The Language, the Expression, and the (small) Automaton

Jacques Sakarovitch

CNRS / ENST

This talk is based on an invited talk I gave at the CIAA 2005 conference in Nice (France).

The third part: 'Can expressions code for automata' is based on a joint work with Sylvain Lombardy, published in *RAIRO — Theoretical Informatics and Aplications* **39** (2005).

Part I

Plato's cave of formal language theory

First notation

First notation

In what follows,

First notation

In what follows,

- A is an *alphabet*, *i.e.* a finite set of *letters*.

First notation

In what follows,

- A is an *alphabet*, *i.e.* a finite set of *letters*.
- A^* is the set of *words*.

First notation

In what follows,

- A is an *alphabet*, *i.e.* a finite set of *letters*.
- A^* is the set of *words*.
- Any subset L of A^* is a *language*.

First notation

In what follows,

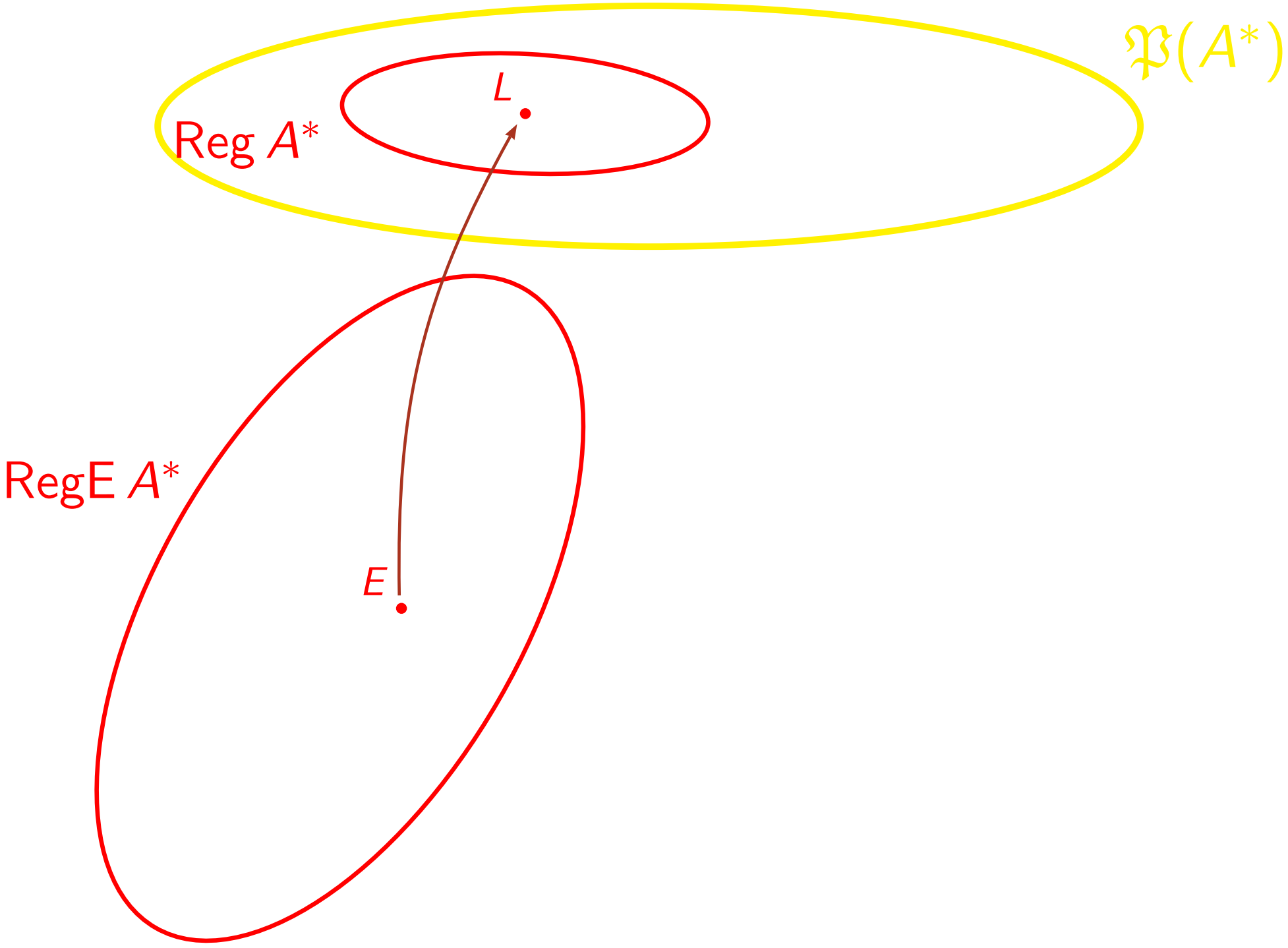
- A is an *alphabet*, *i.e.* a finite set of *letters*.
- A^* is the set of *words*.
- Any subset L of A^* is a *language*.
- The set of *regular expressions* over A^* is denoted $\mathrm{RegE}\, A^*$; the set of *regular languages* over A^* is denoted $\mathrm{Reg}\, A^*$.

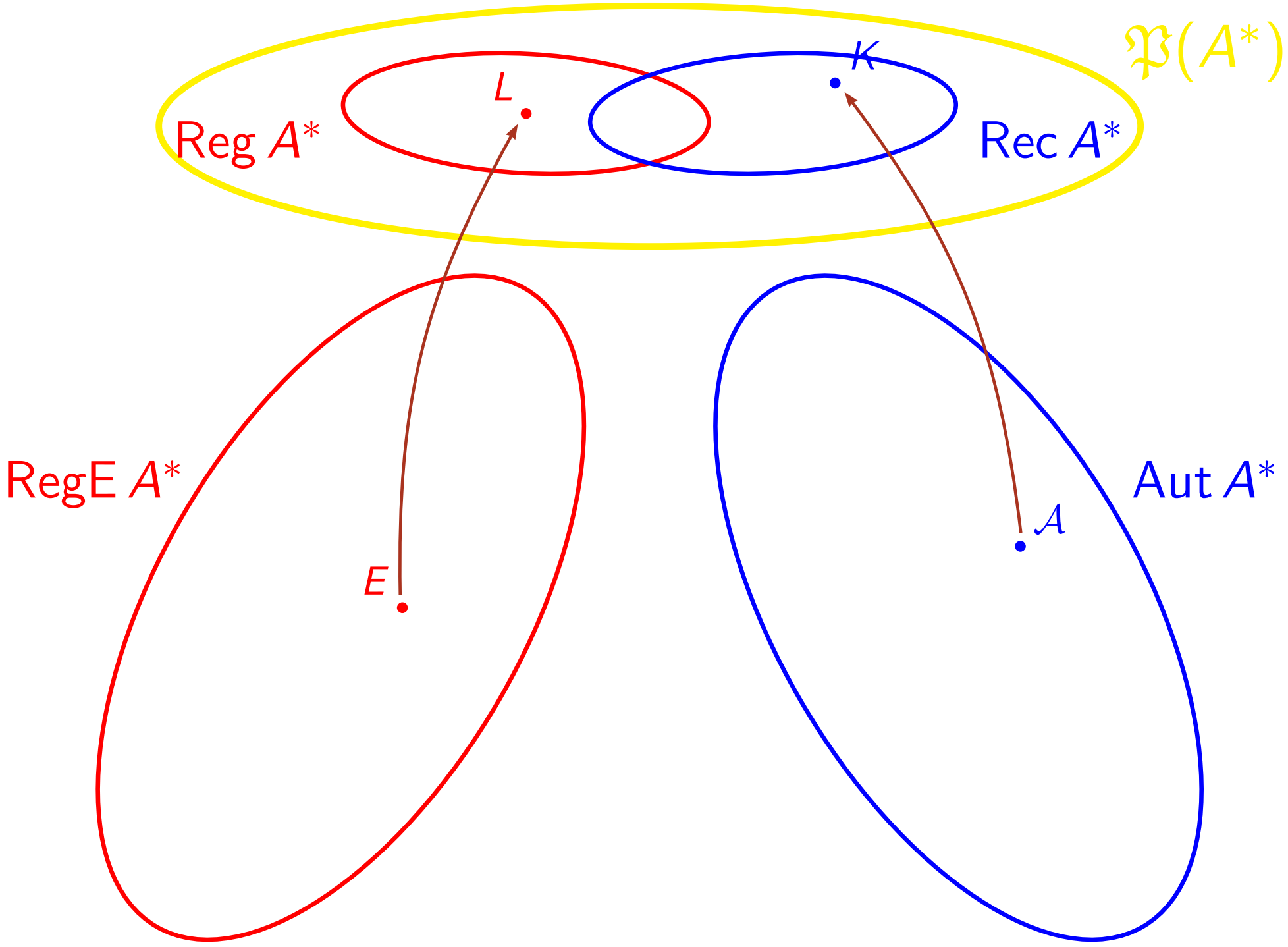
First notation

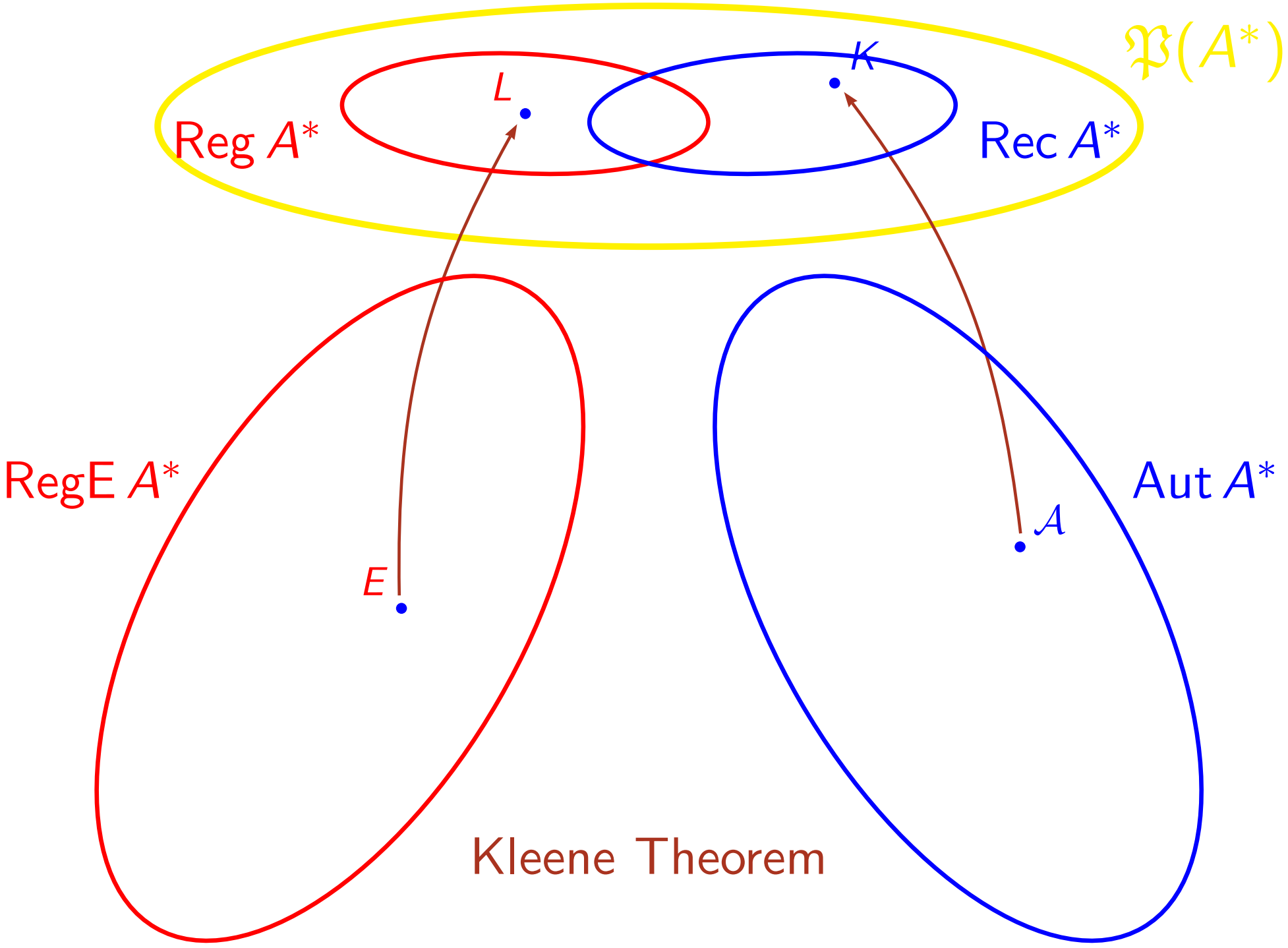
In what follows,

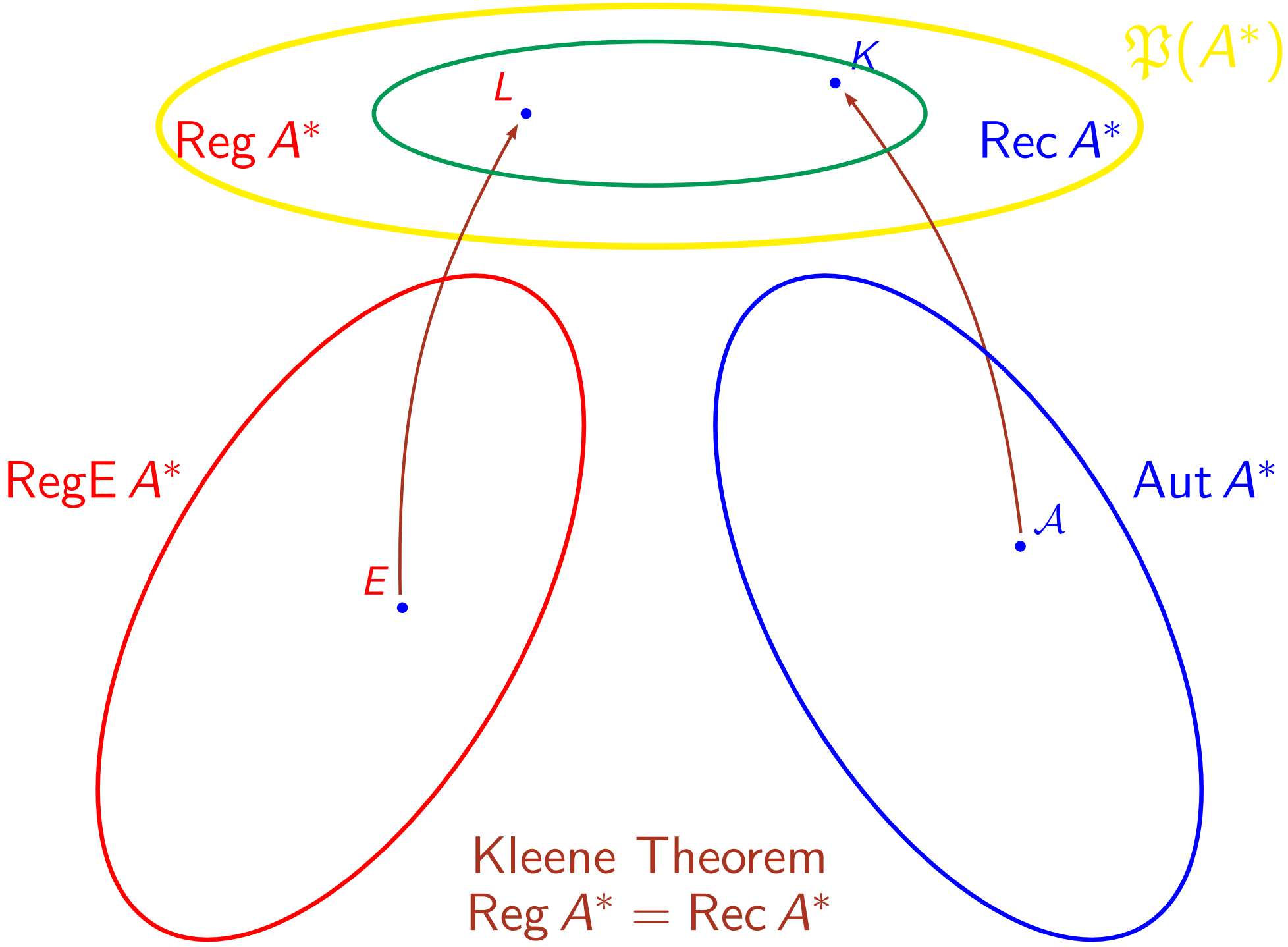
- A is an *alphabet*, *i.e.* a finite set of *letters*.
- A^* is the set of *words*.
- Any subset L of A^* is a *language*.
- The set of *regular expressions* over A^* is denoted $\mathrm{RegE}\, A^*$; the set of *regular languages* over A^* is denoted $\mathrm{Reg}\, A^*$.
- The set of *finite automata* over A^* is denoted $\mathrm{Aut}\, A^*$; the set of *recognizable languages* over A^* is denoted $\mathrm{Rec}\, A^*$.

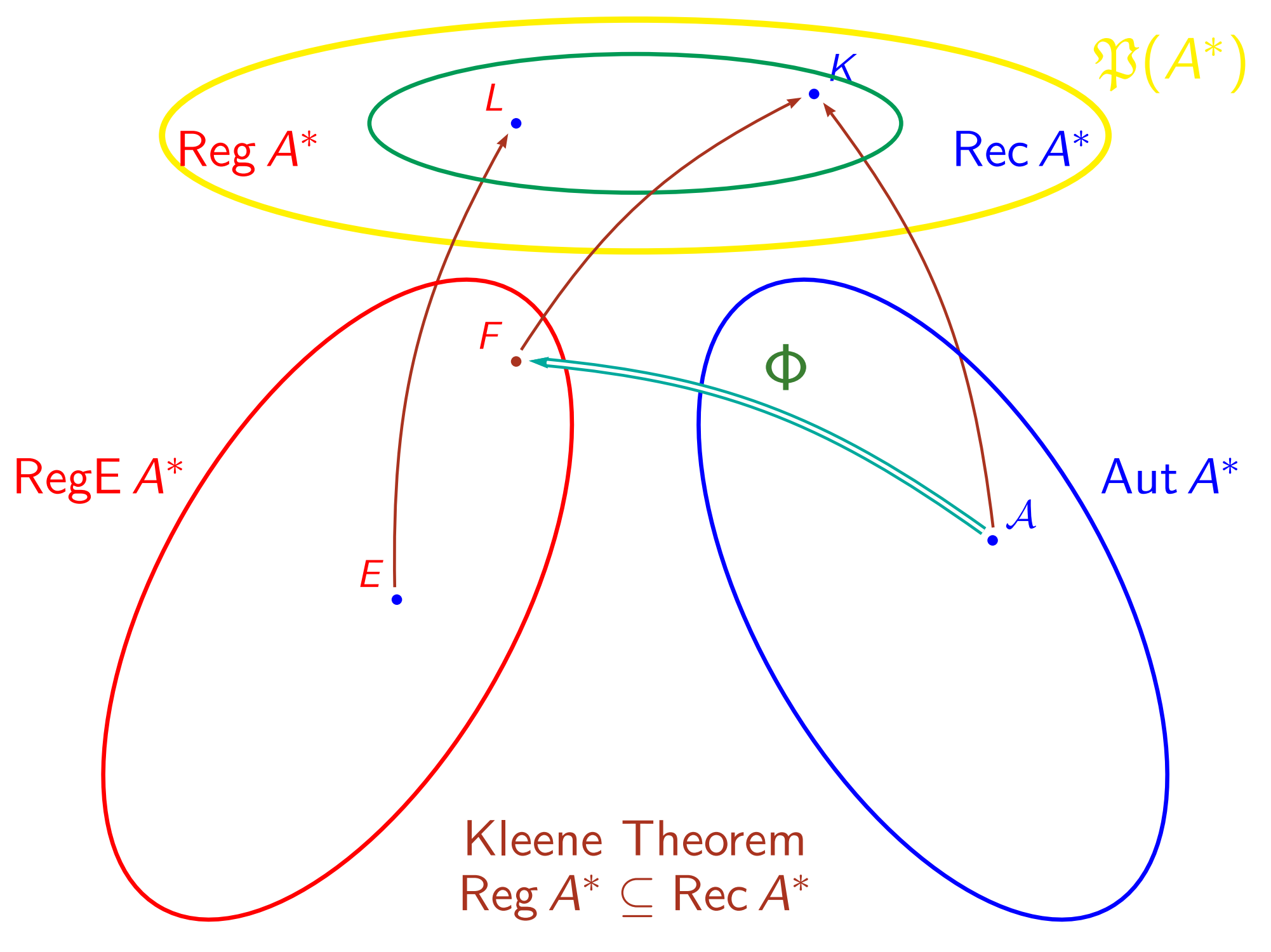
$\mathfrak{P}(A^*)$

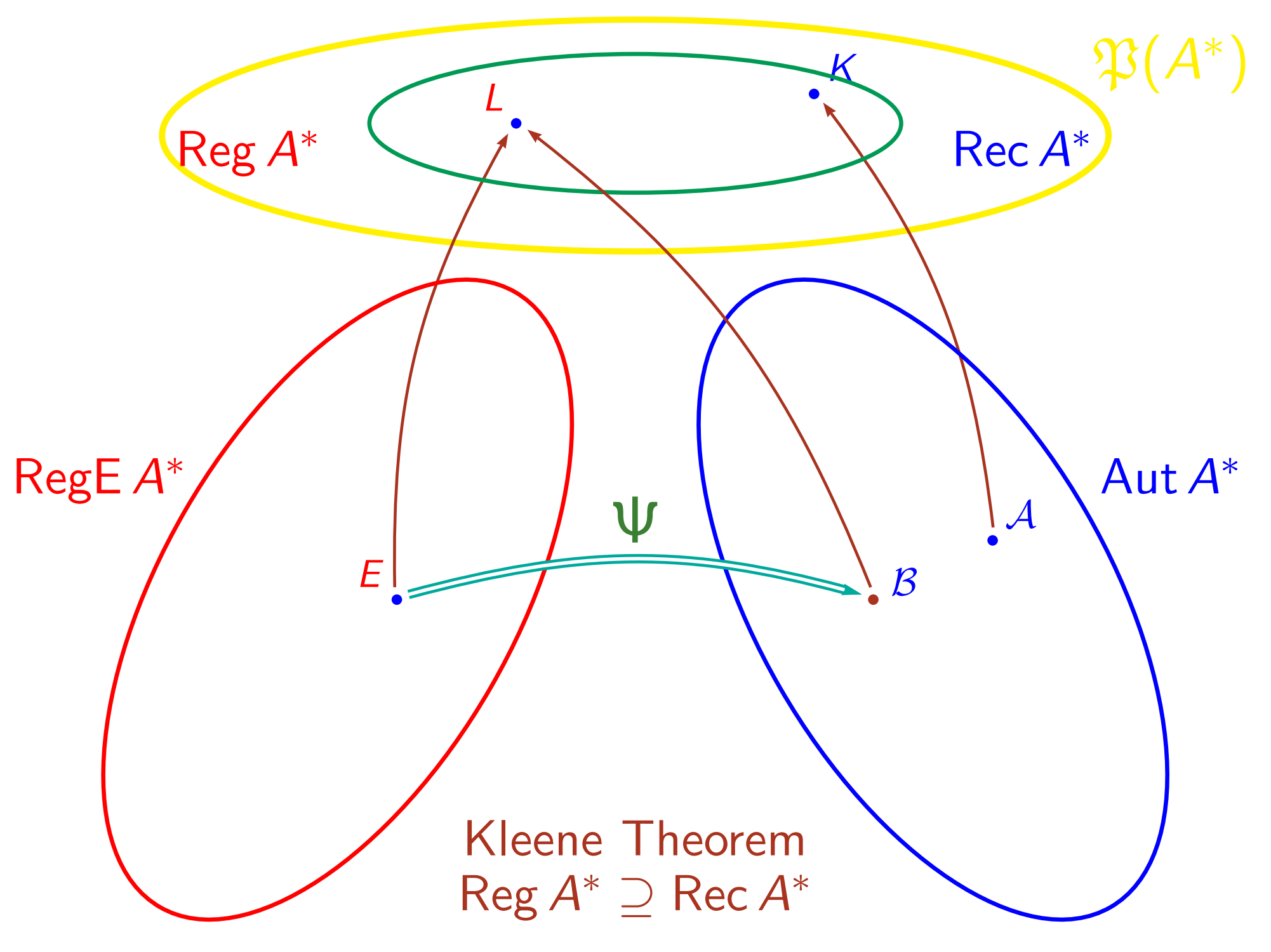

$\mathfrak{P}(A^*)$
Reg A^*
L
RegE A^*
E

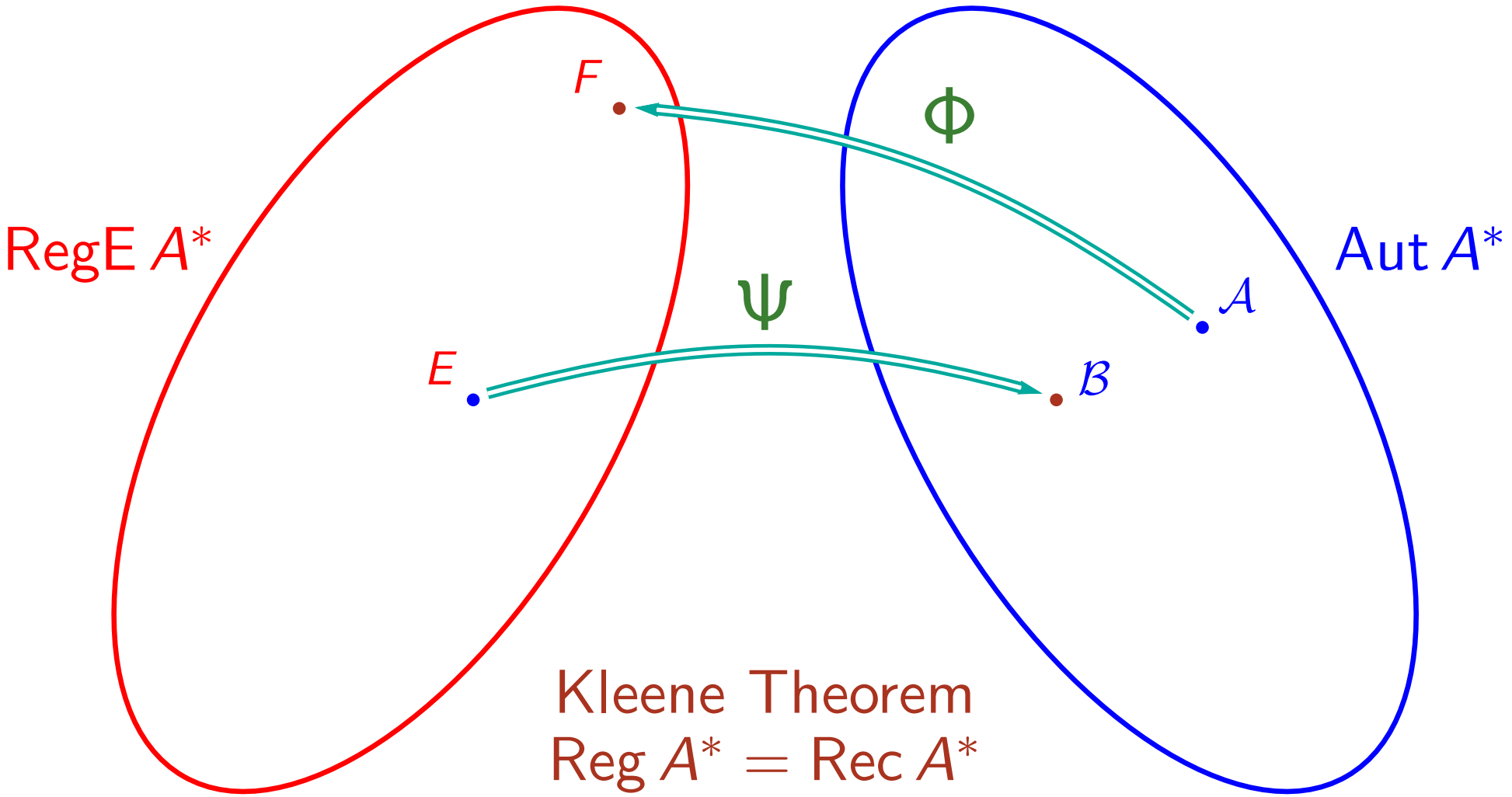

$\mathfrak{P}(A^*)$
Reg A^*
L
K
Rec A^*
RegE A^*
E
$\mathcal{A}$
Aut A^*


$\mathfrak{P}(A^*)$
Reg A^*
L
K
Rec A^*
RegE A^*
E
$\mathcal{A}$
Aut A^*
Kleene Theorem


$\mathfrak{P}(A^*)$
Reg A^*
Rec A^*
L
K
RegE A^*
Aut A^*
E
$\mathcal{A}$
Kleene Theorem
Reg A^* = Rec A^*


$\mathfrak{P}(A^*)$
Reg A^*
Rec A^*
L
K
F
Φ
RegE A^*
Aut A^*
$\mathcal{A}$
E
Kleene Theorem
Reg $A^* \subseteq$ Rec A^*


$\mathfrak{P}(A^*)$
Reg A^*
Rec A^*
L
K
RegE A^*
Aut A^*
Ψ
E
$\mathcal{B}$
$\mathcal{A}$
Kleene Theorem
Reg $A^* \supseteq$ Rec A^*



