Automata and Logic

Aart Middeldorp and Samuel Frontull

Outline

Definitions

- regular expression α over alphabet Σ:

$$a \in \Sigma \qquad \epsilon \qquad \varnothing \qquad \beta + \gamma \qquad \beta\gamma \qquad \beta^*$$

- set of strings $L(\alpha) \subseteq \Sigma^*$ matched by regular expression α:

$$L(a) = \{a\} \qquad L(\varnothing) = \varnothing \qquad L(\beta\gamma) = L(\beta)L(\gamma)$$

$$L(\epsilon) = \{\epsilon\} \qquad L(\beta + \gamma) = L(\beta) \cup L(\gamma) \qquad L(\beta^*) = L(\beta)^*$$

- regular expressions α and β are equivalent $(\alpha \equiv \beta)$ if $L(\alpha) = L(\beta)$

Theorem

finite automata and regular expressions are equivalent:

$$\text{for all } A \subseteq \Sigma^* \quad A \text{ is regular} \quad \Longleftrightarrow \quad A = L(\alpha) \text{ for some regular expression } \alpha$$

Definitions

- homomorphism is mapping $h\colon \Sigma^* \to \Gamma^*$ such that $h(\epsilon) = \epsilon$ and $h(xy) = h(x)h(y)$
- if $A \subseteq \Sigma^*$ then $h(A) = \{h(x) \mid x \in A\} \subseteq \Gamma^*$ "image of A under h"
- if $B \subseteq \Gamma^*$ then $h^{-1}(B) = \{x \mid h(x) \in B\} \subseteq \Sigma^*$ "preimage of B under h"

Theorem

regular sets are effectively closed under homomorphic image and preimage

Theorem

problems

instance: DFA M and string x	instance: DFA M	instance: DFAs M and N
question: $x \in L(M)$?	question: $L(M) = \varnothing$?	question: $L(M) = L(N)$?

are decidable

Automata

- (deterministic, nondeterministic, alternating) finite automata
- regular expressions
- (alternating) Büchi automata

Logic

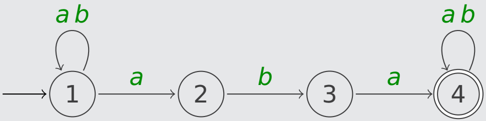
- (weak) monadic second–order logic
- Presburger arithmetic
- linear–time temporal logic

Outline

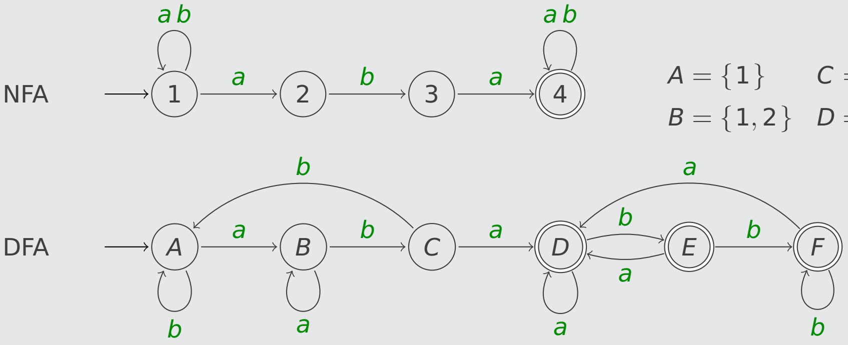
1. Summary of Previous Lecture
2. **Minimization**
3. Intermezzo
4. Weak Monadic Second–Order Logic
5. Further Reading

Example ❶

NFA



Example ❶

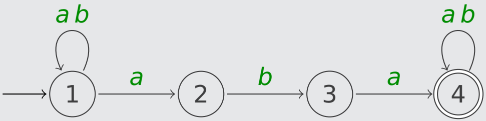


$A = \{1\}$ $\quad C = \{1,3\}$ $\quad E = \{1,3,4\}$

$B = \{1,2\}$ $\quad D = \{1,2,4\}$ $\quad F = \{1,4\}$

Example ①

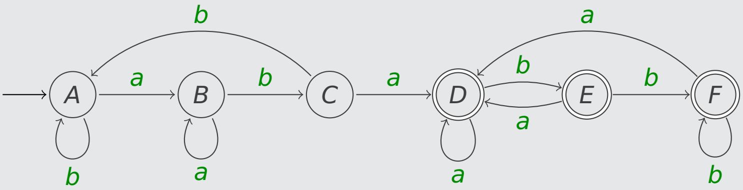
NFA



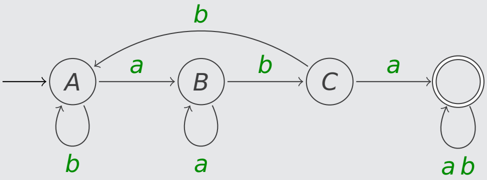
$A = \{1\}$ $C = \{1,3\}$ $E = \{1,3,4\}$

$B = \{1,2\}$ $D = \{1,2,4\}$ $F = \{1,4\}$

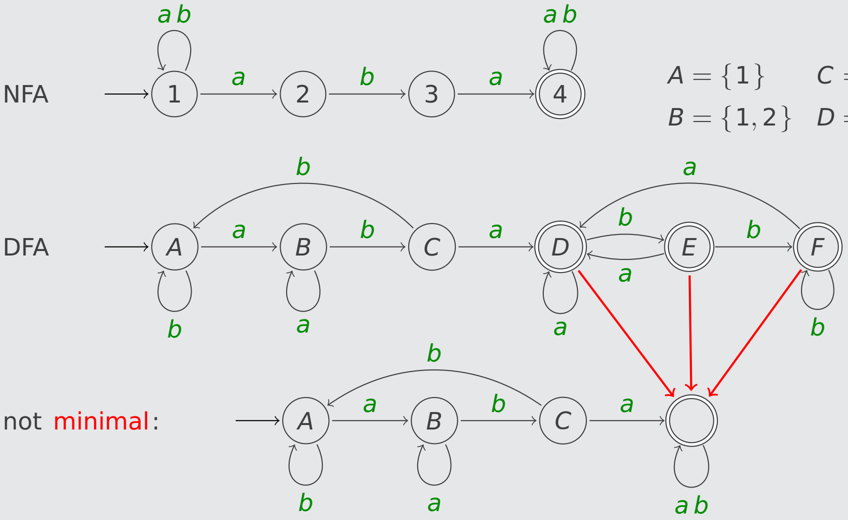
DFA



not minimal:



Example ❶

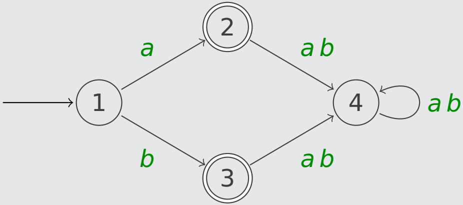


$A = \{1\}$ $\quad C = \{1,3\}$ $\quad E = \{1,3,4\}$

$B = \{1,2\}$ $\quad D = \{1,2,4\}$ $\quad F = \{1,4\}$

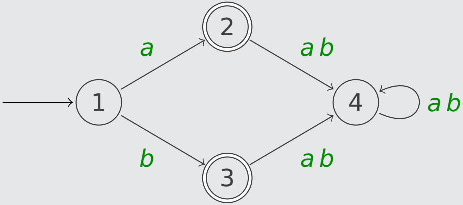
Example ❷

DFA

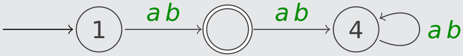


Example ❷

DFA

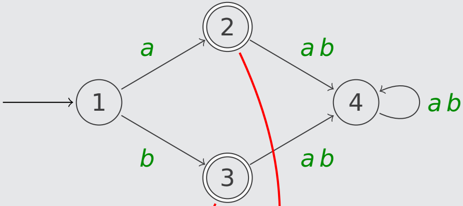


not minimal: states 2 and 3 can be collapsed

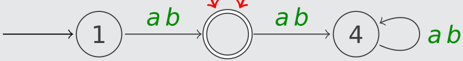


Example ❷

DFA

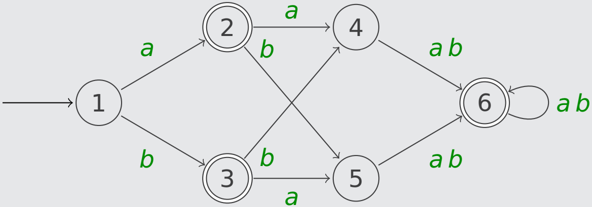


not minimal: states 2 and 3 can be collapsed



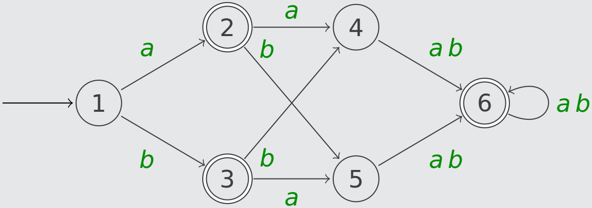
Example ③

DFA

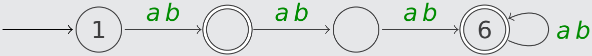


Example ③

DFA

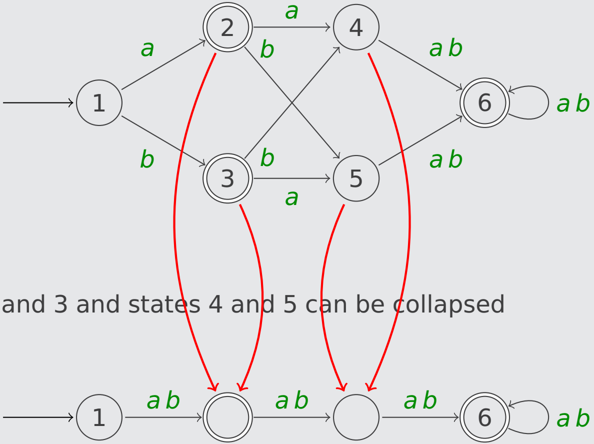


not minimal: states 2 and 3 and states 4 and 5 can be collapsed



Example ③

DFA



not minimal: states 2 and 3 and states 4 and 5 can be collapsed

Definitions

DFA $M = (Q, \Sigma, \delta, s, F)$

- state p is inaccessible if $\widehat{\delta}(s, x) \neq p$ for all $x \in \Sigma^*$

Definitions

DFA $M = (Q, \Sigma, \delta, s, F)$

- state p is inaccessible if $\widehat{\delta}(s,x) \neq p$ for all $x \in \Sigma^*$
- states p and q are distinguishable if

$$\big(\widehat{\delta}(p,x) \in F \wedge \widehat{\delta}(q,x) \notin F\big) \ \vee \ \big(\widehat{\delta}(p,x) \notin F \wedge \widehat{\delta}(q,x) \in F\big)$$

  for some $x \in \Sigma^*$

Definitions

DFA $M = (Q, \Sigma, \delta, s, F)$

- state p is inaccessible if $\widehat{\delta}(s, x) \neq p$ for all $x \in \Sigma^*$
- states p and q are distinguishable if

$$\big(\widehat{\delta}(p,x) \in F \wedge \widehat{\delta}(q,x) \notin F\big) \ \vee \ \big(\widehat{\delta}(p,x) \notin F \wedge \widehat{\delta}(q,x) \in F\big)$$

  for some $x \in \Sigma^*$

Minimization Algorithm

DFA $M = (Q, \Sigma, \delta, s, F)$

① remove inaccessible states

Definitions

DFA $M = (Q, \Sigma, \delta, s, F)$

- state p is inaccessible if $\widehat{\delta}(s, x) \neq p$ for all $x \in \Sigma^*$
- states p and q are distinguishable if

$$\big(\widehat{\delta}(p,x) \in F \wedge \widehat{\delta}(q,x) \notin F\big) \ \vee\ \big(\widehat{\delta}(p,x) \notin F \wedge \widehat{\delta}(q,x) \in F\big)$$

  for some $x \in \Sigma^*$

Minimization Algorithm

DFA $M = (Q, \Sigma, \delta, s, F)$

① remove inaccessible states

② for every two different states determine whether they are distinguishable (marking)

Definitions

DFA $M = (Q, \Sigma, \delta, s, F)$

- state p is inaccessible if $\widehat{\delta}(s,x) \neq p$ for all $x \in \Sigma^*$
- states p and q are distinguishable if

$$\left(\widehat{\delta}(p,x) \in F \wedge \widehat{\delta}(q,x) \notin F\right) \vee \left(\widehat{\delta}(p,x) \notin F \wedge \widehat{\delta}(q,x) \in F\right)$$

  for some $x \in \Sigma^*$

Minimization Algorithm

DFA $M = (Q, \Sigma, \delta, s, F)$

① remove inaccessible states

② for every two different states determine whether they are distinguishable (marking)

③ collapse indistinguishable states

Marking Algorithm

given DFA $M = (Q, \Sigma, \delta, s, F)$ without inaccessible states

① tabulate all unordered pairs $\{p, q\}$ with $p,\ q \in Q,$ initially unmarked

Marking Algorithm

given DFA $M = (Q, \Sigma, \delta, s, F)$ without inaccessible states

① tabulate all unordered pairs $\{p, q\}$ with $p,\ q \in Q,$ initially unmarked

② mark $\{p, q\}$ if $p \in F$ and $q \notin F$ or $p \notin F$ and $q \in F$

Marking Algorithm

given DFA $M = (Q, \Sigma, \delta, s, F)$ without inaccessible states

① tabulate all unordered pairs $\{p, q\}$ with $p,\ q \in Q,$ initially unmarked

② mark $\{p, q\}$ if $p \in F$ and $q \notin F$ or $p \notin F$ and $q \in F$

③ repeat until no change:

mark $\{p, q\}$ if $\{\delta(p, a), \delta(q, a)\}$ is marked for some $a \in \Sigma$

Marking Algorithm

given DFA $M = (Q, \Sigma, \delta, s, F)$ without inaccessible states

① tabulate all unordered pairs $\{p, q\}$ with $p,\ q \in Q,$ initially unmarked

② mark $\{p, q\}$ if $p \in F$ and $q \notin F$ or $p \notin F$ and $q \in F$

③ repeat until no change:

mark $\{p, q\}$ if $\{\delta(p, a), \delta(q, a)\}$ is marked for some $a \in \Sigma$

Notation

$p \approx q \quad \Longleftrightarrow \quad$ states p and q are indistinguishable

Marking Algorithm

given DFA $M = (Q, \Sigma, \delta, s, F)$ without inaccessible states

① tabulate all unordered pairs $\{p, q\}$ with $p,\ q \in Q,$ initially unmarked

② mark $\{p, q\}$ if $p \in F$ and $q \notin F$ or $p \notin F$ and $q \in F$

③ repeat until no change:

mark $\{p, q\}$ if $\{\delta(p, a), \delta(q, a)\}$ is marked for some $a \in \Sigma$

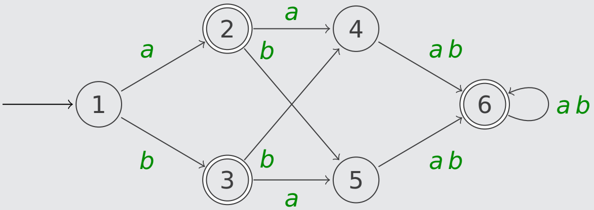
Notation

$p \approx q \quad \Longleftrightarrow \quad$ states p and q are indistinguishable

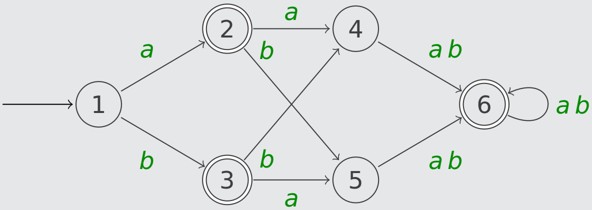
Lemma

$p \approx q \quad \Longleftrightarrow \quad \{p, q\}$ is unmarked

Example



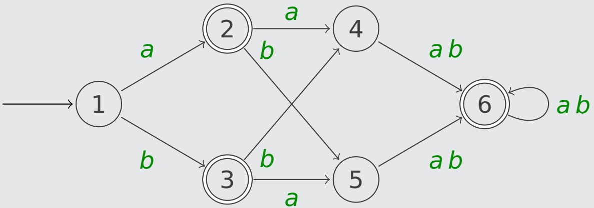
Example



1					
✓	2				
✓		3			
	✓	✓	4		
	✓	✓		5	
✓			✓	✓	6

1. final / nonfinal states are distinguishable

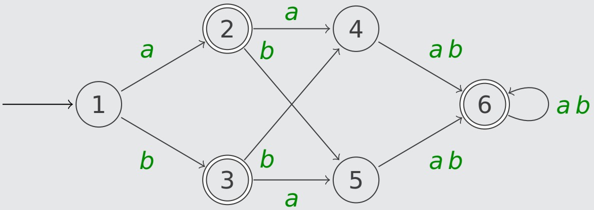
Example



1					
✓	2				
✓		3			
	✓	✓	4		
	✓	✓		5	
✓	✓	✓	✓	✓	6

1. final / nonfinal states are distinguishable
2. $\{2,6\} \xrightarrow{a} \{4,6\}$ $\{3,6\} \xrightarrow{a} \{5,6\}$

