COL352 Lecture 17

Contents

1 Recap

Discussion about grammars, parse trees, yields, and context free languages.

2 Definitions

Definition 1

A (non-deterministic) pushdown automaton ((N)PDA) is a 6-tuple $(Q, \Sigma, \Gamma, \Delta, q_0, A)$ where

1. Q – finite nonempty set of states
2. Σ – finite nonempty input alphabet
3. Γ – finite stack alphabet
4. $q_0 \in Q$ – initial state
5. A – set of accepting states
6. $\Delta \subseteq Q \times \Sigma_\epsilon \times \Gamma_\epsilon \times Q \times \Gamma_\epsilon$, where X_ϵ is defined as $X \cup \{\epsilon\}$

Note that in an NFA, $\Delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times Q$.
Informally, a string is accepted if $\exists$ a run which reads the entire string and does what ?????

Definition 2

Let $P = (Q, \Sigma, \Gamma, \Delta, q_0, A)$ be a PDA. An instantaneous description (i.d.) of P is a tuple (q, x, α) where $q \in Q$, $x \in \Sigma^*$, $\alpha \in \Gamma^*$. (The set of instantaneous descriptions is $Q \times \Sigma^* \times \Gamma^*$).

Informally, it consists of the current state, the string left to be read, and the description of the current stack.

Definition 3

Let $P = (Q, \Sigma, \Gamma, \Delta, q_0, A)$ be a PDA. The relation $\vdash_P$ (read as "changes to") is defined on the set of i.d.s as follows:

If $(q, a, B, q', B') \in \Delta$, then $(q, ax, B\alpha) \vdash_P (q', x, B'\alpha)$, and no other pairs of i.d.s are related.

In other words:

$$(q, y, \beta) \vdash_P (q', y', \beta') \iff$$
$$\exists a \in \Sigma_\epsilon, B \in \Gamma_\epsilon, \alpha \in \Gamma^*, B' \in \Gamma_\epsilon \text{ such that } y = ay', \beta = B\alpha, \beta' = B'\alpha, (q, a, B, q', B') \in \Delta$$

Definition 4

$\vdash_P^*$ is defined as the reflexive transitive closure of $\vdash$ (read as "changes to in finitely many steps").

Definition 5

$x \in \Sigma^*$ is said to be accepted by PDA $P = (Q, \Sigma^*, \Gamma, \Delta, q_0, A)$ iff

$$(q_0, x, \epsilon) \vdash_P^* (q, \epsilon, \alpha)$$

for some $q \in A$ and some $\alpha \in \Gamma^*$.

New

Definition 6

The language recognized by PDA P denoted by $\mathcal{L}(P)$ is $\{x \in \Sigma^* \mid P \text{ accepts } x\}$.

3 Content

Plan for the next 4 lectures is to show that L is generated by the grammar iff L is recognized by a PDA.

The forward implication will be done in the next two lectures.
Recall that $\Delta \subseteq Q \times \Sigma_\epsilon \times \Gamma_\epsilon \times Q \times \Gamma_\epsilon$, $(q, a, B, q', B') \in \Delta$. q stands for the current state, a for current input, B for the current stack top to be popped, q' for the next state, and B' for the element to be pushed onto stack.

Suppose that instead, we do $\Delta \in Q \times \Sigma_\epsilon \times \Gamma_\epsilon \times Q \times \Gamma^*$ (i.e., we can push strings onto the stack). Call such a PDA a fast PDA.

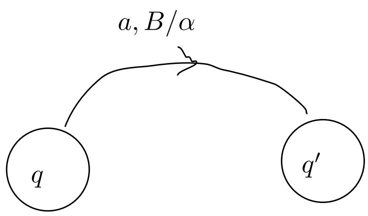
Theorem 1

L is recognized by a PDA $\iff$ L is recognized by a fast PDA.

Proof. The forward direction is obvious.

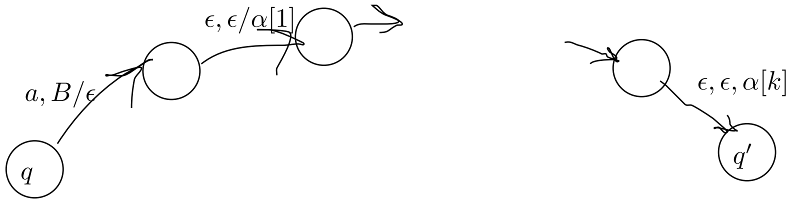
Now we look at a sketch of the proof of the non-obvious reverse direction.

L is recognized by a fast PDA, say P. A state transition looks like the following:



$a \in \Sigma_\epsilon, \alpha \in \Gamma^*$
$B \in \Gamma_\epsilon$

We do the following:



Check that P accepts x iff P' accepts x. Now since Δ was finite, Δ' is finite too. $\square$

Corollary 1

In order to show that L generated by grammar $\implies$ L recognized by a PDA, it is sufficient to show that L generated by grammar $\implies$ L recognized by a fast PDA.

Theorem 2

Every language which is generated by a grammar is also recognized by a fast PDA.

Proof. We begin by an example.

Example 1

Consider $N = \{A, B\}$, $\Sigma = \{0, 1\}$, $R = \{A \to 0BB, A \to 1, B \to A1A, B \to 0\}$, with initial nonterminal as A.

We construct the PDA as follows.

$Q = \{q_i, q, q_a\}, A = \{q_a\}$, initial state $= q_i$, $\Sigma = \{0, 1\}, \Gamma = N \cup \Gamma \cup \{\perp\}$.

$\Delta = \Delta_{special} \cup \Delta_{match} \cup \Delta_{produce}$.

$\Delta_{match} = \{(q, 0, 0, q, \epsilon), (q, 1, 1, q, \epsilon)\}$. This corresponds to removing a from the stack, where $a \in \Sigma$.

$\Delta_{produce} = \{(q, \epsilon, A, q, BB0), (q, \epsilon, A, q, 1), (q, \epsilon, B, q, A1A), (q, \epsilon, B, q, 0)\}$. This corresponds to the production rules.

$\Delta_{special} = \{(q_i, \epsilon, \epsilon, q, \perp, \perp A), (q, \epsilon, \perp, q_a, \epsilon)\}$. This corresponds to initialization and cleanup.

Consider the following leftmost derivation of 01110: $A \implies 0BB \implies 0A1AB \implies 011AB \implies 0111B \implies 01110$.

Then we have

$$\begin{aligned}
(q_i, 01110, \epsilon) &\vdash (q, 01110, A\perp) \\
&\vdash (q, 01110, A\perp) \\
&\vdash (q, 01110, 0BB\perp) \\
&\vdash (q, 1110, BB\perp) \\
&\vdash (q, 1110, A1AB\perp) \\
&\vdash (q, 1110, 11AB\perp) \\
&\vdash (q, 110, 1AB\perp) \\
&\vdash (q, 10, AB\perp) \\
&\vdash (q, 10, 1B\perp) \\
&\vdash (q, 0, B\perp) \\
&\vdash (q, 0, 0\perp) \\
&\vdash (q, \epsilon, \perp) \\
&\vdash (q_a, \epsilon, \epsilon)
\end{aligned}$$

□

Corollary 2

Every language which is generated by a grammar is also recognized by a PDA.