COL352 Lecture 21

Contents

1 Recap

Grammar $\implies$ PDA completed.

More precisely, we showed in the last to last lecture that L is recognized by a fast PDA $\iff$ L is recognized by a PDA, and in the last lecture that L is recognized by a fast PDA if L is generated by a grammar.

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, $\Delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times Q$.

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 $Q \times \Sigma^* \times \Gamma^*$).

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 $\alpha \in \Gamma^*$.

Definition 6

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

Definition 7

We define a simple PDA P to be a PDA such that

1. $\Delta = \Delta_{push} \uplus \Delta_{pop}$, where
   (a) Δ_{push} contains transitions (q, a, ϵ, q', B) where $q, q' \in Q, a \in \Sigma_\epsilon, B \in \Gamma$ (i.e., not allowed to pop, must push), and
   (b) Δ_{pop} contains transitions (q, a, B, q', ϵ) where $q, q' \in Q, a \in \Sigma_\epsilon, B \in \Gamma$ (i.e., must pop, not allowed to push).
2. $|A| = 1$, i.e., unique accepting state.
3. If x is accepted, then x is accepted with an empty stack, i.e., $(q_{init}, x, \epsilon) \vdash^* (q_{acc}, \epsilon, \alpha)$ for some $\alpha \in \Sigma^*$ iff $q_{init}, x, \epsilon) \vdash^* (q_{acc}, \epsilon, \epsilon)$.

3 Content

Now we need the following: If L is recognized by a PDA, L is generated by a grammar.

We shall define something called a simple PDA, and show that it is as powerful as a PDA.

Recall that PDA $P = (Q, \Sigma, \Gamma, \Delta, q_{init}, A)$ where $q_{init} \in Q, A \subseteq Q$, $\Delta \subseteq Q \times \Sigma_\epsilon \times \Gamma_\epsilon \times Q \times \Gamma_\epsilon$.
A simple PDA is defined as in the previous section.

Claim 0.1

L is recognized by a PDA iff L is recognized by a simple PDA.

Proof. $\Longleftarrow$ is trivial. We'll look at the other direction.

Suppose L is recognized by PDA $P = (Q, \Sigma, \Gamma, \Delta, q_{init}, A)$.

To ensure condition 2, add a new state q_{acc} to Q, make it the unique accepting state, and add transitions $(q, \epsilon, \epsilon, q_{acc}, \epsilon)$ for each $q \in A$.

To ensure condition 3, add transitions $(q_{acc}, \epsilon, B, q_{acc}, \epsilon)$ $\forall B \in \Gamma$.

To ensure condition 1, we need to break each transition that pushes as well as pops into two, i.e., replace the transition (q, a, B, q', C) where $a \in \Sigma_\epsilon, B, C \in \Gamma$ by (q, a, B, q'', ϵ) and $(q'', \epsilon, C, q', \epsilon)$, and replace $(q, a, \epsilon, q', \epsilon)$ with $(q, a, \epsilon, q'', \$)$ and $(q'', \epsilon, \$, q', \epsilon)$.

$P' = (Q \uplus \{q_{acc}\} \uplus \text{intermediate states}, \Sigma, \Gamma \uplus \{\$\}, q_{init}, \{q_{acc}\})$. □

Now our goal shall be the following.

Question 1

Given simple PDA $P = (Q, \Sigma, \Gamma, \Delta, q_{init}, \{q_{acc}\})$, $\Delta = \Delta_{push} \uplus \Delta_{pop}$, construct a grammar $G = (N, \Sigma, R, S)$ such that

$$\forall x \in \Sigma^* : ((q_{init}, x, \epsilon) \vdash^* (q_{acc}, \epsilon, \epsilon) \iff S \xrightarrow{*} x)$$

Note 1

Idea: $N = \{V_{qq'} \mid (q,q') \in Q \times Q\}$. R should ensure that $V_{qq'} \overset{*}{\Longrightarrow} x \iff (q,x,\epsilon) \vdash^* (q',\epsilon,\epsilon)$ (i.e., $\exists$ a run of P on x starting from q with empty stack and ending in q' with empty stack).

We also set S to $v_{q_{init}q_{acc}}$.

The derivation should be something like $(q,x,\epsilon) \vdash \_ \vdash \_ \cdots \vdash \_ \vdash (q',\epsilon,\epsilon)$.

Is there an intermediate instantaneous description in which the stack is empty? If there is, then in the first part of the run, we have read off some prefix of x, and in the second part of the run, we have read off the remaining suffix of x, say $x = x_1x_2$. We hope that $V_{qr} \overset{*}{\Longrightarrow} x_1$ and $V_{rq'} \overset{*}{\Longrightarrow} x_2$.

We want $V_{qq'} \overset{*}{\Longrightarrow} x_1x_2$, so we add the rule $V_{qq'} \to V_{qr}V_{rq'}$ to R, whihc will give us $V_{qq'} \Longrightarrow V_{qr}V_{rq'} \overset{*}{\Longrightarrow} x_1x_2 = x$.

