Formalizing DFAs

DFAs

A DFA, M, is a quintuple, $M = (Q, \Sigma, q_0, \delta, A)$, where

- Q is a finite set of states,
- Σ is a finite set of symbols (an alphabet),
- $q_0 \in Q$ is a special start state,
- $A \subseteq Q$ is the set of accepting states,
- $\delta : Q \times \Sigma \to Q$ is a transition function.

Formalizing DFAs

DFAs

A DFA, M, is a quintuple, $M = (Q, \Sigma, q_0, \delta, A)$, where

- Q is a finite set of states,
- Σ is a finite set of symbols (an alphabet),
- $q_0 \in Q$ is a special start state,
- $A \subseteq Q$ is the set of accepting states,
- $\delta : Q \times \Sigma \to Q$ is a transition function.

δ maps a state and an input character to another state.
Recall, $Q \times \Sigma$ is the set of all ordered pairs (q, σ) such that $q \in Q$ and $\sigma \in \Sigma$.

Formalizing DFAs

DFAs

A DFA, M, is a quintuple, $M = (Q, \Sigma, q_0, \delta, A)$, where

- Q is a finite set of states,
- Σ is a finite set of symbols (an alphabet),
- $q_0 \in Q$ is a special start state,
- $A \subseteq Q$ is the set of accepting states,
- $\delta : Q \times \Sigma \to Q$ is a transition function.

δ maps a state and an input character to another state.
Recall, $Q \times \Sigma$ is the set of all ordered pairs (q, σ) such that $q \in Q$ and $\sigma \in \Sigma$.
$\delta(q, \sigma)$ denotes the state of the computation when you start in state q and the next character is σ.

Formalizing DFAs

DFAs

A DFA, M, is a quintuple, $M = (Q, \Sigma, q_0, \delta, A)$, where

- Q is a finite set of states,
- Σ is a finite set of symbols (an alphabet),
- $q_0 \in Q$ is a special start state,
- $A \subseteq Q$ is the set of accepting states,
- $\delta : Q \times \Sigma \rightarrow Q$ is a transition function.

δ maps a state and an input character to another state.
Recall, $Q \times \Sigma$ is the set of all ordered pairs (q, σ) such that $q \in Q$ and $\sigma \in \Sigma$.
$\delta(q, \sigma)$ denotes the state of the computation when you start in state q and the next character is σ.

Note that δ is a total function: it is defined *for every possible input pair*.
This assumes a trap state, and assures that the function table for δ doesn't have any empty cells.

Example

$M = (Q, \Sigma, q_0, \delta, A)$ where $q_0 \in Q$ is the start state, $A \subseteq Q$ are the accepting states,
$\delta : Q \times \Sigma \to Q$
$M = (\{q_0, q_1, q_2\}, \{a, b\}, q_0, \delta, \{q_1\})$, where δ is as follows:

	a	b
q_0	q_1	q_2
q_1	q_1	q_1
q_2	q_2	q_2

Example

$M = (Q, \Sigma, q_0, \delta, A)$ where $q_0 \in Q$ is the start state, $A \subseteq Q$ are the accepting states, $\delta : Q \times \Sigma \to Q$

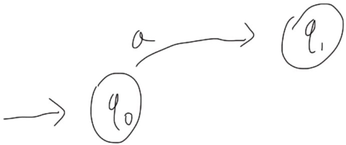
$M = (\{q_0, q_1, q_2\}, \{a, b\}, q_0, \delta, \{q_1\})$, where δ is as follows:

	a	b
q_0	q_1	q_2
q_1	q_1	q_1
q_2	q_2	q_2

Example

$M = (Q, \Sigma, q_0, \delta, A)$ where $q_0 \in Q$ is the start state, $A \subseteq Q$ are the accepting states,
$\delta : Q \times \Sigma \to Q$
$M = (\{q_0, q_1, q_2\}, \{a, b\}, q_0, \delta, \{q_1\})$, where δ is as follows:

	a	b
q_0	q_1	q_2
q_1	q_1	q_1
q_2	q_2	q_2

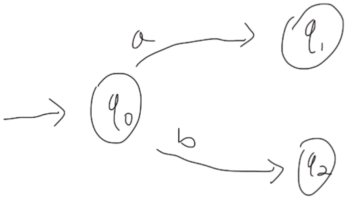


Example

$M = (Q, \Sigma, q_0, \delta, A)$ where $q_0 \in Q$ is the start state, $A \subseteq Q$ are the accepting states, $\delta : Q \times \Sigma \to Q$

