A probabilistic Kleene Theorem

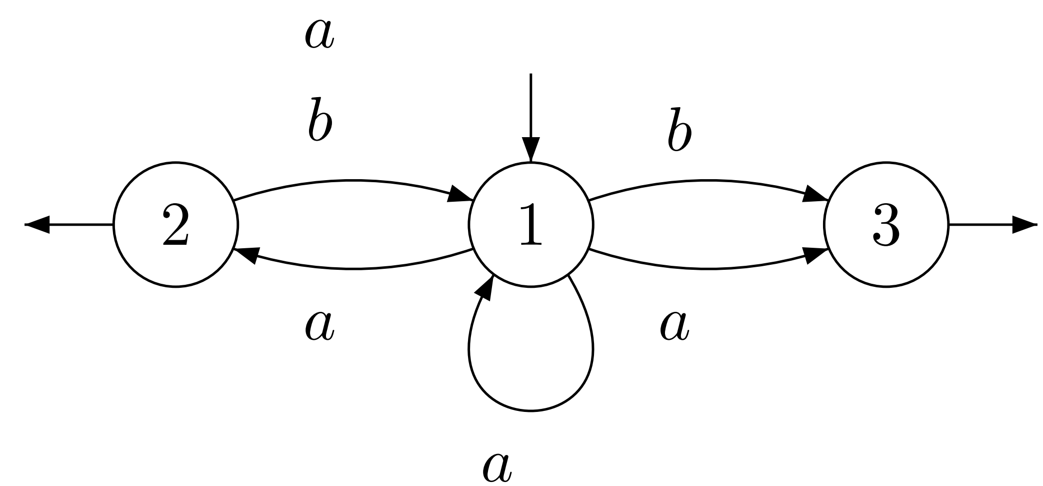
Benjamin Monmege
LSV, ENS Cachan, CNRS, France

MOVEP 2012, Marseille

Part of works published at ATVA'12 with Benedikt Bollig, Paul Gastin and Marc Zeitoun

Kleene's Theorem

Finite State Automata



same expressivity

Regular Expressions

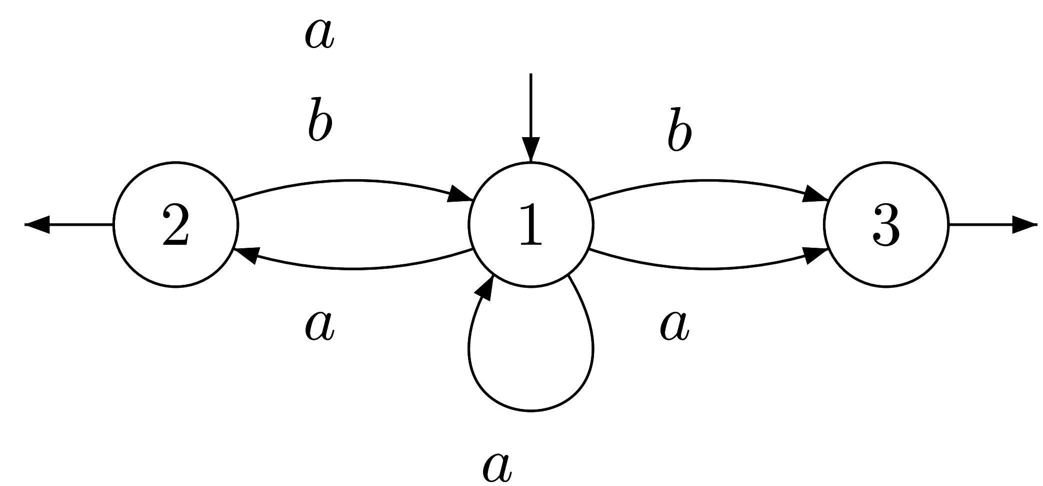
E ::= a | E+E | E · E | E*

Motivations

- Theoretically: relate denotational and computational models
- Practically: easier to write specifications using regular expressions vs. easier to check properties (emptiness, inclusion...) with automata
- Goal: translate expressions to automata, as efficiently as possible

Kleene's Theorem

Finite State Automata



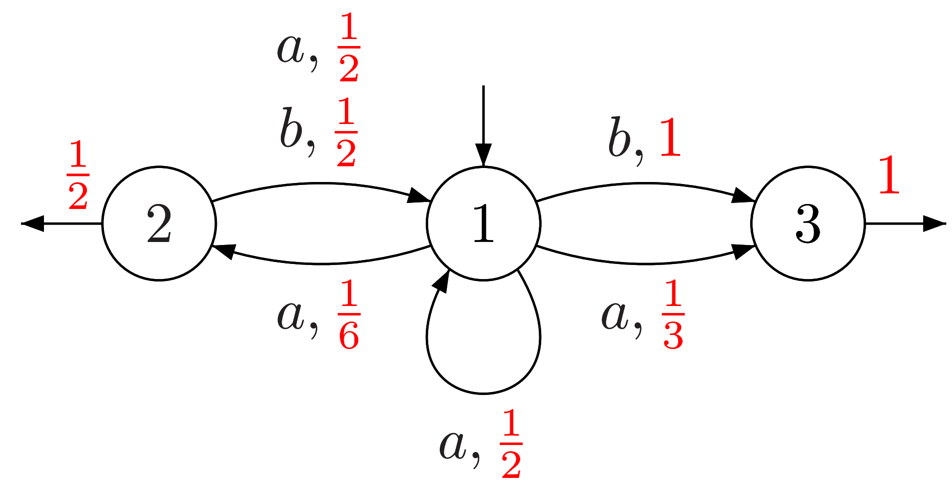
same expressivity

Regular Expressions

$$E ::= \quad a \mid E+E \mid E \cdot E \mid E^*$$

Kleene's Theorem

Weighted Finite State Automata



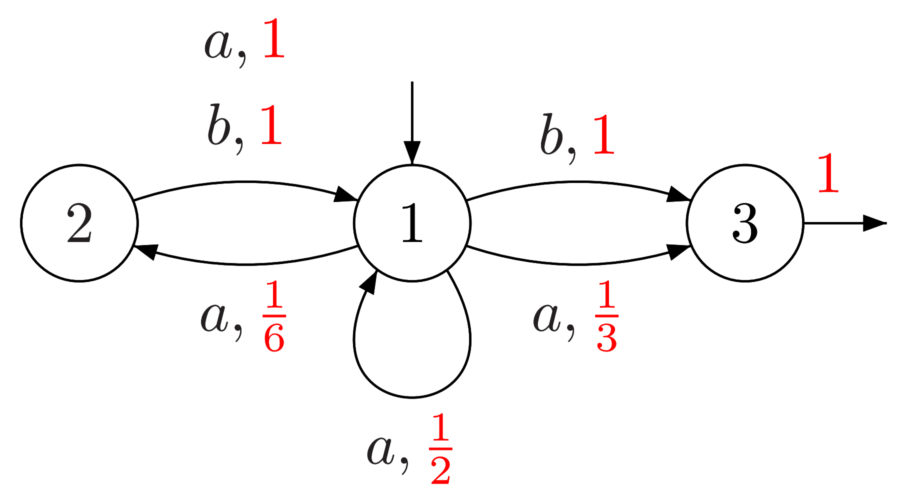
same expressivity

Regular Expressions

$E ::= \quad a \mid E+E \mid E \cdot E \mid E^*$

Kleene's Theorem

Weighted Finite State Automata



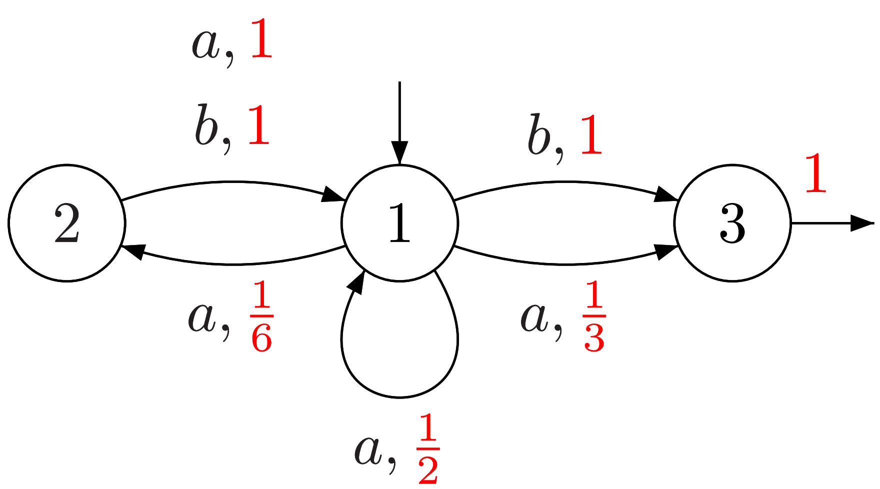
same expressivity ↕

Regular Expressions

$$E ::= \quad a \mid E+E \mid E \cdot E \mid E^*$$

Kleene's Theorem

Weighted Finite State Automata



$\Sigma^* \rightarrow \mathbb{R}$

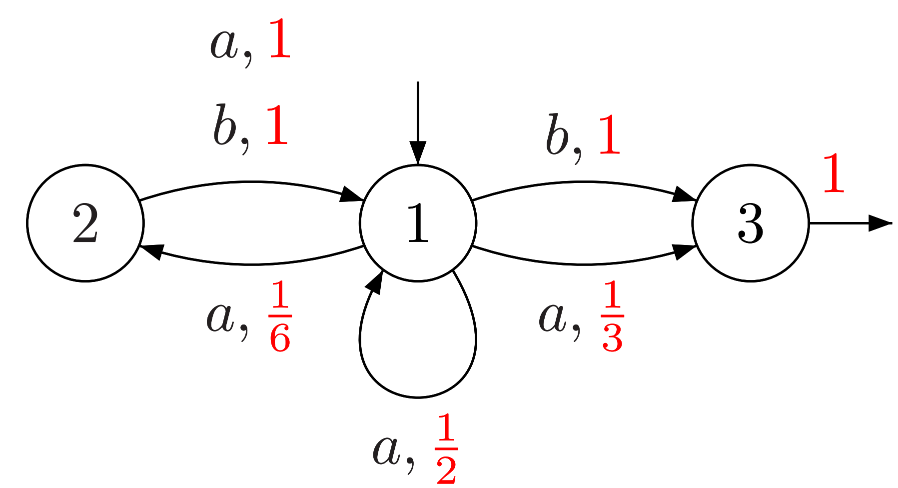
same expressivity

Regular Expressions

E ::= *a* | E+E | E · E | E*

Kleene's Theorem

Weighted Finite State Automata



$\Sigma^* \rightarrow \mathbb{R}$

same expressivity

a a b

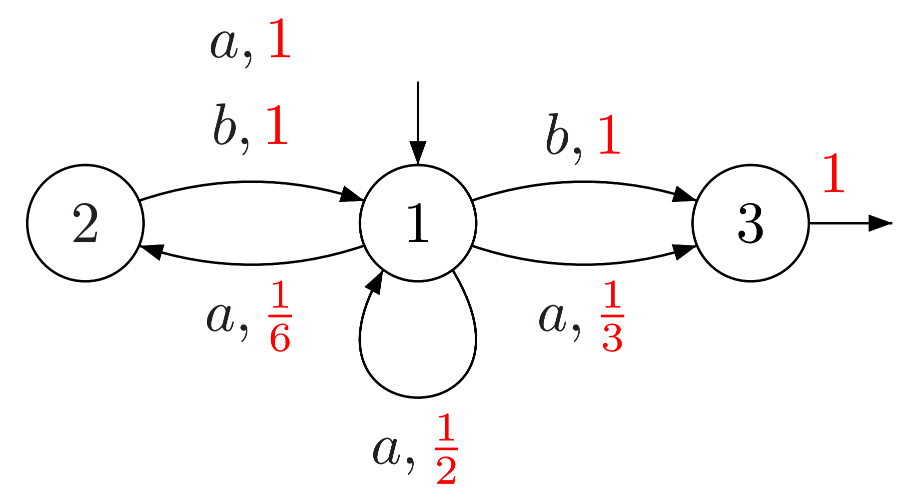
two runs:
1→2→1→3 of weight 1/6x1x1x1=1/6
and 1→1→1→3 of weight 1/2x1/2x1x1=1/4

Regular Expressions

$$E ::= \quad a \mid E+E \mid E \cdot E \mid E^*$$

Kleene's Theorem

Weighted Finite State Automata



$\Sigma^* \to \mathbb{R}$

same expressivity

a a b

two runs:

1→2→1→3 of weight 1/6x1x1x1=1/6

and 1→1→1→3 of weight 1/2x1/2x1x1=1/4

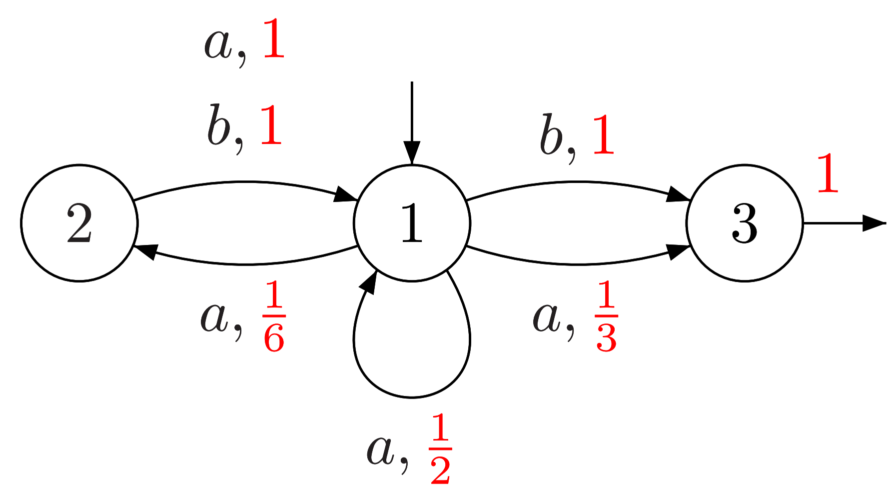
hence, *a a b* recognized with weight 1/6+1/4=5/12

Regular Expressions

$E ::= \quad a \mid E+E \mid E \cdot E \mid E^*$

Kleene's Theorem

Weighted Finite State Automata



$\Sigma^* \rightarrow \mathbb{R}$

same expressivity

a a b

two runs:

1→2→1→3 of weight 1/6x1x1x1=1/6

and 1→1→1→3 of weight 1/2x1/2x1x1=1/4

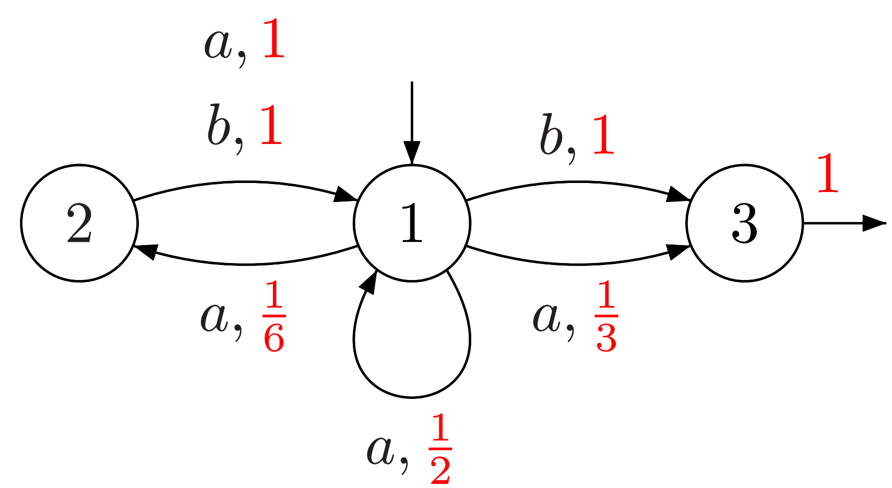
hence, *a a b* recognized with weight 1/6+1/4=5/12

Weighted Regular Expressions

E ::= *p* | *a* | E+E | E · E | E* ← E proper

Schützenberger Kleene's Theorem

Weighted Finite State Automata



$\Sigma^* \rightarrow \mathbb{R}$

same expressivity

a a b

two runs:

1→2→1→3 of weight 1/6x1x1x1=1/6

and 1→1→1→3 of weight 1/2x1/2x1x1=1/4

hence, *a a b* recognized with weight 1/6+1/4=5/12

Weighted Regular Expressions

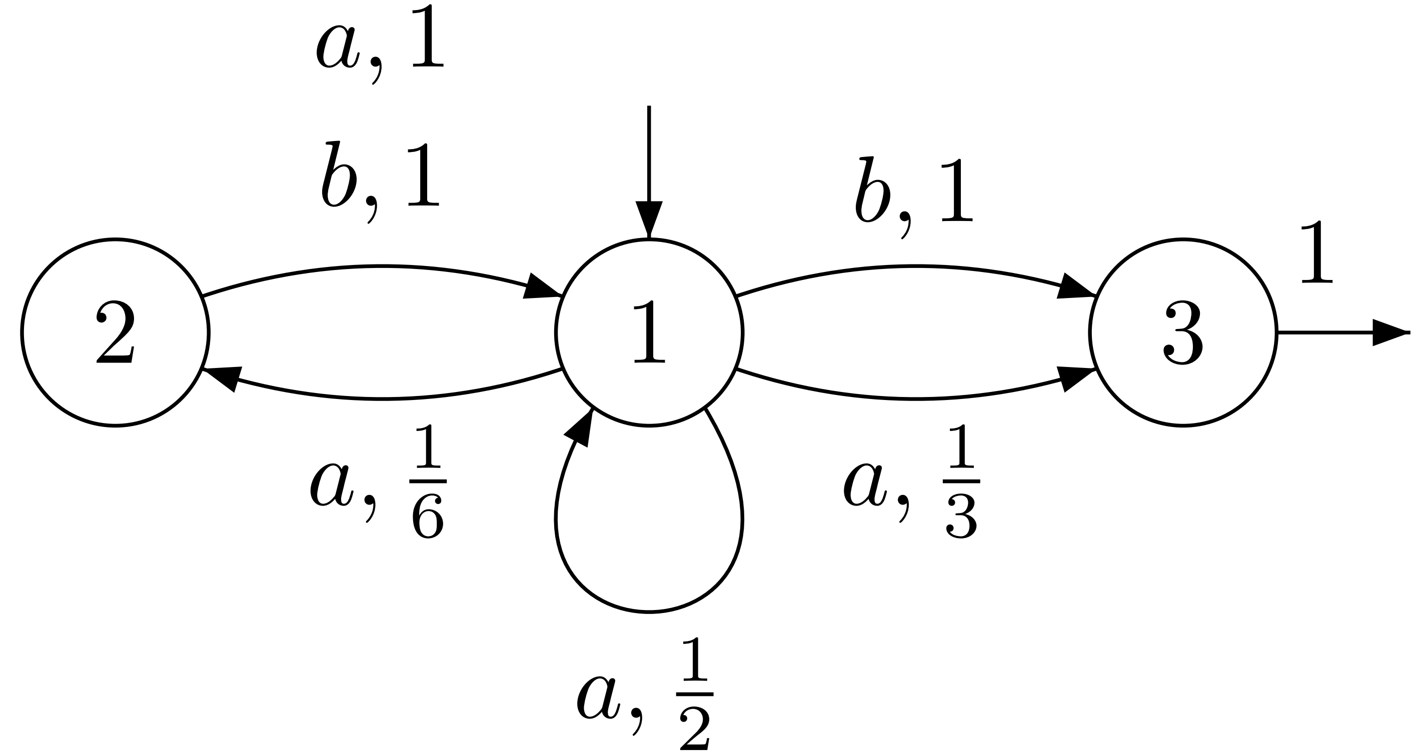
E ::= *p* | *a* | E+E | E · E | E* ← E proper

[1] S. Kleene (1956). Representation of events in nerve nets and finite automata.

[2] M.-P. Schützenberger (1961). On the Definition of a Family of Automata. Information and Control.

For an overview about Weighted Automata, see, e.g., Handbook of Weighted Automata. Editors: Manfred Droste, Werner Kuich, and Heiko Vogler. EATCS Monographs in Theoretical Computer Science. Springer, 2009.

Probabilistic case?



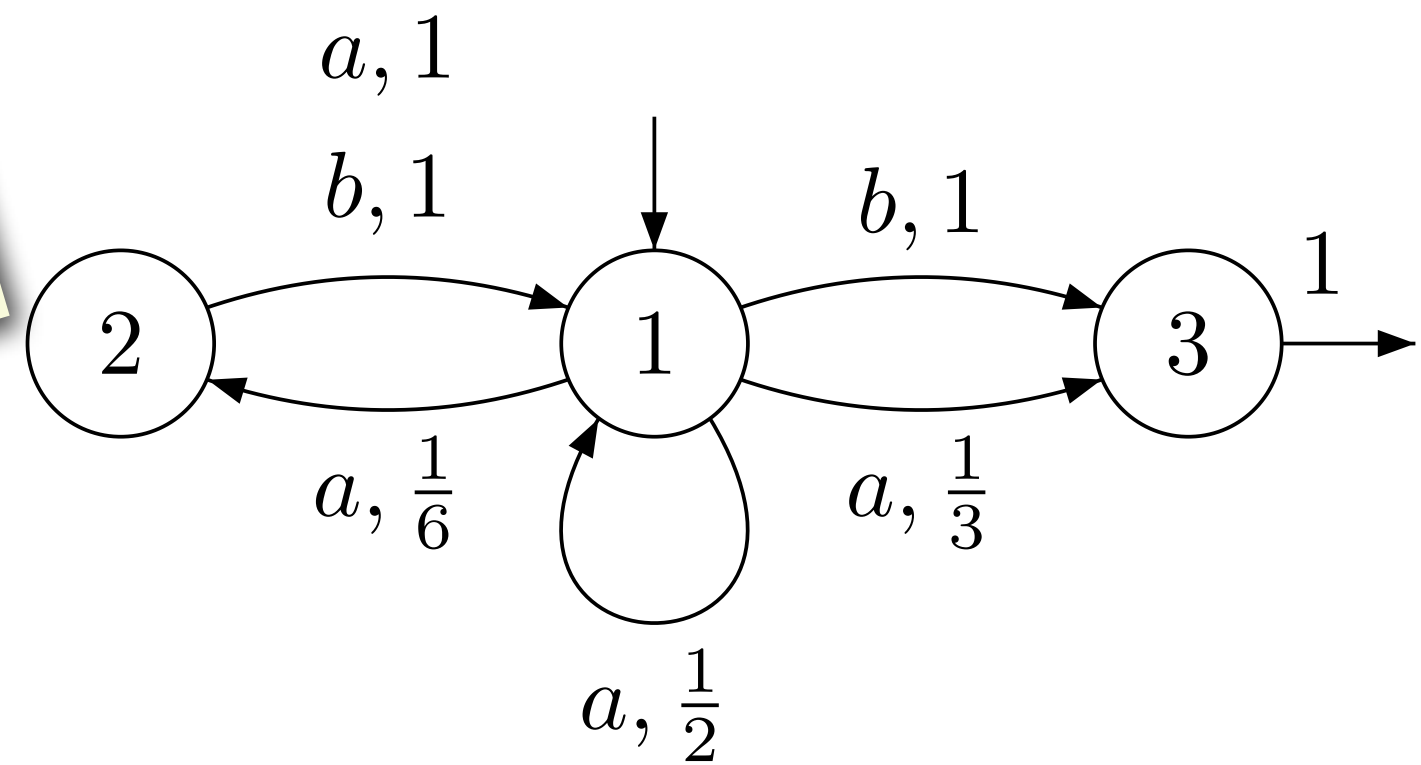
$$\mathcal{A} = (Q, \iota, Acc, \mathbb{P})$$

$$\mathbb{P} : Q \times \Sigma \times Q \to [0, 1]$$

$$Acc(q) + \sum_{q' \in Q} \mathbb{P}(q, a, q') \leq 1 \text{ for all } (q, a) \in Q \times A$$

Probabilistic case?

