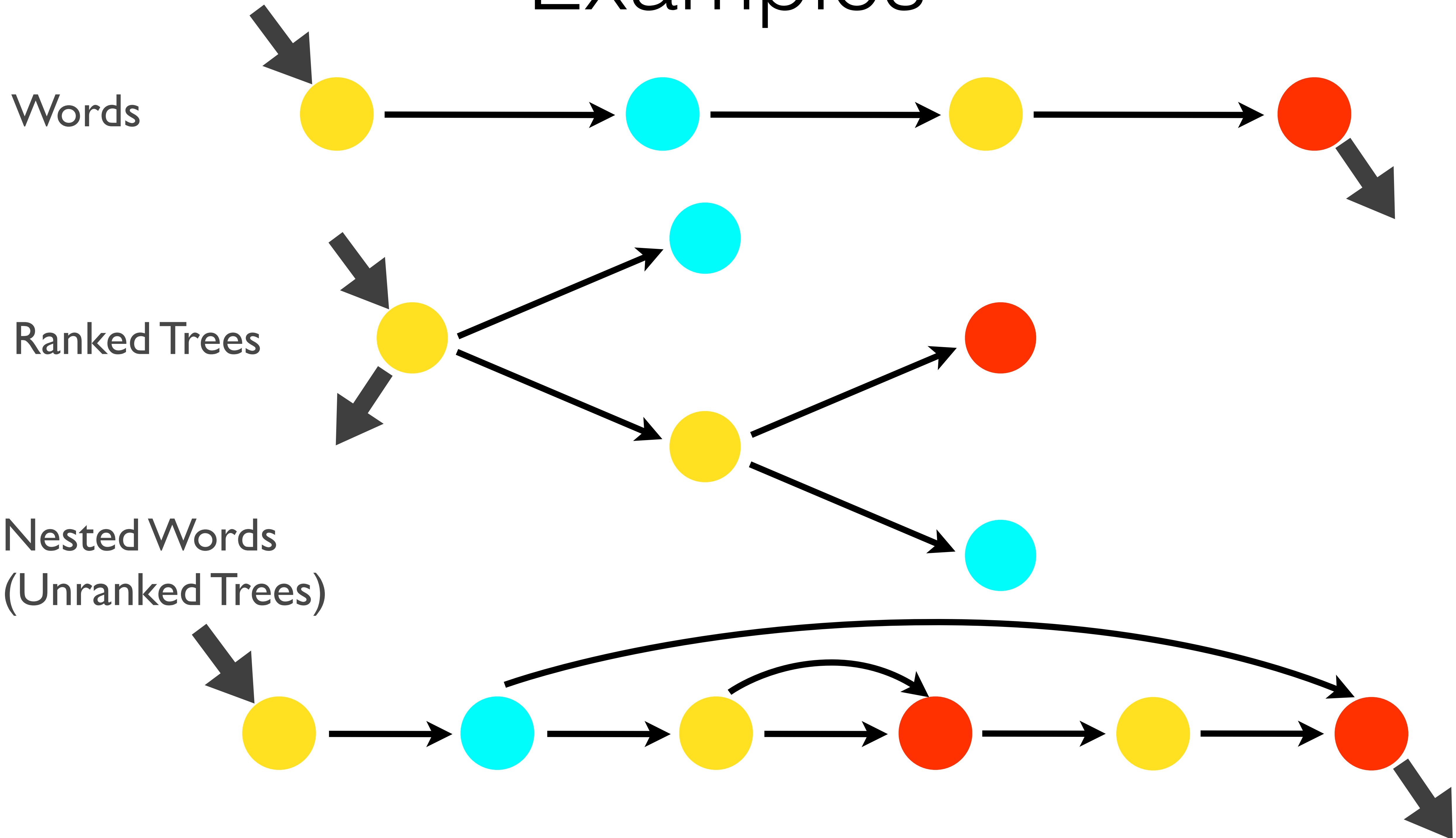
$$D = \{\rightarrow, \downarrow\} \cup \{\leftarrow, \uparrow\}$$

$$G = (V, (E_d)_{d \in D}, \lambda)$$

V	set of vertices
λ	labels of vertices
D	set of directions
E_d	set of *d*-edges

deterministic
(hence bounded degree)

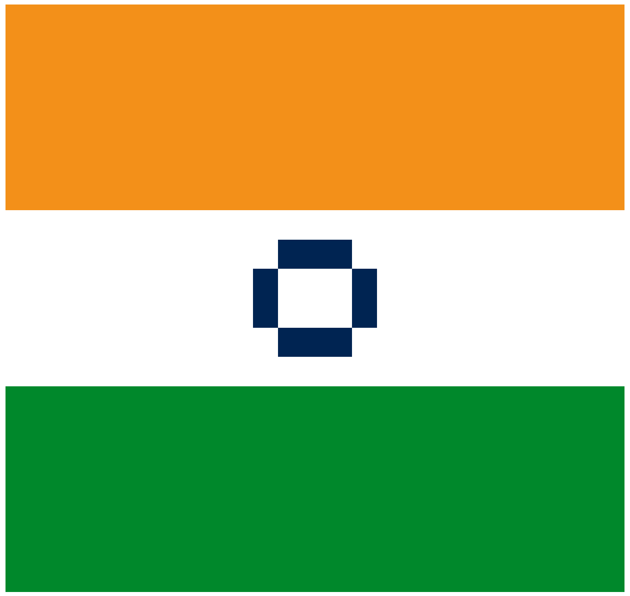
Examples



Mazurkiewicz Traces   Pictures   Graphs of bounded degree

Labyrinths   ...

Logical Specifications: Query Examples



Is there a line of green pixels?

How many lines of green pixels are there?

What is the size of the picture?

What is the average lightness?

What is the size of the biggest monochromatic rectangle?

Logical Specifications: Query Examples

Is there a line of green pixels?

$$\exists x \forall y [(R^*_{\rightarrow}(x,y) \vee R^*_{\rightarrow}(y,x)) \Rightarrow P_{\blacksquare}(y)] ?1 : 0$$

How many lines of green pixels are there?

What is the size of the picture?

Boolean fragment: first-order logic

$$\varphi ::= \top \mid (x = y) \mid \mathsf{init}(x) \mid \mathsf{final}(x) \mid P_a(x) \mid R_d(x,y) \mid R_d^\star(x,y) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi$$

$$G, \sigma \models P_a(x) \quad \text{iff.} \quad \lambda(\sigma(x)) = a$$

$$G, \sigma \models R_d(x,y) \quad \text{iff.} \quad (\sigma(x), \sigma(y)) \in E_d$$

$$G, \sigma \models R_d^\star(x,y) \quad \text{iff. there is a } d\text{-path from } \sigma(x) \text{ to } \sigma(y)$$

Logical Specifications: Query Examples

Is there a line of green pixels?

$$\exists x \forall y[(R^*_{\rightarrow}(x,y) \vee R^*_{\rightarrow}(y,x)) \Rightarrow P_{\blacksquare}(y)]?1 : 0$$

How many lines of green pixels are there?

What is the size of the picture?

What is the average lightness?

What is the size of the biggest monochromatic rectangle?

Logical Specifications: Query Examples

Is there a line of green pixels?

$$\exists x \forall y [(R^*_{\rightarrow}(x, y) \vee R^*_{\rightarrow}(y, x)) \Rightarrow P_{\blacksquare}(y)] ? 1 : 0$$

How many lines of green pixels are there?

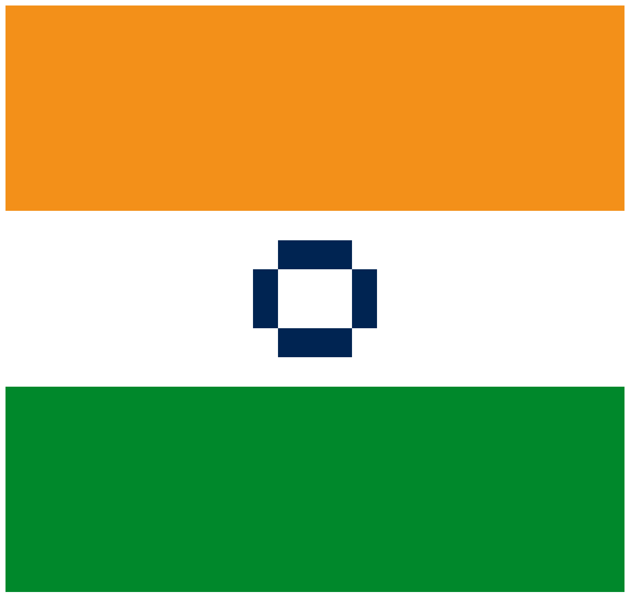
$$\sum_{x} \neg \exists z \, R_{\rightarrow}(z, x) \wedge \forall y \, R^*_{\rightarrow}(x, y) \Rightarrow P_{\blacksquare}(y) ? 1 : 0$$

What is the size of the picture?

What is the average lightness?

What is the size of the biggest monochromatic rectangle?

Logical Specifications: Query Examples



Is there a line of green pixels?

$$\exists x \forall y [(R^*_\rightarrow(x,y) \vee R^*_\rightarrow(y,x)) \Rightarrow P_{\blacksquare}(y)]\,?1 : 0$$

How many lines of green pixels are there?

$$\sum_x \neg \exists z\, R_\rightarrow(z,x) \wedge \forall y\, R^*_\rightarrow(x,y) \Rightarrow P_{\blacksquare}(y)\,?1 : 0$$

What is the size of the picture?

$$\left(\sum_x \neg \exists y R_\rightarrow(y,x)\,?1 : 0 \right) \times \left(\sum_x \neg \exists y R_\downarrow(y,x)\,?1 : 0 \right)$$

What is the average lightness?

What is the size of the biggest monochromatic rectangle?

Logical Specifications: Query Examples

$(\mathbb{N}, +, \times, 0, 1)$

Is there a line of green pixels?

$$\exists x \forall y [(R^*_{\rightarrow}(x,y) \vee R^*_{\rightarrow}(y,x)) \Rightarrow P_{\blacksquare}(y)] ? 1 : 0$$

How many lines of green pixels are there?

$$\sum_x \neg \exists z\, R_{\rightarrow}(z,x) \wedge \forall y\, R^*_{\rightarrow}(x,y) \Rightarrow P_{\blacksquare}(y) ? 1 : 0$$

What is the size of the picture?

$$\left(\sum_x \neg \exists y R_{\rightarrow}(y,x) ? 1 : 0 \right) \times \left(\sum_x \neg \exists y R_{\downarrow}(y,x) ? 1 : 0 \right)$$

What is the average lightness?

What is the size of the biggest monochromatic rectangle?

Logical Specifications: Query Examples

$(\mathbb{N}, +, \times, 0, 1)$

Is there a line of green pixels?

$$\exists x \forall y [(R^*_{\rightarrow}(x,y) \vee R^*_{\rightarrow}(y,x)) \Rightarrow P_{\color{green}\blacksquare}(y)] ?1 : 0$$

How many lines of green pixels are there?

$$\sum_x \neg \exists z\, R_{\rightarrow}(z,x) \wedge \forall y\, R^*_{\rightarrow}(x,y) \Rightarrow P_{\color{green}\blacksquare}(y) ?1 : 0$$

What is the size of the picture?

$$\left(\sum_x \neg \exists y R_{\rightarrow}(y,x) ?1 : 0 \right) \times \left(\sum_x \neg \exists y R_{\downarrow}(y,x) ?1 : 0 \right)$$

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

What is the average lightness?

$$\sum_x P_{\color{orange}\blacksquare}(x) ?100 : 0 + \sum_x P_{\color{green}\blacksquare}(x) ?150 : 0 + \sum_x P_{\color{navy}\blacksquare}(x) ?220 : 0$$

What is the size of the biggest monochromatic rectangle?

Logical Specifications: Query Examples

$(\mathbb{N}, +, \times, 0, 1)$

Is there a line of green pixels?

$$\exists x \forall y [(R^*_{\rightarrow}(x,y) \vee R^*_{\rightarrow}(y,x)) \Rightarrow P_{\color{green}\blacksquare}(y)] ? 1 : 0$$

How many lines of green pixels are there?

$$\color{red}{\sum_x} \color{black} \neg \exists z\, R_{\rightarrow}(z,x) \wedge \forall y\, R^*_{\rightarrow}(x,y) \Rightarrow P_{\color{green}\blacksquare}(y) ? 1 : 0$$

What is the size of the picture?

$$\left(\sum_x \neg \exists y R_{\rightarrow}(y,x) ? 1 : 0 \right) \color{red}{\times} \color{black} \left(\sum_x \neg \exists y R_{\downarrow}(y,x) ? 1 : 0 \right)$$

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

What is the average lightness?

$$\sum_x P_{\color{orange}\blacksquare}(x) ? 100 : 0 + \sum_x P_{\color{green}\blacksquare}(x) ? 150 : 0 + \sum_x P_{\color{navy}\blacksquare}(x) ? 220 : 0$$

What is the size of the biggest monochromatic rectangle?

$$\max_{x,y} \left[\varphi_{mono}(x,y) ? 1 : 0 + \left(\sum_z \varphi_{rect}(x,y,z) ? 1 : 0 \right) \right]$$

Logical Specifications: Query Examples

$(\mathbb{N}, +, \times, 0, 1)$

Is there a line of green pixels?

$$\exists x \forall y [(R^*_{\rightarrow}(x,y) \vee R^*_{\rightarrow}(y,x)) \Rightarrow P_{\blacksquare}(y)]\,?1:0$$

How many lines of green pixels are there?

$$\sum_x \neg \exists z\, R_{\rightarrow}(z,x) \wedge \forall y\, R^*_{\rightarrow}(x,y) \Rightarrow P_{\blacksquare}(y)\,?1:0$$

What is the size of the picture?

$$\left(\sum_x \neg \exists y R_{\rightarrow}(y,x)\,?1:0\right) \times \left(\sum_x \neg \exists y R_{\downarrow}(y,x)\,?1:0\right)$$

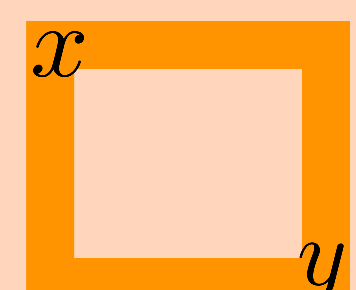
$(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$

What is the average lightness?

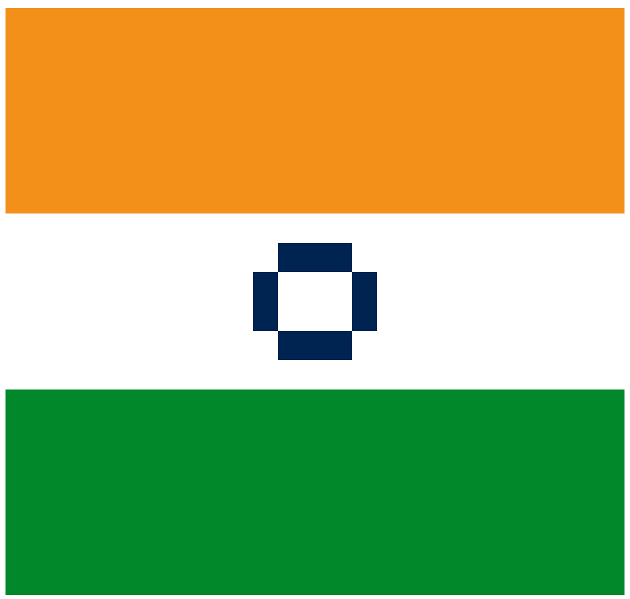
$$\sum_x P_{\blacksquare}(x)\,?100:0 + \sum_x P_{\blacksquare}(x)\,?150:0 + \sum_x P_{\blacksquare}(x)\,?220:0$$

What is the size of the biggest monochromatic rectangle?

$$\max_{x,y}\left[\varphi_{mono}(x,y)\,?1:0 + \left(\sum_z \varphi_{rect}(x,y,z)\,?1:0\right)\right]$$



Logical Specifications: Query Examples



$(\mathbb{N}, +, \times, 0, 1)$

Is there a line of green pixels?

$$\exists x \forall y [(R^*_\rightarrow(x,y) \lor R^*_\rightarrow(y,x)) \Rightarrow P_{\blacksquare}(y)] ?1 : 0$$

How many lines of green pixels are there?

$$\sum_x \neg \exists z\, R_\rightarrow(z,x) \land \forall y\, R^*_\rightarrow(x,y) \Rightarrow P_{\blacksquare}(y) ?1 : 0$$

What is the size of the picture?

$$\left(\sum_x \neg \exists y R_\rightarrow(y,x) ?1 : 0\right) \times \left(\sum_x \neg \exists y R_\downarrow(y,x) ?1 : 0\right)$$

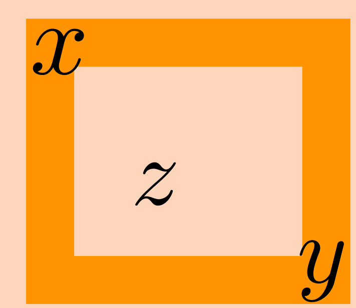
$(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$

What is the average lightness?

$$\sum_x P_{\blacksquare}(x) ?100 : 0 + \sum_x P_{\blacksquare}(x) ?150 : 0 + \sum_x P_{\blacksquare}(x) ?220 : 0$$

What is the size of the biggest monochromatic rectangle?

$$\max_{x,y} \left[\varphi_{mono}(x,y) ?1 : 0 + \left(\sum_z \varphi_{rect}(x,y,z) ?1 : 0 \right) \right]$$



Weighted FO logic

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

$$\Psi ::= s \mid \varphi\, ?\, \Psi : \Psi$$

$$\Phi ::= \mathbf{0} \mid \varphi\, ?\, \Phi : \Phi \mid \Phi + \Phi \mid \textstyle\sum_x \Phi \mid \textstyle\sum_X \Phi \mid \textstyle\prod_x \Psi$$

Weighted FO logic

We can keep Boolean MSO or restrict to FO…

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

$$\Phi ::= s \,\big|\, \varphi ? \Phi : \Phi \,\big|\, \Phi + \Phi \,\big|\, \Phi \times \Phi \,\big|\, \sum_x \Phi \,\big|\, \prod_x \Phi$$

Reintroduction of the product

Unconditional product quantification

Weighted FO logic

We can keep Boolean MSO or restrict to FO…

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

$$\Phi ::= s \mid \varphi ? \Phi : \Phi \mid \Phi + \Phi \mid \Phi \times \Phi \mid \sum_x \Phi \mid \prod_x \Phi$$

Reintroduction of the product

Unconditional product quantification

$$\varphi_2 = \forall x\, \exists y\, (y \leqslant x \wedge P_a(y)) \qquad [\![\varphi_2]\!](a^n) = n!$$

Weighted FO logic

We can keep Boolean MSO or restrict to FO…

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi \mid \forall X\, \varphi$$

$$\Phi ::= s \mid \varphi ? \Phi : \Phi \mid \Phi + \Phi \mid \Phi \times \Phi \mid \sum\nolimits_x \Phi \mid \prod\nolimits_x \Phi$$

Reintroduction of the product

Unconditional product quantification

$$[\![\prod\nolimits_x \prod\nolimits_y 2]\!](w) = 2^{|w|^2}$$

Pebble weighted automata

$I \in \mathbb{S}^Q$ $T \in \mathbb{S}^Q$

$$\mathcal{A} = (Q, A, I, \delta, T)$$

$$\delta : Q \times \mathrm{Test} \times \mathrm{Move} \times Q \to \mathbb{S}$$

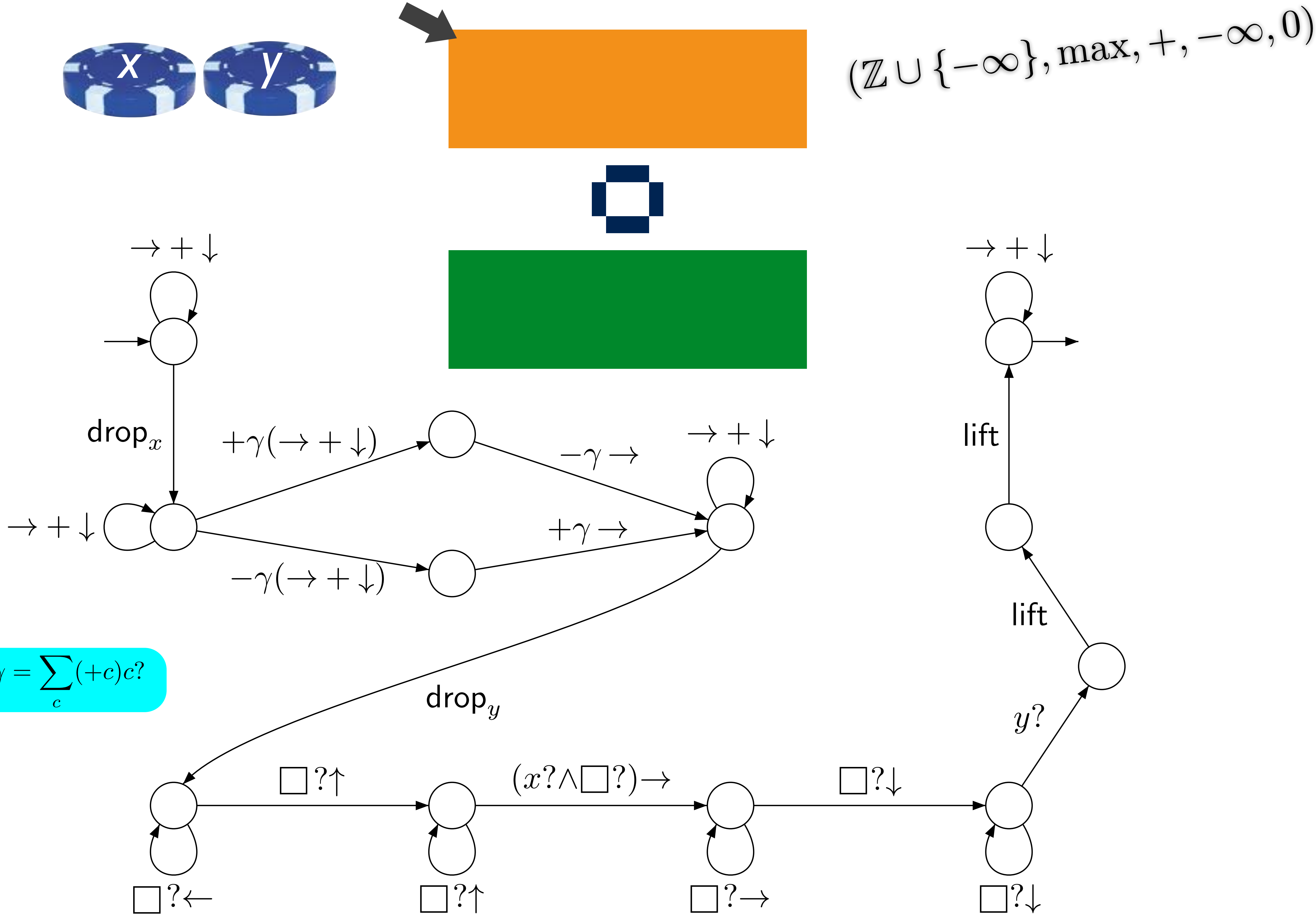
$$\mathrm{Move} = D \cup \{\mathrm{drop}_x, \mathrm{lift} \mid x \in \mathrm{Peb}\}$$

Run as a finite sequence of configurations (W, σ, q, i, π)

with free pebbles $\sigma : \mathrm{Peb} \to \mathrm{pos}(W)$

and a stack of currently dropped pebbles $\pi \in (\mathrm{Peb} \times \mathrm{pos}(W))^*$

Pebble Weighted Automata: An Example

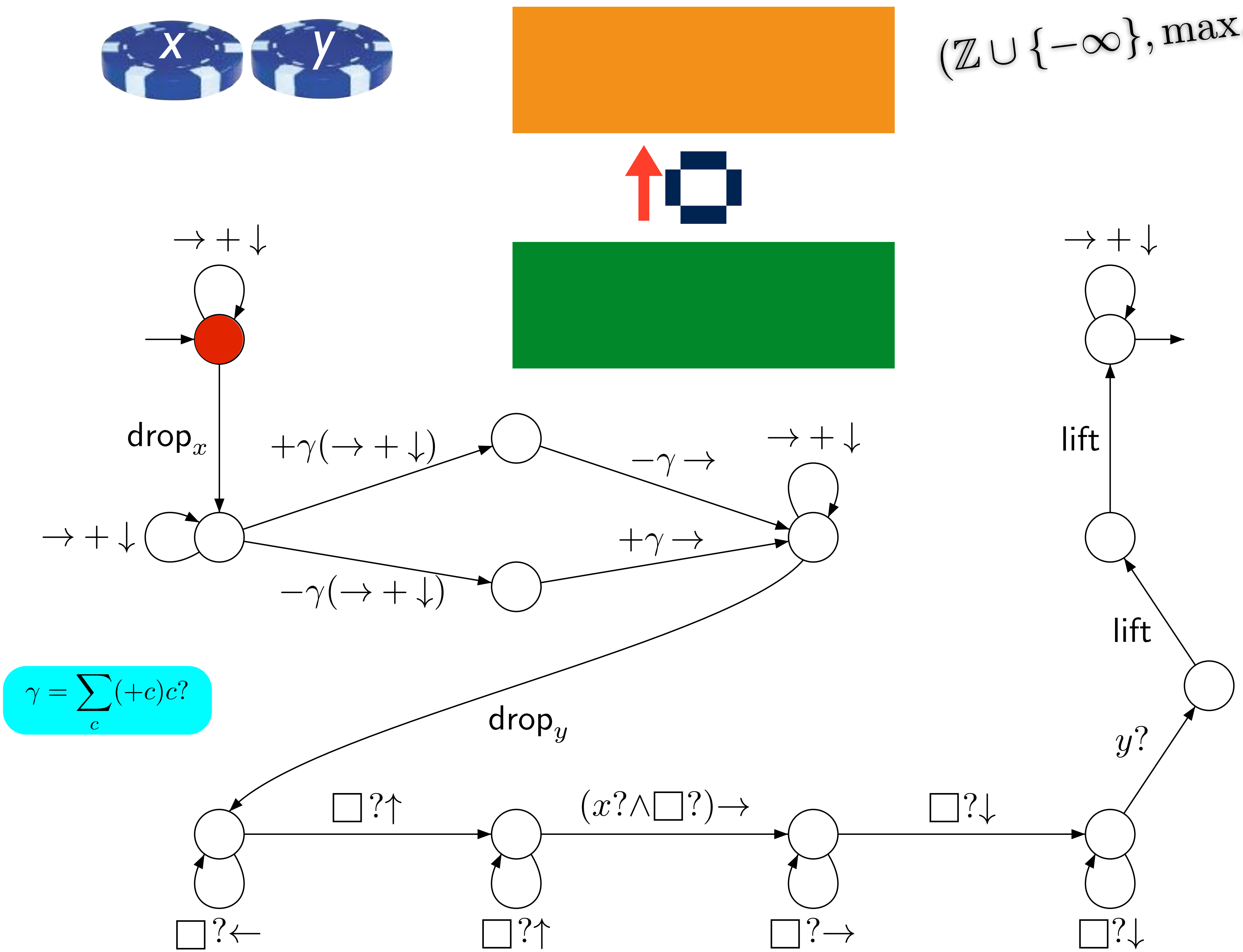


$\gamma = \sum_c (+c)c?$

Pebble automata over words and trees: [Globerman&Harel 96], [Engelfriet&Hoogeboom 99]

Pebble Weighted Automata: An Example

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

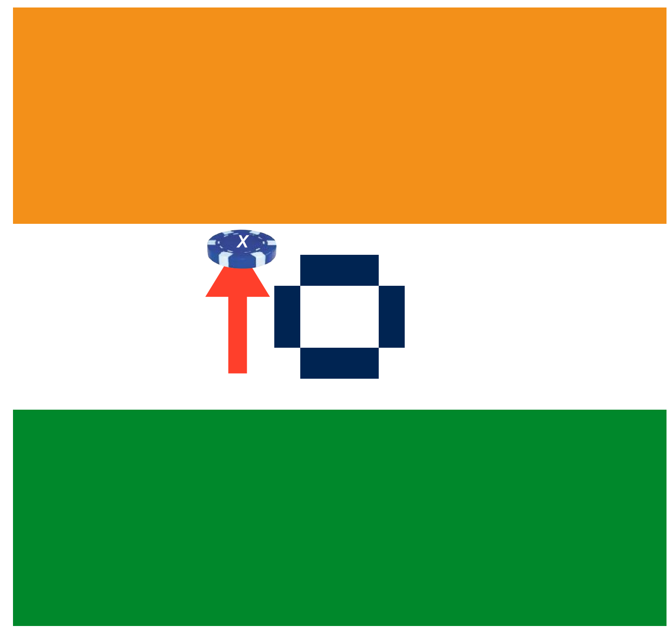


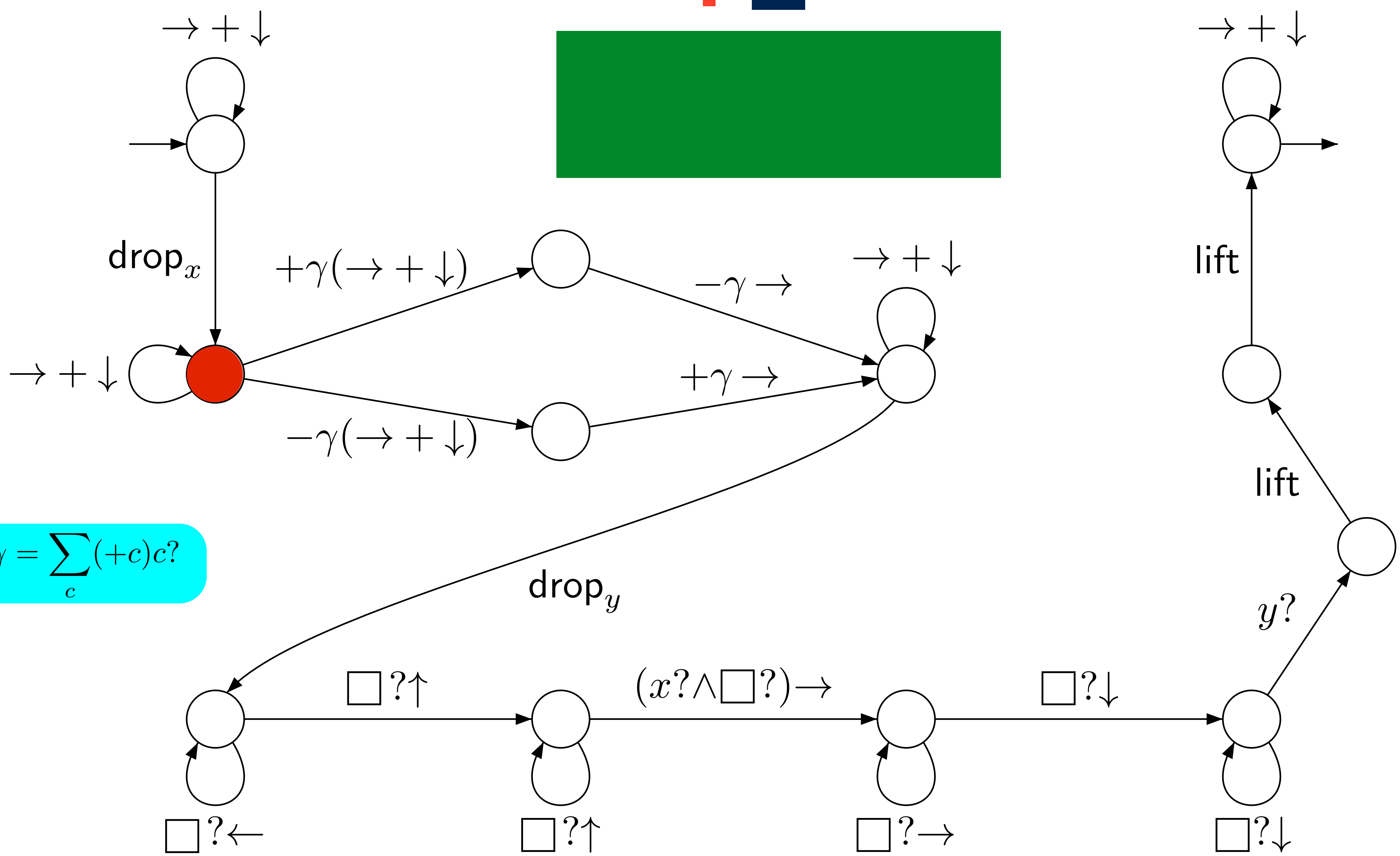
©GasTeX

Pebble automata over words and trees: [Globerman&Harel 96], [Engelfriet&Hoogeboom 99]