Now suppose the answer is no. Then the first transition must be a push transition and the last must be a pop transition, i.e., we go from (q,x,ϵ) to $(r,\_,B)$ where $(q,a,\epsilon,r,B) \in \Delta$ where $a = \epsilon$ or $x[1]$, and from $(r',\_,B')$ to (q',ϵ,ϵ) where (r',b,B',q,ϵ) where $b = \epsilon$ or $x[n]$. Note that the stack has never been empty, so $B' = B$ (look at it from the perspective of the evolution of the stack over time).

So since $(q,x_1x_2,\epsilon) \vdash^* (r,x_1,\epsilon)$, and x_1 is not touched, we should have $(q,x_1,\epsilon) \vdash^* (r,\epsilon,\epsilon)$ maybe. In better terms, we have $(q,x,\epsilon) \vdash (r,x'b,B) \vdash \cdots \vdash (r',b,B) \vdash (q',\epsilon,\epsilon)$. B is always in the stack for the intermediate places.

So we'll add $V_{qq'} \to aV_{rr'}b$ for $q,q',r,r' \in Q, a,b \in \Sigma_\epsilon$ if $\exists B \in \Gamma$ suh that $(q,a,\epsilon,r,B) \in \Delta_{push}$ and $(r',b,B,q',\epsilon) \in \Delta_{pop}$. Runs with 0 transitions: consume ϵ, must start and end in the same state. So we'll add $V_{qq} \to \epsilon \forall q \in Q$.

Theorem 1

Let $P = (Q,\Sigma,\Gamma,\Delta_{push} \uplus \Delta_{pop}, q_{init}, \{q_{acc}\})$ be a simple PDA. Let $G = (N,\Sigma,R_1 \cup R_2 \cup R_3, V_{q_{init}q_{pop}})$ be the grammar where

1. $N = \{V_{qq'} \mid (q,q') \in Q \times Q\}$
2. $R_1 = \{V_{qq} \to \epsilon \mid q \in Q\}$, $R_2 = \{V_{qq'} \to V_{qr}V_{rq'} \mid (q,q',r) \in Q \times Q \times Q\}$, and $R_3 = \{V_{qq'} \to aV_{rr'}b \mid (q,q',r,r') \in Q \times Q \times Q \times Q, a,b \in \Sigma_\epsilon, \exists B \in \Gamma : (q,a,\epsilon,r,B) \in \Delta_{push}, (r',b,B,q',\epsilon) \in \Delta_{pop}\}$

Then $\forall q,q' \in Q, x \in \Sigma^*$, $V_{qq'} \overset{*}{\Longrightarrow} x$ iff $(q,x,\epsilon) \vdash^* (q',\epsilon,\epsilon)$.

Proof.

Claim 1.1

P,G as before, then $\forall q,q' \in Q, x \in \Sigma^*$, if $(q,x,\epsilon) \vdash^* (q',\epsilon,\epsilon)$, then $V_{qq'} \overset{*}{\Longrightarrow} x$.

Proof. By induction on the number of transitions in a shortest run of P from (q,x,ϵ) to (q',ϵ,ϵ).

If the number of transitions is 0, then $x = \epsilon$, so $q = q'$ and since $V_{qq} \to \epsilon \in R$, we have $V_{qq'} \overset{*}{\Longrightarrow} x$.

If the number of transitions is more than 0.

1. $\exists$ an intermediate I.D. in the run having an empty stack. Let r be the state in that I.D., suppose the run consumes x_1 before r and x_2 after r. Then $x = x_1x_2, U = (r,x_2,\epsilon)$. Now break the run into two and show that it devolves into case 1 in the note (exercise).
2. No intermediate I.D. has an empty stack, so $\exists B \in \Gamma, a,b \in \Gamma_\epsilon$ and states r,r' such that $x = ax'b, (q,ax'b,\epsilon) \vdash (r,x;b,B) \vdash^* (r',b,B) \vdash (q',\epsilon,\epsilon)$, and the run from $(r,x'b,B)$ to (r',b,B) doesn't pop the bottom-most B.

   Since $(q,ax'b,\epsilon) \vdash (r,x'b,B), (q,a,\epsilon,r,B) \in Delta_{push}$.

   Since $(r',b,B) \vdash (q',\epsilon,\epsilon), (r',b,B,q',\epsilon) \in \Delta_{pop}$.

   Both of these together imply that $V_{qq} \to aV_{rr'}b \in R_3$.

   Moreover, $(r,x',\epsilon) \vdash^* (r',\epsilon,\epsilon)$. By induction hypothesis, we have $V_{rr'} \overset{*}{\Longrightarrow} x'$ since this run has 2 less transitions than the original run. So we have $V_{qq'} \implies aV_{rr'}b \overset{*}{\implies} x$.

$\square$

Claim 1.2

P, G as before, then $\forall q, q' \in Q, x \in \Sigma^*$, if $V_{qq'} \overset{*}{\implies} x$, then $(q, x, \epsilon) \vdash^* (q', \epsilon, \epsilon)$.

Proof.

