WEEK 10: LECTURE NOTES

Normal Forms for CFGs

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

- productions in G satisfy certain restrictions

$$A \to \alpha, \; A \in V, \; \alpha \in (V \cup T)^*$$

Two of them
- Chomsky Normal form (CNF)
- Greibach Normal form (GNF)

Chomsky Normal form

(Restrictions on the length of RHS and nature of symbols in RHS of production).

A CFG G is in CNF, if every production is of the form $A \to a$ or $A \to BC$ and $S \to \varepsilon$ is in G if $\varepsilon \in L(G)$. When $\varepsilon \in L(G)$ we assume that S doesnot appear on the R.H.S. of any production

Example: $S \to AB \mid \varepsilon$, $A \to a$, $B \to b$

- For a CFG in CNF, the parse tree has the following property :- Every node has at most two descendants,
  - either two internal vertices or a single leaf

Reduction to CNF

Example: $S \to ABC \mid aC$, $A \to a$, $B \to b$, $C \to c$

- $S \to ABC$, is not in the required form
  Replace them by $S \to AE$, $E \to BC$
- $S \to aC$ is not in the required form
  Replace them by $S \to DC$, $D \to a$

Equivalent CFG

$S \to AE$, $E \to BC$, $S \to DC$

$D \to a$, $A \to a$, $B \to b$

$C \to c$

Theorem (Reduction to CFN)

For every CFG, there is an equivalent grammar G_2 in CNF

Proof:

Step 1: Elimination of null productions and unit productions

(except for $S \to \varepsilon$, if $\varepsilon \in L(G)$ and S doesnot appear on R.H.S. of any productions)

Let $G = (V, T, P, S)$ be the resulting CFG

Step 2: Elimination of terminals on R.H.S.

We define $G_1 = (V', T, P', S')$, where P' and S' are constructed as follows:

i) All productions of the form $A \to a$ or $A \to BC$ are included in P'. All variables in V are included in V'

ii) Consider $A \to X_1 X_2 \cdots X_n$ with some terminal in R.H.S.. If X_i is a terminal, say a_i, add new variable C_{a_i} to V' and $C_{a_i} \to a_i$ in P'

In production $A \to X_1 X_2 \cdots X_n$, every terminal on the R.H.S. is replaced by the corresponding new variable and the variables on the R.H.S. are retained. The resulting production is added to P'

Thus we get $G_1 = (V', T, P', S')$

Step 3 : Restricting the number of variables on R.H.S.

For any production in P', the R.H.S. consists of either a single terminal (or ε, if $S \to \varepsilon$) or two or more variables.

Define $G_2 = (V'', T, P'', S)$ as follows:

i. All productions in P' are added to P'' if they are in the required form. All variables in V' are added to V''

ii. Consider $A \to A_1 A_2 \ldots A_n$ when $n \geq 3$, $A_i \in V'$.
We introduce new productions $A \to A_1 C_1$, $C_1 \to A_2 C_2, \ldots$ $C_{n-2} \to A_{n-1} A_n$ and new variables $C_1, C_2, \cdots C_{n-2}$
These are added to P'' and V'' respectively.

Then $G_2 = (V'', T, P'', S)$ is the required CNF.

Example: Reduce the following grammar G to CNF.

G is $S \to aAD$, $A \to aB \mid bAB$, $B \to b$, $D \to d$

Sol$\underline{n}$:

Step 1: No null productions or unit productions

Step 2: i. $B \to b$, $D \to d$ are included in P'

ii. $S \to aAD$ gives rise to $S \to C_a AD$, $C_a \to a$

$A \to aB$ gives rise to $A \to C_a B$,

$A \to bAB$ gives rise to $A \to C_b AB$, $C_b \to b$

$\therefore$ $G' = (V', T, P', S)$, $V' = \{S, A, B, D, C_a, C_b\}$

$$P' = \{S \to C_a AD,\ A \to C_a B \mid C_b AB,\ B \to b,\ D \to d,\ C_a \to a,\ C_b \to b\}$$

Step 3:

i. $A \to C_a B$, $B \to b$, $D \to d$, $C_a \to a$, $C_b \to b$, are included in P_2

ii. $S \to C_a AD$ is replaced by $S \to C_a C_1$, $C_1 \to AD$

$A \to C_b AB$ is replaced by $A \to C_b C_2$, $C_2 \to AB$

$G_2 = (\{S, A, B, D, C_a, C_b, C_1, C_2\}, \{a, b, d\}, P_2, S)$

where $P_2 = \{ S \to C_a C_1,\ A \to C_a B \mid C_b C_2,\ C_1 \to AD,\ C_2 \to AB,$
$B \to b,\ D \to d,\ C_a \to a,\ C_b \to b \}$

G_2 is in CNF and is equivalent to G.

Step 4: Claim: $L(G) = L(G_2)$

To complete the proof we have to show that

$$L(G) = L(G_1) = L(G_2)$$

To prove $L(G) \subseteq L(G_1)$

Let $w \in L(G)$. If $A \to X_1 X_2 \dots X_n$ is used in the derivation of w, the same effect can be achieved by using the corresponding productions in P' and the productions in the new variables.

Hence, $A \overset{*}{\underset{G_1}{\Rightarrow}} X_1 X_2 \dots X_n$.