Pebble Weighted Automata: An Example

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



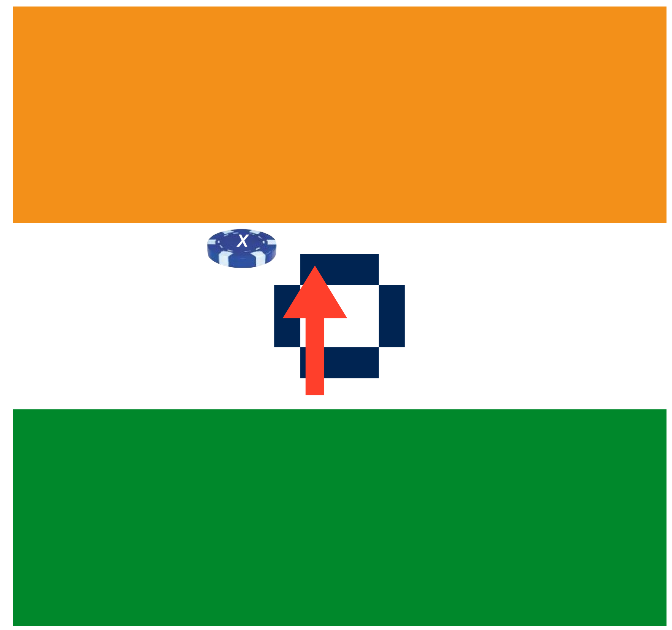


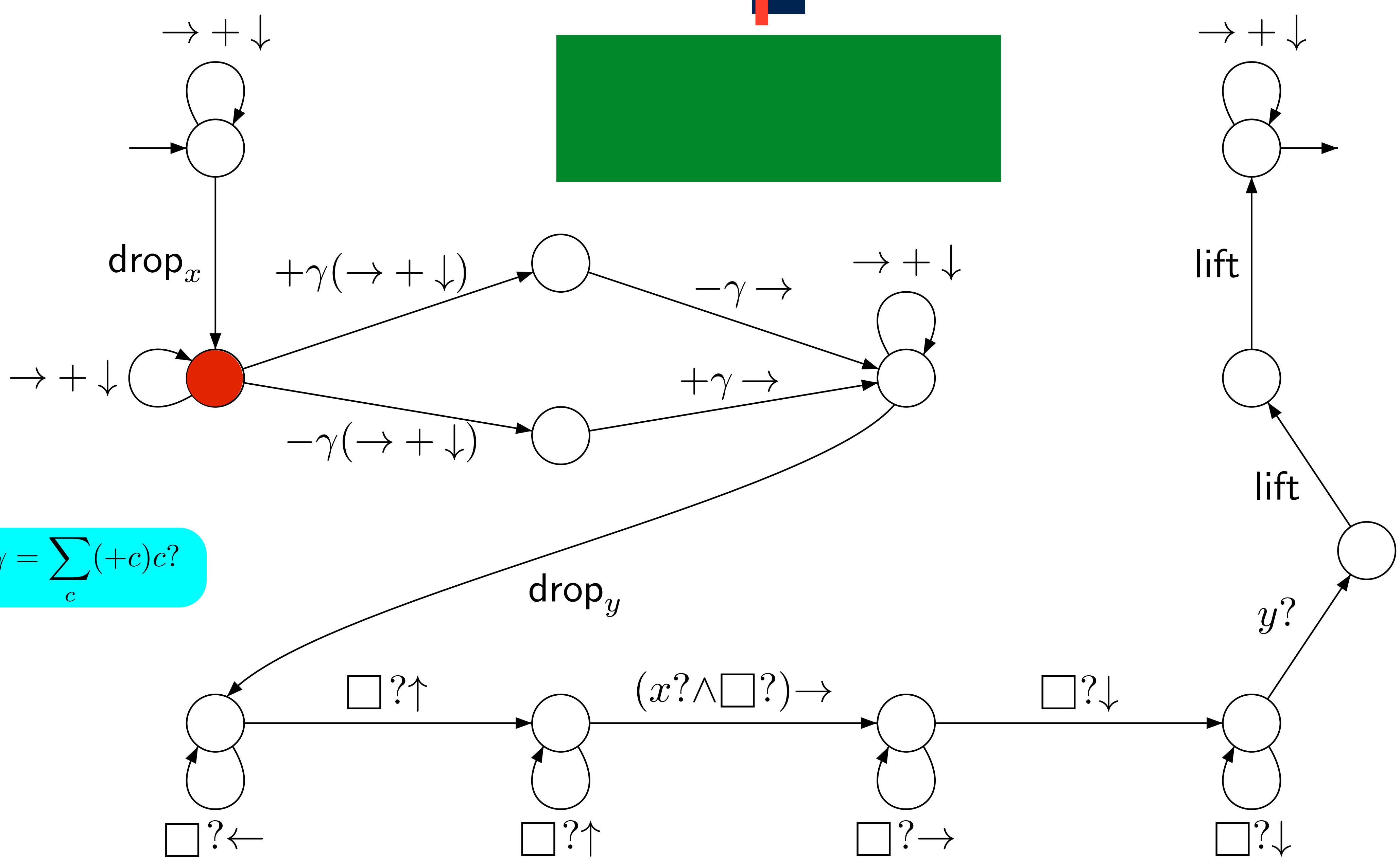
$\gamma = \sum_c (+c)c?$

Pebble automata over words and trees: [Globerman&Harel 96], [Engelfriet&Hoogeboom 99]

Pebble Weighted Automata: An Example

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



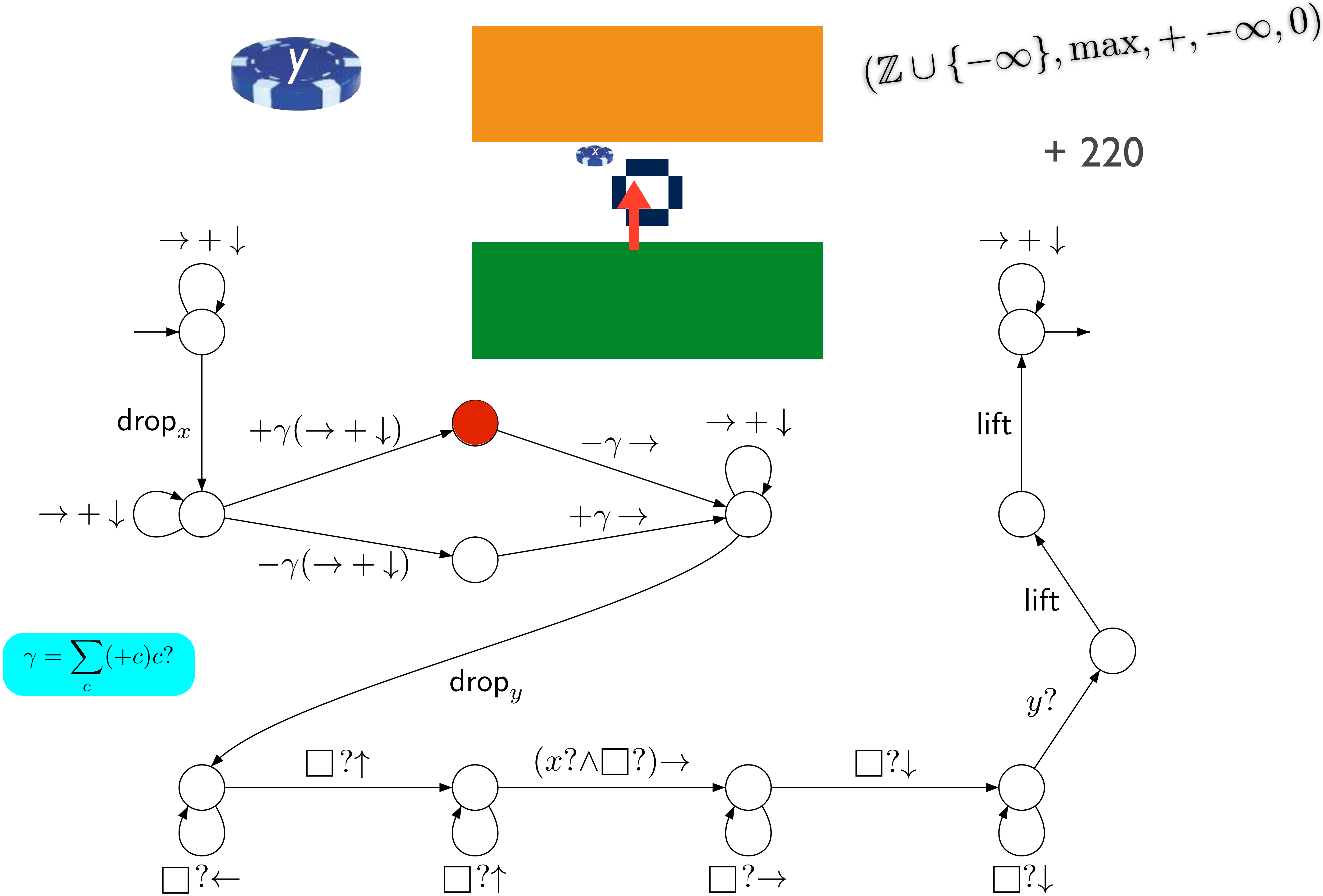


$\gamma = \sum_c (+c)c?$

©GasTeX

Pebble automata over words and trees: [Globerman&Harel 96], [Engelfriet&Hoogeboom 99]

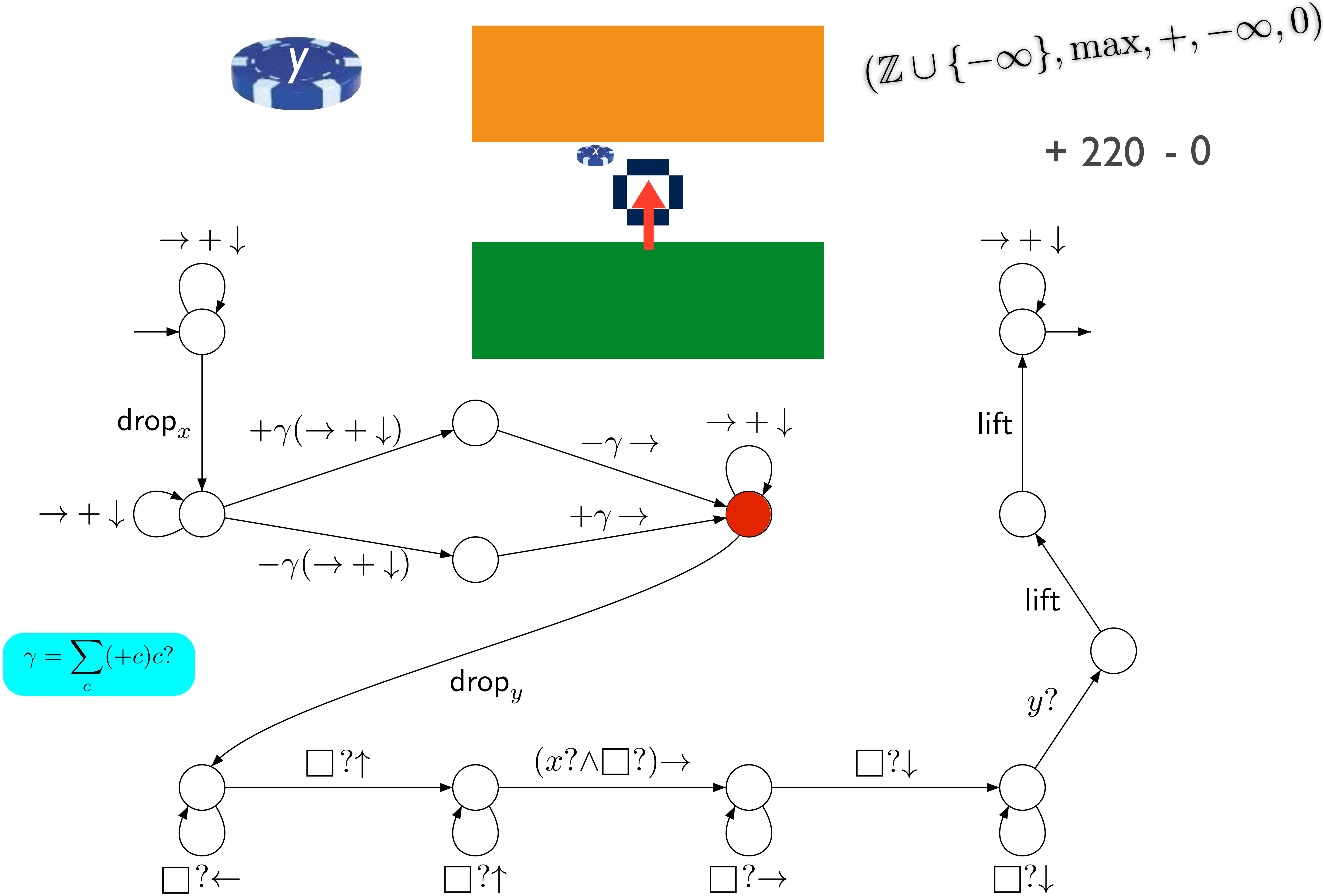
Pebble Weighted Automata: An Example



©GasTeX

Pebble automata over words and trees: [Globerman&Harel 96], [Engelfriet&Hoogeboom 99]

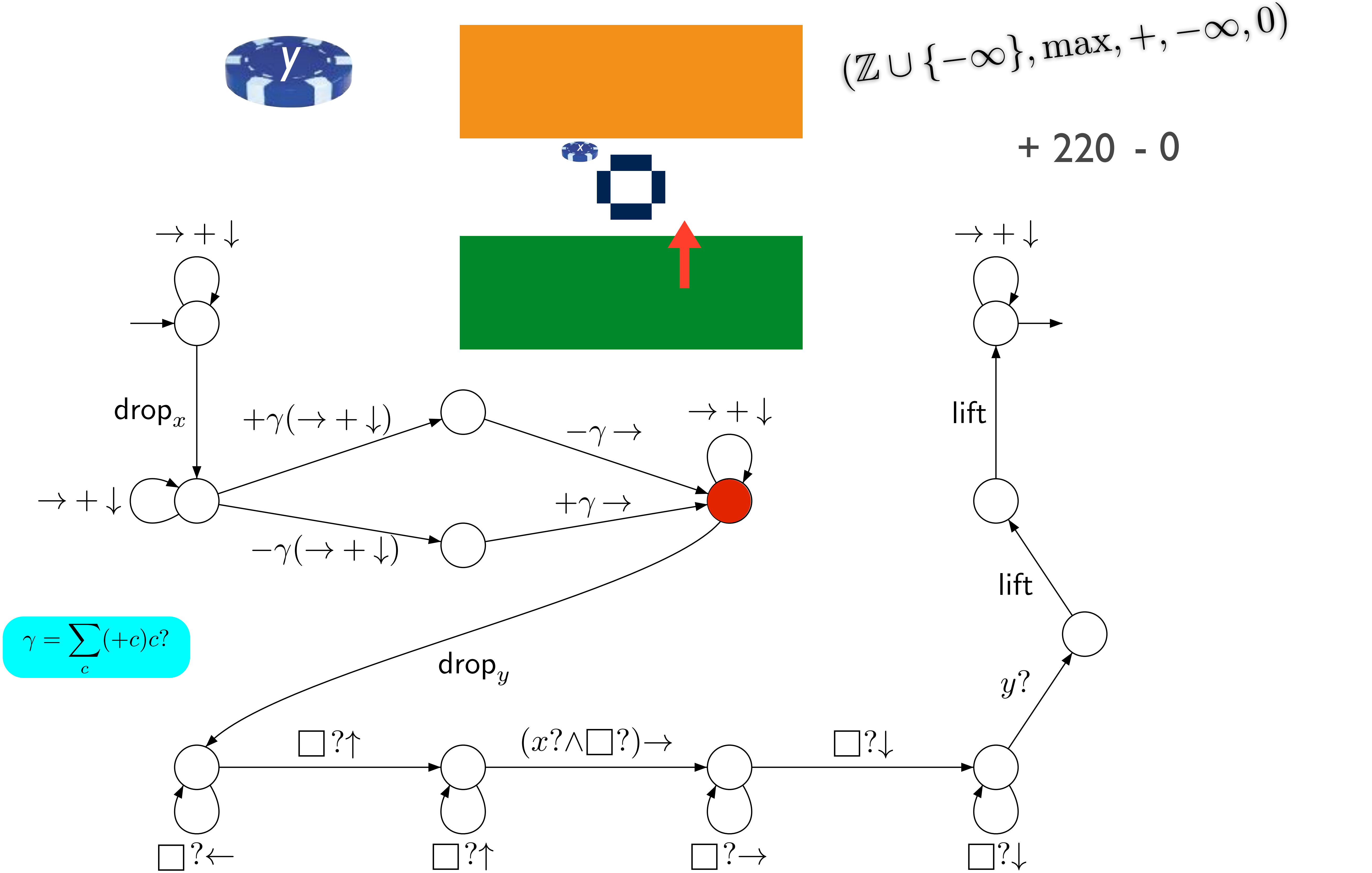
Pebble Weighted Automata: An Example



©GasTeX

Pebble automata over words and trees: [Globerman&Harel 96], [Engelfriet&Hoogeboom 99]

Pebble Weighted Automata: An Example

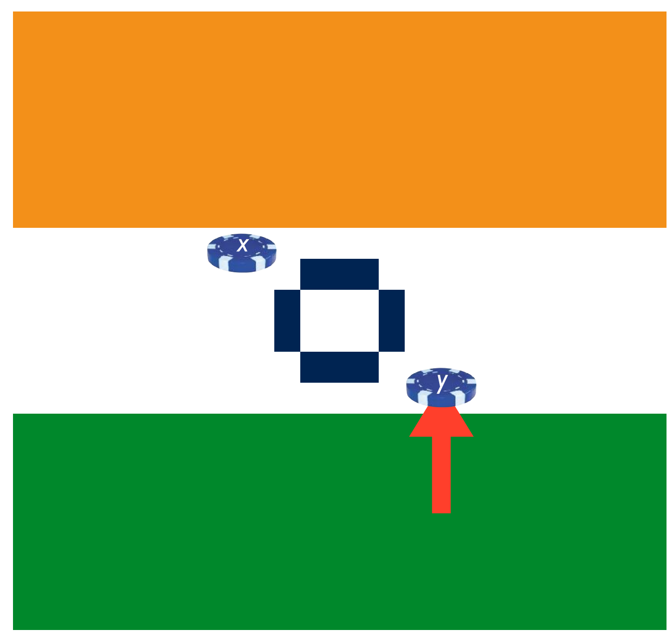


©GasTeX

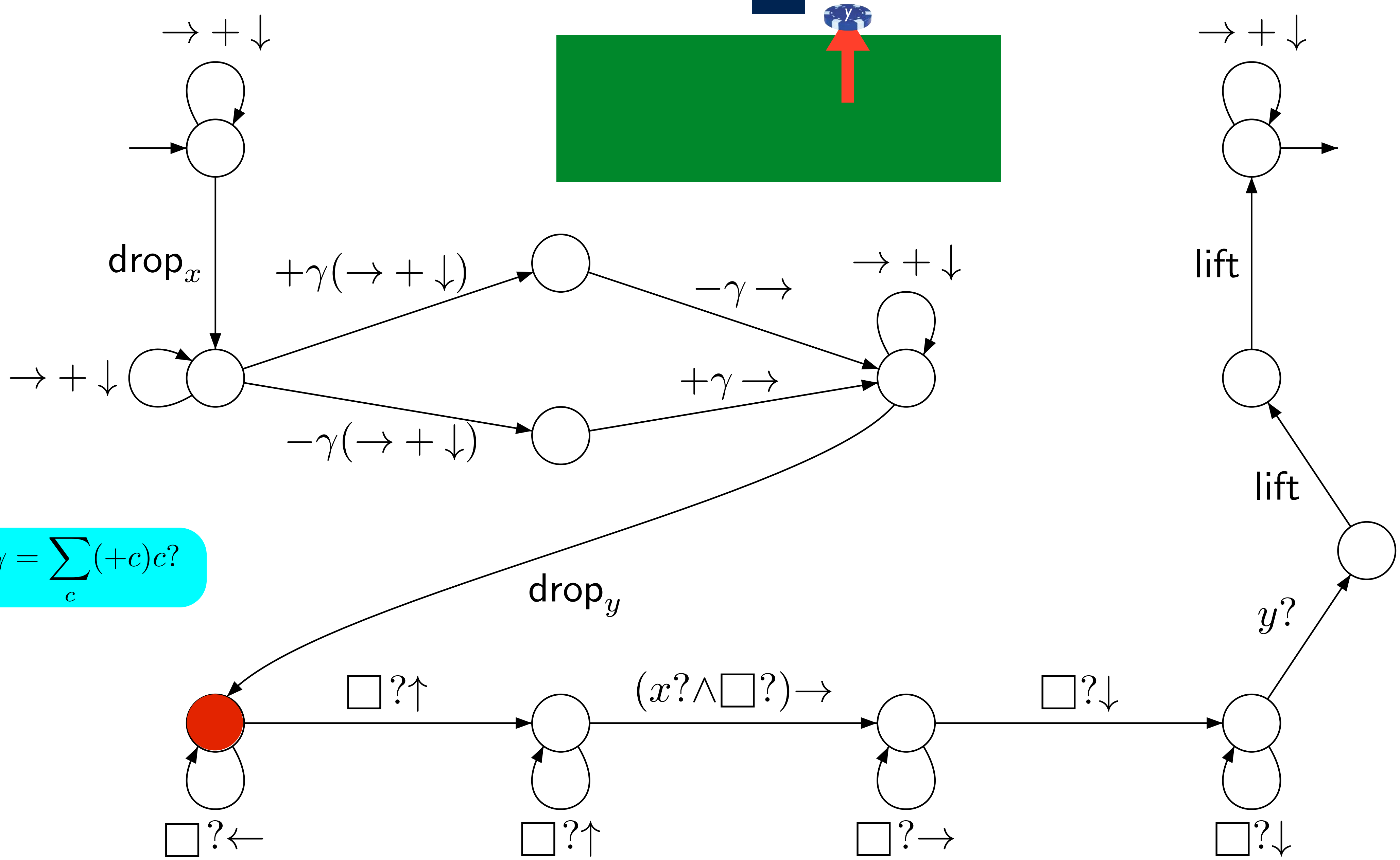
Pebble automata over words and trees: [Globerman&Harel 96], [Engelfriet&Hoogeboom 99]

Pebble Weighted Automata: An Example

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



+ 220 - 0



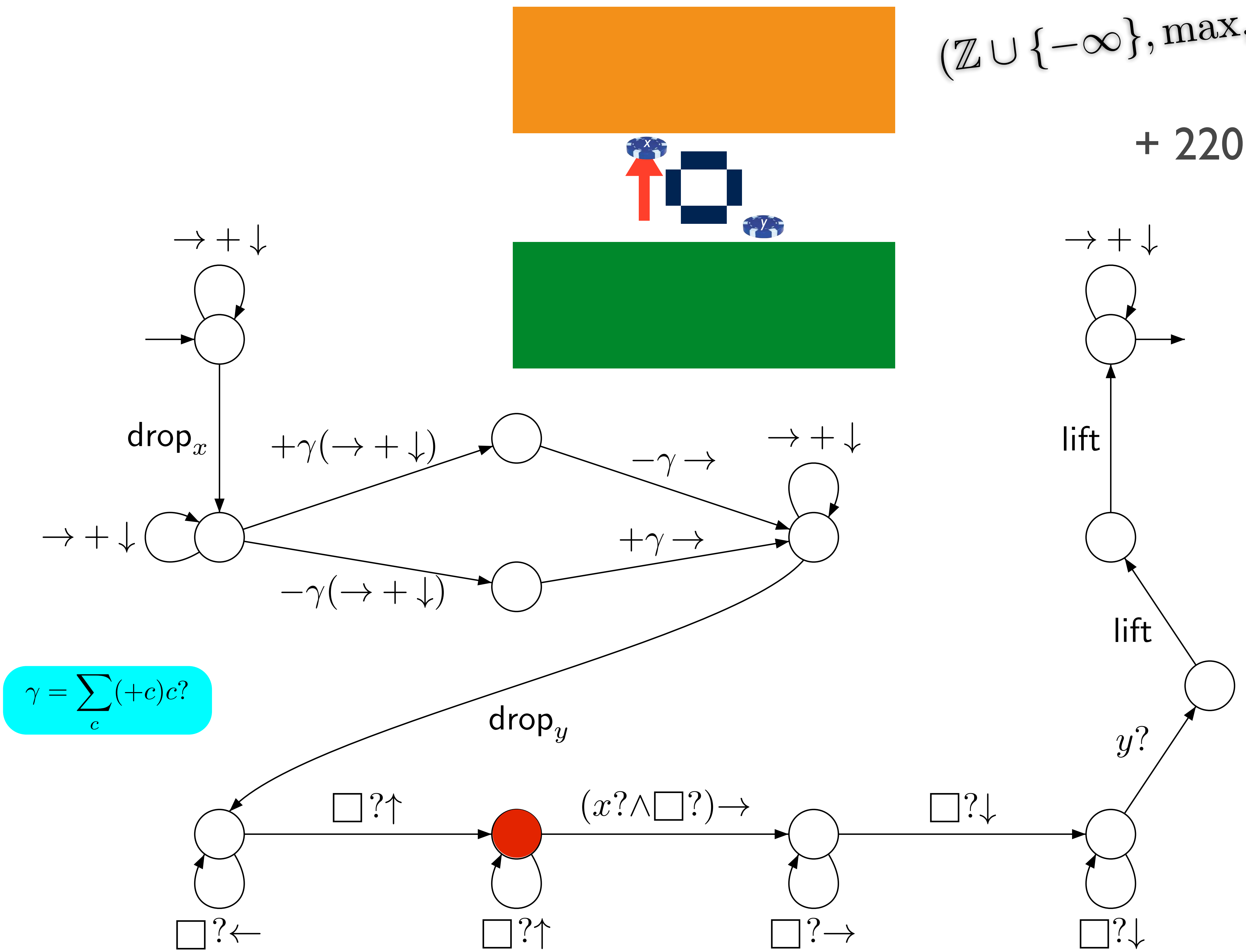
$\gamma = \sum_c (+c)c?$

Pebble automata over words and trees: [Globerman&Harel 96], [Engelfriet&Hoogeboom 99]

Pebble Weighted Automata: An Example

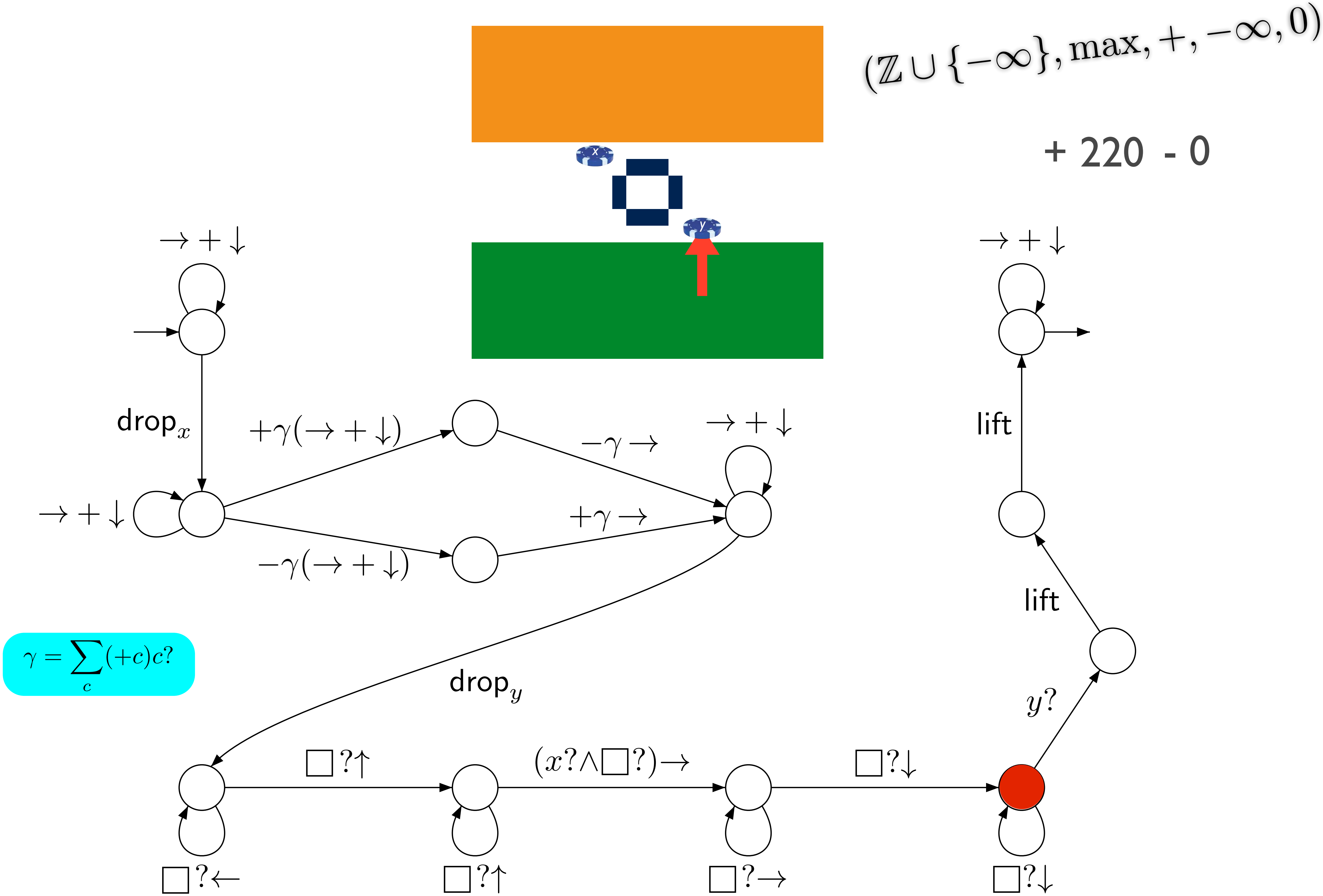
$(\mathbb{Z} \cup \{-\infty\}, \max, +, -\infty, 0)$

+ 220 - 0



Pebble automata over words and trees: [Globerman&Harel 96], [Engelfriet&Hoogeboom 99]