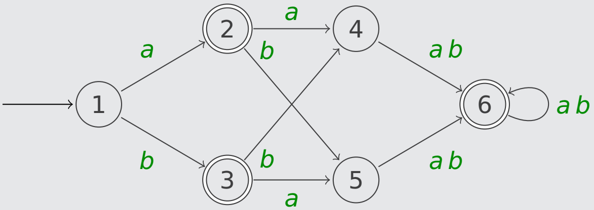
Example



1					
✓	2				
✓		3			
✓	✓	✓	4		
✓	✓	✓		5	
✓	✓	✓	✓	✓	6

1. final / nonfinal states are distinguishable
2. $\{2,6\} \xrightarrow{a} \{4,6\}$ $\quad \{3,6\} \xrightarrow{a} \{5,6\}$
3. $\{1,4\} \xrightarrow{a} \{2,6\}$ $\quad \{1,5\} \xrightarrow{a} \{2,6\}$

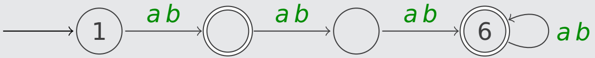
Example



1					
✓	2				
✓		3			
✓	✓	✓	4		
✓	✓	✓		5	
✓	✓	✓	✓	✓	6

1. final / nonfinal states are distinguishable
2. $\{2,6\} \xrightarrow{a} \{4,6\} \quad \{3,6\} \xrightarrow{a} \{5,6\}$
3. $\{1,4\} \xrightarrow{a} \{2,6\} \quad \{1,5\} \xrightarrow{a} \{2,6\}$

collapse states 2 and 3 and states 4 and 5:



Definition

states p and q of DFA $M = (Q, \Sigma, \delta, s, F)$ are indistinguishable $(p \approx q)$ if for all $x \in \Sigma^*$

$$\widehat{\delta}(p, x) \in F \iff \widehat{\delta}(q, x) \in F$$

Definition

states p and q of DFA $M = (Q, \Sigma, \delta, s, F)$ are indistinguishable $(p \approx q)$ if for all $x \in \Sigma^*$

$$\widehat{\delta}(p, x) \in F \iff \widehat{\delta}(q, x) \in F$$

Lemma

$\approx$ is equivalence relation on Q

Definition

states p and q of DFA $M = (Q, \Sigma, \delta, s, F)$ are indistinguishable ($p \approx q$) if for all $x \in \Sigma^*$

$$\widehat{\delta}(p, x) \in F \iff \widehat{\delta}(q, x) \in F$$

Lemma

$\approx$ is equivalence relation on Q:

1. $\forall\, p \in Q \qquad p \approx p$ (reflexivity)

Definition

states p and q of DFA $M = (Q, \Sigma, \delta, s, F)$ are indistinguishable ($p \approx q$) if for all $x \in \Sigma^*$

$$\widehat{\delta}(p, x) \in F \quad \Longleftrightarrow \quad \widehat{\delta}(q, x) \in F$$

Lemma

$\approx$ is equivalence relation on Q:

1. $\forall\, p \in Q \qquad p \approx p$ (reflexivity)
2. $\forall\, p,\ q \in Q \qquad p \approx q \implies q \approx p$ (symmetry)

Definition

states p and q of DFA $M = (Q, \Sigma, \delta, s, F)$ are indistinguishable $(p \approx q)$ if for all $x \in \Sigma^*$

$$\widehat{\delta}(p, x) \in F \iff \widehat{\delta}(q, x) \in F$$

Lemma

$\approx$ is equivalence relation on Q:

1. $\forall\, p \in Q \qquad p \approx p$ (reflexivity)
2. $\forall\, p, q \in Q \qquad p \approx q \implies q \approx p$ (symmetry)
3. $\forall\, p, q, r \in Q \qquad p \approx q \wedge q \approx r \implies p \approx r$ (transitivity)

Definition

states p and q of DFA $M = (Q, \Sigma, \delta, s, F)$ are indistinguishable ($p \approx q$) if for all $x \in \Sigma^*$

$$\widehat{\delta}(p, x) \in F \iff \widehat{\delta}(q, x) \in F$$

Lemma

$\approx$ is equivalence relation on Q:

1. $\forall\, p \in Q \quad p \approx p$ (reflexivity)
2. $\forall\, p, q \in Q \quad p \approx q \implies q \approx p$ (symmetry)
3. $\forall\, p, q, r \in Q \quad p \approx q \wedge q \approx r \implies p \approx r$ (transitivity)

Notation

$[p]_\approx = \{q \in Q \mid p \approx q\}$ denotes equivalence class of p

Definition (Collapsing Indistinguishable States)

DFA $M/\approx$ is defined as $(Q', \Sigma, \delta', s', F')$ with

- $Q' = \{[p]_\approx \mid p \in Q\}$

Definition (Collapsing Indistinguishable States)

DFA $M/{\approx}$ is defined as $(Q', \Sigma, \delta', s', F')$ with

- $Q' = \{[p]_\approx \mid p \in Q\}$
- $\delta'([p]_\approx, a) = [\delta(p, a)]_\approx$

Definition (Collapsing Indistinguishable States)

DFA $M/{\approx}$ is defined as $(Q', \Sigma, \delta', s', F')$ with

- $Q' = \{[p]_{\approx} \mid p \in Q\}$
- $\delta'([p]_{\approx}, a) = [\delta(p, a)]_{\approx}$
- $s' = [s]_{\approx}$

Definition (Collapsing Indistinguishable States)

DFA $M/{\approx}$ is defined as $(Q', \Sigma, \delta', s', F')$ with

- $Q' = \{[p]_\approx \mid p \in Q\}$
- $\delta'([p]_\approx, a) = [\delta(p, a)]_\approx$
- $s' = [s]_\approx$
- $F' = \{[p]_\approx \mid p \in F\}$

Definition (Collapsing Indistinguishable States)

DFA $M/\approx$ is defined as $(Q', \Sigma, \delta', s', F')$ with

- $Q' = \{ [p]_\approx \mid p \in Q \}$
- $\delta'([p]_\approx, a) = [\delta(p,a)]_\approx$ well–defined: $p \approx q \implies \delta(p,a) \approx \delta(q,a)$
- $s' = [s]_\approx$
- $F' = \{ [p]_\approx \mid p \in F \}$

Definition (Collapsing Indistinguishable States)

DFA $M/\!\approx$ is defined as $(Q', \Sigma, \delta', s', F')$ with

- $Q' = \{[p]_\approx \mid p \in Q\}$
- $\delta'([p]_\approx, a) = [\delta(p,a)]_\approx$ well–defined: $p \approx q \implies \delta(p,a) \approx \delta(q,a)$
- $s' = [s]_\approx$
- $F' = \{[p]_\approx \mid p \in F\}$

Lemma

1. $\widehat{\delta}'([p]_\approx, x) = [\widehat{\delta}(p,x)]_\approx$ for all $x \in \Sigma^*$

for all $p \in Q$

Definition (Collapsing Indistinguishable States)

DFA $M/{\approx}$ is defined as $(Q', \Sigma, \delta', s', F')$ with

- $Q' = \{[p]_\approx \mid p \in Q\}$
- $\delta'([p]_\approx, a) = [\delta(p,a)]_\approx$ well–defined: $p \approx q \implies \delta(p,a) \approx \delta(q,a)$
- $s' = [s]_\approx$
- $F' = \{[p]_\approx \mid p \in F\}$

Lemma

1. $\widehat{\delta}'([p]_\approx, x) = [\widehat{\delta}(p,x)]_\approx$ for all $x \in \Sigma^*$
2. $p \in F \iff [p]_\approx \in F'$

for all $p \in Q$

Theorem

$L(M/\approx) = L(M)$

Theorem

$L(M/{\approx}) = L(M)$

Proof

$$x \in L(M/{\approx}) \iff \widehat{\delta}'([s]_{\approx}, x) \in F'$$

Theorem

$L(M/{\approx}) = L(M)$

Proof

$$x \in L(M/{\approx}) \iff \widehat{\delta}'([s]_\approx, x) \in F' \iff [\widehat{\delta}(s,x)]_\approx \in F'$$

Theorem

$L(M/{\approx}) = L(M)$

Proof

$$x \in L(M/{\approx}) \iff \widehat{\delta}'([s]_{\approx}, x) \in F' \iff [\widehat{\delta}(s, x)]_{\approx} \in F' \iff \widehat{\delta}(s, x) \in F$$

Theorem

$L(M/{\approx}) = L(M)$

Proof

$$x \in L(M/{\approx}) \iff \widehat{\delta'}([s]_{\approx}, x) \in F' \iff [\widehat{\delta}(s,x)]_{\approx} \in F' \iff \widehat{\delta}(s,x) \in F \iff x \in L(M)$$

Theorem

$L(M/{\approx}) = L(M)$

Proof

$$x \in L(M/{\approx}) \iff \widehat{\delta'}([s]_{\approx}, x) \in F' \iff [\widehat{\delta}(s,x)]_{\approx} \in F' \iff \widehat{\delta}(s,x) \in F \iff x \in L(M)$$

Question

is $M/{\approx}$ minimum–state DFA for $L(M)$?