Thus, $L(G) \subseteq L(G_1)$

To prove $L(G_1) \subseteq L(G)$

Let $w \in L(G_1)$

To show $w \in L(G)$, it is enough to prove

$$A \overset{*}{\underset{G}{\Rightarrow}} w \text{ if } A \in V,\ A \overset{*}{\underset{G_1}{\Rightarrow}} w \qquad (1)$$

We prove it by induction on the number of steps in

$A \overset{*}{\underset{G_1}{\Rightarrow}} w$

Base:

if $A \underset{G_1}{\Rightarrow} w$ then $A \to w$ is in P'

By construction of $\underline{P'}$, $\underline{w}$ is a single terminal.

so, $A \to w$ in P i.e. $A \underset{G}{\Rightarrow} w$

Induction step:

Let us assume (1) for derivations in at most k steps.

Let. $A \overset{k+1}{\underset{G_1}{\Rightarrow}} w$

We can split this derivation as

$$A \underset{G_1}{\Rightarrow} A_1 A_2 \cdots A_n \overset{k}{\underset{G_1}{\Rightarrow}} w_1 \cdots w_n = w$$

s.t. $A_i \overset{*}{\underset{G_1}{\Rightarrow}} w_i$

Now each A_i is i. either in V, or

ii. a new variable, say C_{a_i}

i. when $A_i \in V$, then $A_i \overset{*}{\underset{G_1}{\Rightarrow}} w_i$ is a derivation in at most k-steps, so by induction hypothesis,

$$A_i \overset{*}{\underset{G}{\Rightarrow}} w_i$$

ii. when $A_i = C_{a_i}$, the production $C_{a_i} \to a_i$ is applied to get $A_i \overset{*}{\underset{G_1}{\Rightarrow}} w_i$

The production $A \to A_1 A_2 \dots A_n$ in P' is induced by a production $A \to X_1 X_2 \dots X_n$ in P

where $\begin{cases} X_i = A_i & \text{if } A_i \in V \\ X_i = w_i & \text{if } A_i = C_{a_i} \end{cases}$

So, $A \underset{G}{\Rightarrow} X_1 X_2 \dots X_n \underset{G}{\overset{*}{\Rightarrow}} w_1 w_2 \dots w_n$

i.e. $A \underset{G}{\overset{*}{\Rightarrow}} w$

Thus (1) is true for all derivations

$\therefore \; L(G) = L(G_1)$

To prove $L(G_1) = L(G_2)$

The effect of applying $A \to A_1 A_2 \dots A_m$ in a derivation for $w \in L(G_1)$ can be achieved by applying the productions

$A \to A_1 C_1, \; C_1 \to A_2 C_2, \dots \; C_{m-2} \to A_{m-1} A_m$

Hence it is easy to say that $L(G_1) \subseteq L(G_2)$

To prove $L(G_2) \subseteq L(G_1)$, we can prove an auxiliary result

$A \underset{G_1}{\overset{*}{\Rightarrow}} w$ if $A \in V'$, $A \underset{G_2}{\overset{*}{\Rightarrow}} w$ ——— (2)

(2) can be proved by induction on the number of steps in $A \underset{G_2}{\overset{*}{\Rightarrow}} w$

Applying (1) to S, we get

$L(G_2) \subseteq L(G_1)$.

Example: Find a grammar in CNF equivalent to

$S \to aAbB,\ A \to aA \mid a,\ B \to bB \mid b$

Soln:

- No unit production, no null productions
- eliminate terminals on R.H.S.

Let $G_1 = (V_1, \{a, b\}, P_1, S)$ where V_1 and P_1 are constructed as follows:

i. $A \to a$, $B \to b$ are added to P_1

ii. $S \to aAbB$, $A \to aA$, $B \to bB$ yields

$S \to C_a A C_b B$, $A \to C_a A$, $B \to C_b B$, $C_a \to a$, $C_b \to b$

which are added to P_1

$V_1 = \{S, A, B, C_a, C_b\}$

- Restrict the number of variables on R.H.S.

P_1 consists of $S \to C_a A C_b B$, $A \to C_a A$, $B \to C_b B$, $C_a \to a$, $C_b \to b$, $A \to a$, $B \to b$

Let $G_2 = (V_2, \{a, b\}, P_2, S)$ where P_2 and V_2 are constructed as:

i. $S \to C_a A C_b B$ is replaced by

$S \to C_a C_1$, $C_1 \to A C_2$, $C_2 \to C_b B$

which are added to P_2.

ii. Remaining productions in P_1 are added to P_2

$V_2 = \{S, A, B, C_a, C_b, C_1, C_2\}$

Example: find a grammar in CNF equivalent to the grammar $S \to \sim S \mid [S \supset S] \mid p \mid q$

(S being the only variable)

Sol^{n}:

- No unit productions, no null productions
- Eliminate terminals on R.H.S.

Let $G_1 = (V_1, T, P, S)$, where P_1, V_1 are constructed as follows:

1. $S \to p$, $S \to q$ are added to P_1

ii. $S \to \sim S$ induced $S \to AS$, $A \to \sim$, which are added to P_1

iii. $S \to [S \supset S]$ induces

$S \to BSCSD$, $B \to [$, $C \to \supset$, $D \to]$ which are added to P_1

- Restrict the number of variables on R.H.S.

Let $G_2 = (V_2, T, P_2, S)$ where P_2 and V_2 are constructed as

- P_2 consists of

i. $S \to p \mid q \mid AS$, $A \to \sim$, $B \to [$, $C \to \supset$, $D \to]$

ii. $S \to BSCSD$ is replaced by

$S \to BC_1$, $C_1 \to SC_2$, $C_2 \to CC_3$, $C_3 \to SD$ which are added to P_2

$V_2 = \{S, A, B, C, D, C_1, C_2, C_3\}$

$\therefore$ G_2 in in CNF and is equivalent to the given grammar

Greibach Normal Form (GNF)

Useful in some proofs and constructions
(e.g. CFG generated by pushdown automata is in GNF)

Definition:

A CFG $G = (V, T, P, S)$ is in GNF if every production is of the form $A \to a\alpha$, where $a \in T$ and $\alpha \in V^*$ (α may be ε) and $S \to \varepsilon$ is in G if $\varepsilon \in L(G)$

When $\varepsilon \in L(G)$, we assume that S doesnot appear on the R.H.S. of any production.

e.g. $S \to aAB \mid \varepsilon$, $A \to bC$, $B \to b$, $C \to c$ is in GNF.