Kleene Theorem

$\mathrm{Reg}\, A^* = \mathrm{Rec}\, A^*$

Part II

The Phi algorithms

The Φ algorithms

The Φ algorithms

That is, computing an expression from an automaton

The Φ algorithms

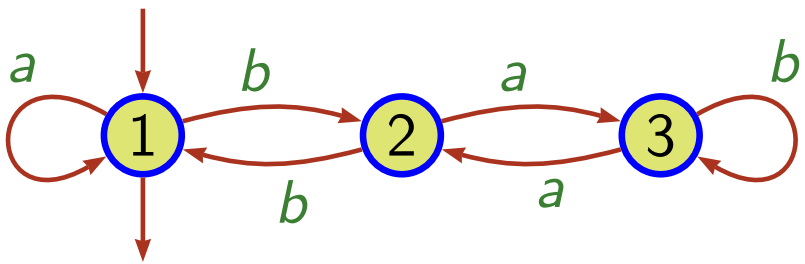
That is, computing an expression from an automaton

- From a theoretical point of view
- From an experimental point of view

The Φ algorithms

That is, computing an expression from an automaton

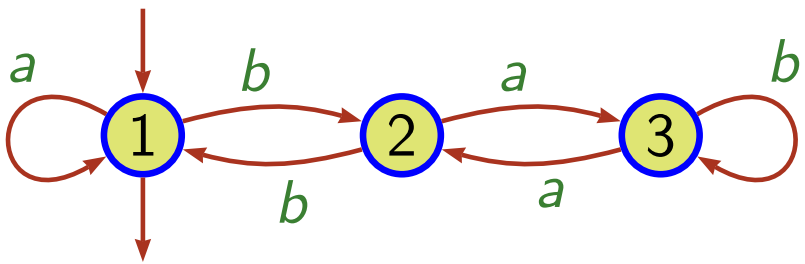
- From a theoretical point of view
- From an experimental point of view



The Φ algorithms

That is, computing an expression from an automaton

- From a theoretical point of view
- From an experimental point of view

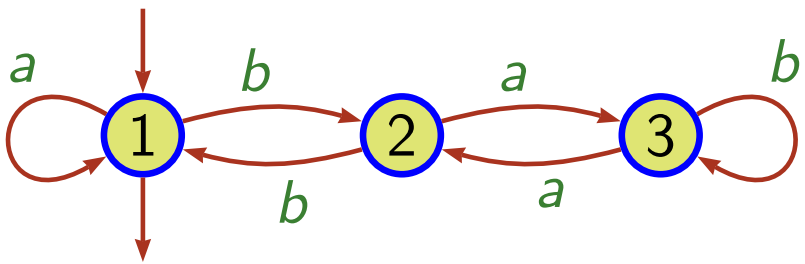


$$\left\langle \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} a & b & 0 \\ b & 0 & a \\ 0 & a & b \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right\rangle$$

The Φ algorithms

That is, computing an expression from an automaton

- From a theoretical point of view
- From an experimental point of view



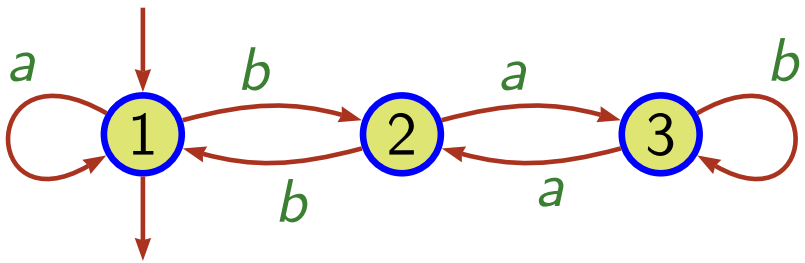
$$\left\langle \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} a & b & 0 \\ b & 0 & a \\ 0 & a & b \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right\rangle$$

$$\mathcal{A} = \langle I, X, T \rangle \qquad L(\mathcal{A}) = I \cdot X^* \cdot T$$

The Φ algorithms

That is, computing an expression from an automaton

- From a theoretical point of view
- From an experimental point of view



$$\left\langle \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} a & b & 0 \\ b & 0 & a \\ 0 & a & b \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right\rangle$$

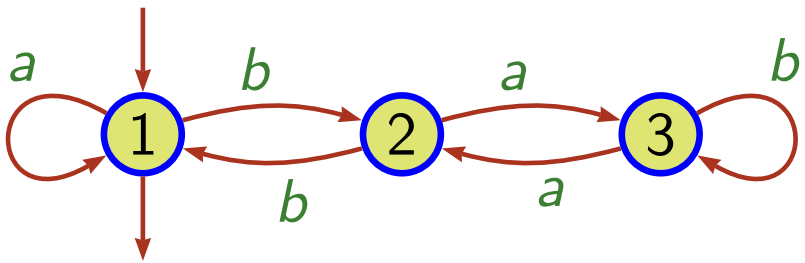
$$\mathcal{A} = \langle I, X, T \rangle \qquad L(\mathcal{A}) = I \cdot X^* \cdot T$$

Computing the star of a matrix with entries in $\mathfrak{P}(A^*)$

The Φ algorithms

That is, computing an expression from an automaton

- From a theoretical point of view
- From an experimental point of view



$$\left\langle \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} a & b & 0 \\ b & 0 & a \\ 0 & a & b \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right\rangle$$

$$\mathcal{A} = \langle I, X, T \rangle \qquad L(\mathcal{A}) = I \cdot X^* \cdot T$$

Computing the star of a matrix with entries in $\mathfrak{P}(A^*)$

Computing the *quasi-inverse* of a matrix with entries in $\mathfrak{P}(A^*)$

The Φ algorithms

- Theoretical point of view : methods of computations

1. Direct computation of the entries of X^* :

2. Iterative computation of X^* :

3. Recursive computation of X^* :

4. Computation of $X^* \cdot T$ as a fixed point:

The Φ algorithms

- Theoretical point of view : methods of computations

1. Direct computation of the entries of X^* :
   state elimination method (Brzozowski–McCluskey)
2. Iterative computation of X^* :
   McNaughton–Yamada algorithm
3. Recursive computation of X^* : Conway(?) algorithm
4. Computation of $X^* \cdot T$ as a fixed point:
   solution of a system of linear equations

The Φ algorithms

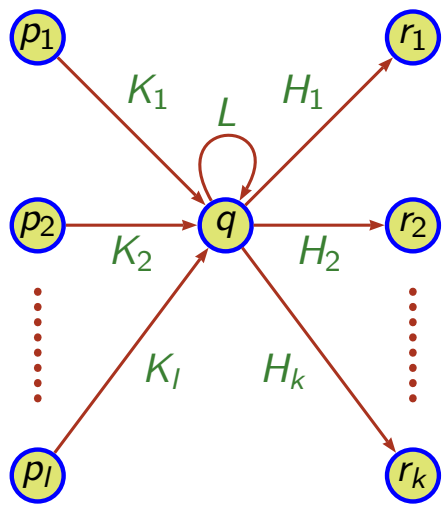
- Theoretical point of view : methods of computations

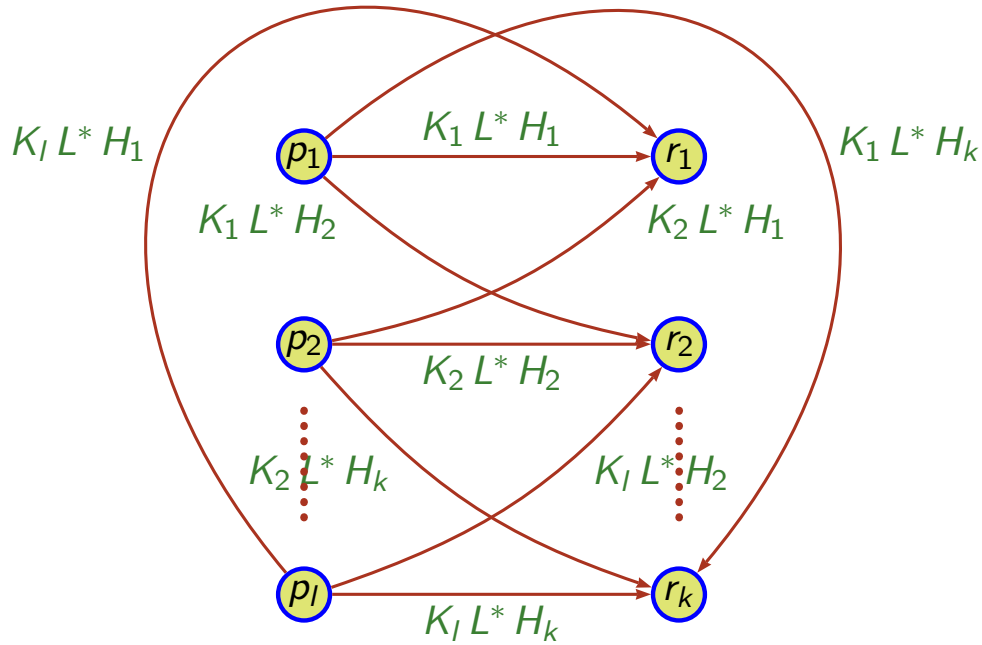
1. Direct computation of the entries of X^* :
   state elimination method (Brzozowski–McCluskey)

The Φ algorithms

- Theoretical point of view : methods of computations

1. Direct computation of the entries of X^* :
   state elimination method (Brzozowski–McCluskey)

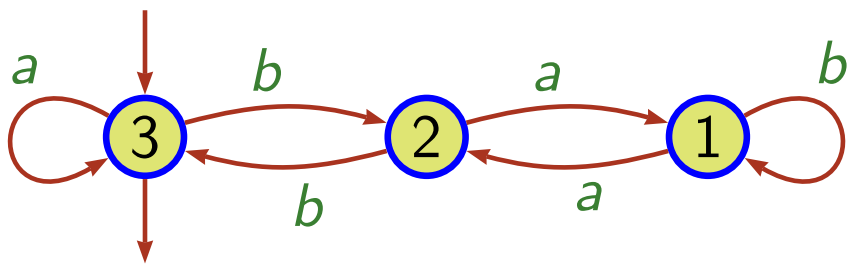




The Φ algorithms

- Theoretical point of view : methods of computations

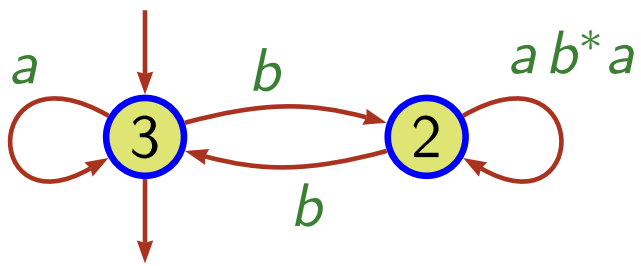
1. Direct computation of the entries of X^* :
   state elimination method (Brzozowski–McCluskey)



The Φ algorithms

- Theoretical point of view : methods of computations

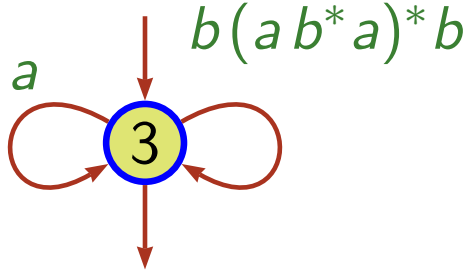
1. Direct computation of the entries of X^* :
   state elimination method (Brzozowski–McCluskey)



The Φ algorithms

- Theoretical point of view : methods of computations

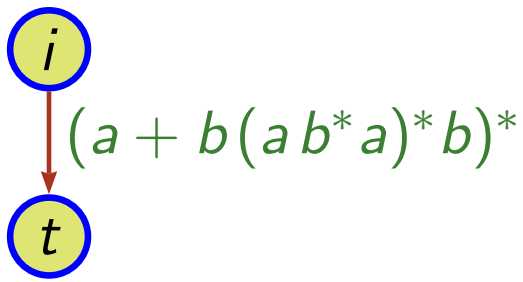
1. Direct computation of the entries of X^* :
   state elimination method (Brzozowski–McCluskey)



The Φ algorithms

- Theoretical point of view : methods of computations

1. Direct computation of the entries of X^* :
   state elimination method (Brzozowski–McCluskey)



The Φ algorithms

- Theoretical point of view : methods of computations

1. Direct computation of the entries of X^*
2. Iterative computation of X^*
3. Recursive computation of X^*
4. Computation of $X^* \cdot T$ as a fixed point

The Φ algorithms

- Theoretical point of view : methods of computations

1. Direct computation of the entries of X^*
2. Iterative computation of X^*
3. Recursive computation of X^*
4. Computation of $X^* \cdot T$ as a fixed point

ONE method, different perspectives

The Φ algorithms

- Theoretical point of view : methods of computations

1. Direct computation of the entries of X^*
2. Iterative computation of X^*
3. Recursive computation of X^*
4. Computation of $X^* \cdot T$ as a fixed point

ONE method, different perspectives

Comparison between the expressions obtained with each method

The Φ algorithms

- Theoretical point of view : methods of computations

1. Direct computation of the entries of X^*
2. Iterative computation of X^*
3. Recursive computation of X^*
4. Computation of $X^* \cdot T$ as a fixed point

ONE method, different perspectives

Comparison between the expressions obtained with each method

For each method, the actual computation
depends on an order on the set of states.

The Φ algorithms

- Theoretical point of view : methods of computations

1. Direct computation of the entries of X^*
2. Iterative computation of X^*
3. Recursive computation of X^*
4. Computation of $X^* \cdot T$ as a fixed point

ONE method, different perspectives

Comparison between the expressions obtained with each method

For each method, the actual computation
depends on an order on the set of states.

Comparison between the expressions obtained
in each method with distinct orders

Axiomatisation of regular expressions

Trivial and natural identities

$$E + 0 \equiv 0 + E \equiv E \ , \quad E \cdot 0 \equiv 0 \cdot E \equiv 0 \ , \quad E \cdot 1 \equiv 1 \cdot E \equiv E \qquad (\mathbf{T})$$

$$(E + F) + G \equiv E + (F + G) \ , \quad (E \cdot F) \cdot G \equiv E \cdot (F \cdot G) \qquad (\mathbf{A})$$

$$E \cdot (F + G) \equiv E \cdot F + E \cdot G \ , \quad (E + F) \cdot G \equiv E \cdot G + F \cdot G \qquad (\mathbf{D})$$

$$E + F \equiv F + E \qquad (\mathbf{C})$$

Axiomatisation of regular expressions

Trivial and natural identities

$$E + 0 \equiv 0 + E \equiv E \ , \quad E \cdot 0 \equiv 0 \cdot E \equiv 0 \ , \quad E \cdot 1 \equiv 1 \cdot E \equiv E \qquad (\mathbf{T})$$

$$(E + F) + G \equiv E + (F + G) \ , \quad (E \cdot F) \cdot G \equiv E \cdot (F \cdot G) \qquad (\mathbf{A})$$

$$E \cdot (F + G) \equiv E \cdot F + E \cdot G \ , \quad (E + F) \cdot G \equiv E \cdot G + F \cdot G \qquad (\mathbf{D})$$

$$E + F \equiv F + E \qquad (\mathbf{C})$$

Aperiodic identities.

$$E^* \equiv 1 + E \cdot E^* \ , \quad E^* \equiv 1 + E^* \cdot E \qquad (\mathbf{U})$$

$$(E + F)^* \equiv E^* \cdot (F \cdot E^*)^* \ , \quad (E + F)^* \equiv (E^* \cdot F)^* \cdot E^* \qquad (\mathbf{S})$$

$$(E \cdot F)^* \equiv 1 + E \cdot (F \cdot E)^* \cdot F \qquad (\mathbf{P})$$

Axiomatisation of regular expressions

Trivial and natural identities

$$E + 0 \equiv 0 + E \equiv E \ , \quad E \cdot 0 \equiv 0 \cdot E \equiv 0 \ , \quad E \cdot 1 \equiv 1 \cdot E \equiv E \qquad (\mathbf{T})$$

$$(E + F) + G \equiv E + (F + G) \ , \quad (E \cdot F) \cdot G \equiv E \cdot (F \cdot G) \qquad (\mathbf{A})$$

$$E \cdot (F + G) \equiv E \cdot F + E \cdot G \ , \quad (E + F) \cdot G \equiv E \cdot G + F \cdot G \qquad (\mathbf{D})$$

$$E + F \equiv F + E \qquad (\mathbf{C})$$

Aperiodic identities.

$$E^* \equiv 1 + E \cdot E^* \ , \quad E^* \equiv 1 + E^* \cdot E \qquad (\mathbf{U})$$

$$(E + F)^* \equiv E^* \cdot (F \cdot E^*)^* \ , \quad (E + F)^* \equiv (E^* \cdot F)^* \cdot E^* \qquad (\mathbf{S})$$

$$(E \cdot F)^* \equiv 1 + E \cdot (F \cdot E)^* \cdot F \qquad (\mathbf{P})$$

Cyclic identities.

$$E^* \equiv E^{<n} \cdot (E^n)^* \qquad (\mathbf{Z}_n)$$

Axiomatisation of regular expressions

Natural identities

$$(E + F) + G \equiv E + (F + G) \ , \quad (E \cdot F) \cdot G \equiv E \cdot (F \cdot G) \qquad (\mathbf{A})$$

$$E \cdot (F + G) \equiv E \cdot F + E \cdot G \ , \quad (E + F) \cdot G \equiv E \cdot G + F \cdot G \qquad (\mathbf{D})$$

$$E + F \equiv F + E \qquad (\mathbf{C})$$

Aperiodic identities.

$$E^* \equiv 1 + E \cdot E^* \ , \quad E^* \equiv 1 + E^* \cdot E \qquad (\mathbf{U})$$

$$(E + F)^* \equiv E^* \cdot (F \cdot E^*)^* \ , \quad (E + F)^* \equiv (E^* \cdot F)^* \cdot E^* \qquad (\mathbf{S})$$

$$(E \cdot F)^* \equiv 1 + E \cdot (F \cdot E)^* \cdot F \qquad (\mathbf{P})$$

Cyclic identities.

$$E^* \equiv E^{<n} \cdot (E^n)^* \qquad (\mathbf{Z}_n)$$

Idempotency identities.

$$E + E \equiv E \qquad (\mathbf{I})$$

$$(E^*)^* \equiv E^* \qquad (\mathbf{J})$$

The Φ algorithms

$$\mathcal{A} = \langle\, Q, A, X, I, T \,\rangle$$

1. McNaughton–Yamada algorithm $\quad \mathrm{M}_{\mathcal{A}}$
2. Recursive computation of X^* $\quad \mathrm{C}_{\mathcal{A}}$
3. State elimination method $\quad \mathrm{E}_{\mathcal{A}}$
4. Solution of a system of linear equations $\quad \mathrm{S}_{\mathcal{A}}$

The Φ algorithms

$\mathcal{A} = \langle\, Q, A, X, I, T \,\rangle$ ω ordering on Q

1. McNaughton–Yamada algorithm $\mathrm{M}(\omega)$
2. Recursive computation of X^* $\mathrm{C}(\omega)$
3. State elimination method $\mathrm{E}(\omega)$
4. Solution of a system of linear equations $\mathrm{S}(\omega)$

The Φ algorithms

$\mathcal{A} = \langle\, Q, A, X, \{p\}, \{q\} \,\rangle$ ω ordering on Q

1. McNaughton–Yamada algorithm $[\mathrm{M}(\omega)]_{p,q}$
2. Recursive computation of X^* $[\mathrm{C}(\omega)]_{p,q}$
3. State elimination method $\mathrm{E}(\omega, p, q)$
4. Solution of a system of linear equations $[\mathrm{S}(\omega, q)]_p$

The Φ algorithms

$\mathcal{A} = \langle\, Q, A, X, \{p\}, \{q\} \,\rangle$ ω ordering on Q

1. McNaughton–Yamada algorithm $[\mathrm{M}(\omega)]_{p,q}$
2. Recursive computation of X^* $[\mathrm{C}(\omega)]_{p,q}$
3. State elimination method $\mathrm{E}(\omega, p, q)$
4. Solution of a system of linear equations $[\mathrm{S}(\omega, q)]_p$

Proposition

$$[\mathrm{S}(\omega, q)]_p = \mathrm{E}(\omega, p, q)$$

The Φ algorithms

$\mathcal{A} = \langle\, Q, A, X, \{p\}, \{q\} \,\rangle$ ω ordering on Q

1. McNaughton–Yamada algorithm $[\mathrm{M}(\omega)]_{p,q}$
2. Recursive computation of X^* $[\mathrm{C}(\omega)]_{p,q}$
3. State elimination method $\mathrm{E}(\omega, p, q)$
4. Solution of a system of linear equations $[\mathrm{S}(\omega, q)]_p$

Proposition (S. 03)

$$\mathbf{U} \vdash\!\!\!\!- \quad [\mathrm{M}(\omega)]_{p,q} \equiv \mathrm{E}(\omega, p, q)$$

The Φ algorithms

$\mathcal{A} = \langle\, Q, A, X, \{p\}, \{q\} \,\rangle$ ω ordering on Q

1. McNaughton–Yamada algorithm $[\mathrm{M}(\omega)]_{p,q}$
2. Recursive computation of X^* $[\mathrm{C}(\omega)]_{p,q}$
3. State elimination method $\mathrm{E}(\omega, p, q)$
4. Solution of a system of linear equations $[\mathrm{S}(\omega, q)]_p$

Theorem (Krob 9?)

For any two orderings ω and ω' on Q

$$\mathbf{S} \wedge \mathbf{P} \quad \vdash \quad \mathrm{E}(\omega, p, q) \quad \equiv \quad \mathrm{E}(\omega', p, q)$$

The Φ algorithms

$\mathcal{A} = \langle\, Q, A, X, \{p\}, \{q\} \,\rangle$ $\qquad$ ω ordering on Q

1. McNaughton–Yamada algorithm $\qquad [\mathrm{M}(\omega)]_{p,q}$
2. Recursive computation of X^* $\qquad [\mathrm{C}(\omega)]_{p,q}$
3. State elimination method $\qquad \mathrm{E}(\omega, p, q)$
4. Solution of a system of linear equations $\qquad [\mathrm{S}(\omega, q)]_p$

Conjecture

For any ordering ω on Q there exists an ordering ω' such that

$$\mathbf{U} \vdash \qquad [\mathrm{C}(\omega)]_{p,q} \quad \equiv \quad \mathrm{E}(\omega', p, q)$$

The Φ algorithms

$\mathcal{A} = \langle\, Q, A, X, \{p\}, \{q\} \,\rangle$ $\qquad$ ω ordering on Q

1. McNaughton–Yamada algorithm $\qquad [\mathrm{M}(\omega)]_{p,q}$
2. Recursive computation of X^* $\qquad [\mathrm{C}(\omega)]_{p,q}$
3. State elimination method $\qquad \mathrm{E}(\omega, p, q)$
4. Solution of a system of linear equations $\qquad [\mathrm{S}(\omega, q)]_p$

Conjecture

For any recursive division τ of Q

there exists an ordering ω' such that

$$\mathbf{U} \vdash \qquad [\mathrm{C}(\tau)]_{p,q} \quad \equiv \quad \mathrm{E}(\omega', p, q)$$

The Φ algorithms

$\mathcal{A} = \langle\, Q, A, X, \{p\}, \{q\} \,\rangle$ ω ordering on Q

1. McNaughton–Yamada algorithm $[\mathrm{M}(\omega)]_{p,q}$
2. Recursive computation of X^* $[\mathrm{C}(\omega)]_{p,q}$
3. State elimination method $\mathrm{E}(\omega, p, q)$
4. Solution of a system of linear equations $[\mathrm{S}(\omega, q)]_p$

Conjecture

For any recursive division τ *of* Q
there exists an ordering ω' *such that*

$$\mathbf{S} \wedge \mathbf{P} \quad \vdash \quad [\mathrm{C}(\tau)]_{p,q} \quad \equiv \quad \mathrm{E}(\omega', p, q)$$

The Φ algorithms : from an experimental point of view

The Φ algorithms : from an experimental point of view

The size of E computed from $\mathcal{A}$
may be exponential in the number of states of $\mathcal{A}$.

The Φ algorithms : from an experimental point of view

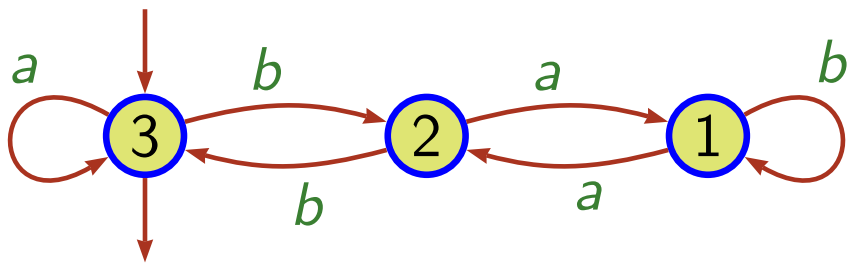
The size of E computed from $\mathcal{A}$
may be exponential in the number of states of $\mathcal{A}$.

The size of E computed from $\mathcal{A}$
may vary dramatically with the *order* put on the states of $\mathcal{A}$.

The Φ algorithms : from an experimental point of view

The size of E computed from $\mathcal{A}$
may be exponential in the number of states of $\mathcal{A}$.

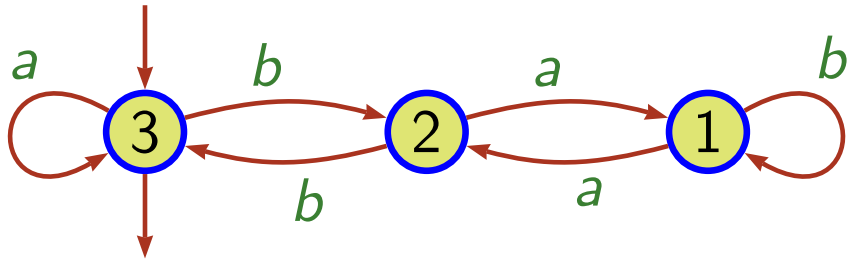
The size of E computed from $\mathcal{A}$
may vary dramatically with the *order* put on the states of $\mathcal{A}$.



The Φ algorithms : from an experimental point of view

The size of E computed from $\mathcal{A}$
may be exponential in the number of states of $\mathcal{A}$.

The size of E computed from $\mathcal{A}$
may vary dramatically with the *order* put on the states of $\mathcal{A}$.

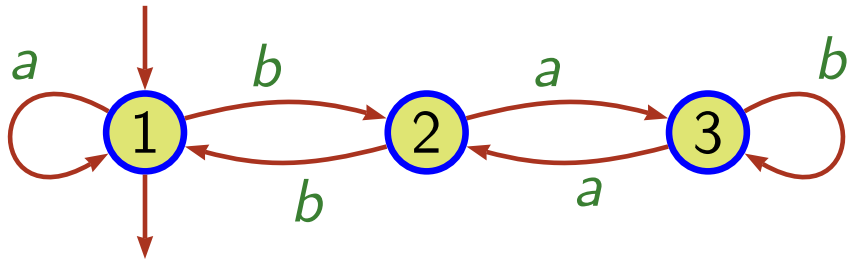


$$\mathrm{E}_2 = (a + b(ab^*a)^*b)^*$$

The Φ algorithms : from an experimental point of view

The size of E computed from $\mathcal{A}$
may be exponential in the number of states of $\mathcal{A}$.

The size of E computed from $\mathcal{A}$
may vary dramatically with the *order* put on the states of $\mathcal{A}$.



$$\mathsf{E}_2 = (a + b(ab^*a)^*b)^*$$

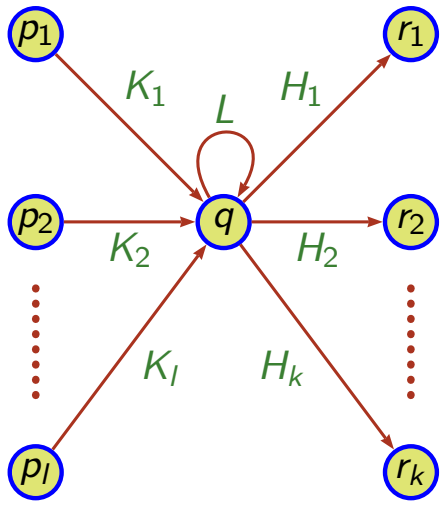
$$\mathsf{E}_1 = a^* + a^*b(ba^*b)^*ba^* + a^*b(ba^*b)^*a(b + a(ba^*b)^*a)^*a(ba^*b)^*ba^*$$

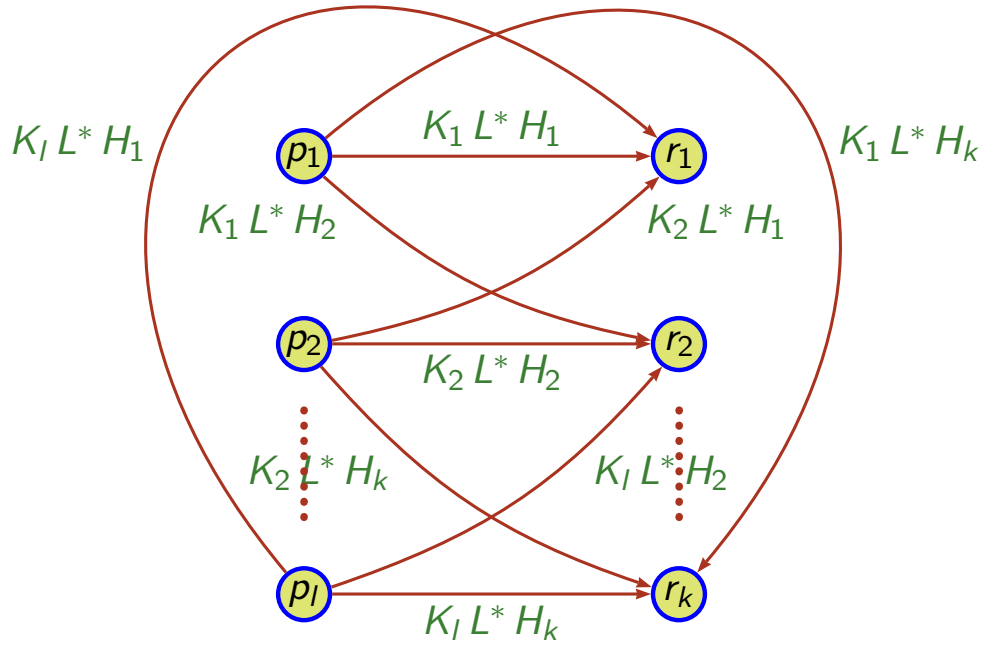
The Φ algorithms : from an experimental point of view

The use of heuristics for the determination of the ordering of states proves to be (very) useful.