Reactive Probabilistic Finite Automata



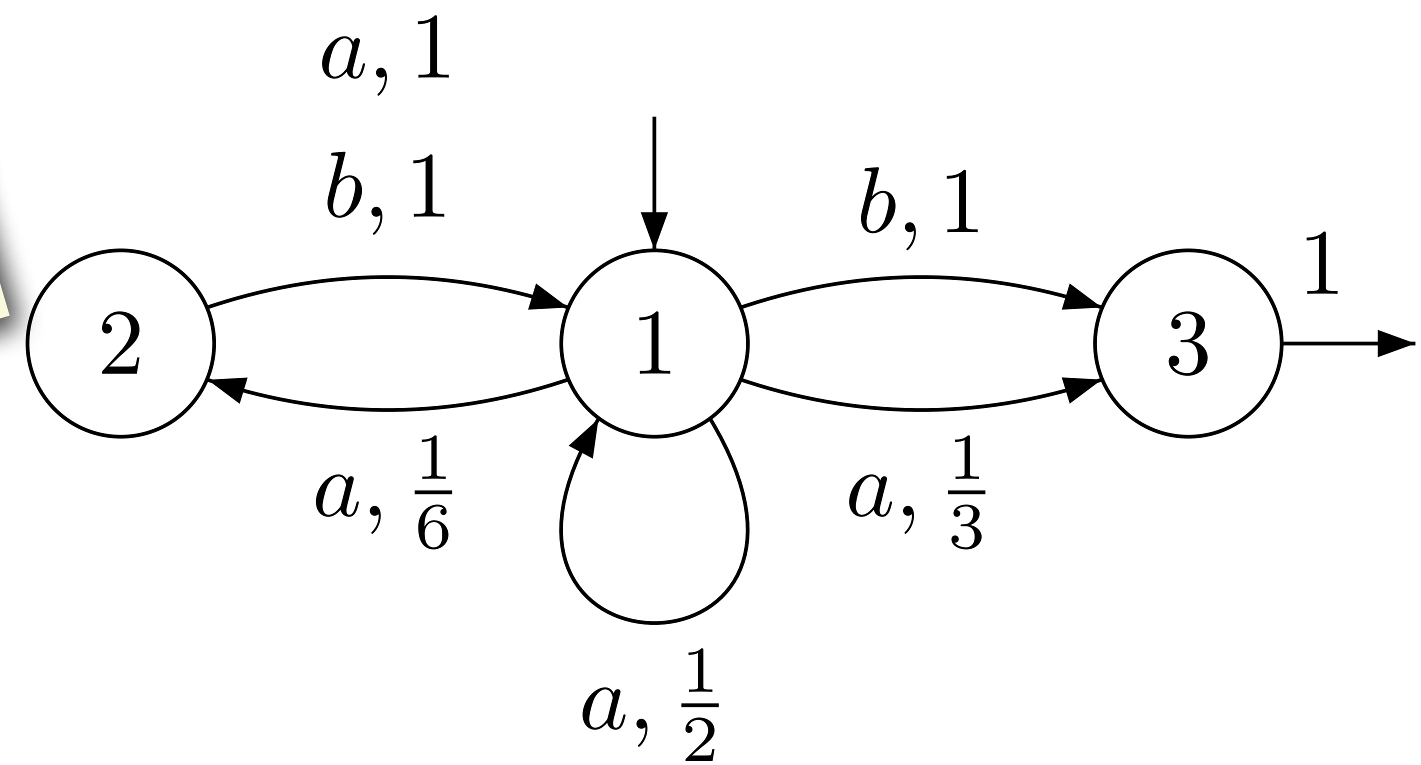
$$\mathcal{A} = (Q, \iota, Acc, \mathbb{P})$$

$$\mathbb{P} : Q \times \Sigma \times Q \to [0, 1]$$

$$Acc(q) + \sum_{q' \in Q} \mathbb{P}(q, a, q') \leq 1 \text{ for all } (q, a) \in Q \times A$$

Probabilistic case?

Reactive Probabilistic Finite Automata



$$\mathcal{A} = (Q, \iota, Acc, \mathbb{P})$$

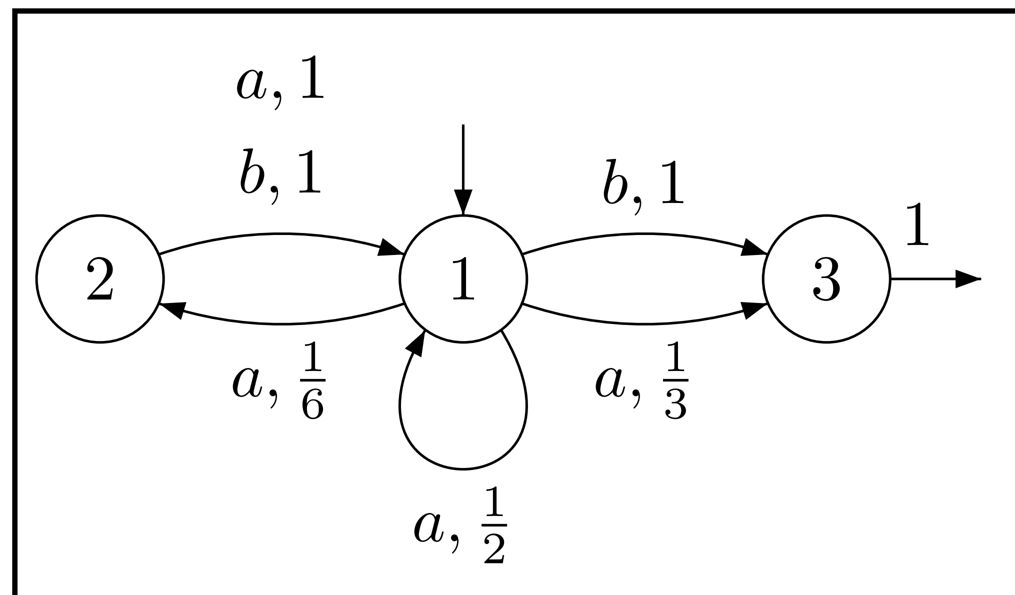
Applying Schützenberger's Theorem over these special Weighted Automata, we obtain regular expressions (proper)

ll $(q, a) \in Q \times A$

$q \in Q$

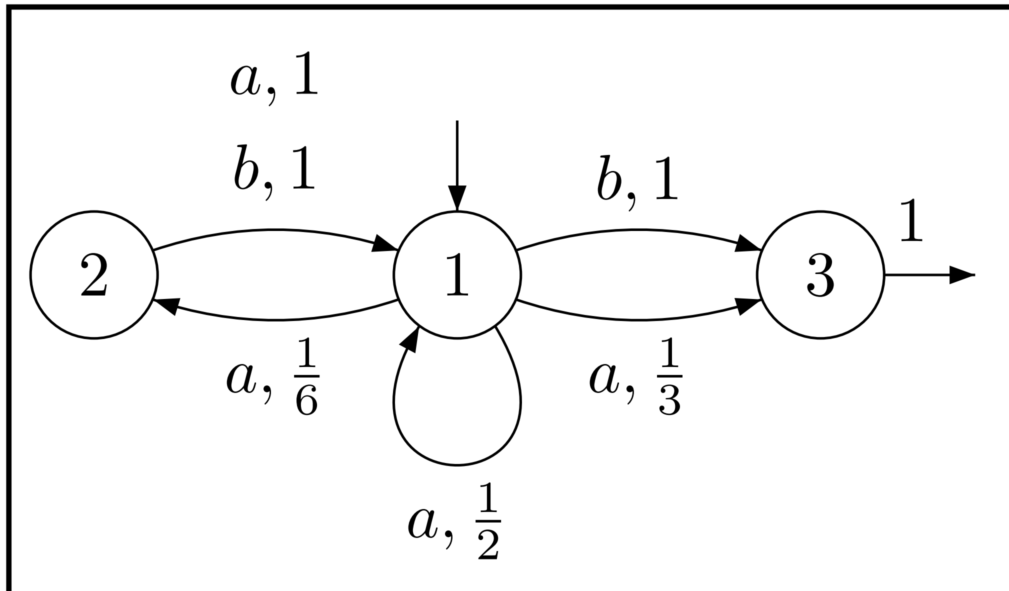
$$(\tfrac{1}{6}a(a+b) + \tfrac{1}{2}a)^*(\tfrac{1}{3}a + b)$$

What kind of expressions?



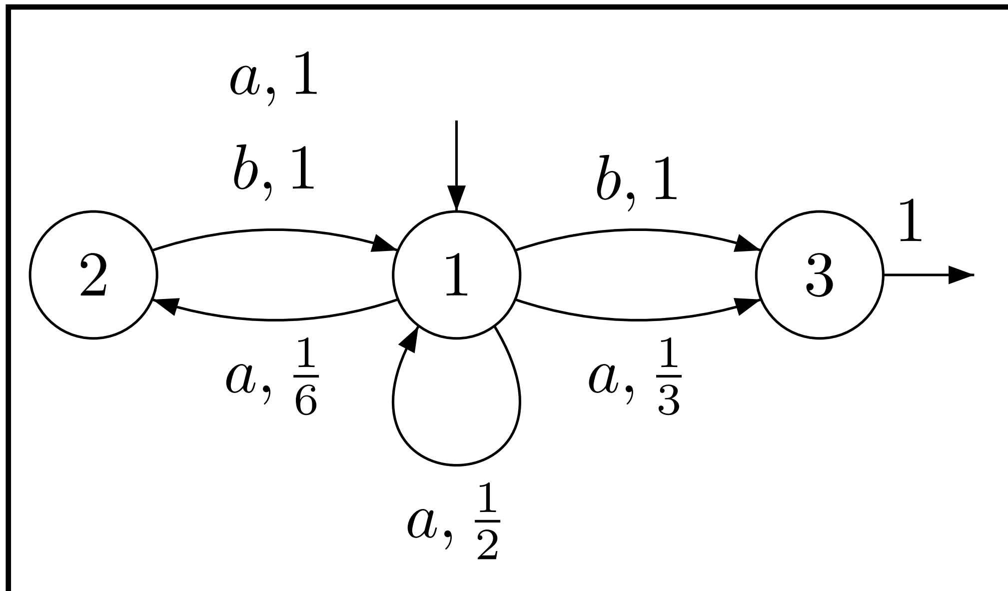
What kind of expressions?

$$(\frac{1}{6}a(a+b) + \frac{1}{2}a)^*(\frac{1}{3}a+b)$$



What kind of expressions?

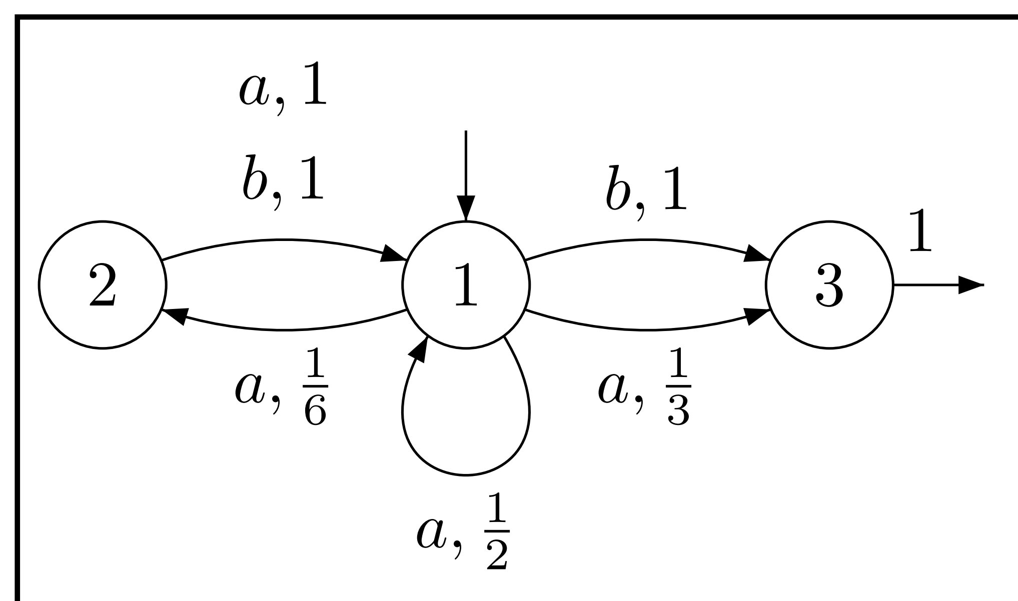
$$(\tfrac{1}{6}a(a+b)+\tfrac{1}{2}a)^*(\tfrac{1}{3}a+b)$$



What kind of expressions?

$$(\tfrac{1}{6}a(a+b)+\tfrac{1}{2}a)^*(\tfrac{1}{3}a+b)$$ ✔

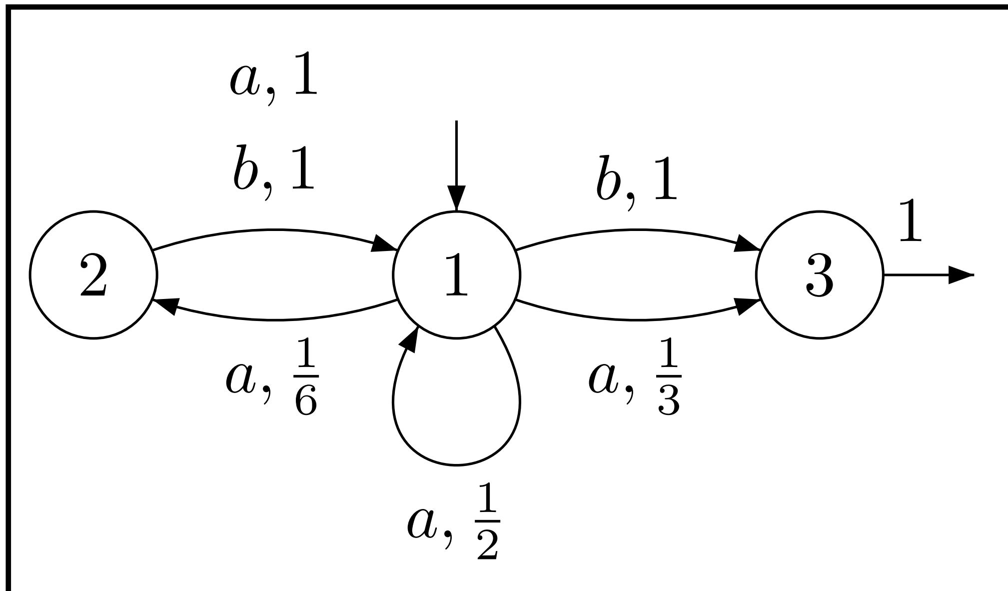
$$(\tfrac{1}{6}a(a+b)+\tfrac{1}{2}a)^*(a+b)$$



What kind of expressions?

$$(\tfrac{1}{6}a(a+b) + \tfrac{1}{2}a)^*(\tfrac{1}{3}a + b)$$

$$(\tfrac{1}{6}a(a+b) + \tfrac{1}{2}a)^*(a + b)$$

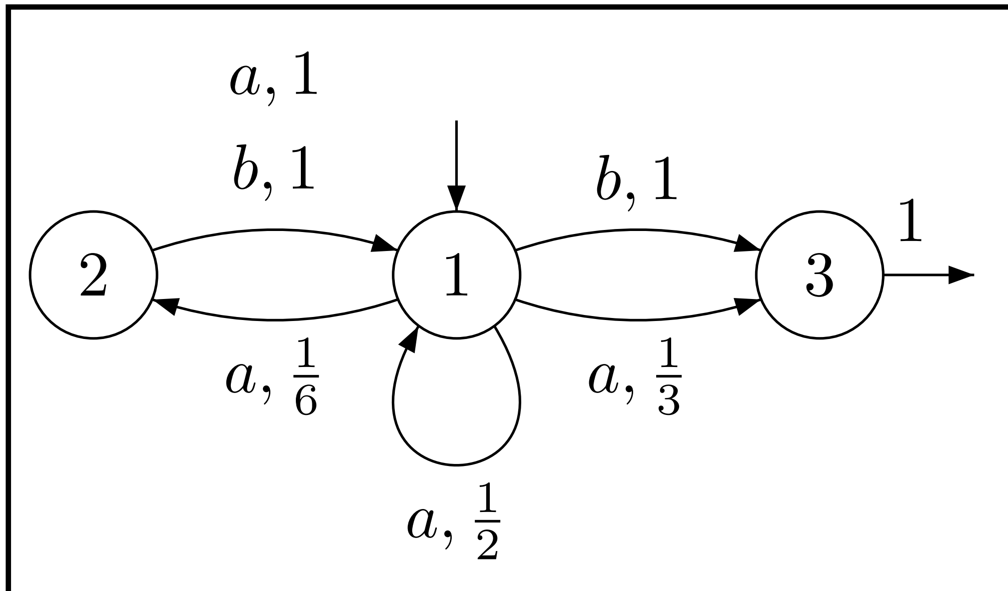


What kind of expressions?

$$(\tfrac{1}{6}a(a+b)+\tfrac{1}{2}a)^*(\tfrac{1}{3}a+b)$$

$$(\tfrac{1}{6}a(a+b)+\tfrac{1}{2}a)^*(a+b)$$

$$(\tfrac{1}{6}a(a+b)+\tfrac{1}{2}a)^*(\tfrac{1}{3}a+\tfrac{1}{2}b)$$

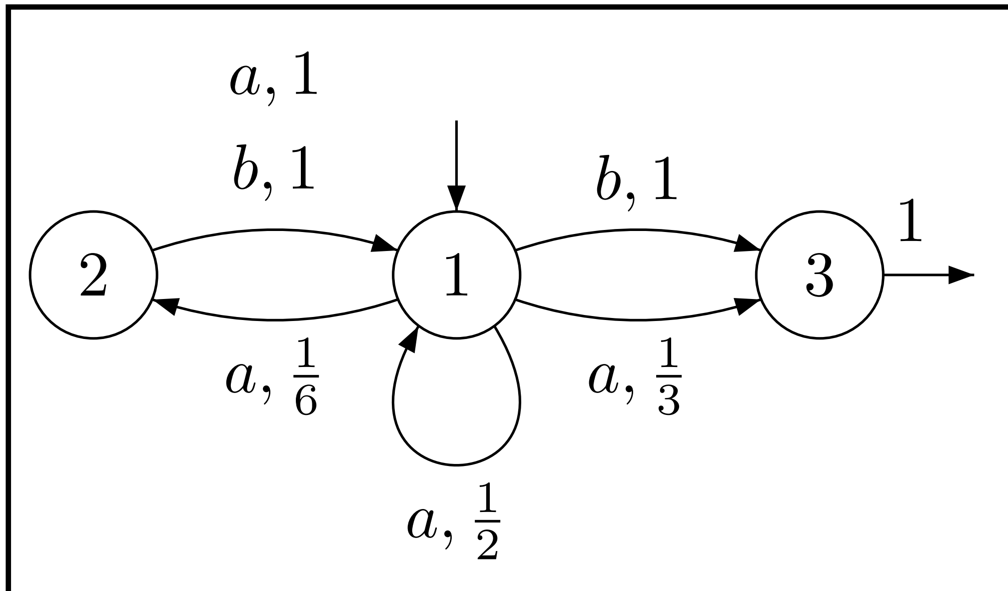


What kind of expressions?

$$(\frac{1}{6}a(a+b)+\frac{1}{2}a)^*(\frac{1}{3}a+b)$$

$$(\frac{1}{6}a(a+b)+\frac{1}{2}a)^*(a+b)$$

$$(\frac{1}{6}a(a+b)+\frac{1}{2}a)^*(\frac{1}{3}a+\frac{1}{2}b)$$



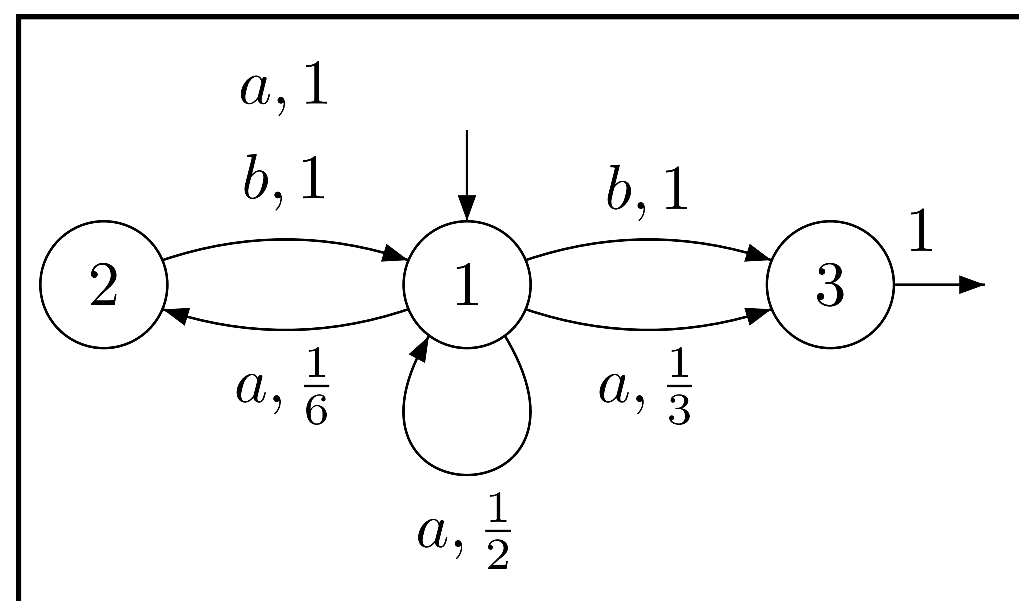
What kind of expressions?

$$(\tfrac{1}{6}a(a+b)+\tfrac{1}{2}a)^*(\tfrac{1}{3}a+b)$$ ✔

$$(\tfrac{1}{6}a(a+b)+\tfrac{1}{2}a)^*(a+b)$$ ✘

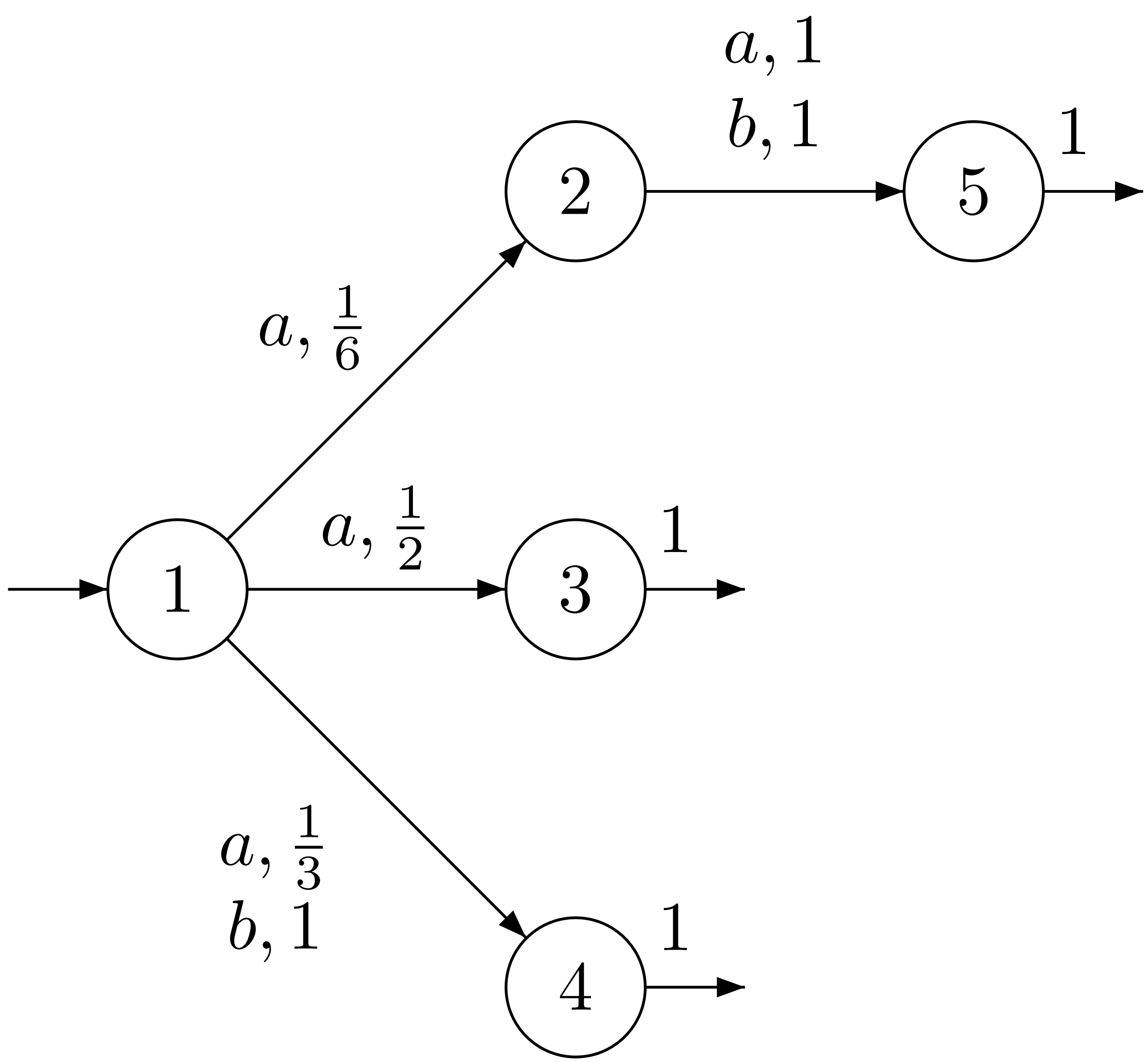
$$(\tfrac{1}{6}a(a+b)+\tfrac{1}{2}a)^*(\tfrac{1}{3}a+\tfrac{1}{2}b)$$ ✔

Searching for a natural fragment of weighted regular expressions representing probabilistic behaviors



