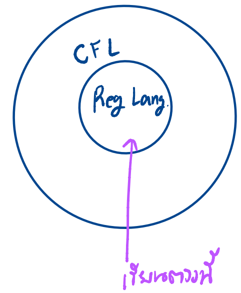
Reg Lang.

DFA, NFA
Tools for
บอกว่า lang.
เป็น regular lang. มั้ย

reg Ex
อธิบายภาษา Reg Lang
เช่น $(a+b)^*$

Regular Grammar
เป็นตัว Generate Reg. Lang.



Grammars

Grammars

Grammars express languages

Example: the English language

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

อ่านว่า Derive

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

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

(Non terminal) Variable

terminal symbol
เป็น สัญลักษณ์ ที่ไม่ Derive ต่อไปได้แล้ว / ตัวสุดท้ายแล้ว

$$\langle article \rangle \rightarrow \underline{a}$$

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

$$\langle noun \rangle \rightarrow cat$$

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

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

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

เอามาสร้างเป็นประโยค

A derivation of "the dog walks":

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

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:

ประกอบด้วย 8 sentence
หรือ 8 string

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

ภาษ. production

Notation

Gamma ในภาษาคอม

Production Rules (Gamma)

เรียกว่า

$$\langle noun \rangle \rightarrow cat$$

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

Variable

Terminal

Example

Grammar:

$$S \to aSb$$

$$S \to \lambda$$

Derivation of sentence ab:

มี 2 ทางเลือก

λ

$$S \Rightarrow aSb \Rightarrow ab$$

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

Grammar:

$$S \to aSb$$

$$S \to \lambda$$

เพื่อเป็น str ที่ยาวขึ้น

Derivation of sentence $aabb$:

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

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

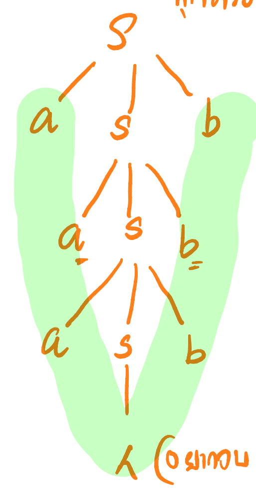
Other derivations:

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

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

Language of the grammar

ทุกครั้งที่แตก S จะเป็น pattern aSb เสมอ



(อย่าลืม ให้ S = λ)

เอาลูกมาต่อกัน

aaa bbb

ดูแล้ว detect ได้เลย

$$S \rightarrow aSb$$

$$S \rightarrow \lambda$$

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

More Notation

สัญลักษณ์ Grammar

เขียน

แบบทางการ

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 :

Variable

$$S \to aSb$$

$$S \to \lambda$$

Gamma ไม่ต้องการ

กฎทั้ง 2 ข้อมาใส่

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

(start variable)

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

↳ λ ไม่เอามาใส่ เอาเฉพาะตัวอักษรที่มองเห็น

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

ประกอบด้วย

More Notation

Sentential Form:

A sentence that contains variables and terminals

Example:

ผสมกันกับ terminal กับvariable

เป็น str ที่ประกอบด้วย terminal

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

Sentential Forms

sentence

เห็นแค่ตัวเริ่ม ทำ Derivation ตั้งแต่ 0 ครั้งขึ้นไป

กับตัวท้ายสุด

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

มองไม่เห็น

Instead of:

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

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

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

str ใดๆ ที่ทำ Derivertion * แล้วกลับมาเป็นตัวเดิม

By default:

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

Example

Grammar

$$S \to aSb$$

$$S \to \lambda$$

Derivations

1

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

2

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

3

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

4

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

Another Grammar Example

Grammar G :

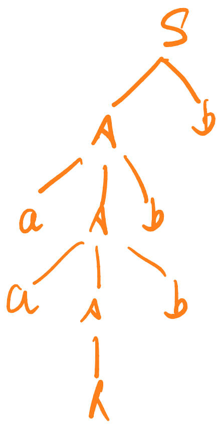
$$S \to Ab$$

$$A \to aAb$$

$$A \to \lambda$$

หา L จาก Gamma โดยตรง

ภาษาของ Gamma $a^n b^n b \; ; \; n \geq 0$



มองไม่ออก ทำแบบนี้

Derivations:

$$S \to Ab \to b$$

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

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

More Derivations

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

$$\Rightarrow aaaaAbbbbb \Rightarrow aaaabbbbb$$

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

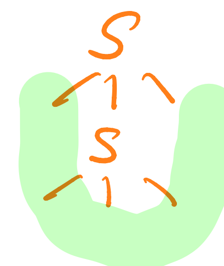
$$S \overset{*}{\Rightarrow} aaaaaabbbbbbb$$

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

Language of a Grammar

For a grammar G

with start variable S : ของ Gamma G



ต่อกัน = w

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

Str ประกอบด้วย terminal ทั้งหมด

String of terminals

Example

For grammar G:

$$S \to Ab$$

$$A \to aAb$$

$$A \to \lambda$$

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

ทำ Derivertion มาเรื่อยจะได้ไปแบบนี้

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

A Convenient Notation

$$A \rightarrow aAb$$
$$A \rightarrow \lambda$$

เขียนได้แบบนี้

$$\Longrightarrow \quad A \rightarrow \underline{aAb} \mid \underline{\lambda}$$

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

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

Linear Grammars

Linear Grammars

2 ปท.
left
right

Grammars with

จำนวน Variable มีได้ไม่เกิน 1 ทาง ขวา

at most **one variable** at the **right side**

ของ

of a production

linear gamma

Examples:

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

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

$$A \to \lambda$$

A Non-Linear Grammar

Grammar G :

$$S \rightarrow \underset{\sim\sim}{S}S$$

มี 2 ตัว ขีดแยกมาจาก เดียว ทั้งหมดคือ ไม่ใช่ linear

$$S \rightarrow \lambda$$

จำนวนของ a จำนวน b
$n_a(w) = n_b(w)$

$$S \rightarrow aSb$$

ทุกครั้งที่สร้าง a มีจะสร้าง b ขึ้นมา 1 ตัว

$$S \rightarrow bSa$$

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

Number of a in string w

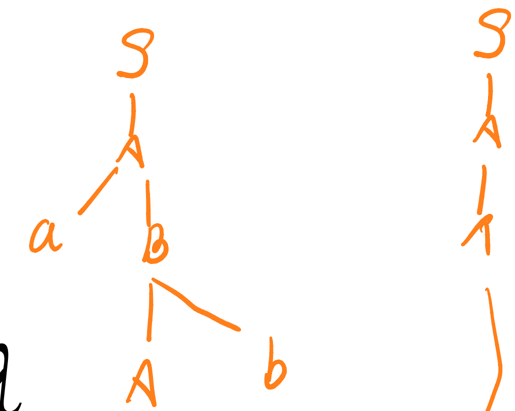
Another Linear Grammar

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

ขวามือสุด

All productions have form:

$$A \to xB$$

or

$$A \to x$$

string of terminals

Example:

$$S \to abS$$

$$S \to a$$

Left-Linear Grammars

ซ้ายสุด

All productions have form: $A \to Bx$

or

$A \to x$

string of terminals

Example:

$$S \to Aab$$

$$A \to Aab \mid B$$ = ดูจากมีพทแล้ว B อยู่ left

$$B \to a$$

Regular Grammars

Regular Grammars มีทคือ Regular Languages

A regular grammar is any คือ

right-linear or left-linear grammar

อย่างใดอย่างหนึ่ง ไม่ฟนกัพ ไม่ฟนกัพ

Examples:

ไม่ฟนกัพ

$$G_1$$

$$S \rightarrow abS$$

$$S \rightarrow a$$

$$G_2$$

$$S \rightarrow Aab$$

$$A \rightarrow Aab \mid B$$

$$B \rightarrow a$$

Observation

สร้าง

Regular grammars generate regular languages

Examples:

$$G_1$$

$$S \to abS$$

$$S \to a$$

ภาษาใดก็ตามที่เขียน ได้คือ Reg. Lang.