The Φ algorithms : from an experimental point of view

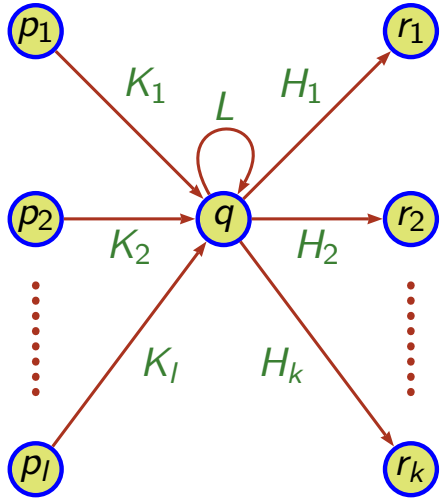
The use of heuristics for the determination of the ordering of states proves to be (very) useful.

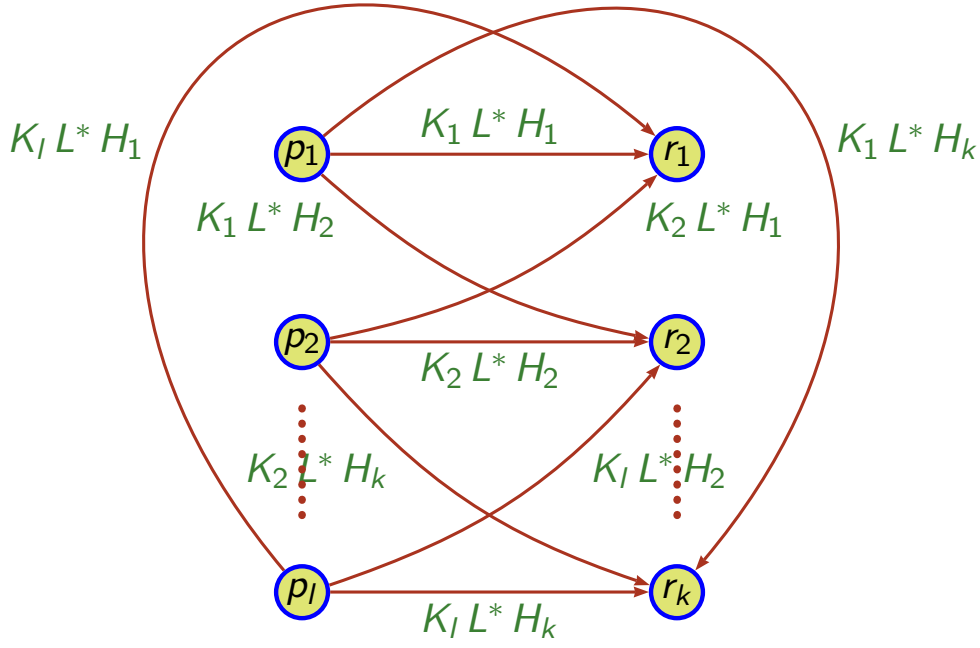




The Φ algorithms : from an experimental point of view

The use of heuristics for the determination of the ordering of states proves to be (very) useful.

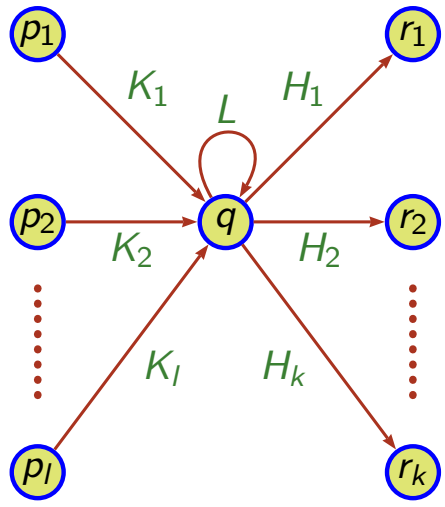


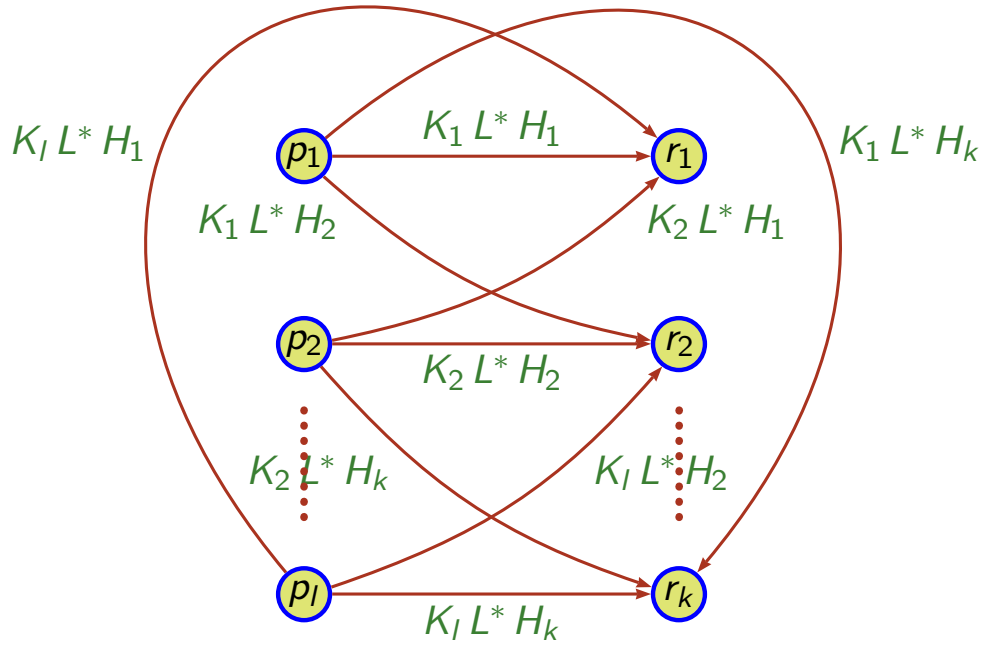


- The naive heuristic

The Φ algorithms : from an experimental point of view

The use of heuristics for the determination of the ordering of states proves to be (very) useful.

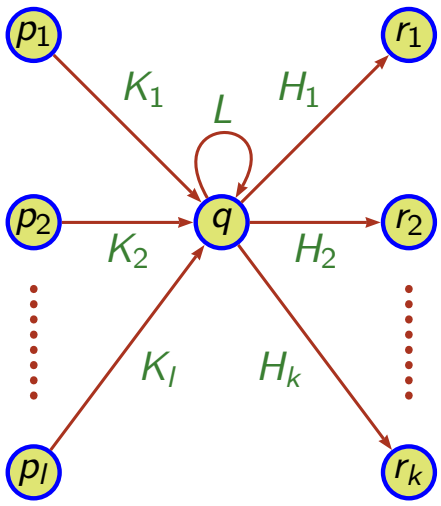


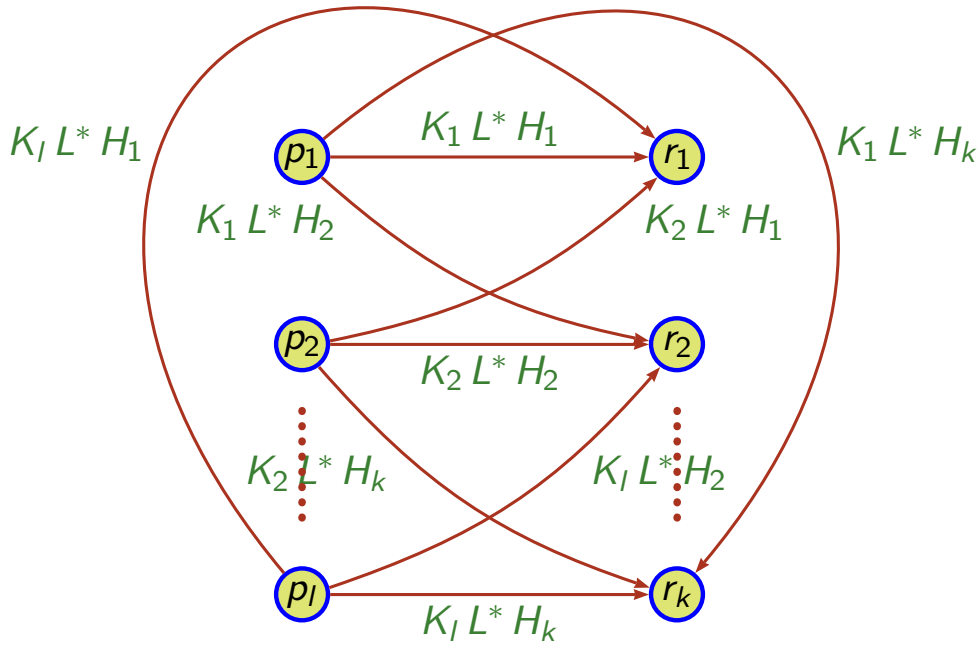


- The naive heuristic
- The Delgado–Morais heuristic (CIAA 04)

The Φ algorithms : from an experimental point of view

The use of heuristics for the determination of the ordering of states proves to be (very) useful.





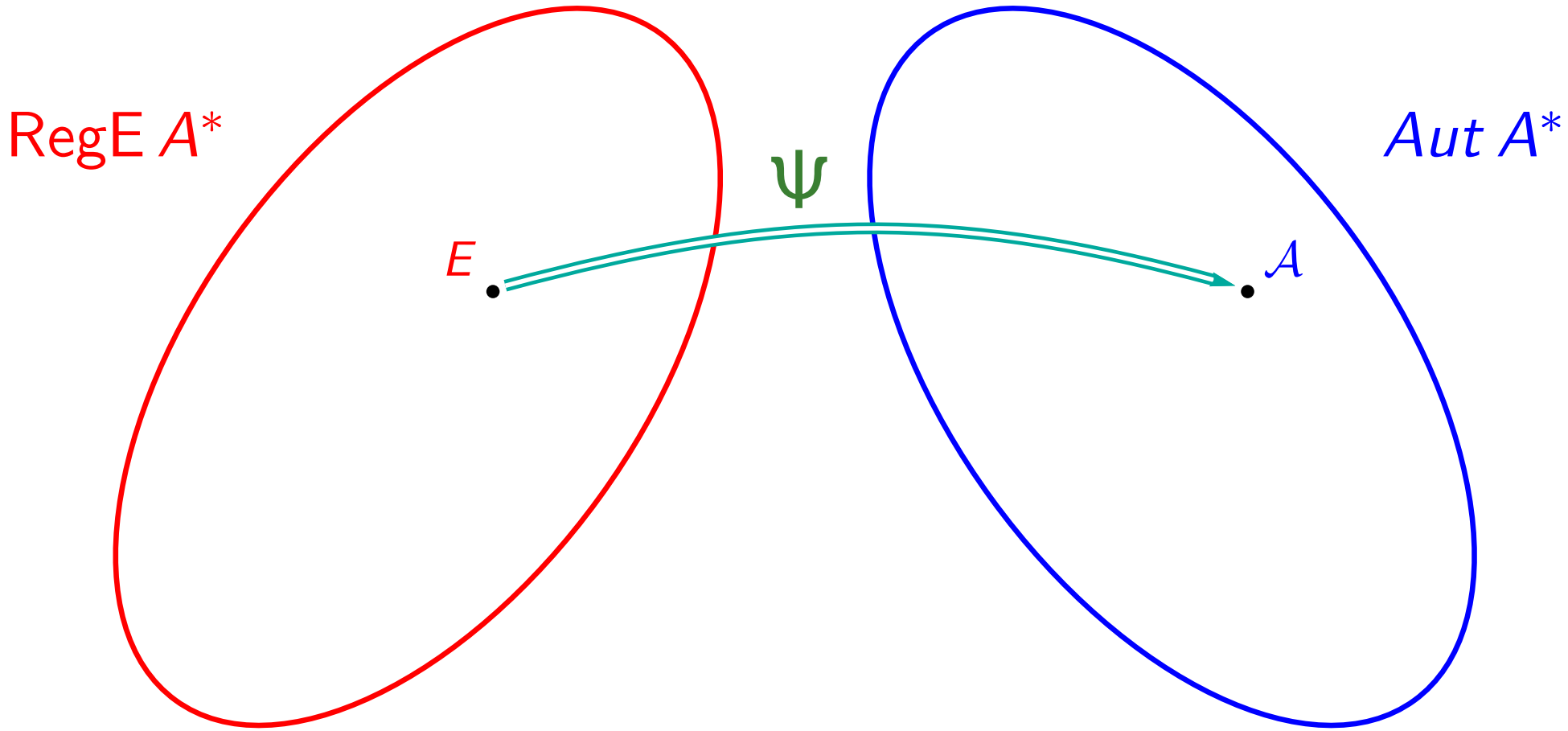
The proof of the heuristic is in the computing.

Part III

The Psi algorithms

The Ψ algorithms

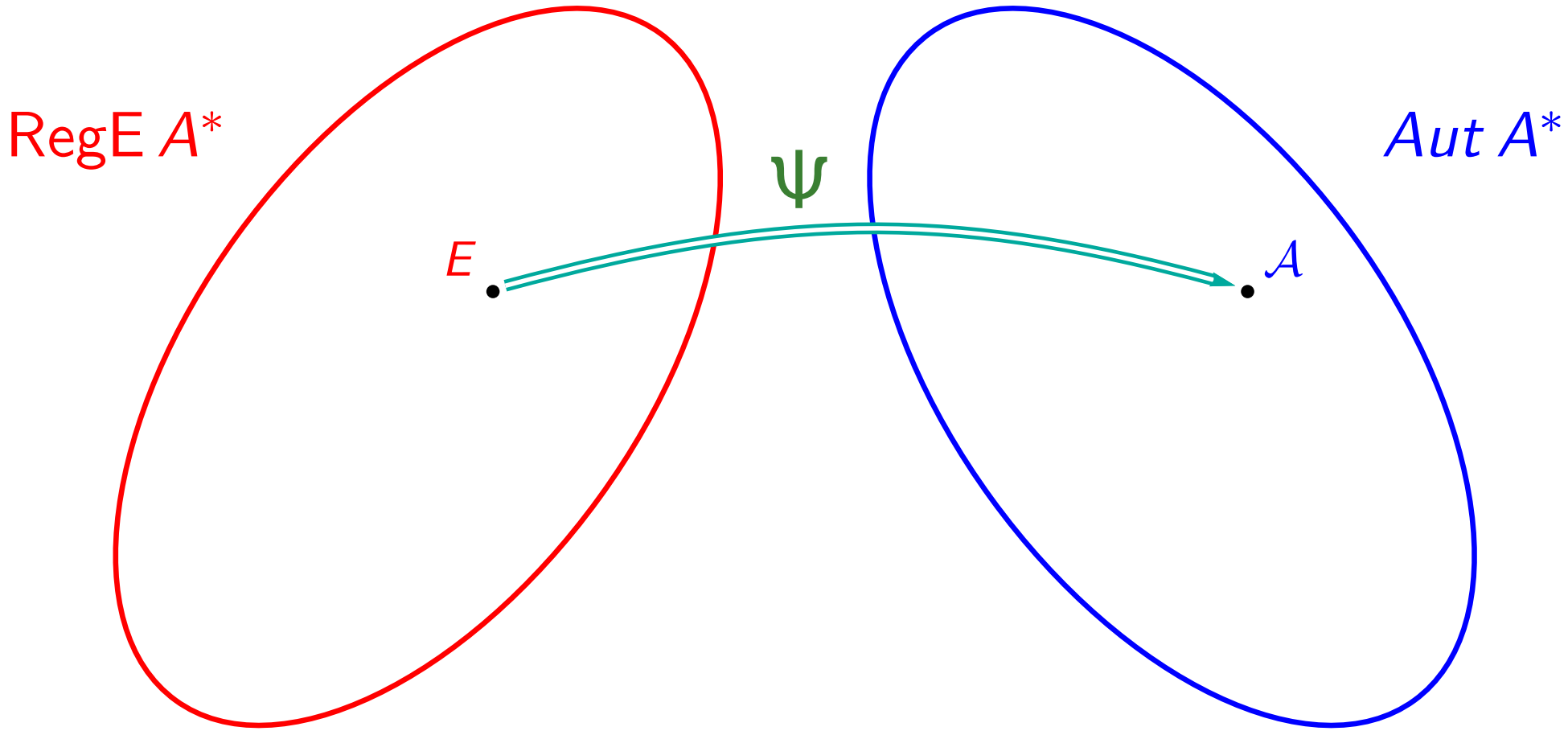
$\mathsf{E} \in \mathsf{RegE}\ A^*$



The Ψ algorithms

$\mathrm{E} \in \mathrm{RegE}\, A^*$

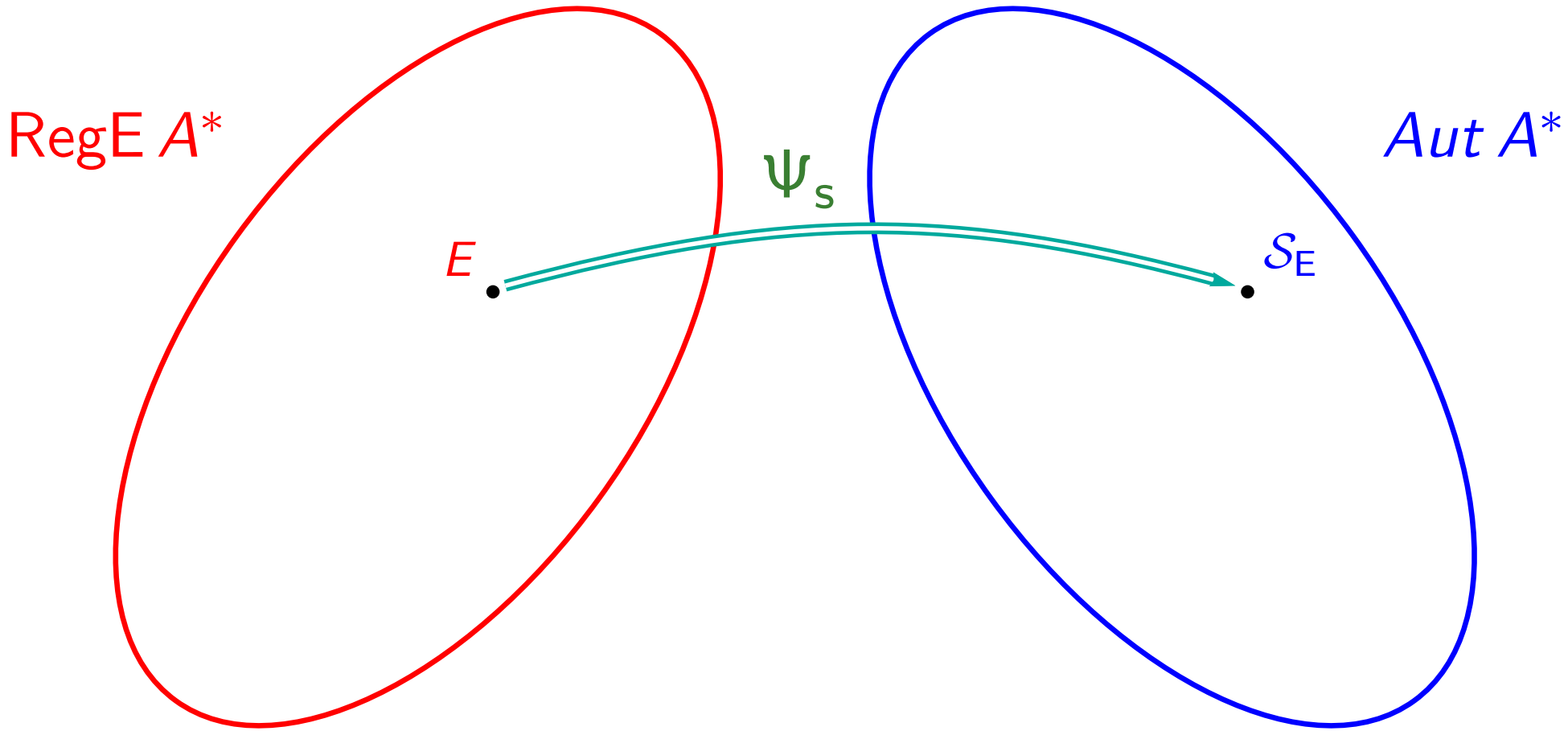
- From a theoretical point of view
- From an experimental point of view



The Ψ algorithms

$\mathrm{E} \in \mathrm{RegE}\, A^*$

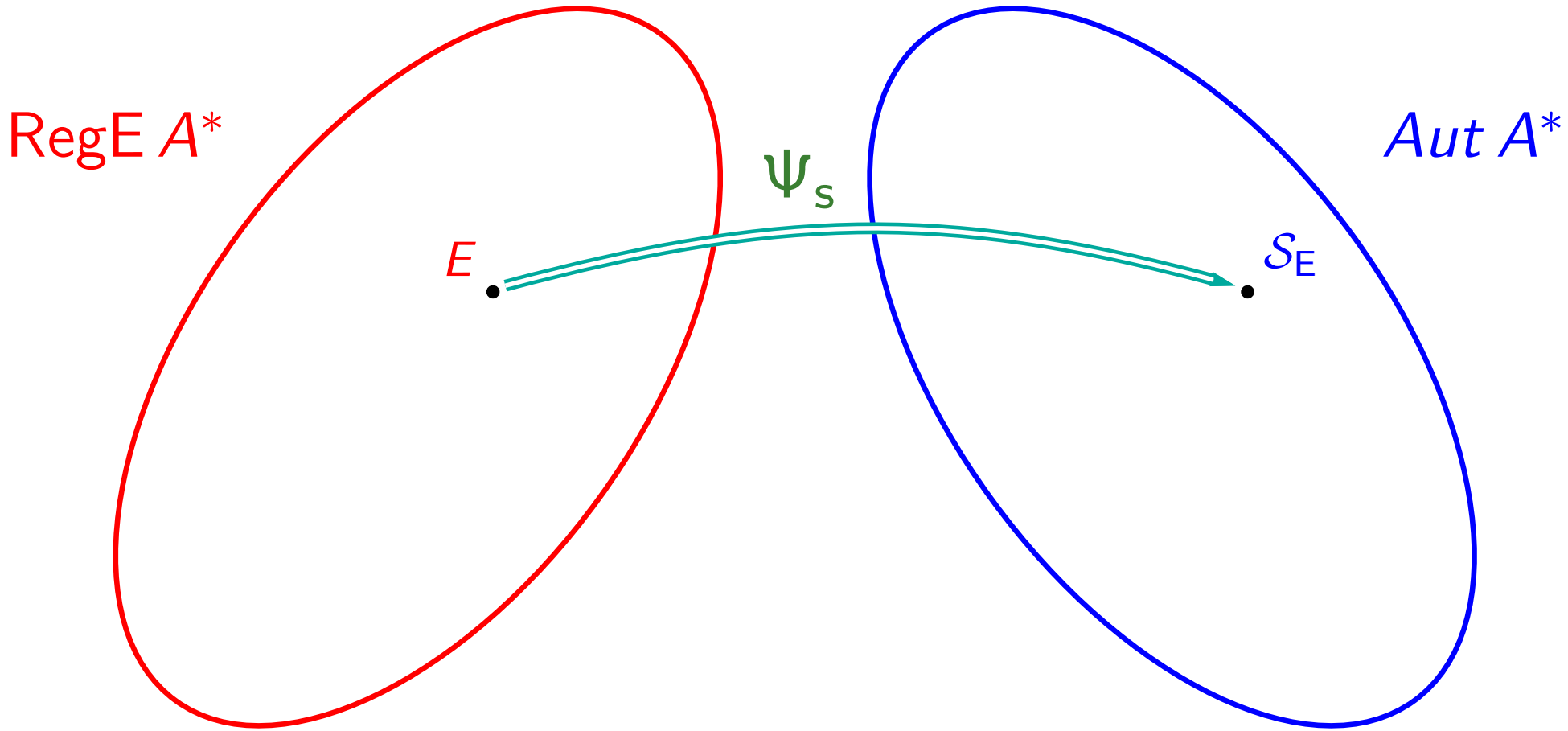
- The standard automaton of E



The Ψ algorithms

$\mathrm{E} \in \mathrm{RegE}\, A^*$

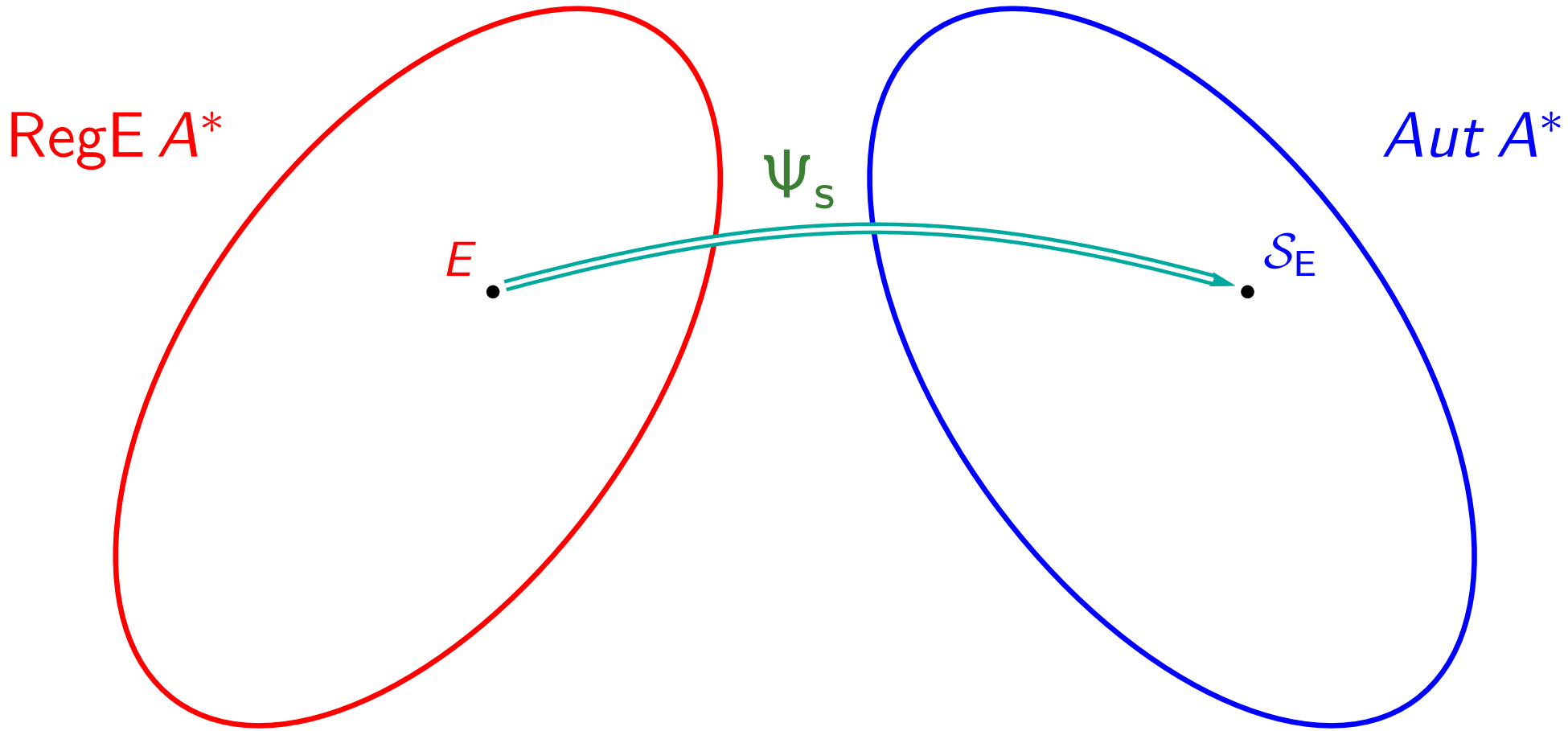
- The standard automaton of E position, Glushkov



