CMPE 322/327 - Theory of Computation

Week 8: DFA Equivalence Checking & Context Free Grammars & Ambiguity

Burak Ekici

April 11-15, 2022

Outline

Definition

Kleene Algebra consists of set K with distinguished elements $0, 1 \in K$ and operations $* : K \to K$ and $+, \times : K \times K \to K$

Definition

Kleene Algebra consists of set K with distinguished elements $0, 1 \in K$ and operations $* : K \to K$ and $+, \times : K \times K \to K$

Notation

- ab for $a \times b$ $\quad$ a^* for $*(a)$

Definition

Kleene Algebra consists of set K with distinguished elements $0, 1 \in K$ and operations $* : K \to K$ and $+, \times : K \times K \to K$

Notation

- ab for $a \times b$ $\quad$ a^* for $*(a)$
- binding precedence: $* > \times > +$

Definition

Kleene Algebra consists of set K with distinguished elements $0, 1 \in K$ and operations $* : K \to K$ and $+, \times : K \times K \to K$ such that

$$
\begin{aligned}
a + (b + c) &= (a + b) + c \\
a + b &= b + a \\
a + a &= a \\
a + 0 &= a
\end{aligned}
$$

for all $a, b, c \in K$

Notation

- ab for $a \times b$ $\quad$ a^* for $*(a)$
- binding precedence: $* > \times > +$

Definition

Kleene Algebra consists of set K with distinguished elements $0, 1 \in K$ and operations $* : K \to K$ and $+, \times : K \times K \to K$ such that

$$
\begin{array}{rclcrclcrcl}
a + (b + c) & = & (a + b) + c & \quad & a0 & = & 0 & \quad & a(bc) & = & (ab)c \\
a + b & = & b + a & & 0a & = & 0 & & & & \\
a + a & = & a & & 1a & = & a & & & & \\
a + 0 & = & a & & a1 & = & a & & & &
\end{array}
$$

for all $a, b, c \in K$

Notation

- ab for $a \times b$ $\qquad$ a^* for $*(a)$
- binding precedence: $* > \times > +$

Definition

Kleene Algebra consists of set K with distinguished elements $0, 1 \in K$ and operations $* : K \to K$ and $+, \times : K \times K \to K$ such that

$$
\begin{aligned}
a + (b + c) &= (a + b) + c & a0 &= 0 & a(bc) &= (ab)c \\
a + b &= b + a & 0a &= 0 & (a + b)c &= ac + bc \\
a + a &= a & 1a &= a & a(b + c) &= ab + ac \\
a + 0 &= a & a1 &= a & &
\end{aligned}
$$

for all $a, b, c \in K$

Notation

- ab for $a \times b$ $\quad a^*$ for $*(a)$
- binding precedence: $* > \times > +$

Definition

Kleene Algebra consists of set K with distinguished elements $0, 1 \in K$ and operations $* : K \to K$ and $+, \times : K \times K \to K$ such that

$$\begin{aligned} a + (b + c) &= (a + b) + c \\ a + b &= b + a \\ a + a &= a \\ a + 0 &= a \end{aligned} \qquad \begin{aligned} a0 &= 0 \\ 0a &= 0 \\ 1a &= a \\ a1 &= a \end{aligned} \qquad \begin{aligned} a(bc) &= (ab)c \\ (a + b)c &= ac + bc \\ a(b + c) &= ab + ac \end{aligned} \qquad \begin{aligned} 1 + aa^* &= a^* \\ 1 + a^*a &= a^* \end{aligned}$$

for all $a, b, c \in K$

Notation

- ab for $a \times b$ $\qquad a^*$ for $*(a)$
- binding precedence: $* > \times > +$

Definition

Kleene Algebra consists of set K with distinguished elements $0, 1 \in K$ and operations $* : K \to K$ and $+, \times : K \times K \to K$ such that

$$
\begin{aligned}
a + (b + c) &= (a + b) + c \\
a + b &= b + a \\
a + a &= a \\
a + 0 &= a
\end{aligned}
\qquad
\begin{aligned}
a0 &= 0 \\
0a &= 0 \\
1a &= a \\
a1 &= a
\end{aligned}
\qquad
\begin{aligned}
a(bc) &= (ab)c \\
(a + b)c &= ac + bc \\
a(b + c) &= ab + ac
\end{aligned}
\qquad
\begin{aligned}
1 + aa^* &= a^* \\
1 + a^* a &= a^* \\
ac \leqslant c &\implies a^* c \leqslant c \\
ca \leqslant c &\implies ca^* \leqslant c
\end{aligned}
$$

for all $a, b, c \in K$

Notation

- ab for $a \times b$ $\quad a^*$ for $*(a)$ $\quad a \leqslant b$ for $a + b = b$
- binding precedence: $* > \times > +$

Definition

Kleene Algebra consists of set K with distinguished elements $0, 1 \in K$ and operations $* : K \to K$ and $+, \times : K \times K \to K$ such that (A.1) – (A.13)

$$
\begin{array}{rcl}
a + (b + c) & = & (a + b) + c \\
a + b & = & b + a \\
a + a & = & a \\
a + 0 & = & a
\end{array}
\qquad
\begin{array}{rcl}
a0 & = & 0 \\
0a & = & 0 \\
1a & = & a \\
a1 & = & a
\end{array}
\qquad
\begin{array}{rcl}
a(bc) & = & (ab)c \\
(a + b)c & = & ac + bc \\
a(b + c) & = & ab + ac
\end{array}
\qquad
\begin{array}{c}
1 + aa^* = a^* \\
1 + a^* a = a^* \\
ac \leqslant c \implies a^* c \leqslant c \\
ca \leqslant c \implies ca^* \leqslant c
\end{array}
$$

for all $a, b, c \in K$

Notation

- ab for $a \times b$ $\quad a^*$ for $*(a)$ $\quad a \leqslant b$ for $a + b = b$
- binding precedence: $* > \times > +$

Definition

Kleene Algebra consists of set K with distinguished elements $0, 1 \in K$ and operations $* : K \to K$ and $+, \times : K \times K \to K$ such that (A.1) – (A.13)

$$
\begin{array}{rcl}
a + (b + c) & = & (a + b) + c \\
a + b & = & b + a \\
a + a & = & a \\
a + 0 & = & a
\end{array}
\qquad
\begin{array}{rcl}
a0 & = & 0 \\
0a & = & 0 \\
1a & = & a \\
a1 & = & a
\end{array}
\qquad
\begin{array}{rcl}
a(bc) & = & (ab)c \\
(a + b)c & = & ac + bc \\
a(b + c) & = & ab + ac
\end{array}
\qquad
\begin{array}{c}
1 + aa^* = a^* \\
1 + a^* a = a^* \\
ac \leqslant c \implies a^* c \leqslant c \\
ca \leqslant c \implies ca^* \leqslant c
\end{array}
$$

(A.14) $b + ac \leqslant c \implies a^* b \leqslant c$

for all $a, b, c \in K$

Notation

- ab for $a \times b$ $\quad a^*$ for $*(a)$ $\quad a \leqslant b$ for $a + b = b$
- binding precedence: $* > \times > +$

Definition

Kleene Algebra consists of set K with distinguished elements $0, 1 \in K$ and operations $* : K \to K$ and $+, \times : K \times K \to K$ such that (A.1) – (A.13)

$$
\begin{array}{rcl}
a + (b + c) & = & (a + b) + c \\
a + b & = & b + a \\
a + a & = & a \\
a + 0 & = & a
\end{array}
\qquad
\begin{array}{rcl}
a0 & = & 0 \\
0a & = & 0 \\
1a & = & a \\
a1 & = & a
\end{array}
\qquad
\begin{array}{rcl}
a(bc) & = & (ab)c \\
(a + b)c & = & ac + bc \\
a(b + c) & = & ab + ac
\end{array}
\qquad
\begin{array}{c}
1 + aa^* \;=\; a^* \\
1 + a^* a \;=\; a^* \\
ac \leqslant c \implies a^* c \leqslant c \\
ca \leqslant c \implies ca^* \leqslant c
\end{array}
$$

(A.14) $b + ac \leqslant c \implies a^* b \leqslant c$ (A.15) $b + ca \leqslant c \implies ba^* \leqslant c$

for all $a, b, c \in K$

Notation

- ab for $a \times b$ a^* for $*(a)$ $a \leqslant b$ for $a + b = b$
- binding precedence: $* > \times > +$

Definition

Kleene Algebra consists of set K with distinguished elements $0, 1 \in K$ and operations $* : K \to K$ and $+, \times : K \times K \to K$ such that (A.1) – (A.13)

$$\begin{aligned} a + (b + c) &= (a + b) + c \\ a + b &= b + a \\ a + a &= a \\ a + 0 &= a \end{aligned} \qquad \begin{aligned} a0 &= 0 \\ 0a &= 0 \\ 1a &= a \\ a1 &= a \end{aligned} \qquad \begin{aligned} a(bc) &= (ab)c \\ (a + b)c &= ac + bc \\ a(b + c) &= ab + ac \end{aligned} \qquad \begin{aligned} 1 + aa^* &= a^* \\ 1 + a^*a &= a^* \\ ac \leqslant c &\implies a^*c \leqslant c \\ ca \leqslant c &\implies ca^* \leqslant c \end{aligned}$$

(A.14) $b + ac \leqslant c \implies a^*b \leqslant c$ (A.15) $b + ca \leqslant c \implies ba^* \leqslant c$
(A.16) $(a + b)^* = (a^*b)^*a^*$

for all $a, b, c \in K$

Notation

- ab for $a \times b$ a^* for $*(a)$ $a \leqslant b$ for $a + b = b$
- binding precedence: $* > \times > +$

Definition

Kleene Algebra consists of set K with distinguished elements $0, 1 \in K$ and operations $*: K \to K$ and $+, \times : K \times K \to K$ such that (A.1) – (A.13)

$$
\begin{array}{rcl}
a + (b + c) & = & (a + b) + c \\
a + b & = & b + a \\
a + a & = & a \\
a + 0 & = & a
\end{array}
\qquad
\begin{array}{rcl}
a0 & = & 0 \\
0a & = & 0 \\
1a & = & a \\
a1 & = & a
\end{array}
\qquad
\begin{array}{rcl}
a(bc) & = & (ab)c \\
(a + b)c & = & ac + bc \\
a(b + c) & = & ab + ac
\end{array}
\qquad
\begin{array}{rcl}
1 + aa^* & = & a^* \\
1 + a^*a & = & a^* \\
ac \leqslant c & \Longrightarrow & a^*c \leqslant c \\
ca \leqslant c & \Longrightarrow & ca^* \leqslant c
\end{array}
$$

(A.14) $b + ac \leqslant c \implies a^*b \leqslant c$ (A.15) $b + ca \leqslant c \implies ba^* \leqslant c$
(A.16) $(a + b)^* = (a^*b)^*a^*$ (A.17) $a(ba)^* = (ab)^*a$

for all $a, b, c \in K$

Notation

- ab for $a \times b$ $\quad a^*$ for $*(a)$ $\quad a \leqslant b$ for $a + b = b$
- binding precedence: $* > \times > +$

Theorem

regular sets over some alphabet Σ form Kleene algebra

Theorem

regular sets over some alphabet Σ form Kleene algebra

Example

$$\beta_a \quad \equiv \quad a^* b(a^* b)^* a^* + a^*$$

Theorem

regular sets over some alphabet Σ form Kleene algebra

Example

$$\begin{aligned}\beta_a &\equiv a^*b(a^*b)^*a^* + a^* \\ &= xx^*y + y\end{aligned}$$

Theorem

regular sets over some alphabet Σ form Kleene algebra

Example

$$\begin{aligned} \beta_a &\equiv a^*b(a^*b)^*a^* + a^* \\ &= xx^*y + y \qquad x := (a^*b) \end{aligned}$$

Theorem

regular sets over some alphabet Σ form Kleene algebra

Example

$$\begin{aligned} \beta_a &\equiv a^* b (a^* b)^* a^* + a^* \\ &= x x^* y + y \qquad x := (a^* b) \qquad y := (a^*) \end{aligned}$$

Theorem

regular sets over some alphabet Σ form Kleene algebra

Example

$$\begin{aligned}
\beta_a &\equiv a^*b(a^*b)^*a^* + a^* \\
&= xx^*y + y \qquad x := (a^*b) \qquad y := (a^*) \\
&\equiv (xx^* + \boldsymbol{\varepsilon})y
\end{aligned}$$

Theorem

regular sets over some alphabet Σ form Kleene algebra

Example

$$
\begin{aligned}
\beta_a &\equiv a^*b(a^*b)^*a^* + a^* \\
&= xx^*y + y \qquad x := (a^*b) \qquad y := (a^*) \\
&\equiv (xx^* + \boldsymbol{\varepsilon})y \\
&\equiv x^*y
\end{aligned}
$$

Theorem

regular sets over some alphabet Σ form Kleene algebra

Example

$$
\begin{aligned}
\beta_a &\equiv a^*b(a^*b)^*a^* + a^* \\
&= xx^*y + y \qquad x := (a^*b) \qquad y := (a^*) \\
&\equiv (xx^* + \boldsymbol{\varepsilon})y \\
&\equiv x^*y \\
&= (a^*b)^*a^*
\end{aligned}
$$

Theorem

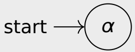
regular sets over some alphabet Σ form Kleene algebra

Example

$$
\begin{aligned}
\beta_a &\equiv a^* b(a^* b)^* a^* + a^* \\
&= xx^* y + y \qquad x := (a^* b) \qquad y := (a^*) \\
&\equiv (xx^* + \boldsymbol{\varepsilon})y \\
&\equiv x^* y \\
&= (a^* b)^* a^* \\
&\equiv (a+b)^*
\end{aligned}
$$

Example

- $\alpha = (a + b)^*$



Lemma

every regular expression α can be transformed into equivalent DFA using derivatives

Example

- $\alpha = (a+b)^*, \ \alpha_a \equiv \alpha$

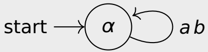
$$\text{start} \longrightarrow \boxed{\alpha} \circlearrowleft a$$

Lemma

every regular expression α can be transformed into equivalent DFA using derivatives

Example

- $\alpha = (a+b)^*$, $\alpha_a \equiv \alpha$, $\alpha_b \equiv \alpha$

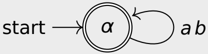


Lemma

every regular expression α can be transformed into equivalent DFA using derivatives

Example

- $\alpha = (a+b)^*$, $\alpha_a \equiv \alpha$, $\alpha_b \equiv \alpha$, $\alpha\downarrow$

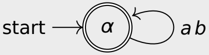


Lemma

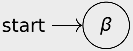
every regular expression α can be transformed into equivalent DFA using derivatives

Example

- $\alpha = (a + b)^*$, $\alpha_a \equiv \alpha$, $\alpha_b \equiv \alpha$, $\alpha\downarrow$



- $\beta = (a^* b)^* a^*$

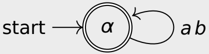


Lemma

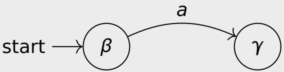
every regular expression α can be transformed into equivalent DFA using derivatives

Example

- $\alpha = (a + b)^*$, $\alpha_a \equiv \alpha$, $\alpha_b \equiv \alpha$, $\alpha\downarrow$



- $\beta = (a^* b)^* a^*$, $\beta_a \equiv a^* b(a^* b)^* a^* + a^* = \gamma$

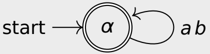


Lemma

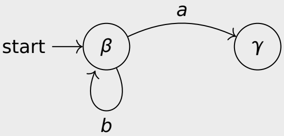
every regular expression α can be transformed into equivalent DFA using derivatives

Example

- $\alpha = (a + b)^*$, $\alpha_a \equiv \alpha$, $\alpha_b \equiv \alpha$, $\alpha\downarrow$



- $\beta = (a^* b)^* a^*$, $\beta_a \equiv a^* b(a^* b)^* a^* + a^* = \gamma$, $\beta_b \equiv \beta$

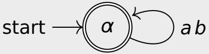


Lemma

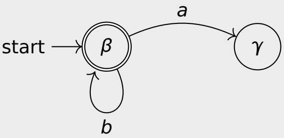
every regular expression α can be transformed into equivalent DFA using derivatives

Example

- $\alpha = (a+b)^*,\ \alpha_a \equiv \alpha\ , \alpha_b \equiv \alpha\ ,\ \alpha\downarrow$



- $\beta = (a^*b)^*a^*\ , \beta_a \equiv a^*b(a^*b)^*a^* + a^* = \gamma\ , \beta_b \equiv \beta\ ,\ \beta\downarrow$

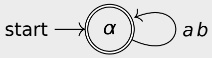


Lemma

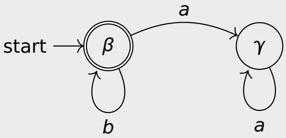
every regular expression α can be transformed into equivalent DFA using derivatives

Example

- $\alpha = (a+b)^*$, $\alpha_a \equiv \alpha$, $\alpha_b \equiv \alpha$, $\alpha{\downarrow}$



- $\beta = (a^* b)^* a^*$, $\beta_a \equiv a^* b (a^* b)^* a^* + a^* = \gamma$, $\beta_b \equiv \beta$, $\beta{\downarrow}$, $\gamma_a \equiv \gamma$

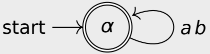


Lemma

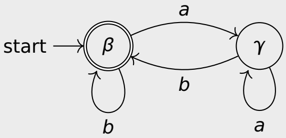
every regular expression α can be transformed into equivalent DFA using derivatives

Example

- $\alpha = (a+b)^*$, $\alpha_a \equiv \alpha$, $\alpha_b \equiv \alpha$, $\alpha\downarrow$



- $\beta = (a^* b)^* a^*$, $\beta_a \equiv a^* b (a^* b)^* a^* + a^* = \gamma$, $\beta_b \equiv \beta$, $\beta\downarrow$, $\gamma_a \equiv \gamma$, $\gamma_b \equiv \beta$

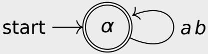


Lemma

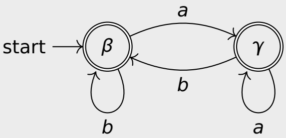
every regular expression α can be transformed into equivalent DFA using derivatives

Example

- $\alpha = (a+b)^*$, $\alpha_a \equiv \alpha$, $\alpha_b \equiv \alpha$, $\alpha\downarrow$



- $\beta = (a^*b)^*a^*$, $\beta_a \equiv a^*b(a^*b)^*a^* + a^* = \gamma$, $\beta_b \equiv \beta$, $\beta\downarrow$, $\gamma_a \equiv \gamma$, $\gamma_b \equiv \beta$, $\gamma\downarrow$

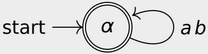


Lemma

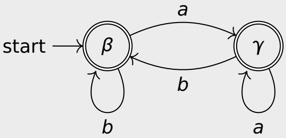
every regular expression α can be transformed into equivalent DFA using derivatives

