Pushdown Automata (PDA)

regular expression $\longleftrightarrow$ FA (DFA)

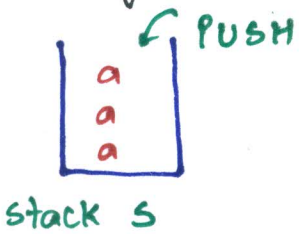
context-free grammar $\longleftrightarrow$ PDA (non-deterministic device)

* 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
- 

  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

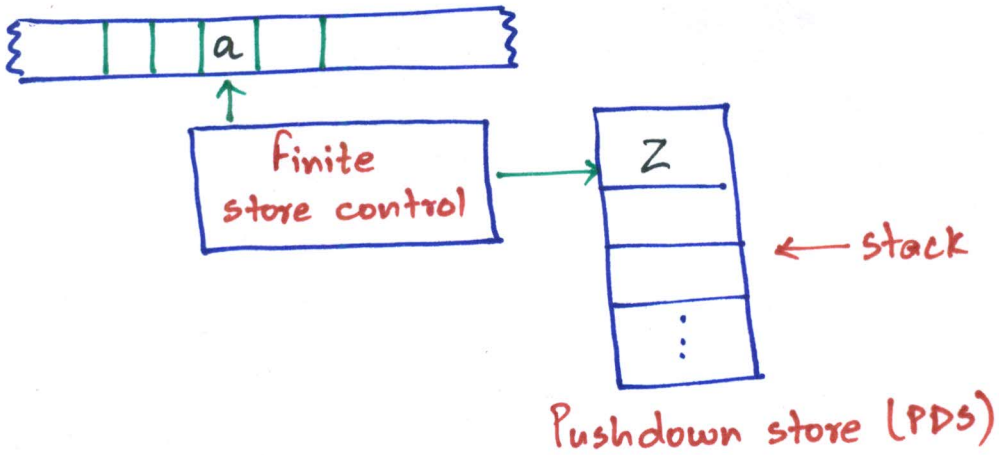
- PDA is essentially a FA with control of both an input tape and a stack.

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



- lower case letters near the front of alphabet
  ↳ 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 $a \in Z$, z is a stack symbol and $\gamma_i \in M^*$, $1 \le i \le m$

- 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

$$\begin{bmatrix} z_1 \\ \vdots \\ z_{k-1} \\ z_k \end{bmatrix}$$

- 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 <u>input head is not advanced.</u>

Example:

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

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

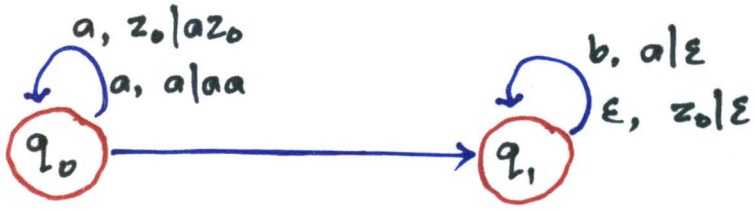
$F = \{q_f\}$

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.

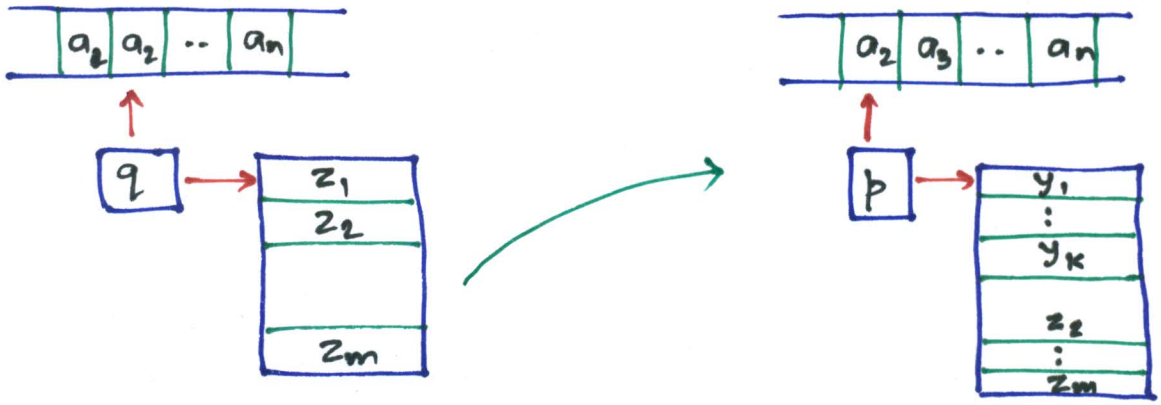
Move relation $\vdash_M$

$(q, aw, Z\alpha) \vdash_M (p, w\beta\alpha)$ if $\delta(q, a, Z)$ contains (p, β)

(a may be ε or an input symbol)

$p \in Q$, $\beta \in \Gamma^*$

- 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: $\vdash_M$, $\vdash_M^*$, $\vdash_M^i$ 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

initial

$(q_1, 001100, R)$ → $(q_2, 001100, \varepsilon)$ halt

$(q_1, 01100, BR)$ → $(q_2, 1100, R)$ → $(q_2, 1100, \varepsilon)$ halt

$(q_1, 1100, BBR)$

$(q_1, 100, GBBR)$ → $(q_2, 00, BBR)$ → $(q_2, 0, BR)$ → (q_2, ε, R) → $(q_2, \varepsilon, \varepsilon)$ accept

$(q_1, 00, GGBBR)$

$(q_1, 0, BGGBBR)$ → $(q_2, \varepsilon, GGBBR)$ halt

$(q_1, \varepsilon, BBGGBBR)$

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

where δ is defined by the following rules

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

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

new incoming symbol

stack top symbol

$$\begin{cases} \delta(q, a, b) = \{(q, \varepsilon)\} & \text{(b is erased in PDS)} \\ \delta(q, b, a) = \{(q, \varepsilon)\} & \text{(a is erased in PDS)} \\ \delta(q, \varepsilon, z_0) = \{(q, \varepsilon)\} \end{cases}$$

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

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

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

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

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:

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

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

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

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

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

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

M accepts $L = \{wcw^R \mid w \in \{a,b\}^*\}$ 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. $(q_0, bacab, z_0) \vdash^* (q_f, \varepsilon, z_0)$

