Lecture 6

Regular Grammars

COT 4420

Theory of Computation

Section 3.3

Grammar

- A **grammar** G is defined as a quadruple

  $G = (V, T, S, P)$

  V is a finite set of **variables**

  T is a finite set of **terminal** symbols

  S $\in$ V is a special variable called **start symbol**

  P is a finite set of **production rules** of the form

  $$x \rightarrow y$$

  where $x \in (V \cup T)^+$, $y \in (V \cup T)^*$

Linear Grammars

- Grammars with at most one variable at the right side of the production.

- Example: $S \rightarrow aSb \mid \lambda$
- Example: $S \rightarrow Ab$

  $A \rightarrow aAb \mid \lambda$

Right-linear Grammar

- A grammar G=(V, T, S, P) is said to be right-linear if all productions are of the form:

$A \rightarrow xB$

or

$A \rightarrow x$ $\qquad x \in T^*, \quad A, B \in V$

string of terminals

Left-linear Grammar

- A grammar $G=(V, T, S, P)$ is said to be left-linear if all productions are of the form:

$$A \rightarrow Bx$$

or

$$A \rightarrow x \qquad x \in T^*, \quad A,B \in V$$

string of terminals

Linear Grammars
Example

- Right-Linear Grammar

  $S \rightarrow abS \mid a$

- Left-Linear Grammar

  $S \rightarrow Aab$

  $A \rightarrow Aab \mid B$

  $B \rightarrow a$

Regular Grammars

- A regular grammar is any right-linear or left-linear grammar.

G_1:

$S \rightarrow abS \mid a$

G_2:

$S \rightarrow Aab$

$A \rightarrow Aab \mid B$

$B \rightarrow a$

Regular Grammars

What about this Grammar?

$S \rightarrow A$

$A \rightarrow aB \mid \lambda$

$B \rightarrow Ab$

- Is this grammar linear?
- Is this grammar regular?

This grammar is neither right-linear nor left-linear ➔ it is not regular.

Regular grammars and Regular languages

Regular grammars generate regular languages.

Example:

G_1:

$S \rightarrow abS \mid a$

$L(G_1) = (ab)^*a$

G_2:

$S \rightarrow Aab$

$A \rightarrow Aab \mid B$

$B \rightarrow a$

$L(G_2) = aab(ab)^*$

Theorem

Languages generated by Regular Grammars = Regular languages

Part 1) Any regular grammar generates a regular language

Part2) Any regular language is generated by a regular grammar

Proof - Part 1

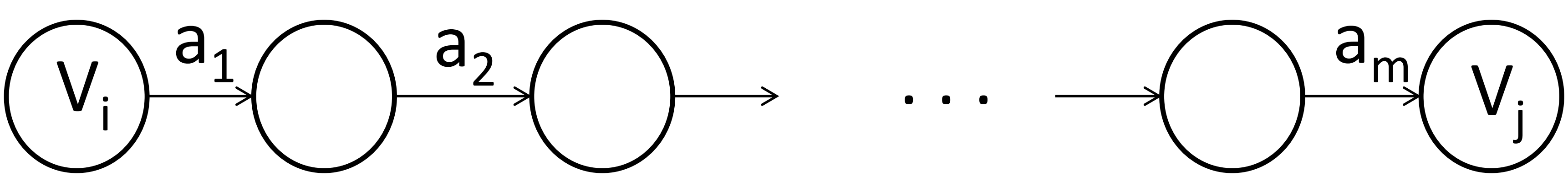
Theorem: Let G = (V, T, S, P) be a right-linear grammar. Then L(G) is a regular language.

In order to prove this we construct an NFA M such that L(M) = L(G).

Proof - Part 1
Right-linear grammar to NFA

1. For every Nonterminal there is a state in our NFA (The start symbol is the starting state)
2. For a production rule of the form $V_i \rightarrow a_1a_2 \ldots a_m V_j$ the automaton will have transitions to connect V_i and V_j such that

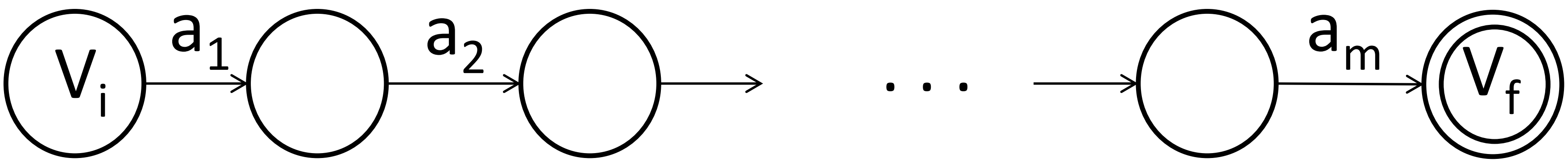
$$\delta^*(V_i, a_1a_2 \ldots a_m) = V_j$$



Proof - Part 1
Right-linear grammar to NFA

3. For each production $V_i \rightarrow a_1a_2....a_m$ the corresponding transition will be

$$\delta^*(V_i, a_1a_2...a_m) = V_f \qquad V_f\text{: a final state}$$



Right-linear grammar to NFA Example

Grammar G is right-linear:

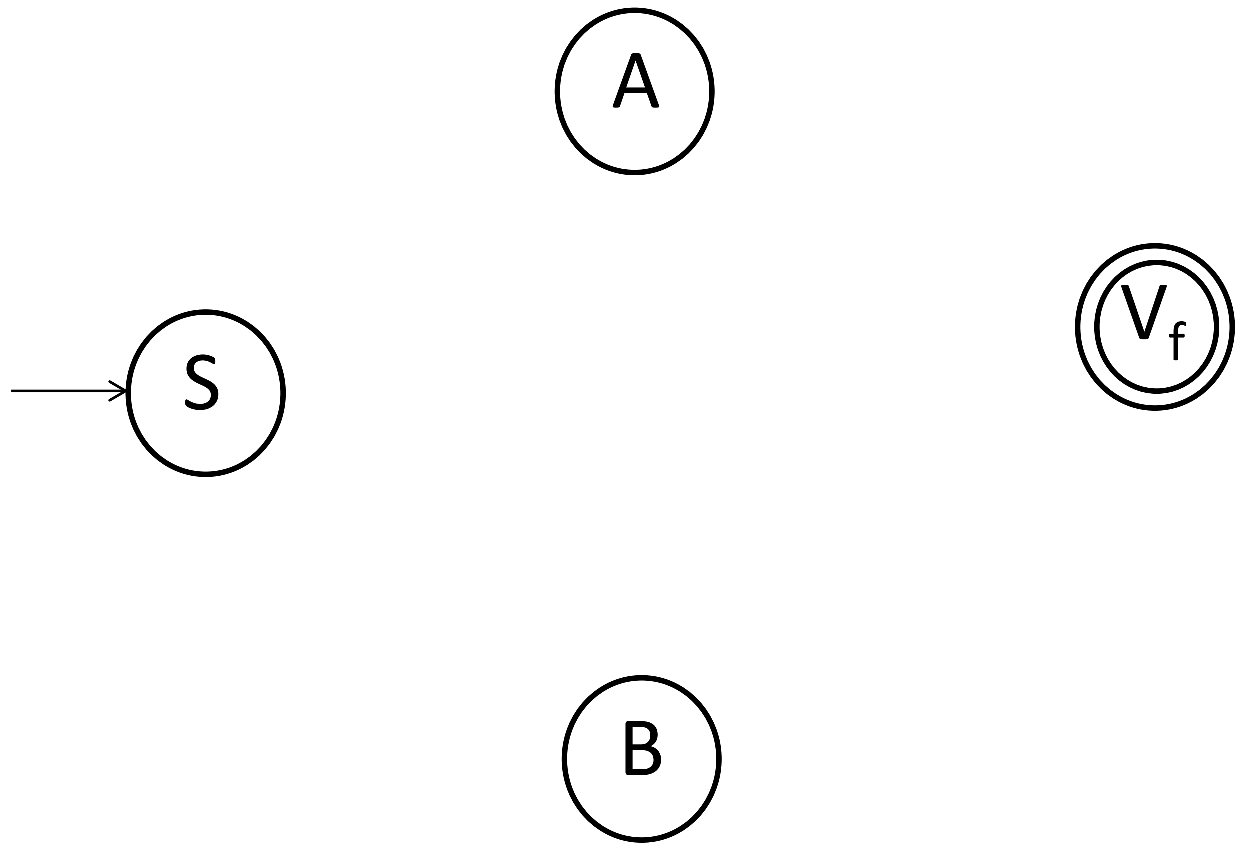
S → aA

S → B

A → aaB

B → bB

B → a



Right-linear grammar to NFA Example

Grammar G is right-linear:

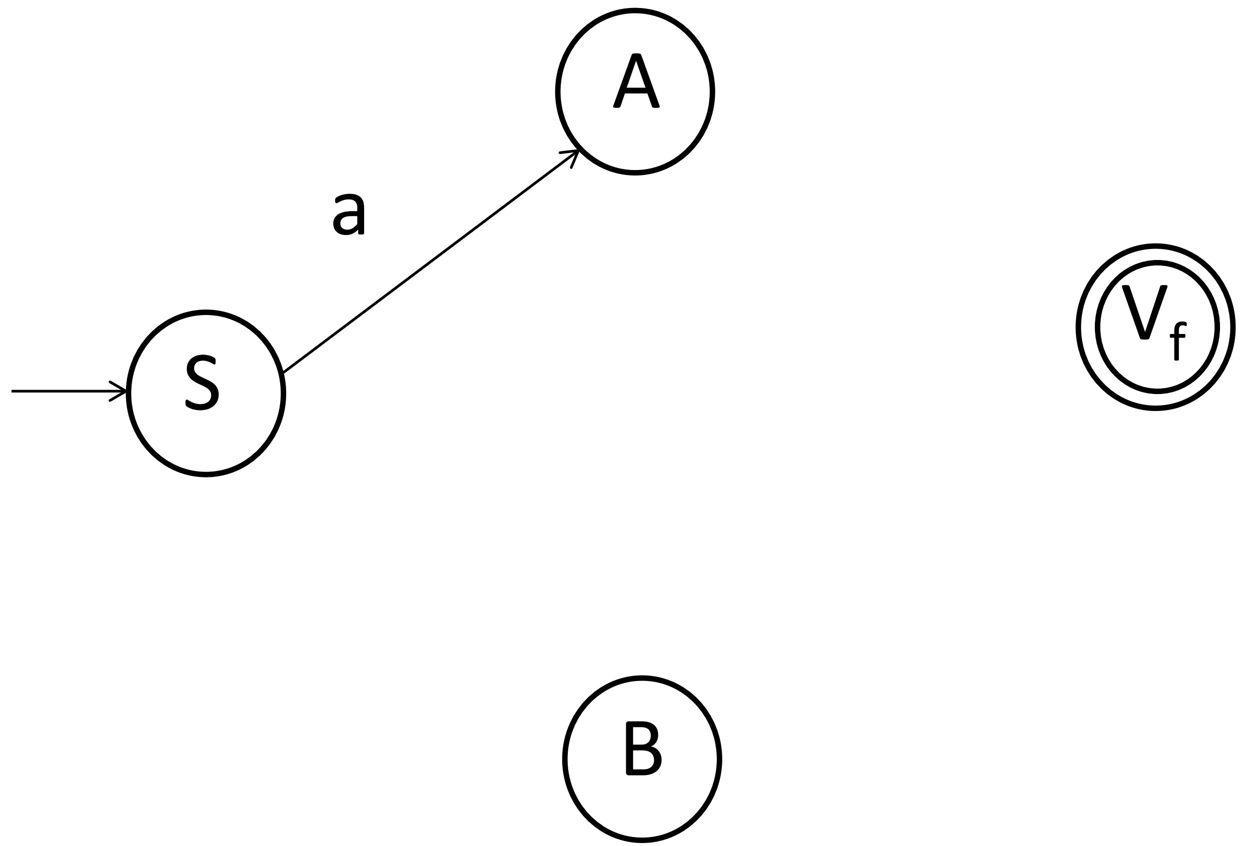
S → aA

S → B

A → aaB

B → bB

B → a



Right-linear grammar to NFA Example

Grammar G is right-linear:

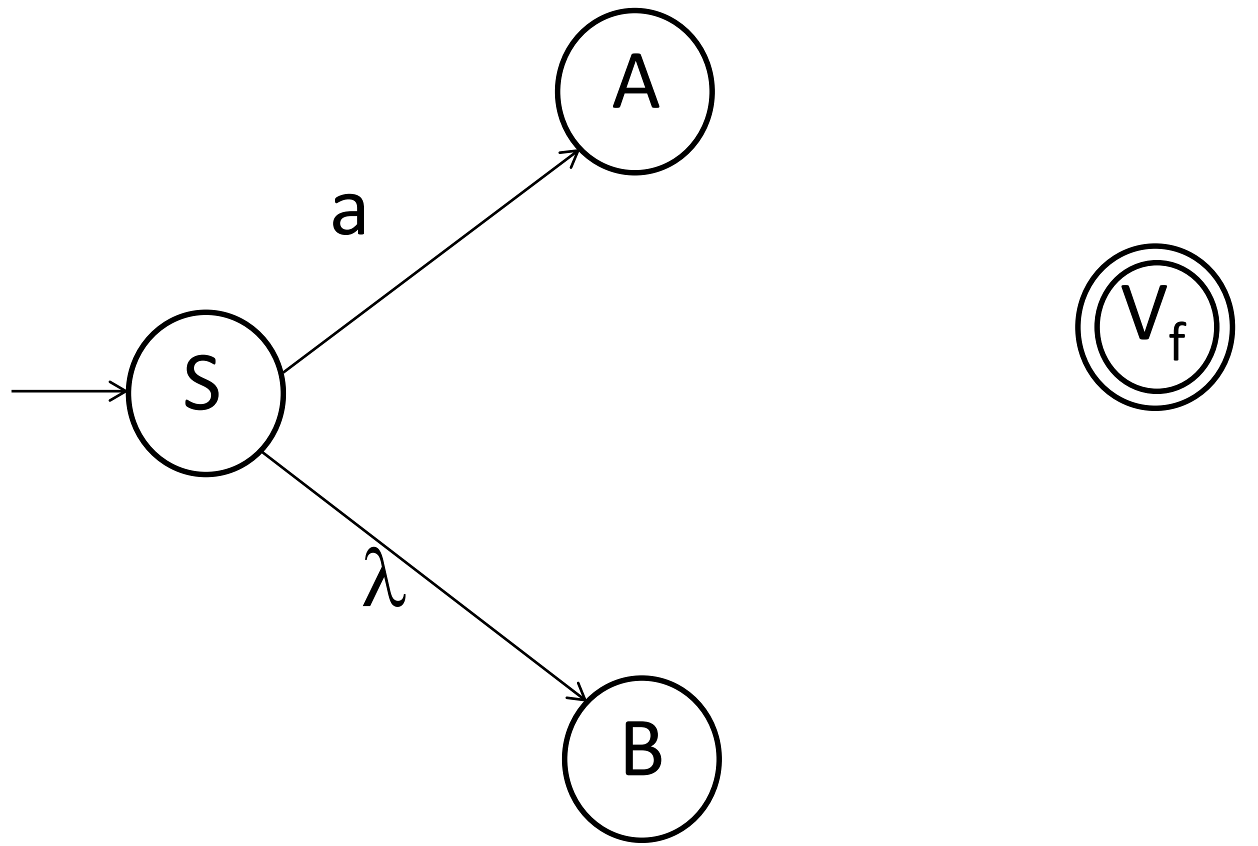
S → aA

S → B

A → aaB

B → bB

B → a



Right-linear grammar to NFA Example

Grammar G is right-linear:

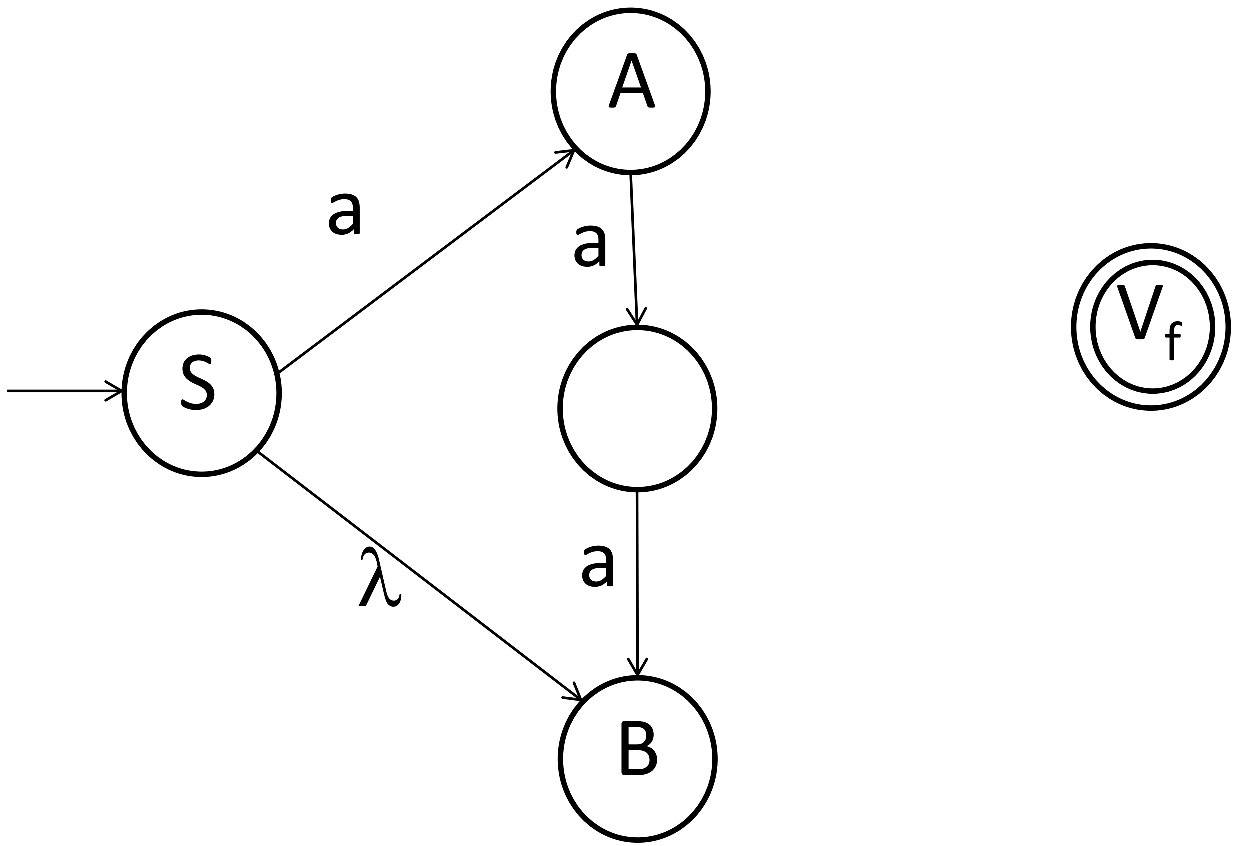
S → aA

S → B

A → aaB

B → bB

B → a



Right-linear grammar to NFA Example

Grammar G is right-linear:

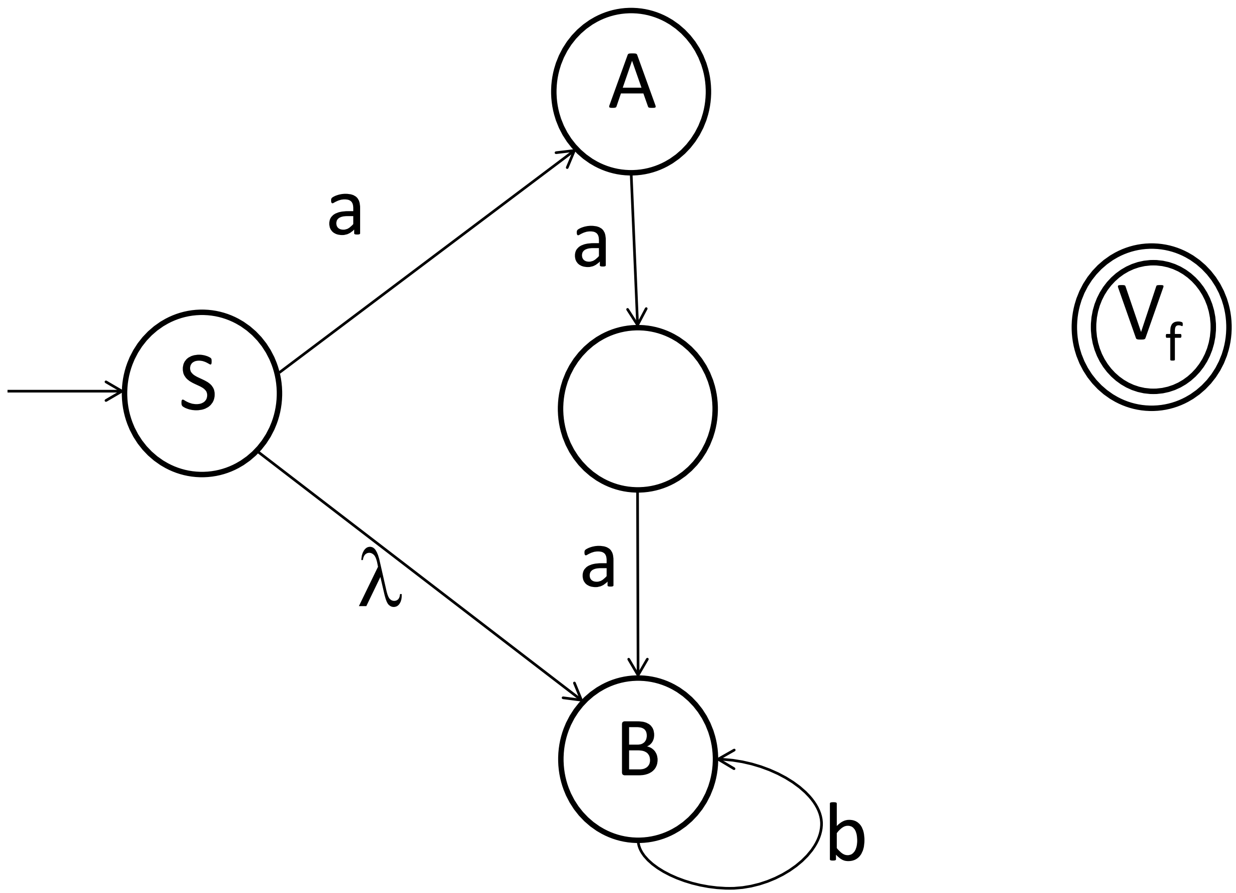
S → aA

S → B

A → aaB

B → bB

B → a



Right-linear grammar to NFA Example

Grammar G is right-linear:

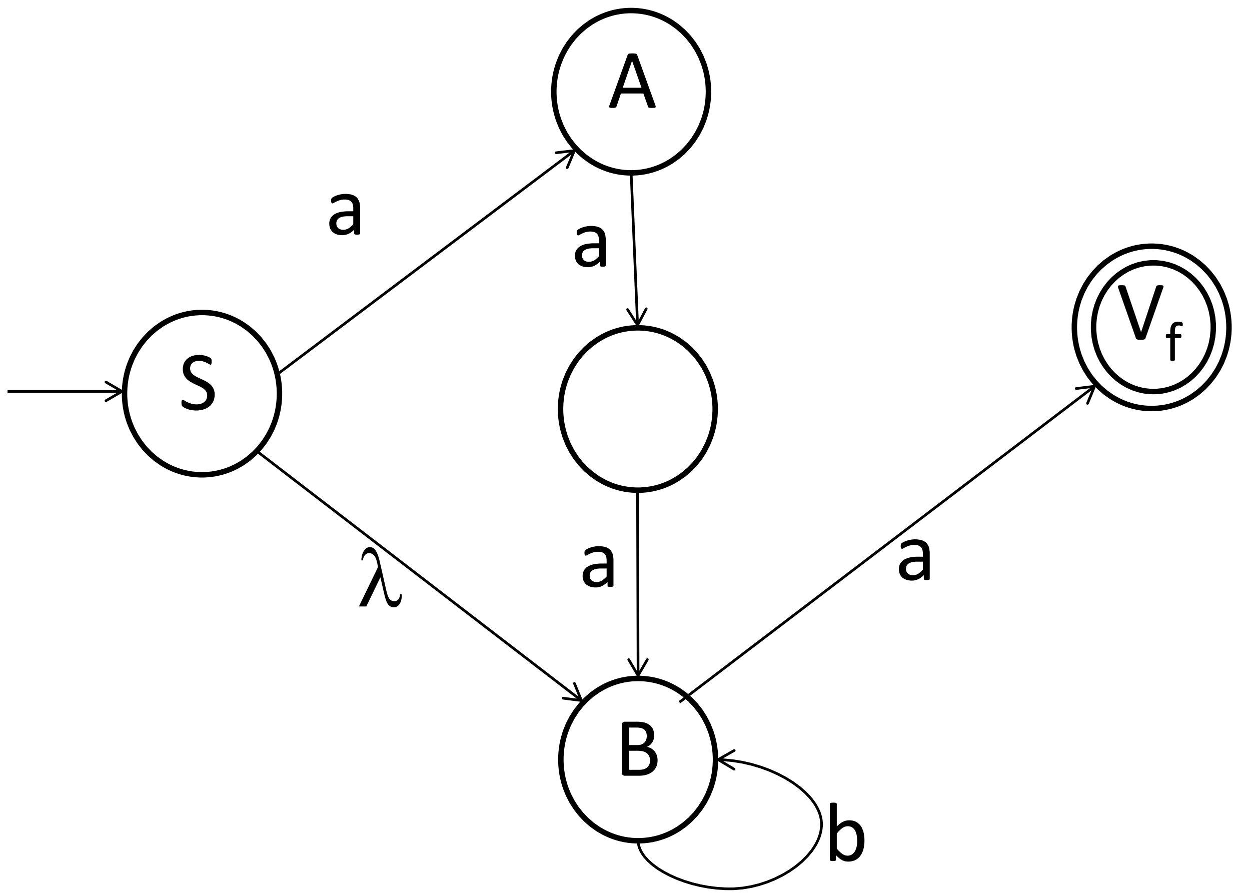
S → aA

S → B

A → aaB

B → bB

B → a



Right-linear grammar to NFA Example

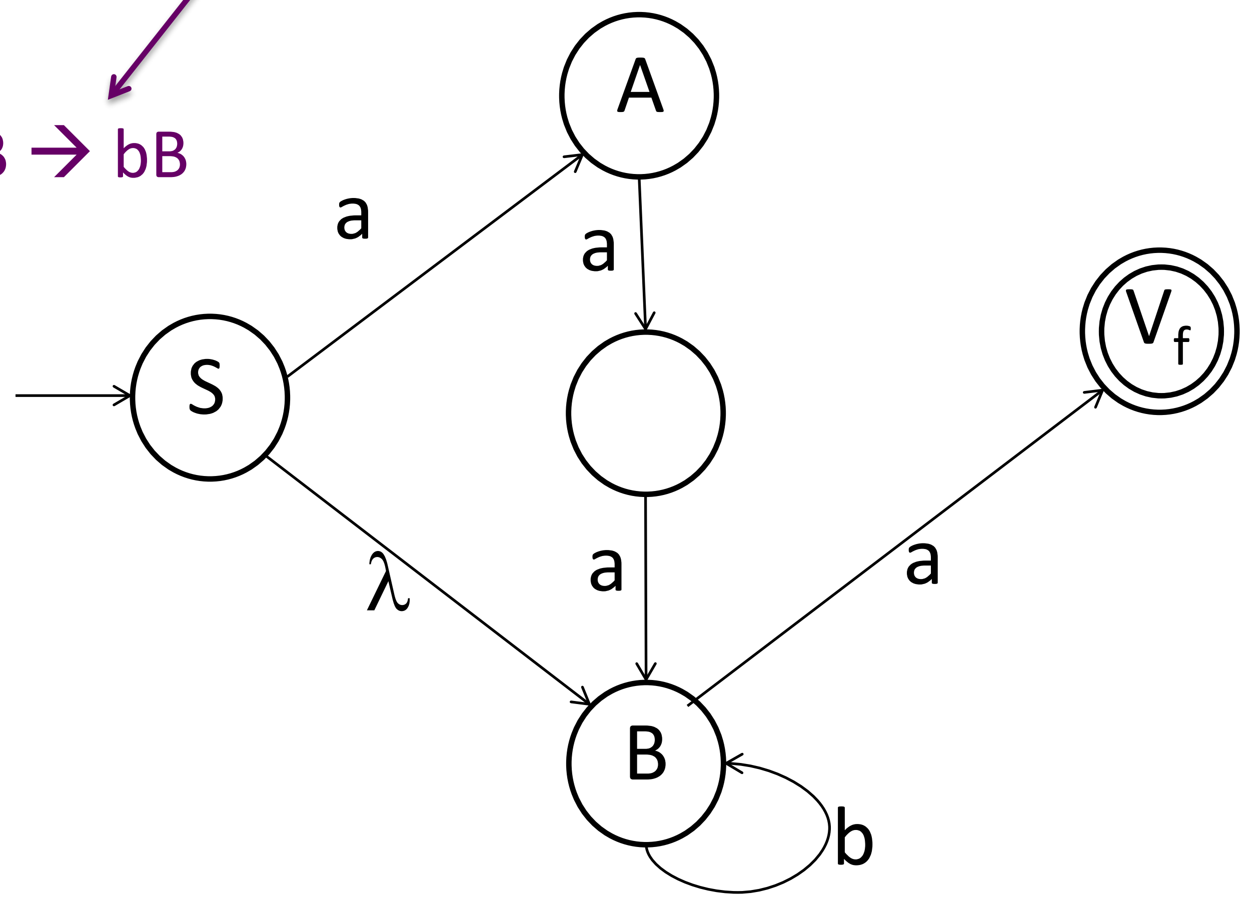
S → aA

B → a

S => aA => aaaB => aaabB => aaaba

A → aaB

B → bB



Proof - Part 1
In the case of Left-linear grammar

Theorem: Let G = (V, T, S, P) be a left-linear grammar. Then L(G) is a regular language.

Proof idea: We construct a right-linear grammar G' such that $L(G) = L(G')^R$

Proof - Part 1
In the case of Left-linear grammar

G is a left-linear grammar of the form:

$A \rightarrow Bv_1v_2...v_k$ or

$A \rightarrow v_1v_2...v_k$

Construct right-linear grammar G':

$A \rightarrow v_k....v_2v_1B$

$A \rightarrow v_k....v_2v_1$

Proof - Part 1
In the case of Left-linear grammar

It is easy to see that $L(G) = L(G')^R$

Since G' is right-linear, L(G') is a regular language. Therefore, $L(G')^R$ is also a regular language → L(G) is a regular language.

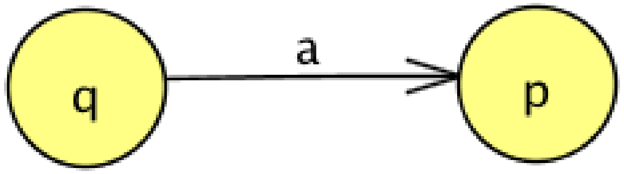
Proof – Part 2

Theorem: If L is a regular language on the alphabet Σ, then there exists a right-linear grammar G = (V, Σ, S, P) such that L = L(G).

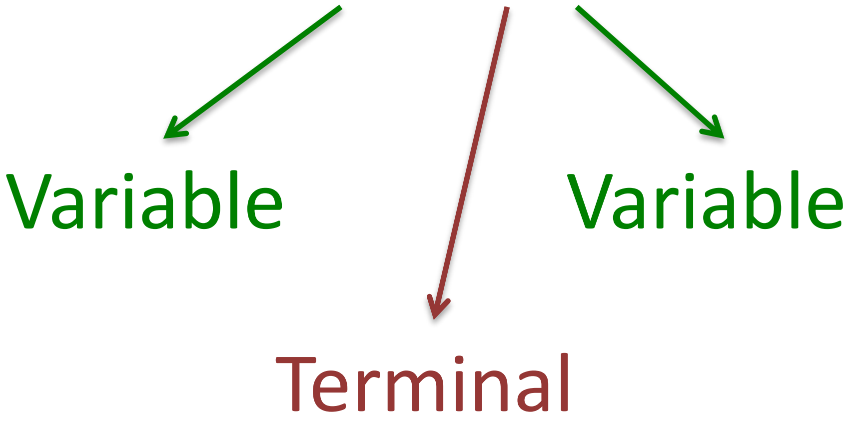
Let M be the NFA with L = L(M), we construct a regular grammar G such that L(M) = L(G)

