23MT2014

THEORY OF COMPUTATION

Topic:

N-PDA FOR CFG

Session - 15

N-pda for CFG

Conversion of CFG to NPDA

Construction Principles

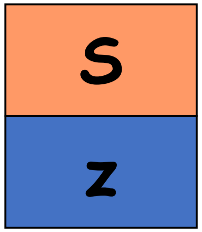
- We will construct an nPDA that can in some way, carry out a leftmost derivation of any string in the language

- We assume that the given CFG is in Greibach Normal Form.

Construction Principles

- The PDA will represent the derivation by keeping the variables in the right part of the sentinel form on its stack, while the left part consisting entirely of terminals, is identical with the input read.

- To begin with we put the Start symbol S on top of the stack.

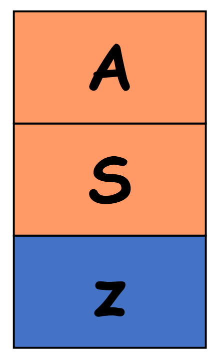


Stack

Construction Principles

Then for the productions of the type

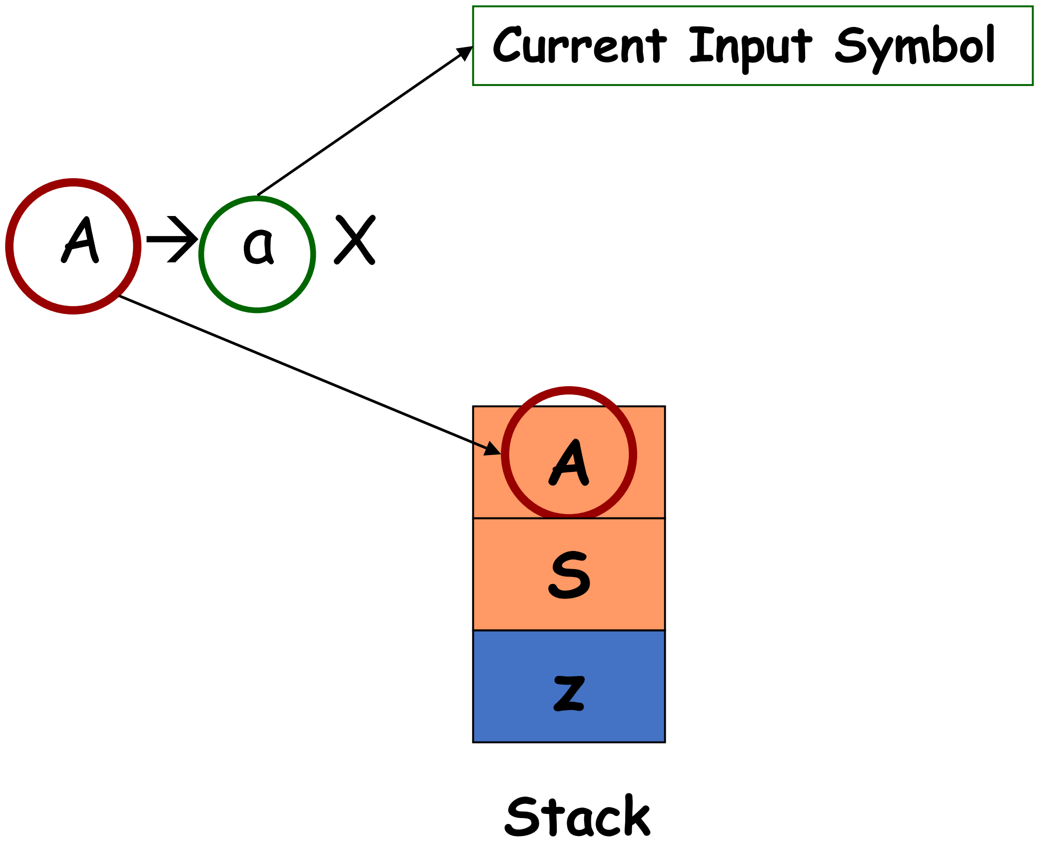
$$A \rightarrow a \; X$$



Stack

Construction Principles

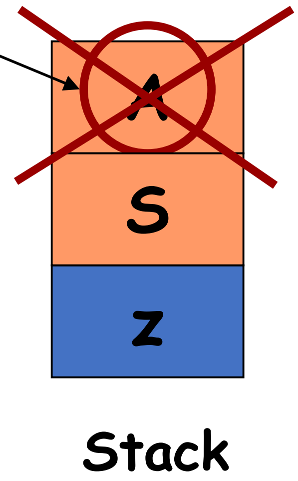
Then for the productions of the type



Construction Principles

Then for the productions of the type

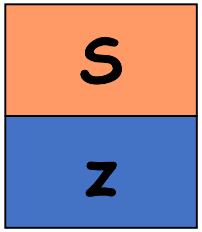
A → a X



Construction Principles

Then for the productions of the type

$$A \rightarrow a\ X$$

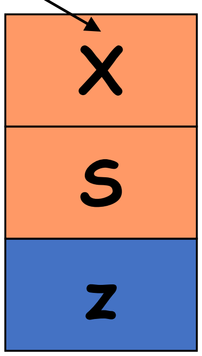


Stack

Construction Principles

Then for the productions of the type

$$A \rightarrow a\ X$$



Stack

Example #1

Construct a PDA that accepts the language generated by the Grammar with productions

$$S \rightarrow a\,S\,bb \mid a$$

Step #1: **Check whether the given Grammar is in GNF, if not convert it into GNF**

Converting the the Grammar in to GNF we have:

$$S \rightarrow a\,S\,A \mid a$$

$$A \rightarrow b\,B$$

$$B \rightarrow b$$

Generally

The PDA will have three states, q_0, q_1, q_2, out of which q_0 is the initial state and q_2 is the final state.

Step #2: **First the Start Symbol S is pushed on to the Stack**

$$\delta (q_0, \lambda , z) = \{ (q_1, Sz) \}$$