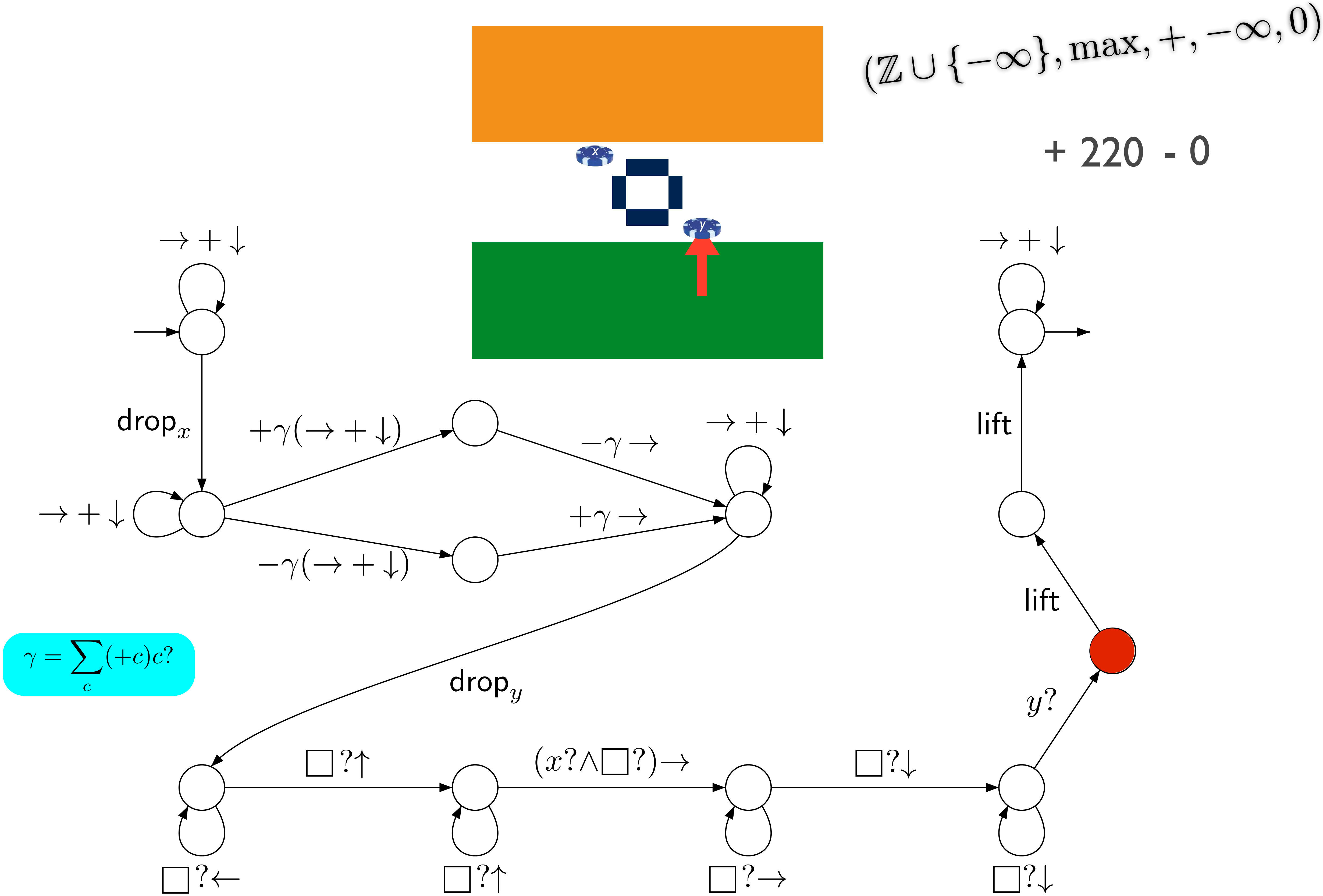
Pebble Weighted Automata: An Example



©GasTeX

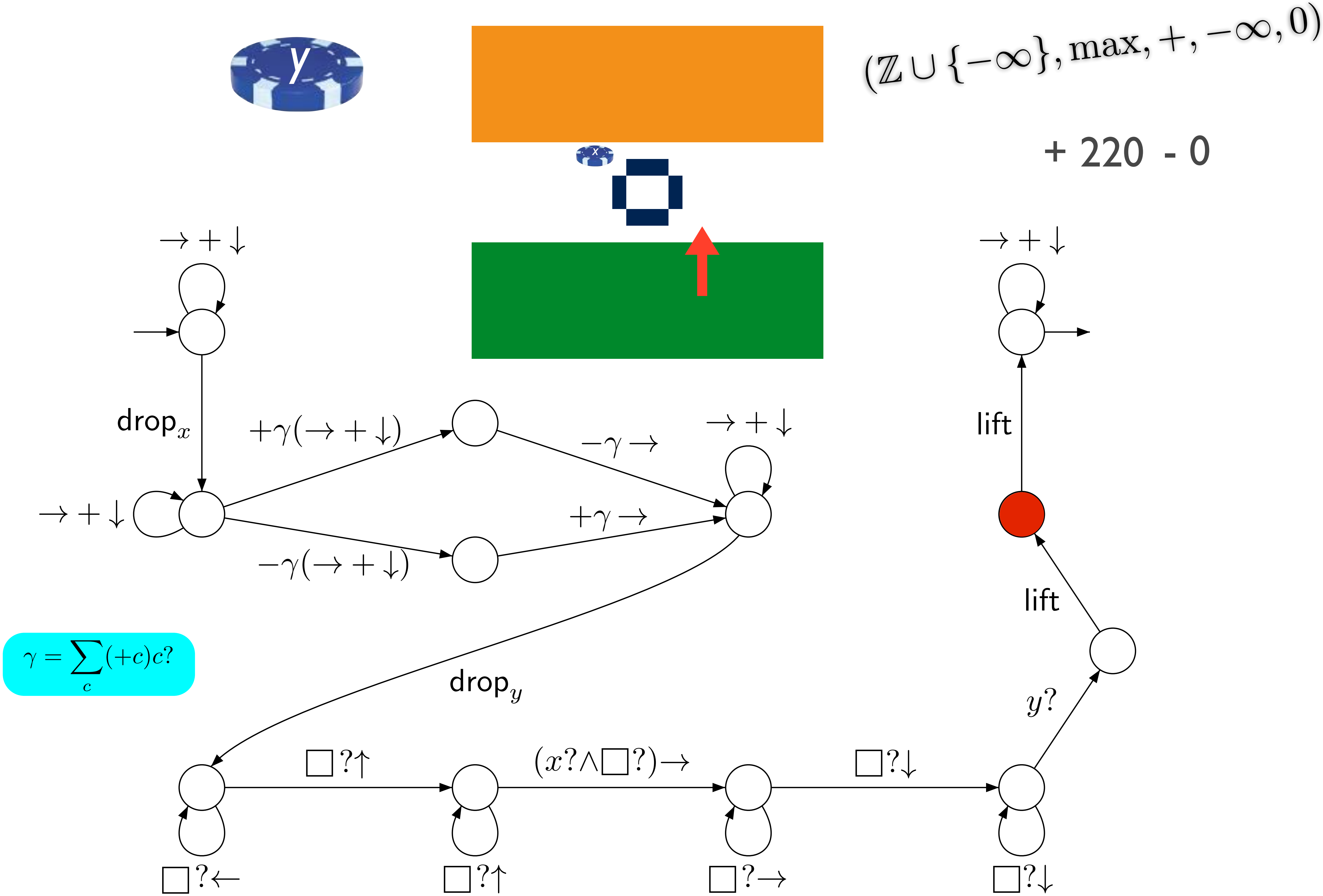
Pebble automata over words and trees: [Globerman&Harel 96], [Engelfriet&Hoogeboom 99]

Pebble Weighted Automata: An Example



Pebble automata over words and trees: [Globerman&Harel 96], [Engelfriet&Hoogeboom 99]

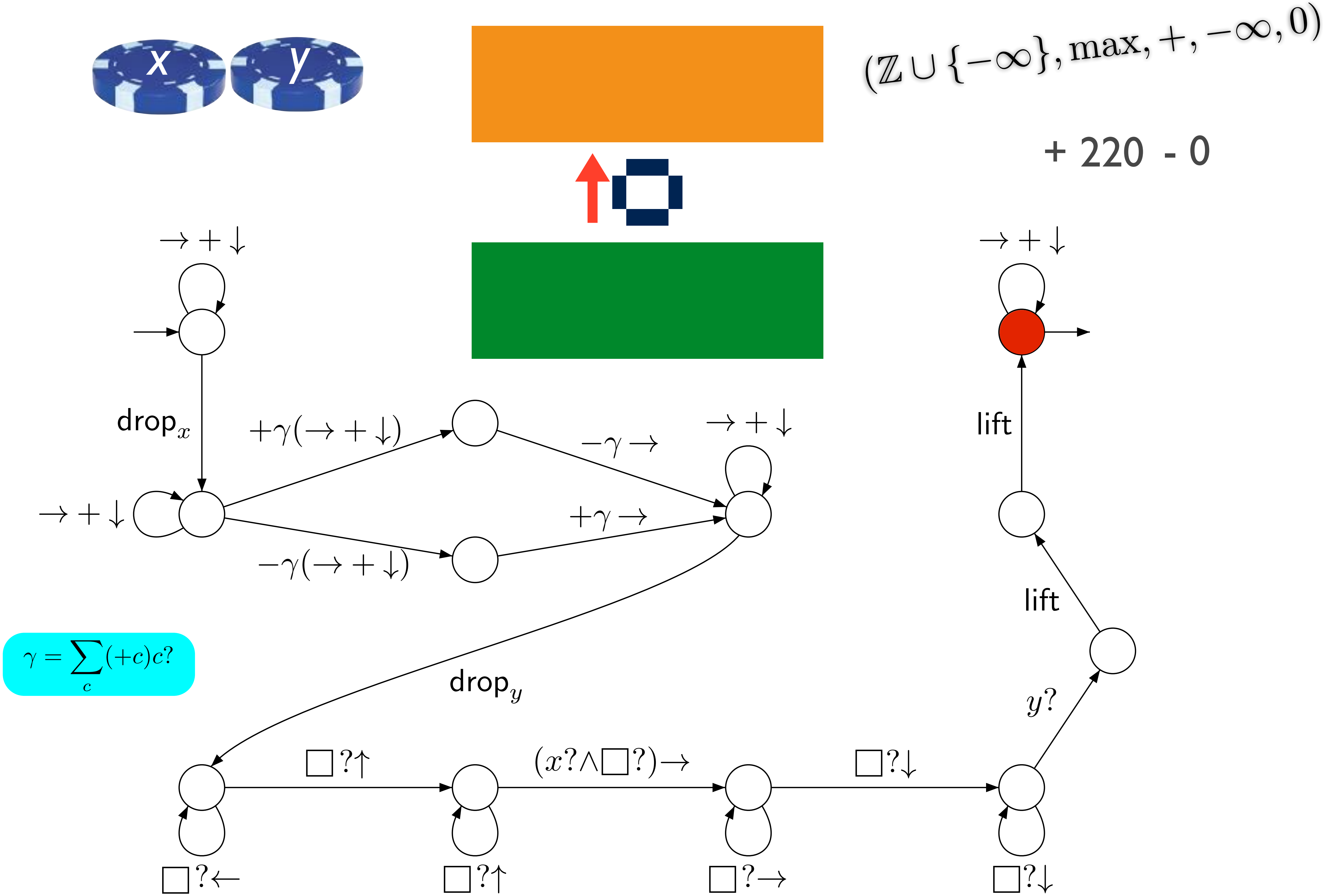
Pebble Weighted Automata: An Example



©GasTeX

Pebble automata over words and trees: [Globerman&Harel 96], [Engelfriet&Hoogeboom 99]

Pebble Weighted Automata: An Example

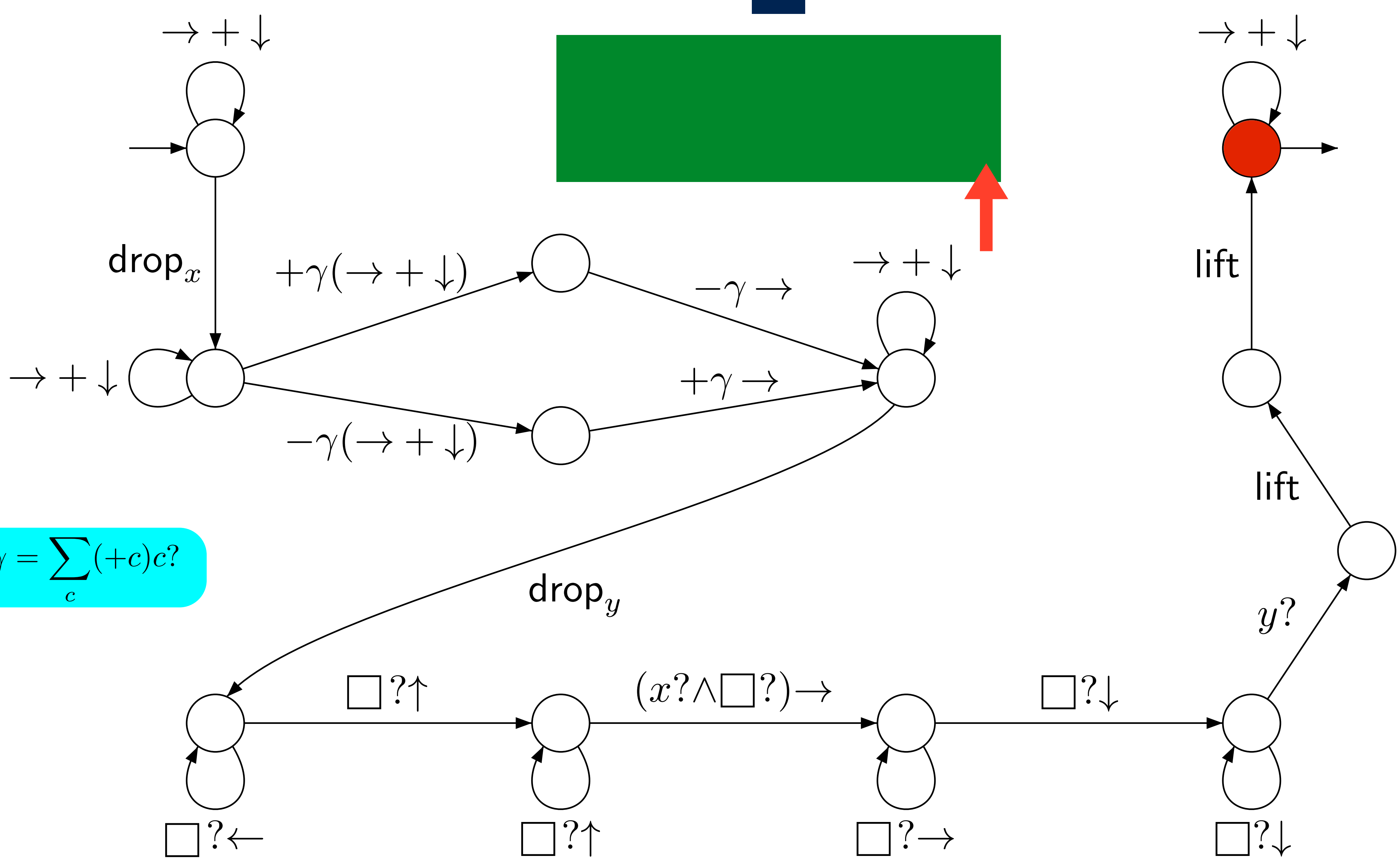


Pebble automata over words and trees: [Globerman&Harel 96], [Engelfriet&Hoogeboom 99]

Pebble Weighted Automata: An Example

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

+ 220 - 0



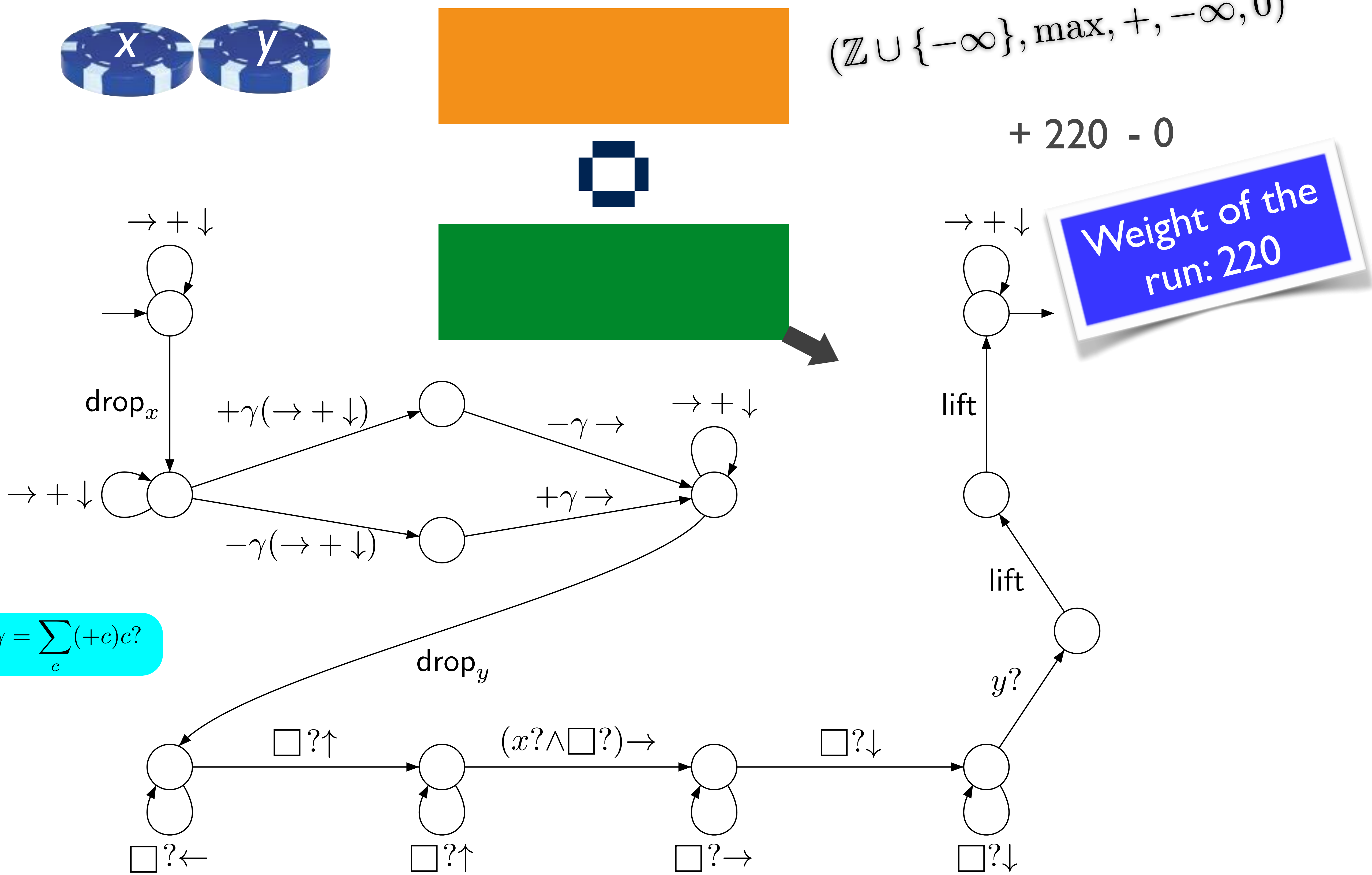
$\gamma = \sum_c (+c)c?$

©GasTeX

Pebble automata over words and trees: [Globerman&Harel 96], [Engelfriet&Hoogeboom 99]

Pebble Weighted Automata: An Example

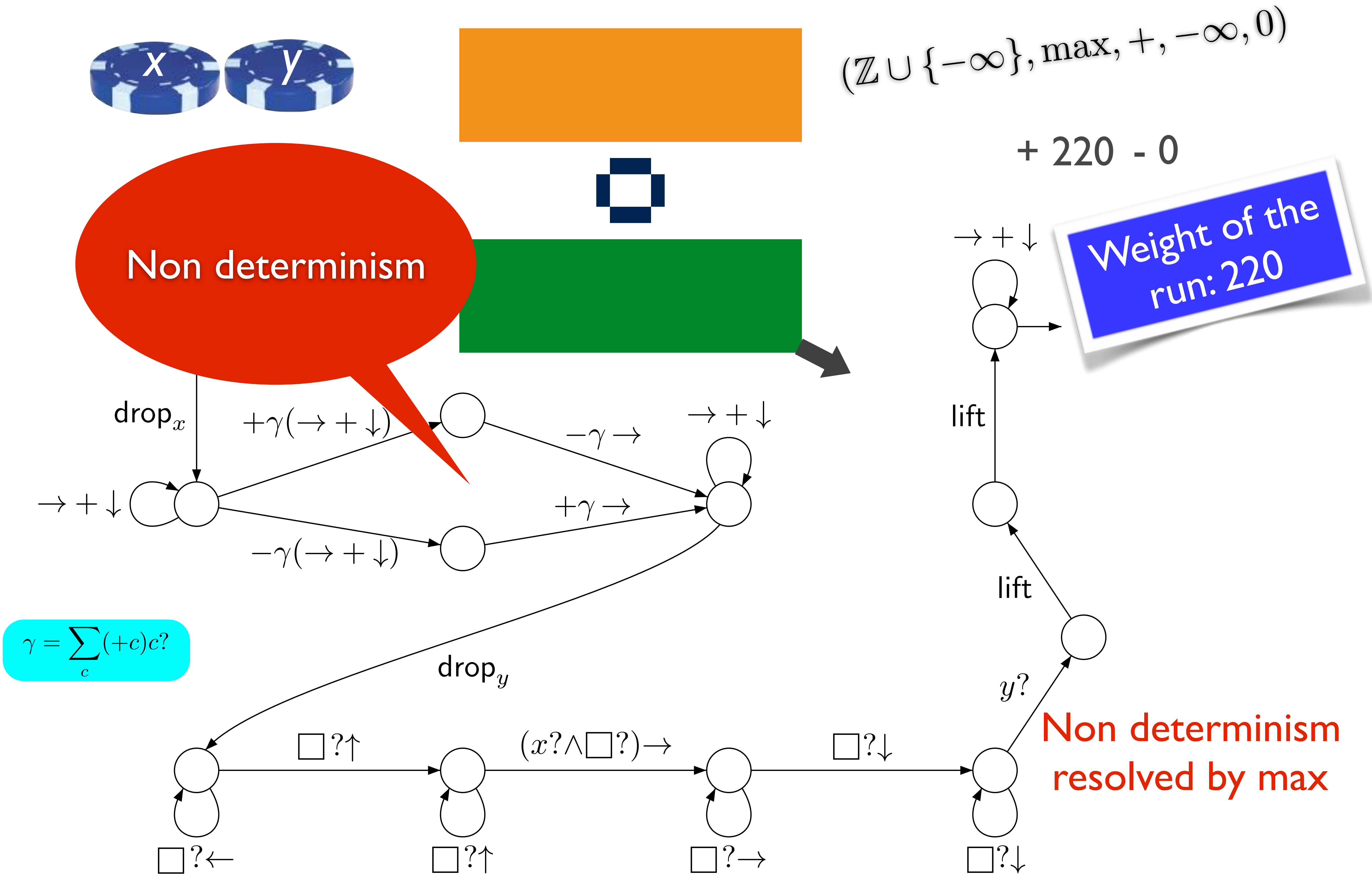
$$(\mathbb{Z} \cup \{-\infty\}, \max, +, -\infty, 0)$$



$$\gamma = \sum_c (+c)c?$$

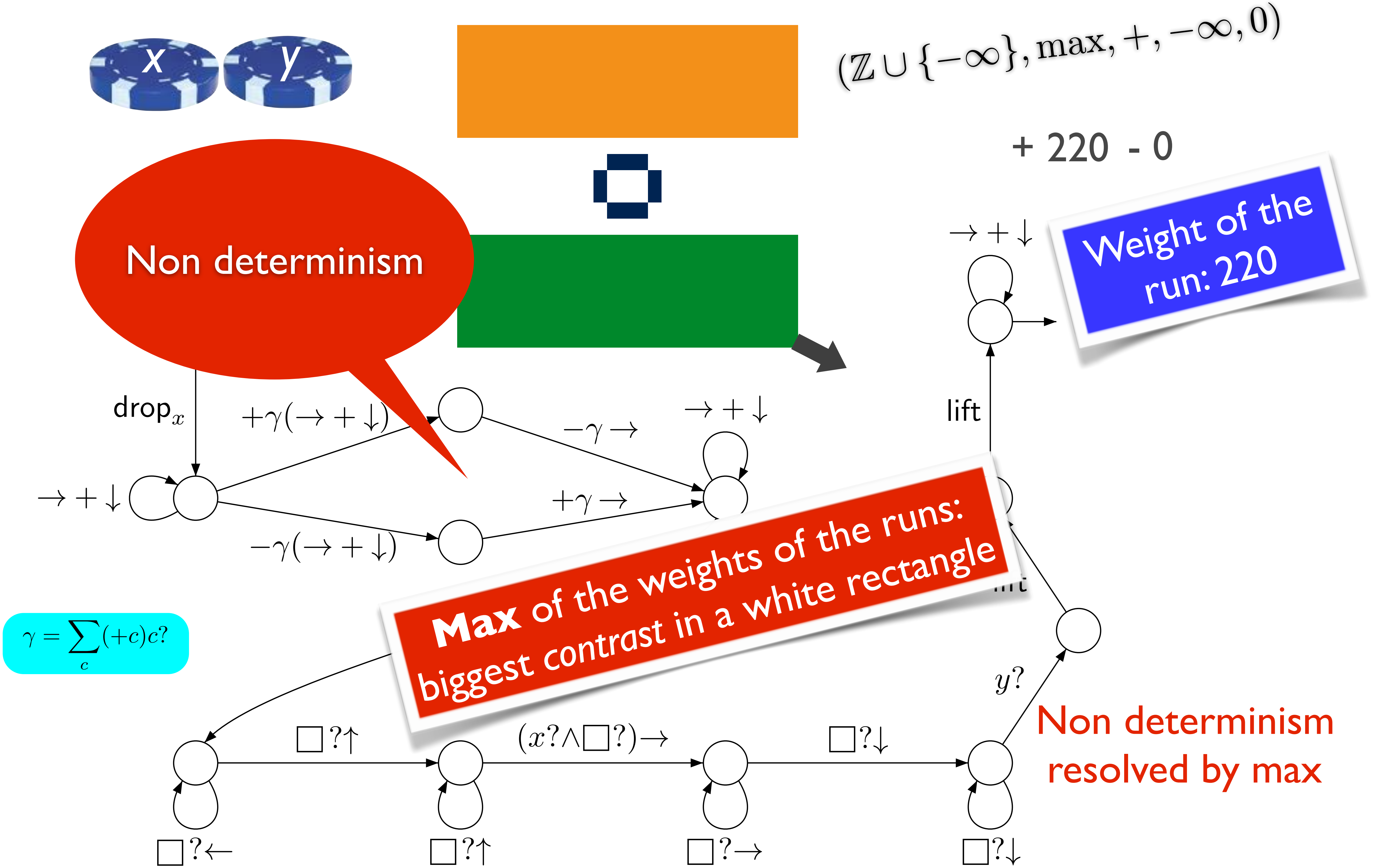
Pebble automata over words and trees: [Globerman&Harel 96], [Engelfriet&Hoogeboom 99]

Pebble Weighted Automata: An Example



Pebble automata over words and trees: [Globerman&Harel 96], [Engelfriet&Hoogeboom 99]

Pebble Weighted Automata: An Example



Pebble automata over words and trees: [Globerman&Harel 96], [Engelfriet&Hoogeboom 99]

Translation from Logics to Automata

Theorem: Consider a *searchable* class of graph. Every wFO formula can then be translated into a Pebble Weighted Automaton equivalent over this class of graphs.

WFO ⟶ PWA

Over words: [Bollig&Gastin&Monmege&Zeitoun 2010]
Over nested words: [Bollig&Gastin&Monmege&Zeitoun 2013]

Translation from Logics to Automata

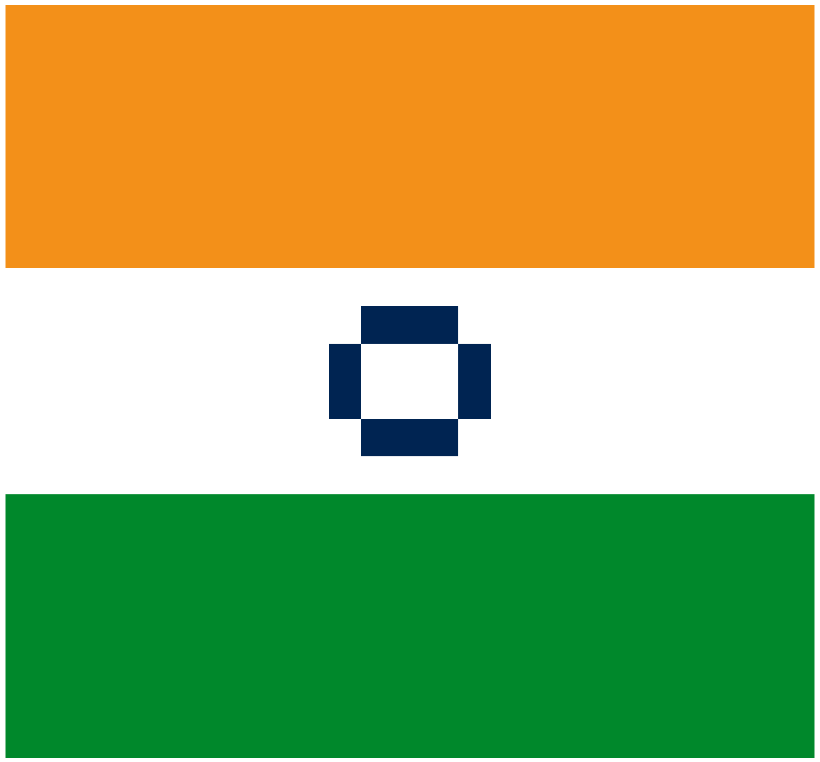
Theorem: Consider a *searchable* class of graph. Every wFO formula can then be translated into a Pebble Weighted Automaton equivalent over this class of graphs.

WFO ⟶ PWA

Which complexity?