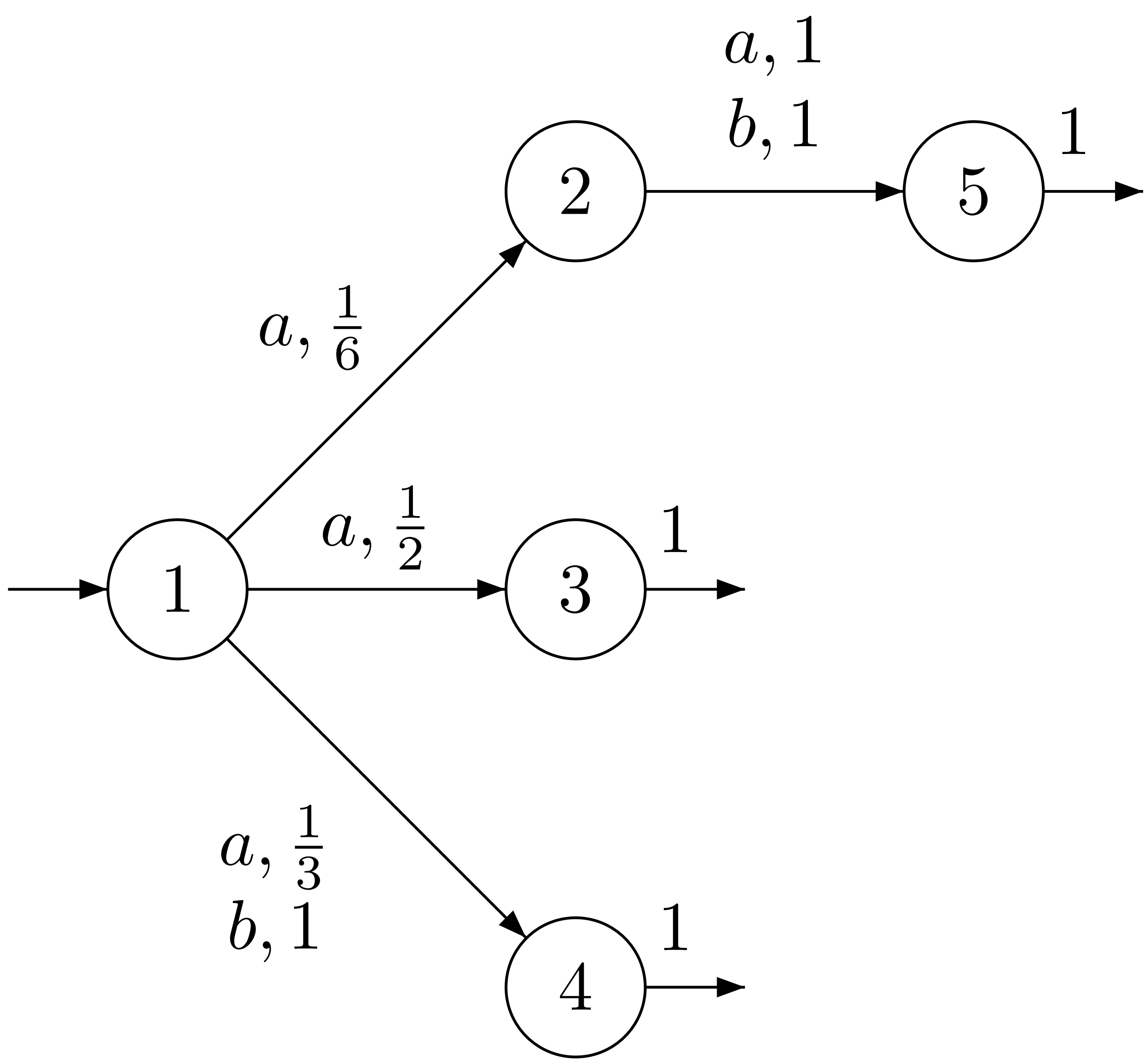
Constructing Probabilistic Expressions

How to iterate?



Constructing Probabilistic Expressions

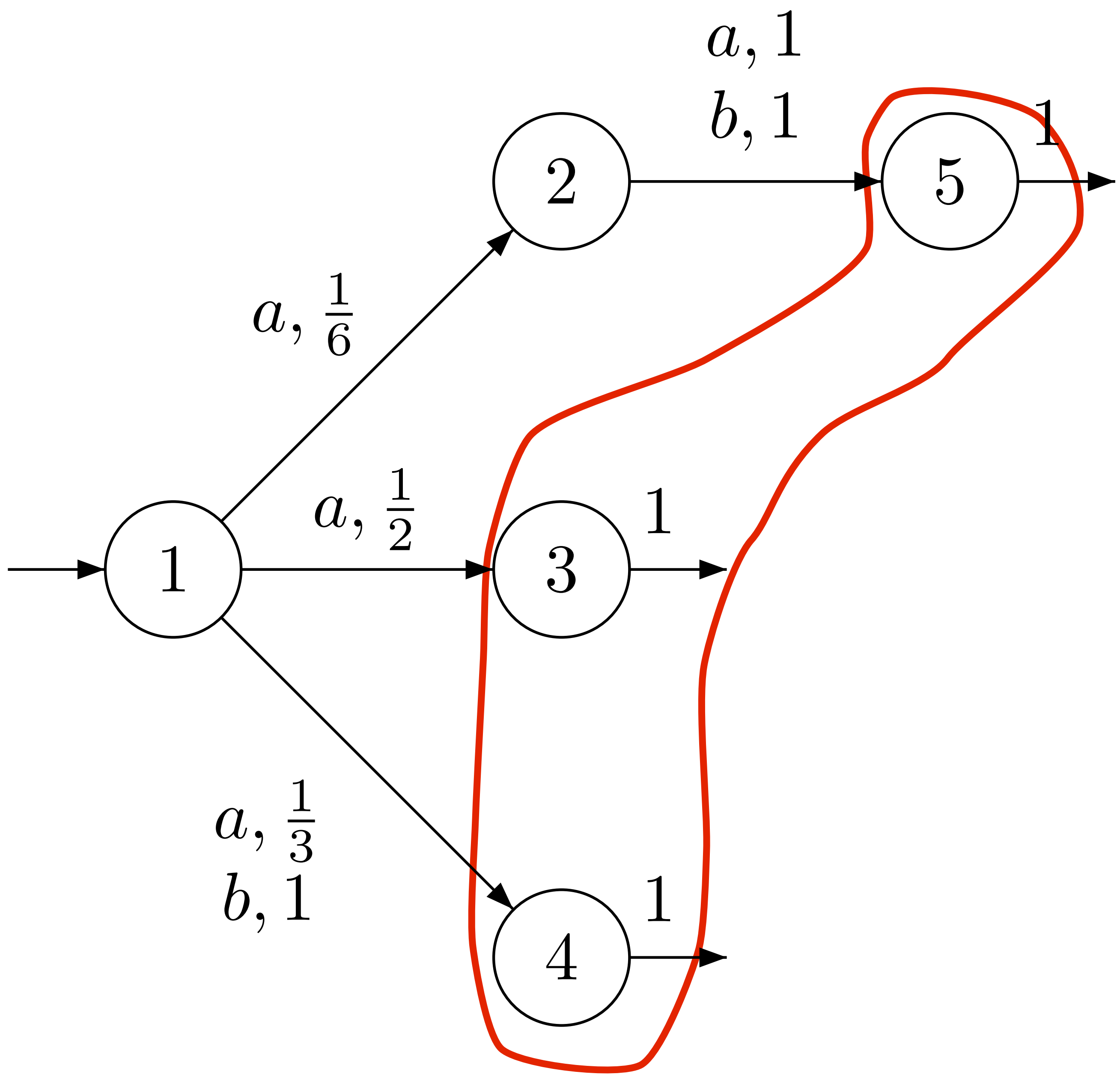
How to iterate?



$$\frac{1}{6}a(a+b) + \frac{1}{2}a + (\frac{1}{3}a + b)$$

Constructing Probabilistic Expressions

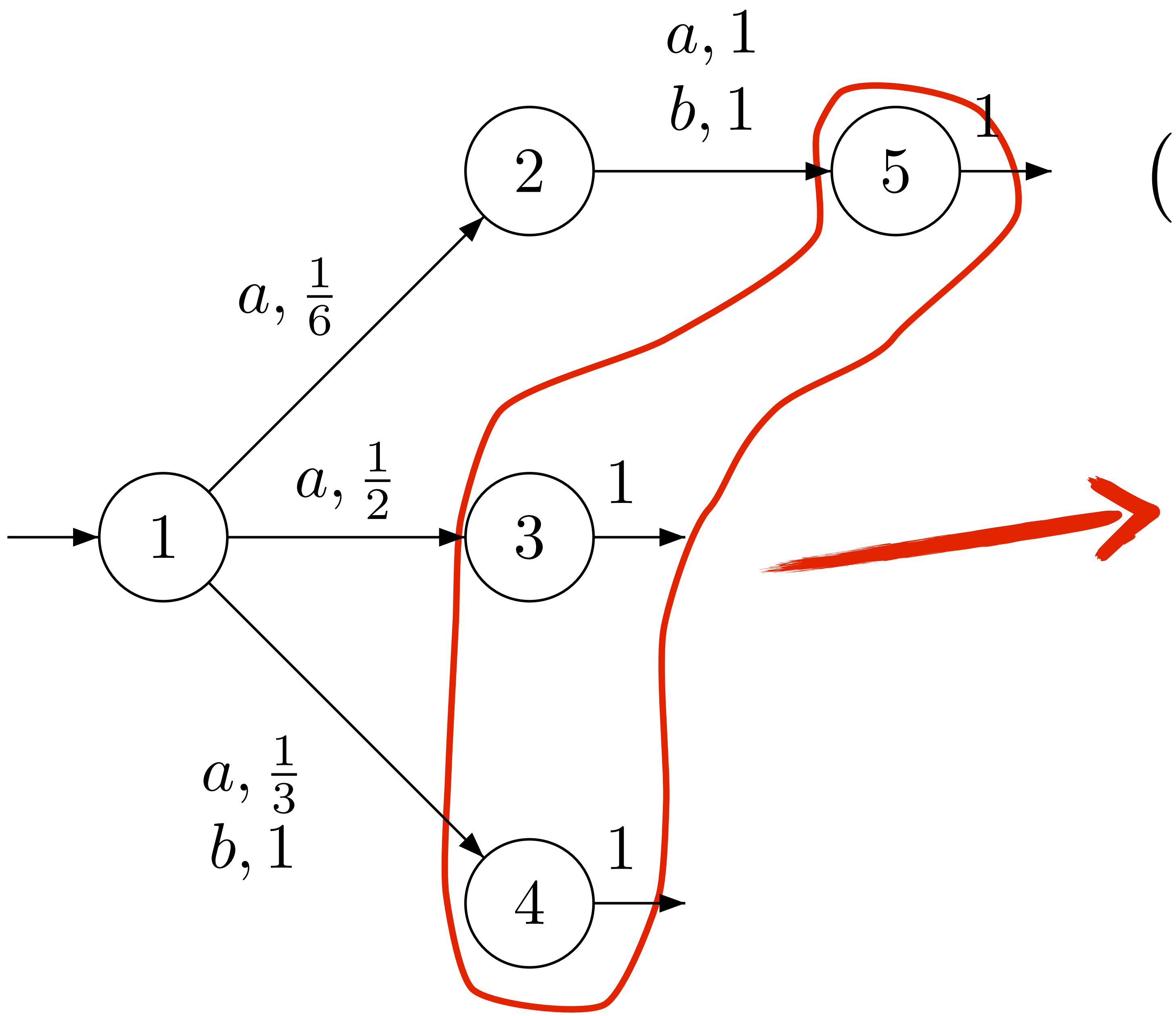
How to iterate?



$$\frac{1}{6}a(a+b) + \frac{1}{2}a + (\frac{1}{3}a + b)$$

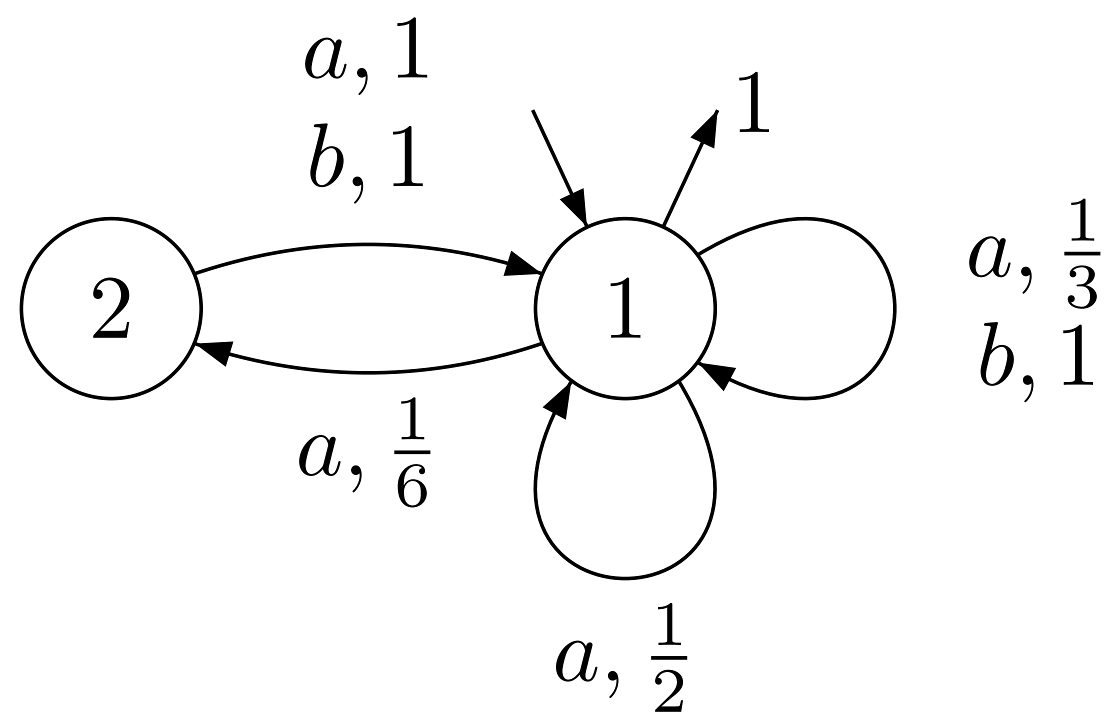
Constructing Probabilistic Expressions

How to iterate?



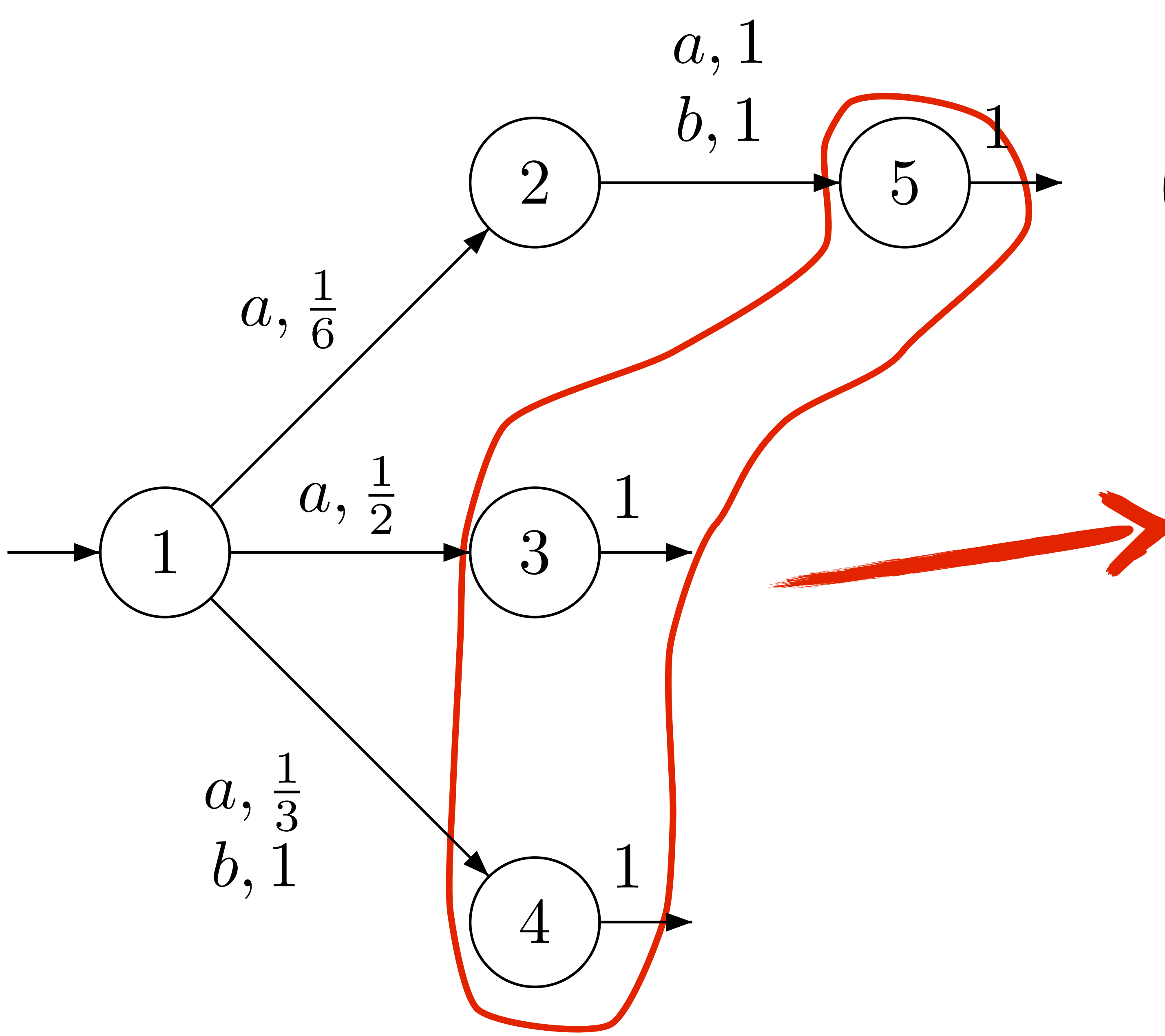
$$\frac{1}{6}a(a+b) + \frac{1}{2}a + (\frac{1}{3}a + b)$$

$$\left(\frac{1}{6}a(a+b) + \frac{1}{2}a + (\frac{1}{3}a + b)\right)^*$$



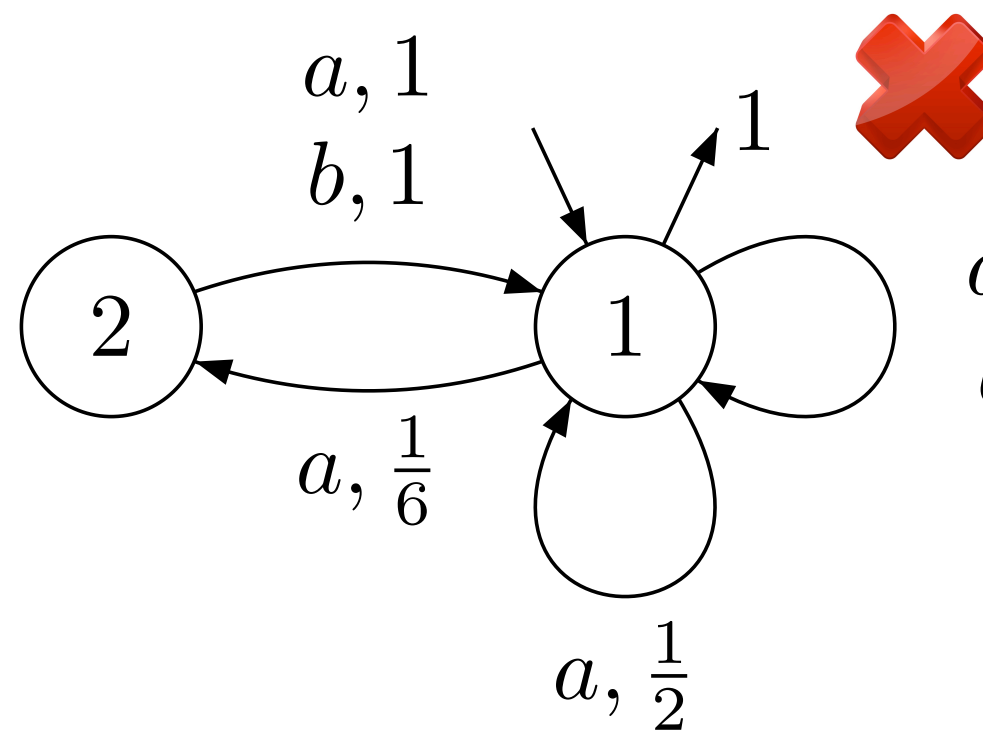
Constructing Probabilistic Expressions

How to iterate?



$\frac{1}{6}a(a+b) + \frac{1}{2}a + (\frac{1}{3}a + b)$

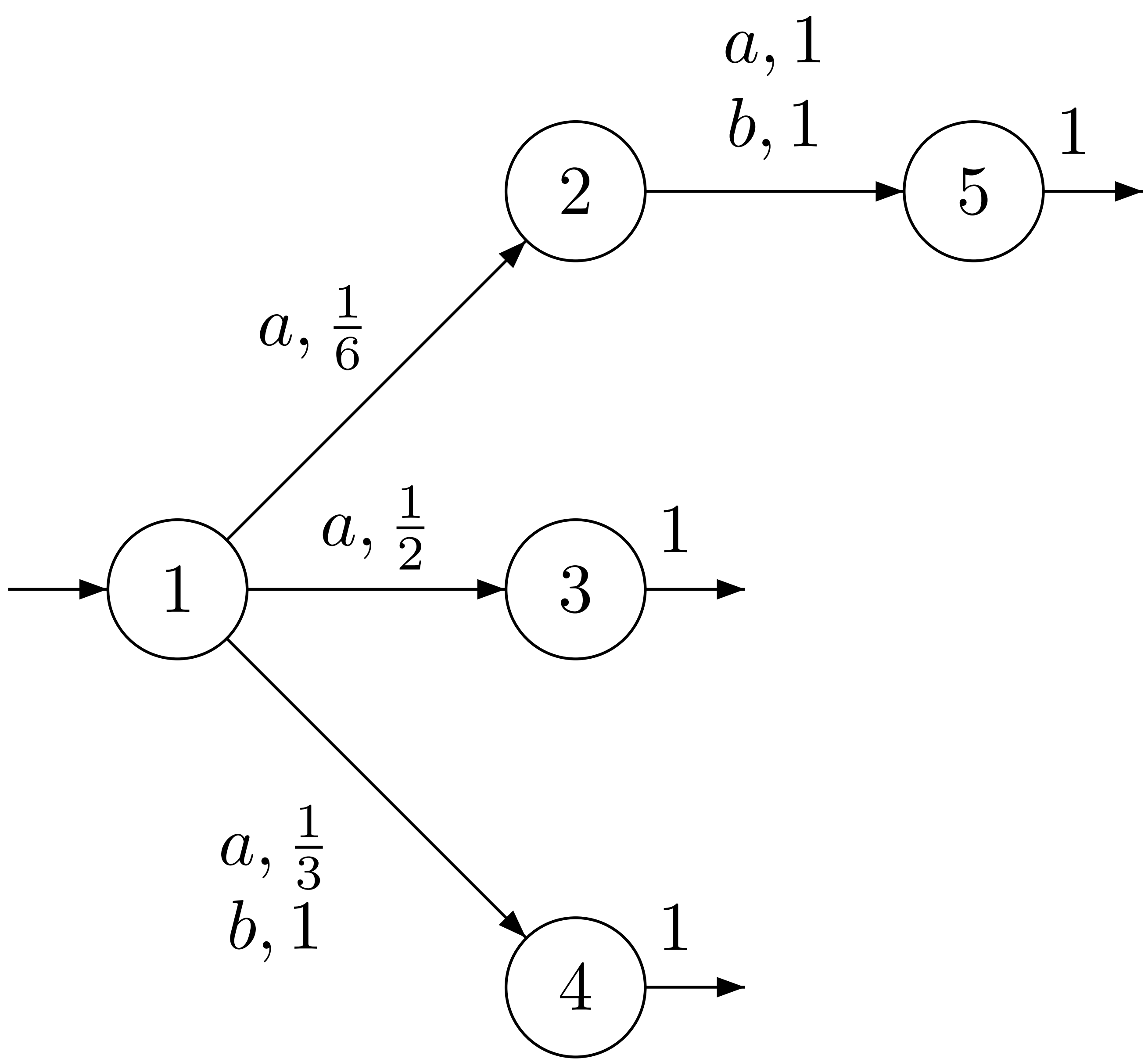
$(\frac{1}{6}a(a+b) + \frac{1}{2}a + (\frac{1}{3}a + b))^*$



Not a valid Probabilistic Automaton anymore (acceptance condition not fulfilled)