Step #3: **Now we will convert the productions in to transition function one by one:**

Prod: 1

$$S \rightarrow a\ S\ A$$

$$\boxed{\delta (q1, a , S) = \{ (q1, SA) \}}$$

Step #4:

Prod: 2

$S \rightarrow a\ (\lambda\ )$

$\delta\ (\ q1,\ a\ ,\ S\ )\ =\ \{\ (\ q1,\ \lambda\ )\ \}$

Step #5:

Prod: 3

$A \rightarrow b\ B$

$\delta\ (\ q1,\ b\ ,\ A\ )\ =\ \{\ (\ q1,\ B\ )\ \}$

Step #6: Prod: 4

$B \rightarrow b\ (\lambda\ )$

$\delta\ (\ q1,\ b\ ,\ B\ )\ =\ \{\ (\ q1,\ \lambda\ )\ \}$

Step #6: For Entering in to the final state

$\delta\ (\ q1,\ \lambda\ ,\ z\ )\ =\ \{\ (\ q2,\ \lambda\ )\ \}$

The PDA for the given Grammar is:

CFG

$S \rightarrow a\,S\,A \mid a$

$A \rightarrow b\,B$

$B \rightarrow b$

Equivalent PDA

$\delta(q_0, \lambda, z) = \{(q_1, Sz)\}$

$\delta(q1, a, S) = \{(q1, SA)\}$

$\delta(q1, a, S) = \{(q1, \lambda)\}$

$\delta(q1, b, A) = \{(q1, B)\}$

$\delta(q1, b, B) = \{(q1, \lambda)\}$

$\delta(q1, \lambda, z) = \{(q2, \lambda)\}$

q_2 is the final state

CFG for nPDA

Conversion of PDA to CFG

Assumptions regarding the conversion

- The PDA to be Converted has a single final state q_f that is entered if and only if the stack is empty;
- All the transitions must have the form

  - $\delta(q_i, a, A) = (c_1, c_2, c_3 \ldots c_n\}$ where

    $c_i = (q_j, \lambda)$ OR $c_i = (q_j, BC)$

(ie) Each move either increases or decreases the stack content by a single symbol.

We find a grammar whose variables are of the form $(q_i\ A\ q_j)$ and whose productions are such that

$$(q_i\ A\ q_j) \rightarrow^* v\ ,$$

if and only if the npda erases A from the stack while reading v and going from state q_i to q_j. "Erasing" means that A and its effects (ie all successive strings by which it is replaced) are removed from the stack bringing the symbol originally below A to the top.

