NL*—Angluin-style learning of NFA

Benedikt Bollig[1] Peter Habermehl[2]
Carsten Kern[3] Martin Leucker[4]

[1]LSV, ENS Cachan

[2]LIAFA Paris Diderot (Paris 7)

[3]RWTH Aachen University

[4] Technical University Munich

Pasadena, July 15th, IJCAI 2009

Learning

Here, learning means:

Given exemplifying behavior of a system

Learn a *model* conforming to the given behavior

Learning

Here, learning means:

Given exemplifying behavior of a system
in terms of words

Learn a *model* conforming to the given behavior
in terms of a regular language (deterministic finite automaton, DFA)

Learning

Here, learning means:

Given exemplifying behavior of a system
in terms of words

Learn a *model* conforming to the given behavior
in terms of a regular language (deterministic finite automaton, DFA)

Active Learning

- The learner is given positive and negative examples

Learning

Here, learning means:

Given exemplifying behavior of a system
in terms of words

Learn a *model* conforming to the given behavior
in terms of a regular language (deterministic finite automaton, DFA)

Active Learning

- The learner is given positive and negative examples
- The learner can actively ask specific questions

RWTH AACHEN UNIVERSITY

Learning

Here, learning means:

Given exemplifying behavior of a system
in terms of words

Learn a *model* conforming to the given behavior
in terms of a regular language (deterministic finite automaton, DFA)

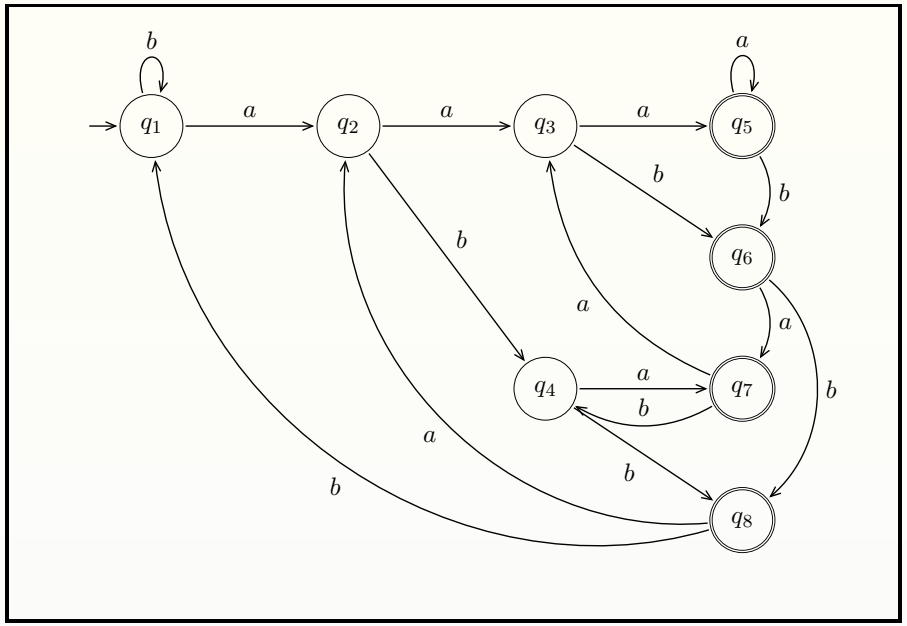
Active Learning

- The learner is given positive and negative examples
- The learner can actively ask specific questions

Occam's razor:

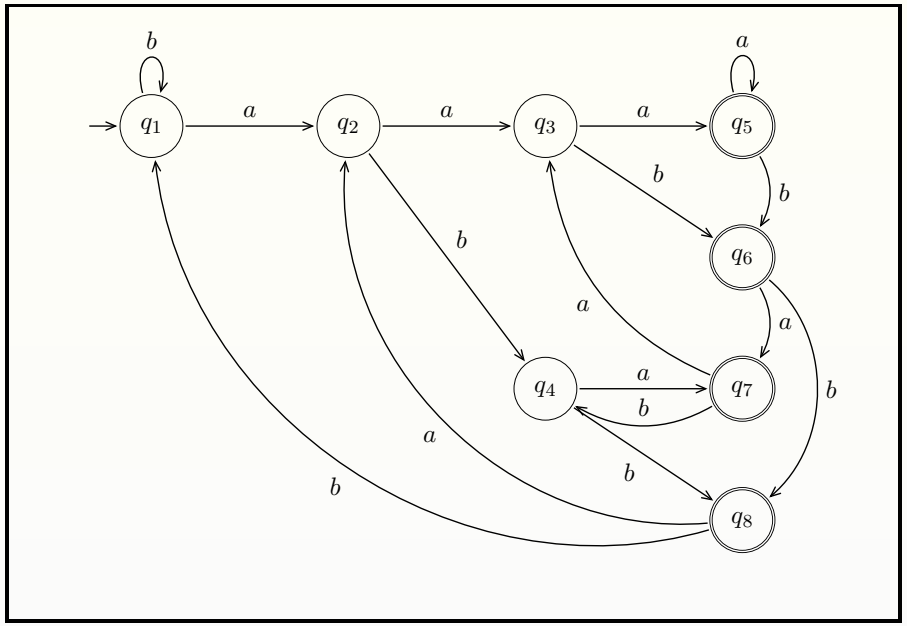
"In case of different explanations, choose the *simplest* one."
$\Rightarrow$ Learn the *minimal* DFA conforming to given examples

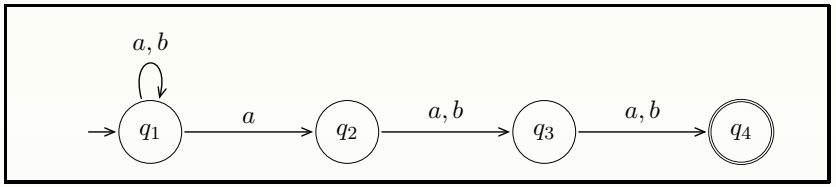
But there is a problem . . .



minimal DFA can be huge!

But there is a problem . . .



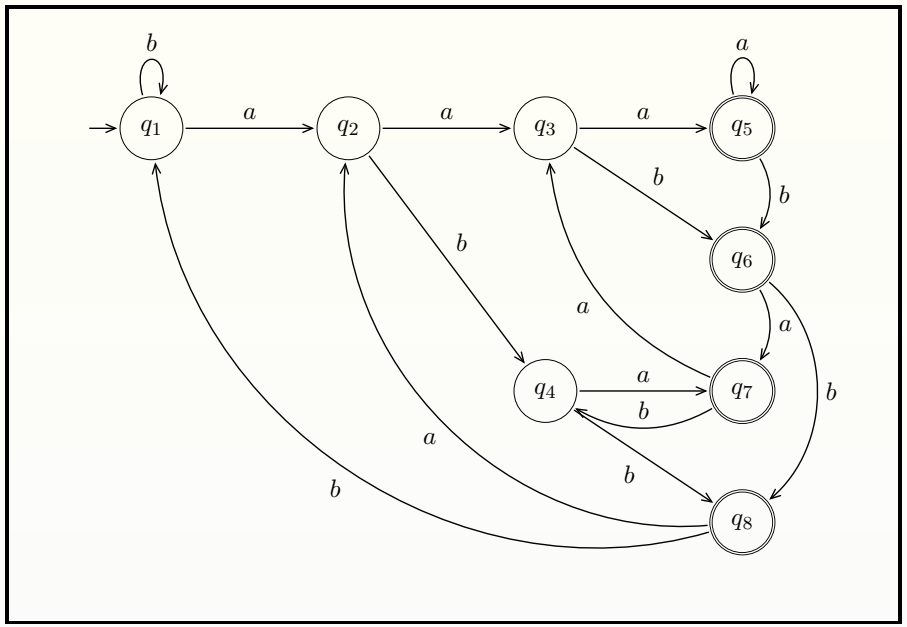


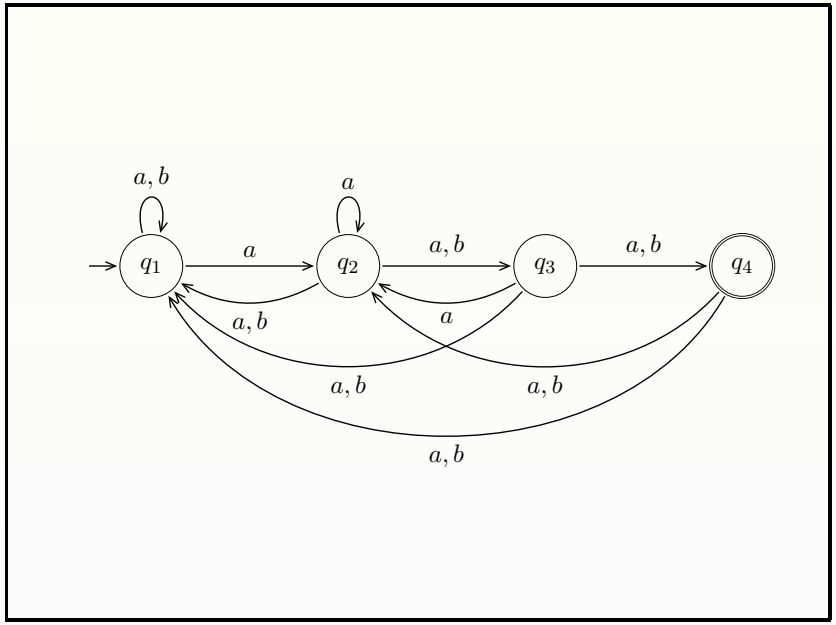
minimal DFA can be huge!

What about NFA?

Can we learn (a certain subclass of) NFA?

But there is a problem . . .





minimal DFA can be huge!

What about NFA?

Can we learn (a certain subclass of) NFA?

Yes, we can!

