Formal languages and Automata
形式语言与自动机

Chapter7 PUSHDOWN AUTOMATA

Beijing University of Posts and Telecommunications

北京邮电大学

刘咏彬 liiuyb@bupt.edu.cn

2023.12

Creating Pushdown Automata (PDA) with Stack Storage Mechanism

- What is required to recognize a context-free language?
  - $L = \{a^n b^n : n \geq 0\}$, we must count the number of a's.
  - $L = \{ww^R\}$,we need to store and match a sequence of symbols in reverse order.
  - Stack Storage Mechanism

- How a Pushdown Automaton (PDA) Functions:
  - Each move of the control unit reads a symbol from the input file
  - The control unit's moves are determined by the current input,the state of the control unit and the symbol on top of the stack.
  - The result of each move is a new state for the control unit and a change at the top of the stack.

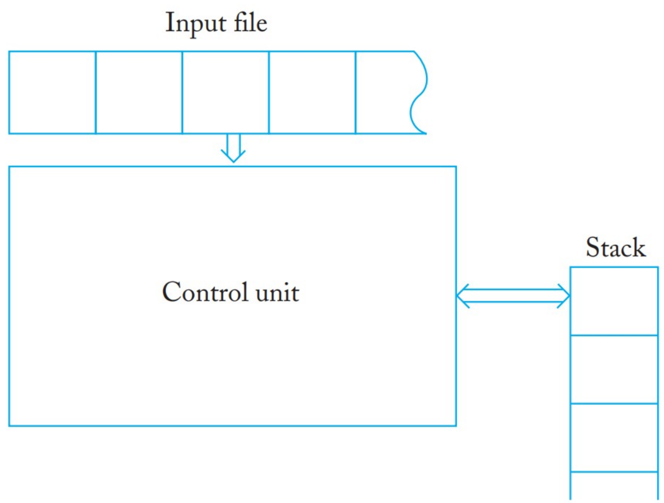
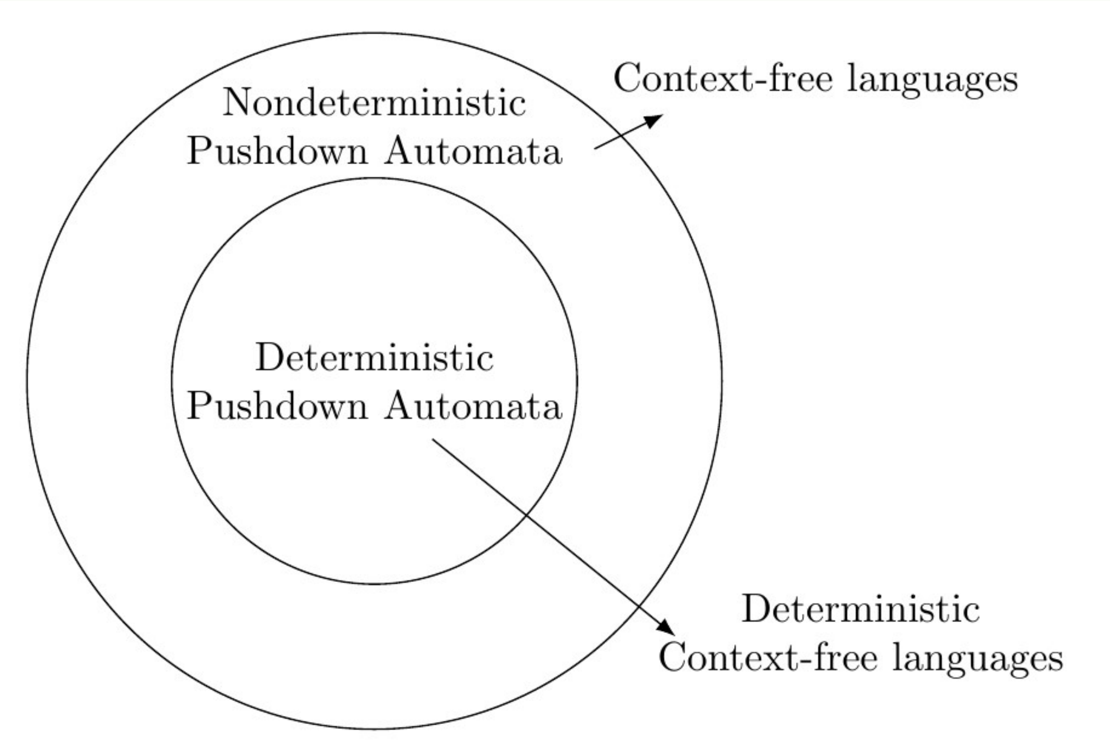


