Grammars

Grammars

Grammars express languages

Example: the English language

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

ประโยคหนึ่ง

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

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

"Varible" "terminal symbol"

$$\langle \mathit{article} \rangle \xrightarrow{\text{derive}} a$$

$$\langle \mathit{article} \rangle \rightarrow \mathit{the}$$

$$\langle \mathit{noun} \rangle \xrightarrow{\text{der}} \mathit{cat}$$

$$\langle \mathit{noun} \rangle \rightarrow \mathit{dog}$$

$$\langle \mathit{verb} \rangle \xrightarrow{\text{der}} \mathit{runs}$$

$$\langle \mathit{verb} \rangle \rightarrow \mathit{walks}$$

A derivation of "the dog walks":

$$
\begin{aligned}
\langle sentence \rangle &\overset{\text{derm}}{\Rightarrow} \langle noun\_phrase \rangle \ \langle predicate \rangle \\
&\Rightarrow \langle noun\_phrase \rangle \ \langle verb \rangle \\
&\Rightarrow \langle article \rangle \ \langle noun \rangle \ \langle verb \rangle \\
&\Rightarrow the \ \langle noun \rangle \ \langle verb \rangle \\
&\Rightarrow the \ \ dog \ \langle verb \rangle \\
&\Rightarrow the \ \ dog \ \ walks
\end{aligned}
$$

A derivation of "a cat runs":

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

Language of the grammar:

L = { "a cat runs", string ของภาษา หรือ sentense
"a cat walks",
"the cat runs",
"the cat walks",
"a dog runs",
"a dog walks",
"the dog runs",
"the dog walks" }

แกรมมาในการตอม

Notation

Production Rules

$$\underbrace{\begin{array}{l}\langle noun \rangle \rightarrow cat \\ \langle noun \rangle \rightarrow dog\end{array}}^{\text{Production Rules}}$$

“ Variable ”

“ Terminal ”

แปร

Variable ก็ได้

Example

Grammar:

$$S \rightarrow aSb$$

แกรมม่าเป็นสธารภาษาของโปรแกรม

$$S \rightarrow \lambda$$

Derivation of sentence ab:

$$S \Rightarrow aSb \Rightarrow ab$$

กรณีที่เราเลือก

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

Grammar: $S \to aSb$

$S \to \lambda$

ถ้าอยากได้ string อื่นๆในภาษานี้

ก็ derive เรื่อยๆให้สิ้นแรง

Derivation of sentence $aabb$:

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

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

Other derivations:

[illegible]

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

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb$$
$$\Rightarrow aaaaSbbbb \Rightarrow aaaabbbb$$

Language of the grammar

$$S \to aSb$$

$$S \to \lambda$$

เห็น pattern ของมันแล้ว

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

More Notation

Grammar $G=(V,T,S,P)$ เขียนแบบทางการ

$V:$ 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: → คือ →

A sentence that contains variables and terminals

Example:

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

Sentential Forms   sentence

มีการผสมกันทั้ง variable, terminal   string ที่

We write:

ทำกันตั้งแต่ 0 ครั้งเป็นต้นไป

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

เขียนย่อ

แต่ทำให้เรามองไม่เห็นการ derive

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: $w \stackrel{*}{\Rightarrow} w$

Example

Grammar

$$S \rightarrow aSb$$

$$S \rightarrow \lambda$$

ภาษาของ grammar นี้คือ $L = \{a^n b^n ; n \geq 0$

Derivations

$*$ เหมือนการเขียนรวบเลย

$$S \Rightarrow \lambda$$

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

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

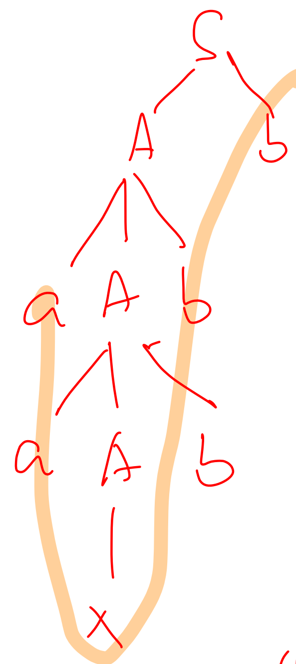
$$S \overset{*}{\Rightarrow} aaabbb$$

Another Grammar Example

Grammar G **:** $S \to Ab$

$$A \to aAb$$

$$A \to \lambda$$



$a^n b^n b$

Derivations: ก็นับลงหาพ

$$S \to Ab \to b$$

$$S \to Ab \to aAbb \to abb$$

$$S \to Ab \to aAbb \to aaAbbb \to aabbb$$

More Derivations

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

$$\Rightarrow aaaaAbbbbb \Rightarrow aaaabbbbb$$

ใครจะอะไรออกมา brute force ไป

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

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

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

Language of a Grammar

For a grammar G .เรามี grammar G

with start variable S :

langu. ของ gram. G คือ set ของ string ๆ โดยเริ่มต้นจาก S

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

สามารถ derive ออกมาได้ 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: $S \overset{*}{\Rightarrow} a^n b^n b$

ถ้าเริ่มต้นจาก S ทำ derivation ไปเรื่อยๆ จะได้ string ในรูป

A Convenient Notation

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

ประหยัดบรรทัดในการเขียน

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

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

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

Linear Grammars

ขึ้นและมีความเกี่ยวข้องกันกับภาษา regular

Linear Grammars

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

grammar ที่
มีแค่ ≤ 1 var อยู่ทางขวาของ production

1 var

Examples:

$$S \to aSb \qquad\qquad S \to Ab$$

0 var

$$S \to \lambda \qquad\qquad A \to aAb$$

$$A \to \lambda$$

A Non-Linear Grammar

Grammar G :

$$S \rightarrow SS$$ 2ตัว ; ปมที่ต้นเป็นทอง

↓

ที่รวมหมดเป็นกลาง

$$S \rightarrow \lambda$$

$$S \rightarrow aSb$$

$$S \rightarrow bSa$$

จำนวน a = b

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

Number of a in string w

Another Linear Grammar

Grammar G :

$$S \rightarrow A$$

$$A \rightarrow aB \mid \lambda$$

$$B \rightarrow Ab$$

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

Right-Linear Grammars

All productions have form: $A \rightarrow xB$

เหมือนเส้นแต่ตำแหน่ง var อยู่ขวามือ

or

$A \rightarrow x$

string of terminals

Example: $S \rightarrow abS$

$S \rightarrow a$

Left-Linear Grammars

อยู่ทางซ้าย

All productions have form:

$$A \to Bx$$

or

$$A \to x$$

string of terminals

Example:

$$S \to Aab$$

$$A \to Aab \mid B$$

left or right up to context

$$B \to a$$

Regular Grammars

Regular Grammars

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

L(regular grammar) = Regular language

Examples:

G_1

$$S \rightarrow abS$$ right

$$S \rightarrow a$$

G_2

left

$$S \rightarrow Aab$$

$$A \rightarrow Aab \mid B$$

$$B \rightarrow a$$

Observation

Regular grammars generate regular languages

Examples:

G_1

$$S \rightarrow abS$$

$$S \rightarrow a$$

∴ เป็น regular language

สร้าง regular expression ออกมาได้

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

G_2

$$S \rightarrow Aab$$

$$A \rightarrow Aab \mid B$$

$$B \rightarrow a$$

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

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 $L(M) = L(G)$

ภาษาของ regular grammar = regular ถึงว่าไง

Grammar G is right-linear

Example:

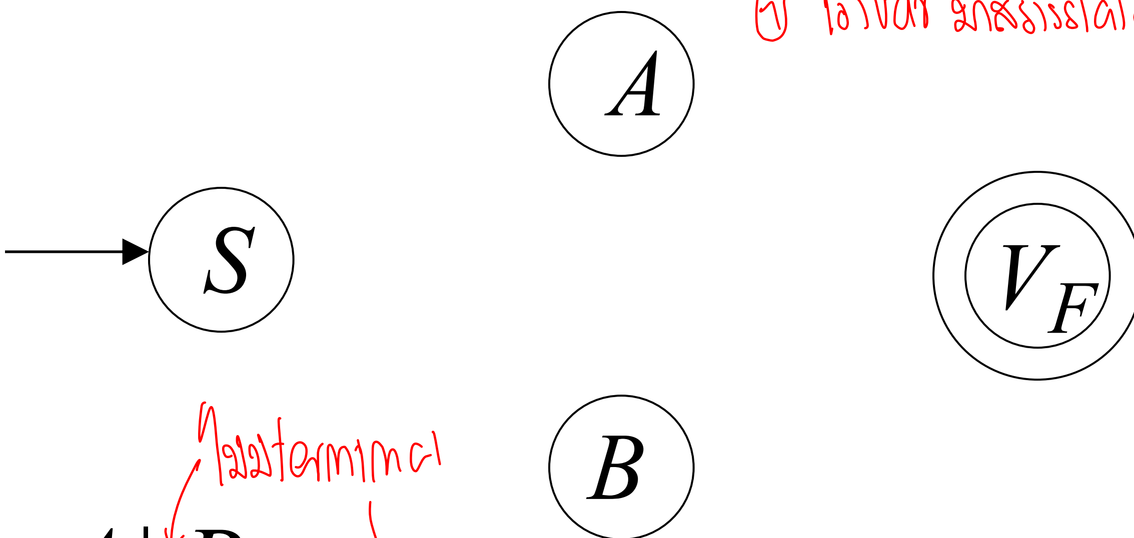
$$S \to aA \mid B$$

$$A \to aa\,B$$

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

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

① เอาvar มาสร้างstate



② สร้างขึ้นมาลอยๆ

special final state

ไม่มีterminal

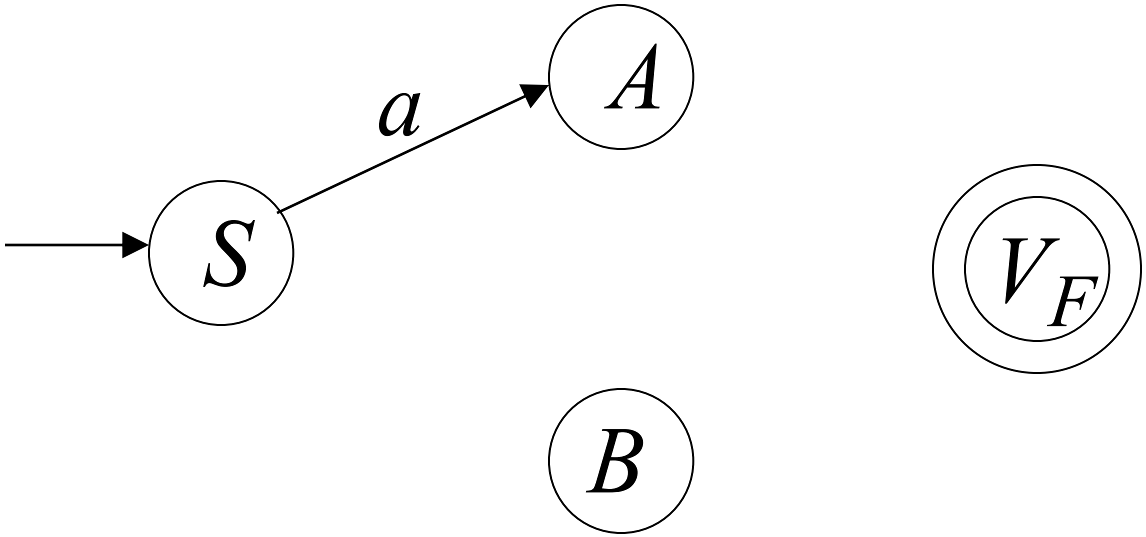
actual = λ

$$S \rightarrow aA \mid B$$

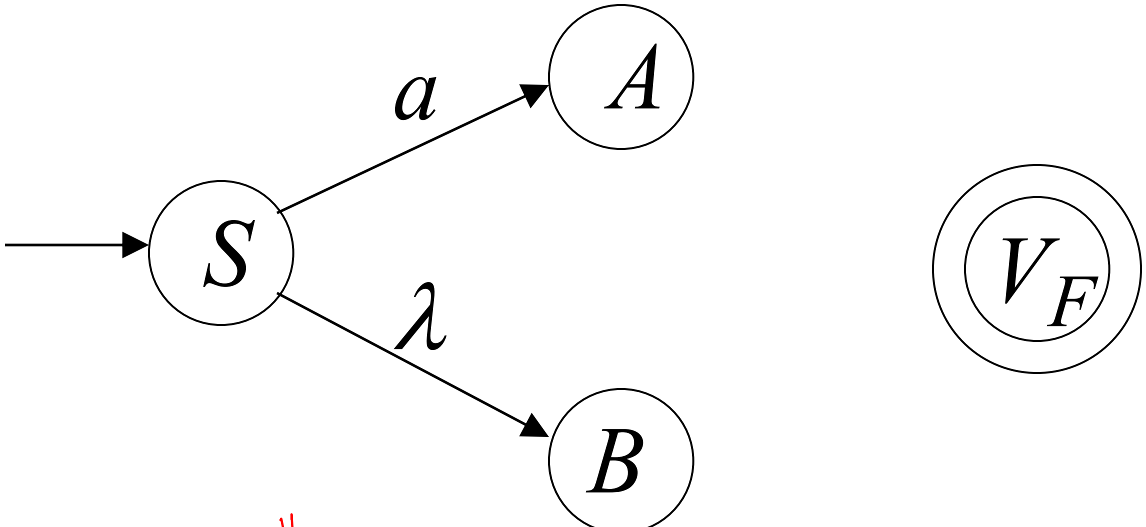
$$A \rightarrow aa\,B$$

$$B \rightarrow b\,B \mid a$$

Add edges for each production:

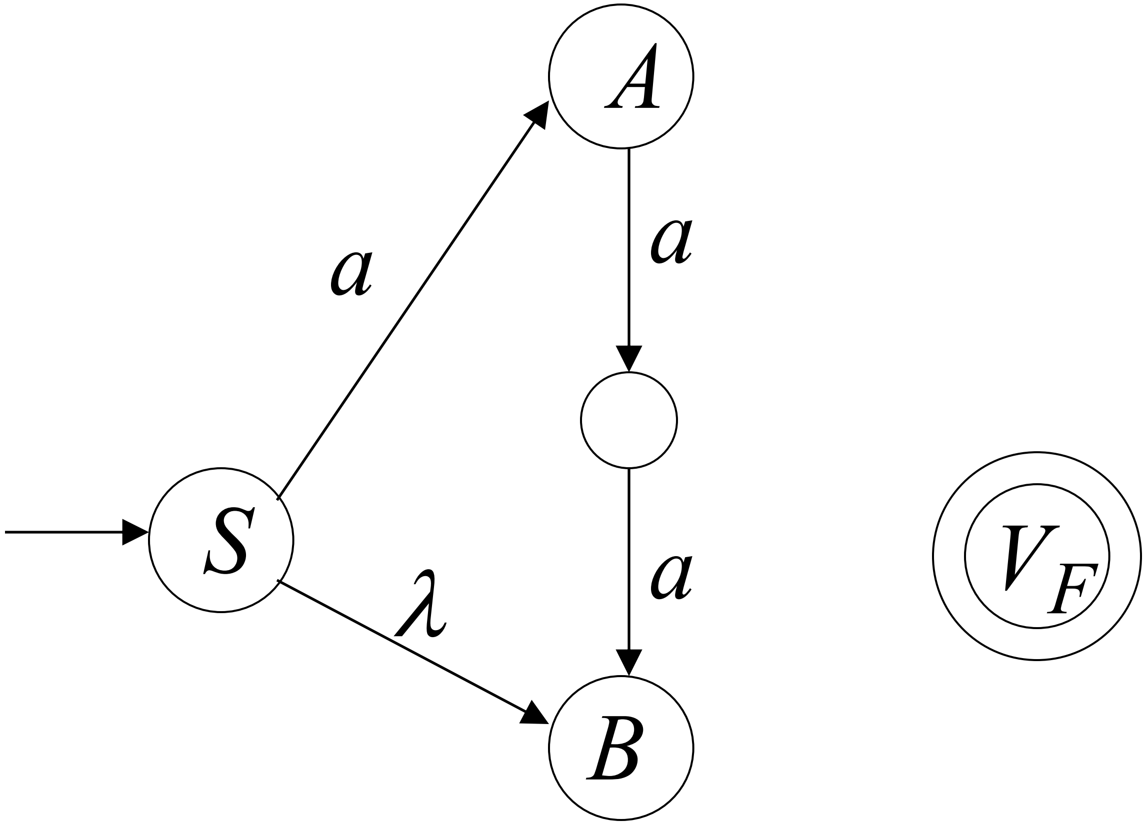


$$S \rightarrow aA$$



nothing

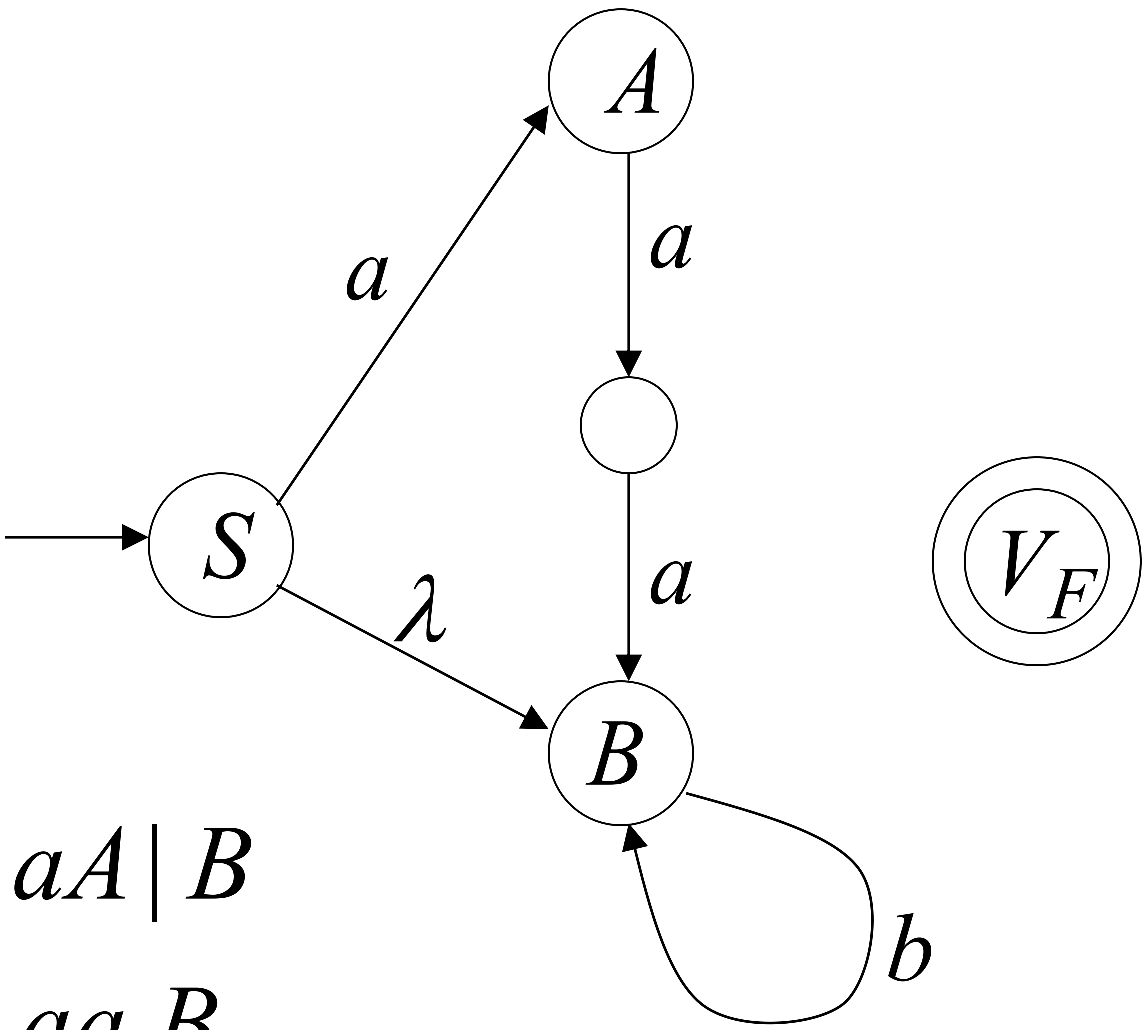
$$S \rightarrow aA \mid B$$



$$S \to aA \mid B$$

terminal 2 ตัว เป็นธรรมดาต้องสร้าง state ใหม่

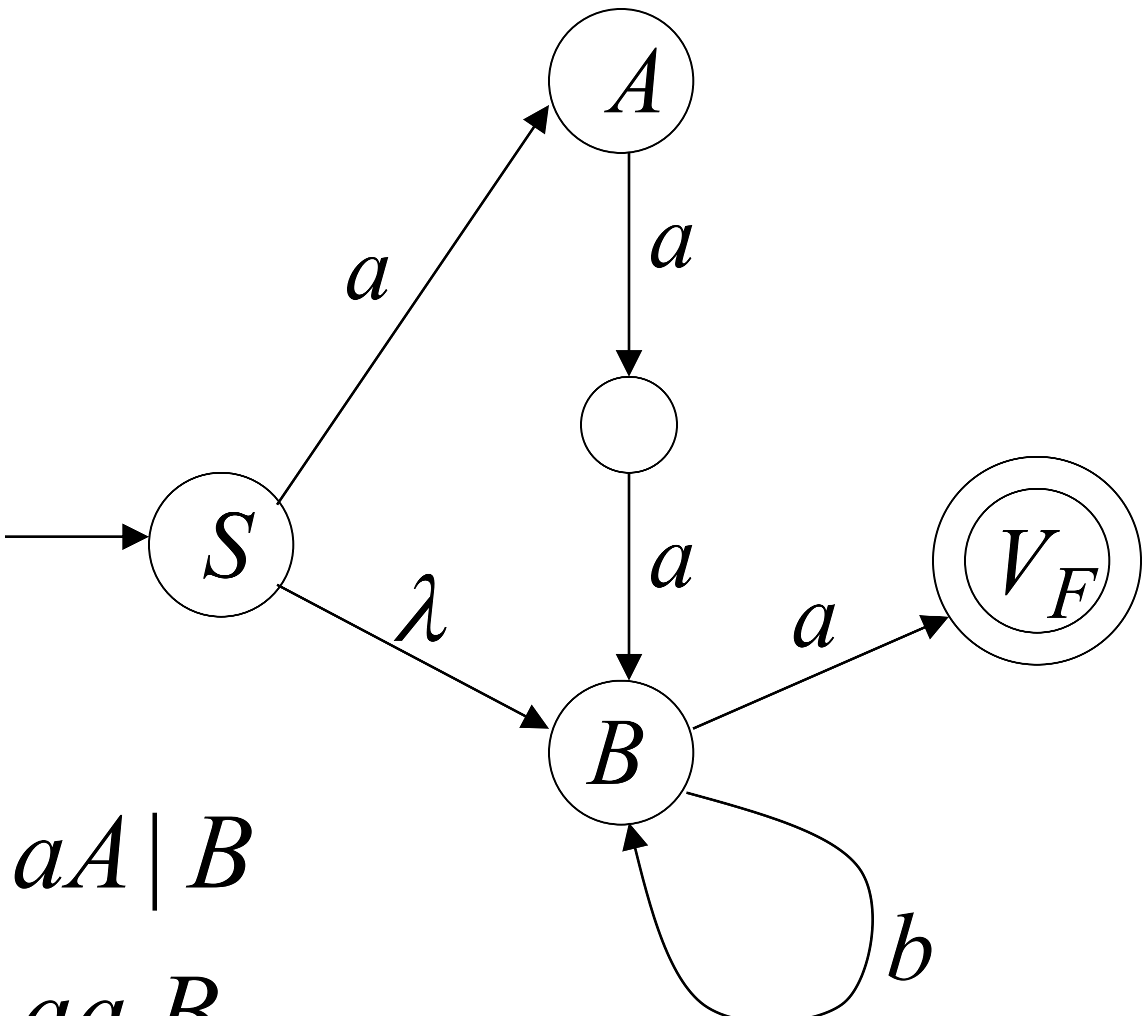
$$A \to aa\,B$$



$$S \rightarrow aA \mid B$$

$$A \rightarrow aa\ B$$

$$B \rightarrow bB$$

