

Ambiguity

Sometimes grammar can generate the same string in several different ways.

Such strings will have several different parse trees.

In some applications, this may be undesirable.

If a grammar generates a string in several different ways, we say that string is derived ambiguously in that grammar.

If grammar generates some string ambiguously, we say that the grammar is ambiguous.

Example:

$$\langle \text{Expr} \rangle \rightarrow \underbrace{\langle \text{Expr} \rangle + \langle \text{Expr} \rangle}_{\text{---}} \mid \underbrace{\langle \text{Expr} \rangle \times \langle \text{Expr} \rangle}_{\text{---}}$$
$$(\langle \text{Expr} \rangle) \mid \underline{a}$$

$$V = \{\langle \text{Expr} \rangle\} \quad \Sigma = \{a, (,), +, \times\}$$

$$S = \langle \text{Expr} \rangle$$

a+a×a try to generate this

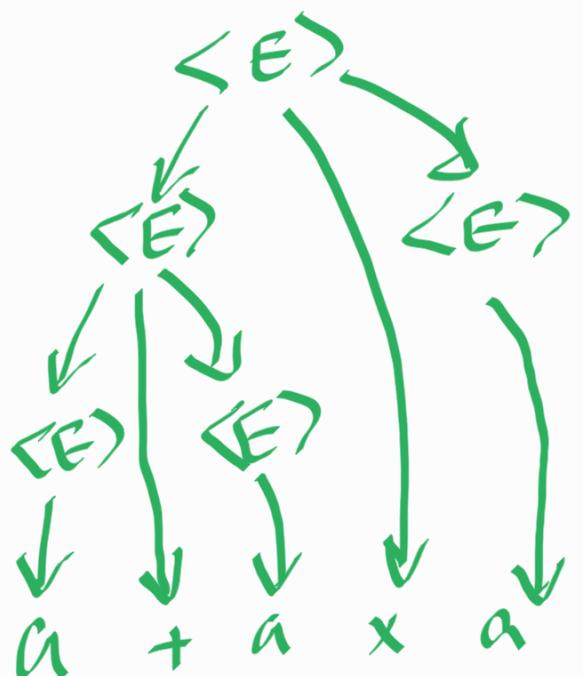
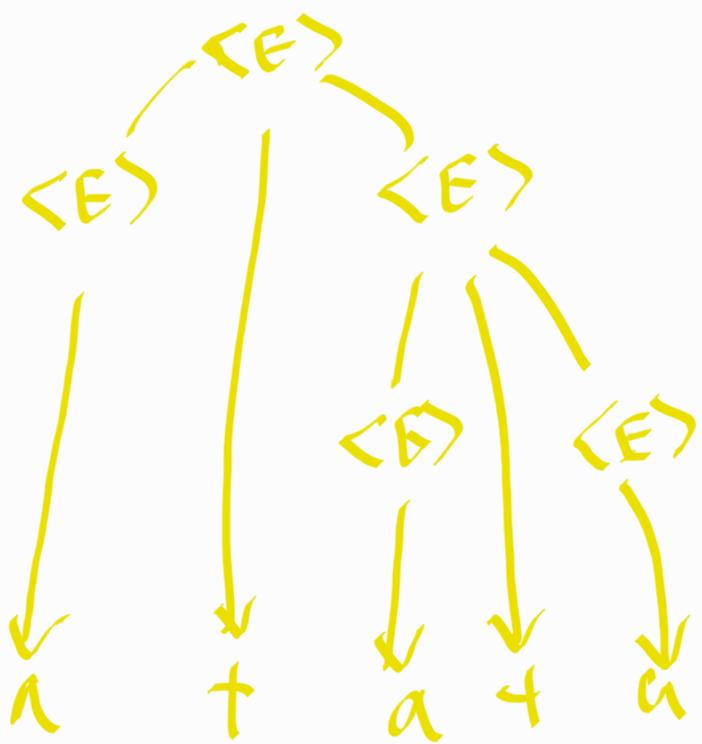
$$\langle E \rangle \Rightarrow \langle E \rangle + \underline{\langle E \rangle} \Rightarrow \langle E \rangle + \langle E \rangle \times \langle E \rangle$$

$$a + a \times a$$

$$\langle E \rangle \Rightarrow \underline{\langle E \rangle} \times \langle E \rangle \Rightarrow \langle E \rangle + \langle E \rangle \times \langle E \rangle$$

$$\Downarrow$$

$$a + a \times a$$



Chomsky Normal Form (CNF)

when working with CFG, it is often convenient to have them in simplified form.

Very useful when we have algorithms working on CFGs.