$M = (\{q_0, q_1, q_2\}, \{a, b\}, q_0, \delta, \{q_1\})$, where δ is as follows:

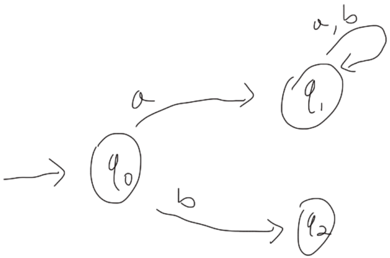
	a	b
q_0	q_1	q_2
q_1	q_1	q_1
q_2	q_2	q_2



Example

$M = (Q, \Sigma, q_0, \delta, A)$ where $q_0 \in Q$ is the start state, $A \subseteq Q$ are the accepting states,
$\delta : Q \times \Sigma \to Q$
$M = (\{q_0, q_1, q_2\}, \{a, b\}, q_0, \delta, \{q_1\})$, where δ is as follows:

	a	b
q_0	q_1	q_2
q_1	q_1	q_1
q_2	q_2	q_2

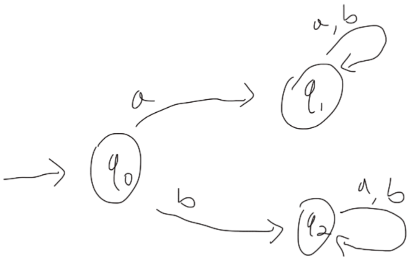


Example

$M = (Q, \Sigma, q_0, \delta, A)$ where $q_0 \in Q$ is the start state, $A \subseteq Q$ are the accepting states, $\delta : Q \times \Sigma \to Q$

$M = (\{q_0, q_1, q_2\}, \{a, b\}, q_0, \delta, \{q_1\})$, where δ is as follows:

	a	b
q_0	q_1	q_2
q_1	q_1	q_1
q_2	q_2	q_2

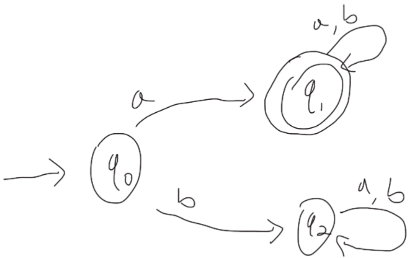


Example

$M = (Q, \Sigma, q_0, \delta, A)$ where $q_0 \in Q$ is the start state, $A \subseteq Q$ are the accepting states, $\delta : Q \times \Sigma \to Q$

$M = (\{q_0, q_1, q_2\}, \{a, b\}, q_0, \delta, \{q_1\})$, where δ is as follows:

	a	b
q_0	q_1	q_2
q_1	q_1	q_1
q_2	q_2	q_2



δ^*

closure of δ

For a given DFA, $M = (Q, \Sigma, q_0, \delta, A)$, δ^* is a function that takes a state and a string as inpu, and produces a resulting state. That is, $\delta^* : Q \times \Sigma^* \to Q$, and

- For any $q \in Q$, $\delta^*(q, \Lambda) = q$,
- For any $q \in Q$, any $\sigma \in \Sigma$, and any $x \in \Sigma^*$, $\delta^*(q, x\sigma) = \delta(\delta^*(q, x), \sigma)$

DFA from RG

Lemma 9.2

If $L = \mathcal{L}(G)$ for some deterministic regular grammar G, then there exists a DFA M such that $L = \mathcal{L}(M)$.

.

DFA from RG

Lemma 9.2

If $L = \mathcal{L}(G)$ for some deterministic regular grammar G, then there exists a DFA M such that $L = \mathcal{L}(M)$.

Let $G = (V, \Sigma, S, P)$. We define DFA $M = (Q, \Sigma, q_0, \delta, A)$ as follows.

.