Lemma 1

Let $G = (V, T, P, S)$ be a CFG. Let $A \to \alpha_1 B \alpha_2$ be an A-production in P and $B \to \beta_1 \mid \beta_2 \mid \cdots \mid \beta_r$ be the set of all B-productions.

Let $G_1 = (V, T, P_1, S)$ be obtained from G by deleting the production $A \to \alpha_1 B \alpha_2$ from P_1 and adding the production

$$A \to \alpha_1 \beta_1 \alpha_2 \mid \alpha_1 \beta_2 \alpha_2 \mid \cdots \mid \alpha_1 \beta_r \alpha_2$$

Then,

$$L(G) = L(G_1)$$

Proof:

To prove $L(G) \subseteq L(G_1)$

Let $w \in L(G)$

If we apply $A \rightarrow \alpha_1 B \alpha_2$ in the derivation for $w \in L(G)$, then we have to apply $B \rightarrow \beta_i$ for some i at a later step.

Also note that $A \rightarrow \alpha_1 B \alpha_2$ is the only production in G not in G_1.

The effect of applying $A \rightarrow \alpha_1 B \alpha_2$ and eliminating B in grammar G is the same as applying $A \rightarrow \alpha_1 \beta_i \alpha_2$ for some i in G_1

Hence $w \in L(G_1)$

To prove $L(G_1) \subseteq L(G)$

Instead of applying $A \rightarrow \alpha_1 \beta_i \alpha_2$ in a derivation of G_1 we can apply $A \rightarrow \alpha_1 B \alpha_2$ and $B \rightarrow \beta_i$ in G to get

$$A \overset{*}{\underset{G}{\Longrightarrow}} \alpha_1 \beta_i \alpha_2$$

Thus if $w \in L(G_1)$ and $A \rightarrow \alpha_1 \beta_i \alpha_2$ is used for the derivative of w, then by replacing the single $A \rightarrow \alpha_1 \beta_i \alpha_2$ by two steps $A \rightarrow \alpha_1 B \alpha_2$ and $B \rightarrow \beta_i$ in the derivative of w, we obtain $w \in L(G)$

Hence $L(G_1) \subseteq L(G)$

Lemma 2

Let $G = (V, T, P, S)$ be a CFG. Let the set of A-productions be

$$A \to A\alpha_1 \mid A\alpha_2 \mid \cdots \mid A\alpha_r \mid \beta_1 \mid \beta_2 \mid \cdots \mid \beta_s$$

Let z be a new variable.

Let $G_1 = (V \cup \{z\}, T, P, S)$, where P_1 is defined as follows:

i. The set of A-productions in P_1 are

$$A \to \beta_1 \mid \beta_2 \mid \cdots \mid \beta_s$$

$$A \to \beta_1 z \mid \beta_2 z \mid \cdots \mid \beta_s z$$

ii. The set of z-productions in P_1 are

$$z \to \alpha_1 \mid \alpha_2 \mid \cdots \mid \alpha_r$$

$$z \to \alpha_1 z \mid \alpha_2 z \mid \cdots \mid \alpha_r z$$

iii. The productions for the other variables are as in G.

Then G_1 is a CFG and equivalent to G

i.e. $L(G_1) = L(G)$

Proof:

In a leftmost derivation, a sequence of productions of the form $A \to A\alpha_i$ must event end with a production $A \to \beta_i$

The sequence of replacements in G

$$A \underset{G}{\Rightarrow} A\alpha_{i_1} \underset{G}{\Rightarrow} A\alpha_{i_2}\alpha_{i_1} \underset{G}{\Rightarrow} \cdots \underset{G}{\Rightarrow} A\alpha_{i_p}\alpha_{i_{p-1}} \cdots \alpha_{i_1}$$

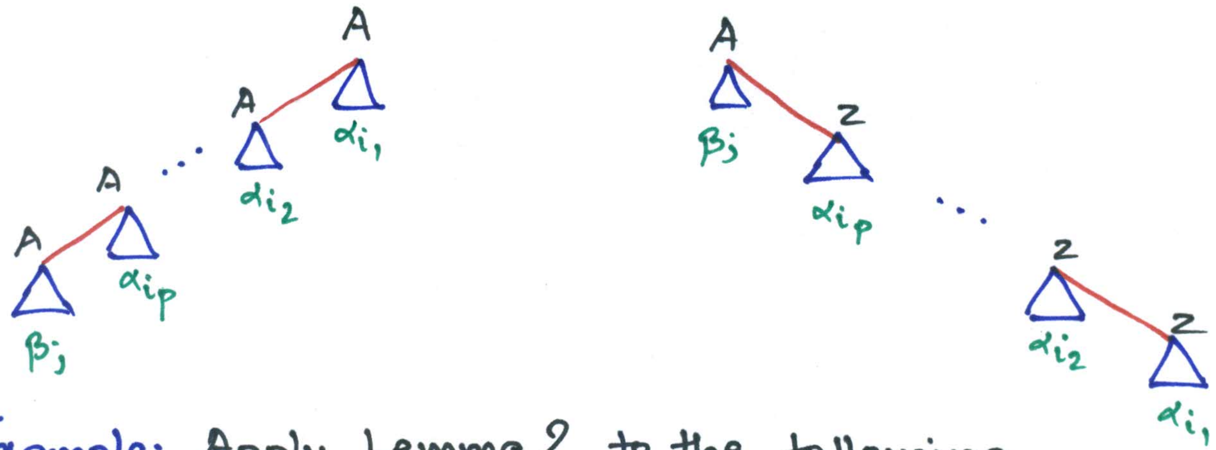
$$\cdots \underset{G}{\Rightarrow} \beta_j \alpha_{i_p} \alpha_{i_{p-1}} \cdots \alpha_{i_1}$$