Theorem

$L(M/\approx) = L(M)$

Proof

$$x \in L(M/\approx) \iff \widehat{\delta}'([s]_\approx, x) \in F' \iff [\widehat{\delta}(s,x)]_\approx \in F' \iff \widehat{\delta}(s,x) \in F \iff x \in L(M)$$

Question

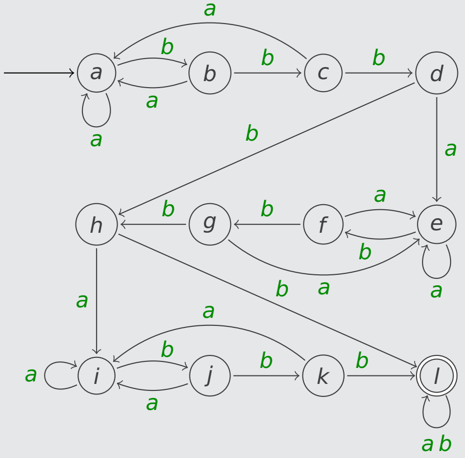
is $M/\approx$ minimum–state DFA for $L(M)$?

Lemma

$M/\approx$ cannot be collapsed further

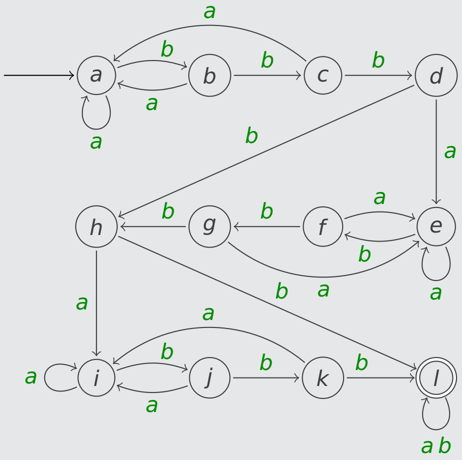
Example

DFA for set of strings over $\{a, b\}$ containing at least three occurrences of three consecutive b's, overlapping permitted:



Example

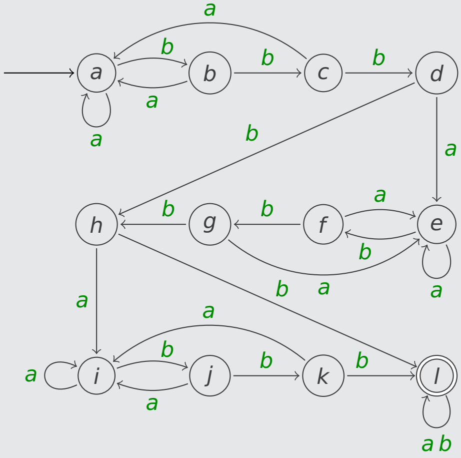
DFA for set of strings over $\{a, b\}$ containing at least three occurrences of three consecutive b's, overlapping permitted:



a											
	b										
		c									
			d								
				e							
					f						
						g					
							h				
								i			
									j		
										k	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	l

Example

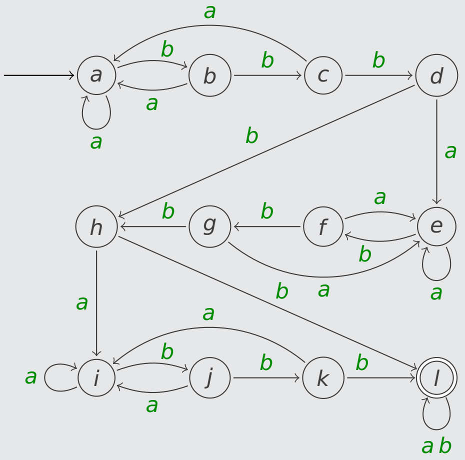
DFA for set of strings over $\{a, b\}$ containing at least three occurrences of three consecutive b's, overlapping permitted:



	a	b	c	d	e	f	g	h	i	j	k
a											
b											
c											
d											
e											
f											
g											
h	✓	✓	✓	✓	✓	✓	✓				
i								✓			
j								✓			
k	✓	✓	✓	✓	✓	✓	✓		✓	✓	
l	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Example

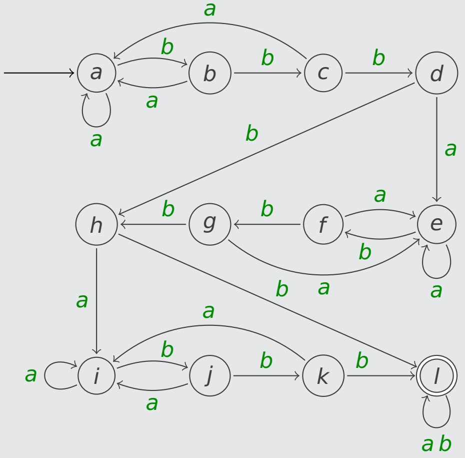
DFA for set of strings over $\{a, b\}$ containing at least three occurrences of three consecutive b's, overlapping permitted:



	a	b	c	d	e	f	g	h	i	j	k
b											
c											
d	✓	✓	✓								
e				✓							
f				✓							
g	✓	✓	✓		✓	✓					
h	✓	✓	✓	✓	✓	✓	✓				
i				✓			✓	✓			
j	✓	✓	✓		✓	✓		✓	✓		
k	✓	✓	✓	✓	✓	✓	✓		✓	✓	
l	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Example

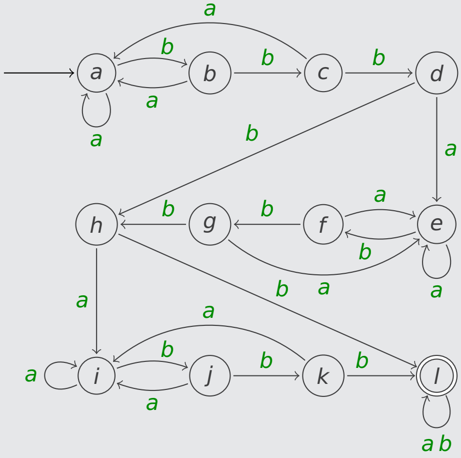
DFA for set of strings over $\{a, b\}$ containing at least three occurrences of three consecutive b's, overlapping permitted:



	a	b	c	d	e	f	g	h	i	j	k
c	✓	✓									
d	✓	✓	✓								
e			✓	✓							
f	✓	✓		✓	✓						
g	✓	✓	✓		✓	✓					
h	✓	✓	✓	✓	✓	✓	✓				
i	✓	✓		✓	✓		✓	✓			
j	✓	✓	✓		✓	✓		✓	✓		
k	✓	✓	✓	✓	✓	✓	✓		✓	✓	
l	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Example

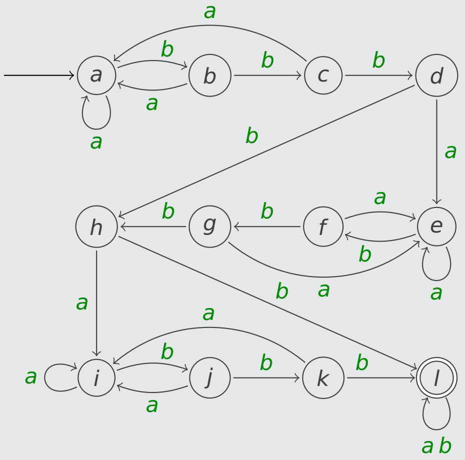
DFA for set of strings over $\{a, b\}$ containing at least three occurrences of three consecutive b's, overlapping permitted:



	a	b	c	d	e	f	g	h	i	j	k
b	✓										
c	✓	✓									
d	✓	✓	✓								
e	✓		✓	✓							
f	✓	✓		✓	✓						
g	✓	✓	✓		✓	✓					
h	✓	✓	✓	✓	✓	✓	✓				
i	✓	✓	✓	✓	✓	✓	✓	✓			
j	✓	✓	✓	✓	✓	✓	✓	✓	✓		
k	✓	✓	✓	✓	✓	✓	✓		✓	✓	
l	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Example