Example

- $\alpha = (a+b)^*$, $\alpha_a \equiv \alpha$, $\alpha_b \equiv \alpha$, $\alpha\!\downarrow$



- $\beta = (a^*b)^*a^*$, $\beta_a \equiv a^*b(a^*b)^*a^* + a^* = \gamma$, $\beta_b \equiv \beta$, $\beta\!\downarrow$, $\gamma_a \equiv \gamma$, $\gamma_b \equiv \beta$, $\gamma\!\downarrow$



Lemma

every regular expression α can be transformed into equivalent DFA using derivatives
(and ‘easy’ Kleene algebra axioms for simplification)

Definition

bisimulation is binary relation $\sim$ between languages over alphabet Σ

Definition

bisimulation is binary relation $\sim$ between languages over alphabet Σ

1. $A_a \sim B_a$ for all $a \in \Sigma$

Definition

bisimulation is binary relation $\sim$ between languages over alphabet Σ such that if $A \sim B$ then

1. $A_a \sim B_a$ for all $a \in \Sigma$
2. $A{\downarrow} \iff B{\downarrow}$

Definition

bisimulation is binary relation $\sim$ between languages over alphabet Σ such that if $A \sim B$ then

1. $A_a \sim B_a$ for all $a \in \Sigma$
2. $A{\downarrow} \iff B{\downarrow}$

Theorem

$L(\alpha) = L(\beta) \iff L(\alpha) \sim L(\beta)$ for some bisimulation $\sim$

Definition

bisimulation is binary relation $\sim$ between languages over alphabet Σ such that if $A \sim B$ then

1. $A_a \sim B_a \quad$ for all $a \in \Sigma$
2. $A{\downarrow} \iff B{\downarrow}$

Theorem

$L(\alpha) = L(\beta) \iff L(\alpha) \sim L(\beta)$ for some bisimulation $\sim$

Example

$\alpha = (a+b)^*$ and $\beta = (a^*b)^*a^*$

Definition

bisimulation is binary relation $\sim$ between languages over alphabet Σ such that if $A \sim B$ then

1. $A_a \sim B_a \quad$ for all $a \in \Sigma$
2. $A{\downarrow} \iff B{\downarrow}$

Theorem

$L(\alpha) = L(\beta) \iff L(\alpha) \sim L(\beta)$ for some bisimulation $\sim$

Example

$\alpha = (a + b)^*$ and $\beta = (a^* b)^* a^*$

	a	b	
α	α	α	$\downarrow$

Definition

bisimulation is binary relation $\sim$ between languages over alphabet Σ such that if $A \sim B$ then

1. $A_a \sim B_a$ for all $a \in \Sigma$
2. $A{\downarrow} \iff B{\downarrow}$

Theorem

$L(\alpha) = L(\beta) \iff L(\alpha) \sim L(\beta)$ for some bisimulation $\sim$

Example

$\alpha = (a+b)^*$ and $\beta = (a^*b)^*a^*$ and $\gamma = a^*b(a^*b)^*a^* + a^*$

	a	b	
α	α	α	$\downarrow$

	a	b	
β	γ	β	$\downarrow$

Definition

bisimulation is binary relation $\sim$ between languages over alphabet Σ such that if $A \sim B$ then

1. $A_a \sim B_a$ for all $a \in \Sigma$
2. $A{\downarrow} \iff B{\downarrow}$

Theorem

$L(\alpha) = L(\beta) \iff L(\alpha) \sim L(\beta)$ for some bisimulation $\sim$

Example

$\alpha = (a+b)^*$ and $\beta = (a^*b)^*a^*$ and $\gamma = a^*b(a^*b)^*a^* + a^*$

	a	b	
α	α	α	$\downarrow$

	a	b	
β	γ	β	$\downarrow$
γ	γ	β	$\downarrow$

Definition

bisimulation is binary relation $\sim$ between languages over alphabet Σ such that if $A \sim B$ then

1. $A_a \sim B_a$ for all $a \in \Sigma$
2. $A{\downarrow} \iff B{\downarrow}$

Theorem

$L(\alpha) = L(\beta) \iff L(\alpha) \sim L(\beta)$ for some bisimulation $\sim$

Example

$\alpha = (a+b)^*$ and $\beta = (a^*b)^*a^*$ and $\gamma = a^*b(a^*b)^*a^* + a^*$

	a	b	
α	α	α	$\downarrow$

	a	b	
β	γ	β	$\downarrow$
γ	γ	β	$\downarrow$

hence $\{(L(\alpha), L(\beta)), (L(\alpha), L(\gamma))\}$ is bisimulation and thus $L(\alpha) = L(\beta) = L(\gamma)$

Example

$\alpha = ab^*(a+b)^*b$ and $\beta = aa^*(b^*a)^*b$

Example

$\alpha = ab^*(a+b)^*b$ and $\beta = aa^*(b^*a)^*b$

derivatives

$$\begin{aligned}
\alpha_a &= b^*(a+b)^*b =: \alpha_1 & \alpha_b &= \emptyset \\
(\alpha_1)_a &= (a+b)^*b =: \alpha_2 & (\alpha_1)_b &= b^*(a+b)^*b + (a+b)^*b + \varepsilon =: \alpha_3 \\
(\alpha_2)_a &= \alpha_2 & (\alpha_2)_b &= (a+b)^*b + \varepsilon =: \alpha_4 \\
(\alpha_3)_a &= \alpha_2 & (\alpha_3)_b &= \alpha_3 \\
(\alpha_4)_a &= \alpha_2 & (\alpha_4)_b &= \alpha_4
\end{aligned}$$

$$\begin{aligned}
\beta_a &= a^*(b^*a)^*b =: \beta_1 & \beta_b &= \emptyset \\
(\beta_1)_a &= a^*(b^*a)^*b + (b^*a)^*b =: \beta_2 & (\beta_1)_b &= b^*a(b^*a)^*b + \varepsilon =: \beta_3 \\
(\beta_2)_a &= \beta_2 & (\beta_2)_b &= \beta_3 \\
(\beta_3)_a &= (b^*a)^*b =: \beta_4 & (\beta_3)_b &= b^*a(b^*a)^*b =: \beta_5 \\
(\beta_4)_a &= \beta_4 & (\beta_4)_b &= \beta_3 \\
(\beta_5)_a &= \beta_4 & (\beta_5)_b &= \beta_5
\end{aligned}$$

Example

$\alpha = ab^*(a+b)^*b$ and $\beta = aa^*(b^*a)^*b$

tables

	a	b	
α	α_1	$\varnothing$	$\uparrow$
α_1	α_2	α_3	$\uparrow$
α_2	α_2	α_4	$\uparrow$
α_3	α_2	α_3	$\downarrow$
α_4	α_2	α_4	$\downarrow$
$\varnothing$	$\varnothing$	$\varnothing$	$\uparrow$

	a	b	
β	β_1	$\varnothing$	$\uparrow$
β_1	β_2	β_3	$\uparrow$
β_2	β_2	β_3	$\uparrow$
β_3	β_4	β_5	$\downarrow$
β_4	β_4	β_3	$\uparrow$
β_5	β_4	β_5	$\uparrow$
$\varnothing$	$\varnothing$	$\varnothing$	$\uparrow$

Example

$\alpha = ab^*(a+b)^*b$ and $\beta = aa^*(b^*a)^*b$

tables

	a	b	
α	α_1	$\emptyset$	$\uparrow$
α_1	α_2	α_3	$\uparrow$
α_2	α_2	α_4	$\uparrow$
α_3	α_2	α_3	$\downarrow$
α_4	α_2	α_4	$\downarrow$
$\emptyset$	$\emptyset$	$\emptyset$	$\uparrow$

	a	b	
β	β_1	$\emptyset$	$\uparrow$
β_1	β_2	β_3	$\uparrow$
β_2	β_2	β_3	$\uparrow$
β_3	β_4	β_5	$\downarrow$
β_4	β_4	β_3	$\uparrow$
β_5	β_4	β_5	$\uparrow$
$\emptyset$	$\emptyset$	$\emptyset$	$\uparrow$

- any bisimulation $\sim$ satisfying $L(\alpha) \sim L(\beta)$ requires $L(\alpha_1) \sim L(\beta_1)$

Example

$\alpha = ab^*(a+b)^*b$ and $\beta = aa^*(b^*a)^*b$

tables

	a	b	
α	α_1	$\emptyset$	$\uparrow$
α_1	α_2	α_3	$\uparrow$
α_2	α_2	α_4	$\uparrow$
α_3	α_2	α_3	$\downarrow$
α_4	α_2	α_4	$\downarrow$
$\emptyset$	$\emptyset$	$\emptyset$	$\uparrow$

	a	b	
β	β_1	$\emptyset$	$\uparrow$
β_1	β_2	β_3	$\uparrow$
β_2	β_2	β_3	$\uparrow$
β_3	β_4	β_5	$\downarrow$
β_4	β_4	β_3	$\uparrow$
β_5	β_4	β_5	$\uparrow$
$\emptyset$	$\emptyset$	$\emptyset$	$\uparrow$

- any bisimulation $\sim$ satisfying $L(\alpha) \sim L(\beta)$ requires $L(\alpha_3) \sim L(\beta_3)$

Example

$\alpha = ab^*(a+b)^*b$ and $\beta = aa^*(b^*a)^*b$

tables

	a	b	
α	α_1	$\emptyset$	$\uparrow$
α_1	α_2	α_3	$\uparrow$
α_2	α_2	α_4	$\uparrow$
α_3	α_2	α_3	$\downarrow$
α_4	α_2	α_4	$\downarrow$
$\emptyset$	$\emptyset$	$\emptyset$	$\uparrow$

	a	b	
β	β_1	$\emptyset$	$\uparrow$
β_1	β_2	β_3	$\uparrow$
β_2	β_2	β_3	$\uparrow$
β_3	β_4	β_5	$\downarrow$
β_4	β_4	β_3	$\uparrow$
β_5	β_4	β_5	$\uparrow$
$\emptyset$	$\emptyset$	$\emptyset$	$\uparrow$

- any bisimulation $\sim$ satisfying $L(\alpha) \sim L(\beta)$ requires $L(\alpha_3) \sim L(\beta_5)$

Example

$\alpha = ab^*(a+b)^*b$ and $\beta = aa^*(b^*a)^*b$

tables

	a	b	
α	α_1	$\emptyset$	$\uparrow$
α_1	α_2	α_3	$\uparrow$
α_2	α_2	α_4	$\uparrow$
α_3	α_2	α_3	$\downarrow$
α_4	α_2	α_4	$\downarrow$
$\emptyset$	$\emptyset$	$\emptyset$	$\uparrow$

	a	b	
β	β_1	$\emptyset$	$\uparrow$
β_1	β_2	β_3	$\uparrow$
β_2	β_2	β_3	$\uparrow$
β_3	β_4	β_5	$\downarrow$
β_4	β_4	β_3	$\uparrow$
β_5	β_4	β_5	$\uparrow$
$\emptyset$	$\emptyset$	$\emptyset$	$\uparrow$

- any bisimulation $\sim$ satisfying $L(\alpha) \sim L(\beta)$ requires $L(\alpha_3) \sim L(\beta_5)$
- $L(\alpha_3)\downarrow$ and $L(\beta_5)\uparrow$

Example

$\alpha = ab^*(a+b)^*b$ and $\beta = aa^*(b^*a)^*b$

tables

	a	b	
α	α_1	$\emptyset$	$\uparrow$
α_1	α_2	α_3	$\uparrow$
α_2	α_2	α_4	$\uparrow$
α_3	α_2	α_3	$\downarrow$
α_4	α_2	α_4	$\downarrow$
$\emptyset$	$\emptyset$	$\emptyset$	$\uparrow$

	a	b	
β	β_1	$\emptyset$	$\uparrow$
β_1	β_2	β_3	$\uparrow$
β_2	β_2	β_3	$\uparrow$
β_3	β_4	β_5	$\downarrow$
β_4	β_4	β_3	$\uparrow$
β_5	β_4	β_5	$\uparrow$
$\emptyset$	$\emptyset$	$\emptyset$	$\uparrow$

- any bisimulation $\sim$ satisfying $L(\alpha) \sim L(\beta)$ requires $L(\alpha_3) \sim L(\beta_5)$
- $L(\alpha_3)\downarrow$ and $L(\beta_5)\uparrow$ ↯

Example

$\alpha = ab^*(a+b)^*b$ and $\beta = aa^*(b^*a)^*b$

tables

	a	b	
α	α_1	$\varnothing$	$\uparrow$
α_1	α_2	α_3	$\uparrow$
α_2	α_2	α_4	$\uparrow$
α_3	α_2	α_3	$\downarrow$
α_4	α_2	α_4	$\downarrow$
$\varnothing$	$\varnothing$	$\varnothing$	$\uparrow$

	a	b	
β	β_1	$\varnothing$	$\uparrow$
β_1	β_2	β_3	$\uparrow$
β_2	β_2	β_3	$\uparrow$
β_3	β_4	β_5	$\downarrow$
β_4	β_4	β_3	$\uparrow$
β_5	β_4	β_5	$\uparrow$
$\varnothing$	$\varnothing$	$\varnothing$	$\uparrow$

- any bisimulation $\sim$ satisfying $L(\alpha) \sim L(\beta)$ requires $L(\alpha_3) \sim L(\beta_5)$
- $L(\alpha_3)\downarrow$ and $L(\beta_5)\uparrow$ ⚡
- $L(\alpha) \neq L(\beta)$

Example

$\alpha = ab^*(a+b)^*b$ and $\beta = aa^*(b^*a)^*b$

tables

	a	b	
α	α_1	$\varnothing$	$\uparrow$
α_1	α_2	α_3	$\uparrow$
α_2	α_2	α_4	$\uparrow$
α_3	α_2	α_3	$\downarrow$
α_4	α_2	α_4	$\downarrow$
$\varnothing$	$\varnothing$	$\varnothing$	$\uparrow$

	a	b	
β	β_1	$\varnothing$	$\uparrow$
β_1	β_2	β_3	$\uparrow$
β_2	β_2	β_3	$\uparrow$
β_3	β_4	β_5	$\downarrow$
β_4	β_4	β_3	$\uparrow$
β_5	β_4	β_5	$\uparrow$
$\varnothing$	$\varnothing$	$\varnothing$	$\uparrow$

- any bisimulation $\sim$ satisfying $L(\alpha) \sim L(\beta)$ requires $L(\alpha_3) \sim L(\beta_5)$
- $L(\alpha_3)\downarrow$ and $L(\beta_5)\uparrow$ ⚡
- $L(\alpha) \neq L(\beta)$ $\quad$ (witness: $abb \in L(\alpha) \setminus L(\beta)$)

Outline

New Algorithm for Testing Equivalence of NFAs

New Algorithm for Testing Equivalence of NFAs

- bisimulation up to congruence

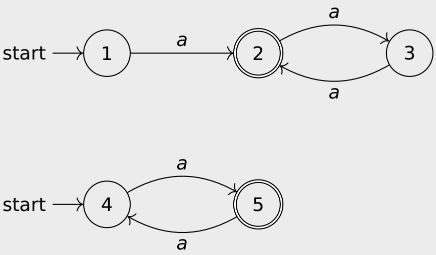
New Algorithm for Testing Equivalence of NFAs

- bisimulation up to congruence
- developed by F. Bonchi and D. Pous (POPL 2013, CACM 2015)

New Algorithm for Testing Equivalence of NFAs

- bisimulation up to congruence
- developed by F. Bonchi and D. Pous (POPL 2013, CACM 2015)
- first: DFAs (by example)

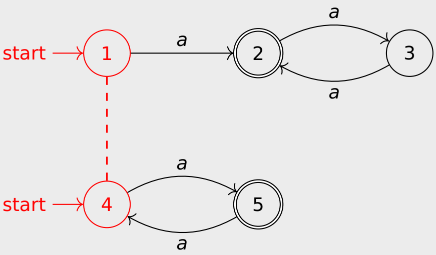
Example



New Algorithm for Testing Equivalence of NFAs

- bisimulation up to congruence
- developed by F. Bonchi and D. Pous (POPL 2013, CACM 2015)
- first: DFAs (by example)

Example

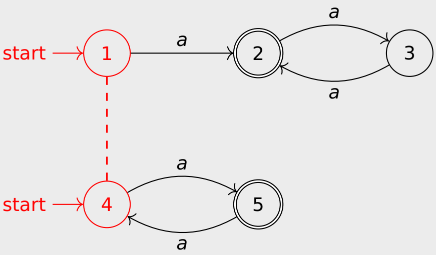


$1 \sim 4$?

New Algorithm for Testing Equivalence of NFAs

- bisimulation up to congruence
- developed by F. Bonchi and D. Pous (POPL 2013, CACM 2015)
- first: DFAs (by example)

Example

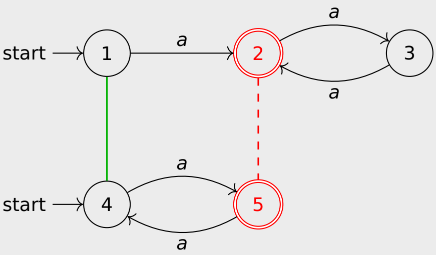


$1 \sim 4$? $1\uparrow$ $4\uparrow$

New Algorithm for Testing Equivalence of NFAs

- bisimulation up to congruence
- developed by F. Bonchi and D. Pous (POPL 2013, CACM 2015)
- first: DFAs (by example)

Example



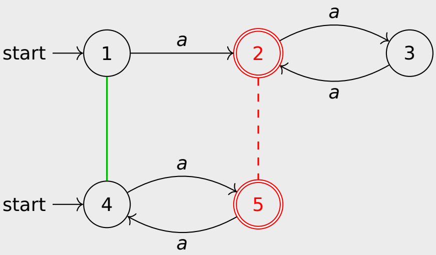
$1 \sim 4$ $1\uparrow$ $4\uparrow$

$2 \sim 5$?

New Algorithm for Testing Equivalence of NFAs

- bisimulation up to congruence
- developed by F. Bonchi and D. Pous (POPL 2013, CACM 2015)
- first: DFAs (by example)

Example



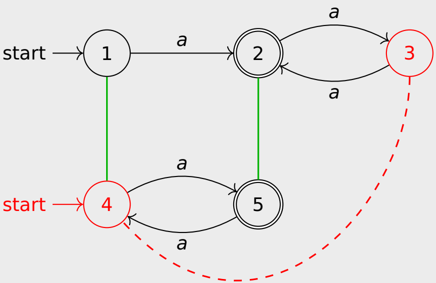
$1 \sim 4$ $\quad 1\uparrow \quad 4\uparrow$

$2 \sim 5$? $\quad 2\downarrow \quad 5\downarrow$

New Algorithm for Testing Equivalence of NFAs

- bisimulation up to congruence
- developed by F. Bonchi and D. Pous (POPL 2013, CACM 2015)
- first: DFAs (by example)

Example



$1 \sim 4$ $\quad 1\!\uparrow \quad 4\!\uparrow$

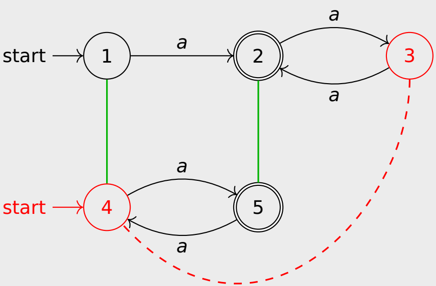
$2 \sim 5$ $\quad 2\!\downarrow \quad 5\!\downarrow$

$3 \sim 4$?

