CS310 Automata Theory – 2016-2017

Nutan Limaye

Indian Institute of Technology, Bombay
nutan@cse.iitb.ac.in

Lecture 15: Extensions of DFA/NFAs
February 06, 2017

Last class

Proof of the main claim: The class of languages recognized by 2DFAs is exactly REG.

Pushdown automata: NFA + Stack

Formal definition of 2DFA

Definition

A 2DFA $A = (Q, \Sigma \cup \{\#, \$\}, \delta, q_0, q_{acc}, q_{rej})$, where

Q :	set of states,	Σ:	input alphabet
$\#$:	left endmarker	$\$$:	right endmarker
q_0:	start state		
q_{acc}:	accept state	q_{rej}:	reject state

$$\delta : Q \times (\Sigma \cup \{\#, \$\} \to Q \times \{L, R\}$$

The following conditions are forced:

$\forall q \in Q$, $\exists q', q'' \in Q$ s.t. $\delta(q, \#) = (q', R)$ and $\delta(q, \$) = (q'', L)$.

Power of 2DFAs

Lemma

The class of language recognized by 2DFAs is regular.

Proof.

Let $T_x : Q \times \{\bowtie\} \rightarrow Q \times \{\perp\}$, which is defined as follows:

$T_x(p) := q$ if whenever A enters x on p it leaves x on q.

$T_x(\bowtie) := q$ q is the state in which A emerges on x the first time.

$T_x(q) := \perp$ if A loops on x forever.

$\square$

Power of 2DFAs

Lemma

The class of language recognized by 2DFAs is regular.

Proof.

Let $T_x : Q \times \{\bowtie\} \to Q \times \{\perp\}$, which is defined as follows:

$T_x(p) := q$ if whenever A enters x on p it leaves x on q.

$T_x(\bowtie) := q$ q is the state in which A emerges on x the first time.

$T_x(q) := \perp$ if A loops on x forever.

Total number of functions of the type

$T_x \leq (|Q| + 1)^{(|Q|+1)}$

$T_x = T_y \Rightarrow \forall z(xz \in F \Leftrightarrow yz \in F)$ (Prove this.)

$T_x = T_y \Leftrightarrow x \equiv_A y$

Pushdown automata

NFA + Stack

$$L_{a,b} = \{a^n b^n \mid n \geq 0\}.$$

$$PAL = \{w \cdot w^R \mid w \in \Sigma^*\}.$$

Pushdown automata: formal definition

Definition

A non-deterministic pushdown automaton (NPDA)
$A = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$, where

Q:	set of states	Σ:	input alphabet
Γ:	stack alphabet	q_0:	start state
$\perp$:	start symbol	F:	set of final states

$\delta \subseteq Q \times \Sigma \times \Gamma \times Q \times \Gamma^*$.

Understanding δ

For $q \in Q, a \in \Sigma$ and $X \in \Gamma$, if $\delta(q, a, X) = (p, \gamma)$,
then p is the new state and γ replaces X in the stack.

if $\gamma = \epsilon$ then X is popped.
if $\gamma = X$ then X stays unchanges on the top of the stack.
if $\gamma = \gamma_1\gamma_2\dots\gamma_k$ then X is replaced by γ_k
and $\gamma_1\gamma_2\dots\gamma_{k-1}$ are pushed on top of that.

Configuration of an NPDA

Definition (Configurations)

A configuration of an NPDA $A = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$ is a three tuple (q, w, γ), where $q \in Q$, $w \in \Sigma^*$, and $\gamma \in \Gamma^*$.

if $(p, \gamma) \in \delta(q, a, X)$ then $\forall w \in \Sigma^*$ and $\gamma' \in \Gamma^*$,

$$(q, a \cdot w, X\gamma') \ \vdash \ (p, w, \gamma \cdot \gamma')$$

Let I, J are two configurations of A.

We say that $I \vdash^k J$ iff $\exists I'$ such that $I \vdash I'$ and $I' \vdash^{k-1} J$.

Language recognized by pushdown automata

Definition

We say that a word is accepted by an NPDA A if $(q_0, w, \perp) \vdash^* (q, \epsilon, \epsilon)$, where $q \in Q$. acceptance by an empty stack.

A language L is said to be recognized by an NPDA A if the set $\{w \mid w \text{ is accepted by } A\}$ is the same as L.

The class of languages recognized by NPDAs is called Context-free languages.

Another notion of acceptance of words:

We say that a word is accepted by an NPDA A if $(q_0, w, \perp) \vdash^* (q, \epsilon, \gamma)$, where $q \in F$. acceptance by a final state.

Context-free languages

Examples

$$\text{PAL} = \{w \cdot w^R \mid w \in \Sigma^*\}.$$

$$\text{Balanced} = \{w \in \{(,),[,]\} \mid w \text{ balanced string of paranthesis }\}.$$

$$L_{a/b/c} = \{a^i b^j c^k \mid i \neq j \text{ or } j \neq k\}.$$

$$L_{a/b/c} = \{a^i b^j c^k \mid i \neq j \text{ and } j \neq k\}. \ ?$$

Non-context-free languages

Lemma (Pumping lemma for CFLs)

Say L is a language over the alphabet Σ^*. If

- ☹ *for all* $n \in \mathbb{N}$,
- ☺ $\exists z \in \Sigma^*$, *such that*
- ☹ *for all possible ways of breaking* z *into* $z = u \cdot v \cdot w \cdot x \cdot y$, *s.t.* $|v \cdot w \cdot x| \leq n$ *and* $|v \cdot x| > 0$,
- ☺ $\exists i \in \mathbb{N}$ *s. t.* $u \cdot v^i \cdot w \cdot x^i \cdot y \notin L$,

then L is not a CFL.