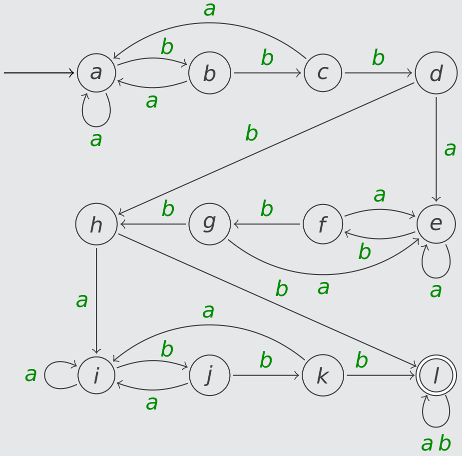
DFA for set of strings over $\{a, b\}$ containing at least three occurrences of three consecutive b's, overlapping permitted:



	a	b	c	d	e	f	g	h	i	j	k
a											
b	✓										
c	✓	✓									
d	✓	✓	✓								
e	✓	✓	✓	✓							
f	✓	✓	✓	✓	✓						
g	✓	✓	✓		✓	✓					
h	✓	✓	✓	✓	✓	✓	✓				
i	✓	✓	✓	✓	✓	✓	✓	✓			
j	✓	✓	✓	✓	✓	✓	✓	✓	✓		
k	✓	✓	✓	✓	✓	✓	✓		✓	✓	
l	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Example

DFA for set of strings over $\{a, b\}$ containing at least three occurrences of three consecutive b's, overlapping permitted:



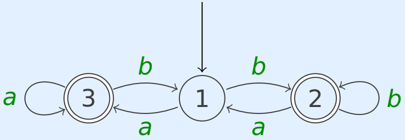
	a	b	c	d	e	f	g	h	i	j	k
b	✓										
c	✓	✓									
d	✓	✓	✓								
e	✓	✓	✓	✓							
f	✓	✓	✓	✓	✓						
g	✓	✓	✓		✓	✓					
h	✓	✓	✓	✓	✓	✓	✓				
i	✓	✓	✓	✓	✓	✓	✓	✓			
j	✓	✓	✓	✓	✓	✓	✓	✓	✓		
k	✓	✓	✓	✓	✓	✓	✓		✓	✓	
l	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

states d, g and h, k can be merged

Outline

Question

Which statements about the following DFA A are true ?



A the DFA is minimal

B states 2 and 3 are distinguishable

C $L(A) = L(a^*ba^*)$

D all states can be merged

Outline

1. Summary of Previous Lecture
2. Minimization
3. Intermezzo
4. **Weak Monadic Second–Order Logic**
5. Further Reading

Definitions

- first–order variables $V_1 = \{x, y, \dots\}$ ranging over natural numbers

Definitions

- first–order variables $V_1 = \{x, y, \ldots\}$ ranging over natural numbers
- second–order variables $V_2 = \{X, Y, \ldots\}$ ranging over finite sets of natural numbers

Definitions

- first–order variables $V_1 = \{x, y, \ldots\}$ ranging over natural numbers
- second–order variables $V_2 = \{X, Y, \ldots\}$ ranging over finite sets of natural numbers
- formulas of weak monadic second–order logic

$$\varphi ::= \bot$$

Definitions

- first–order variables $V_1 = \{x, y, \ldots\}$ ranging over natural numbers
- second–order variables $V_2 = \{X, Y, \ldots\}$ ranging over finite sets of natural numbers
- formulas of weak monadic second–order logic

$$\varphi \ ::= \ \bot \ \mid \ x < y$$

with x, $y \in V_1$

Definitions

- first–order variables $V_1 = \{x, y, \ldots\}$ ranging over natural numbers
- second–order variables $V_2 = \{X, Y, \ldots\}$ ranging over finite sets of natural numbers
- formulas of weak monadic second–order logic

$$\varphi ::= \bot \mid x < y \mid X(x)$$

with $x, y \in V_1$ and $X \in V_2$

Definitions

- first–order variables $V_1 = \{x, y, \ldots\}$ ranging over natural numbers
- second–order variables $V_2 = \{X, Y, \ldots\}$ ranging over finite sets of natural numbers
- formulas of weak monadic second–order logic

$$\varphi \ ::= \ \bot \ \mid \ x < y \ \mid \ X(x) \ \mid \ \neg\varphi \ \mid \ \varphi_1 \vee \varphi_2$$

with $x, y \in V_1$ and $X \in V_2$

Definitions

- first–order variables $V_1 = \{x, y, \dots\}$ ranging over natural numbers
- second–order variables $V_2 = \{X, Y, \dots\}$ ranging over finite sets of natural numbers
- formulas of weak monadic second–order logic

$$\varphi \ ::= \ \bot \ \mid \ x < y \ \mid \ X(x) \ \mid \ \neg\varphi \ \mid \ \varphi_1 \vee \varphi_2 \ \mid \ \exists\, x.\, \varphi \ \mid \ \exists\, X.\, \varphi$$

with x, $y \in V_1$ and $X \in V_2$

Definitions

- first–order variables $V_1 = \{x, y, \dots\}$ ranging over natural numbers
- second–order variables $V_2 = \{X, Y, \dots\}$ ranging over finite sets of natural numbers
- formulas of weak monadic second–order logic

$$\varphi \ ::= \ \bot \ | \ x < y \ | \ X(x) \ | \ \neg\varphi \ | \ \varphi_1 \vee \varphi_2 \ | \ \exists x.\, \varphi \ | \ \exists X.\, \varphi$$

with x, $y \in V_1$ and $X \in V_2$

Abbreviations

$$\varphi \wedge \psi \ := \ \neg(\neg\varphi \vee \neg\psi) \qquad\qquad \varphi \rightarrow \psi \ := \ \neg\varphi \vee \psi$$

Definitions

- first–order variables $V_1 = \{x, y, \ldots\}$ ranging over natural numbers
- second–order variables $V_2 = \{X, Y, \ldots\}$ ranging over finite sets of natural numbers
- formulas of weak monadic second–order logic

$$\varphi \ ::= \ \bot \ \mid \ x < y \ \mid \ X(x) \ \mid \ \neg\varphi \ \mid \ \varphi_1 \vee \varphi_2 \ \mid \ \exists\, x.\, \varphi \ \mid \ \exists\, X.\, \varphi$$

with x, $y \in V_1$ and $X \in V_2$

Abbreviations

$$\varphi \wedge \psi \ := \ \neg(\neg\varphi \vee \neg\psi) \qquad\qquad \varphi \to \psi \ := \ \neg\varphi \vee \psi$$

$$\forall\, x.\, \varphi \ := \ \neg\exists\, x.\, \neg\varphi \qquad\qquad \forall\, X.\, \varphi \ := \ \neg\exists\, X.\, \neg\varphi$$

Definitions

- first–order variables $V_1 = \{x, y, \dots\}$ ranging over natural numbers
- second–order variables $V_2 = \{X, Y, \dots\}$ ranging over finite sets of natural numbers
- formulas of weak monadic second–order logic

$$\varphi \ ::= \ \bot \mid x < y \mid X(x) \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \exists x.\, \varphi \mid \exists X.\, \varphi$$

with x, $y \in V_1$ and $X \in V_2$

Abbreviations

$$\varphi \wedge \psi \ := \ \neg(\neg\varphi \vee \neg\psi) \qquad \varphi \to \psi \ := \ \neg\varphi \vee \psi$$

$$\forall x.\, \varphi \ := \ \neg\exists x.\, \neg\varphi \qquad \forall X.\, \varphi \ := \ \neg\exists X.\, \neg\varphi$$

$$x \leqslant y \ := \ \neg(y < x) \qquad x = y \ := \ x \leqslant y \wedge y \leqslant x$$

Definitions

- first–order variables $V_1 = \{x, y, \ldots\}$ ranging over natural numbers
- second–order variables $V_2 = \{X, Y, \ldots\}$ ranging over finite sets of natural numbers
- formulas of weak monadic second–order logic

$$\varphi \ ::= \ \bot \mid x < y \mid X(x) \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \exists\, x.\, \varphi \mid \exists\, X.\, \varphi$$

  with x, $y \in V_1$ and $X \in V_2$

Abbreviations

$$\begin{aligned} \varphi \wedge \psi &:= \neg(\neg\varphi \vee \neg\psi) \\ \forall\, x.\, \varphi &:= \neg\exists\, x.\, \neg\varphi \\ x \leqslant y &:= \neg(y < x) \\ \top &:= \neg\bot \end{aligned}$$

$$\begin{aligned} \varphi \to \psi &:= \neg\varphi \vee \psi \\ \forall\, X.\, \varphi &:= \neg\exists\, X.\, \neg\varphi \\ x = y &:= x \leqslant y \wedge y \leqslant x \\ x = 0 &:= \neg\exists\, y.\, y < x \end{aligned}$$