Regular Expression

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

$$G_2$$

$$S \to Aab$$

$$A \to Aab \mid B$$

$$B \to a$$

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

Regular Grammars Generate Regular Languages

Theorem

ภาษาที่

Languages
Generated by

ออกมาจาก

Regular Grammars

=

Regular
Languages

Theorem - Part 1

ๆเป็น subset ของขวา

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

แปลง grammar เป็น NFA

Any regular grammar generates a regular language

Theorem - Part 2

เป็น subset ของ ซ้าย

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

จาก NFA → L Grammar

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

1 $L(G)$ ต้องแปลงเป็น .... ได้

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

Grammar G is right-linear

Example:

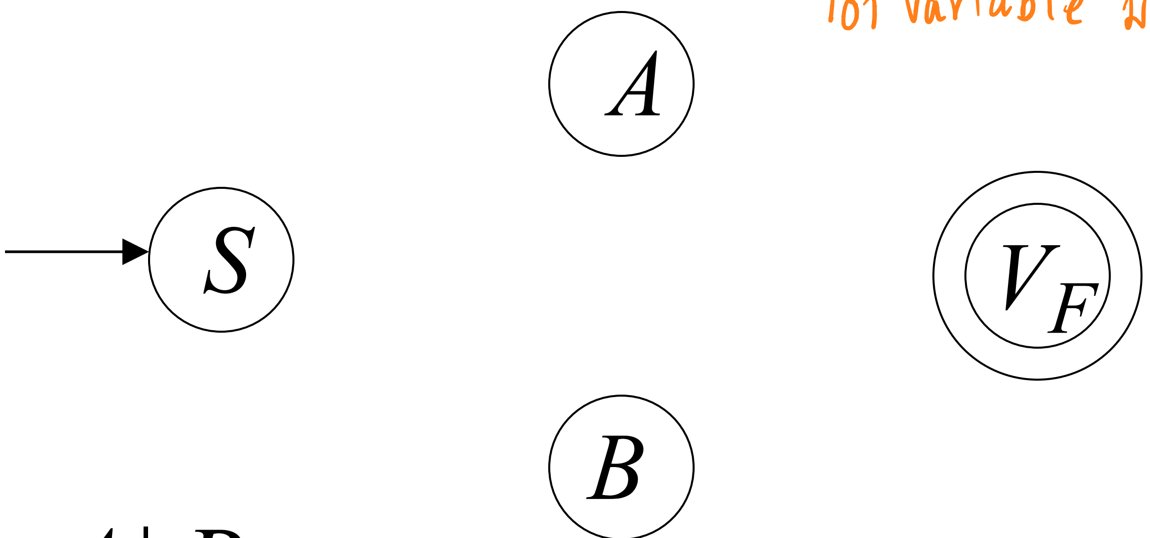
$$S \to aA \mid B$$

$$A \to aa\,B$$

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

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

เอา variable มาทำเป็น state



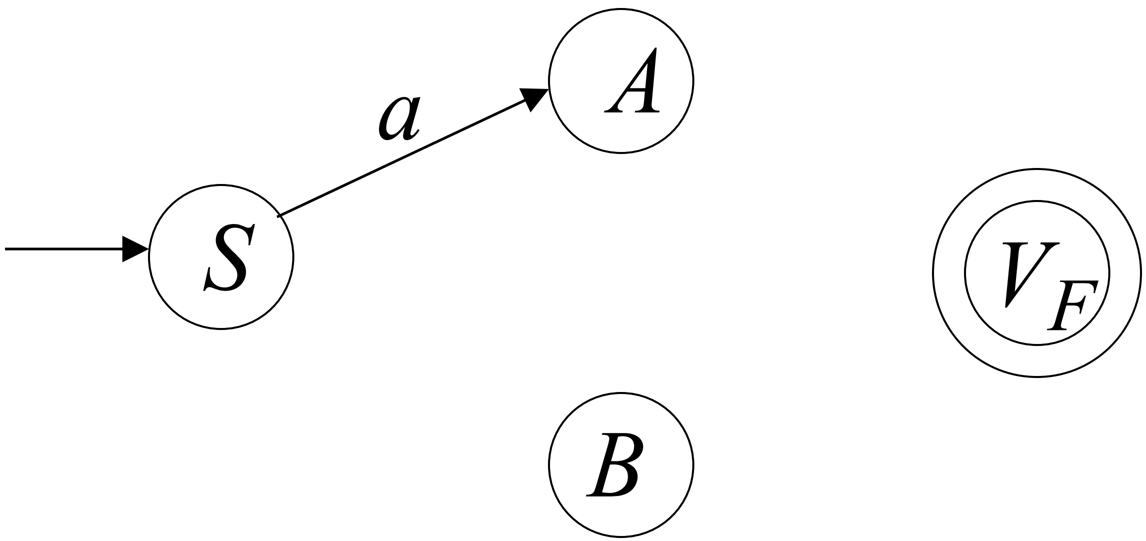
special final state

$$S \to aA \mid B$$

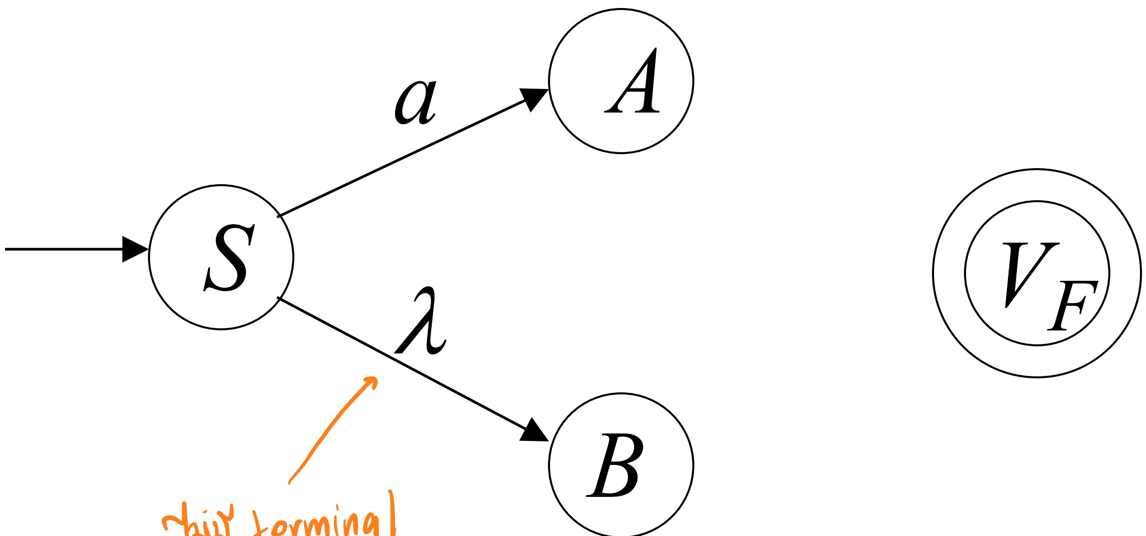
$$A \to aa\,B$$

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

Add edges for each production:

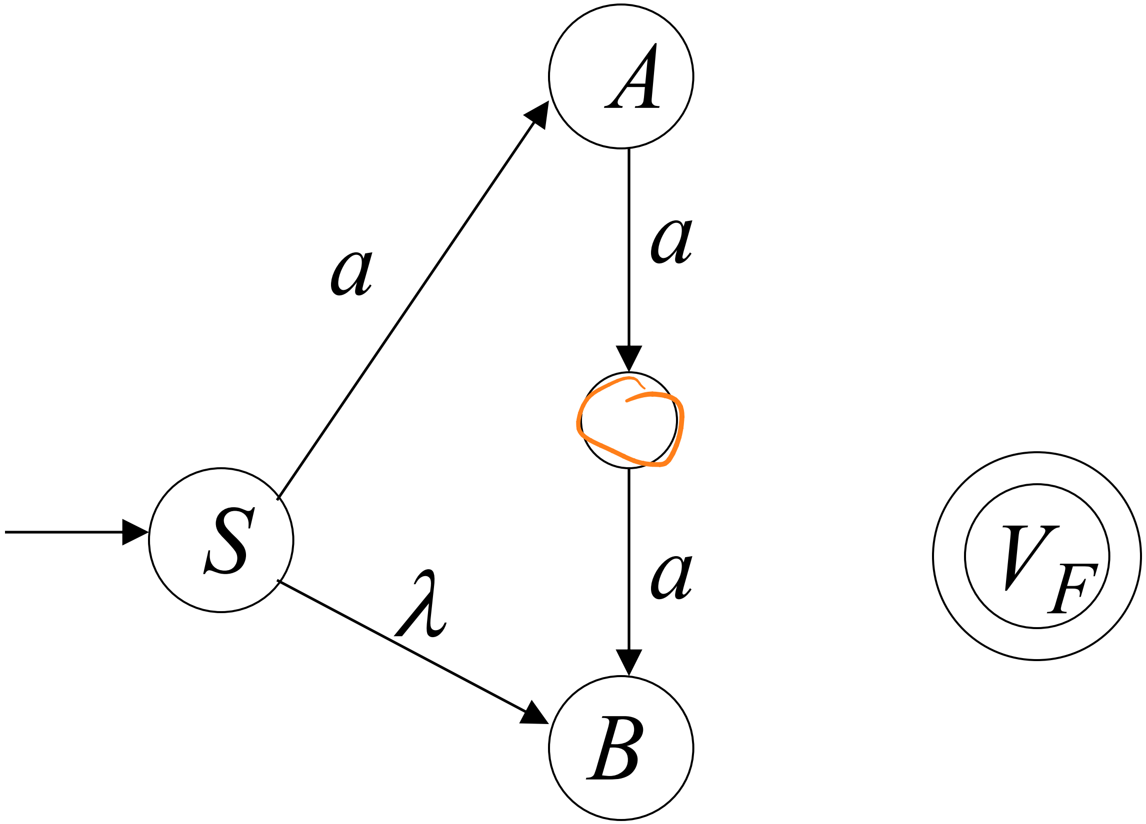


$$S \to aA$$



ไม่มี terminal

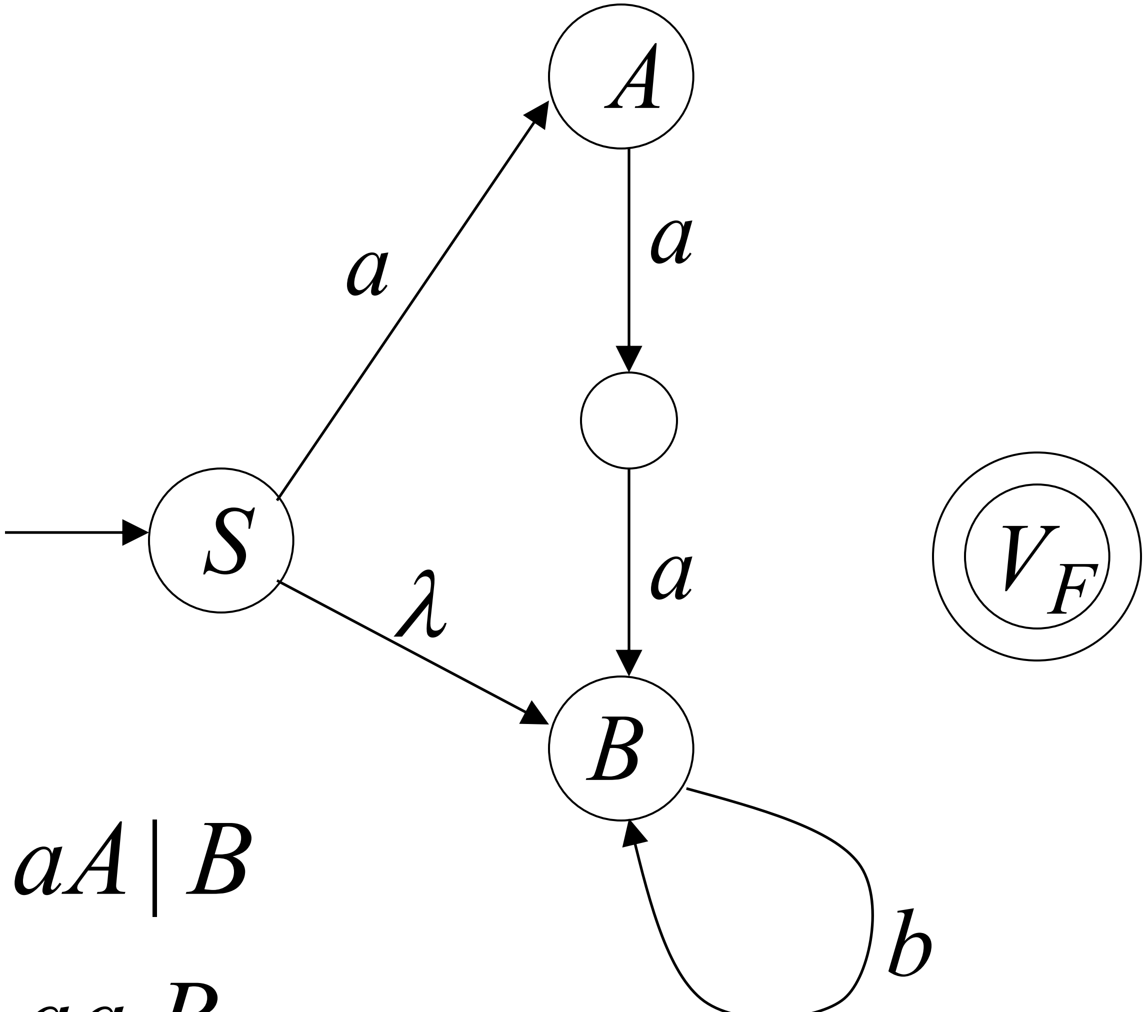
$$S \rightarrow aA \mid B$$



$$S \to aA \mid B$$

มี terminal 2 ตัว ต้องมี state มารองรับ

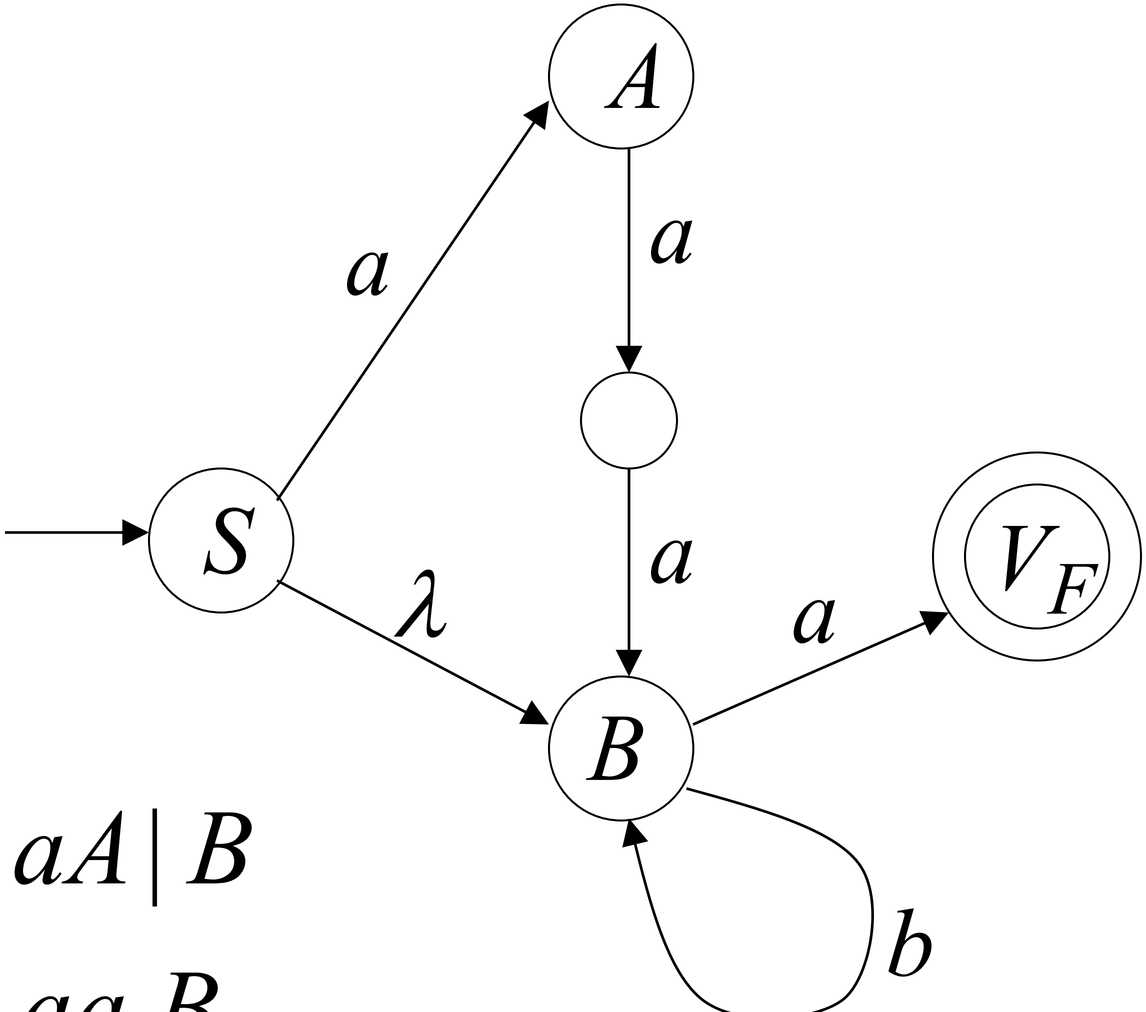
$$A \to aa\, B$$



$$S \rightarrow aA \mid B$$

$$A \rightarrow aa\,B$$

$$B \rightarrow bB$$



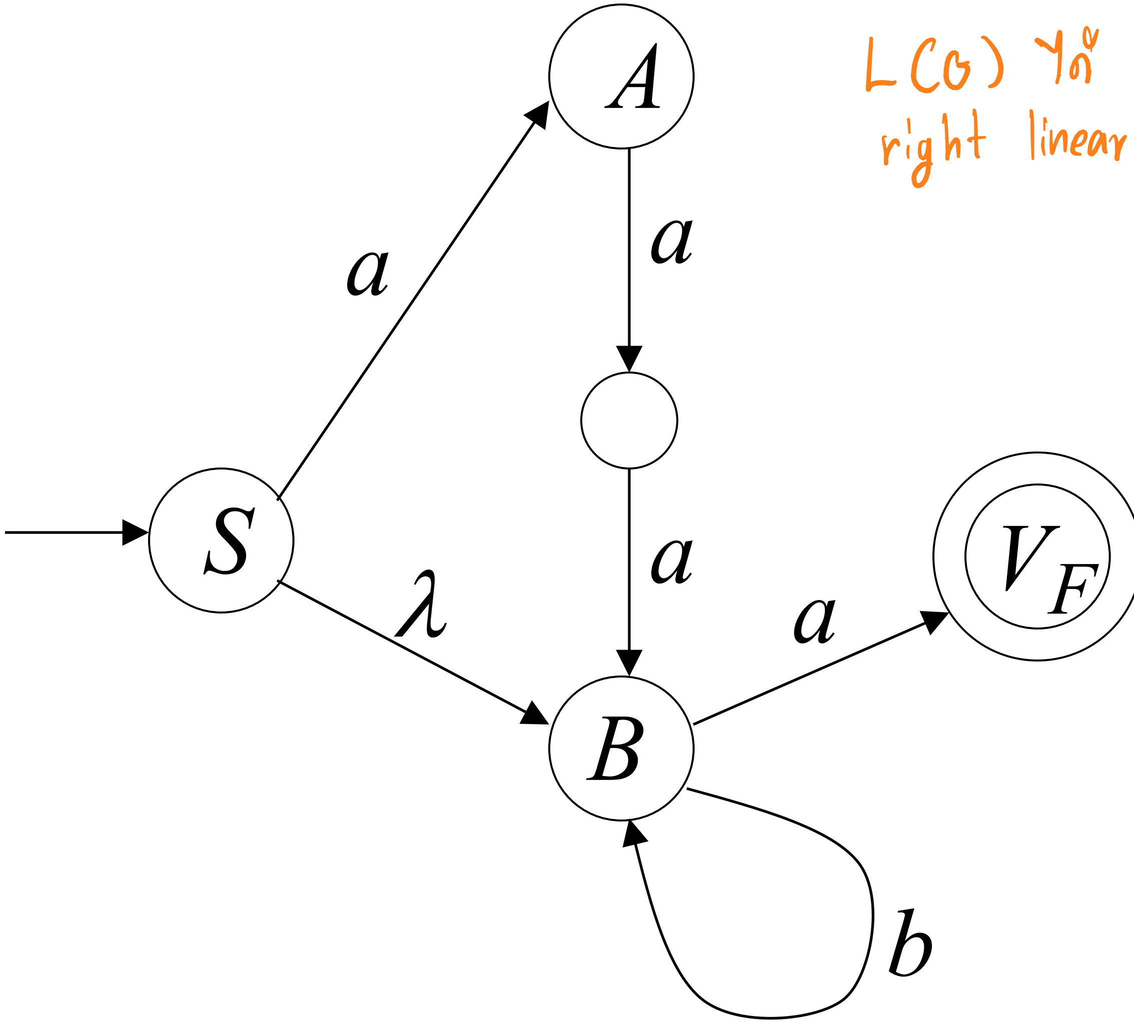