Proceeding in a similar way; we can show that

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

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)

WEEK 11 : Lecture Notes

Deterministic PDA:

A PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ is deterministic if

i) $\delta(q, a, z)$ is either empty or a singleton set for each $q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$ and $z \in \Gamma$

ii) $\delta(q, \varepsilon, z) \neq 0$ implies $\delta(q, a, z) = \phi \ \forall \ a \in \Sigma$

- DFA, NFA are equivalent with respect to the languages accepted
- Same is not true for PDA
- ww^R is accepted by non-deterministic PDA, but is not accepted by any deterministic PDA

Example

Consider the PDA

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

where δ is defined by:

$\delta(q_0, a, z_0) = \{(q_0, az_0)\}$, $\delta(q_0, b, z_0) = \{(q_0, bz_0)\}$

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

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

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

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

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

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

Then,

$$N(M) = \{wcw^R \mid w \in \{a, b\}^*\}$$

Pushdown Automata and Context-free languages

- Class of languages accepted by PDA's are precisely the class of CFLs.

→ language accepted by PDAs final state are exactly the languages accepted by PDA's by empty stack, i.e. $N(M_1) = L$ iff $L(M_2) = L$, where M_1, M_2 are accepted

→ languages accepted by empty state are exactly the CFLs i.e. $N(M) = L$ iff L is a CFL

Equivalence of acceptance by final state and empty state

Theorem:

If L is $L(M_2)$ for some PDA M_2 then L is $N(M_1)$ for some PDA M_1

Proof:

Let $M_2 = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ be a PDA such that $L = L(M_2)$

Let $M_1 = (Q \cup \{q_e, q_0'\}, \Sigma, \Gamma \cup \{X_0\}, \delta', q_0,$

where δ' is defined as follows:

R_1: $\delta'(q_0', \varepsilon, X_0) = \{(q_0, z_0 X_0)\}$

R_2: $\delta'(q, a, z) = \delta(q, a, z)$ $\forall (q, a, z) \in Q \times \{\Sigma \cup \{\varepsilon\}\} \times \Gamma$

R_3: $\delta'(q, \varepsilon, z)$ contains (q_e, ε) $\forall q \in F$ and $z \in \Gamma \cup \{X_0\}$

R_4: $\delta'(q_e, \varepsilon, z)$ contains (q_e, ε) $\forall z \in \Gamma \cup \{X_0\}$

- R_1 causes M_1 to enter the initial ID of M_2, except that M_1 will have its own bottom of the stack marker which is below the symbol of M_2's stack.

- R_2 allow M_1 to simulate M_2

- Should M_2 ever enter a final state R_3 and R_4 allow M_1 the choice of entering state q_e and erasing its stack

Note: M_2 may possibly erase its entire stack for some input x not in $L(M_2)$ (i.e. without entering to a final state of F)

This is why M_1 has its own special bottom-of-stack marker X_0

Otherwise, M_1 in simulating M_2 would also erase its entire stack, thereby accepting x when it should not.

To prove $L(M_2) \subseteq N(M_1)$

Let $x \in L(M_2)$

Then $(q_0, x, z_0) \overset{*}{\underset{M_2}{\vdash}} (q, \varepsilon, \gamma)$ for some $q \in F$

Now consider M_1 with input x

By R_1

$$(q_0', x, X_0) \underset{M_1}{\vdash} (q_0, x, z_0 X_0)$$

By R_2, every move of M_2 is a legal move of M_1, thus

$$(q_0, x, z_0) \overset{*}{\underset{M_1}{\vdash}} (q, \varepsilon, \gamma)$$

and so

$$(q_0', x, X_0) \underset{M_1}{\vdash} (q_0, x, z_0 X_0)$$

$$\underset{M_1}{\overset{*}{\vdash}} (q, \varepsilon, \gamma X_0)$$

$$\underset{M_1}{\overset{*}{\vdash}} (q_e, \varepsilon, \varepsilon)$$

By R_3 and R_4

and M_1 accepts x by empty stack.

Thus $x \in N(M_1)$

To prove $N(M_1) \subseteq L(M_2)$

If M_1 accepts x by empty stack, it is easy to show that the sequence of moves must be

i. one move by R_1

ii. then a sequence of moves by R_2 in which M_1 simulates acceptance of x by M_2 followed by the erasing of M_1's stack using R_3 and R_4

Thus $x \in L(M_2)$

Theorem

If L is $N(M_1)$ for some PDA M_1 then L is $L(M_2)$ for some PDA M_2

Proof:

Let $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, \phi)$ be a PDA such that

$$L = N(M_1)$$

Let $M_2 = (Q \cup \{q_0', q_f\}, \Sigma, \Gamma \cup \{X_0\}, \delta', q_0', X_0, \{q_f\})$

where δ' is defined as:

R_1: $\delta'(q_0', \varepsilon, X_0) = \{(q_0, z_0 X_0)\}$

R_2: $\delta'(q, a, z) = \delta(q, a, z)$ $\forall$ $q \in Q, a \in \Sigma \cup \{\varepsilon\}, z \in \Gamma$

R_3: $\delta'(q, \varepsilon, X_0)$ contains (q_f, ε) $\forall$ $q \in Q$

- R_1 causes M_2 to enter the initial ID of M_1, except that M_2 will have its own bottom-of-stack marker X_0
- R_2 allows M_2 to simulate M_1

  Should M_1 ever erase its entire stack, then M_2, when simulating M_1 will erase it's entire stack except the symbol X_0 at the bottom.

- R_3 causes M_2, when X_0 appears, to enter a final state thereby accepting the input x

The proof that $L(M_2) = N(M_1)$ is similar to the previous proof.

Example: Consider the PDA M_1 given by

$$M_1 = (\{q_0, q_1\}, \{a, b\}, \{a, z_0\}, \delta, q_0, z_0, \phi)$$

where δ is given by

R_1: $\delta(q_0, a, z_0) = \{(q_0, az_0)\}$

R_2: $\delta(q_0, a, a) = \{(q_0, aa)\}$

R_3: $\delta(q_0, b, a) = \{(q_1, \varepsilon)\}$

R_4: $\delta(q_1, b, a) = \{(q_1, \varepsilon)\}$

R_5: $\delta(q_1, \varepsilon, z_0) = \{(q_1, \varepsilon)\}$

Determine $N(M_1)$. Also construct a PDA M_2 such that $L(M_2) = N(M_1)$

Soln:

- R_1 stores a in PDS
  R_2 repeatedly stores a in PDS (thus string a^n in PDS)
- R_3 is used to erase a from PDS when b is encountered for the first time and PDA transits to state q
- R_4 repeatedly erases a from PDS when b is encountered and PDA remain in the same state q
- R_5 empties PDS if z_0 remains in stack after processing the entire input.

