Learning Regular Languages over Large Alphabets

Oded Maler   Irini Eleftheria Mens

CNRS-VERIMAG
University of Grenoble
2, avenue de Vignate
38610 Gieres
France

January 10, 2014

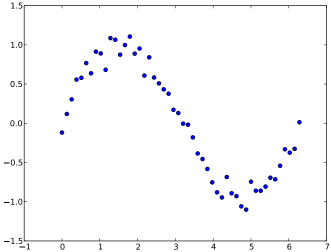
Outline

Machine learning in general

- given a sample $M = \{(x, y) \mid x \in X,\ y \in Y\}$
- find a representation $f : X \to Y$ such that $f(x) = y$
- predict or identify $f(x)$ for all $x \in X$

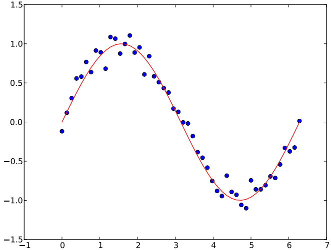
Machine learning in general

- given a sample $M = \{(x, y) \mid x \in X,\, y \in Y\}$
- find a representation $f : X \to Y$ such that $f(x) = y$
- predict or identify $f(x)$ for all $x \in X$



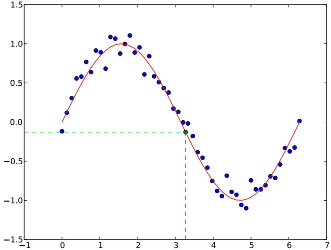
Machine learning in general

- given a sample $M = \{(x, y) \mid x \in X,\, y \in Y\}$
- find a representation $f : X \to Y$ such that $f(x) = y$
- predict or identify $f(x)$ for all $x \in X$



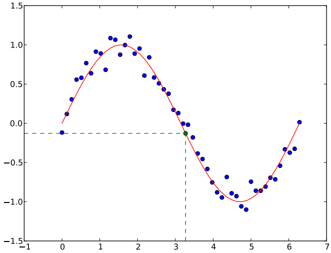
Machine learning in general

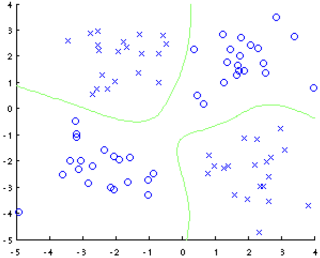
- given a sample $M = \{(x, y) \mid x \in X,\, y \in Y\}$
- find a representation $f : X \to Y$ such that $f(x) = y$
- predict or identify $f(x)$ for all $x \in X$



Machine learning in general

- given a sample $M = \{(x, y) \mid x \in X, y \in Y\}$
- find a representation $f : X \to Y$ such that $f(x) = y$
- predict or identify $f(x)$ for all $x \in X$

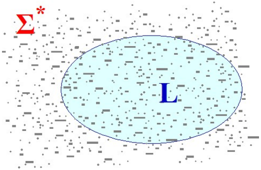




Learning regular languages

Learning regular languages

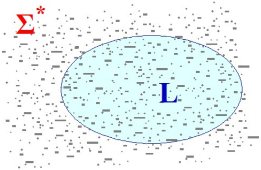
- Σ alphabet
- $L \subseteq \Sigma^*$ an unknown regular language (*target language*)



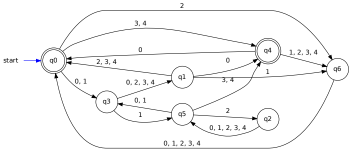
Learning regular languages

- Σ alphabet
- $L \subseteq \Sigma^*$ an unknown regular language (*target language*)

Find a DFA A such that $L = L(A)$



?



Queries

Queries

- Membership Queries

Queries

- Membership Queries

$w \in \Sigma^* \longrightarrow$ **MQ** $w \in L$? $\longrightarrow$ yes / no

- Equivalence Queries

$h \longrightarrow$ **EQ** $L = L(h)$? $\longrightarrow$ yes / c-ex

General Idea of Angluin's algorithm L^*

Repeat:

- ask MQs to complete the observation table
- find a closed and consistent observation table
- ask an EQ for it
- use the counterexample to update the table

Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

- $S \subseteq \Sigma^*$ prefixes

Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

- $S \subseteq \Sigma^*$ prefixes
- $R = S \cdot \Sigma \setminus S$ boundary

Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

- $S \subseteq \Sigma^*$ prefixes
- $R = S \cdot \Sigma \setminus S$ boundary
- $E \subseteq \Sigma^*$ suffixes

Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

- $S \subseteq \Sigma^*$ prefixes
- $R = S \cdot \Sigma \setminus S$ boundary
- $E \subseteq \Sigma^*$ suffixes
- $f : S \cup R \times E \to \{-, +\}$ classification function

Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

- $S \subseteq \Sigma^*$ prefixes
- $R = S \cdot \Sigma \setminus S$ boundary
- $E \subseteq \Sigma^*$ suffixes
- $f : S \cup R \times E \rightarrow \{-, +\}$ classification function

Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

- $S \subseteq \Sigma^*$ prefixes
- $R = S \cdot \Sigma \setminus S$ boundary
- $E \subseteq \Sigma^*$ suffixes
- $f : S \cup R \times E \to \{-, +\}$ classification function

Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

- $S \subseteq \Sigma^*$ prefixes
- $R = S \cdot \Sigma \setminus S$ boundary
- $E \subseteq \Sigma^*$ suffixes
- $f : S \cup R \times E \to \{-, +\}$ classification function
- $f_s : E \to \{-, +\}$ for all $s \in S \cup R$
  $$f_s(e) = f(s \cdot e)$$

Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

- $S \subseteq \Sigma^*$ prefixes
- $R = S \cdot \Sigma \setminus S$ boundary
- $E \subseteq \Sigma^*$ suffixes
- $f : S \cup R \times E \to \{-, +\}$ classification function
- $f_s : E \to \{-, +\}$ for all $s \in S \cup R$
$$f_s(e) = f(s \cdot e)$$

Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

- $S \subseteq \Sigma^*$ prefixes
- $R = S \cdot \Sigma \setminus S$ boundary
- $E \subseteq \Sigma^*$ suffixes
- $f : S \cup R \times E \to \{-, +\}$ classification function
- $f_s : E \to \{-, +\}$ for all $s \in S \cup R$
  $$f_s(e) = f(s \cdot e)$$

Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

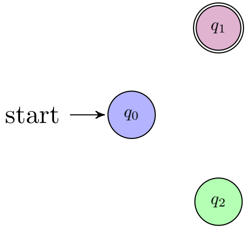
- $S \subseteq \Sigma^*$ prefixes
- $R = S \cdot \Sigma \setminus S$ boundary
- $E \subseteq \Sigma^*$ suffixes
- $f : S \cup R \times E \to \{-, +\}$ classification function
- $f_s : E \to \{-, +\}$ for all $s \in S \cup R$
  $$f_s(e) = f(s \cdot e)$$

Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

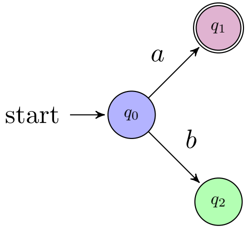
Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$



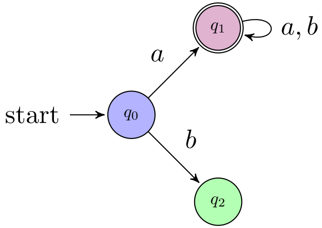
Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$



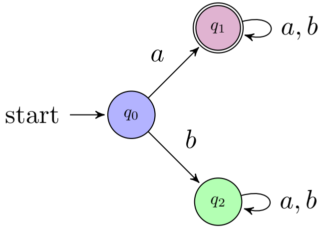
Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$



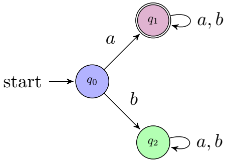
Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$



Observation Table - $T = (\Sigma, S, R, E, f)$

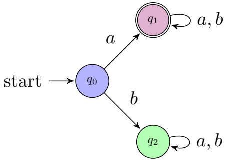
	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$



Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

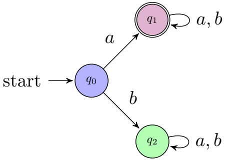
- $S \cup R$ is prefix-closed



Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

- $S \cup R$ is prefix-closed
- E is suffix-closed

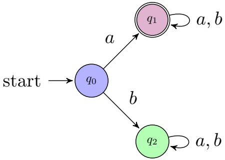


Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

- $S \cup R$ is prefix-closed
- E is suffix-closed

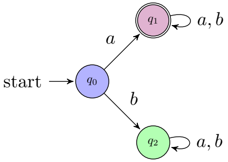
  - T closed



Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

- $S \cup R$ is prefix-closed
- E is suffix-closed

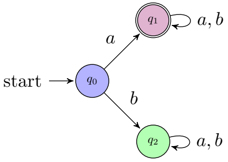


- T closed
- T consistent

Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

- $S \cup R$ is prefix-closed
- E is suffix-closed

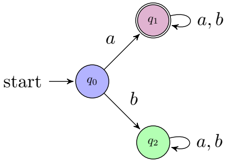


- T closed
- T consistent
- T reduced

Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

- $S \cup R$ is prefix-closed
- E is suffix-closed

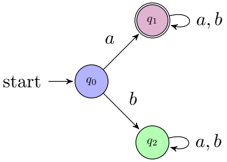


- T closed
  $\forall r \in R, \exists s \in S, f_r = f_s$
- T consistent
  $\forall s, s' \in S, \forall a \in \Sigma,\ f_s = f_{s'} \Rightarrow f_{s \cdot a} = f_{s' \cdot a}$
- T reduced
  $\forall s, s' \in S, f_s \neq f_{s'}$

Observation Table - $T = (\Sigma, S, R, E, f)$

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$+$	$-$

- $S \cup R$ is prefix-closed
- E is suffix-closed



- T closed
  $\forall r \in R, \exists s \in S, f_r = f_s$
- T consistent
  $\forall s, s' \in S, \forall a \in \Sigma,\ f_s = f_{s'} \Rightarrow f_{s \cdot a} = f_{s' \cdot a}$
- T reduced
  $\forall s, s' \in S, f_s \neq f_{s'}$