Definition

Chomsky Normal Form (CNF)

Context free grammar is in chomsky normal form if every rule is in the form of:

$$A \longrightarrow BC$$

$$A \longrightarrow a$$

a is a terminal, and A, B, C are any variable—except that B & C may not be start variable

we also allow $S \xrightarrow{\leftarrow} \epsilon$ (only for start variable)

Theorem 2.9

Any context-free language is generated by a context-free grammar in Chomsky Normal Form.

Proof' (By construction)

Instead going through detailed proof, we will go through an example to understand this.

$$S \rightarrow ASA \mid aB$$
$$A \rightarrow B \mid S$$
$$B \rightarrow b \mid \epsilon$$

Let's convert this to CNF_{II}

1. Add a new starting variable.

(This guarantees that start variable does not occur on the right hand side)

2. Take care of ϵ -rules.

we remove $A \rightarrow \epsilon$ rules where A is not the start variable.

for each occurrence of A on the right hand side of a rule, we add a new rule with that occurrence deleted.

Ex $R \rightarrow uAvAw$ is replaced with

* $R \rightarrow uAvAw \mid uvAw \mid uAvw \mid uw$

$R \rightarrow A$ $\Rightarrow R \rightarrow E$ (only if we did not remove this prev)

3. Handle all unit rules

(i) we remove unit rules $A \rightarrow B$

(ii) when we see $B \rightarrow u$, we add $A \rightarrow a$ unless we have already removed it earlier.

u is a string of variables and terminals.

4. Convert all remaining rules into proper form.

(i) we replace each rule in the form of

$A \rightarrow u_1 u_2 \dots u_k$, where

$k \geq 3$, and u_i is a variable or terminal symbol, with rules

$A \rightarrow u_1 A_1 \quad A_1 \rightarrow u_2 A_2 \dots \quad A_{k-2} \rightarrow u_{k-1} A_k$

We replace any terminal u_i in the preceding rule(s) with the new variable u_i and $u_i \rightarrow u_i$

$A \rightarrow u_1 u_2 u_3 \dots u_k$

$A \rightarrow u_1 A_1$

$\underbrace{A_1 \rightarrow u_2 u_3 u_4 \dots u_k}_{\hookrightarrow A_1 \rightarrow u_2 A_2}$

$A_2 \rightarrow u_3 u_4 \dots u_k$

Let's do an example

$$S \rightarrow ASA | aB$$

$$A \rightarrow B | S$$

$$B \rightarrow b | \epsilon$$

$S_0 \rightarrow S$ Adding a new starting variable.

$$S \rightarrow ASA | a\underline{B}$$

$$A \rightarrow \underline{B} | S$$

$$B \rightarrow b | \epsilon$$

removing $B \rightarrow \epsilon$

$$S_0 \rightarrow S$$

$$S \rightarrow ASA | aB | a$$

$$A \rightarrow B | S | \cancel{\epsilon}$$

$$B \rightarrow b$$

removing $A \rightarrow \epsilon$

$s_0 \rightarrow S$ $S \rightarrow ASA | SA | AS | S | aB | a$ $A \rightarrow \cancel{B} | S$ $B \rightarrow b$

removing $A \rightarrow B$

 $s_0 \rightarrow S$ $S \rightarrow ASA | SA | AS | S | aB | a$ $A \rightarrow \cancel{S} | b$ $B \rightarrow b$

remove $A \rightarrow S$

 $s_0 \rightarrow S$ $S \rightarrow ASA | SA | AS | S | aB | a$ $A \rightarrow b | ASA | SA | AS | aB | a$ $B \rightarrow b$

remove $s \rightarrow s$

$s_0 \rightarrow s$

$s \rightarrow ASA | SA | AS | aB | a$

$A \rightarrow b | ASA | SA | AS | aB | a$

$B \rightarrow b$

$s_0 \rightarrow \underline{ASA} | SA | AS | aB | a$

$s \rightarrow ASA | SA | AS | aB | a$

$A \rightarrow b | ASA | SA | AS | aB | a$

$B \rightarrow b$

$s_0 \rightarrow AU_1 | SA | AS | aB | a$

$U_1 \rightarrow SA$

$s \rightarrow \underline{ASA} | SA | AS | aB | a$

$A \rightarrow b | ASA | SA | AS | aB | a$

$B \rightarrow b$

$s_0 \rightarrow A u_1 | sA | As | aB | a$

$u_1 \rightarrow SA$

$s \rightarrow A u_2 | sA | As | aB | a$

$u_2 \rightarrow SA$

$A \rightarrow b | AsA | sA | As | aB | a$

$b \rightarrow b$

Context free grammar

"context free" means that the application of a production rule depends only on the non-terminal being replaced, not on its surrounding

$$\alpha A \beta \rightarrow \alpha \gamma \beta$$

A can only
be replaced
by γ only if
it appears between
a α & β .