The Ψ algorithms

$\mathrm{E} \in \mathrm{RegE}\, A^*$

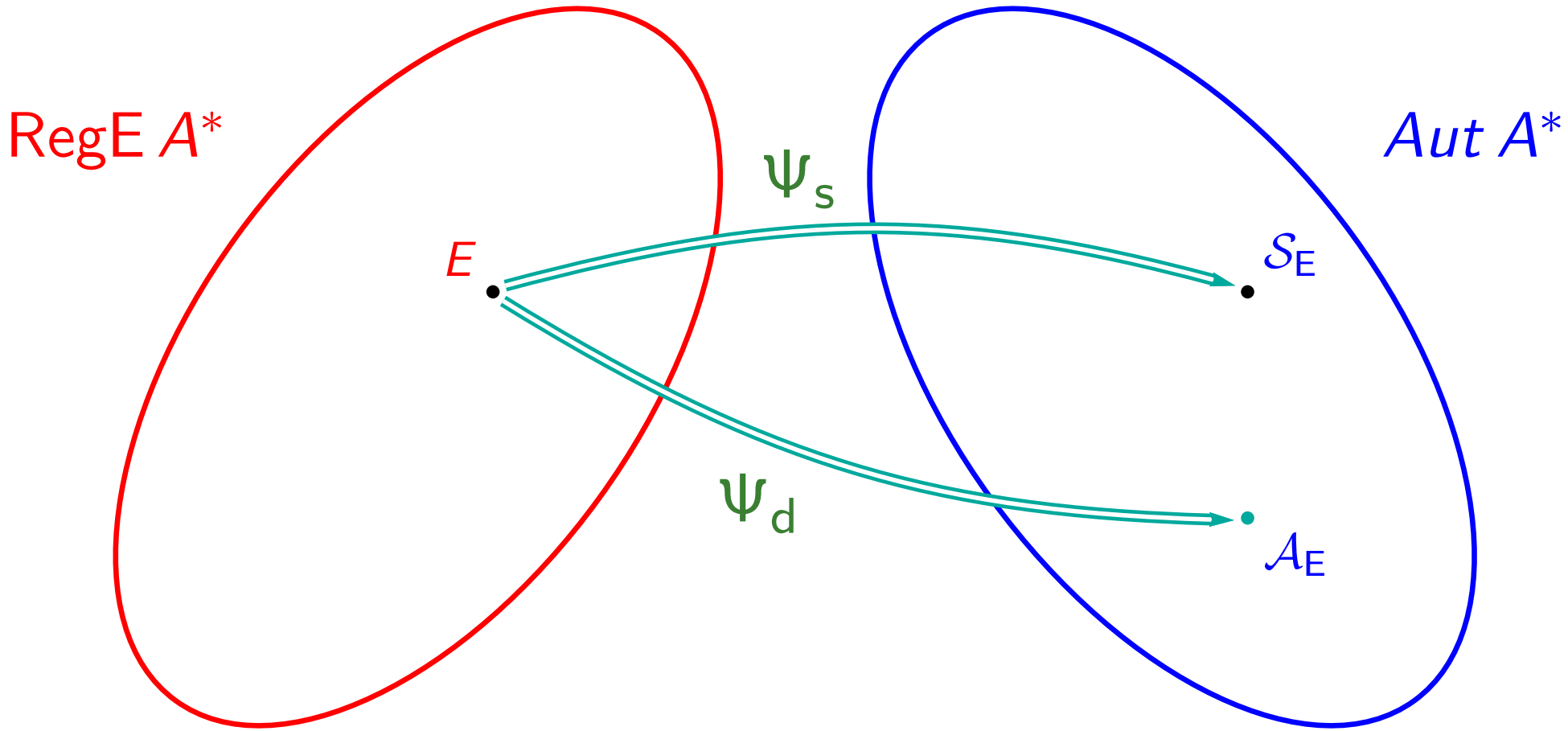
- The standard automaton of E Thompson + closure



The Ψ algorithms

$\mathrm{E} \in \mathrm{RegE}\, A^*$

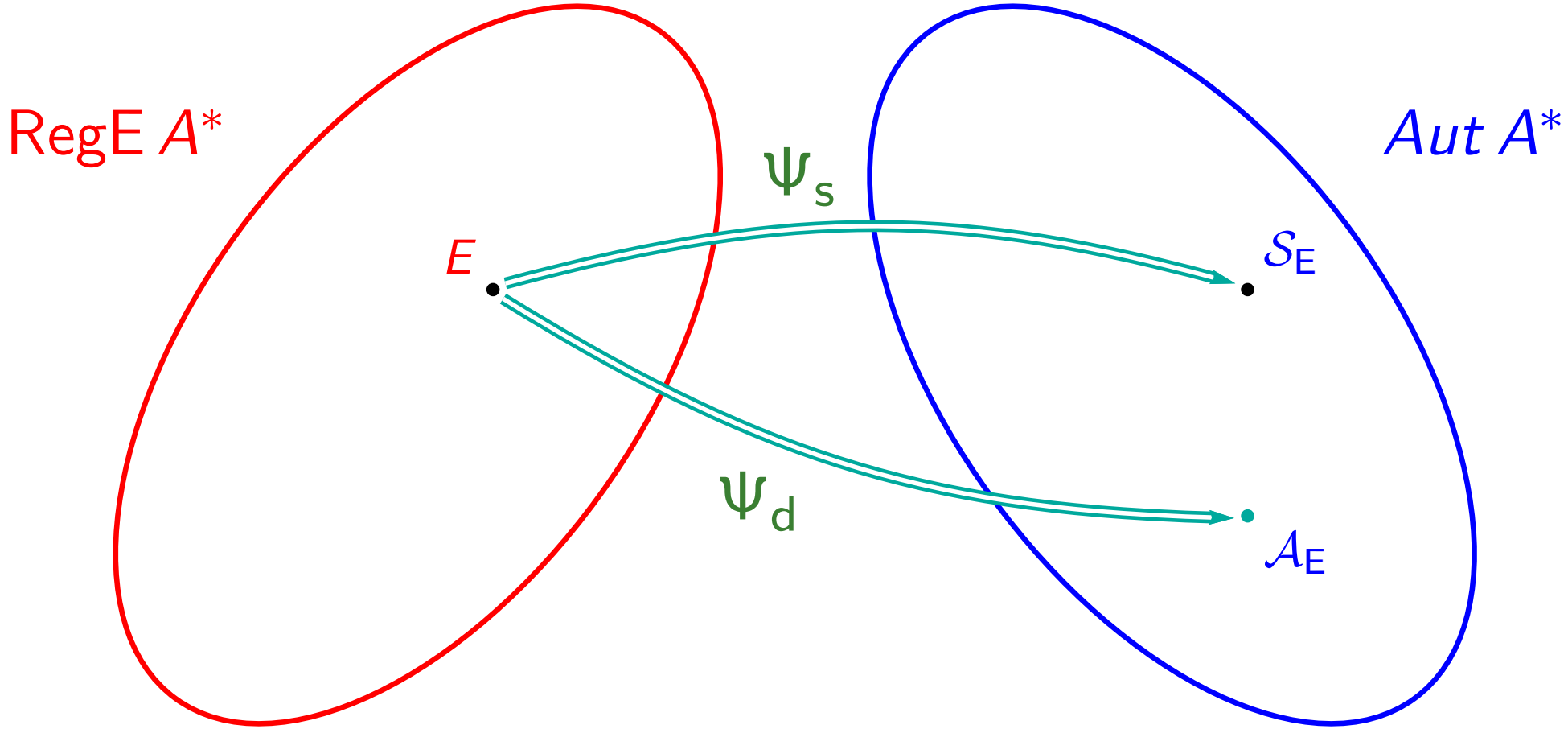
- The standard automaton of E
- The derived term automaton of E



The Ψ algorithms

$\mathrm{E} \in \mathrm{RegE}\, A^*$

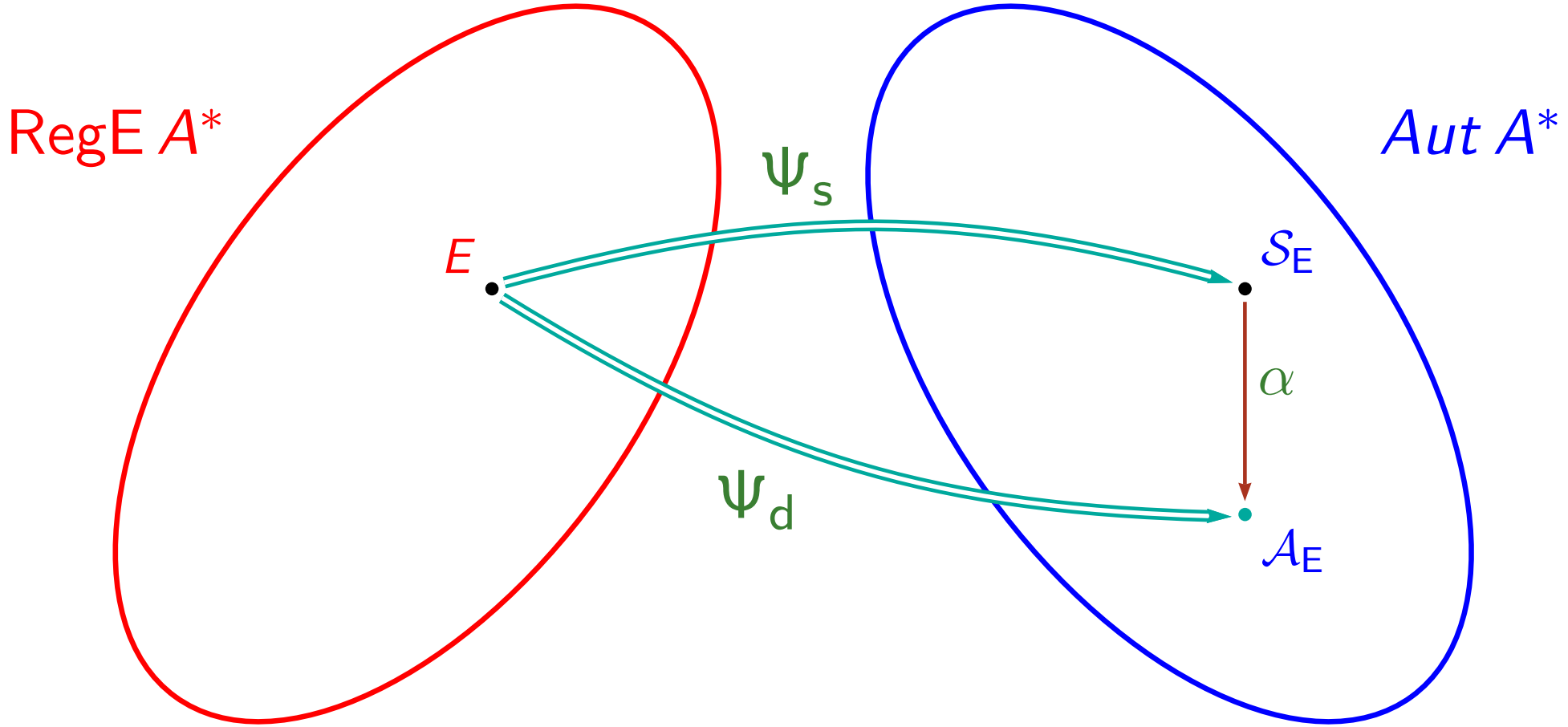
- The standard automaton of E
- The derived term automaton of E Brzozowski–Antimirov



The Ψ algorithms

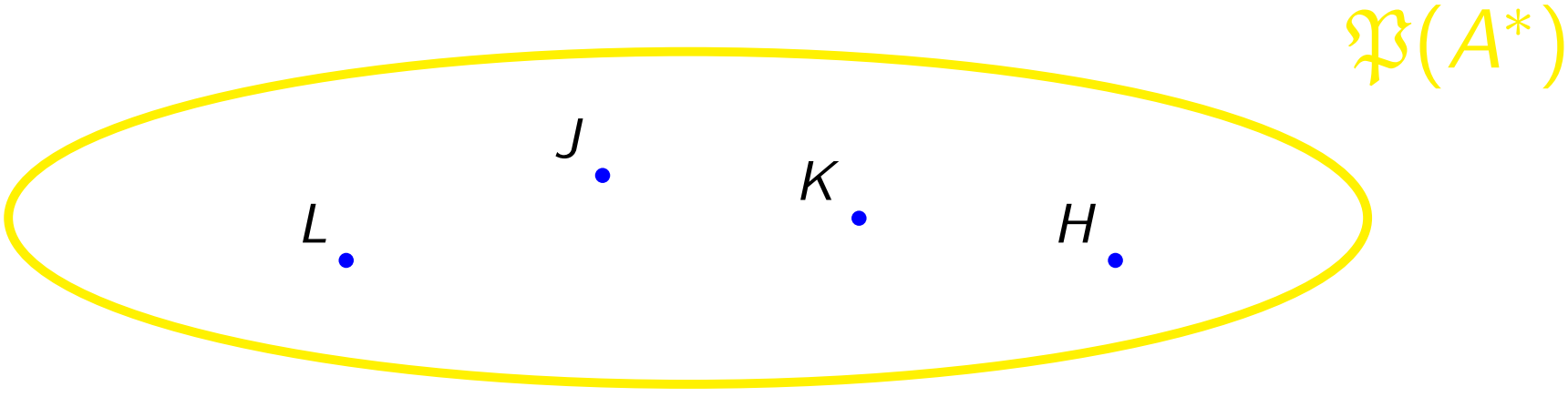
$\mathrm{E} \in \mathrm{RegE}\, A^*$

- The standard automaton of E
- The derived term automaton of E Brzozowski–Antimirov

