CS310 : Automata Theory 2019

Lecture 18: Deterministic PDA

Instructor: Ashutosh Gupta

IITB, India

Compile date: 2019-02-12

Topic 18.1

PDA to CFG

PDA to CFG

Theorem 18.1

Let $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, \{\})$ be a PDA. Then, there is a CFG grammar $G = (N, \Sigma, P, S)$ such that $L^\epsilon(A) = L(G)$.

Proof.

We construct the grammar $G = (N, \Sigma, P, S)$ as follows.

Let $N = \{[qBq'] \mid B \in \Gamma \text{ and } q, q' \in Q\} \cup \{S\}$.

For each $(p, Y_1 ... Y_k) \in \delta(q, a, X)$ and $r_1, ..., r_k \in Q$, we add the following rule

$$[qXr_k] \to a[pY_1r_1] \dots [r_{k-1}Y_kr_k] \in P$$

We also need initialization rules, for each $q \in Q$

$$S \to [q_0 Z_0 q] \in P.$$

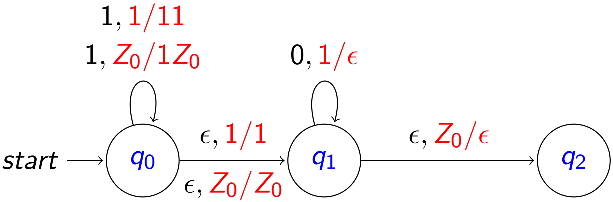
Exercise 18.1

a. How many nonterminals? *b. How many transitions?* ...

Example: PDA to CFG

Example 18.1

Consider PDA $A = (\{q_0, q_1, q_2\}, \{0, 1\}, \{Z_0, 1\}, \delta, q_0, Z_0, \{\})$



We construct CFG, (N, Σ, P, S) *as follows*

$$N = \{S, [q_0 Z_0 q_0], [q_0 Z_0 q_1], [q_0 Z_0 q_2], ..., [q_2 1 q_0], [q_2 1 q_1], [q_2 1 q_2]\}$$

Exercise 18.2

What is the size of N*?*

Example : PDA to CFG (contd.)

Due to transition $(q_0, 1Z_0) \in \delta(q_0, 1, Z_0)$, P contains

- $[q_0 Z_0 q_1] \rightarrow 1[q_0 1 q_2][q_2 Z_0 q_1]$
- $[q_0 Z_0 q_0] \rightarrow 1[q_0 1 q_0][q_0 Z_0 q_0]$
- ... How many more?

Due to transition $(q_1, \epsilon) \in \delta(q_1, 0, 1)$, P contains

- $[q_1 1 q_1] \rightarrow 0$
- ... How many more?

P also contains the following start transitions

- $S \rightarrow [q_0 Z_0 q_2]$
- ... How many more?

Exercise 18.3

Give rules in P due to transition $(q_1, 1) \in \delta(q_0, \epsilon, 1)$

PDA to CFG II

Proof(contd.).

We only(why?) need to prove

$$[qXp] \overset{*}{\Rightarrow} w \text{ iff } (q, w, X) \vdash^* (p, \epsilon, \epsilon).$$

($\Leftarrow$)

Let us suppose $(q, w, X) \vdash^* (p, \epsilon, \epsilon)$.

We show $[qXp] \overset{*}{\Rightarrow} w$ by induction on the number of moves of A.

base case:

Single step. Then $(p, \epsilon) \in \delta(q, w, X)$, where $w \in \Sigma \cup \{\epsilon\}$.

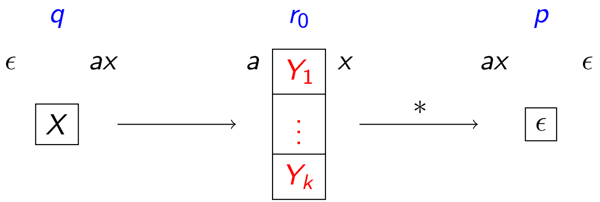
By the construction of G, $[qXp] \rightarrow w \in P$. Therefore, $[qXp] \overset{*}{\Rightarrow} w$. ...

PDA to CFG III

Proof(contd.).

induction step:

Let $w = ax$, where $a \in \Sigma \cup \{\epsilon\}$. Suppose,

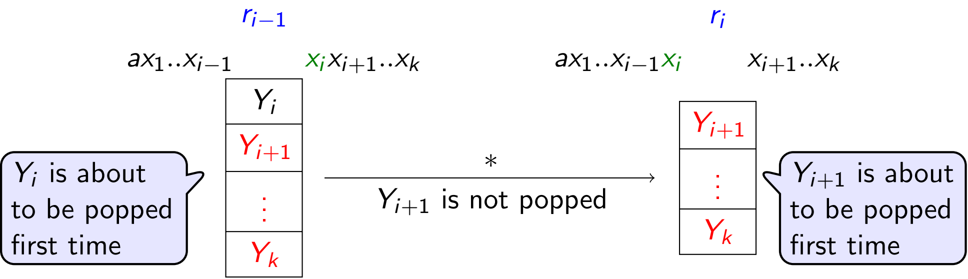


Y_i has to be eventually popped.(why?) ...

PDA to CFG IV

Proof(contd.).

Let $x = ax_1x_2...x_k$ and $r_k = p$ such that



Therefore, we have $(r_{i-1}, x_i, Y_i) \vdash^* (r_i, \epsilon, \epsilon)$ in less than n steps.(why?)

Due to induction hypothesis, $[r_{i-1} Y_i r_i] \overset{*}{\Rightarrow} x_i$.

...

