cse303
ELEMENTS OF THE THEORY OF COMPUTATION

Professor Anita Wasilewska

LECTURE 11

CHAPTER 3
CONTEXT-FREE LANGUAGES

1. Context-free Grammars
2. Parse Trees
3. Pushdown Automata
4. Pushdown automata and context -free grammars
5. Languages that are not context- free

CHAPTER 3
Review Exercises

Exercises

Exercise 1

Prove that the language

$$L = \{a^m b^n : \quad m \geq n\}$$

is context- free

We construct a grammar G such that

$$L(G) = \{a^m b^n : \quad m \geq n\}$$

as follows

$$G = (V, \Sigma, R, S)$$

where $V = \{a, b, S\}, \quad \Sigma = \{a, b\}$

$$R = \{S \to aS \mid aSb \mid e\}$$

Derivation examples

$$S \Rightarrow aS \Rightarrow aaS \Rightarrow aaaSb \Rightarrow aaab$$

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabbb$$

Exercises

Exercise 2

Construct a grammar G such that

$$L(G) = \{a^m b^n : \quad m > n\}$$

We adopt rules

$$R = \{S \to aS \mid aSb \mid a\}$$

Exercise 3

Prove that

$$L = \{a^m b^n : \quad m < n\}$$

is context- free

Observe that

$$L = \{a^n b^n : \; n \geq 0\}\{b\}^+$$

and hence L is context- free as regular language is context-free and context-free languages are **closed** under concatenation

Exercises

Exercise 4

Prove that

$$L = \{a^m b^n : \quad m \neq n\}$$

is context- free

Observe that

$$\{a^m b^n : \quad m \neq n\} = \{a^m b^n : \quad m > n\} \cup \{a^m b^n : \quad m < n\}$$

and hence L is context- free as context-free languages are **closed** under union and we have just proved that both union components are context-free

Exercises

Exercise 5

We proved that

The context-free languages are **not closed** under complementation

We know that

$$L = \{a^n b^n : \ n \geq 0\}$$

is context- free

Prove that the complement of L is also **context- free**

Observe that

$$\Sigma^* - L = \{a^m b^n : \ m \neq n\} \cup \Sigma^* a \Sigma^* b \Sigma^* a \Sigma^* \cup \Sigma^* b \Sigma^* a \Sigma^* b \Sigma^*$$

and hence $\Sigma^* - L$ is context- free as **union** of context-free languages

Exercises

Exercise 6

Let L be a context-free language and let R be regular

Prove that $L - R$ is context-free

What about $R - L$?

Observe that

$$L - R = L \cap (\Sigma^* - R)$$

and regular languages are **closed** under intersection and

Theorem 3

The **intersection** of a context-free language with a regular language is a context-free language

$R - L$ does not need to be CF

Take $R = \Sigma^*$

$$R - L = \Sigma^* - L$$

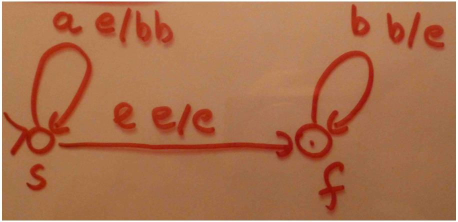
and CF languages are **not closed** under complementation

Exercise 7

Construct a PD automaton M such that

$$L(M) = \{a^n b^{2n} : \quad n \geq 0\}$$

The **diagram** of M is



The components of M are

$$K = \{s, f\}, \quad \Sigma = \{a, b\}, \ \Gamma = \{a\}, \ s, \ F = \{f\}$$

Exercises

The set Δ of transitions is

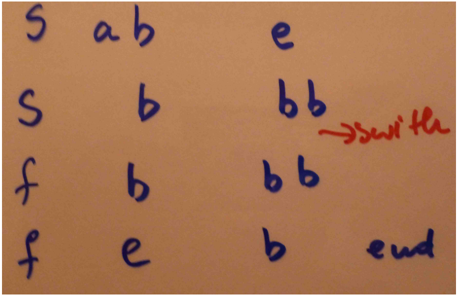
$$\Delta = \{((s,a,e),(s,bb)),\ \ ((s,e,e),(f,e)),\ \ ((f,b,b),(f,e))\}$$

Show that $ab \notin L(M)$

M is **non-deterministic**, so we have to consider **all possible** computations M starting at the word ab

Here they are:

1.



