Chapter 17: Context-Free Languages

I. Theory of Automata

→ II. Theory of Formal Languages

III. Theory of Turing Machines …

- Theorem. The set of context-free languages is closed under union, concatenation, and Kleene closure.

- Union. L_1+L_2

  L_1 and L_2 are generated by two context-free grammars G_1 and G_2. We replace each nonterminal X in G_1 by X_1, and each nonterminal X in G_2 by X_2. We add the productions:

  $S \to S_1 \quad S \to S_2$

  L_1+L_2 is the language generated by this new CFG.

Example: L_1 = PALINDROME: $S \to aSa \mid bSb \mid a \mid b \mid \Lambda$

$L_2 = a^nb^n$: $S \to aSb \mid \Lambda$

L_1+L_2 : $S \to S_1 \mid S_2$

$S_1 \to aS_1a \mid bS_1b \mid a \mid b \mid \Lambda$

$S_2 \to aS_2b \mid \Lambda$

Example: $L_1 = \{aa,bb\}$: $S \to aA \mid bB \quad A \to a \quad B \to b$

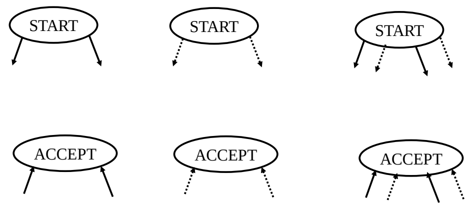
$L_2 = \{\Lambda\}$: $S \to \Lambda$

L_1+L_2 : $S \to S_1 \mid S_2$

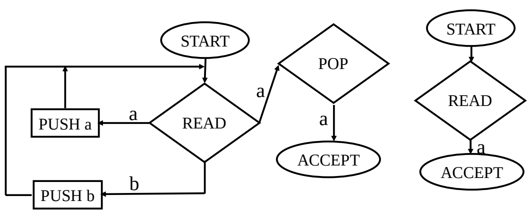
$S_1 \to aA_1 \mid bB_1 \quad A_1 \to a \quad B_1 \to b$

$S_2 \to \Lambda$

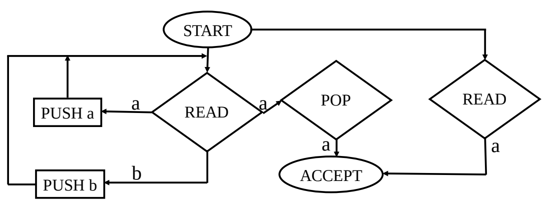
Proof by machines



(contain aa) (begin with a)



L1 + L2



- Concatenation. L_1L_2

  Similar to union except we add:

  $S \to S_1S_2$
- Kleene star. L^*

  We replace S by S_1 and add:

  $S \to S_1S \mid \Lambda$

  $S \Rightarrow S_1S \Rightarrow S_1S_1S \Rightarrow S_1S_1S_1S \Rightarrow \ldots$

- Theorem. The intersection of two context-free languages may or may not be context-free.

- $L1 = a^nb^na^m$

  $S \to XA$

  $X \to aXb$ / ab

  $A \to aA$ / a
- $L2 = a^nb^ma^m$

  $S \to AX$

  $X \to bXa$ / ba

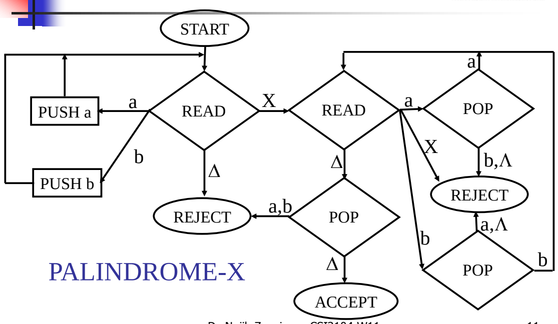
  $A \to aA$ / a
- $L1 \cap L2 = a^nb^na^n$

  (is not a context-free language)

- Theorem. The intersection of a context-free language and a regular language is always context-free.

Proof: By constructive algorithm using pushdown automata, similar to the intersection algorithm for two finite automata.

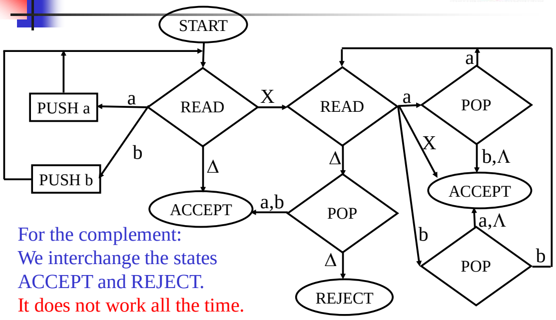
(see manual)



PALINDROME-X

- Theorem. The complement of a context-free language may or may not be context-free.

When the PDA is deterministic with other properties, we could use for the complement a similar technique as for FA



For the complement:
We interchange the states
ACCEPT and REJECT.
It does not work all the time.