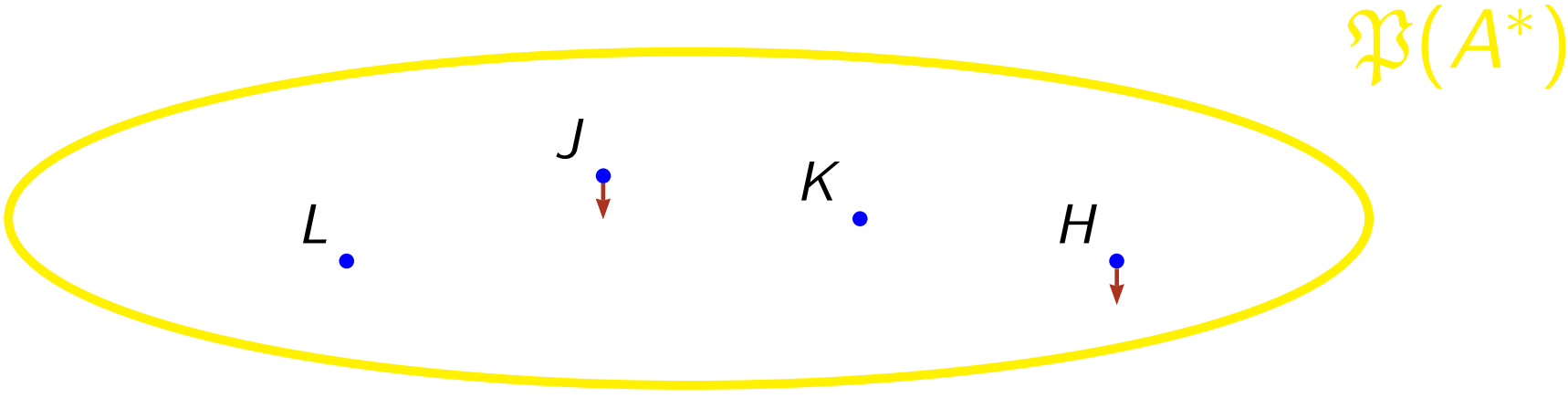


The Brzozowski derivatives

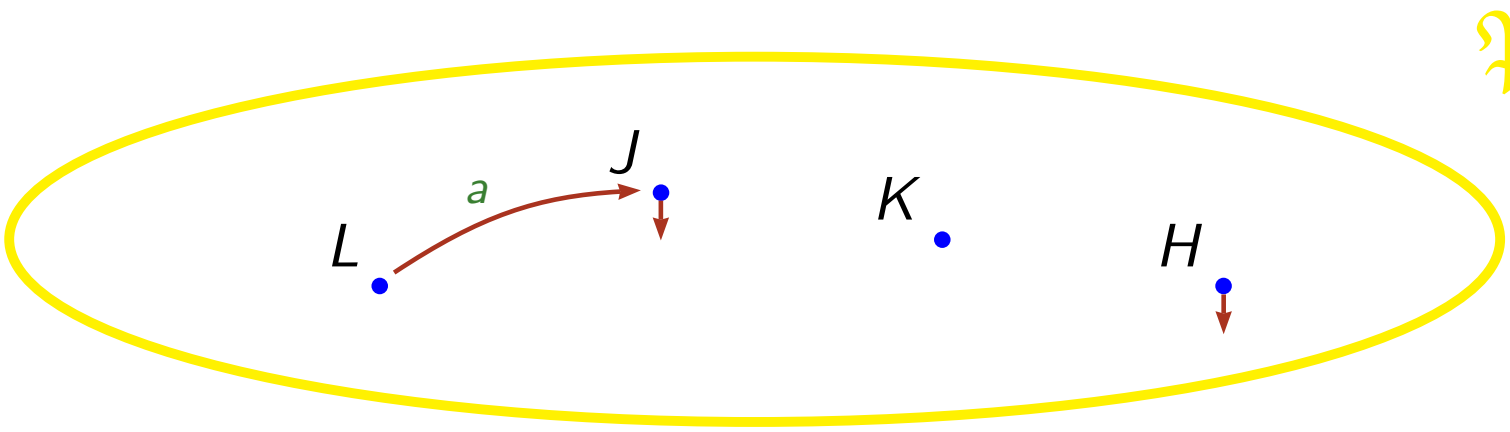
The Brzozowski derivatives



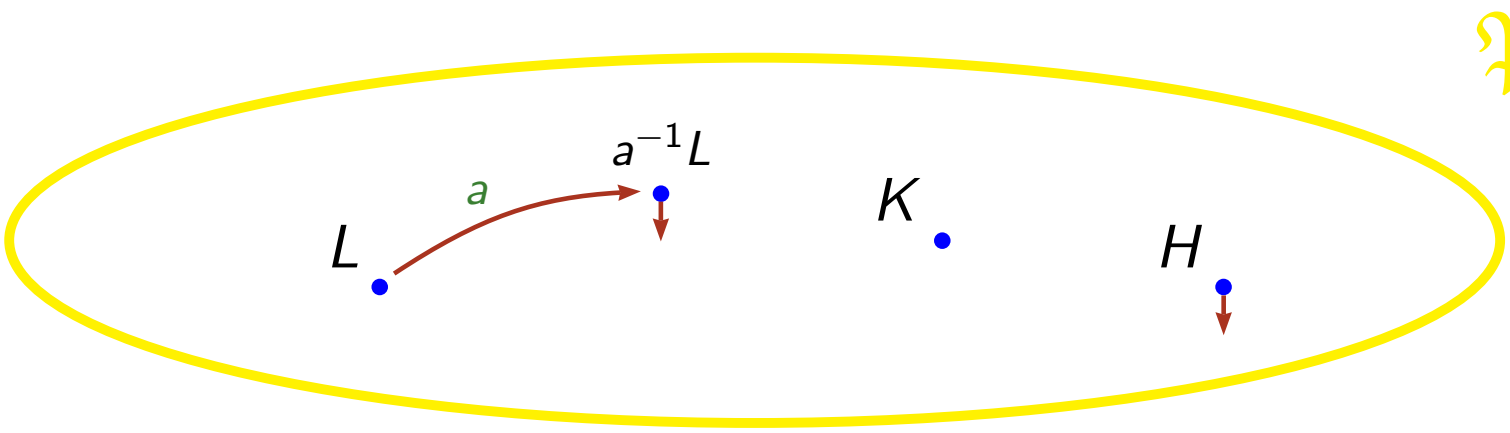
The Brzozowski derivatives



The Brzozowski derivatives

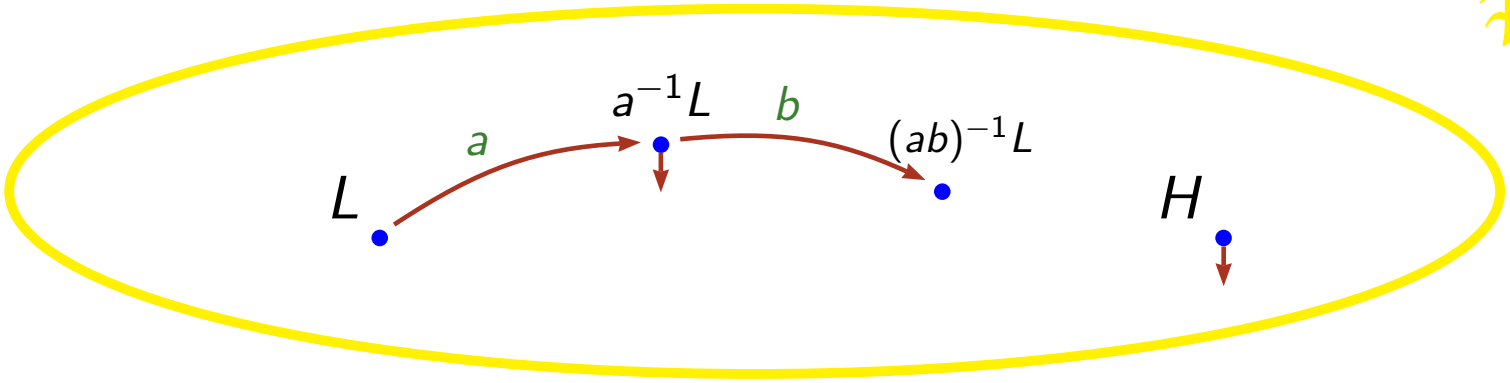


The Brzozowski derivatives



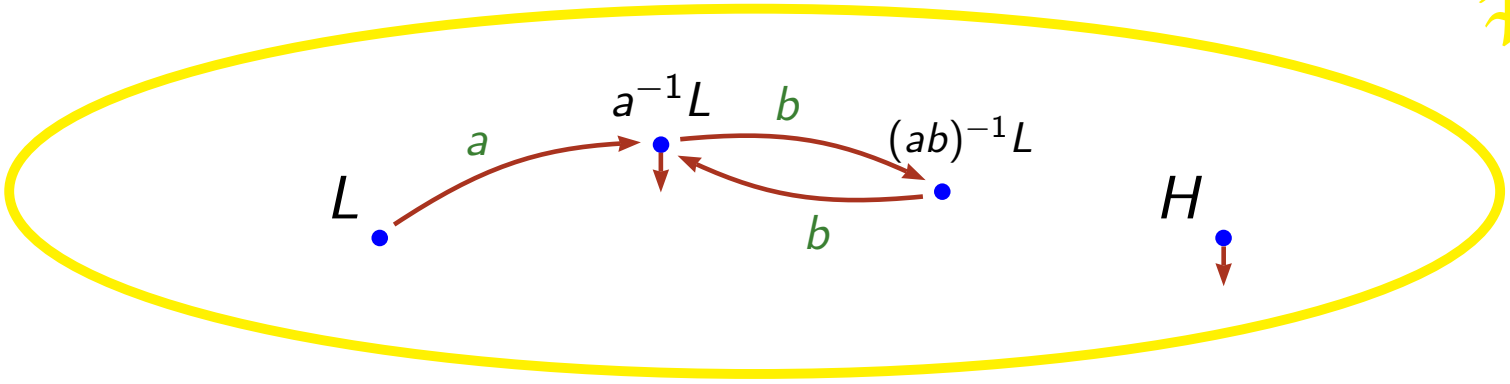
The Brzozowski derivatives

$\mathfrak{P}(A^*)$



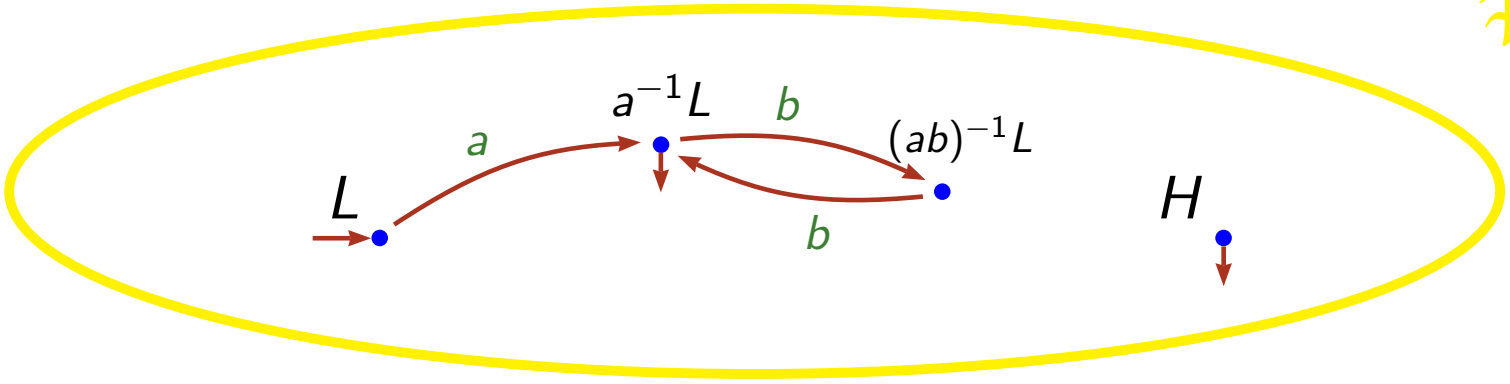
The Brzozowski derivatives

$\mathfrak{P}(A^*)$

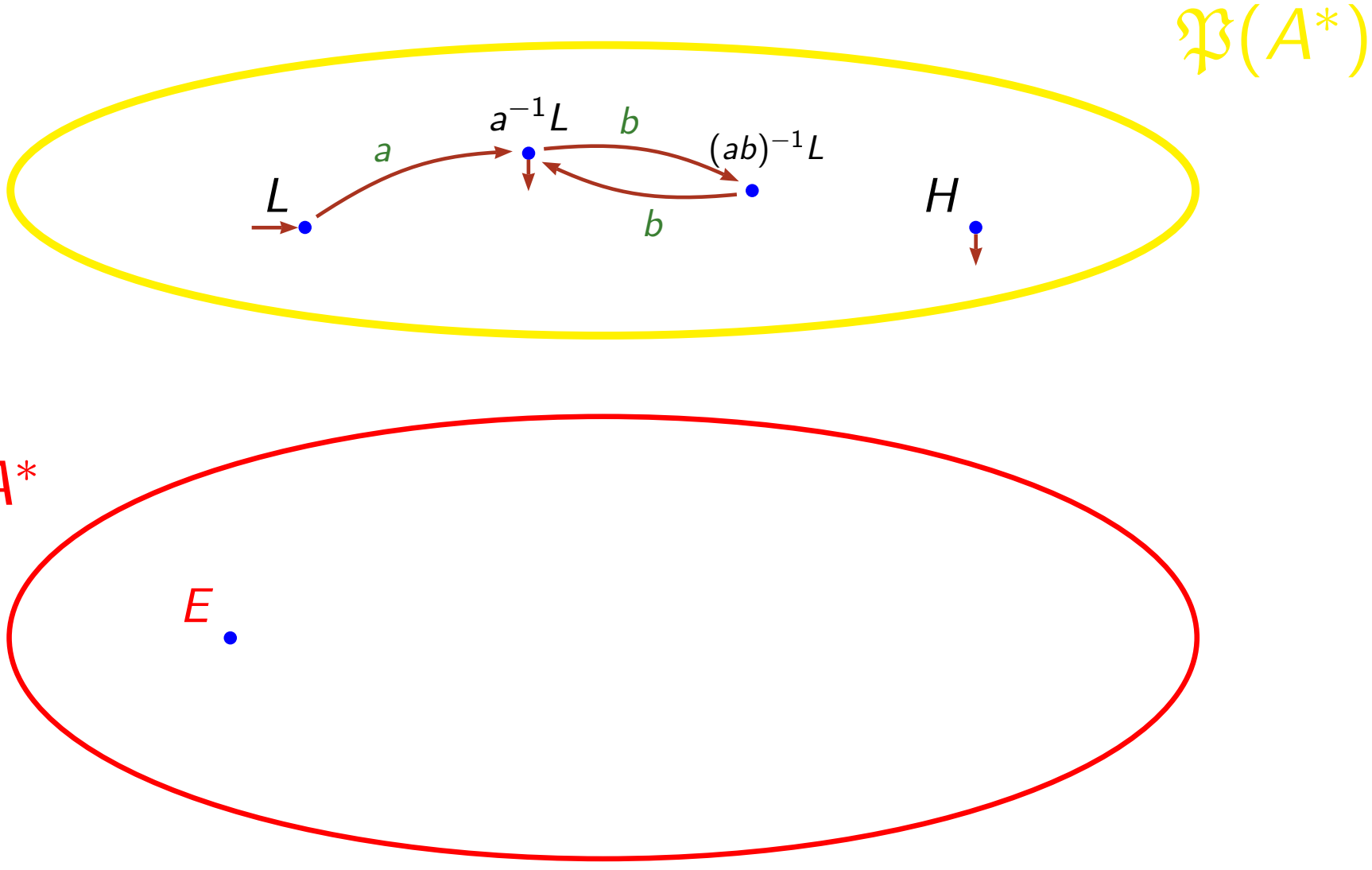


The Brzozowski derivatives

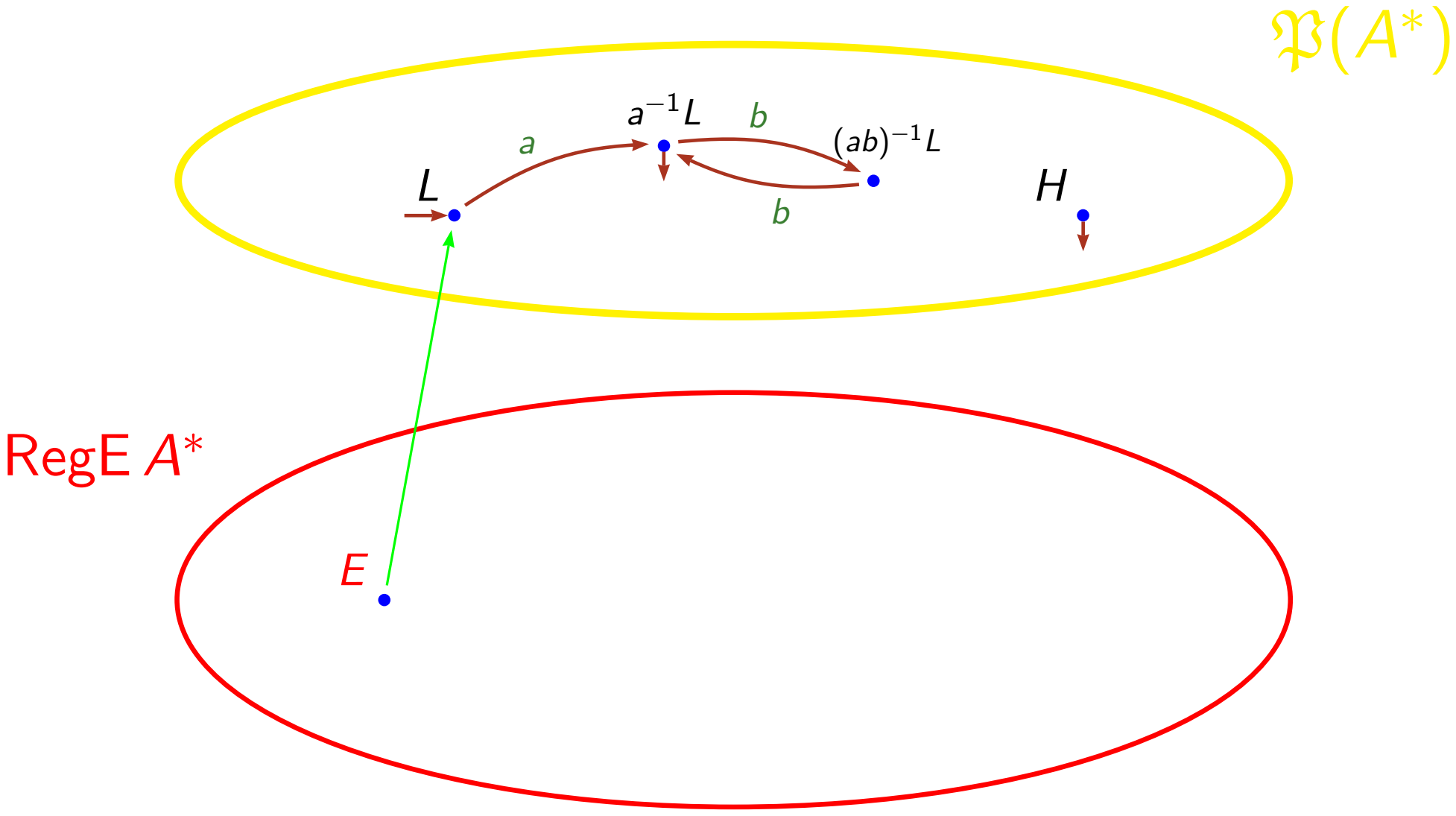
$\mathfrak{P}(A^*)$



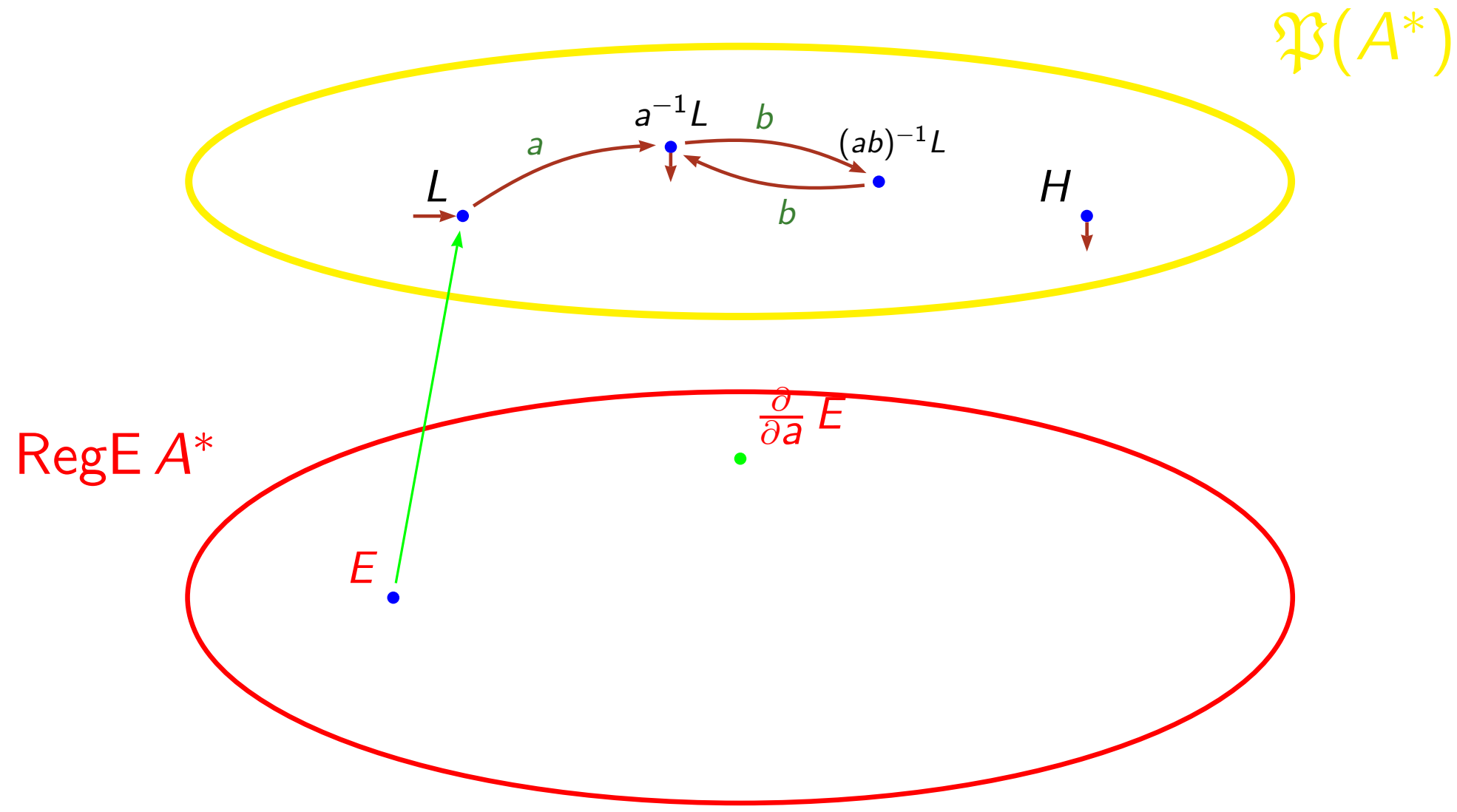
The Brzozowski derivatives



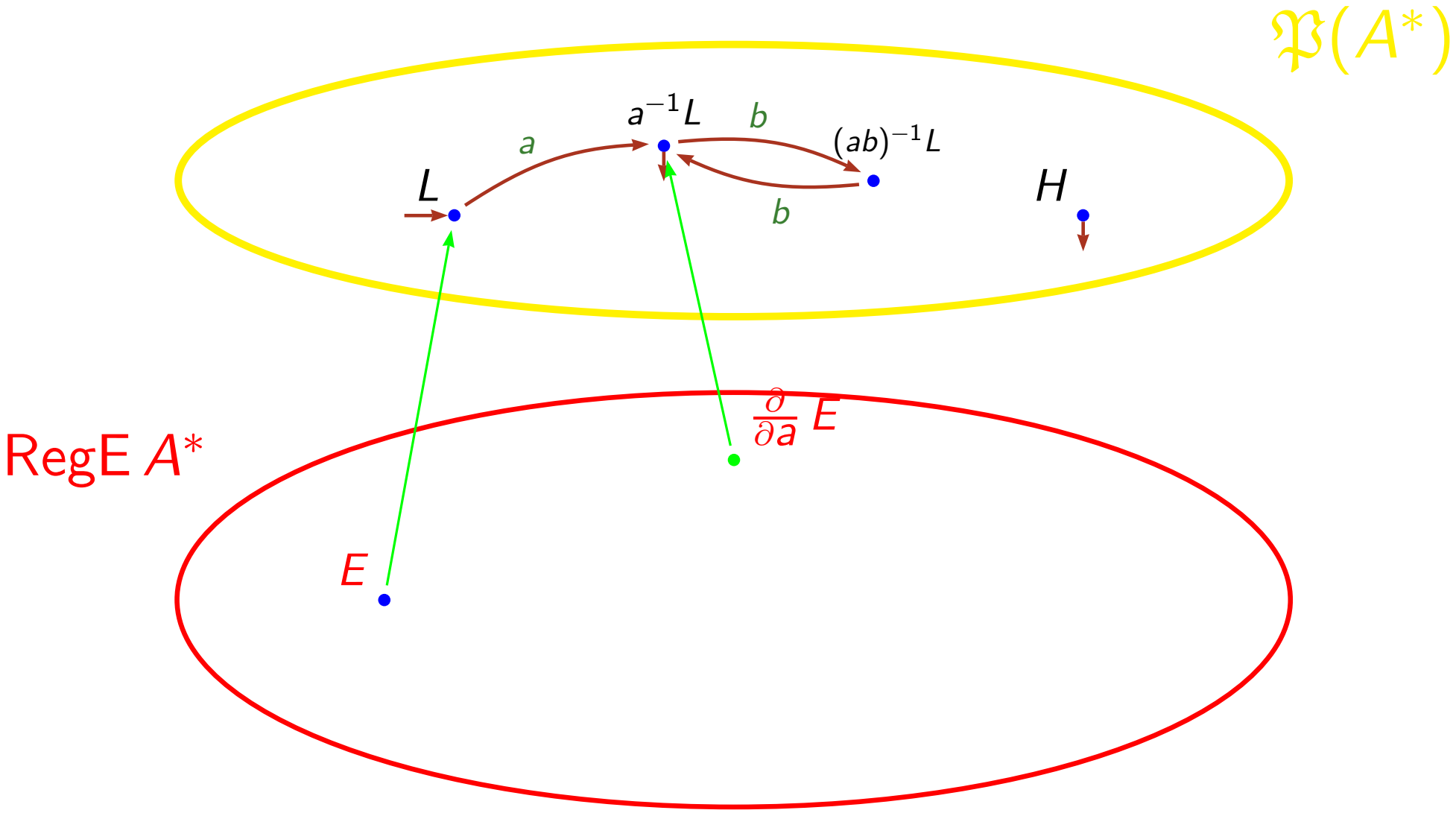
The Brzozowski derivatives



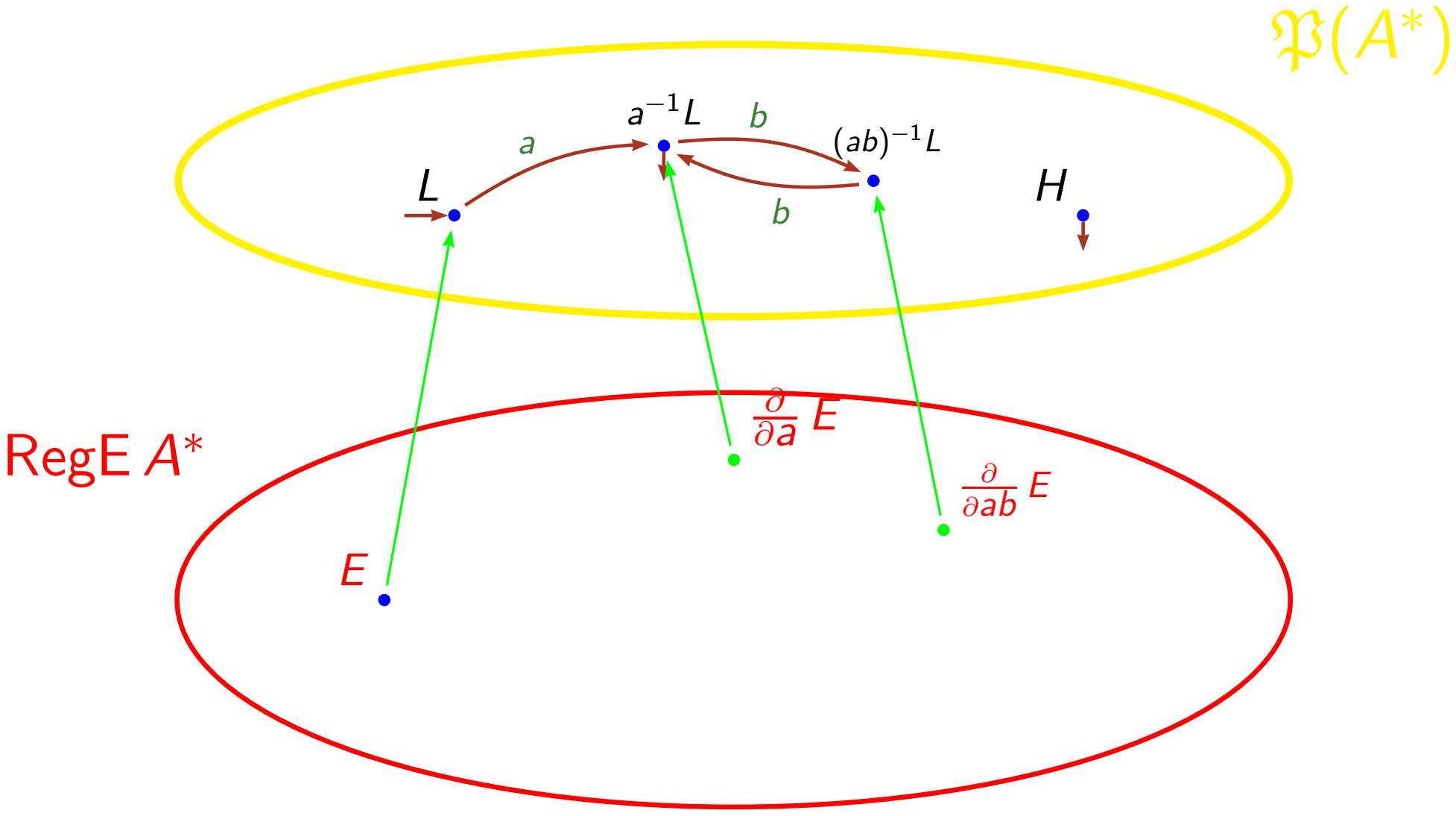
The Brzozowski derivatives



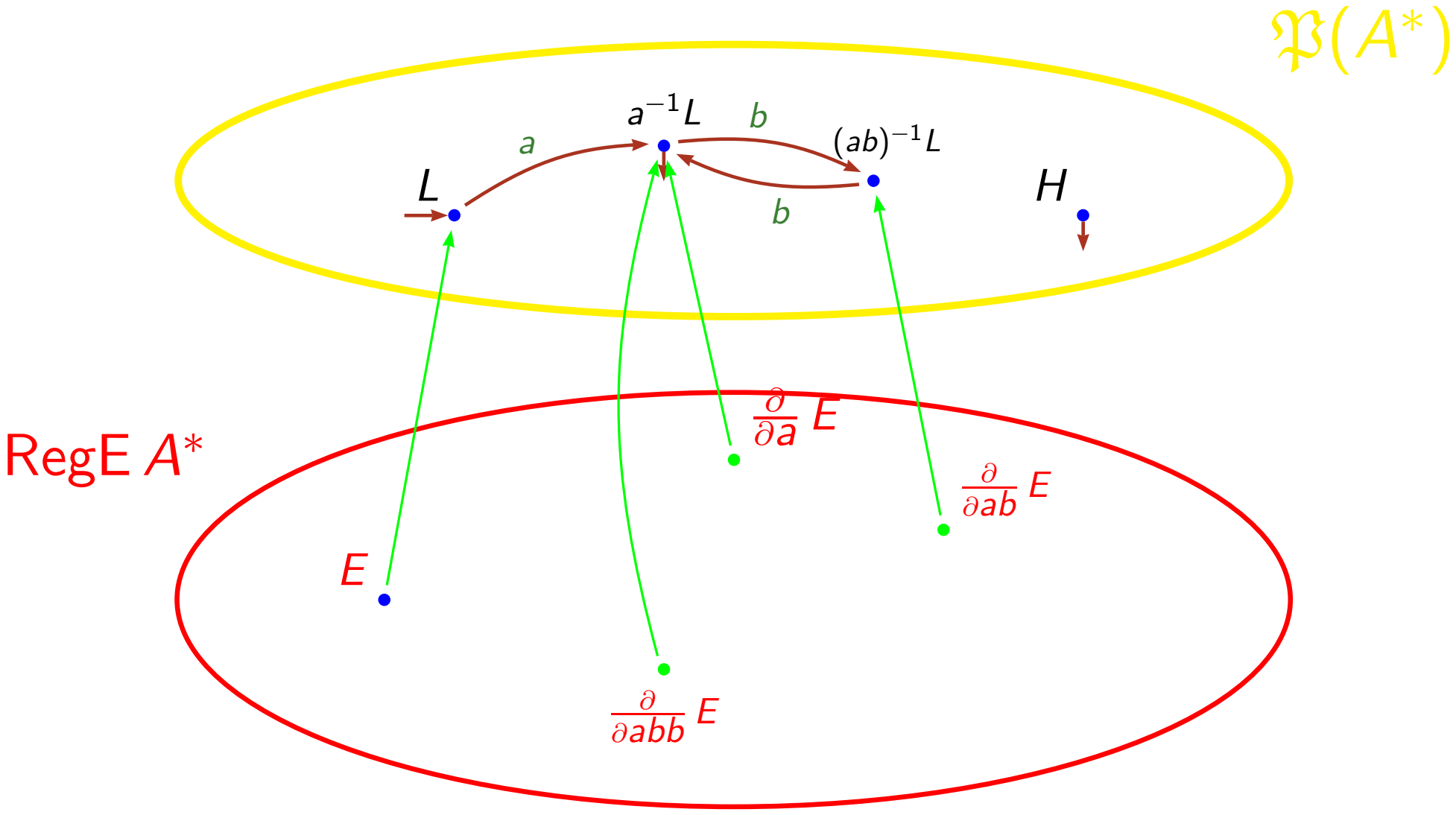
The Brzozowski derivatives



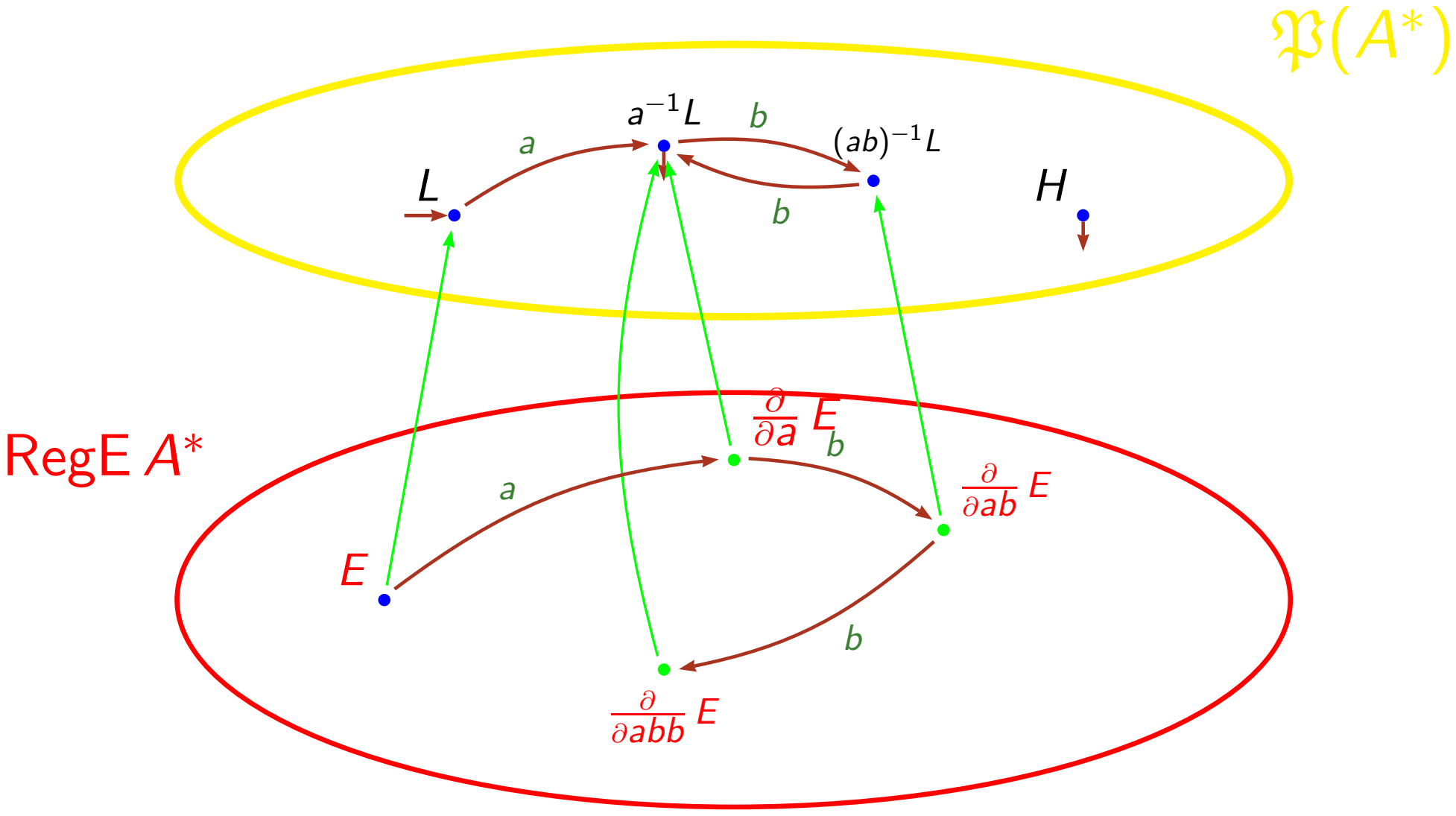
The Brzozowski derivatives



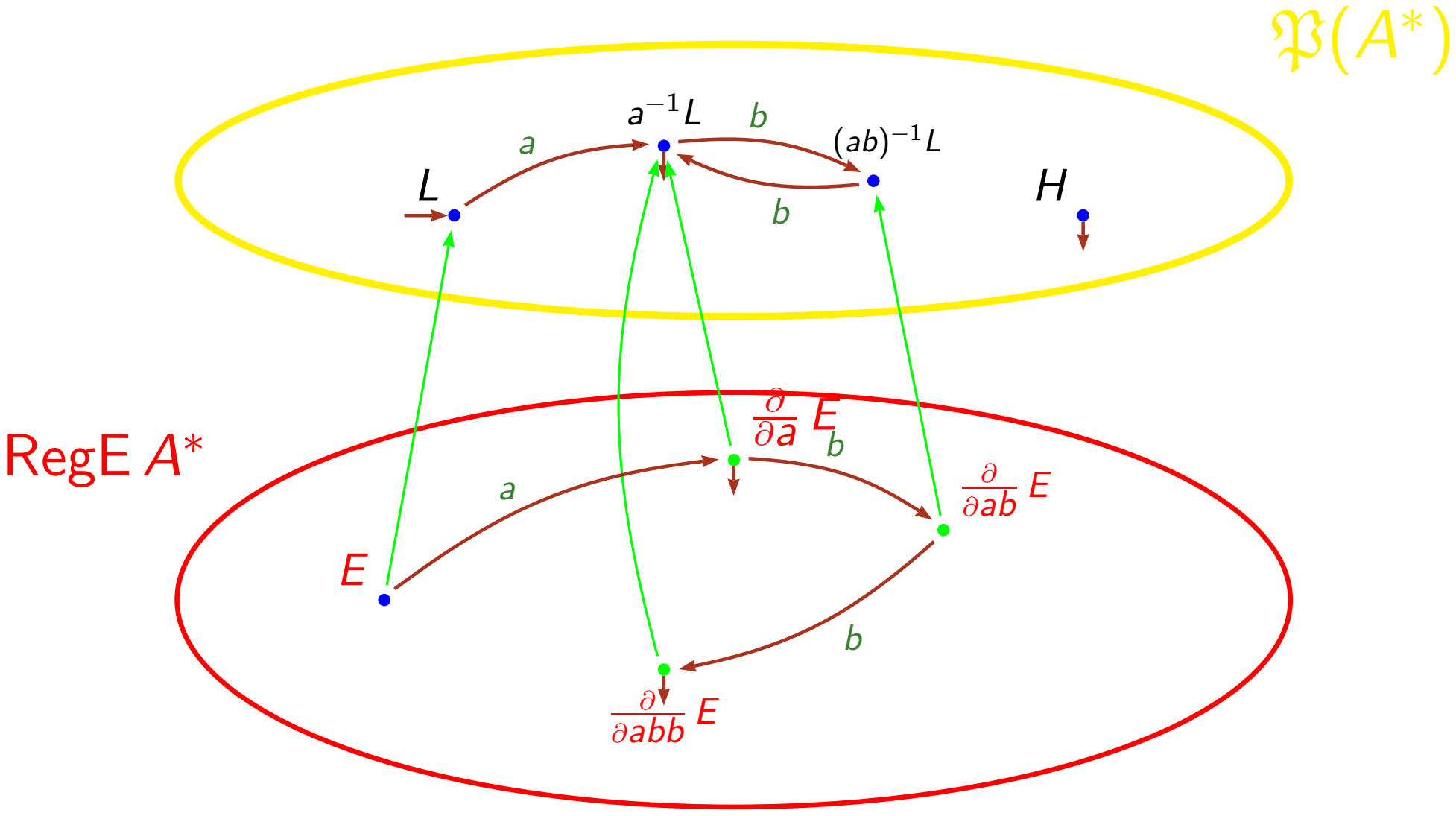
The Brzozowski derivatives



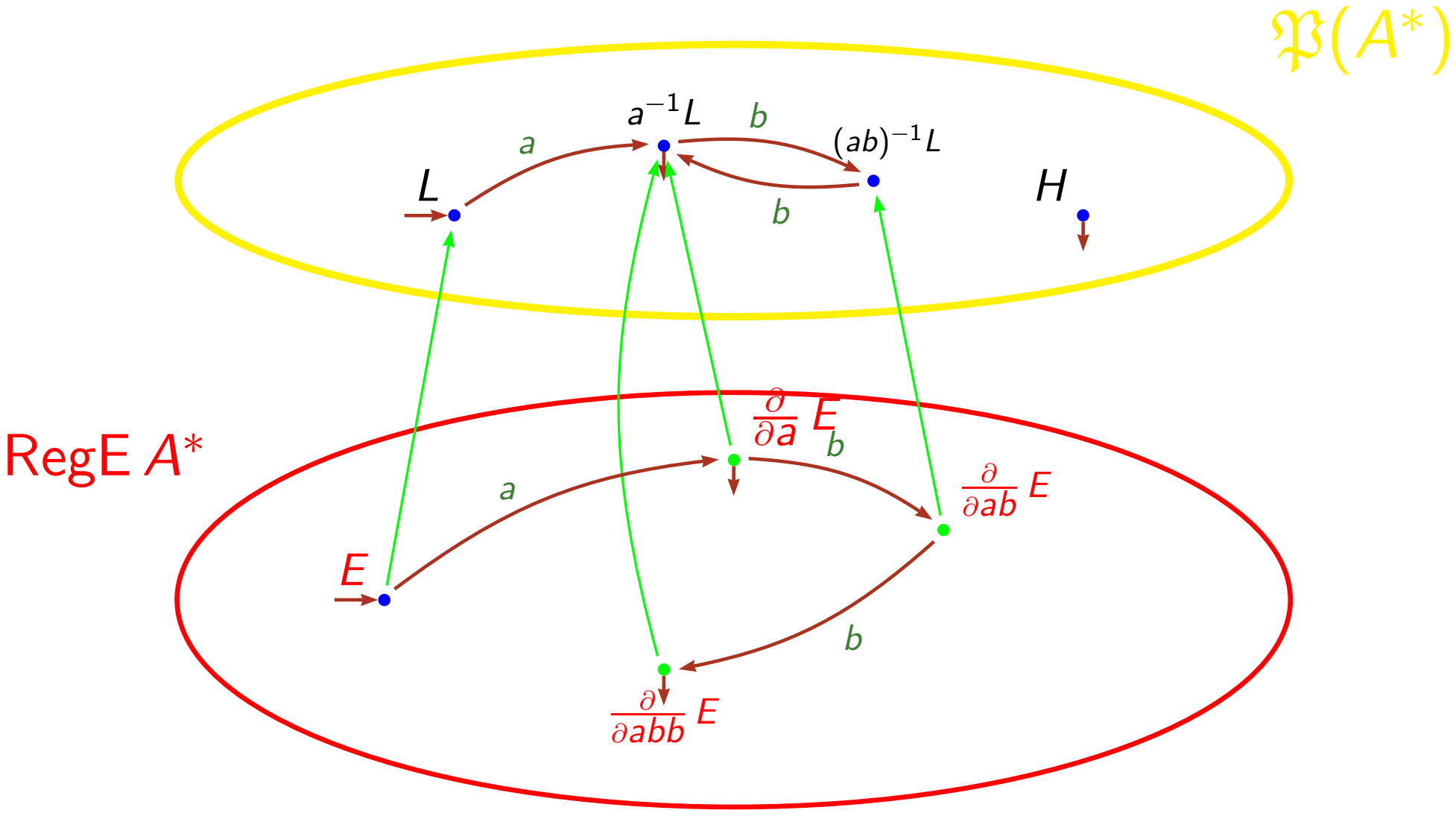
The Brzozowski derivatives



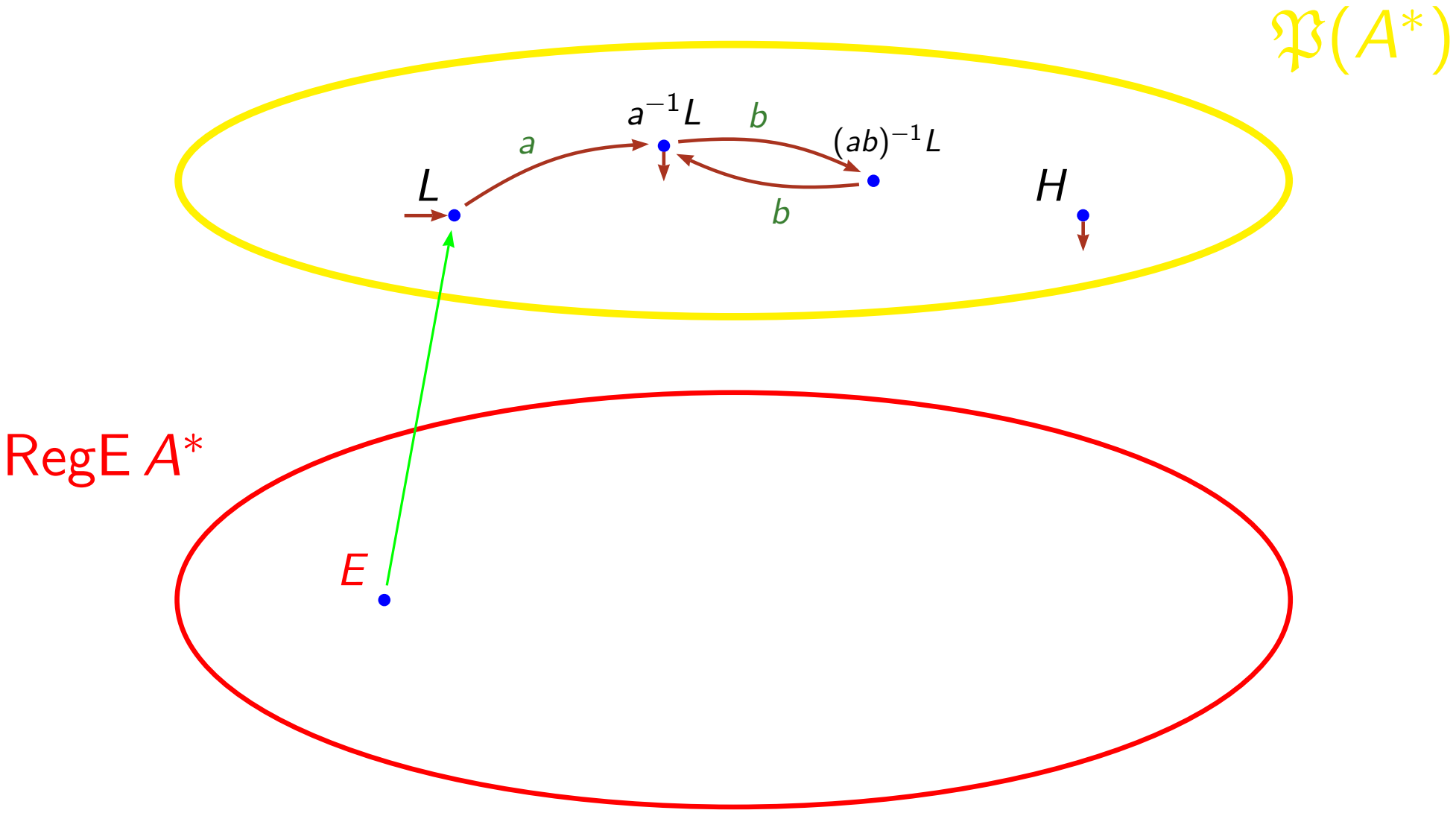
The Brzozowski derivatives



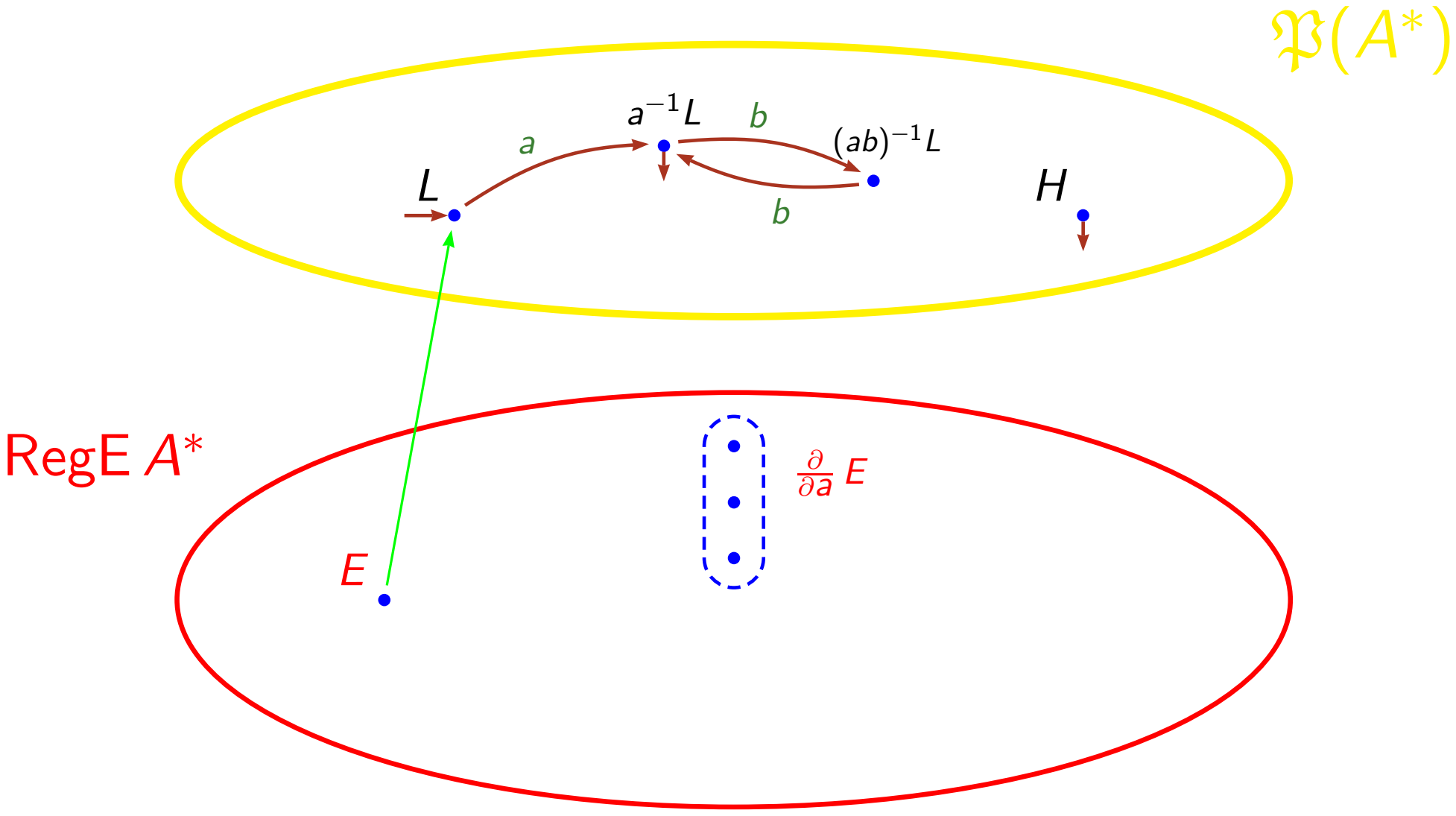
The Brzozowski derivatives



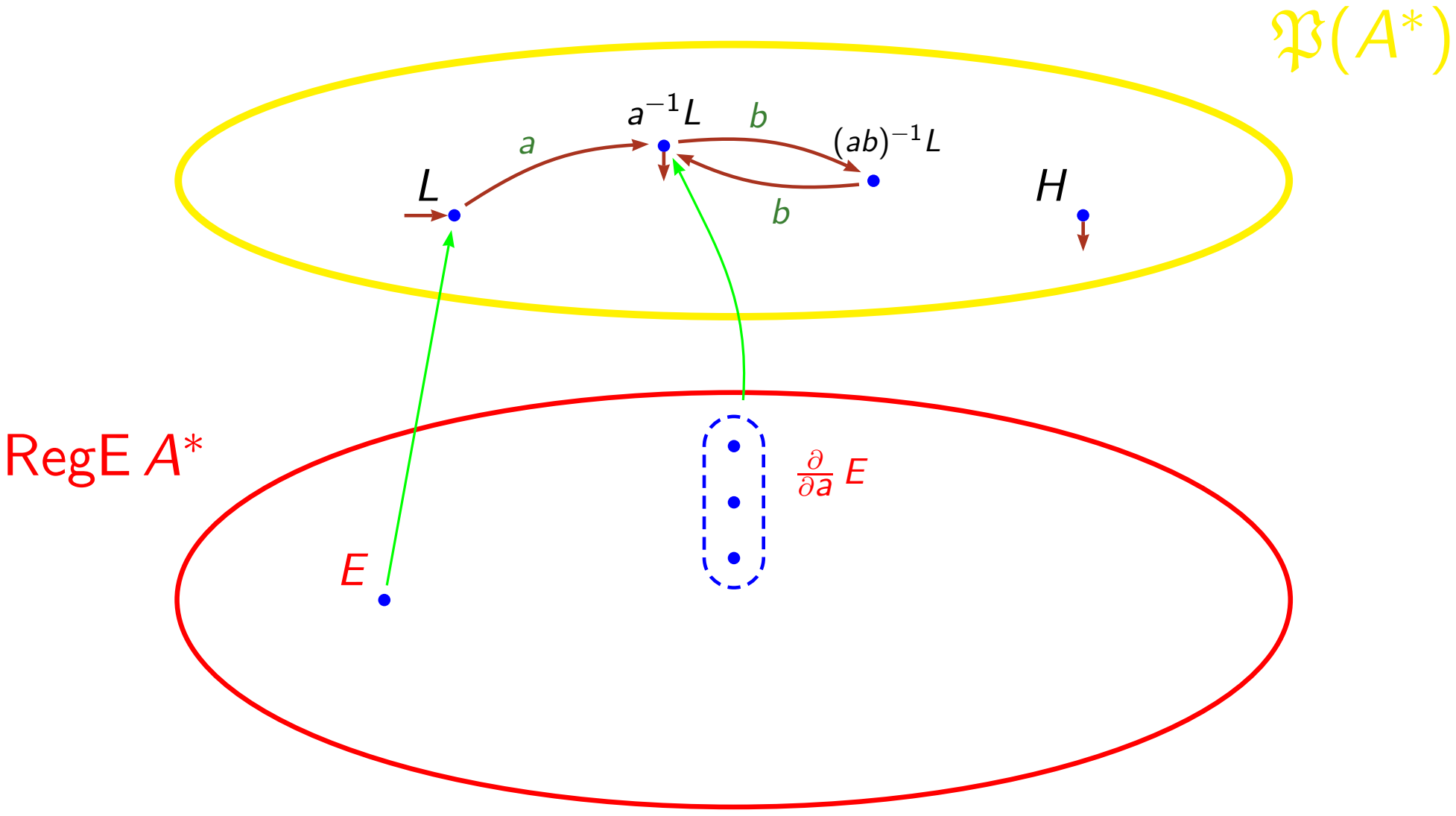
The Brzozowski–Antimirov derivation



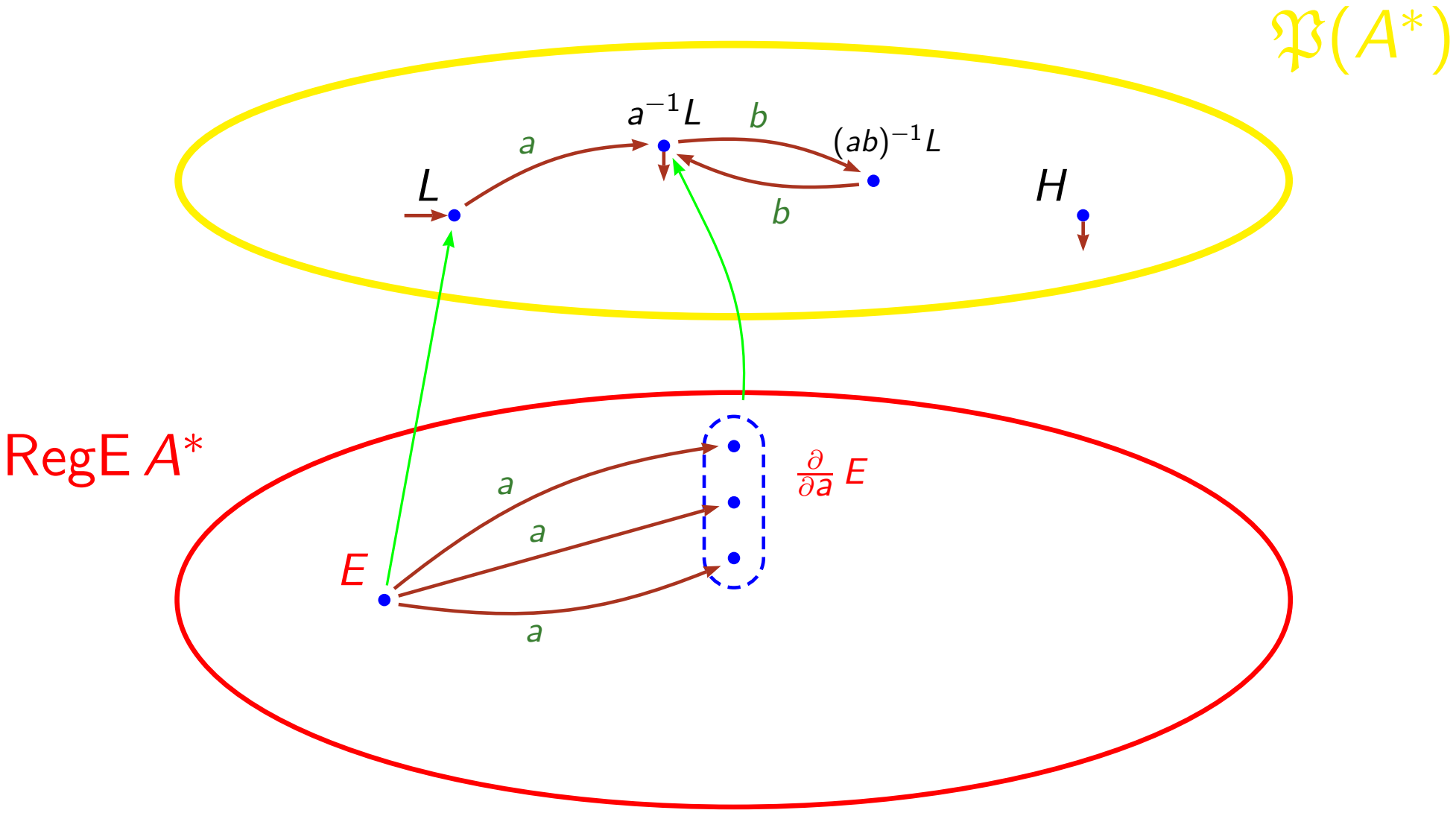
The Brzozowski–Antimirov derivation



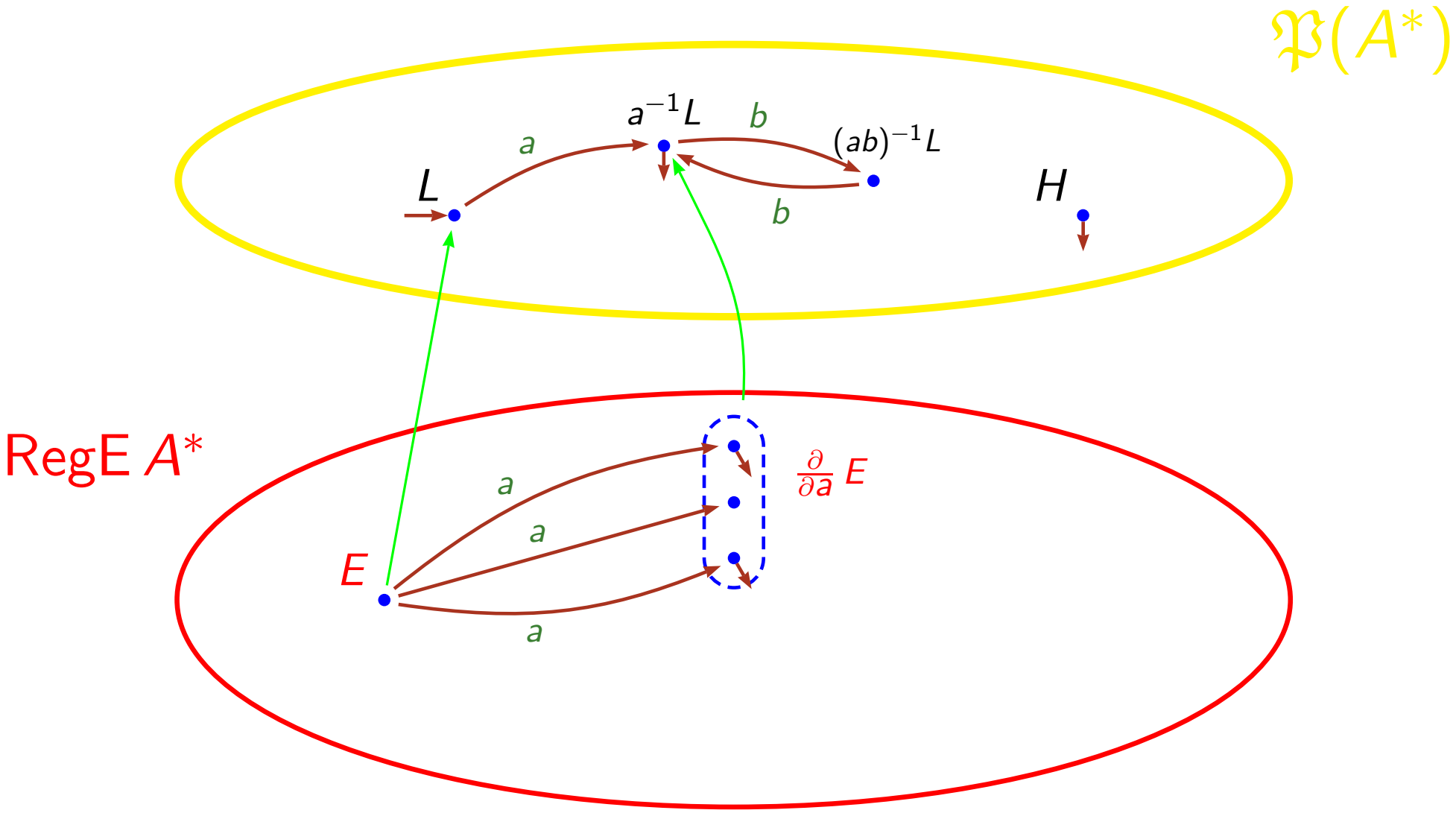
The Brzozowski–Antimirov derivation



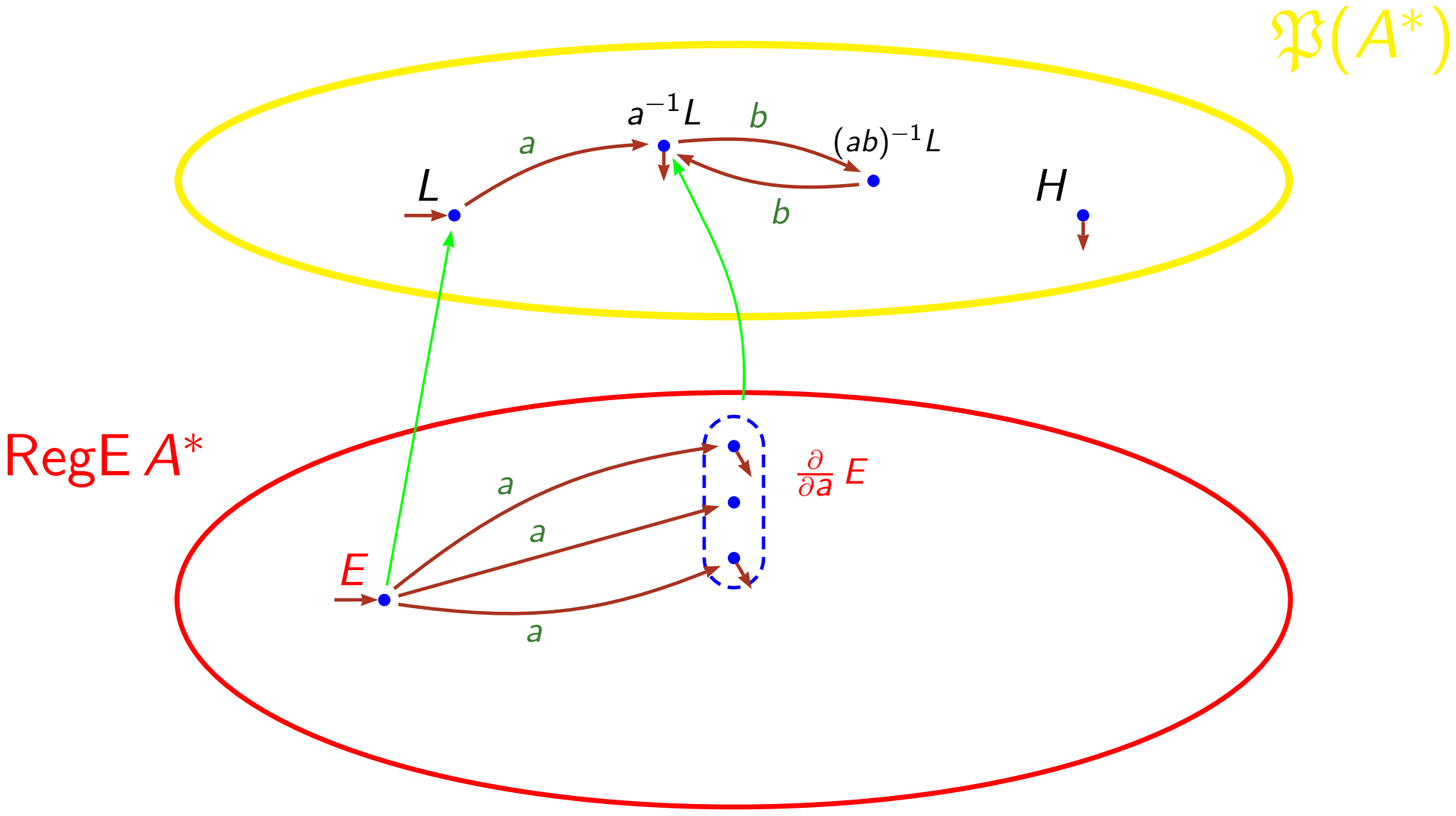
The Brzozowski–Antimirov derivation



The Brzozowski–Antimirov derivation



The Brzozowski–Antimirov derivation



The Brzozowski–Antimirov derivation

Theorem (Brzozowski 64)

$\mathcal{B}_{\mathsf{E}}$ *is a DFA which recognizes* $L(\mathsf{E})$.

Theorem (Antimirov 96)

$\mathcal{A}_{\mathsf{E}}$ *is a NFA which accepts* $L(\mathsf{E})$