Thus, $(q_0, a^n b^n, z_0) \overset{*}{\vdash} (q_0, b^n, a^n z_0)$

$\vdash (q_1, b^{n-1}, a^{n-1} z_0)$ by R_1 and R_2

$\overset{*}{\vdash} (q_1, \varepsilon, z_0)$ by R_3 and R_4

$\vdash (q_1, \varepsilon\, \varepsilon)$ by R_5

Thus $a^n b^n \in N(M)$

i.e. $\{a^n b^n \mid n \geq 1\} \subseteq N(M)$

To prove $N(M) \subseteq \{a^n b^n \mid n \geq 1\}$

- Let $w \in N(M)$

  $\Rightarrow$ Then $(q_0, w, z_0) \overset{*}{\underset{M}{\vdash}} (q_1, \varepsilon, \varepsilon)$

  (Note: PDS can be emptied only when M is in state q_1)

- Also w must start with a, otherwise we cannot make any move.
- We store a in PDS if current input symbol is a and the topmost symbol of PDS is either a or z_0
- PDA erases a in PDS if current input symbol is b
- PDA enters ID $(q_1, \varepsilon, \varepsilon)$ only by R_5
- PDA can reach ID (q_1, ε, z_0) only by erasing a's in PDS, which is possible.
- When # of b's = # of a's and so $w = a^n b^n$

Thus $N(M) = \{a^n b^n \mid n \geq 1\}$

Construction of M_2 such that $L(M_2) = N(M_1)$

$$M_1 = (\{q_0, q_1\}, \{a, b\}, \{a, z_0\}, \delta, q_0, z_0, \phi)$$

$\downarrow$ construct

$$M_2 = (\{q_0, q_1, q_0', q_f\}, \{ab\}, \{a, z_0, x_0\}, \delta', q_0', x_0, \{q_f\})$$

where δ' is defined by

added rule $\rightarrow$ $\delta'(q_0', \varepsilon, x_0) = \{(q_0, z_0 x_0)\}$

follow given rules
$$\begin{cases} \delta'(q_0, a, z_0) = \delta(q_0, a, z_0) = \{(q_0, a z_0)\} \\ \delta'(q_0, a, a) = \{(q_0, aa)\} \\ \delta'(q_0, b, a) = \{(q_1, \varepsilon)\} \\ \delta'(q_1, b, a) = \{(q_1, \varepsilon)\} \\ \delta'(q_1, \varepsilon, z_0) = \{(q_1, \varepsilon)\} \end{cases}$$

added rules
$$\begin{cases} \delta'(q_0, \varepsilon, x_0) = \{(q_f, \varepsilon)\} \\ \delta'(q_1, \varepsilon, x_0) = \{(q_f, \varepsilon)\} \end{cases}$$

Thus

$$L(M_2) = N(M_1)$$

Equivalence of PDA's and CFL's

Theorem:

If L is a CFL, then there exists a PDA M such that $L = N(M)$

Proof:

Assume that $\varepsilon \notin L(G)$

(The construction can be modified if $\varepsilon \in L(G)$)

Let $G = (V, T, P, S)$ be a CFG in GNF generation L

Let $M = (\{q\}, T, V, \delta, q, S, \phi)$

where $\delta(q, a, A)$ contains (q, γ) whenever $A \to a\gamma$ is in P.

Claim:

$S \overset{*}{\Rightarrow} x\alpha$ by a leftmost derivation iff

$$(q, x, S) \overset{*}{\underset{M}{\vdash}} (q, \varepsilon, \alpha)$$

where $x \in T^*$, $\alpha \in V^*$

Example:

Construct a PDA M equivalent to the following CFG: $S \to 0BB$, $B \to 0S \mid 1S \mid 0$

Test whether 010^4 is in $N(M)$.

Sol^n:

$M = (\{q\}, \{0,1\}, \{S, B\}, \delta, q, S, \phi)$ where δ is defined by

$\delta(q, 0, S) = \{(q, BB)\}$, $\quad \delta(q, 1, B) = \{(q, B)\}$

$\delta(q, 0, B) = \{(q, S), (q, \varepsilon)\}$

So, $(q, 010^4, S) \vdash (q, 10^4, BB)$

$\vdash (q, 0^4, SB)$

$\vdash (q, 0^3, BBB)$

$\vdash (q, 0^2, BB)$

$\vdash (q, 0, B)$

$\vdash (q, \varepsilon, \varepsilon)$

accept

Thus, $010^4 \in N(M)$

Claim:

$S \overset{*}{\Rightarrow} x\alpha$, $x \in T^*$, $\alpha \in V^*$ ——— (1)

iff $(q, x, S) \overset{*}{\underset{M}{\vdash}} (q, \varepsilon, \alpha)$

Proof:

<u>if part</u>

Let $(q, x, S) \overset{i}{\vdash} (q, \varepsilon, \alpha)$ and show by induction on i that $S \overset{*}{\Rightarrow} x\alpha$

<u>Base: $i=0$:</u> $(q, x, S) \overset{0}{\vdash} (q, \varepsilon, \alpha)$

i.e. $x = \varepsilon$, $\alpha = S$ and

$S \overset{*}{\Rightarrow} S \; (= x\alpha)$ holds

<u>Induction step</u>

Let $i \geq 0$ and $x = ya$, $y \in T^*$, $a \in T$

Consider next to last step:

$$\underbrace{(q, ya, S) \overset{i-1}{\vdash} (q, a, \beta)}_{(q, y, S) \overset{i-1}{\vdash} (q, \varepsilon, \beta)} \vdash (q, \varepsilon, \alpha), \quad \beta \in V^*$$

By induction hypothesis

$$S \overset{*}{\Rightarrow} y\beta, \quad y \in T^*, \ \beta \in V^*$$

Also, the last move

$$(q, a, \beta) \vdash (q, \varepsilon, \alpha)$$

implies

$$\beta = A\gamma \text{ for some } A \in V$$

$$\left(\begin{array}{l} (q, a, A\gamma) \vdash (q, \varepsilon, \eta\gamma) \text{ if } \delta(q, a, A) \text{ contains } (q, \eta) \\ \qquad = (q, \varepsilon, \alpha) \text{ i.e. if } A \to a\eta \text{ is in } P \\ \quad \text{implies } \alpha = \eta\gamma \end{array} \right)$$

Thus, $S \overset{*}{\Rightarrow} y\beta \Rightarrow ya\eta\gamma = ya\alpha = x\alpha$

and we conclude the 'if' part of (1)

<u>only if part:</u>

Let $S \overset{i}{\Rightarrow} x\alpha$, $x \in T^*$, $\alpha \in V^*$ by a leftmost derivation.