Constructing Probabilistic Expressions

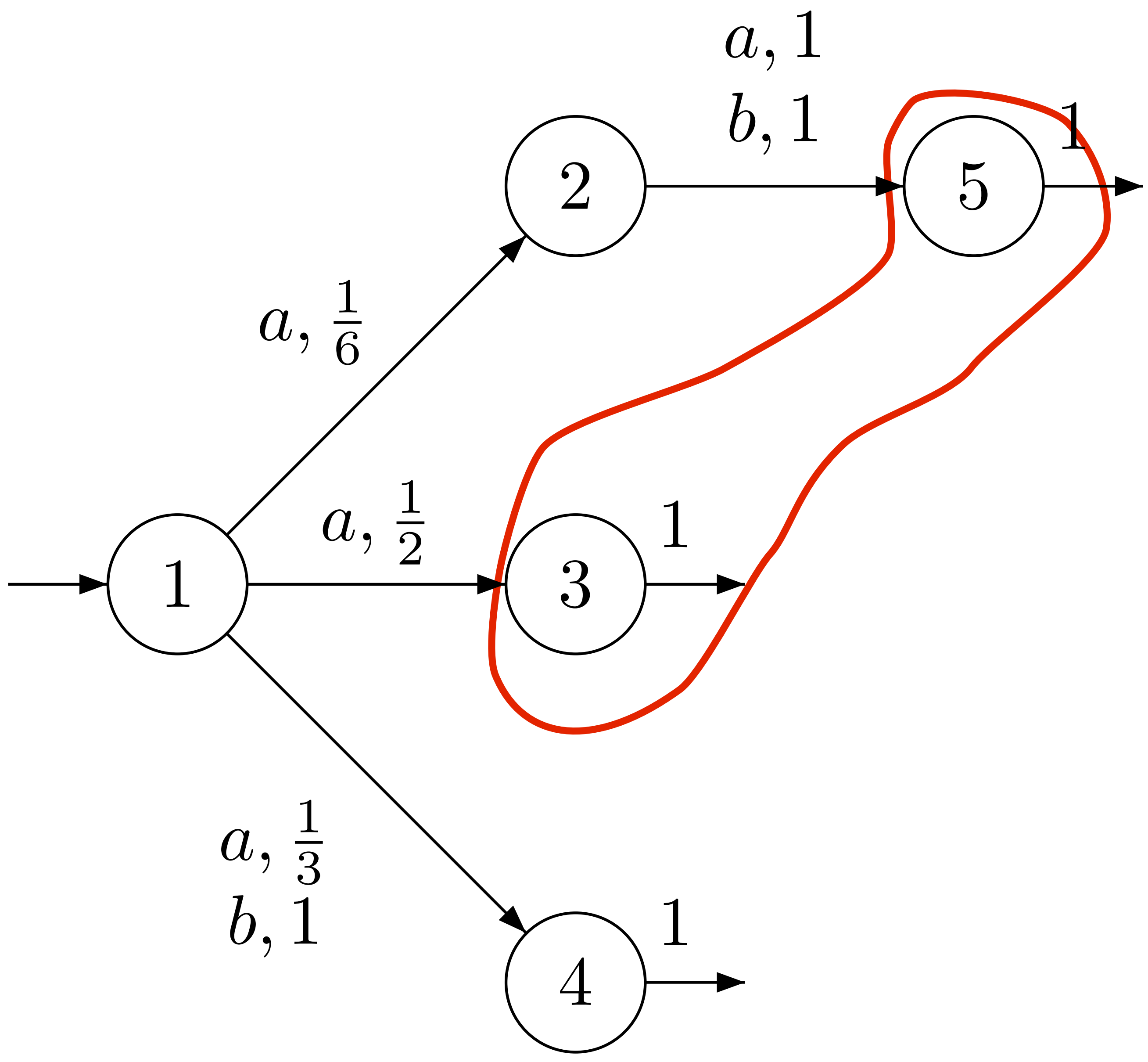
How to iterate?



$$\frac{1}{6}a(a+b) + \frac{1}{2}a + (\frac{1}{3}a + b)$$

Constructing Probabilistic Expressions

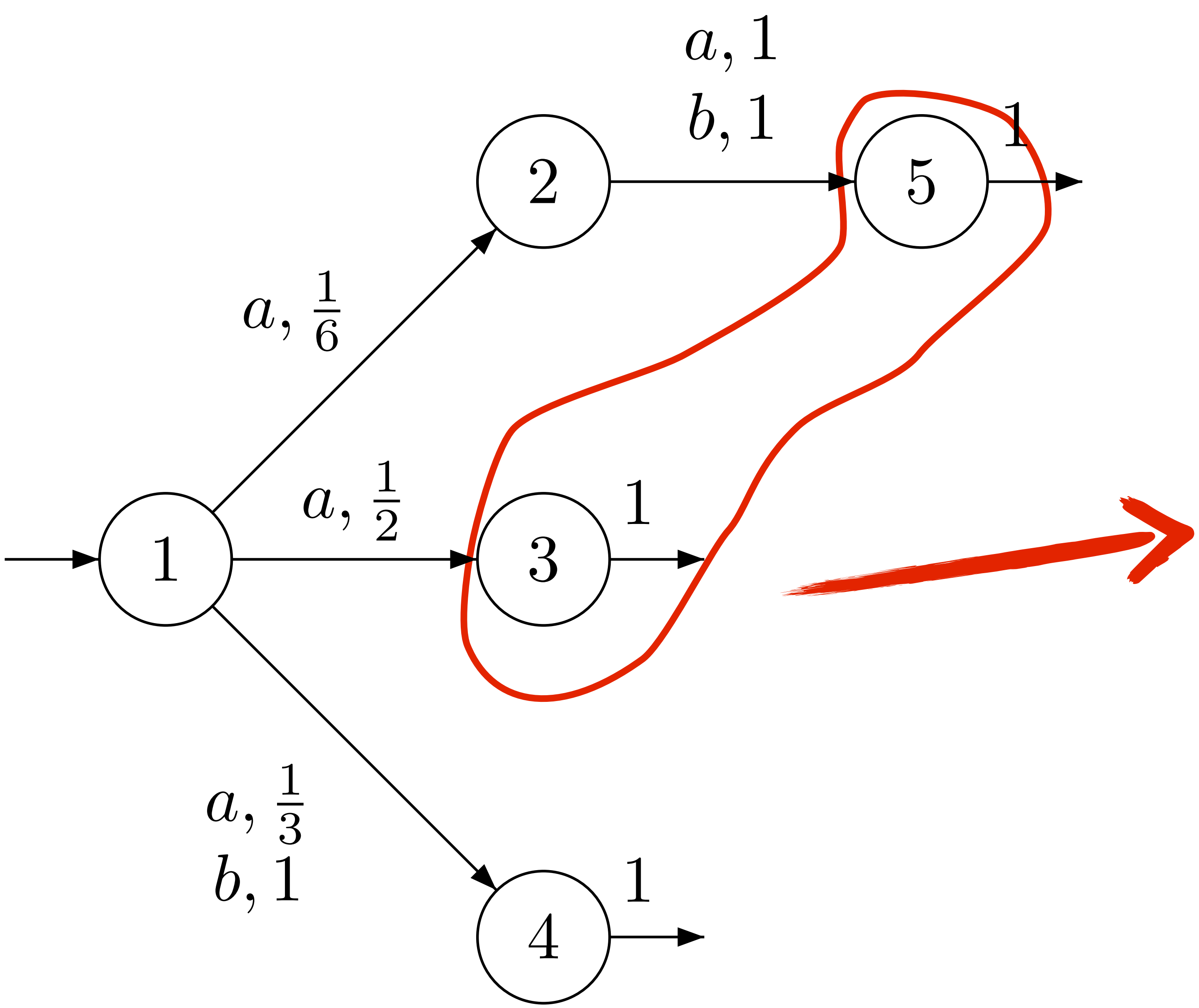
How to iterate?



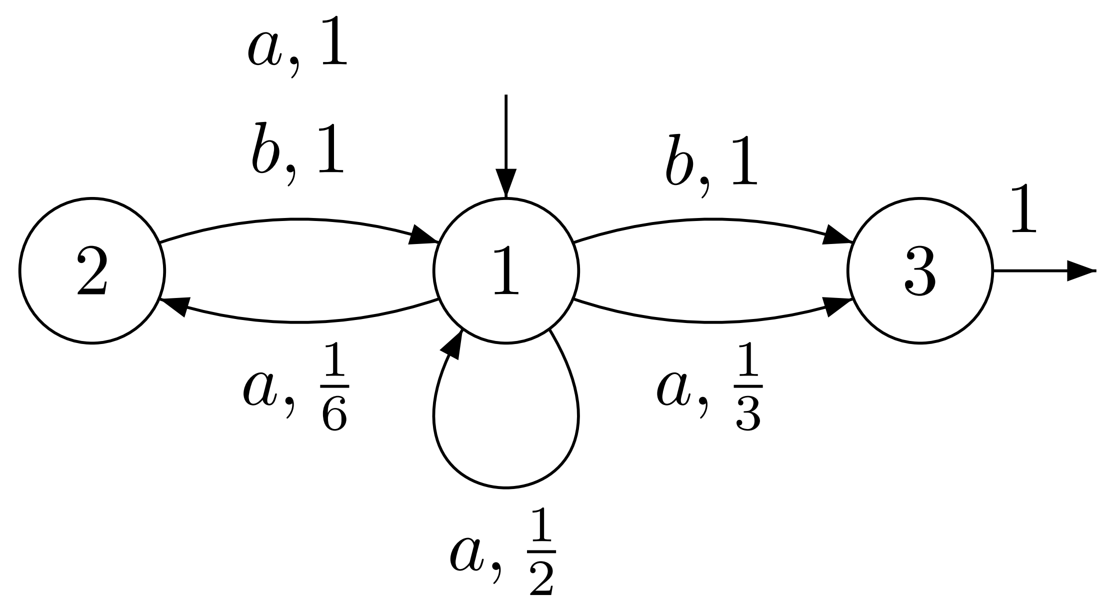
$$\frac{1}{6}a(a+b) + \frac{1}{2}a + (\frac{1}{3}a + b)$$

Constructing Probabilistic Expressions

How to iterate?



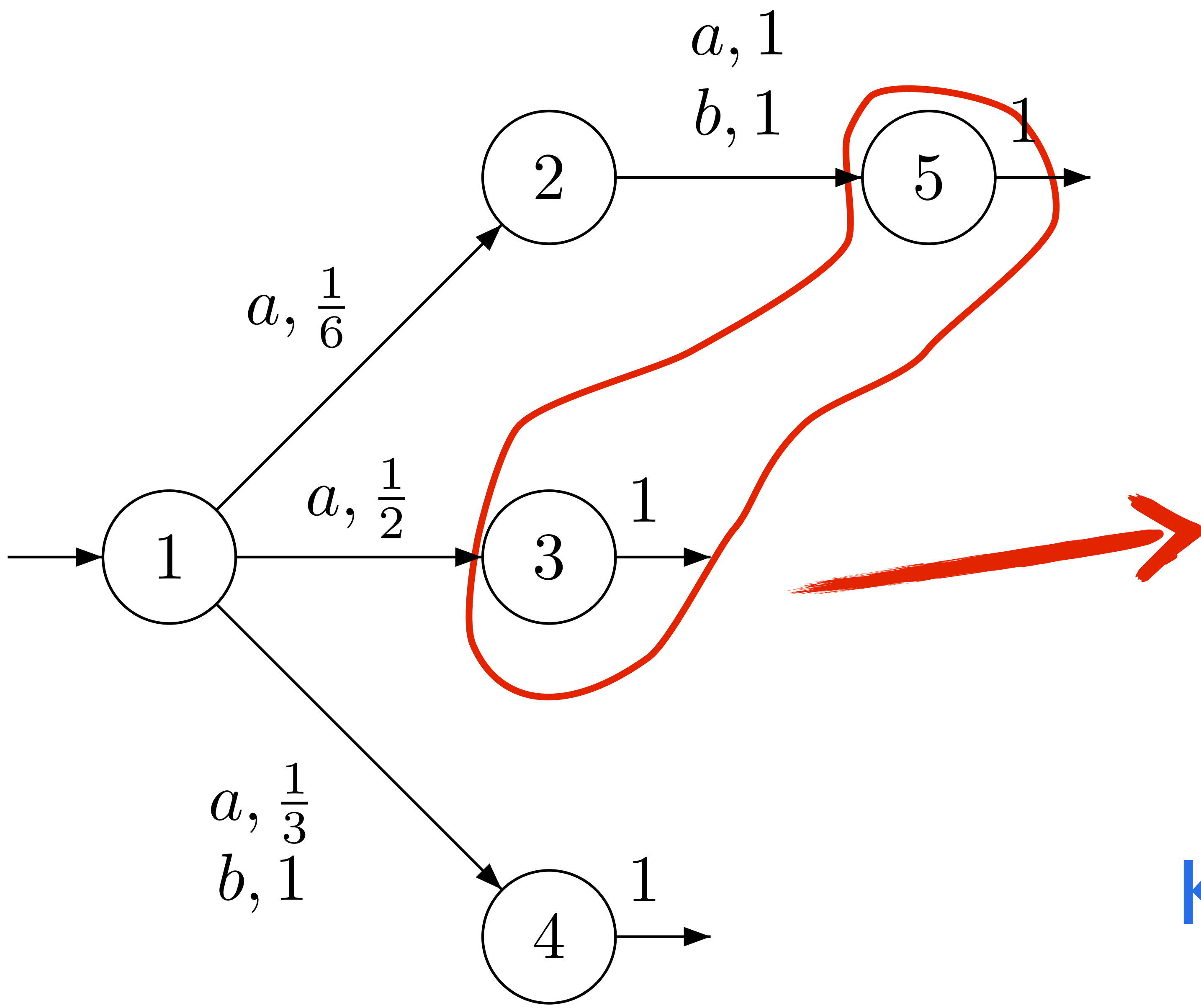
$$(\tfrac{1}{6}a(a+b) + \tfrac{1}{2}a)^*(\tfrac{1}{3}a + b)$$



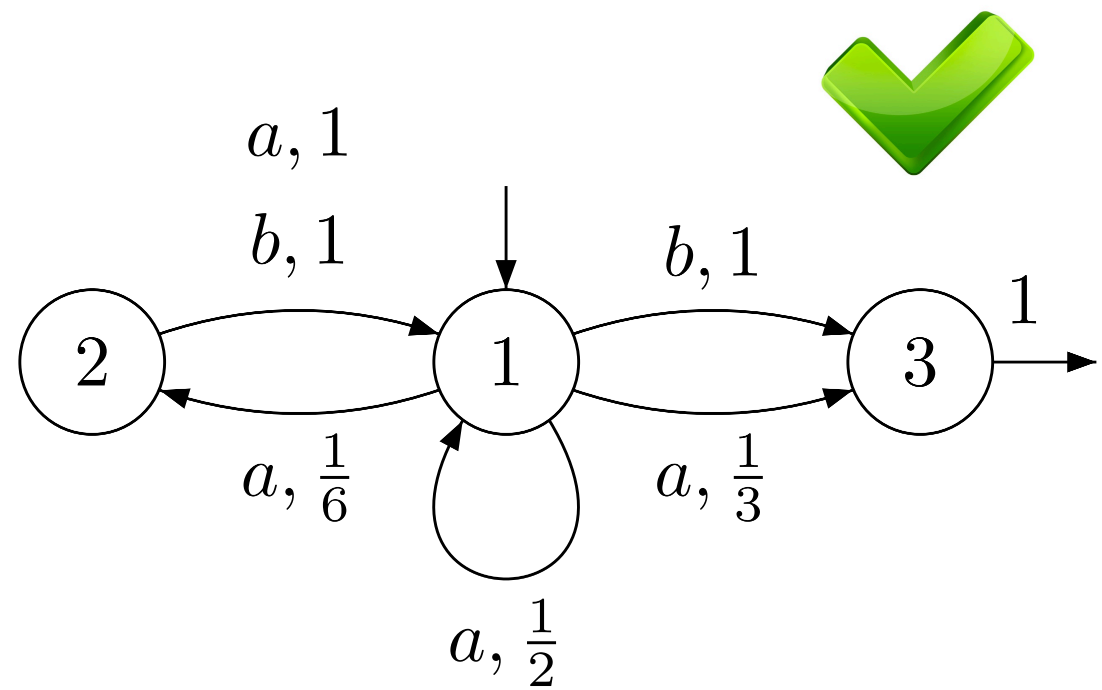
$$\tfrac{1}{6}a(a+b) + \tfrac{1}{2}a + (\tfrac{1}{3}a + b)$$

Constructing Probabilistic Expressions

How to iterate?



$$\left(\tfrac{1}{6}a(a+b) + \tfrac{1}{2}a\right)^*\left(\tfrac{1}{3}a + b\right)$$



$$\tfrac{1}{6}a(a+b) + \tfrac{1}{2}a + \left(\tfrac{1}{3}a + b\right)$$

Keep some *branch* for termination of the Probabilistic Automaton

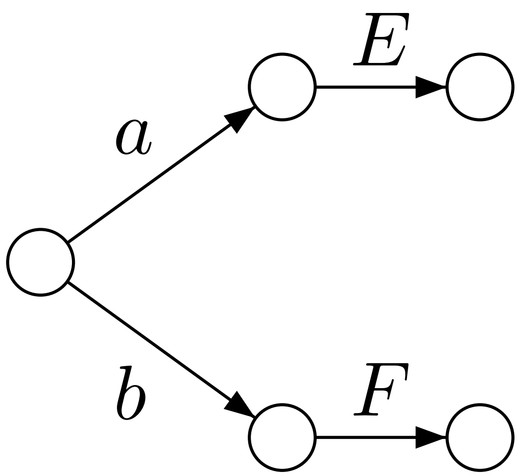
Probabilistic Expressions

Probabilistic Expressions

- $a \in A$ and $p \in [0,1]$ are PREs

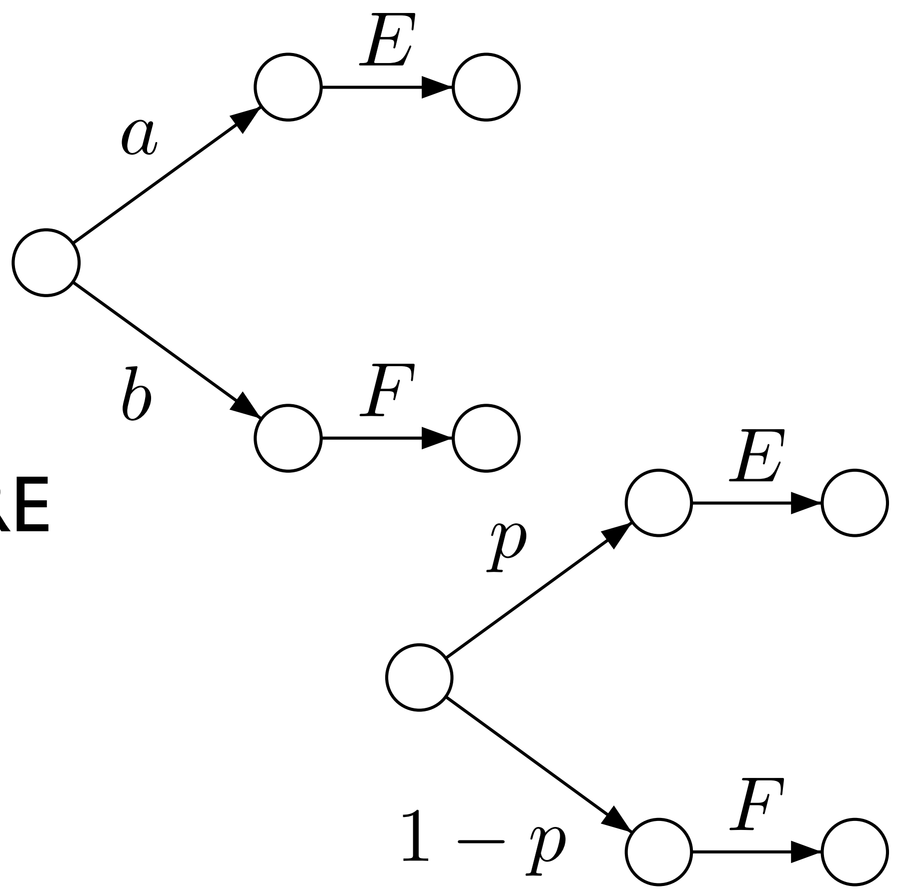
Probabilistic Expressions

- $a \in A$ and $p \in [0,1]$ are PREs
- if $(E_a)_{a \in A}$ are PREs, then $\sum_{a \in A} a \cdot E_a$ is a PRE



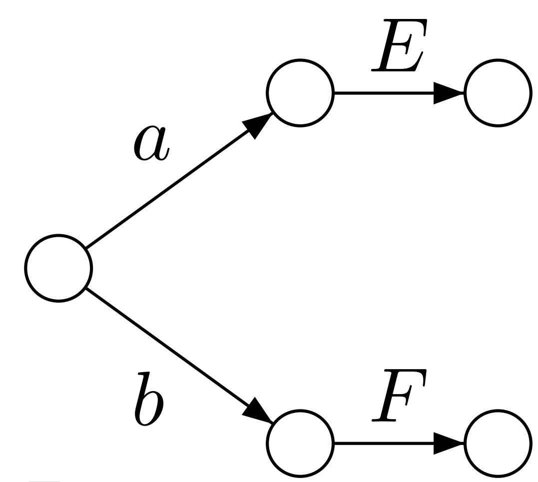
Probabilistic Expressions

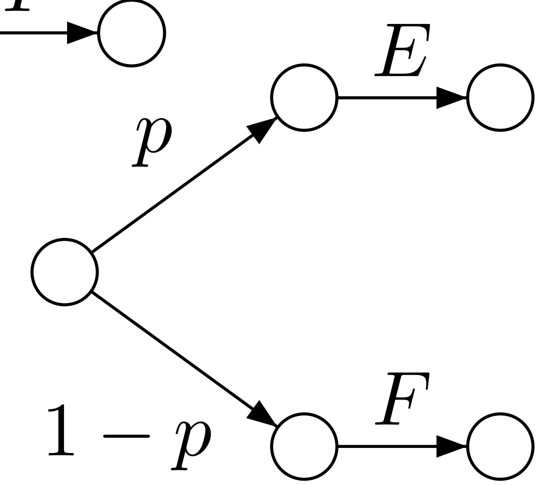
- $a \in A$ and $p \in [0,1]$ are PREs
- if $(E_a)_{a \in A}$ are PREs, then $\sum_{a \in A} a \cdot E_a$ is a PRE
- if E and F are PREs, then $p \cdot E + (1-p) \cdot F$ is a PRE



Probabilistic Expressions

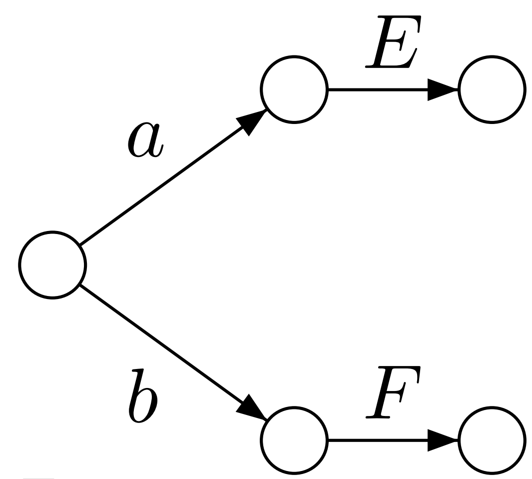
- $a \in A$ and $p \in [0,1]$ are PREs
- if $(E_a)_{a \in A}$ are PREs, then $\sum_{a \in A} a \cdot E_a$ is a PRE
- if E and F are PREs, then $p \cdot E + (1\text{-}p) \cdot F$ is a PRE
- if E and F are PREs, then $E \cdot F$ is a PRE

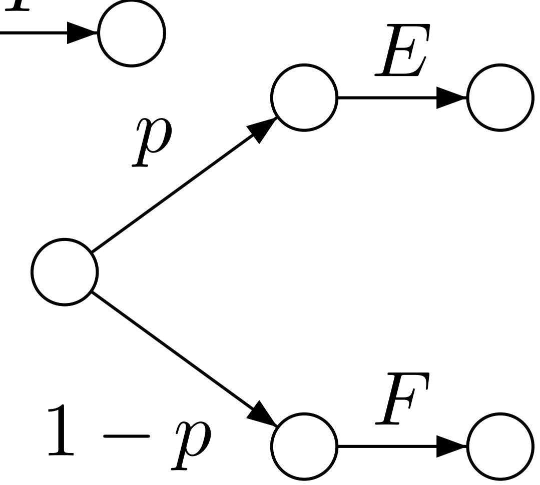


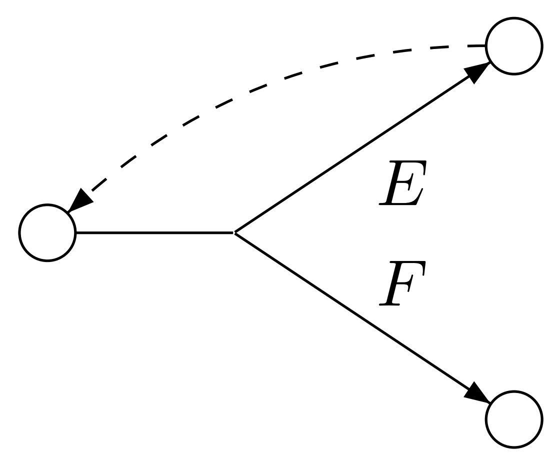


Probabilistic Expressions

- $a \in A$ and $p \in [0,1]$ are PREs
- if $(E_a)_{a \in A}$ are PREs, then $\sum_{a \in A} a \cdot E_a$ is a PRE
- if E and F are PREs, then $p \cdot E + (1-p) \cdot F$ is a PRE
- if E and F are PREs, then $E \cdot F$ is a PRE
- if E+F is a PRE, then $E^* \cdot F$ is a PRE