Angluin's algorithm

1. Initialize table $T = (\Sigma,\ S = \{\epsilon\},\ E = \{\epsilon\},\ R = \Sigma,\ f)$
2. Ask MQs for ϵ and each $\sigma \in \Sigma$ to fill in T
3. **Repeat**
4. **While** T is not closed or not consistent:
5. **If** T not consistent **then**:
6. find $s_1, s_2 \in S$, $a \in \Sigma$, $e \in E : f_{s_1} = f_{s_2}$ and $f(s_1 \cdot a, e) \neq f(s_2 \cdot a, e)$.
7. add $a \cdot e$ to E
8. fill in T by asking membership queries
9. **If** T not closed **then**:
10. find $s_1 \in S$, and $a \in \Sigma : f_{s_1 \cdot a} \neq f_s, \forall s \in S$
11. add $s_1 \cdot a$ to S and $s_1 \cdot a \cdot \sigma$ to R, $\forall \sigma \in \Sigma$
12. fill in T by asking membership queries
13. Once T is closed and consistent, ask EQ for T
14. **If** answer is a counterexample **then**:
15. add it and all prefixes to S
16. fill in T by asking MQs
17. **Until** answer is yes
18. **Return** T

example ($\Sigma = \{a, b\}$)

observation table

	ϵ
ϵ	

example ($\Sigma = \{a, b\}$)

observation table

	ϵ
ϵ	
a	
b	

example $(\Sigma = \{a, b\})$

observation table

	ϵ
ϵ	$-$
a	$+$
b	$-$

example ($\Sigma = \{a, b\}$)

observation table

	ϵ
ϵ	$-$
a	$+$
b	$-$

example ($\Sigma = \{a, b\}$)

observation table

	ϵ
ϵ	$-$
a	$+$
b	$-$

example ($\Sigma = \{a, b\}$)

observation table

	ϵ
ϵ	$-$
a	$+$
b	$-$
aa	
ab	

example ($\Sigma = \{a, b\}$)

observation table

	ϵ
ϵ	$-$
a	$+$
b	$-$
aa	$+$
ab	$+$

example ($\Sigma = \{a, b\}$)

observation table

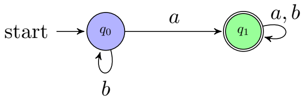
	ϵ
ϵ	$-$
a	$+$
b	$-$
aa	$+$
ab	$+$

example ($\Sigma = \{a, b\}$)

observation table

	ϵ
ϵ	$-$
a	$+$
b	$-$
aa	$+$
ab	$+$

hypothesis automaton

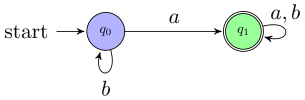


example ($\Sigma = \{a, b\}$)

observation table

	ϵ
ϵ	$-$
a	$+$
b	$-$
aa	$+$
ab	$+$

hypothesis automaton



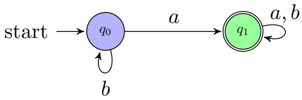
counterexample: $-ba$

example ($\Sigma = \{a, b\}$)

observation table

	ϵ
ϵ	$-$
a	$+$
ba	$-$
b	$-$
aa	$+$
ab	$+$

hypothesis automaton



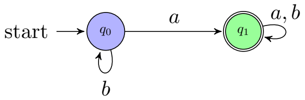
counterexample: $-ba$

example $(\Sigma = \{a, b\})$

observation table

	ϵ
ϵ	$-$
a	$+$
b	$-$
ba	$-$
aa	$+$
ab	$+$

hypothesis automaton



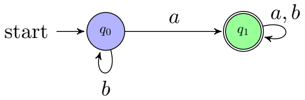
counterexample: $-ba$

example ($\Sigma = \{a, b\}$)

observation table

	ϵ
ϵ	$-$
a	$+$
b	$-$
ba	$-$
aa	$+$
ab	$+$
bb	
baa	
bab	

hypothesis automaton



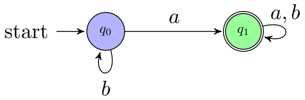
counterexample: $-ba$

example ($\Sigma = \{a, b\}$)

observation table

	ϵ
ϵ	$-$
a	$+$
b	$-$
ba	$-$
aa	$+$
ab	$+$
bb	$-$
baa	$-$
bab	$-$

hypothesis automaton



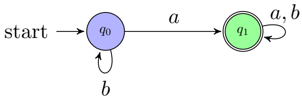
counterexample: $-ba$

example ($\Sigma = \{a, b\}$)

observation table

	ϵ
ϵ	$-$
a	$+$
b	$-$
ba	$-$
aa	$+$
ab	$+$
bb	$-$
baa	$-$
bab	$-$

hypothesis automaton



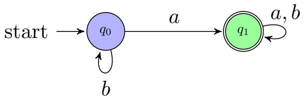
counterexample: $-ba$

example ($\Sigma = \{a, b\}$)

observation table

	ϵ
ϵ	$-$
a	$+$
b	$-$
$b\,a$	$-$
aa	$+$
ab	$+$
bb	$-$
baa	$-$
bab	$-$

hypothesis automaton



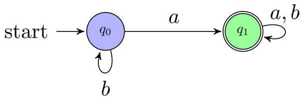
counterexample: $-ba$

example ($\Sigma = \{a, b\}$)

observation table

	ϵ	a
ϵ	$-$	
a	$+$	
b	$-$	
$b\ a$	$-$	
aa	$+$	
ab	$+$	
bb	$-$	
baa	$-$	
bab	$-$	

hypothesis automaton



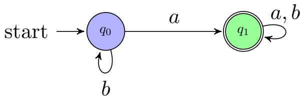
counterexample: $-ba$

example ($\Sigma = \{a, b\}$)

observation table

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
$b\ a$	$-$	$-$
aa	$+$	
ab	$+$	
bb	$-$	
baa	$-$	
bab	$-$	

hypothesis automaton

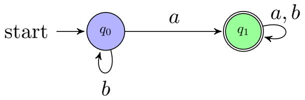


example ($\Sigma = \{a, b\}$)

observation table

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
$b\ a$	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

hypothesis automaton

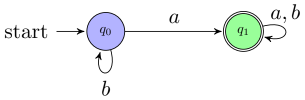


example ($\Sigma = \{a, b\}$)

observation table

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

hypothesis automaton

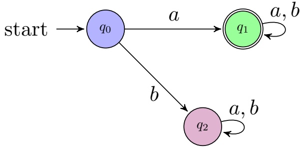


example ($\Sigma = \{a, b\}$)

observation table

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

hypothesis automaton

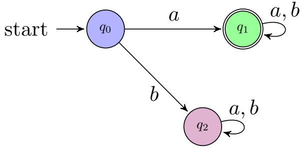


example ($\Sigma = \{a, b\}$)

observation table

	ϵ	a
ϵ	$-$	$+$
a	$+$	$+$
b	$-$	$-$
ba	$-$	$-$
aa	$+$	$+$
ab	$+$	$+$
bb	$-$	$-$
baa	$-$	$-$
bab	$-$	$-$

hypothesis automaton



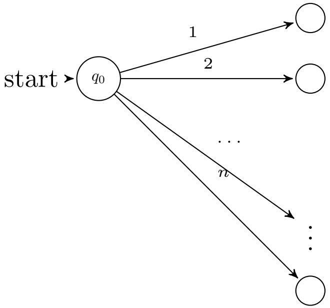
True

it is not enough...

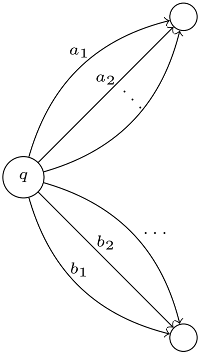
Let $\Sigma = \mathbb{N}$

observation table

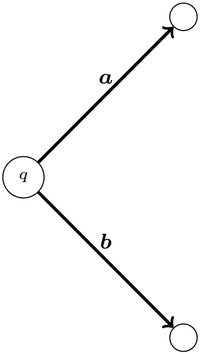
	ϵ
ϵ	
1	
2	
3	
4	
5	
$\vdots$	



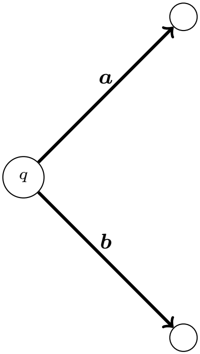
group them..



group them..

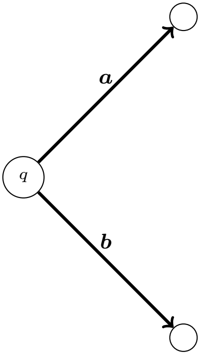


group them..



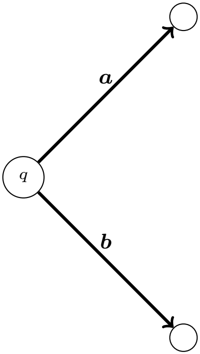
- $\psi_q : \Sigma \to \boldsymbol{\Sigma}$
  - $\psi(a_1) = \boldsymbol{a}, \psi(a_2) = \boldsymbol{a}, \ldots$
  - $\psi(b_1) = \boldsymbol{b}, \psi(b_2) = \boldsymbol{b}, \ldots$

group them..



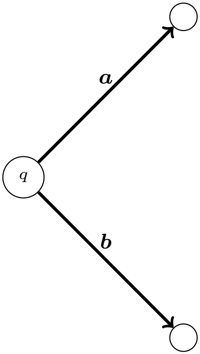
- $\psi_q : \Sigma \to \boldsymbol{\Sigma}$
  - $\psi(a_1) = \boldsymbol{a}, \psi(a_2) = \boldsymbol{a}, \ldots$
  - $\psi(b_1) = \boldsymbol{b}, \psi(b_2) = \boldsymbol{b}, \ldots$
- $[a] = \{a_1, a_2, \ldots\}$, $[b] = \{b_1, b_2, \ldots\}$

group them..