DFA from RG

Lemma 9.2

If $L = \mathcal{L}(G)$ for some deterministic regular grammar G, then there exists a DFA M such that $L = \mathcal{L}(M)$.

Let $G = (V, \Sigma, S, P)$. We define DFA $M = (Q, \Sigma, q_0, \delta, A)$ as follows.
$Q = V$,
$q_0 = S$,
$A = \{p \mid p \to \Lambda \in P\}$,

.

DFA from RG

Lemma 9.2

If $L = \mathcal{L}(G)$ for some deterministic regular grammar G, then there exists a DFA M such that $L = \mathcal{L}(M)$.

Let $G = (V, \Sigma, S, P)$. We define DFA $M = (Q, \Sigma, q_0, \delta, A)$ as follows.
$Q = V$,
$q_0 = S$,
$A = \{p \mid p \to \Lambda \in P\}$, and,
for every $p \in Q, \sigma \in \Sigma$, $\delta(p, \sigma) = q$ if $p \to \sigma q \in P$.

.

DFA from RG

Lemma 9.2

If $L = \mathcal{L}(G)$ for some deterministic regular grammar G, then there exists a DFA M such that $L = \mathcal{L}(M)$.

Let $G = (V, \Sigma, S, P)$. We define DFA $M = (Q, \Sigma, q_0, \delta, A)$ as follows.
$Q = V$,
$q_0 = S$,
$A = \{p \mid p \to \Lambda \in P\}$, and,
for every $p \in Q, \sigma \in \Sigma$, $\delta(p, \sigma) = q$ if $p \to \sigma q \in P$.
If δ is not a complete function, we can add a trap state to Q.

.

DFA from RG

Lemma 9.2

If $L = \mathcal{L}(G)$ for some deterministic regular grammar G, then there exists a DFA M such that $L = \mathcal{L}(M)$.

Let $G = (V, \Sigma, S, P)$. We define DFA $M = (Q, \Sigma, q_0, \delta, A)$ as follows.
$Q = V$,
$q_0 = S$,
$A = \{p \mid p \to \Lambda \in P\}$, and,
for every $p \in Q, \sigma \in \Sigma$, $\delta(p, \sigma) = q$ if $p \to \sigma q \in P$.
If δ is not a complete function, we can add a trap state to Q.

Claim: For every $x \in \Sigma^*$ and $A \in V$, $S \stackrel{*}{\Rightarrow} xA$ iff $\delta^*(q_0, x) = A$.

.

DFA from RG

Lemma 9.2

If $L = \mathcal{L}(G)$ for some deterministic regular grammar G, then there exists a DFA M such that $L = \mathcal{L}(M)$.

Let $G = (V, \Sigma, S, P)$. We define DFA $M = (Q, \Sigma, q_0, \delta, A)$ as follows.
$Q = V$,
$q_0 = S$,
$A = \{p \mid p \to \Lambda \in P\}$, and,
for every $p \in Q, \sigma \in \Sigma$, $\delta(p, \sigma) = q$ if $p \to \sigma q \in P$.
If δ is not a complete function, we can add a trap state to Q.

Claim: For every $x \in \Sigma^*$ and $A \in V$, $S \overset{*}{\Rightarrow} xA$ iff $\delta^*(q_0, x) = A$.
Proof is by induction on the length of x. When $|x| = 0$, note that S can generate A iff $A = S$ (because G is regular), and $\delta(q_0, \Lambda) = S$.

.

DFA from RG

Lemma 9.2

If $L = \mathcal{L}(G)$ for some deterministic regular grammar G, then there exists a DFA M such that $L = \mathcal{L}(M)$.

Let $G = (V, \Sigma, S, P)$. We define DFA $M = (Q, \Sigma, q_0, \delta, A)$ as follows.
$Q = V$,
$q_0 = S$,
$A = \{p \mid p \to \Lambda \in P\}$, and,
for every $p \in Q, \sigma \in \Sigma$, $\delta(p, \sigma) = q$ if $p \to \sigma q \in P$.
If δ is not a complete function, we can add a trap state to Q.

