Reverse of a Regular Language

Theorem:

The reverse L^R of a regular language L is a regular language

Proof idea:

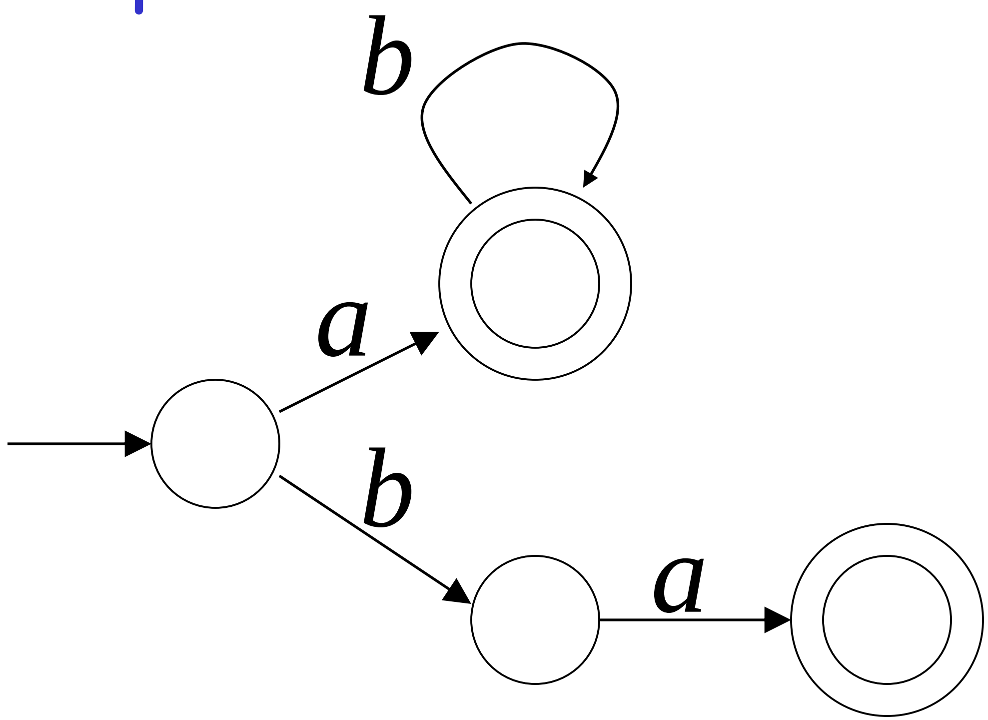
Construct NFA that accepts L^R :

invert the transitions of the NFA that accepts L

Proof

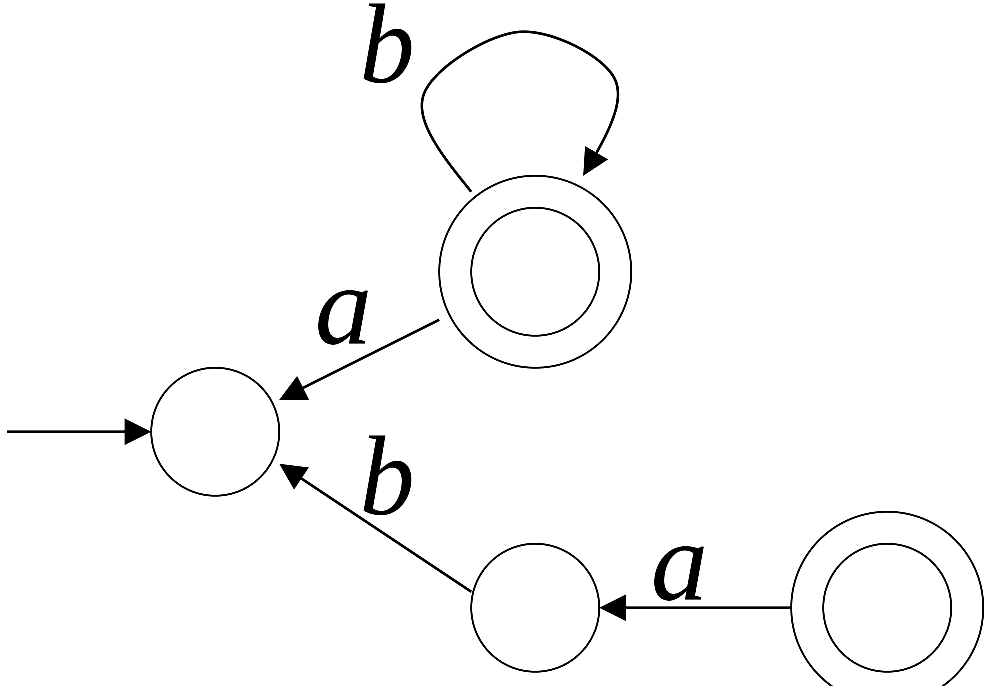
Since L is regular,
there is NFA that accepts L

Example:

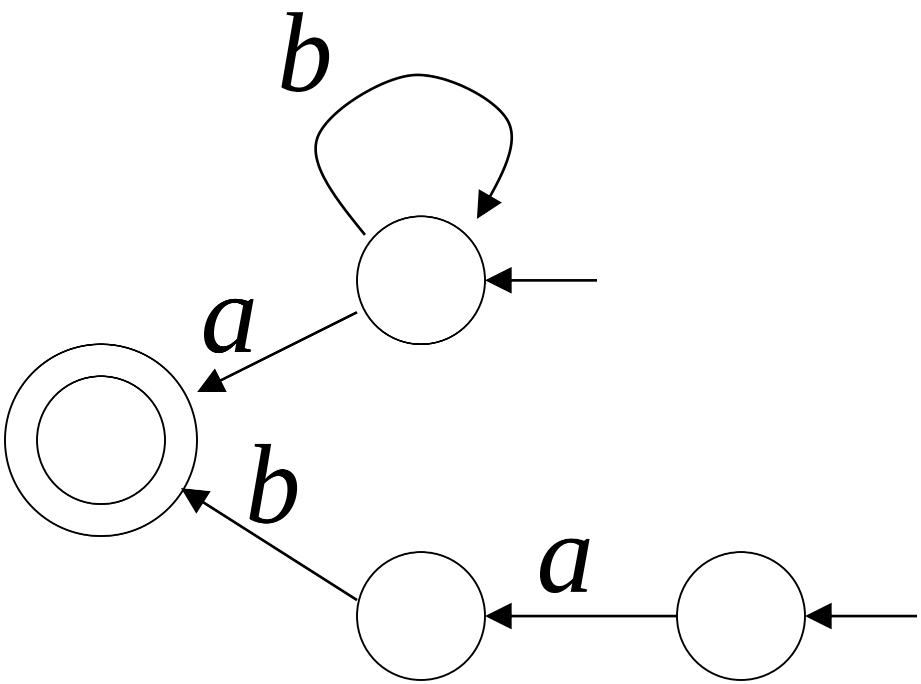


$$L = ab^* + ba$$

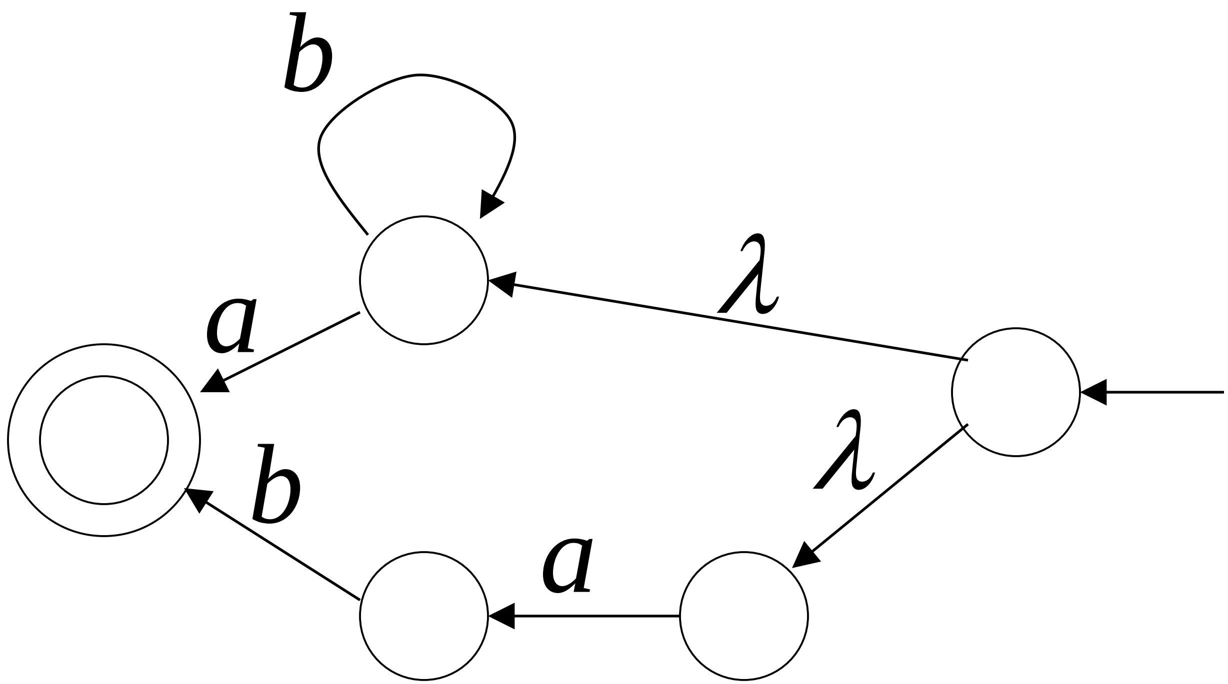
Invert Transitions



Make old initial state a final state



Add a new initial state

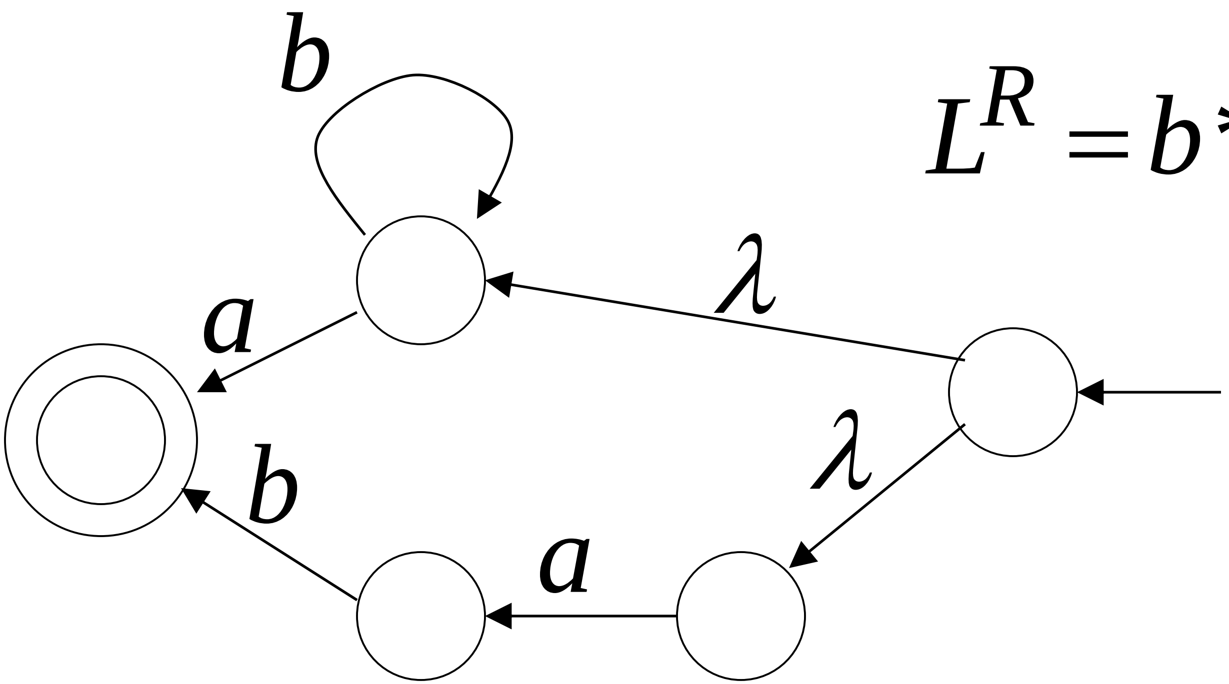


Resulting machine accepts L^R

L^R is regular

$$L = ab^* + ba$$

$$L^R = b^*a + ab$$



Grammars

Grammars

Grammars express languages

Example: the English language

$$\langle sentence \rangle \rightarrow \langle noun\_phrase \rangle \; \langle predicate \rangle$$

$$\langle noun\_phrase \rangle \rightarrow \langle article \rangle \; \langle noun \rangle$$

$$\langle predicate \rangle \rightarrow \langle verb \rangle$$

$$\langle article \rangle \rightarrow a$$

$$\langle article \rangle \rightarrow the$$

$$\langle noun \rangle \rightarrow boy$$

$$\langle noun \rangle \rightarrow dog$$

$$\langle verb \rangle \rightarrow runs$$

$$\langle verb \rangle \rightarrow walks$$

A derivation of "the boy walks":

$$\begin{aligned}\langle sentence \rangle &\Rightarrow \langle noun\_phrase \rangle \ \langle predicate \rangle \\ &\Rightarrow \langle noun\_phrase \rangle \ \langle verb \rangle \\ &\Rightarrow \langle article \rangle \ \langle noun \rangle \ \langle verb \rangle \\ &\Rightarrow the \ \langle noun \rangle \ \langle verb \rangle \\ &\Rightarrow the \ boy \ \langle verb \rangle \\ &\Rightarrow the \ boy \ walks\end{aligned}$$

A derivation of "a dog runs":

$$
\begin{aligned}
\langle sentence \rangle &\Rightarrow \langle noun\_phrase \rangle \ \langle predicate \rangle \\
&\Rightarrow \langle noun\_phrase \rangle \ \langle verb \rangle \\
&\Rightarrow \langle article \rangle \ \langle noun \rangle \ \langle verb \rangle \\
&\Rightarrow a \ \langle noun \rangle \ \langle verb \rangle \\
&\Rightarrow a \ dog \ \langle verb \rangle \\
&\Rightarrow a \ dog \ runs
\end{aligned}
$$

Language of the grammar:

L = { "a boy runs",
"a boy walks",
"the boy runs",
"the boy walks",
"a dog runs",
"a dog walks",
"the dog runs",
"the dog walks" }

Notation

$$\langle noun \rangle \to boy$$

$$\langle noun \rangle \to dog$$

Variable
or
Non-terminal

Production
rule

Terminal

Another Example

Grammar: $S \rightarrow aSb$

$S \rightarrow \lambda$

Derivation of sentence ab:

$$S \Rightarrow aSb \Rightarrow ab$$

$$S \rightarrow aSb \qquad\qquad S \rightarrow \lambda$$

Grammar: $S \to aSb$

$S \to \lambda$

Derivation of sentence $aabb$:

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$$

$S \to aSb$ $\qquad$ $S \to \lambda$

Other derivations:

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabbb$$

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb$$

$$\Rightarrow aaaaSbbbb \Rightarrow aaaabbbb$$

Language of the grammar

$$S \to aSb$$

$$S \to \lambda$$

$$L = \{a^n b^n : n \geq 0\}$$

More Notation

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

$V:$ Set of variables

$T:$ Set of terminal symbols

$S:$ Start variable

$P:$ Set of Production rules

Example

Grammar G :

$$S \to aSb$$

$$S \to \lambda$$

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

$$V = \{S\} \qquad T = \{a, b\}$$

$$P = \{S \to aSb,\ S \to \lambda\}$$

More Notation

Sentential Form:

A sentence that contains variables and terminals

Example:

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabbb$$

Sentential Forms　　　　sentence

We write: $S \overset{*}{\Rightarrow} aaabbb$

Instead of:

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabbb$$

In general we write: $w_1 \overset{*}{\Rightarrow} w_n$

If: $w_1 \Rightarrow w_2 \Rightarrow w_3 \Rightarrow \cdots \Rightarrow w_n$

By default: $w \overset{*}{\Rightarrow} w$

Example

Grammar

$$S \to aSb$$

$$S \to \lambda$$

Derivations

$$S \overset{*}{\Rightarrow} \lambda$$

$$S \overset{*}{\Rightarrow} ab$$

$$S \overset{*}{\Rightarrow} aabb$$

$$S \overset{*}{\Rightarrow} aaabbb$$

Example

Grammar

$$S \rightarrow aSb$$

$$S \rightarrow \lambda$$

Derivations

$$S \overset{*}{\Rightarrow} aaSbb$$

$$aaSbb \overset{*}{\Rightarrow} aaaaaSbbbbb$$

Another Grammar Example

Grammar G:

$$S \to Ab$$

$$A \to aAb$$

$$A \to \lambda$$

Derivations:

$$S \Rightarrow Ab \Rightarrow b$$

$$S \Rightarrow Ab \Rightarrow aAbb \Rightarrow abb$$

$$S \Rightarrow Ab \Rightarrow aAbb \Rightarrow aaAbbbb \Rightarrow aabbbb$$

More Derivations

$$S \Rightarrow Ab \Rightarrow aAbb \Rightarrow aaAbbb \Rightarrow aaaAbbbb$$

$$\Rightarrow aaaaAbbbbb \Rightarrow aaaabbbbb$$

$$S \stackrel{*}{\Rightarrow} aaaabbbbb$$

$$S \stackrel{*}{\Rightarrow} aaaaaabbbbbbb$$

$$S \stackrel{*}{\Rightarrow} a^n b^n b$$

Language of a Grammar

For a grammar G
with start variable S :

$$L(G) = \{w: \quad S \overset{*}{\Rightarrow} w\}$$

String of terminals

Example

For grammar G :

$$S \to Ab$$

$$A \to aAb$$

$$A \to \lambda$$

$$L(G) = \{a^n b^n b : \quad n \geq 0\}$$

Since: $S \overset{*}{\Rightarrow} a^n b^n b$

A Convenient Notation

$$A \to aAb$$
$$A \to \lambda$$

$$\Longrightarrow \quad A \to aAb \mid \lambda$$

$$\langle article \rangle \to a$$
$$\langle article \rangle \to the$$

$$\Longrightarrow \quad \langle article \rangle \to a \mid the$$

Linear Grammars

Linear Grammars

Grammars with
at most one variable at the right side
of a production

Examples:

$$S \rightarrow aSb$$
$$S \rightarrow \lambda$$

$$S \rightarrow Ab$$
$$A \rightarrow aAb$$
$$A \rightarrow \lambda$$

A Non-Linear Grammar

Grammar G:

$$S \to SS$$

$$S \to \lambda$$

$$S \to aSb$$

$$S \to bSa$$

$$L(G) = \{w: \quad n_a(w) = n_b(w)\}$$

Number of a in string w

Another Linear Grammar

Grammar G :

$$S \to A$$

$$A \to aB \mid \lambda$$

$$B \to Ab$$

$$L(G) = \{a^n b^n : n \geq 0\}$$

Right-Linear Grammars

All productions have form: $A \to xB$

or

$A \to x$

string of terminals

Example: $S \to abS$

$S \to a$

Left-Linear Grammars

All productions have form: $A \to Bx$

or

$A \to x$

string of terminals

Example:

$$S \to Aab$$

$$A \to Aab \mid B$$

$$B \to a$$

Regular Grammars

Regular Grammars

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

Examples:

G_1

$$S \to abS$$
$$S \to a$$

G_2

$$S \to Aab$$
$$A \to Aab \mid B$$
$$B \to a$$

Observation

Regular grammars generate regular languages

Examples:

G_1

$S \to abS$

$S \to a$

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

G_2

$S \to Aab$

$A \to Aab \mid B$

$B \to a$

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

Regular Grammars Generate Regular Languages

Theorem

$$\left\{ \begin{array}{l} \text{Languages} \\ \text{Generated by} \\ \text{Regular Grammars} \end{array} \right\} = \left\{ \begin{array}{l} \text{Regular} \\ \text{Languages} \end{array} \right\}$$

Theorem - Part 1

$$\left\{ \begin{array}{l} \text{Languages} \\ \text{Generated by} \\ \text{Regular Grammars} \end{array} \right\} \subseteq \left\{ \begin{array}{l} \text{Regular} \\ \text{Languages} \end{array} \right\}$$

Any regular grammar generates a regular language

Theorem - Part 2

$$\left\{ \begin{array}{l} \text{Languages} \\ \text{Generated by} \\ \text{Regular Grammars} \end{array} \right\} \supseteq \left\{ \begin{array}{l} \text{Regular} \\ \text{Languages} \end{array} \right\}$$

Any regular language is generated by a regular grammar

Proof - Part 1

$$\left\{\begin{array}{l}\text{Languages}\\ \text{Generated by}\\ \text{Regular Grammars}\end{array}\right\} \subseteq \left\{\begin{array}{l}\text{Regular}\\ \text{Languages}\end{array}\right\}$$

The language $L(G)$ generated by any regular grammar G is regular

The case of Right-Linear Grammars

Let G be a right-linear grammar

We will prove: $L(G)$ is regular

Proof idea: We will construct NFA M with $L(M) = L(G)$

Grammar G is right-linear

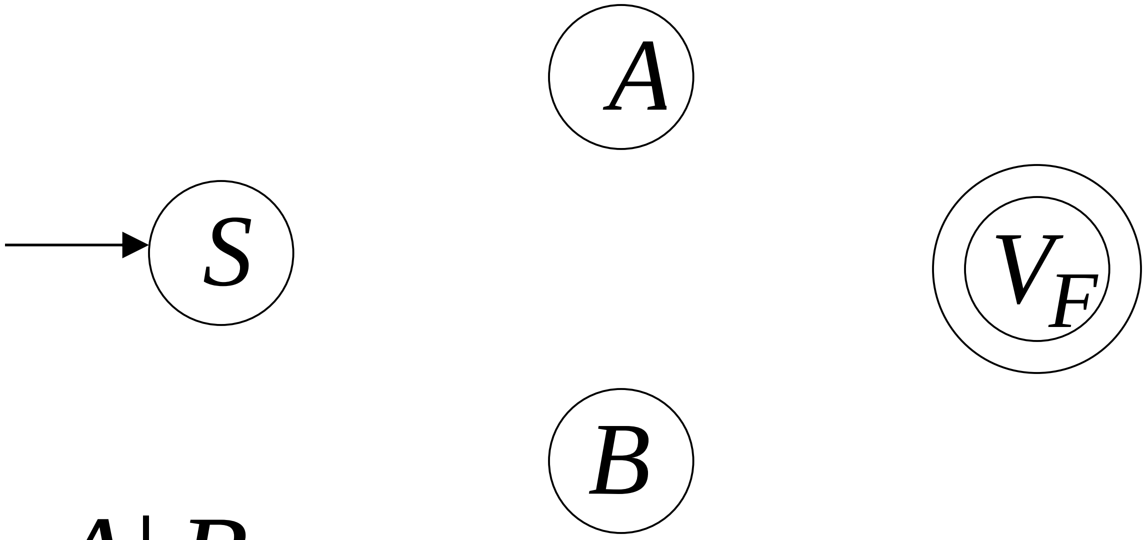
Example:

$$S \to aA \mid B$$

$$A \to aa\,B$$

$$B \to b\,B \mid a$$

Construct NFA M such that every state is a grammar variable:



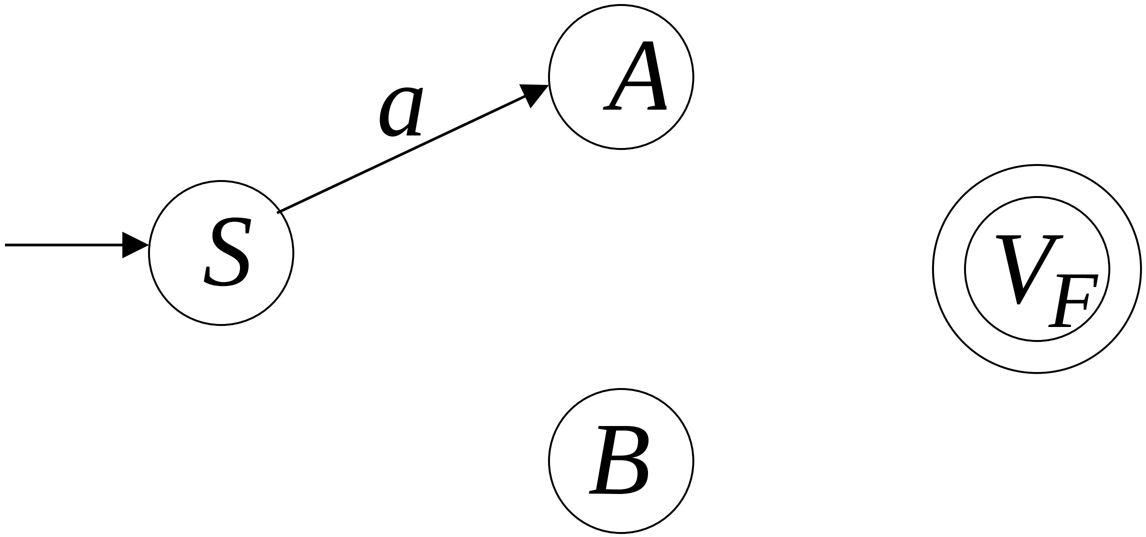
special final state

$$S \to aA \mid B$$

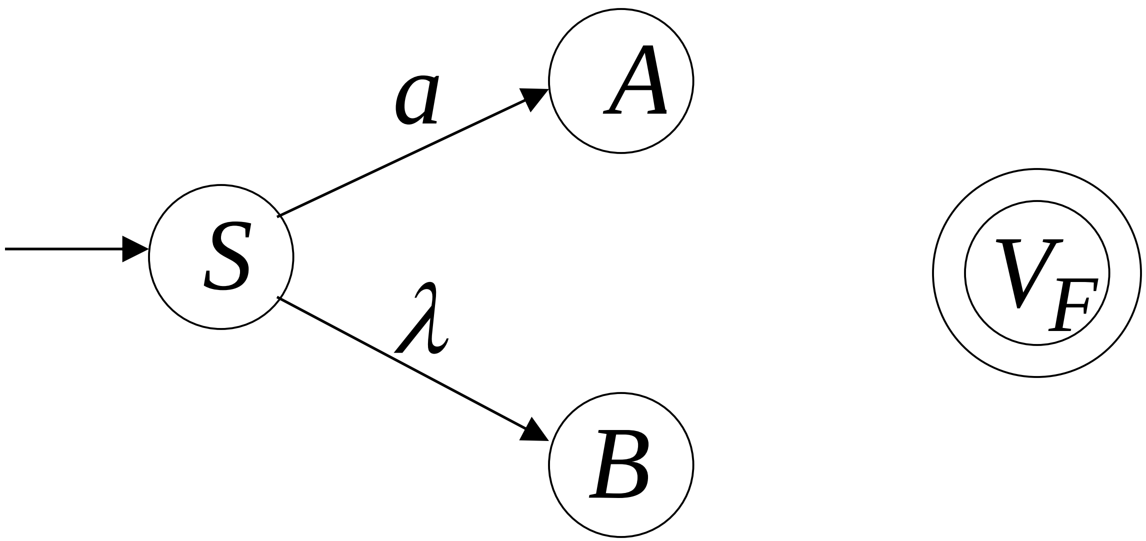
$$A \to aa\,B$$

$$B \to b\,B \mid a$$

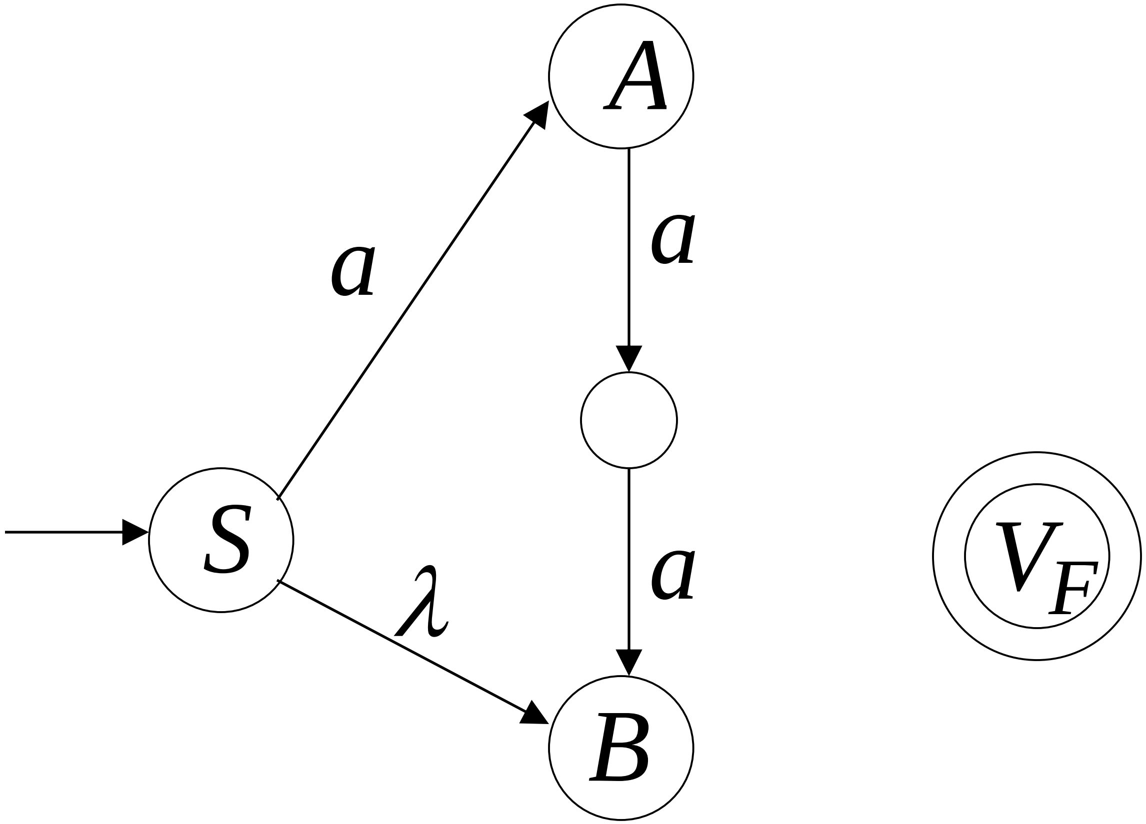
Add edges for each production:



$$S \rightarrow aA$$

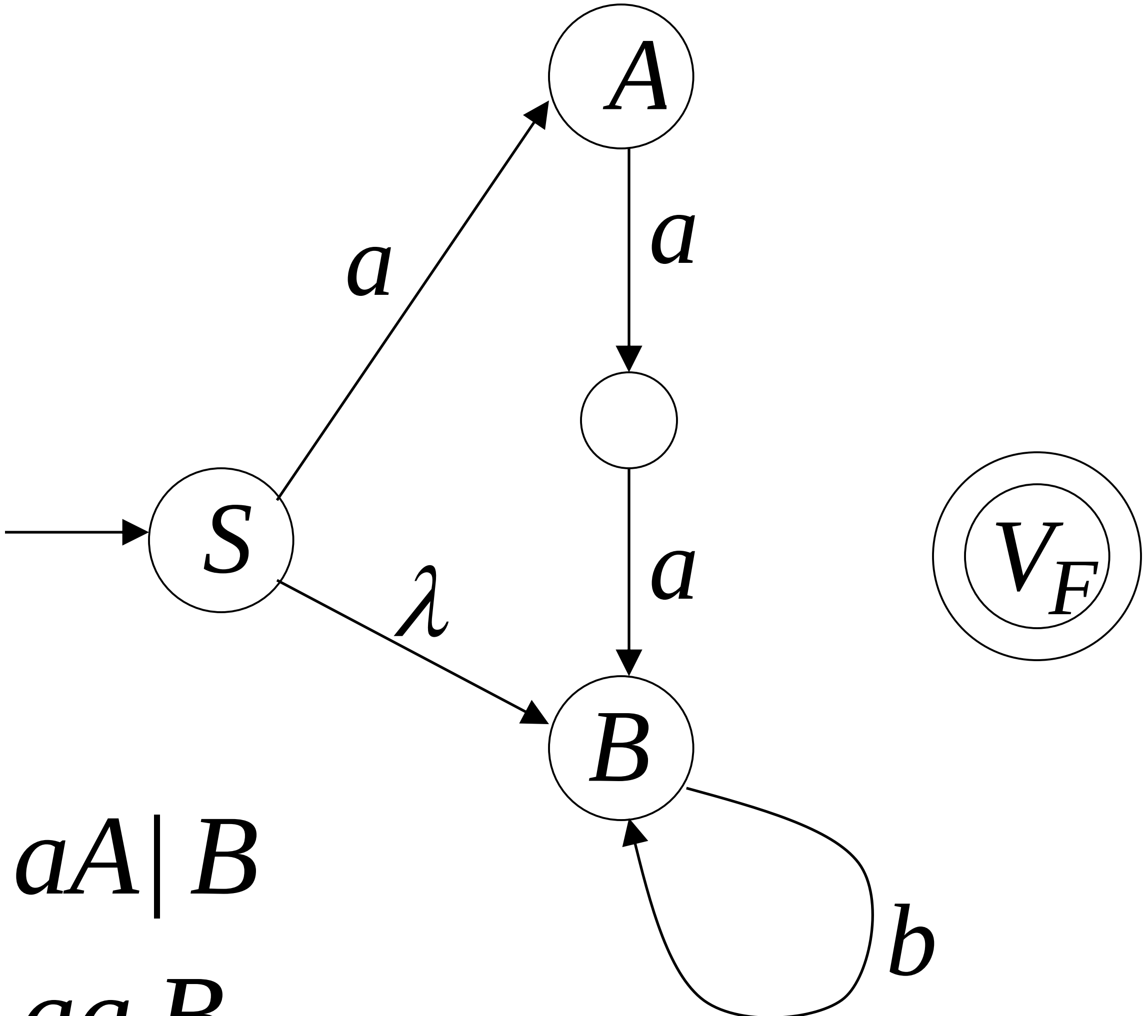


$$S \rightarrow aA \mid B$$



$$S \to aA \mid B$$

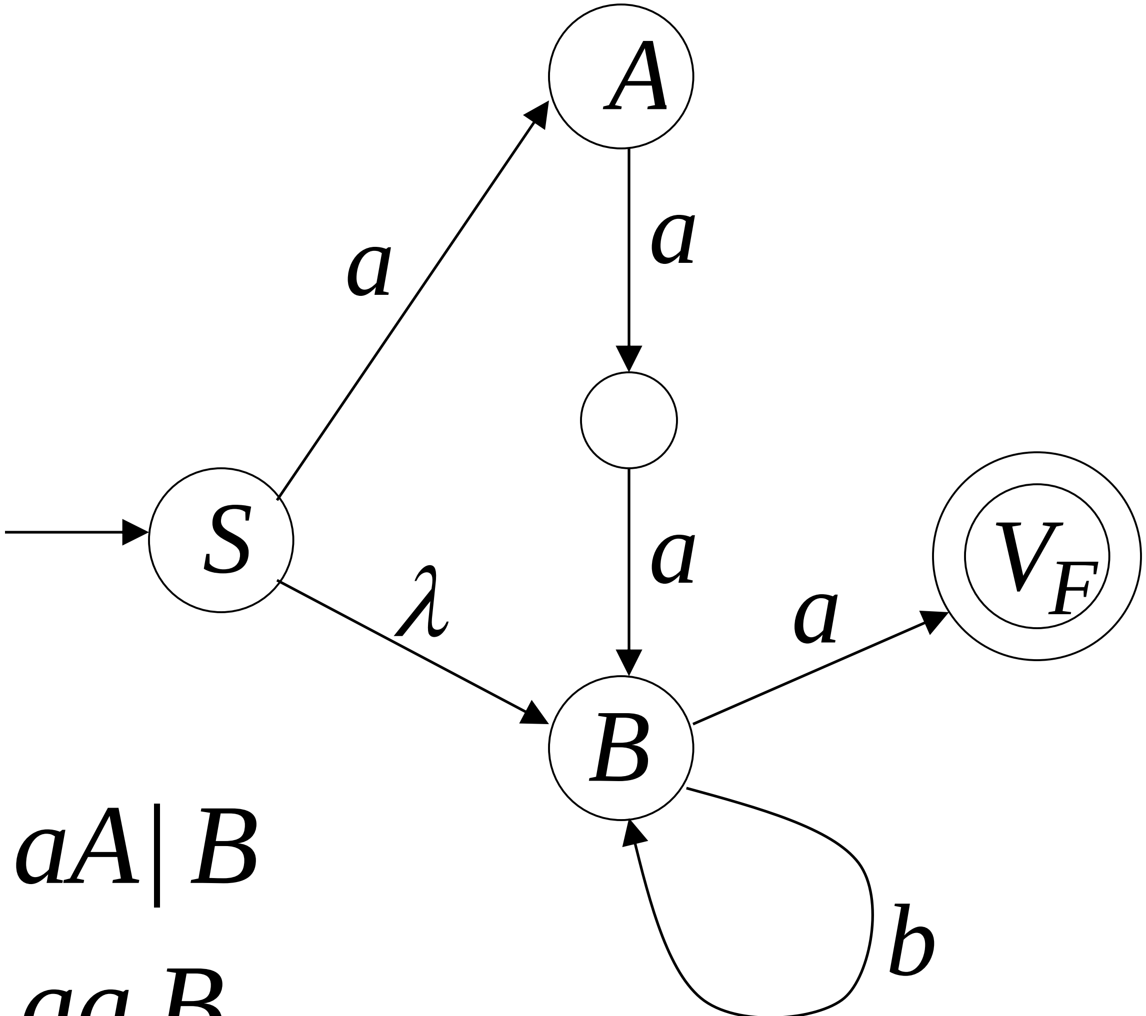
$$A \to aa\,B$$



$$S \to aA \mid B$$

$$A \to aa\,B$$

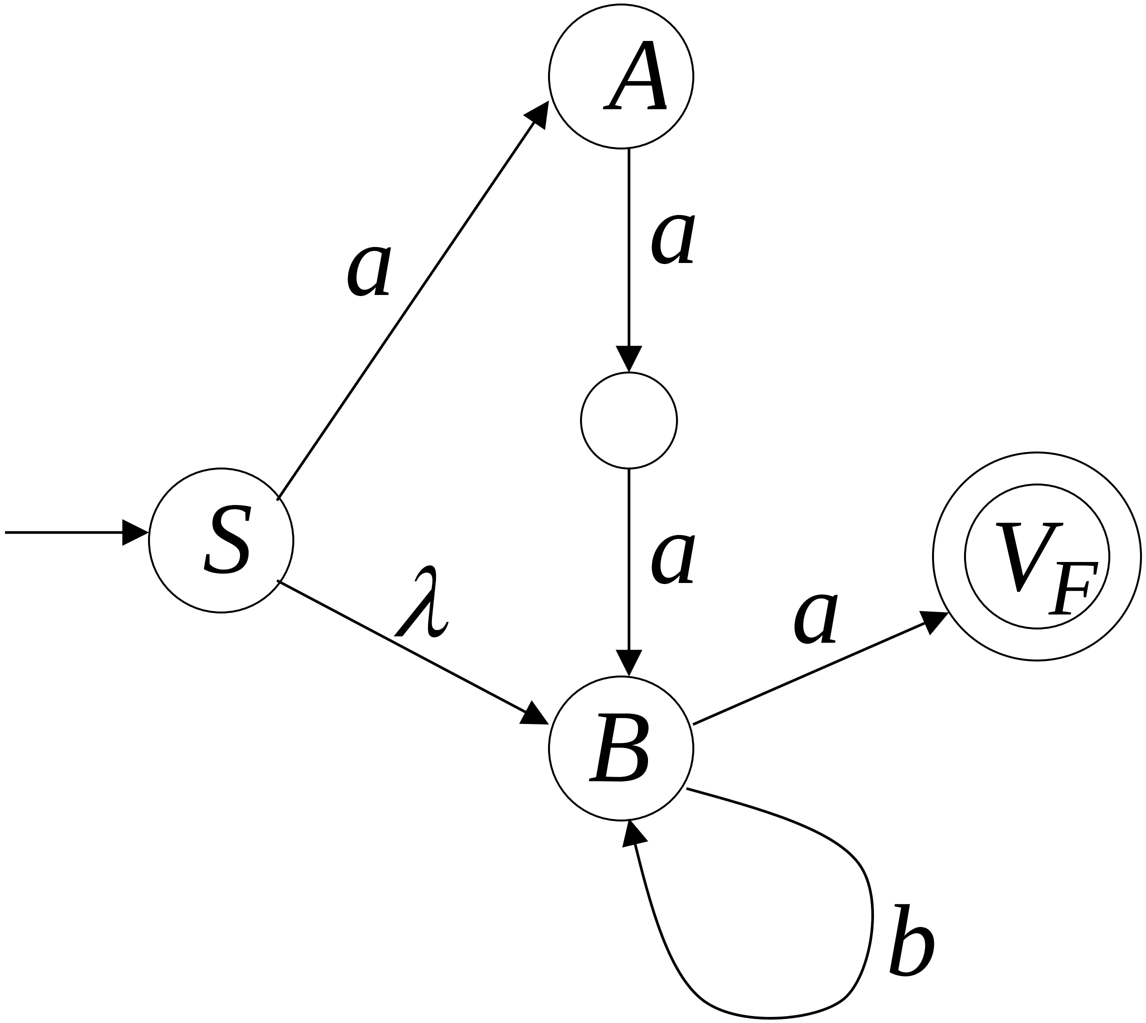
$$B \to bB$$



$$S \to aA \mid B$$

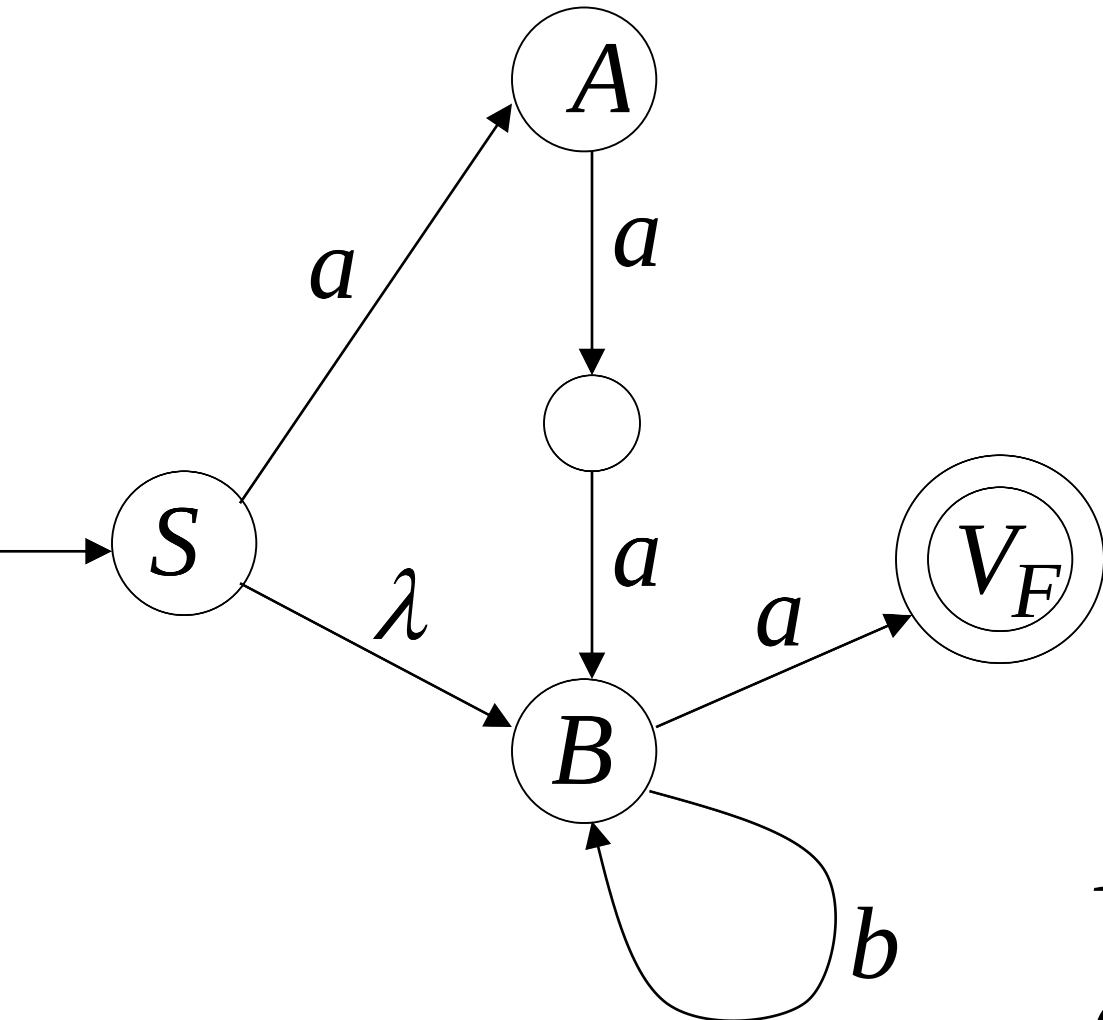
$$A \to aa\,B$$

$$B \to bB \mid a$$



$$S \Rightarrow aA \Rightarrow aaaB \Rightarrow aaabB \Rightarrow aaabc$$

NFA M



Grammar G

$$S \to aA \mid B$$

$$A \to aa\,B$$

$$B \to bB \mid a$$

$$L(M) = L(G) =$$
$$aaab^*a + b^*a$$

In General

A right-linear grammar G