New Algorithm for Testing Equivalence of NFAs

- bisimulation up to congruence
- developed by F. Bonchi and D. Pous (POPL 2013, CACM 2015)
- first: DFAs (by example)

Example



$1 \sim 4$ $1\!\uparrow$ $4\!\uparrow$

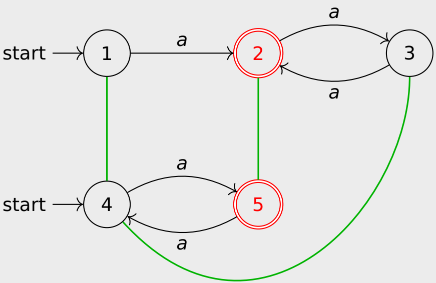
$2 \sim 5$ $2\!\downarrow$ $5\!\downarrow$

$3 \sim 4$? $3\!\uparrow$ $4\!\uparrow$

New Algorithm for Testing Equivalence of NFAs

- bisimulation up to congruence
- developed by F. Bonchi and D. Pous (POPL 2013, CACM 2015)
- first: DFAs (by example)

Example

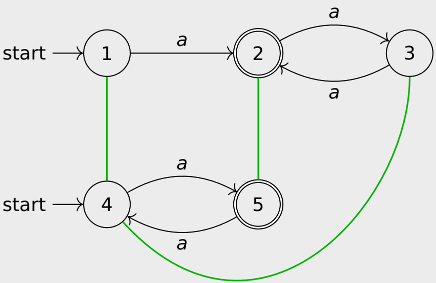


$1 \sim 4$	$1\uparrow$	$4\uparrow$
$2 \sim 5$	$2\downarrow$	$5\downarrow$
$3 \sim 4$	$3\uparrow$	$4\uparrow$
$2 \sim 5$		

New Algorithm for Testing Equivalence of NFAs

- bisimulation up to congruence
- developed by F. Bonchi and D. Pous (POPL 2013, CACM 2015)
- first: DFAs (by example)

Example



$1 \sim 4$	$1\uparrow$	$4\uparrow$
$2 \sim 5$	$2\downarrow$	$5\downarrow$
$3 \sim 4$	$3\uparrow$	$4\uparrow$
$2 \sim 5$	$\sim$ is bisimulation	

Remark

bisimulation relates states with same observable behaviour

Remark

bisimulation relates states with same observable behaviour

Definition

- bisimulation is binary relation R on states Q of DFA $M = (Q, \Sigma, \delta, s, F)$ such that for all $p \, R \, q$
  1. $p \in F \iff q \in F$

Remark

bisimulation relates states with same observable behaviour

Definition

- bisimulation is binary relation R on states Q of DFA $M = (Q, \Sigma, \delta, s, F)$ such that for all $p\,R\,q$
  1. $p \in F \iff q \in F$
  2. $\delta(p, a)\ R\ \delta(q, a) \quad$ for all $a \in \Sigma$

Remark

bisimulation relates states with same observable behaviour

Definition

- bisimulation is binary relation R on states Q of DFA $M = (Q, \Sigma, \delta, s, F)$ such that for all $p\,R\,q$
  1. $p \in F \iff q \in F$
  2. $\delta(p, a)\ R\ \delta(q, a)$ for all $a \in \Sigma$
- states p and q are bisimilar ($p \sim q$) if $p\ R\ q$ for some bisimulation R

Remark

bisimulation relates states with same observable behaviour

Definition

- **bisimulation** is binary relation R on states Q of DFA $M = (Q, \Sigma, \delta, s, F)$ such that for all $p\,R\,q$
  1. $p \in F \iff q \in F$
  2. $\delta(p, a)\ R\ \delta(q, a)$ for all $a \in \Sigma$
- states p and q are **bisimilar** ($p \sim q$) if $p\ R\ q$ for some bisimulation R

Example

- $=$ (identity relation)

Remark

bisimulation relates states with same observable behaviour

Definition

- bisimulation is binary relation R on states Q of DFA $M = (Q, \Sigma, \delta, s, F)$ such that for all $p\,R\,q$
  1. $p \in F \iff q \in F$
  2. $\delta(p, a)\ R\ \delta(q, a)$ for all $a \in \Sigma$
- states p and q are bisimilar ($p \sim q$) if $p\ R\ q$ for some bisimulation R

Example

- $=$ (identity relation)
- $\approx$ (indistinguishability relation of lecture 5)

Remark

bisimulation relates states with same observable behaviour

Definition

- bisimulation is binary relation R on states Q of DFA $M = (Q, \Sigma, \delta, s, F)$ such that for all $p\,R\,q$
  1. $p \in F \quad \Longleftrightarrow \quad q \in F$
  2. $\delta(p, a)\ R\ \delta(q, a) \quad$ for all $a \in \Sigma$
- states p and q are bisimilar $(p \sim q)$ if $p\ R\ q$ for some bisimulation R

Remarks

- bisimilarity is equivalence relation

Remark

bisimulation relates states with same observable behaviour

Definition

- **bisimulation** is binary relation R on states Q of DFA $M = (Q, \Sigma, \delta, s, F)$ such that for all $p\,R\,q$
  1. $p \in F \iff q \in F$
  2. $\delta(p, a)\ R\ \delta(q, a)$ for all $a \in \Sigma$
- states p and q are **bisimilar** ($p \sim q$) if $p\ R\ q$ for some bisimulation R

Remarks

- bisimilarity is equivalence relation
- $L(M, p) := \{x \in \Sigma^* \mid \widehat{\delta}(p, x) \in F\}$

Remark

bisimulation relates states with same observable behaviour

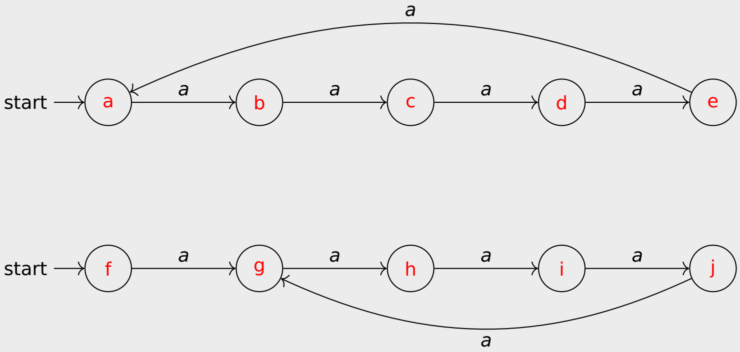
Definition

- bisimulation is binary relation R on states Q of DFA $M = (Q, \Sigma, \delta, s, F)$ such that for all $p\,R\,q$
  1. $p \in F \iff q \in F$
  2. $\delta(p, a)\ R\ \delta(q, a)$ for all $a \in \Sigma$
- states p and q are bisimilar ($p \sim q$) if $p\ R\ q$ for some bisimulation R

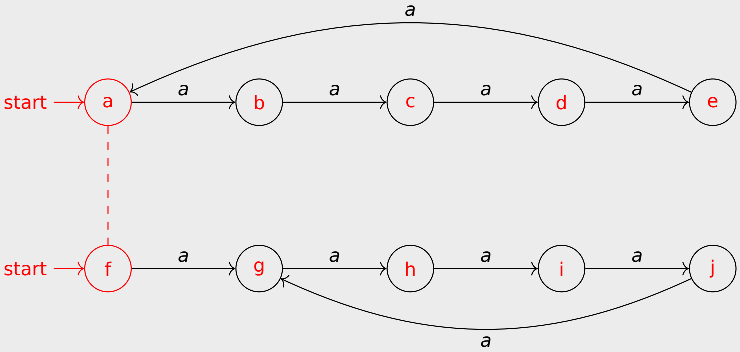
Remarks

- bisimilarity is equivalence relation
- $L(M, p) := \{x \in \Sigma^* \mid \widehat{\delta}(p, x) \in F\}$
- $p \sim q \iff L(M, p) \sim L(M, q)$

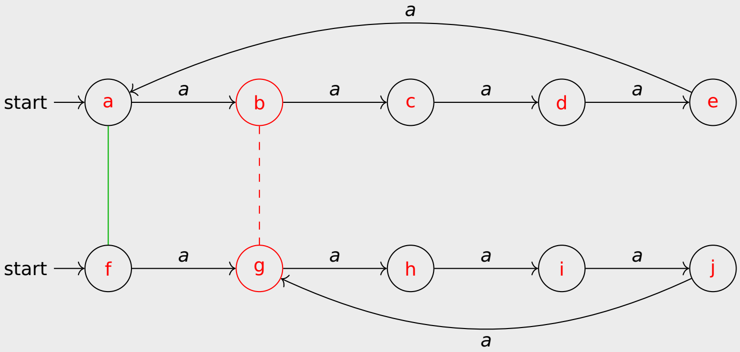
Example



Example

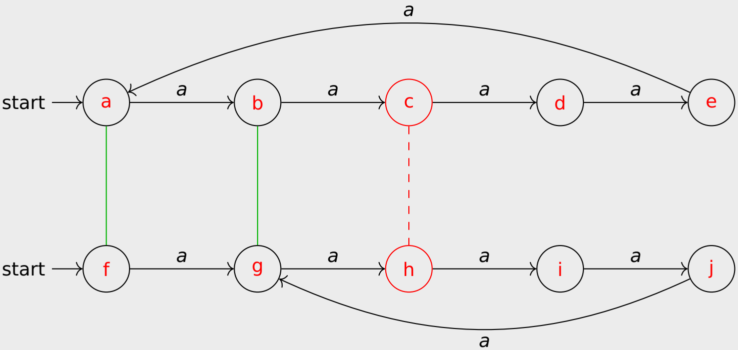


Example



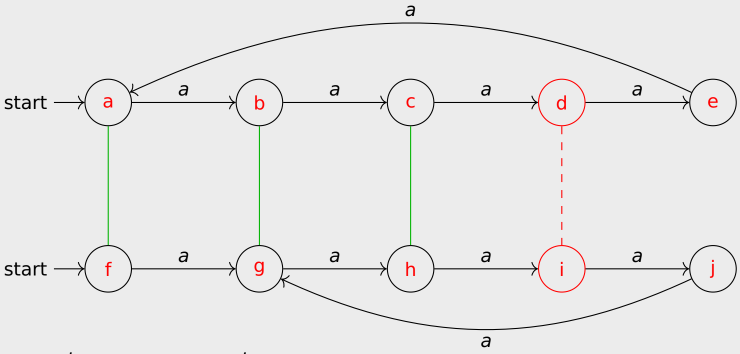
$a \sim f$

Example



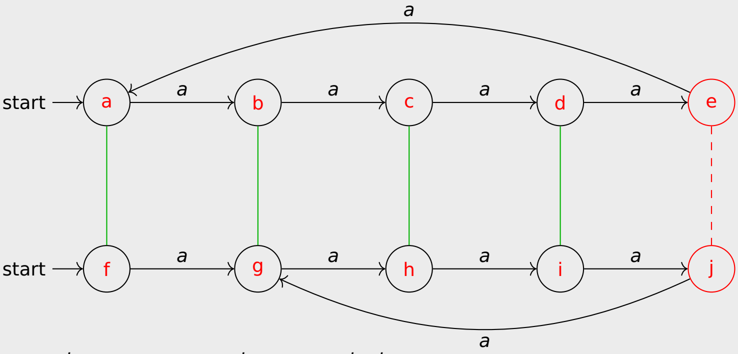
$a \sim f$ $b \sim g$

Example



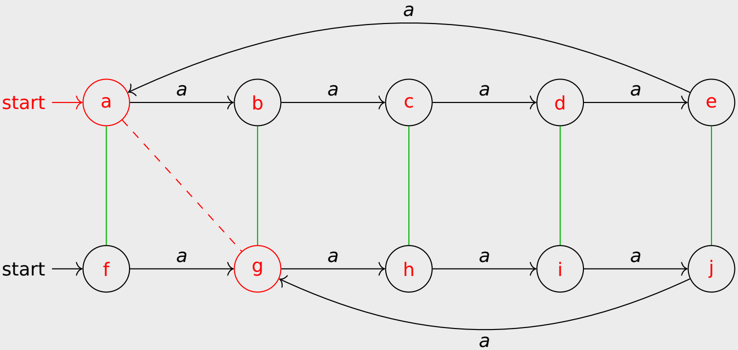
$a \sim f$ $b \sim g$ $c \sim h$

Example



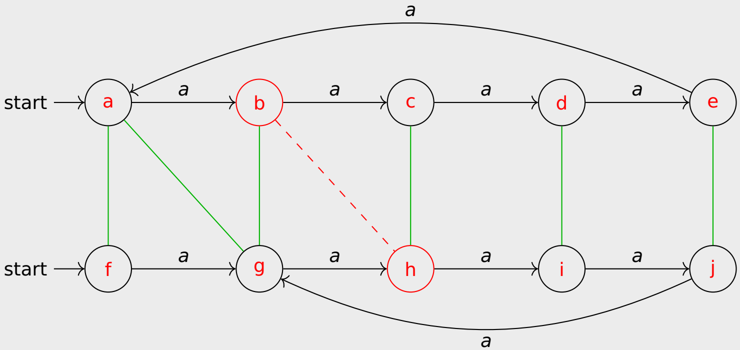
$a \sim f$ $b \sim g$ $c \sim h$ $d \sim i$

Example



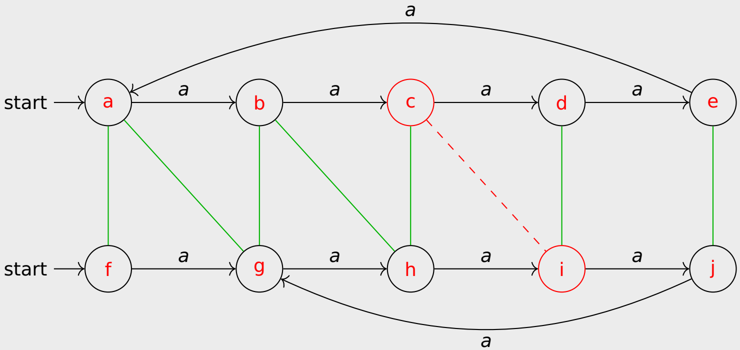
$a \sim f$ $\quad$ $b \sim g$ $\quad$ $c \sim h$ $\quad$ $d \sim i$ $\quad$ $e \sim j$

Example



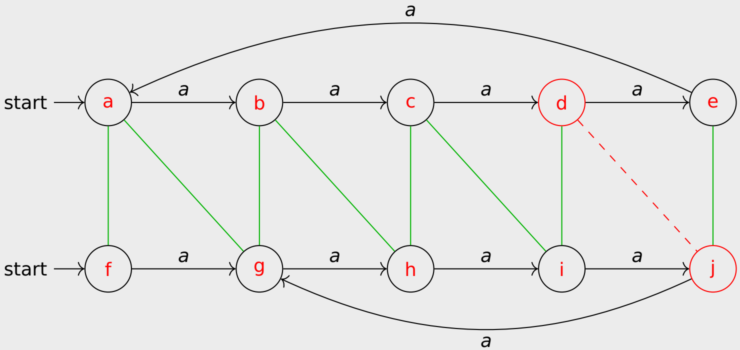
$a \sim f$ $b \sim g$ $c \sim h$ $d \sim i$ $e \sim j$ $a \sim g$

Example



$a \sim f$ $\quad b \sim g$ $\quad c \sim h$ $\quad d \sim i$ $\quad e \sim j$ $\quad a \sim g$ $\quad b \sim h$

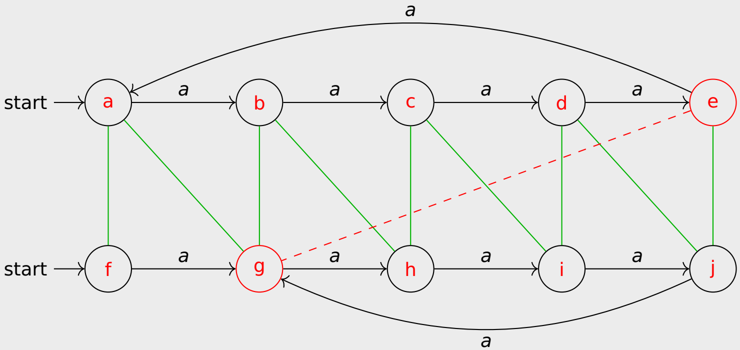
Example



$a \sim f$ $b \sim g$ $c \sim h$ $d \sim i$ $e \sim j$ $a \sim g$ $b \sim h$

$c \sim i$

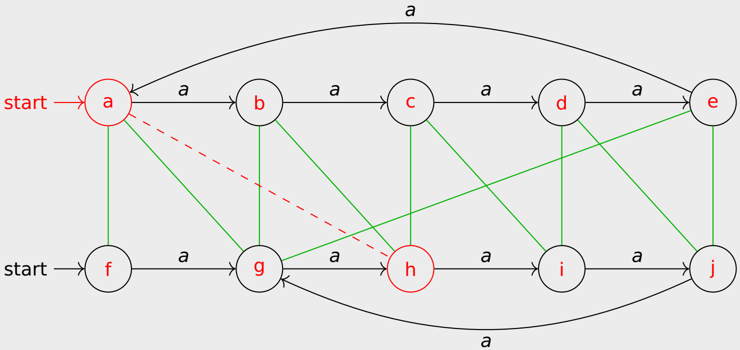
Example



$a \sim f$ $\quad b \sim g$ $\quad c \sim h$ $\quad d \sim i$ $\quad e \sim j$ $\quad a \sim g$ $\quad b \sim h$