- $\psi_q : \Sigma \to \mathbf{\Sigma}$
  - $\psi(a_1) = \boldsymbol{a}, \psi(a_2) = \boldsymbol{a}, \ldots$
  - $\psi(b_1) = \boldsymbol{b}, \psi(b_2) = \boldsymbol{b}, \ldots$
- $[a] = \{a_1, a_2, \ldots\}$, $[b] = \{b_1, b_2, \ldots\}$
- $\mathbf{\Sigma}_q = \{\boldsymbol{a}, \boldsymbol{b}, \ldots\}$, $\mathbf{\Sigma} = \biguplus_{q \in Q} \mathbf{\Sigma}_q$

group them..



- $\psi_q : \Sigma \rightarrow \boldsymbol{\Sigma}$
    - $\psi(a_1) = \boldsymbol{a}, \psi(a_2) = \boldsymbol{a}, \ldots$
    - $\psi(b_1) = \boldsymbol{b}, \psi(b_2) = \boldsymbol{b}, \ldots$
- $[a] = \{a_1, a_2, \ldots\}$, $[b] = \{b_1, b_2, \ldots\}$
- $\boldsymbol{\Sigma}_q = \{\boldsymbol{a}, \boldsymbol{b}, \ldots\}$, $\boldsymbol{\Sigma} = \biguplus_{q \in Q} \boldsymbol{\Sigma}_q$

$\mu(\boldsymbol{a}) \subset [\boldsymbol{a}]$ evidences

deterministic symbolic automaton

$\boldsymbol{\mathcal{A}} = (\Sigma, \boldsymbol{\Sigma}, \psi, Q, \delta, q_0, F)$

deterministic symbolic automaton

$\mathcal{A} = (\Sigma, \mathbf{\Sigma}, \psi, Q, \delta, q_0, F)$, where

- Σ is the input alphabet,

deterministic symbolic automaton

$\boldsymbol{\mathcal{A}} = (\Sigma, \boldsymbol{\Sigma}, \psi, Q, \delta, q_0, F)$, where

- Σ is the input alphabet,
- Q is a finite set of states,

deterministic symbolic automaton

$\boldsymbol{\mathcal{A}} = (\Sigma, \boldsymbol{\Sigma}, \psi, Q, \delta, q_0, F)$, where

- Σ is the input alphabet,
- Q is a finite set of states,
- q_0 is the initial state,

deterministic symbolic automaton

$\boldsymbol{\mathcal{A}} = (\Sigma, \boldsymbol{\Sigma}, \psi, Q, \delta, q_0, F)$, where

- Σ is the input alphabet,
- Q is a finite set of states,
- q_0 is the initial state,
- F is the set of accepting states,

deterministic symbolic automaton

$\boldsymbol{\mathcal{A}} = (\Sigma, \boldsymbol{\Sigma}, \psi, Q, \delta, q_0, F)$, where

- Σ is the input alphabet,
- Q is a finite set of states,
- q_0 is the initial state,
- F is the set of accepting states,
- $\boldsymbol{\Sigma}$ is a finite alphabet, decomposable into $\boldsymbol{\Sigma} = \biguplus_{q \in Q} \boldsymbol{\Sigma}_q$,

deterministic symbolic automaton

$\boldsymbol{\mathcal{A}} = (\Sigma, \boldsymbol{\Sigma}, \psi, Q, \delta, q_0, F)$, where

- Σ is the input alphabet,
- Q is a finite set of states,
- q_0 is the initial state,
- F is the set of accepting states,
- $\boldsymbol{\Sigma}$ is a finite alphabet, decomposable into $\boldsymbol{\Sigma} = \biguplus_{q \in Q} \boldsymbol{\Sigma}_q$,
- $\psi = \{\psi_q : q \in Q\}$ is a family of total surjective functions $\psi_q : \Sigma \to \boldsymbol{\Sigma}_q$,

deterministic symbolic automaton

$\mathcal{A} = (\Sigma, \mathbf{\Sigma}, \psi, Q, \delta, q_0, F)$, where

- Σ is the input alphabet,
- Q is a finite set of states,
- q_0 is the initial state,
- F is the set of accepting states,
- $\mathbf{\Sigma}$ is a finite alphabet, decomposable into $\mathbf{\Sigma} = \biguplus_{q \in Q} \mathbf{\Sigma}_q$,
- $\psi = \{\psi_q : q \in Q\}$ is a family of total surjective functions $\psi_q : \Sigma \to \mathbf{\Sigma}_q$,
- $\delta : Q \times \mathbf{\Sigma} \to Q$ is a partial transition function decomposable into a family of total functions $\delta_q : \{q\} \times \mathbf{\Sigma}_q \to Q$,

Symbolic Algorithm

Repeat

- create symbolic letters for all states and find evidences
- ask MQs to complete the observation table
- find observation table that is closed, consistent and evidence compatible
- ask an EQ for it
- use the counterexample to update the table
  - Discover new state (vertical expansion)
  - Refine symbolic letters (horizontal expansion)

Angluin's algorithm

1. Initialize table $T = (\Sigma, S, E, R, f)$
2. Ask MQs for ϵ and each $\sigma \in \Sigma$ to fill in T
3. **Repeat**
4.   **While** $\boldsymbol{T}$ is not closed or not consistent:
5.   **If** $\boldsymbol{T}$ not consistent **then**:
6.   find $s_1, s_2 \in S$, $a \in \Sigma$, $e \in E : f_{s_1} = f_{s_2}$ and $f(s_1 \cdot a, e) \neq f(s_2 \cdot a, e)$.
7.   add $a \cdot e$ to E
8.   fill in $\boldsymbol{T}$ by asking membership queries
9.   **If** $\boldsymbol{T}$ not closed **then**:
10.   find $s_1 \in S$, and $a \in \Sigma : f_{s_1 \cdot a} \neq f_s, \forall s \in S$
11.   add $s_1 \cdot a$ to S and $s_1 \cdot a \cdot \sigma$ to R, $\forall \sigma \in \Sigma$
12.   fill in $\boldsymbol{T}$ by asking membership queries
13.   ask EQ for $\boldsymbol{T}$
14.   **If** answer is a counter-example w **then**:
15.   add it and all prefixes to S
16.   fill in T by asking MQs
17. **Until** answer to EQ is positive
18. **Return** $\boldsymbol{T}$

Symbolic algorithm

1. Initialize table $T = (\Sigma, S, E, R, f)$
2. Ask MQs for ϵ and each $\sigma \in \Sigma$ to fill in T
3. **Repeat**
4. **While** $\boldsymbol{T}$ is not closed or not consistent:
5. **If** $\boldsymbol{T}$ not consistent **then**:
6. find $s_1, s_2 \in S$, $a \in \Sigma$, $e \in E : f_{s_1} = f_{s_2}$ and $f(s_1 \cdot a, e) \neq f(s_2 \cdot a, e)$.
7. add $a \cdot e$ to E
8. fill in $\boldsymbol{T}$ by asking membership queries
9. **If** $\boldsymbol{T}$ not closed **then**:
10. find $s_1 \in S$, and $a \in \Sigma : f_{s_1 \cdot a} \neq f_s, \forall s \in S$
11. add $s_1 \cdot a$ to S and $s_1 \cdot a \cdot \sigma$ to R, $\forall \sigma \in \Sigma$
12. fill in $\boldsymbol{T}$ by asking membership queries
13. ask EQ for $\boldsymbol{T}$
14. **If** answer is a counter-example w **then**:
15. add it and all prefixes to S
16. fill in T by asking MQs
17. **Until** answer to EQ is positive
18. **Return** $\boldsymbol{T}$

Symbolic algorithm

1. Initialize table $\boldsymbol{T} = (\Sigma, \boldsymbol{\Sigma}, \boldsymbol{S}, \boldsymbol{R}, \psi, E, \boldsymbol{f}, \mu)$
2. Ask MQs for ϵ and each $\sigma \in \Sigma$ to fill in T
3. **Repeat**
4. **While** $\boldsymbol{T}$ is not closed or not consistent:
5. **If** $\boldsymbol{T}$ not consistent **then**:
6. find $s_1, s_2 \in S$, $a \in \Sigma$, $e \in E : f_{s_1} = f_{s_2}$ and $f(s_1 \cdot a, e) \neq f(s_2 \cdot a, e)$.
7. add $a \cdot e$ to E
8. fill in $\boldsymbol{T}$ by asking membership queries
9. **If** $\boldsymbol{T}$ not closed **then**:
10. find $s_1 \in S$, and $a \in \Sigma : f_{s_1 \cdot a} \neq f_s, \forall s \in S$
11. add $s_1 \cdot a$ to S and $s_1 \cdot a \cdot \sigma$ to R, $\forall \sigma \in \Sigma$
12. fill in $\boldsymbol{T}$ by asking membership queries
13. ask EQ for $\boldsymbol{T}$
14. **If** answer is a counter-example w **then**:
15. add it and all prefixes to S
16. fill in T by asking MQs
17. **Until** answer to EQ is positive
18. **Return** $\boldsymbol{T}$

Symbolic algorithm

1. Initialize table $\boldsymbol{T} = (\Sigma, \boldsymbol{\Sigma}, \boldsymbol{S}, \boldsymbol{R}, \psi, E, \boldsymbol{f}, \mu)$
   $\boldsymbol{\Sigma} = \boldsymbol{a}$; $\mu(\boldsymbol{a}) = a_0$; $\psi_\epsilon(a) = \boldsymbol{a}, \forall a \in \Sigma$
   $\boldsymbol{S} = \{\boldsymbol{\epsilon}\}$; $\boldsymbol{R} = \{\boldsymbol{a}\}$; $E = \{\epsilon\}$
2. Ask MQs for ϵ and each $\sigma \in \Sigma$ to fill in T
3. **Repeat**
4.   **While** $\boldsymbol{T}$ is not closed or not consistent:
5.   **If** $\boldsymbol{T}$ not consistent **then**:
6.   find $s_1, s_2 \in S$, $a \in \Sigma$, $e \in E : f_{s_1} = f_{s_2}$ and $f(s_1 \cdot a, e) \neq f(s_2 \cdot a, e)$.
7.   add $a \cdot e$ to E
8.   fill in $\boldsymbol{T}$ by asking membership queries
9.   **If** $\boldsymbol{T}$ not closed **then**:
10.   find $s_1 \in S$, and $a \in \Sigma : f_{s_1 \cdot a} \neq f_s, \forall s \in S$
11.   add $s_1 \cdot a$ to S and $s_1 \cdot a \cdot \sigma$ to R, $\forall \sigma \in \Sigma$
12.   fill in $\boldsymbol{T}$ by asking membership queries
13.   ask EQ for $\boldsymbol{T}$
14.   **If** answer is a counter-example w **then**:
15.   add it and all prefixes to S
16.   fill in T by asking MQs
17. **Until** answer to EQ is positive
18. **Return** $\boldsymbol{T}$