Proof – Part 2
NFA to right-linear grammar

- For any transition



Add production: q → ap



Proof – Part 2
NFA to right-linear grammar

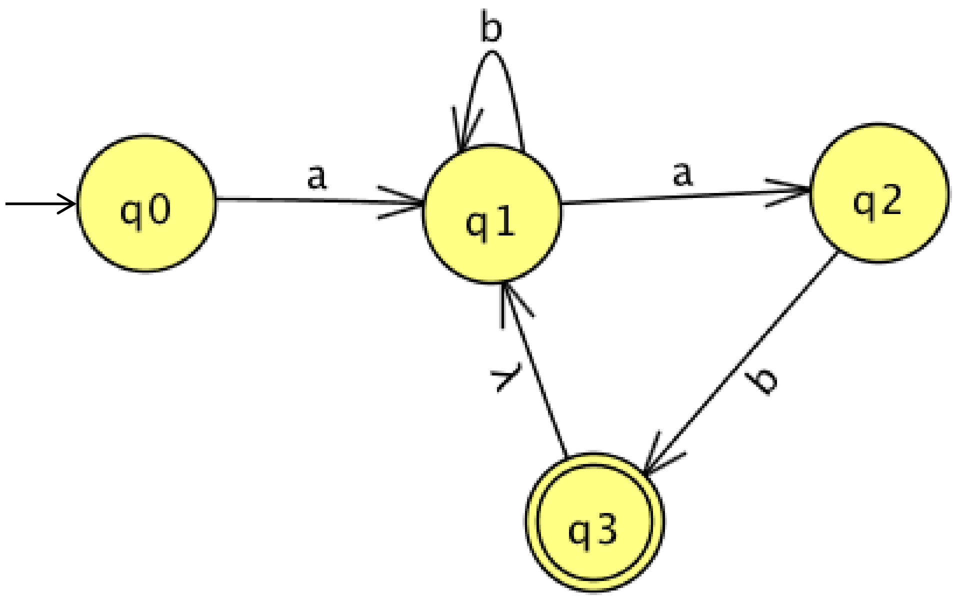
- The starting state is your start symbol