$c \sim i$ $\quad d \sim j$

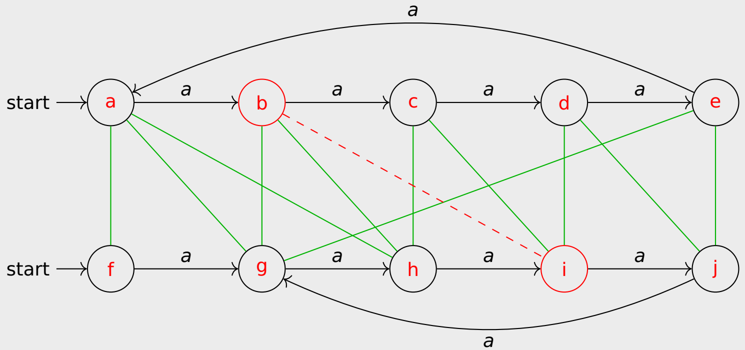
Example



$a \sim f$ $b \sim g$ $c \sim h$ $d \sim i$ $e \sim j$ $a \sim g$ $b \sim h$

$c \sim i$ $d \sim j$ $e \sim g$

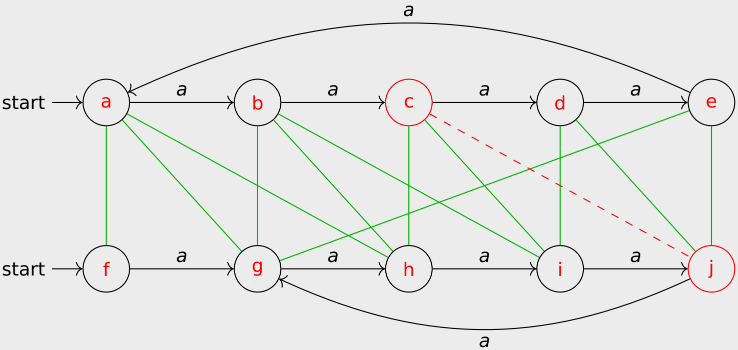
Example



$a \sim f$ $b \sim g$ $c \sim h$ $d \sim i$ $e \sim j$ $a \sim g$ $b \sim h$

$c \sim i$ $d \sim j$ $e \sim g$ $a \sim h$

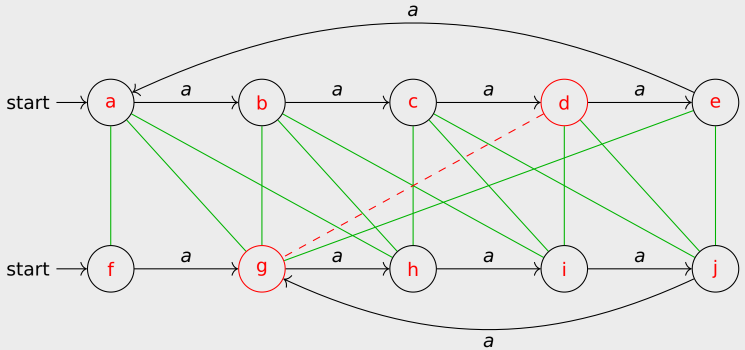
Example



$a \sim f$ $\quad b \sim g$ $\quad c \sim h$ $\quad d \sim i$ $\quad e \sim j$ $\quad a \sim g$ $\quad b \sim h$

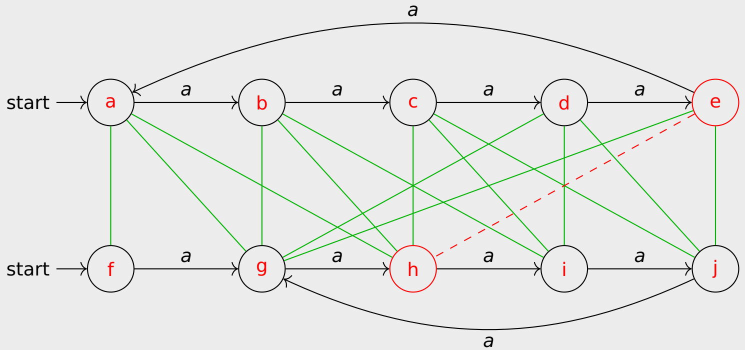
$c \sim i$ $\quad d \sim j$ $\quad e \sim g$ $\quad a \sim h$ $\quad b \sim i$

Example



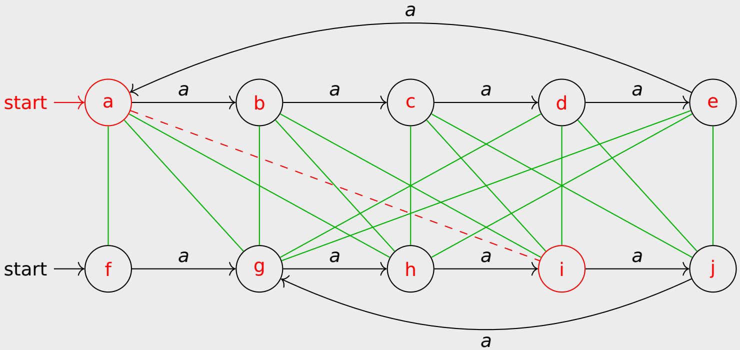
$a \sim f$	$b \sim g$	$c \sim h$	$d \sim i$	$e \sim j$	$a \sim g$	$b \sim h$
$c \sim i$	$d \sim j$	$e \sim g$	$a \sim h$	$b \sim i$	$c \sim j$	

Example



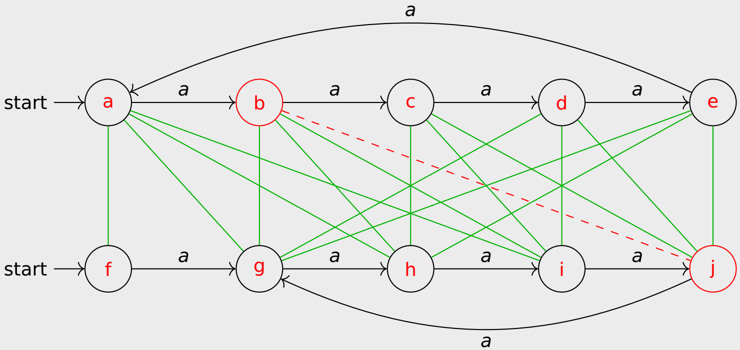
$a \sim f$	$b \sim g$	$c \sim h$	$d \sim i$	$e \sim j$	$a \sim g$	$b \sim h$
$c \sim i$	$d \sim j$	$e \sim g$	$a \sim h$	$b \sim i$	$c \sim j$	$d \sim g$

Example



$a \sim f$	$b \sim g$	$c \sim h$	$d \sim i$	$e \sim j$	$a \sim g$	$b \sim h$
$c \sim i$	$d \sim j$	$e \sim g$	$a \sim h$	$b \sim i$	$c \sim j$	$d \sim g$
$e \sim h$						

Example

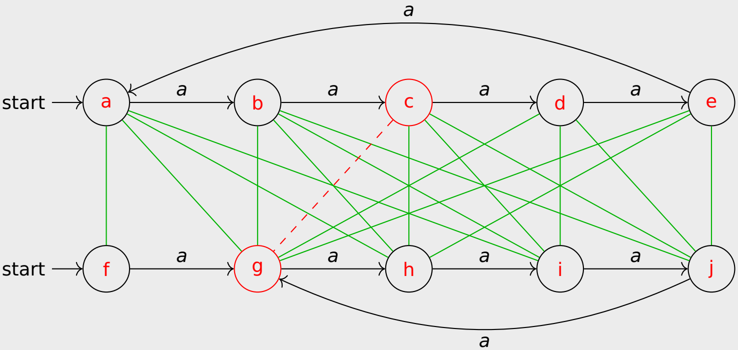


$a \sim f$ $b \sim g$ $c \sim h$ $d \sim i$ $e \sim j$ $a \sim g$ $b \sim h$

$c \sim i$ $d \sim j$ $e \sim g$ $a \sim h$ $b \sim i$ $c \sim j$ $d \sim g$

$e \sim h$ $a \sim i$

Example

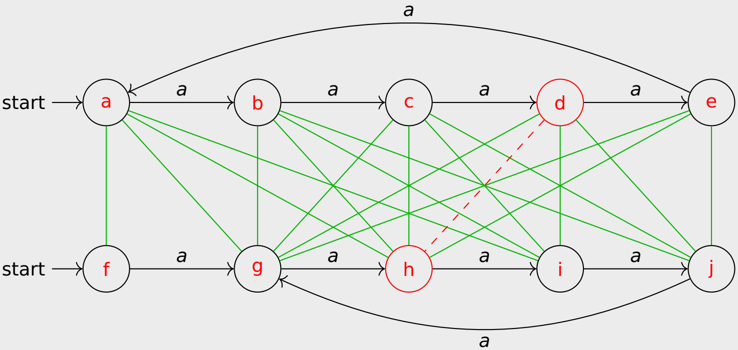


$a \sim f$ $\quad b \sim g$ $\quad c \sim h$ $\quad d \sim i$ $\quad e \sim j$ $\quad a \sim g$ $\quad b \sim h$

$c \sim i$ $\quad d \sim j$ $\quad e \sim g$ $\quad a \sim h$ $\quad b \sim i$ $\quad c \sim j$ $\quad d \sim g$

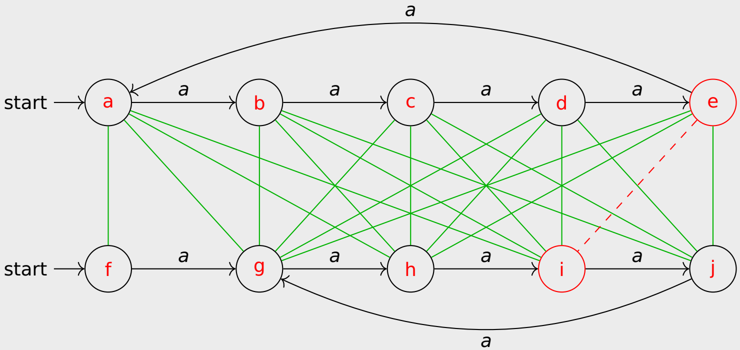
$e \sim h$ $\quad a \sim i$ $\quad b \sim j$

Example



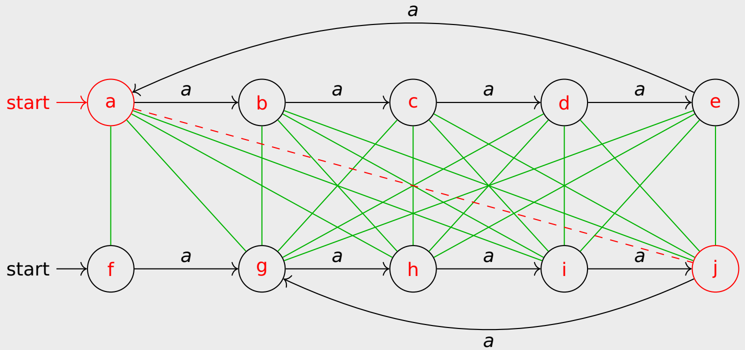
$a \sim f$ $b \sim g$ $c \sim h$ $d \sim i$ $e \sim j$ $a \sim g$ $b \sim h$
$c \sim i$ $d \sim j$ $e \sim g$ $a \sim h$ $b \sim i$ $c \sim j$ $d \sim g$
$e \sim h$ $a \sim i$ $b \sim j$ $c \sim g$

Example



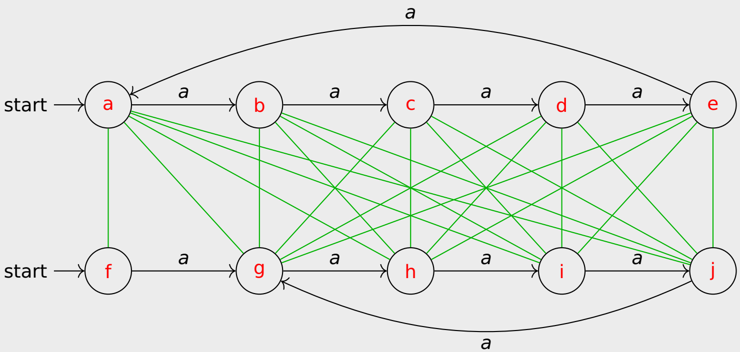
$a \sim f$	$b \sim g$	$c \sim h$	$d \sim i$	$e \sim j$	$a \sim g$	$b \sim h$
$c \sim i$	$d \sim j$	$e \sim g$	$a \sim h$	$b \sim i$	$c \sim j$	$d \sim g$
$e \sim h$	$a \sim i$	$b \sim j$	$c \sim g$	$d \sim h$		

Example



$a \sim f$	$b \sim g$	$c \sim h$	$d \sim i$	$e \sim j$	$a \sim g$	$b \sim h$
$c \sim i$	$d \sim j$	$e \sim g$	$a \sim h$	$b \sim i$	$c \sim j$	$d \sim g$
$e \sim h$	$a \sim i$	$b \sim j$	$c \sim g$	$d \sim h$	$e \sim i$	

Example

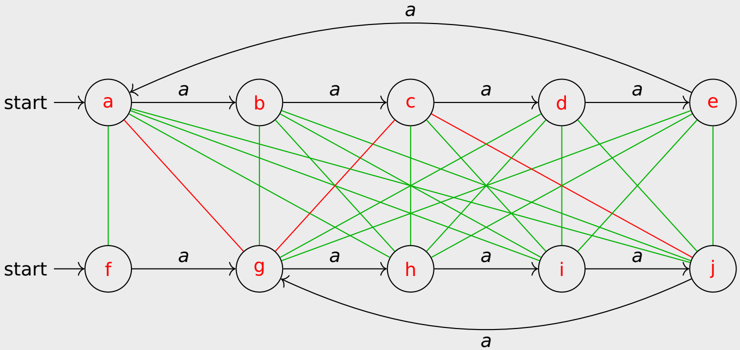


$a \sim f$	$b \sim g$	$c \sim h$	$d \sim i$	$e \sim j$	$a \sim g$	$b \sim h$
$c \sim i$	$d \sim j$	$e \sim g$	$a \sim h$	$b \sim i$	$c \sim j$	$d \sim g$
$e \sim h$	$a \sim i$	$b \sim j$	$c \sim g$	$d \sim h$	$e \sim i$	$a \sim j$

bisimulation

21 comparisons

Example



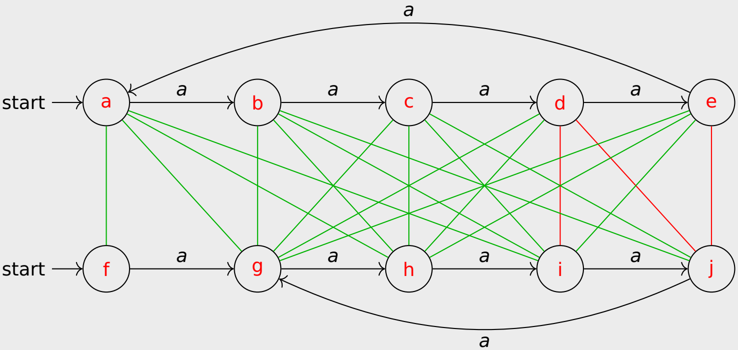
$a \sim f$	$b \sim g$	$c \sim h$	$d \sim i$	$e \sim j$	$a \sim g$	$b \sim h$
$c \sim i$	$d \sim j$	$e \sim g$	$a \sim h$	$b \sim i$	$c \sim j$	$d \sim g$
$e \sim h$	$a \sim i$	$b \sim j$	$c \sim g$	$d \sim h$	$e \sim i$	$a \sim j$

bisimulation

21 comparisons

$a \sim j$ follows from $a \sim g, c \sim g, c \sim j$

Example



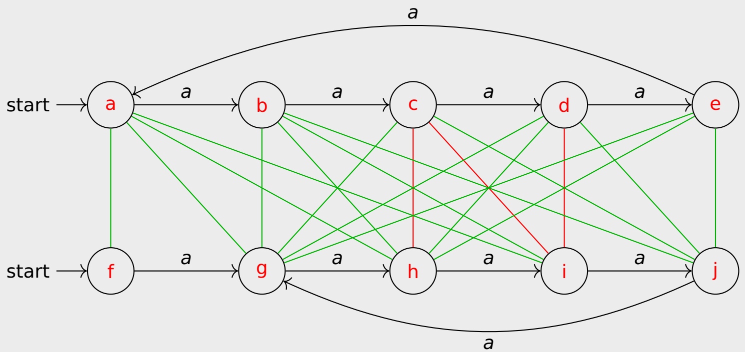
$a \sim f$	$b \sim g$	$c \sim h$	$d \sim i$	$e \sim j$	$a \sim g$	$b \sim h$
$c \sim i$	$d \sim j$	$e \sim g$	$a \sim h$	$b \sim i$	$c \sim j$	$d \sim g$
$e \sim h$	$a \sim i$	$b \sim j$	$c \sim g$	$d \sim h$	$e \sim i$	

bisimulation

20 comparisons

$e \sim i$ follows from $d \sim i, d \sim j, e \sim j$

Example



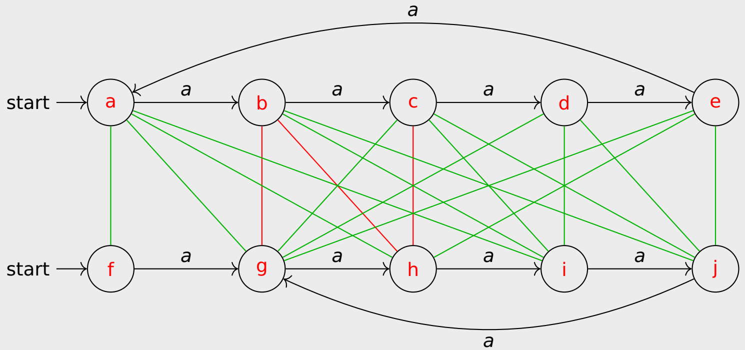
$a \sim f$	$b \sim g$	$c \sim h$	$d \sim i$	$e \sim j$	$a \sim g$	$b \sim h$
$c \sim i$	$d \sim j$	$e \sim g$	$a \sim h$	$b \sim i$	$c \sim j$	$d \sim g$
$e \sim h$	$a \sim i$	$b \sim j$	$c \sim g$	$d \sim h$		

bisimulation

19 comparisons

$d \sim h$ follows from $c \sim h, c \sim i, d \sim i$

Example



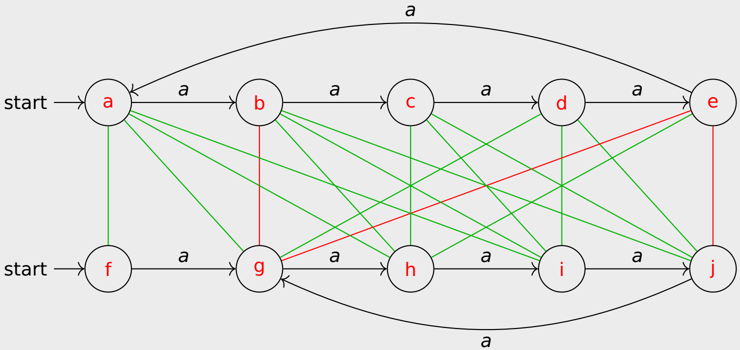
$a \sim f$	$b \sim g$	$c \sim h$	$d \sim i$	$e \sim j$	$a \sim g$	$b \sim h$
$c \sim i$	$d \sim j$	$e \sim g$	$a \sim h$	$b \sim i$	$c \sim j$	$d \sim g$
$e \sim h$	$a \sim i$	$b \sim j$	$c \sim g$			

bisimulation

18 comparisons

$c \sim g$ follows from $b \sim g, b \sim h, c \sim h$

Example



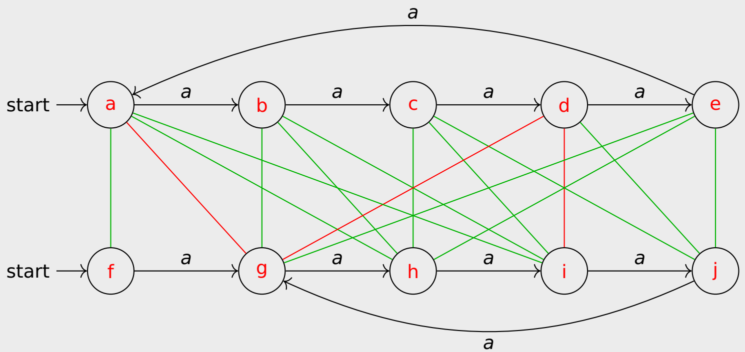
$a \sim f$	$b \sim g$	$c \sim h$	$d \sim i$	$e \sim j$	$a \sim g$	$b \sim h$
$c \sim i$	$d \sim j$	$e \sim g$	$a \sim h$	$b \sim i$	$c \sim j$	$d \sim g$
$e \sim h$	$a \sim i$	$b \sim j$				

bisimulation

17 comparisons

$b \sim j$ follows from $b \sim g, e \sim g, e \sim j$

Example



$a \sim f$ $\quad b \sim g$ $\quad c \sim h$ $\quad d \sim i$ $\quad e \sim j$ $\quad a \sim g$ $\quad b \sim h$