Over words: [Bollig&Gastin&Monmege&Zeitoun 2010]
Over nested words: [Bollig&Gastin&Monmege&Zeitoun 2013]

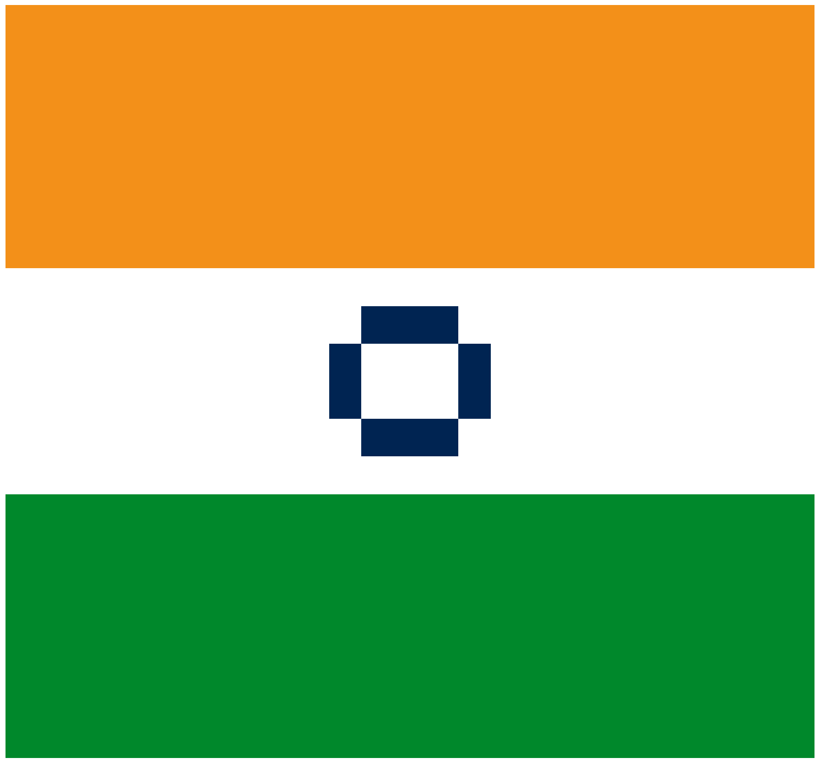
Translation from Logics to Automata



$$\sum_{x} P_{\blacksquare}(x)$$

$(\mathbb{N} \cup \{+\infty\}, +, \times, 0, 1)$

Translation from Logics to Automata



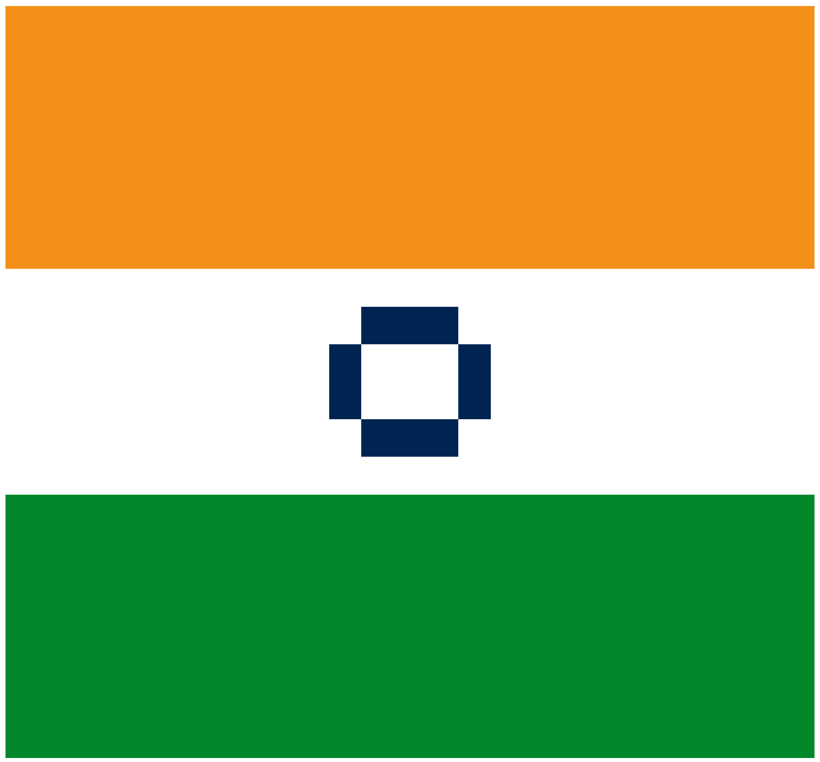
$(\mathbb{N} \cup \{+\infty\}, +, \times, 0, 1)$

$$\sum_x P_{\blacksquare}(x)$$

use *non-determinism* to count

- a run per position
- each run has the value of the subformula

Translation from Logics to Automata

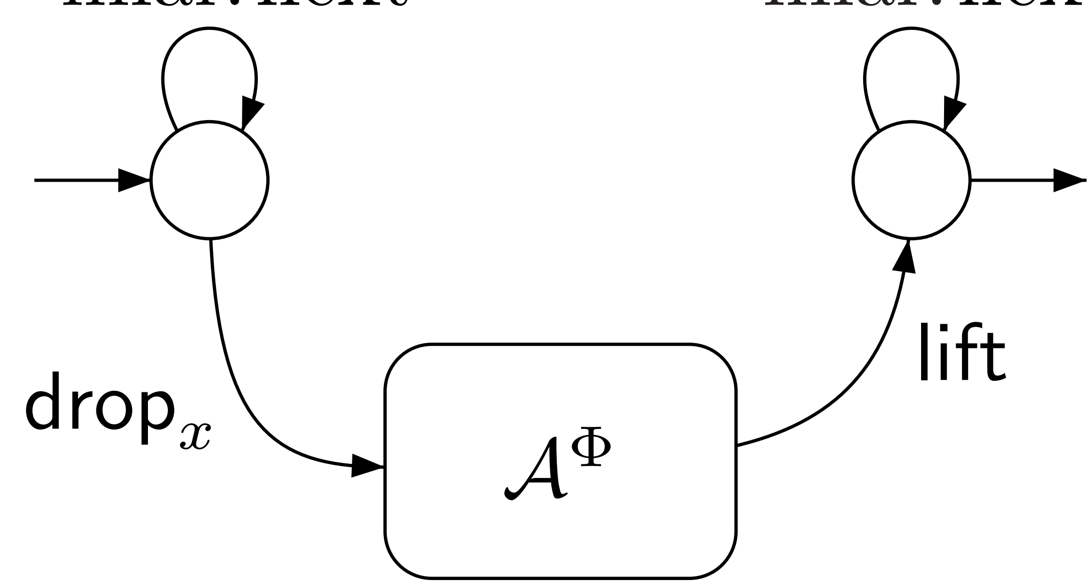


$(\mathbb{N} \cup \{+\infty\}, +, \times, 0, 1)$

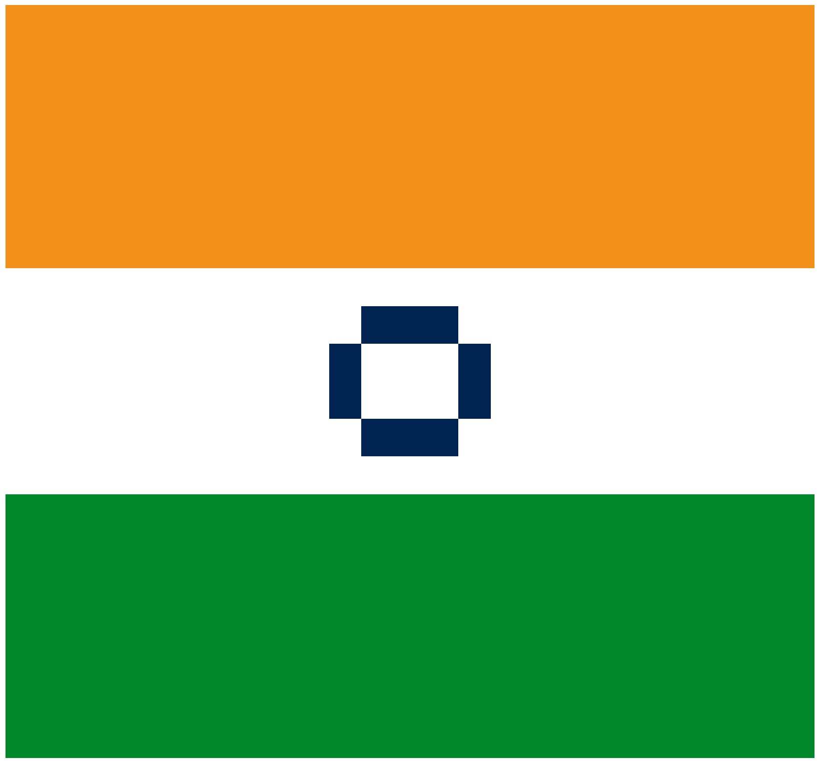
$$\sum_x \Phi(x)$$

use *non-determinism* to count

- a run per position
- each run has the value of the subformula



Translation from Logics to Automata

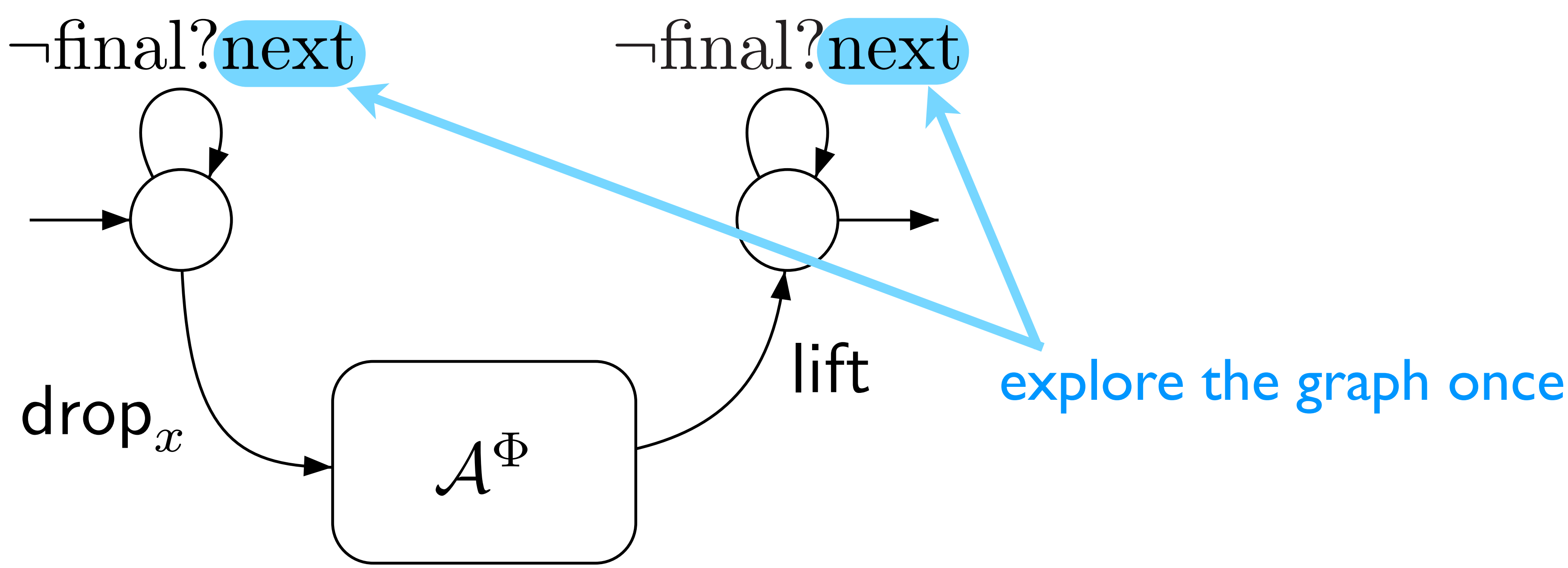


$(\mathbb{N} \cup \{+\infty\}, +, \times, 0, 1)$

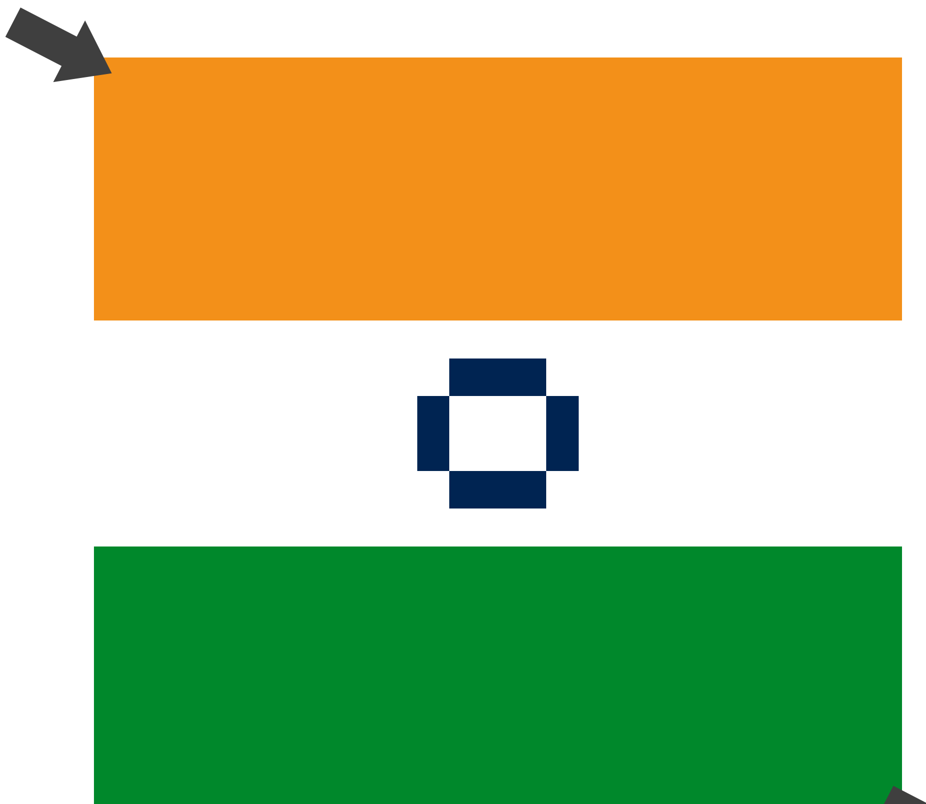
$$\sum_x \Phi(x)$$

use *non-determinism* to count

- a run per position
- each run has the value of the subformula



Translation from Logics to Automata



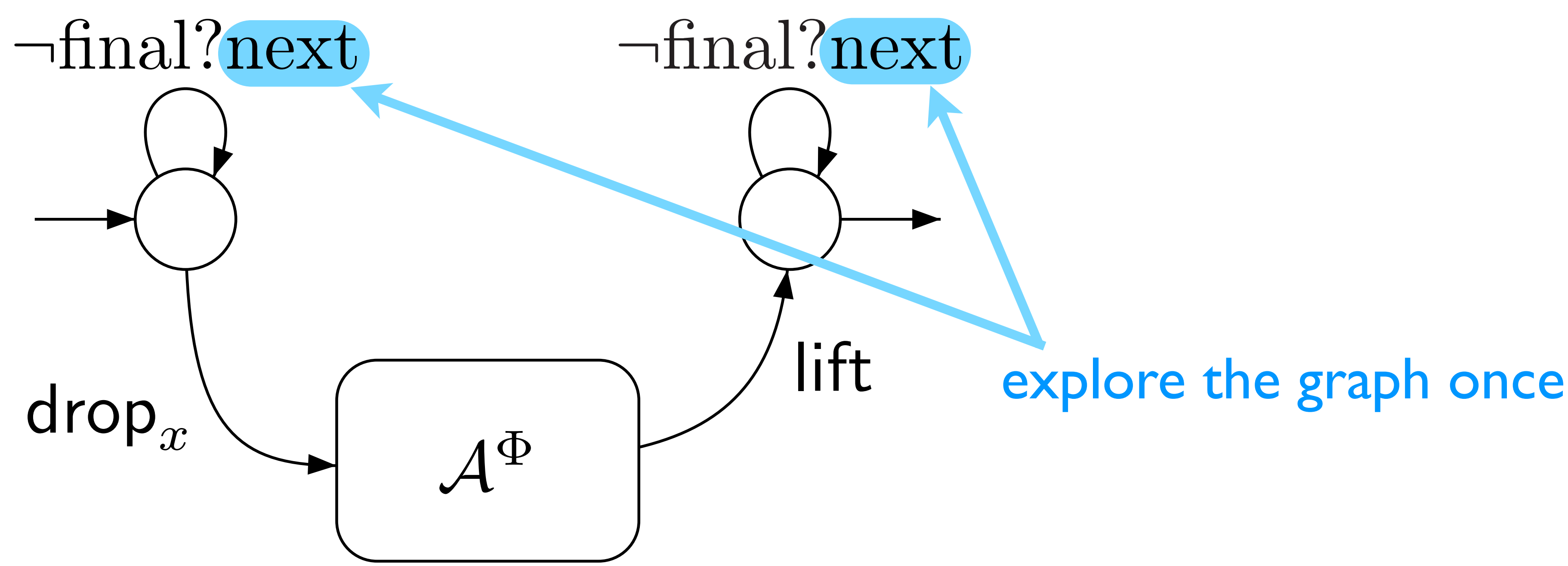
Searchable Graphs

$$G = (V, (E_d)_{d \in D}, \lambda, v^{(i)}, v^{(f)}, \leq)$$

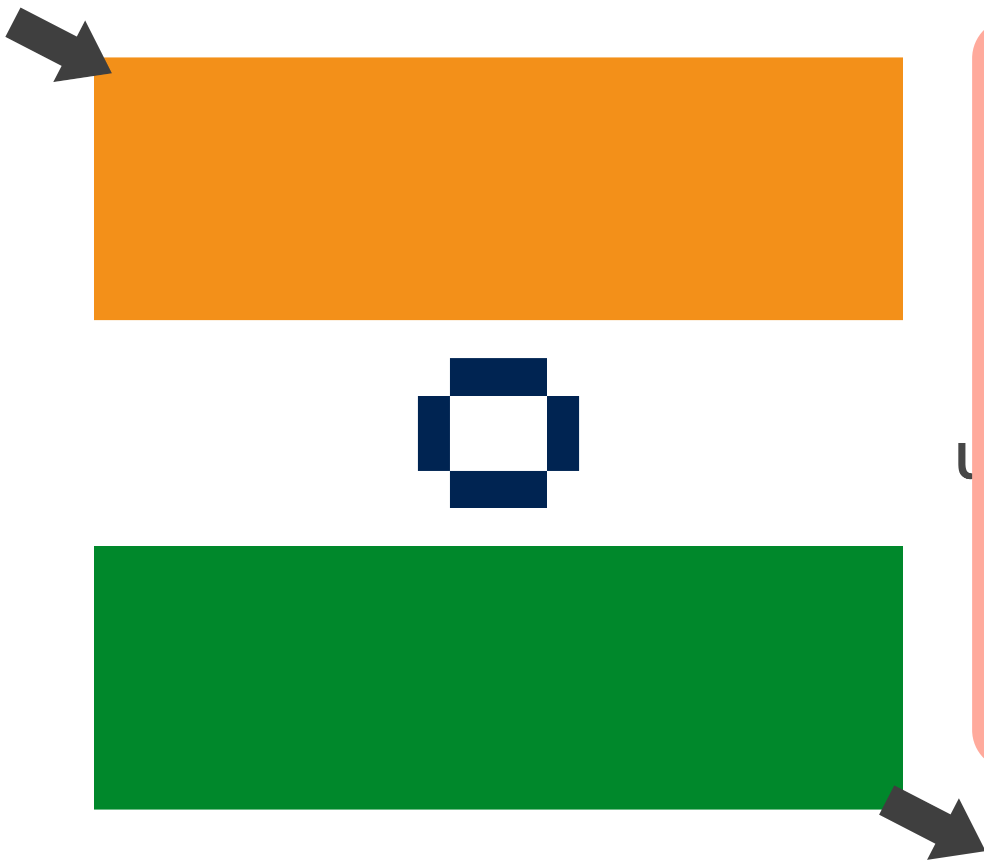
$v^{(i)}$ initial vertex

$v^{(f)}$ final vertex

$\leq$ total order over vertices, computable with navigating automata



Translation from Logics to Automata



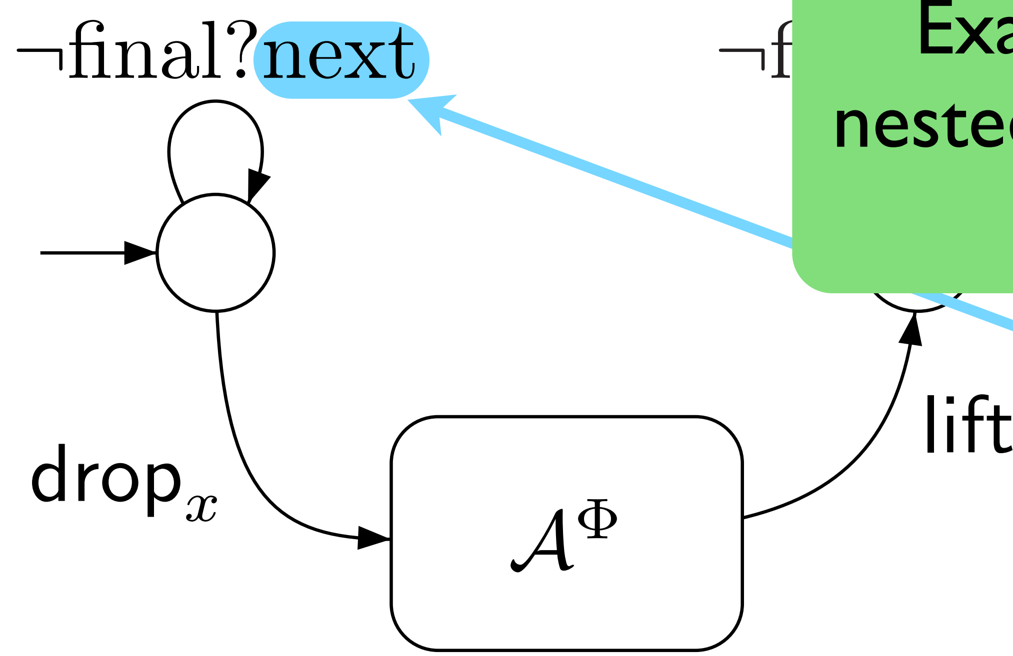
Searchable Graphs

$$G = (V, (E_d)_{d \in D}, \lambda, v^{(i)}, v^{(f)}, \leq)$$

$v^{(i)}$ initial vertex

$v^{(f)}$ final vertex

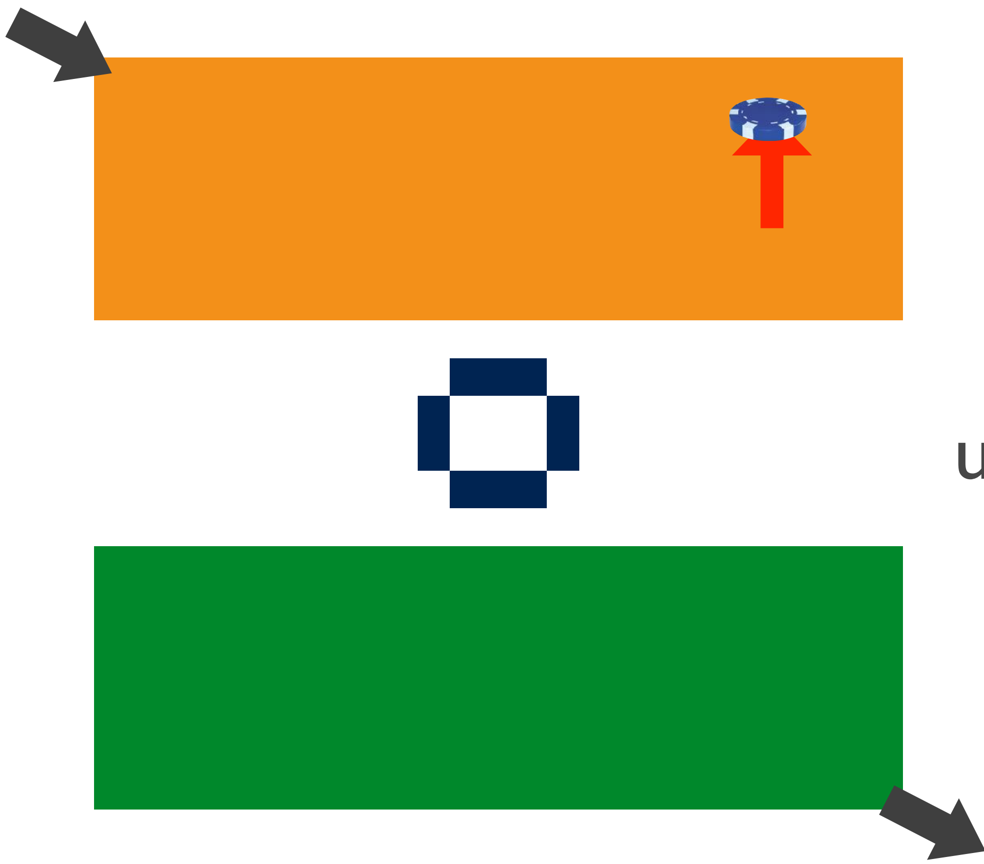
$\leq$ total order over vertices, computable with navigating automata



Examples: words, trees, nested words, Mazurkiewicz traces, pictures...