Claim: For every $x \in \Sigma^*$ and $A \in V$, $S \stackrel{*}{\Rightarrow} xA$ iff $\delta^*(q_0, x) = A$.
Proof is by induction on the length of x. When $|x| = 0$, note that S can generate A iff $A = S$ (because G is regular), and $\delta(q_0, \Lambda) = S$.

Suppose the claim holds for all x of length $k \geq 0$.

.

DFA from RG

Lemma 9.2

If $L = \mathcal{L}(G)$ for some deterministic regular grammar G, then there exists a DFA M such that $L = \mathcal{L}(M)$.

Let $G = (V, \Sigma, S, P)$. We define DFA $M = (Q, \Sigma, q_0, \delta, A)$ as follows.
$Q = V$,
$q_0 = S$,
$A = \{p \mid p \to \Lambda \in P\}$, and,
for every $p \in Q, \sigma \in \Sigma$, $\delta(p, \sigma) = q$ if $p \to \sigma q \in P$.
If δ is not a complete function, we can add a trap state to Q.

Claim: For every $x \in \Sigma^*$ and $A \in V$, $S \overset{*}{\Rightarrow} xA$ iff $\delta^*(q_0, x) = A$.
Proof is by induction on the length of x. When $|x| = 0$, note that S can generate A iff $A = S$ (because G is regular), and $\delta(q_0, \Lambda) = S$.

Suppose the claim holds for all x of length $k \geq 0$.
$(S \overset{*}{\Rightarrow} xA) \to (\delta^*(q_0, x) = A)$
Suppose that in the $k+1$st step of a derivation of string yb of length $k+1$, we use the rule $A \to bC$:
we have $S \overset{*}{\Rightarrow} yA \Rightarrow ybC$.

.

DFA from RG

Lemma 9.2

If $L = \mathcal{L}(G)$ for some deterministic regular grammar G, then there exists a DFA M such that $L = \mathcal{L}(M)$.

Let $G = (V, \Sigma, S, P)$. We define DFA $M = (Q, \Sigma, q_0, \delta, A)$ as follows.
$Q = V$,
$q_0 = S$,
$A = \{p \mid p \to \Lambda \in P\}$, and,
for every $p \in Q, \sigma \in \Sigma$, $\delta(p, \sigma) = q$ if $p \to \sigma q \in P$.
If δ is not a complete function, we can add a trap state to Q.

Claim: For every $x \in \Sigma^*$ and $A \in V$, $S \stackrel{*}{\Rightarrow} xA$ iff $\delta^*(q_0, x) = A$.
Proof is by induction on the length of x. When $|x| = 0$, note that S can generate A iff $A = S$ (because G is regular), and $\delta(q_0, \Lambda) = S$.

Suppose the claim holds for all x of length $k \geq 0$.
$(S \stackrel{*}{\Rightarrow} xA) \to (\delta^*(q_0, x) = A)$
Suppose that in the $k+1$st step of a derivation of string yb of length $k+1$, we use the rule $A \to bC$:
we have $S \stackrel{*}{\Rightarrow} yA \Rightarrow ybC$.
By the inductive hypothesis, we have $\delta^*(q_0, y) = A$.

DFA from RG

Lemma 9.2

If $L = \mathcal{L}(G)$ for some deterministic regular grammar G, then there exists a DFA M such that $L = \mathcal{L}(M)$.

Let $G = (V, \Sigma, S, P)$. We define DFA $M = (Q, \Sigma, q_0, \delta, A)$ as follows.
$Q = V$,
$q_0 = S$,
$A = \{p \mid p \to \Lambda \in P\}$, and,
for every $p \in Q, \sigma \in \Sigma$, $\delta(p, \sigma) = q$ if $p \to \sigma q \in P$.
If δ is not a complete function, we can add a trap state to Q.

Claim: For every $x \in \Sigma^*$ and $A \in V$, $S \overset{*}{\Rightarrow} xA$ iff $\delta^*(q_0, x) = A$.
Proof is by induction on the length of x. When $|x| = 0$, note that S can generate A iff $A = S$ (because G is regular), and $\delta(q_0, \Lambda) = S$.

