Grammars

A grammar defines a language

could be regular or not $\qquad a^n b^n$

A grammar for english sentences like

I love mom

You want candy $\qquad$ structure is NVO

we see candy

$S \to NVO$

$N \to I \mid you \mid we$

$V \to love \mid want \mid see$

$O \to mom \mid candy$

Starting with S, we can generate a sentence by substitutions:

$S \Rightarrow NVO \Rightarrow I\ VO \Rightarrow I\ \text{want}\ O \Rightarrow I\ \text{want mom.}$

Fancier: $S \to NVO \mid NVMO$

$N \to I \mid you \mid we$

$V \to love \mid want \mid see$

$O \to mom \mid candy$

$M \to my \mid a \mid the$

This allows "I see my mom"

More abstract:

$$S \to aS \mid X$$
$$X \to bX \mid \varepsilon$$

To build a string, start with "the starting symbol" S, then do substitutions with the $\to$

"the productions"

This grammar can generate aaabb

A derivation:

$$S \to aS \mid X$$
$$X \to bX \mid \varepsilon$$

$$S \Rightarrow aS \Rightarrow aaS \Rightarrow aaaS \Rightarrow aaaX$$
$$\Rightarrow aaabX \Rightarrow aaabbX \Rightarrow aaabb\varepsilon = aaabb$$

The set of all strings which can be obtained by derivations from the grammar G is called the language generated by G,

written $L(G)$

For

$$G: \begin{array}{l} S \to aS \mid X \\ X \to bX \mid \varepsilon \end{array}, \quad L(G) = \{a^n b^m\}$$

$$G: \quad S \to aSb \mid \varepsilon$$

we can do:

$$S \Rightarrow aSb \Rightarrow a\varepsilon b = ab$$

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb = a^2 b^2$$

Here, $L(G) = \{a^n b^n\}$ (nonregular)

---

fancier:

$$G: \begin{array}{l} S \to aSb \mid \varepsilon \\ aS \to Sa \end{array}$$

now we can make abab

$$S \Rightarrow aSb \Rightarrow Sab \Rightarrow aSbab \Rightarrow abab$$

$L(G)$ is hard to describe as a set.

Formally:

Terms: Alphabet letters (a, b's) in the strings of $L(G)$ are called the terminal symbols

The other ones (S, X, etc) are nonterminal symbols.

Def A grammar is a tuple $G = (V, \Sigma, S, P)$

where: V is a finite set of symbols (the nonterminal alphabet)

Σ is a finite set of symbols (the terminal alphabet)

$S \in V$ (the starting symbol)

P is a finite set of productions

each production must look like

$x \to y$ where x & y are any strings over $V \cup \Sigma$,

and $x \neq \varepsilon$.

$$G: \quad \begin{aligned} S &\to aS \mid X \\ X &\to bX \mid \varepsilon \end{aligned}$$

Write the formal description of G:

$$G = \Big(\{S, X\}, \{a, b\}, S, \{ S \to aS, S \to X, X \to bX, X \to \varepsilon \} \Big)$$

For strings x, y on $V \cup \Sigma$, we write

$x \Rightarrow y$ when we can obtain y from x by doing a single production.

We write $x \Rightarrow^* y$ when y can be obtainde from x by doing several productions.

<u>Def</u> The <u>language</u> <u>of</u> <u>G</u> $= (V, \Sigma, S, P)$ is:

$$L(G) = \{ x \in \Sigma^* \mid S \Rightarrow^* x \}$$

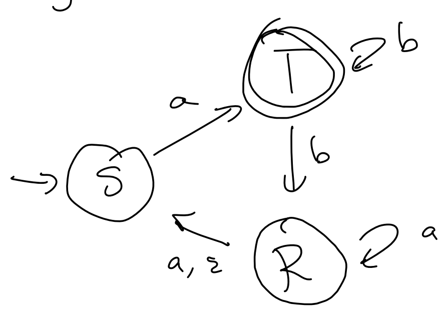
How do langs of grammars compare with langs of NFA / DFA?

Can any grammar lang be generated by NFA? NO

$a^n b^n$ has a grammar, but no NFA.

Can any NFA lang. be generated by grammar? Yes!

Any NFA can converted to a grammar



Each state becomes a nonterminal

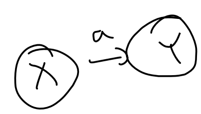
Start state is S.

$S \to aT$

$T \to bT \mid bR \mid \varepsilon$

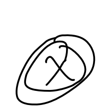
$R \to aR \mid aS \mid S$

each NFA arrow like



becomes $X \to aY$

and any NFA accepting state



makes

$X \to \varepsilon$