explore the graph once

Translation from Logics to Automata

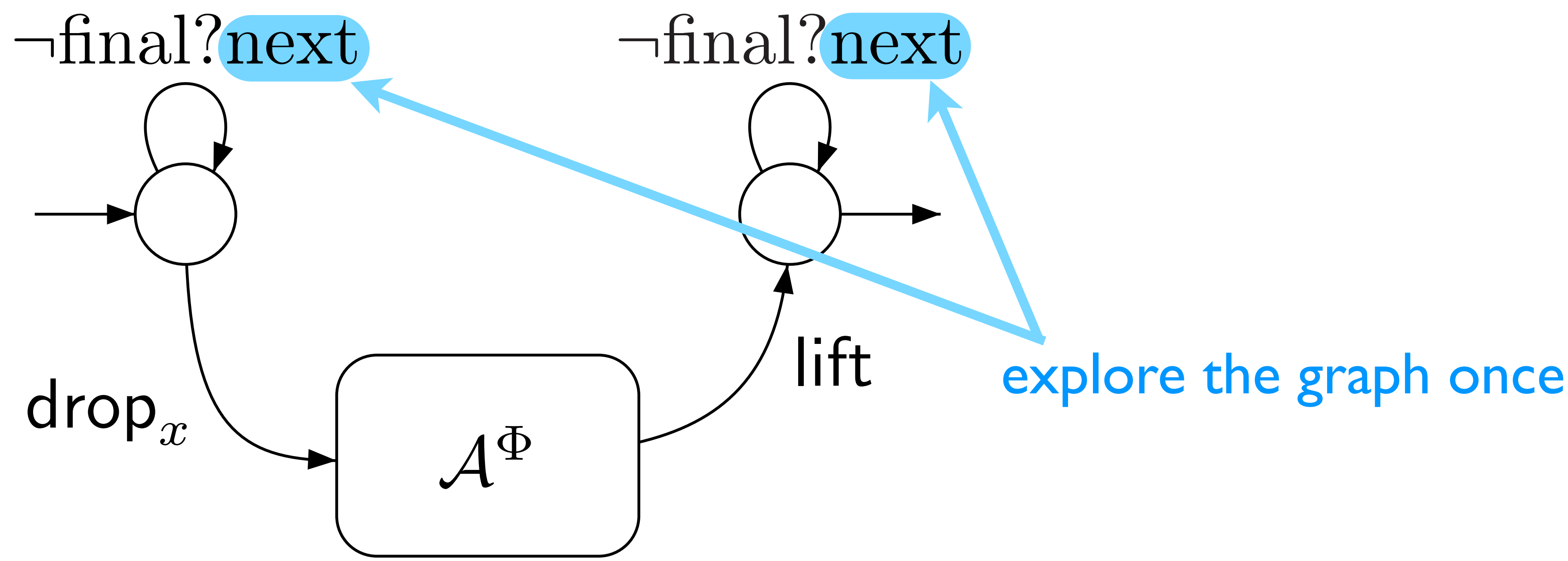


$(\mathbb{N} \cup \{+\infty\}, +, \times, 0, 1)$

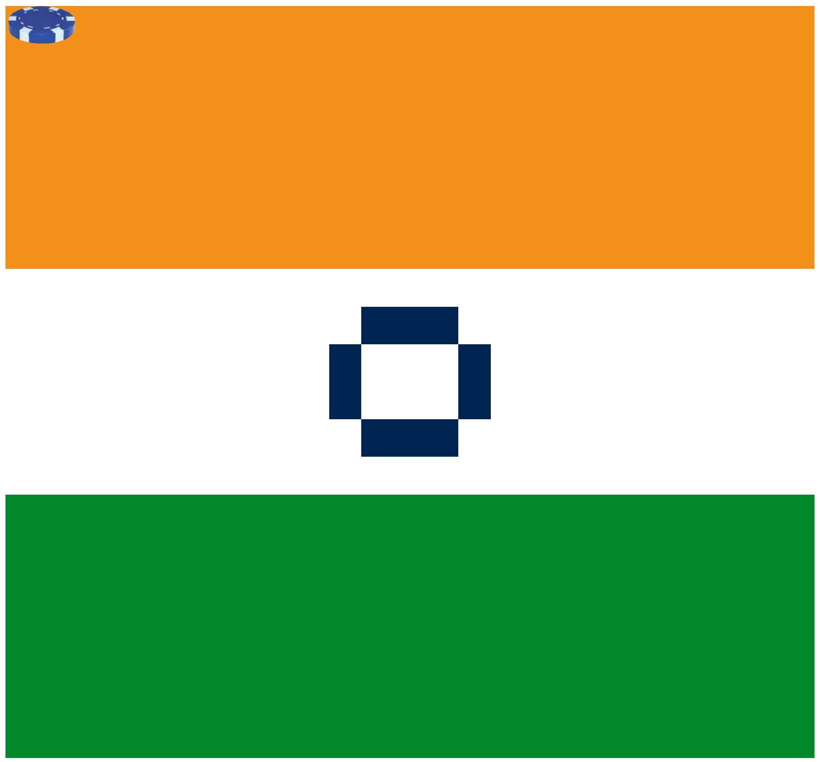
$$\sum_x \Phi(x)$$

use *non-determinism* to count

- a run per position
- each run has the value of the subformula

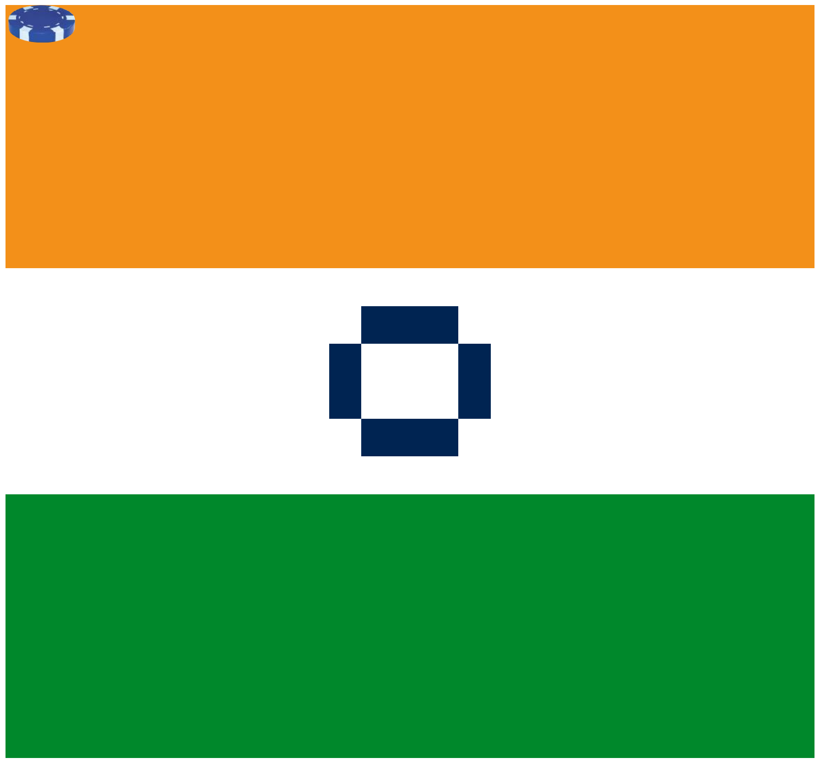


Translation from Logics to Automata



$$\prod_{x} \Phi(x)$$

Translation from Logics to Automata

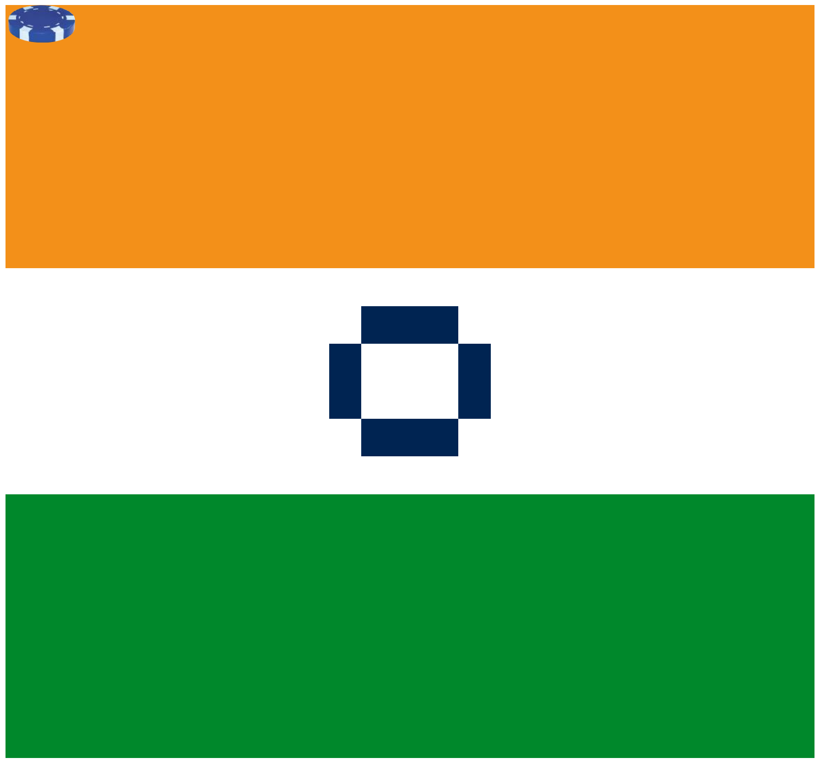


$$\prod_x \Phi(x)$$

use *sequentialization* to multiply

- a single accepting run
- multiply the values of subformula along this run

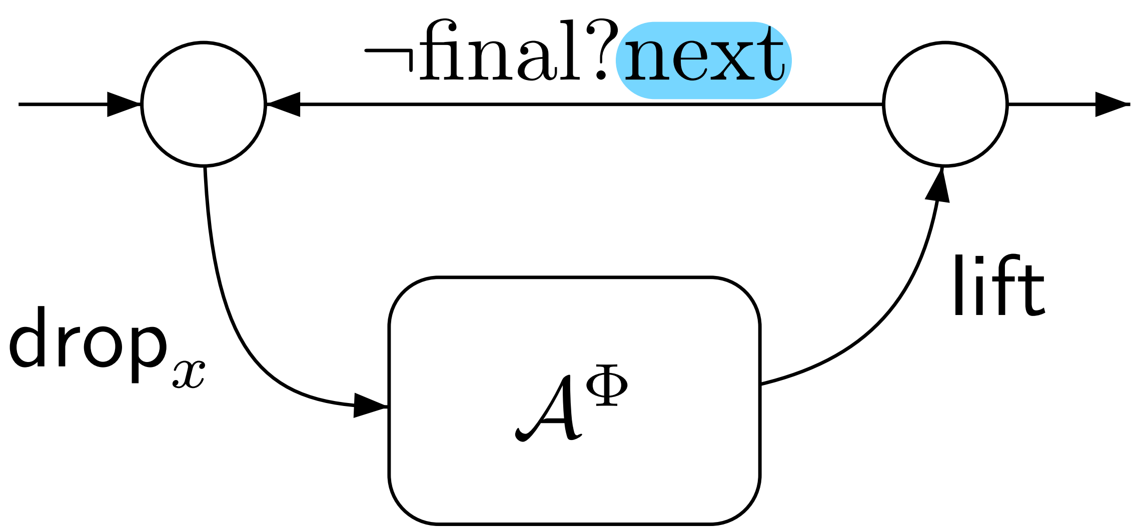
Translation from Logics to Automata



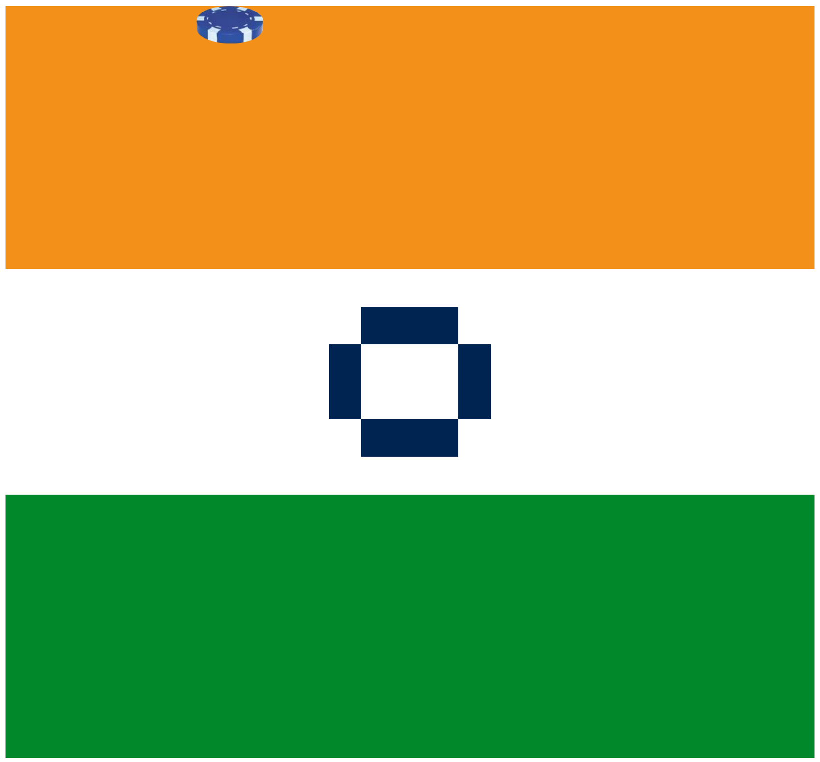
$$\prod_x \Phi(x)$$

use *sequentialization* to multiply

- a single accepting run
- multiply the values of subformula along this run



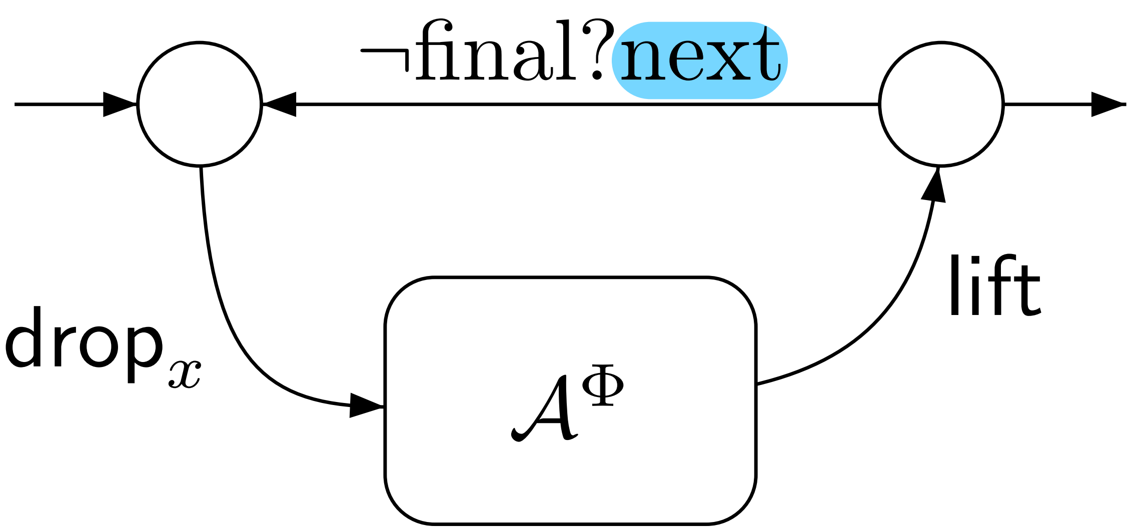
Translation from Logics to Automata



$$\prod_{x} \Phi(x)$$

use *sequentialization* to multiply

- a single accepting run
- multiply the values of subformula along this run



Translation from Logics to Automata

Challenging for the *Boolean* part:
we need unambiguous automata

Translation from Logics to Automata

Challenging for the *Boolean* part:
we need unambiguous automata

Use deterministic automata
of size non-elementary...

Translation from Logics to Automata

Challenging for the *Boolean* part:
we need unambiguous automata

Use deterministic automata
of size non-elementary...

Take advantage of the
navigation and the pebbles to
build linear sized automata

Translation from Logics to Automata

$$(\exists x\, \varphi(x))?3 : 5$$

$$(\mathbb{N} \cup \{+\infty\}, +, \times, 0, 1)$$

Translation from Logics to Automata

$$(\exists x\, \varphi(x))?3 : 5$$

$(\mathbb{N} \cup \{+\infty\}, +, \times, 0, 1)$

use *unambiguous* *non-determinism* to check

- a single accepting run
- run has value 3 or 5 depending on the truth value of the Boolean subformula

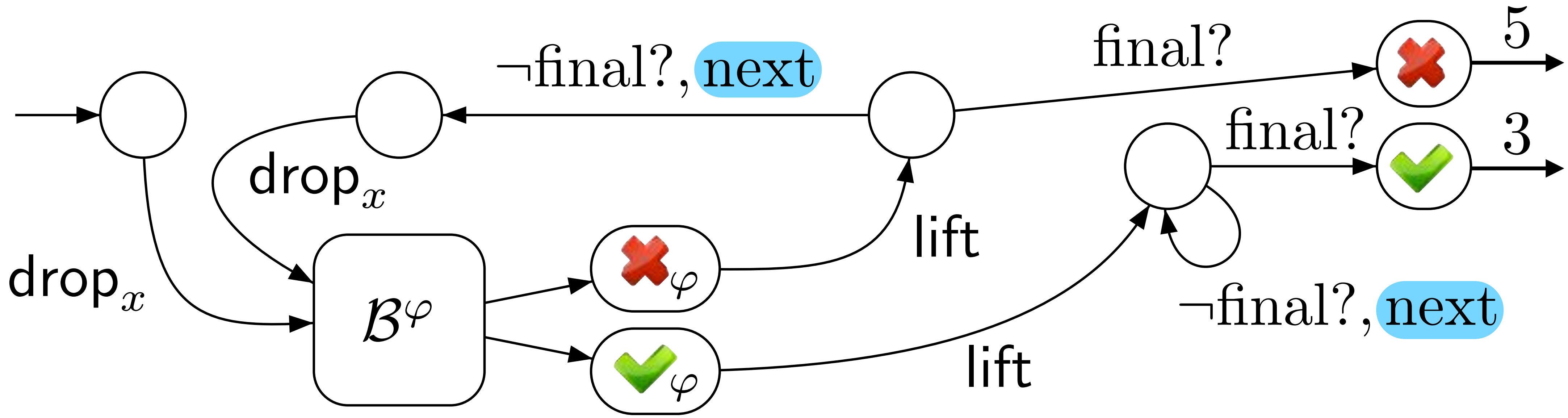
Translation from Logics to Automata

$(\mathbb{N} \cup \{+\infty\}, +, \times, 0, 1)$

$$(\exists x\, \varphi(x))?3 : 5$$

use *unambiguous* *non-determinism* to check

- a single accepting run
- run has value 3 or 5 depending on the truth value of the Boolean subformula



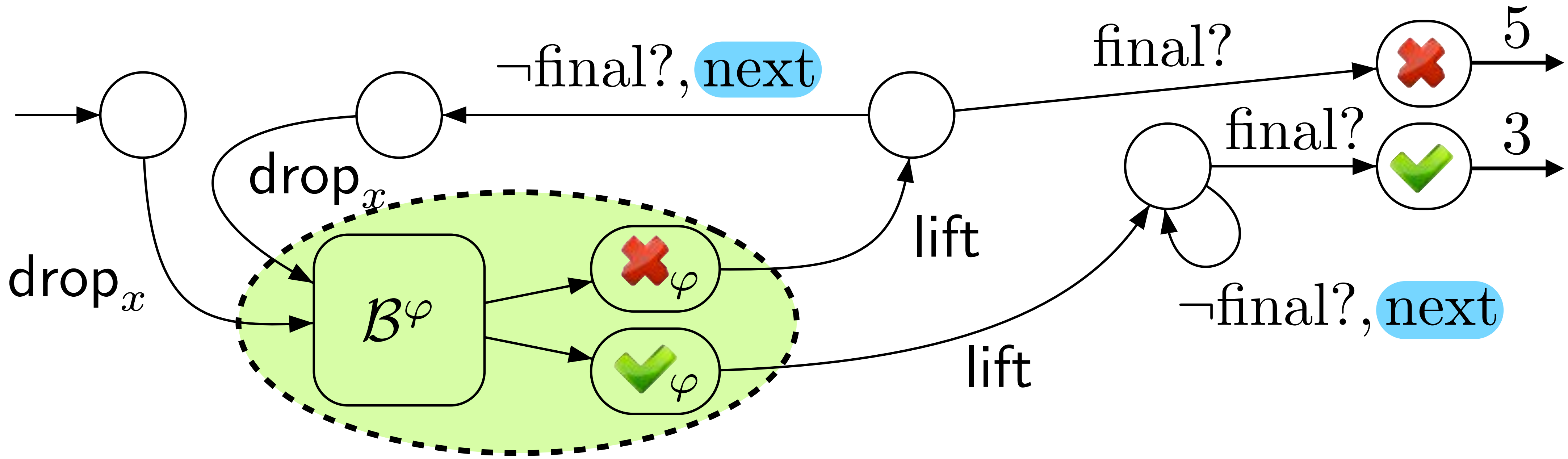
Translation from Logics to Automata

$$(\mathbb{N} \cup \{+\infty\}, +, \times, 0, 1)$$

$$(\exists x\, \varphi(x))?3 : 5$$

use *unambiguous* *non-determinism* to check

- a single accepting run
- run has value 3 or 5 depending on the truth value of the Boolean subformula



unambiguous automaton for formula $\varphi(x)$

Translation from Logics to Automata

Theorem: Consider a searchable class of graph. Every wFO formula can be translated into a Pebble Weighted Automaton equivalent over this class of graphs.

$$\text{WFO} \xrightarrow{\text{linear time}} \text{PWA}$$

Obtained automata are of linear size with respect to the size of the formula

Logic equivalent to PWA?

- Weighted FO misses a counting capability…
- Solution: weighted transitive closure operation

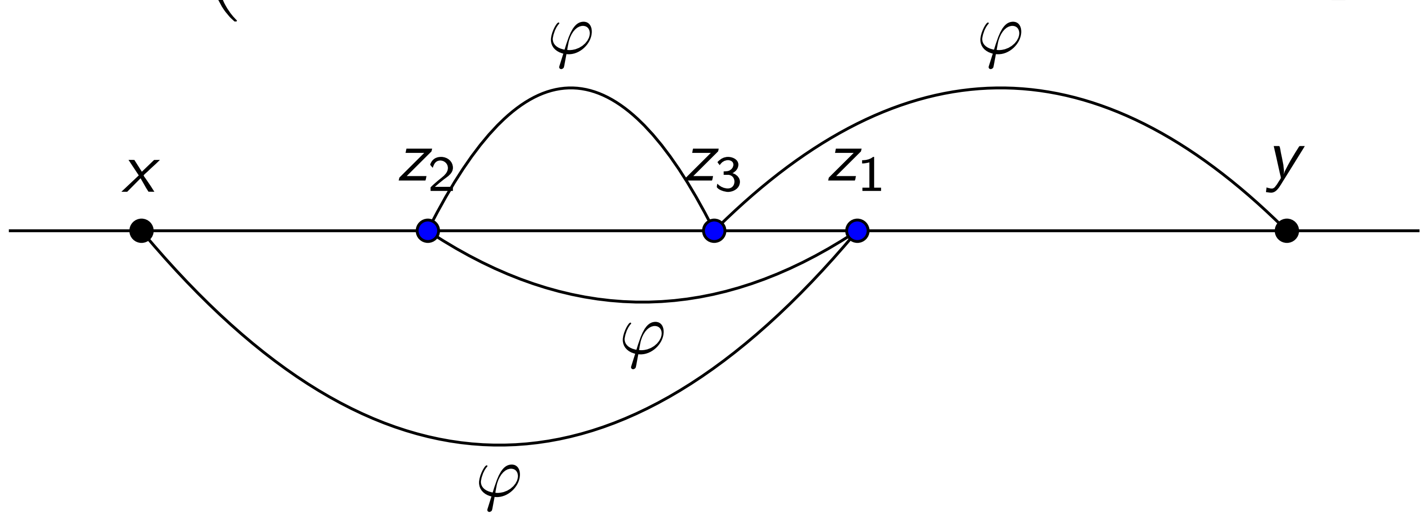
Over words: [Bollig&Gastin&Monmege&Zeitoun 2010]

Logic equivalent to PWA?

- Weighted FO misses a counting capability…
- Solution: weighted transitive closure operation

$$\varphi^1(x,y) = \varphi(x,y)$$

$$\varphi^n(x,y) = \exists z_0 \cdots \exists z_n \Big(x = z_0 \wedge z_n = y \wedge \mathsf{diff}(z_0,\dots,z_n) \wedge \big[\textstyle\bigwedge_{1\leq \ell \leq n} \varphi(z_{\ell-1}, z_\ell)\big]\Big)$$



Over words: [Bollig&Gastin&Monmege&Zeitoun 2010]

Logic equivalent to PWA?

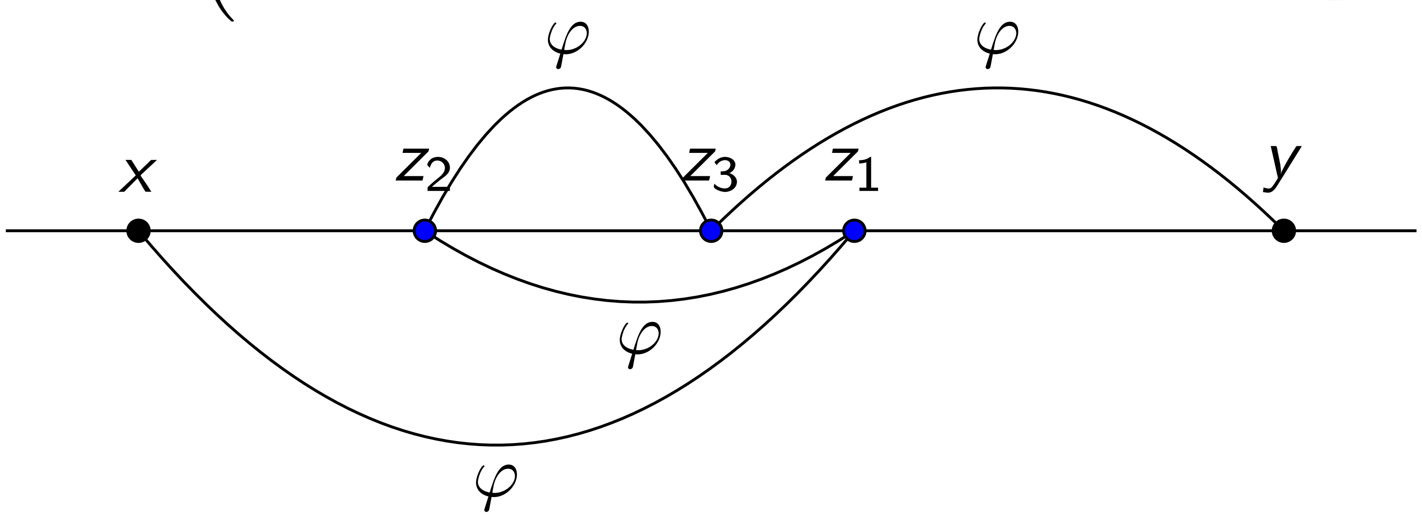
- Weighted FO misses a counting capability…

- Solution: weighted transitive closure operation

$$\varphi^1(x, y) = \varphi(x, y)$$

$$\varphi^n(x, y) = \exists z_0 \cdots \exists z_n \Big(x = z_0 \wedge z_n = y \wedge \mathrm{diff}(z_0, \ldots, z_n) \wedge \big[\textstyle\bigwedge_{1 \leq \ell \leq n} \varphi(z_{\ell-1}, z_\ell)\big]\Big)$$

$$\mathrm{TC}_{xy}\varphi = \textstyle\bigvee_{n \geq 1} \varphi^n$$



Over words: [Bollig&Gastin&Monmege&Zeitoun 2010]

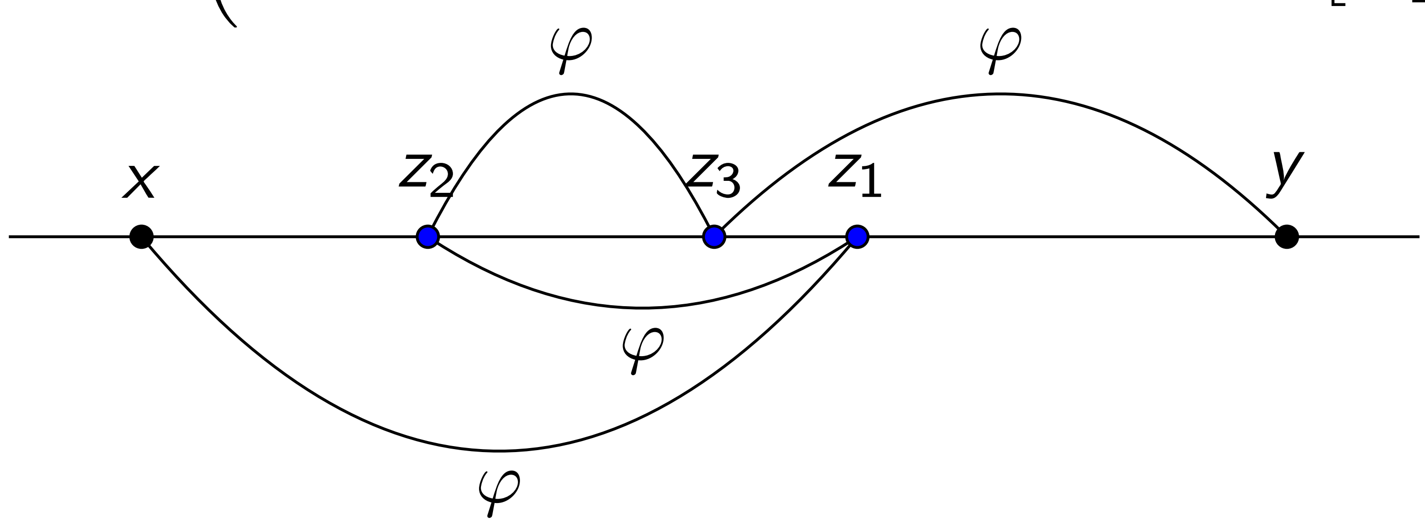
Logic equivalent to PWA?

- Weighted FO misses a counting capability…
- Solution: weighted transitive closure operation

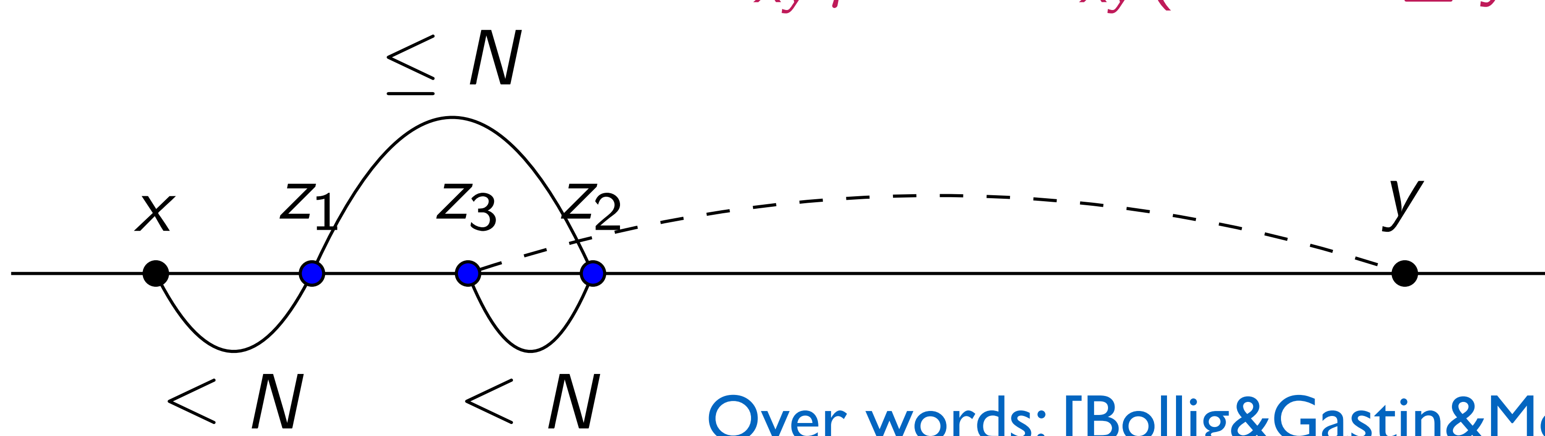
$$\varphi^1(x,y) = \varphi(x,y)$$

$$\varphi^n(x,y) = \exists z_0 \cdots \exists z_n \Big(x = z_0 \wedge z_n = y \wedge \mathrm{diff}(z_0,\dots,z_n) \wedge \big[\bigwedge_{1\leq \ell \leq n} \varphi(z_{\ell-1}, z_\ell)\big]\Big)$$

$$\mathrm{TC}_{xy}\varphi = \bigvee_{n\geq 1} \varphi^n$$



Bounded transitive closure : $N\text{-}\mathrm{TC}_{xy}\varphi = \mathrm{TC}_{xy}(x - N \leq y \leq x + N \wedge \varphi)$



Over words: [Bollig&Gastin&Monmege&Zeitoun 2010]

Logic equivalent to PWA?

- Weighted FO misses a counting capability…
- Solution: weighted transitive closure operation

Theorem: Weighted First Order logic with weighted transitive closure and Pebble Weighted Automata are equivalent for *zonable* and searchable classes of graphs.

Over words: [Bollig&Gastin&Monmege&Zeitoun 2010]

Logic equivalent to PWA?

- Weighted FO misses a counting capability…
- Solution: weighted transitive closure operation

Theorem: Weighted First Order logic with weighted transitive closure and Pebble Weighted Automata are equivalent for *zonable* and searchable classes of graphs.

Examples: words, trees, nested words, Mazurkiewicz traces, pictures...

Over words: [Bollig&Gastin&Monmege&Zeitoun 2010]

Deciding wFO?

Input: A pebble weighted automata / A formula of wFO+BTC
Question: Does there exist an equivalent formula in wFO?

Deciding wFO?

Input: A pebble weighted automata / A formula of wFO+BTC
Question: Does there exist an equivalent formula in wFO?

Deciding FO?

Input: A finite automaton
Question: Does there exist an equivalent formula in FO?

Deciding FO?

Input: A finite automaton
Question: Does there exist an equivalent formula in FO?

[Schützenberger 65, McNaughton&Papert 71, Diekert&Gastin 2008]

Deciding FO?

Input: A finite automaton
Question: Does there exist an equivalent formula in FO?

Theorem: For a language L of finite (or infinite) words, TFAE
- L is FO definable
- L is aperiodic
- L is LTL definable
- L is accepted by some counter-free automaton
- L is accepted by some aperiodic automaton…

[Schützenberger 65, McNaughton&Papert 71, Diekert&Gastin 2008]

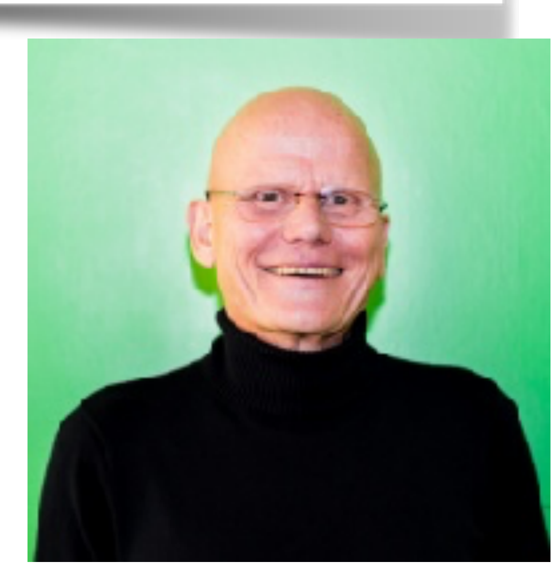
Deciding FO?

Input: A finite automaton
Question: Does there exist an equivalent formula in FO?

Theorem: For a language L of finite (or infinite) words, TFAE

- L is FO definable
- L is aperiodic
- L is LTL definable
- L is accepted by some counter-free automaton
- L is accepted by some aperiodic automaton…

$$\exists m \geq 1 \qquad p \xrightarrow{u^m} q \iff p \xrightarrow{u^{m+1}} q$$



[Schützenberger 65, McNaughton&Papert 71, Diekert&Gastin 2008]

Deciding FO?

Input: A finite automaton
Question: Does there exist an equivalent formula in FO?

Theorem: For a language L of finite (or infinite) words, TFAE
- L is FO definable
- L is aperiodic
- L is LTL definable
- L is accepted by some counter-free automaton
- L is accepted by some aperiodic automaton…

$$\exists m \geq 1 \qquad p \xrightarrow{u^m} q \iff p \xrightarrow{u^{m+1}} q$$

PSPACE-complete… using algebra

[Schützenberger 65, McNaughton&Papert 71, Diekert&Gastin 2008]

Deciding core-wFO?

Input: A weighted automaton / A formula of core-wMSO
Question: Does there exist an equivalent formula in core-wFO?

Deciding core-wFO?

Input: A weighted automaton / A formula of core-wMSO
Question: Does there exist an equivalent formula in core-wFO?

$$\varphi ::= \top \mid P_a(x) \mid x \leq y \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x \varphi \qquad \text{(FO)}$$

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

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

Deciding core-wFO?

Input: A weighted automaton / A formula of core-wMSO
Question: Does there exist an equivalent formula in core-wFO?

$$\varphi ::= \top \mid P_a(x) \mid x \leq y \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x \varphi \qquad \text{(FO)}$$

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

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

Deciding core-wFO?

Input: A weighted automaton / A formula of core-wMSO
Question: Does there exist an equivalent formula in core-wFO?

Deciding core-wFO?

Input: A weighted automaton / A formula of core-wMSO
Question: Does there exist an equivalent formula in core-wFO?

Theorem: [Droste&Gastin 2019]
core-wFO = aperiodic poly-ambiguous WA

Deciding core-wFO?

Input: A weighted automaton / A formula of core-wMSO
Question: Does there exist an equivalent formula in core-wFO?

Theorem: [Droste&Gastin 2019]
core-wFO = aperiodic poly-ambiguous WA
core-wFO without $\sum_x$ = aperiodic finitely-ambiguous WA

Deciding core-wFO?

Input: A weighted automaton / A formula of core-wMSO
Question: Does there exist an equivalent formula in core-wFO?

Theorem: [Droste&Gastin 2019]
core-wFO = aperiodic poly-ambiguous WA
core-wFO without $\sum_x$ = aperiodic finitely-ambiguous WA
core-wFO without + and $\sum_x$ = aperiodic unambiguous WA

Deciding core-wFO?

Input: A weighted automaton / A formula of core-wMSO
Question: Does there exist an equivalent formula in core-wFO?

Theorem: [Droste&Gastin 2019]
core-wFO = aperiodic poly-ambiguous WA
core-wFO without $\sum_x$ = aperiodic finitely-ambiguous WA
core-wFO without + and $\sum_x$ = aperiodic unambiguous WA

Decision procedure?… algebra is missing

A special case: the transducers

Functions {

- Two-way Deterministic Finite-State Transducers
- Functional One-way Finite-State Transducers
- MSOT (à la Courcelle)
- Copyless Streaming String Transducers (Alur et al)

A special case: the transducers

Functions

- Two-way Deterministic Finite-State Transducers
- Functional One-way Finite-State Transducers
- MSOT (à la Courcelle)
- Copyless Streaming String Transducers (Alur et al)

Relations

- Two-way Non-Deterministic Finite-State Transducers
- Non-Deterministic Finite-State Transducers
- Non-deterministic Copyless Streaming String Transducers (Alur et Deshmukh)
- NMSOT (with free second-order variables)

A special case: the transducers

Functions
- Two-way Deterministic Finite-State Transducers
- Functional One-way Finite-State Transducers
- MSOT (à la Courcelle)
- Copyless Streaming String Transducers (Alur et al)

Relations
- Two-way Non-Deterministic Finite-State Transducers
- Non-Deterministic Finite-State Transducers
- Non-deterministic Copyless Streaming String Transducers (Alur et Deshmukh)
- NMSOT (with free second-order variables)

→ only finite valued relations…

Transduction as weights

- Desire: weight transitions with words… Difficult to equip A* with a semiring structure: how to combine several accepting runs?