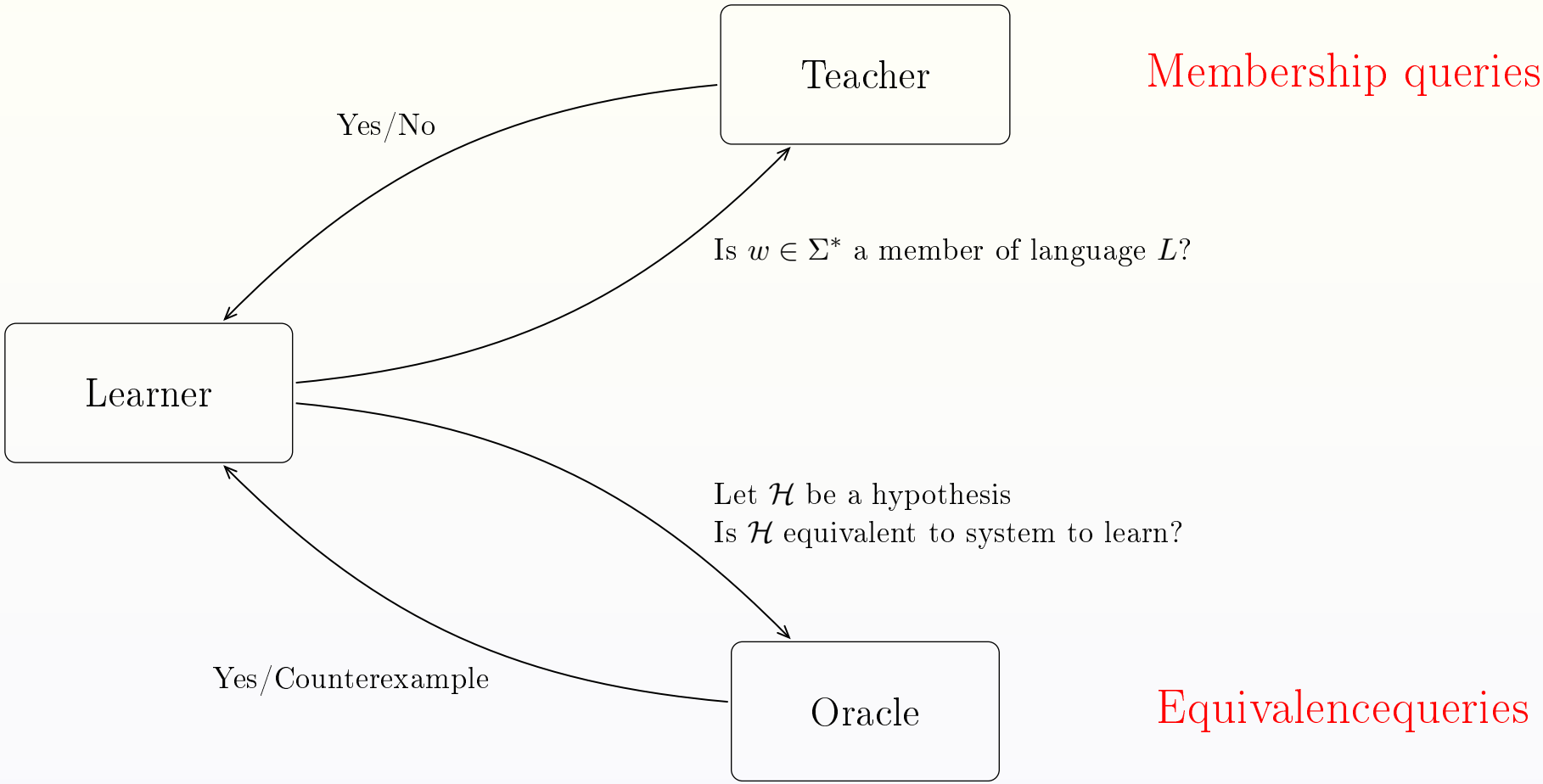
Outline

RWTH AACHEN UNIVERSITY

Presentation outline

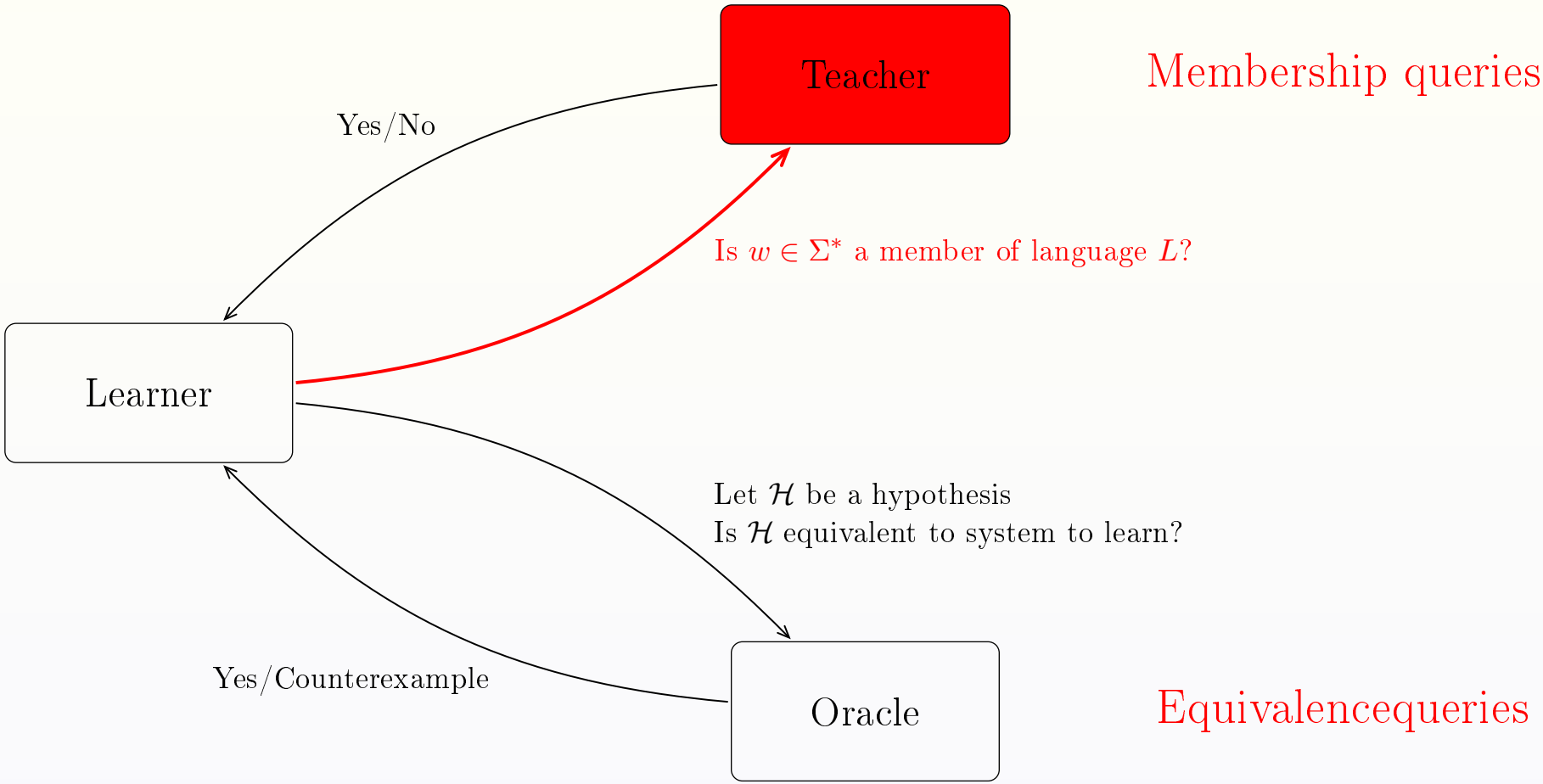
RWTH AACHEN UNIVERSITY

Algorithm - Overview



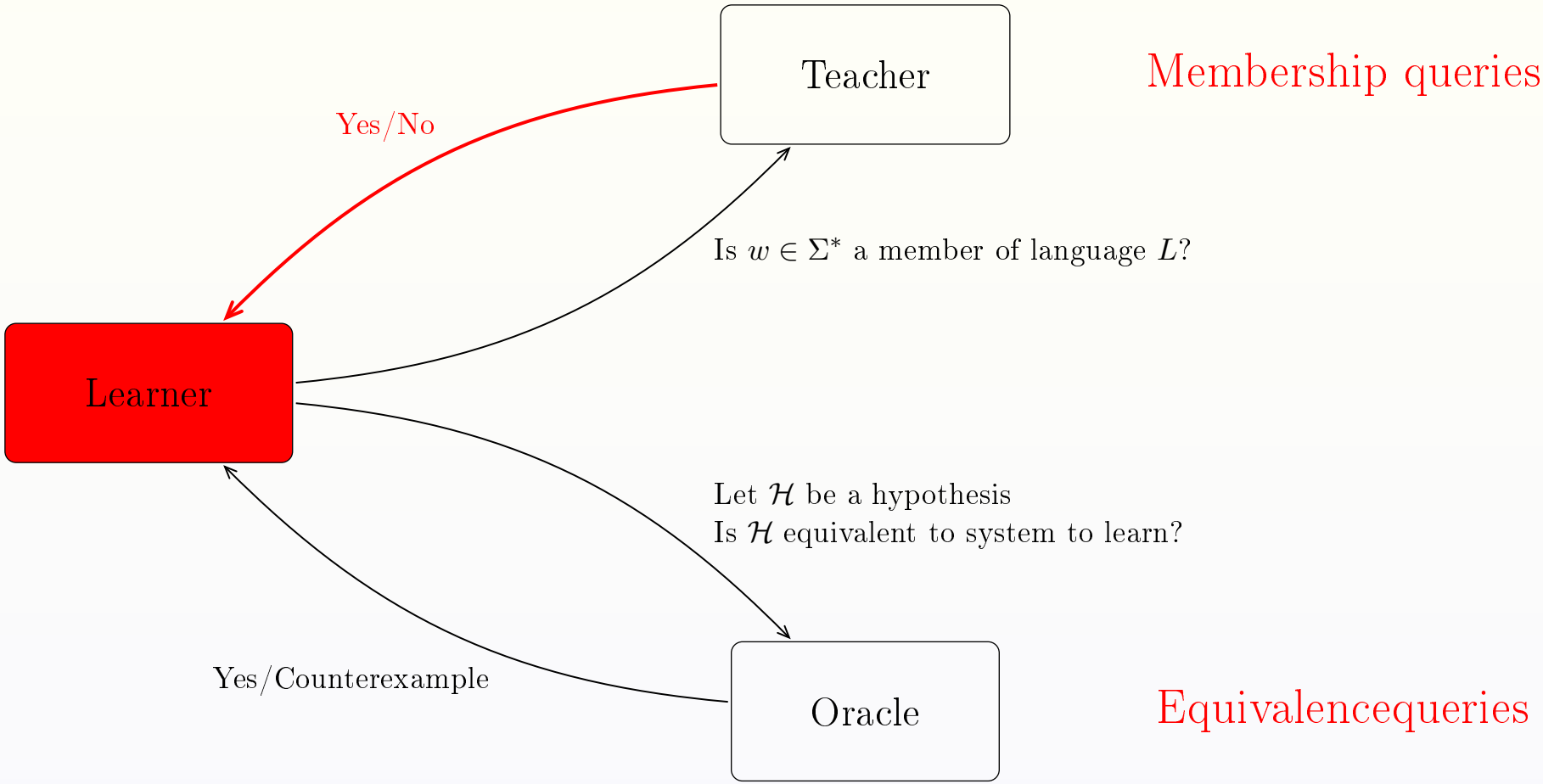
- L: (regular) language to learn
- Counterexample: $w \in (L(\mathcal{H}) \setminus L) \cup (L \setminus L(\mathcal{H}))$

Algorithm - Overview



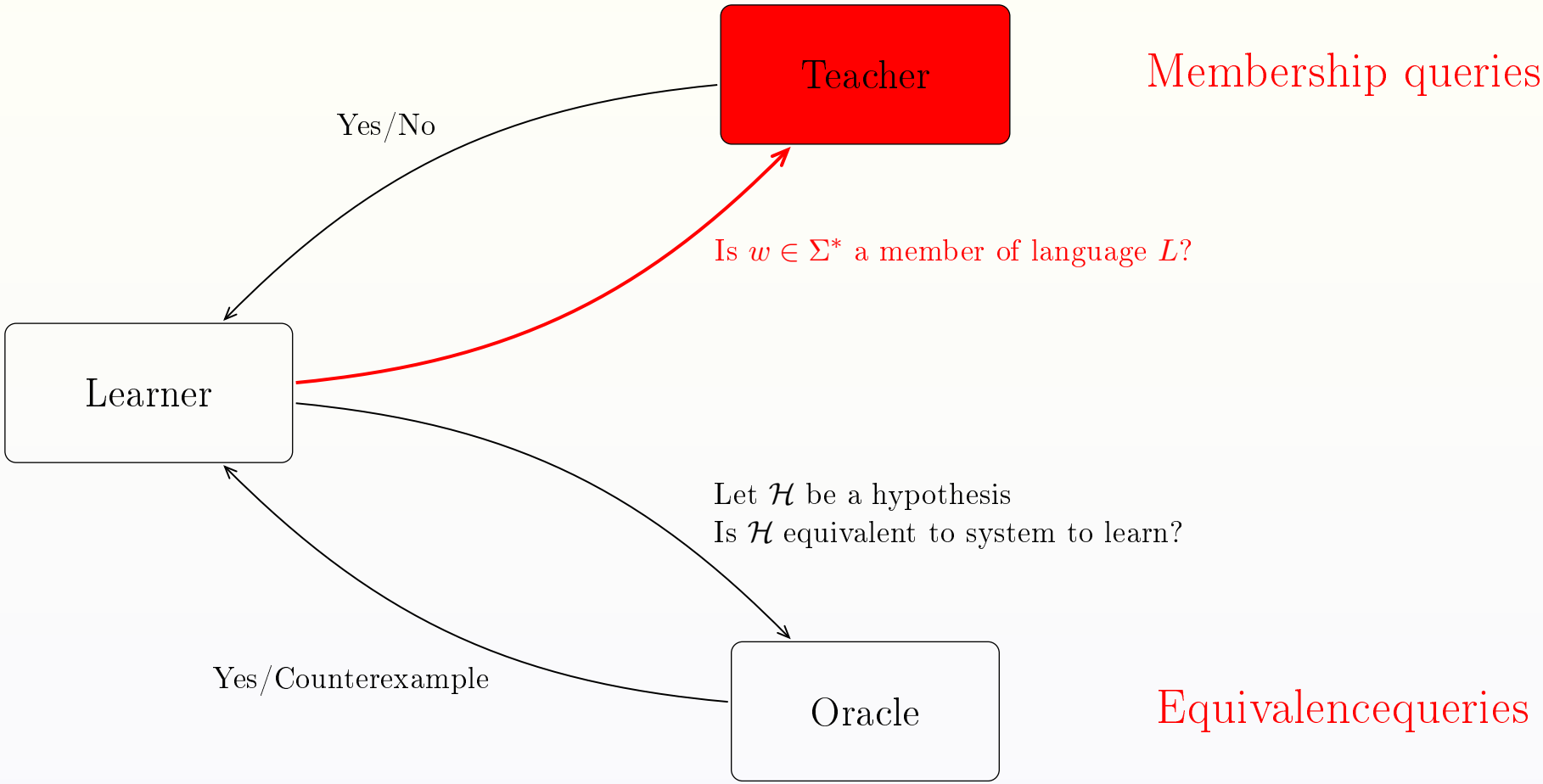
- L: (regular) language to learn
- Counterexample: $w \in (L(\mathcal{H}) \setminus L) \cup (L \setminus L(\mathcal{H}))$

Algorithm - Overview



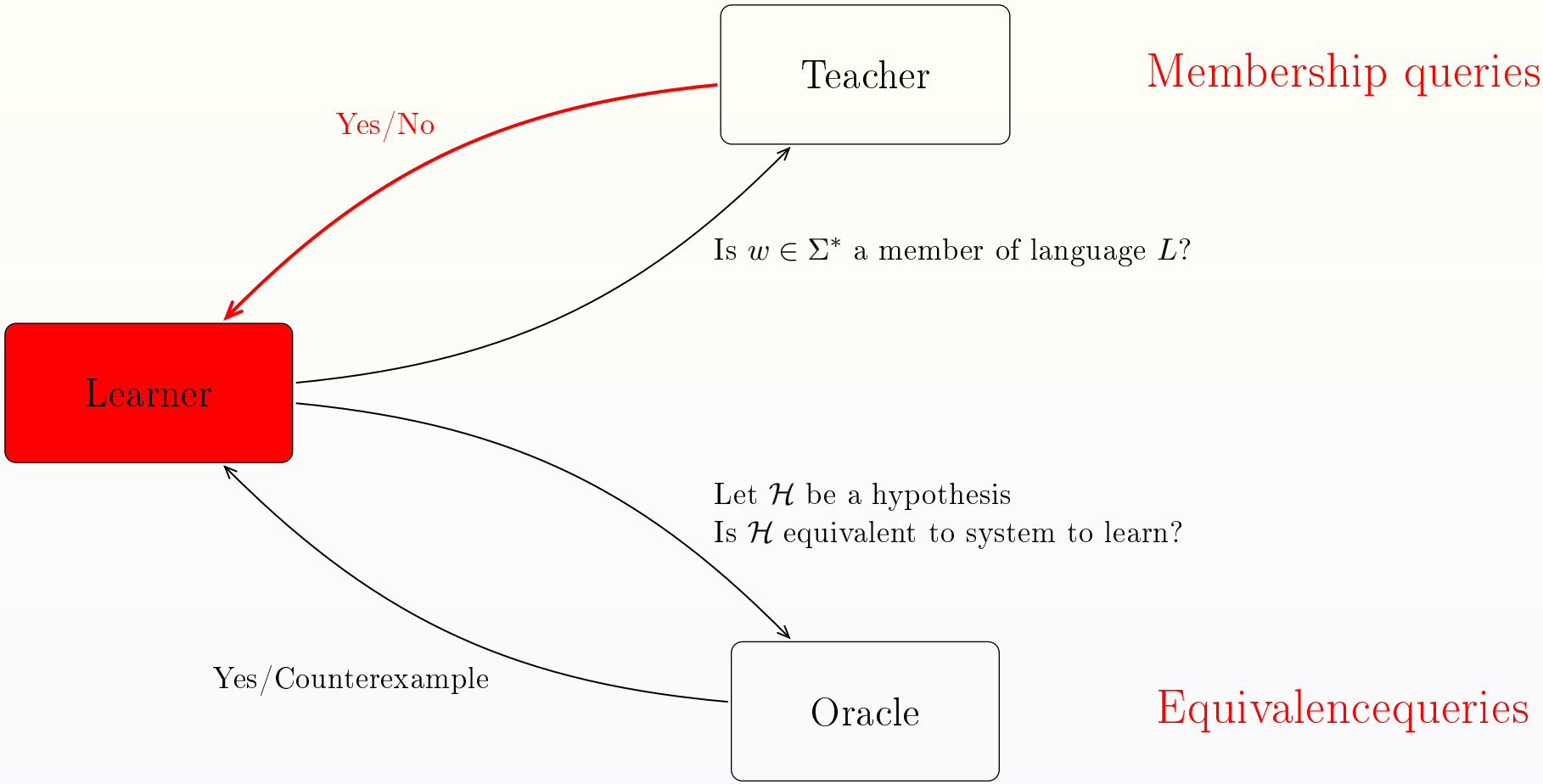
- L: (regular) language to learn
- Counterexample: $w \in (L(\mathcal{H}) \setminus L) \cup (L \setminus L(\mathcal{H}))$

Algorithm - Overview



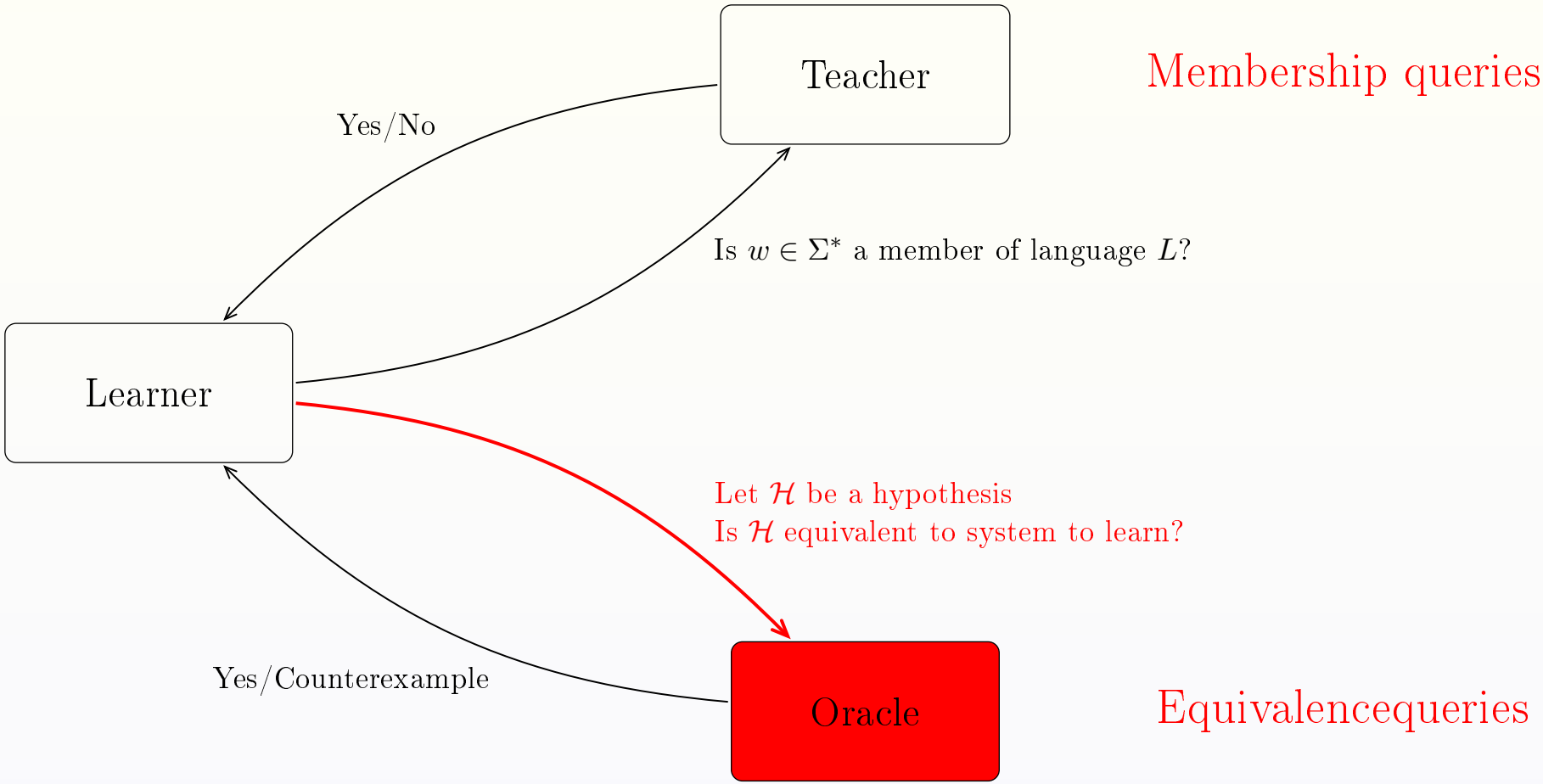
- L: (regular) language to learn
- Counterexample: $w \in (L(\mathcal{H}) \setminus L) \cup (L \setminus L(\mathcal{H}))$