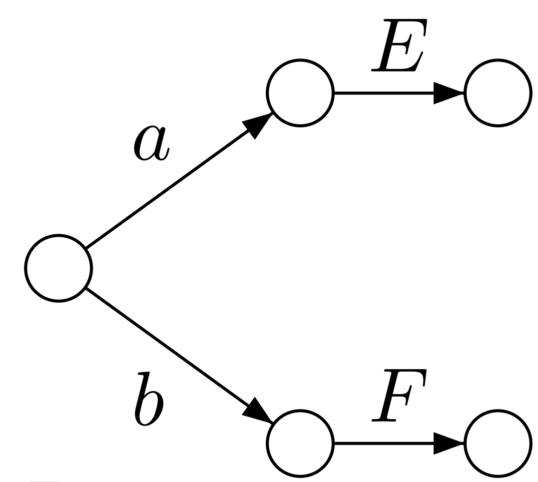


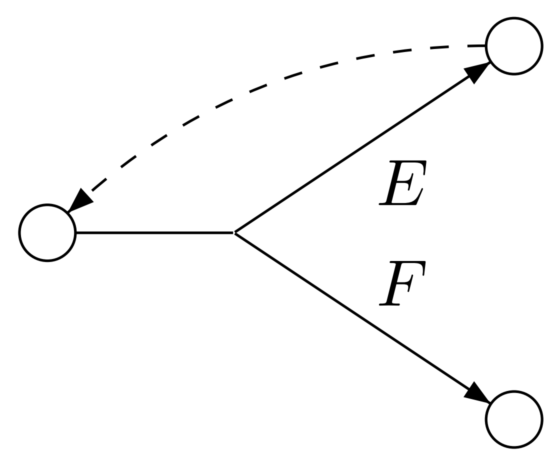


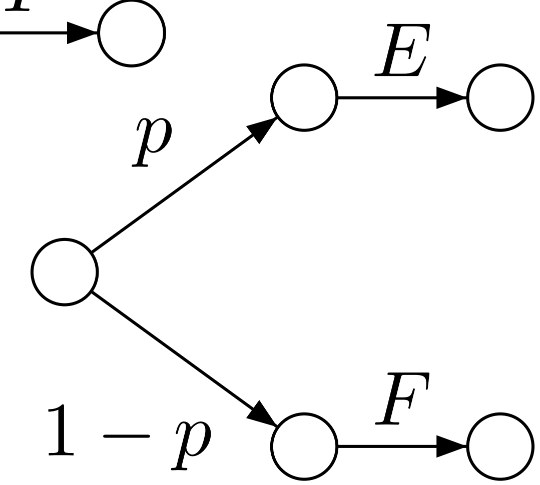


Probabilistic Expressions

- $a \in A$ and $p \in [0,1]$ are PREs
- if $(E_a)_{a \in A}$ are PREs, then $\sum_{a \in A} a \cdot E_a$ is a PRE
- if E and F are PREs, then $p \cdot E + (1-p) \cdot F$ is a PRE
- if E and F are PREs, then $E \cdot F$ is a PRE
- if E+F is a PRE, then $E^* \cdot F$ is a PRE





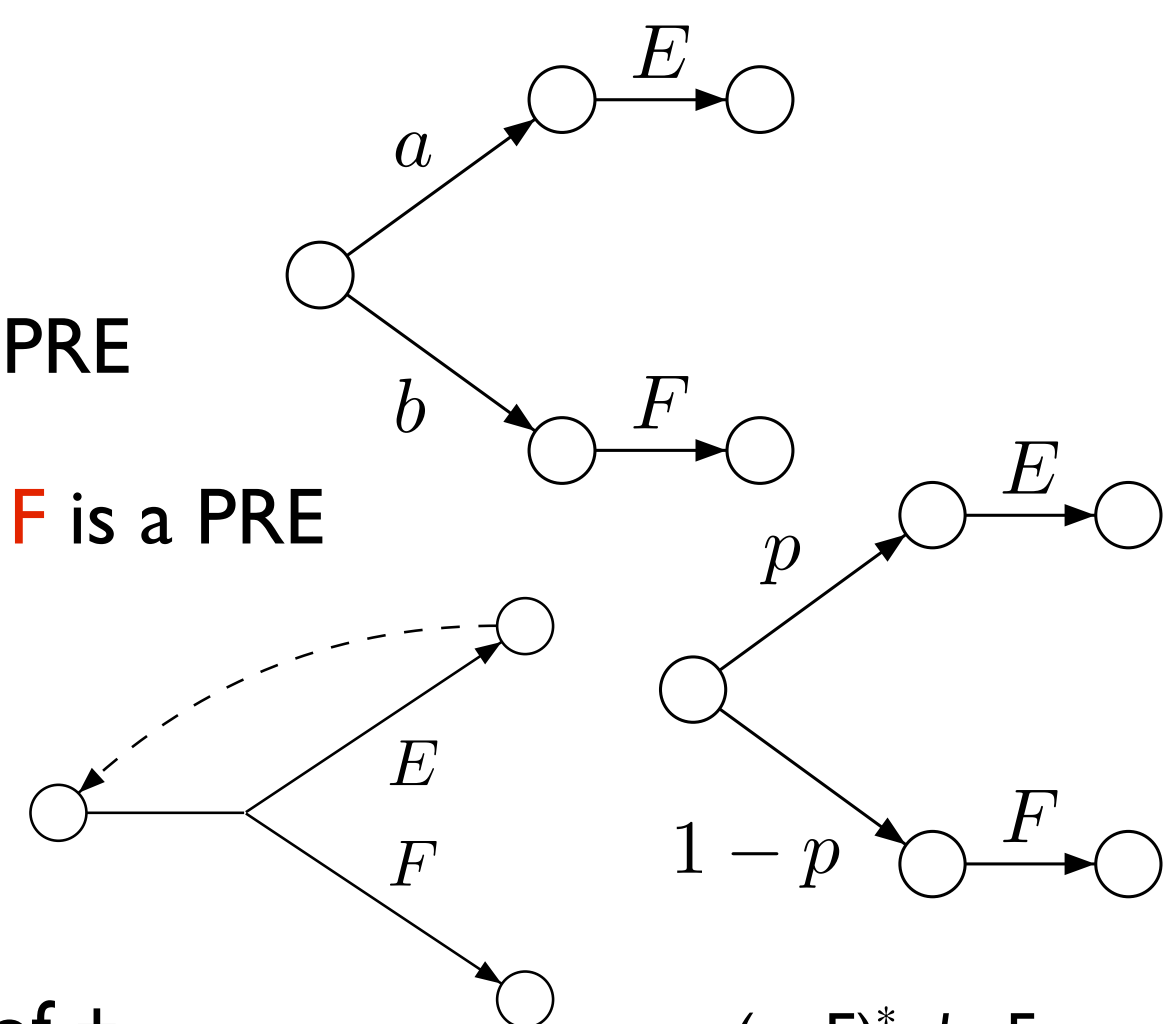


$(a \cdot E)^* \cdot b \cdot F$

$(p \cdot E)^* \cdot (1-p) \cdot F$

Probabilistic Expressions

- $a \in A$ and $p \in [0,1]$ are PREs
- if $(E_a)_{a \in A}$ are PREs, then $\sum_{a \in A} a \cdot E_a$ is a PRE
- if E and F are PREs, then $p \cdot E + (1\text{-}p) \cdot F$ is a PRE
- if E and F are PREs, then $E \cdot F$ is a PRE
- if E+F is a PRE, then $E^* \cdot F$ is a PRE
- Closure of PRE under commutativity of +, associativity of + and ·, distributivity of · over +

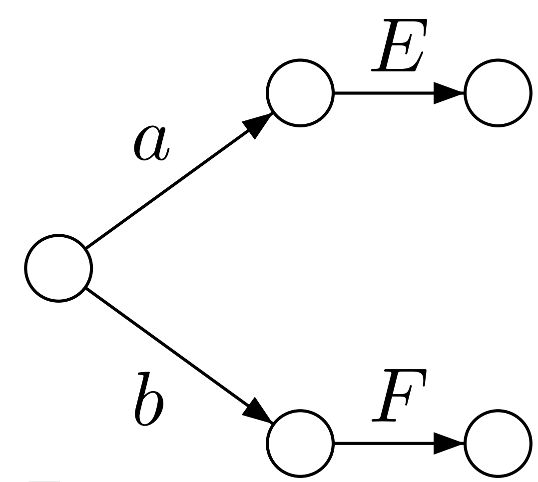


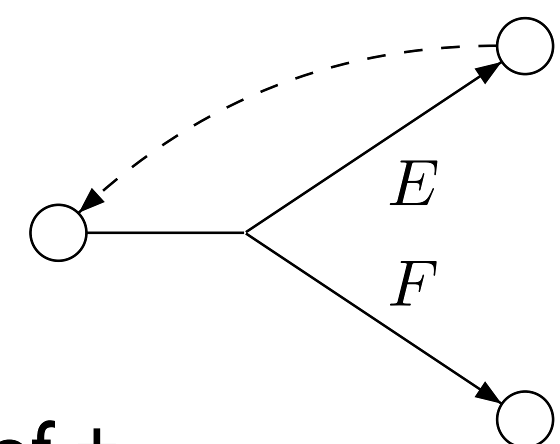
$(a \cdot E)^* \cdot b \cdot F$

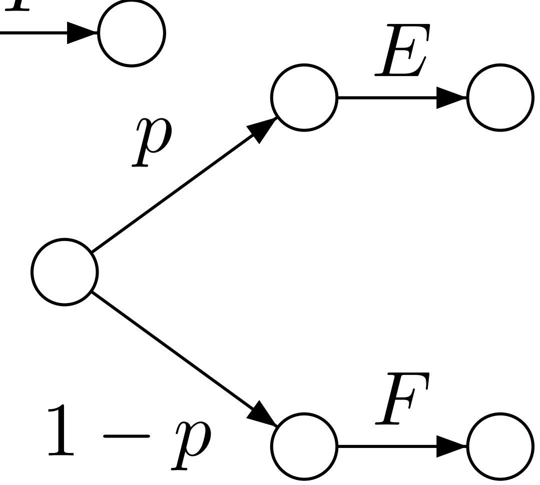
$(p \cdot E)^* \cdot (1\text{-}p) \cdot F$

Probabilistic Expressions

- $a \in A$ and $p \in [0,1]$ are PREs
- if $(E_a)_{a\in A}$ are PREs, then $\sum_{a\in A} a \cdot E_a$ is a PRE
- if E and F are PREs, then $p \cdot E + (1\text{-}p) \cdot F$ is a PRE
- if E and F are PREs, then $E \cdot F$ is a PRE
- if E+F is a PRE, then $E^* \cdot F$ is a PRE
- Closure of PRE under commutativity of +, associativity of + and $\cdot$, distributivity of $\cdot$ over +







$(a \cdot E)^* \cdot b \cdot F$

$(p \cdot E)^* \cdot (1\text{-}p) \cdot F$

Semantics given as a fragment of regular expressions in complete semirings...

$$\mathbb{P}(E \cdot F, u) = \sum_{u=vw} \mathbb{P}(E, v) \times \mathbb{P}(F, w)$$

Example

$$a + b$$

$$a^*b \qquad \frac{1}{2}a + \frac{1}{2}b$$

$$a^*b(\frac{1}{2}a + \frac{1}{2}b)$$

$$a^*b\frac{1}{2}a + a^*b\frac{1}{2}b$$

$$(a^*b\frac{1}{2}a)^*a^*b\frac{1}{2}b$$

Example

Deterministic choice

$$a + b$$

$$\downarrow$$

$$a^*b \qquad \tfrac{1}{2}a + \tfrac{1}{2}b$$

$$\searrow \quad \swarrow$$

$$a^*b(\tfrac{1}{2}a + \tfrac{1}{2}b)$$

$$\downarrow$$

$$a^*b\frac{1}{2}a + a^*b\frac{1}{2}b$$

$$\downarrow$$

$$(a^*b\frac{1}{2}a)^*a^*b\frac{1}{2}b$$

Example

Deterministic choice

$$a + b$$

Star rule

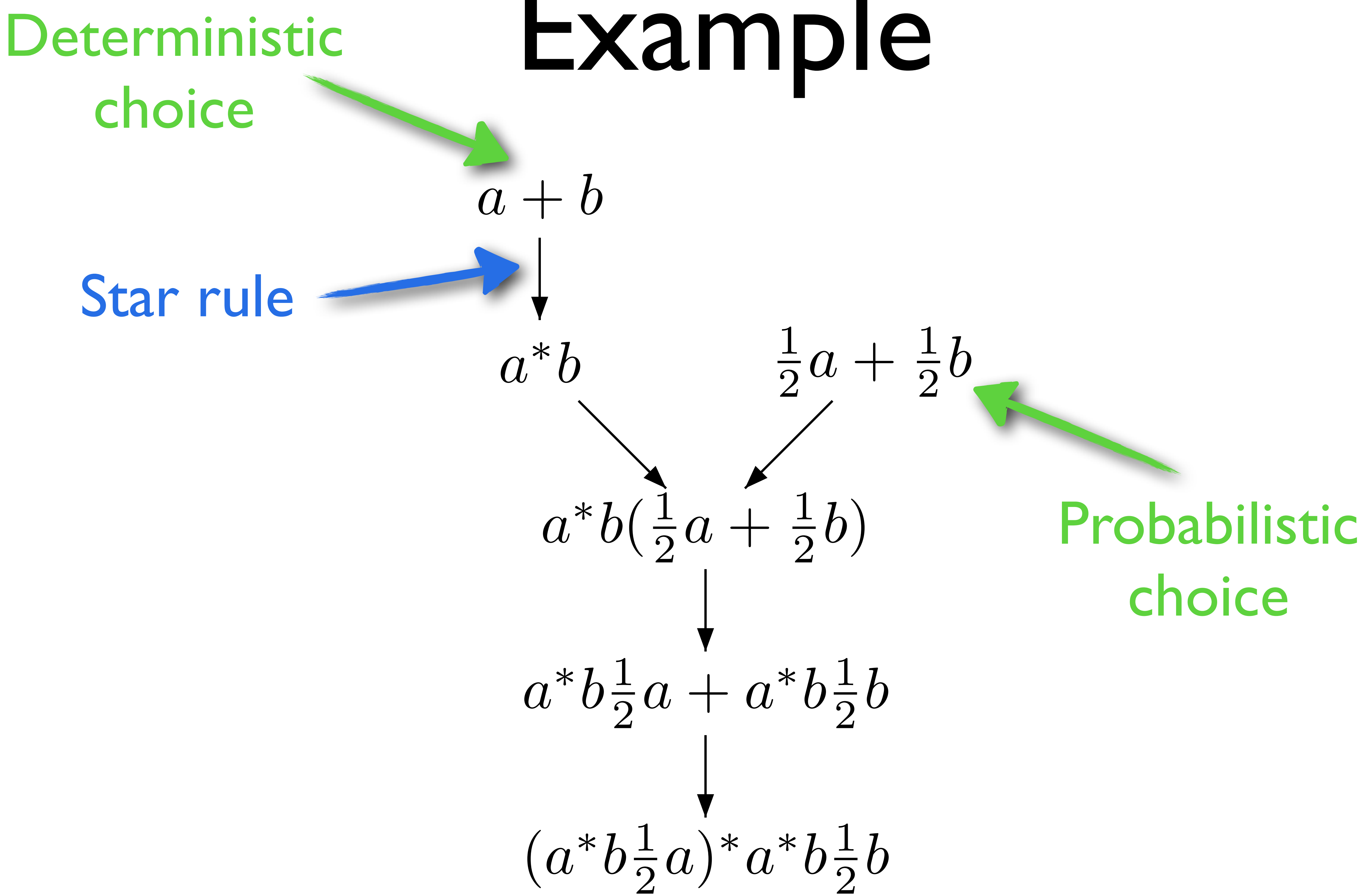
$$a^* b \qquad \tfrac{1}{2}a + \tfrac{1}{2}b$$

$$a^* b(\tfrac{1}{2}a + \tfrac{1}{2}b)$$

$$a^* b\tfrac{1}{2}a + a^* b\tfrac{1}{2}b$$

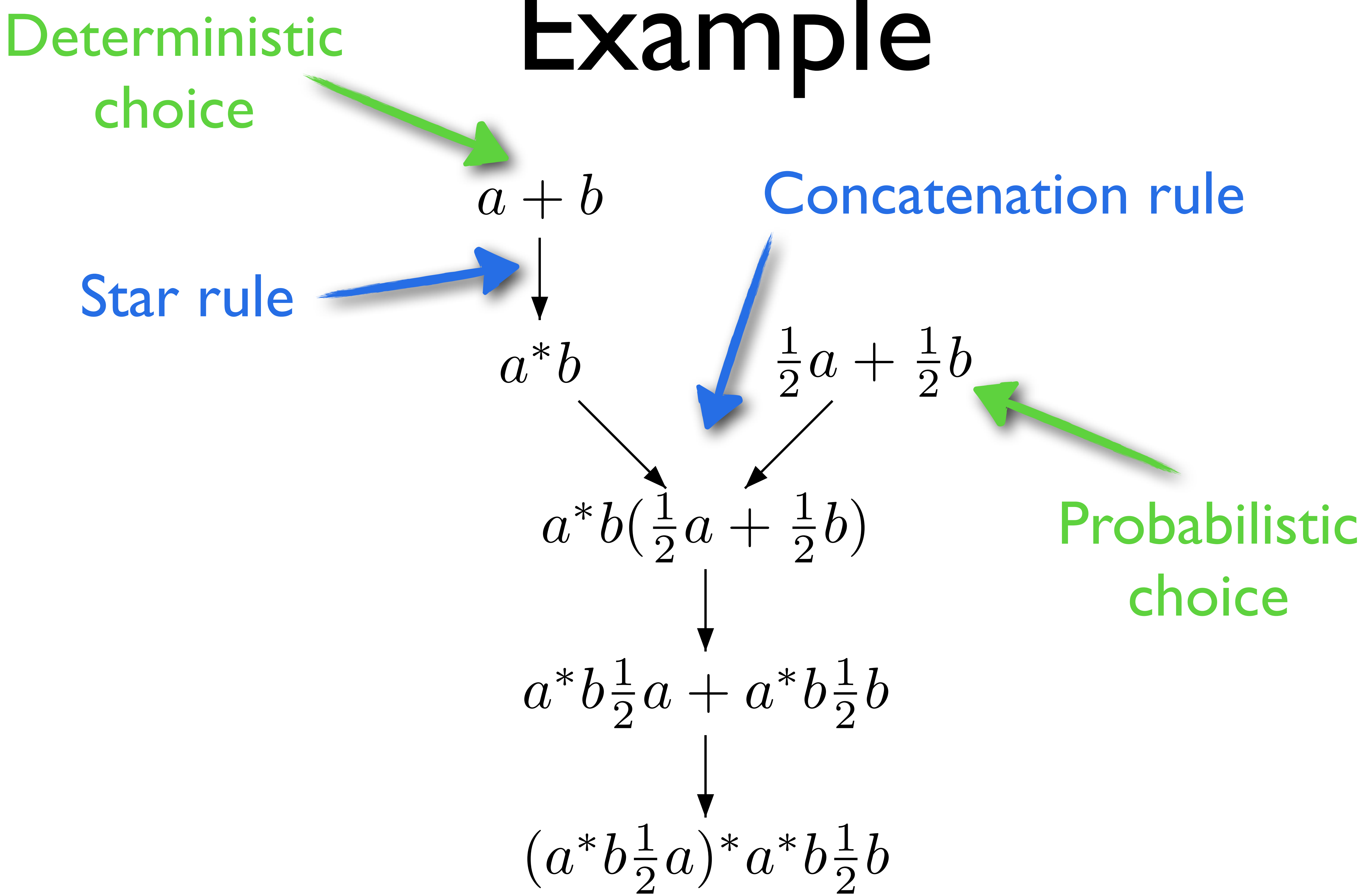
$$(a^* b\tfrac{1}{2}a)^* a^* b\tfrac{1}{2}b$$

Example

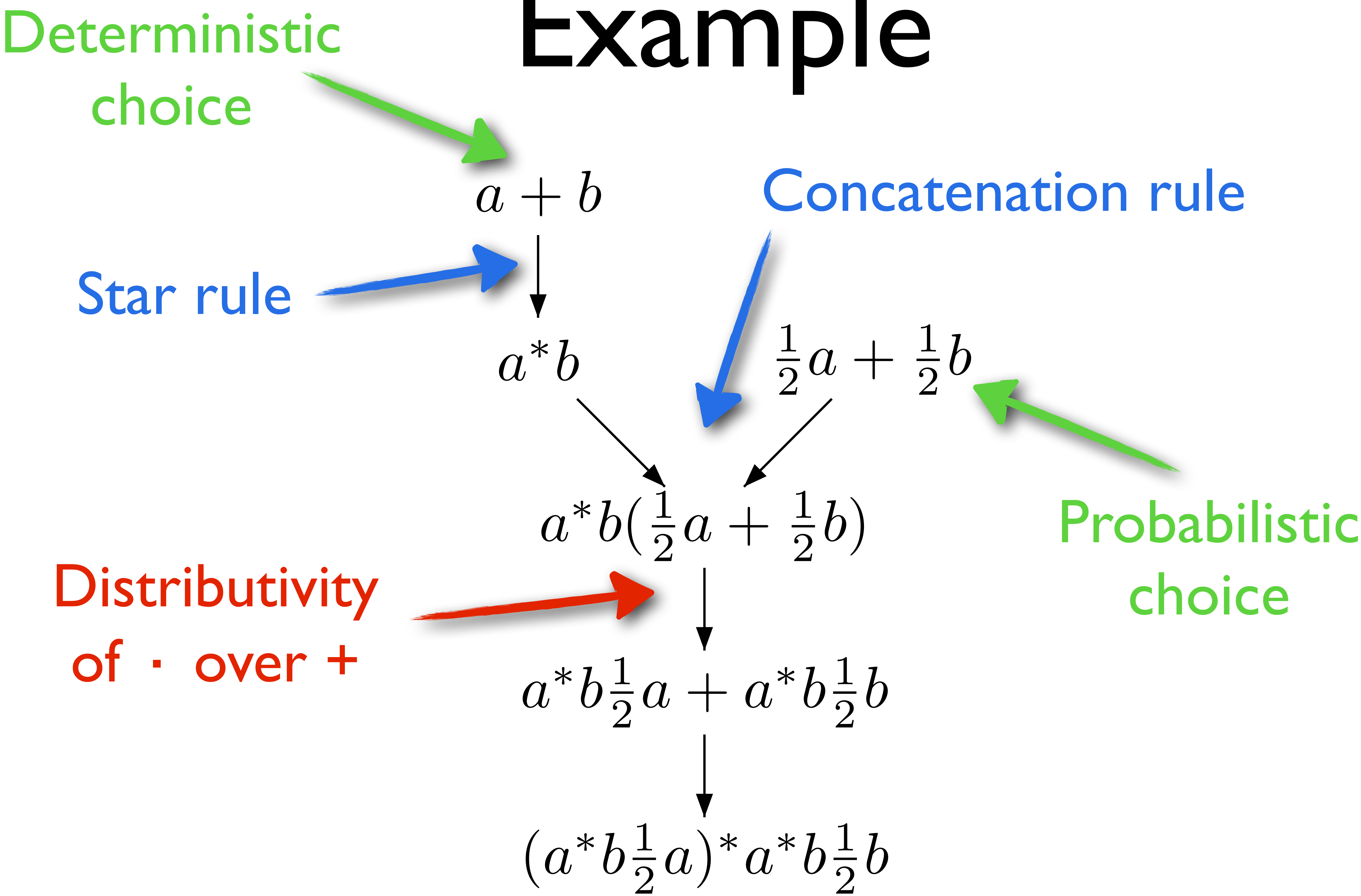


Deterministic choice
Star rule
Probabilistic choice
$a + b$
a^*b
$\frac{1}{2}a + \frac{1}{2}b$
$a^*b(\frac{1}{2}a + \frac{1}{2}b)$
$a^*b\frac{1}{2}a + a^*b\frac{1}{2}b$
$(a^*b\frac{1}{2}a)^*a^*b\frac{1}{2}b$

