Equivalence of CFG and PDA

CFG to PDA Conversion

For every CFG G, there exists a PDA M accepting L(G) and for every PDA M, there exists a CFG generating L(M)

CFG to PDA Conversion

Let L be a CFL, There exists a CFG G=(V,T,P,S) generating L

Assume that G is in GNF Form

Thus, G does not contain Epsilon

$A \to a\alpha$

alumni

SB

GNF

free of Λ

V^*

① For each production A->aα of G

Add δ(q,a,A)=(q, α)

$\delta(q, a, A) = (q, \alpha)$

Input

top

② For each production A->a of G

Add δ(q,a,A)=(q, ε)

CFG to PDA Conversion-Eg 1

Construct a PDA equivalent to the following grammar:

S->aAA

A->aS|bS|a

① Is it in GNF? Yes

[Single state PDA]

② for $S \rightarrow aAA$

Transition Rule ① $\delta(q, a, S) = (q, AA)$

for $A \rightarrow aS | bS | a$, $\alpha \Rightarrow V^* = \Lambda$

② $\delta(q, a, A) = (q, S)$ ✓

$\Lambda = \epsilon$

③ $\delta(q, b, A) = (q, S)$

$A \rightarrow a\alpha$

④ $\delta(q, a, A) = (q, \epsilon)$ ✓

$\delta(q, a, A) = (q, \alpha)$

$\delta(q, a, S) = (q, AA)$

$\delta(q, a, A) = \{(q, S), (q, \epsilon)\} \Rightarrow$ NDPA $\Rightarrow$ Non-Determi-nism

$\delta(q, b, A) = (q, S)$

CFG to PDA Conversion-Eg 2

Construct a PDA equivalent to the following grammar:

S->aSa|bSb|a

$A \to a$
$B \to b$

① Now, converted in GNF i.e. $A \to a\alpha$ format

$S \to aSA \mid bSB \mid a$

$\delta(q, a, S) = (q, SA)$
$\delta(q, b, S) = (q, SB)$
$\delta(q, a, S) = (q, \epsilon)$

$\to$ for $S \to a\underbrace{SA}_{\alpha} \mid b\underbrace{SB}_{\alpha} \mid a\underbrace{\epsilon}_{\alpha}$

for $A \to a$
$\delta(q, a, A) = (q, \Lambda)$

for $B \to b$
$\delta(q, b, B) = (q, \Lambda)$

Final Rules

$\delta(q, a, S) = \{(q, SA), (q, \epsilon)\}$ Non-Determinism
$\delta(q, b, S) = (q, SB)$
$\delta(q, a, A) = (q, \Lambda)$
$\delta(q, b, B) = (q, \Lambda)$

CFG to PDA Conversion-Eg 3

Construct a PDA equivalent to the following grammar:

E->+EE|*EE|id ϵ

$\delta(q, +, E) = (q, EE)$

$\delta(q, *, E) = (q, EE)$

$\delta(q, id, E) = (q, \epsilon)$

final rules

PDA to CFG Conversion

- Given a PDA accepting a language L, we can obtain a CFG generating L

PDA to CFG Conversion

- Construct a CFG G=(V,T,P,S) where
- V is a set of objects of the form [q,A,p]
- where p and q are in Q and A is in **r**

- If (q,a,A)=(r,BB) then the move tells that if PDA M starts in state q with A on the top of the stack and gets “a” as next symbol in the input, then it enters into state r and replaces A by BB

PDA to CFG Conversion

- **The object** [q,A,p] derives to a string that allows PDA M to

1) Erase A from the top of the stack,
2) By starting in state q and
3) Ending up in state p using sequence of moves

PDA to CFG Conversion

- To erase A from the top of the stack by starting in state q and ending in state p, the PDA M has to consume a from input and enter into the state r
- Then by starting in state r get B erased by entering into any state say s using sequence of moves followed by getting again B erased by starting in a state s and ending up in state p

Sequence of moves:-

- Object [q,A,p] derives to a[r,B,s][s,B,p]
- Hence P contains the rules of the form
- **[q,a,p]->a[r,B,s][s,B,p]**

PDA to CFG Conversion

- Since strings accepted by PDA M are those , that allows PDA M to erase Z_0 from the top of the stack by starting in state q0 and ending up in any state
- The start symbol S will derive to objects [q0,z0,q] for every q in Q

PDA to CFG Conversion

Give the CFG generating the language accepted by the following PDA:-

M=({q0,q1},{0,1},{Z0,X}, **δ,q0,Z0,** ∅) where **δ is given below:**

1) **δ(q0,1,Z0)=(q0,XZ0)**
2) **δ(q0,1,X)=(q0,XX)**
3) **δ(q0,0,X)=(q1,X)**
4) **δ(q0, ε,Z0)=(q0, ε)**
5) **δ(q1,1,X)=(q1, ε)**
6) **δ(q1,0,Z0)=(q0,Z0)**

PDA to CFG Conversion

M=({q0,q1},{0,1},{Z0,X}, **δ,q0,Z0,** Ø) where **δ is given below:**

The productions are

- S->[q0,Z0,q0]
- S->[q0,Z0,q1]

PDA M to erase Z_0 from the top of the stack by starting in state q0 and ending up in any state
The start symbol S will derive to objects [q0,z0,q] for every q in Q

PDA to CFG Conversion

For Move $\delta(q0,1,Z0)=(q0,XZ0)$

- The productions are

The object [q,A,p] derives to a string that allows PDA M to

1) Erase A from the top of the stack,
2) By starting in state q and
3) Ending up in state p using sequence of moves

For Move $\delta(q0,1,X)=(q0,XX)$

- The productions are