can be replaced in G_1 by

$$A \underset{G_1}{\Rightarrow} \beta_j Z \underset{G_1}{\Rightarrow} \beta_j \alpha_{i_p} Z \underset{G_1}{\Rightarrow} \beta_j \alpha_{i_p} \alpha_{i_{p-1}} Z$$

$$\cdots \underset{G_1}{\Rightarrow} \beta_j \alpha_{i_p} \alpha_{i_{p-1}} \cdots \alpha_{i_2} Z \underset{G_1}{\Rightarrow} \beta_j \alpha_{i_p} \alpha_{i_{p-1}} \cdots \alpha_{i_2} \alpha_{i_1}$$

The reverse transformation can be also made.
Thus $L(G) = L(G_1)$



Example: Apply Lemma 2 to the following A-productivity in a CFG G.

$$A \to \underbrace{aBD}_{\beta_1} \mid \underbrace{bDB}_{\beta_2} \mid \underbrace{c}_{\beta_3} \qquad A \to A\underbrace{B}_{\alpha_1} \mid A\underbrace{D}_{\alpha_2}$$

Soln:

$$G = (V, T, P, S) \longrightarrow G' = (V', T, P', S), \quad V' = V \cup \{Z\}$$

P':

i. $A \to aBD \mid bDB \mid c$

$A \to aBDZ \mid bDBZ \mid cZ$

ii. $Z \to B \mid D$

$Z \to BZ \mid DZ$

A-productions

$A \to \beta_1 \mid \beta_2 \mid \beta_3$

$A \to \beta_1 Z \mid \beta_2 Z \mid \beta_3 Z$

Z-productions

$Z \to \alpha_1 \mid \alpha_2$

$Z \to \alpha_1 Z \mid \alpha_2 Z$

Theorem (Greibach Normal form or GNF)

Every CFL L without can be generated by a CFG for which each production is of the form $A \to a\alpha$, where A is a terminal and α is a (possibly empty) string of variables.

Proof: (Construction of a CFG in GNF)

Step 1: Eliminate null productions and then construct a grammar G in CNF generating L

We rename the variables as $A_1, A_2, \ldots A_n$ with $S = A_1$

Let $G = (\{A_1, A_2, \ldots A_n\}, T, P, A_1)$

Step 2: (Modify the productions so that if $A_i \to A_j \gamma$ is a production, then $j > i$)

Starting with and proceeding to we do this as follows:

* { Assume productions have been modified so that for $1 \le i \le k$, $A_i \to A_j \gamma$ is a production only if $j > 1$

We now modify the A_k-productions (Apply lemma 1)

- If $A_k \to A_j \gamma$ is a production with $j < k$, then we generate a new set of productions by substituting for A_j the R.H.S. of each A_j-productions according to Lemma 1.

i.e. we remove $A_k \to A_j \gamma$ and add productions $A_k \to \beta\gamma$ for all A_j-productions $A_j \to \beta$

as $j < k$, $A_j \to \beta$ satisfies '*'

Repeating the process $k-1$ times at most, we obtain productions of the form $A_k \to A_l \gamma$, $l \geq k$

Step 3: (Apply lemma 2)

- The production with $l = k$ are then replaced according to Lemma 2, introducing a new variable Z_k

i.e. we remove each $A_k \to A_k \gamma$, and productions

$$Z_k \to \gamma,\ Z_k \to \gamma Z_k$$

and for each $A_k \to \beta$, where β doesnot start with A_k, we add production $A_k \to \beta Z_k$

- By repeating the above process for each original variable, we have only productions of the form:

i. $A_i \to A_j \gamma \quad , \ j > i$

ii. $A_i \to a\gamma \quad , \ a \in T$

iii. $Z_i \to \gamma \quad , \ \gamma \in (V \cup \{Z_1, \dots, Z_{i-1}\})^*$

Any A_n-production is of the form $A_n \to a$ where $a \in T$ as A_n is the highest numbered variable

Step 4: (Modify A_i-productions to the form $A_i \to a\gamma$ for $i = 1, 2, \dots n-1$ using lemma 1.)

- A_n- productions are of the form $A_n \to a\gamma$
- A_{n-1} productions are of the form $A_{n-1} \to a\gamma'$ or $A_{n-1} \to A_n \gamma'$

Eliminate production $A_{n-1} \to A_n \gamma'$ by using lemma 1.

The resulting productions are in the required form.

- Repeat the construction by considering $A_{n-2}, A_{n-3}, \ldots A_1$

Step 5: (Modify z_i-productions using lemma 1)

- z_i- productions are of the form $z_i \rightarrow a\gamma$ or $z_i \rightarrow A_k \gamma$ for some k.

Remark:

It is not necessary to convert all the productions to the form required for CNF in the 1st step.

We may allow productions of the form

- $A \rightarrow a\alpha$ where $a \in T, \alpha \in V^*$
- $A \rightarrow \alpha$ where $\alpha \in V^*$ with $|\alpha| \geq 2$

Remark

A CFG in GNF generates a CFL without ϵ

Example:

Convert the grammar $S \to AB, A \to BS|b, B \to SA|a$ into GNF.