Symbolic algorithm

1. Initialize table $\boldsymbol{T} = (\Sigma, \boldsymbol{\Sigma}, \boldsymbol{S}, \boldsymbol{R}, \psi, E, \boldsymbol{f}, \mu)$
   $\boldsymbol{\Sigma} = \boldsymbol{a}$; $\mu(\boldsymbol{a}) = a_0$; $\psi_\epsilon(a) = \boldsymbol{a}, \forall a \in \Sigma$
   $\boldsymbol{S} = \{\boldsymbol{\epsilon}\}$; $\boldsymbol{R} = \{\boldsymbol{a}\}$; $E = \{\epsilon\}$
2. Ask MQs for ϵ and $\mu(\boldsymbol{a})$ to fill in T
3. **Repeat**
4. **While** $\boldsymbol{T}$ is not closed or not consistent:
5. **If** $\boldsymbol{T}$ not consistent **then**:
6. find $s_1, s_2 \in S$, $a \in \Sigma$, $e \in E : f_{s_1} = f_{s_2}$ and $f(s_1 \cdot a, e) \neq f(s_2 \cdot a, e)$.
7. add $a \cdot e$ to E
8. fill in $\boldsymbol{T}$ by asking membership queries
9. **If** $\boldsymbol{T}$ not closed **then**:
10. find $s_1 \in S$, and $a \in \Sigma : f_{s_1 \cdot a} \neq f_s, \forall s \in S$
11. add $s_1 \cdot a$ to S and $s_1 \cdot a \cdot \sigma$ to R, $\forall \sigma \in \Sigma$
12. fill in $\boldsymbol{T}$ by asking membership queries
13. ask EQ for $\boldsymbol{T}$
14. **If** answer is a counter-example w **then**:
15. add it and all prefixes to S
16. fill in T by asking MQs
17. **Until** answer to EQ is positive
18. **Return** $\boldsymbol{T}$

Symbolic algorithm

1. Initialize table $\boldsymbol{T} = (\Sigma, \boldsymbol{\Sigma}, \boldsymbol{S}, \boldsymbol{R}, \psi, E, \boldsymbol{f}, \mu)$

   $\boldsymbol{\Sigma} = \boldsymbol{a}$; $\mu(\boldsymbol{a}) = a_0$; $\psi_\epsilon(a) = \boldsymbol{a}, \forall a \in \Sigma$

   $\boldsymbol{S} = \{\boldsymbol{\epsilon}\}$; $\boldsymbol{R} = \{\boldsymbol{a}\}$; $E = \{\epsilon\}$
2. Ask MQs for ϵ and $\mu(\boldsymbol{a})$ to fill in T
3. **Repeat**

9. **If** $\boldsymbol{T}$ not closed **then**:
10. find $s_1 \in S$, and $a \in \Sigma : f_{s_1 \cdot a} \neq f_s, \forall s \in S$
11. add $s_1 \cdot a$ to S and $s_1 \cdot a \cdot \sigma$ to R, $\forall \sigma \in \Sigma$
12. fill in $\boldsymbol{T}$ by asking membership queries
13. ask EQ for $\boldsymbol{T}$
14. **If** answer is a counter-example w **then**:
15. add it and all prefixes to S
16. fill in T by asking MQs
17. **Until** answer to EQ is positive
18. **Return** $\boldsymbol{T}$

Symbolic algorithm

1. Initialize table $\boldsymbol{T} = (\Sigma, \boldsymbol{\Sigma}, \boldsymbol{S}, \boldsymbol{R}, \psi, E, \boldsymbol{f}, \mu)$
   $\boldsymbol{\Sigma} = \boldsymbol{a}$; $\mu(\boldsymbol{a}) = a_0$; $\psi_\epsilon(a) = \boldsymbol{a}, \forall a \in \Sigma$
   $\boldsymbol{S} = \{\boldsymbol{\epsilon}\}$; $\boldsymbol{R} = \{\boldsymbol{a}\}$; $E = \{\epsilon\}$
2. Ask MQs for ϵ and $\mu(\boldsymbol{a})$ to fill in T
3. **Repeat**
4. **If** $\boldsymbol{T}$ not closed **then**:
5. find $s_1 \in S$, and $a \in \Sigma : f_{s_1 \cdot a} \neq f_s, \forall s \in S$
6. add $s_1 \cdot a$ to S and $s_1 \cdot a \cdot \sigma$ to R, $\forall \sigma \in \Sigma$
7. fill in $\boldsymbol{T}$ by asking membership queries
8. ask EQ for $\boldsymbol{T}$
9. **If** answer is a counter-example w **then**:
10. add it and all prefixes to S
11. fill in T by asking MQs
12. **Until** answer to EQ is positive
13. **Return** $\boldsymbol{T}$

Symbolic algorithm

1. Initialize table $\boldsymbol{T} = (\Sigma, \boldsymbol{\Sigma}, \boldsymbol{S}, \boldsymbol{R}, \psi, E, \boldsymbol{f}, \mu)$
   $\boldsymbol{\Sigma} = \boldsymbol{a};\ \mu(\boldsymbol{a}) = a_0;\ \psi_\epsilon(a) = \boldsymbol{a}, \forall a \in \Sigma$
   $\boldsymbol{S} = \{\boldsymbol{\epsilon}\};\ \boldsymbol{R} = \{\boldsymbol{a}\};\ E = \{\epsilon\}$
2. Ask MQs for ϵ and $\mu(\boldsymbol{a})$ to fill in T
3. **Repeat**
4. **If** $\boldsymbol{T}$ not closed **then**:
5. make_closed($\boldsymbol{T}$)

8. ask EQ for $\boldsymbol{T}$
9. **If** answer is a counter-example w **then**:
10. add it and all prefixes to S
11. fill in T by asking MQs
12. **Until** answer to EQ is positive
13. **Return** $\boldsymbol{T}$

Symbolic algorithm

1. Initialize table $\boldsymbol{T} = (\Sigma, \boldsymbol{\Sigma}, \boldsymbol{S}, \boldsymbol{R}, \psi, E, \boldsymbol{f}, \mu)$
   $\boldsymbol{\Sigma} = \boldsymbol{a}$; $\mu(\boldsymbol{a}) = a_0$; $\psi_\epsilon(a) = \boldsymbol{a}, \forall a \in \Sigma$
   $\boldsymbol{S} = \{\boldsymbol{\epsilon}\}$; $\boldsymbol{R} = \{\boldsymbol{a}\}$; $E = \{\epsilon\}$
2. Ask MQs for ϵ and $\mu(\boldsymbol{a})$ to fill in T
3. **Repeat**
4. **If** $\boldsymbol{T}$ not closed **then**:
5. make_closed($\boldsymbol{T}$)
6. ask EQ for $\boldsymbol{T}$
7. **If** answer is a counter-example w **then**:
8. Add counter-example to the concrete sample
9. treat_counter-example(w,$\boldsymbol{T}$)
10. **Until** answer to EQ is positive
11. **Return** $\boldsymbol{T}$

Symbolic algorithm

make_closed($\boldsymbol{T}$)

1. **While** there exists $\boldsymbol{r} \in \boldsymbol{R}$ such that $\forall \boldsymbol{s} \in \boldsymbol{S}$, $\boldsymbol{f_r} \neq \boldsymbol{f_s}$
2. Let $\boldsymbol{a} \notin \boldsymbol{\Sigma}$ a new symbolic letter
3. $\boldsymbol{\Sigma'} = \boldsymbol{\Sigma} \cup \{\boldsymbol{a}\}$
4. $\mu(\boldsymbol{a}) = a_0$
5. $\psi' = \psi \cup \{\psi_{\boldsymbol{r}}\}$ with $\psi_{\boldsymbol{r}}(a) = \boldsymbol{a}$, $\forall a \in \Sigma$
6. $\boldsymbol{S'} = \boldsymbol{S} \cup \{\boldsymbol{r}\}$; $\boldsymbol{R'} = (\boldsymbol{R} \backslash \{\boldsymbol{r}\}) \cup \{\boldsymbol{r} \cdot \boldsymbol{a}\}$
7. Ask MQ for all words in $\bigcup_{e \in E} \mu(\boldsymbol{r} \cdot \boldsymbol{a}) \cdot e$
8. $\boldsymbol{T} = (\Sigma, \boldsymbol{\Sigma'}, \boldsymbol{S'}, \boldsymbol{R'}, \psi', E, \boldsymbol{f'}, \mu')$
9. **return** $\boldsymbol{T}$

Symbolic algorithm

treat_counter-example(w,$\boldsymbol{T}$)

Symbolic algorithm

treat_counter-example(w,$\boldsymbol{T}$)

1. **Repeat**:
2. Find factorization $w = u \cdot a \cdot v$, $a \in \Sigma$, u, $v \in \Sigma^*$ such that
   $\exists \boldsymbol{u} \in \boldsymbol{M}$, $u \in \mu(\boldsymbol{u})$ and $\forall \boldsymbol{u'} \in \boldsymbol{M}$, $u \cdot a \notin \mu(\boldsymbol{u'})$

Symbolic algorithm

treat_counter-example(w,$\boldsymbol{T}$)

1. **Repeat**:
2. Find factorization $w = u \cdot a \cdot v$, $a \in \Sigma$, u, $v \in \Sigma^*$ such that $\exists \boldsymbol{u} \in \boldsymbol{M}$, $u \in \mu(\boldsymbol{u})$ and $\forall \boldsymbol{u'} \in \boldsymbol{M}$, $u \cdot a \notin \mu(\boldsymbol{u'})$
3. **If $\boldsymbol{u} \in \boldsymbol{R}$ then**