Algorithm - Overview



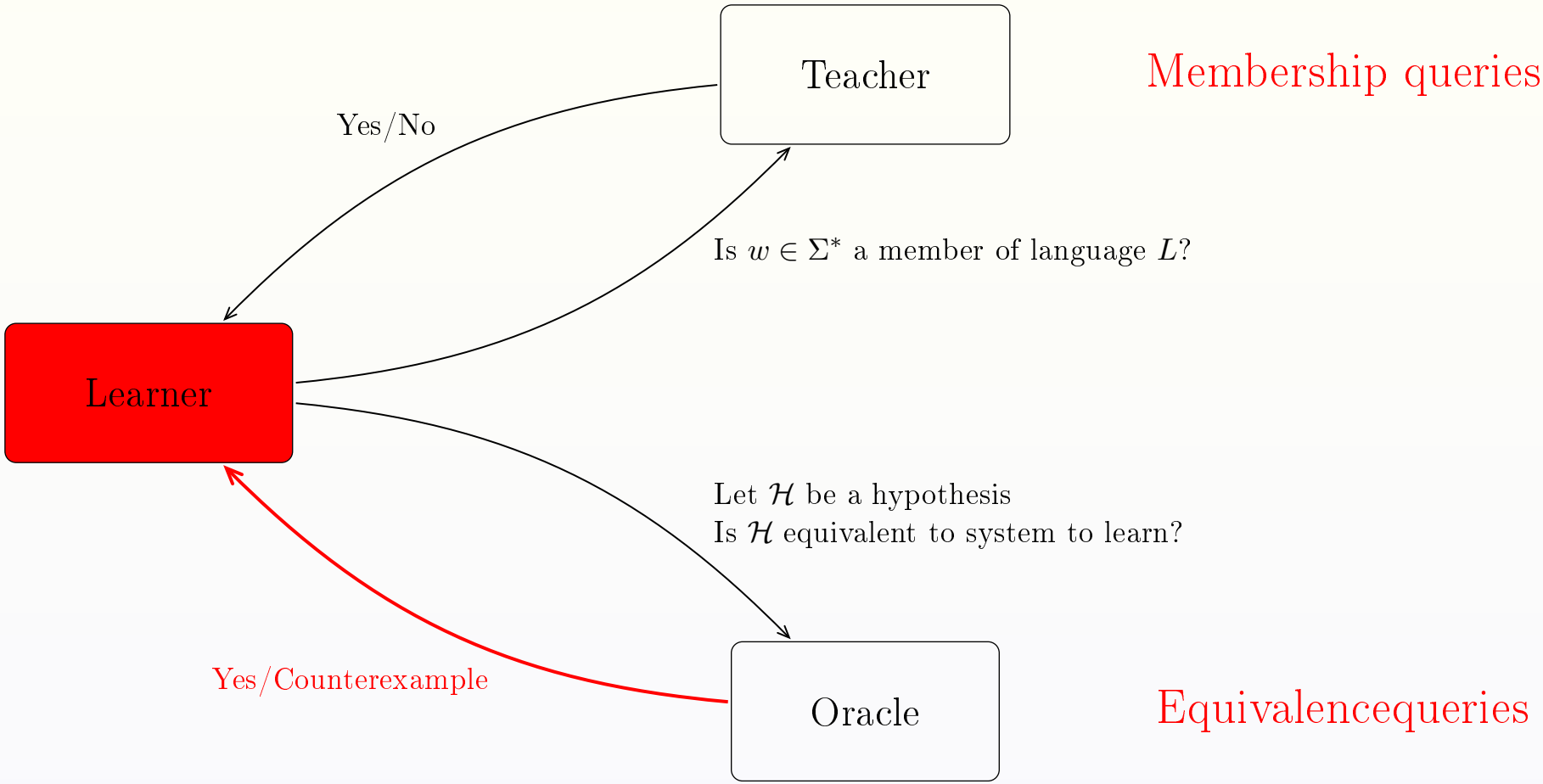
- L: (regular) language to learn
- Counterexample: $w \in (L(\mathcal{H}) \setminus L) \cup (L \setminus L(\mathcal{H}))$

Algorithm - Overview



- L: (regular) language to learn
- Counterexample: $w \in (L(\mathcal{H}) \setminus L) \cup (L \setminus L(\mathcal{H}))$

Algorithm - Overview



- L: (regular) language to learn
- Counterexample: $w \in (L(\mathcal{H}) \setminus L) \cup (L \setminus L(\mathcal{H}))$

Table-based learning

$\mathcal{T}$	ε
ε	
a	
b	
aa	
ab	

$\varepsilon \in L?$

Table-based learning

$\mathcal{T}$	ε
ε	$+$
a	
b	
aa	
ab	

$a \in L?$

Table-based learning

$\mathcal{T}$	ε
ε	$+$
a	$-$
b	
aa	
ab	

$b, aa, ab \in L?$

Table-based learning

$\mathcal{T}$	ε
ε	$+$
a	$-$
b	$-$
aa	$-$
ab	$+$

Table-based learning

$\mathcal{T}$	ε
ε	$+$
a	$-$
b	$-$
aa	$-$
ab	$+$

To derive an automaton:

- $\mathcal{T}$ must be closed, i.e., all states are derivable from $\mathcal{T}$

Table-based learning

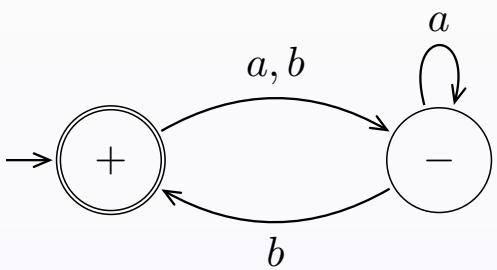
$\mathcal{T}$	ε
ε	$+$
a	$-$
b	$-$
aa	$-$
ab	$+$

To derive an automaton:

- $\mathcal{T}$ must be closed, i.e., all states are derivable from $\mathcal{T}$
- $\mathcal{T}$ must be consistent, i.e., there are no contradicting transitions

Table-based learning

$\mathcal{T}$	ε
ε	$+$
a	$-$
b	$-$
aa	$-$
ab	$+$



To derive an automaton:

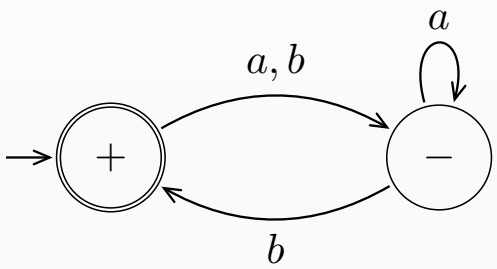
- $\mathcal{T}$ must be closed, i.e., all states are derivable from $\mathcal{T}$
- $\mathcal{T}$ must be consistent, i.e., there are no contradicting transitions

To this end:

- upper rows serve to derive states
- lower rows serve to derive transitions

Table-based learning

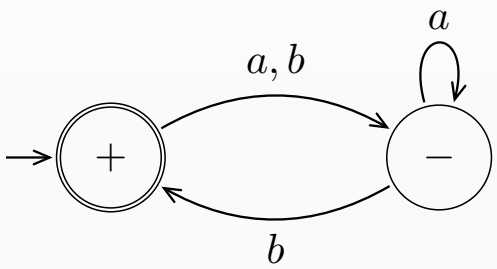
$\mathcal{T}$	ε
ε	$+$
a	$-$
b	$-$
aa	$-$
ab	$+$



$bb \notin L$!

Table-based learning

$\mathcal{T}$	ε
ε	$+$
a	$-$
b	$-$
aa	$-$
ab	$+$



$bb \notin L$!

Counterexample can be added to:

Table-based learning

$\mathcal{T}$	ε
ε	$+$
a	$-$
b	$-$
bb	$-$
aa	$-$
ab	$+$
ba	$-$
bba	$-$
bbb	$-$

Counterexample can be added to:

- the rows (L^*)

Table-based learning

$\mathcal{T}$	ε	bb	b
ε	$+$	$-$	$-$
a	$-$	$-$	$+$
b	$-$	$-$	$-$
aa	$-$	$-$	$-$
ab	$+$	$-$	$-$

Counterexample can be added to:

- the rows (L^*)
- the columns ($\mathrm{L}^*_{\mathrm{col}}$)

Angluin's Algorithm L*

Theorem (Complexity of L*)

Let:

- n: number of states of the minimal DFA $\mathcal{A}_L$ for regular language L,
- m: length of the biggest counterexample

Then, L* returns after at most:

the minimal DFA $\mathcal{A}$.

Angluin's Algorithm L*

Theorem (Complexity of L*)

Let:

- n: number of states of the minimal DFA $\mathcal{A}_L$ for regular language L,
- m: length of the biggest counterexample

Then, L* returns after at most:

- n equivalence queries and

the minimal DFA $\mathcal{A}$.

Angluin's Algorithm L*

Theorem (Complexity of L*)

Let:

- n: number of states of the minimal DFA $\mathcal{A}_L$ for regular language L,
- m: length of the biggest counterexample

Then, L* returns after at most:

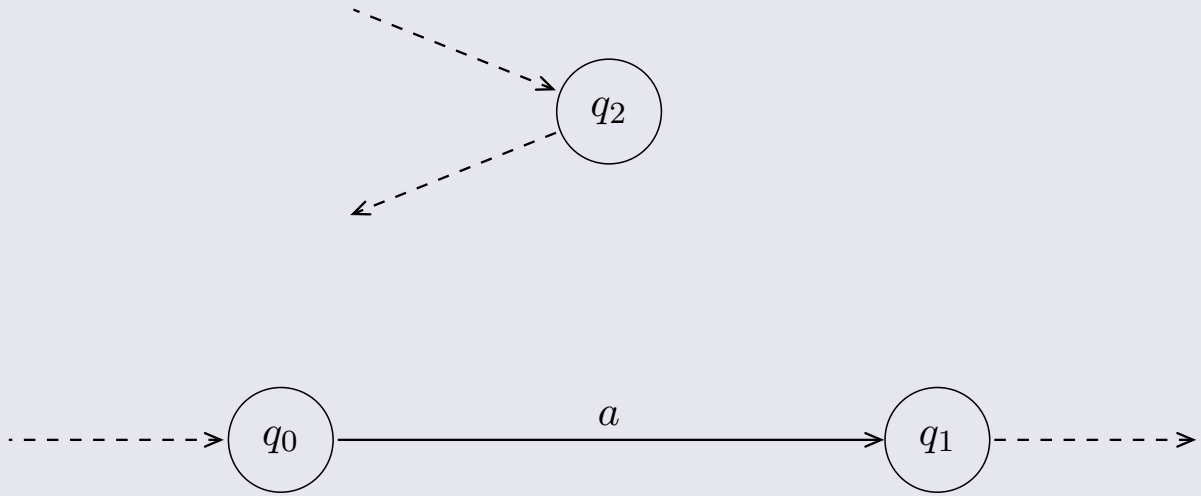
- n equivalence queries and
- $O(m|\Sigma|n^2)$ membership queries

the minimal DFA $\mathcal{A}$.

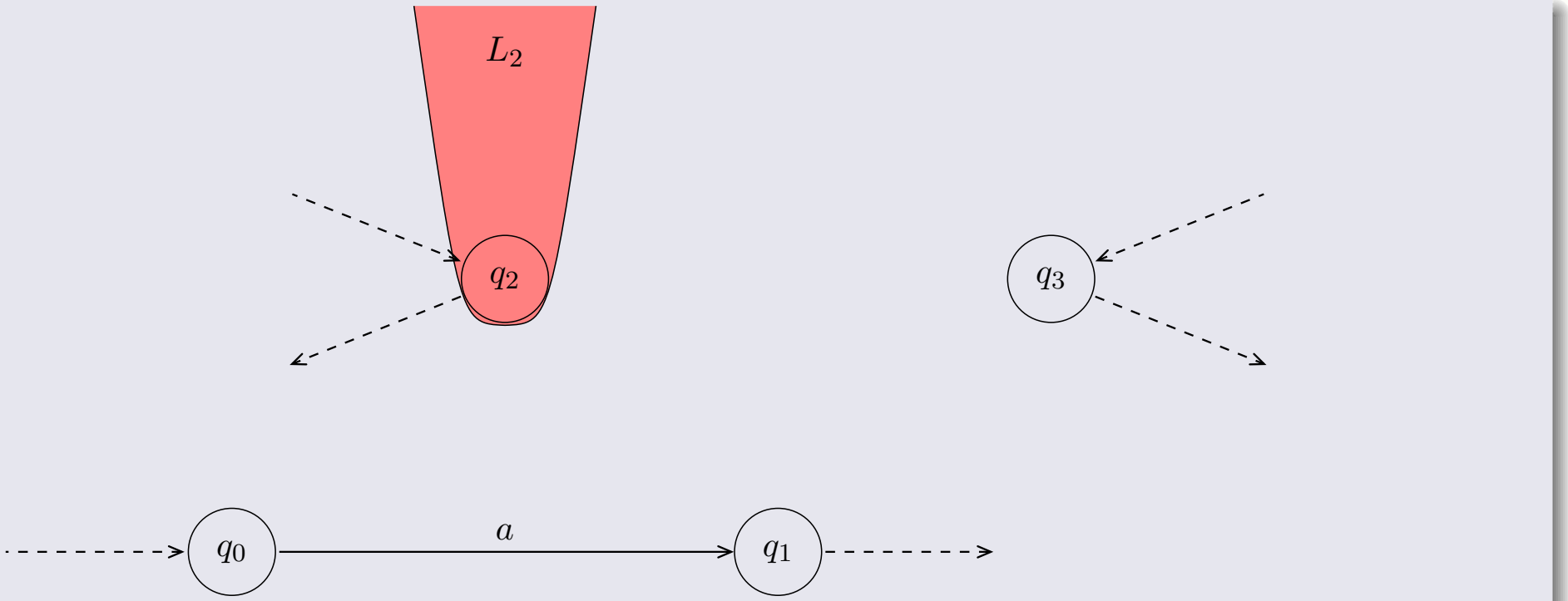
Presentation outline

RWTH AACHEN UNIVERSITY

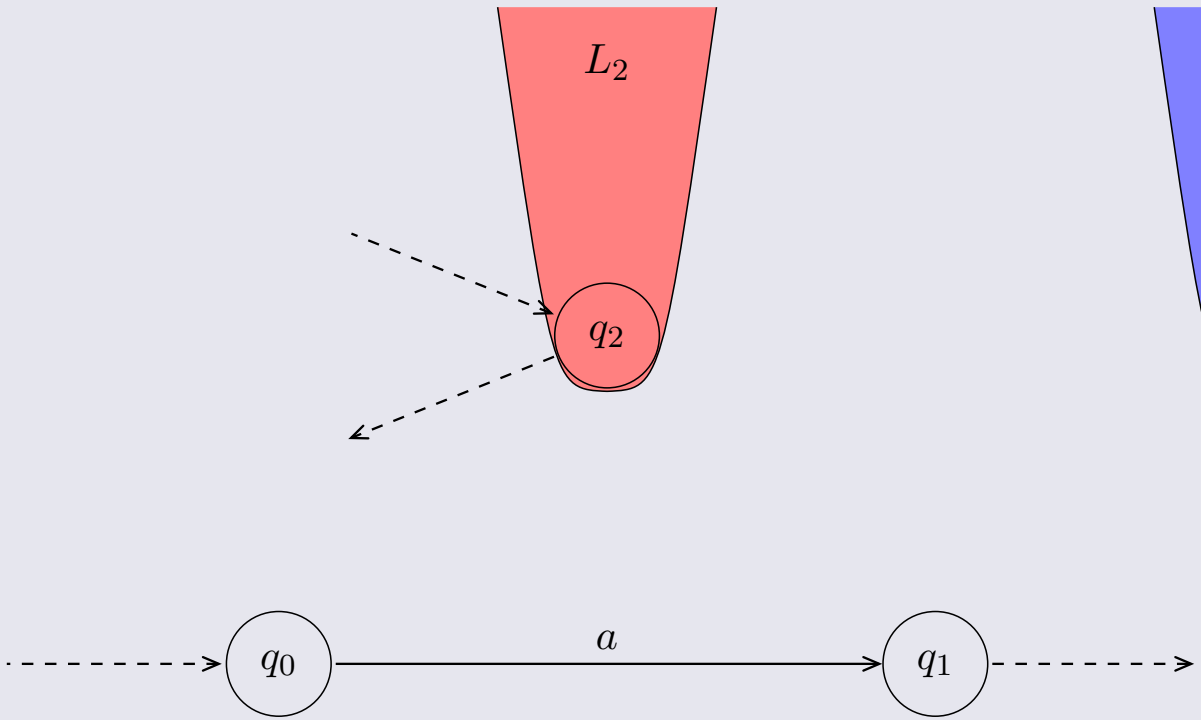
Residual Finite-State Automata [Denis et al.]