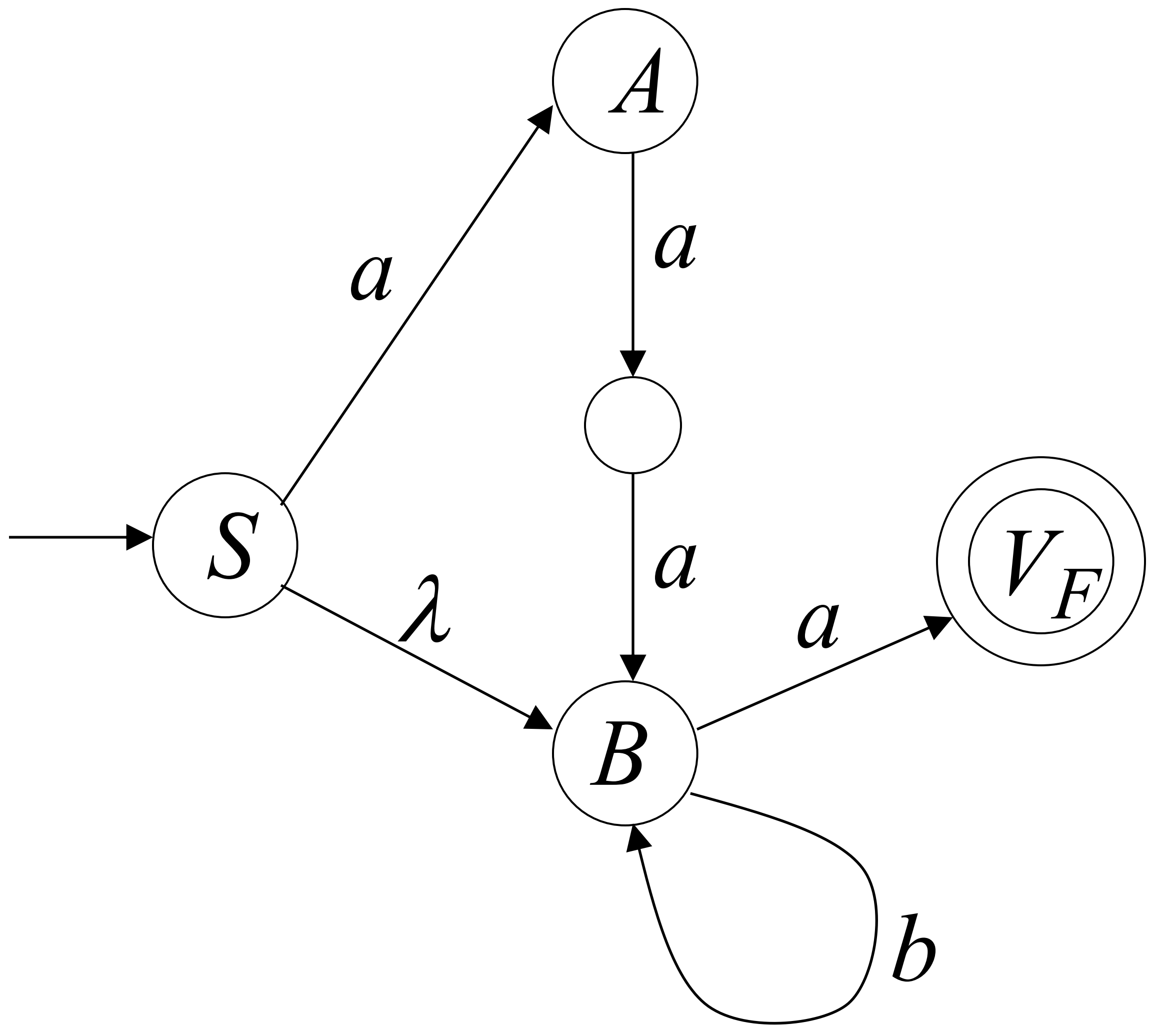


$$S \rightarrow aA \mid B$$

$$A \rightarrow aa\,B$$

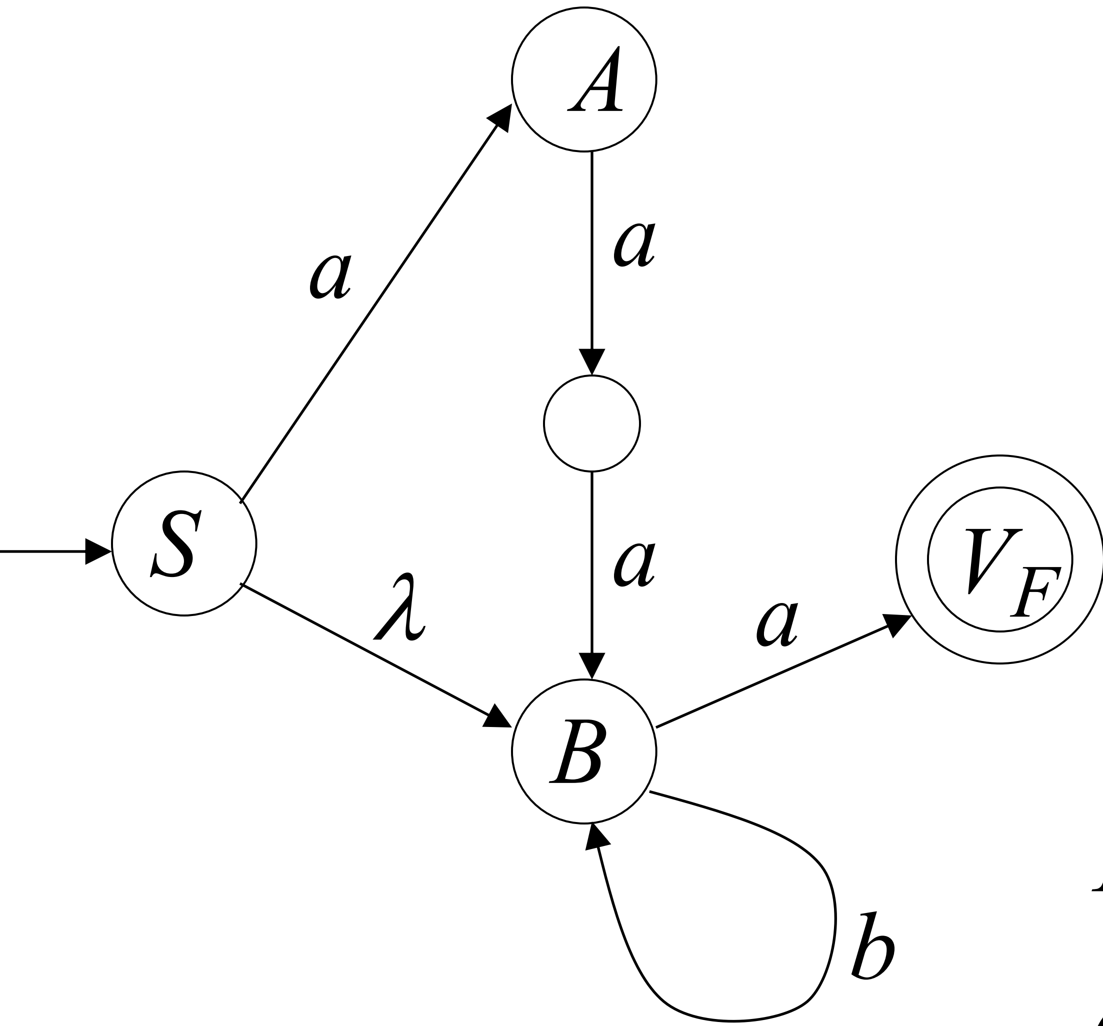
$$B \rightarrow bB \mid a$$

ถ้าไม่มี variable ตามหลังชี้เข้า final state ได้เลย



$$S \Rightarrow aA \Rightarrow aaaB \Rightarrow aaabB \Rightarrow aaaba$$

NFA M



Grammar

G

$$S \to aA \mid B$$

$$A \to aa\, B$$

$$B \to bB \mid a$$