$$S \to aA \mid B$$

$$A \to aa\ B$$

ไม่ varible ตามมา → จะไปที่ final state

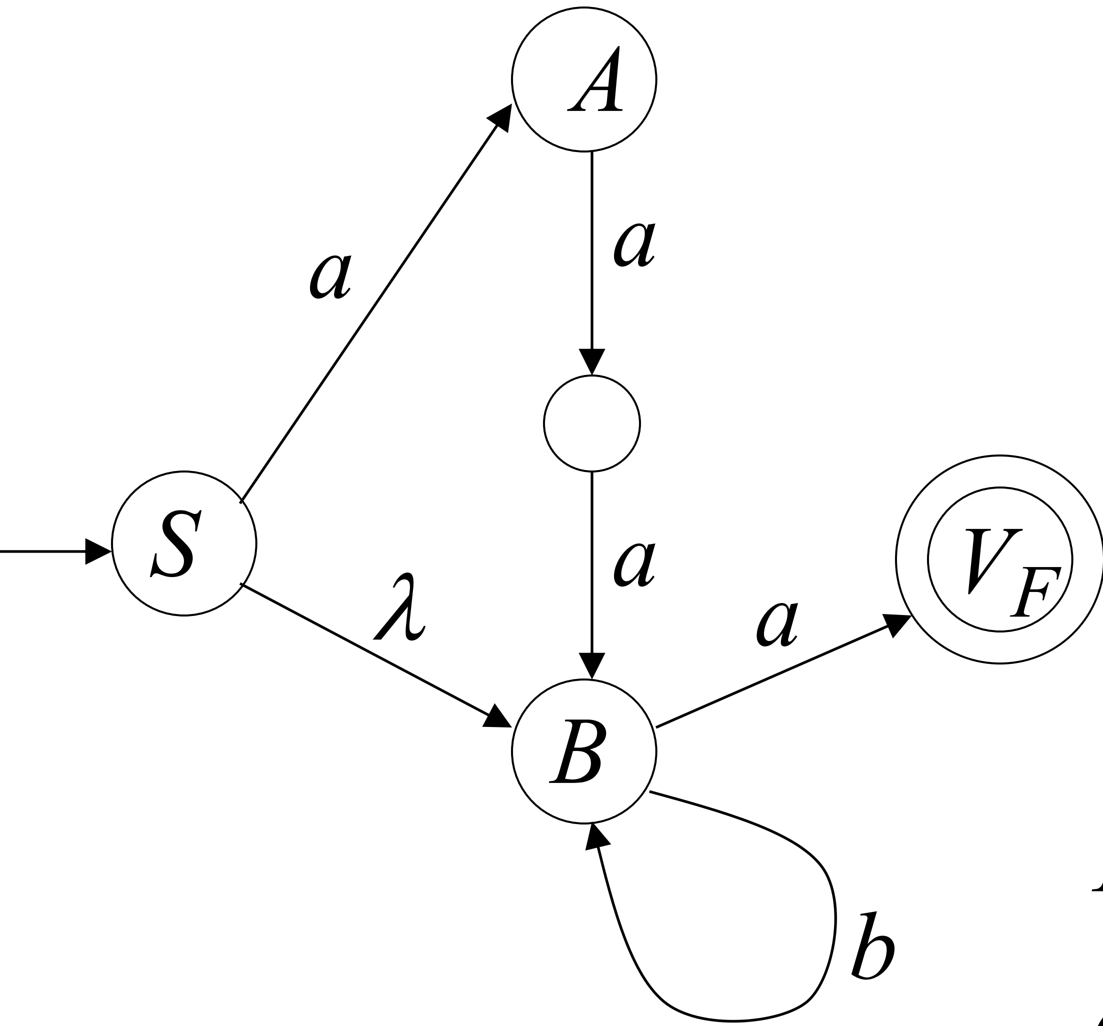
$$B \to bB \mid a$$

NFA ที่สร้างมาจาก
L(G) ได้ = มันเป็น Reg Lang.
right linear gamma



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

NFA M



Grammar

G

$$S \to aA \mid B$$

$$A \to aa\,B$$

$$B \to bB \mid a$$

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