Suppose the claim holds for all x of length $k \geq 0$.
$(S \overset{*}{\Rightarrow} xA) \to (\delta^*(q_0, x) = A)$
Suppose that in the $k + 1$st step of a derivation of string yb of length $k + 1$, we use the rule $A \to bC$:
we have $S \overset{*}{\Rightarrow} yA \Rightarrow ybC$.
By the inductive hypothesis, we have $\delta^*(q_0, y) = A$.
By the way we constructed M, we have $\delta(A, b) = C$
Therefore: $\delta^*(q_0, yb) = C$.

DFA from RG

Lemma 9.2

If $L = \mathcal{L}(G)$ for some deterministic regular grammar G, then there exists a DFA M such that $L = \mathcal{L}(M)$.

Let $G = (V, \Sigma, S, P)$. We define DFA $M = (Q, \Sigma, q_0, \delta, A)$ as follows.
$Q = V$,
$q_0 = S$,
$A = \{p \mid p \to \Lambda \in P\}$, and,
for every $p \in Q, \sigma \in \Sigma$, $\delta(p, \sigma) = q$ if $p \to \sigma q \in P$.
If δ is not a complete function, we can add a trap state to Q.

Claim: For every $x \in \Sigma^*$ and $A \in V$, $S \overset{*}{\Rightarrow} xA$ iff $\delta^*(q_0, x) = A$.
Proof is by induction on the length of x. When $|x| = 0$, note that S can generate A iff $A = S$ (because G is regular), and $\delta(q_0, \Lambda) = S$.

Suppose the claim holds for all x of length $k \geq 0$.
$(\delta^*(q_0, x) = A) \to (S \overset{*}{\Rightarrow} xA)$

DFA from RG

Lemma 9.2

If $L = \mathcal{L}(G)$ for some deterministic regular grammar G, then there exists a DFA M such that $L = \mathcal{L}(M)$.

Let $G = (V, \Sigma, S, P)$. We define DFA $M = (Q, \Sigma, q_0, \delta, A)$ as follows.
$Q = V$,
$q_0 = S$,
$A = \{p \mid p \to \Lambda \in P\}$, and,
for every $p \in Q, \sigma \in \Sigma$, $\delta(p, \sigma) = q$ if $p \to \sigma q \in P$.
If δ is not a complete function, we can add a trap state to Q.

Claim: For every $x \in \Sigma^*$ and $A \in V$, $S \stackrel{*}{\Rightarrow} xA$ iff $\delta^*(q_0, x) = A$.
Proof is by induction on the length of x. When $|x| = 0$, note that S can generate A iff $A = S$ (because G is regular), and $\delta(q_0, \Lambda) = S$.

Suppose the claim holds for all x of length $k \geq 0$.
$(\delta^*(q_0, x) = A) \to (S \stackrel{*}{\Rightarrow} xA)$
Suppose that on some input yb of length $k + 1$, the $k + 1$st transition in M was $\delta(A, b) = C$.

DFA from RG

Lemma 9.2

If $L = \mathcal{L}(G)$ for some deterministic regular grammar G, then there exists a DFA M such that $L = \mathcal{L}(M)$.

Let $G = (V, \Sigma, S, P)$. We define DFA $M = (Q, \Sigma, q_0, \delta, A)$ as follows.
$Q = V$,
$q_0 = S$,
$A = \{p \mid p \to \Lambda \in P\}$, and,
for every $p \in Q, \sigma \in \Sigma$, $\delta(p, \sigma) = q$ if $p \to \sigma q \in P$.
If δ is not a complete function, we can add a trap state to Q.

Claim: For every $x \in \Sigma^*$ and $A \in V$, $S \stackrel{*}{\Rightarrow} xA$ iff $\delta^*(q_0, x) = A$.
Proof is by induction on the length of x. When $|x| = 0$, note that S can generate A iff $A = S$ (because G is regular), and $\delta(q_0, \Lambda) = S$.

Suppose the claim holds for all x of length $k \geq 0$.
$(\delta^*(q_0, x) = A) \to (S \stackrel{*}{\Rightarrow} xA)$
Suppose that on some input yb of length $k + 1$, the $k + 1$st transition in M was $\delta(A, b) = C$.
We have: $\delta^*(q_0, y) = A$.