Symbolic algorithm

treat_counter-example(w,$\boldsymbol{T}$)

1. **Repeat**:
2. Find factorization $w = u \cdot a \cdot v$, $a \in \Sigma$, u, $v \in \Sigma^*$ such that
   $\exists \boldsymbol{u} \in \boldsymbol{M}$, $u \in \mu(\boldsymbol{u})$ and $\forall \boldsymbol{u'} \in \boldsymbol{M}$, $u \cdot a \notin \mu(\boldsymbol{u'})$
3. **If** $\boldsymbol{u} \in \boldsymbol{R}$ **then**
4. **If** $a = a_0$

Symbolic algorithm

treat_counter-example(w,$\boldsymbol{T}$)

Repeat:

Find factorization $w = u \cdot a \cdot v$, $a \in \Sigma$, $u, v \in \Sigma^*$ such that

$$\exists \boldsymbol{u} \in \boldsymbol{M},\ u \in \mu(\boldsymbol{u}) \text{ and } \forall \boldsymbol{u'} \in \boldsymbol{M},\ u \cdot a \notin \mu(\boldsymbol{u'})$$

If $\boldsymbol{u} \in \boldsymbol{R}$ **then**

- **If** $a = a_0$
  - Create a new symbolic letter $\boldsymbol{a} \notin \boldsymbol{\Sigma}$
  - $\boldsymbol{\Sigma'} = \boldsymbol{\Sigma} \cup \{\boldsymbol{a}\}$
  - $\mu(\boldsymbol{a}) = \{a_0\}$
  - $\psi' = \psi \cup \{\psi_{\boldsymbol{u}}\}$, with $\psi_{\boldsymbol{u}}(\sigma) = \boldsymbol{a}, \forall \sigma \in \Sigma$
  - $\boldsymbol{S'} = \boldsymbol{S} \cup \{\boldsymbol{u}\}$ 1 new state
  - $\boldsymbol{R'} = (\boldsymbol{R} \backslash \boldsymbol{u}) \cup \{\boldsymbol{u} \cdot \boldsymbol{a}\}$ 1 new row
  - $E' = E \cup \{\text{suffixes of } v\}$
  - Ask membership queries for all words in $\bigcup_{e \in E'} \mu(\boldsymbol{u} \cdot \boldsymbol{a}) \cdot e$

Symbolic algorithm

treat_counter-example(w,$\boldsymbol{T}$)

1. **Repeat**:
2. Find factorization $w = u \cdot a \cdot v$, $a \in \Sigma$, u, $v \in \Sigma^*$ such that
   $\exists \boldsymbol{u} \in \boldsymbol{M}$, $u \in \mu(\boldsymbol{u})$ and $\forall \boldsymbol{u'} \in \boldsymbol{M}$, $u \cdot a \notin \mu(\boldsymbol{u'})$
3. **If** $\boldsymbol{u} \in \boldsymbol{R}$ **then**
4. **If** $a = a_0$
   ... 1 new state,1 new row

Symbolic algorithm

treat_counter-example(w,$\boldsymbol{T}$)

1. **Repeat**:
2. Find factorization $w = u \cdot a \cdot v$, $a \in \Sigma$, u, $v \in \Sigma^*$ such that
   $\exists \boldsymbol{u} \in \boldsymbol{M}$, $u \in \mu(\boldsymbol{u})$ and $\forall \boldsymbol{u'} \in \boldsymbol{M}$, $u \cdot a \notin \mu(\boldsymbol{u'})$
3. **If** $\boldsymbol{u} \in \boldsymbol{R}$ **then**
4. **If** $a = a_0$
   ... 1 new state,1 new row
5. **Else**

Symbolic algorithm

treat_counter-example(w,$\boldsymbol{T}$)

1. **Repeat:**
2. Find factorization $w = u \cdot a \cdot v$, $a \in \Sigma$, $u, v \in \Sigma^*$ such that
   $\exists \boldsymbol{u} \in \boldsymbol{M}$, $u \in \mu(\boldsymbol{u})$ and $\forall \boldsymbol{u'} \in \boldsymbol{M}$, $u \cdot a \notin \mu(\boldsymbol{u'})$
3. **If** $\boldsymbol{u} \in \boldsymbol{R}$ **then**
4. **If** $a = a_0$
   ... 1 new state,1 new row
5. **Else**
6. Create two new symbolic letters $\boldsymbol{a}, \boldsymbol{a'} \notin \boldsymbol{\Sigma}$
7. $\boldsymbol{\Sigma'} = \boldsymbol{\Sigma} \cup \{\boldsymbol{a}, \boldsymbol{a'}\}$
8. $\mu(\boldsymbol{a}) = a_0$ and $\mu(\boldsymbol{a'}) = \{a\}$
9. $\psi' = \psi \cup \{\psi_{\boldsymbol{u}}\}$, with $\psi_{\boldsymbol{u}}(\sigma) = \begin{cases} \boldsymbol{a} & \text{if } \sigma < a \\ \boldsymbol{a'} & \text{otherwise} \end{cases}$
10. $\boldsymbol{S'} = \boldsymbol{S} \cup \{\boldsymbol{u}\}$ 1new state
11. $\boldsymbol{R'} = (\boldsymbol{R} \backslash \boldsymbol{u}) \cup \{\boldsymbol{u} \cdot \boldsymbol{a}, \boldsymbol{u} \cdot \boldsymbol{a'}\}$ 2 new rows
12. $E' = E \cup \{\text{suffixes of } v\}$
13. Ask membership queries for all words in

$$\bigcup_{e \in E'} \Big(\mu(\boldsymbol{u} \cdot \boldsymbol{a}) \cup \mu(\boldsymbol{u} \cdot \boldsymbol{a'})\Big) \cdot e$$

Symbolic algorithm

treat_counter-example(w,$\boldsymbol{T}$)

1. **Repeat**:
2. Find factorization $w = u \cdot a \cdot v$, $a \in \Sigma$, u, $v \in \Sigma^*$ such that
   $\exists \boldsymbol{u} \in \boldsymbol{M}$, $u \in \mu(\boldsymbol{u})$ and $\forall \boldsymbol{u'} \in \boldsymbol{M}$, $u \cdot a \notin \mu(\boldsymbol{u'})$
3. **If** $\boldsymbol{u} \in \boldsymbol{R}$ **then**
4. **If** $a = a_0$
   ... 1 new state,1 new row
5. **Else**
   ... 1 new state,2 new rows
6. $\boldsymbol{T'} = (\Sigma, \boldsymbol{\Sigma'}, \boldsymbol{S'}, \boldsymbol{R'}, \psi', E', \boldsymbol{f'}, \mu)$

Symbolic algorithm

treat_counter-example(w,$\boldsymbol{T}$)

1. **Repeat**:
2. Find factorization $w = u \cdot a \cdot v$, $a \in \Sigma$, u, $v \in \Sigma^*$ such that
   $\exists \boldsymbol{u} \in \boldsymbol{M}$, $u \in \mu(\boldsymbol{u})$ and $\forall \boldsymbol{u'} \in \boldsymbol{M}$, $u \cdot a \notin \mu(\boldsymbol{u'})$
3. **If** $\boldsymbol{u} \in \boldsymbol{R}$ **then**
4. **If** $a = a_0$
   ... 1 new state,1 new row
5. **Else**
   ... 1 new state,2 new rows
6. $\boldsymbol{T'} = (\Sigma, \boldsymbol{\Sigma'}, \boldsymbol{S'}, \boldsymbol{R'}, \psi', E', \boldsymbol{f'}, \mu)$
7. **Else** ($\boldsymbol{u} \in \boldsymbol{S}$)

Symbolic algorithm

treat_counter-example(w,$\boldsymbol{T}$)

1. **Repeat**:
2. Find factorization $w = u \cdot a \cdot v$, $a \in \Sigma$, u, $v \in \Sigma^*$ such that
   $\exists \boldsymbol{u} \in \boldsymbol{M}$, $u \in \mu(\boldsymbol{u})$ and $\forall \boldsymbol{u'} \in \boldsymbol{M}$, $u \cdot a \notin \mu(\boldsymbol{u'})$
3. **If $\boldsymbol{u} \in \boldsymbol{R}$ then**

   ... new state
4. **Else ($\boldsymbol{u} \in \boldsymbol{S}$)**

Symbolic algorithm

treat_counter-example(w,$\boldsymbol{T}$)

1. **Repeat**:
2. Find factorization $w = u \cdot a \cdot v$, $a \in \Sigma$, u, $v \in \Sigma^*$ such that
   $\exists \boldsymbol{u} \in \boldsymbol{M}$, $u \in \mu(\boldsymbol{u})$ and $\forall \boldsymbol{u'} \in \boldsymbol{M}$, $u \cdot a \notin \mu(\boldsymbol{u'})$
3. **If $\boldsymbol{u} \in \boldsymbol{R}$ then**

   ... new state
4. **Else ($\boldsymbol{u} \in \boldsymbol{S}$)**
5. Find $\boldsymbol{a} \in \boldsymbol{\Sigma_u}$ such that $a \in [\boldsymbol{a}]$

Symbolic algorithm

treat_counter-example(w,$\boldsymbol{T}$)

1. **Repeat**:
2. Find factorization $w = u \cdot a \cdot v$, $a \in \Sigma$, u, $v \in \Sigma^*$ such that
   $\exists \boldsymbol{u} \in \boldsymbol{M}$, $u \in \mu(\boldsymbol{u})$ and $\forall \boldsymbol{u'} \in \boldsymbol{M}$, $u \cdot a \notin \mu(\boldsymbol{u'})$
3. **If** $\boldsymbol{u} \in \boldsymbol{R}$ **then**

   ... new state
4. **Else** $(\boldsymbol{u} \in \boldsymbol{S})$
5. Find $\boldsymbol{a} \in \boldsymbol{\Sigma_u}$ such that $a \in [\boldsymbol{a}]$
6. **If** there is no $\boldsymbol{a}' \in \boldsymbol{\Sigma} : f_a = f_{\mu(\boldsymbol{a}')}$ on E **then**