and has less than $\ell(\mathsf{E}) + 2$ *states.*

$\mathcal{B}_{\mathsf{E}}$ $=$ *Determinized* ($\mathcal{A}_{\mathsf{E}}$).

Theorem (Champarnaud–Ziadi 02)

$\mathcal{A}_{\mathsf{E}}$ *is a quotient of* $\mathcal{S}_{\mathsf{E}}$.

The Ψ algorithms from an experimental point of view

The construction of $\mathcal{S}_{\mathsf{E}}$ is of quadratic complexity.

The Ψ algorithms from an experimental point of view

The construction of $\mathcal{S}_{\mathsf{E}}$ is of quadratic complexity.

Theorem (Champarnaud–Ziadi 01)

$\mathcal{A}_{\mathsf{E}}$ *can be constructed with a quadratic complexity.*

The Ψ algorithms from an experimental point of view

The construction of $\mathcal{S}_{\mathsf{E}}$ is of quadratic complexity.

Theorem (Champarnaud–Ziadi 01)

$\mathcal{A}_{\mathsf{E}}$ can be constructed with a quadratic complexity.

Observation

It appears that the computation of $\mathcal{A}_{\mathsf{E}}$ is particularly effective — sizewize and by comparison with $\mathcal{S}_{\mathsf{E}}$ — when E is obtained by a Ψ algorithm from a finite automaton.

The Ψ algorithms from an experimental point of view

The construction of $\mathcal{S}_{\mathsf{E}}$ is of quadratic complexity.

Theorem (Champarnaud–Ziadi 01)

$\mathcal{A}_{\mathsf{E}}$ can be constructed with a quadratic complexity.

Observation

It appears that the computation of $\mathcal{A}_{\mathsf{E}}$ is particularly effective — sizewize and by comparison with $\mathcal{S}_{\mathsf{E}}$ — when E is obtained by a Ψ algorithm from a finite automaton.

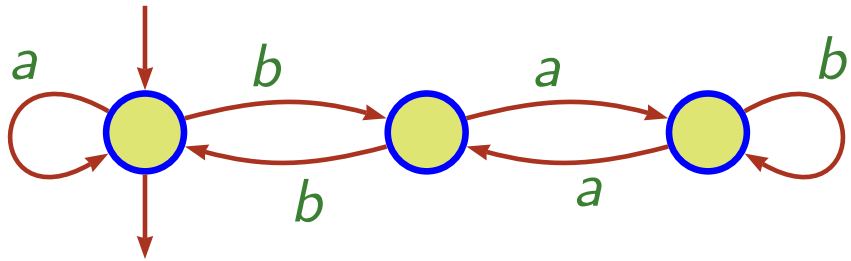
Observation

However, it seems that, even in the case of an expression E obtained by a Ψ algorithm from a finite automaton, the computation of $\mathcal{S}_{\mathsf{E}}$ followed by a quotient is far more effective than the direct computation of $\mathcal{A}_{\mathsf{E}}$.

Part IV

Do the expressions code for automata?