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

In General

เขียนเอง

A right-linear grammar G

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

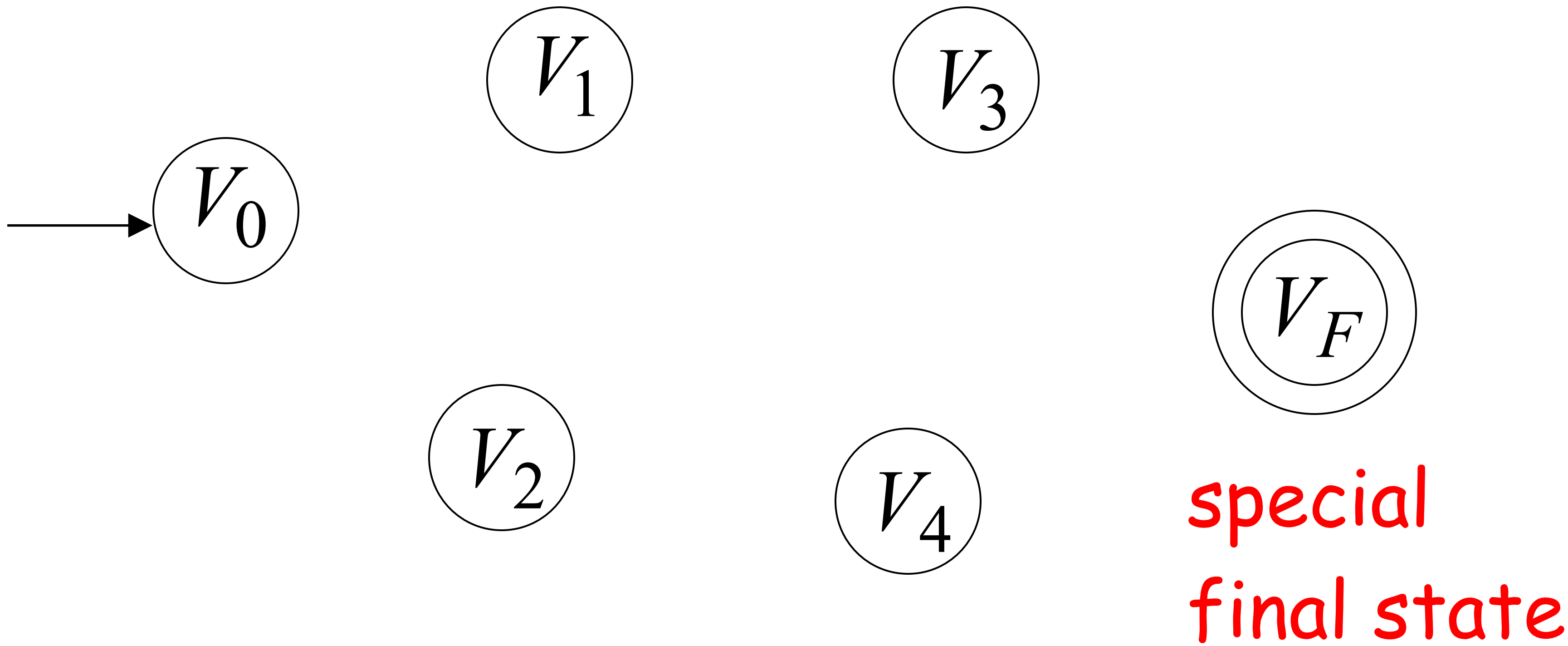
and productions: $V_i \to a_1 a_2 \cdots a_m V_j$

or

$V_i \to a_1 a_2 \cdots a_m$

We construct the NFA M such that:

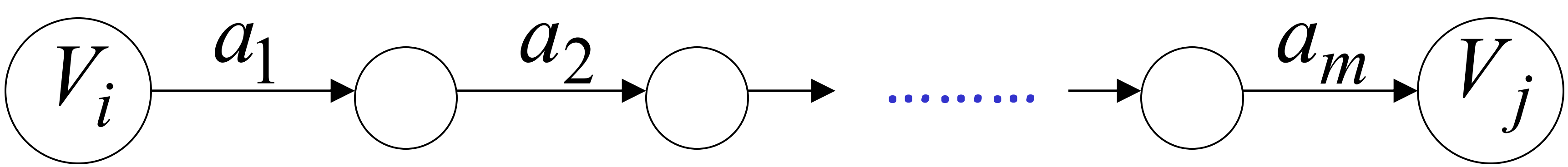
each variable V_i corresponds to a node:



JO

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

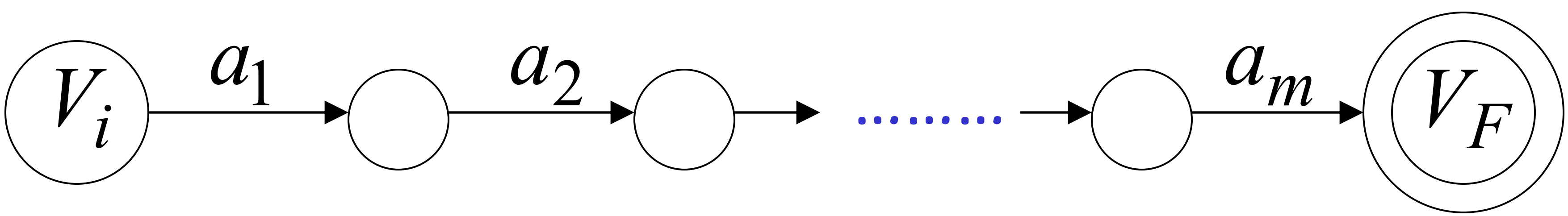
we add transitions and intermediate nodes



50

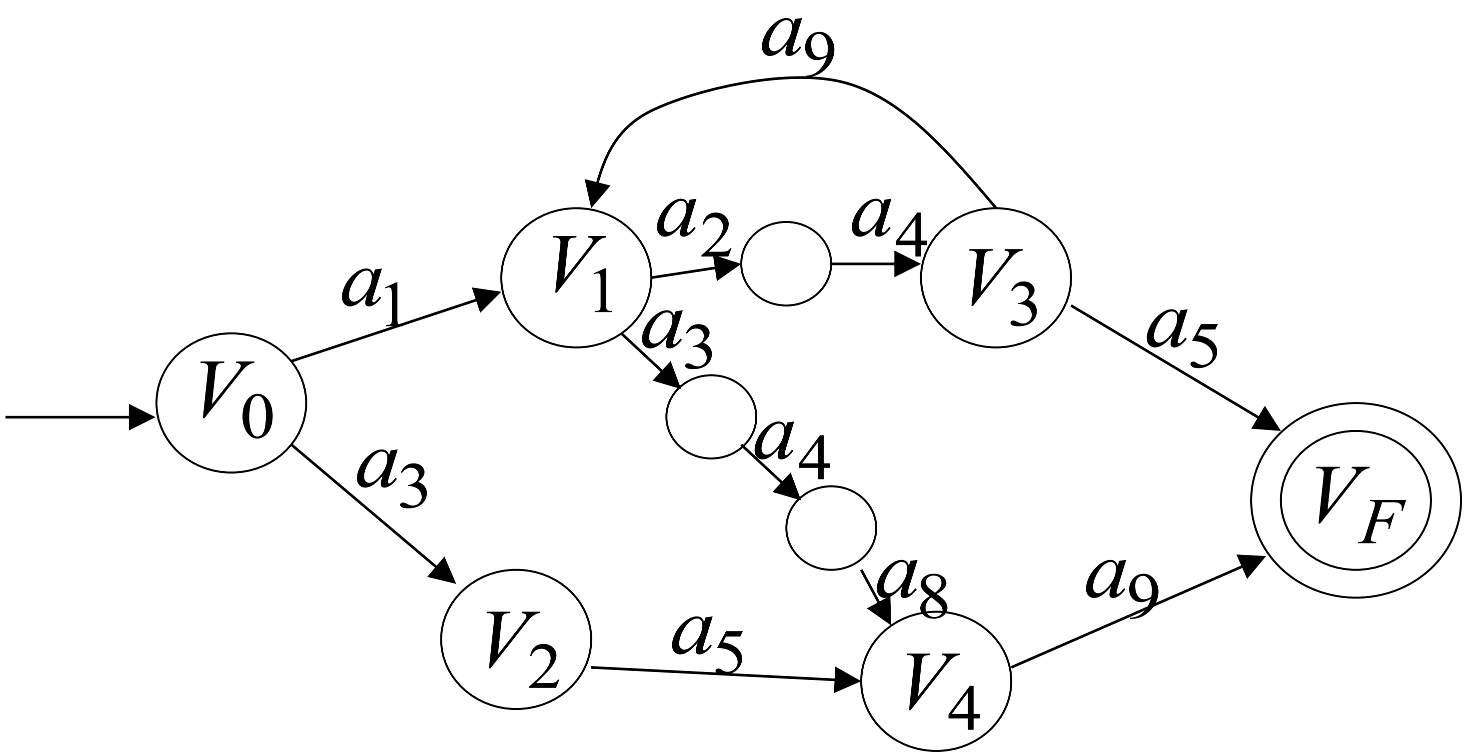
For each production: $V_i \to a_1 a_2 \cdots a_m$

we add transitions and intermediate nodes



50

Resulting NFA M looks like this:



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

พิสูจน์

The case of Left-Linear Grammars

แปลงแบบ righ → NFA ไม่ได้ เพราะ Variable อยู่หน้าสุด
ซึ่งทำ NFA = มีปัญหาแน่นอน

Let G be a left-linear grammar

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

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

ได้

ถ้า reverse

Proof idea:

สร้าง

We will construct a right-linear คือ G'

grammar G' with $L(\overset{\text{left}}{G}) = L(\overset{\text{right}}{G'})^R$

มัน คือ Reg. Lang.
พิสูจน์แล้ว

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

$$A \to Ba_1a_2\cdots a_k$$

$$A \to a_1a_2\cdots a_k$$

Construct right-linear grammar G'

Left linear G

$$A \to Ba_1a_2 \cdots a_k$$

$$A \to Bv$$

$$\Downarrow$$

Right linear G'

$$A \to a_k \cdots a_2a_1B$$

$$A \to v^R B$$

กรณีไม่มี Variable

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:

left linear gamma G

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

Regular Language

reverse

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:

เริ่ม

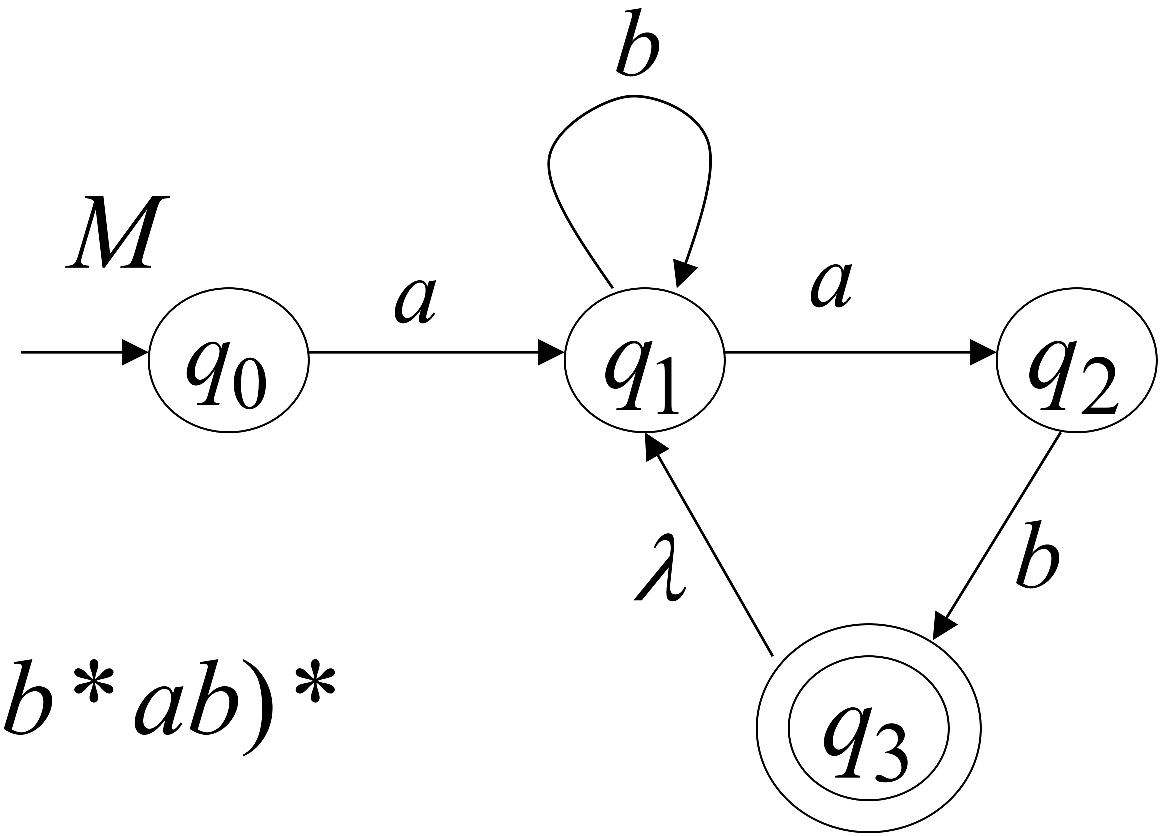
Let M be the NFA with $L = L(M)$.