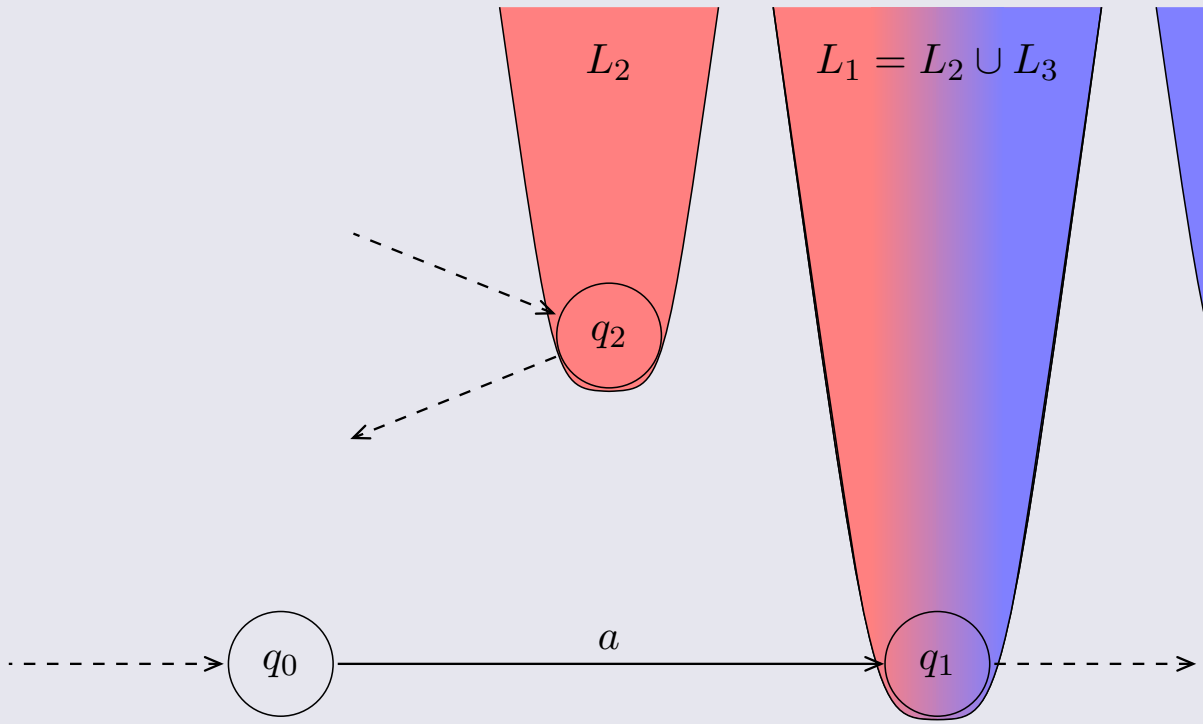
Residual Finite-State Automata [Denis et al.]



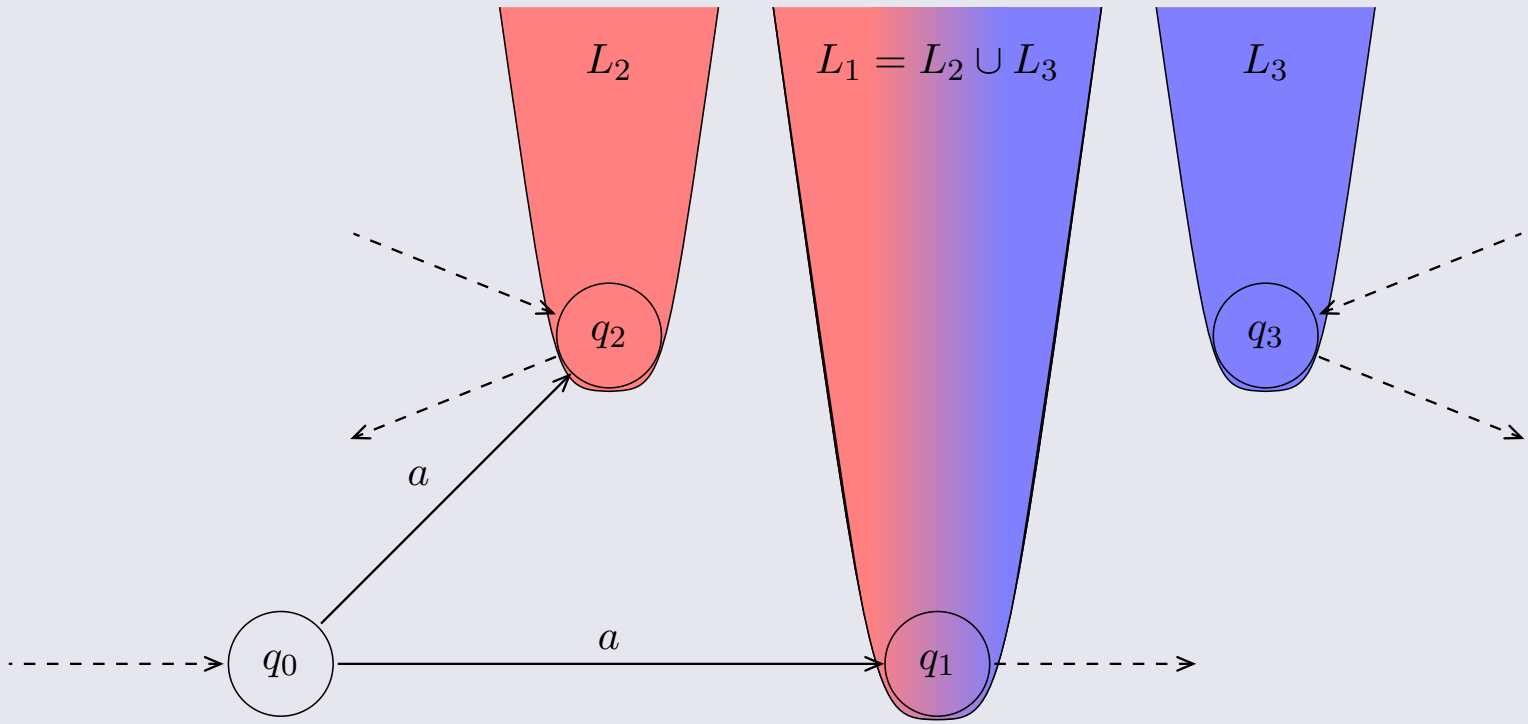
Residual Finite-State Automata [Denis et al.]



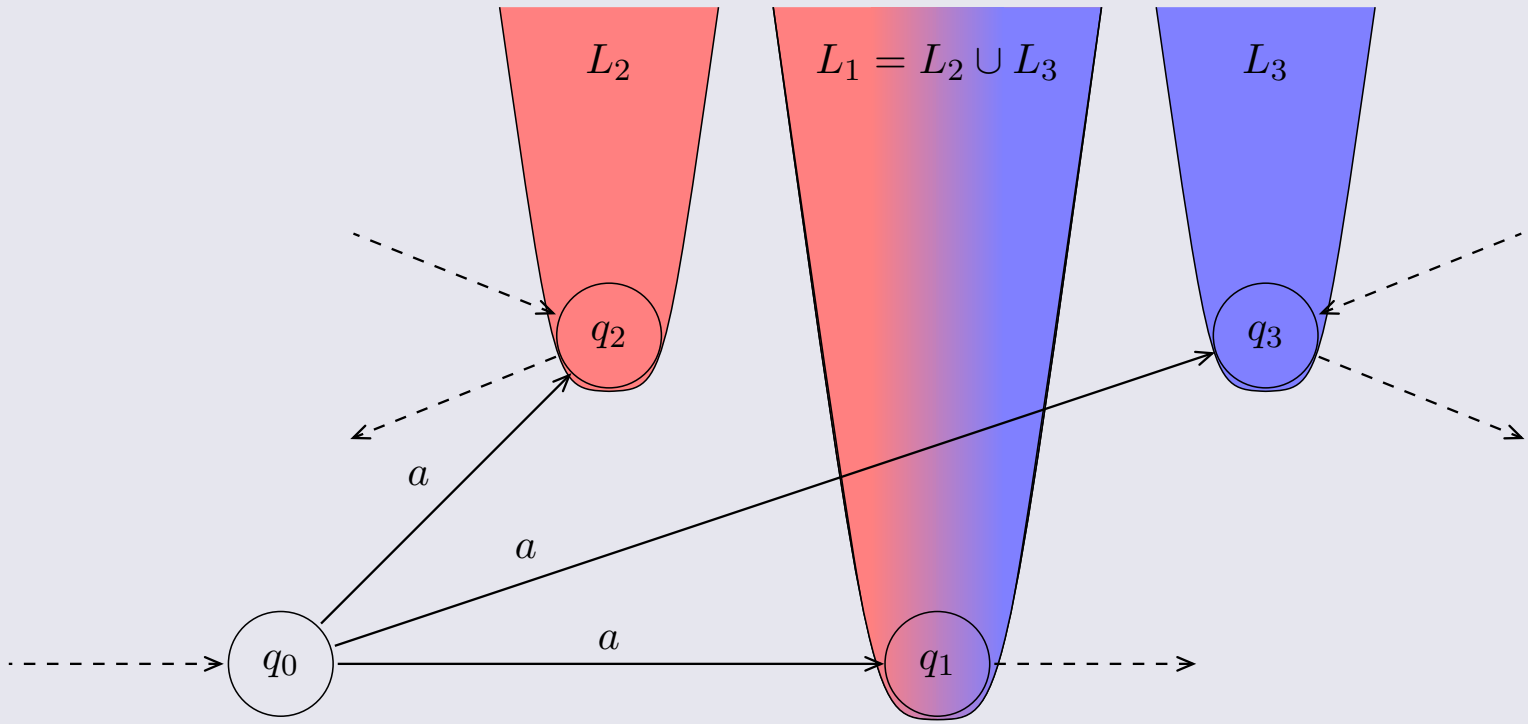
Residual Finite-State Automata [Denis et al.]



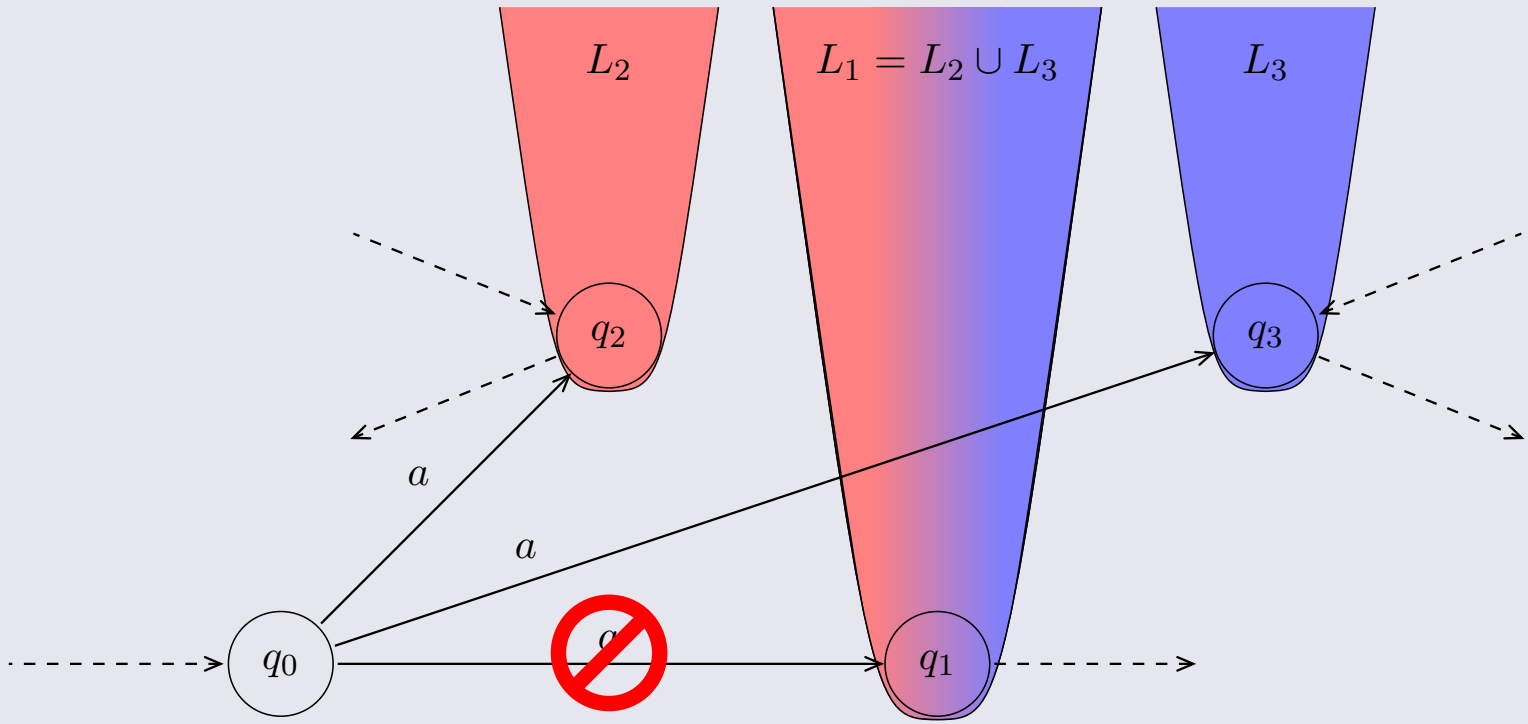
Residual Finite-State Automata [Denis et al.]



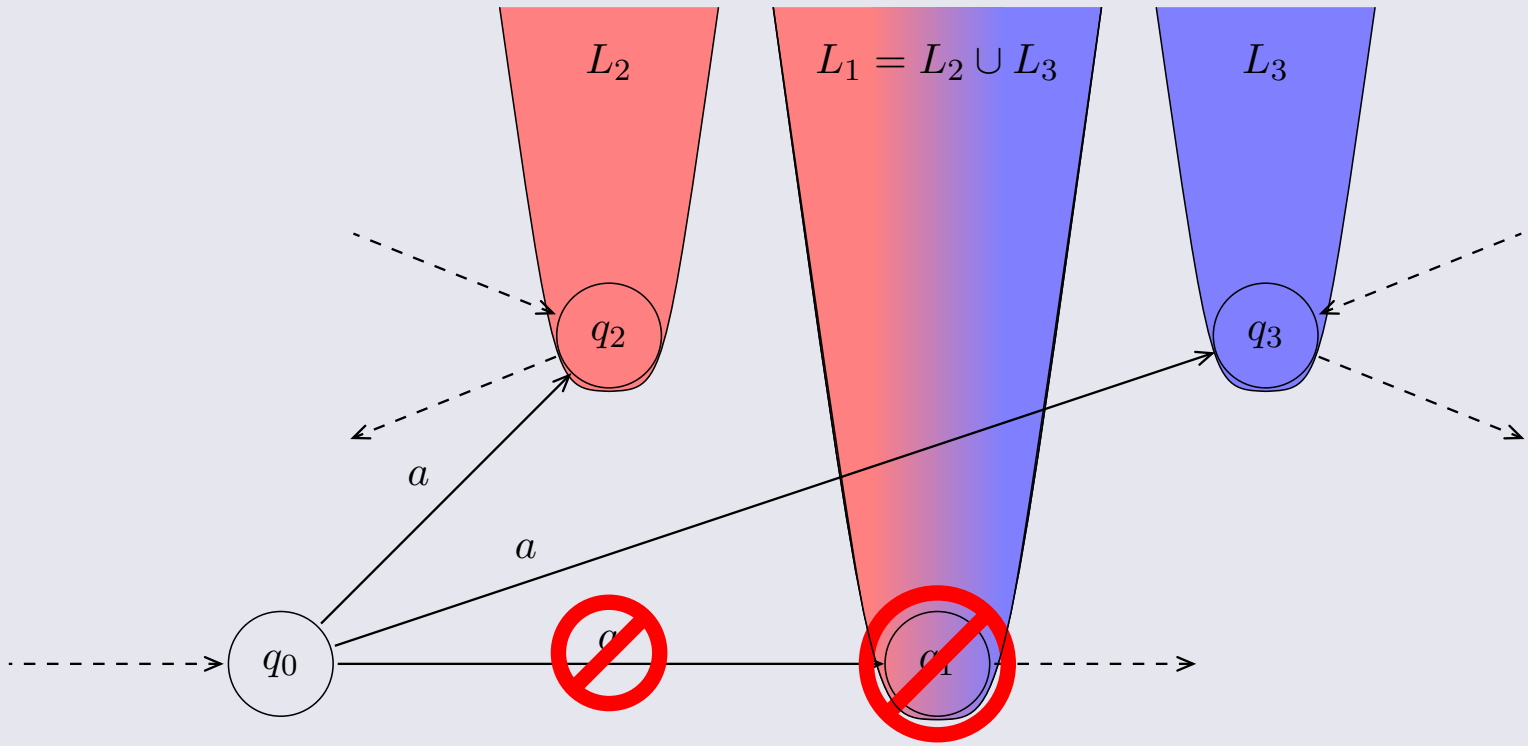
Residual Finite-State Automata [Denis et al.]



