Closure properties of CFLs

- There are some differences when compared with regular languages.
- We deviate from both text books (to simplify the things).
- We want to skip homomorphism.

CFLs are --

- Closed under
  - Union
  - Concatenation
  - Kleene star
  - Reversal
  - Intersection with regular languages
  -
  -

- Not closed under
  - Intersection
  - Difference
  - Complement
  - Repetition
  -
  -

CFLs are closed under Union

- L_1 and L_2 are CFLs $\Rightarrow L_1 \cup L_2$ is a CFL.

CFLs are closed under Union

- $L_1$and L_2 are CFLs $\Rightarrow L_1 \cup L_2$ is a CFL.
- Proof: Let the CFG for these are $G_1 = (V_1, T_1, P_1, S_1)$ and $G_2 = (V_2, T_2, P_2, S_2)$.

CFLs are closed under Union

- $L_1$and L_2 are CFLs $\Rightarrow L_1 \cup L_2$ is a CFL.
- Proof: Let the CFG for these are $G_1 = (V_1, T_1, P_1, S_1)$ and $G_2 = (V_2, T_2, P_2, S_2)$.
- We rename variables so that V_1 and V_2 such that they are disjoint and does not have a variable with name S. Note T_1 and T_2 need not be disjoint. Perhaps they are same.
- Now the grammar $(V_1 \cup V_2 \cup \{S\}, T_1 \cup T_2, P_1 \cup P_2 \cup \{S \to S_1 | S_2\}, S)$ will generate the union.

Closed under concatenation

- Add the production $S \to S_1 S_2$

Closed under Kleene star

- Add productions $S \rightarrow S_1 S | \epsilon$

Closed under reversal

Theorem 7.25: If L is a CFL, then so is L^R.

PROOF: Let $L = L(G)$ for some CFL $G = (V, T, P, S)$. Construct $G^R = (V, T, P^R, S)$, where P^R is the "reverse" of each production in P. That is, if $A \to \alpha$ is a production of G, then $A \to \alpha^R$ is a production of G^R. It is an easy induction on the lengths of derivations in G and G^R to show that $L(G^R) = L^R$. Essentially, all the sentential forms of G^R are reverses of sentential forms of G, and vice-versa. We leave the formal proof as an exercise. $\square$

Not closed under intersection☹

Not closed under intersection☹

- Proof [by counter example]:

$L = \{0^n 1^n 2^n \mid n \geq 1\}$ is not a context-free language.

However, the following two languages *are* context-free:

$$L_1 = \{0^n 1^n 2^i \mid n \geq 1, i \geq 1\}$$
$$L_2 = \{0^i 1^n 2^n \mid n \geq 1, i \geq 1\}$$

- And, $L = L_1 \cap L_2$

A grammar for L_1 is:

$$
\begin{aligned}
&S \rightarrow AB \\
&A \rightarrow 0A1 \mid 01 \\
&B \rightarrow 2B \mid 2
\end{aligned}
$$

In this grammar, A generates all strings of the form $0^n 1^n$, and B generates all strings of 2's.

A grammar for L_2 is:

$$
\begin{aligned}
&S \rightarrow AB \\
&A \rightarrow 0A \mid 0 \\
&B \rightarrow 1B2 \mid 12
\end{aligned}
$$

CFLs are closed when intersected with regular languages

Theorem 7.27: If L is a CFL and R is a regular language, then $L \cap R$ is a CFL.

CFLs are closed when intersected with regular languages

Theorem 7.27: If L is a CFL and R is a regular language, then $L \cap R$ is a CFL.