$c \sim i$ $\quad d \sim j$ $\quad e \sim g$ $\quad a \sim h$ $\quad b \sim i$ $\quad c \sim j$ $\quad d \sim g$

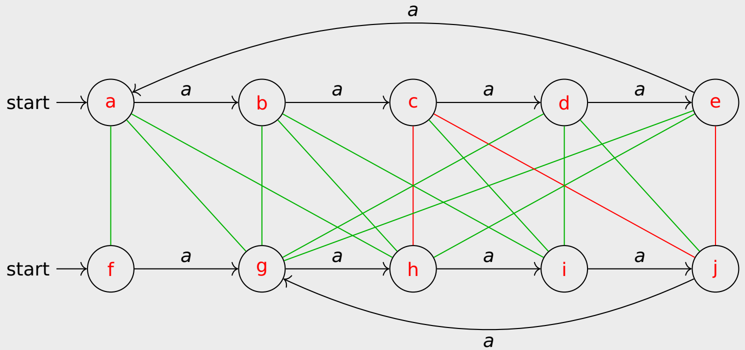
$e \sim h$ $\quad a \sim i$

bisimulation

$a \sim i$ follows from $a \sim g, d \sim g, d \sim i$

16 comparisons

Example



$a \sim f$ $\quad b \sim g$ $\quad c \sim h$ $\quad d \sim i$ $\quad e \sim j$ $\quad a \sim g$ $\quad b \sim h$

$c \sim i$ $\quad d \sim j$ $\quad e \sim g$ $\quad a \sim h$ $\quad b \sim i$ $\quad c \sim j$ $\quad d \sim g$

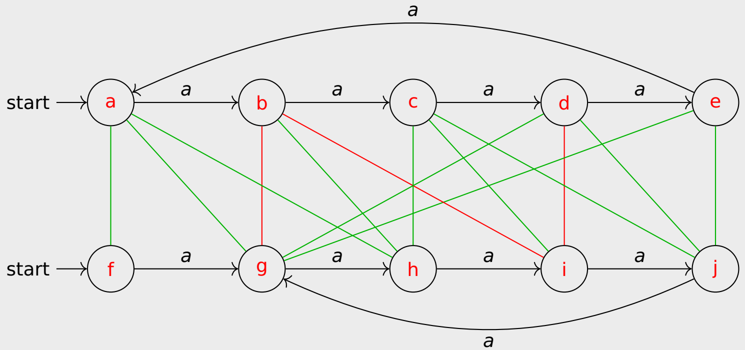
$e \sim h$

bisimulation

$e \sim h$ follows from $c \sim h, c \sim j, e \sim j$

15 comparisons

Example



$a \sim f$ $b \sim g$ $c \sim h$ $d \sim i$ $e \sim j$ $a \sim g$ $b \sim h$

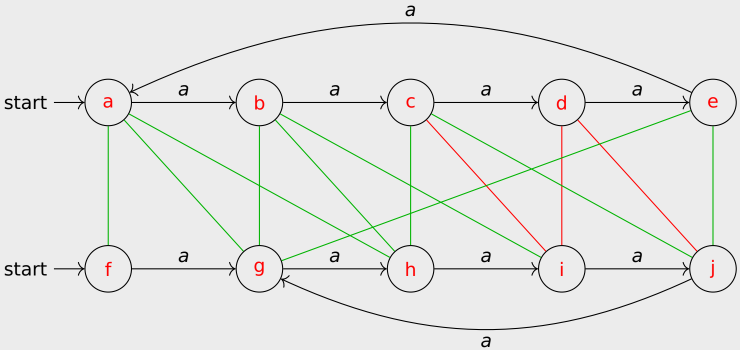
$c \sim i$ $d \sim j$ $e \sim g$ $a \sim h$ $b \sim i$ $c \sim j$ $d \sim g$

bisimulation

$d \sim g$ follows from $b \sim g$, $b \sim i$, $d \sim i$

14 comparisons

Example



$a \sim f$ $b \sim g$ $c \sim h$ $d \sim i$ $e \sim j$ $a \sim g$ $b \sim h$

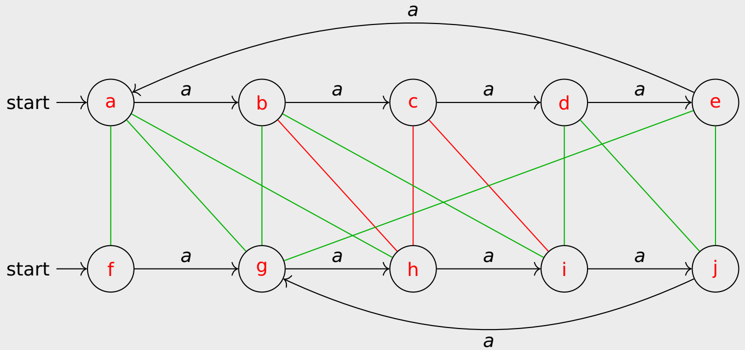
$c \sim i$ $d \sim j$ $e \sim g$ $a \sim h$ $b \sim i$ $c \sim j$

bisimulation

$c \sim j$ follows from $c \sim i, d \sim i, d \sim j$

13 comparisons

Example



$a \sim f$ $\quad b \sim g$ $\quad c \sim h$ $\quad d \sim i$ $\quad e \sim j$ $\quad a \sim g$ $\quad b \sim h$

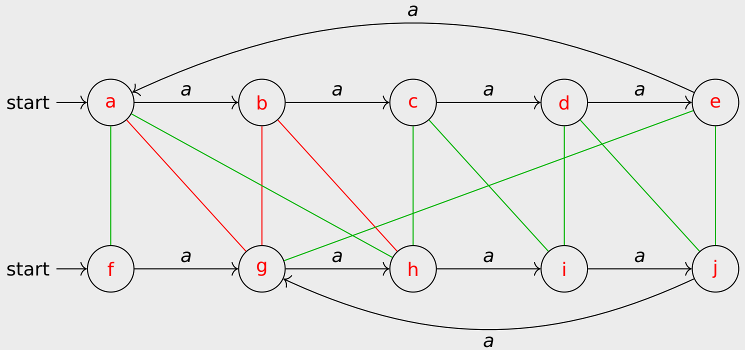
$c \sim i$ $\quad d \sim j$ $\quad e \sim g$ $\quad a \sim h$ $\quad b \sim i$

bisimulation

$b \sim i$ follows from $b \sim h, c \sim h, c \sim i$

12 comparisons

Example



$a \sim f$ $b \sim g$ $c \sim h$ $d \sim i$ $e \sim j$ $a \sim g$ $b \sim h$

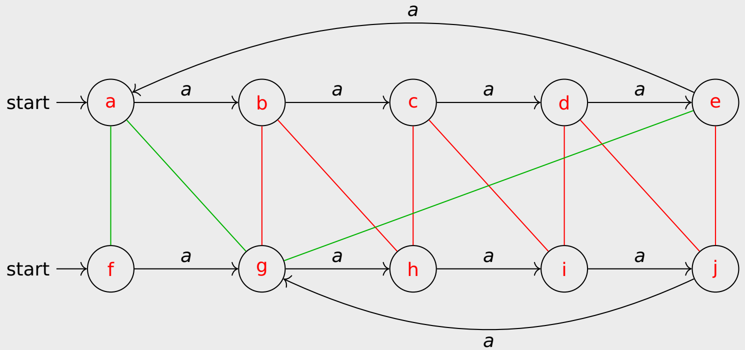
$c \sim i$ $d \sim j$ $e \sim g$ $a \sim h$

bisimulation

$a \sim h$ follows from $a \sim g, b \sim g, b \sim h$

11 comparisons

Example



$a \sim f$ $\quad$ $b \sim g$ $\quad$ $c \sim h$ $\quad$ $d \sim i$ $\quad$ $e \sim j$ $\quad$ $a \sim g$ $\quad$ $b \sim h$

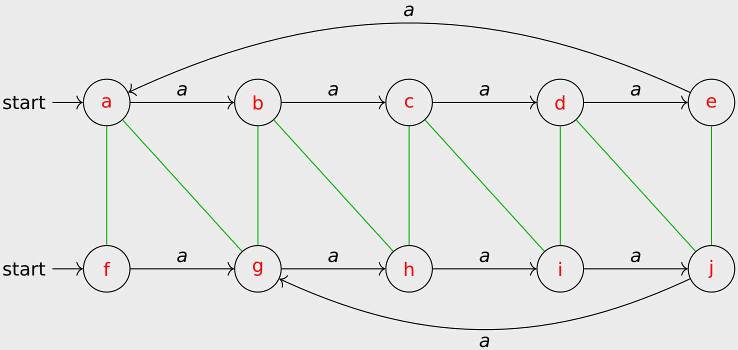
$c \sim i$ $\quad$ $d \sim j$ $\quad$ $e \sim g$

bisimulation

10 comparisons

$e \sim g$ follows from $b \sim g, b \sim h, c \sim h, c \sim i, d \sim i, d \sim j, e \sim j$

Example



$a \sim f$ $b \sim g$ $c \sim h$ $d \sim i$ $e \sim j$ $a \sim g$ $b \sim h$

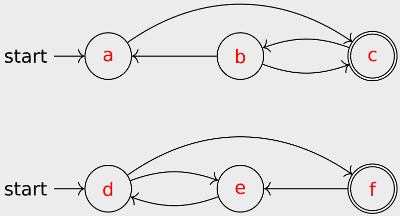
$c \sim i$ $d \sim j$

bisimulation up to equivalence

9 instead of 21 comparisons

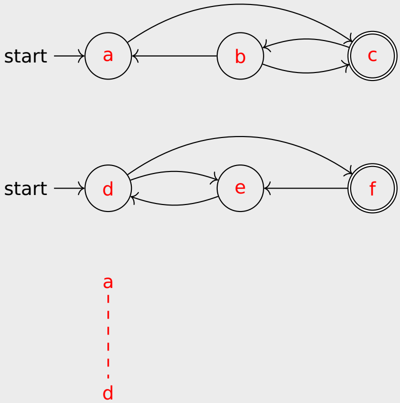
for NFAs on-the-fly determinization is incorporated

Example



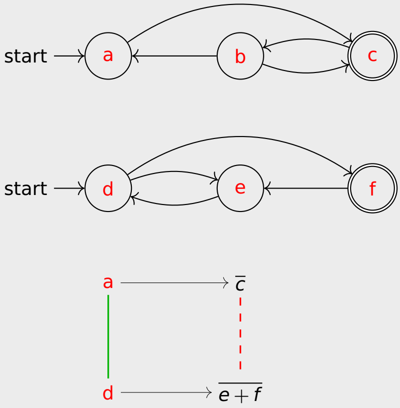
for NFAs on-the-fly determinization is incorporated

Example



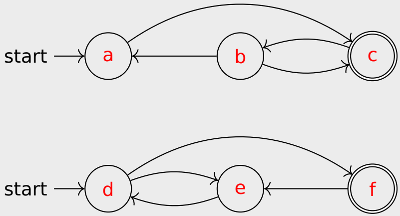
for NFAs on-the-fly determinization is incorporated

Example



for NFAs on-the-fly determinization is incorporated

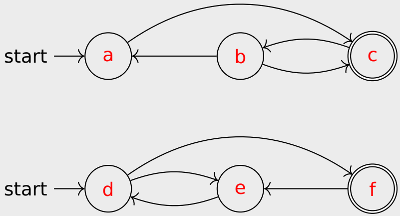
Example



$$
\begin{tikzcd}
a \arrow[r] \arrow[d, dash] & \overline{c} \arrow[r] \arrow[d, dash] & b \arrow[d, dash, dashed] \\
d \arrow[r] & \overline{e+f} \arrow[r] & d+e
\end{tikzcd}
$$

for NFAs on-the-fly determinization is incorporated

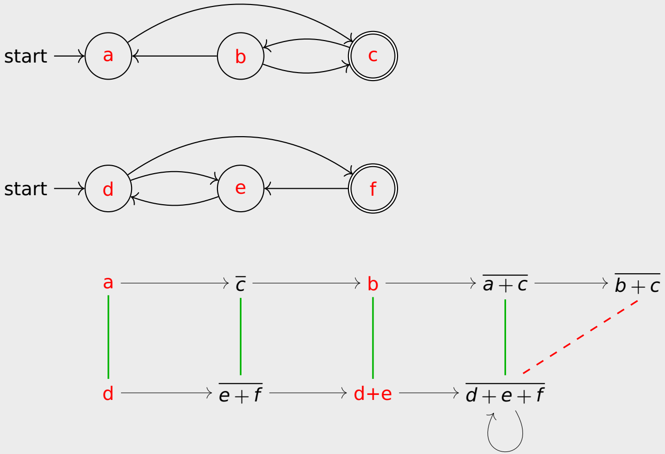
Example



$$\begin{tikzcd}
a \arrow[r] \arrow[d, dash] & \overline{c} \arrow[r] \arrow[d, dash] & b \arrow[r] \arrow[d, dash] & \overline{a+c} \arrow[d, dash, dashed] \\
d \arrow[r] & \overline{e+f} \arrow[r] & d+e \arrow[r] & \overline{d+e+f}
\end{tikzcd}$$

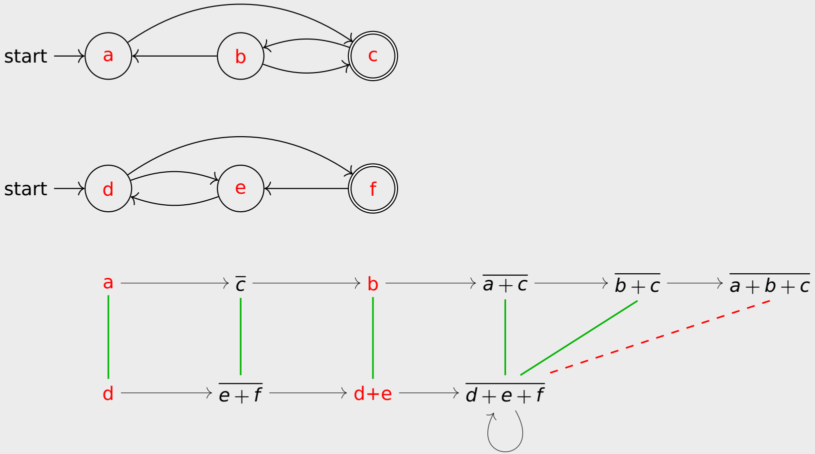
for NFAs on-the-fly determinization is incorporated

Example



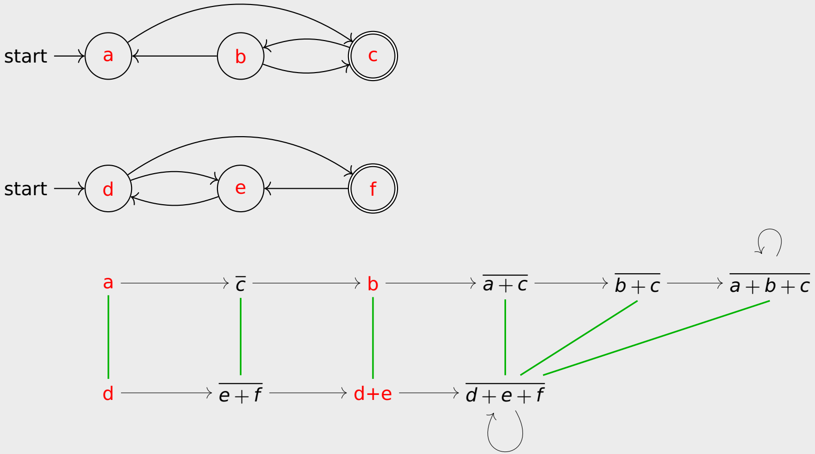
for NFAs on-the-fly determinization is incorporated

Example



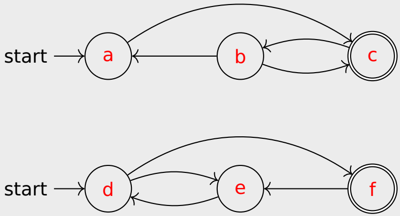
for NFAs on-the-fly determinization is incorporated

Example

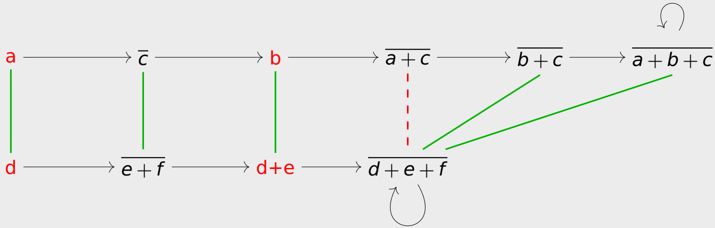


for NFAs on-the-fly determinization is incorporated

Example

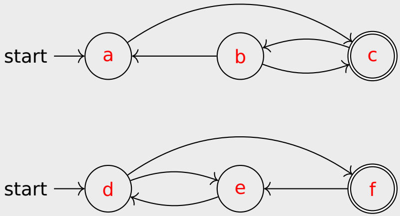


$$a \sim d \ \wedge \ c \sim e + f$$
$$\implies \ a + c \sim d + e + f$$



for NFAs on-the-fly determinization is incorporated

Example



$$a \sim d \ \wedge \ c \sim e + f \implies a + c \sim d + e + f$$

$$\begin{tikzcd}
a \arrow[r] \arrow[d, dash] & \overline{c} \arrow[r] \arrow[d, dash] & b \arrow[d, dash] \\
d \arrow[r] & \overline{e+f} \arrow[r] & d+e
\end{tikzcd}$$