We show by induction on i that

$$(q, x, S) \overset{*}{\vdash} (q, \varepsilon, \alpha)$$

<u>Base:</u> $i = 0$

$x = \varepsilon$, $\alpha = S$ as $S \overset{0}{\Rightarrow} x\alpha$

$$\therefore (q, x, S) = (q, \varepsilon, S) \overset{0}{\vdash} (q, \varepsilon, \alpha)$$

<u>Inductive step:</u>

Let $i \geq 0$ and suppose $S \overset{i-1}{\Rightarrow} yA\gamma \Rightarrow ya\eta\gamma$

where $x = ya$, $\alpha = \eta\gamma$ and $A \to a\eta$ is in P as G is in GNF

By induction hypothesis

$$(q, y, s) \overset{*}{\underset{M}{\vdash}} (q, \varepsilon, A\gamma)$$

and thus $(q, ya, s) \overset{*}{\underset{M}{\vdash}} (q, a, A\gamma)$

Since $A \to a\eta$ is in P, it follows that $\delta(q, a, A)$ contains (q, η).

Thus $(q, x, s) \overset{*}{\underset{M}{\vdash}} (q, a, A\gamma)$

$$\underset{M}{\vdash} (q, \varepsilon, \eta\gamma) = (q, \varepsilon, \alpha)$$

and the 'only if' part of ① follows.

Alternative construction

If G is not in GNF, set PDA M to be

$$M = (\{q\}, T, V \cup T, \delta, q, s, \phi)$$

where δ is defined as follows:

1. $\delta(q, \varepsilon, A) = \{(q, \alpha) \mid A \to \alpha \text{ is in } P\}$
2. $\delta(q, a, a) = \{(q, \varepsilon)\}$ for every $a \in T$

Example: Convert the CFG G given below to a PDA.

$E \to I \mid E+E \mid E*E \mid (E)$

$I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

Soln:

$$M = (\{q\}, \underbrace{\{a, b, 0, 1, +, *, (,)\}}_{= T}, \{I, E\} \cup T, \delta, q, s, \phi)$$

where δ is defined as

$\delta(q, \varepsilon, E) = \{(q, I), (q, E+E), (q, E*E), (q, (E))\}$

$\delta(q, \varepsilon, I) = \{(q, a), (q, b), (q, Ia), (q, Ib), (q, I0), (q, I1)\}$

$\delta(q, c, c) = \{(q, \varepsilon)\} \quad \forall\, c \in T$

Example:

G is a CFG with productions

$$S \to 0BB, \quad B \to 0S \mid 1S \mid 0$$

Convert it to PDA M such that

$$N(M) = L(G)$$

Sol$^{\underline{n}}$:

$$M = (\{q\}, \underbrace{\{0,1\}}_{T}, \underbrace{\{S, B, 0, 1\}}_{V \cup T}, \delta, q, S, \phi)$$

where δ is defined as

$$\delta(q, \varepsilon, S) = \{(q, 0BB)\}$$

$$\delta(q, \varepsilon, B) = \{(q, 0S), (q, 1S), (q, 0)\}$$

$$\delta(q, 0, 0) = \{(q, \varepsilon)\}$$

$$\delta(q, 1, 1) = \{(q, \varepsilon)\}$$

Check $010^4 \in N(M)$ or not.

$$\begin{aligned}
(q, 010^4, S) &\vdash (q, 010^4, 0BB) \\
&\vdash (q, 10^4, BB) \\
&\vdash (q, 10^4, 1SB) \\
&\vdash (q, 0^4, SB) \\
&\vdash (q, 0^4, 0BBB) \\
&\vdash (q, 0^3, BBB) \\
&\vdash (q, 0^3, 0BB) \\
&\vdash (q, 0^2, BB) \\
&\vdash (q, 0^2, 0B) \\
&\vdash (q, 0, B) \\
&\vdash (q, 0, 0) \\
&\vdash (q, \varepsilon, \varepsilon) \quad \text{accept}
\end{aligned}$$

Theorem

The following three statements are equivalent

1. L is a CFL
2. $L = N(M_1)$ for some PDA M_1
3. $L = L(M_2)$ for some PDA M_2

(Proofs skipped, only constructions are given)

- $(3) \Rightarrow (2)$

$$M_2 = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$$

such that $L = L(M_2)$

$\downarrow$ construct M_1 s.t. $L = N(M_1)$

$$M_1 = (Q \cup \{q_e, q_0'\}, \Sigma, \Gamma \cup \{X_0\}, \delta', q_0', X_0, \phi)$$

where δ' is defined as follows

R_1: $\delta'(q_0', \varepsilon, X_0) = \{(q_0, Z_0 X_0)\}$

$\delta'(q, a, Z) = \delta(q, a, Z) \ \forall q \in Q, a \in \Sigma \cup \{\varepsilon\}, Z \in \Gamma$

$\delta'(q, \varepsilon, Z)$ contains $(q_e, \varepsilon) \ \forall q \in F, Z \in \Gamma \cup \{X_0\}$

$\delta'(q_e, \varepsilon, Z)$ contains $(q_e, \varepsilon) \ \forall Z \in \Gamma \cup \{X_0\}$

- (2) $\Rightarrow$ (3)

$M_1 = (Q, \Sigma, \Gamma, \delta, q_0, z_0, \phi)$ such that $L = N(M_1)$

$\downarrow$ construct M_2 s.t. $L = L(M_2)$

$M_2 = (Q \cup \{q_0', q_f\}, \Sigma, \Gamma \cup \{X_0\}, \delta', q_0', X_0, \{q_f\})$

where δ' is defined as follows:

R_1: $\delta'(q_0', \varepsilon, X_0) = \{(q_0, z_0 X_0)\}$

R_2: $\delta'(q, a, z) = \delta(q, a, z)$ $\forall\, q \in Q, a \in \Sigma \cup \{\varepsilon\}$ and $z \in \Gamma$

R_3: $\delta'(q, \varepsilon, X_0)$ contains (q_f, ε) $\forall\, q \in Q$

- (1) $\Rightarrow$ (2)

$G = (V, T, P, S)$ such that $L = L(G)$

$\downarrow$ construct M_1 s.t. $L = N(M_1)$

$M_1 = (\{q\}, T, V, \delta, q, S, \phi)$

where $\delta(q, a, A)$ contains (q, r)

whenever $A \to ar$ is in P.