Symbolic algorithm

treat_counter-example(w,$\boldsymbol{T}$)

1. **Repeat**:
2. Find factorization $w = u \cdot a \cdot v$, $a \in \Sigma$, u, $v \in \Sigma^*$ such that
   $\exists \boldsymbol{u} \in \boldsymbol{M}$, $u \in \mu(\boldsymbol{u})$ and $\forall \boldsymbol{u'} \in \boldsymbol{M}$, $u \cdot a \notin \mu(\boldsymbol{u'})$
3. **If** $\boldsymbol{u} \in \boldsymbol{R}$ **then**
   ... new state
4. **Else** ($\boldsymbol{u} \in \boldsymbol{S}$)
5. Find $\boldsymbol{a} \in \boldsymbol{\Sigma_u}$ such that $a \in [\boldsymbol{a}]$
6. **If** there is no $\boldsymbol{a}' \in \boldsymbol{\Sigma}$: $f_a = f_{\mu(\boldsymbol{a}')}$ on E **then**
7. Create a new symbolic letter $\boldsymbol{b'} \notin \boldsymbol{\Sigma}$; $\boldsymbol{\Sigma'} = \boldsymbol{\Sigma} \cup \{\boldsymbol{a'}\}$
8. $\mu(\boldsymbol{a'}) = \{a\}$; $\boldsymbol{R'} = \boldsymbol{R} \cup \{\boldsymbol{ua'}\}$
9. Ask membership queries for all words in $\bigcup_{e \in E} \mu(\boldsymbol{u} \cdot \boldsymbol{a'}) \cdot e$

Symbolic algorithm

treat_counter-example(w,$\boldsymbol{T}$)

1. **Repeat**:
2. Find factorization $w = u \cdot a \cdot v$, $a \in \Sigma$, u, $v \in \Sigma^*$ such that
   $\exists \boldsymbol{u} \in \boldsymbol{M}$, $u \in \mu(\boldsymbol{u})$ and $\forall \boldsymbol{u'} \in \boldsymbol{M}$, $u \cdot a \notin \mu(\boldsymbol{u'})$
3. **If** $\boldsymbol{u} \in \boldsymbol{R}$ **then**

   ... new state
4. **Else** ($\boldsymbol{u} \in \boldsymbol{S}$)
5. Find $\boldsymbol{a} \in \boldsymbol{\Sigma_u}$ such that $a \in [\boldsymbol{a}]$
6. **If** there is no $\boldsymbol{a}' \in \boldsymbol{\Sigma}$: $f_a = f_{\mu(\boldsymbol{a}')}$ on E **then**
7. Create a new symbolic letter $\boldsymbol{b'} \notin \boldsymbol{\Sigma}$; $\boldsymbol{\Sigma'} = \boldsymbol{\Sigma} \cup \{\boldsymbol{a'}\}$
8. $\mu(\boldsymbol{a'}) = \{a\}$; $\boldsymbol{R'} = \boldsymbol{R} \cup \{\boldsymbol{ua'}\}$
9. Ask membership queries for all words in $\bigcup_{e \in E} \mu(\boldsymbol{u} \cdot \boldsymbol{a'}) \cdot e$
10. $$\psi_{\boldsymbol{u}}(\sigma) = \begin{cases} \psi_{\boldsymbol{u}}(\sigma) & \text{if } \sigma \notin [\boldsymbol{a}] \\ \boldsymbol{a} & \text{if } \sigma \in [\boldsymbol{a}] \text{ and } \sigma < a \\ \boldsymbol{a'} & \text{otherwise} \end{cases}$$ refinement

Symbolic algorithm

treat_counter-example(w,$\boldsymbol{T}$)

1. **Repeat**:
2. Find factorization $w = u \cdot a \cdot v$, $a \in \Sigma$, $u, v \in \Sigma^*$ such that
   $\exists \boldsymbol{u} \in \boldsymbol{M}$, $u \in \mu(\boldsymbol{u})$ and $\forall \boldsymbol{u'} \in \boldsymbol{M}$, $u \cdot a \notin \mu(\boldsymbol{u'})$
3. **If** $\boldsymbol{u} \in \boldsymbol{R}$ **then**
   ... new state
4. **Else** $(\boldsymbol{u} \in \boldsymbol{S})$
5. Find $\boldsymbol{a} \in \boldsymbol{\Sigma_u}$ such that $a \in [\boldsymbol{a}]$
6. **If** there is no $\boldsymbol{a}' \in \boldsymbol{\Sigma}$: $f_a = f_{\mu(\boldsymbol{a}')}$ on E **then**
7. Create a new symbolic letter $\boldsymbol{b'} \notin \boldsymbol{\Sigma}$; $\boldsymbol{\Sigma'} = \boldsymbol{\Sigma} \cup \{\boldsymbol{a'}\}$
8. $\mu(\boldsymbol{a'}) = \{a\}$; $\boldsymbol{R'} = \boldsymbol{R} \cup \{\boldsymbol{ua'}\}$
9. Ask membership queries for all words in $\bigcup_{e \in E} \mu(\boldsymbol{u} \cdot \boldsymbol{a'}) \cdot e$
10. $\psi_{\boldsymbol{u}}(\sigma) = \begin{cases} \psi_{\boldsymbol{u}}(\sigma) & \text{if } \sigma \notin [\boldsymbol{a}] \\ \boldsymbol{a} & \text{if } \sigma \in [\boldsymbol{a}] \text{ and } \sigma < a \\ \boldsymbol{a'} & \text{otherwise} \end{cases}$ refinement
11. $\boldsymbol{T} = (\Sigma, \boldsymbol{\Sigma'}, \boldsymbol{S}, \boldsymbol{R'}, \psi, E, \boldsymbol{f'}, \mu)$

Symbolic algorithm

treat_counter-example(w,$\boldsymbol{T}$)

1. **Repeat**:
2. Find factorization $w = u \cdot a \cdot v$, $a \in \Sigma$, u, $v \in \Sigma^*$ such that
   $\exists \boldsymbol{u} \in \boldsymbol{M}$, $u \in \mu(\boldsymbol{u})$ and $\forall \boldsymbol{u'} \in \boldsymbol{M}$, $u \cdot a \notin \mu(\boldsymbol{u'})$
3. **If $\boldsymbol{u} \in \boldsymbol{R}$ then**
   ... new state
4. **Else ($\boldsymbol{u} \in \boldsymbol{S}$)**
   ... refinement

Symbolic algorithm

treat_counter-example(w,$\boldsymbol{T}$)

1. **Repeat**:
2. Find factorization $w = u \cdot a \cdot v$, $a \in \Sigma$, u, $v \in \Sigma^*$ such that
   $\exists \boldsymbol{u} \in \boldsymbol{M}$, $u \in \mu(\boldsymbol{u})$ and $\forall \boldsymbol{u'} \in \boldsymbol{M}$, $u \cdot a \notin \mu(\boldsymbol{u'})$
3. **If** $\boldsymbol{u} \in \boldsymbol{R}$ **then**
   ... new state
4. **Else** ($\boldsymbol{u} \in \boldsymbol{S}$)
   ... refinement
5. **If** $\boldsymbol{T}$ is not closed **then**
6. make_closed($\boldsymbol{T}$)

Symbolic algorithm

treat_counter-example(w,$\boldsymbol{T}$)

1. **Repeat**:
2. Find factorization $w = u \cdot a \cdot v$, $a \in \Sigma$, u, $v \in \Sigma^*$ such that
   $\exists \boldsymbol{u} \in \boldsymbol{M}$, $u \in \mu(\boldsymbol{u})$ and $\forall \boldsymbol{u'} \in \boldsymbol{M}$, $u \cdot a \notin \mu(\boldsymbol{u'})$
3. **If** $\boldsymbol{u} \in \boldsymbol{R}$ **then**

   ... new state
4. **Else** ($\boldsymbol{u} \in \boldsymbol{S}$)

   ... refinement
5. **If** $\boldsymbol{T}$ is not closed **then**
6. make_closed($\boldsymbol{T}$)
7. **Until** $\boldsymbol{T}$ classifies w correctly
8. **return** $\boldsymbol{T}$

example $(\Sigma = \{1, 2, 3, \dots, 100\})$

observation table

$\psi = \{\psi_s\}_{s \in S}$

hypothesis automaton

	ϵ
ϵ	

example ($\Sigma = \{1, 2, 3, \ldots, 100\}$)

observation table $\psi = \{\psi_s\}_{s \in S}$ hypothesis automaton

	ϵ
ϵ	$-$

example $(\Sigma = \{1, 2, 3, \ldots, 100\})$

observation table

	ϵ
ϵ	$-$

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

hypothesis automaton

example ($\Sigma = \{1, 2, 3, \ldots, 100\}$)

observation table

	ϵ
ϵ	$-$
1 $\boldsymbol{a_0}$	

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 100\}$

hypothesis automaton

example $(\Sigma = \{1, 2, 3, \ldots, 100\})$

observation table

	ϵ
ϵ	−
$\overset{1}{\boldsymbol{a_0}}$	−

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 100\}$

hypothesis automaton

example ($\Sigma = \{1, 2, 3, \ldots, 100\}$)

observation table

	ϵ
ϵ	$-$
$\boldsymbol{a_0}$ (1)	$-$

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 100\}$

hypothesis automaton

example $(\Sigma = \{1, 2, 3, \dots, 100\})$

observation table

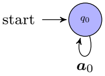
	ϵ
ϵ	$-$
$\boldsymbol{a_0}$ (1)	$-$

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \dots, 100\}$

hypothesis automaton



example ($\Sigma = \{1, 2, 3, \ldots, 100\}$)

observation table

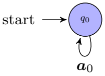
	ϵ
ϵ	—
$\overset{1}{\boldsymbol{a_0}}$	—

