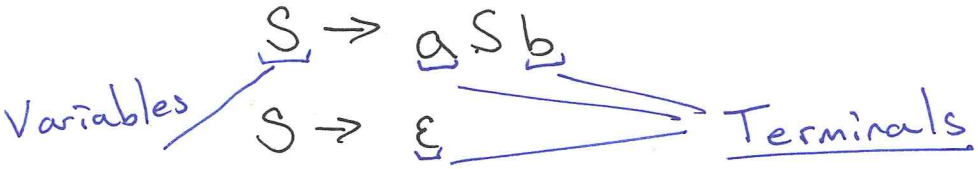
Context-Free Grammars

$$A = \{ a^n b^n \mid n \geq 0 \}$$

A grammar is a set of substitution rules, called productions



The terminals are the elements of the alphabet Σ

Grammars are used to describe a language by generating all strings of the language.

1. Write down the start variable
   - left-hand side of the first rule unless otherwise specified.
2. Find a variable and a rule that starts with that variable. Replace that variable with the right hand side of the rule.
3. Repeat step 2 until no variables remain

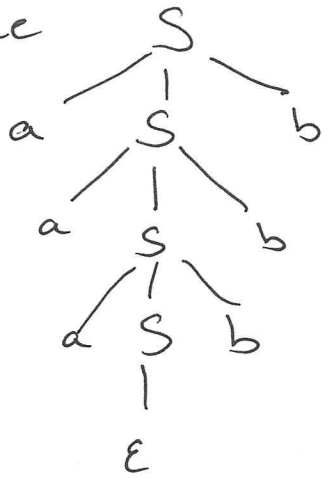
This sequence of substitutions to obtain a string is called a derivation

$w = aaabbb$

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

yields

Parse Tree



All strings generated in this way constitute the language of the grammar

$L(G)$

$S \rightarrow aSb \mid \epsilon$

Formal Definition of a Context-Free Grammar

$G = (V, \Sigma, R, S)$

V = a finite set of variables

Σ = a finite set of terminals

R = a finite set of rules

$S \in V$ is the start variable

$\Rightarrow$ yields (one step)

$\overset{*}{\Rightarrow}$ derives (one or more steps)

$\{w \mid w \text{ is a palindrome}\}$ $\quad \Sigma = \{a, b\}$

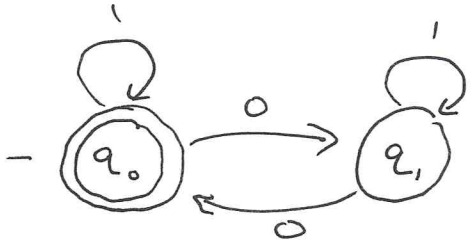
$A \rightarrow aAa$

$A \rightarrow bAb$

$A \rightarrow a \mid b \mid \varepsilon$

$S \rightarrow (S) \mid SS \mid \varepsilon$

$\{w \mid w$ is a set of properly nested parentheses$\}$



$\{w \mid w$ contains an even number of 0s$\}$

$A \rightarrow 0B$

$A \rightarrow 1A$

$B \rightarrow 0A$

$B \rightarrow 1B$

$A \rightarrow \varepsilon$

Context-free grammars that generate Regular Languages

A context-free language is any language that can be generated with a context-free grammar.

Theorem

The class of context-free languages are closed under union.

Proof by construction:

Let A and B be context-free languages.

Let $G_1 = (V_1, \Sigma, R_1, S_1)$ s.t. $L(G_1) = A$

Let $G_2 = (V_2, \Sigma, R_2, S_2)$ s.t. $L(G_2) = B$

Construct G to ~~recognize~~ generate $A \cup B$

$$G = (V, \Sigma, R, S)$$

$$V = V_1 \cup V_2 \cup \{S\}$$

$$R = R_1 \cup R_2 \cup \{ S \to S_1 \mid S_2 \}$$

$A = \{ a^n b^n \mid n \geq 0 \}$

$B = \{ a^{2n} \mid n \geq 0 \}$

G_1 $\quad S_1 \rightarrow a S_1 b \mid \varepsilon$

G_2 $\quad S_2 \rightarrow aa S_2 \mid \varepsilon$

G $\quad S \rightarrow S_1 \mid S_2$

Does this help us to find if context-free languages are closed under intersection?

No

Theorem

The class of context-free languages is closed under concatenation. For any context-free languages A, B $A \circ B$ is context-free.