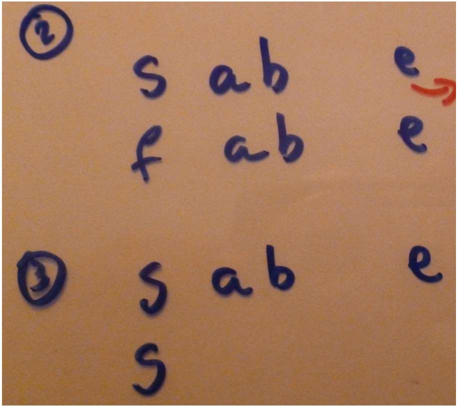
Exercises

The set Δ of transitions is

$$\Delta = \{((s, a, e), (s, bb)), \quad ((s, e, e), (f, e)), \quad ((f, b, b), (f, e))\}$$

There are only two more "switches"

2. and **3.**



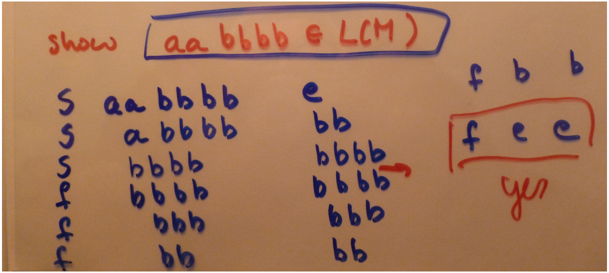
Exercises

The set Δ of transitions is

$$\Delta = \{((s, a, e), (s, bb)), \quad ((s, e, e), (f, e)), \quad ((f, b, b), (f, e))\}$$

Show that $aabbbb \in L(M)$

Here is a computation of M accepting *aabbbb*



Configuration and Transition

In order to define a notion of computation of M on an input string $w \in \Sigma^*$ we introduce, as always, a notion of a **configuration** and **transition** relation

A **configuration** is any tuple

$$(q, w, \gamma) \in K \times \Sigma^* \times \Gamma^*$$

where $q \in K$ represents a current state of M and $w \in \Sigma^*$ is unread part of the input, and γ is a content of the stack read top-down

The **transition relation** acts between two **configurations** and hence $\vdash_M$ is a certain binary relation defined on $K \times \Sigma^* \times \Gamma^*$, i.e.

$$\vdash_M \subseteq (K \times \Sigma^* \times \Gamma^*)^2$$

Formal definition follows

Transition Relation

Given

$$M = (K, \Sigma, \Gamma, \Delta, s, F))$$

Transition relation

Definition

For any $p, q \in K, \ u, x \in \Sigma^*, \ \alpha, \beta, \gamma$

$$(p, ux, \beta\alpha) \vdash_M (q, x, \gamma\alpha)$$

if and only if

$$((p, u, \beta), (q, \gamma)) \in \Delta$$

Language L(M)

We **denote** as usual, the reflexive, transitive closure of $\vdash_M$ denoted by $\vdash_M{}^*$ and define

Definition

$$L(M) = \{w \in \Sigma^* : \ (s, \ w, \ e) \vdash_M{}^* (p, \ e, \ , e) \ \text{ for certain } p \in F\}$$

M **accepts** $w \in \Sigma^*$ if and only if $w \in L(M)$

In plain English:

$(s, \ w, \ e)$ means:

start with w and empty **stack**

$(p, \ e, \ e)$ for certain $p \in F$ means:

finish in a final state after reading w and emptying all of the **stack**

Exercises

Exercise

Here is M

M operates as follows

Δ **pushes** *aa* on the top of the stock while *M* is reading *b*,

switches to *f* (final state) **non-deterministically**;

and **pops** *a* while **reading** *a* - all in final state

M **puts** on the stock **two** *a*'s for each *b*,

and then **removes** all *a*'s from the stock **comparing** them with *a*'s included in the word

all theses actions while *M* is in the final state

Write L(M)

$$L(M) = \{b^n a^{2n} : \ n \geq 0\}$$

Exercises

Trace formally a computation of M that leads to the **acceptance** of the string bbaaaa