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 100\}$

hypothesis automaton



counterexample + 23

example ($\Sigma = \{1, 2, 3, \dots, 100\}$)

observation table

	ϵ
ϵ	$-$
$\overset{1}{\boldsymbol{a}_0}$	$-$
$\overset{23}{\boldsymbol{a}_1}$	$+$

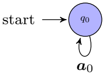
$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \dots, 23\}$

$[\boldsymbol{a}_1] = \{23, \dots, 100\}$

hypothesis automaton



counterexample $+\ 23$

example $(\Sigma = \{1, 2, 3, \ldots, 100\})$

observation table

	ϵ
ϵ	$-$
1 $\boldsymbol{a_0}$	$-$
23 $\boldsymbol{a_1}$	$+$

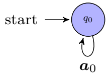
$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 23\}$

$[\boldsymbol{a}_1] = \{23, \ldots, 100\}$

hypothesis automaton



counterexample $+$ 23

example ($\Sigma = \{1, 2, 3, \ldots, 100\}$)

observation table

	ϵ
ϵ	$-$
$\boldsymbol{a_1}$ (23)	$+$
$\boldsymbol{a_0}$ (1)	$-$

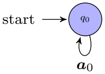
$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 23\}$

$[\boldsymbol{a}_1] = \{23, \ldots, 100\}$

hypothesis automaton



counterexample $+$ 23

example $(\Sigma = \{1, 2, 3, \dots, 100\})$

observation table

	ϵ
ϵ	$-$
$\overset{23}{\boldsymbol{a_1}}$	$+$
$\overset{1}{\boldsymbol{a_0}}$	$-$
$\overset{23}{\boldsymbol{a_1}}\overset{1}{\boldsymbol{a_2}}$	

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

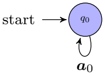
$[\boldsymbol{a}_0] = \{1, 2, \dots, 23\}$

$[\boldsymbol{a}_1] = \{23, \dots, 100\}$

$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \dots, 100\}$

hypothesis automaton



counterexample $+\ 23$

example $(\Sigma = \{1, 2, 3, \ldots, 100\})$

observation table

	ϵ
ϵ	$-$
$\overset{23}{a_1}$	$+$
$\overset{1}{a_0}$	$-$
$\overset{23}{a_1}\overset{1}{a_2}$	$+$

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

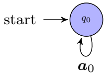
$[\boldsymbol{a}_0] = \{1, 2, \ldots, 23\}$

$[\boldsymbol{a}_1] = \{23, \ldots, 100\}$

$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \ldots, 100\}$

hypothesis automaton



counterexample $+\ 23$

example $(\Sigma = \{1, 2, 3, \dots, 100\})$

observation table

	ϵ
ϵ	$-$
$\overset{23}{\boldsymbol{a}_1}$	$+$
$\overset{1}{\boldsymbol{a}_0}$	$-$
$\overset{23}{\boldsymbol{a}_1}\overset{1}{\boldsymbol{a}_2}$	$+$

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

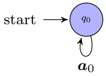
$[\boldsymbol{a}_0] = \{1, 2, \dots, 23\}$

$[\boldsymbol{a}_1] = \{23, \dots, 100\}$

$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \dots, 100\}$

hypothesis automaton



example $(\Sigma = \{1, 2, 3, \ldots, 100\})$

observation table

	ϵ
ϵ	$-$
$\overset{23}{\boldsymbol{a_1}}$	$+$
$\overset{1}{\boldsymbol{a_0}}$	$-$
$\overset{23}{\boldsymbol{a_1}}\overset{1}{\boldsymbol{a_2}}$	$+$

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

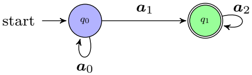
$[\boldsymbol{a}_0] = \{1, 2, \ldots, 23\}$

$[\boldsymbol{a}_1] = \{23, \ldots, 100\}$

$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \ldots, 100\}$

hypothesis automaton



example $(\Sigma = \{1, 2, 3, \ldots, 100\})$

observation table

	ϵ
ϵ	$-$
$\overset{23}{\boldsymbol{a_1}}$	$+$
$\overset{1}{\boldsymbol{a_0}}$	$-$
$\overset{23}{\boldsymbol{a_1}}\overset{1}{\boldsymbol{a_2}}$	$+$

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

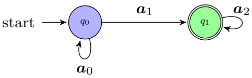
$[\boldsymbol{a}_0] = \{1, 2, \ldots, 23\}$

$[\boldsymbol{a}_1] = \{23, \ldots, 100\}$

$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \ldots, 100\}$

hypothesis automaton



counterexample $- 65$

example ($\Sigma = \{1, 2, 3, \ldots, 100\}$)

observation table

	ϵ
ϵ	$-$
$\overset{23}{\boldsymbol{a_1}}$	$+$
$\overset{1}{\boldsymbol{a_0}}$	$-$
$\overset{23}{\boldsymbol{a_1}}\overset{1}{\boldsymbol{a_2}}$	$+$
$\overset{65}{\boldsymbol{a_3}}$	$-$

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 23\}$

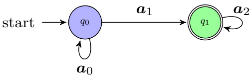
$[\boldsymbol{a}_1] = \{23, \ldots, 65\}$

$[\boldsymbol{a}_3] = \{65, \ldots, 100\}$

$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \ldots, 100\}$

hypothesis automaton



counterexample $- 65$

example $(\Sigma = \{1, 2, 3, \ldots, 100\})$

observation table

	ϵ
ϵ	$-$
$\overset{23}{a_1}$	$+$
$\overset{1}{a_0}$	$-$
$\overset{23}{a_1}\overset{1}{a_2}$	$+$
$\overset{65}{a_3}$	$-$

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[a_0] = \{1, 2, \ldots, 23\}$

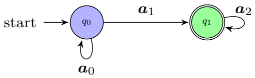
$[a_1] = \{23, \ldots, 65\}$

$[a_3] = \{65, \ldots, 100\}$

ψ_{a_1}

$[a_2] = \{1, 2, \ldots, 100\}$

hypothesis automaton



counterexample $-$ 65

example $(\Sigma = \{1, 2, 3, \ldots, 100\})$

observation table

	ϵ
ϵ	$-$
$\overset{23}{\boldsymbol{a_1}}$	$+$
$\overset{1}{\boldsymbol{a_0}}$	$-$
$\overset{23}{\boldsymbol{a_1}}\overset{1}{\boldsymbol{a_2}}$	$+$
$\overset{65}{\boldsymbol{a_3}}$	$-$

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 23\}$

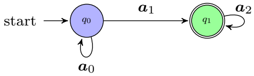
$[\boldsymbol{a}_1] = \{23, \ldots, 65\}$

$[\boldsymbol{a}_3] = \{65, \ldots, 100\}$

$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \ldots, 100\}$

hypothesis automaton



counterexample $-$ 65

example ($\Sigma = \{1, 2, 3, \ldots, 100\}$)

observation table

	ϵ
ϵ	$-$
$\overset{23}{\boldsymbol{a}_1}$	$+$
$\overset{1}{\boldsymbol{a}_0}$	$-$
$\overset{23}{\boldsymbol{a}_1}\overset{1}{\boldsymbol{a}_2}$	$+$
$\overset{65}{\boldsymbol{a}_3}$	$-$

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 23\}$

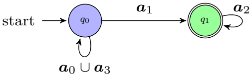
$[\boldsymbol{a}_1] = \{23, \ldots, 65\}$

$[\boldsymbol{a}_3] = \{65, \ldots, 100\}$

$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \ldots, 100\}$

hypothesis automaton



example $(\Sigma = \{1, 2, 3, \ldots, 100\})$

observation table

	ϵ
ϵ	$-$
$\boldsymbol{a_1}$ (23)	$+$
$\boldsymbol{a_0}$ (1)	$-$
$\boldsymbol{a_1 a_2}$ (23 1)	$+$
$\boldsymbol{a_3}$ (65)	$-$

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 23\}$

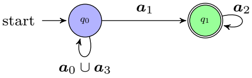
$[\boldsymbol{a}_1] = \{23, \ldots, 65\}$

$[\boldsymbol{a}_3] = \{65, \ldots, 100\}$

$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \ldots, 100\}$

hypothesis automaton



counterexample -1 23

example ($\Sigma = \{1, 2, 3, \ldots, 100\}$)

observation table

	ϵ
ϵ	$-$
$\boldsymbol{a_0}$ (1)	$-$
$\boldsymbol{a_1}$ (23)	$+$
$\boldsymbol{a_1 a_2}$ (23 1)	$+$
$\boldsymbol{a_3}$ (65)	$-$

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 23\}$

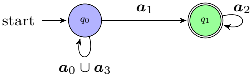
$[\boldsymbol{a}_1] = \{23, \ldots, 65\}$

$[\boldsymbol{a}_3] = \{65, \ldots, 100\}$

$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \ldots, 100\}$

hypothesis automaton



counterexample $-1\ 23$

example $(\Sigma = \{1, 2, 3, \ldots, 100\})$

observation table

	ϵ
ϵ	$-$
$\overset{1}{a_0}$	$-$
$\overset{23}{a_1}$	$+$
$\overset{23}{a_1}\overset{1}{a_2}$	$+$
$\overset{65}{a_3}$	$-$
$\overset{1}{a_0}\overset{1}{a_4}$	

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 23\}$

$[\boldsymbol{a}_1] = \{23, \ldots, 65\}$

$[\boldsymbol{a}_3] = \{65, \ldots, 100\}$

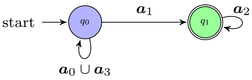
$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \ldots, 100\}$

$\psi_{\boldsymbol{a}_0}$

$[\boldsymbol{a}_4] = \{1, 2, \ldots, 100\}$

hypothesis automaton



counterexample $-1\ 23$

example ($\Sigma = \{1, 2, 3, \ldots, 100\}$)

observation table

	ϵ
ϵ	$-$
$\overset{1}{a_0}$	$-$
$\overset{23}{a_1}$	$+$
$\overset{23}{a_1}\overset{1}{a_2}$	$+$
$\overset{65}{a_3}$	$-$
$\overset{1}{a_0}\overset{1}{a_4}$	$-$

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 23\}$

$[\boldsymbol{a}_1] = \{23, \ldots, 65\}$

$[\boldsymbol{a}_3] = \{65, \ldots, 100\}$

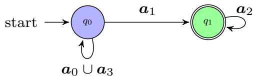
$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \ldots, 100\}$