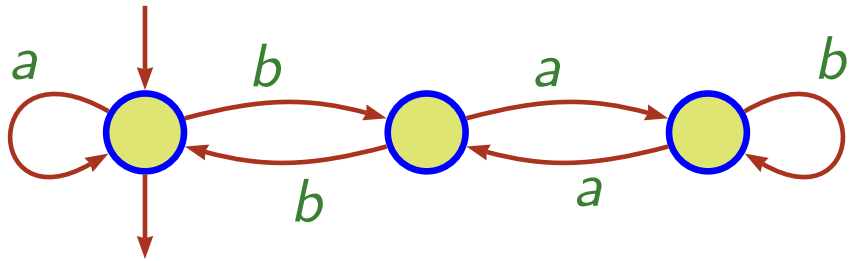
Back to our problem with state elimination



$$\mathsf{E}_1 = a^* + a^* b(ba^* b)^* ba^* + a^* b(ba^* b)^* a(b + a(ba^* b)^* a)^* a(ba^* b)^* ba^*$$

$$\mathsf{E}_2 = (a + b(ab^* a)^* b)^*$$

Back to our problem with state elimination

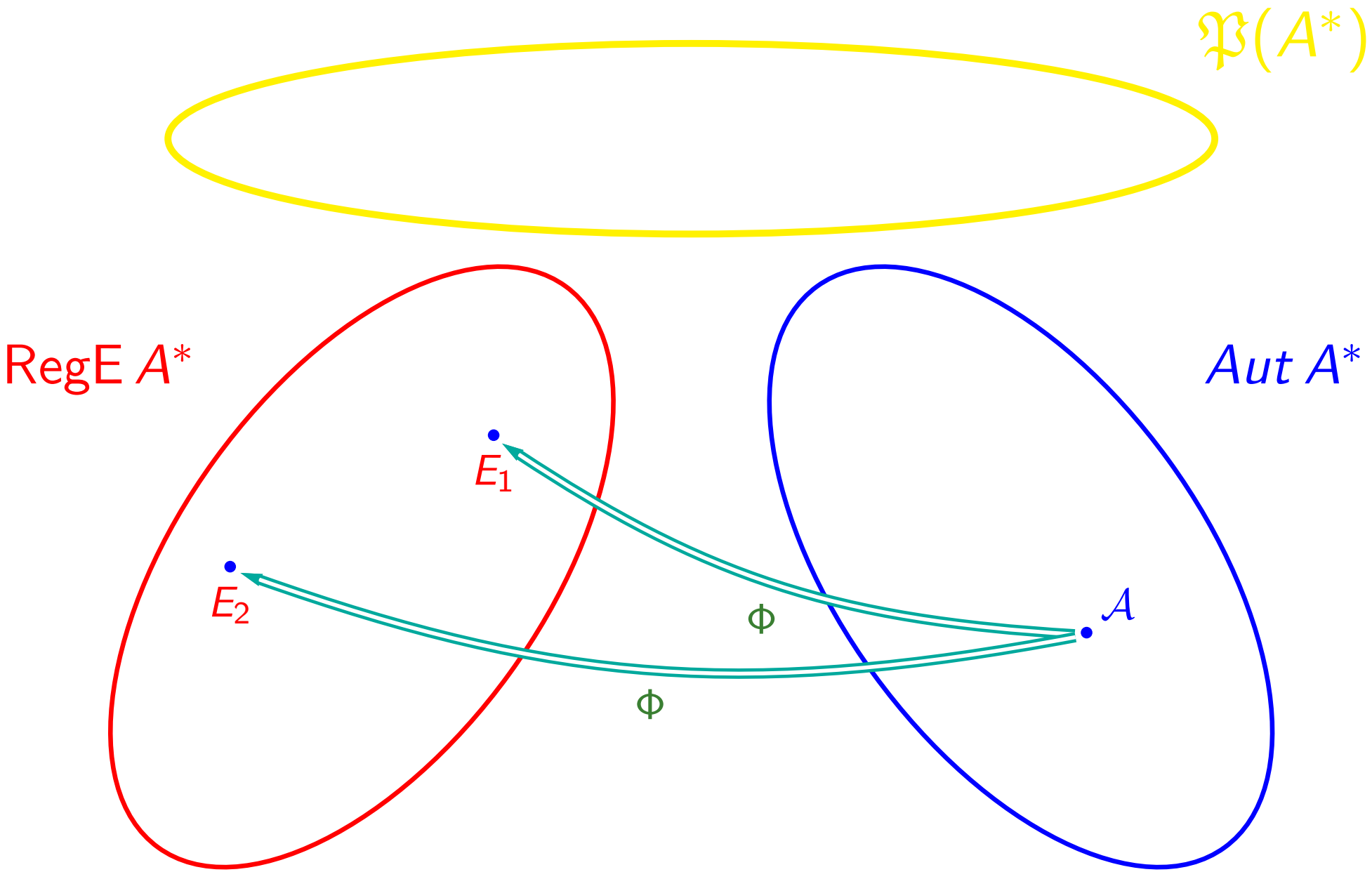


$$\mathsf{E}_1 = a^* + a^* b(ba^* b)^* ba^* + a^* b(ba^* b)^* a(b + a(ba^* b)^* a)^* a(ba^* b)^* ba^*$$

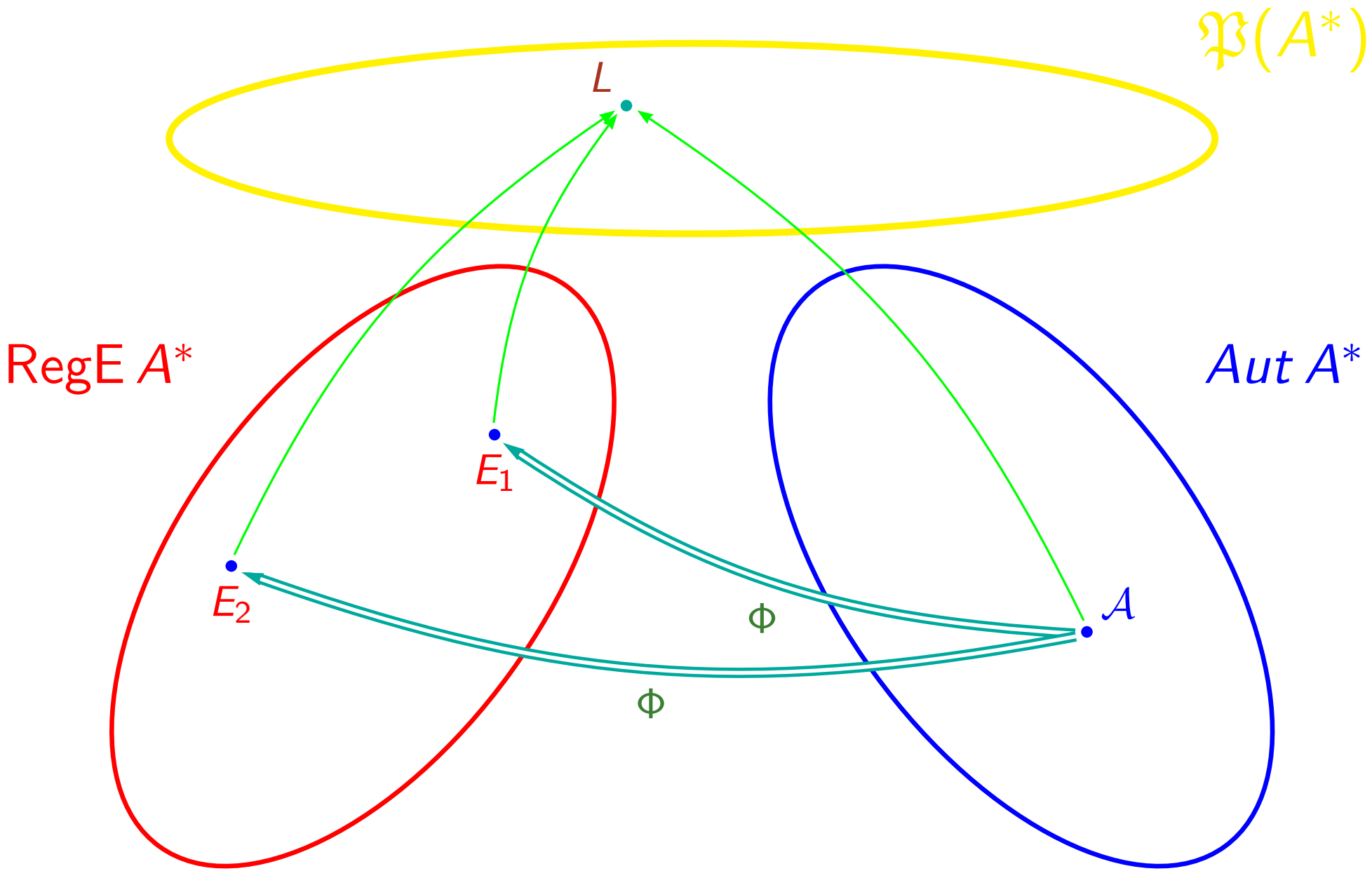
$$\mathsf{E}_2 = (a + b(ab^* a)^* b)^*$$

How to find an invariant?

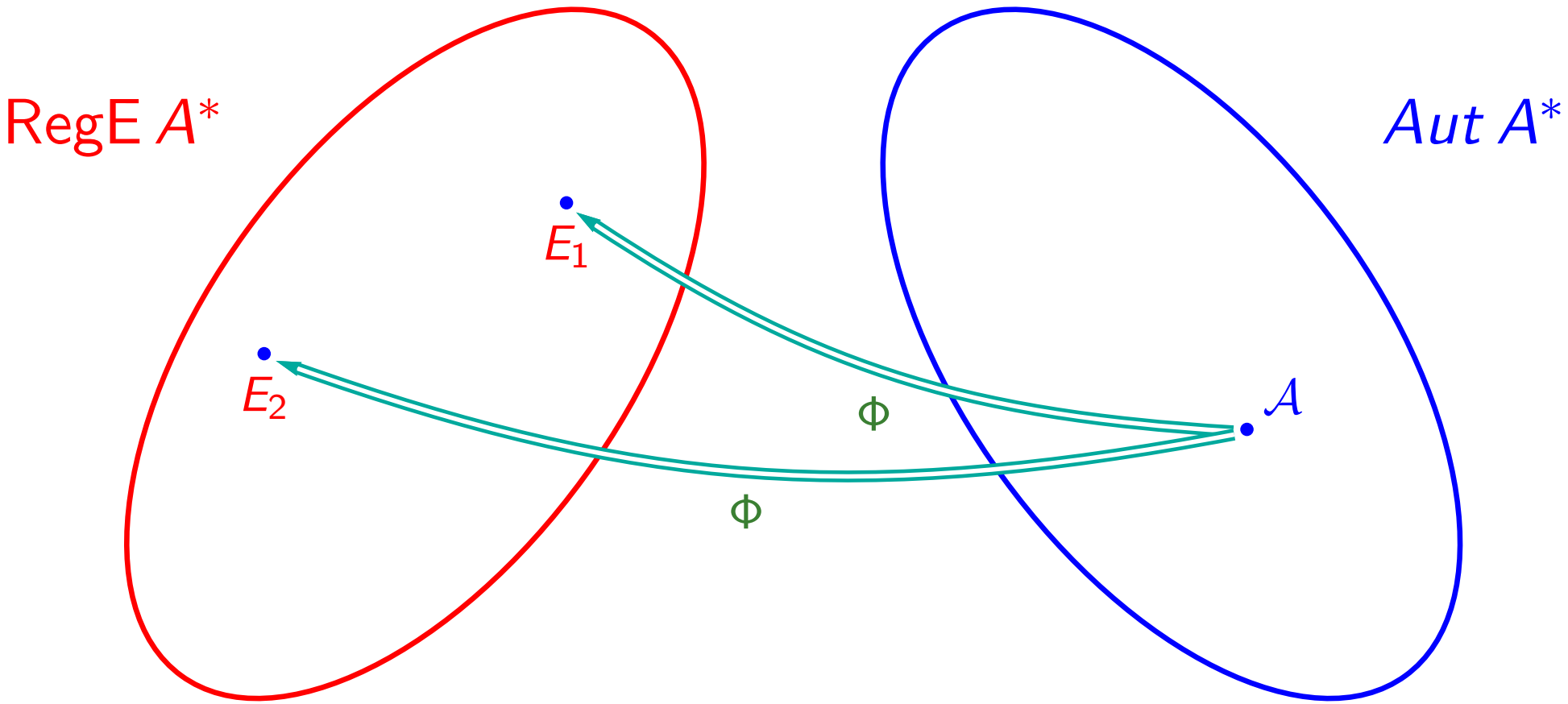
Back to our problem with state elimination



Back to our problem with state elimination

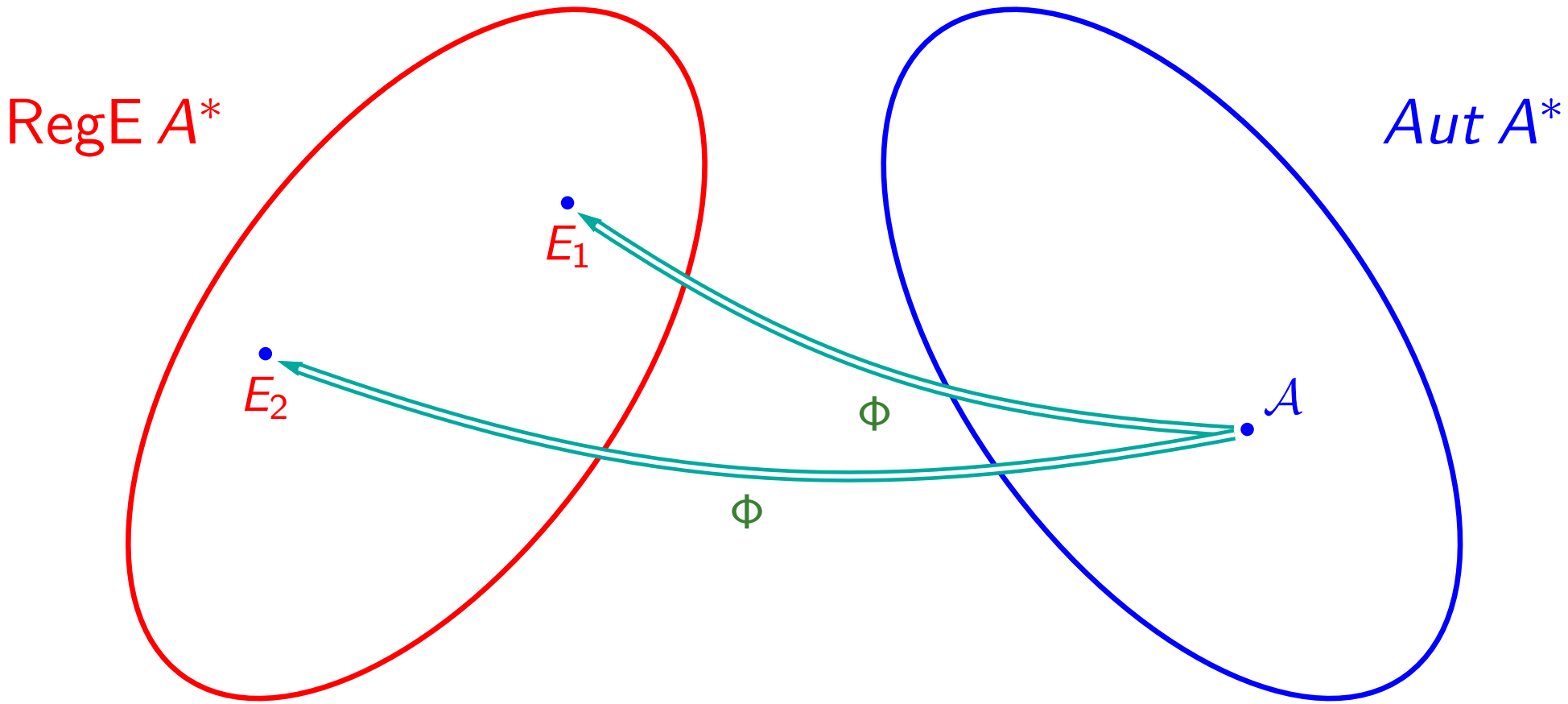


Back to our problem with state elimination



Back to our problem with state elimination

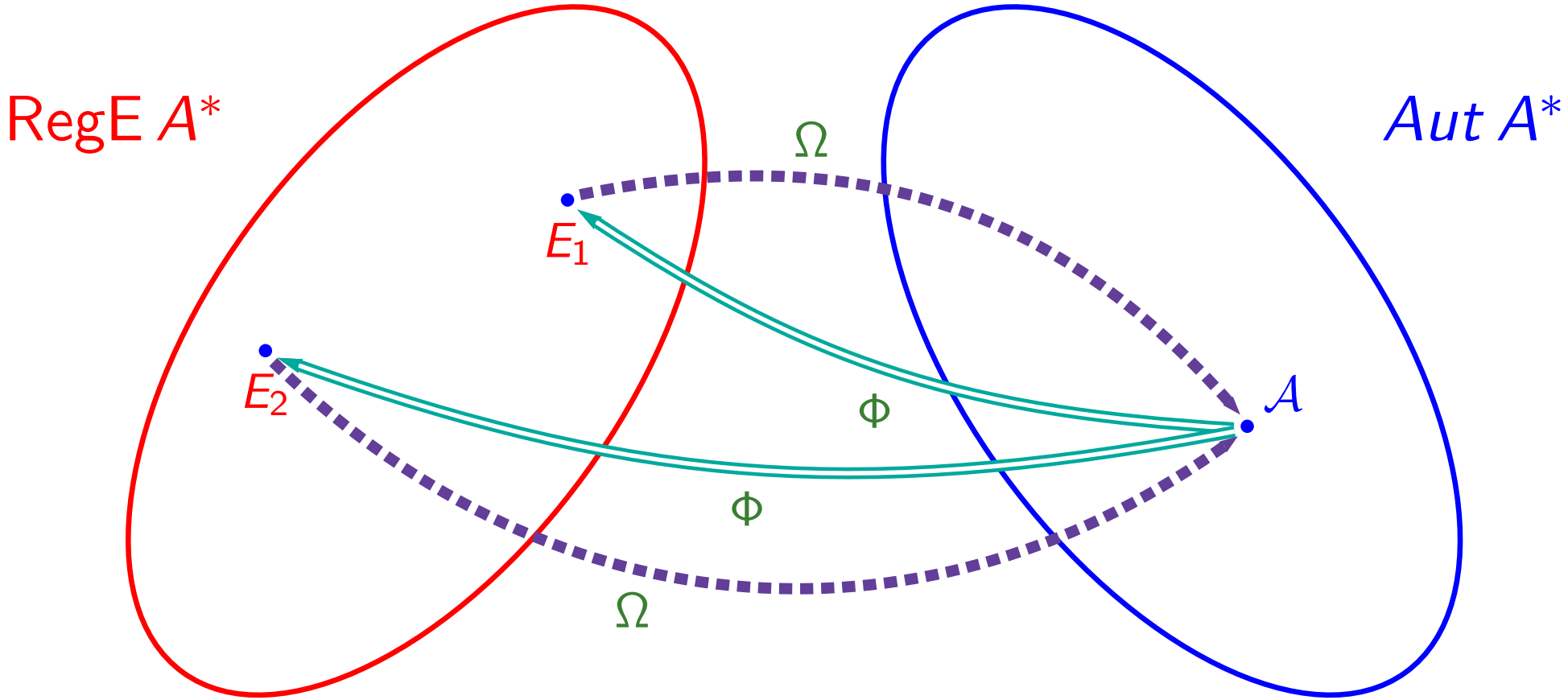
The automaton $\mathcal{A}$ is a most natural invariant!



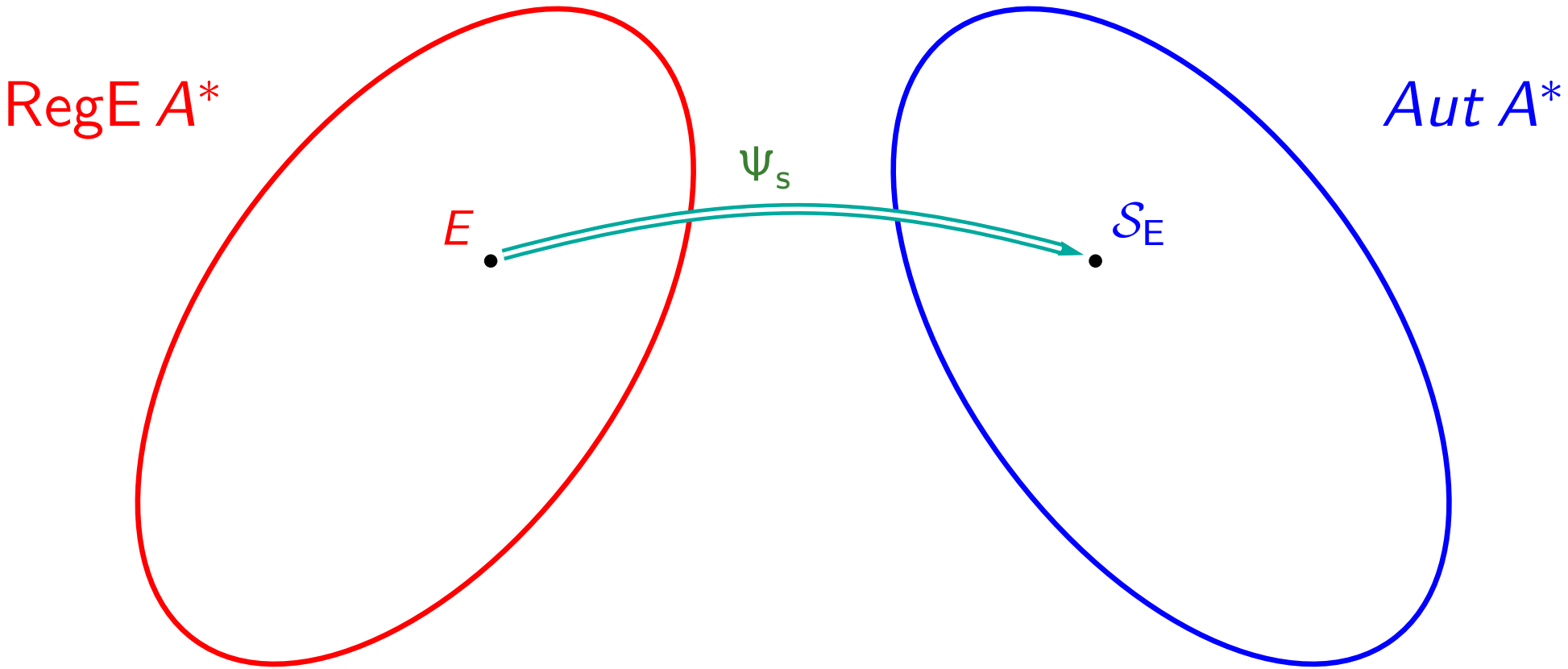
Back to our problem with state elimination

The automaton $\mathcal{A}$ is a most natural invariant!

We look for an algorithm Ω



An inverse for the Ψ algorithm



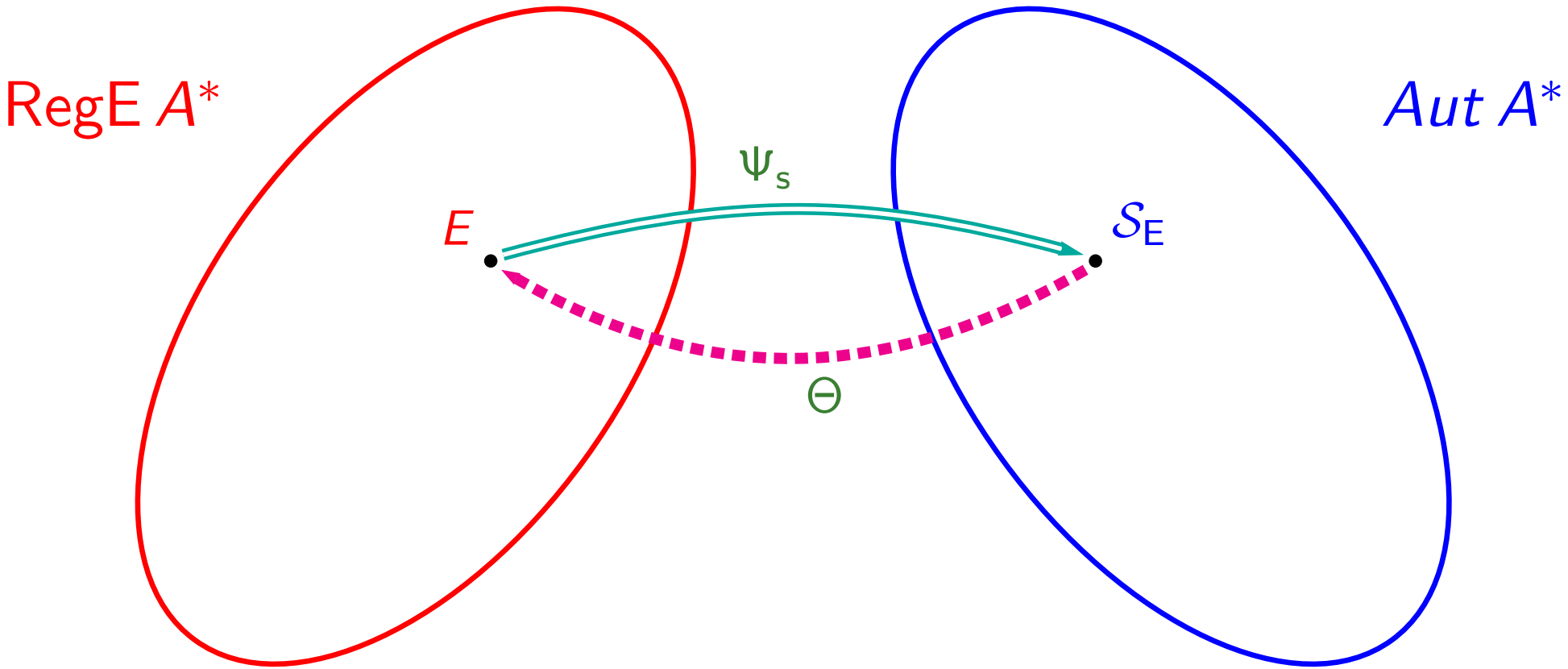
An inverse for the Ψ algorithm

Theorem (Caron–Ziadi 00)

There exists an algorithm Θ *such that*

for any star normal form expression E *it holds:*

$$\Theta(\Psi_s(\mathrm{E})) = \mathrm{E}$$



Back to Brzozowski–Antimirov derivation

Definition (Brzozowski 64 — Antimirov 96)

$\mathsf{E} \in \mathrm{RegE}\, A^*$ $\quad \frac{\partial}{\partial a}\,\mathsf{E}$ is defined by induction.

$$\frac{\partial}{\partial a}\,0 = \frac{\partial}{\partial a}\,1 = \emptyset\,, \qquad \frac{\partial}{\partial a}\,b = \begin{cases} \{1\} & \text{if} \quad b = a \\ \emptyset & \text{otherwise} \end{cases}$$

$$\frac{\partial}{\partial a}(\mathsf{E}+\mathsf{F}) = \frac{\partial}{\partial a}\,\mathsf{E} \cup \frac{\partial}{\partial a}\,\mathsf{F}$$

$$\frac{\partial}{\partial a}(\mathsf{E}\cdot\mathsf{F}) = \left[\frac{\partial}{\partial a}\,\mathsf{E}\right]\cdot\mathsf{F} \cup \mathsf{c}(\mathsf{E})\,\frac{\partial}{\partial a}\,\mathsf{F}$$

$$\frac{\partial}{\partial a}(\mathsf{E}^*) = \left[\frac{\partial}{\partial a}\,\mathsf{E}\right]\cdot\mathsf{E}^*$$

Back to Brzozowski–Antimirov derivation

Definition (Brzozowski 64 — Antimirov 96)

$\mathsf{E} \in \mathrm{RegE}\, A^*$ $\quad \frac{\partial}{\partial a}\mathsf{E}$ is defined by induction.

$$\frac{\partial}{\partial a} 0 = \frac{\partial}{\partial a} 1 = \emptyset\,, \qquad \frac{\partial}{\partial a} b = \begin{cases} \{1\} & \text{if} \quad b = a \\ \emptyset & \text{otherwise} \end{cases}$$

$$\frac{\partial}{\partial a}(\mathsf{E}+\mathsf{F}) = \frac{\partial}{\partial a}\mathsf{E} \cup \frac{\partial}{\partial a}\mathsf{F}$$

$$\frac{\partial}{\partial a}(\mathsf{E}\cdot\mathsf{F}) = \left[\frac{\partial}{\partial a}\mathsf{E}\right]\cdot\mathsf{F} \cup \mathrm{c}(\mathsf{E})\ \frac{\partial}{\partial a}\mathsf{F}$$

$$\frac{\partial}{\partial a}(\mathsf{E}^*) = \left[\frac{\partial}{\partial a}\mathsf{E}\right]\cdot\mathsf{E}^*$$

$$\frac{\partial}{\partial a}\left[\bigcup_{i\in I}\mathsf{E}_i\right] = \bigcup_{i\in I}\frac{\partial}{\partial a}\mathsf{E}_i\,, \qquad \left[\bigcup_{i\in I}\mathsf{E}_i\right]\cdot\mathsf{F} = \bigcup_{i\in I}(\mathsf{E}_i\cdot\mathsf{F})\,.$$

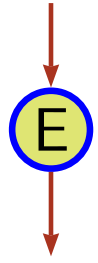
$$\frac{\partial}{\partial fa}\mathsf{E} = \frac{\partial}{\partial a}\left(\frac{\partial}{\partial f}\mathsf{E}\right)$$

A first example

$$\mathsf{E} = (a + b(ab^*a)^*b)^* = \mathsf{H}^*$$

A first example

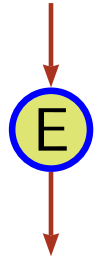
$$\mathsf{E} = (a + b(ab^*a)^*b)^* = \mathsf{H}^*$$



A first example

$$\mathsf{E} = (a + b(ab^*a)^*b)^* = \mathsf{H}^*$$

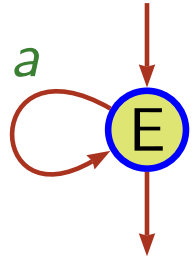
$$\frac{\partial}{\partial a}\mathsf{E} = \left[\frac{\partial}{\partial a}\mathsf{H}\right]\mathsf{E} = \mathsf{E}$$



A first example

$$\mathsf{E} = (a + b(ab^*a)^*b)^* = \mathsf{H}^*$$

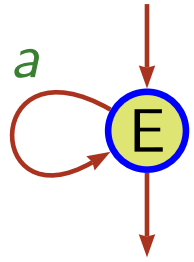
$$\frac{\partial}{\partial a}\mathsf{E} = \left[\frac{\partial}{\partial a}\mathsf{H}\right]\mathsf{E} = \mathsf{E}$$



A first example

$$\mathsf{E} = (a + b(ab^*a)^*b)^* = \mathsf{H}^*$$

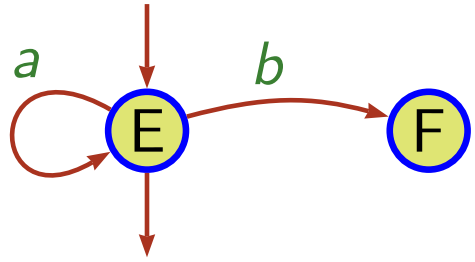
$$\frac{\partial}{\partial a}\mathsf{E} = \left[\frac{\partial}{\partial a}\mathsf{H}\right]\mathsf{E} = \mathsf{E} \qquad \frac{\partial}{\partial b}\mathsf{E} = \left[\frac{\partial}{\partial b}\mathsf{H}\right]\mathsf{E} = (ab^*a)^*b\,\mathsf{E} = \mathsf{F}$$



A first example

$$\mathsf{E} = (a + b(ab^*a)^*b)^* = \mathsf{H}^*$$

$$\frac{\partial}{\partial a}\mathsf{E} = \left[\frac{\partial}{\partial a}\mathsf{H}\right]\mathsf{E} = \mathsf{E} \qquad \frac{\partial}{\partial b}\mathsf{E} = \left[\frac{\partial}{\partial b}\mathsf{H}\right]\mathsf{E} = (ab^*a)^*b\,\mathsf{E} = \mathsf{F}$$

