

# CS 121 Section 5

Harvard University

October 11 & 12, 2012

## Overview

This week we will focus on reviewing the core concepts involved with PDAs, Pumping lemma for CFL, and closure properties of context-free language.

## 1 Concept Review

### 1.1 PDA

Intuitively a Pushdown Automata (PDA) represents a NFA with an additional (unbounded size) stack. The addition of the stack makes PDAs much powerful than NFA. For example, the language  $L = \{a^n b^n : n \in \mathbb{N}\}$  is recognized by a PDA, but not by any NFA. Intuitively, PDAs the stack gives PDAs the ability to perform simple counting tasks.

### 1.2 PDAs v.s. CFGs

PDAs and Context-Free Grammars are equivalent in power.

**Theorem 1.1.** *The CFLs are the languages accepted by PDAs.*

### 1.3 Pumping lemma for Context-Free Language

**Lemma 1.1.** *If  $L$  is context-free, then there is a number  $p$  such that any  $s \in L$  of length at least  $p$  can be divided into  $s = wxyz$ , where*

- $v \neq \epsilon$  or  $y \neq \epsilon$ ,
- $|vxy| \leq p$ , and
- $w^i x y^i z \in L$  for every  $i \geq 0$ .

## 1.4 Closure properties of Context-Free Language

The Context-Free Languages are closed under:

- Union
- Concatenation
- Kleene \*
- Intersection with a regular set

The Context-Free Languages are **not** closed under:

- Complement
- Intersection

## 2 Exercises

**Exercise 2.1.** *Explain and justify the following statement: "Almost all languages are not context-free."*

**Exercise 2.2.** *Show that  $L = \{a^n b^{2n}\}$  is context-free by giving a PDA that accepts it. Draw the state diagram and write the 6-tuple  $(Q, \Sigma, \Gamma, \delta, q_0, F)$*

**Exercise 2.3.** *Determine, with proof, whether or not each of the following languages is context-free.*

1.  $\{ww : w \in \Sigma^*\}$
2.  $L = \{w \text{ is not of the form } a^n b^n\}$
3.  $\{w \in \{(,)\}^* : w \text{ is not properly parenthesized}\}$
4.  $L = \{a^i b^j c^k \mid 0 \leq i \leq j \leq k\}$

**Exercise 2.4.** *DPDAs*

1. *Give a formal definition of a deterministic pushdown automaton (DPDA). (Note, there are a few different ways of doing this. Some definitions are equivalent, but some are not. However, any definition that eliminates nondeterminism from PDAs will yield less powerful machines.)*
2. *Now show that the DPDAs defined in part 1 are weaker than PDAs.*