If we can find such a grammar, and if we choose $(q_0\ z\ q_f)$ as its start symbol, then

$$(q_0\ w\ q_f) \rightarrow^* w,$$

if and only if the npda removes z while reading w and going from q_0 to q_f.

Therefore, the language generated by the grammar will be identical to the language accepted by the PDA

Construction of CFG

Let us look at the possible Transitions that can be made by the pda.

1

$$\delta(q_i, a, A) = (q_j, \lambda)$$

$$\boxed{(q_i A q_j) \rightarrow a}$$

2

$$\delta(q_i, a, A) = (q_j, BA)$$

$$\boxed{(q_i A q_k) \rightarrow a (q_j B q_l)(q_l A q_k)}$$

where q_k and q_l take on all possible values in Q

Example

Convert the npda to a CFG.

$$\delta(q_0, a, z) = \{(q_0, Az)\}$$

$$\delta(q_0, a, A) = \{(q_0, A)\}$$

$$\delta(q_0, b, A) = \{(q_1, \lambda)\}$$

$$\delta(q_1, \lambda, z) = \{(q_2, \lambda)\}$$

The above npda satisfies the 1st Condition but not the second one.

So we introduce a intermediate state q_3 to satisfy the second one and then we go for the conversion

The new set of transition rules are:

$\delta(q_0, a, z) = \{(q_0, Az)\}$ Trans No: 1 type 2

$\delta(q_0, a, A) = \{(q_3, \lambda)\}$ Trans No: 2 type 1

$\delta(q_3, \lambda, z) = \{(q_0, Az)\}$ Trans No: 3 type 2

$\delta(q_0, b, A) = \{(q_1, \lambda)\}$ Trans No: 4 type 1

$\delta(q_1, \lambda, z) = \{(q_2, \lambda)\}$ Trans No: 5 type 1

First take the transitions of type 1:

$\delta(q_0, a, A) = \{(q_3, \lambda)\}$ Trans No: 2 type 1

$$\boxed{(q_0\, A\, q_3) \rightarrow a}$$

$\delta(q_0, b, A) = \{(q_1, \lambda)\}$ Trans No: 4 type 1

$(q_0 \; A \; q_1) \rightarrow b$

$\delta(q_1, \lambda, z) = \{(q_2, \lambda)\}$ Trans No: 5 type 1

$(q_1 \; z \; q_2) \rightarrow \lambda$

Now take the Transitions of Type 2:

$$\delta(q_0, a, z) = \{(q_0, Az)\}$$ **Trans No: 1 type 2**

$(q_0 z q_0) \rightarrow a\,(q_0\ A\ q_0)\,(q_0\ z\ q_0) \mid a\,(q_0\ A\ q_1)\,(q_1\ z\ q_0)$

$a\,(q_0\ A\ q_2)\,(q_2\ z\ q_0) \mid a\,(q_0\ A\ q_3)\,(q_3\ z\ q_0)$

$(q_0 z q_1) \rightarrow a\,(q_0\ A\ q_0)\,(q_0\ z\ q_1) \mid a\,(q_0\ A\ q_1)\,(q_1\ z\ q_1)$

$a\,(q_0\ A\ q_2)\,(q_2\ z\ q_1) \mid a\,(q_0\ A\ q_3)\,(q_3\ z\ q_1)$

$(q_0 z q_2) \rightarrow a\,(q_0\ A\ q_0)\,(q_0\ z\ q_2) \mid a\,(q_0\ A\ q_1)\,(q_1\ z\ q_2)$

$a\,(q_0\ A\ q_2)\,(q_2\ z\ q_2) \mid a\,(q_0\ A\ q_3)\,(q_3\ z\ q_2)$

$(q_0 z q_3) \rightarrow a\,(q_0\ A\ q_0)\,(q_0\ z\ q_3) \mid a\,(q_0\ A\ q_1)\,(q_1\ z\ q_3)$

$a\,(q_0\ A\ q_3)\,(q_2\ z\ q_3) \mid a\,(q_0\ A\ q_3)\,(q_3\ z\ q_3)$

Trans No: 3 type 2

$$\delta(q_3, \lambda, z) = \{(q_0, Az)\}$$

$(q_3 z q_0) \rightarrow (q_0\ A\ q_0)(q_0\ z\ q_0) \mid (q_0\ A\ q_1)(q_1\ z\ q_0)$

