3.3 Regular languages and grammars [Linz §3.3]

Q. What is the relation between **regular** languages and languages generated by **grammars**?

A. We will see: Every **regular** language can be generated by a **grammar**, but **not conversely**!

To get an equivalence, we need a **special kind** of grammar.

Definition:

(1) A grammar $G = (V, \Sigma, S, P)$ is **right-linear (RL)** if all its productions are of the form:

$$A \longrightarrow xB$$

$$A \longrightarrow x$$

where $x \in \Sigma^*$.

(2) G is **left-linear (LL)** if all its productions are of the form:

$$A \longrightarrow Bx$$

$$A \longrightarrow x$$

(3) G is **linear** if it is either **RL** or **LL**.

Note: Linz (Def. 3.3) calls such grammars “regular”.

[L, Theorem 3.3]

Theorem : Let G be an **RLG**.

Then $L(G)$ is **regular**.

Proof : Given an RLG G, we will show how to **construct** an **NFA** M that accepts $L(G)$.

M is constructed as follows,

Non-terminals represent **nodes**,

Terminals represent **edges**.

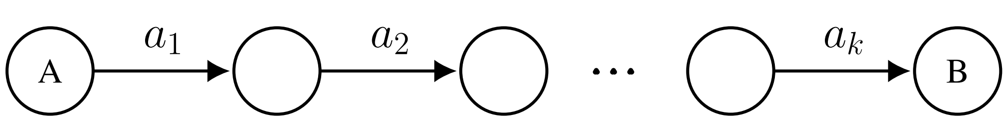
(Start with production $S \longrightarrow$ ...)

Take a production $A \longrightarrow$ ...

where $A = S$, or you already have a node labeled A,

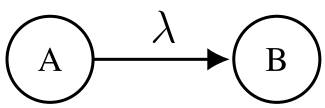
<u>Case 1</u> Production is $\boldsymbol{A \longrightarrow a_1 \dots a_k B}$ $(k \geq 1)$

Construct:



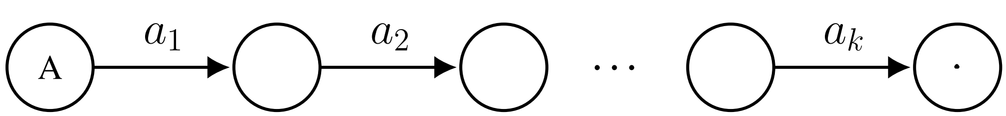
Case 2 Production is $\boldsymbol{A \longrightarrow B}$.

Construct:



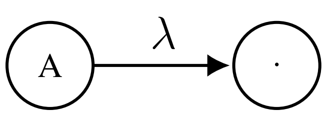
Case 3 Production is $\boldsymbol{A \longrightarrow a_1 \ldots a_k}$ $(k \geq 1)$

Construct:



Case 4 Production is $\boldsymbol{A \longrightarrow \lambda}$

Construct:



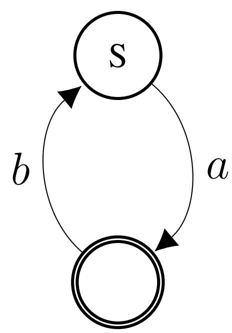
OR make (A) a final node.

Note:
If production is $A \longrightarrow xB$, where you have already encountered B, i.e. there already is a node B, then **loop back** to node B.

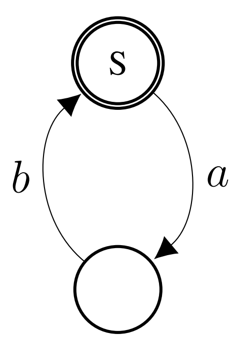
In this way, you construct an **nfa** $\boldsymbol{M}$ s.t. $\boldsymbol{L(M) = L(G)}$.

Examples: Construct **nfa's** from RLG's ($\Sigma = \{a, b\}$)

1) $\boldsymbol{S \longrightarrow abS \mid a}$



2) $\boldsymbol{S \longrightarrow abS \mid \lambda}$



Conversely, [L, Theorem 3.4]

Theorem : For any **nfa** M, we can construct an RLG G s.t.

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

.

Proof : (outline)

Again, the **non-terminals** of G are the **nodes** of M, and the **terminals** of G are the **edges** of M.

Cases

– For each edge $A \xrightarrow{a} B$ of M,

add to G the prod. $A \longrightarrow aB$.

– For each edge $A \xrightarrow{\lambda} B$ of M,

add to G the prod. $A \longrightarrow B$.

– For each **final node** in M, add the production $A \longrightarrow \lambda$. $\square$

Corollary : For each **regular expression** r, $\exists$ RLG G s.t.

$$L(G) = L(r).$$

Proof : **Combine** Theorems on p. 3-7 and above. $\square$