has variables: $V_0, V_1, V_2, \ldots$

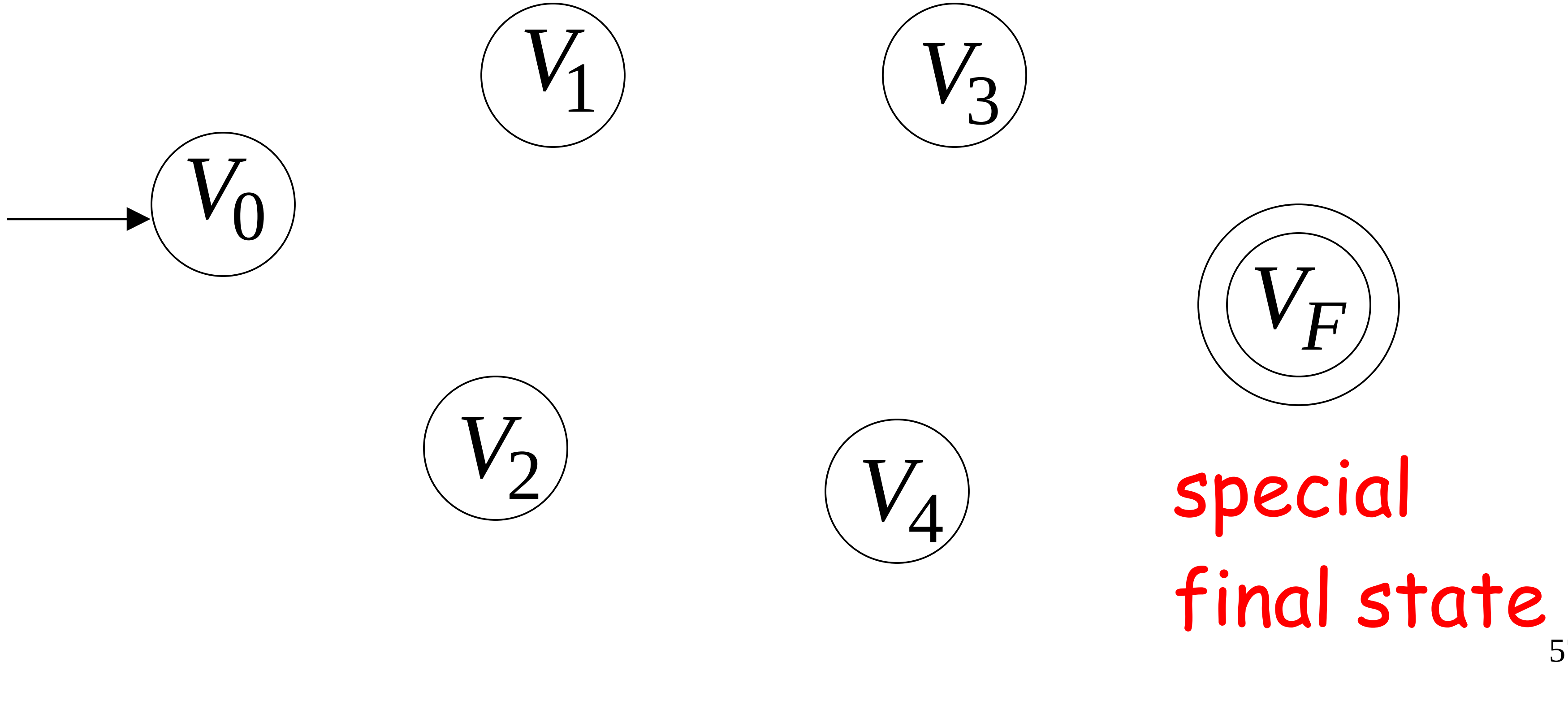
and productions: $V_i \rightarrow a_1 a_2 \cdots a_m V_j$

or

$$V_i \rightarrow a_1 a_2 \cdots a_m$$

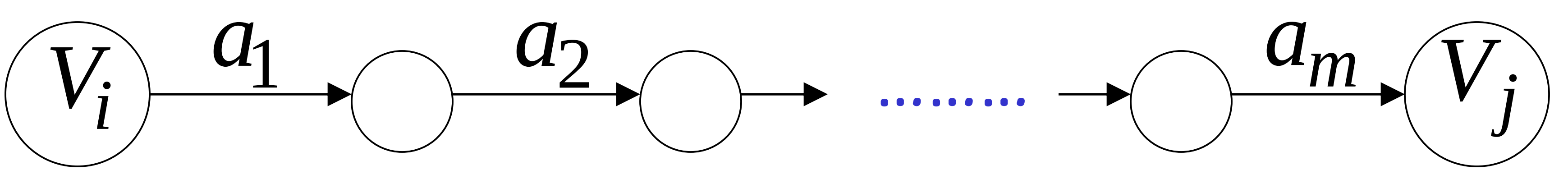
We construct the NFA M such that:

each variable V_i corresponds to a node:



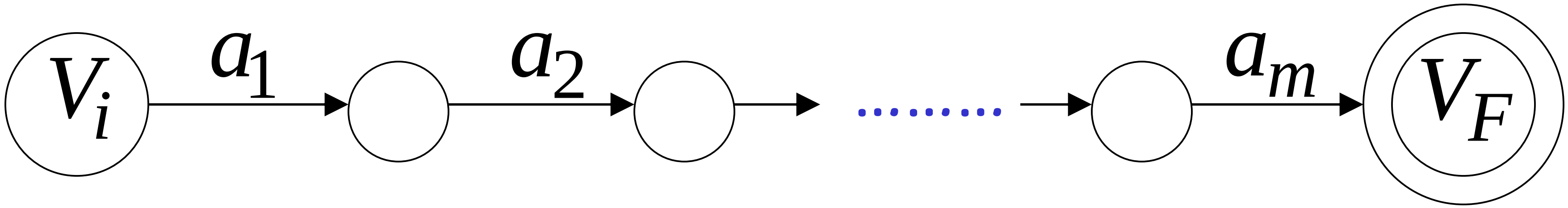
For each production: $V_i \to a_1 a_2 \cdots a_m V_j$

we add transitions and intermediate nodes

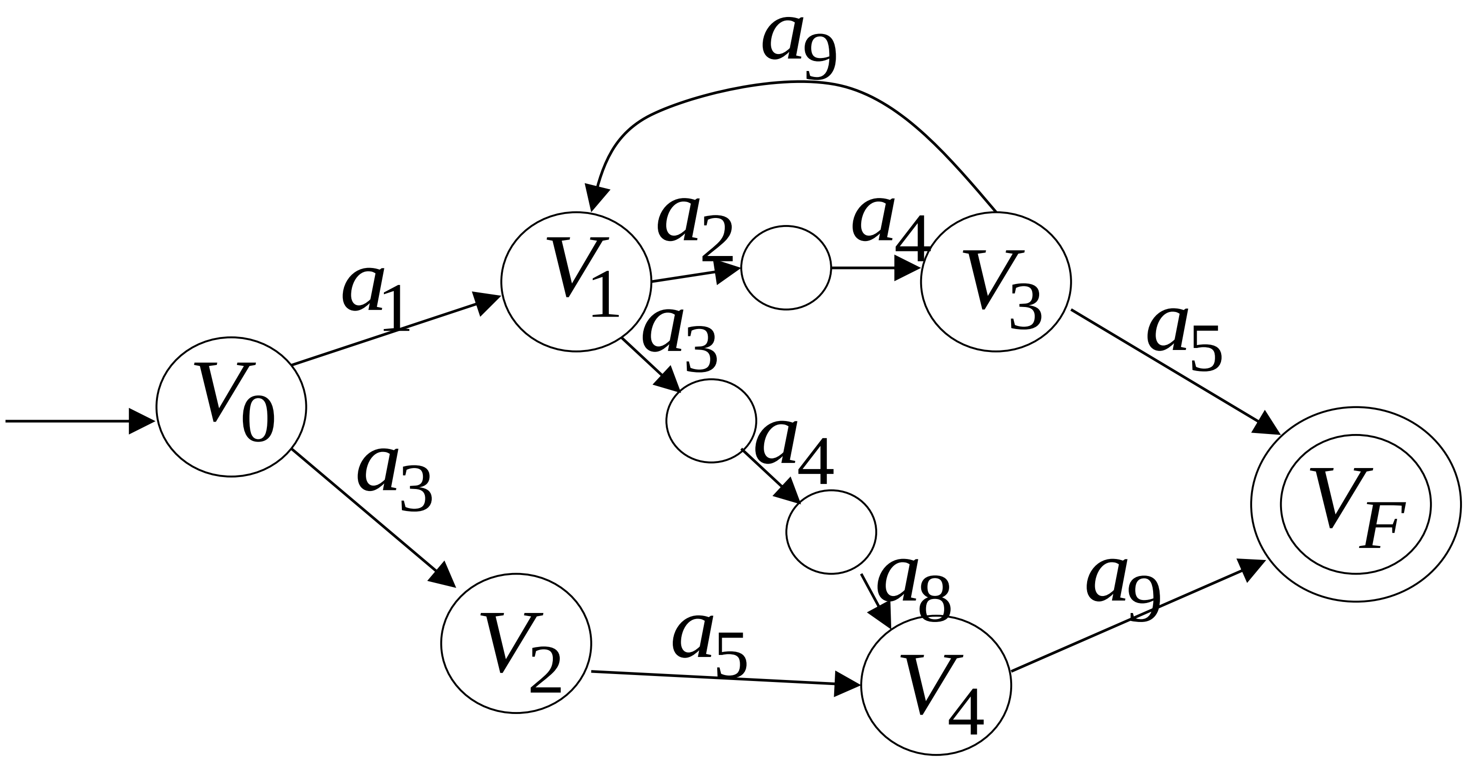


For each production: $V_i \to a_1 a_2 \cdots a_m$

we add transitions and intermediate nodes



Resulting NFA M looks like this:



It holds that: $L(G) = L(M)$

The case of Left-Linear Grammars

Let G be a left-linear grammar

We will prove: $L(G)$ is regular

Proof idea:

We will construct a right-linear grammar G' with $L(G) = L(G')^R$

Since G is left-linear grammar the productions look like:

$$A \to B a_1 a_2 \cdots a_k$$

$$A \to a_1 a_2 \cdots a_k$$

Construct right-linear grammar G'

Left linear G

$$A \to B a_1 a_2 \cdots a_k$$

$$A \to B v$$

Right linear G'

$$A \to a_k \cdots a_2 a_1 B$$

$$A \to v^R B$$

Construct right-linear grammar G'

Left linear G

$$A \rightarrow a_1 a_2 \cdots a_k$$

$$A \rightarrow v$$

$$\Downarrow$$

Right linear G'

$$A \rightarrow a_k \cdots a_2 a_1$$

$$A \rightarrow v^R$$

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

Since G' is right-linear, we have:

$L(G')$ $\Rightarrow$ $L(G')^R$ $\Rightarrow$ $L(G)$

Regular Language	Regular Language	Regular Language

Proof - Part 2

$$\left\{ \begin{array}{l} \text{Languages} \\ \text{Generated by} \\ \text{Regular Grammars} \end{array} \right\} \supseteq \left\{ \begin{array}{l} \text{Regular} \\ \text{Languages} \end{array} \right\}$$

Any regular language L is generated by some regular grammar G

Any regular language L is generated by some regular grammar G

Proof idea:

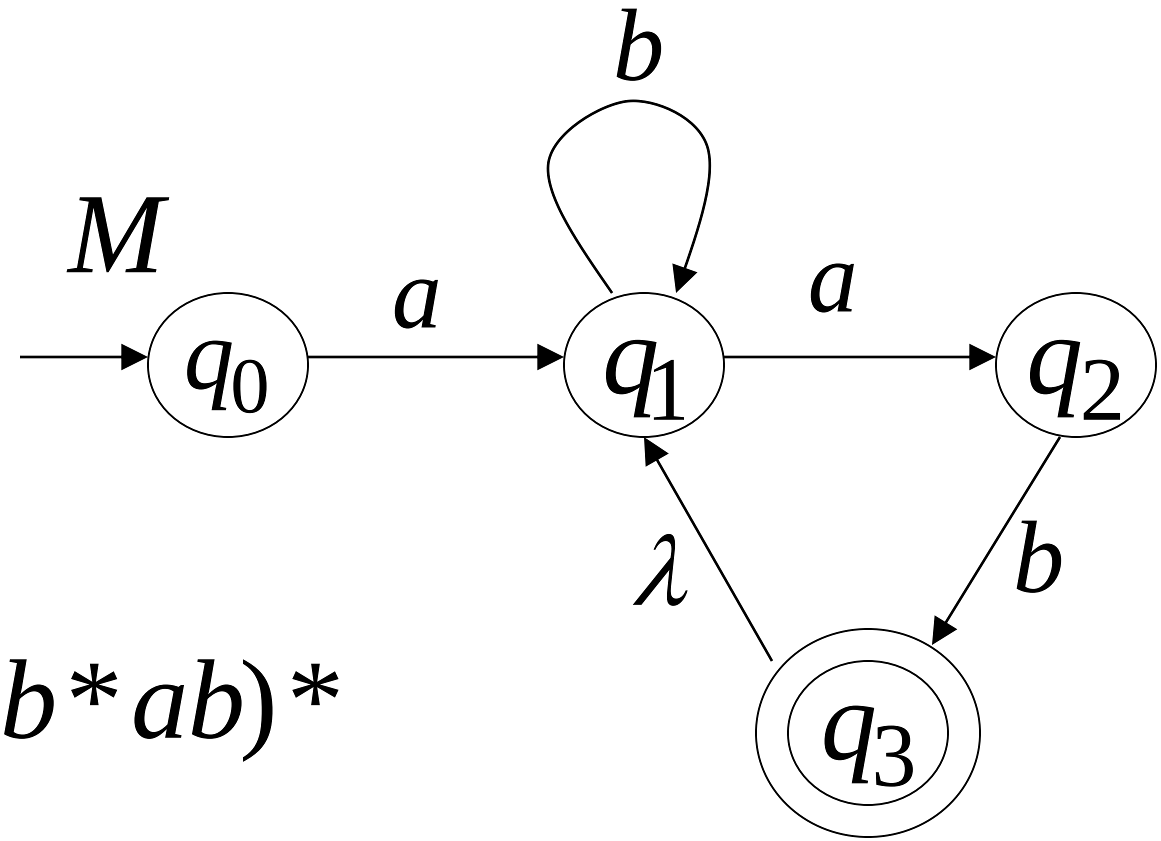
Let M be the NFA with $L = L(M)$.

Construct from M a regular grammar G such that $L(M) = L(G)$

Since L is regular

there is an NFA M such that $L = L(M)$

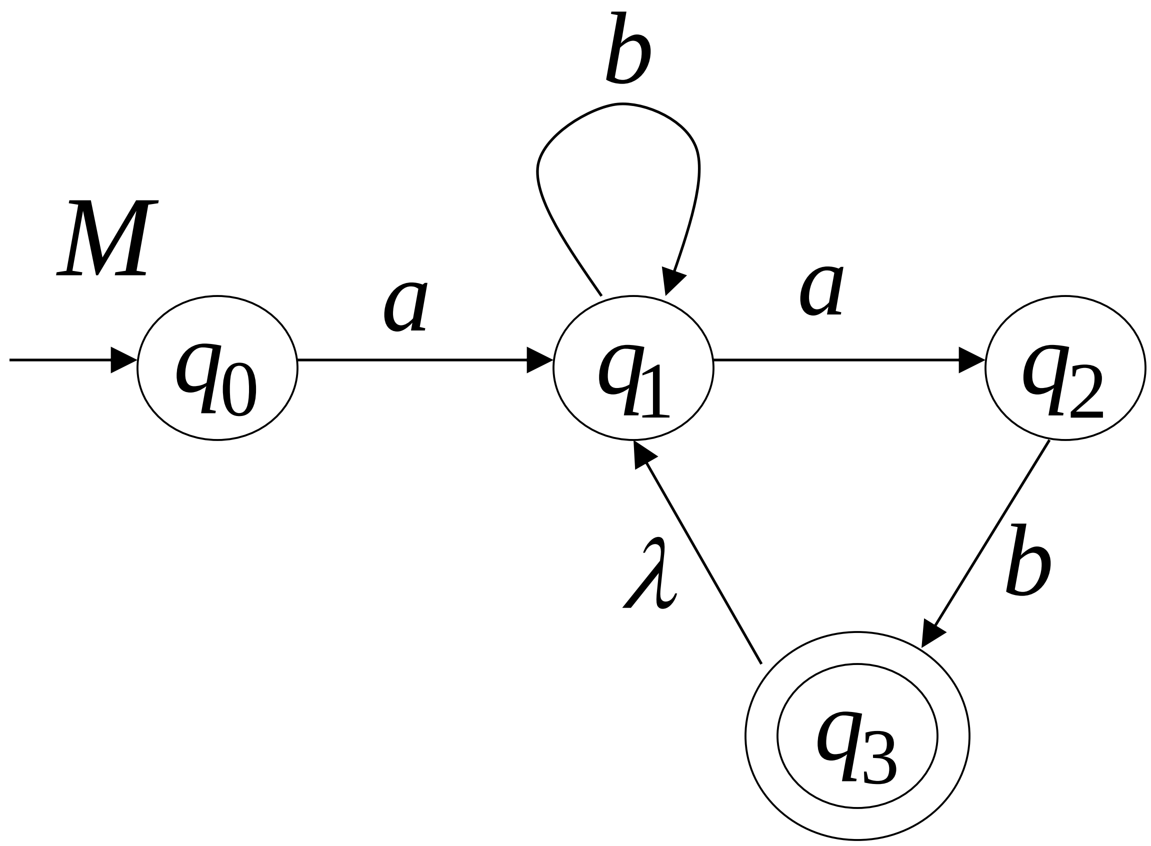
Example:



$$L = ab^*ab(b^*ab)^*$$

$$L = L(M)$$

Convert M to a right-linear grammar

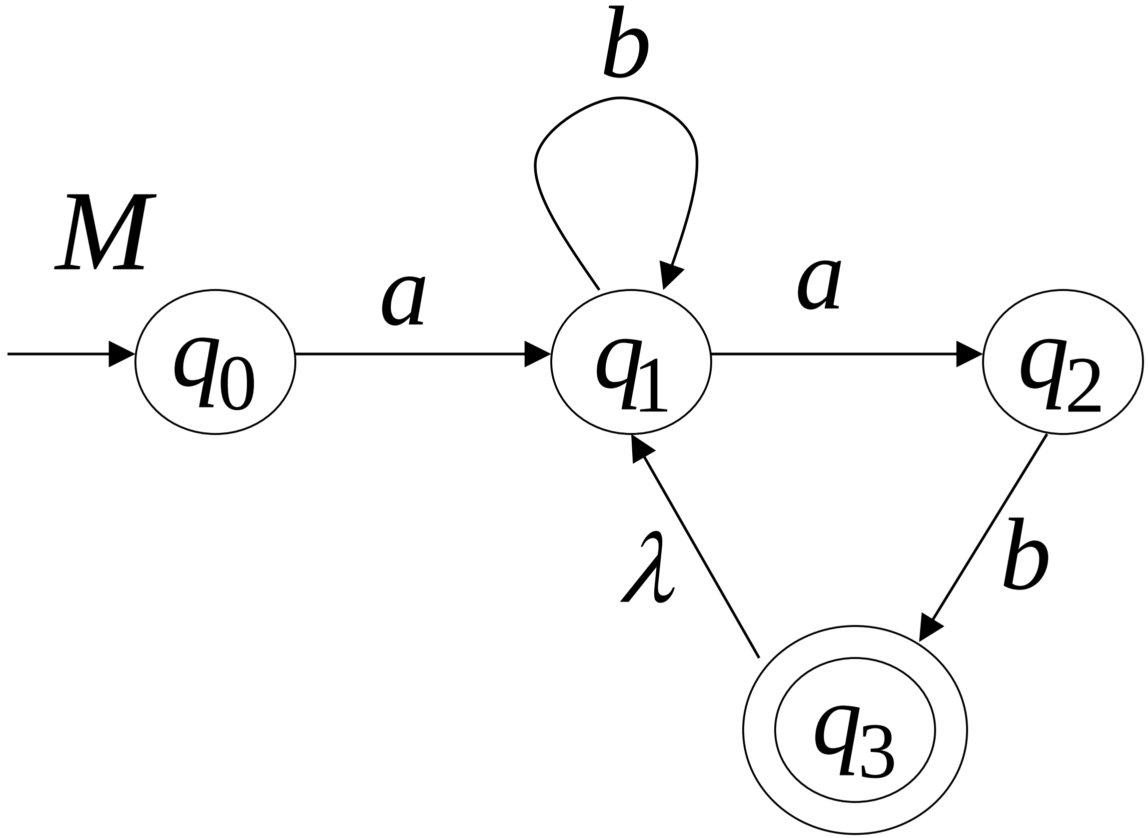


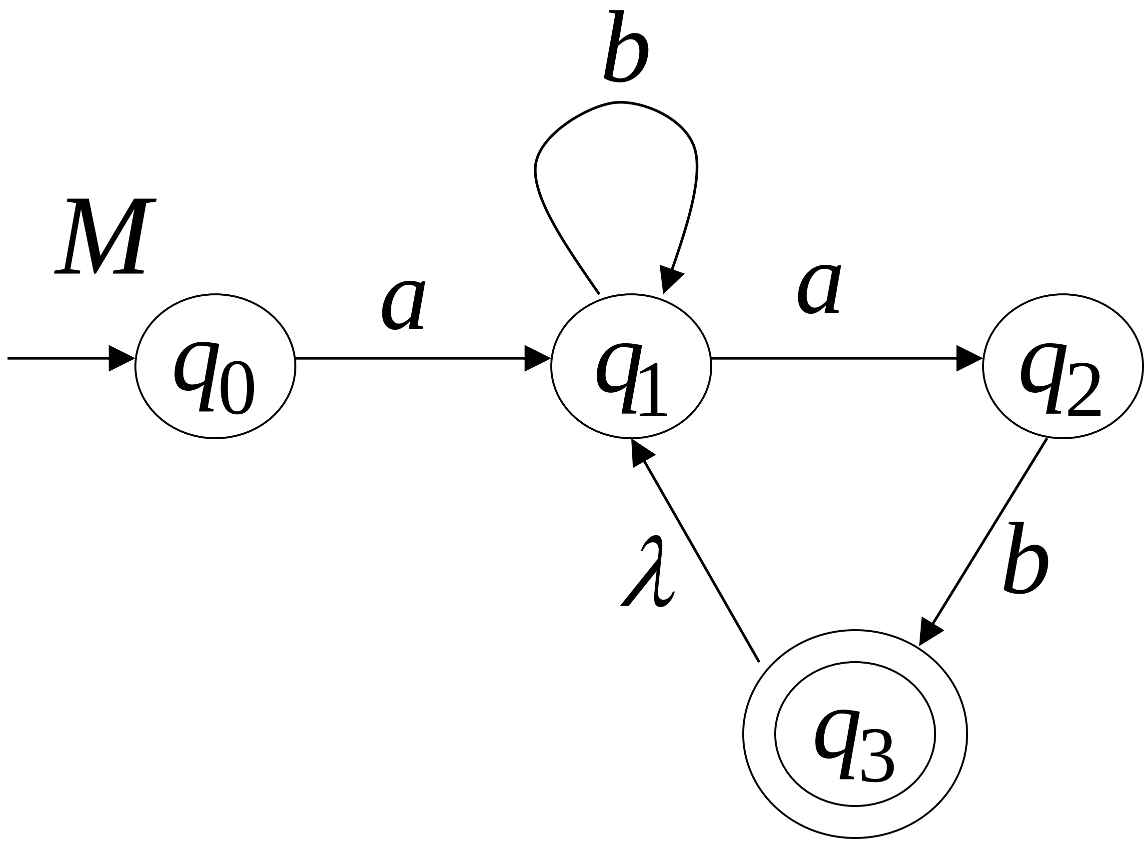
$$q_0 \to aq_1$$

$$q_0 \to aq_1$$

$$q_1 \to bq_1$$

$$q_1 \to aq_2$$





$$q_0 \to aq_1$$

$$q_1 \to bq_1$$

$$q_1 \to aq_2$$

$$q_2 \to bq_3$$

$$L(G) = L(M) = L$$

G

$$q_0 \to aq_1$$

$$q_1 \to bq_1$$

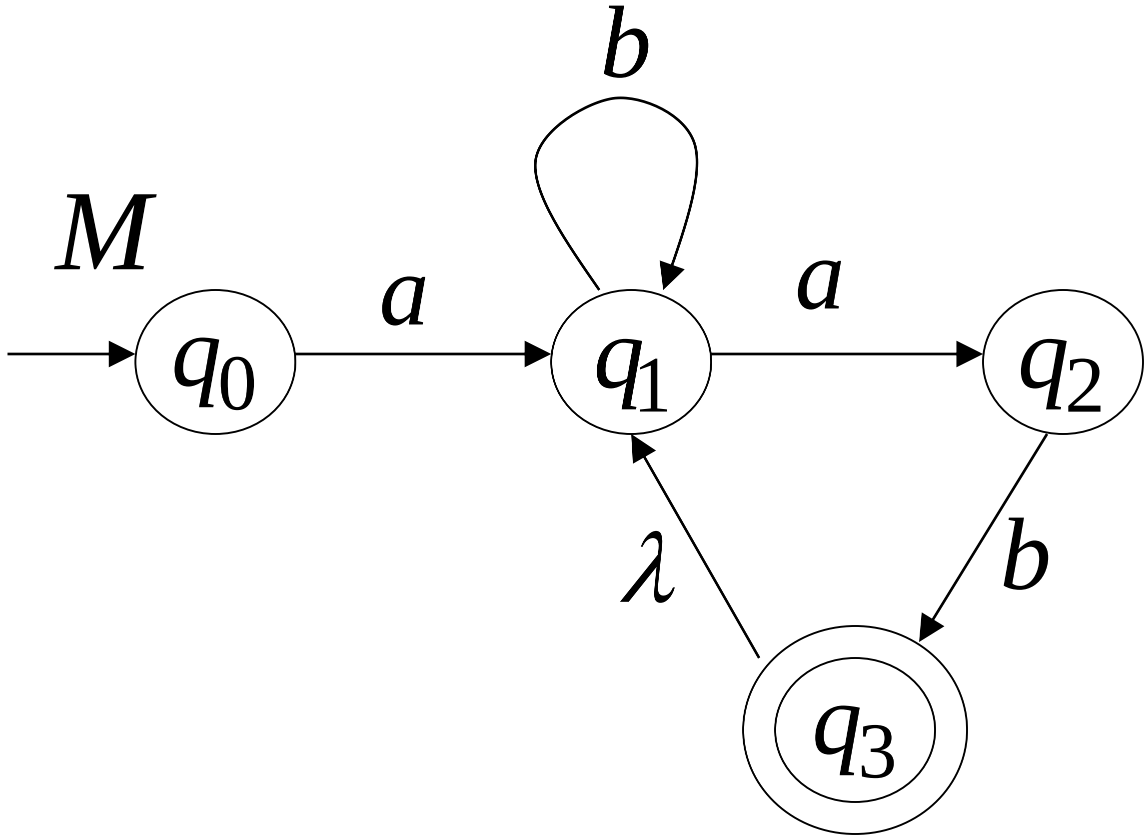
$$q_1 \to aq_2$$

$$q_2 \to bq_3$$

$$q_3 \to q_1$$

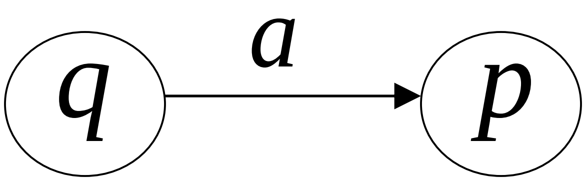
$$q_3 \to \lambda$$

M



In General

For any transition:

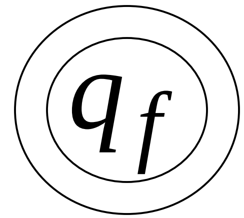


Add production:

$$q \rightarrow ap$$

variable terminal variable

For any final state:



Add production: $q_f \to \lambda$

Since G is right-linear grammar

G is also a regular grammar

with $L(G) = L(M) = L$