- Works for deterministic or unambiguous automata: functional transducers

- For relations: semiring of languages

$$(2^{A^*}, \cup, \cdot, \varnothing, \{\varepsilon\})$$

Examples

$$\prod_x \left(P_x(a)?\{aa\} : \left(P_x(b)?\{bb\} : \varnothing\right)\right)$$

Examples

$\prod_x (P_x(a)?\{aa\} : (P_x(b)?\{bb\} : \varnothing))$ *aba* -> *aabbaa*

Examples

$$\prod_x (P_x(a)?\{aa\} : (P_x(b)?\{bb\} : \varnothing))$$

aba -> *aabbaa*

$$\prod_x (P_x(\star)?\{insert\} : (P_x(a)?\{a\} : (P_x(b) : \{b\})))$$

Examples

$\prod_x (P_x(a)?\{aa\} : (P_x(b)?\{bb\} : \varnothing))$ *aba* -> *aabbaa*

$\prod_x (P_x(\star)?\{insert\} :(P_x(a)?\{a\} :(P_x(b) : \{b\})))$ *a*b* -> *ainsertb*

Examples

$$\prod_x (P_x(a)?\{aa\} : (P_x(b)?\{bb\} : \varnothing))$$ *aba* -> *aabbaa*

$$\prod_x (P_x(\star)?\{insert\} :(P_x(a)?\{a\} :(P_x(b) : \{b\})))$$ *a*b* -> *ainsertb*

$$\sum_y P_y(\star)?\Big[\prod_x (x = y)?\{insert\} : (P_x(\star)?\{\varepsilon\} : (P_x(a)?\{a\} : (P_x(b)?\{b\})))\Big]$$

Examples

$$\prod_x (P_x(a)?\{aa\} : (P_x(b)?\{bb\} : \varnothing))$$

aba -> *aabbaa*

$$\prod_x (P_x(\star)?\{insert\} : (P_x(a)?\{a\} : (P_x(b) : \{b\})))$$

*a*b* -> *ainsertb*

$$\sum_y P_y(\star)?\left[\prod_x (x = y)?\{insert\} : (P_x(\star)?\{\varepsilon\} : (P_x(a)?\{a\} : (P_x(b)?\{b\})))\right]$$

*a*b*a* -> {*ainsertba*,*abinserta*}

Relation

Examples

$$\prod_x P_x(a)?\{a\} : (P_x(b)?\{\varepsilon\}) \times \prod_x P_x(a)?\{\varepsilon\} : (\,P_x(b) : \{c\})$$

Examples

$$\prod_x P_x(a)?\{a\}:(P_x(b)?\{\varepsilon\}) \times \prod_x P_x(a)?\{\varepsilon\}:(P_x(b):\{c\})$$

ababbaabb -> *aaaaccccc*

Examples

$$\prod_x P_x(a)?\{a\}:(P_x(b)?\{\varepsilon\})\times\prod_x P_x(a)?\{\varepsilon\}:($$

Not comp. by 1-way
Func Transducer

ababbaabb -> *aaaaccccc*

Examples

Not comp. by 1-way
Func Transducer

$$\prod_x P_x(a)?\{a\} : (P_x(b)?\{\varepsilon\}) \times \prod_x P_x(a)?\{\varepsilon\} : ($$

ababbaabb -> *aaaaccccc*

$$\prod_x P_x(a)?\{a\} : (P_x(b)?\{\varepsilon\}) + \prod_x P_x(a)?\{\varepsilon\} : (P_x(b) \times \{c\})$$

Examples

$$\prod_x P_x(a)?\{a\}:(P_x(b)?\{\varepsilon\})\times\prod_x P_x(a)?\{\varepsilon\}:($$

Not comp. by 1-way
Func Transducer

ababbaabb -> *aaaaccccc*

$$\prod_x P_x(a)?\{a\}:(P_x(b)?\{\varepsilon\})+\prod_x P_x(a)?\{\varepsilon\}:(P_x(b)\times\{c\})$$

ababbaabb -> {*aaaa*,*ccccc*}

Examples

$$\prod_x P_x(a)?\{a\}:(P_x(b)?\{\varepsilon\})\times\prod_x P_x(a)?\{\varepsilon\}:($$

Not comp. by 1-way
Func Transducer

ababbaabb -> *aaaaccccc*

$$\prod_x P_x(a)?\{a\}:(P_x(b)?\{\varepsilon\})+\prod_x P_x(a)?\{\varepsilon\}:(P_x(b)\times\{c\})$$

ababbaabb -> {*aaaa*,*ccccc*}

$$\prod_x P_x(a)?\{a,\varepsilon\}:(P_x(b)?\{b,\varepsilon\})$$

Examples

$$\prod_x P_x(a)?\{a\}:(P_x(b)?\{\varepsilon\})\times\prod_x P_x(a)?\{\varepsilon\}:($$

Not comp. by 1-way
Func Transducer

ababbaabb -> *aaaaccccc*

$$\prod_x P_x(a)?\{a\}:(P_x(b)?\{\varepsilon\})+\prod_x P_x(a)?\{\varepsilon\}:(P_x(b)\times\{c\})$$

ababbaabb -> {*aaaa,ccccc*}

$$\prod_x P_x(a)?\{a,\varepsilon\}:(P_x(b)?\{b,\varepsilon\})$$

aba -> {*ε,a,b,ab,ba,aa,aba*}

Examples

$$\prod_x P_x(a)?\{a\}:(P_x(b)?\{\varepsilon\}) \times \prod_x P_x(a)?\{\varepsilon\}:($$

Not comp. by 1-way Func Transducer

ababbaabb -> *aaaaccccc*

$$\prod_x P_x(a)?\{a\}:(P_x(b)?\{\varepsilon\}) + \prod_x P_x(a)?\{\varepsilon\}:(P_x(b) \times \{c\})$$

ababbaabb -> {*aaaa*,*ccccc*}

$$\prod_x P_x(a)?\{a,\varepsilon\}:(P_x(b)?\{b,\varepsilon\})$$

aba -> {*ε*,*a*,*b*,*ab*,*ba*,*aa*,*aba*}

$$\prod_x P_x(a)?A^*aA^*:(P_x(b)?A^*bA^*)$$

Examples

$$\prod_x P_x(a)?\{a\}:(P_x(b)?\{\varepsilon\}) \times \prod_x P_x(a)?\{\varepsilon\}:($$

Not comp. by 1-way
Func Transducer

ababbaabb -> *aaaaccccc*

$$\prod_x P_x(a)?\{a\}:(P_x(b)?\{\varepsilon\}) + \prod_x P_x(a)?\{\varepsilon\}:(P_x(b) \times \{c\})$$

ababbaabb -> {*aaaa*,*ccccc*}

$$\prod_x P_x(a)?\{a,\varepsilon\}:(P_x(b)?\{b,\varepsilon\})$$

aba

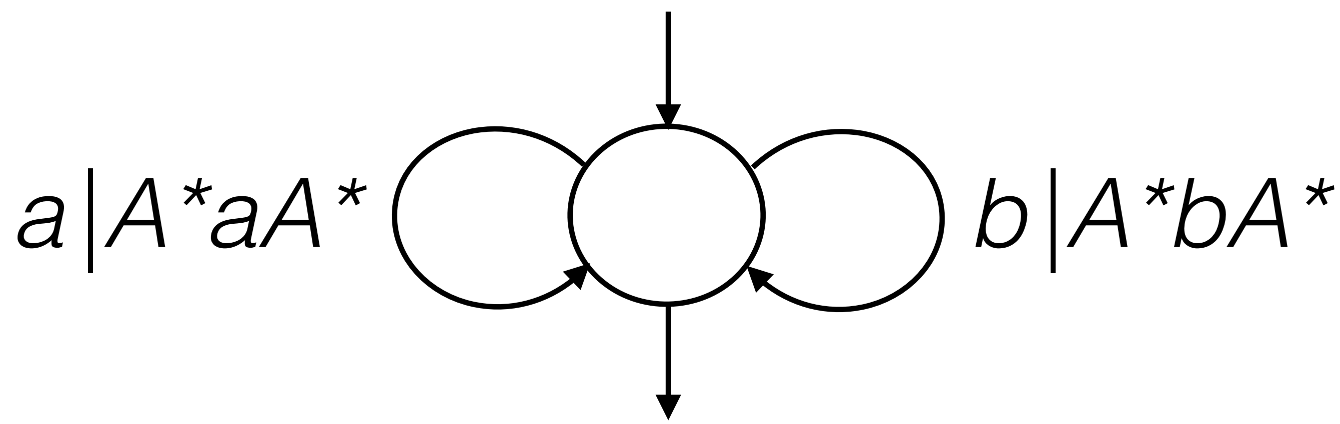
Infinitely-valued relation

$$\prod_x P_x(a)?\,A^* a A^*:(P_x(b)?\,A^* b A^*)$$

aba -> *A*aA*bA*aA**

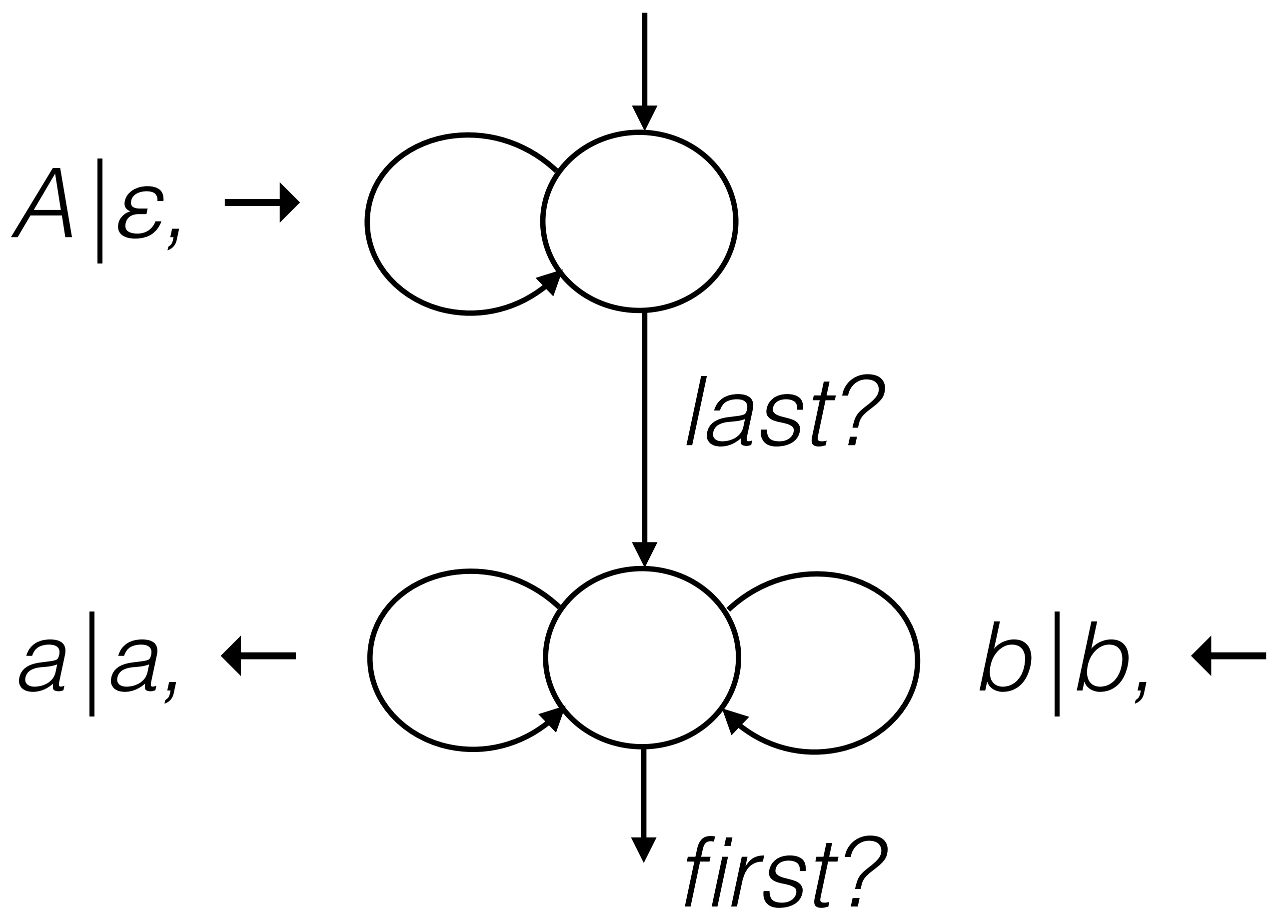
Transducers

$$\prod_x P_x(a)?\, A^* a A^* : (P_x(b)?\, A^* b A^*)$$

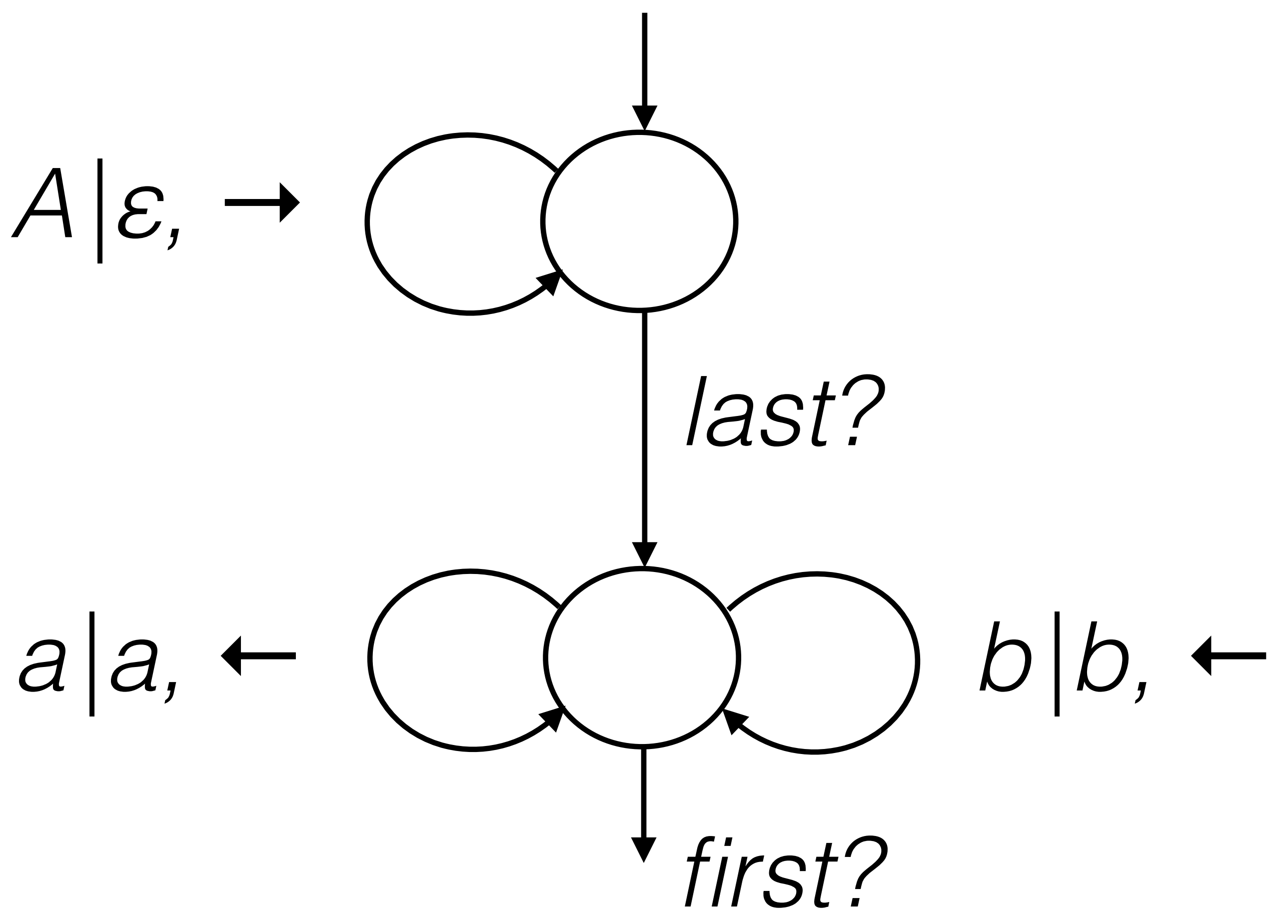


Infinite-valued, but <u>deterministic</u>

Reverse?

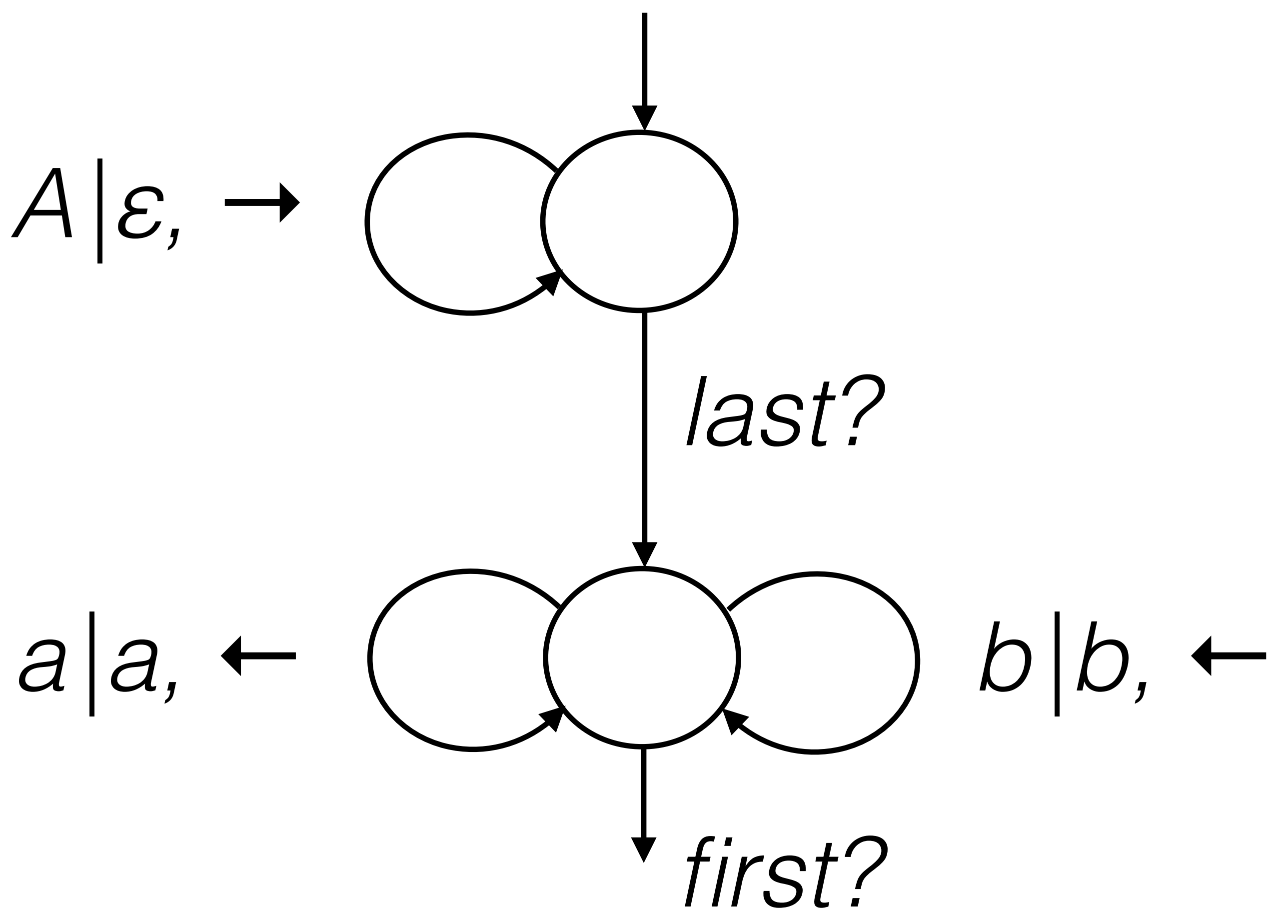


Reverse?



Impossible in FO…
… because of order of interpretation of product

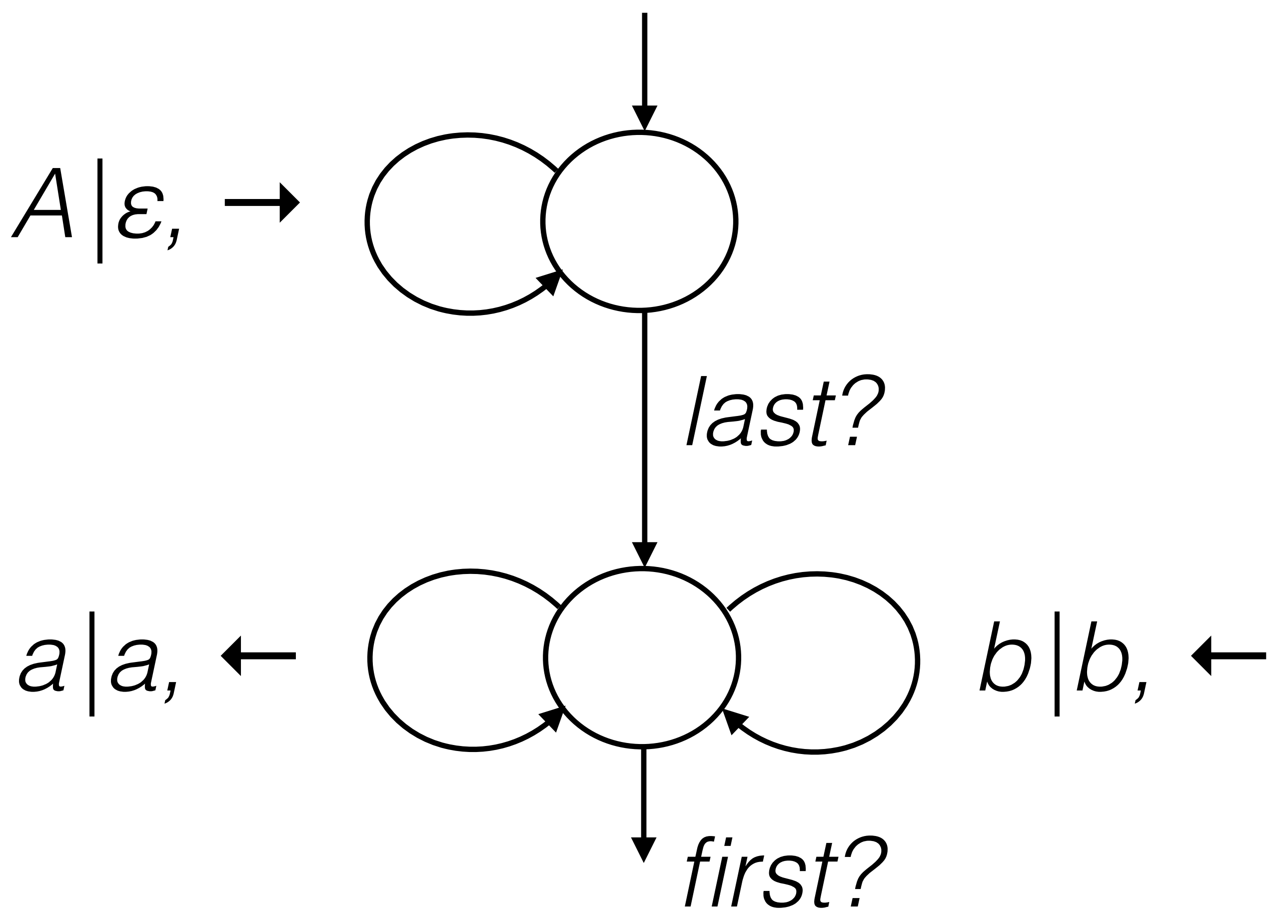
Reverse?



Impossible in FO…
… because of order of
interpretation of product

Solution: in this non-commutative setting,
add right-to-left products

Reverse?



Impossible in FO…
… because of order of interpretation of product

Solution: in this non-commutative setting, add right-to-left products

$$\coprod_x P_x(a)?\{a\} : (P_x(b)?\{b\})$$

Transitive closure

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi$$

$$\Phi ::= L \mid \varphi ? \Phi : \Phi \mid \Phi + \Phi \mid \Phi \times \Phi \mid \sum_x \Phi \mid \prod_x \Phi \mid N\text{-}TC_{x,y} \Phi$$

Transitive closure

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi$$

$$\Phi ::= L \mid \varphi ? \Phi : \Phi \mid \Phi + \Phi \mid \Phi \times \Phi \mid \sum_x \Phi \mid \prod_x \Phi \mid N\text{-}TC_{x,y} \Phi$$

Regular language

Transitive closure

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi$$

$$\Phi ::= L \mid \varphi ? \Phi : \Phi \mid \Phi + \Phi \mid \Phi \times \Phi \mid \sum_x \Phi \mid \prod_x \Phi \mid N\text{-}TC_{x,y} \Phi$$

Regular language

Able to define right-to-left product

$$\coprod_x \Phi(x) := [1\text{-}TC_{x,y}(y = x - 1 \,?\, \Phi(x))](\mathit{last}, \mathit{first}) \times \Phi(\mathit{first})$$

Transitive closure

$$\varphi ::= \top \mid P_a(x) \mid x \leqslant y \mid x \in X \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\, \varphi$$

$$\Phi ::= L \mid \varphi ? \Phi : \Phi \mid \Phi + \Phi \mid \Phi \times \Phi \mid \sum_x \Phi \mid \prod_x \Phi \mid N\text{-}TC_{x,y}\, \Phi$$

Regular language

Able to define right-to-left product

$$\coprod_x \Phi(x) \coloneqq [1\text{-}TC_{x,y}(y = x - 1 ? \Phi(x))](\mathit{last}, \mathit{first}) \times \Phi(\mathit{first})$$

Theorem: Pebble Transducers = wFO + bounded-TC

with regular language productions

linear transformation from logic to transducers

Functional transductions

Functional transductions

Theorem: Polyregular functions [Bojańczyk 2018]
Deterministic pebble automata

Functional transductions

Theorem: Polyregular functions [Bojańczyk 2018]
Deterministic pebble automata
= For–transducers

Functional transductions

Theorem: Polyregular functions [Bojańczyk 2018]
Deterministic pebble automata
= For–transducers
= Smallest class of transductions closed under composition, containing iterated reverse and squaring

Functional transductions

Theorem: Polyregular functions [Bojańczyk 2018]
Deterministic pebble automata
= For–transducers
= Smallest class of transductions closed under composition, containing iterated reverse and squaring
= A fragment of λ-calculus…

Functional transductions

Theorem: Polyregular functions [Bojańczyk 2018]
Deterministic pebble automata
= For–transducers
= Smallest class of transductions closed under composition, containing iterated reverse and squaring
= A fragment of λ-calculus…

= wFO without + and $\sum_x$, all weights being words, underlying boolean logic being MSO

Functional transductions

Theorem: Polyregular functions [Bojańczyk 2018]
Deterministic pebble automata
= For–transducers
= Smallest class of transductions closed under composition, containing iterated reverse and squaring
= A fragment of λ-calculus…

= wFO without + and $\sum_x$, all weights being words, underlying boolean logic being MSO

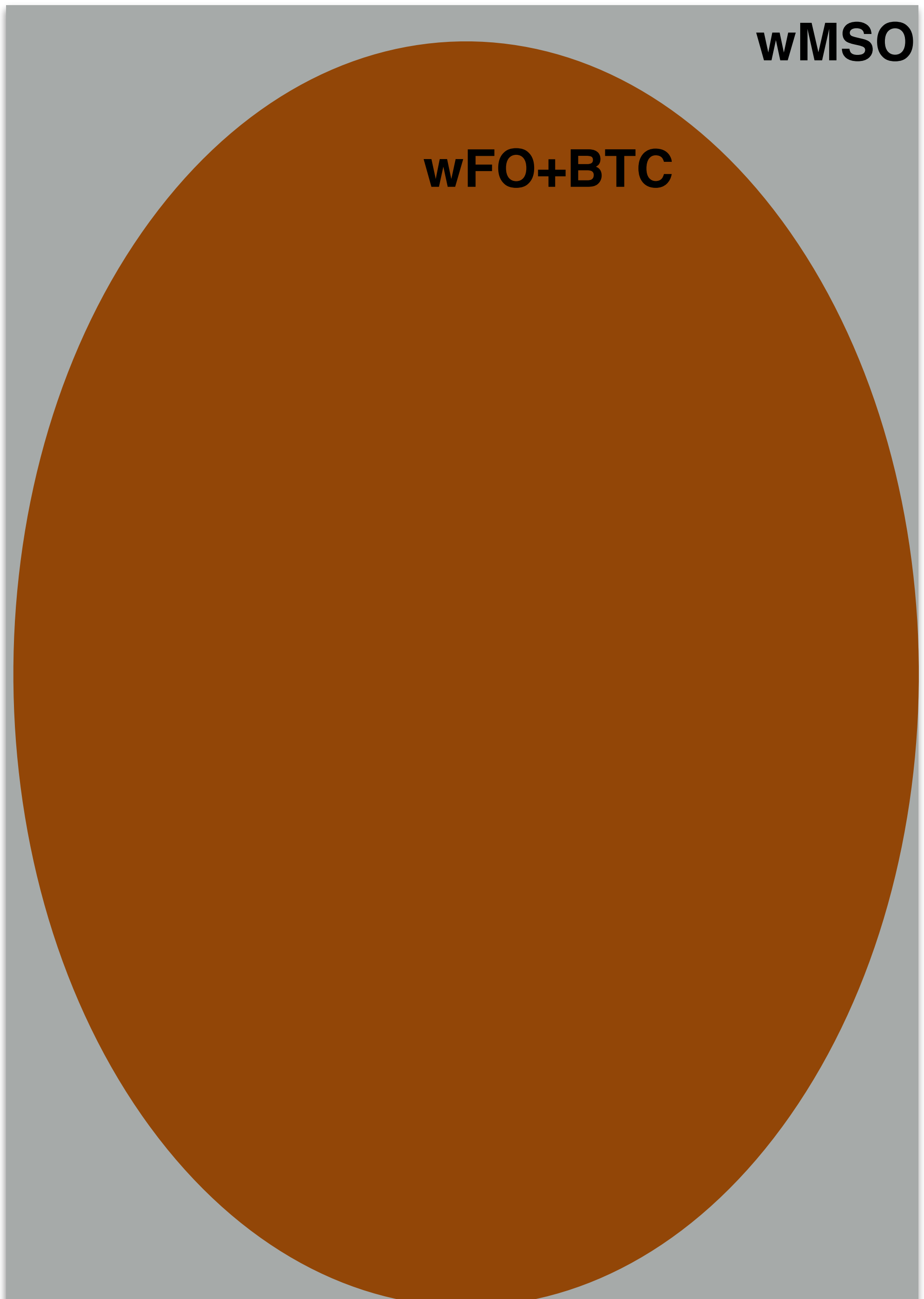
Similar characterizations for relational transductions / weighted functions ?

