CS 205
Lecture 23
9/3/21

Deterministic Pushdown Automata (DPDA)

Def : DPDA $= (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

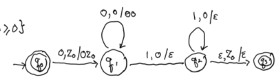
s.t.

(1) $\forall q \in Q.\ \forall X \in \Gamma.$ if $\delta(q, \varepsilon, X) \neq \emptyset$ then
$\forall a \in \Sigma .\ \delta(q, a, X) = \emptyset.$

(2) $\forall q \in Q.\ \forall X \in \Gamma .\ \forall a \in \Sigma \cup \{\varepsilon\}.$
$|\delta(q, a, X)| \leq 1.$

Ex $L = \{0^n 1^n \mid n \geq 0\}$

DPDA P



Thm If L is regular then there is a DPDA P s.t. $L = L(P)$.

Fact : DPDA accept a richer class of languages than regular languages.
e.g. $L = \{0^n 1^n \mid n \geq 0\}$

Def : L has the prefix property if there are no two distinct strings x, y in L s.t. x is a prefix of y.

Prop : If P is a DPDA and $L = N(P)$ then L has the prefix property.

Proof : If $x \in N(P)$ then $\langle q_0, x, Z_0 \rangle \vdash^* \langle p, \varepsilon, \varepsilon \rangle$
Therefore, P does not accept an input of the form xu, since P necessarily empties its stack after reading x and is stuck.

Thm : $L = N(P)$ for some DPDA P iff L has the prefix property and $L = L(P')$ for some DPDA P'.

Thm : $L = N(P)$ for some DPDA P then L has an unambiguous CFG G.

Thm : If $L = L(P)$ for some DPDA P then L has an unambiguous CFG G.

Thm : There are CFLs that are not accepted by any DPDA.
Ex : $L = \{ w w^R \mid w \in \{0,1\}^* \}$

Normal Forms for CFG

(1) Chomsky Normal Form (CNF) : Productions are of the form $A \to BC$ or $A \to a$.

(2) Griebach Normal Form (GNF) : $A \to a\alpha \quad \alpha \in V^*$

Eliminating ε- productions

Given CFG G, produce CFG G' s.t. G' has no rule of the form $A \to \varepsilon$ and $L(G) = L(G')$.

Assume : $\varepsilon \notin L(G)$.

Def : A variable A of CFG G is nullable if $A \overset{*}{\Rightarrow} \varepsilon$.

Inductive definition :
1. If $A \to \varepsilon$ is a prodn. then A is nullable
2. If $A \to B_1 B_2 B_3 \dots B_k$ is a prodn. and each B_i is nullable then A is nullable.

Algo for eliminating ε- productions

For each prodn $A \to X_1 \dots X_k$ create a prodn.
$A \to \alpha_1 \dots \alpha_k$ where

$$\alpha_i = \begin{cases} X_i & \text{if } X_i \text{ is not nullable} \\ X_i \text{ or } \varepsilon & \text{if } X_i \text{ is nullable} \end{cases}$$

and not all the X_i are ε.

By this algorithm we will remove all rules of the form $A \to \varepsilon$.

Ex $S \to AB$, $A \to aAA \mid \varepsilon$, $B \to bBB \mid \varepsilon$

Nullable variables : A, B, S

New rules
$S \to AB \mid A \mid B$
$A \to aAA \mid aA \mid a$
$B \to bBB \mid bB \mid b$

Eliminating Unit Productions

Unit prodn ; $A \to B$ (variables)

Ex $E \to T \mid E+T$
$T \to F \mid T*F$
...

Algo. for eliminating unit prodns.

1. Determine pairs $\langle A, B \rangle$ (Unit pairs) s.t. $A \overset{*}{\underset{u}{\Rightarrow}} B$ → using only unit productions

   Note : It is possible to have $A \overset{*}{\Rightarrow} B$ w/o using unit productions.
   e.g. $A \to BC$ and $C \to \varepsilon$.

2. If $\langle A, B \rangle$ is a unit pair then add the prodns.
   $A \to \beta_1 \mid \beta_2 \mid \dots \mid \beta_k$
   where $B \to \beta_1 \mid \beta_2 \mid \dots \mid \beta_k$ are all the non-unit productions of B.

3. Remove all unit productions.

How to determine the unit pairs?

Induction
- Base Case : $\langle A, A \rangle$ is a unit pair for any A.
- Ind. Step : If $\langle A, B \rangle$ is a unit pair and $B \to C$ is a prodn. then $\langle A, C \rangle$ is a unit pair.