Alternative construction of M_1 from G

$G = (V, T, P, S)$ not in GNF with $L = L(G)$

$\downarrow$ construct M_1 s.t. $L = N(M_1)$

$M_1 = (\{q\}, T, V \cup T, \delta, q, S, \phi)$

where δ is defined as

(i) $\delta(q, \varepsilon, A) = \{(q, \alpha) \mid A \to \alpha \text{ is in } P\}$

(ii) $\delta(q, a, a) = \{(q, \varepsilon)\}$ for every $a \in T$

- (2) $\Rightarrow$ (1)

$M_1 = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, \phi)$ such that $L = N(M_1)$

$\downarrow$ construct $G = (V, T, P, S)$ s.t. $L = L(G)$

$G = (V, T, P, S)$, $T = \Sigma$, $V = S \cup \{[q, z, p] \mid q, p \in Q, z \in \Gamma\}$

and P is set of following productions

i. $S \to [q_0, z_0, p] \ \forall \ p \in Q$

ii. $(q_1, \varepsilon) \in \delta(q, a, z)$ induces production

$$[q, z, q_1] \to a$$

iii. $(q_1, z_1 z_2 \dots z_m) \in \delta(q, a, z)$ yields productions

$$[q, z, q_{m+1}] \to a\,[q_1, z_1, q_2]\,[q_2, z_2, q_3] \dots$$
$$\dots [q_m, z_m, q_{m+1}]$$

for each $q_1, q_2 \dots q_{m+1} \in Q$

$z_1, z_2, \dots z_m \in \Gamma$

and $a \in \Sigma \cup \{\varepsilon\}$

Theorem

If L is $N(M)$ for some PDA M, then L is a CFL.

Proof:

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

(construction of G)

We define $G = (V, \Sigma, P, S)$ where

$$V = \{S\} \cup \{[q, z, p] \mid q, p \in Q,\ z \in \Gamma\}$$

P is the set of productions:

1. $S \to [q_0, z_0, q]$ for each $q \in Q$

2a. $[q, z, q_{m+1}] \to a[q_1, z_1, q_2][q_2, z_2, q_3] \ldots [q_m, z_m, q_{m+1}]$

for each $q, q_1, q_2, \ldots q_{m+1} \in Q$

each $a \in \Sigma \cup \{\varepsilon\}$ and

$z_1, z_2, \ldots z_m \in \Gamma$ such that

$(q_1, z_1 z_2 \ldots z_m) \in \delta(q, a, z)$

2b. if $m = 0$, then the production is

$[q, z, q_1] \to a$ induced by $(q_1, \varepsilon) \in \delta(q, a, z)$

corresponding to move erasing a pushdown symbol

$\therefore \quad L(G) = N(M)$

Example: Construct a CFG G which accepts $N(M)$ where

$$M = (\{q_0, q_1\}, \{a, b\}, \{z_0, z\}, \delta, q_0, z_0, \phi)$$

and δ is given by

$$\delta(q_0, b, z_0) = \{(q_0, \cancel{z}\, z_0)\}$$
$$\delta(q_0, \varepsilon, z_0) = \{(q_0, z)\}$$
$$\delta(q_0, b, z) = \{(q_0, zz)\}$$
$$\delta(q_0, a, z) = \{(q_1, z)\}$$
$$\delta(q_1, b, z) = \{(q_1, \varepsilon)\}$$
$$\delta(q_1, a, z_0) = \{(q_0, z_0)\}$$

<u>Sol$^{\underline{n}}$:</u>

Let

$G = (V, T, P, S)$ generating $N(M)$

$T = \{a, b\}$

$V = S \cup \{[q_0, z_0, q_0], [q_0, z, q_0], [q_0, z_0, q_1], [q_0, z, q_1], [q_1, z_0, q_0], [q_1, z, q_0], [q_1, z_0, q_1], [q_1, z, q_1]\}$

The productions are:

S- productions:

1. $S \to [q_0, z_0, q_0] \mid [q_0, z_0, q_1]$

Other productions:

2. $\delta(q_0, b, z_0) = \{(q_0, zz_0)\}$ yields

$$[q_0, z_0, q_0] \to b[q_0, z, q_0][q_0, z_0, q_0]$$
$$[q_0, z_0, q_0] \to b[q_0, z, q_1][q_1, z_0, q_0]$$
$$[q_0, z_0, q_1] \to b[q_0, z, q_0][q_0, z_0, q_1]$$
$$[q_0, z_0, q_1] \to b[q_0, z, q_1][q_1, z_0, q_1]$$

3. $\delta(q_0, \varepsilon, z_0) = \{(q_0, \varepsilon)\}$ yields

$[q_0, z_0, q_0] \rightarrow \varepsilon$

4. $\delta(q_0, b, z) = \{(q_0, zz)\}$ yields

$[q_0, z, q_0] \rightarrow b[q_0, z, q_0][q_0, z, q_0]$

$[q_0, z, q_0] \rightarrow b[q_0, z, q_1][q_1, z, q_0]$

$[q_0, z, q_1] \rightarrow b[q_0, z, q_0][q_0, z, q_1]$

$[q_0, z, q_1] \rightarrow b[q_0, z, q_1][q_1, z, q_1]$

5. $\delta(q_0, a, z) = \{(q_1, z)\}$ yields.