Proof by construction

$$S \to AB$$

$$A \to aAb$$
$$A \to \varepsilon$$

$$B \to aBb \mid bBa \mid BB \mid \varepsilon$$

Let $G_1 = (V_1, \Sigma, R_1, S_1)$ and $L(G_1) = A$

Let $G_2 = (V_2, \Sigma, R_2, S_2)$ and $L(G_2) = B$

Construct $G = (V, \Sigma, R, S)$ to generate $A \circ B$

$$V = V_1 \cup V_2 \cup \{S\}$$

$$R = R_1 \cup R_2 \cup \{S \to S_1 S_2\}$$

Theorem

The class of context-free languages are closed under Kleene Star. If A is context-free then A^* is context-free.

Proof by Construction

$\{a^n b^n \mid n \geq 0\}^*$ $\qquad S \rightarrow aSb \mid \varepsilon$

new start state $\rightarrow T \rightarrow ST \mid \varepsilon$

Let $G = (V, \Sigma, R, S)$ where $L(G) = A$

Construct $G' = (V', \Sigma, R', S')$ to generate A^*

$$V' = V \cup \{S'\}$$

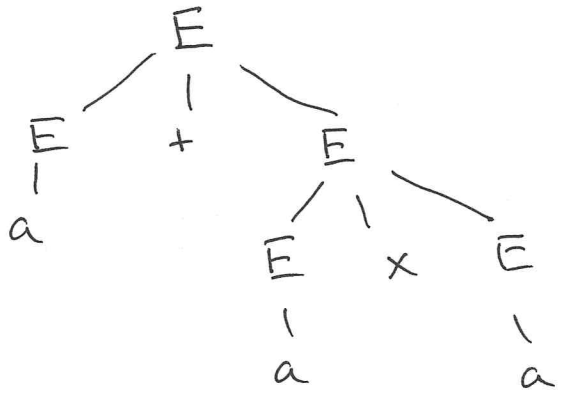
$$R' = R \cup \{S' \rightarrow SS', \; S' \rightarrow \varepsilon\}$$

Ambiguity

$$E \rightarrow E+E \mid E-E \mid$$
$$E \times E \mid E \div E \mid$$
$$(E) \mid a$$

$s = a + a \times a$

$E \Rightarrow E+E \Rightarrow a+E \Rightarrow a+E \times E \Rightarrow a+a \times E$
$\Rightarrow a+a \times a$



$S = a + a \times a$

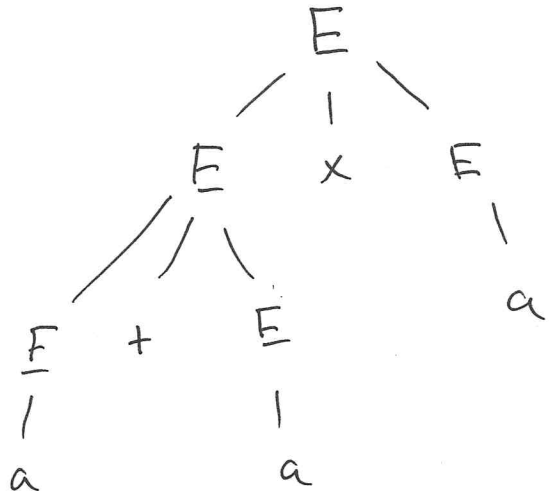
$E \Rightarrow E \times E$

$\Rightarrow E + E \times E$

$\Rightarrow a + E \times E$

$\Rightarrow a + a \times E$

$\Rightarrow a + a \times a$



Def: If a string s can be derived in several different ways then that string is derived ambiguously in that grammar.

If a grammar generates some string ambiguously then that grammar is ambiguous

Why is Ambiguity a Problem?

Why didn't we discuss ambiguity with regards to regular languages?

- with regular languages we don't really care about assigning internal structure to the strings.

Why are context-free languages different?

- The internal structure is often important with a string from a context-free language. And when we want to assign a meaning to a string, we almost always want to assign a unique meaning.