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

Since L is regular

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

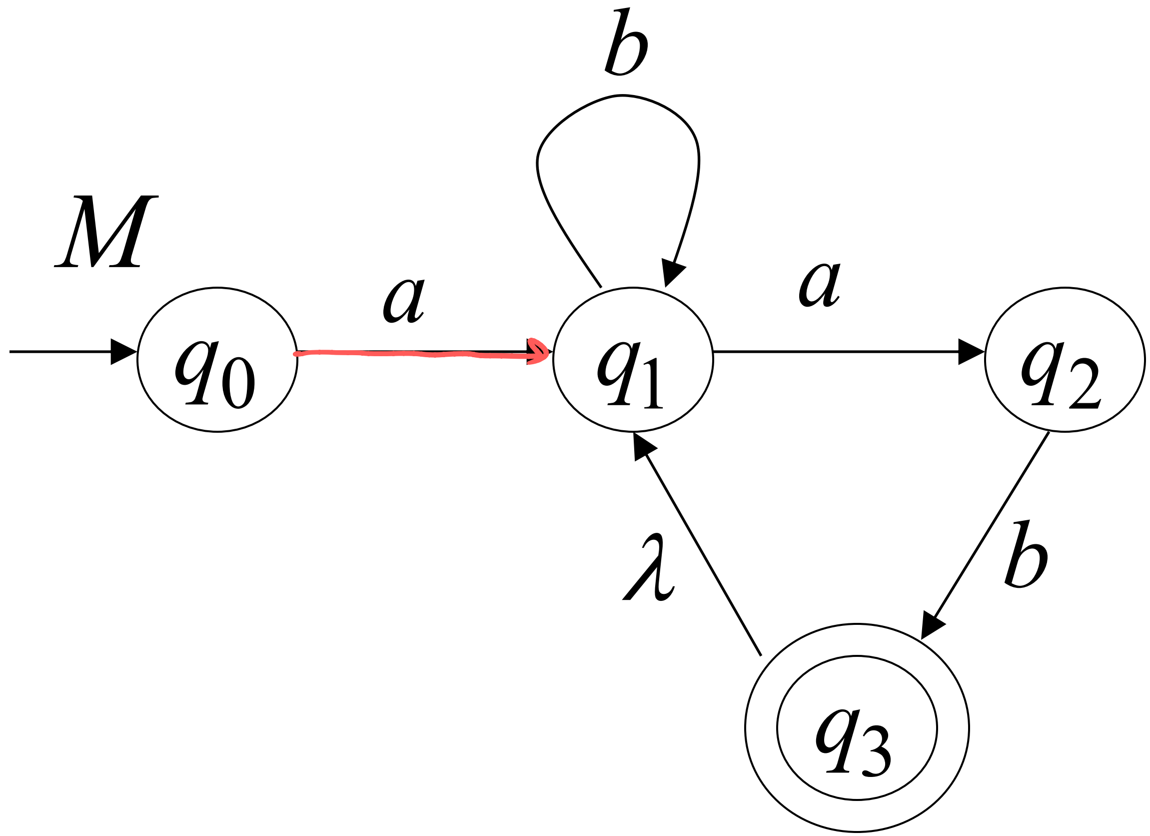
Example:



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

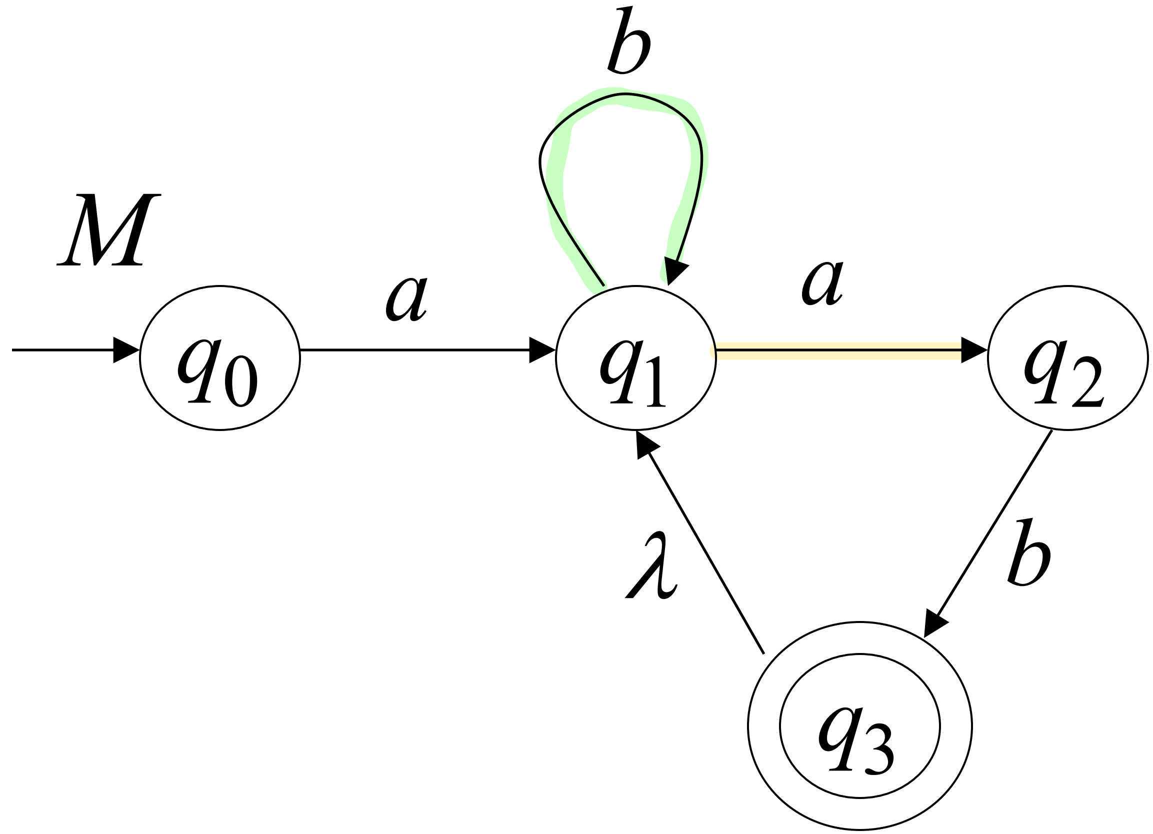
$$L = L(M)$$

Convert M to a right-linear grammar



Derive ทีฟ

$$q_0 \to aq_1$$



$$q_0 \to aq_1$$

$$q_1 \to bq_1$$

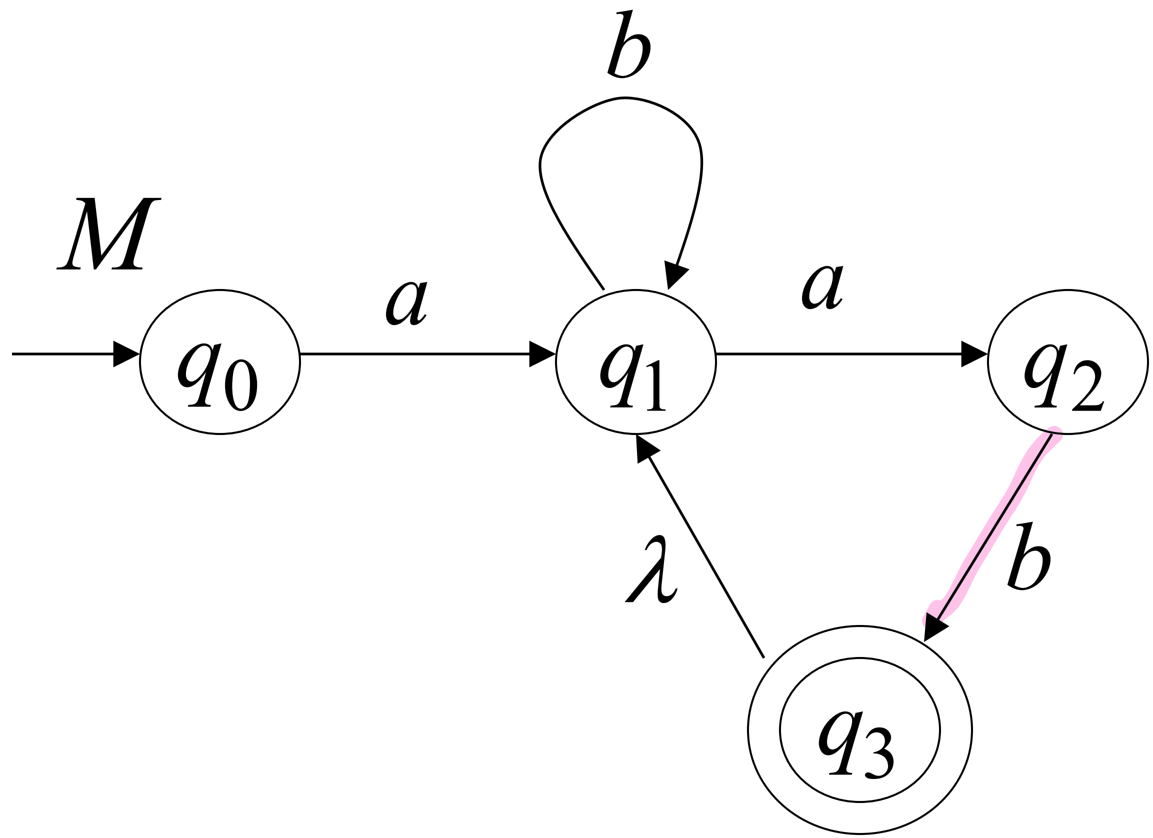
$$q_1 \to aq_2$$

$$q_0 \to aq_1$$

$$q_1 \to bq_1$$

$$q_1 \to aq_2$$

$$q_2 \to bq_3$$



$$L(\underbrace{G}_{\text{right linear Gamma}}) = L(M) = L$$

G

$q_0 \to aq_1$

$q_1 \to bq_1$

$q_1 \to aq_2$

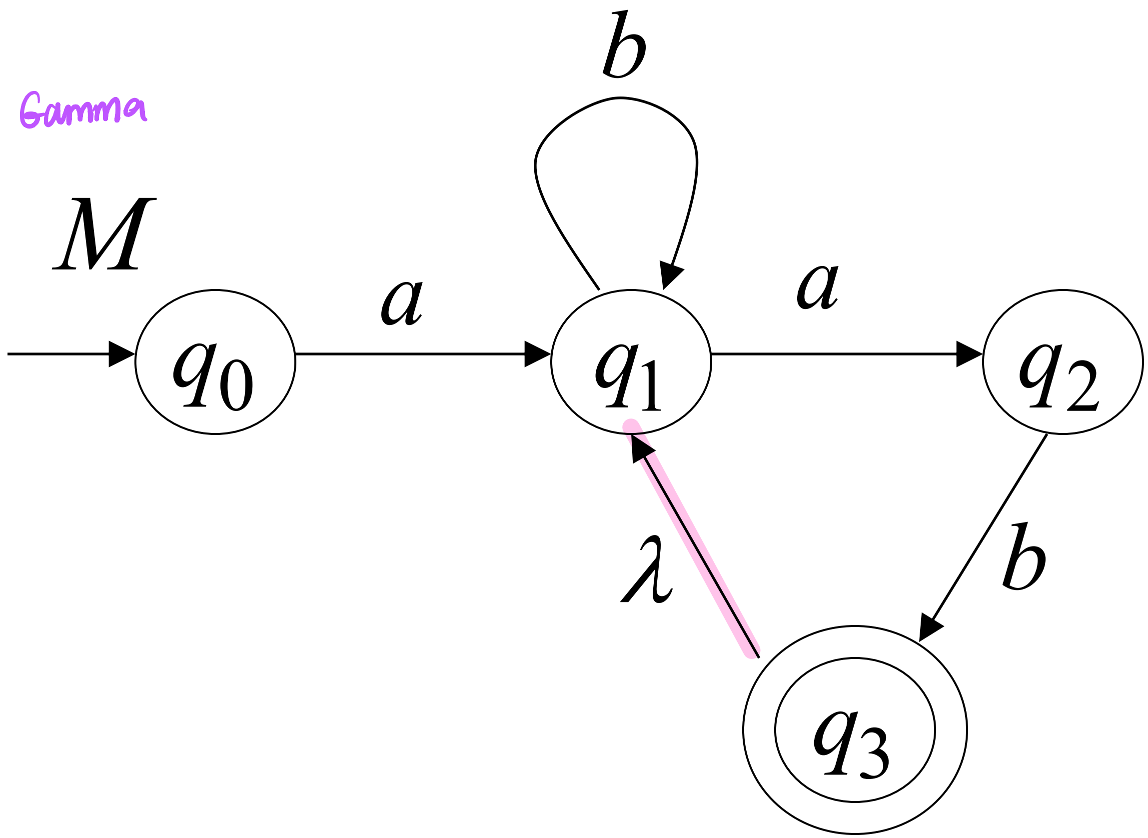
$q_2 \to bq_3$

$q_3 \to q_1$

$q_3 \to \lambda$

เพราะ q_3 คือ final state

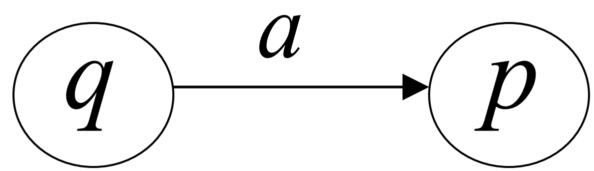
เลยเขียนมาวกตัดทิ้งได้เลย เพราะ



In General

10

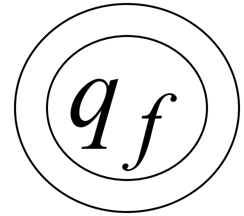
For any transition:



Add production:

$$q \to ap$$

variable terminal variable

For any final state: 

Add production: $q_f \to \lambda$

50

Since G is right-linear grammar

G is also a regular grammar

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