The **accepting** computation is:

$$(s, bbaaaa, e) \vdash_M (s, baaaa, aa) \vdash_M (s, aaaa, aaaa)$$

$$\vdash_M (f, aaaa, aaaa) \vdash_M (f, aaa, aaa)$$

$$\vdash_M (f, aa, aa) \vdash_M (f, a, a) \vdash_M (f, e, e)$$

Alternative definition of $M = (K, \ \Sigma, \ \Gamma, \ \Delta, \ s, \ F)$ is

$$\Delta = \{((s, b, e), (s, b)), \ ((s, e, e), (f, e)), \ ((f, aa, b), (f, e))\}$$

Exercise 8

Prove that L(G) is NOT regular, for

$$G = (V,\ \Sigma,\ S,\ R)$$

where $V = \{S,\ (,)\},\ \ \Sigma = \{(,)\}$

$$R = \{S \rightarrow\ SS \mid (S) \mid e\}$$

Proof by contradiction

Assume that L(G) is **regular**. The language $L_1 = (^*)^*$ is regular and regular languages are closed under $\cap$, so

$$L(G) \cap L_1 = (^n)^n$$

is regular. **Contradiction**

Exercises

Exercise 9

Given a context-free grammar

$$G = (V,\ \Sigma,\ S,\ R)$$

where $V = \{S,\ (,)\},\ \ \Sigma = \{(,)\}$

$$R = \{S \rightarrow\ SS \mid (S) \mid e\}$$

1. **Construct** a PD automaton M, such that

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

2. **Show** that $()() \in L(M)$

Exercises

We use construction described in the proof of our **Main Theorem**, in particular, in the proof of the its first part, i.e.

Lemma 1

Each context free language is **accepted** by **some** PD automaton

Proof

Let $G = (V, \Sigma, R, S)$ be a context-free grammar; we construct a PD automaton M, such that L(G) = L(M) as follows

M has only two states: initial s and a final f

M remains in state f after its first move

Δ contains the following transitions

1. ((s, e, e), (f, S))

2. ((f, e, A), (f, x)) - for each rule $A \rightarrow x$

3. ((f, c, c), (f, e)) - for each $c \in \Sigma$

Exercises

The rules of G are:

$$R = \{S \to \ SS \mid (S) \mid e\}$$

General case: Δ contains the following transitions

1. ((s, e, e), (f, S))

2. ((f, e, A), (f, x)) - for each rule $A \to x$

3. ((f, c, c), (f, e)) - for each $c \in \Sigma$

The transitions of M are

$$\Delta = \{ ((s, e, e), (f, S)), \ ((f, e, S), (f, SS)),$$

$$((f, e, S), (f, (S))), \ ((f, e, S), (f, e)),$$

$$((f, (, ( ), (f, e)), \ ((f,),)), (f, e)) \}$$

Exercises

We **trace formally** a computation of M that leads to the **acceptance** of the string ()()

The **accepting** computation is:

$$(s, ()(), e) \vdash_M (f, ()(), S)) \vdash_M (f, ()(), SS))$$

$$\vdash_M (f, ()(), (S)S) \vdash_M (f,)(), S)S) \vdash_M (f,)(),)S))$$

$$\vdash_M (f, (), S) \vdash_M (f, (), (S)) \vdash_M (f,), S)) \vdash_M (f,),)) \vdash_M (f, e, e)$$

We proved that

$$()() \in L(M)$$

Exercise 9

Construct a regular grammar G such that

$$L(G) = b^* \cup a$$

We use the **closure** of CF languages over over UNION construction

Let $G_1,\ G_2$ be two regular grammars with has sets of rules

$$R_1:\quad S_1 \to bS_1 \mid e$$

$$R_2:\quad S_2 \to a$$

Obviously

$$L(G_1) = b^* \quad \text{and} \quad L(G_2) = a$$

Exercises

We construct $G = G_1 \cup G_2$ as follows

We **add** new initial state S such that $S \neq S_1, S_2$, and make S_1, S_2 the internal states

The rules for $G = G_1 \cup G_2$ are

$$S \to S_1 \,|S_2, \quad S_1 \to bS_1 \mid e, \quad S_2 \to a$$

We re-write them as

$$S \to A \,|B, \quad B \to bB \mid e, \quad A \to a$$

Here is another, direct grammar G with rules

$$S \to B \,|a, \quad B \to bB \mid e$$

Exercises

Exercise 10

Let G be a CF grammar with $\Sigma = \{a, b\}$ the following rules

$$S \rightarrow aSb \mid A \mid B$$

$$A \rightarrow abS \mid aSBb$$

$$B \rightarrow AB \mid Ba \mid bSaB$$

Describe L(G)