Changing EXPR grammar to take into account precedence rules.

$$E \rightarrow E + T \mid E - T \mid T$$

$$T \rightarrow T \times F \mid T \div F \mid F$$

$$F \rightarrow (E) \mid a$$

$a + a \times a$

$E \Rightarrow E+T \Rightarrow T+T \Rightarrow F+T$

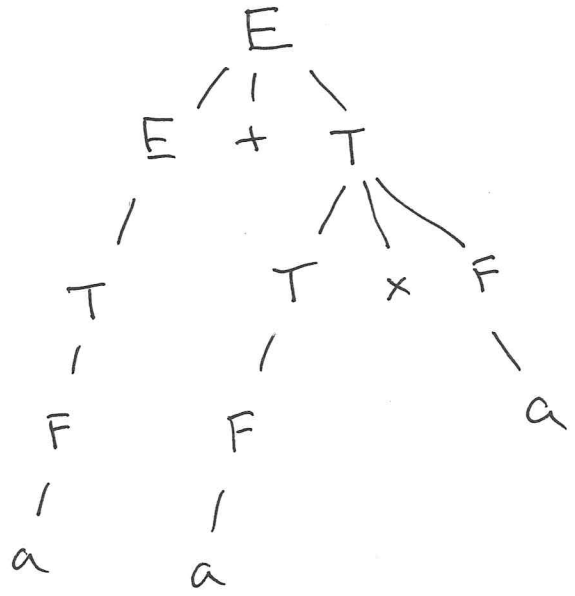
$\Rightarrow a+T$

$\Rightarrow a+T \times F$

$\Rightarrow a+F \times F$

$\Rightarrow a+a \times F$

$\Rightarrow a+a \times a$



Def

A context-free language that can be generated only by an ambiguous grammar is called inherently ambiguous.

$$\{a^i b^j c^k \mid i = j \text{ or } j = k\}$$

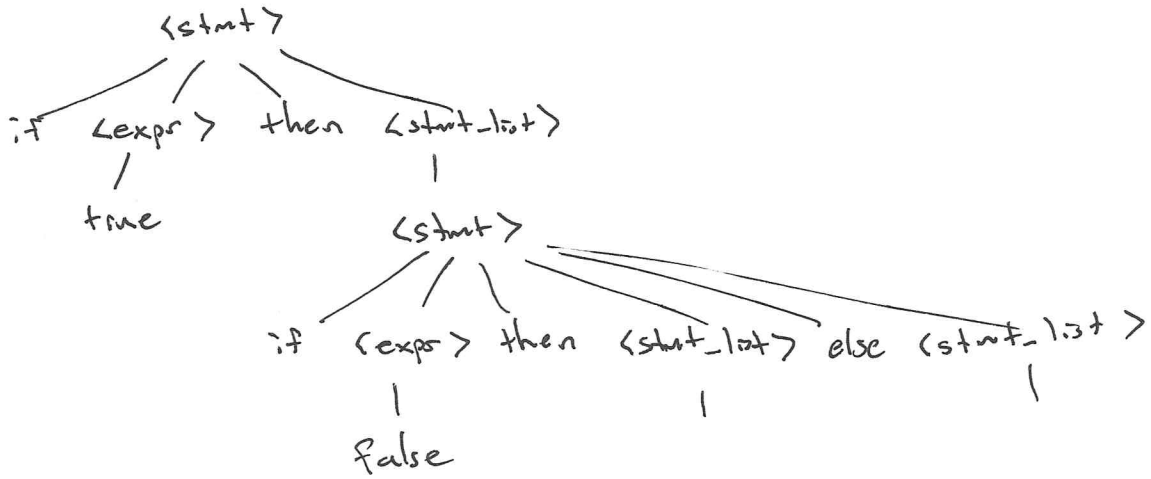
<expr> ::= true | false

<stmt> ::= if <expr> then <stmt_list>
| if <expr> then <stmt_list> else <stmt_list>

<stmt_list> ::= <stmt>
...

```
if true then
   if false then
else
```

```
if true then
if false then
else
```



Chomsky Normal Form

In a CNF grammar $G = (V, \Sigma, R, S)$ all rules have one of the following two forms.

- $X \to a$ where $a \in \Sigma$
- $X \to BC$ where $B, C \in V$

All parse trees generated from CNF grammars have a branching factor of two.

which makes the functionally binary trees.

Why is this useful?

- Parsers can exploit efficient data structures for storing and manipulating binary trees.
- Every derivation of a string w contains $|w| - 1$ applications of some rule of the form $X \to BC$ and $|w|$ applications of some rule of the form $X \to a$

This makes it much easier to define procedure to determine whether w can be generated by a CNF grammar G.

Chomsky Normal Form

In a CNF grammar all rules have one of the following two forms

$$X \to a$$

$$X \to BC \quad (B \text{ and } C \text{ cannot be the start variable})$$

If X is the start variable we also permit

$$X \to \varepsilon$$

Theorem

Any context-free language is generated by a context-free grammar in Chomsky normal form.