$(q_0\ A\ q_2)(q_2\ z\ q_0) \mid (q_0\ A\ q_3)(q_3\ z\ q_0)$

$(q_3 z q_1) \rightarrow (q_0\ A\ q_0)(q_0\ z\ q_1) \mid (q_0\ A\ q_1)(q_1\ z\ q_1)$

$(q_0\ A\ q_2)(q_2\ z\ q_1) \mid (q_0\ A\ q_3)(q_3\ z\ q_1)$

$(q_3 z q_2) \rightarrow (q_0\ A\ q_0)(q_0\ z\ q_2) \mid (q_0\ A\ q_1)(q_1\ z\ q_2)$

$(q_0\ A\ q_2)(q_2\ z\ q_2) \mid (q_0\ A\ q_3)(q_3\ z\ q_2)$

$(q_3 z q_3) \rightarrow (q_0\ A\ q_0)(q_0\ z\ q_3) \mid (q_0\ A\ q_1)(q_1\ z\ q_3)$

$(q_0\ A\ q_3)(q_2\ z\ q_3) \mid (q_0\ A\ q_3)(q_3\ z\ q_3)$

The Start Variable will be $(q_0\ z\ q_2)$.

The string aab is accepted by the given pda with the following configurations

$$
\begin{aligned}
(q_0, aab, z) &\Rightarrow (q_0, ab, Az) \\
&\Rightarrow (q3, b, z) \\
&\Rightarrow (q_0, b, Az) \\
&\Rightarrow (q1, \lambda, z) \\
&\Rightarrow (q2, \lambda, \lambda)
\end{aligned}
$$

The Corresponding Derivation with G is:

$$
\begin{aligned}
(q_0 z q_2) &\Rightarrow a\ (q_0\ A\ q_3)\ (q_3\ z\ q_2) \\
&\Rightarrow a\ a\ (q_3\ z\ q_2) \\
&\Rightarrow a\ a\ (q_0\ A\ q_1)\ (q_1\ z\ q_2) \\
&\Rightarrow a\ a\ b\ (q_1\ z\ q_2) \\
&\Rightarrow a\ a\ b.
\end{aligned}
$$

- Find a context-free grammar that generates the language accepted by the NPDA.

$M = (\{q0, q1\}, \{a, b\}, \{A, z\}, \delta, q0, z, \{q1\})$,

with transitions

$\delta(q0, a, z) = \{(q0, Az)\}$,

$(q0, b, A) = \{(q0, AA)\}$,

$\delta(q0, a, A) = \{(q1)\lambda\}$

Self Assessment Questions

Q.1 Which of the following statements is true regarding Non-deterministic Pushdown Automata (NPDA)?

a) NPDA can recognize more languages than Deterministic Pushdown Automata (DPDA).
b) NPDA can recognize only regular languages.
c) NPDA can recognize only context-free languages.
d) NPDA can recognize only regular expressions.

Answer: a) NPDA can recognize more languages than Deterministic Pushdown Automata (DPDA).

Self Assessment Questions

Q.2. Which of the following is a key characteristic of Context-Free Grammars (CFG)?

a) CFG can generate any language.
b) CFG can generate only regular languages.
c) CFG can generate only context-free languages.
d) CFG can generate any recursively enumerable language.

Answer: c) CFG can generate only context-free languages.

Self Assessment Questions

Q.3. What is the relationship between Non-deterministic Pushdown Automata (NPDA) and Context-Free Grammars (CFG)?

a) For every CFG, there exists an equivalent NPDA that recognizes the same language.
b) NPDA and CFG are completely unrelated models of computation.
c) NPDA can recognize more languages than CFG.
d) CFG can generate only regular languages, while NPDA can generate any language.

Answer: c) CFG can generate only context-free languages.

Terminal Questions

Q.1.Explain the concept of non-determinism in Non-deterministic Pushdown Automata (NPDA). How does non-determinism affect the language recognition process?
Define a Context-Free Grammar (CFG).

Q.2. Describe the components of a CFG and their roles in generating languages.

Q.3. Discuss the relationship between Non-deterministic Pushdown Automata (NPDA) and Context-Free Grammars (CFG). How are they related in terms of language recognition? Explain with an example.

THANK YOU

(DEEMED TO BE UNIVERSITY)

Team – TOC