By definition

$$L(G) = \{w \in \Sigma^* : S \overset{*}{\underset{G}{\Rightarrow}} w\}$$

and hence we have that

$$L(G) = \emptyset$$

Exercises

Exercise 11

Let G be a CF grammar with $\Sigma = \{a, b\}$ the following rules

$$S \rightarrow aaA \mid B \mid abB \mid e$$

$$A \rightarrow bS \mid a$$

$$B \rightarrow bS$$

Construct a **nondeterministic** finite automaton M such that

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

We follow the poof of the **L-G Theorem**

L-G Theorem

Language L is **regular** if and only if there exists a **regular** grammar G such that

$$L = L(G)$$

Proof part 1

Suppose that L is **regular**; then L is accepted by a **deterministic** finite automaton

$$M = (K,\ \Sigma,\ \delta,\ s,\ F)$$

We **construct** a regular grammar G as follows

$$G = (V,\ \Sigma,\ R,\ S)$$

for $V = \Sigma \cup K, \quad S = s$

$$R = \{q \ \rightarrow\ ap : \quad \delta(q, a) = p\} \ \cup\ \{q \ \rightarrow\ e : \quad q \in F\}$$

L-G Theorem Part 2

Proof part 2

Let now G be any **regular** grammar

$$G = (V, \Sigma, R, S)$$

We define a **nondeterministic** automaton M such that

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

as follows

$$M = (K, \Sigma, \Delta, s, F)$$

$K = (V - \Sigma) \cup \{f\}$ where f is a new element

$$s = S, \quad F = \{f\}$$

Proof of L-G Theorem Part 2

The set Δ of transitions is

$$\Delta = \{(A, w, B) : \ A \rightarrow wB \in R; \ A, B \in V - \Sigma, \ \ w \in \Sigma^*\}$$

$$\cup \{(A, w, f) : \ A \rightarrow w \in R; \ A, B \in V - \Sigma, \ \ w \in \Sigma^*\}$$

Once again, derivations are mimicked by the moves, i.e, for any

$A_1, \ldots, A_n \in V - \Sigma, \ \ w_1, \ldots w_n \in \Sigma^*$

$$A_1 \Rightarrow_G \ w_1 A_2 \ \Rightarrow_G \cdots \Rightarrow_G \ w_1 \ldots w_{n-1} A_n \Rightarrow_G \ w_1 \ldots w_n$$

if and only if

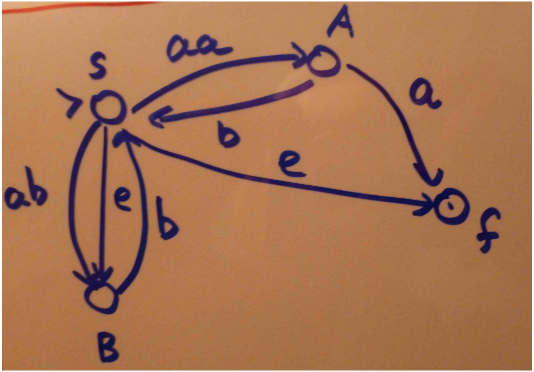
$$(A_1, w_1 \ldots w_n) \vdash_M (A_2, w_2 \ldots w_n) \vdash_M \ \ldots \vdash_M (A_n, w_n) \vdash_M (f, e)$$

Exercise 11 Solution

Rules of G

$$S \rightarrow aaA \mid B \mid abB \mid e, \;\; A \rightarrow bS \mid a, \;\; B \rightarrow bS$$

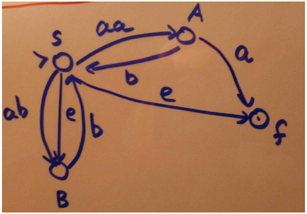
The **diagram** of M is



Trace a computation of M that leads to the acceptance of abbaaa and **compare** it with derivation of abbaaa in G

Exercises

The **diagram** of M is



Here is the **computation** of M and **derivation** of in G