Summary

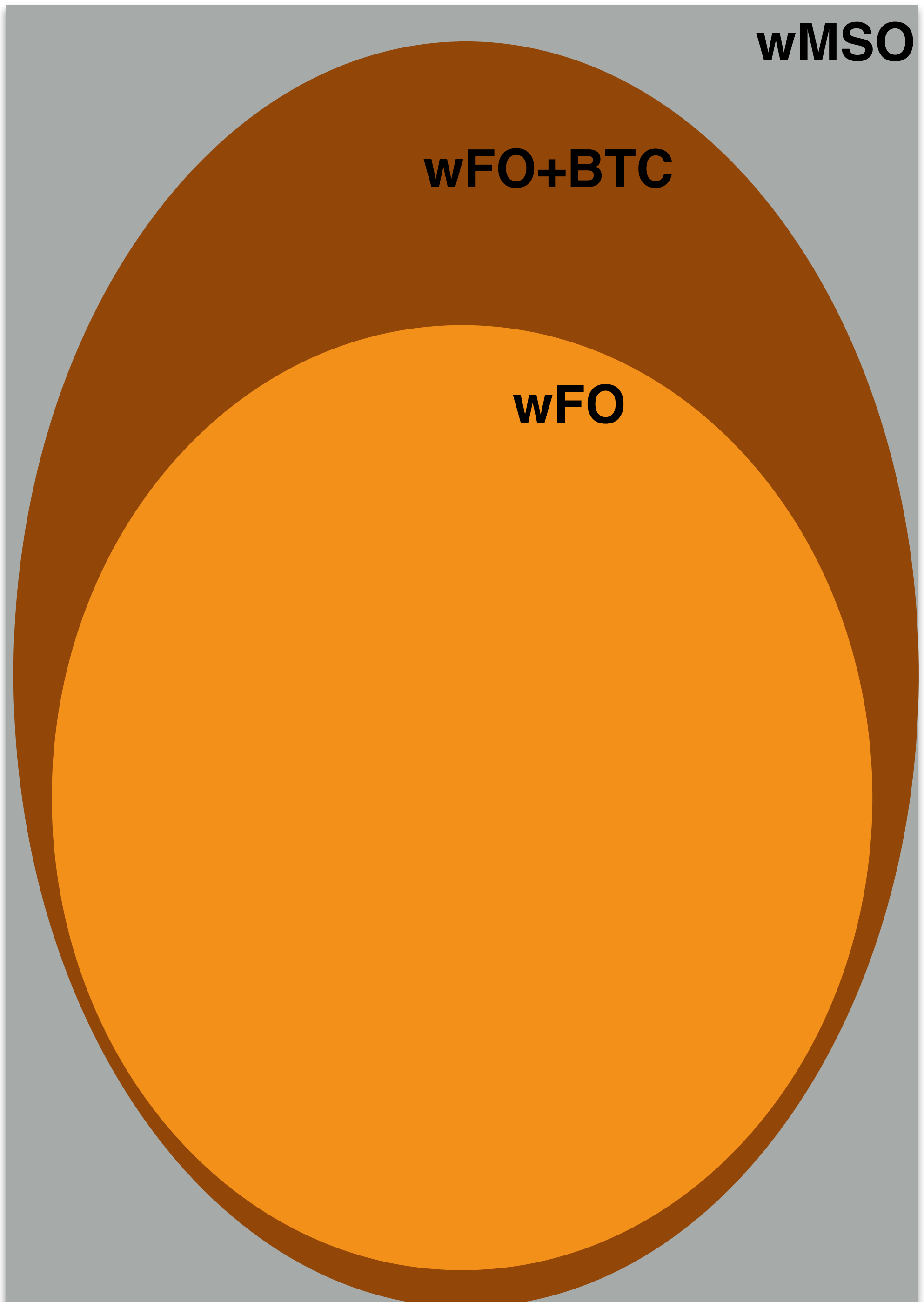
Summary

wMSO

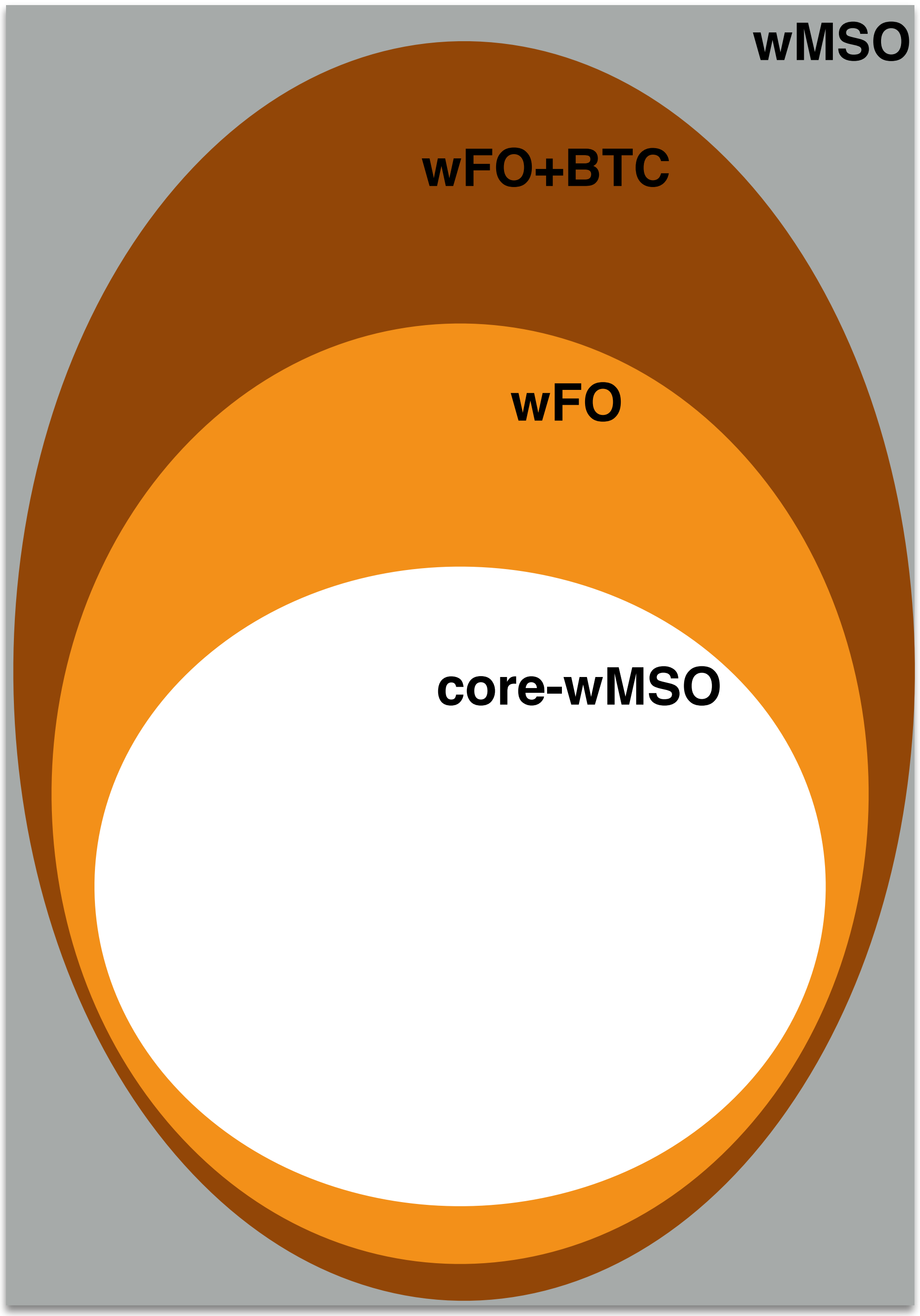
Summary



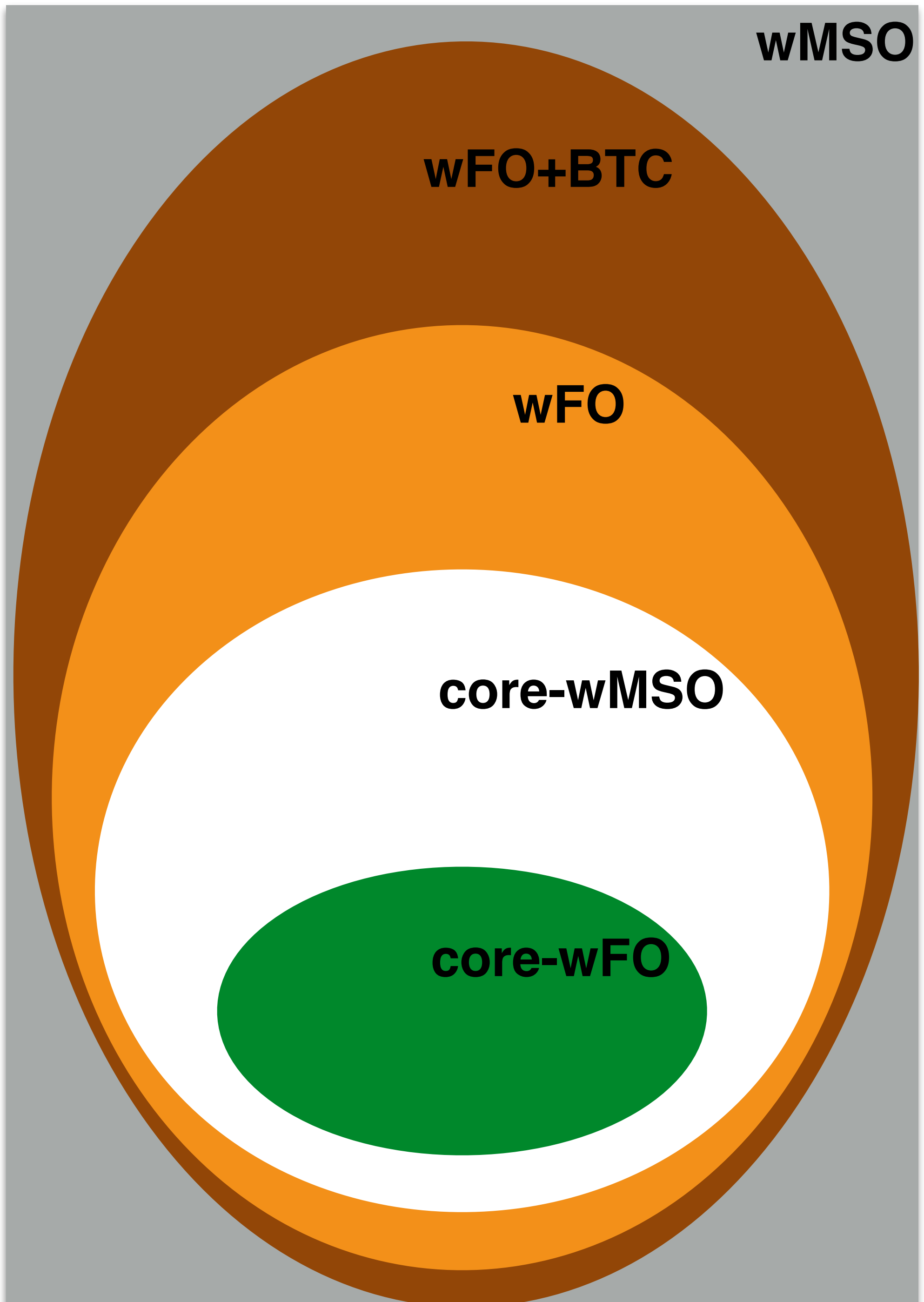
Summary



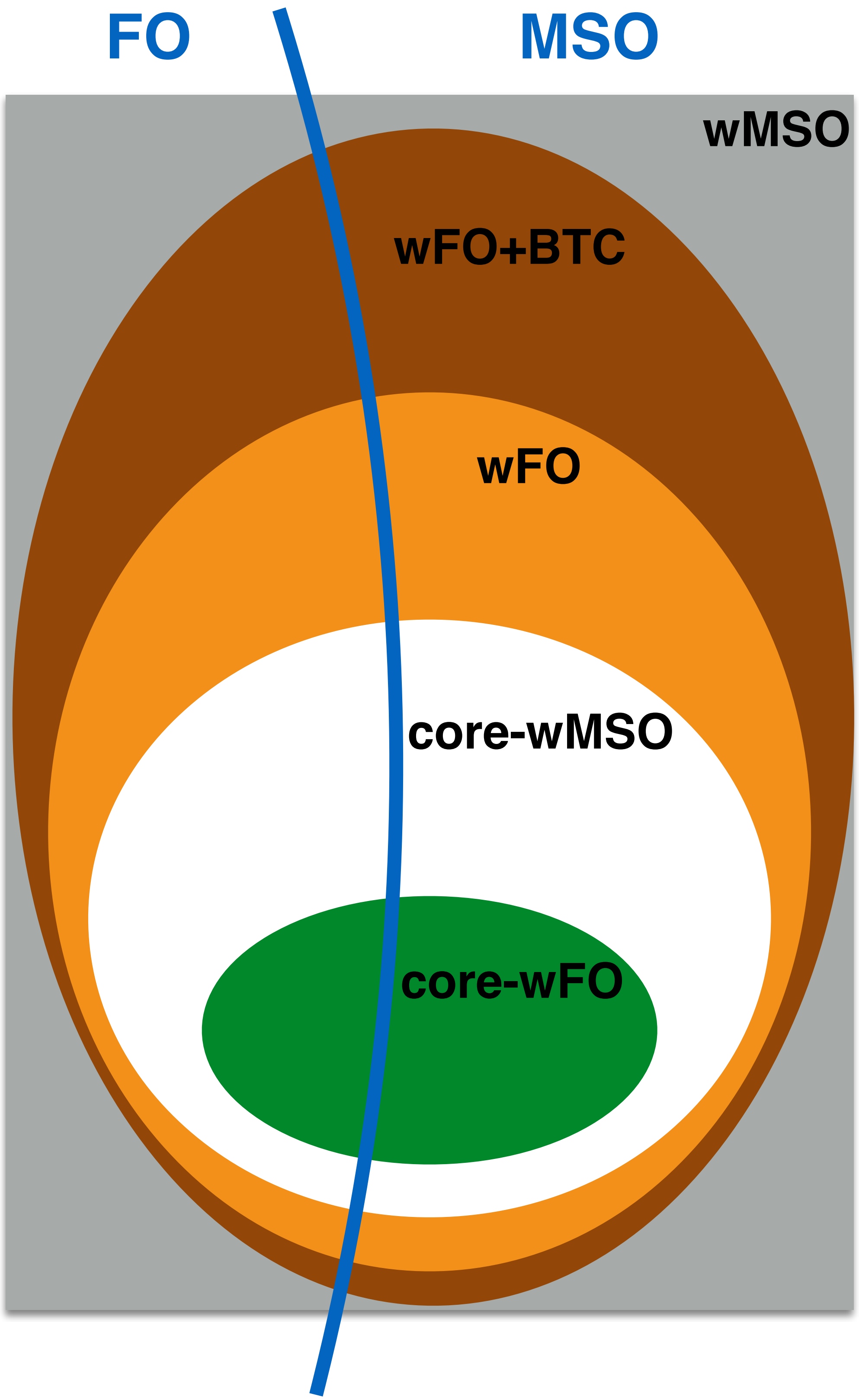
Summary



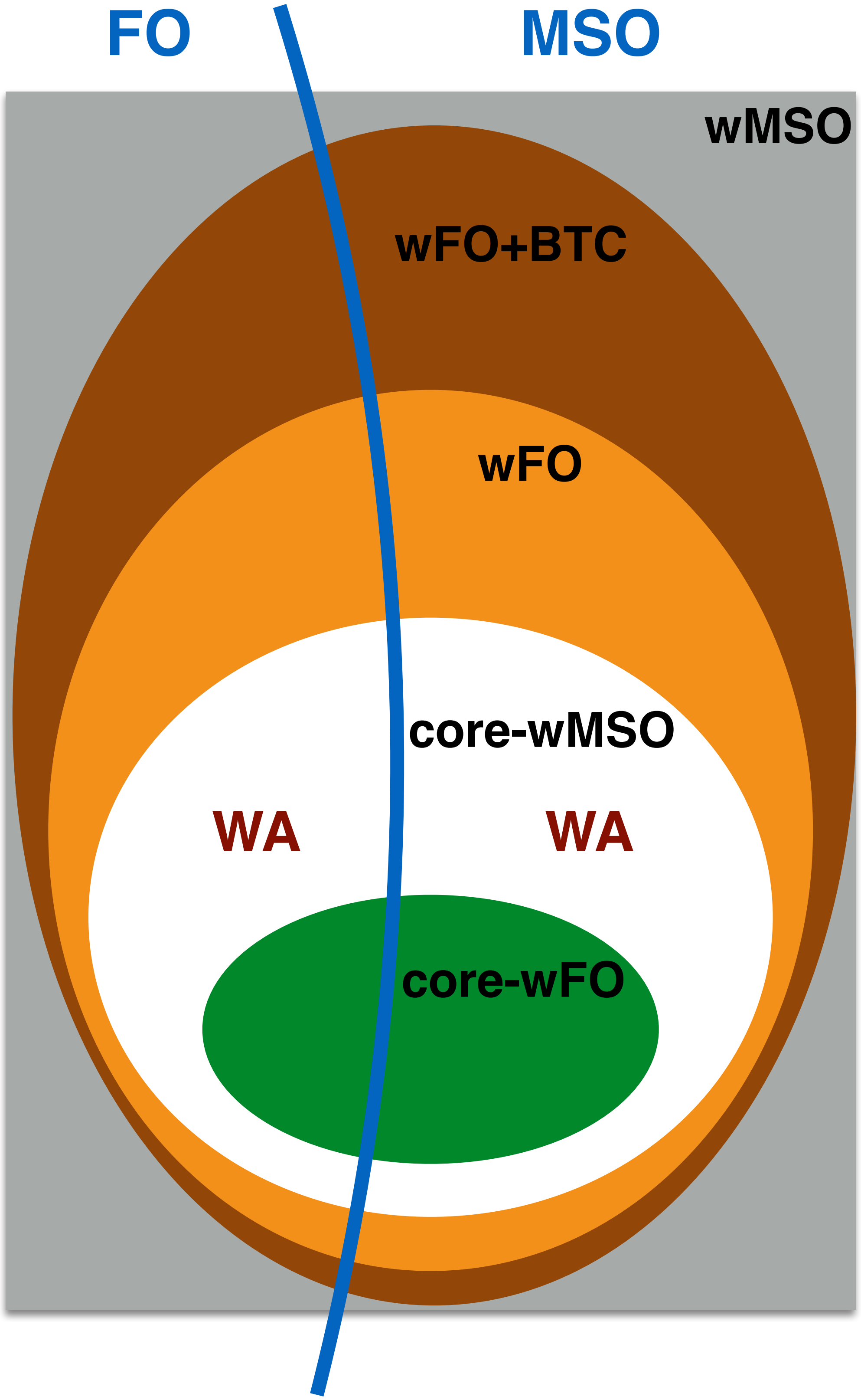
Summary



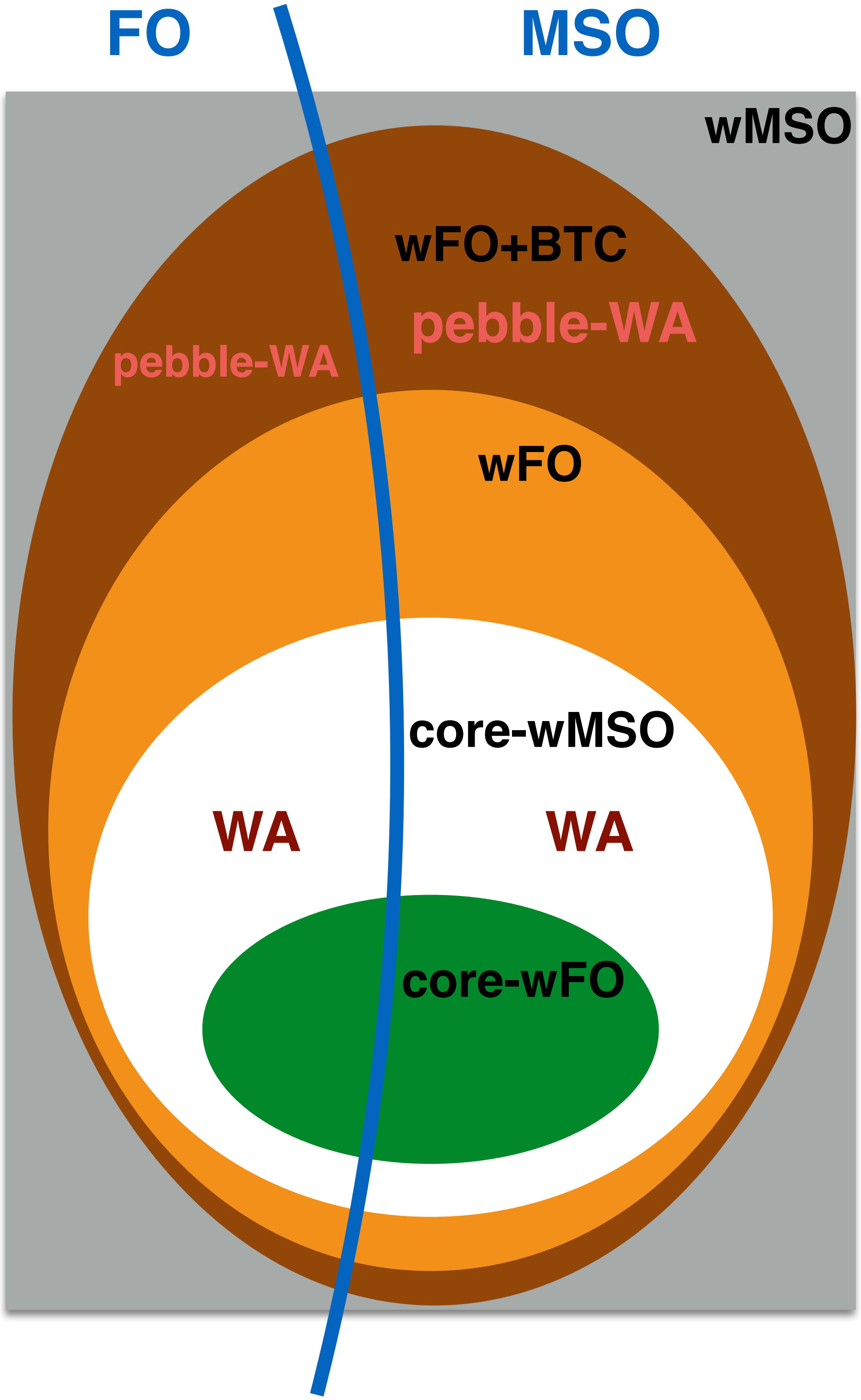
Summary



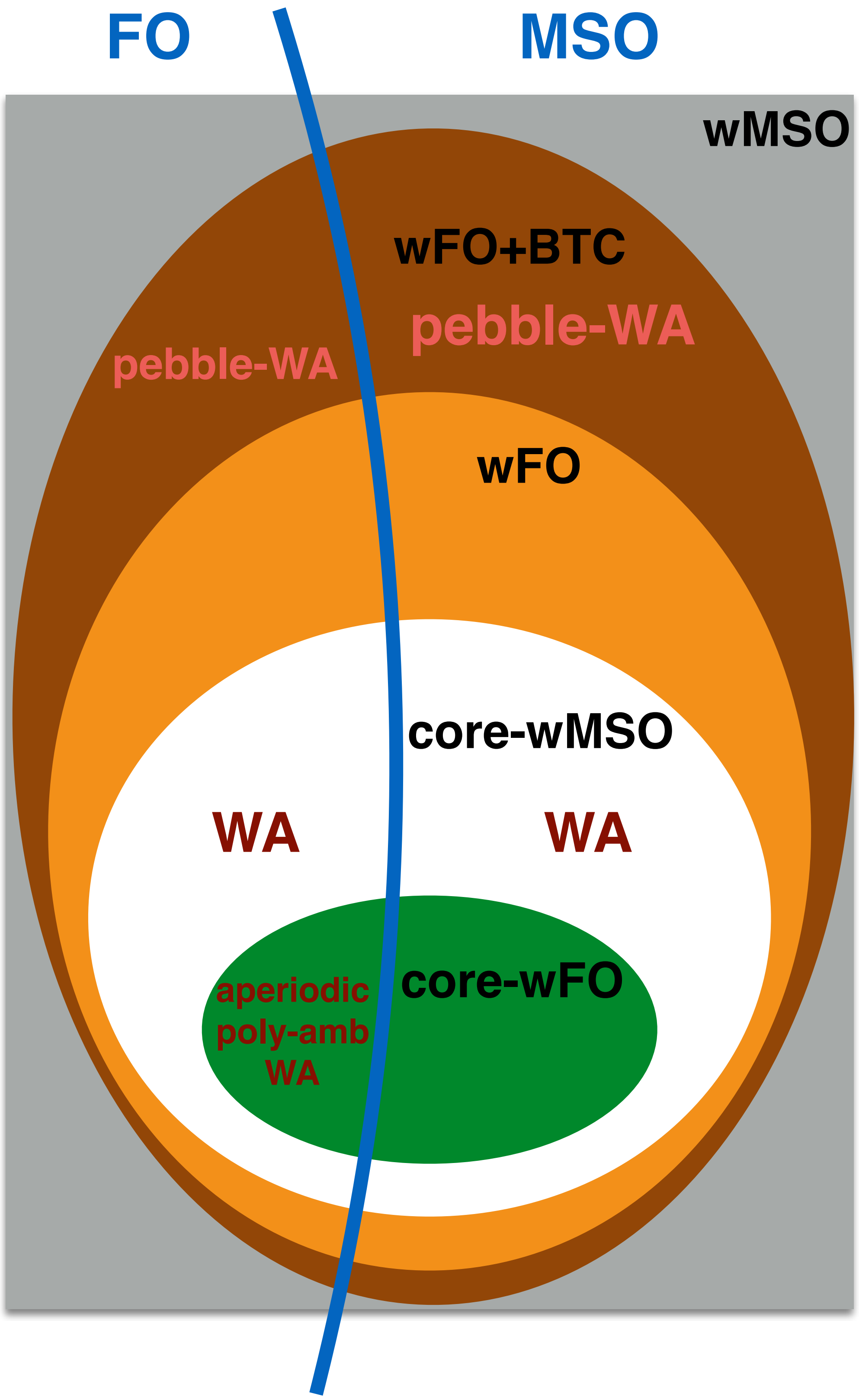
Summary



Summary

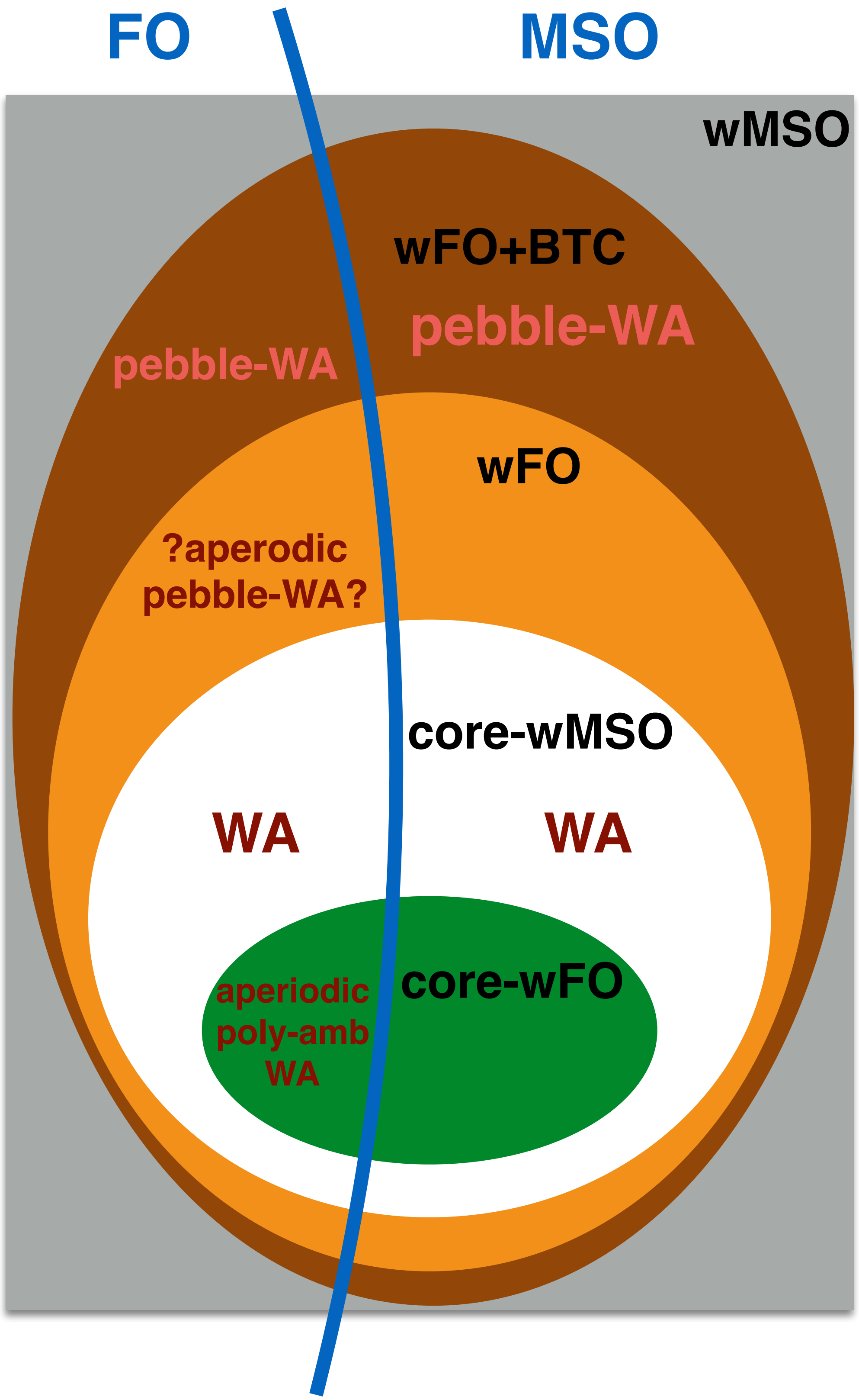

FO
MSO
wMSO
wFO+BTC
pebble-WA
pebble-WA
wFO
core-wMSO
WA
WA
core-wFO

Summary



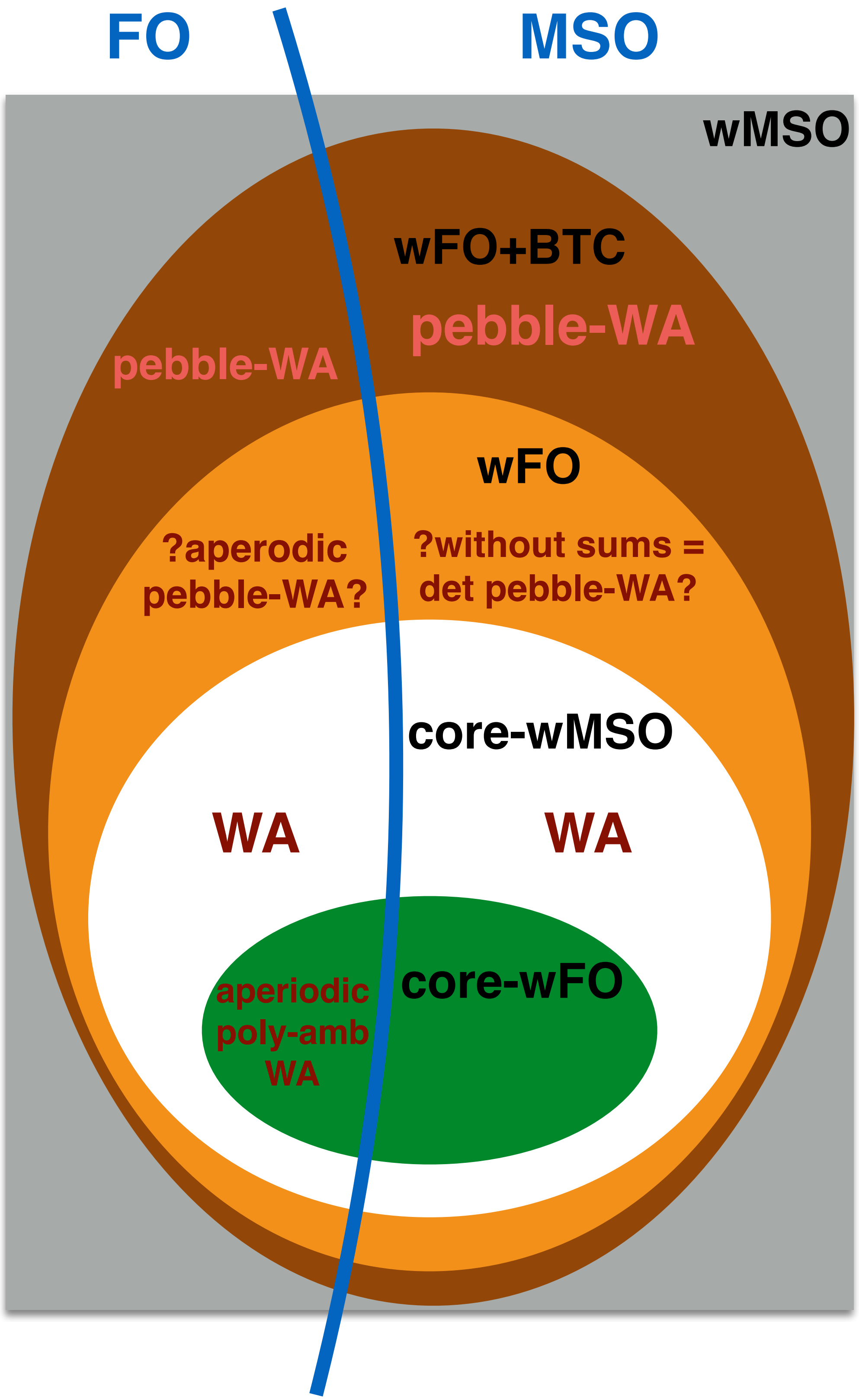
FO
MSO
wMSO
wFO+BTC
pebble-WA
pebble-WA
wFO
core-wMSO
WA
WA
core-wFO
aperiodic
poly-amb
WA