Proof idea

Convert an arbitrary grammar into Chomsky normal form.

Several stages

At each stage the grammar must generate the same language.

For some grammar G

1. Apply some transformation to G that eliminates undesireable property 1. Show that the language generated by G is unchanged.
2. Apply another transformation to G that eliminates undesireable property 2. Show that the language generated by G is unchanged and...

   that undesireable property 1 has not been reintroduced.
3. Continue until the grammar has the desired form.

Undesireable properties
→ add new start variable

1. ε rules ($A \to \varepsilon$ where $A \neq S$)
2. remove all unit rules
3. convert all remaining rules to the proper form.

   - remove mixed rules

Example 0

$S \to aSb$

$S \to \varepsilon$

① $S_0 \to S$

$S \to aSb$

$S \to \varepsilon$

② $S_0 \to S \mid \varepsilon$

$S \to aSb \mid ab$

③ $S_0 \to aSb \mid ab \mid \varepsilon$

$S \to aSb \mid ab$

④ $S_0 \to U_a S U_b \mid U_a U_b \mid \varepsilon$

$S \to U_a S U_b \mid U_a U_b$

$U_a \to a$

$U_b \to b$

$U_a T$

$T \to S U_b$

⑤ $S_0 \to U_a T \mid U_a U_b \mid \varepsilon$

$T \to S U_b$

$S \to U_a T \mid U_a U_b$

$U_a \to a$

$U_b \to b$

Example 3

$S \to RaRaR$

$R \to aR \mid bR \mid \varepsilon$

ε-rule removal

$S \to RaRaR \mid aRaR \mid RaRa \mid RaaR \mid Raa \mid aRa \mid aaR \mid aa$

$R \to aR \mid bR \mid a \mid b$

unit rules ✓

$S \to R U_a R U_a R \mid U_a R U_a R \mid R U_a R U_a \mid R U_a U_a R \mid R U_a U_a \mid U_a R U_a \mid U_a U_a R \mid U_a U_a$

$R \to U_a R \mid U_b R \mid a \mid b$

$U_a \to a$

$U_b \to b$

remove long

$S \to R S_1 \mid U_a S_2 \mid U_a U_b$

$S_1 \to U_a S_3 \mid U_a U_a$

$S_2 \to R S_4 \mid R U_a \mid U_a R$

$S_3 \to R S_5 \mid R U_a \mid U_a R$

$S_4 \to U_a R$

$S_5 \to U_a R$

$R \to U_a R$

$R \to U_b R \mid a \mid b$

$U_a \to a$

$U_b \to b$

Example 1

$A \to aBCa$

$B \to C \mid a$

$C \to D \mid b$

$D \to bD \mid \varepsilon$

add new start variable

$S \to A$

$A \to aBCa$

$B \to C \mid a$

$C \to D \mid b$

$D \to bD \mid \varepsilon$

remove ε-rules

$S \to A$

$A \to aBCa$

$B \to C \mid a$

$C \to D \mid b \mid \varepsilon$

$D \to bD \mid b$

$S \to A$

$A \to aBCa \mid aBa$

$B \to C \mid a \mid \varepsilon$

$C \to D \mid b$

$D \to bD \mid b$

$S \to A$

$A \to aBCa \mid aBa \mid aCa \mid aa$

$B \to C \mid a$

$C \to D \mid b$

$D \to bD \mid b$

remove unit rules

$S \to aBCa \mid aBa \mid aCa \mid aa$

$B \to D \mid b \mid a$

$C \to bD \mid b \mid b$

$D \to bD \mid b$

$S \to aBCa \mid aBa \mid aCa \mid aa$

$B \to bD \mid b \mid a$

$C \to bD \mid b$

$D \to bD \mid b$

remove mixed

$S \to TBCT \mid TBT \mid TCT \mid TT$

$B \to RD \mid b \mid a$

$C \to RD \mid b$

$D \to RD \mid b$

$T \to a$

$R \to b$

Remove long

$S \rightarrow TS_1 \mid TS_2 \mid TS_3 \mid TT$

$S_1 \rightarrow BS_4$ $\quad$ $S_4 \rightarrow CT$

$S_2 \rightarrow BT$ $\quad$ $S_3 \rightarrow CT$

$B \rightarrow RD \mid b \mid a$

$C \rightarrow RD \mid b$

$D \rightarrow RD \mid b$

$T \rightarrow a$

$R \rightarrow b$