Definitions

- first–order variables $V_1 = \{x, y, \dots\}$ ranging over natural numbers
- second–order variables $V_2 = \{X, Y, \dots\}$ ranging over finite sets of natural numbers
- formulas of weak monadic second–order logic

$$\varphi \ ::= \ \bot \mid x < y \mid X(x) \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \exists\, x.\, \varphi \mid \exists\, X.\, \varphi$$

with x, $y \in V_1$ and $X \in V_2$

Abbreviations

$$\varphi \wedge \psi \ := \ \neg(\neg\varphi \vee \neg\psi) \qquad \varphi \rightarrow \psi \ := \ \neg\varphi \vee \psi$$

$$\forall\, x.\, \varphi \ := \ \neg\exists\, x.\, \neg\varphi \qquad \forall\, X.\, \varphi \ := \ \neg\exists\, X.\, \neg\varphi$$

$$x \leqslant y \ := \ \neg(y < x) \qquad x = y \ := \ x \leqslant y \wedge y \leqslant x$$

$$\top \ := \ \neg\bot \qquad x = 0 \ := \ \neg\exists\, y.\, y < x$$

$$X(0) \ := \ \exists\, x.\, (X(x) \wedge x = 0)$$

Definitions

- first–order variables $V_1 = \{x, y, \ldots\}$ ranging over natural numbers
- second–order variables $V_2 = \{X, Y, \ldots\}$ ranging over finite sets of natural numbers
- formulas of weak monadic second–order logic

$$\varphi \ ::= \ \bot \mid x < y \mid X(x) \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \exists x.\, \varphi \mid \exists X.\, \varphi$$

with x, $y \in V_1$ and $X \in V_2$

Abbreviations

$$\begin{aligned}
\varphi \wedge \psi &:= \neg(\neg\varphi \vee \neg\psi) & \varphi \to \psi &:= \neg\varphi \vee \psi \\
\forall x.\, \varphi &:= \neg\exists x.\, \neg\varphi & \forall X.\, \varphi &:= \neg\exists X.\, \neg\varphi \\
x \leqslant y &:= \neg(y < x) & x = y &:= x \leqslant y \wedge y \leqslant x \\
\top &:= \neg\bot & x = 0 &:= \neg\exists y.\, y < x \\
X(0) &:= \exists x.\, (X(x) \wedge x = 0) & &
\end{aligned}$$

Definitions

- first–order variables $V_1 = \{x, y, \ldots\}$ ranging over natural numbers
- second–order variables $V_2 = \{X, Y, \ldots\}$ ranging over finite sets of natural numbers
- formulas of weak monadic second–order logic

$$\varphi \;::=\; \bot \mid x < y \mid X(x) \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \exists\, x.\, \varphi \mid \exists\, X.\, \varphi$$

with x, $y \in V_1$ and $X \in V_2$

Abbreviations

$$\begin{aligned}
\varphi \wedge \psi &:= \neg(\neg\varphi \vee \neg\psi) & \varphi \to \psi &:= \neg\varphi \vee \psi \\
\forall\, x.\, \varphi &:= \neg\exists\, x.\, \neg\varphi & \forall\, X.\, \varphi &:= \neg\exists\, X.\, \neg\varphi \\
x \leqslant y &:= \neg(y < x) & x = y &:= x \leqslant y \wedge y \leqslant x \\
\top &:= \neg\bot & x = 0 &:= \neg\exists\, y.\, y < x \\
X(0) &:= \exists\, x.\; X(x) \wedge x = 0
\end{aligned}$$

Remarks

- $X(x)$ represents $x \in X$

Remarks

- $X(x)$ represents $x \in X$
- MSO is weak MSO without restriction to finite sets

Remarks

- $X(x)$ represents $x \in X$
- MSO is weak MSO without restriction to finite sets

Examples

- $(\forall\, x.\, X(x) \to Y(x)) \wedge (\exists\, y.\, \neg X(y) \wedge Y(y))$

Remarks

- $X(x)$ represents $x \in X$
- MSO is weak MSO without restriction to finite sets

Examples

- $(\forall x.\, X(x) \to Y(x)) \wedge (\exists y.\, \neg X(y) \wedge Y(y))$
- $\exists X.\, (\forall x.\, x = 0 \to X(x)) \wedge (\forall x.\, X(x) \to \exists y.\, x < y \wedge X(y))$

Remarks

- $X(x)$ represents $x \in X$
- MSO is weak MSO without restriction to finite sets

Examples

- $(\forall\, x.\, X(x) \rightarrow Y(x)) \wedge (\exists\, y.\, \neg X(y) \wedge Y(y))$
- $\exists\, X.\, (\forall\, x.\, x = 0 \rightarrow X(x)) \wedge (\forall\, x.\, X(x) \rightarrow \exists\, y.\, x < y \wedge X(y))$
- $\exists\, X.\, X(0) \wedge \big(\forall\, y.\, \forall\, z.\, z = y + 1 \wedge z \leqslant x \rightarrow (X(y) \leftrightarrow \neg X(z))\big) \wedge X(x)$

Remarks

- $X(x)$ represents $x \in X$
- MSO is weak MSO without restriction to finite sets

Examples

- $(\forall x. X(x) \rightarrow Y(x)) \wedge (\exists y. \neg X(y) \wedge Y(y))$
- $\exists X. (\forall x. x = 0 \rightarrow X(x)) \wedge (\forall x. X(x) \rightarrow \exists y. x < y \wedge X(y))$
- $\exists X. X(0) \wedge \big(\forall y. \forall z. z = y + 1 \wedge z \leqslant x \rightarrow (X(y) \leftrightarrow \neg X(z))\big) \wedge X(x) \quad \Longleftrightarrow \quad x$ is even

Remarks

- $X(x)$ represents $x \in X$
- MSO is weak MSO without restriction to finite sets

Examples

- $(\forall x.\, X(x) \rightarrow Y(x)) \wedge (\exists y.\, \neg X(y) \wedge Y(y))$
- $\exists X.\, (\forall x.\, x = 0 \rightarrow X(x)) \wedge (\forall x.\, X(x) \rightarrow \exists y.\, x < y \wedge X(y))$
- $\exists X.\, X(0) \wedge \big(\forall y.\, \forall z.\, z = y + 1 \wedge z \leqslant x \rightarrow (X(y) \leftrightarrow \neg X(z))\big) \wedge X(x) \quad \Longleftrightarrow \quad x$ is even

Remark

$z = y + 1$ abbreviates $y < z \wedge \neg\exists x.\, (y < x \wedge x < z)$

Definitions

- assignment α is mapping from variables $x \in V_1$ to $\mathbb{N}$ and $X \in V_2$ to finite subsets of $\mathbb{N}$

Definitions

- assignment α is mapping from variables $x \in V_1$ to $\mathbb{N}$ and $X \in V_2$ to finite subsets of $\mathbb{N}$
- assignment α satisfies formula φ ($\alpha \vDash \varphi$):

$$\alpha \nvDash \bot$$

Definitions

- assignment α is mapping from variables $x \in V_1$ to $\mathbb{N}$ and $X \in V_2$ to finite subsets of $\mathbb{N}$
- assignment α satisfies formula φ ($\alpha \vDash \varphi$):

$$\alpha \nvDash \bot$$

$$\alpha \vDash x < y \quad\iff\quad \alpha(x) < \alpha(y)$$

Definitions

- assignment α is mapping from variables $x \in V_1$ to $\mathbb{N}$ and $X \in V_2$ to finite subsets of $\mathbb{N}$
- assignment α satisfies formula φ ($\alpha \vDash \varphi$):

$$
\begin{aligned}
&\alpha \nvDash \bot \\
&\alpha \vDash x < y && \iff && \alpha(x) < \alpha(y) \\
&\alpha \vDash X(x) && \iff && \alpha(x) \in \alpha(X)
\end{aligned}
$$

Definitions

- assignment α is mapping from variables $x \in V_1$ to $\mathbb{N}$ and $X \in V_2$ to finite subsets of $\mathbb{N}$
- assignment α satisfies formula φ ($\alpha \vDash \varphi$):

$$\begin{aligned}
&\alpha \nvDash \bot \\
&\alpha \vDash x < y && \iff && \alpha(x) < \alpha(y) \\
&\alpha \vDash X(x) && \iff && \alpha(x) \in \alpha(X) \\
&\alpha \vDash \neg\varphi && \iff && \alpha \nvDash \varphi \\
&\alpha \vDash \varphi_1 \vee \varphi_2 && \iff && \alpha \vDash \varphi_1 \text{ or } \alpha \vDash \varphi_2
\end{aligned}$$