สร้าง NFA ได้ → เป็น regular language

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

$$aaab^*a + b^*a$$

In General

ทำความเข้าใจเอง

A right-linear grammar G

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

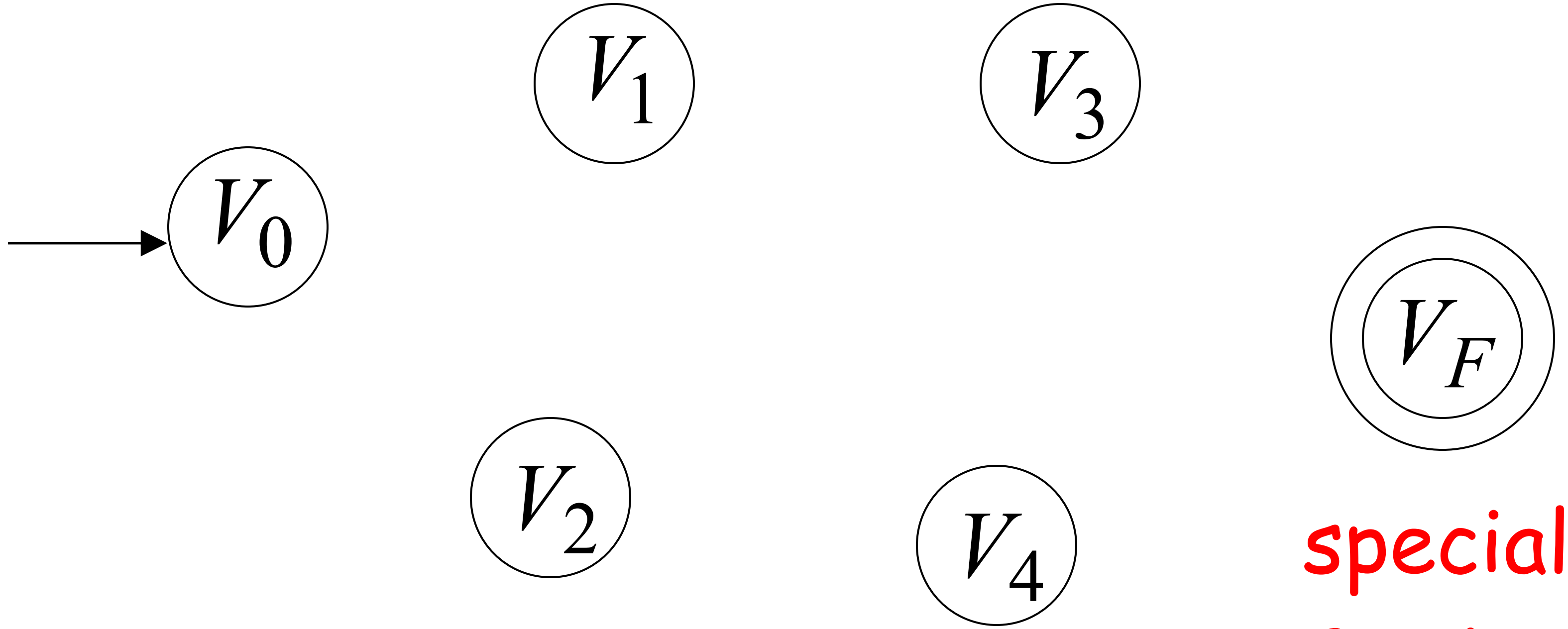
and productions: $V_i \to a_1 a_2 \cdots a_m V_j$

or

$V_i \to a_1 a_2 \cdots a_m$

We construct the NFA M such that:

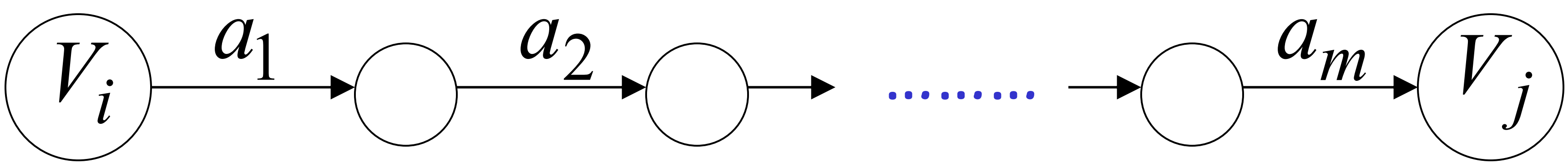
each variable V_i corresponds to a node:



special
final state

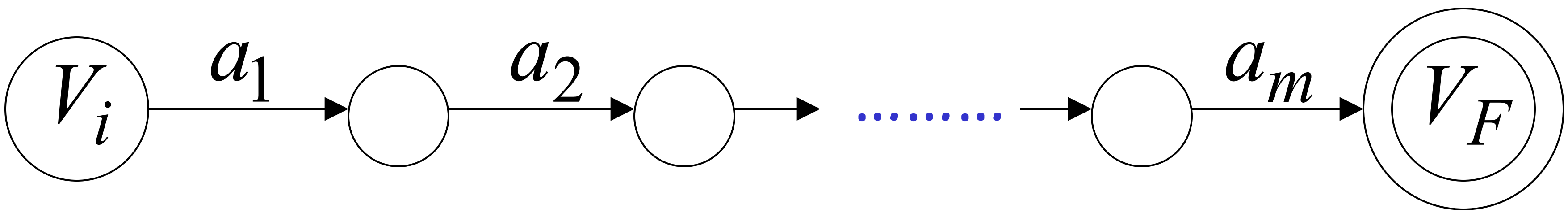
For each production: $V_i \to a_1 a_2 \cdots a_m V_j$

we add transitions and intermediate nodes

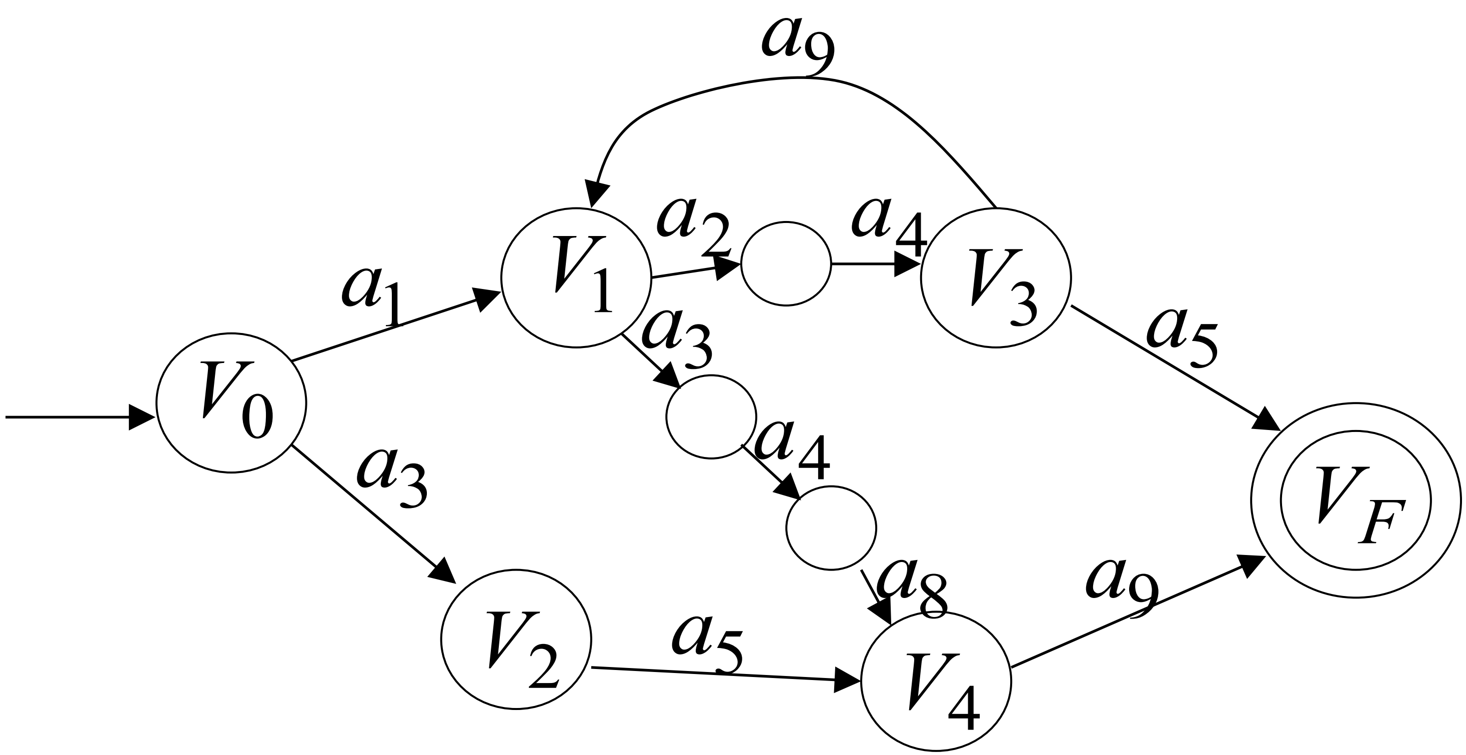


For each production: $V_i \to a_1 a_2 \cdots a_m$

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

เปลี่ยน left linear เป็น right

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 $L(G) = L(G')^R$

ถ้าทำได้ → เป็น regular

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

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

$$A \to Bv$$

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

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

$$A \to v$$

Right linear G'

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

$$A \to v^R$$

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

Since G' is right-linear, we have:

เอา regular มา reverse ได้ regular เหมือนเดิม

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

Regular Language | Regular Language | Regular Language

Proof - Part 2

พิสูจน์ส่วนเท่าการกับเมื่อได้

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

Any regular language L is generated by some regular grammar G

Any regular language L is generated by some regular grammar G

Proof idea:

เริ่มกับ

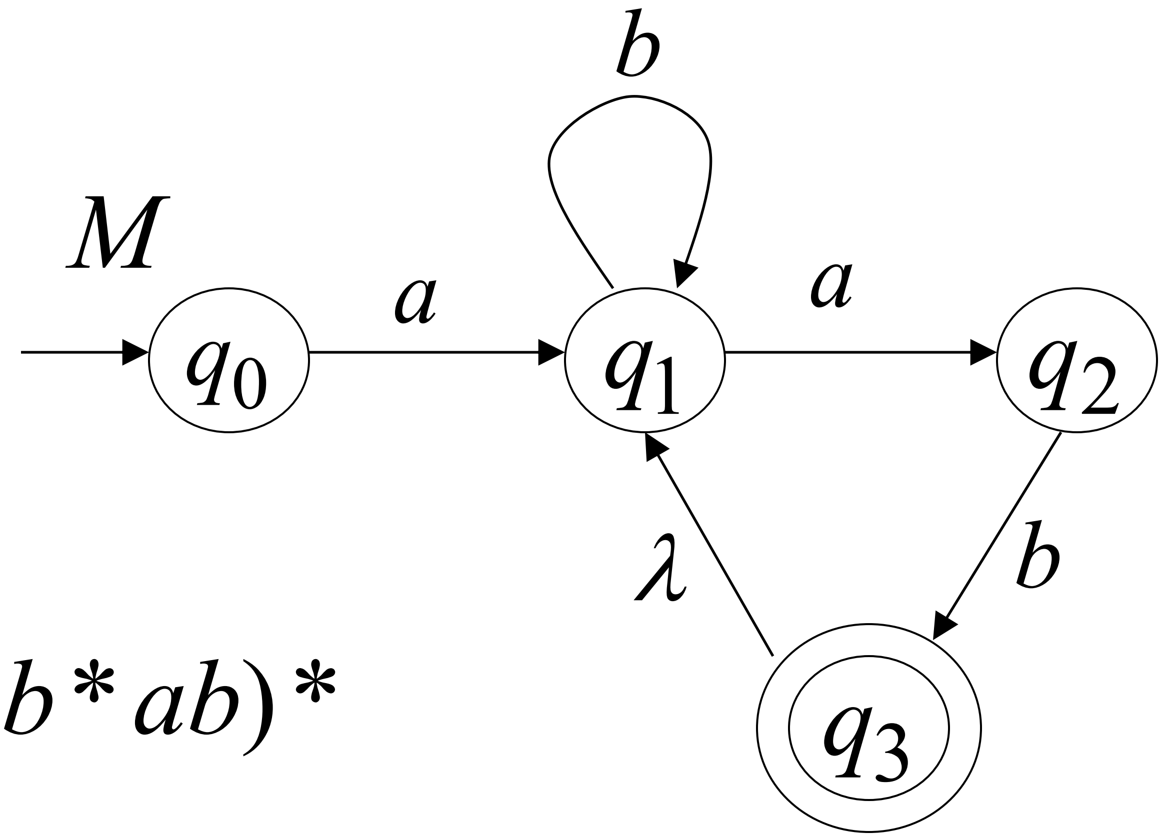
Let M be the NFA with $L = L(M)$.