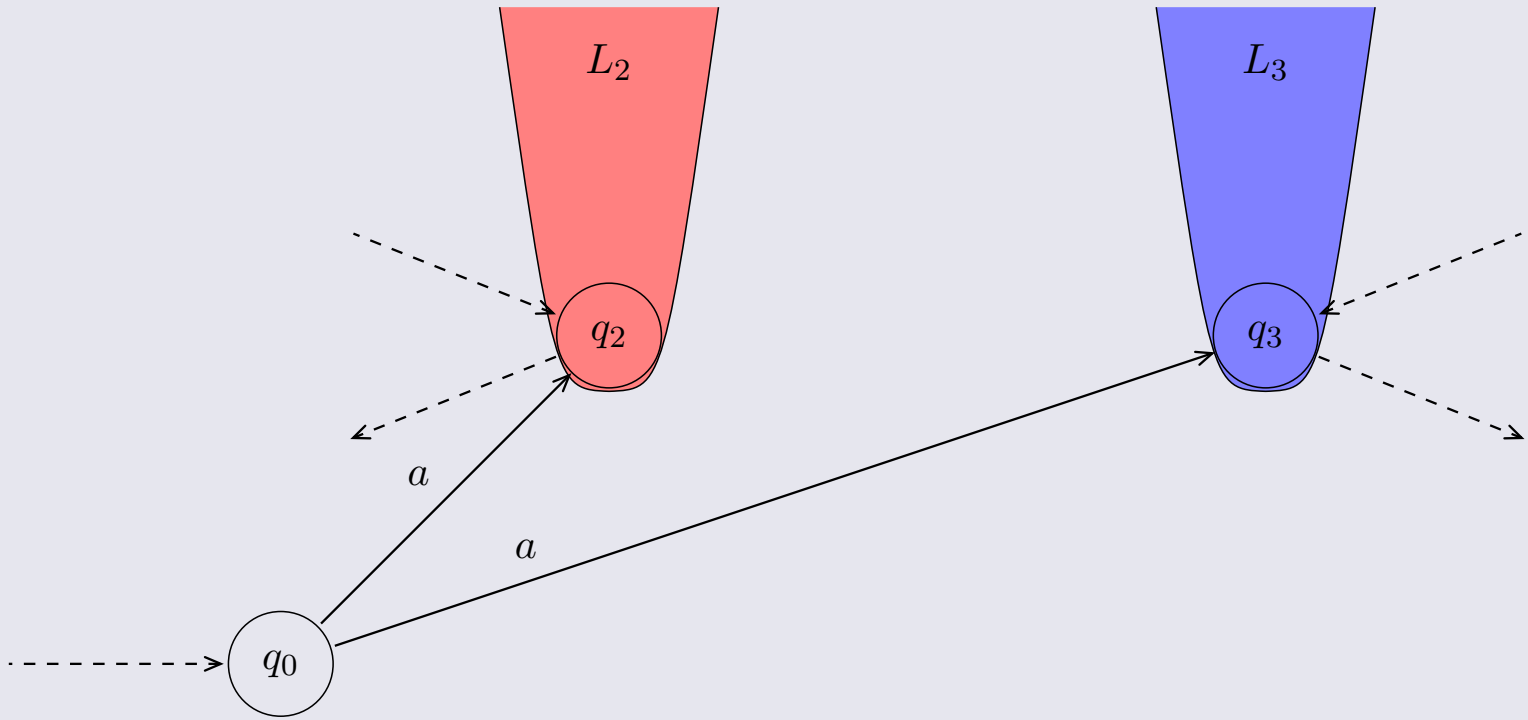
Residual Finite-State Automata [Denis et al.]



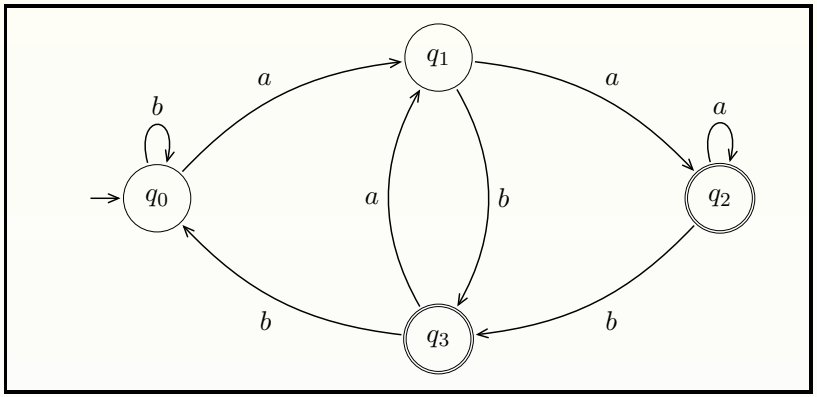
Residual Finite-State Automata [Denis et al.]



Residual Finite-State Automata [Denis et al.]



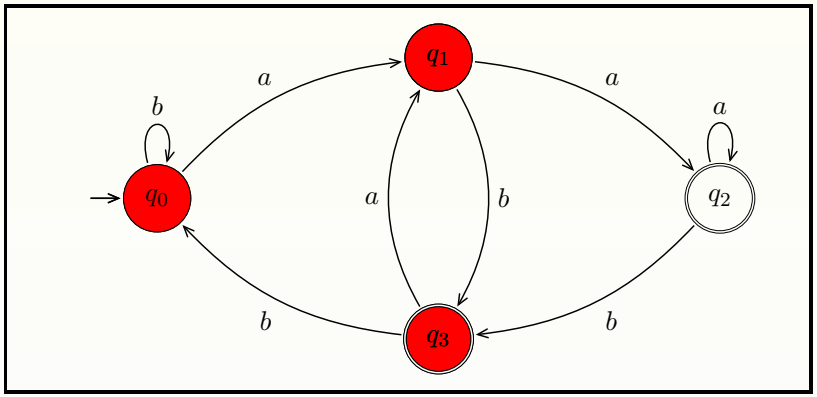
Example



Residual languages for $L = \Sigma^* a \Sigma$

$$L_{q_0} = \Sigma^* a \Sigma$$
$$L_{q_1} = \Sigma^* a \Sigma \cup \Sigma$$
$$L_{q_2} = \Sigma^* a \Sigma \cup \Sigma \cup \{\varepsilon\}$$
$$L_{q_3} = \Sigma^* a \Sigma \cup \{\varepsilon\}$$

Example



Residual languages for $L = \Sigma^* a \Sigma$

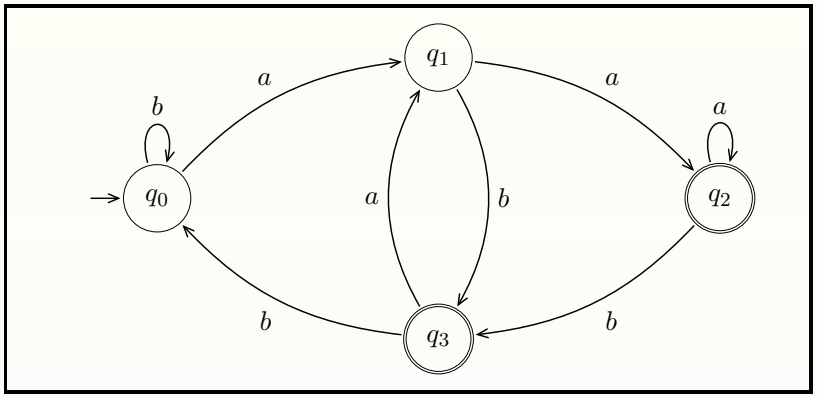
$$L_{q_0} = \Sigma^* a \Sigma$$

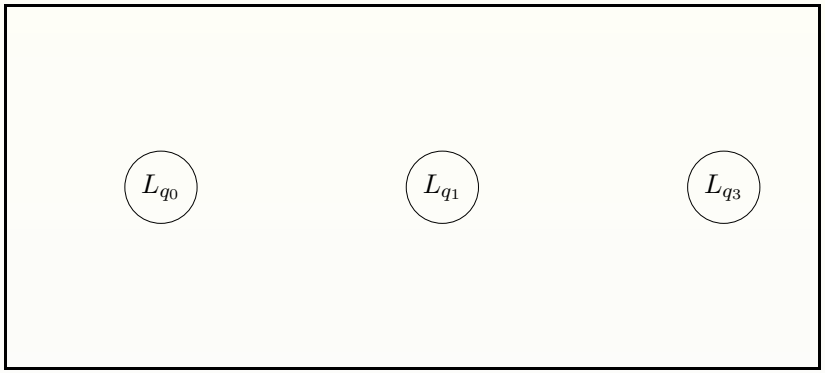
$$L_{q_1} = \Sigma^* a \Sigma \cup \Sigma$$

$$L_{q_2} = \Sigma^* a \Sigma \cup \Sigma \cup \{\varepsilon\}$$

$$L_{q_3} = \Sigma^* a \Sigma \cup \{\varepsilon\}$$

Example

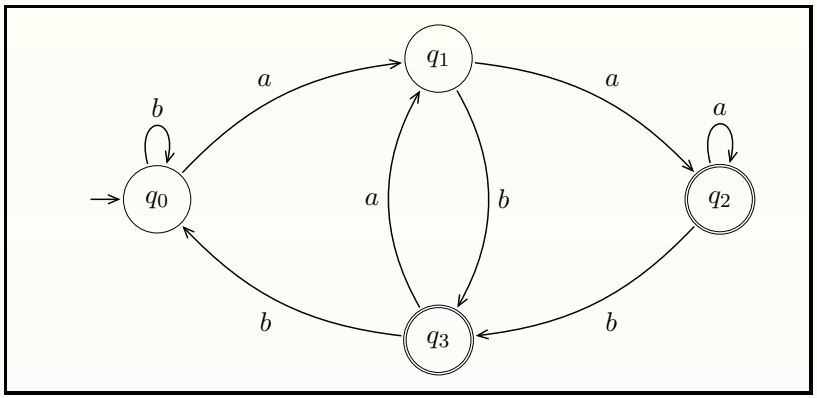


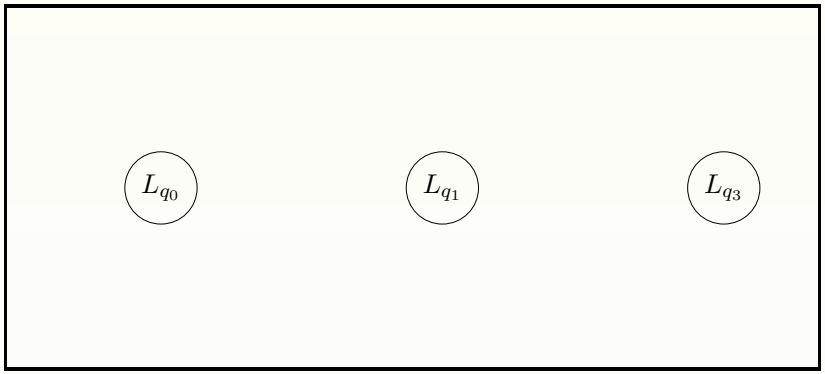


Residual languages for $L = \Sigma^* a \Sigma$

$$L_{q_0} = \Sigma^* a \Sigma$$
$$L_{q_1} = \Sigma^* a \Sigma \cup \Sigma$$
$$L_{q_2} = \Sigma^* a \Sigma \cup \Sigma \cup \{\varepsilon\}$$
$$L_{q_3} = \Sigma^* a \Sigma \cup \{\varepsilon\}$$

Example





Residual languages for $L = \Sigma^* a \Sigma$

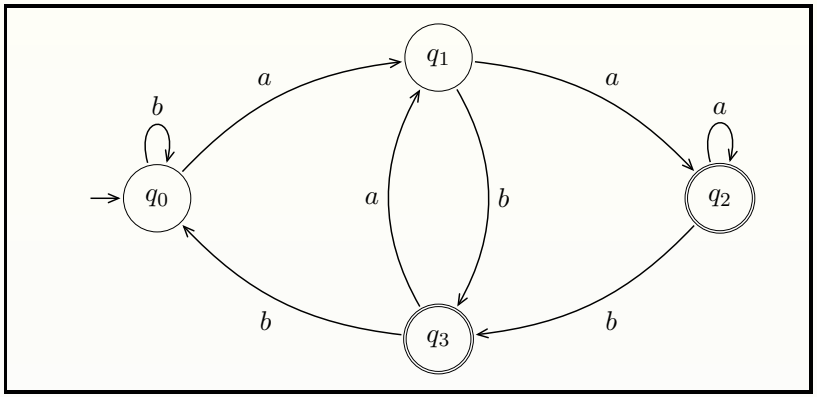
$$L_{q_0} = \Sigma^* a \Sigma \qquad \text{(initial state)}$$

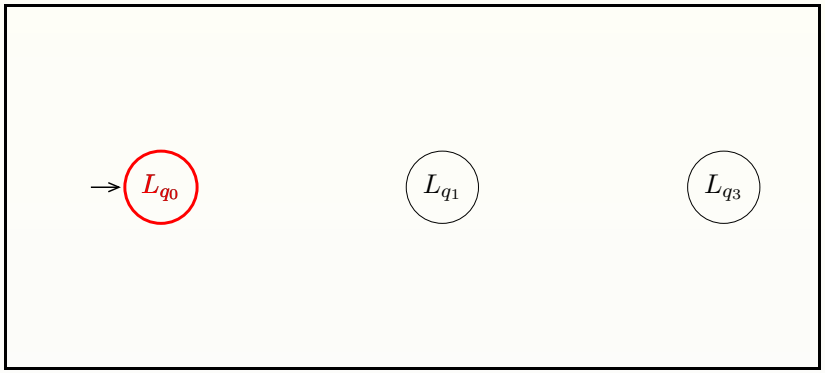
$$L_{q_1} = \Sigma^* a \Sigma \cup \Sigma$$

$$L_{q_2} = \Sigma^* a \Sigma \cup \Sigma \cup \{\varepsilon\}$$

$$L_{q_3} = \Sigma^* a \Sigma \cup \{\varepsilon\}$$

Example

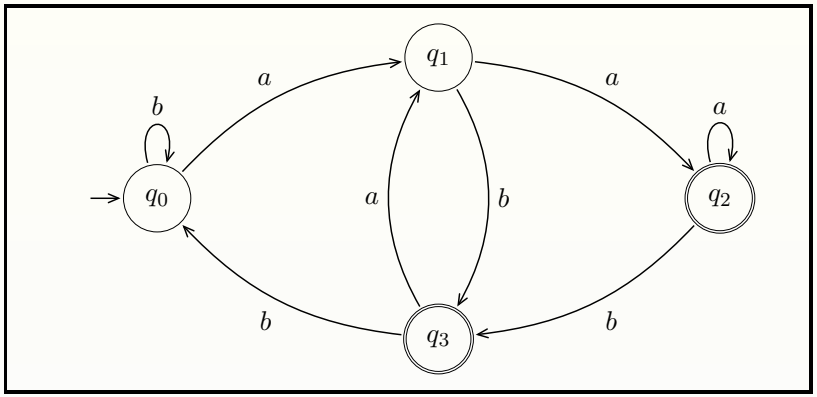


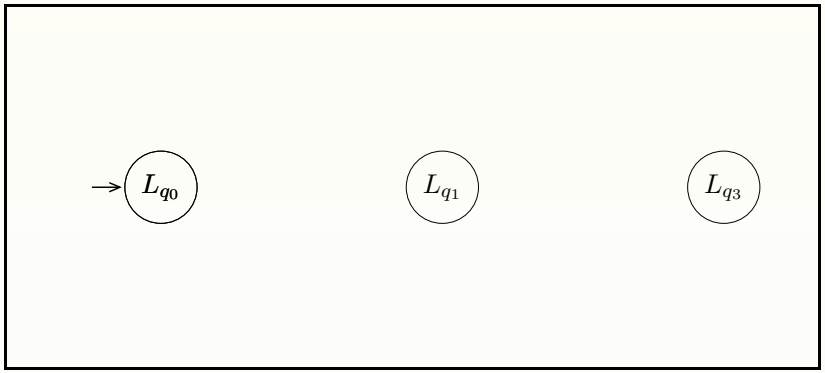


Residual languages for $L = \Sigma^* a \Sigma$

$$L_{q_0} = \Sigma^* a \Sigma$$
$$L_{q_1} = \Sigma^* a \Sigma \cup \Sigma$$
$$L_{q_2} = \Sigma^* a \Sigma \cup \Sigma \cup \{\varepsilon\}$$
$$L_{q_3} = \Sigma^* a \Sigma \cup \{\varepsilon\}$$