PDA to CFG V

Proof(contd.).

Due to the properties of derivations

$$ax_1..x_{i-1}[r_{i-1}Y_ir_i][r_iY_{i+1}r_{i+1}]..[r_{k-1}Y_kr_k] \overset{*}{\Rightarrow} ax_1..x_i[r_iY_{i+1}r_{i+1}]..[r_{k-1}Y_kr_k]$$

Due to the construction of the grammar, we must have

$$[qXp] \Rightarrow a[r_0Y_1r_1]..[r_{k-1}Y_kr_k].$$

By stitching the above derivations, we obtain

$$[qXp] \overset{*}{\Rightarrow} w$$

($\Rightarrow$)
Since the above proof is fully reversible, we only need to play it backwards. □

Topic 18.2

Deterministic PDA

Deterministic PDA (DPDA)

Definition 18.1

A PDA $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ is deterministic *if for each $q \in Q$ and $X \in \Gamma$*

- $|\delta(q, a, X)| \leq 1$ *for each* $a \in \Sigma \cup \{\epsilon\}$
- *if* $|\delta(q, a, X)| = 1$ *for some* $a \in \Sigma$*, then* $|\delta(q, \epsilon, X)| = 0$

If there is spontaneous move, there is no other move

Commentary:

Power of DPDA

DPDA are distinctly less powerful than PDAs.

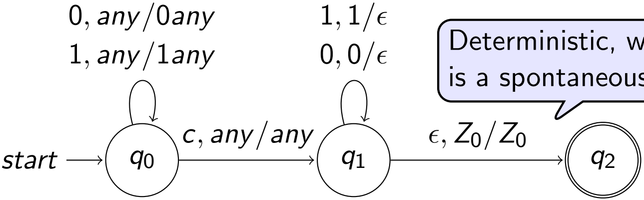
Formal proof is beyond the scope of this course!

Example 18.2

The following language can not be recognized by DPDA.

$$L_{wwr} = \{ww^R | w \in \{0,1\}^*\}$$

However $\{wcw^R | w \in \{0,1\}^*\}$ *can be recognized by the following DPDA.*



Deterministic, while there is a spontaneous transition

DPDA and regular languages

DPDA recognized some languages that are not regular (the previous example).

Exercise 18.4

Show that if L is a regular language, then $L = L(P)$ for some DPDA P.

Exercise 18.5

Give a (regular) language L such that no DPDA recognizes it by empty stack?

Prefix property

Definition 18.2

A language L has prefix property if there are no two $x, y \in L$ such that x is a prefix of y.

Example 18.3

- $\{wcw^R | w \in \{0,1\}^*\}$ *has prefix property*
- 0^* *does not have prefix property*

Exercise 18.6

Do the following languages have prefix property?

- $\{101, 1001\}$
- $\{101, 1011\}$
- $\{ww^R | w \in \{0,1\}^*\}$
- $\epsilon \in L$ *and* $|L| > 1$

Prefix property and empty stack recognition

Theorem 18.2

A language L is $L^{\epsilon}(P)$ for some DPDA P iff L has the prefix property and $L = L(P')$ for some DPDA P'.

Proof.

$(\Rightarrow)$

Let distinct $x, y \in L$ such that x is prefix of y.

P gets stuck after x. no way to accept y.Contradiction.

The usual construction for "by final state" automaton preserves determinism.

$(\Leftarrow)$

There must be no useful outgoing transitions from accepting states.

Therefore, we can delete them if there are any.

Now we can use the usual construction to obtain "by empty stack" DPDA P from P'. □

Unambiguous grammar and DPDA

We can prove that the languages recognized by DPDA are not inherently ambiguous.

However, the reverse does not hold.

Example 18.4

The following is an unambiguous grammar for L_{wwr}, which cannot be recognized by a DPDA.

$$S \rightarrow 0S0 \mid 1S1 \mid \epsilon$$

Unambious grammars for DPDA

Theorem 18.3

If language $L = L^{\epsilon}(P)$ for some DPDA P, then L has an unambiguous grammar.

Proof.

Due to theorem 18.1, we construct a grammar that G such that $L(G) = L$.

claim: G is unambiguous
Consider $w \in L$.
There is exactly one run of P on w.
Therefore, at each step at least one of the production rules can be applied on the leftmost symbol.(why?)
At some step, let it be a rule due to the transition $\delta(q, a, X) = \{(r, Y_1 ... Y_k)\}$.
There may be many production rules generated due the above transition.
Since there is unique run, therefore Y_i must be popped at unique stater_i.
Therefore, exactly one of the production rule will lead to acceptance. □

Unambious grammars for DPDA

Theorem 18.4
If language $L = L(P)$ for some DPDA P, then L has an unambiguous grammar.

Proof.
Let \$ be a fresh symbol with respect to L.
Consider language $L\$$.
$L\$$ has prefix property.(why?)
$L\$ = L(P')$ for some DPDA P'.(why?)
Due to thereom 18.2, there exists DPFA P'' such that $L\$ = L^{\epsilon}(P'')$.
Due to theorem 18.3, there is an unambiguous grammar G' such that $L(G') = L\$$.

We obtain G be turning \$ into nonterminal and adding a production rule $\$ \rightarrow \epsilon$ in G'. Clearly, $L(G) = L$. ...

Unambious grammars for DPDA

Proof.

Any leftmost derivation in G, $\$ \rightarrow \epsilon$ would be applied at the end.

All other derivations in the run are same as G'.
Therefore, there can be no choices in the middle of the derivation of G.
There is excactly one derivation for \$.

G is unambiguous. $\square$

End of Lecture 18