Continued from here

By induction on the number of productions in a shortest derivation of x from$V_{qq'}$.

Base case: # productions = 1, so $x = \epsilon, q = q', (q, x, \epsilon) \vdash^* (q', \epsilon, \epsilon)$ is trivial because $\vdash^*$ is reflexive.

Inductive case: # productions > 1.

1. Case 1: 1st production applied is $V_{qq} \to V_{qr}V_{rq'}$ for some $r \in Q$. So $V_{qq'} \implies V_{qr}V_{rq'} \overset{*}{\implies} x$, so $\exists x_1, x_2$ such that $V_{qr} \overset{*}{\implies} x_2, V_{rq'} \overset{*}{\implies} x_2$ and $x = x_1x_2$. By the inductive hypothesis, we have $(q, x_1, \epsilon) \vdash^* (r, \epsilon, \epsilon)$ and $(r, x_2, \epsilon) \vdash^* (q', \epsilon, \epsilon)$. The first implies that $(q, x_1x_2, \epsilon) \vdash^* (r, x_2, \epsilon)$, and by transition closure property, we have $(q, x_1x_2, \epsilon) \vdash^* (q', \epsilon, \epsilon)$.
2. Case 2: 1st production rule applied is $V_{qq'} \to aV_{rr'}b$ for some $r, r' \in Q$, $a, b \in \Sigma_\epsilon$. So $\exists x' : x = ax'b$ and $V_{rr'} \overset{*}{\implies} x'$. By induction hypothesis, since number of productions is less than the current number of productions, we have $(r, x', \epsilon) \vdash^* (r, \epsilon, \epsilon)$. So we have $(r, x'b, \epsilon) \vdash^* (r', b, \epsilon)$. Since this relation is in R_3, we have that $\exists B \in \Gamma$ such that $(q, a, \epsilon, r, B) \in \Delta_{push}) \in \Delta_{push}$ and $(r', b, B, q', \epsilon) \in \Delta_{pop}$. So we have $(q, x, \epsilon) = (q, ax'b, \epsilon) \vdash (r, x'b, B) \vdash^* (r', b, B) \vdash (q', \epsilon, \epsilon)$, so $(q, x, \epsilon) \vdash^* (q, \epsilon, \epsilon)$.

$\square$

$\square$

Corollary 1

$\mathcal{L}(G) = \mathcal{L}(P)$

Proof. $V_{q_{init}q_{acc}} \overset{*}{\implies} x$ iff $(q_{init}, x, \epsilon) \vdash^* (q_{acc}, \epsilon, \epsilon)$. $\square$

Question 2

How is the class of context free languages related to the class of regular languages?

Answer. The class of regular languages is 2^{Σ^*}. Regular languages $\subset$ context free languages since every DFA can be simulated by some PDA (without using stack).

Exercise: Given a DFA D, describe an easy construction of a grammar G such that $\mathcal{L}(D) = \mathcal{L}(G)$.

Question 3

What kinds of closure properties can we think of for context free languages? Union? Intersection? Complementation? Concatenation? $*$?

Answer. Let $G = (N, \Sigma, R, S)$.
For $*$: $G' = (N \uplus \{T\}, \Sigma, R \cup \{T \to TT, T \to S, T \to \epsilon\})$ generates L^*.
Let $G = (N, \Sigma, R, S)$.
For $\cup$: $G_1 = (N_1, \Sigma, R_1, S_1)$ generates L_1 and $G_2 = (N_2, \Sigma, R_2, S_2)$ generates L_2, then $(N_1 \uplus N_2 \uplus T, \Sigma, R_1 \cup R_2 \cup \{T \to S_1, T \to S_2\}, T)$ generates $L_1 \cup L_2$.

$(N_1 \uplus N_2 \uplus T, \Sigma, R_1 \cup R_2 \cup \{T \to S_1S_2\}, T)$ generates L_1L_2.

Not closed under intersection (would imply complementation by contradiction and contrapositive).

Let $L_1 = \{a^n b^n c^* \mid n \in \mathbb{N} \cup \{0\}\}$, and $L_2 = \{a^* b^n c^n \mid n \in \mathbb{N} \cup \{0\}\}$. Then $L_1 \cap L_2 = \{a^n b^n c^n \mid n \in \mathbb{N} \cup \{0\}\}$, which is probably not context free. We'll use a version of the pumping lemma for the DFA.

Note 2

Suppose L is a context free language generated by a grammar G. Suppose $w \in L$ is a "long enough" string. Consider a smallest parse tree T of w. Since w is "long enough", T is "tall enough". Look at the longest root-to-leaf path in T, say p.

Since p is long enough, some two nodes on P are labelled by the same non-terminal, say $A \in N$. Let $uw'z$ be w such that the upper A derives w', and let vxy be w'. Look at tree (very helpful). Break it into S, A's tree (having u, z), A, A's tree (having v, y) and A's tree (having x).

By copy pasting the second tree into itself again and again, we can get $\forall i : uv^i xy^i z \in L$. This doesn't give us anything if $y = v = \epsilon$, so we enforce smallest tree constraints.