Example





Residual languages for $L = \Sigma^* a \Sigma$

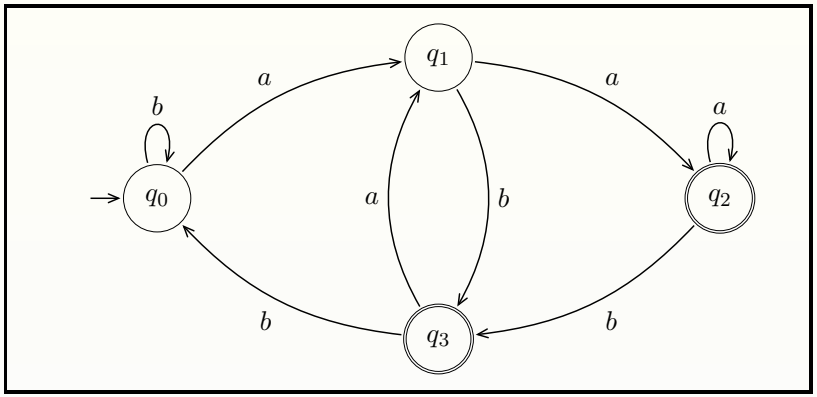
$$L_{q_0} = \Sigma^* a \Sigma$$

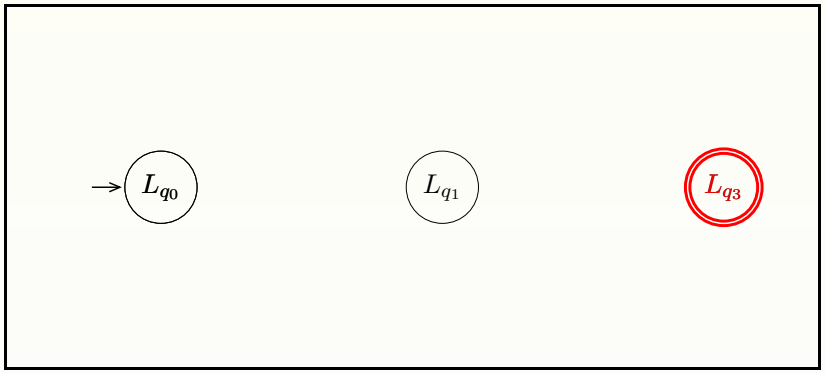
$$L_{q_1} = \Sigma^* a \Sigma \cup \Sigma$$

$$L_{q_2} = \Sigma^* a \Sigma \cup \Sigma \cup \{\varepsilon\}$$

$$L_{q_3} = \Sigma^* a \Sigma \cup \{\varepsilon\} \qquad \text{(final state)}$$

Example

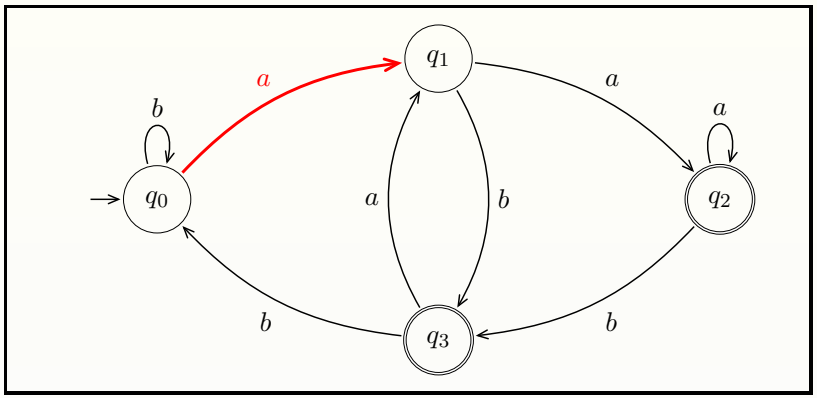


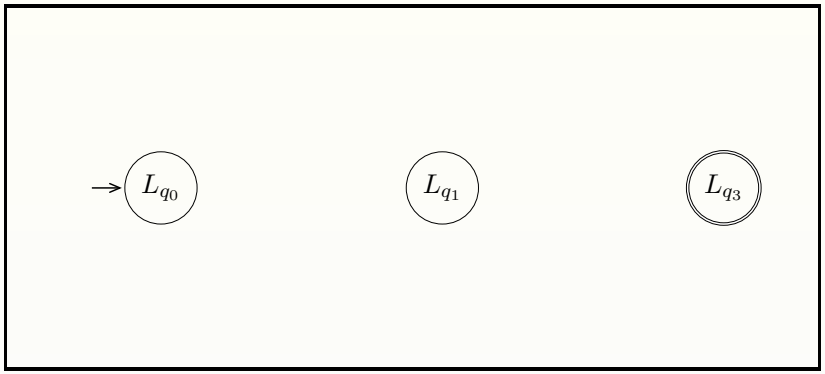


Residual languages for $L = \Sigma^* a \Sigma$

$$L_{q_0} = \Sigma^* a \Sigma$$
$$L_{q_1} = \Sigma^* a \Sigma \cup \Sigma$$
$$L_{q_2} = \Sigma^* a \Sigma \cup \Sigma \cup \{\varepsilon\}$$
$$L_{q_3} = \Sigma^* a \Sigma \cup \{\varepsilon\}$$

Example





Residual languages for $L = \Sigma^* a \Sigma$

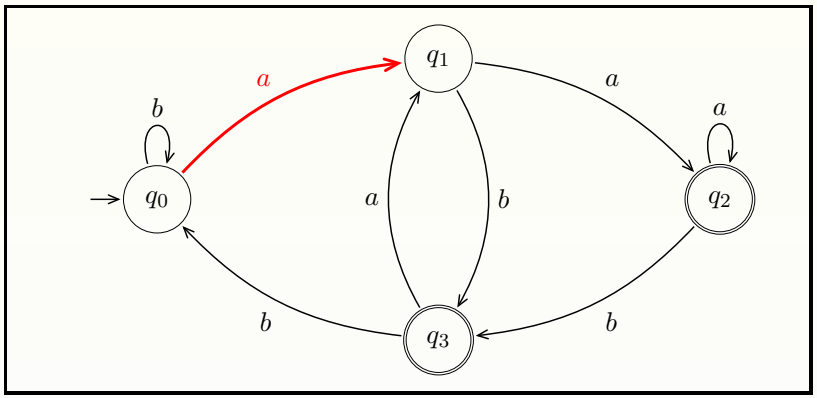
$$L_{q_0} = \Sigma^* a \Sigma \qquad (a\text{-transitions})$$

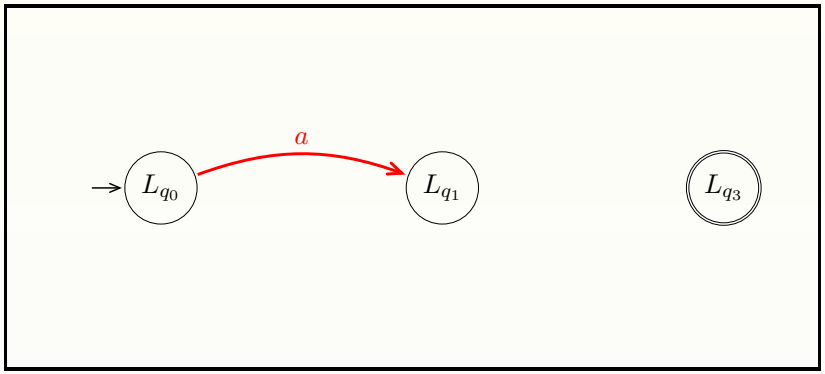
$$L_{q_1} = \Sigma^* a \Sigma \cup \Sigma$$

$$L_{q_2} = \Sigma^* a \Sigma \cup \Sigma \cup \{\varepsilon\}$$

$$L_{q_3} = \Sigma^* a \Sigma \cup \{\varepsilon\}$$

Example





Residual languages for $L = \Sigma^* a \Sigma$

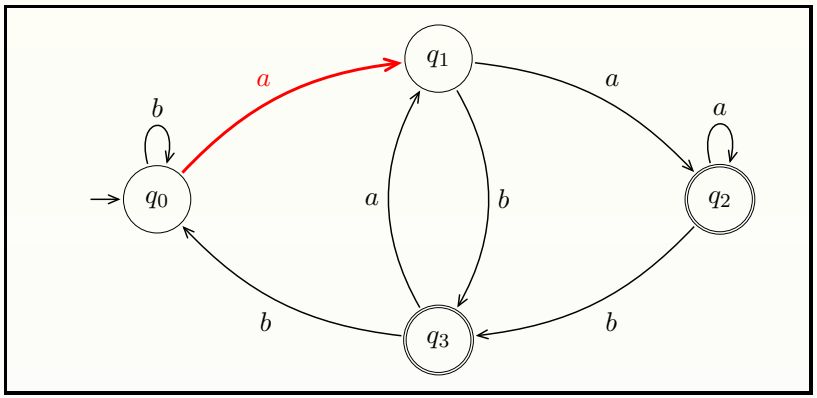
$$L_{q_0} = \Sigma^* a \Sigma \qquad (a\text{-transitions})$$

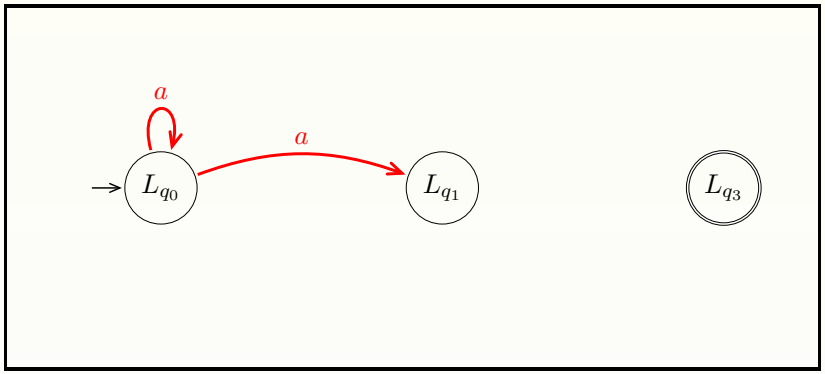
$$L_{q_1} = \Sigma^* a \Sigma \cup \Sigma$$

$$L_{q_2} = \Sigma^* a \Sigma \cup \Sigma \cup \{\varepsilon\}$$

$$L_{q_3} = \Sigma^* a \Sigma \cup \{\varepsilon\}$$

Example





Residual languages for $L = \Sigma^* a \Sigma$

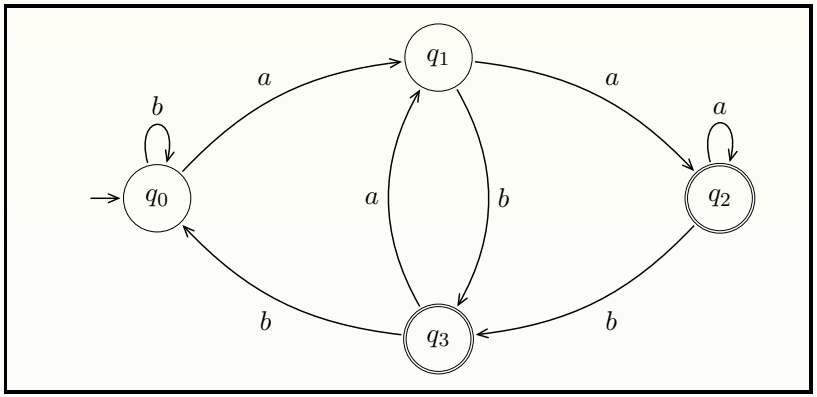
$$L_{q_0} = \Sigma^* a \Sigma \qquad (a\text{-transitions})$$

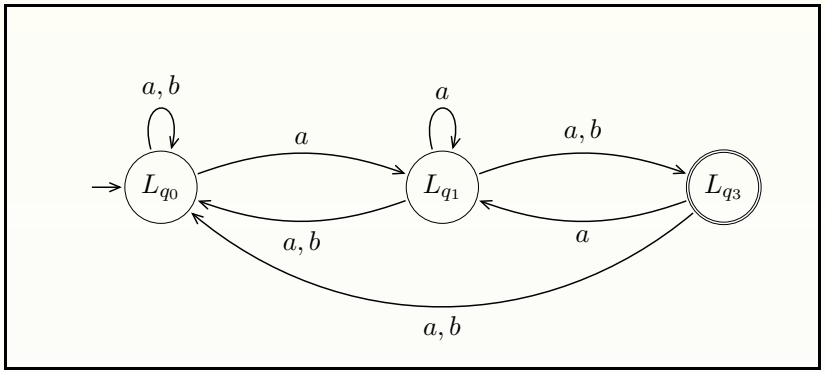
$$L_{q_1} = \Sigma^* a \Sigma \cup \Sigma$$

$$L_{q_2} = \Sigma^* a \Sigma \cup \Sigma \cup \{\varepsilon\}$$

$$L_{q_3} = \Sigma^* a \Sigma \cup \{\varepsilon\}$$

Example



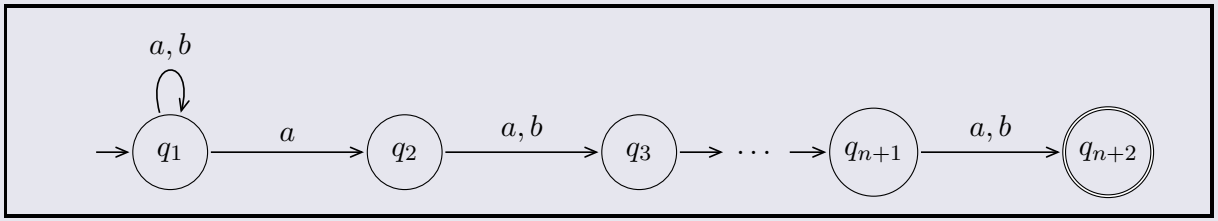


Residual languages for $L = \Sigma^* a \Sigma$

$$L_{q_0} = \Sigma^* a \Sigma$$
$$L_{q_1} = \Sigma^* a \Sigma \cup \Sigma$$
$$L_{q_2} = \Sigma^* a \Sigma \cup \Sigma \cup \{\varepsilon\}$$
$$L_{q_3} = \Sigma^* a \Sigma \cup \{\varepsilon\}$$

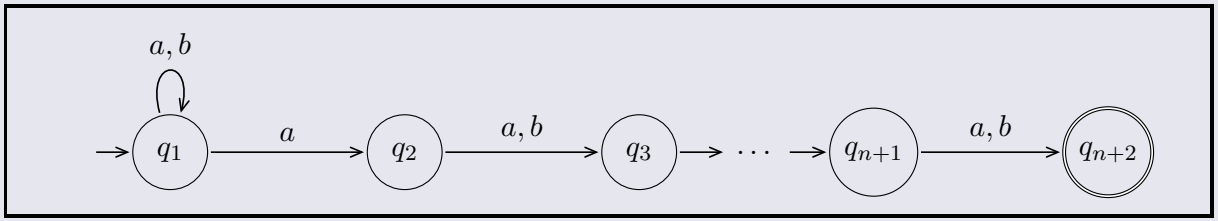
It's worth considering RFSA...

$L_n = \{w \in \Sigma^* | w$ has an a at the $(n+1)$-last position$\}$

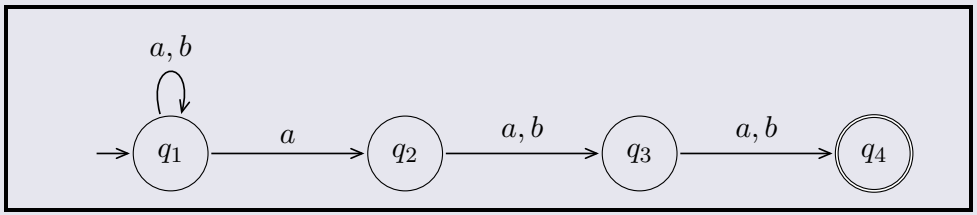


It's worth considering RFSA...