$\psi_{\boldsymbol{a}_0}$

$[\boldsymbol{a}_4] = \{1, 2, \ldots, 100\}$

hypothesis automaton



counterexample $-1\ 23$

example $(\Sigma = \{1, 2, 3, \dots, 100\})$

observation table

	ϵ	23
ϵ	$-$	
$\overset{1}{\boldsymbol{a_0}}$	$-$	
$\overset{23}{\boldsymbol{a_1}}$	$+$	
$\overset{23}{\boldsymbol{a_1}}\overset{1}{\boldsymbol{a_2}}$	$+$	
$\overset{65}{\boldsymbol{a_3}}$	$-$	
$\overset{1}{\boldsymbol{a_0}}\overset{1}{\boldsymbol{a_4}}$	$-$	

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \dots, 23\}$

$[\boldsymbol{a}_1] = \{23, \dots, 65\}$

$[\boldsymbol{a}_3] = \{65, \dots, 100\}$

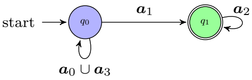
$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \dots, 100\}$

$\psi_{\boldsymbol{a}_0}$

$[\boldsymbol{a}_4] = \{1, 2, \dots, 100\}$

hypothesis automaton



counterexample $-1\ 23$

example ($\Sigma = \{1, 2, 3, \ldots, 100\}$)

observation table

	ϵ	23
ϵ	$-$	$+$
$\overset{1}{\boldsymbol{a_0}}$	$-$	
$\overset{23}{\boldsymbol{a_1}}$	$+$	
$\overset{23}{\boldsymbol{a_1}}\overset{1}{\boldsymbol{a_2}}$	$+$	
$\overset{65}{\boldsymbol{a_3}}$	$-$	
$\overset{1}{\boldsymbol{a_0}}\overset{1}{\boldsymbol{a_4}}$	$-$	

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 23\}$

$[\boldsymbol{a}_1] = \{23, \ldots, 65\}$

$[\boldsymbol{a}_3] = \{65, \ldots, 100\}$

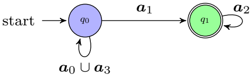
$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \ldots, 100\}$

$\psi_{\boldsymbol{a}_0}$

$[\boldsymbol{a}_4] = \{1, 2, \ldots, 100\}$

hypothesis automaton



counterexample $-1\ 23$

example $(\Sigma = \{1, 2, 3, \ldots, 100\})$

observation table

	ϵ	23
ϵ	$-$	$+$
$\overset{1}{\boldsymbol{a_0}}$	$-$	$-$
$\overset{23}{\boldsymbol{a_1}}$	$+$	$+$
$\overset{23}{\boldsymbol{a_1}}\overset{1}{\boldsymbol{a_2}}$	$+$	$+$
$\overset{65}{\boldsymbol{a_3}}$	$-$	$-$
$\overset{1}{\boldsymbol{a_0}}\overset{1}{\boldsymbol{a_4}}$	$-$	$-$

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 23\}$

$[\boldsymbol{a}_1] = \{23, \ldots, 65\}$

$[\boldsymbol{a}_3] = \{65, \ldots, 100\}$

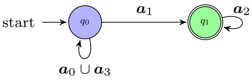
$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \ldots, 100\}$

$\psi_{\boldsymbol{a}_0}$

$[\boldsymbol{a}_4] = \{1, 2, \ldots, 100\}$

hypothesis automaton



counterexample $-1\ 23$

example $(\Sigma = \{1, 2, 3, \ldots, 100\})$

observation table

	ϵ	23
ϵ	$-$	$+$
$\overset{1}{\boldsymbol{a_0}}$	$-$	$-$
$\overset{23}{\boldsymbol{a_1}}$	$+$	$+$
$\overset{23}{\boldsymbol{a_1}}\overset{1}{\boldsymbol{a_2}}$	$+$	$+$
$\overset{65}{\boldsymbol{a_3}}$	$-$	$-$
$\overset{1}{\boldsymbol{a_0}}\overset{1}{\boldsymbol{a_4}}$	$-$	$-$

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 23\}$

$[\boldsymbol{a}_1] = \{23, \ldots, 65\}$

$[\boldsymbol{a}_3] = \{65, \ldots, 100\}$

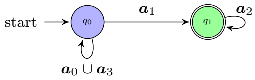
$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \ldots, 100\}$

$\psi_{\boldsymbol{a}_0}$

$[\boldsymbol{a}_4] = \{1, 2, \ldots, 100\}$

hypothesis automaton



example ($\Sigma = \{1, 2, 3, \ldots, 100\}$)

observation table

	ϵ	23
ϵ	$-$	$+$
$\overset{1}{\boldsymbol{a}_0}$	$-$	$-$
$\overset{23}{\boldsymbol{a}_1}$	$+$	$+$
$\overset{23}{\boldsymbol{a}_1}\overset{1}{\boldsymbol{a}_2}$	$+$	$+$
$\overset{65}{\boldsymbol{a}_3}$	$-$	$-$
$\overset{1}{\boldsymbol{a}_0}\overset{1}{\boldsymbol{a}_4}$	$-$	$-$

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 23\}$

$[\boldsymbol{a}_1] = \{23, \ldots, 65\}$

$[\boldsymbol{a}_3] = \{65, \ldots, 100\}$

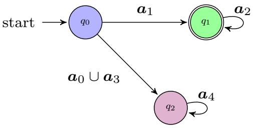
$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \ldots, 100\}$

$\psi_{\boldsymbol{a}_0}$

$[\boldsymbol{a}_4] = \{1, 2, \ldots, 100\}$

hypothesis automaton



example $(\Sigma = \{1, 2, 3, \ldots, 100\})$

observation table

	ϵ	23
ϵ	$-$	$+$
$\overset{1}{\boldsymbol{a_0}}$	$-$	$-$
$\overset{23}{\boldsymbol{a_1}}$	$+$	$+$
$\overset{23}{\boldsymbol{a_1}}\overset{1}{\boldsymbol{a_2}}$	$+$	$+$
$\overset{65}{\boldsymbol{a_3}}$	$-$	$-$
$\overset{1}{\boldsymbol{a_0}}\overset{1}{\boldsymbol{a_4}}$	$-$	$-$

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 23\}$

$[\boldsymbol{a}_1] = \{23, \ldots, 65\}$

$[\boldsymbol{a}_3] = \{65, \ldots, 100\}$

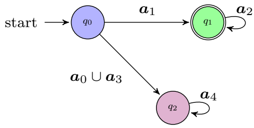
$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \ldots, 100\}$

$\psi_{\boldsymbol{a}_0}$

$[\boldsymbol{a}_4] = \{1, 2, \ldots, 100\}$

hypothesis automaton



True

example ($\Sigma = \{1, 2, 3, \ldots, 100\}$)

observation table

	ϵ	23
ϵ	−	+
$\boldsymbol{a}_0$ (1)	−	−
$\boldsymbol{a}_1$ (23)	+	+
$\boldsymbol{a}_1\boldsymbol{a}_2$ (23 1)	+	+
$\boldsymbol{a}_3$ (65)	−	−
$\boldsymbol{a}_0\boldsymbol{a}_4$ (1 1)	−	−

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 23\}$

$[\boldsymbol{a}_1] = \{23, \ldots, 65\}$

$[\boldsymbol{a}_3] = \{65, \ldots, 100\}$

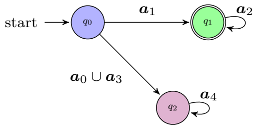
$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \ldots, 100\}$

$\psi_{\boldsymbol{a}_0}$

$[\boldsymbol{a}_4] = \{1, 2, \ldots, 100\}$

hypothesis automaton



True

$$M = \{\epsilon, 1, 23, 65, 1\,1, 1\,23, 23\,1, 23\,23, 65\,23, 1\,1\,23, 23\,1\,23\}$$

$$|M| = 11,\ |MQ| = 8,\ |EQ| = 4$$

example $(\Sigma = \{1, 2, 3, \ldots, 100\})$

observation table

	ϵ	23
ϵ	$-$	$+$
$\overset{1}{\boldsymbol{a}_0}$	$-$	$-$
$\overset{23}{\boldsymbol{a}_1}$	$+$	$+$
$\overset{23}{\boldsymbol{a}_1}\overset{1}{\boldsymbol{a}_2}$	$+$	$+$
$\overset{65}{\boldsymbol{a}_3}$	$-$	$-$
$\overset{1}{\boldsymbol{a}_0}\overset{1}{\boldsymbol{a}_4}$	$-$	$-$

$\psi = \{\psi_s\}_{s \in S}$

ψ_ϵ

$[\boldsymbol{a}_0] = \{1, 2, \ldots, 23\}$

$[\boldsymbol{a}_1] = \{23, \ldots, 65\}$

$[\boldsymbol{a}_3] = \{65, \ldots, 100\}$

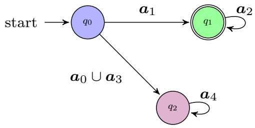
$\psi_{\boldsymbol{a}_1}$

$[\boldsymbol{a}_2] = \{1, 2, \ldots, 100\}$

$\psi_{\boldsymbol{a}_0}$

$[\boldsymbol{a}_4] = \{1, 2, \ldots, 100\}$

hypothesis automaton



True

$$M = \{\epsilon, 1, 23, 65, 1\,1, 1\,23, 23\,1, 23\,23, 65\,23, 1\,1\,23, 23\,1\,23\}$$

$|M| = 200,\ |MQ| = 199,\ |EQ| = 2$

$$|M| = 11,\ |MQ| = 8,\ |EQ| = 4$$

Conclusion

- learning languages over large or infinite alphabets
- finite number of intervals
- without using variables

Future work

- PAC learning
  - the counterexample is not smallest
  - learn without equivalence queries
- learning time series
- learning languages over boolean vectors
- ...

Future work

- PAC learning
  - the counterexample is not smallest
  - learn without equivalence queries
- learning time series
- learning languages over boolean vectors
- ...

Thank you!