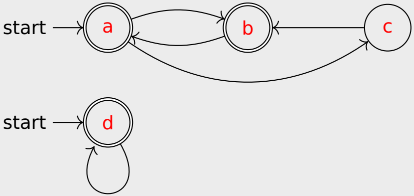
Notation

sets are denoted as sums of their elements; overlining indicates final state sets

Notation

sets are denoted as sums of their elements; overlining indicates final state sets

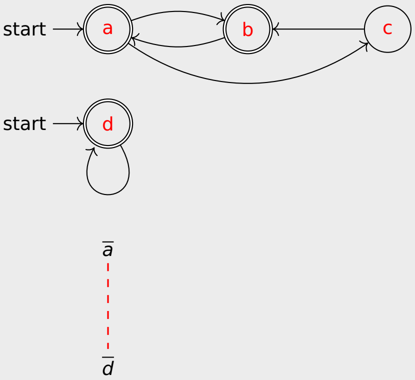
Example



Notation

sets are denoted as sums of their elements; overlining indicates final state sets

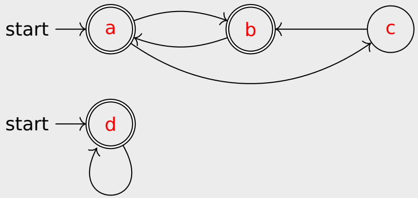
Example

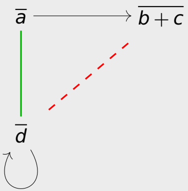


Notation

sets are denoted as sums of their elements; overlining indicates final state sets

Example

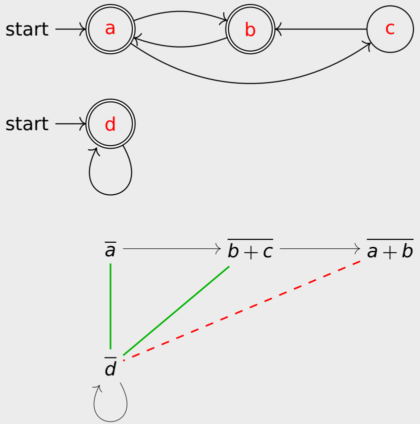




Notation

sets are denoted as sums of their elements; overlining indicates final state sets

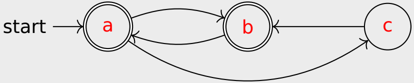
Example

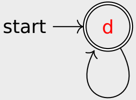


Notation

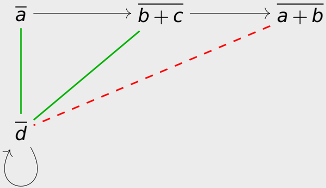
sets are denoted as sums of their elements; overlining indicates final state sets

Example





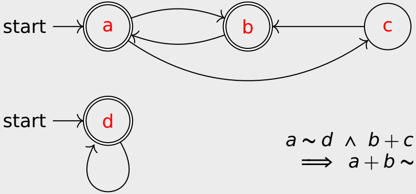
$$a \sim d$$
$$\implies \quad a + b \sim d + b$$



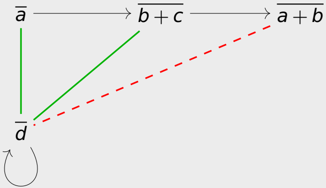
Notation

sets are denoted as sums of their elements; overlining indicates final state sets

Example



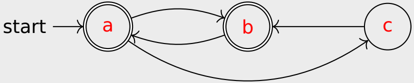
$$a \sim d \ \wedge \ b + c \sim d$$
$$\implies a + b \sim d + b \sim b + c + b$$

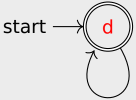


Notation

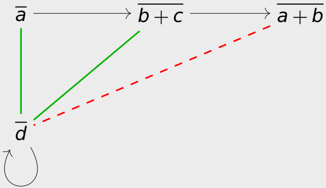
sets are denoted as sums of their elements; overlining indicates final state sets

Example





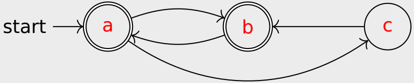
$$a \sim d \ \wedge \ b + c \sim d$$
$$\implies a + b \sim d + b \sim b + c + b = b + c$$

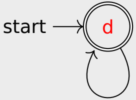


Notation

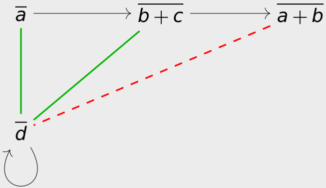
sets are denoted as sums of their elements; overlining indicates final state sets

Example





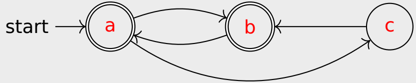
$$a \sim d \;\wedge\; b + c \sim d$$
$$\implies a + b \sim d + b \sim b + c + b = b + c \sim d$$

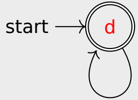


Notation

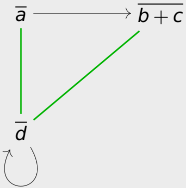
sets are denoted as sums of their elements; overlining indicates final state sets

Example





$$a \sim d \ \wedge \ b + c \sim d$$
$$\implies \ a + b \sim d + b \sim b + c + b = b + c \sim d$$



bisimulation up to congruence

Definition

given binary relation R on sets of states of NFA $N = (Q, \Sigma, \Delta, S, F)$

- c(R) is the smallest equivalence relation that includes R and is closed under union:

$$\frac{X_1 \text{ c(R) } Y_1 \qquad X_2 \text{ c(R) } Y_2}{X_1 \cup X_2 \text{ c(R) } Y_1 \cup Y_2}$$

Definition

given binary relation R on sets of states of NFA $N = (Q, \Sigma, \Delta, S, F)$

- c(R) is the smallest equivalence relation that includes R and is closed under union:

$$\frac{X_1 \text{ c(R) } Y_1 \qquad X_2 \text{ c(R) } Y_2}{X_1 \cup X_2 \text{ c(R) } Y_1 \cup Y_2}$$

- binary relation R on sets of states of NFA $N = (Q, \Sigma, \Delta, S, F)$ is bisimulation up to congruence if for all $Q_1, Q_2 \subseteq Q$ with $Q_1 \, R \, Q_2$:

Definition

given binary relation R on sets of states of NFA $N = (Q, \Sigma, \Delta, S, F)$

- c(R) is the smallest equivalence relation that includes R and is closed under union:

$$\frac{X_1 \text{ c(R) } Y_1 \qquad X_2 \text{ c(R) } Y_2}{X_1 \cup X_2 \text{ c(R) } Y_1 \cup Y_2}$$

- binary relation R on sets of states of NFA $N = (Q, \Sigma, \Delta, S, F)$ is bisimulation up to congruence if for all $Q_1, Q_2 \subseteq Q$ with $Q_1 \, R \, Q_2$:

  1. $Q_1 \cap F \neq \varnothing \quad \iff \quad Q_2 \cap F \neq \varnothing$

Definition

given binary relation R on sets of states of NFA $N = (Q, \Sigma, \Delta, S, F)$

- c(R) is the smallest equivalence relation that includes R and is closed under union:

$$\frac{X_1 \text{ c(R) } Y_1 \qquad X_2 \text{ c(R) } Y_2}{X_1 \cup X_2 \text{ c(R) } Y_1 \cup Y_2}$$

- binary relation R on sets of states of NFA $N = (Q, \Sigma, \Delta, S, F)$ is bisimulation up to congruence if for all $Q_1, Q_2 \subseteq Q$ with $Q_1 \, R \, Q_2$:
  1. $Q_1 \cap F \neq \varnothing \iff Q_2 \cap F \neq \varnothing$
  2. $\widehat{\Delta}(Q_1, a) \text{ c(R) } \widehat{\Delta}(Q_2, a)$ for all $a \in \Sigma$

Definition

given binary relation R on sets of states of NFA $N = (Q, \Sigma, \Delta, S, F)$

- c(R) is the smallest equivalence relation that includes R and is closed under union:

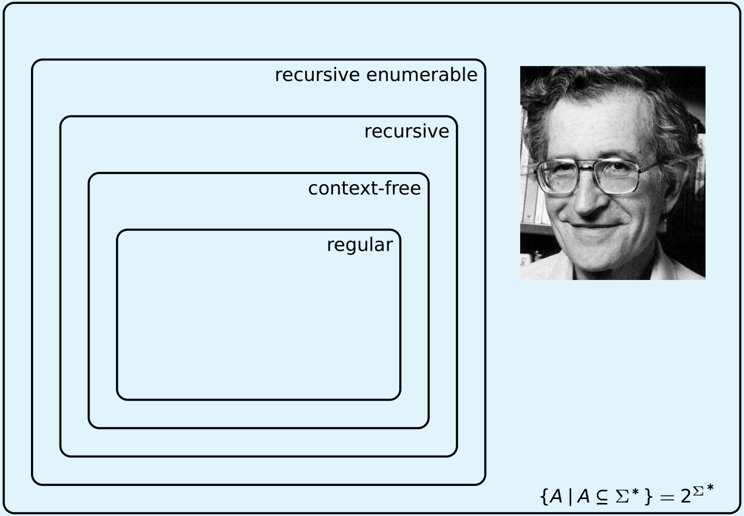
$$\frac{X_1 \ \mathrm{c(R)}\ Y_1 \qquad X_2 \ \mathrm{c(R)}\ Y_2}{X_1 \cup X_2 \ \mathrm{c(R)}\ Y_1 \cup Y_2}$$

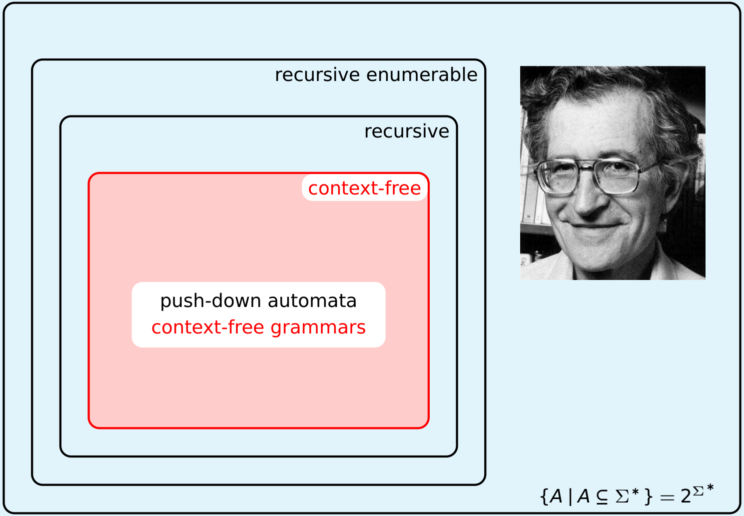
- binary relation R on sets of states of NFA $N = (Q, \Sigma, \Delta, S, F)$ is bisimulation up to congruence if for all $Q_1, Q_2 \subseteq Q$ with $Q_1 \, R \, Q_2$:
  1. $Q_1 \cap F \neq \varnothing \iff Q_2 \cap F \neq \varnothing$
  2. $\widehat{\Delta}(Q_1, a) \ \mathrm{c(R)}\ \widehat{\Delta}(Q_2, a)$ for all $a \in \Sigma$

Theorem

NFAs $N_1 = (Q_1, \Sigma, \Delta_1, S_1, F_1)$ and $N_2 = (Q_2, \Sigma, \Delta_2, S_2, F_2)$ are equivalent $\iff$ S_1 R S_2 for some bisimulation up to congruence R on $N_1 \cup N_2$

Outline


recursive enumerable
recursive
context-free
regular
$\{A \mid A \subseteq \Sigma^*\} = 2^{\Sigma^*}$


recursive enumerable
recursive
context-free
push-down automata
context-free grammars
$\{A \mid A \subseteq \Sigma^*\} = 2^{\Sigma^*}$

Definitions

- context-free grammar (CFG) is quadruple $G = (N, \Sigma, P, S)$ with

Definitions

- context-free grammar (CFG) is quadruple $G = (N, \Sigma, P, S)$ with
  1. N : finite set of nonterminals

Definitions

- context-free grammar (CFG) is quadruple $G = (N, \Sigma, P, S)$ with
  1. N : finite set of nonterminals
  2. Σ : finite set of terminals, disjoint from N

Definitions

- context-free grammar (CFG) is quadruple $G = (N, \Sigma, P, S)$ with
  1. N : finite set of nonterminals
  2. Σ : finite set of terminals, disjoint from N
  3. P : finite set of productions of the form $A \rightarrow \alpha$ with $A \in N$ and $\alpha \in (N \cup \Sigma)^*$

Definitions

- context-free grammar (CFG) is quadruple $G = (N, \Sigma, P, S)$ with
  1. N : finite set of nonterminals
  2. Σ : finite set of terminals, disjoint from N
  3. P : finite set of productions of the form $A \to \alpha$ with $A \in N$ and $\alpha \in (N \cup \Sigma)^*$
  4. $S \in N$: start symbol

Definitions

- context-free grammar (CFG) is quadruple $G = (N, \Sigma, P, S)$ with
  1. N : finite set of nonterminals
  2. Σ : finite set of terminals, disjoint from N
  3. P : finite set of productions of the form $A \to \alpha$ with $A \in N$ and $\alpha \in (N \cup \Sigma)^*$
  4. $S \in N$: start symbol
- one step derivation relation $\xrightarrow[G]{1}$ on $(N \cup \Sigma)^*$: $\beta A \gamma \xrightarrow[G]{1} \beta \alpha \gamma$ if $A \to \alpha \in P$ and $\beta, \gamma \in (N \cup \Sigma)^*$

Example

CFG $G = (N, \Sigma, P, S)$

1. $N = \{S\}$
2. $\Sigma = \{\mathrm{a}, \mathrm{b}\}$
3. $P = \{S \rightarrow aSb,\ S \rightarrow \varepsilon\}$

Example

CFG $G = (N, \Sigma, P, S)$

1. $N = \{S\}$
2. $\Sigma = \{a, b\}$
3. $P = \{S \to aSb, S \to \varepsilon\}$

two derivations:

$$S \xrightarrow[G]{1} aSb$$

Example

CFG $G = (N, \Sigma, P, S)$

1. $N = \{S\}$
2. $\Sigma = \{a, b\}$
3. $P = \{S \rightarrow aSb,\ S \rightarrow \varepsilon\}$

two derivations:

$$S \xrightarrow[G]{1} aSb \xrightarrow[G]{1} ab$$

Example

CFG $G = (N, \Sigma, P, S)$

1. $N = \{S\}$
2. $\Sigma = \{\mathrm{a}, \mathrm{b}\}$
3. $P = \{S \rightarrow aSb, S \rightarrow \varepsilon\}$

two derivations:

$$S \xrightarrow[G]{1} aSb \xrightarrow[G]{1} ab \qquad S \xrightarrow[G]{1} \mathrm{a}S\mathrm{b} \xrightarrow[G]{1} aaSbb$$

Example

CFG $G = (N, \Sigma, P, S)$

1. $N = \{S\}$
2. $\Sigma = \{\mathrm{a}, \mathrm{b}\}$
3. $P = \{S \rightarrow aSb, S \rightarrow \varepsilon\}$

two derivations:

$$S \xrightarrow[G]{1} aSb \xrightarrow[G]{1} ab \qquad S \xrightarrow[G]{1} \mathrm{aSb} \xrightarrow[G]{1} aaSbb \xrightarrow[G]{1} aabb$$

Example

CFG $G = (N, \Sigma, P, S)$

1. $N = \{S\}$
2. $\Sigma = \{a, b\}$
3. $P = \{S \to aSb, S \to \varepsilon\} = \{S \to aSb \mid \varepsilon\}$

two derivations:

$$S \xrightarrow[G]{1} aSb \xrightarrow[G]{1} ab \qquad S \xrightarrow[G]{1} aSb \xrightarrow[G]{1} aaSbb \xrightarrow[G]{1} aabb$$

Definitions

- context-free grammar (CFG) is quadruple $G = (N, \Sigma, P, S)$ with
  1. N : finite set of nonterminals
  2. Σ : finite set of terminals, disjoint from N
  3. P : finite set of productions of the form $A \to \alpha$ with $A \in N$ and $\alpha \in (N \cup \Sigma)^*$
  4. $S \in N$: start symbol
- one step derivation relation $\xrightarrow[G]{1}$ on $(N \cup \Sigma)^*$: $\quad \beta A \gamma \xrightarrow[G]{1} \beta\alpha\gamma \quad$ if $A \to \alpha \in P$ and $\beta, \gamma \in (N \cup \Sigma)^*$
- $\xrightarrow[G]{n} = (\xrightarrow[G]{1})^n \quad \forall n \geqslant 0$

Definitions

- context-free grammar (CFG) is quadruple $G = (N, \Sigma, P, S)$ with
  1. N : finite set of nonterminals
  2. Σ : finite set of terminals, disjoint from N
  3. P : finite set of productions of the form $A \to \alpha$ with $A \in N$ and $\alpha \in (N \cup \Sigma)^*$
  4. $S \in N$: start symbol
- one step derivation relation $\xrightarrow[G]{1}$ on $(N \cup \Sigma)^*$: $\quad \beta A \gamma \xrightarrow[G]{1} \beta \alpha \gamma \quad$ if $A \to \alpha \in P$ and $\beta, \gamma \in (N \cup \Sigma)^*$
- $\xrightarrow[G]{n} = (\xrightarrow[G]{1})^n \quad \forall\, n \geqslant 0 \qquad\qquad \xrightarrow[G]{*} = \bigcup_{n \geqslant 0} \xrightarrow[G]{n}$

Definitions

- context-free grammar (CFG) is quadruple $G = (N, \Sigma, P, S)$ with
  1. N : finite set of nonterminals
  2. Σ : finite set of terminals, disjoint from N
  3. P : finite set of productions of the form $A \to \alpha$ with $A \in N$ and $\alpha \in (N \cup \Sigma)^*$
  4. $S \in N$: start symbol
- one step derivation relation $\xrightarrow[G]{1}$ on $(N \cup \Sigma)^*$: $\beta A \gamma \xrightarrow[G]{1} \beta \alpha \gamma$ if $A \to \alpha \in P$ and $\beta, \gamma \in (N \cup \Sigma)^*$
- $\xrightarrow[G]{n} = (\xrightarrow[G]{1})^n \quad \forall\, n \geqslant 0 \qquad\qquad \xrightarrow[G]{*} = \bigcup_{n \geqslant 0} \xrightarrow[G]{n}$
- members of the set $(N \cup \Sigma)^*$ are called strings