$L_n = \{w \in \Sigma^* | w$ has an a at the $(n+1)$-last position$\}$

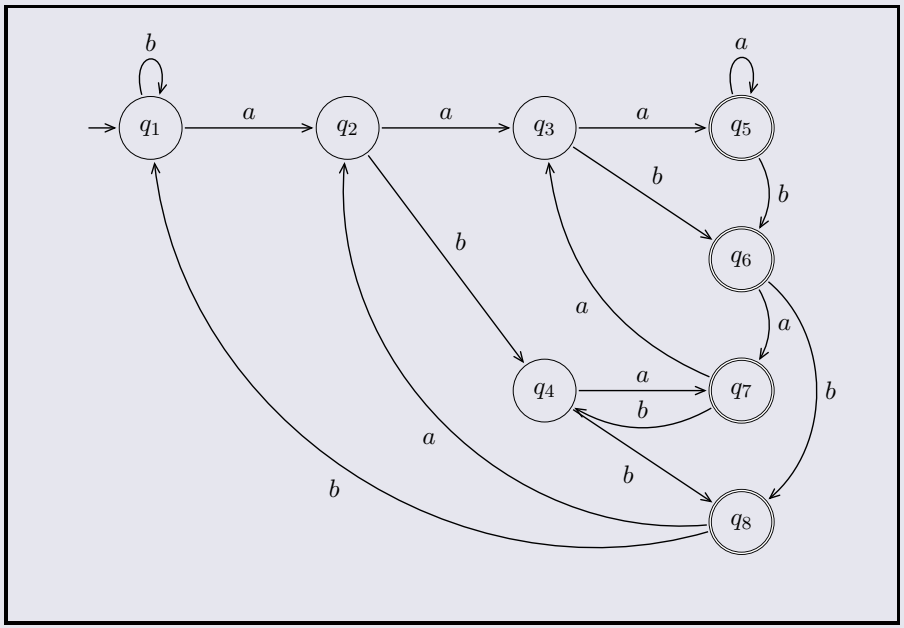


$L_2 = \{w \in \Sigma^* | w$ has an a at the 3^{rd}-last position$\}$



Minimal DFA and RFSA

Minimal DFA and RFSA for L_2:

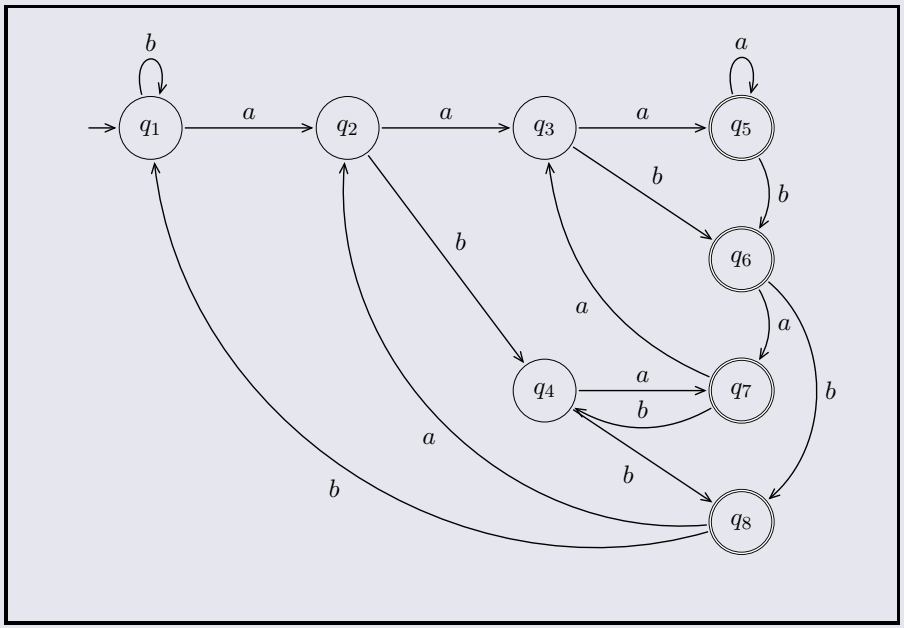


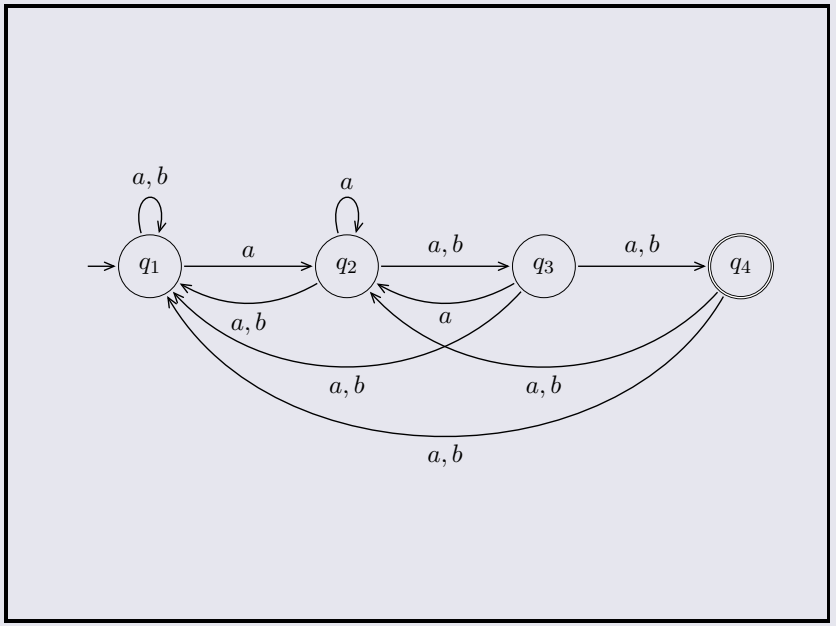
Automata for language L_n:

- minimal DFA general case: 2^{n+1} states

Minimal DFA and RFSA

Minimal DFA and RFSA for L_2:





Automata for language L_n:

- minimal DFA general case: 2^{n+1} states
- *canonical* RFSA general case: $n + 2$ states

Presentation outline

RWTH AACHEN UNIVERSITY

Tables in NL*

$\mathcal{T}$	ε	a	aa
ε	$-$	$-$	$+$
a	$-$	$+$	$+$
ab	$+$	$-$	$+$
b	$-$	$-$	$+$
aa	$+$	$+$	$+$
aba	$-$	$+$	$+$
abb	$-$	$-$	$+$

From tables to RFSA

- we deal with tables

Tables in NL*

$\mathcal{T}$	ε	a	aa
ε	$-$	$-$	$+$
a	$-$	$+$	$+$
ab	$+$	$-$	$+$
b	$-$	$-$	$+$
aa	$+$	$+$	$+$
aba	$-$	$+$	$+$
abb	$-$	$-$	$+$

From tables to RFSA

- we deal with tables
- table rows approximate residual languages

Tables in NL*

$\mathcal{T}$	ε	a	aa
ε	$-$	$-$	$+$
a	$-$	$+$	$+$
ab	$+$	$-$	$+$
b	$-$	$-$	$+$
aa	$+$	$+$	$+$
aba	$-$	$+$	$+$
abb	$-$	$-$	$+$

From tables to RFSA

- we deal with tables
- table rows approximate residual languages
- not all rows represent states

Tables in NL*

$\mathcal{T}$	ε	a	aa
ε	$-$	$-$	$+$
a	$-$	$+$	$+$
ab	$+$	$-$	$+$
b	$-$	$-$	$+$
aa	$+$	$+$	$+$
aba	$-$	$+$	$+$
abb	$-$	$-$	$+$

From tables to RFSA

- we deal with tables
- table rows approximate residual languages
- not all rows represent states
- as long as there is no other evidence: equal rows represent equal residual languages

Tables in NL*

$\mathcal{T}$	ε	a	aa
ε	$-$	$-$	$+$
a	$-$	$+$	$+$
ab	$+$	$-$	$+$
b	$-$	$-$	$+$
aa	$+$	$+$	$+$
aba	$-$	$+$	$+$
abb	$-$	$-$	$+$

From tables to RFSA

- we deal with tables
- table rows approximate residual languages
- not all rows represent states
- as long as there is no other evidence: equal rows represent equal residual languages
- transition relation respects language inclusion

Tables in NL*

$\mathcal{T}$	ε	a	aa
ε	$-$	$-$	$+$
a	$-$	$+$	$+$
ab	$+$	$-$	$+$
b	$-$	$-$	$+$
aa	$+$	$+$	$+$
aba	$-$	$+$	$+$
abb	$-$	$-$	$+$

From tables to RFSA

- we deal with tables
- table rows approximate residual languages
- not all rows represent states
- as long as there is no other evidence: equal rows represent equal residual languages
- transition relation respects language inclusion

Tables in NL*

$\mathcal{T}$	ε	a	aa
ε	$-$	$-$	$+$
a	$-$	$+$	$+$
ab	$+$	$-$	$+$
b	$-$	$-$	$+$
aa	$+$	$+$	$+$
aba	$-$	$+$	$+$
abb	$-$	$-$	$+$

From tables to RFSA

- we deal with tables
- table rows approximate residual languages
- not all rows represent states
- as long as there is no other evidence: equal rows represent equal residual languages
- transition relation respects language inclusion

Tables in NL*

$\mathcal{T}$	ε	a	aa	$\ldots$
ε	$-$	$-$	$+$	$\ldots$
a	$-$	$+$	$+$	$\ldots$
ab	$+$	$-$	$+$	$\ldots$
b	$-$	$-$	$+$	$\ldots$
aa	$+$	$+$	$+$	$\ldots$
aba	$-$	$+$	$+$	$\ldots$
abb	$-$	$-$	$+$	$\ldots$

From tables to RFSA

- we deal with tables
- table rows approximate residual languages
- not all rows represent states
- as long as there is no other evidence: equal rows represent equal residual languages
- transition relation respects language inclusion
- treatment of counterexamples:
  - add to columns
  - otherwise non-termination

Table properties

$\mathcal{T}$	ε	a	aa
ε	$-$	$-$	$+$
a	$-$	$+$	$+$
ab	$+$	$-$	$+$
b	$-$	$-$	$+$
aa	$+$	$+$	$+$
aba	$-$	$+$	$+$
abb	$-$	$-$	$+$

Closedness

- all states identifiable from the table

Table properties

$\mathcal{T}$	ε	a	aa
ε	$-$	$-$	$+$
a	$-$	$+$	$+$
ab	$+$	$-$	$+$
b	$-$	$-$	$+$
aa	$+$	$+$	$+$
aba	$-$	$+$	$+$
abb	$-$	$-$	$+$

Closedness

- all states identifiable from the table
- all *non-composed* rows have to be in the upper part of the table

Table properties

$\mathcal{T}$	ε	a	aa
ε	$-$	$-$	$+$
a	$-$	$+$	$+$
ab	$+$	$-$	$+$
b	$-$	$-$	$+$
aa	$+$	$+$	$+$
aba	$-$	$+$	$+$
abb	$-$	$-$	$+$

Closedness

- all states identifiable from the table
- all *non-composed* rows have to be in the upper part of the table
- all other rows can be composed by upper rows

Table properties

$\mathcal{T}$	ε	a	aa
ε	$-$	$-$	$+$
a	$-$	$+$	$+$
ab	$+$	$-$	$+$
b	$-$	$-$	$+$
aa	$+$	$+$	$+$
aba	$-$	$+$	$+$
abb	$-$	$-$	$+$

Closedness

- all states identifiable from the table
- all *non-composed* rows have to be in the upper part of the table
- all other rows can be composed by upper rows

Consistency

- transition relation respects language inclusion

Table properties

$\mathcal{T}$	ε	a	aa
ε	$-$	$-$	$+$
a	$-$	$+$	$+$
ab	$+$	$-$	$+$
b	$-$	$-$	$+$
aa	$+$	$+$	$+$
aba	$-$	$+$	$+$
abb	$-$	$-$	$+$

Closedness

- all states identifiable from the table
- all *non-composed* rows have to be in the upper part of the table
- all other rows can be composed by upper rows

Consistency

- transition relation respects language inclusion

Table properties

$\mathcal{T}$	ε	a	aa
ε	$-$	$-$	$+$
a	$-$	$+$	$+$
ab	$+$	$-$	$+$
b	$-$	$-$	$+$
aa	$+$	$+$	$+$
aba	$-$	$+$	$+$
abb	$-$	$-$	$+$

Closedness

- all states identifiable from the table
- all *non-composed* rows have to be in the upper part of the table
- all other rows can be composed by upper rows

Consistency

- transition relation respects language inclusion

Towards correctness

Theorem (Complexity of NL*)

Let:

- n: number of states of minimal DFA $\mathcal{A}_L$ for regular language L,
- m: length of the biggest counterexample

Then, NL* returns after at most:

the canonical RFSA $\mathcal{R}(L)$.

Towards correctness

Theorem (Complexity of NL*)

Let:

- n: number of states of minimal DFA $\mathcal{A}_L$ for regular language L,
- m: length of the biggest counterexample

Then, NL* returns after at most:

- $O(n^2)$ equivalence queries and

the canonical RFSA $\mathcal{R}(L)$.

Towards correctness

Theorem (Complexity of NL*)

Let:

- n: number of states of minimal DFA $\mathcal{A}_L$ for regular language L,
- m: length of the biggest counterexample

Then, NL* returns after at most:

- $O(n^2)$ equivalence queries and
- $O(m|\Sigma|n^3)$ membership queries

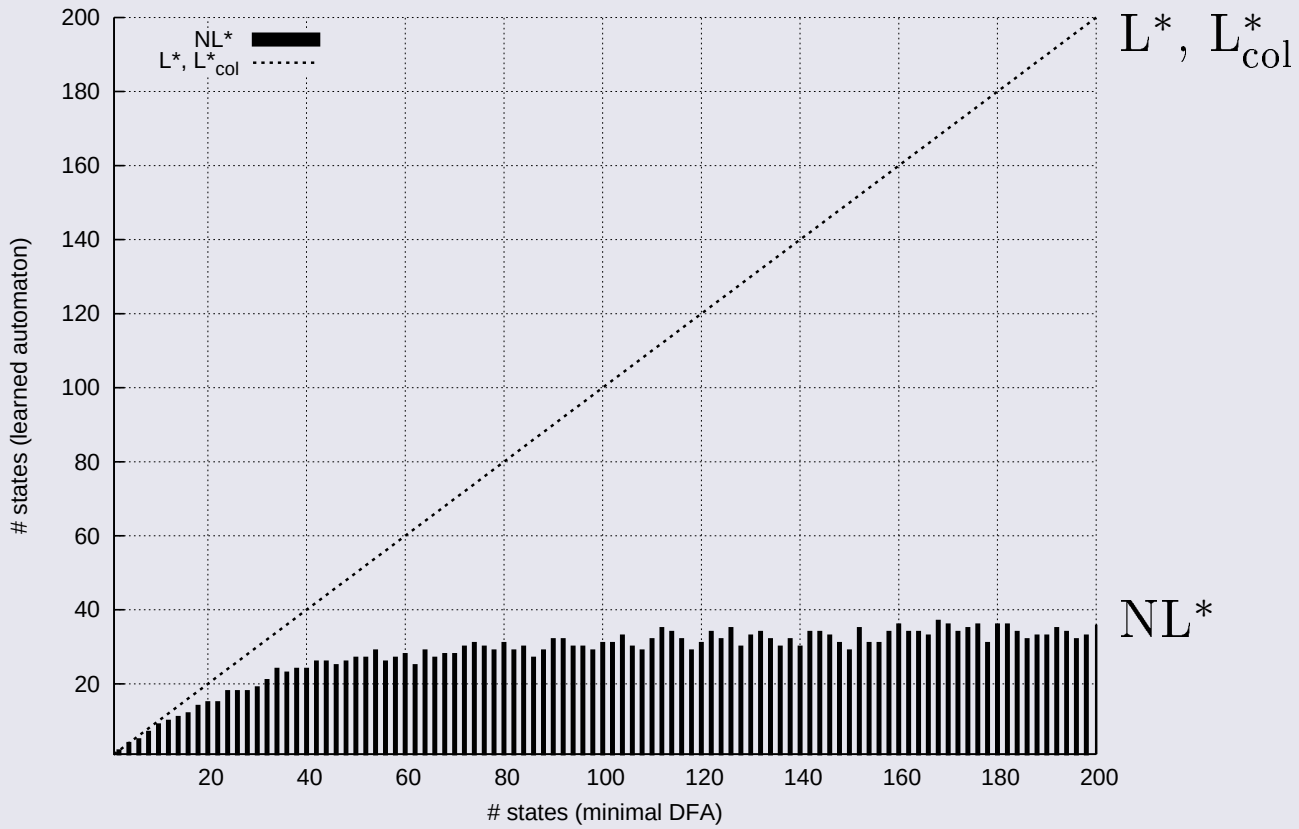
the canonical RFSA $\mathcal{R}(L)$.