Sol^n:

Step 1: No null productions, CNF

rename variable: $A_1 = S,\ A_2 = A,\ A_3 = B$

productions are: $A_1 \to A_2A_3,\ A_2 \to A_3A_1 | b,\ A_3 \to A_1A_2 | a$

Step 2:

i. A_1 productions $A_1 \to A_2A_3$ is in required form

ii. A_2 productions $A_2 \to A_3A_1 | b$ are in the required form

iii. $A_3 \to a$ is in required form

Apply Lemma 1 to $A_3 \to A_1A_2$:

replace $A_3 \to A_1A_2$ by the following productions

$A_3 \to A_2A_3A_2$ as $A_1 \to A_2A_3$ is an A_1-productions

Apply lemma 1 again to $A_3 \to A_2A_3A_2$

Resulting productions are $A_3 \to A_3A_1A_3A_2 | bA_3A_2$

Thus A_3- productions are

$$A_3 \to a | A_3A_1A_3A_2 | bA_3$$

Step 3: Apply Lemma 2 to $A_3 \to A_3A_1A_3A_2$

introduce a new variable Z_3

the resulting productions are:

$$A_3 \to a | bA_3A_2 | aZ_3 | bA_3A_2Z_3$$

$$Z_3 \to A_1A_3A_2 | A_1A_3A_2Z_3$$

$(A_3 \to \beta_1 | \beta_2 | \beta_1 Z_3 | \beta_2 Z_3,\quad Z_3 \to \alpha_1 | \alpha_1 Z_3)$

Step 4:

i. A_3- productions are

$$\boxed{A_3 \to a \mid bA_3A_2 \mid aZ_3 \mid bA_3A_2Z_3}$$

ii. A_2- productions are

$$A_2 \to A_3A_1 \mid b$$

- We retain $A_2 \to b$
- We replace $A_2 \to A_3A_1$ using lemma 1.

The resulting productions are

$$A_2 \to aA_1 \mid bA_3A_2A_1 \mid aZ_3A_1 \mid bA_3A_2Z_3A_1$$

The set of all modified A_2- productions is

$$\boxed{A_2 \to b \mid aA_1 \mid bA_3A_2A_1 \mid aZ_3A_1 \mid bA_3A_2Z_3A_1}$$

iii. A_1 productions are $A_1 \to A_2A_3$

Apply lemma 1 to $A_1 \to A_2A_3$ to get

$$\boxed{A_1 \to bA_3 \mid aA_1A_3 \mid bA_3A_2A_1A_3 \mid aZ_3A_1A_3 \mid bA_3A_2Z_3A_1A_3}$$

Step 5:

The Z_3- productions are

$$Z_3 \to A_1A_3Z_2 \mid A_1A_3A_2Z_3$$

Apply Lemma 1 to modify Z_3- productions

The resulting Z_3- productions are

$$Z_3 \to bA_3A_3A_2 \mid aA_1A_3A_3A_2 \mid bA_3A_2A_1A_3A_3A_2 \mid aZ_3A_1A_3A_3A_2 \mid bA_3A_2Z_3A_1A_3A_2$$

Example:

Construct a CFG in GNF equivalent to the CFG

$S \to AA \mid a, \quad A \to SS \mid b$

Soln:

Step 1: No null productions, the CFG is in CNF.

Rename the variables: $A_1 = S, \; A_2 = A$

So, the productions are: $A_1 \to A_2 A_2 \mid a, \; A_2 \to A_1 A_1 \mid b$

Step 2:

1. A_1- productions are in the required form. They are $A_1 \to A_2 A_2 \mid a$
2. $A_2 \to b$ is in the required form.

Apply lemma 1 to $A_2 \to A_1 A_1$

As $A_1 \to A_2 A_2 \mid a$ are A_1- productions, we replace $A_2 \to A_1 A_1$ by the following productions:

$$A_2 \to A_2 A_2 A_1 \mid a A_1$$

Thus A_2- productions are $A_2 \to b \mid A_2 A_2 A_1 \mid a A_1$

Step 3:

Apply Lemma 2 to $\underline{A_2 \to A_2 A_2 A_1}$

- Introduce a new variable Z_2
- $A_2 \to A_2 A_2 A_1$ is replaced by $A_2 \to b Z_2 \mid a A_1 Z_2$

$$Z_2 \to A_2 A_1 \mid A_2 A_1 Z_2$$

Step 4:

1. A_2 - productions are $\boxed{A_2 \to b \mid aA_1 \mid bZ_2 \mid aA_1Z_2}$ — ①

A_1 - productions are

$$A_1 \to A_2A_2 \mid a$$

We retain $A_1 \to a$

We replace $A_1 \to A_2A_2$ using Lemma 1 and get the following productions

$$A_1 \to bA_2 \mid aA_1A_2 \mid bZ_2A_2 \mid aA_1Z_2A_2$$

The set of all modified A_1- productions are

$$\boxed{A_1 \to a \mid bA_2 \mid aA_1A_2 \mid bZ_2A_2 \mid aA_1Z_2A_2} \quad \text{— ②}$$

Step 5: Modify Z_2- productions using Lemma 1

Z_2- productions $Z_2 \to A_2A_1 \mid A_2A_1Z_2$ are replaced by the following productions

$$\boxed{\begin{array}{l} Z_2 \to bA_1 \mid aA_1A_1 \mid bZ_2A_1 \mid aA_1Z_2A_1 \\ Z_2 \to bA_1Z_2 \mid aA_1A_1Z_2 \mid bZ_2A_1Z_2 \mid aA_1Z_2A_1Z_2 \end{array}} \quad \text{— ③}$$