Example

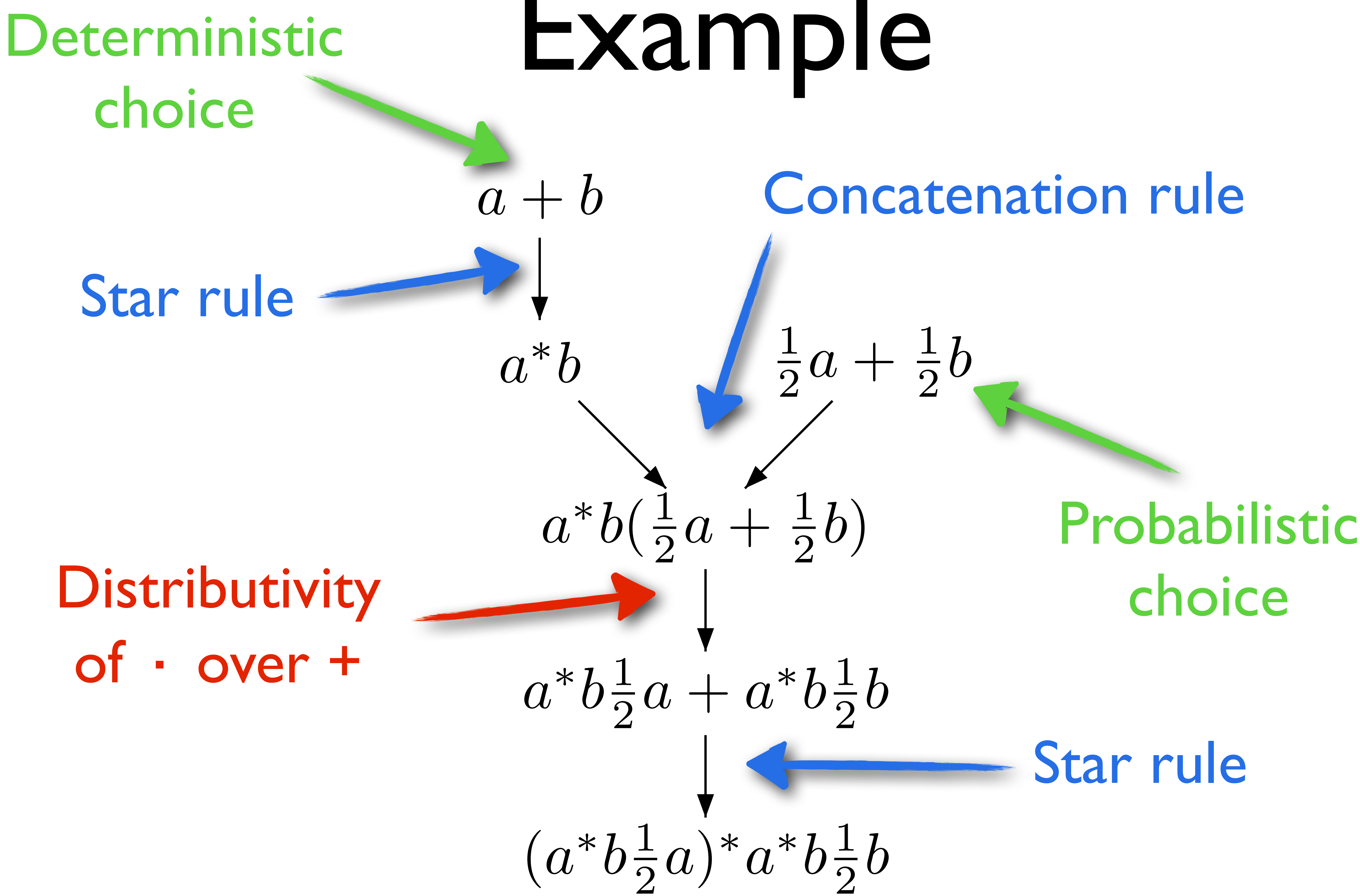

Deterministic choice
$a + b$
Star rule
a^*b
Concatenation rule
$\frac{1}{2}a + \frac{1}{2}b$
Probabilistic choice
$a^*b(\frac{1}{2}a + \frac{1}{2}b)$
$a^*b\frac{1}{2}a + a^*b\frac{1}{2}b$
$(a^*b\frac{1}{2}a)^*a^*b\frac{1}{2}b$

Example

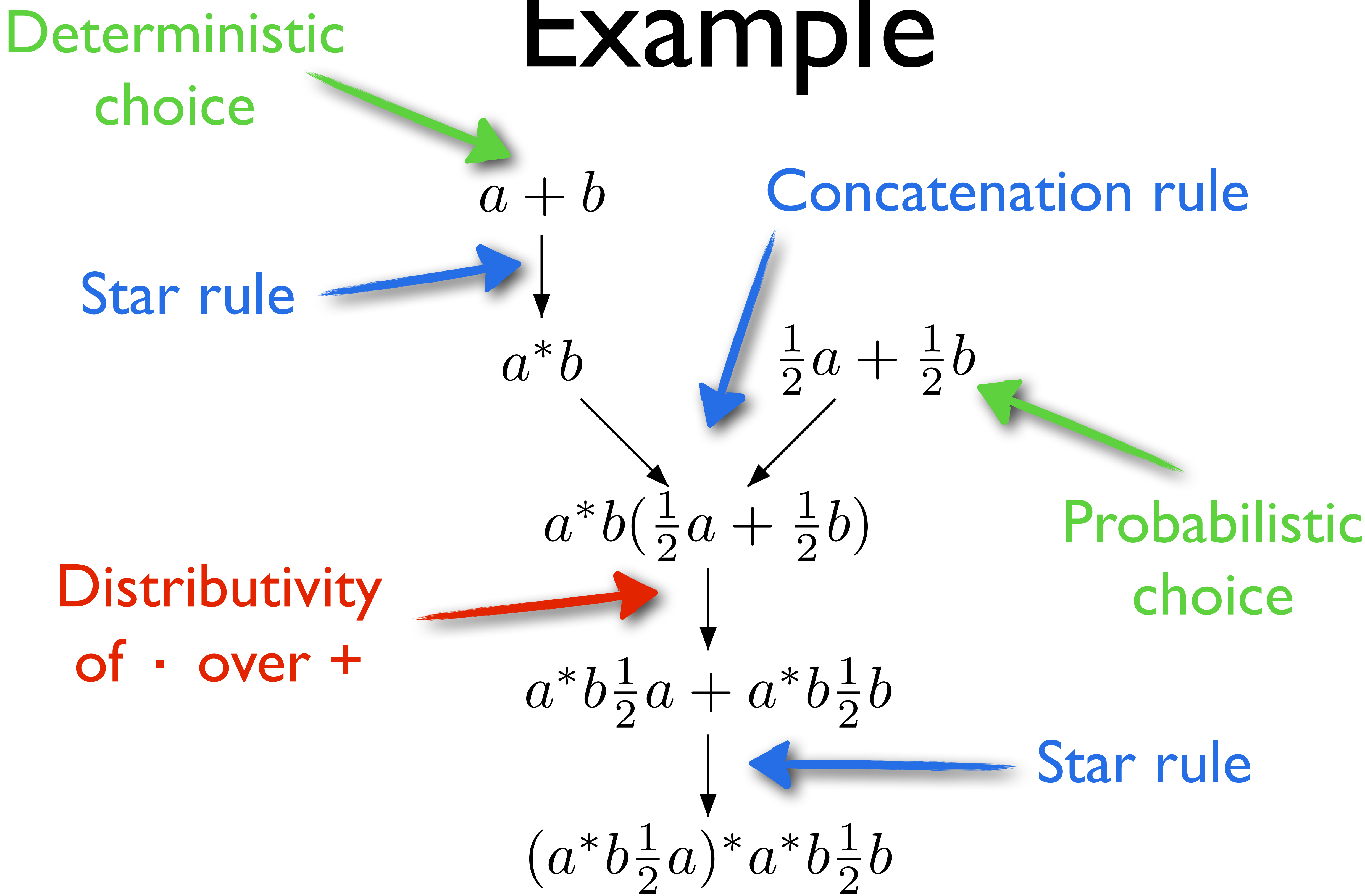


Deterministic choice
$a + b$
Star rule
$a^* b$
Concatenation rule
$\frac{1}{2}a + \frac{1}{2}b$
Probabilistic choice
$a^* b(\frac{1}{2}a + \frac{1}{2}b)$
Distributivity of · over +
$a^* b\frac{1}{2}a + a^* b\frac{1}{2}b$
$(a^* b\frac{1}{2}a)^* a^* b\frac{1}{2}b$

Example


Deterministic choice
$a + b$
Star rule
$a^* b$
Concatenation rule
$\frac{1}{2} a + \frac{1}{2} b$
Probabilistic choice
$a^* b(\frac{1}{2} a + \frac{1}{2} b)$
Distributivity of · over +
$a^* b \frac{1}{2} a + a^* b \frac{1}{2} b$
Star rule
$(a^* b \frac{1}{2} a)^* a^* b \frac{1}{2} b$

Example



The choice in the star is made *far* from the beginning...

Probabilistic Kleene-Schützenberger Theorem

- Every PRE can be translated into an equivalent Probabilistic automaton.
- Every Probabilistic automaton can be denoted by an equivalent PRE.

From Automata to Expressions

- Usual procedures (Brozozwski-McCluskey, elimination, McNaughton-Yamada...) keeping probabilistic constraints in mind
- Requires to prove some (useful) properties of PREs, e.g., if E+F and G are PREs, then E+F · G is a PRE

[1] J. A. Brzozowski and E. J. McCluskey (1963). Signal Flow Graph Techniques for Sequential Circuit State Diagrams. IEEE Trans. on Electronic Computers 12.

From Expressions to Automata

[1] V. M. Glushkov (1961). The abstract theory of automata. Russian Math. Surveys 16.
[2] G. Berry and R. Sethi (1986). From regular expressions to deterministic automata. Theoretical Computer Science 48.

From Expressions to Automata

- We have to prove closure properties of Probabilistic Automata (for example constructing standard automata, [1,2])

[1] V. M. Glushkov (1961). The abstract theory of automata. Russian Math. Surveys 16.
[2] G. Berry and R. Sethi (1986). From regular expressions to deterministic automata. Theoretical Computer Science 48.

From Expressions to Automata

- We have to prove closure properties of Probabilistic Automata (for example constructing standard automata, [1,2])
- Problem: « if E+F is a PRE, then $E^* \cdot F$ is a PRE » is not an *inductive* rule

[1] V. M. Glushkov (1961). The abstract theory of automata. Russian Math. Surveys 16.
[2] G. Berry and R. Sethi (1986). From regular expressions to deterministic automata. Theoretical Computer Science 48.

From Expressions to Automata

- We have to prove closure properties of Probabilistic Automata (for example constructing standard automata, [1,2])
- Problem: « if E+F is a PRE, then $E^* \cdot F$ is a PRE » is not an *inductive* rule
- Probabilistic Automata in a normal form: accepting states labelled by subexpressions they are computing

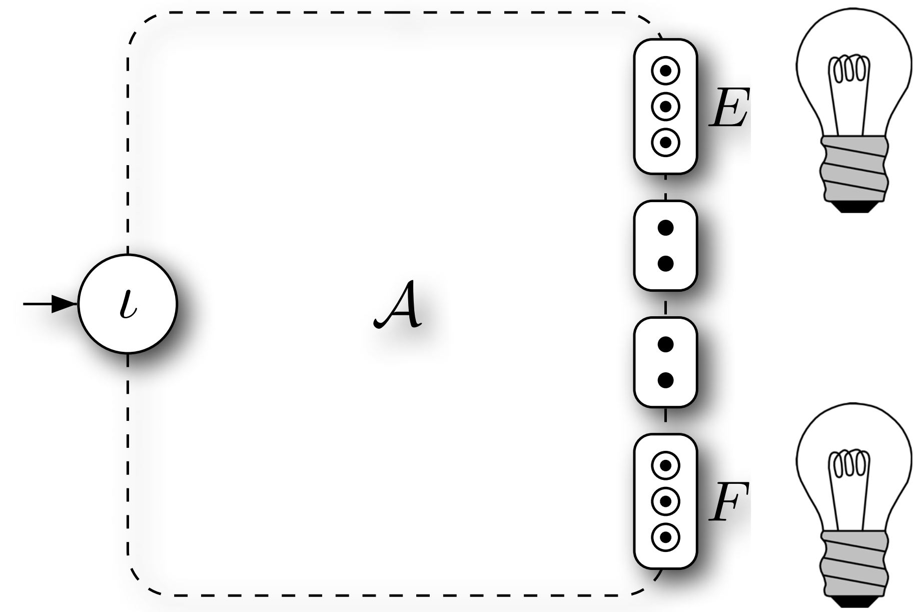
[1] V. M. Glushkov (1961). The abstract theory of automata. Russian Math. Surveys 16.
[2] G. Berry and R. Sethi (1986). From regular expressions to deterministic automata. Theoretical Computer Science 48.

From Expressions to Automata

- We have to prove closure properties of Probabilistic Automata (for example constructing standard automata, [1,2])

- Problem: « if E+F is a PRE, then $E^* \cdot F$ is a PRE » is not an *inductive* rule

- Probabilistic Automata in a normal form: accepting states labelled by subexpressions they are computing



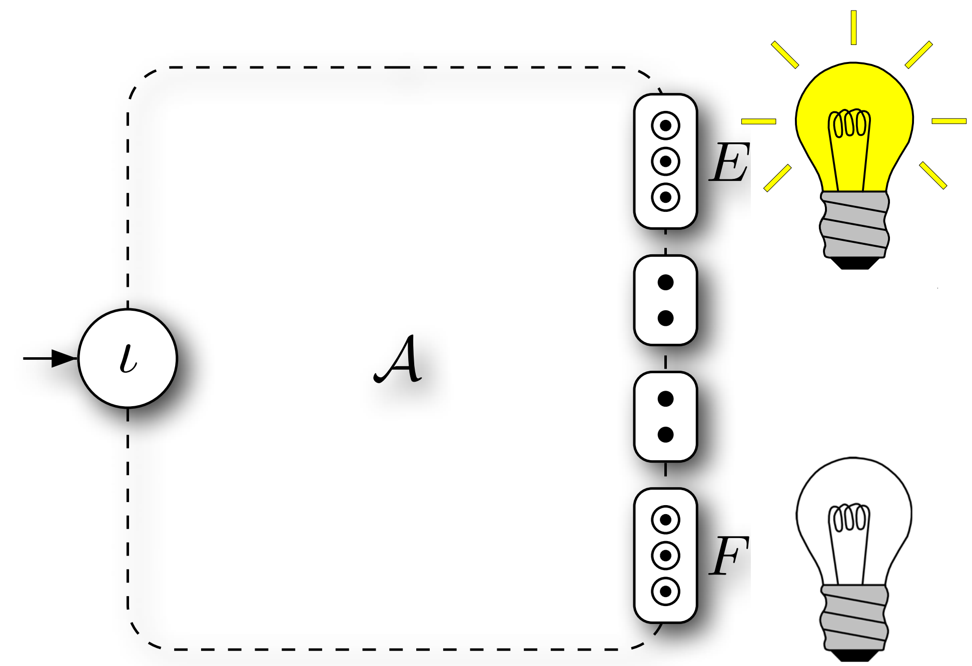
[1] V. M. Glushkov (1961). The abstract theory of automata. Russian Math. Surveys 16.
[2] G. Berry and R. Sethi (1986). From regular expressions to deterministic automata. Theoretical Computer Science 48.

From Expressions to Automata

- We have to prove closure properties of Probabilistic Automata (for example constructing standard automata, [1,2])

- Problem: « if E+F is a PRE, then $E^* \cdot F$ is a PRE » is not an *inductive* rule

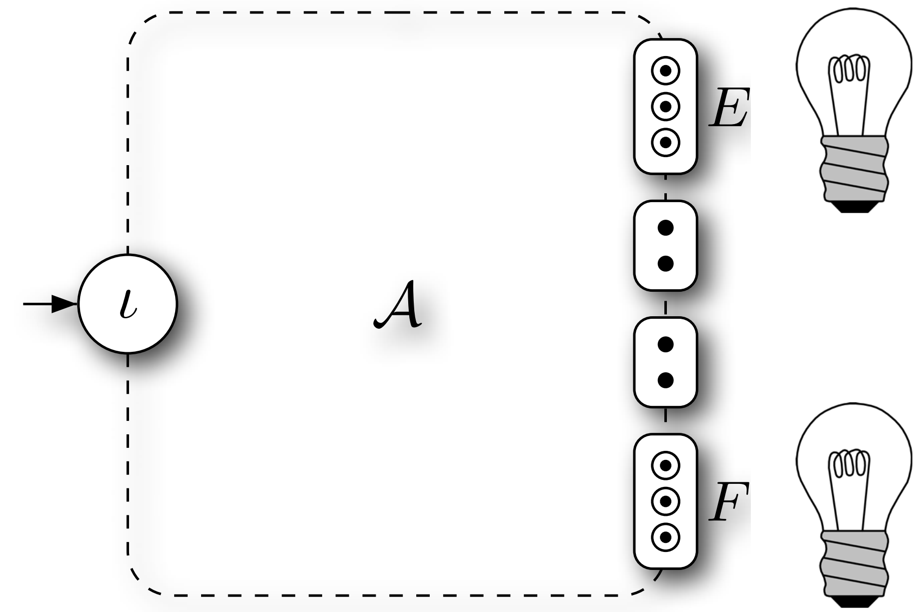
- Probabilistic Automata in a normal form: accepting states labelled by subexpressions they are computing



[1] V. M. Glushkov (1961). The abstract theory of automata. Russian Math. Surveys 16.
[2] G. Berry and R. Sethi (1986). From regular expressions to deterministic automata. Theoretical Computer Science 48.

From Expressions to Automata

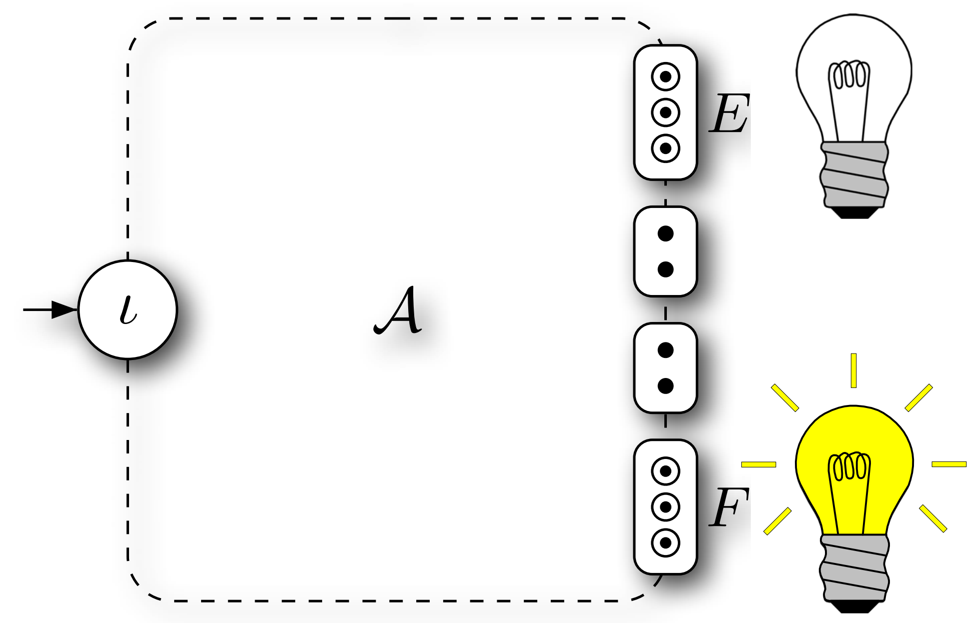
- We have to prove closure properties of Probabilistic Automata (for example constructing standard automata, [1,2])
- Problem: « if E+F is a PRE, then $E^* \cdot F$ is a PRE » is not an *inductive* rule
- Probabilistic Automata in a normal form: accepting states labelled by subexpressions they are computing



[1] V. M. Glushkov (1961). The abstract theory of automata. Russian Math. Surveys 16.
[2] G. Berry and R. Sethi (1986). From regular expressions to deterministic automata. Theoretical Computer Science 48.

From Expressions to Automata

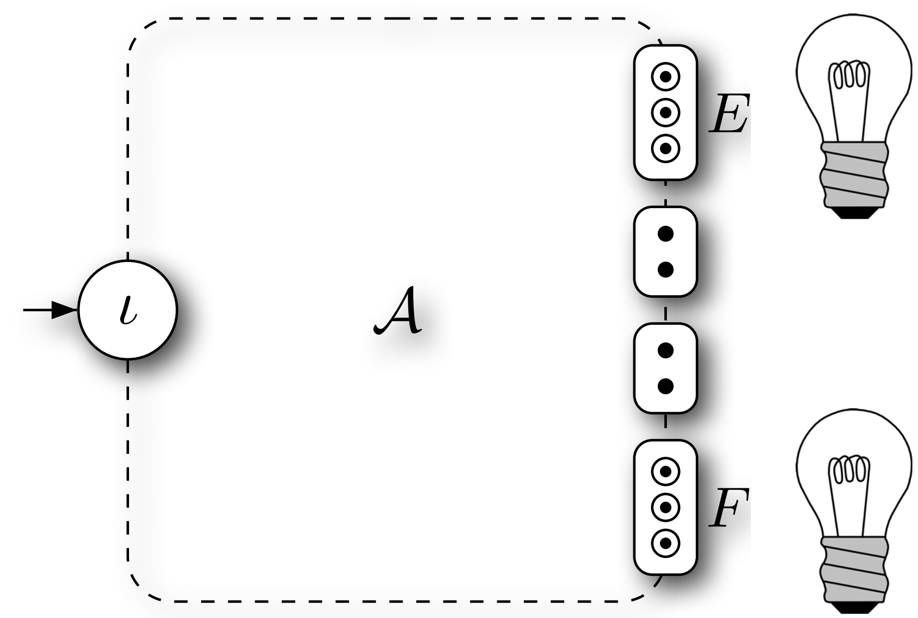
- We have to prove closure properties of Probabilistic Automata (for example constructing standard automata, [1,2])
- Problem: « if E+F is a PRE, then $E^* \cdot F$ is a PRE » is not an *inductive* rule
- Probabilistic Automata in a normal form: accepting states labelled by subexpressions they are computing



[1] V. M. Glushkov (1961). The abstract theory of automata. Russian Math. Surveys 16.
[2] G. Berry and R. Sethi (1986). From regular expressions to deterministic automata. Theoretical Computer Science 48.

From Expressions to Automata

- We have to prove closure properties of Probabilistic Automata (for example constructing standard automata, [1,2])
- Problem: « if E+F is a PRE, then $E^* \cdot F$ is a PRE » is not an *inductive* rule
- Probabilistic Automata in a normal form: accepting states labelled by subexpressions they are computing



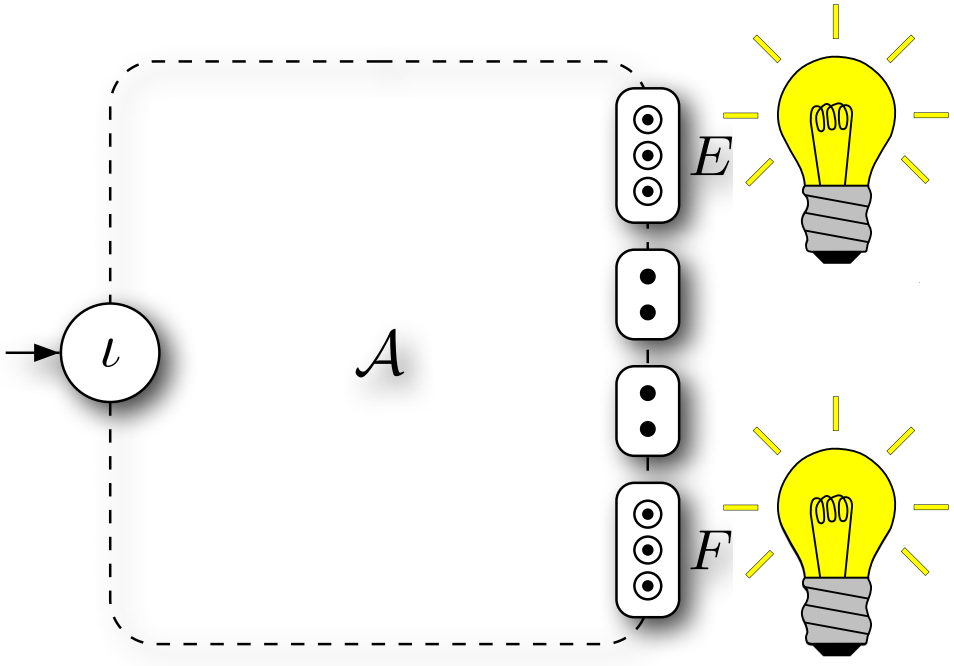
[1] V. M. Glushkov (1961). The abstract theory of automata. Russian Math. Surveys 16.
[2] G. Berry and R. Sethi (1986). From regular expressions to deterministic automata. Theoretical Computer Science 48.