Definitions

- context-free grammar (CFG) is quadruple $G = (N, \Sigma, P, S)$ with
  1. N : finite set of nonterminals
  2. Σ : finite set of terminals, disjoint from N
  3. P : finite set of productions of the form $A \to \alpha$ with $A \in N$ and $\alpha \in (N \cup \Sigma)^*$
  4. $S \in N$: start symbol
- one step derivation relation $\xrightarrow[G]{1}$ on $(N \cup \Sigma)^*$: $\beta A \gamma \xrightarrow[G]{1} \beta \alpha \gamma$ if $A \to \alpha \in P$ and $\beta, \gamma \in (N \cup \Sigma)^*$
- $\xrightarrow[G]{n} = (\xrightarrow[G]{1})^n \quad \forall n \geqslant 0$ $\qquad \xrightarrow[G]{*} = \bigcup_{n \geqslant 0} \xrightarrow[G]{n}$
- members of the set $(N \cup \Sigma)^*$ are called strings
- string s is called a sentential form if $S \xrightarrow[G]{*} s$ (derivable from the start symbol S)

Definitions

- context-free grammar (CFG) is quadruple $G = (N, \Sigma, P, S)$ with
  1. N : finite set of nonterminals
  2. Σ : finite set of terminals, disjoint from N
  3. P : finite set of productions of the form $A \to \alpha$ with $A \in N$ and $\alpha \in (N \cup \Sigma)^*$
  4. $S \in N$: start symbol
- one step derivation relation $\xrightarrow[G]{1}$ on $(N \cup \Sigma)^*$: $\quad \beta A \gamma \xrightarrow[G]{1} \beta\alpha\gamma \quad$ if $A \to \alpha \in P$ and $\beta, \gamma \in (N \cup \Sigma)^*$
- $\xrightarrow[G]{n} = (\xrightarrow[G]{1})^n \quad \forall n \geqslant 0 \qquad\qquad \xrightarrow[G]{*} = \bigcup_{n \geqslant 0} \xrightarrow[G]{n}$
- members of the set $(N \cup \Sigma)^*$ are called strings
- string s is called a sentential form if $S \xrightarrow[G]{*} s$ (derivable from the start symbol S)
- sentential form x is called a sentence if $x \in \Sigma^*$ (consisting terminal symbols only)

Definitions

- context-free grammar (CFG) is quadruple $G = (N, \Sigma, P, S)$ with
  1. N : finite set of nonterminals
  2. Σ : finite set of terminals, disjoint from N
  3. P : finite set of productions of the form $A \to \alpha$ with $A \in N$ and $\alpha \in (N \cup \Sigma)^*$
  4. $S \in N$: start symbol
- one step derivation relation $\xrightarrow[G]{1}$ on $(N \cup \Sigma)^*$: $\quad \beta A \gamma \xrightarrow[G]{1} \beta \alpha \gamma \quad$ if $A \to \alpha \in P$ and $\beta, \gamma \in (N \cup \Sigma)^*$
- $\xrightarrow[G]{n} = (\xrightarrow[G]{1})^n \quad \forall n \geqslant 0 \qquad\qquad \xrightarrow[G]{*} = \bigcup_{n \geqslant 0} \xrightarrow[G]{n}$
- members of the set $(N \cup \Sigma)^*$ are called strings
- string s is called a sentential form if $S \xrightarrow[G]{*} s$ (derivable from the start symbol S)
- sentential form x is called a sentence if $x \in \Sigma^*$ (consisting terminal symbols only)
- language generated by G: $\quad L(G) = \{x \in \Sigma^* \mid S \xrightarrow[G]{*} x\}$

Example

CFG $G = (N, \Sigma, P, S)$

1. $N = \{S\}$
2. $\Sigma = \{\mathrm{a}, \mathrm{b}\}$
3. $P = \{S \to aSb,\ S \to \varepsilon\} = \{S \to \mathrm{a}S\mathrm{b} \mid \varepsilon\}$

two derivations:

$$S \xrightarrow[G]{1} aSb \xrightarrow[G]{1} ab \qquad S \xrightarrow[G]{1} \mathrm{a}S\mathrm{b} \xrightarrow[G]{1} aaSbb \xrightarrow[G]{1} aabb$$

Lemma

$L(G) = \{\mathrm{a}^n\mathrm{b}^n \mid n \geqslant 0\}$

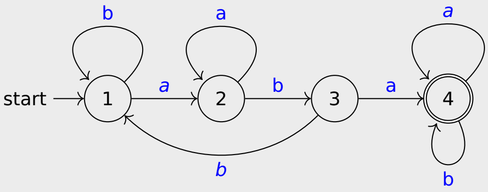
Definitions

- context-free grammar (CFG) is quadruple $G = (N, \Sigma, P, S)$ with
  1. N : finite set of nonterminals
  2. Σ : finite set of terminals, disjoint from N
  3. P : finite set of productions of the form $A \to \alpha$ with $A \in N$ and $\alpha \in (N \cup \Sigma)^*$
  4. $S \in N$: start symbol
- one step derivation relation $\xrightarrow[G]{1}$ on $(N \cup \Sigma)^*$: $\quad \beta A \gamma \xrightarrow[G]{1} \beta \alpha \gamma \quad$ if $A \to \alpha \in P$ and $\beta, \gamma \in (N \cup \Sigma)^*$
- $\xrightarrow[G]{n} = (\xrightarrow[G]{1})^n \quad \forall n \geqslant 0 \qquad\qquad \xrightarrow[G]{*} = \bigcup_{n \geqslant 0} \xrightarrow[G]{n}$
- members of the set $(N \cup \Sigma)^*$ are called strings
- string s is called a sentential form if $S \xrightarrow[G]{*} s$ (derivable from the start symbol S)
- sentential form x is called a sentence if $x \in \Sigma^*$ (consisting terminal symbols only)
- language generated by G: $\quad L(G) = \{x \in \Sigma^* \mid S \xrightarrow[G]{*} x\}$
- set $B \subseteq \Sigma^*$ is context-free if $B = L(G)$ for some CFG G

Outline

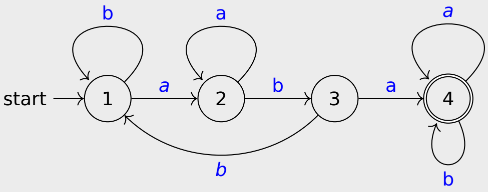
Example

DFA M



Example

DFA M



CFG G_M

$1 \to a\,2 \mid b\,1$

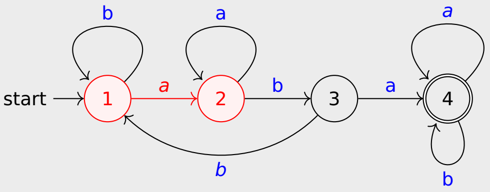
$2 \to a\,2 \mid b\,3$

$3 \to a\,4 \mid b\,1$

$4 \to a\,4 \mid b\,4 \mid \varepsilon$

Example

DFA M



CFG G_M

$1 \to a\,2 \mid b\,1$

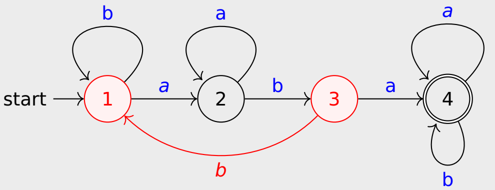
$2 \to a\,2 \mid b\,3$

$3 \to a\,4 \mid b\,1$

$4 \to a\,4 \mid b\,4 \mid \varepsilon$

Example

DFA M



CFG G_M

$1 \rightarrow a\,2 \mid b\,1$

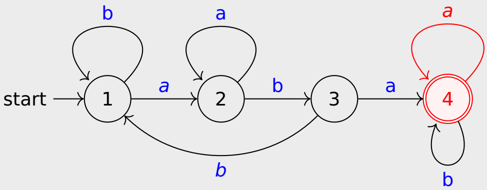
$2 \rightarrow a\,2 \mid b\,3$

$3 \rightarrow a\,4 \mid b\,1$

$4 \rightarrow a\,4 \mid b\,4 \mid \varepsilon$

Example

DFA M



CFG G_M

$1 \to a\,2 \mid b\,1$

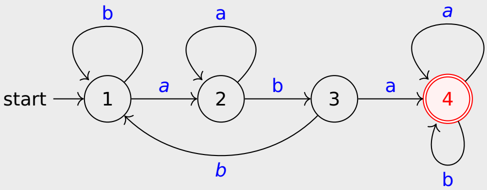
$2 \to a\,2 \mid b\,3$

$3 \to a\,4 \mid b\,1$

$4 \to a\,4 \mid b\,4 \mid \varepsilon$

Example

DFA M

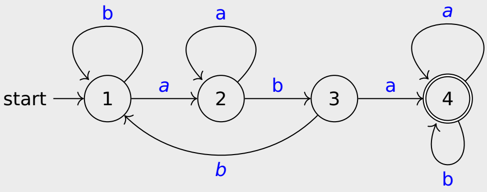


CFG G_M

$$1 \rightarrow a\,2 \mid b\,1$$
$$2 \rightarrow a\,2 \mid b\,3$$
$$3 \rightarrow a\,4 \mid b\,1$$
$$4 \rightarrow a\,4 \mid b\,4 \mid \varepsilon$$

Example

DFA M



CFG G_M

$1 \to a\,2 \mid b\,1$

$2 \to a\,2 \mid b\,3$

$3 \to a\,4 \mid b\,1$

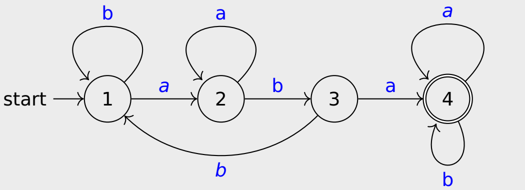
$4 \to a\,4 \mid b\,4 \mid \varepsilon$

DFA 1

CFG 1

Example

DFA M



CFG G_M

$1 \rightarrow a\,2 \mid b\,1$

$2 \rightarrow a\,2 \mid b\,3$

$3 \rightarrow a\,4 \mid b\,1$

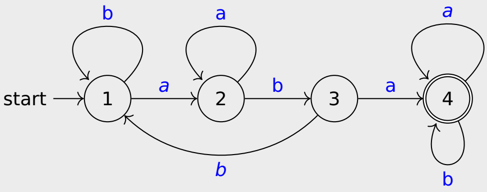
$4 \rightarrow a\,4 \mid b\,4 \mid \varepsilon$

DFA $1 \xrightarrow{b} 1$

CFG $1 \rightarrow b\,1$

Example

DFA M



CFG G_M

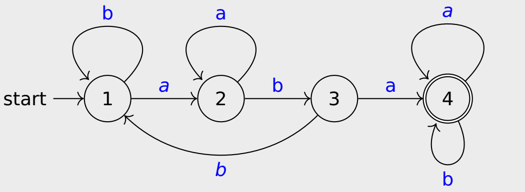
$$1 \to a\,2 \mid b\,1 \qquad 3 \to a\,4 \mid b\,1$$
$$2 \to a\,2 \mid b\,3 \qquad 4 \to a\,4 \mid b\,4 \mid \varepsilon$$

DFA $\quad 1 \overset{b}{\ } 1 \overset{a}{\ } 2$

CFG $\quad 1 \to b\,1 \to b\,a\,2$

Example

DFA M



CFG G_M

$1 \rightarrow a\,2 \mid b\,1$

$2 \rightarrow a\,2 \mid b\,3$

$3 \rightarrow a\,4 \mid b\,1$

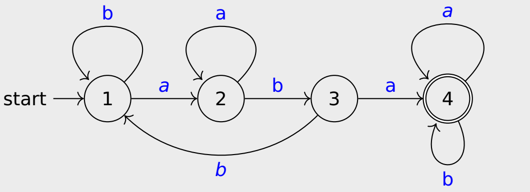
$4 \rightarrow a\,4 \mid b\,4 \mid \varepsilon$

DFA $1 \xrightarrow{b} 1 \xrightarrow{a} 2 \xrightarrow{b} 3$

CFG $1 \rightarrow b\,1 \rightarrow b\,a\,2 \rightarrow b\,a\,b\,3$

Example

DFA M



CFG G_M

$1 \to a\,2 \mid b\,1$

$2 \to a\,2 \mid b\,3$

$3 \to a\,4 \mid b\,1$

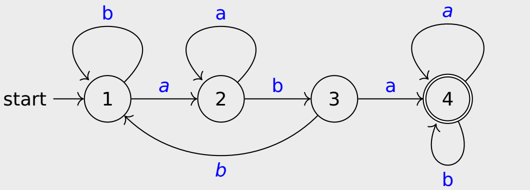
$4 \to a\,4 \mid b\,4 \mid \varepsilon$

DFA $1 \xrightarrow{b} 1 \xrightarrow{a} 2 \xrightarrow{b} 3 \xrightarrow{a} 4$

CFG $1 \to b\,1 \to b\,a\,2 \to b\,a\,b\,3 \to b\,a\,b\,a\,4$

Example

DFA M



CFG G_M

$1 \to a\,2 \mid b\,1$

$2 \to a\,2 \mid b\,3$

$3 \to a\,4 \mid b\,1$

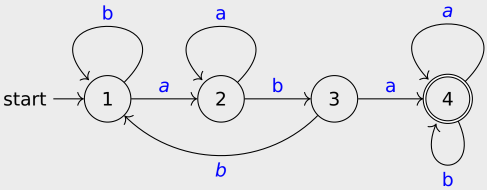
$4 \to a\,4 \mid b\,4 \mid \varepsilon$

DFA $1 \xrightarrow{b} 1 \xrightarrow{a} 2 \xrightarrow{b} 3 \xrightarrow{a} 4 \xrightarrow{a} 4$

CFG $1 \to b\,1 \to b\,a\,2 \to b\,a\,b\,3 \to b\,a\,b\,a\,4 \to b\,a\,b\,a\,a\,4$

Example

DFA M



CFG G_M

$1 \to a\,2 \mid b\,1$

$2 \to a\,2 \mid b\,3$

$3 \to a\,4 \mid b\,1$

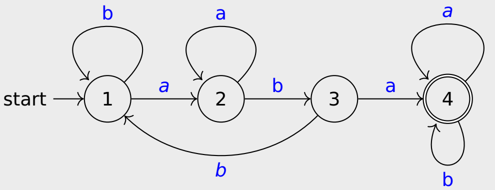
$4 \to a\,4 \mid b\,4 \mid \varepsilon$

DFA $1 \overset{b}{\;} 1 \overset{a}{\;} 2 \overset{b}{\;} 3 \overset{a}{\;} 4 \overset{a}{\;} 4$

CFG $1 \to b\,1 \to b\,a\,2 \to b\,a\,b\,3 \to b\,a\,b\,a\,4 \to b\,a\,b\,a\,a\,4 \to b\,a\,b\,a\,a$

Example

DFA M



CFG G_M

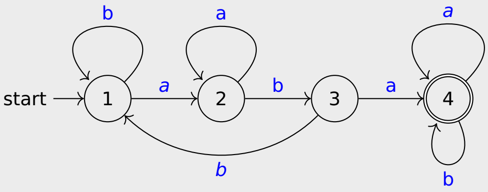
$$1 \to a\,2 \mid b\,1 \qquad 3 \to a\,4 \mid b\,1$$
$$2 \to a\,2 \mid b\,3 \qquad 4 \to a\,4 \mid b\,4 \mid \varepsilon$$

DFA $1 \xrightarrow{b} 1 \xrightarrow{a} 2 \xrightarrow{b} 3 \xrightarrow{a} 4 \xrightarrow{a} 4 \in L(M)$

CFG $1 \to b\,1 \to b\,a\,2 \to b\,a\,b\,3 \to b\,a\,b\,a\,4 \to b\,a\,b\,a\,a\,4 \to b\,a\,b\,a\,a \in L(G_M)$

Example

DFA M



strongly right-linear CFG G_M

$$1 \to a\,2 \mid b\,1 \qquad 3 \to a\,4 \mid b\,1$$

$$2 \to a\,2 \mid b\,3 \qquad 4 \to a\,4 \mid b\,4 \mid \varepsilon$$

DFA $\quad 1 \xrightarrow{b} 1 \xrightarrow{a} 2 \xrightarrow{b} 3 \xrightarrow{a} 4 \xrightarrow{a} 4 \in L(M)$

CFG $\quad 1 \to b\,1 \to b\,a\,2 \to b\,a\,b\,3 \to b\,a\,b\,a\,4 \to b\,a\,b\,a\,a\,4 \to b\,a\,b\,a\,a \in L(G_M)$

Definition

CFG $G = (N, \Sigma, P, S)$ is strongly right-linear if

$$\alpha = aB \in \Sigma N \quad \text{or} \quad \alpha = \varepsilon$$

for all $A \to \alpha$ in P

Definition

CFG $G = (N, \Sigma, P, S)$ is strongly right-linear if

$$\alpha = aB \in \Sigma N \quad \text{or} \quad \alpha = \varepsilon$$

for all $A \to \alpha$ in P

Lemma

L is regular $\iff$ L is generated by strongly right-linear CFG

Definition

CFG $G = (N, \Sigma, P, S)$ is strongly right-linear if

$$\alpha = aB \in \Sigma N \quad \text{or} \quad \alpha = \varepsilon$$

for all $A \to \alpha$ in P

Lemma

L is regular $\iff$ L is generated by strongly right-linear CFG

Proof. ($\implies$)

- DFA $M = (Q, \Sigma, \delta, s, F)$
- $L(M) = L(G_M)$ for strongly right-linear CFG $G_M = \{Q, \Sigma, P, s\}$ with

$$P = \{p \to aq \mid \delta(p, a) = q\} \cup \{q \to \varepsilon \mid q \in F\}$$

Definition

CFG $G = (N, \Sigma, P, S)$ is strongly right-linear if

$$\alpha = aB \in \Sigma N \quad \text{or} \quad \alpha = \varepsilon$$

for all $A \to \alpha$ in P

Lemma

L is regular $\iff$ L is generated by strongly right-linear CFG

Proof. ($\Longleftarrow$)

- strongly right-linear CFG $G = (N, \Sigma, P, S)$
- $L(G) = L(M_G)$ for NFA $M_G = (N, \Sigma, \Delta, \{S\}, F)$ with

$$\Delta(A, a) = \{B \mid A \to aB \in P\} \quad \text{and} \quad F = \{A \mid A \to \varepsilon \in P\}$$

Definition

CFG $G = (N, \Sigma, P, S)$ is strongly right-linear if

$$\alpha = aB \in \Sigma N \quad \text{or} \quad \alpha = \varepsilon$$

for all $A \to \alpha$ in P

Lemma

L is regular $\iff$ L is generated by strongly right-linear CFG

Corollary

every regular set is context-free

Outline

Example

CFG G: $S \to [S] \mid SS \mid \varepsilon$

Example

CFG G: $S \to [S] \mid SS \mid \varepsilon$ three derivations of [[]]:

1. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$

Example

CFG G: $S \rightarrow [S] \mid SS \mid \varepsilon$ three derivations of $[[]]$:

1. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$
2. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [SS] \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$

Example

CFG G: $S \to [S] \mid SS \mid \varepsilon$ three derivations of $[[]]$:

1. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$
2. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [SS] \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$
3. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [SS] \xrightarrow[G]{1} [S[S]] \xrightarrow[G]{1} [S[]] \xrightarrow[G]{1} [[]]$

Example

CFG G: $S \to [S] \mid SS \mid \epsilon$ three derivations of [[]]:

➊ $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$

➋ $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [SS] \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$

➌ $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [SS] \xrightarrow[G]{1} [S[S]] \xrightarrow[G]{1} [S[]] \xrightarrow[G]{1} [[]]$

Definition

- in leftmost derivation always leftmost nonterminal is replaced ➊ ➋

Example

CFG G: $S \to [S] \mid SS \mid \varepsilon$ three derivations of $[[]]$:

1. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$
2. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [SS] \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$
3. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [SS] \xrightarrow[G]{1} [S[S]] \xrightarrow[G]{1} [S[]] \xrightarrow[G]{1} [[]]$

Definition

- in leftmost derivation always leftmost nonterminal is replaced 1 2
- in rightmost derivation always rightmost nonterminal is replaced 1 3

Example

CFG G: $S \rightarrow [S] \mid SS \mid \varepsilon$ three derivations of [[]]:

➊ $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$

➋ $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [SS] \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$

➌ $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [SS] \xrightarrow[G]{1} [S[S]] \xrightarrow[G]{1} [S[]] \xrightarrow[G]{1} [[]]$

Definition

- in leftmost derivation always leftmost nonterminal is replaced ➊ ➋
- in rightmost derivation always rightmost nonterminal is replaced ➊ ➌
- parse tree is representation of derivation in which replacement order is ignored

Example

CFG G: $S \to [S] \mid SS \mid \epsilon$ three derivations of $[[]]$:

1. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$
2. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [SS] \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$
3. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [SS] \xrightarrow[G]{1} [S[S]] \xrightarrow[G]{1} [S[]] \xrightarrow[G]{1} [[]]$

Definition

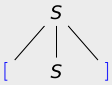
- in leftmost derivation always leftmost nonterminal is replaced 1 2
- in rightmost derivation always rightmost nonterminal is replaced 1 3
- parse tree is representation of derivation in which replacement order is ignored
- CFG is ambiguous if some string has different parse trees

Example

CFG G: $S \to [\,S\,] \mid S\,S \mid \varepsilon$

1. $S \xrightarrow[G]{1} [\,S\,]$

parse tree

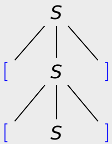


Example

CFG G: $S \rightarrow [S] \mid S\,S \mid \varepsilon$

1. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]]$

parse tree

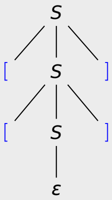


Example

CFG G: $S \to [S] \mid S\,S \mid \varepsilon$

1. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$

parse tree



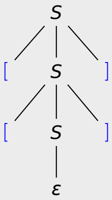
Example

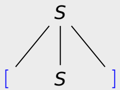
CFG G: $S \to [S] \mid S\,S \mid \varepsilon$

➊ $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$

➋ $S \xrightarrow[G]{1} [S]$

parse trees



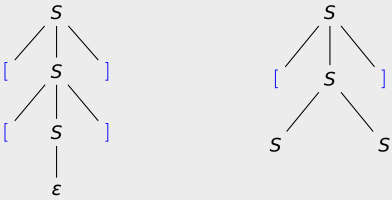


Example

CFG G: $S \to [S] \mid S\,S \mid \varepsilon$

1. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$
2. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [S\,S]$

parse trees



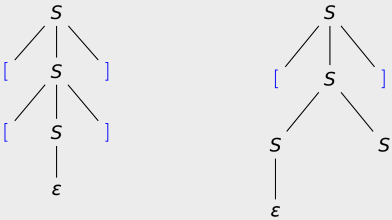
Example

CFG G: $S \rightarrow [S] \mid SS \mid \varepsilon$

1. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$

2. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [SS] \xrightarrow[G]{1} [S]$

parse trees

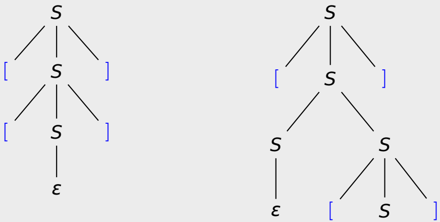


Example

CFG G: $S \rightarrow [S] \mid S\,S \mid \varepsilon$

1. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$
2. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [S\,S] \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]]$

parse trees



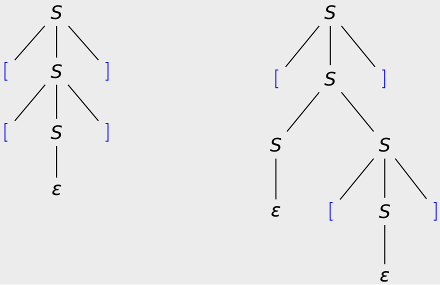
Example

CFG G: $S \to [\,S\,] \mid S\,S \mid \varepsilon$

➊ $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$

➋ $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [SS] \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$

parse trees

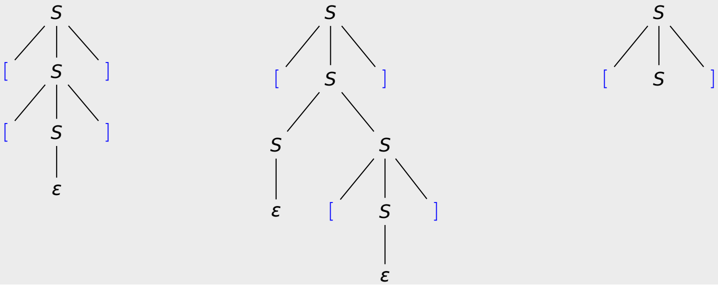


Example

CFG G: $S \rightarrow [S] \mid S\,S \mid \varepsilon$

1. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$
2. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [S\,S] \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$
3. $S \xrightarrow[G]{1} [S]$

parse trees

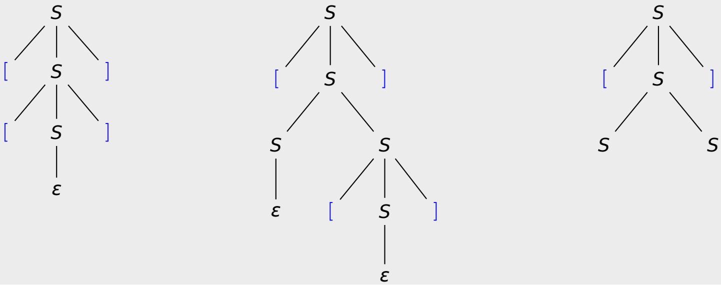


Example

CFG G: $S \to [S] \mid S\,S \mid \varepsilon$

1. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[\,]]$
2. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [S\,S] \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[\,]]$
3. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [S\,S]$

parse trees

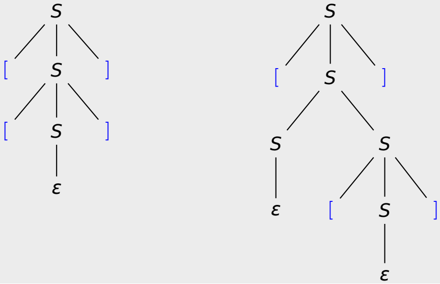


Example

CFG G: $S \to [S] \mid S\,S \mid \varepsilon$

1. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[\,]]$
2. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [S\,S] \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[\,]]$
3. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [S\,S] \xrightarrow[G]{1} [S\,[S]]$

parse trees

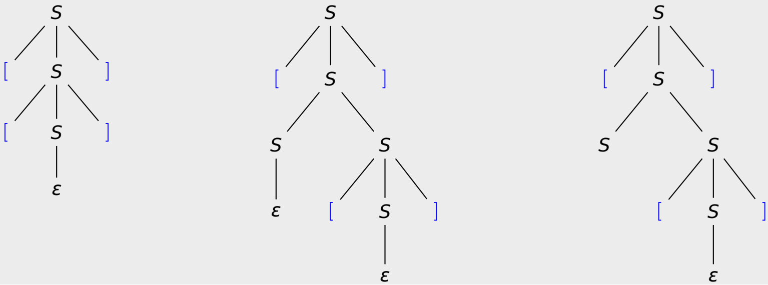


Example

CFG G: $S \to [S] \mid SS \mid \varepsilon$

1. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$
2. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [SS] \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$
3. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [SS] \xrightarrow[G]{1} [S[S]] \xrightarrow[G]{1} [S[]]$

parse trees

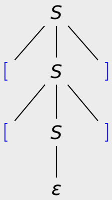


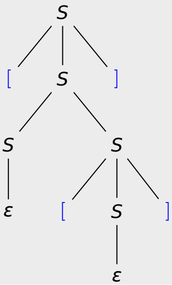
Example

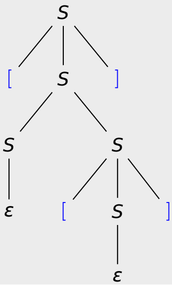
CFG G: $S \to [S] \mid S\,S \mid \varepsilon$

1. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$
2. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [S\,S] \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [[S]] \xrightarrow[G]{1} [[]]$
3. $S \xrightarrow[G]{1} [S] \xrightarrow[G]{1} [S\,S] \xrightarrow[G]{1} [S[S]] \xrightarrow[G]{1} [S[]] \xrightarrow[G]{1} [[]]$

parse trees







Example

- CFG G : $S \to [S] \mid S\,S \mid \varepsilon$
- G is ambiguous

Example

- CFG G : $S \to [S] \mid S\,S \mid \varepsilon$
- CFG G': $S \to \varepsilon \mid T$
  $T \to T\,U \mid U$
  $U \to [\,] \mid [T]$
- G is ambiguous

Example

- CFG G : $S \rightarrow [S] \mid S\,S \mid \varepsilon$
- CFG G': $S \rightarrow \varepsilon \mid T$
  $T \rightarrow T\,U \mid U$
  $U \rightarrow [\,] \mid [T]$
- G is ambiguous G' is unambiguous $L(G) = L(G')$

Example

- CFG G : $S \to [S] \mid SS \mid \varepsilon$
- CFG G': $S \to \varepsilon \mid T$
  $T \to TU \mid U$
  $U \to [\,] \mid [T]$
- G is ambiguous $\quad$ G' is unambiguous $\quad$ $L(G) = L(G')$

$$
\begin{array}{c}
S \\
| \\
T \\
| \\
U \\
| \\
[\,T\,] \\
| \\
[\,U\,] \\
| \\
[\,[\,]\,]
\end{array}
$$

Example

- CFG G : $S \to S - S \mid int$
- G is ambiguous

  with G $7-5-2$ could be parsed as $(7-5)-2$ and $7-(5-2)$

Example

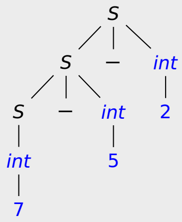
- CFG G : $S \rightarrow S - S \mid int$
- CFG G': $S \rightarrow S - int \mid int$
- G is ambiguous

  with G $7-5-2$ could be parsed as $(7-5)-2$ and $7-(5-2)$

Example

- CFG G : $S \rightarrow S - S \mid int$
- CFG G': $S \rightarrow S - int \mid int$
- G is ambiguous G' is unambiguous $L(G) = L(G')$

with G $7-5-2$ could be parsed as $(7-5)-2$ and $7-(5-2)$
with G' $7-5-2$ could only be parsed as $(7-5)-2$

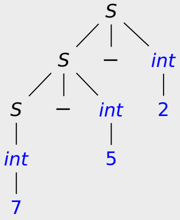


Example

- CFG G : $S \rightarrow S - S \mid int$
- CFG G': $S \rightarrow S - int \mid int$
- G is ambiguous G' is unambiguous $L(G) = L(G')$

  with G $7-5-2$ could be parsed as $(7-5)-2$ and $7-(5-2)$
  with G' $7-5-2$ could only be parsed as $(7-5)-2$

- (if applicable) one way to remove ambiguity is to benefit from associativity of binary operators

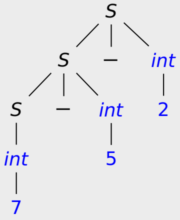


Example

- CFG G : $\quad S \rightarrow S - S \mid int$
- CFG G': $\quad S \rightarrow S - int \mid int$
- G is ambiguous $\qquad G'$ is unambiguous $\qquad L(G) = L(G')$

  with G $\quad 7-5-2$ could be parsed as $\quad (7-5)-2$ and $7-(5-2)$
  with G' $\quad 7-5-2$ could only be parsed as $\quad (7-5)-2$

- (if applicable) one way to remove ambiguity is to benefit from associativity of binary operators
- G' enforces the "-" operator to be left associative



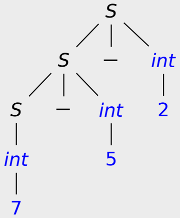
Example

- CFG G : $S \rightarrow S - S \mid int$
- CFG G': $S \rightarrow S - int \mid int$
- G is ambiguous G' is unambiguous $L(G) = L(G')$

  with G $7-5-2$ could be parsed as $(7-5)-2$ and $7-(5-2)$
  with G' $7-5-2$ could only be parsed as $(7-5)-2$

- (if applicable) one way to remove ambiguity is to benefit from associativity of binary operators
- G' enforces the "-" operator to be left associative
- CFG G'': $S \rightarrow int - S \mid int$

  turns the "-" operator into a right associative one



Example

- CFG G : $S \rightarrow S \times S \mid S + S \mid int$
- G is ambiguous

  with G $7 + 5 \times 2$ could be parsed as $7 + (5 \times 2)$ and $(7 + 5) \times 2$

Example

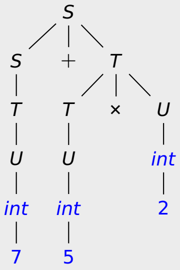
- CFG G : $S \rightarrow S \times S \mid S + S \mid int$
- CFG G': $S \rightarrow S + T \mid T$
  $T \rightarrow T \times U \mid U$
  $U \rightarrow int \mid (S)$
- G is ambiguous

  with G $7 + 5 \times 2$ could be parsed as $7 + (5 \times 2)$ and $(7 + 5) \times 2$

Example

- CFG G : $S \to S \times S \mid S + S \mid int$
- CFG G': $S \to S + T \mid T$
  $T \to T \times U \mid U$
  $U \to int \mid (S)$
- G is ambiguous G' is unambiguous $L(G) = L(G')$

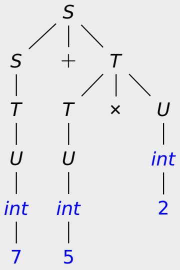
  with G $7 + 5 \times 2$ could be parsed as $7 + (5 \times 2)$ and $(7 + 5) \times 2$
  with G' $7 + 5 \times 2$ could only be parsed as $7 + (5 \times 2)$



Example

- CFG G : $S \to S \times S \mid S + S \mid int$
- CFG G': $S \to S + T \mid T$
  $T \to T \times U \mid U$
  $U \to int \mid (S)$
- G is ambiguous $\quad G'$ is unambiguous $\quad L(G) = L(G')$

  with G $\quad 7 + 5 \times 2$ could be parsed as $\quad 7 + (5 \times 2)$ and $(7 + 5) \times 2$
  with G' $\quad 7 + 5 \times 2$ could only be parsed as $\quad 7 + (5 \times 2)$
- (if applicable) one way to remove ambiguity is to benefit from precedence of operators



Remark

there exist context-free sets without unambiguous grammars

Remark

there exist context-free sets without unambiguous grammars (inherently ambiguous)

Remark

there exist context-free sets without unambiguous grammars (inherently ambiguous)

Example

$A = \{a^i b^j c^k \mid i = j \text{ or } j = k\}$ is context-free and inherently ambiguous

Remark

there exist context-free sets without unambiguous grammars (inherently ambiguous)

Example

$A = \{a^i b^j c^k \mid i = j \text{ or } j = k\}$ is context-free and inherently ambiguous

$A = \{a^i b^i c^k\} \cup \{a^i b^j c^j\}$

Remark

there exist context-free sets without unambiguous grammars (inherently ambiguous)

Example

$A = \{a^i b^j c^k \mid i = j \text{ or } j = k\}$ is context-free and inherently ambiguous

$A = \{a^i b^i c^k\} \cup \{a^i b^j c^j\}$

let $A = L(G)$ such that G:

Remark

there exist context-free sets without unambiguous grammars (inherently ambiguous)

Example

$A = \{a^i b^j c^k \mid i = j \text{ or } j = k\}$ is context-free and inherently ambiguous

$A = \{a^i b^i c^k\} \cup \{a^i b^j c^j\}$

let $A = L(G)$ such that G:

$$S \to T \mid W$$

$$\begin{aligned} T &\to UV & W &\to XY \\ U &\to aUb \mid \varepsilon & X &\to aX \mid \varepsilon \\ V &\to cV \mid \varepsilon & Y &\to bYc \mid \varepsilon \end{aligned}$$

Remark

there exist context-free sets without unambiguous grammars (inherently ambiguous)

Example

$A = \{a^i b^j c^k \mid i = j \text{ or } j = k\}$ is context-free and inherently ambiguous

$A = \{a^i b^i c^k\} \cup \{a^i b^j c^j\}$

let $A = L(G)$ such that G:

$$S \rightarrow T \mid W$$

$$\begin{array}{ll} T \rightarrow UV & W \rightarrow XY \\ U \rightarrow aUb \mid \varepsilon & X \rightarrow aX \mid \varepsilon \\ V \rightarrow cV \mid \varepsilon & Y \rightarrow bYc \mid \varepsilon \end{array}$$

the union we used has a non-empty intersection, where letters a, b and c all are in equal number

Remark

there exist context-free sets without unambiguous grammars (inherently ambiguous)

Example

$A = \{a^i b^j c^k \mid i = j \text{ or } j = k\}$ is context-free and inherently ambiguous

$A = \{a^i b^i c^k\} \cup \{a^i b^j c^j\}$

let $A = L(G)$ such that G:

$$S \to T \mid W$$

$$\begin{array}{ll} T \to UV & W \to XY \\ U \to aUb \mid \varepsilon & X \to aX \mid \varepsilon \\ V \to cV \mid \varepsilon & Y \to bYc \mid \varepsilon \end{array}$$

the union we used has a non-empty intersection, where letters a, b and c all are in equal number

Lemma

there is no CFG G' such that $L(G')$ is unambiguous with $L(G') = A$

Remark

1. given an ambiguous CFG G, the language $L(G)$ may or may not be ambiguous

Remark

1. given an ambiguous CFG G, the language $L(G)$ may or may not be ambiguous
   - one can find an unambiguous CFG G' such that $L(G') = L(G)$

Remark

1. given an ambiguous CFG G, the language $L(G)$ may or may not be ambiguous
   - one can find an unambiguous CFG G' such that $L(G') = L(G)$
2. there is no algorithm to convert ambiguous CFG to unambiguous CFG

Remark

1. given an ambiguous CFG G, the language $L(G)$ may or may not be ambiguous
   - one can find an unambiguous CFG G' such that $L(G') = L(G)$
2. there is no algorithm to convert ambiguous CFG to unambiguous CFG
3. unambiguous context free languages can be parsed by deterministic push down automata

Thanks! & Questions?