- For any final state

Add production: $q_f \rightarrow \lambda$

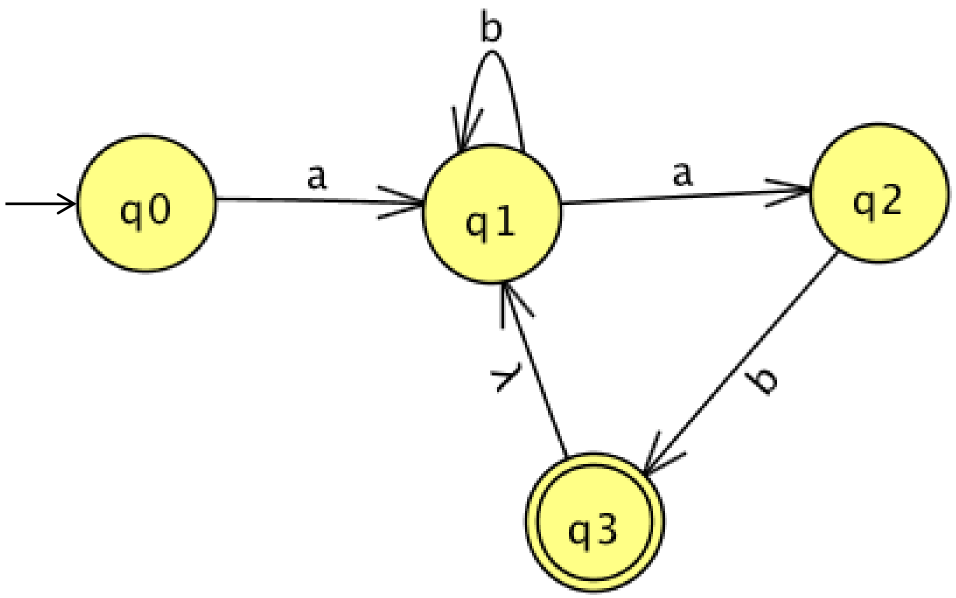
NFA to right-linear grammar Example

$q_0 \rightarrow aq_1$



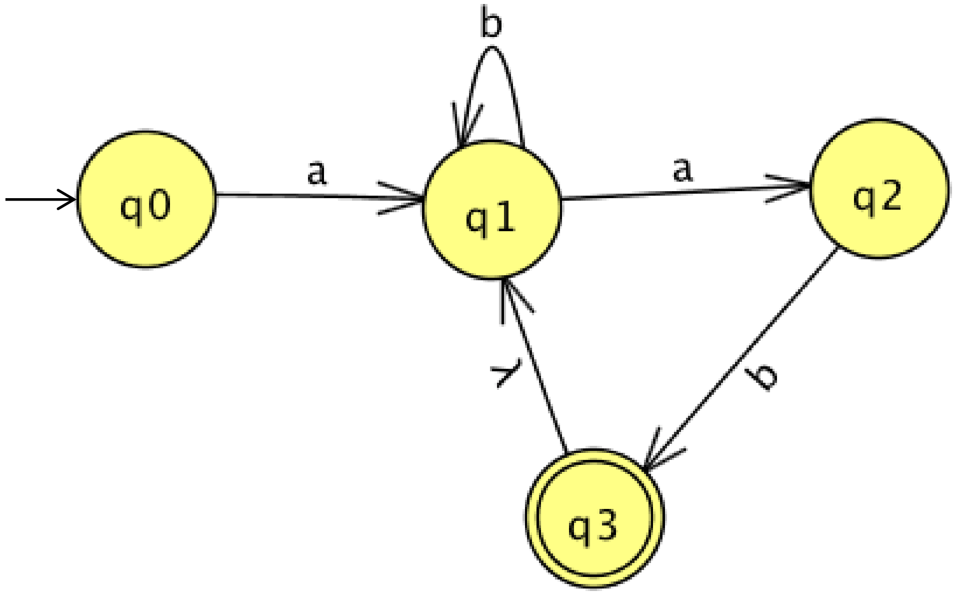
NFA to right-linear grammar Example

$q_0 \rightarrow aq_1$
$q_1 \rightarrow aq_2$
$q_1 \rightarrow bq_1$



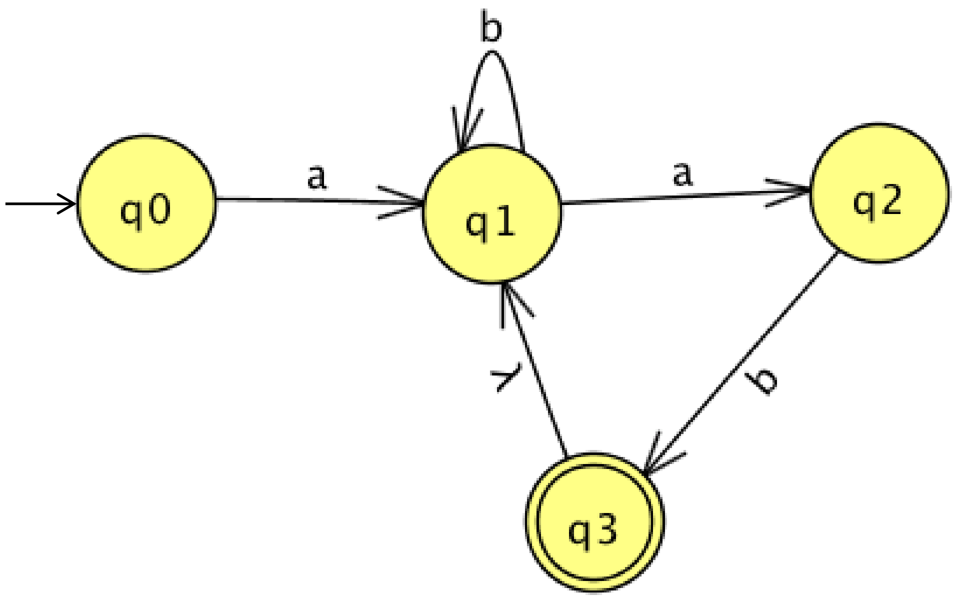
NFA to right-linear grammar Example

$q_0 \rightarrow aq_1$
$q_1 \rightarrow aq_2$
$q_1 \rightarrow bq_1$
$q_2 \rightarrow bq_3$



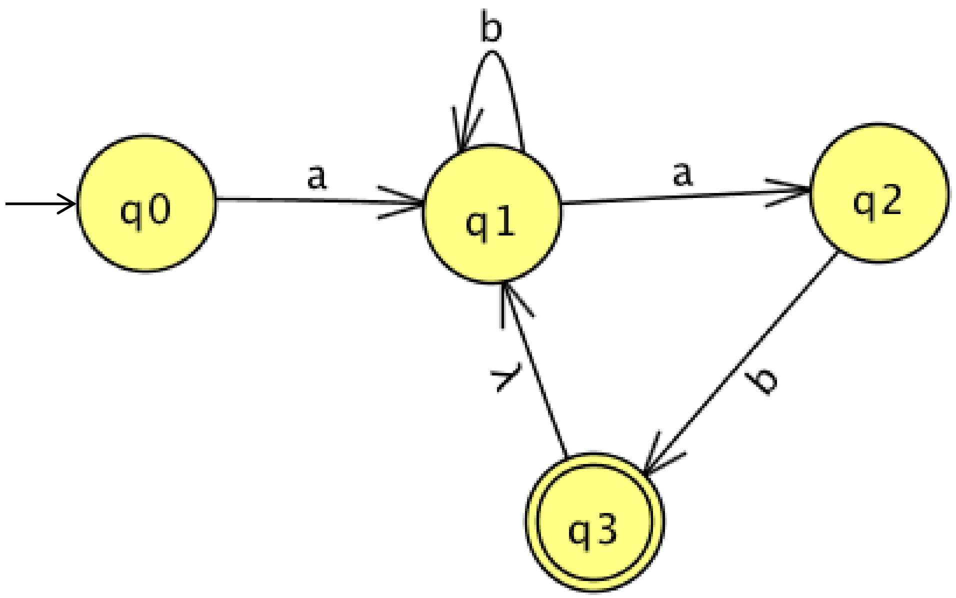
NFA to right-linear grammar Example

$q_0 \rightarrow aq_1$
$q_1 \rightarrow aq_2$
$q_1 \rightarrow bq_1$
$q_2 \rightarrow bq_3$
$q_3 \rightarrow q_1$



NFA to right-linear grammar Example

$q_0 \rightarrow aq_1$
$q_1 \rightarrow aq_2$
$q_1 \rightarrow bq_1$
$q_2 \rightarrow bq_3$
$q_3 \rightarrow q_1$
$q_3 \rightarrow \lambda$



Summary
Regular Languages