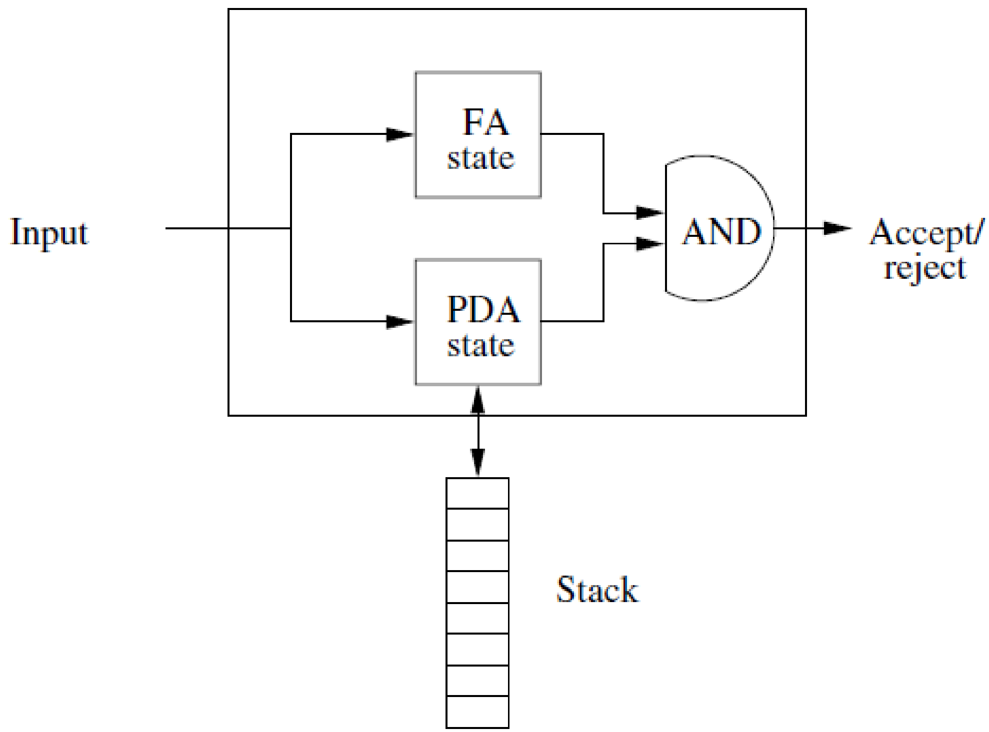

Figure 7.9: A PDA and a FA can run in parallel to create a new PDA

Formally, let $P = (Q_P, \Sigma, \Gamma, \delta_P, q_P, Z_0, F_P)$

be a PDA that accepts L by final state,

Formally, let $P = (Q_P, \Sigma, \Gamma, \delta_P, q_P, Z_0, F_P)$

be a PDA that accepts L by final state,

and let $A = (Q_A, \Sigma, \delta_A, q_A, F_A)$ be a DFA for R.

Formally, let $P = (Q_P, \Sigma, \Gamma, \delta_P, q_P, Z_0, F_P)$

be a PDA that accepts L by final state,

and let $A = (Q_A, \Sigma, \delta_A, q_A, F_A)$ be a DFA for R.

Construct PDA $P' = (Q_P \times Q_A, \Sigma, \Gamma, \delta, (q_P, q_A), Z_0, F_P \times F_A)$

Formally, let $P = (Q_P, \Sigma, \Gamma, \delta_P, q_P, Z_0, F_P)$

be a PDA that accepts L by final state,

and let $A = (Q_A, \Sigma, \delta_A, q_A, F_A)$ be a DFA for R.

Construct PDA $P' = (Q_P \times Q_A, \Sigma, \Gamma, \delta, (q_P, q_A), Z_0, F_P \times F_A)$

We need to define δ for the PDA.

Formally, let $P = (Q_P, \Sigma, \Gamma, \delta_P, q_P, Z_0, F_P)$

be a PDA that accepts L by final state,

and let $A = (Q_A, \Sigma, \delta_A, q_A, F_A)$ be a DFA for R.

Construct PDA $P' = (Q_P \times Q_A, \Sigma, \Gamma, \delta, (q_P, q_A), Z_0, F_P \times F_A)$

We need to define δ for the PDA.

Note that, PDA can change its state with ϵ from input, but DFA cannot. But we can use $\hat{\delta}$, the extended transition for DFA which says $\hat{\delta}(p, a) = p, \text{if } a = \epsilon$

Formally, let $P = (Q_P, \Sigma, \Gamma, \delta_P, q_P, Z_0, F_P)$

be a PDA that accepts L by final state,

and let $A = (Q_A, \Sigma, \delta_A, q_A, F_A)$ be a DFA for R.

Construct PDA $P' = (Q_P \times Q_A, \Sigma, \Gamma, \delta, (q_P, q_A), Z_0, F_P \times F_A)$

where $\delta\big((q,p), a, X\big)$ is defined to be the set of all pairs $\big((r,s), \gamma\big)$ such that:

1. $s = \hat{\delta}_A(p, a)$, and
2. Pair (r, γ) is in $\delta_P(q, a, X)$.

Formally, let $P = (Q_P, \Sigma, \Gamma, \delta_P, q_P, Z_0, F_P)$

be a PDA that accepts L by final state,

and let $A = (Q_A, \Sigma, \delta_A, q_A, F_A)$ be a DFA for R.