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

Since L is regular

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

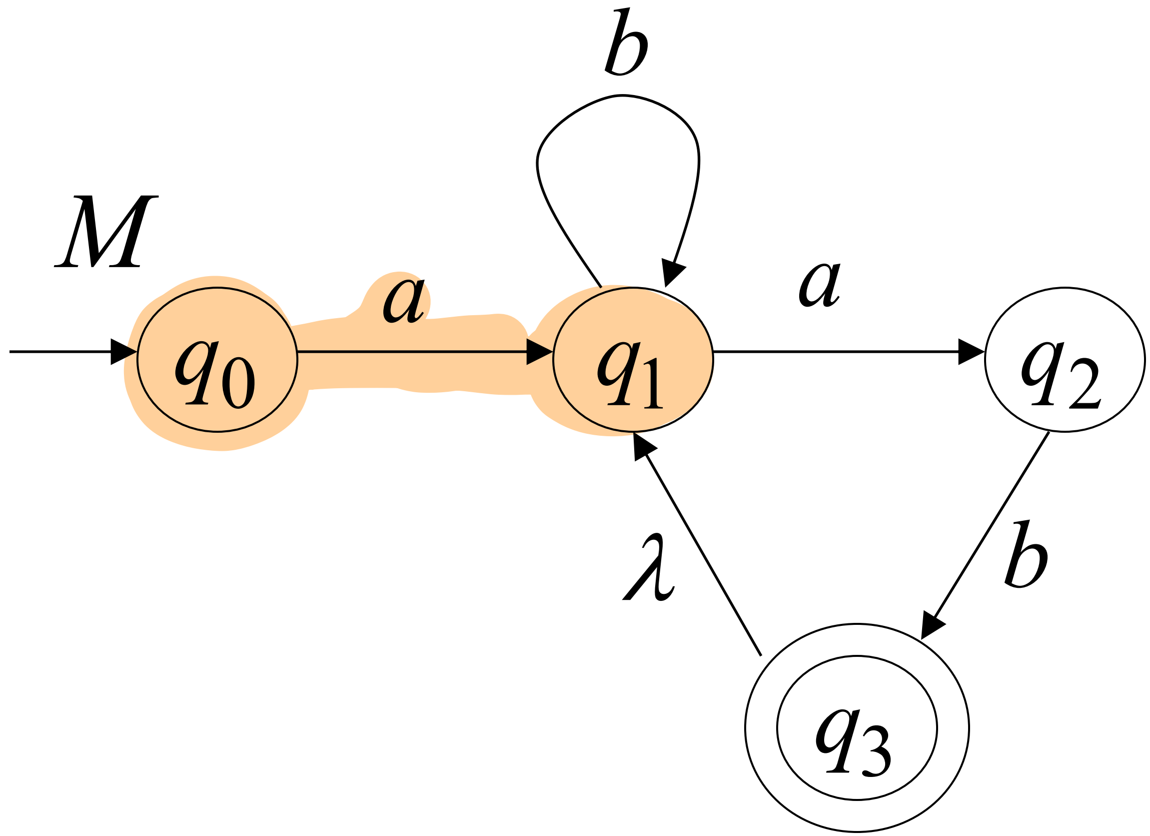
Example:



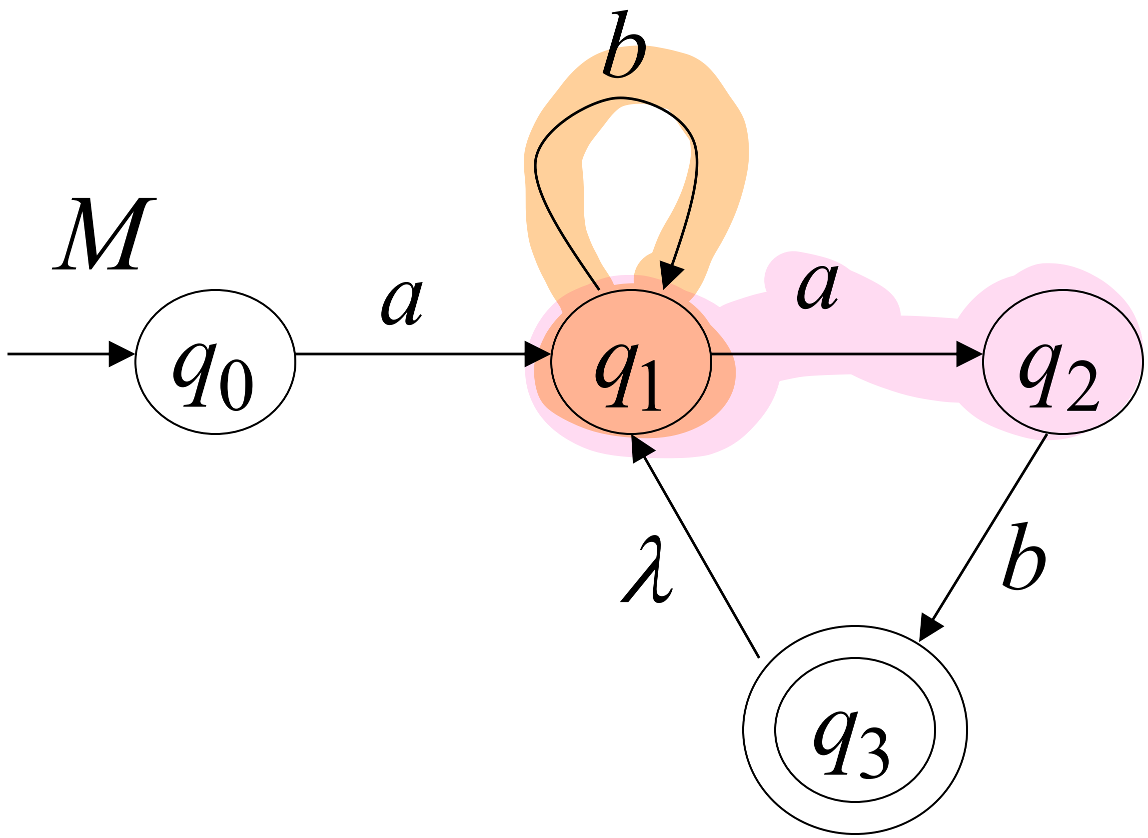
$L = ab^*ab(b^*ab)^*$

$L = L(M)$

Convert M to a right-linear grammar



$$q_0 \to a q_1$$



$q_0 \to aq_1$

$q_1 \to bq_1$

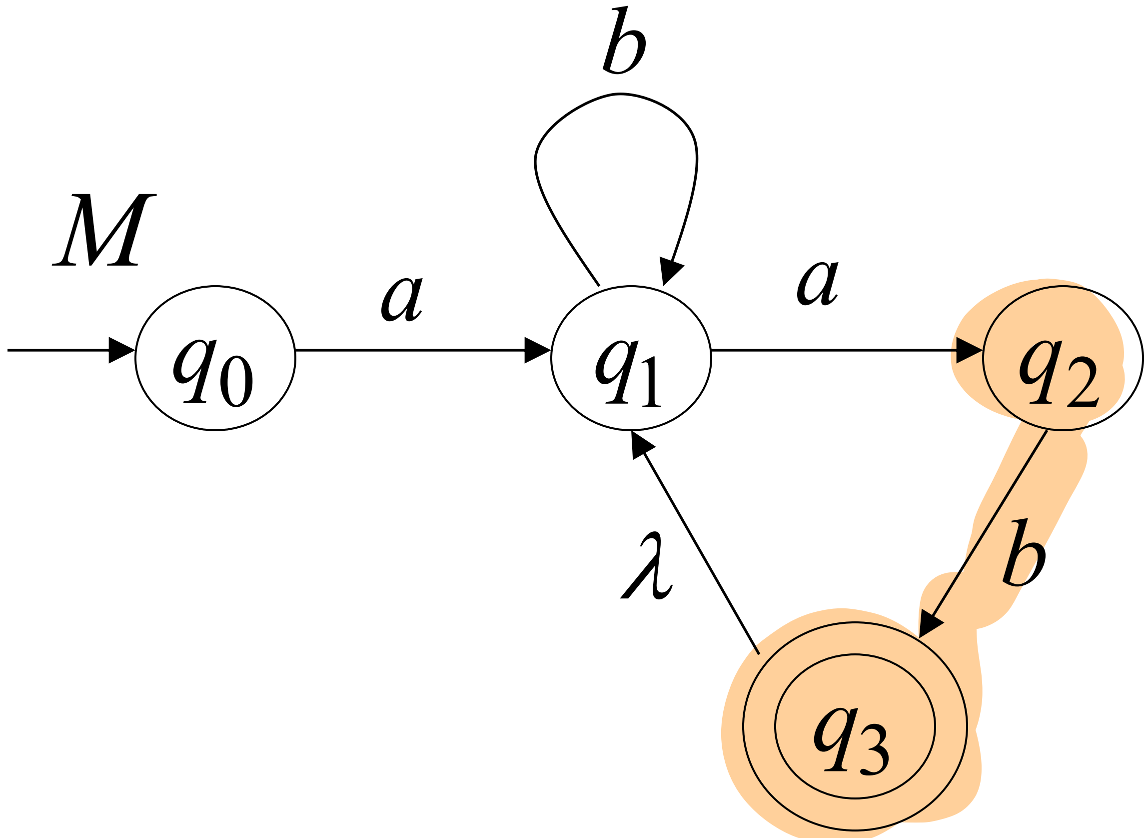
$q_1 \to aq_2$

$$q_0 \to aq_1$$

$$q_1 \to bq_1$$

$$q_1 \to aq_2$$

$$q_2 \to bq_3$$



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

สร้างgrammarได้

G

$$q_0 \rightarrow aq_1$$

$$q_1 \rightarrow bq_1$$

$$q_1 \rightarrow aq_2$$

$$q_2 \rightarrow bq_3$$

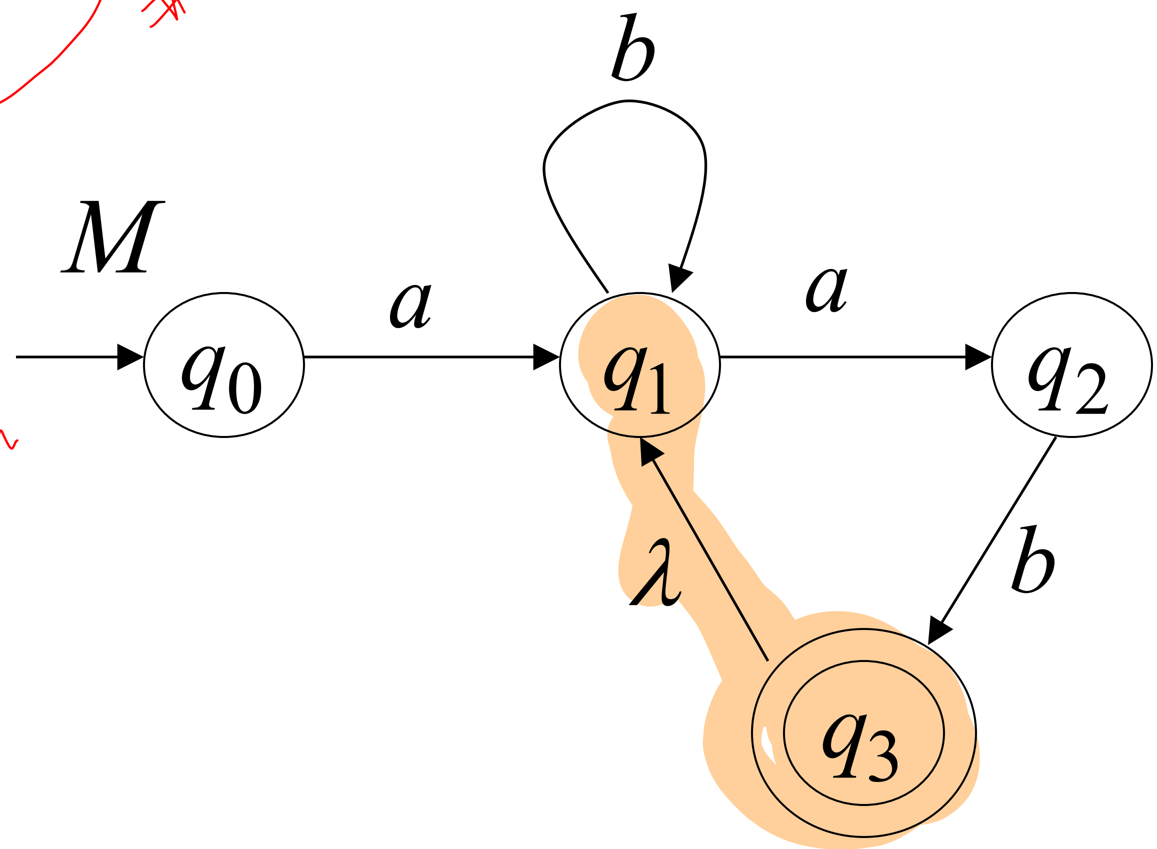
right linear

$$q_3 \rightarrow q_1$$

λ ไม่ต้องเขียน

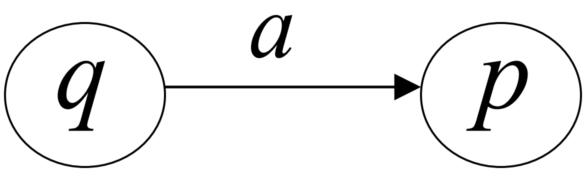
$$q_3 \rightarrow \lambda$$

q_3 เป็น final จะ production แบบนี้ได้เลย



In General

For any transition:

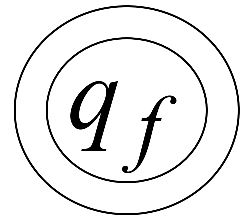


Add production:

$$q \rightarrow ap$$

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