Definitions

- assignment α is mapping from variables $x \in V_1$ to $\mathbb{N}$ and $X \in V_2$ to finite subsets of $\mathbb{N}$
- assignment α satisfies formula φ ($\alpha \vDash \varphi$):

$$
\begin{array}{lcll}
\alpha \nvDash \bot & & & \\
\alpha \vDash x < y & \iff & \alpha(x) < \alpha(y) & \\
\alpha \vDash X(x) & \iff & \alpha(x) \in \alpha(X) & \\
\alpha \vDash \neg\varphi & \iff & \alpha \nvDash \varphi & \\
\alpha \vDash \varphi_1 \vee \varphi_2 & \iff & \alpha \vDash \varphi_1 \text{ or } \alpha \vDash \varphi_2 & \\
\alpha \vDash \exists x.\, \varphi & \iff & \alpha[x \mapsto n] \vDash \varphi & \text{for some } n \in \mathbb{N}
\end{array}
$$

Definitions

- assignment α is mapping from variables $x \in V_1$ to $\mathbb{N}$ and $X \in V_2$ to finite subsets of $\mathbb{N}$
- assignment α satisfies formula φ ($\alpha \vDash \varphi$):

$$
\begin{array}{llll}
\alpha \nvDash \bot & & & \\
\alpha \vDash x < y & \Longleftrightarrow & \alpha(x) < \alpha(y) & \\
\alpha \vDash X(x) & \Longleftrightarrow & \alpha(x) \in \alpha(X) & \\
\alpha \vDash \neg\varphi & \Longleftrightarrow & \alpha \nvDash \varphi & \\
\alpha \vDash \varphi_1 \vee \varphi_2 & \Longleftrightarrow & \alpha \vDash \varphi_1 \text{ or } \alpha \vDash \varphi_2 & \\
\alpha \vDash \exists x.\, \varphi & \Longleftrightarrow & \alpha[x \mapsto n] \vDash \varphi & \text{for some } n \in \mathbb{N} \\
\alpha \vDash \exists X.\, \varphi & \Longleftrightarrow & \alpha[X \mapsto N] \vDash \varphi & \text{for some finite subset } N \subset \mathbb{N}
\end{array}
$$

Definitions

- formula φ is satisfiable if $\alpha \vDash \varphi$ for some assignment α

Definitions

- formula φ is satisfiable if $\alpha \vDash \varphi$ for some assignment α
- formula φ is valid if $\alpha \vDash \varphi$ for all assignments α

Definitions

- formula φ is satisfiable if $\alpha \vDash \varphi$ for some assignment α
- formula φ is valid if $\alpha \vDash \varphi$ for all assignments α
- model of formula φ is assignment α such that $\alpha \vDash \varphi$

Definitions

- formula φ is satisfiable if $\alpha \vDash \varphi$ for some assignment α
- formula φ is valid if $\alpha \vDash \varphi$ for all assignments α
- model of formula φ is assignment α such that $\alpha \vDash \varphi$
- size of model α is smallest n such that

  ① $\alpha(x) < n$ for $x \in V_1$

Definitions

- formula φ is satisfiable if $\alpha \vDash \varphi$ for some assignment α
- formula φ is valid if $\alpha \vDash \varphi$ for all assignments α
- model of formula φ is assignment α such that $\alpha \vDash \varphi$
- size of model α is smallest n such that

  ① $\alpha(x) < n$ for $x \in V_1$

  ② $\alpha(X) \subseteq \{0, \dots, n-1\}$ for $X \in V_2$

Definitions

- formula φ is satisfiable if $\alpha \vDash \varphi$ for some assignment α
- formula φ is valid if $\alpha \vDash \varphi$ for all assignments α
- model of formula φ is assignment α such that $\alpha \vDash \varphi$
- size of model α is smallest n such that
  1. $\alpha(x) < n$ for $x \in V_1$
  2. $\alpha(X) \subseteq \{0, \dots, n-1\}$ for $X \in V_2$

Examples

- $(\forall\, x.\, X(x) \rightarrow Y(x)) \wedge (\exists\, y.\, \neg X(y) \wedge Y(y))$

Definitions

- formula φ is satisfiable if $\alpha \vDash \varphi$ for some assignment α
- formula φ is valid if $\alpha \vDash \varphi$ for all assignments α
- model of formula φ is assignment α such that $\alpha \vDash \varphi$
- size of model α is smallest n such that
  1. $\alpha(x) < n$ for $x \in V_1$
  2. $\alpha(X) \subseteq \{0, \ldots, n-1\}$ for $X \in V_2$

Examples

- $(\forall\, x.\, X(x) \to Y(x)) \wedge (\exists\, y.\, \neg X(y) \wedge Y(y))$ satisfiable

Definitions

- formula φ is satisfiable if $\alpha \vDash \varphi$ for some assignment α
- formula φ is valid if $\alpha \vDash \varphi$ for all assignments α
- model of formula φ is assignment α such that $\alpha \vDash \varphi$
- size of model α is smallest n such that
  1. $\alpha(x) < n$ for $x \in V_1$
  2. $\alpha(X) \subseteq \{0, \dots, n-1\}$ for $X \in V_2$

Examples

- $(\forall x. X(x) \to Y(x)) \wedge (\exists y. \neg X(y) \wedge Y(y))$ satisfiable
- $(\forall x. x = 0 \to X(x)) \wedge (\forall x. X(x) \to \exists y. x < y \wedge X(y))$

Definitions

- formula φ is **satisfiable** if $\alpha \vDash \varphi$ for some assignment α
- formula φ is **valid** if $\alpha \vDash \varphi$ for all assignments α
- **model** of formula φ is assignment α such that $\alpha \vDash \varphi$
- **size** of model α is smallest n such that
  1. $\alpha(x) < n$ for $x \in V_1$
  2. $\alpha(X) \subseteq \{0, \ldots, n-1\}$ for $X \in V_2$

Examples

- $(\forall\, x.\, X(x) \rightarrow Y(x)) \wedge (\exists\, y.\, \neg X(y) \wedge Y(y))$ satisfiable
- $(\forall\, x.\, x = 0 \rightarrow X(x)) \wedge (\forall\, x.\, X(x) \rightarrow \exists\, y.\, x < y \wedge X(y))$ unsatisfiable

Definitions

- formula φ is **satisfiable** if $\alpha \vDash \varphi$ for some assignment α
- formula φ is **valid** if $\alpha \vDash \varphi$ for all assignments α
- **model** of formula φ is assignment α such that $\alpha \vDash \varphi$
- **size** of model α is smallest n such that
  1. $\alpha(x) < n$ for $x \in V_1$
  2. $\alpha(X) \subseteq \{0, \dots, n-1\}$ for $X \in V_2$

Examples

- $(\forall x. X(x) \to Y(x)) \wedge (\exists y. \neg X(y) \wedge Y(y))$ satisfiable
- $(\forall x. x = 0 \to X(x)) \wedge (\forall x. X(x) \to \exists y. x < y \wedge X(y))$ unsatisfiable
- $(\exists x. X(x) \wedge \exists y. X(y) \wedge x \neq y) \wedge (\forall x. X(x) \to \exists y. Y(y) \wedge x < y)$

Definitions

- formula φ is satisfiable if $\alpha \vDash \varphi$ for some assignment α
- formula φ is valid if $\alpha \vDash \varphi$ for all assignments α
- model of formula φ is assignment α such that $\alpha \vDash \varphi$
- size of model α is smallest n such that
  1. $\alpha(x) < n$ for $x \in V_1$
  2. $\alpha(X) \subseteq \{0, \dots, n-1\}$ for $X \in V_2$

Examples

- $(\forall x. X(x) \rightarrow Y(x)) \wedge (\exists y. \neg X(y) \wedge Y(y))$ satisfiable
- $(\forall x. x = 0 \rightarrow X(x)) \wedge (\forall x. X(x) \rightarrow \exists y. x < y \wedge X(y))$ unsatisfiable
- $(\exists x. X(x) \wedge \exists y. X(y) \wedge x \neq y) \wedge (\forall x. X(x) \rightarrow \exists y. Y(y) \wedge x < y)$ satisfiable

Definition

given alphabet Σ

- second–order variables $V_2 = \{P_a \mid a \in \Sigma\}$

Definition

given alphabet Σ and string $x = a_0 \cdots a_{n-1} \in \Sigma^*$

- second–order variables $V_2 = \{P_a \mid a \in \Sigma\}$
- $\alpha_x(P_a) = \{i < n \mid a_i = a\}$

Definition

given alphabet Σ and string $x = a_0 \cdots a_{n-1} \in \Sigma^*$

- second–order variables $V_2 = \{P_a \mid a \in \Sigma\}$
- $\alpha_x(P_a) = \{i < n \mid a_i = a\}$

Notation

$\underline{x}$ for α_x

Definition

given alphabet Σ and string $x = a_0 \cdots a_{n-1} \in \Sigma^*$

- second–order variables $V_2 = \{P_a \mid a \in \Sigma\}$
- $\alpha_x(P_a) = \{i < n \mid a_i = a\}$