$[q_0, z, q_0] \rightarrow a[q_1, z, q_0]$

$[q_0, z, q_1] \rightarrow a[q_1, z, q_1]$

6. $\delta(q_1, a, z_0) = \{(q_0, z)\}$ yields

$[q_1, z_0, q_0] \rightarrow a[q_0, z_0, q_0]$

$[q_1, z_0, q_1] \rightarrow a[q_0, z_0, q_1]$

7. $\delta(q_1, b, z) = \{(q_1, \varepsilon)\}$ yields

$[q_1, z, q_1] \rightarrow b$

Example:

Construct a CFG G which accepts $N(M)$ where

$$M = (\{q\}, \{0,1\}, \{z, A, B\}, \delta, q, z)$$

where δ is defined by

$$\delta(q,0,z) = \{(q, Az)\}, \quad \delta(q,1,z) = \{(q, Bz)\},$$

$$\delta(q,0,A) = \{(q, AA)\}, \quad \delta(q,1,A) = \{(q,\varepsilon)\},$$

$$\delta(q,0,B) = \{(q,\varepsilon)\}, \quad \delta(q,1,B) = \{(q,BB)\}$$

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

Sol^{n}:

$$G = (V, T, P, S), \quad T = \{0,1\}$$

$$V = S \cup \{[q,z,q], [q,A,q], [q,B,q]\}$$

Productions:

- S- productions:

  $S \to [q,z,q]$

- $\delta(q,0,z) = \{(q,Az)\}$ yields

  $[q,z,q] \to 0[q,A,q][q,z,q]$

- $\delta(q,1,z) = \{(q,Bz)\}$ yields

  $[q,z,q] \to 1[q,B,q][q,z,q]$

- $\delta(q,0,A) = \{(q,AA)\}$ yields

  $[q,A,q] \to 0[q,A,q][q,A,q]$

- $\delta(q,1,A) = \{(q,\varepsilon)\}$ yields

  $[q,A,q] \to 1$

- $\delta(q,0,B) = \{(q,\varepsilon)\}$ yields

  $[q,B,q] \to 0$

- $\delta(q, 1, B) = \{(q, BB)\}$ yields

  $[q, B, q] \rightarrow 1\,[q, B, q]\,[q, B, q]$

- $\delta(q, \varepsilon, z) = \{(q, \varepsilon)\}$ yields

  $[q, z, q] \rightarrow \varepsilon$

Let

$X = [q, z, q]$, $C = [q, A, q]$, $D = [q, B, q]$

Then the productions are:

$S \rightarrow X$

$X \rightarrow 0CX$

$X \rightarrow 1DX$

$C \rightarrow 0CC$

$C \rightarrow 1$

$D \rightarrow 0$

$D \rightarrow 1DD$

$X \rightarrow \varepsilon$

Thus,

$G = (\{S, C, D\}, \{0, 1\}, \{S \rightarrow 0CS \mid 1DS \mid \varepsilon,\ C \rightarrow 0CC \mid 1$
$D \rightarrow 1DD \mid 0 \mid S\})$

Example:

Let $M = (\{q_0, q_1\}, \{0,1\}, \{X, Z_0\}, \delta, q_0, Z_0, \phi)$

where δ is given by

$\delta(q_0, 0, Z_0) = \{(q_0, XZ_0)\}$, $\delta(q_1, 1, X) = \{(q_1, \varepsilon)\}$

$\delta(q_0, 0, X) = \{(q_0, XX)\}$, $\delta(q_1, \varepsilon, X) = \{(q_1, \varepsilon)\}$

$\delta(q_0, 1, X) = \{(q_1, \varepsilon)\}$, $\delta(q_1, \varepsilon, Z_0) = \{(q_1, \varepsilon)\}$

To construct a CFG $G = (V, T, P, S)$ generating $N(M)$

Sol$^{\underline{n}}$:

$T = \{0,1\}$

$V = S \cup \{[q_0, Z_0, q_0], [q_0, Z_0, q_1],$
$[q_0, X, q_0], [q_0, X, q_1],$
$[q_1, Z_0, q_0], [q_1, Z_0, q_1],$
$[q_1, X, q_0], [q_1, X, q_1]\}$

Productions in P

1. $S \to [q_0, Z_0, q_0] \mid [q_0, Z_0, q_1]$

2. $\delta(q_0, 0, Z_0) = \{(q_0, X, Z_0)\}$ induces

$[q_0, Z_0, q_0] \to 0[q_0, X, q_0][q_0, Z_0, q_0]$

$[q_0, Z_0, q_0] \to 0[q_0, X, q_1][q_1, Z_0, q_0]$

~~$\delta(q_1, 1, X) = \{(q_1, \varepsilon)\}$ yields~~

$[q_0, Z_0, q_1] \to 0[q_0, X, q_0][q_0, Z_0, q_1]$

$[q_0, Z_0, q_1] \to 0[q_0, X, q_1][q_1, Z_0, q_1]$

3. $\delta(q_1, 1, X) = \{(q_1, \varepsilon)\}$ induces

$[q_1, X, q_1] \to 1$

4. $\delta(q_0, 0, X) = \{(q_0, XX)\}$ induces

$[q_0, X, q_0] \to 0[q_0, X, q_0][q_0, X, q_0]$

$[q_0, X, q_0] \to 0[q_0, X, q_1][q_1, X, q_0]$

$[q_0, X, q_1] \to 0[q_0, X, q_0][q_0, X, q_1]$

$[q_0, X, q_1] \to 0[q_1, X, q_1][q_1, X, q_1]$

5. $\delta(q_1, \varepsilon, X) = \{(q_1, \varepsilon)\}$ yields

$[q_1, X, q_1] \to \varepsilon$

6. $\delta(q_0, 1, X) = \{(q_1, \varepsilon)\}$ yields

$[q_0, X, q_1] \to 1$

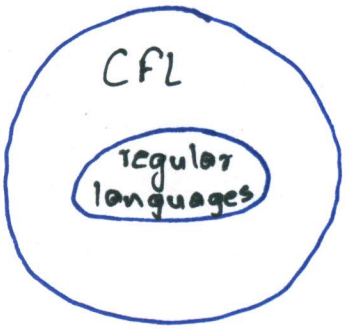
7. $\delta(q_1, \varepsilon, Z_0) = \{(q_1, \varepsilon)\}$ yields

$[q_1, Z_0, q_1] \to \varepsilon$

Week 12: Lecture Notes

Relationship of the regular and context-free languages

- All regular languages are CFL (as every FA can be trivially considered as a PDA)



- If L is NOT a CFL, then it cannot be regular. (Pumping lemma can be used to show that L is not CFL)

- CFL - is closed under
  a. Union b. Concatenation and
  c. Kleen's star

  - trivial language ϕ and $\{a\}$ are definitely CFLs
    - ϕ generated with no rules
    - $\{a\}$ generated with only one rule $S \to a$

- So the class of context-free languages must contain all regular languages, the closure of the trivial languages under these operations.