DFA from RG

Lemma 9.2

If $L = \mathcal{L}(G)$ for some deterministic regular grammar G, then there exists a DFA M such that $L = \mathcal{L}(M)$.

Let $G = (V, \Sigma, S, P)$. We define DFA $M = (Q, \Sigma, q_0, \delta, A)$ as follows.
$Q = V$,
$q_0 = S$,
$A = \{p \mid p \rightarrow \Lambda \in P\}$, and,
for every $p \in Q, \sigma \in \Sigma$, $\delta(p, \sigma) = q$ if $p \rightarrow \sigma q \in P$.
If δ is not a complete function, we can add a trap state to Q.

Claim: For every $x \in \Sigma^*$ and $A \in V$, $S \stackrel{*}{\Rightarrow} xA$ iff $\delta^*(q_0, x) = A$.
Proof is by induction on the length of x. When $|x| = 0$, note that S can generate A iff $A = S$ (because G is regular), and $\delta(q_0, \Lambda) = S$.

Suppose the claim holds for all x of length $k \geq 0$.
$(\delta^*(q_0, x) = A) \rightarrow (S \stackrel{*}{\Rightarrow} xA)$
Suppose that on some input yb of length $k+1$, the $k+1$st transition in M was $\delta(A, b) = C$.
We have: $\delta^*(q_0, y) = A$.
By the inductive hypothesis, we have $S \stackrel{*}{\Rightarrow} yA$

DFA from RG

Lemma 9.2

If $L = \mathcal{L}(G)$ for some deterministic regular grammar G, then there exists a DFA M such that $L = \mathcal{L}(M)$.

Let $G = (V, \Sigma, S, P)$. We define DFA $M = (Q, \Sigma, q_0, \delta, A)$ as follows.
$Q = V$,
$q_0 = S$,
$A = \{p \mid p \to \Lambda \in P\}$, and,
for every $p \in Q, \sigma \in \Sigma$, $\delta(p, \sigma) = q$ if $p \to \sigma q \in P$.
If δ is not a complete function, we can add a trap state to Q.

Claim: For every $x \in \Sigma^*$ and $A \in V$, $S \overset{*}{\Rightarrow} xA$ iff $\delta^*(q_0, x) = A$.
Proof is by induction on the length of x. When $|x| = 0$, note that S can generate A iff $A = S$ (because G is regular), and $\delta(q_0, \Lambda) = S$.

Suppose the claim holds for all x of length $k \geq 0$.
$(\delta^*(q_0, x) = A) \to (S \overset{*}{\Rightarrow} xA)$
Suppose that on some input yb of length $k+1$, the $k+1$st transition in M was $\delta(A, b) = C$.
We have: $\delta^*(q_0, y) = A$.
By the inductive hypothesis, we have $S \overset{*}{\Rightarrow} yA$
By the way we defined M, there must be a rule $A \to bC$.

DFA from RG

Lemma 9.2

If $L = \mathcal{L}(G)$ for some deterministic regular grammar G, then there exists a DFA M such that $L = \mathcal{L}(M)$.

Let $G = (V, \Sigma, S, P)$. We define DFA $M = (Q, \Sigma, q_0, \delta, A)$ as follows.
$Q = V$,
$q_0 = S$,
$A = \{p \mid p \to \Lambda \in P\}$, and,
for every $p \in Q, \sigma \in \Sigma$, $\delta(p, \sigma) = q$ if $p \to \sigma q \in P$.
If δ is not a complete function, we can add a trap state to Q.

Claim: For every $x \in \Sigma^*$ and $A \in V$, $S \stackrel{*}{\Rightarrow} xA$ iff $\delta^*(q_0, x) = A$.
Proof is by induction on the length of x. When $|x| = 0$, note that S can generate A iff $A = S$ (because G is regular), and $\delta(q_0, \Lambda) = S$.

Suppose the claim holds for all x of length $k \geq 0$.
$(\delta^*(q_0, x) = A) \to (S \stackrel{*}{\Rightarrow} xA)$
Suppose that on some input yb of length $k+1$, the $k+1$st transition in M was $\delta(A, b) = C$.
We have: $\delta^*(q_0, y) = A$.
By the inductive hypothesis, we have $S \stackrel{*}{\Rightarrow} yA$
By the way we defined M, there must be a rule $A \to bC$.
Therefore, $S \stackrel{*}{\Rightarrow} ybC$.