Summary



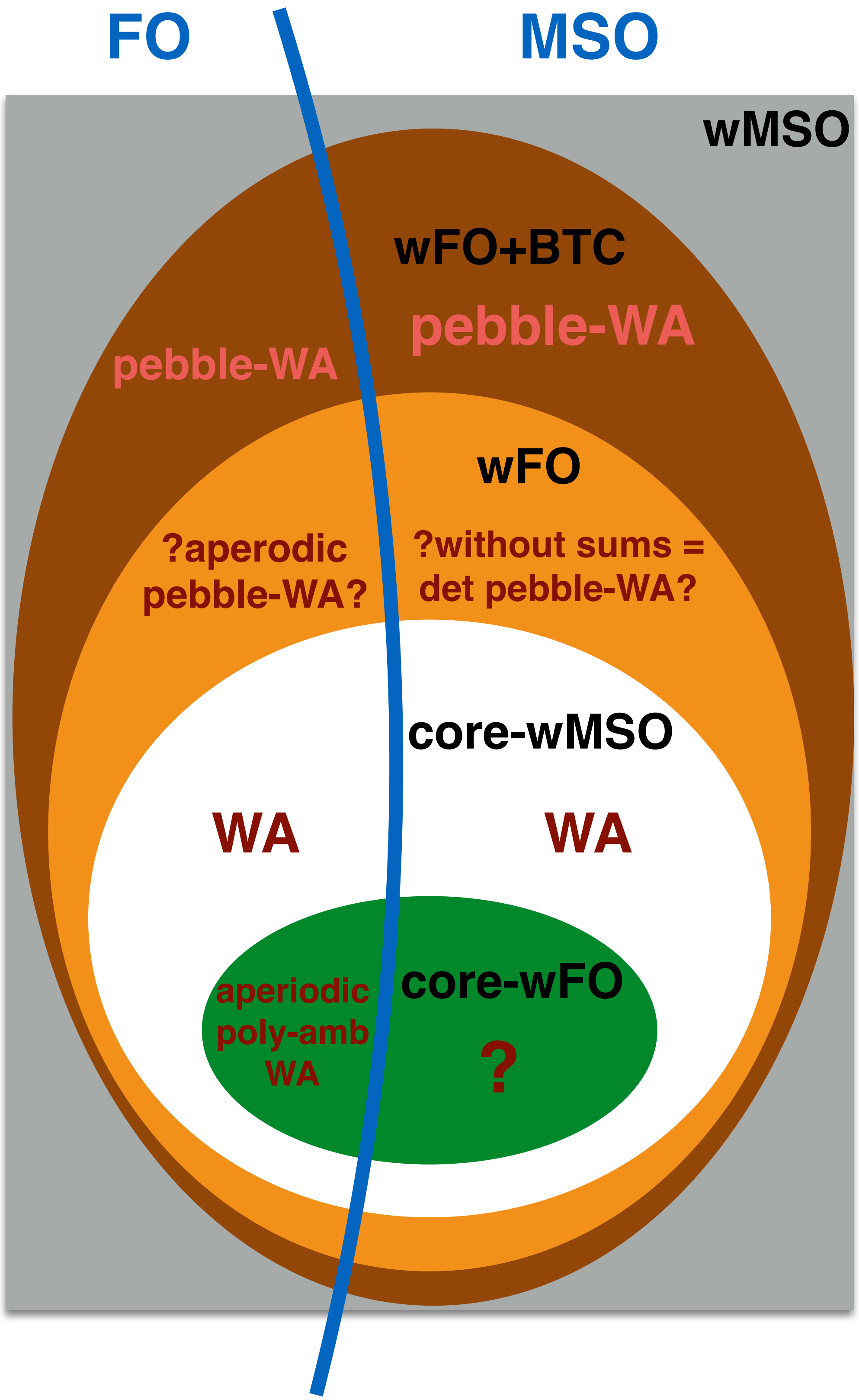
FO
MSO
wMSO
wFO+BTC
pebble-WA
pebble-WA
wFO
?aperodic
pebble-WA?
core-wMSO
WA
WA
core-wFO
aperiodic
poly-amb
WA

Summary



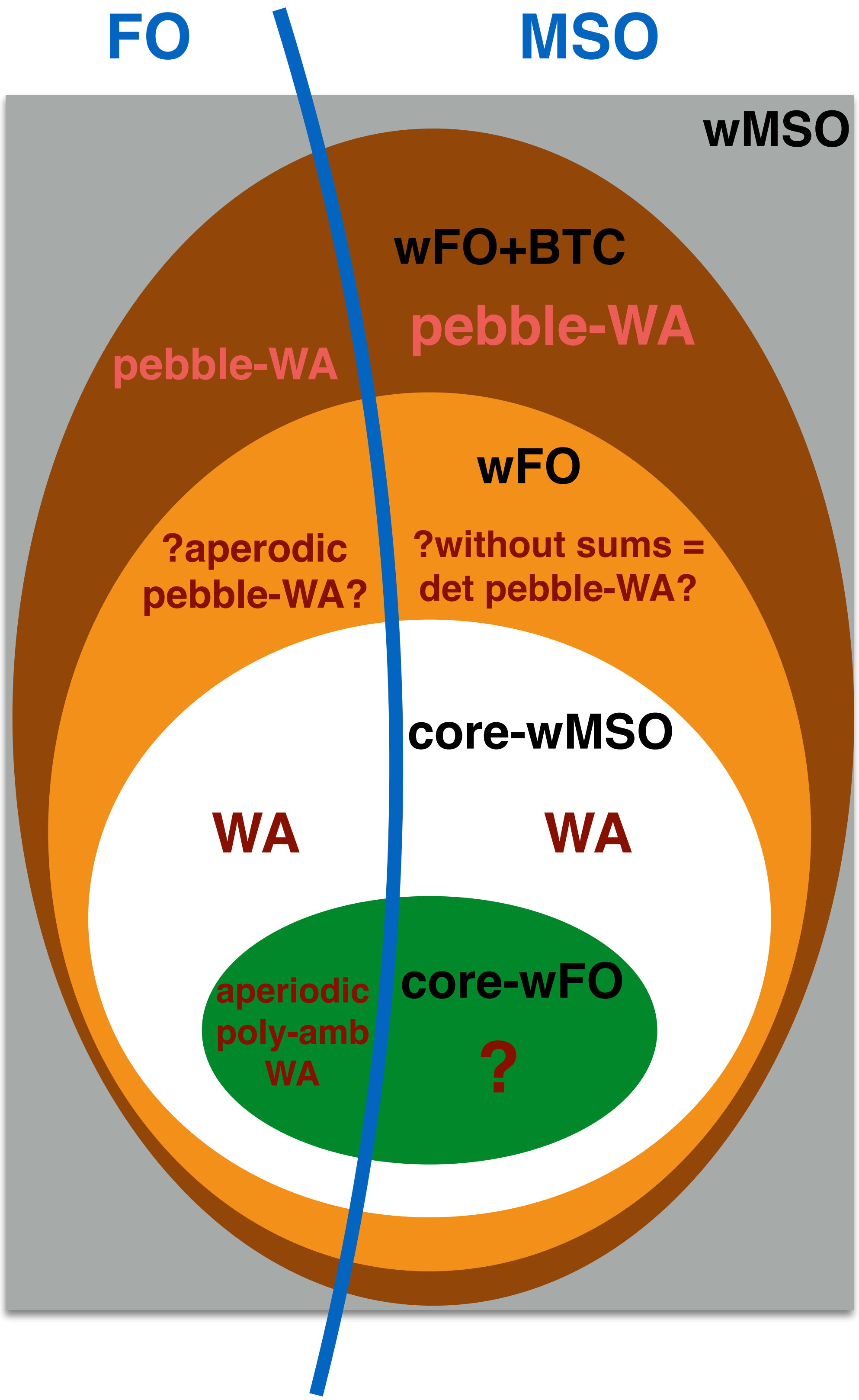
FO
MSO
wMSO
wFO+BTC
pebble-WA
pebble-WA
wFO
?aperodic
pebble-WA?
?without sums =
det pebble-WA?
core-wMSO
WA
WA
aperiodic
poly-amb
WA
core-wFO

Summary



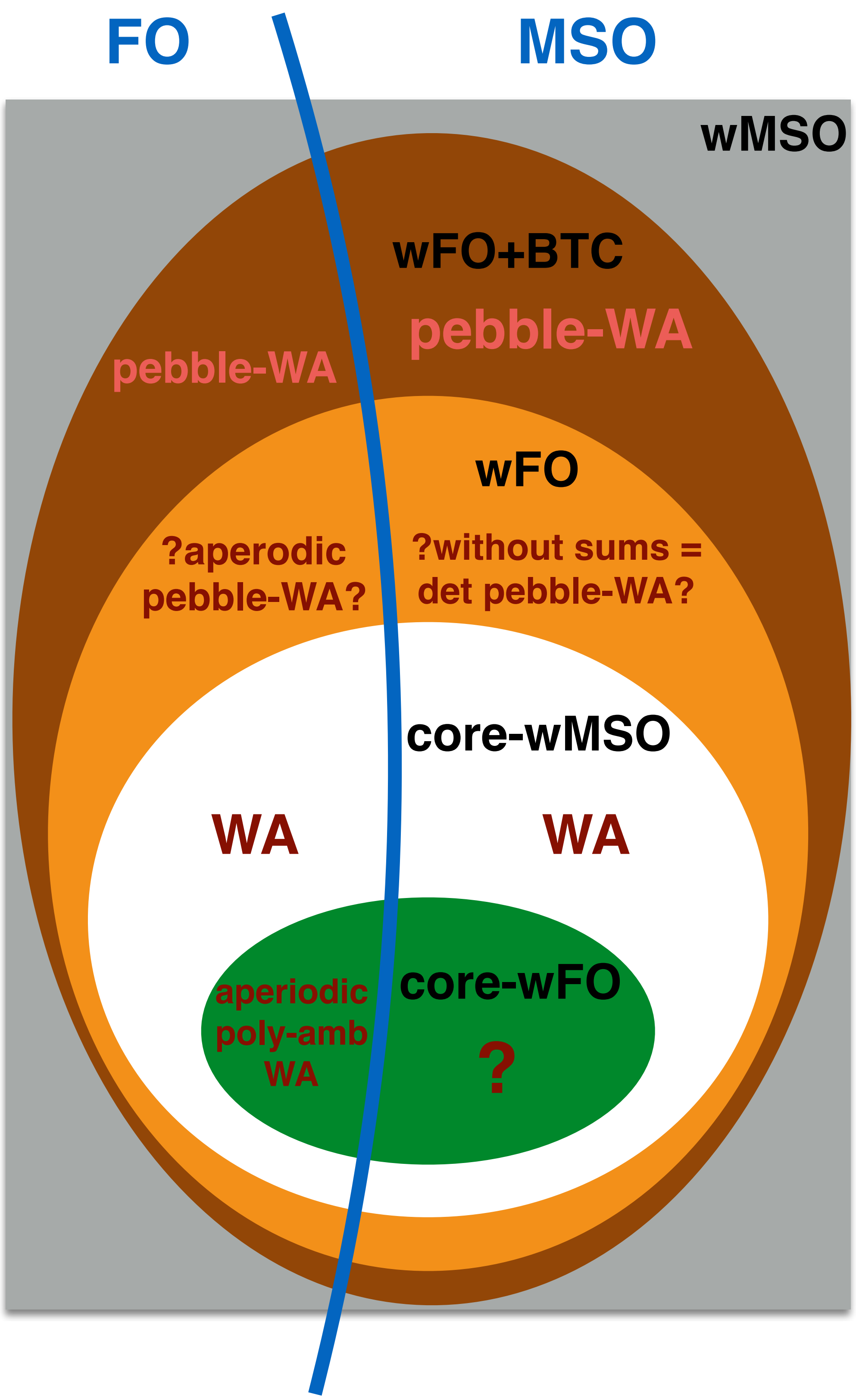
FO
MSO
wMSO
wFO+BTC
pebble-WA
pebble-WA
wFO
?aperodic
pebble-WA?
?without sums =
det pebble-WA?
core-wMSO
WA
WA
aperiodic
poly-amb
WA
core-wFO
?

Summary



Equivalences between **logics** and **automata**

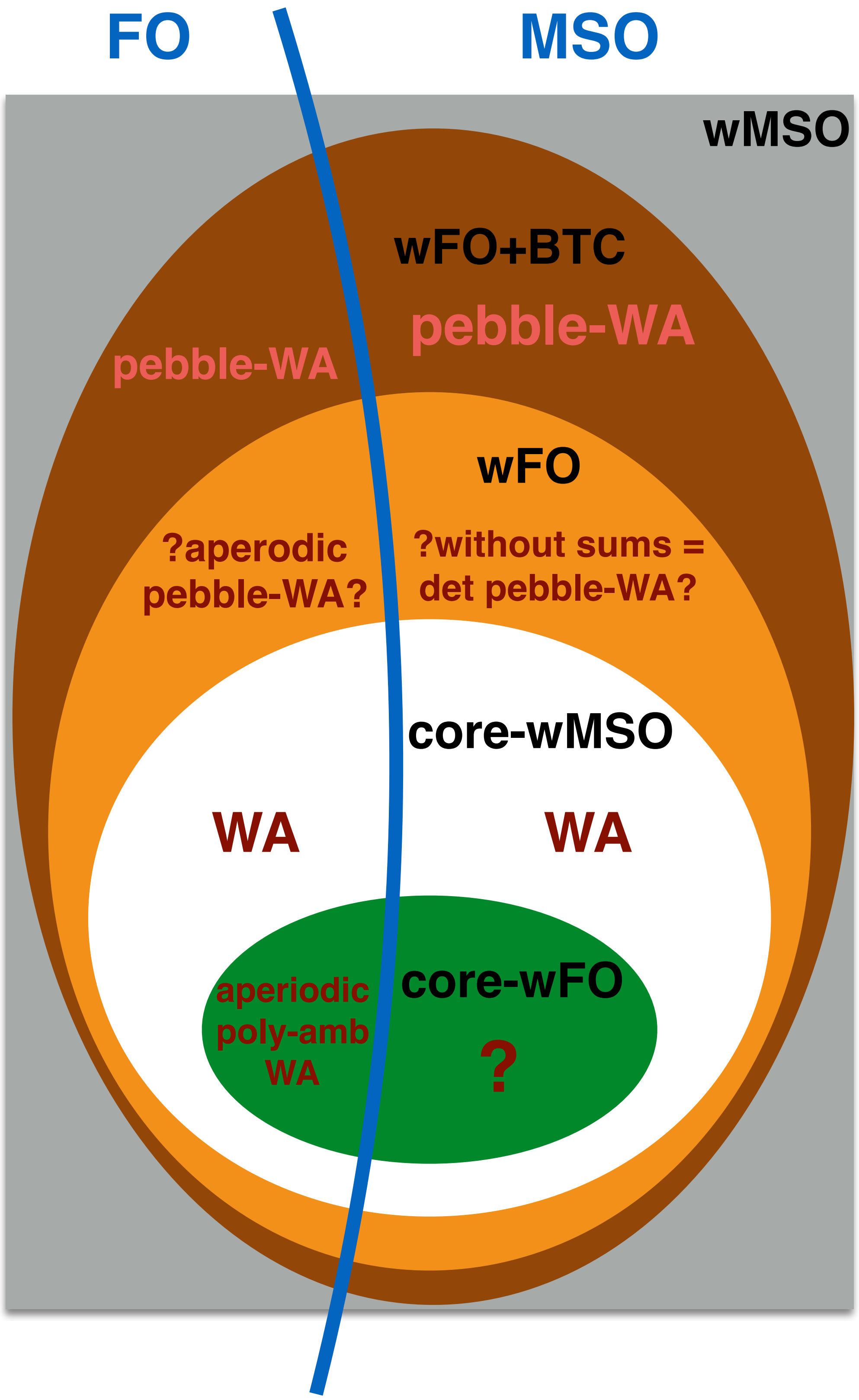
Summary



Equivalences between **logics** and **automata**

Evaluation in $\mathcal{O}(|\varphi| . |w|^{\#\text{vars}})$

Summary

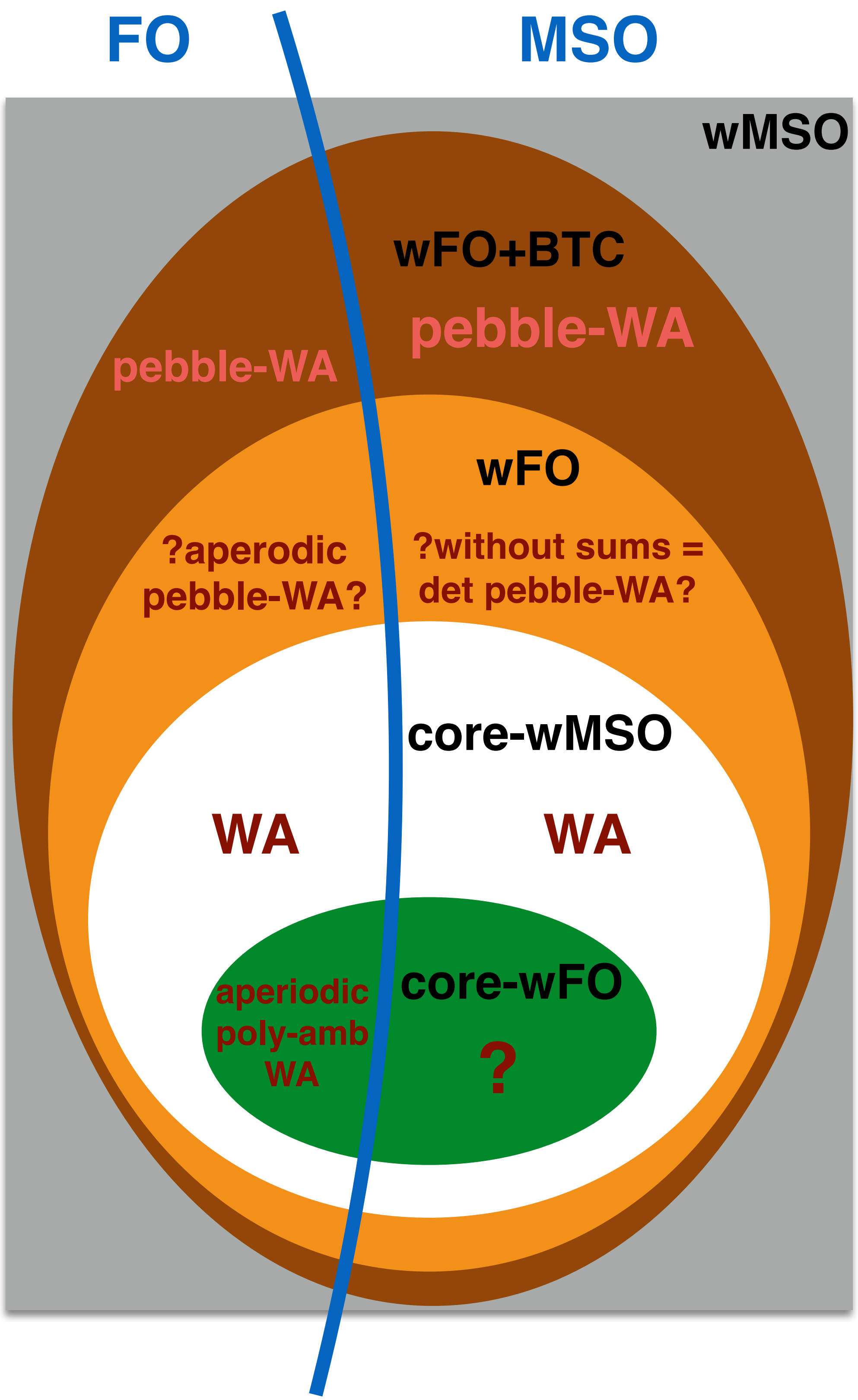


Equivalences between **logics** and **automata**

Evaluation in $\mathcal{O}(|\varphi|.|w|^{\#\text{vars}})$

Decidability procedures:
equivalence,
minimisation,
boundedness

Summary



Equivalences between **logics** and **automata**

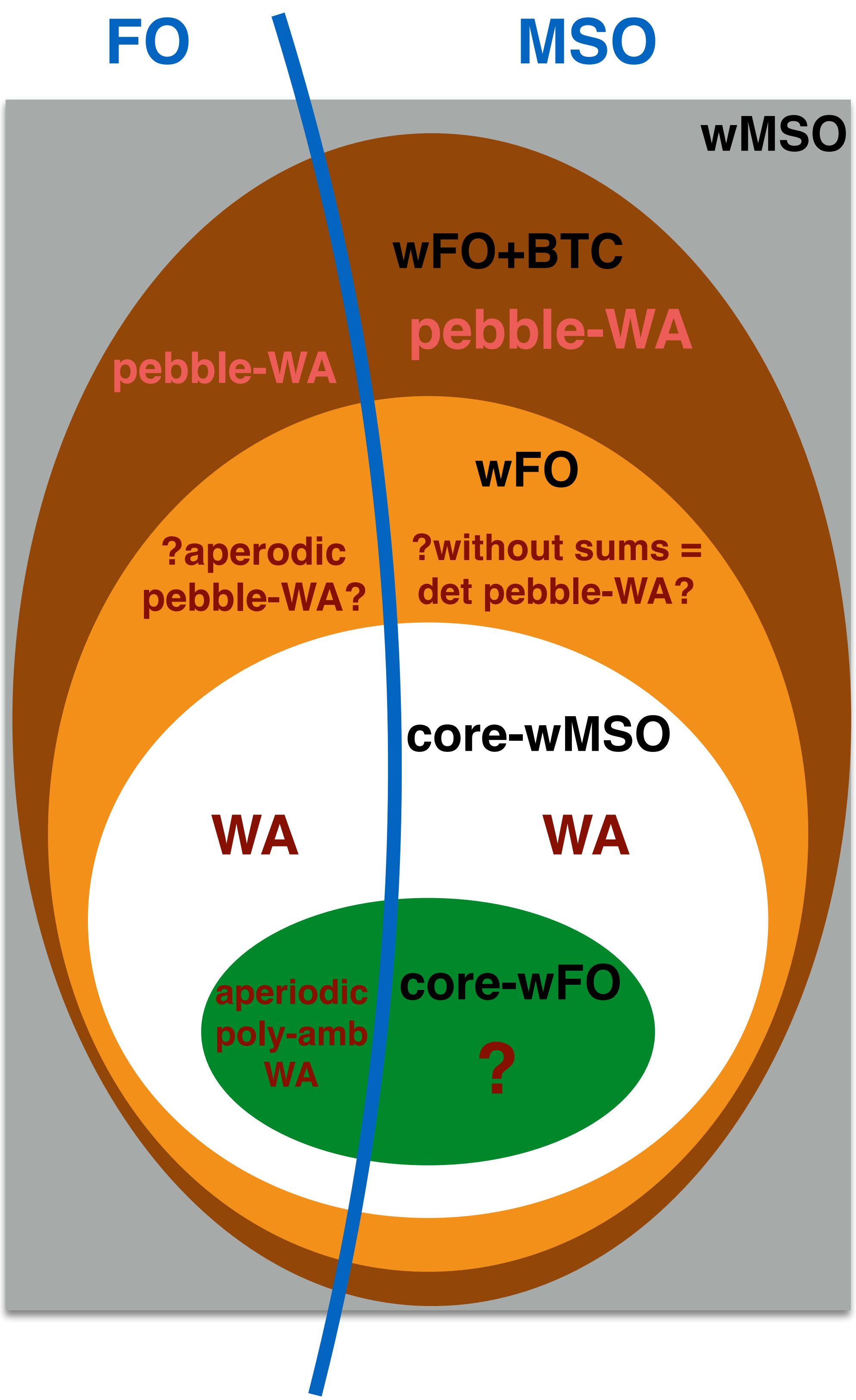
Evaluation in $\mathcal{O}(|\varphi|.|w|^{\#\text{vars}})$

Independant of weight structures

Decidability procedures: equivalence, minimisation, boundedness

Very dependant of weight structures

Summary



Equivalences between **logics** and **automata**

Evaluation in $\mathcal{O}(|\varphi|.|w|^{\#\text{vars}})$

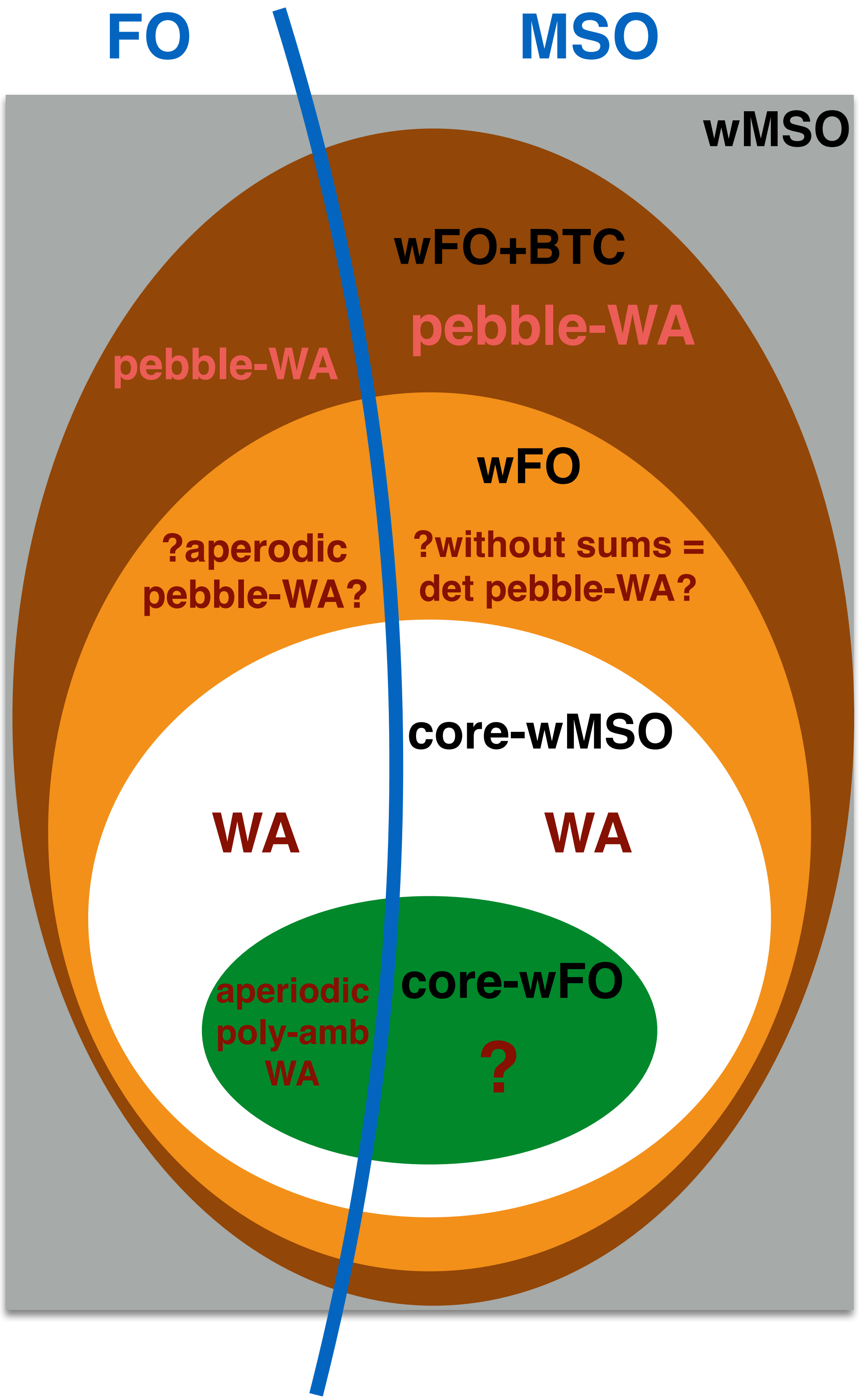
Independant of weight structures

Link with register models?
[Douéneau&Filiot&Gastin 2018]
marbles/invisible-pebbles: fragments of logic?

Decidability procedures:
equivalence,
minimisation,
boundedness

Very dependant of weight structures

Summary



Equivalences between **logics** and **automata**

Evaluation in $\mathcal{O}(|\varphi|.|w|^{\#vars})$

Independant of weight structures

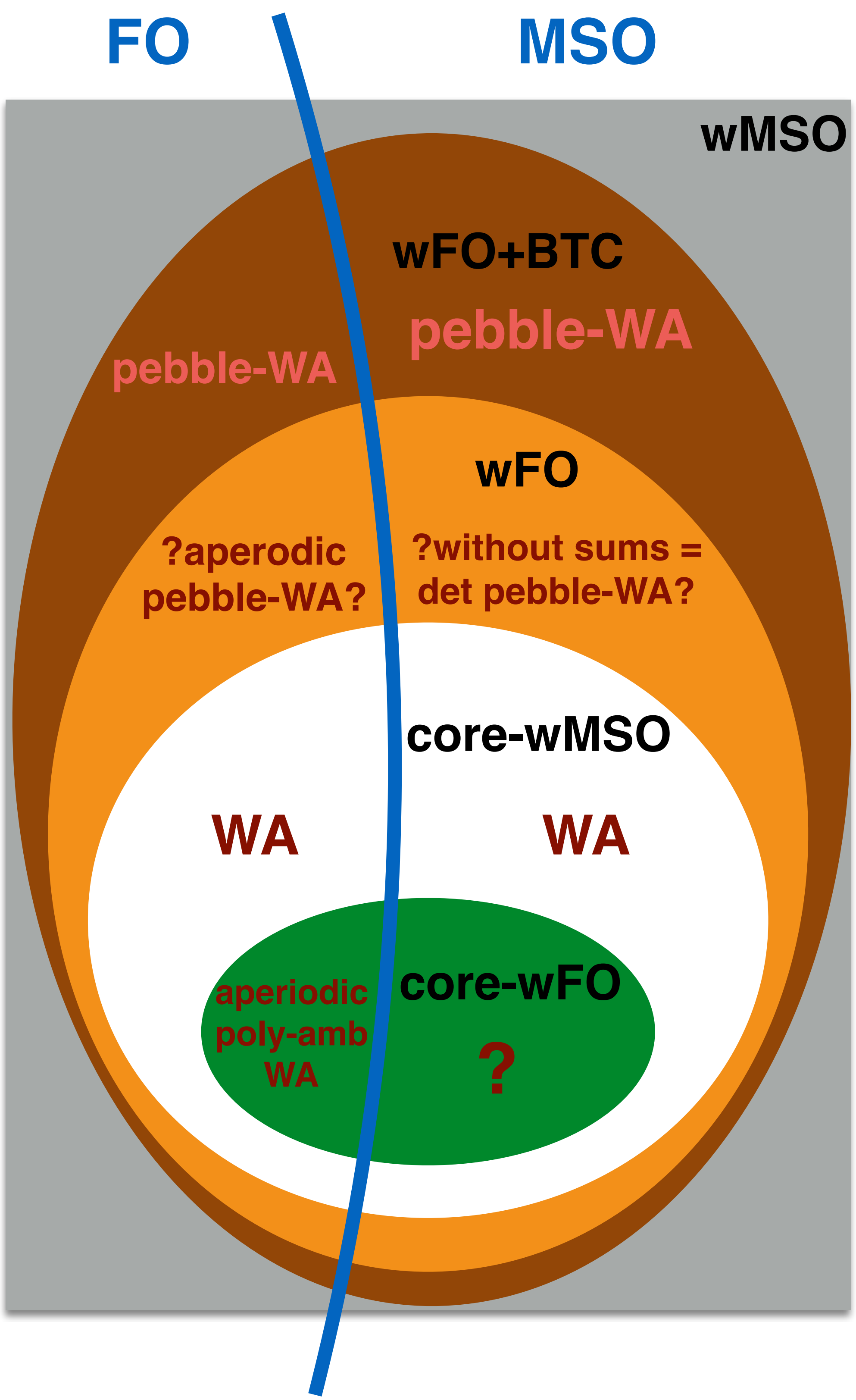
Link with register models?
[Douéneau&Filiot&Gastin 2018]
marbles/invisible-pebbles: fragments of logic?

Decidability procedures:
equivalence,
minimisation,
boundedness

Very dependant of weight structures

2-way→1-way? EXPSPACE for functional transducers
[Filiot&Gauwin&Reynier&Servais 2013,
Baschenis&Gauwin&Muscholl&Puppis 2017+Jecker 2018]
partially-commutative weight structure?
with/without pebbles?

Summary



Equivalences between **logics** and **automata**

Evaluation in $\mathcal{O}(|\varphi|.|w|^{\#\text{vars}})$

Independant of weight structures

Link with register models?
[Douéneau&Filiot&Gastin 2018]
marbles/invisible-pebbles: fragments of logic?

Decidability procedures:
equivalence,
minimisation,
boundedness

Very dependant of weight structures

2-way→1-way? EXPSPACE for functional transducers
[Filiot&Gauwin&Reynier&Servais 2013,
Baschenis&Gauwin&Muscholl&Puppis 2017+Jecker 2018]
partially-commutative weight structure?
with/without pebbles?

Thank you!