Notation

$\underline{x}$ for α_x

Example

$\Sigma = \{a, b\}$

- $\underline{abba}(P_a) = \{0, 3\}$

Definition

given alphabet Σ and string $x = a_0 \cdots a_{n-1} \in \Sigma^*$

- second–order variables $V_2 = \{P_a \mid a \in \Sigma\}$
- $\alpha_x(P_a) = \{i < n \mid a_i = a\}$

Notation

$\underline{x}$ for α_x

Example

$\Sigma = \{a, b\}$

- $\underline{abba}(P_a) = \{0, 3\}$
- $\underline{abba}(P_b) = \{1, 2\}$

Example

$\Sigma = \{a, b\}$

- $\varphi = \forall x. \neg(P_a(x) \wedge P_b(x))$

Example

$\Sigma = \{a, b\}$

- $\varphi = \forall\, x.\, \neg(P_a(x) \wedge P_b(x))$ $\qquad \underline{x} \vDash \varphi$ for all $x \in \Sigma^*$

Example

$\Sigma = \{a, b\}$

- $\varphi = \forall\, x.\, \neg(P_a(x) \wedge P_b(x))$ $\qquad\qquad$ $\underline{x} \vDash \varphi$ for all $x \in \Sigma^*$
- $\psi = \forall\, x.\, \forall\, y.\, (P_a(x) \wedge P_b(y)) \rightarrow x < y$

Example

$\Sigma = \{a, b\}$

- $\varphi = \forall x. \neg(P_a(x) \wedge P_b(x))$ $\qquad$ $\underline{x} \vDash \varphi$ for all $x \in \Sigma^*$
- $\psi = \forall x. \forall y. (P_a(x) \wedge P_b(y)) \rightarrow x < y$ $\qquad$ $\underline{aabbb} \vDash \psi$

Example

$\Sigma = \{a, b\}$

- $\varphi = \forall\, x.\, \neg(P_a(x) \wedge P_b(x))$ $\quad\quad$ $\underline{x} \vDash \varphi$ for all $x \in \Sigma^*$
- $\psi = \forall\, x.\, \forall\, y.\, (P_a(x) \wedge P_b(y)) \rightarrow x < y$ $\quad\quad$ $\underline{aabbb} \vDash \psi$ $\quad$ $\underline{aabab} \nvDash \psi$

Example

$\Sigma = \{a, b\}$

- $\varphi = \forall\, x.\, \neg(P_a(x) \wedge P_b(x))$ $\qquad$ $\underline{x} \vDash \varphi$ for all $x \in \Sigma^*$
- $\psi = \forall\, x.\, \forall\, y.\, (P_a(x) \wedge P_b(y)) \to x < y$ $\qquad$ $\underline{aabbb} \vDash \psi$ $\qquad$ $\underline{aabab} \nvDash \psi$
- $\chi = \forall\, x.\, P_b(x) \to \exists\, y.\, P_a(y) \wedge y < x$

Example

$\Sigma = \{a, b\}$

- $\varphi = \forall\, x.\, \neg(P_a(x) \wedge P_b(x))$ $\qquad$ $\underline{x} \vDash \varphi$ for all $x \in \Sigma^*$
- $\psi = \forall\, x.\, \forall\, y.\, (P_a(x) \wedge P_b(y)) \to x < y$ $\qquad$ $\underline{aabbb} \vDash \psi$ $\qquad$ $\underline{aabab} \nvDash \psi$
- $\chi = \forall\, x.\, P_b(x) \to \exists\, y.\, P_a(y) \wedge y < x$ $\qquad$ $\underline{aaaaa} \vDash \chi$

Example

$\Sigma = \{a, b\}$

- $\varphi = \forall\, x.\, \neg(P_a(x) \wedge P_b(x))$ $\qquad \underline{x} \vDash \varphi$ for all $x \in \Sigma^*$
- $\psi = \forall\, x.\, \forall\, y.\, (P_a(x) \wedge P_b(y)) \rightarrow x < y$ $\qquad \underline{aabbb} \vDash \psi \qquad \underline{aabab} \nvDash \psi$
- $\chi = \forall\, x.\, P_b(x) \rightarrow \exists\, y.\, P_a(y) \wedge y < x$ $\qquad \underline{aaaaa} \vDash \chi \qquad \underline{babab} \nvDash \chi$

Example

$\Sigma = \{a, b\}$

- $\varphi = \forall\, x.\, \neg(P_a(x) \wedge P_b(x))$ $\qquad \underline{x} \vDash \varphi$ for all $x \in \Sigma^*$
- $\psi = \forall\, x.\, \forall\, y.\, (P_a(x) \wedge P_b(y)) \to x < y$ $\qquad \underline{aabbb} \vDash \psi \qquad \underline{aabab} \nvDash \psi$
- $\chi = \forall\, x.\, P_b(x) \to \exists\, y.\, P_a(y) \wedge y < x$ $\qquad \underline{aaaaa} \vDash \chi \qquad \underline{babab} \nvDash \chi$

Definitions

- given alphabet Σ and WMSO formula φ with free variables (exclusively) in $\{P_a \mid a \in \Sigma\}$

$$L(\varphi) = \{x \in \Sigma^* \mid \underline{x} \vDash \varphi\}$$

Example

$\Sigma = \{a, b\}$

- $\varphi = \forall\, x.\, \neg(P_a(x) \wedge P_b(x))$ $\qquad$ $\underline{x} \vDash \varphi$ for all $x \in \Sigma^*$
- $\psi = \forall\, x.\, \forall\, y.\, (P_a(x) \wedge P_b(y)) \to x < y$ $\qquad$ $\underline{aabbb} \vDash \psi$ $\qquad$ $\underline{aabab} \nvDash \psi$
- $\chi = \forall\, x.\, P_b(x) \to \exists\, y.\, P_a(y) \wedge y < x$ $\qquad$ $\underline{aaaaa} \vDash \chi$ $\qquad$ $\underline{babab} \nvDash \chi$

Definitions

- given alphabet Σ and WMSO formula φ with free variables (exclusively) in $\{P_a \mid a \in \Sigma\}$

$$L(\varphi) = \{x \in \Sigma^* \mid \underline{x} \vDash \varphi\}$$

- set $A \subseteq \Sigma^*$ is WMSO definable if $A = L(\varphi)$ for some WMSO formula φ

Examples

$\Sigma = \{a, b\}$

- regular set $L((a+b)^* ab (a+b)^*)$ is WMSO definable by formula

$$\exists\, x.\, \exists\, y.\, P_a(x) \wedge P_b(y) \wedge x < y \wedge \neg \exists\, z.\, x < z \wedge z < y$$

Examples

$\Sigma = \{a, b\}$

- regular set $L((a+b)^*ab(a+b)^*)$ is WMSO definable by formula

$$\exists\, x.\, \exists\, y.\, P_a(x) \wedge P_b(y) \wedge x < y \wedge \neg \exists\, z.\, x < z \wedge z < y$$

- WMSO formula

$$\exists\, x.\, P_a(x) \wedge \forall\, y.\, x < y \rightarrow \neg(P_a(y) \vee P_b(y))$$

Examples

$\Sigma = \{a, b\}$

- regular set $L((a+b)^*ab(a+b)^*)$ is WMSO definable by formula

$$\exists\, x.\, \exists\, y.\, P_a(x) \wedge P_b(y) \wedge x < y \wedge \neg \exists\, z.\, x < z \wedge z < y$$

- WMSO formula

$$\exists\, x.\, P_a(x) \wedge \forall\, y.\, x < y \rightarrow \neg(P_a(y) \vee P_b(y))$$

defines regular set $\{xa \mid x \in \Sigma^*\}$

Examples

$\Sigma = \{a, b\}$

- regular set $L((a+b)^*ab(a+b)^*)$ is WMSO definable by formula

$$\exists x. \exists y. P_a(x) \wedge P_b(y) \wedge x < y \wedge \neg \exists z. x < z \wedge z < y$$

- WMSO formula

$$\exists x. P_a(x) \wedge \forall y. x < y \to \neg(P_a(y) \vee P_b(y))$$

  defines regular set $\{xa \mid x \in \Sigma^*\}$

Theorem

set $A \subseteq \Sigma^*$ is regular if and only if A is WMSO definable

Outline

Kozen

- Lectures 13 and 14

Kozen

- Lectures 13 and 14

Important Concepts

- $\alpha \vDash \varphi$
- indistinguishable states
- minimization algorithm
- model
- MSO
- satisfiability
- validity
- weak monadic second–order logic (WMSO)
- WMSO definability

Kozen

- Lectures 13 and 14

Important Concepts

- $\alpha \vDash \varphi$
- indistinguishable states
- minimization algorithm
- model
- MSO
- satisfiability
- validity
- weak monadic second-order logic (WMSO)
- WMSO definability

homework for October 31