From Expressions to Automata

- We have to prove closure properties of Probabilistic Automata (for example constructing standard automata, [1,2])

- Problem: « if E+F is a PRE, then $E^* \cdot F$ is a PRE » is not an *inductive* rule

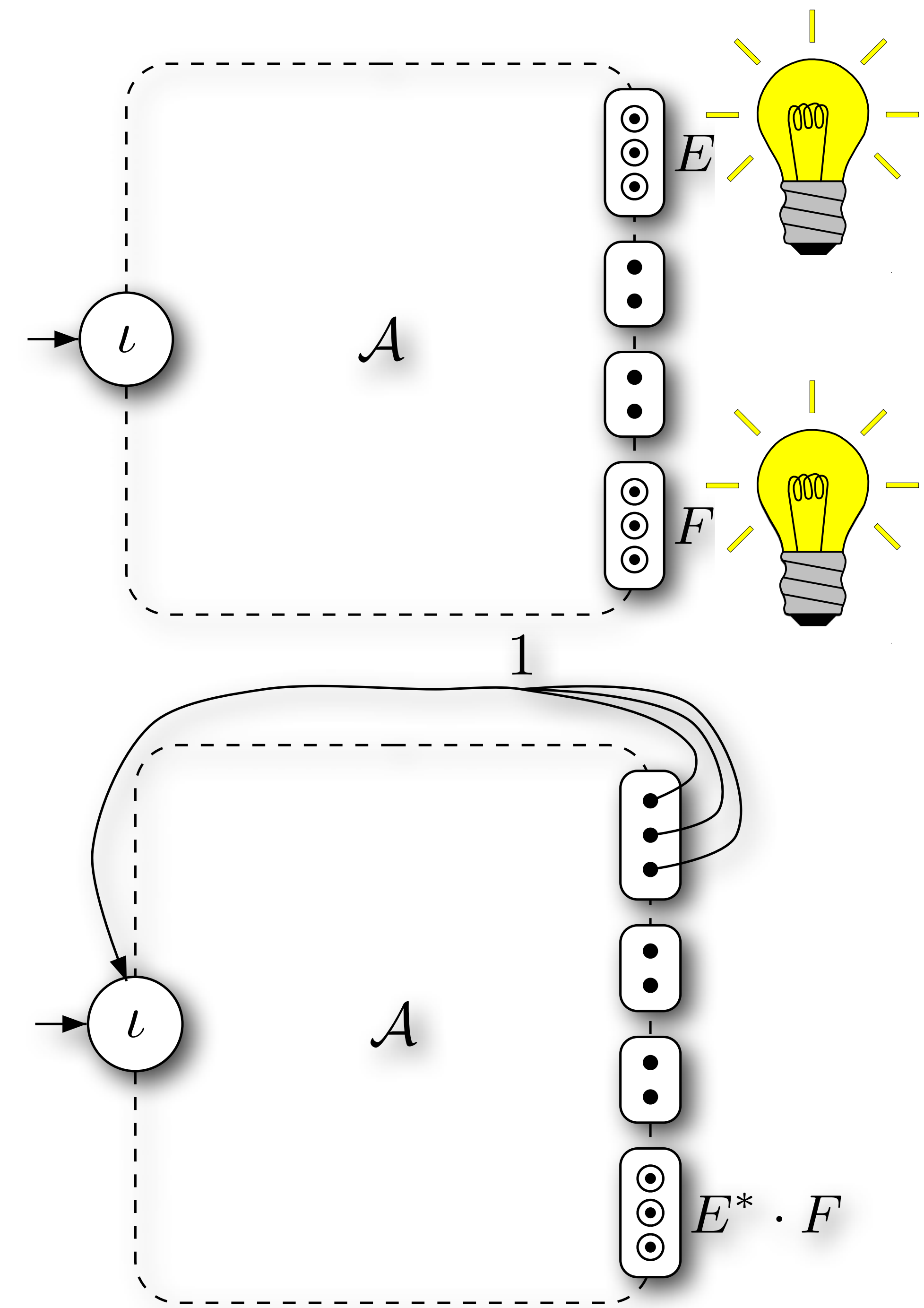
- Probabilistic Automata in a normal form: accepting states labelled by subexpressions they are computing



[1] V. M. Glushkov (1961). The abstract theory of automata. Russian Math. Surveys 16.
[2] G. Berry and R. Sethi (1986). From regular expressions to deterministic automata. Theoretical Computer Science 48.

From Expressions to Automata

- We have to prove closure properties of Probabilistic Automata (for example constructing standard automata, [1,2])
- Problem: « if E+F is a PRE, then $E^* \cdot F$ is a PRE » is not an *inductive* rule
- Probabilistic Automata in a normal form: accepting states labelled by subexpressions they are computing



[1] V. M. Glushkov (1961). The abstract theory of automata. Russian Math. Surveys 16.
[2] G. Berry and R. Sethi (1986). From regular expressions to deterministic automata. Theoretical Computer Science 48.

Corollaries

- Equivalence problem for PREs is decidable: given PREs E and F, does they generate the same semantics? (translation into automata [1])
- Threshold problem for PREs is undecidable: given a PRE E and a threshold *s*, is there a word *w* which is mapped by to a probability greater than *s*? (by reduction to automata [2])

[1] M.-P. Schützenberger (1961). On the Definition of a Family of Automata. Information and Control.
[2] A. Paz. (1971). Introduction to probabilistic automata. Academic Press,

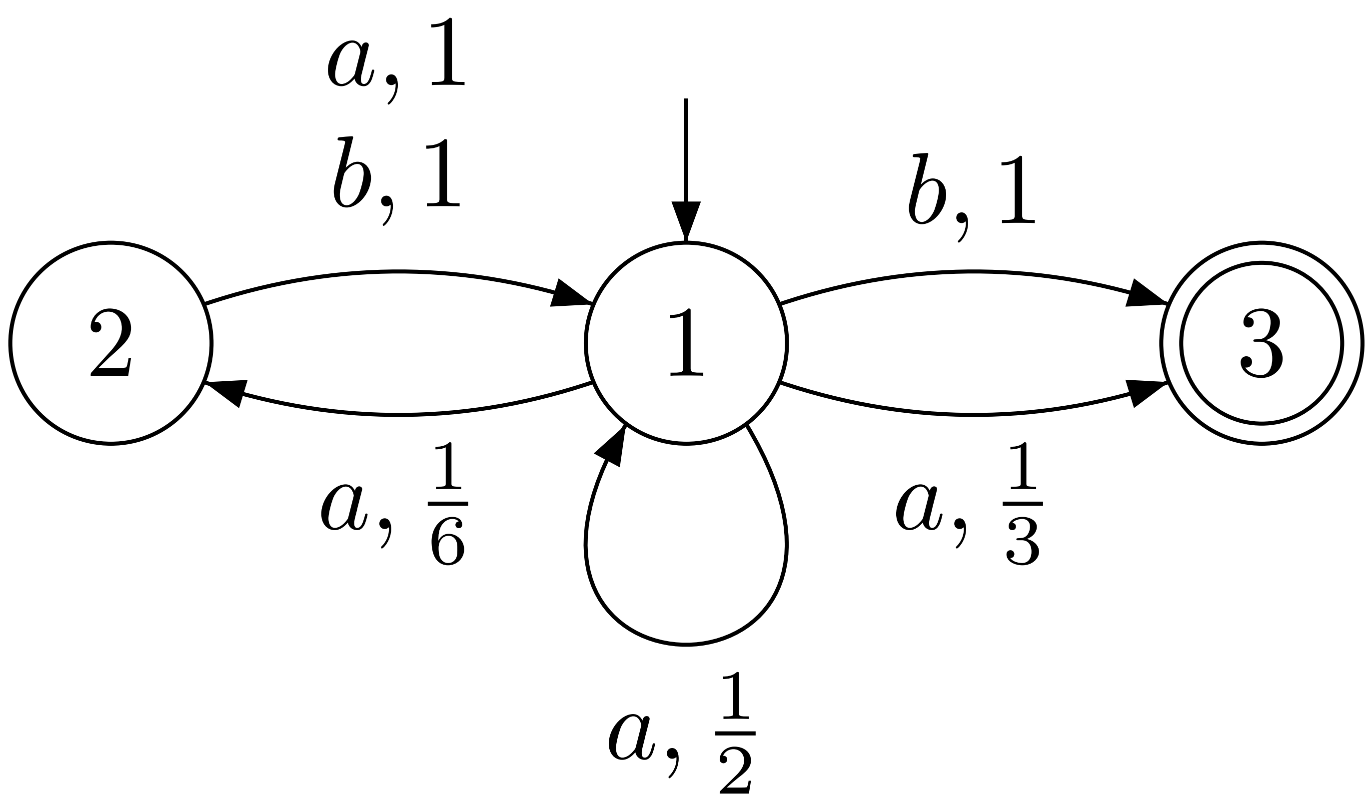
Summary and Future Works

- General Kleene-Schützenberger theorems for Probabilistic models (classical, extended to two-way automata, pebble automata in full paper [1])
- Study of Probabilistic Expressions and their extensions permits us to better understand which behavior Probabilistic Automata can generate
- In [2], we proved that Weighted Automata (with two-way and pebbles) can be evaluated efficiently
- Future work: get logical formalisms generating the same expressivity, and implement quick algorithms to perform translation from PREs to PAs (as there are some for weighted automata, see [2,3] e.g.)

[1] B. Bollig, P. Gastin, B. M. and M. Zeitoun. (2012). A Probabilistic Kleene Theorem. In Proceedings of ATVA'12.
[2] P. Gastin and B. M. (2006). Adding Pebbles to Weighted Automata. In Proceedings of CIAA'12.
[3] C. Allauzen, and M., Mohri, (2006). A Unified Construction of the Glushkov, Follow, and Antimirov Automata. In Proceedings of MFCS'06

Automata Model

Usual Rabin automata...



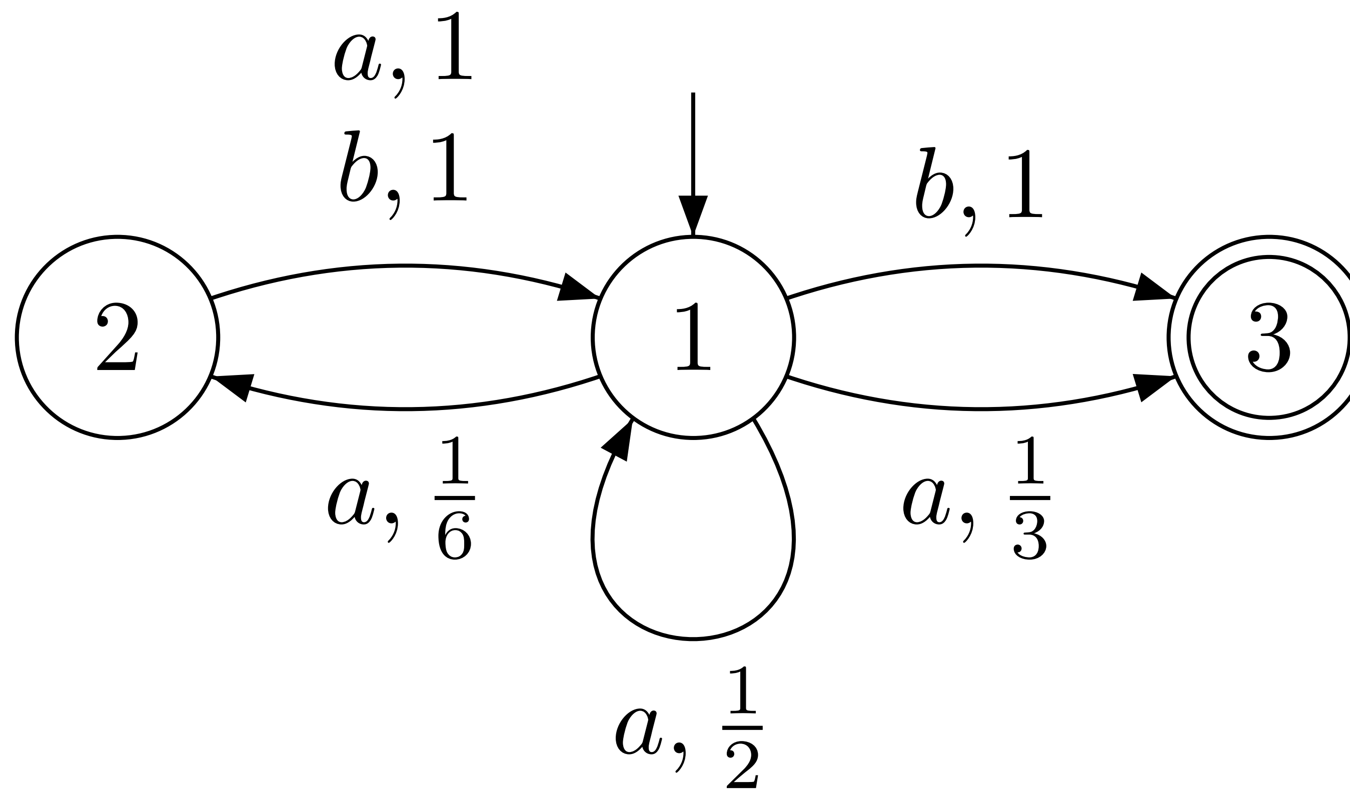
$$\mathcal{A} = (Q, \iota, Acc, \mathbb{P})$$

$$\mathbb{P} : Q \times \Sigma \times Q \to [0, 1]$$

$$Acc(q) + \sum_{q' \in Q} \mathbb{P}(q, a, q') \leq 1 \text{ for all } (q, a) \in Q \times A$$

Automata Model

Usual Rabin automata...



$$\mathcal{A} = (Q, \iota, Acc, \mathbb{P})$$

$$\mathbb{P} : Q \times \Sigma \times Q \to [0, 1]$$

$$Acc(q) + \sum_{q' \in Q} \mathbb{P}(q, a, q') \leq 1 \text{ for all } (q, a) \in Q \times A$$

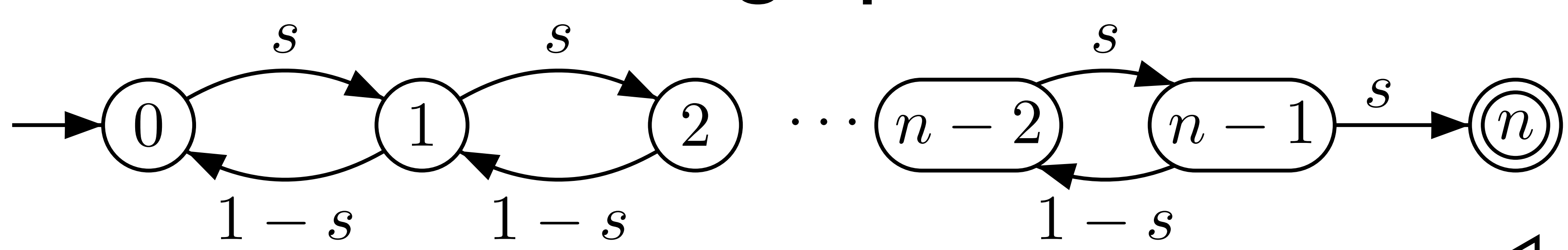
GOAL: Remove all trace of non-determinism

- seems to be a strong restriction

+ indeed we can drop it using a right marker $\triangleleft$ in words

2-way Probabilistic Expressions

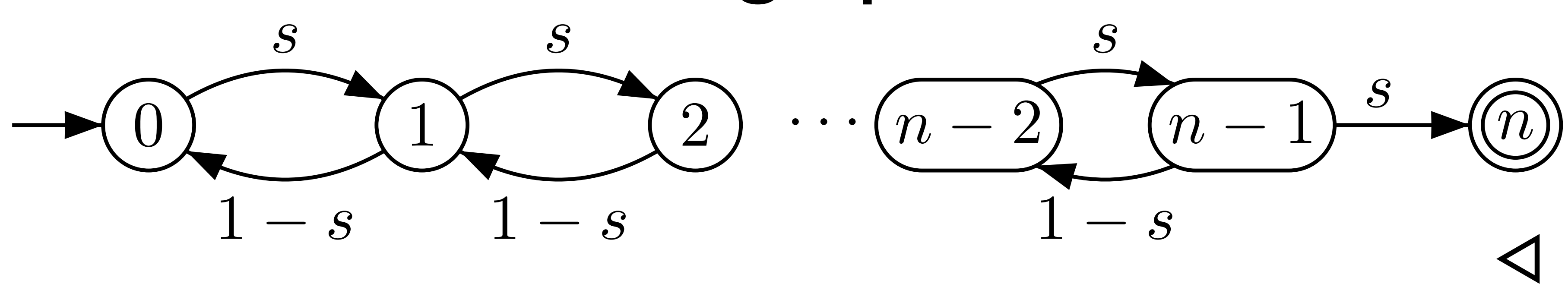
Random Walk over a finite linear graph



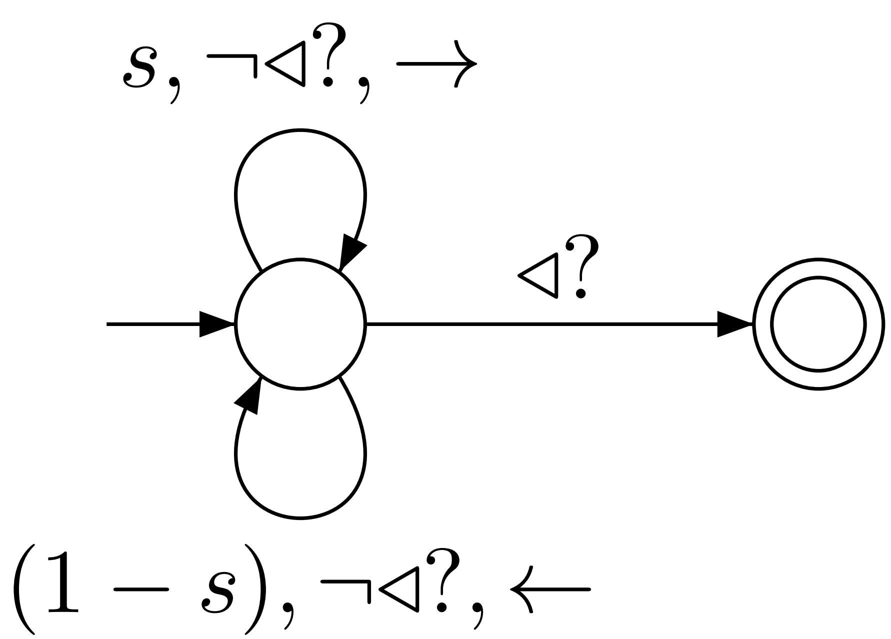
◁

2-way Probabilistic Expressions

Random Walk over a finite linear graph

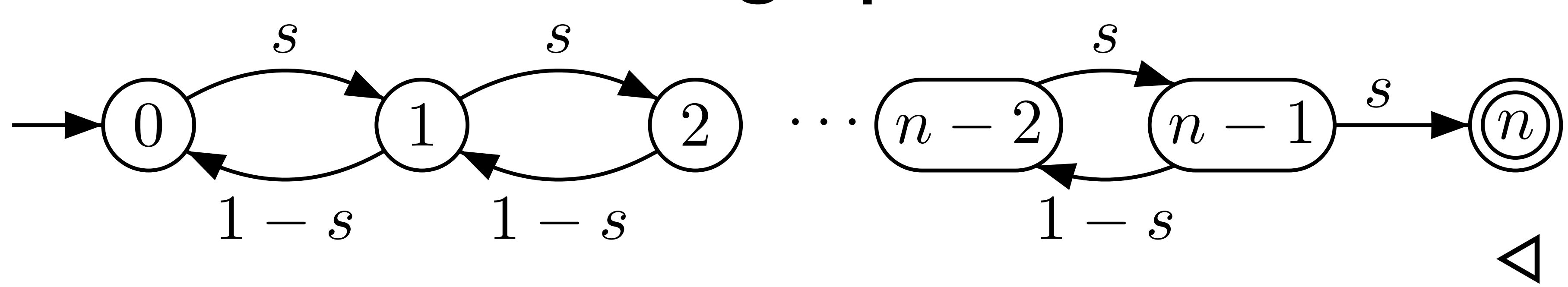


Expressible with
Probabilistic 2-way Automata

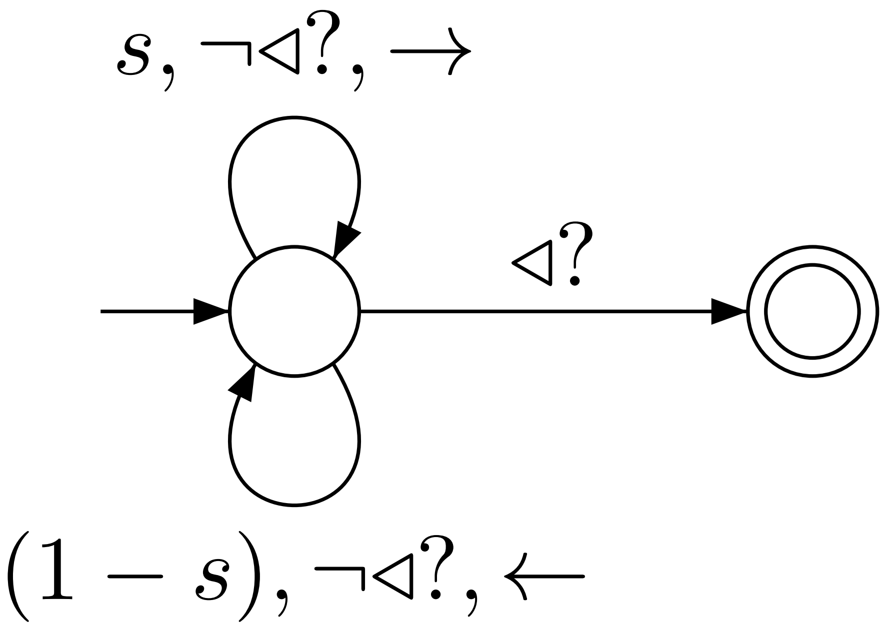


2-way Probabilistic Expressions

Random Walk over a finite linear graph



Expressible with
Probabilistic 2-way Automata

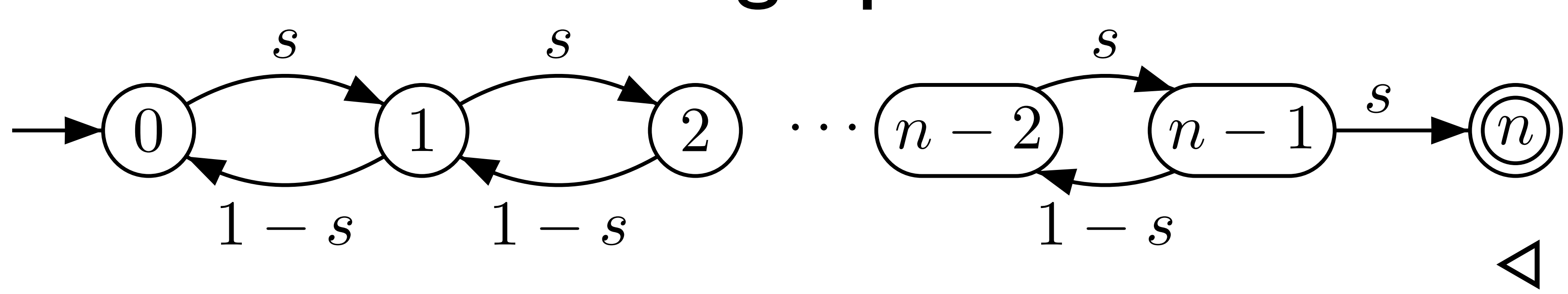


Expressible with
Probabilistic 2-way Expressions

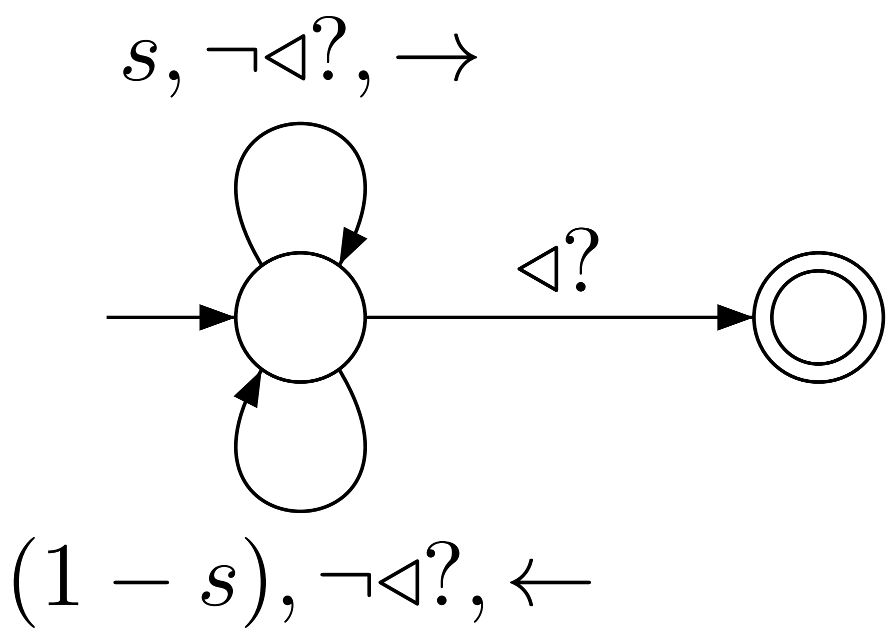
$$E = \left(\neg\triangleleft?\, s\!\rightarrow + \neg\triangleleft?(1-s)\!\leftarrow\right)^{*}\triangleleft?$$