DFA from RG

Lemma 9.2

If $L = \mathcal{L}(G)$ for some deterministic regular grammar G, then there exists a DFA M such that $L = \mathcal{L}(M)$.

Let $G = (V, \Sigma, S, P)$. We define DFA $M = (Q, \Sigma, q_0, \delta, A)$ as follows.
$Q = V$,
$q_0 = S$,
$A = \{p \mid p \to \Lambda \in P\}$, and,
for every $p \in Q, \sigma \in \Sigma$, $\delta(p, \sigma) = q$ if $p \to \sigma q \in P$.
If δ is not a complete function, we can add a trap state to Q.

Claim: For every $x \in \Sigma^*$ and $A \in V$, $S \stackrel{*}{\Rightarrow} xA$ iff $\delta^*(q_0, x) = A$.
Proof is by induction on the length of x. When $|x| = 0$, note that S can generate A iff $A = S$ (because G is regular), and $\delta(q_0, \Lambda) = S$.

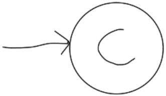
Suppose the claim holds for all x of length $k \geq 0$.
$(\delta^*(q_0, x) = A) \to (S \stackrel{*}{\Rightarrow} xA)$
Suppose that on some input yb of length $k + 1$, the $k + 1$st transition in M was $\delta(A, b) = C$.
We have: $\delta^*(q_0, y) = A$.
By the inductive hypothesis, we have $S \stackrel{*}{\Rightarrow} yA$
By the way we defined M, there must be a rule $A \to bC$.
Therefore, $S \stackrel{*}{\Rightarrow} ybC$.
(Why does this suffice for the proof of the Lemma?)

Example: DFA from G

Consider the grammar $G = (\{C, D, E, F\}, \{a, b\}, C, P)$, where
$P = \{C \to aD; D \to aC; E \to aF; F \to aE; C \to bE; D \to bF; E \to bC;$
$F \to bD; F \to \Lambda\}$

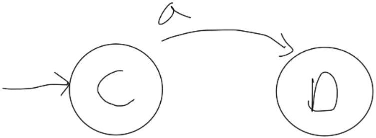
Example: DFA from G

Consider the grammar $G = (\{C, D, E, F\}, \{a, b\}, C, P)$, where
$P = \{C \to aD; D \to aC; E \to aF; F \to aE; C \to bE; D \to bF; E \to bC;$
$F \to bD; F \to \Lambda\}$



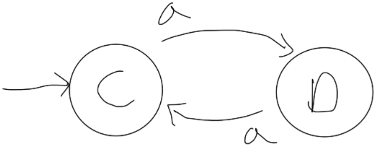
Example: DFA from G

Consider the grammar $G = (\{C, D, E, F\}, \{a, b\}, C, P)$, where
$P = \{C \to aD; D \to aC; E \to aF; F \to aE; C \to bE; D \to bF; E \to bC;$
$F \to bD; F \to \Lambda\}$



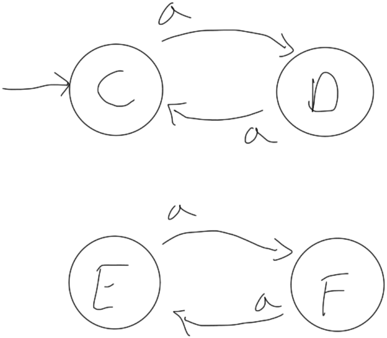
Example: DFA from G

Consider the grammar $G = (\{C, D, E, F\}, \{a, b\}, C, P)$, where
$P = \{C \to aD; D \to aC; E \to aF; F \to aE; C \to bE; D \to bF; E \to bC;$
$F \to bD; F \to \Lambda\}$