Applications of the pumping lemma for CFLs

Let $L_{a,b,c} = \{a^n b^n c^n \mid n \geq 0\}$

- ☹ For any chosen n,
- ☺ let $z = a^n \cdot b^n \cdot c^n$
- ☹ For any split of z into u, v, w, x, y
- ☺ as $|v \cdot w \cdot x| \leq n$
  Either $v \cdot w \cdot x$ has no c's, or no a's.
  Therefore, $u \cdot v^0 \cdot w \cdot x^0 \cdot y \notin L$.

Say L is a language over the alphabet Σ^*. If

- ☹ for all $n \in \mathbb{N}$,
- ☺ $\exists z \in \Sigma^*$, such that
- ☹ for all possible ways of breaking z into $z = u \cdot v \cdot w \cdot x \cdot y$, s.t. $|v \cdot w \cdot x| \leq n$ and $|v \cdot x| > 0$,
- ☺ $\exists i \in \mathbb{N}$ s. t. $u \cdot v^i \cdot w \cdot x^i \cdot y \notin L$,

then L is not a CFL.

Applications of the pumping lemma for CFLs

Let $EQ = \{w \cdot w \mid w \in \{a,b\}^*\}$.

- ☹ For any chosen n,
- ☺ let $z = a^n \cdot b \cdot a^n \cdot b$
- ☹ For any split of z into u, v, w, x, y
- ☺ Note that $|v \cdot w \cdot x| \leq n$.
  (after some case analysis.)
  Therefore, $u \cdot v^0 \cdot w \cdot x^0 \cdot y \notin L$.

Say L is a language over the alphabet Σ^*. If

- ☹ for all $n \in \mathbb{N}$,
- ☺ $\exists z \in \Sigma^*$, such that
- ☹ for all possible ways of breaking z into $z = u \cdot v \cdot w \cdot x \cdot y$, s.t. $|v \cdot w \cdot x| \leq n$ and $|v \cdot x| > 0$,
- ☺ $\exists i \in \mathbb{N}$ s. t. $u \cdot v^i \cdot w \cdot x^i \cdot y \notin L$,

then L is not a CFL.

Context-free grammars

Inductive definition of PAL.

$\epsilon, 0, 1$ are in PAL.

If w is in PAL then $0 \cdot w \cdot 0 \in$ PAL.

If w is in PAL then $1 \cdot w \cdot 1 \in$ PAL.

Context-free grammar for PAL.

$S \to \epsilon$.

$S \to 0$.

$S \to 1$.

$S \to 0S0$.

$S \to 1S1$.

Context-free grammar

Definition

A context-free grammar (CFG) G is given by (V, T, P, S_0), where

- V is a set of variables,
- T is a set of terminal symbols or the alphabet,
- P is a set of productions, $P \subseteq V \times (V \cup T)^*$,
- $S_0 \in V$, a start symbol.

Example: Grammar for PAL.

$$S \to \epsilon.$$
$$S \to 0.$$
$$S \to 1.$$
$$S \to 0S0.$$
$$S \to 1S1.$$

$G_{pal} = (V, T, P, S_0)$ such that

$$V = \{S\},$$
$$T = \{0, 1\},$$
$$P = \{S \to \epsilon, S \to 0, S \to 1, S \to 0S0, S \to 1S1\},$$
$$S_0 = S.$$

Drivations of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0).

Let $w, w' \in (V \cup T)^*$,

let $A \in V$ and let $(A \to v) \in P$ be a production in the grammar, where $v \in (V \cup T)^*$.

Then we say that $w \cdot A \cdot w'$ **derives** $w \cdot v \cdot w'$ **in one step**.

We denote it as follows: $w \cdot A \cdot w' \Rightarrow w \cdot v \cdot w'$.

Definition ($\Rightarrow^*$)

Let G be a CFG given by (V, T, P, S_0).

For all $\alpha \in (V \cup T)^*$, we say that $\alpha \Rightarrow^* \alpha$.

For all $\alpha, \beta, \gamma \in (V \cup T)^*$,

if $\alpha \Rightarrow^* \beta$ and $\beta \Rightarrow \gamma$ then $\alpha \Rightarrow^* \gamma$.

Language of a CFG

Definition

Let G be a CFG given by (V, T, P, S_0). The **language of** G, $L(G)$, is the set of all the strings over T which can be derived from S_0, i.e.

$$L(G) = \{w \in T^* \mid S \Rightarrow^* w\}.$$

Lemma

$L(G_{pal})$ is equal to PAL.
$\forall w \in \{0,1\},^* \ w \in PAL$ if and only if $w = w^R$.

Proof.

By Induction on $|w|$. DIY! □

Equivalence of CFGs and PDAs