Presentation outline

RWTH AACHEN UNIVERSITY

Algorithm - Overview

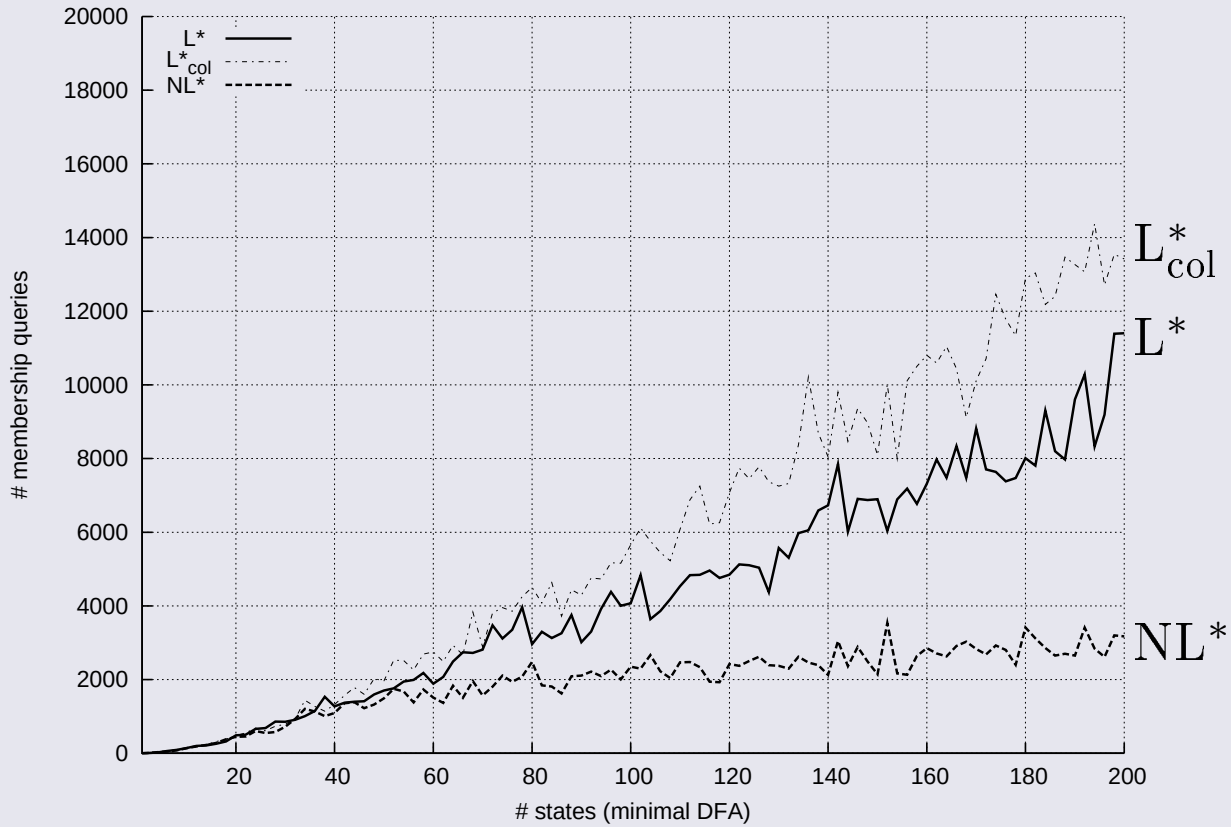
Number of states (L^*, L^*_{col} vs. NL^*)



- $\approx$ 3200 reg. exp. with minimal DFA of 1 to 200 states

Algorithm - Overview

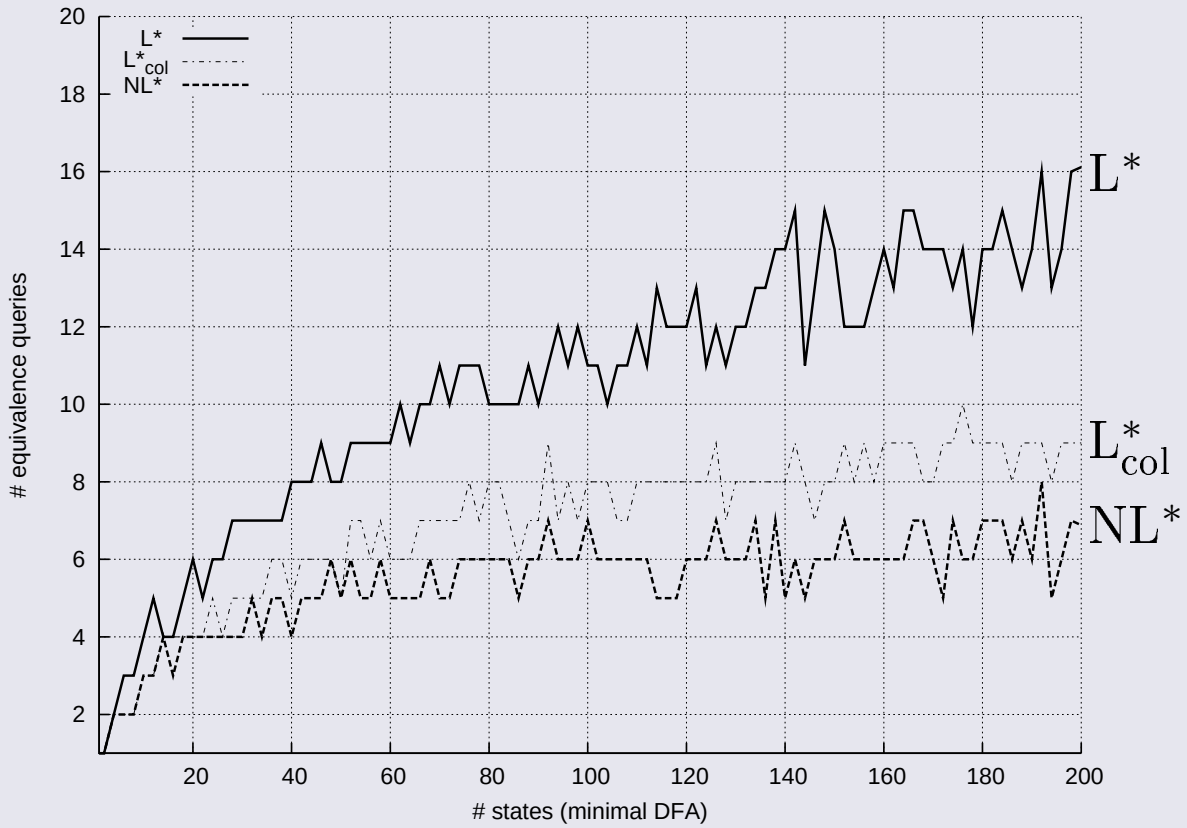
Number of membership queries (L^* vs. L^*_{col} vs. NL^*)



- ≈ 3200 reg. exp. with minimal DFA of 1 to 200 states

Algorithm - Overview

Number of equivalence queries (L^* vs. L^*_{col} vs. NL^*)

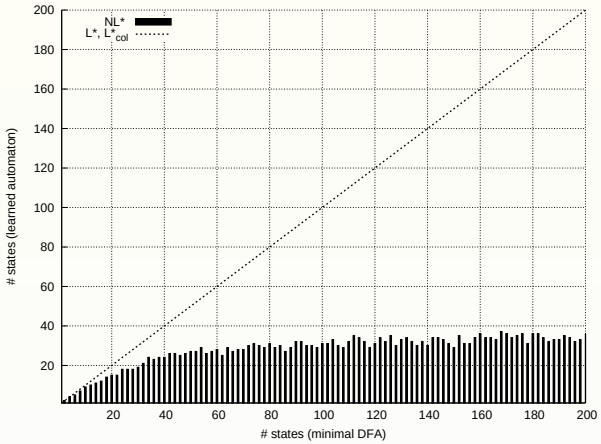


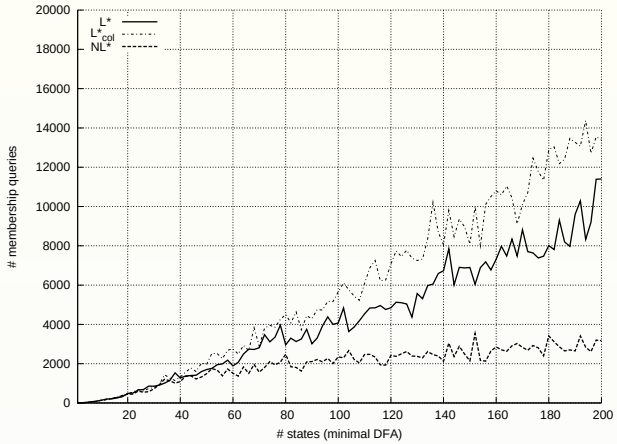
- $\approx$ 3200 reg. exp. with minimal DFA of 1 to 200 states

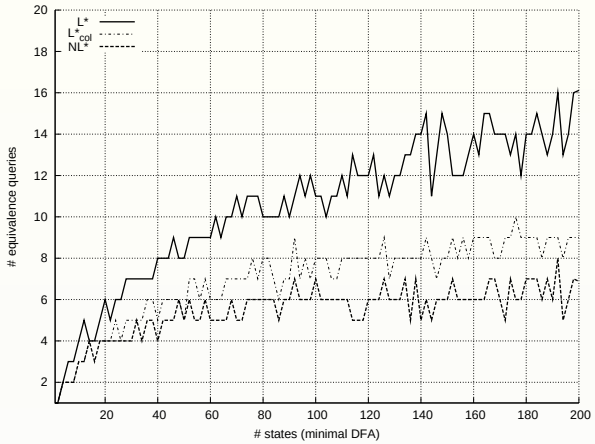
Presentation outline

RWTH AACHEN UNIVERSITY

That's it!



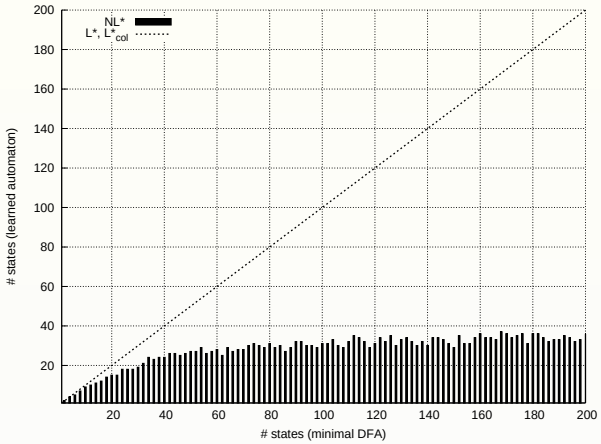


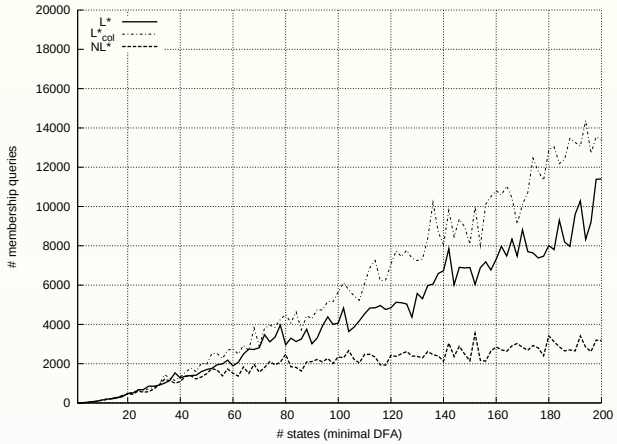


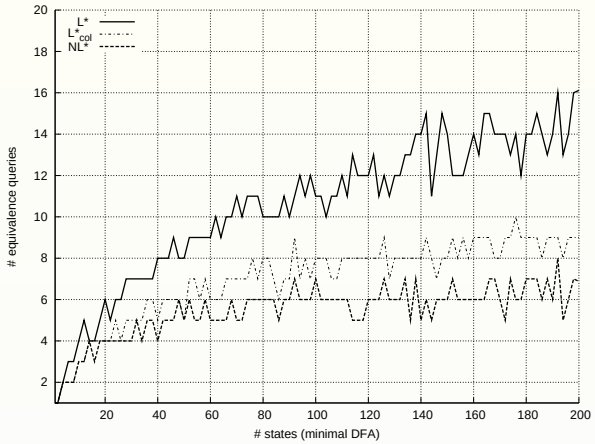
What did *we learn* ?

- NL* outperforms L* by far

That's it!





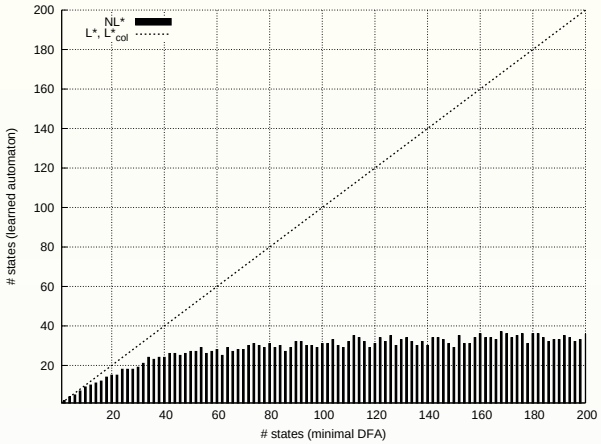


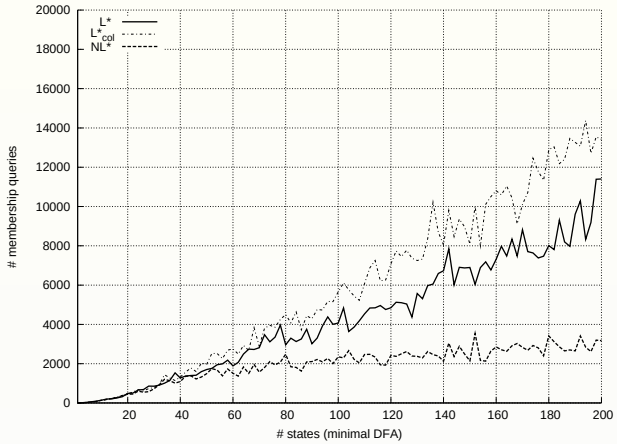
What did *we learn* ?

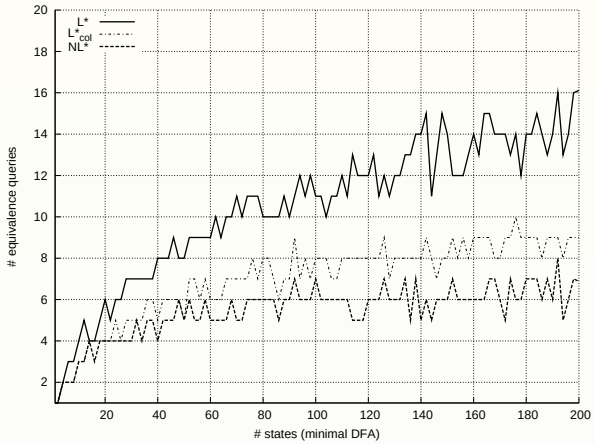
- NL* outperforms L* by far
- Learning using NFA works well in practice!

RWTH AACHEN UNIVERSITY

That's it!





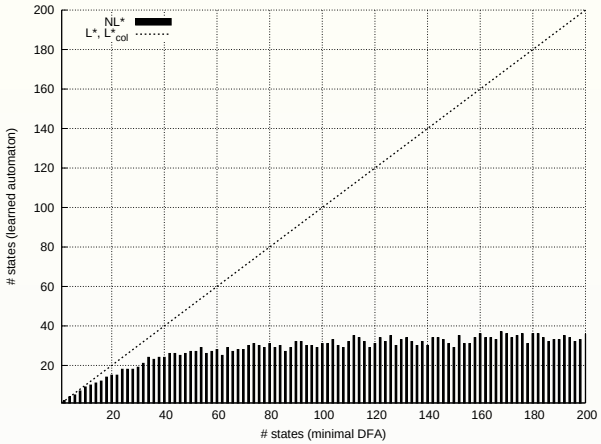


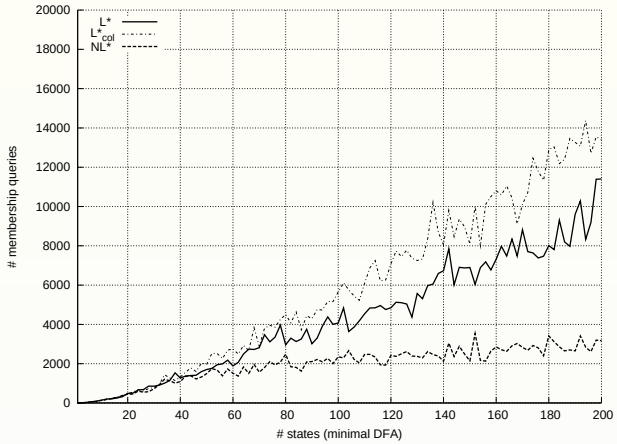
What did *we learn* ?

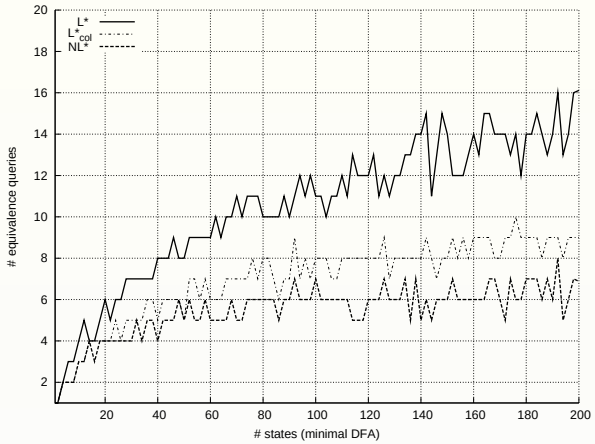
- NL* outperforms L* by far
- Learning using NFA works well in practice!
- Nondeterminism does not always hurt!!!

RWTH AACHEN UNIVERSITY

That's it!







What did *we learn* ?

- NL* outperforms L* by far
- Learning using NFA works well in practice!
- Nondeterminism does not always hurt!!!

Thanks!

RWTH AACHEN UNIVERSITY