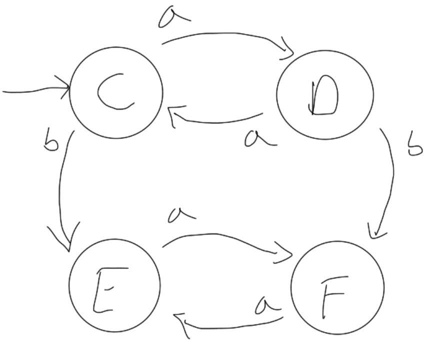
Example: DFA from G

Consider the grammar $G = (\{C, D, E, F\}, \{a, b\}, C, P)$, where
$P = \{C \to aD; D \to aC; E \to aF; F \to aE; C \to bE; D \to bF; E \to bC;$
$F \to bD; F \to \Lambda\}$



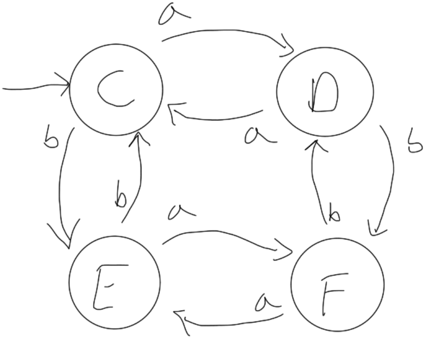
Example: DFA from G

Consider the grammar $G = (\{C, D, E, F\}, \{a, b\}, C, P)$, where
$P = \{C \to aD; D \to aC; E \to aF; F \to aE; C \to bE; D \to bF; E \to bC;$
$F \to bD; F \to \Lambda\}$



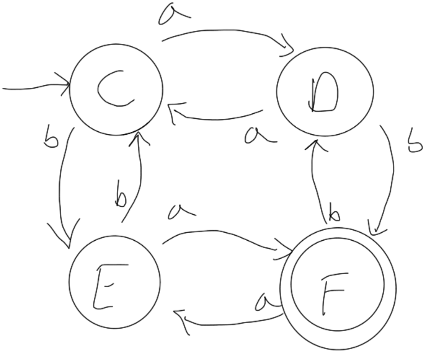
Example: DFA from G

Consider the grammar $G = (\{C, D, E, F\}, \{a, b\}, C, P)$, where
$P = \{C \to aD; D \to aC; E \to aF; F \to aE; C \to bE; D \to bF; E \to bC;$
$F \to bD; F \to \Lambda\}$



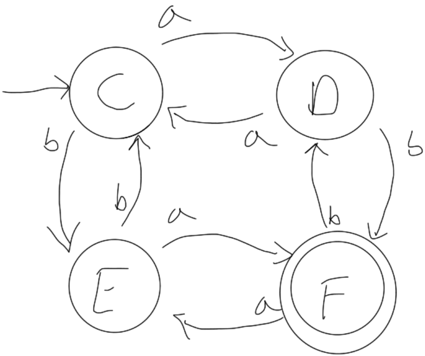
Example: DFA from G

Consider the grammar $G = (\{C, D, E, F\}, \{a, b\}, C, P)$, where
$P = \{C \to aD; D \to aC; E \to aF; F \to aE; C \to bE; D \to bF; E \to bC;$
$F \to bD; F \to \Lambda\}$



Example: DFA from G

Consider the grammar $G = (\{C, D, E, F\}, \{a, b\}, C, P)$, where
$P = \{C \to aD; D \to aC; E \to aF; F \to aE; C \to bE; D \to bF; E \to bC;$
$F \to bD; F \to \Lambda\}$



For example: consider $\delta^*(C, aab) = E$, and $C \Rightarrow aD \Rightarrow aaC \Rightarrow aabE$.

DFA from RG

Lemma 9.3

If $L = \mathcal{L}(M)$ for some deterministic finite automata M, then there exists a deterministic regular grammar G such that $L = \mathcal{L}(G)$.

DFA from RG

Lemma 9.3

If $L = \mathcal{L}(M)$ for some deterministic finite automata M, then there exists a deterministic regular grammar G such that $L = \mathcal{L}(G)$.

The same algorithm works in reverse!

DFA from RG

Lemma 9.3

If $L = \mathcal{L}(M)$ for some deterministic finite automata M, then there exists a deterministic regular grammar G such that $L = \mathcal{L}(G)$.

The same algorithm works in reverse!

Theorem 9.1

L is a regular language if and only if there exists a deterministic finite automata M such that $L = \mathcal{L}(M)$.