2-way Probabilistic Expressions

Random Walk over a finite linear graph



Expressible with
Probabilistic 2-way Automata



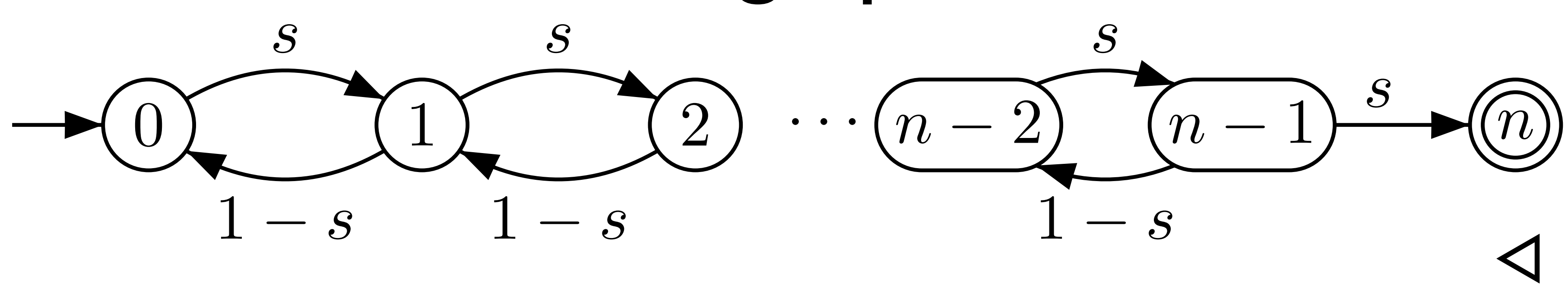
Expressible with
Probabilistic 2-way Expressions

$$E = \left(\neg\triangleleft? s \rightarrow + \neg\triangleleft?(1-s)\leftarrow\right)^* \triangleleft?$$

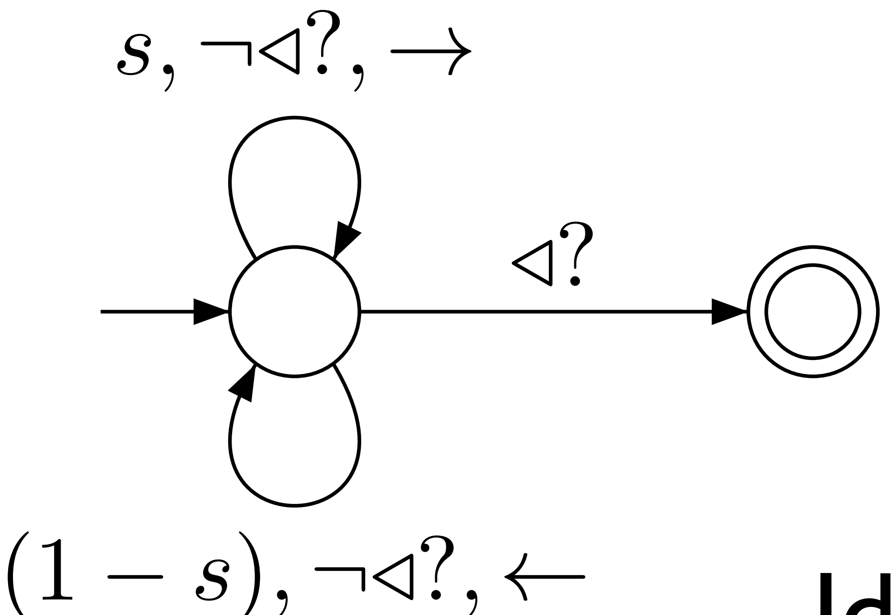
Not expressible with
Probabilistic Expressions / Probabilistic Automata

2-way Probabilistic Expressions

Random Walk over a finite linear graph



Expressible with
Probabilistic 2-way Automata



Expressible with
Probabilistic 2-way Expressions

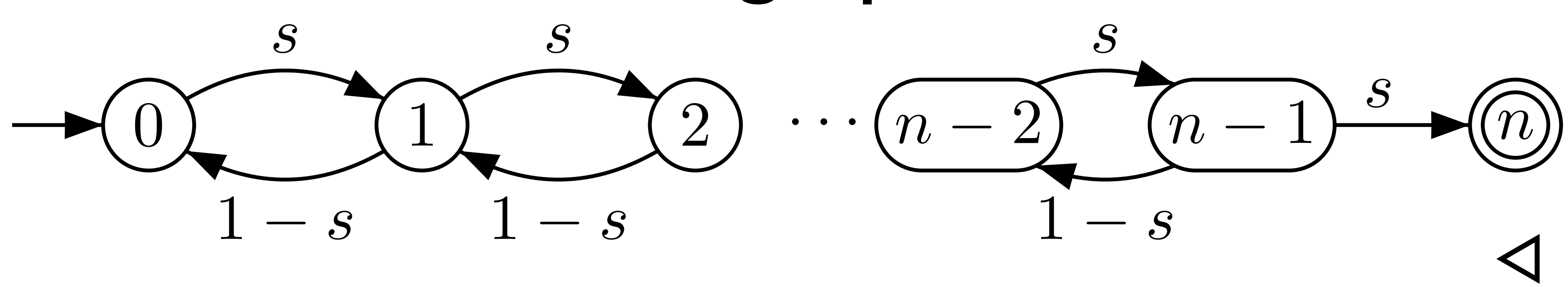
$$E = \left(\neg\triangleleft?s\!\rightarrow + \neg\triangleleft?(1-s)\!\leftarrow\right)^{*}\triangleleft?$$

Not expressible with
Probabilistic Expressions / Probabilistic Automata

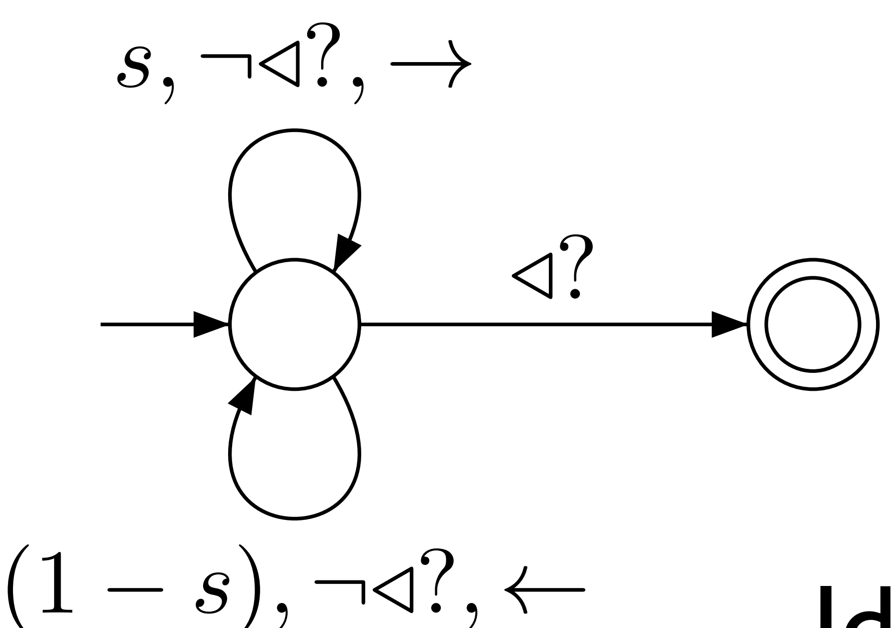
Idea: replace every letter *a* by a test *a?*
followed by a move (either → or ←)

2-way Probabilistic Expressions

Random Walk over a finite linear graph



Expressible with
Probabilistic 2-way Automata



Expressible with
Probabilistic 2-way Expressions

$$E = \left(\neg\triangleleft ?\, s\!\rightarrow + \neg\triangleleft ?\,(1-s)\!\leftarrow\right)^{*} \triangleleft ?$$

Not expressible with
Probabilistic Expressions / Probabilistic Automata

Idea: replace every letter *a* by a test *a?*
followed by a move (either → or ←)

Expressiveness result still holds!

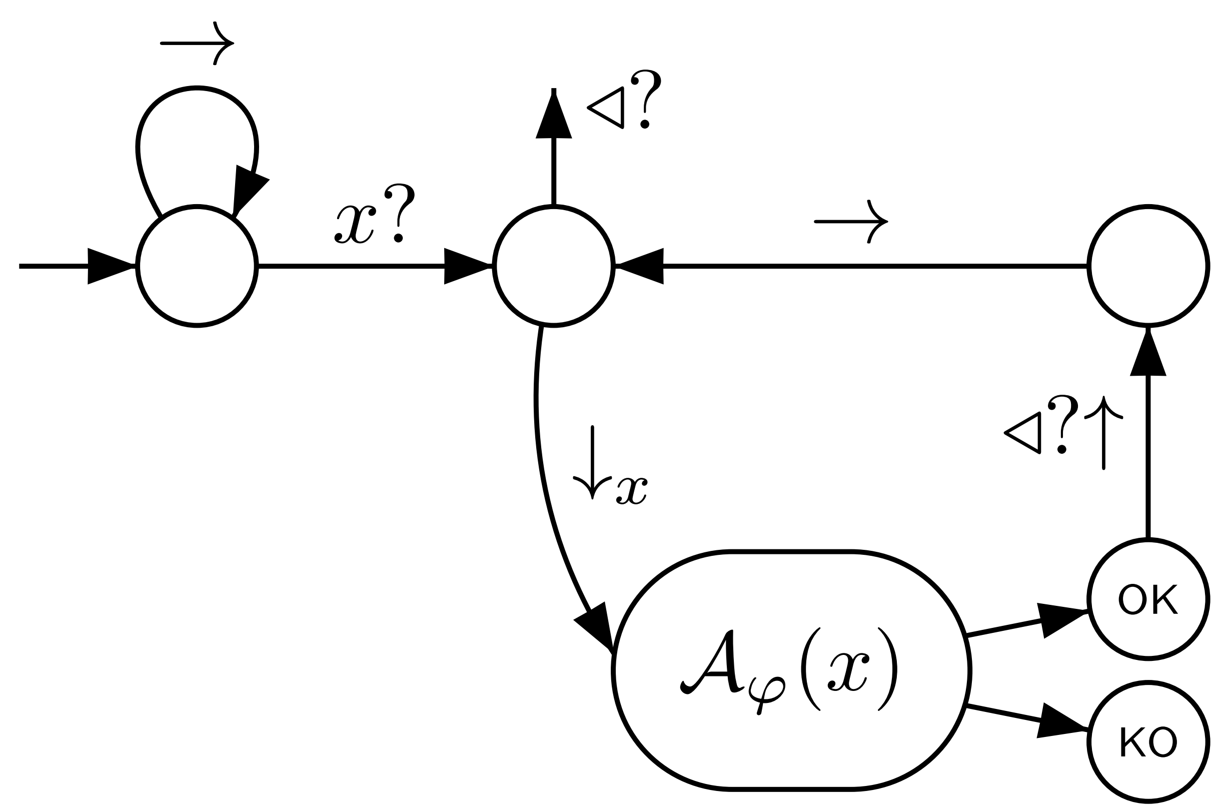
Adding Pebbles: pLTL

Each LTL formula φ has an implicit free variable *x* denoting the position where the formula is evaluated. We use a pebble to mark this position.

Let *P*(φ, *u*, *i*) denote the probability that φ holds on word *u* at position *i*.

$$\mathbb{P}(\mathsf{G}\,\varphi, u, i) \quad = \quad \prod_{j \geq i} \mathbb{P}(\varphi, u, j)$$

$$\mathcal{A}_{\mathsf{G}\,\varphi}(x) \quad =$$



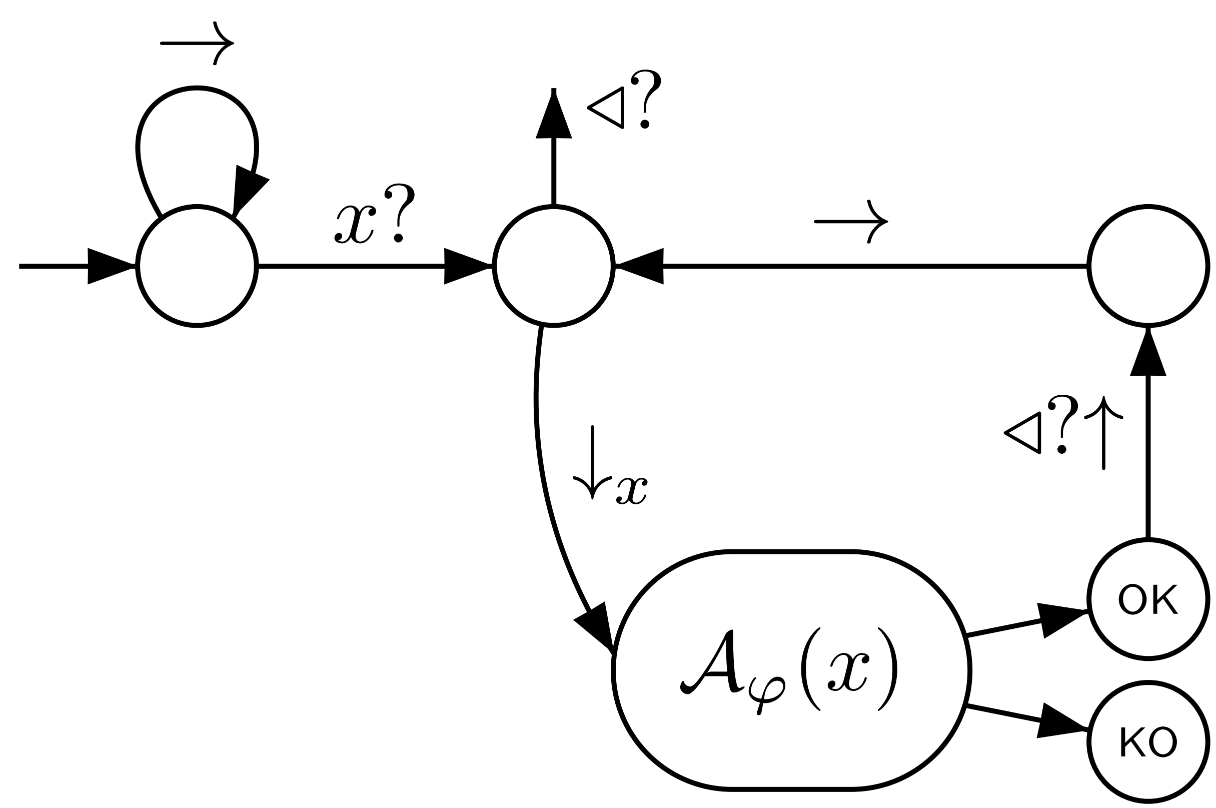
Adding Pebbles: pLTL

Each LTL formula φ has an implicit free variable *x* denoting the position where the formula is evaluated. We use a pebble to mark this position.

Let *P*(φ, *u*, *i*) denote the probability that φ holds on word *u* at position *i*.

$$\mathbb{P}(\mathsf{G}\,\varphi, u, i) \quad = \quad \prod_{j \geq i} \mathbb{P}(\varphi, u, j)$$

$$\mathcal{A}_{\mathsf{G}\,\varphi}(x) \quad =$$



$$E_{\mathsf{G}\,\varphi}(x) = \triangleright ? \rightarrow^* x?\big((x!E_\varphi(x))\rightarrow\big)^* \triangleleft ?$$

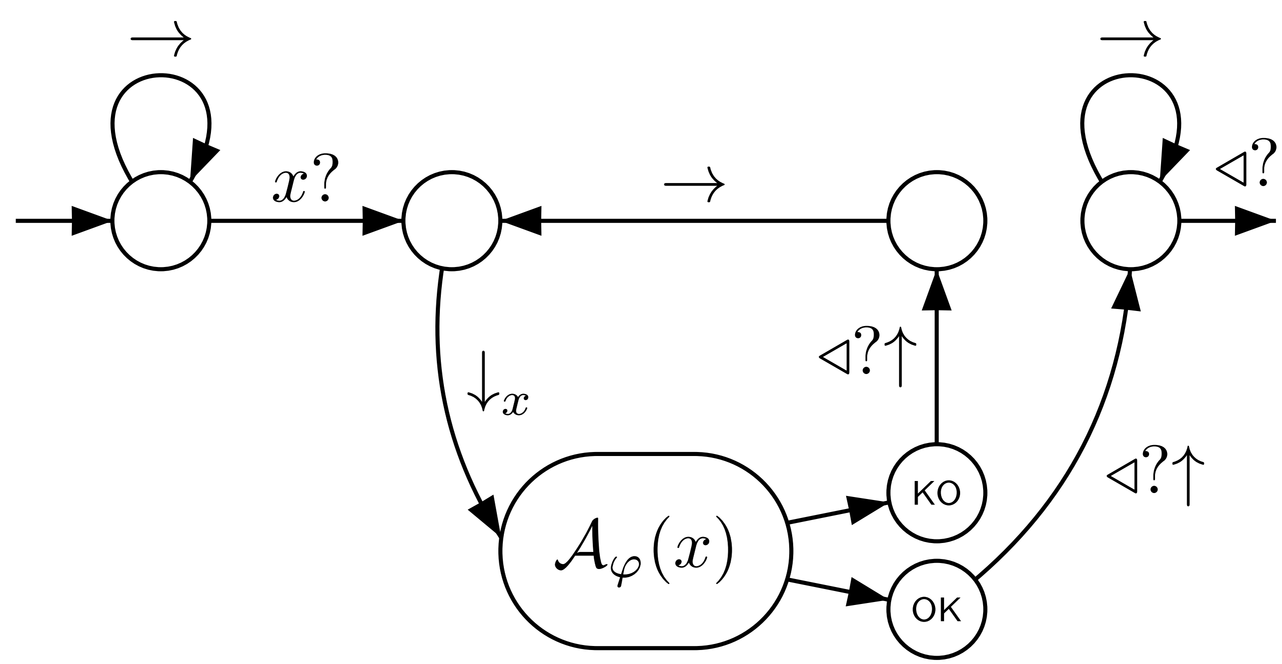
Adding Pebbles: pLTL

Each LTL formula φ has an implicit free variable *x* denoting the position where the formula is evaluated. We use a pebble to mark this position.

Let *P*(φ, *u*, *i*) denote the probability that φ holds on word *u* at position *i*.

$$\begin{aligned}\mathbb{P}(\mathsf{F}\,\varphi, u, i) &= \mathbb{P}(\varphi, u, i) + (1 - \mathbb{P}(\varphi, u, i)) \times \mathbb{P}(\mathsf{F}\,\varphi, u, i+1) \\ &= \sum_{j \geq i} \left(\prod_{i \leq k < j} \mathbb{P}(\neg\varphi, u, k) \right) \times \mathbb{P}(\varphi, u, j)\end{aligned}$$

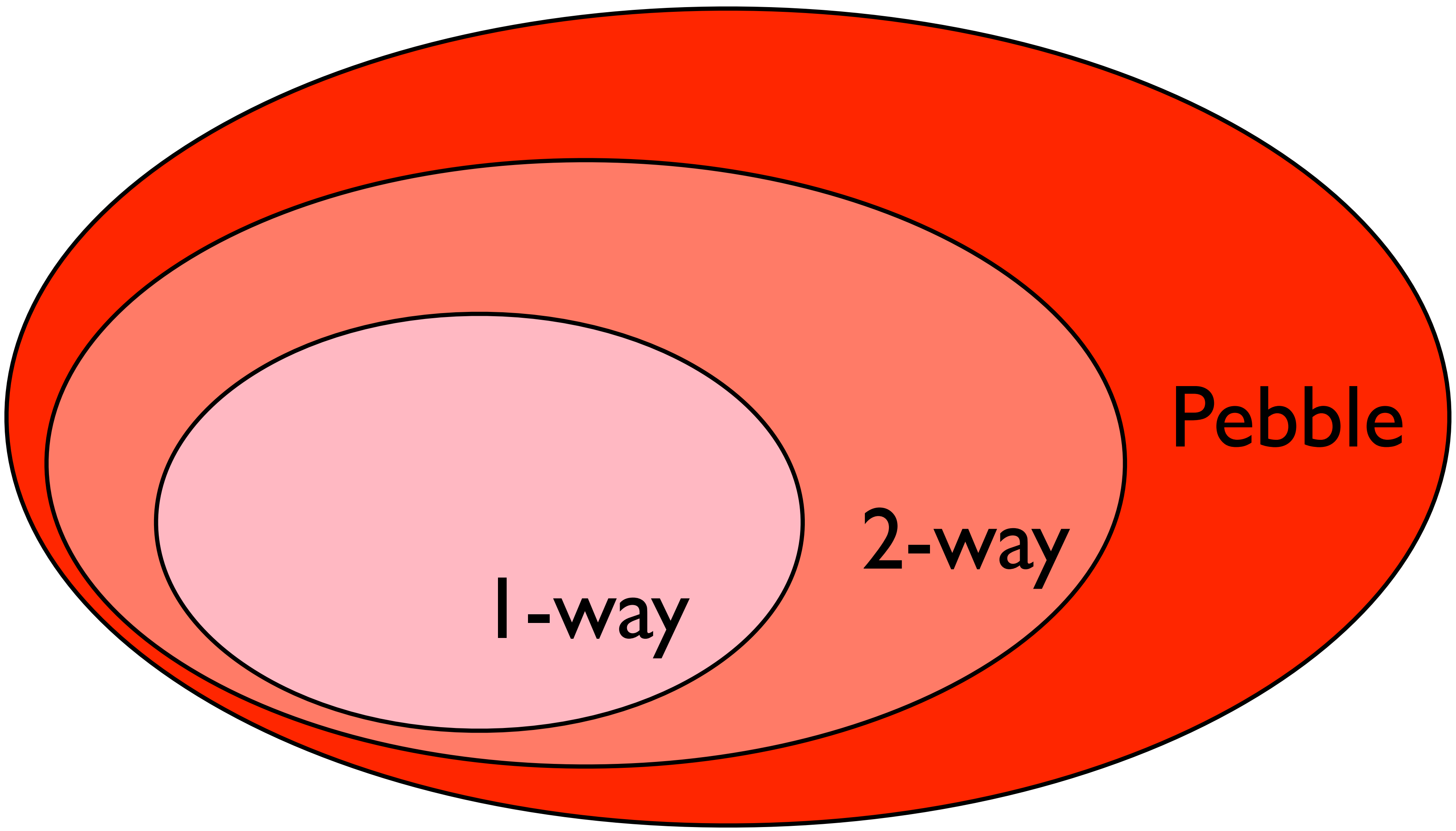
$\mathcal{A}_{\mathsf{F}\,\varphi}(x) =$



$$E_{\mathsf{F}\,\varphi}(x) = \triangleright ? \rightarrow^* x ? \Big((x ! E_{\neg\varphi}(x)) \rightarrow \Big)^* (x ! E_\varphi(x)) \rightarrow^* \triangleleft ?$$

Theorem

- PREs and PAs are expressively equivalent.
- 2-way PREs and 2-way PAs are expressively equivalent.
- Pebble PREs and Pebble PAs are expressively equivalent.



Extensions

- Add 2-way and pebbles in automata and expressions (XPath-like syntax)
- Possibility to *express* more, e.g. *smaller* probabilities (to represent rare events)
- Still a natural way to denote probabilistic properties about words

Conclusion

- General Kleene-Schützenberger theorems for Probabilistic models (classical, two-way, pebbles...)
- Study of Probabilistic Expressions and their extensions permits us to better understand which behavior Probabilistic Automata can generate
- In [1], we proved that Weighted Automata with two-way and pebbles can be evaluated efficiently
- Future work: get logical formalisms generating the same expressivity, and implement quick algorithms to perform translation from PREs to PAs (as there are some for weighted automata, see [2,1] e.g.)

[1] P. Gastin and B. M. (2006). Adding Pebbles to Weighted Automata. In Proceedings of CIAA'12.
[2] C. Allauzen, and M., Mohri, (2006). A Unified Construction of the Glushkov, Follow, and Antimirov Automata. In Proceedings of MFCS'06