Hence the equivalent grammar is

$$G' = (\{A_1, A_2, Z_2\}, \{a, b\}, P_1, A_1)$$

where P_1 consists of A_1- productions, A_2- productions and Z_2- productions given by ①, ②, ③ respectively

Pushdown Automata (PDA)

regular expression $\leftrightarrow$ FA (DFA)

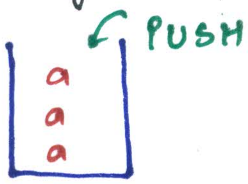
context-free grammar $\leftrightarrow$ PDA (non-deterministic device)

* Deterministic version accepts only a subset of all CFLs

Example:

$L = \{a^n b^n \mid n \geq 1\}$ is not a regular (can be shown using Pumping Lemma), but is a CFL as the CFG

$S \rightarrow aSb \mid ab$ generates L

- A FA cannot accept $L = \{a^n b^n \mid n \geq 1\}$ as it has to remember number of a's in the string and so an infinite number of states
- This difficulty can be avoided by adding an auxilliary memory in the form of a stack
- 

stack S

The a's in the string are added to S (PUSH a)

When b is encountered in the input string an a is removed from S (POP a)

Thus matching of the numbers of a's and of b's is accomplished

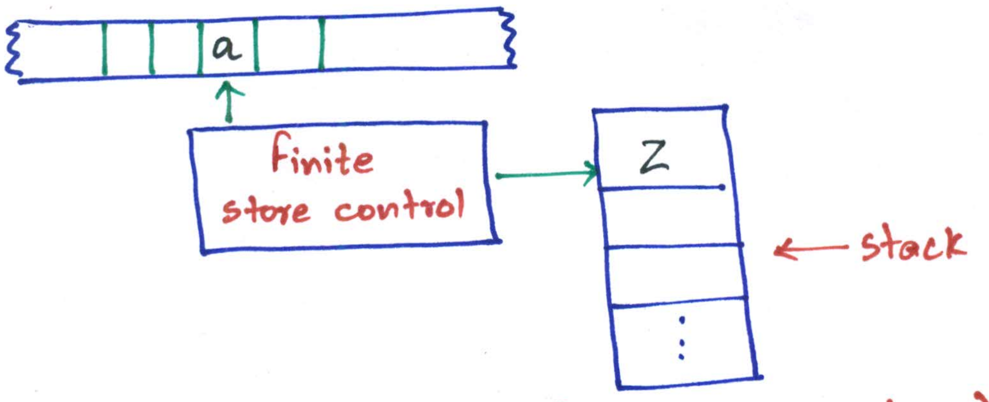
- PDA is essentially a FA with control of both an input tape and a stack.

Definition (PDA)

A pushdown automata M is a system $(Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ where

1. Q is a finite set of states
2. Σ is an alphabet, called the input alphabet
3. Γ is an alphabet, called the stack alphabet
4. q_0 in Q is the initial state
5. z_0 in Γ is a particular stack symbol called the start symbol.
6. $F \subseteq Q$ is the set of final states
7. δ is a mapping from $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma$ to a finite subset of $Q \times \Gamma^*$

PDA



Pushdown store (PDS)

- lower case letters near the front of alphabel
  ↳ input symbol
- lower case letters near the end of alphabet
  ↳ strings of input symbols
- Capital letters → stack symbol
- Greek letters → string of stack symbols

Moves

- $\delta(q, a, z) = \{(p_1, \gamma_1), (p_2, \gamma_2) \ldots (p_m, \gamma_m)\}$

where q and p_i, $1 \le i \le m$ are states $a \in \Sigma$, z is a stack symbol and $\gamma_i \in \Gamma^*$, $1 \le i \le m$

- Interpretation of $\delta(q, a, z)$

  - PDA is state q, with input symbol a and z is the top symbol on the stack.

  - Can for any, enter state p_i,
    replace symbol z by γ_i,
    and advance the input head one symbol

- Convention

  - leftmost symbol of γ_i will be placed highest on the stack and the rightmost symbol lowest on the stack

  i.e. if $\gamma_i = z_1 z_2 \ldots z_k$ then

$$\begin{array}{|c|} z_1 \\ \vdots \\ z_{k-1} \\ z_k \\ \hline \end{array}$$

- Interpretation of $\delta(q, \varepsilon, z) = \{(p_1, \gamma_1) \ldots (p_m, \gamma_m)\}$

  - PDA in state q, independent of the input symbol being scanned and with the top symbol on the stack,

    can enter state p_i and

    replace z by γ_i for any

  - In this case input head is not advanced.

Example:

Let $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ where

$Q: \{q_0, q_1, q_f\}$, $\Sigma = \{a, b\}$, $\Gamma = \{a, z_0\}$

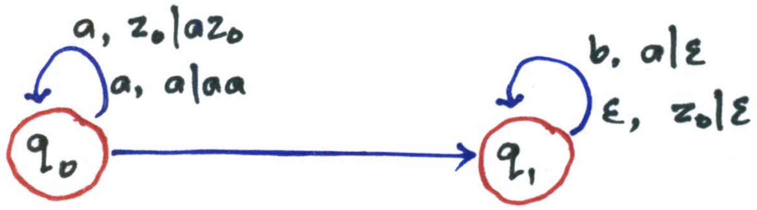
$F = \{q_f\}$

and δ is given by

$$\delta(q_0, a, z_0) = \{(q_0, a, z_0)\}$$
$$\delta(q_1, b, a) = \{(q_1, \varepsilon)\}$$
$$\delta(q_0, a, a) = \{(q_0, aa)\}$$
$$\delta(q_1, \varepsilon, z_0) = \{(q_1, \varepsilon)\}$$
$$\delta(q_0, b, a) = \{(q_1, \varepsilon)\}$$

- PDA cannot take transition when PDS is empty. In this case the PDA halts.