Direct Construction

- $M = (Q, \Sigma, \delta, q_0, F)$ – a DFA
- CFG : $G = (V = \{Q \cup \Sigma\}, T = \Sigma, P, S = q_0)$, P consists of

  $P = \{ q \to ap \mid \delta(q,a) = p \} \cup \{ q \to \varepsilon \mid q \in F \}$
- $L(M) = L(G)$

Properties of CFL

The (BAR-HILLEL) Pumping lemma for CFLs.

Lemma:

Let L be any CFL. Then there is a constant n, depending only on L, such that if z is in L and $|z| \geq n$, then we may write $z = uvwxy$ such that

i. $|vx| \geq 1$

ii. $|vwx| \leq n$, and

iii. $\forall i \geq 0$, uv^iwx^iy is in L

Proof:

Let G be a Chomsky Normal Form (CNF) grammar generating $L - \{\varepsilon\}$

(No grammar can be found if $L = \phi$ or $\{\varepsilon\}$

- However, if $L = \phi$, then the statement of the theorem is still valid as $\exists$ no such string z in ϕ
- CNF grammar G will actually generate $L - \{\varepsilon\}$, which is again not of importance, as we shall surely pick $n > 0$, in which case z cannot be ε any way.)

Observe that if $z \in L(G)$ and z is long, then any parse tree for must contain a long path.

More precisely we have the following claim.

Claim:

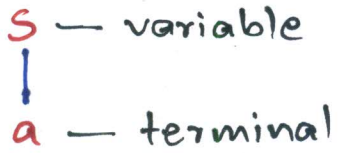
Suppose, we have a parse tree according to a CNF grammar $G = (V, T, P, S)$, and suppose that the yield of the tree is a terminal string w.

If the length of the longest path is n, then

$$|w| \leq 2^{n-1}$$

Proof: We prove it by induction on n.

Base: $n = 1$ — trivial

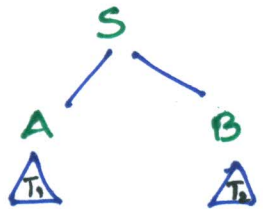


$$|a| = 1 = 2^{n-1} = 2^0$$

Induction: Suppose the longest path has length n, and $n > 1$

As $n > 1$, we cannot start the tree using a production with a terminal.

Apply induction on subtrees rooted at A and B which have no path of length $> n-1$



$\Rightarrow$ yields of subtree rooted at A as well as B have lengths $\leq 2^{n-2}$

$$|w_1| \leq 2^{n-2}, \quad |w_2| \leq 2^{n-2}$$

$\Rightarrow$ yield of the entire tree rooted at S has size

$$\leq 2^{n-2} + 2^{n-2} = 2^{n-1}$$

Hence the proof of the claim.

- Let G have K variables and let $n = 2^K$

Let z is in $L(G)$ and $|z| \geq n$.

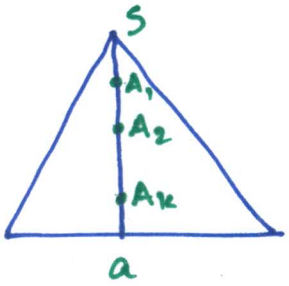
Now by the claim, any parse tree whose longest path is of length K or less must have a yield of length $2^{K-1} = n/2$ or less.

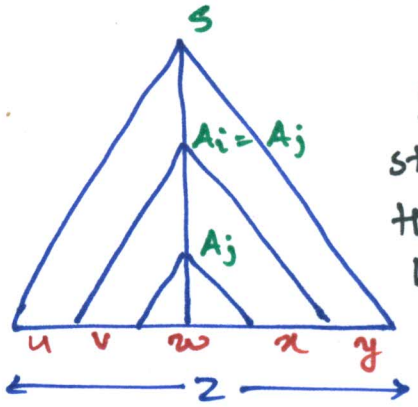
Such a parse tree cannot have yield z, because z is too long, $|z| \geq n$

Thus any parse tree with yield z has a path of length at least $K+1$

But such a path has atleast $K+2$ vertices, all but the last are labeled by variables

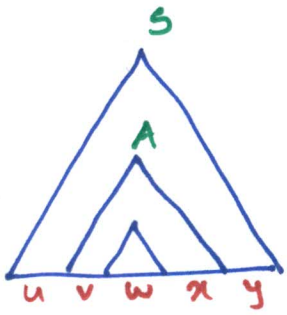
$\Rightarrow$ There must be some vertex that appears twice on the path.

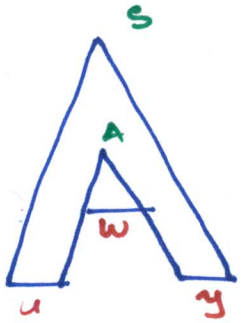


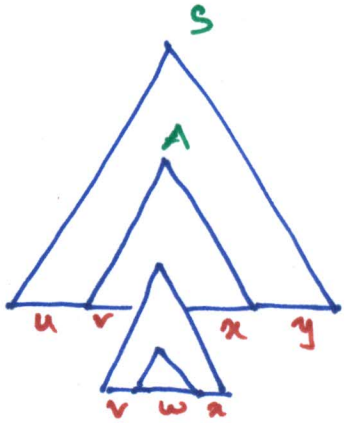


Dividing the string w so that it can be pumped

Every sufficiently long string in L must have a <u>long path</u> in it's parse tree







We can say more

Some variables must appear twice near the bottom of the path. In particular, P be a path that is longer than or as long as any path in the tree.