Construct PDA $P' = (Q_P \times Q_A, \Sigma, \Gamma, \delta, (q_P, q_A), Z_0, F_P \times F_A)$

where $\delta\big((q,p), a, X\big)$ is defined to be the set of all pairs $\big((r,s), \gamma\big)$ such that:

1. $s = \hat{\delta}_A(p, a)$, and
2. Pair (r, γ) is in $\delta_P(q, a, X)$.

We are allowing a to be ϵ

Formally, let $P = (Q_P, \Sigma, \Gamma, \delta_P, q_P, Z_0, F_P)$

be a PDA that accepts L by final state,

and let $A = (Q_A, \Sigma, \delta_A, q_A, F_A)$ be a DFA for R.

Construct PDA $P' = (Q_P \times Q_A, \Sigma, \Gamma, \delta, (q_P, q_A), Z_0, F_P \times F_A)$

where $\delta\big((q,p), a, X\big)$ is defined to be the set of all pairs $\big((r,s), \gamma\big)$ such that:

1. $s = \hat{\delta}_A(p, a)$, and
2. Pair (r, γ) is in $\delta_P(q, a, X)$.

We are allowing a to be ϵ

We could have taken NFA for R instead of a DFA, But the things will be more complicated (it can be done). Note, product of DFAs is simple; can be extended to NFAs also, but things are complicated !!

An application

- Dyck set (strings of balanced parentheses) is a CFL
- $(^*)^*$ is a regular language
- Intersection of above two is a CFL .
  - That is $\{ (^k \ )^k \mid k \geq 1\}$ is a CFL

To show “not CFL”

- The language L where each string has equal number of 0s, 1s and 2s (may be interleaved) is not CFL.

To show “not CFL”

- The language L where each string has equal number of 0s, 1s and 2s (may be interleaved) is not CFL.

- Take $R = 0^*1^*2^*$ and intersect with L.
- The result, if L is a CFL must be CFL.

To show "not CFL"

- The language L where each string has equal number of 0s, 1s and 2s (may be interleaved) is not CFL.

- Take $R = 0^*1^*2^*$ and intersect with L.
- The result, if L is a CFL must be CFL.
- But the result, which is $\{0^n1^n2^n | n \geq 1\}$ is not a CFL. Contradiction.

To show "not CFL"

- The language L where each string has equal number of 0s, 1s and 2s (may be interleaved) is not CFL.

- Take $R = 0^*1^*2^*$ and intersect with L.
- The result, if L is a CFL must be CFL.
- But the result, which is $\{0^n1^n2^n | n \geq 1\}$ is not a CFL. Contradiction.

Of course, pumping lemma for CFLs can directly applied on L and shown to fail.

CFLs are not closed under complementation

- We know $L = \{ww | w \in \{0,1\}^* \}$ is not a CFL.

CFLs are not closed under complementation

- We know $L = \{ww | w \in \{0,1\}^* \}$ is not a CFL.
- But its complement is a CFL !!
- This is a proof by counter example.

CFG for the $\overline{L}$

- $S \rightarrow S_o | S_e$
- $S_o \rightarrow 0R|1R|0|1,\ R \rightarrow 0S_o|1S_o$
- $S_e \rightarrow XY|YX,\ X \rightarrow ZXZ\ |0\ ,$
  $Y \rightarrow ZYZ\ |1\ , Z \rightarrow 0|1$

- S_o generates odd length strings, whereas S_e generates even length strings.

Other proof

- Other proof is through contradiction.
- If it is closed under complementation then it has to be closed under intersection.

Other proof

- Other proof is through contradiction.
- If it is closed under complementation then it has to be closed under intersection.
- Depends on the identity

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

Not closed under set difference

- Proof by contradiction.
- If closed under set difference, then it has to be closed under complementation.

Not closed under set difference

- Proof by contradiction.
- If closed under set difference, then it has to be closed under complementation.
- Assuming it is closed under set difference.
- Let $L_1 = \Sigma^*$ which is a CFL
- Consider any other CFL L_2
- Then $L_1 - L_2$ which is the complement of L_2 must be a CFL. Contradiction.

Set difference with regular is okay☺

- If L is a CFL and R is a regular language, then $L - R$ is a CFL.
- $L - R = L \cap \bar{R}$
- If R is regular then $\bar{R}$ is regular.