- The behaviour of PDA is non-deterministic as the transition given by any element of $\delta(q, a, z)$



Instantaneous Description (ID)

Let $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a PDA

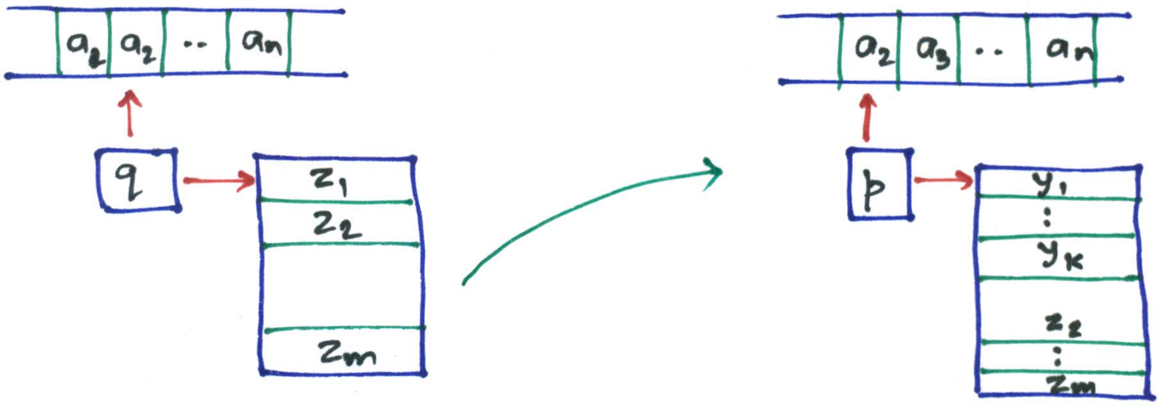
An instantaneous description (ID) is a tuple (q, w, γ) where q is a state, w is a string of input symbols and γ is a string of stack symbols.

Move relation $\vdash_M$

$(q, aw, Z\alpha) \vdash_M (p, w\beta\alpha)$ if $\delta(q, a, z)$ contains (p, β), with $p \in Q$, $\beta \in \Gamma^*$

(a may be ε or an input symbol)

- if $w = a_2 a_3 \cdots a_n$, $\alpha = z_2 \cdots z_m$, $\beta = y_1 y_2 \cdots y_k$



An illustration of the move relation

Relation $\vdash^*_M$ (Reflexive and transitive closure of $\vdash_M$)

- $I \vdash^*_M I$ for each ID I
- $I \vdash^*_M J$ and $J \vdash^*_M K$ imply $I \vdash^*_M K$

Note: $I \vdash^i K$ if ID I can become K after exactly i steps

Note: $\vdash_M$, $\vdash^*_M$, $\vdash^i_M$ subscript M is dropped if PDA M is understood

Accepted Languages

- For a PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ we define $L(M)$ the language accepted by the final state as

$$\{ w \mid (q_0, w, Z_0) \vdash^* (p, \varepsilon, \gamma) \text{ for some } p \text{ in } F \text{ and } \gamma \text{ in } \Gamma^* \}$$

- We define $N(M)$, the language accepted by empty stack (null stack) to be

$$\{ w \mid (q_0, w, Z_0) \vdash^* (p, \varepsilon, \varepsilon) \text{ for some } p \text{ in } Q \}$$

- When acceptance is by null stack, the set of final states is irrelevant, and in this case, we usually let the set of final states be the empty set, i.e. $F = \phi$

- $(q_1, x, \alpha) \vdash^* (q_2, \varepsilon, \beta)$ iff for every $y \in \Sigma^*$

$$(q_1, xy, \alpha) \vdash^* (q_2, y, \beta)$$

- If $(q, x, \alpha) \vdash^* (q', \varepsilon, \gamma)$ then for every $\beta \in \Gamma^*$

$$(q, x, \alpha\beta) \vdash^* (q', \varepsilon, \gamma\beta)$$

- Converse is not true

Example:

The following PDA accepts $\{ww^R \mid w \text{ in } (0+1)^*\}$

$$M = (\{q_1, q_2\}, \{0,1\}, \{R, B, G\}, \delta, q_1, R, \phi)$$

R_1: $\delta(q_1, 0, R) = \{(q_1, BR)\}$

R_2: $\delta(q_1, 1, R) = \{(q_1, GR)\}$

R_3: $\delta(q_1, 0, B) = \{(q_1, BB), (q_2, \varepsilon)\}$

R_4: $\delta(q_1, 0, G) = \{(q_1, BG)\}$

R_5: $\delta(q_1, 1, B) = \{(q_1, GB)\}$

R_6: $\delta(q_1, 1, G) = \{(q_1, G, G), (q_2, \varepsilon)\}$

R_7: $\delta(q_2, 0, B) = \{(q_2, \varepsilon)\}$

R_8: $\delta(q_2, 1, G) = \{(q_2, \varepsilon)\}$

R_9: $\delta(q_1, \varepsilon, R) = \{(q_2, \varepsilon)\}$

R_{10}: $\delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\}$

- $R_1 - R_6$ allow to store the input on the stack

$$\begin{tikzcd}
\text{initial} \arrow[d] & & \\
(q_1, 001100, R) \arrow[d] \arrow[r] & (q_2, 001100, \varepsilon)\ \text{halt} & \\
(q_1, 01100, BR) \arrow[d] \arrow[r] & (q_2, 1100, R) \arrow[r] & (q_2, 1100, \varepsilon)\ \text{halt} \\
(q_1, 1100, BBR) \arrow[d] & & \\
(q_1, 100, GBBR) \arrow[d] \arrow[r] & (q_2, 00, BBR) \arrow[r] & (q_2, 0, BR) \arrow[d] \\
(q_1, 00, GGBBR) \arrow[d] & & (q_2, \varepsilon, R) \arrow[d] \\
(q_1, 0, BGGBBR) \arrow[d] \arrow[r] & (q_2, \varepsilon, GGBBR) & (q_2, \varepsilon, \varepsilon) \\
(q_1, \varepsilon, BBGGBBR) & \text{halt} & \text{accept}
\end{tikzcd}$$

Example:

Construct a PDA M accepting the set of all strings over $\{a, b\}$ with equal number of a's and b's

Sol^{n}:

Let $M = (\{q\}, \{a, b\}, \{z_0, a, b\}, \delta, q, z_0, \phi)$

where δ is defined by the following rules

$$\delta(q, a, z_0) = \{(q, az_0)\}$$
$$\delta(q, b, z_0) = \{(q, bz_0)\}$$

$$\delta(q, a, a) = \{(q, aa)\}$$
$$\delta(q, b, b) = \{(q, bb)\}$$

new incoming symbol

stack top symbol

$$\delta(q, a, b) = \{(q, \varepsilon)\}$$ (b is erased in PDS)

$$\delta(q, b, a) = \{(q, \varepsilon)\}$$ (a is erased in PDS)

$$\delta(q, \varepsilon, z_0) = \{(q, \varepsilon)\}$$

$$(q, w, z_0) \overset{*}{\vdash} (q, \varepsilon, z_0) \vdash (q, \varepsilon, \varepsilon)$$

if w has equal number of a's and b's

Hence $w \in N(M)$

Example 1:

$$M = (\{q_0\}, \{a, b\}, \{z_0\}, \delta, q_0, z_0, \phi)$$

where $\delta(q_0, a, z_0) = \{(q_0, \varepsilon)\}$, $\delta(q_0, b, z_0) = \{(q_0, z_0 z_0)\}$

$$(q_0, aab, z_0 z_0 z_0 z_0) \vdash (q_0, ab, z_0 z_0 z_0)$$
$$\vdash (q_0, b, z_0 z_0)$$
$$\vdash (q_0, \varepsilon, z_0 z_0 z_0)$$

But

$$(q_0, aab, z_0) \vdash (q_0, ab, \varepsilon)$$

Thus, $(q_0, aab, z_0 z_0 z_0 z_0) \vdash^{*} (q_0, \varepsilon, z_0 z_0 z_0)$

but $(q_0, aab, z_0 z_0 z_0) \not\vdash^{*} (q_0, \varepsilon, z_0 z_0)$

$$\vdash (q_0, \underset{\neq \varepsilon}{ab}, z_0 z_0)$$

Example 2:

$$M = (\{q_0, q_1, q_f\}, \{a, b, c\}, \{a, b, z_0\}, \delta, q_0, z_0, \{q_f\})$$

is a PDA, where δ is defined as:

$\delta(q_0, a, z_0) = \{(q_0, a z_0)\}$, $\delta(q_0, b, z_0) = \{(q_0, b z_0)\}$

$\delta(q_0, a, a) = \{(q_0, aa)\}$ $\delta(q_0, b, a) = \{(q_0, ba)\}$

$\delta(q_0, a, b) = \{(q_0, ab)\}$ $\delta(q_0, b, b) = \{(q_0, bb)\}$

$\delta(q_0, c, a) = \{(q_1, a)\}$ $\delta(q_0, c, b) = \{(q_1, b)\}$

$\delta(q_0, c, z_0) = \{(q_1, z_0)\}$ $\delta(q_1, a, a) = \delta(q_1, b, b) = \{(q_1, \varepsilon)\}$

$\delta(q_1, \varepsilon, z_0) = \{(q_f, z_0)\}$

M accepts $L = \{ w c w^R \mid w \in \{a, b\}^* \}$ by final state

i.e. $L(M) = L$

- $(q_0, bacab, z_0) \vdash (q_0, acab, bz_0)$

$$\vdash (q_0, cab, abz_0)$$
$$\vdash (q_1, ab, abz_0)$$
$$\vdash (q_1, b, bz_0)$$
$$\vdash (q_1, \varepsilon, z_0)$$
$$\vdash (q_f, \varepsilon, z_0)$$

i.e. $(q_0, bacab, z_0) \overset{*}{\vdash} (q_f, \varepsilon, z_0)$

Proceeding in a similar way; we can show that

$$(q_0, wcw^R, z_0) \overset{*}{\vdash} (q_f, \varepsilon, z_0) \ \forall\ w \in \{a,b\}^*$$

i.e. $L = \{wcw^R \mid w \in \{a,b\}^*\} \subseteq L(M)$

Reverse inclusion is also true.

- $(q_0, abcbb, z_0) \vdash (q_0, bcbb, az_0)$

$$\vdash (q_0, cbb, baz_0)$$
$$\vdash (q_1, bb, baz_0)$$
$$\vdash (q_1, b, az_0)$$

- PDA halts when it is in ID (q_1, b, az_0) as

$$\delta(q_1, b, a) = \phi$$

- As $\delta(q_0, c, z_0) = \phi$, the PDA cannot make any transition if it starts with an ID of the form (q_0, cw, z_0)