Then there must be two vertices v_1 and v_2 on the path satisfying the following conditions:

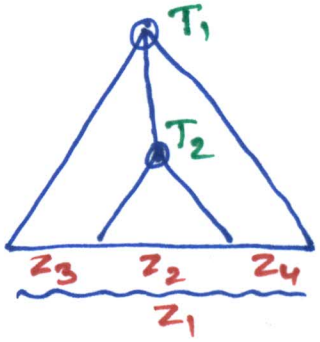
i. The vertices v_1 and v_2 both have the same label, say A

ii. Vertex v_1 is closer to the root than vertex v_2

iii. The portion of the path from v_1 to the leaf is of length at most $k+1$ (i.e. $\leq k+1$)

- To see that v_1 and v_2 can always be found, just proceed up path P from the leaf, keeping track of the labels encountered.
- Of the first $k+2$ vertices, only the leaf has a terminal label.
- The remaining $k+1$ vertices cannot have distinct variable labels
- Subtree T_1 rooted at v_1 represents the derivation of a subword of length $\leq 2^k$, as longest path in T_1 is of length $k+1$, since P was a path of longest length in the entire tree.
- Let z_1 be the yield of subtree T_2
- If T_2 is rooted at v_2 and z_2 is the yield of T_2, then

$z_1 = z_3 z_2 z_4$



- Furthermore, z_3, z_4 cannot be both ε as the first derivation of z_1 must be of the form $A \to BC$, for some variables B, C
- The subtree T_2 must be completely within either the subtree generated by B or the subtree generated by C

We now know that

$$A \overset{*}{\underset{G}{\Rightarrow}} z_3 A z_4 \text{ and } A \overset{*}{\underset{G}{\Rightarrow}} z_2, \text{ where } |z_2 z_3 z_4| \leq 2^k = n$$

But it follows that

$$A \overset{*}{\underset{G}{\Rightarrow}} z_3^i z_2 z_4^i \text{ for each } i \geq 0$$

Thus the string z can clearly be written as

$$z = u z_3 z_2 z_4 y \text{ for some } u, y$$

We let $z_3 = v$, $z_2 = w$, $z_4 = x$ to complete the proof.

Applications of the Pumping for CFL's

(Proof of non-context-freeness of a language)

Example: $L = \{0^n 1^n 2^n \mid n \geq 1\}$ is not a CFL.

Let L be CFL and n be the constant of the pumping lemma for CFLs

Consider $z = 0^n 1^n 2^n$

Write $z = uvwxy$, where $|vwx| \leq n$ and $vx \neq \varepsilon$

Note that vwx cannot involve both 0's and 2's.

Since the last 0 and the first 2 are separated by $n+1$ positions:

1. vwx has no 2's - Then vx consists of only 0's and 1's and has atleast one of these symbols. Then uwy which would have to be in L by

the PL, has n 2's, but fewer than n 0's or fewer than n 1's or both. It therefore doesnot belong in L.

2. vwx has no 0's - similarly uwy has n 0's, but fewer 1's or fewer 2's or both. It therefore cannot be in L

We get contradiction in both, hence L is not a CFL.

Example: $L = \{x \in \{0,1\}^* \mid |x| \text{ is a perfect square}\}$

Let L be a CFL and n be the constant of the PL for the CFLs

Consider $z = 0^{n^2}$

Write $z = uvwxy$ with $|vwx| \leq n$ and $vx \neq \varepsilon$

Let $|vx| = m$, Then $m \leq n$

Now consider the string uv^2wx^2y.

By the PL for CFLs, $uv^2wx^2y \in L$

But $n^2 < |uv^2wx^2y| = n^2 + m \leq n^2 + n < (n+1)^2$

$\Rightarrow |uv^2wx^2y|$ is not a perfect square.

Therefore $uv^2wx^2y \notin L$

Which is a contradiction to the PL for the CFLs.

$\therefore$ L is not a CFL

Example: $L = \{a^i b^j c^i d^j \mid i \geq 1, j \geq 1\}$ is not CFL

Let L be CFL and n be the constant of the PL for CFLs

Consider $z = a^n b^n c^n d^n = uvwxy$ with $|vwx| \leq n$ and

$|vx| \geq 1$ and $uv^i wx^i y \in L \ \forall i \geq 0$

Then vx will have at most two different symbol.

If vx has two different symbols, then they are consequtive

i.e. either a, b, or b, c or c, d

- vx has only a's – then uwy has fewer a's than c's

Similarly, if vx has only b's or only c's or only a's

$\Rightarrow uwy \notin L$

- vx has only a's and b's – then also uwy has fewer a's, then

similarly if vx has only b's and c's or c's and d's

$\Rightarrow uwy \notin L$

Thus L is not CFL.

Example: $L = \{a^p \mid p \text{ is a prime}\}$ is not a CFL

Let L be a CFG and n be the constant of the PL for CFL's

Let $z = a^p \in L$, $p > n$

By PL for CFL, $z = uvwxy$ with $|vwx| \leq n$, $|vx| \geq 1$

and $uv^i wx^i y \in L \ \forall i \geq 0$

for $i = 0$, $uv^0 wx^0 y = uwy \in L \Rightarrow |uwy| = q$ must be a prime

Let $|vx| = r$.

Then $|uv^q wx^q y| = |uwy| + q|vx| = q + rq$

$\neq$ a prime

$\Rightarrow uv^q wx^q y \notin L$

Ogden's Lemma

Let L be a CFL. Then there is a constant n (which may infact be the same as for the Pumping Lemma) such that if z is any word in L with $|z| \geq n$, and we mark any n or more positions of z "distinguished", then we can write $z = uvwxy$ such that

i. vx has atleast one distinguished position

ii. vwx has atmost distinguished position

iii. for all $i \geq 0$, $uv^iwx^iy \in L$

- Bar-Hillel Pumping lemma for CFL is a special case of Ogden's Lemma, in which all positions are distinguished.

- There are certain non-CFL's for which Bar-Hillel pumping lemma for CFLs is of no use.

Example:

$L = \{a^i b^j c^k d^l \mid \text{either } i = 0 \text{ or } j = k = l\}$ is not CFL

- Let $z = b^j c^k d^l = uvwxy \in L$

Then it is always possible to choose u, v, w, x, y so that $uv^m wx^m y \in L \ \forall m$

for example, if we choose vwx to have only one b's

- Let $z = a^i b^j c^j d^j = uvwxy$

Then v, x might consist only of a's in which case

$$uv^m wx^m y \in L \ \forall m$$

$\Rightarrow$ PL is of no use