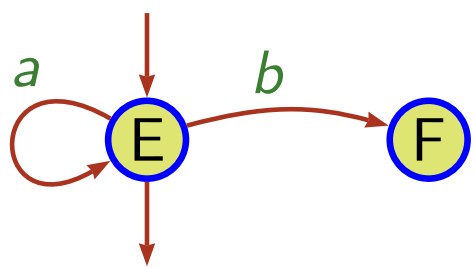


A first example

$$\mathsf{E} = (a + b(ab^*a)^*b)^* = \mathsf{H}^*$$

$$\frac{\partial}{\partial a}\mathsf{E} = \left[\frac{\partial}{\partial a}\mathsf{H}\right]\mathsf{E} = \mathsf{E} \qquad \frac{\partial}{\partial b}\mathsf{E} = \left[\frac{\partial}{\partial b}\mathsf{H}\right]\mathsf{E} = (ab^*a)^*b\,\mathsf{E} = \mathsf{F}$$

$$\frac{\partial}{\partial a}\mathsf{F} = \left[\frac{\partial}{\partial a}(ab^*a)^*b\right]\mathsf{E} = b^*a(ab^*a)^*b\,\mathsf{E} = b^*a\,\mathsf{F} = \mathsf{G}$$

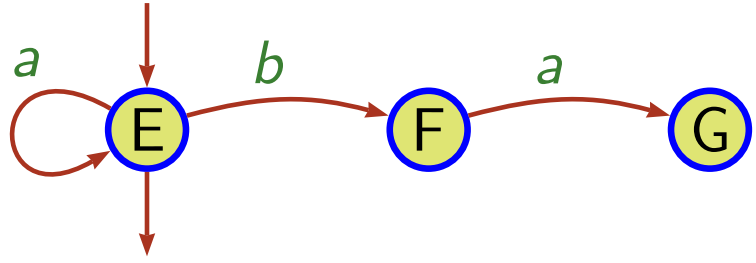


A first example

$$\mathsf{E} = (a + b(ab^*a)^*b)^* = \mathsf{H}^*$$

$$\frac{\partial}{\partial a}\mathsf{E} = \left[\frac{\partial}{\partial a}\mathsf{H}\right]\mathsf{E} = \mathsf{E} \qquad \frac{\partial}{\partial b}\mathsf{E} = \left[\frac{\partial}{\partial b}\mathsf{H}\right]\mathsf{E} = (ab^*a)^*b\,\mathsf{E} = \mathsf{F}$$

$$\frac{\partial}{\partial a}\mathsf{F} = \left[\frac{\partial}{\partial a}(ab^*a)^*b\right]\mathsf{E} = b^*a(ab^*a)^*b\,\mathsf{E} = b^*a\,\mathsf{F} = \mathsf{G}$$



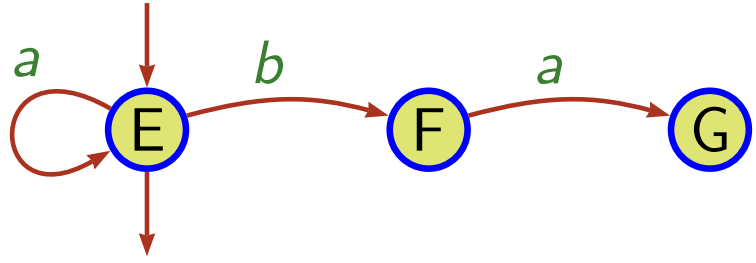
A first example

$$\mathsf{E} = (a + b(ab^*a)^*b)^* = \mathsf{H}^*$$

$$\frac{\partial}{\partial a}\mathsf{E} = \left[\frac{\partial}{\partial a}\mathsf{H}\right]\mathsf{E} = \mathsf{E} \qquad \frac{\partial}{\partial b}\mathsf{E} = \left[\frac{\partial}{\partial b}\mathsf{H}\right]\mathsf{E} = (ab^*a)^*b\,\mathsf{E} = \mathsf{F}$$

$$\frac{\partial}{\partial a}\mathsf{F} = \left[\frac{\partial}{\partial a}(ab^*a)^*b\right]\mathsf{E} = b^*a(ab^*a)^*b\,\mathsf{E} = b^*a\,\mathsf{F} = \mathsf{G}$$

$$\frac{\partial}{\partial b}\mathsf{F} = \left[\frac{\partial}{\partial b}(ab^*a)^*b\right]\mathsf{E} = \mathsf{E}$$



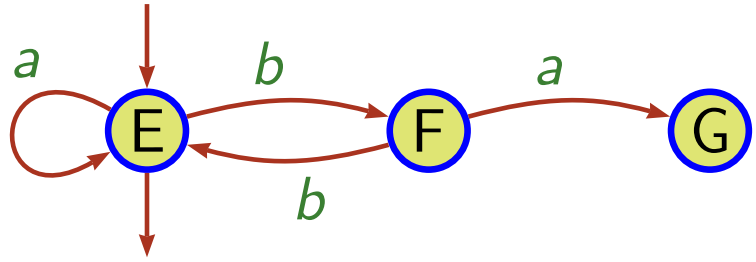
A first example

$$\mathsf{E} = (a + b(ab^*a)^*b)^* = \mathsf{H}^*$$

$$\frac{\partial}{\partial a}\mathsf{E} = \left[\frac{\partial}{\partial a}\mathsf{H}\right]\mathsf{E} = \mathsf{E} \qquad \frac{\partial}{\partial b}\mathsf{E} = \left[\frac{\partial}{\partial b}\mathsf{H}\right]\mathsf{E} = (ab^*a)^*b\,\mathsf{E} = \mathsf{F}$$

$$\frac{\partial}{\partial a}\mathsf{F} = \left[\frac{\partial}{\partial a}(ab^*a)^*b\right]\mathsf{E} = b^*a(ab^*a)^*b\,\mathsf{E} = b^*a\,\mathsf{F} = \mathsf{G}$$

$$\frac{\partial}{\partial b}\mathsf{F} = \left[\frac{\partial}{\partial b}(ab^*a)^*b\right]\mathsf{E} = \mathsf{E}$$



A first example

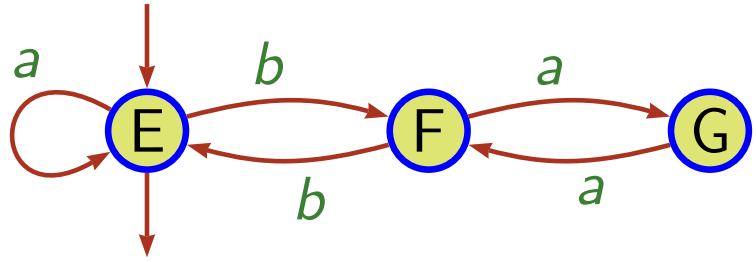
$$\mathsf{E} = (a + b(ab^*a)^*b)^* = \mathsf{H}^*$$

$$\frac{\partial}{\partial a}\mathsf{E} = \left[\frac{\partial}{\partial a}\mathsf{H}\right]\mathsf{E} = \mathsf{E} \qquad \frac{\partial}{\partial b}\mathsf{E} = \left[\frac{\partial}{\partial b}\mathsf{H}\right]\mathsf{E} = (ab^*a)^*b\,\mathsf{E} = \mathsf{F}$$

$$\frac{\partial}{\partial a}\mathsf{F} = \left[\frac{\partial}{\partial a}(ab^*a)^*b\right]\mathsf{E} = b^*a(ab^*a)^*b\,\mathsf{E} = b^*a\mathsf{F} = \mathsf{G}$$

$$\frac{\partial}{\partial b}\mathsf{F} = \left[\frac{\partial}{\partial b}(ab^*a)^*b\right]\mathsf{E} = \mathsf{E}$$

$$\frac{\partial}{\partial a}\mathsf{G} = \left[\frac{\partial}{\partial a}\,b^*a\right]\mathsf{F} = \mathsf{F}b^*a(ab^*a)^*b\,\mathsf{E} = \mathsf{G}$$



A first example

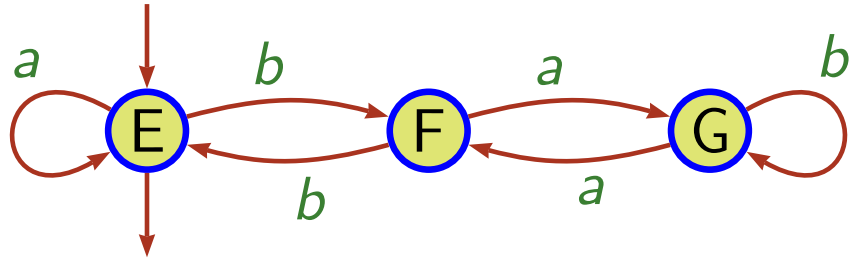
$$\mathsf{E} = (a + b(ab^*a)^*b)^* = \mathsf{H}^*$$

$$\frac{\partial}{\partial a}\mathsf{E} = \left[\frac{\partial}{\partial a}\mathsf{H}\right]\mathsf{E} = \mathsf{E} \qquad \frac{\partial}{\partial b}\mathsf{E} = \left[\frac{\partial}{\partial b}\mathsf{H}\right]\mathsf{E} = (ab^*a)^*b\,\mathsf{E} = \mathsf{F}$$

$$\frac{\partial}{\partial a}\mathsf{F} = \left[\frac{\partial}{\partial a}(ab^*a)^*b\right]\mathsf{E} = b^*a(ab^*a)^*b\,\mathsf{E} = b^*a\,\mathsf{F} = \mathsf{G}$$

$$\frac{\partial}{\partial b}\mathsf{F} = \left[\frac{\partial}{\partial b}(ab^*a)^*b\right]\mathsf{E} = \mathsf{E}$$

$$\frac{\partial}{\partial a}\mathsf{G} = \left[\frac{\partial}{\partial a}\,b^*a\right]\mathsf{F} = \mathsf{F}b^*a(ab^*a)^*b\,\mathsf{E} = \mathsf{G} \qquad \frac{\partial}{\partial b}\mathsf{G} = \left[\frac{\partial}{\partial b}\,b^*a\right]\mathsf{F} = \mathsf{G}$$

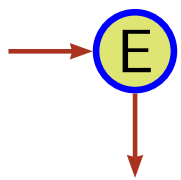


A second example

$$\mathsf{E} = a^* + a^* b(ba^* b)^* ba^* + a^* b(ba^* b)^* a(b + a(ba^* b)^* a)^* a(ba^* b)^* ba^*$$

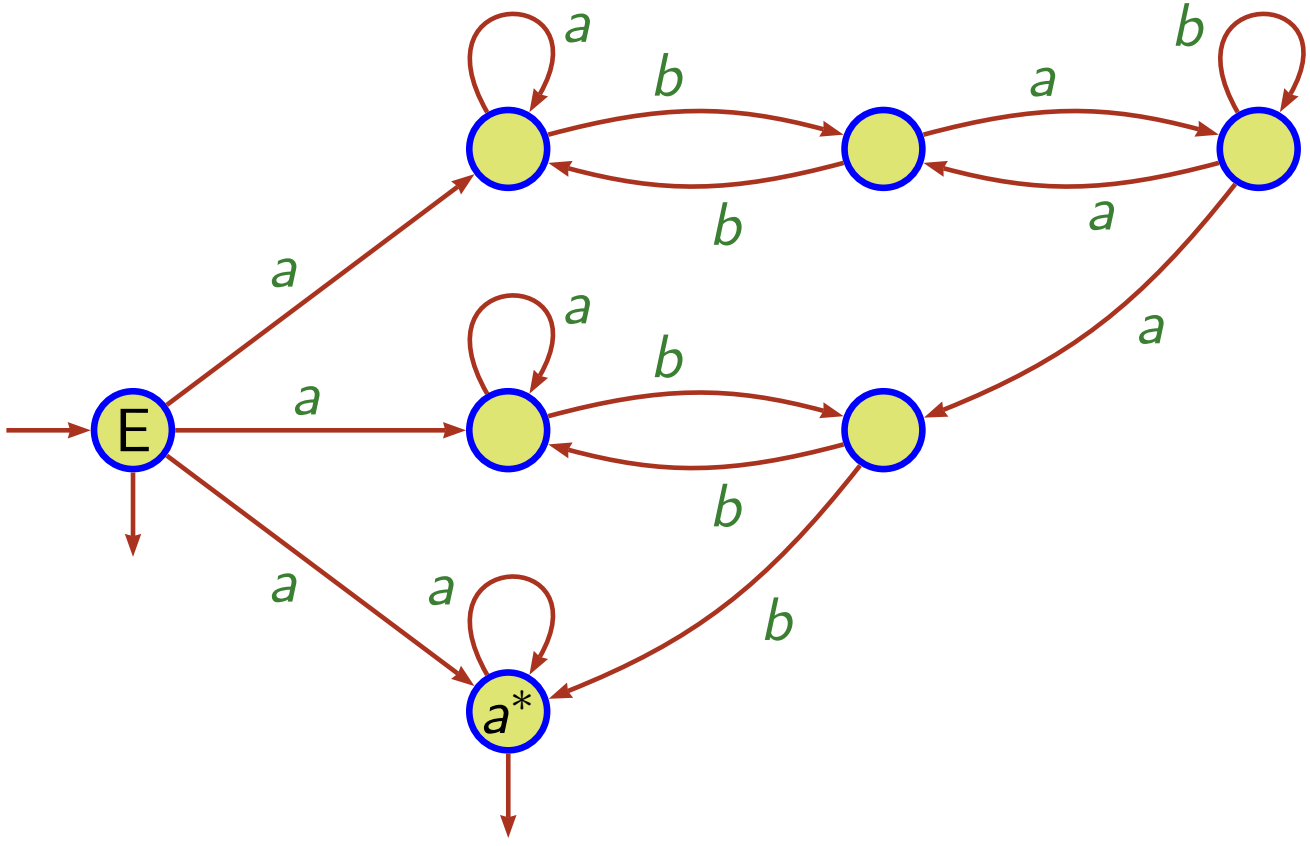
A second example

$$\mathrm{E} = a^* + a^* b(ba^* b)^* ba^* + a^* b(ba^* b)^* a(b + a(ba^* b)^* a)^* a(ba^* b)^* ba^*$$



A second example

$$\mathrm{E} = a^* + a^* b(ba^* b)^* ba^* + a^* b(ba^* b)^* a(b + a(ba^* b)^* a)^* a(ba^* b)^* ba^*$$



Morphisms of automata

Definition

$\varphi\colon \mathcal{A} \to \mathcal{B}$ morphism; $\mathcal{B}$ is a quotient of $\mathcal{A}$
if φ is state surjective and locally Out-surjective .

Morphisms of automata

Definition

$\varphi\colon \mathcal{A} \to \mathcal{B}$ morphism; $\mathcal{B}$ is a co-quotient of $\mathcal{A}$
if φ is state surjective and locally In-surjective .

Morphisms of automata

Definition

$\varphi\colon \mathcal{A} \to \mathcal{B}$ morphism; $\mathcal{B}$ is a co-quotient of $\mathcal{A}$ if φ is state surjective and locally In-surjective .

Proposition

Every automaton $\mathcal{A}$ has a unique minimal quotient $\mathcal{C}$

Morphisms of automata

Definition

$\varphi\colon \mathcal{A} \to \mathcal{B}$ morphism; $\mathcal{B}$ is a co-quotient of $\mathcal{A}$
if φ is state surjective and locally In-surjective .

Proposition

Every automaton $\mathcal{A}$ has a unique minimal co-quotient $\mathcal{D}$

Morphisms of automata

Definition

$\varphi\colon \mathcal{A} \to \mathcal{B}$ morphism; $\mathcal{B}$ is a co-quotient of $\mathcal{A}$
if φ is state surjective and locally In-surjective .

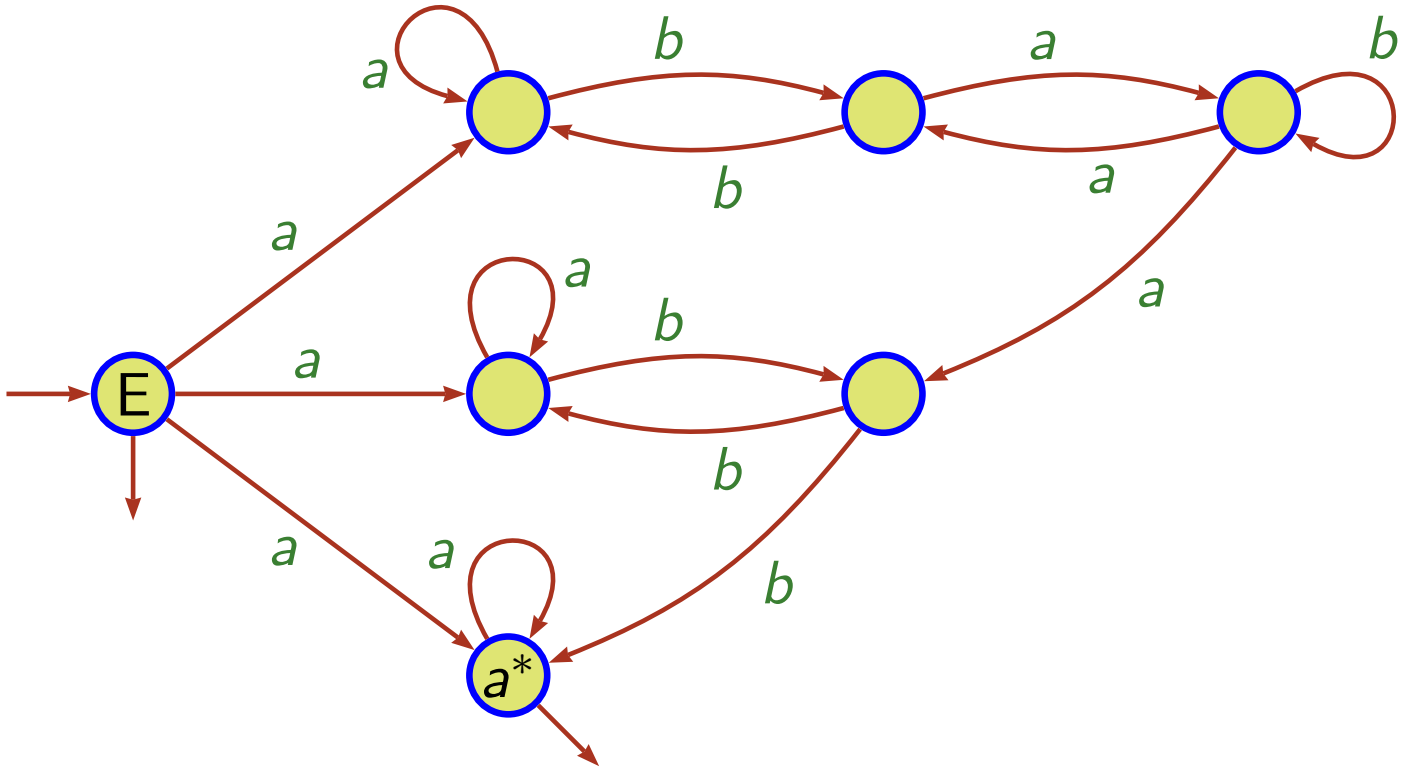
Proposition

Every automaton $\mathcal{A}$ has a unique minimal co-quotient $\mathcal{D}$

$$\mathcal{D} = \Upsilon(\mathcal{A})$$

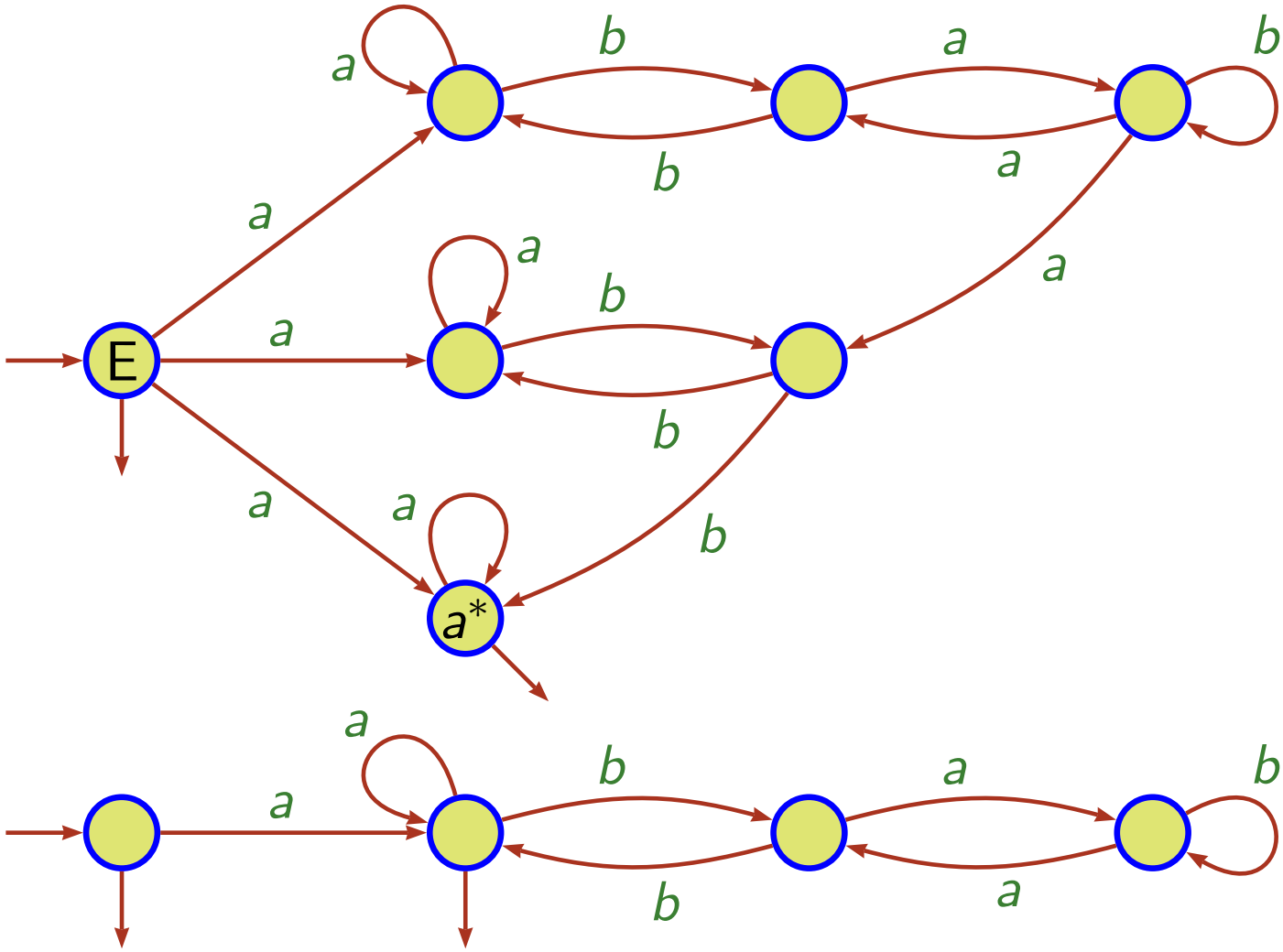
An example of morphism

$$\mathrm{E} = a^* + a^*b(ba^*b)^*ba^* + a^*b(ba^*b)^*a(b + a(ba^*b)^*a)^*a(ba^*b)^*ba^*$$



An example of morphism

$$\mathrm{E} = a^* + a^* b(ba^* b)^* ba^* + a^* b(ba^* b)^* a(b + a(ba^* b)^* a)^* a(ba^* b)^* ba^*$$



The Brzozowski–Antimirov derivation modified

Definition

i) The set of initial derived terms of an expression E is a set $d(\mathsf{E})$ of expressions inductively defined by:

$$d(0) = \{0\}\,, \qquad d(1) = \{1\}\,, \qquad d(a) = \{a\},$$

$$d(\mathsf{E}+\mathsf{F}) = d(\mathsf{E}) \cup d(\mathsf{F})\,, \quad d(\mathsf{E}\cdot\mathsf{F}) = [d(\mathsf{E})]\cdot\mathsf{F}\,, \quad d(\mathsf{E}^*) = \{\mathsf{E}^*\}$$

ii) The set of derived terms of E is redefined as the smallest set that contains the initial derived terms of E and that is closed under derivation.

Using the new derivation

$$\mathsf{E} = a^* + a^* b(ba^* b)^* ba^* + a^* b(ba^* b)^* a(b + a(ba^* b)^* a)^* a(ba^* b)^* ba^*$$

Using the new derivation

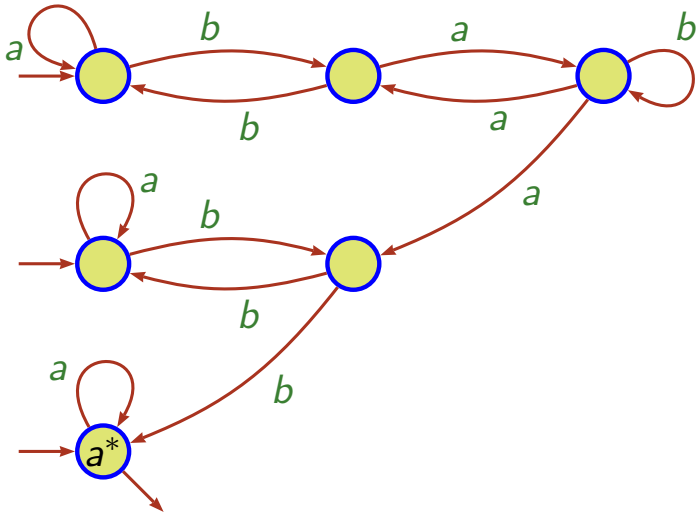
$$\mathsf{E} = a^* + a^* b(ba^* b)^* ba^* + a^* b(ba^* b)^* a(b + a(ba^* b)^* a)^* a(ba^* b)^* ba^*$$

$$d(\mathsf{E}) = \{a^*, a^* b(ba^* b)^* ba^*, a^* b(ba^* b)^* a(b + a(ba^* b)^* a)^* a(ba^* b)^* ba^*\}$$

Using the new derivation

$$\mathrm{E} = a^* + a^*b(ba^*b)^*ba^* + a^*b(ba^*b)^*a(b + a(ba^*b)^*a)^*a(ba^*b)^*ba^*$$

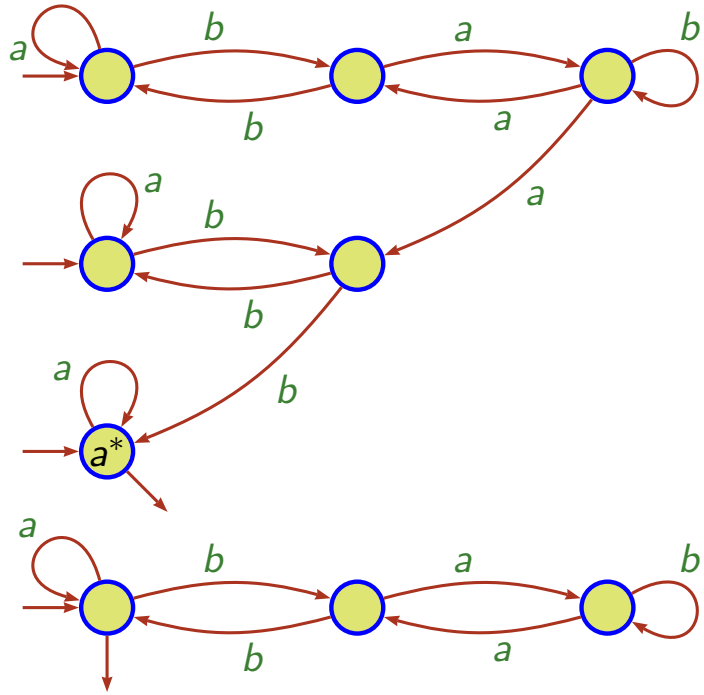
$$d(\mathrm{E}) = \{a^*, a^*b(ba^*b)^*ba^*, a^*b(ba^*b)^*a(b + a(ba^*b)^*a)^*a(ba^*b)^*ba^*\}$$



Using the new derivation

$$\mathsf{E} = a^* + a^*b(ba^*b)^*ba^* + a^*b(ba^*b)^*a(b + a(ba^*b)^*a)^*a(ba^*b)^*ba^*$$

$$d(\mathsf{E}) = \{a^*, a^*b(ba^*b)^*ba^*, a^*b(ba^*b)^*a(b + a(ba^*b)^*a)^*a(ba^*b)^*ba^*\}$$



Theorem (Lombardy–S. 04)

$\mathcal{A}$ *minimal co-deterministic* $\Longrightarrow$ $\mathcal{A} = \Upsilon \circ \Delta \circ \Phi(\mathcal{A})$

Theorem (Lombardy–S. 04 Erratum 05)

$\mathcal{A}$ *minimal co-deterministic* $\Longrightarrow$ $\mathcal{A} = \Upsilon \circ \Delta' \circ \Phi(\mathcal{A})$

Theorem (Lombardy–S. 04 Erratum 05)

$\mathcal{A}$ *minimal co-deterministic* $\Longrightarrow$ $\mathcal{A} = \Upsilon \circ \Delta' \circ \Phi(\mathcal{A})$

Corollary

Λ *partial linearization s.t.* $\Lambda(\mathcal{A})$ *is minimal co-deterministic*

Π *the corresponding delinearization.*

$$\mathcal{A} = \Pi \circ \Upsilon \circ \Delta' \circ \Phi \circ \Lambda(\mathcal{A})$$

Theorem (Lombardy–S. 04 Erratum 05)

$\mathcal{A}$ *minimal co-deterministic* $\Longrightarrow$ $\mathcal{A} = \Upsilon \circ \Delta' \circ \Phi(\mathcal{A})$

Corollary

Λ *partial linearization s.t.* $\Lambda(\mathcal{A})$ *is minimal co-deterministic*

Π *the corresponding delinearization.*

$$\mathcal{A} = \Pi \circ \Upsilon \circ \Delta' \circ \Phi \circ \Lambda(\mathcal{A})$$

$$\mathcal{A} = \Omega \circ \Phi'(\mathcal{A})$$