FIGURE 7.1

Deterministic vs. Nondeterministic PDAs in Context-Free Languages

- The relationship between pushdown automaton and context-free languages
- Lack of equivalence between deterministic and nondeterministic PDAs



Nondeterministic Pushdown Accepter

- **DEFINITION 7.1**

A nondeterministic pushdown accepter (npda,非确定下推自动机) is defined by the septuple（七元组）

$$M = (Q, \Sigma, \Gamma, \delta, q_0, z, F),$$

where

- Q is a finite set of internal states of the control unit,
- Σ is the input alphabet,
- Γ is a finite set of symbols called the stack alphabet（栈符号）,
- $\delta: Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to 2^{Q \times \Gamma^*}$ is the transition function, $2^{Q \times \Gamma^*}$ is the power set of $Q \times \Gamma^*$,
- $q_0 \in Q$ is the initial state of the control unit,
- $z \in \Gamma$ is the stack start symbol（栈开始符号）,
- $F \subseteq Q$ is the set of final states.

EXAMPLE 7.1

$$\delta(q_1, a, b) = \{(q_2, cd), (q_3, \lambda)\}.$$

1. The current input symbol
2. The symbol at the top of the stack,
3. The string that replaces the top of the stack

adds a 1 to the stack when an a is read

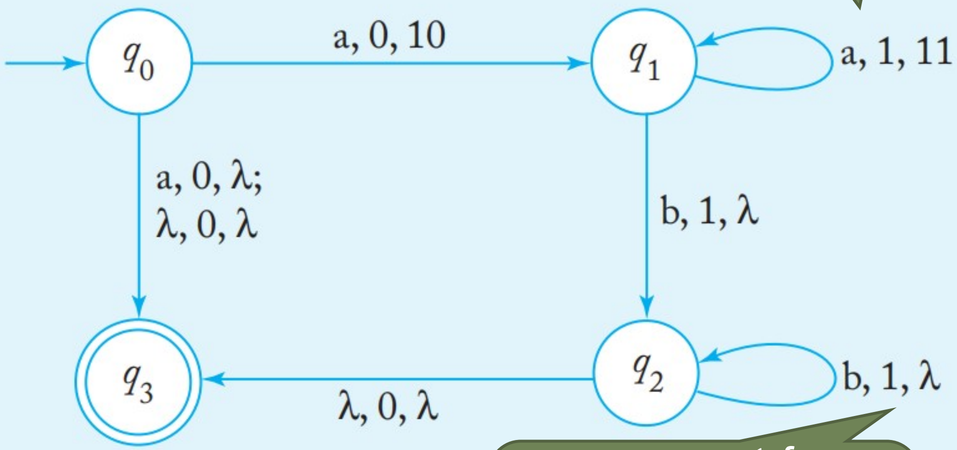
EXAMPLE 7.2 $L = \{a^n b^n : n \geq 0\} \cup \{a\}$.

Consider an npda with

$$Q = \{q_0, q_1, q_2, q_3\},$$
$$\Sigma = \{a, b\},$$
$$\Gamma = \{0, 1\},$$
$$z = 0,$$
$$F = \{q_3\},$$

with initial state q_0 and

$$\delta(q_0, a, 0) = \{(q_1, 10), (q_3, \lambda)\},$$
$$\delta(q_0, \lambda, 0) = \{(q_3, \lambda)\},$$
$$\delta(q_1, a, 1) = \{(q_1, 11)\},$$
$$\delta(q_1, b, 1) = \{(q_2, \lambda)\},$$
$$\delta(q_2, b, 1) = \{(q_2, \lambda)\},$$
$$\delta(q_2, \lambda, 0) = \{(q_3, \lambda)\}.$$



removes a 1 from the top of the stack when a b is encountered

If $\delta(q_i, a, z)$ is unspecified, it defaults to the null set, indicating a dead configuration for the NPDA

Instantaneous Description

- Notation for describing a configuration (格局) of an npda during the processing of a string.

$$(q, w, u)$$

  - Key Elements:
    - The current state of the control unit q,
    - The unread part of the input string w,
    - The current contents of the stack u.

u的最左符号为栈顶符号，最右符号为栈底的符号

- A move from one instantaneous description to another will be denoted by the symbol $\vdash$, thus

$$(q_1, aw, bx) \vdash (q_2, w, yx)$$

is possible if and only if

$$(q_2, y) \in \delta(q_1, a, b)$$

Language Accepted by PDA

- **DEFINITION 7.2**

Let $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$, be a nondeterministic pushdown automaton.

The language accepted by M is the set

$$L(M) = \{w \in \Sigma^*: (q_0, w, z) \vdash^*_M (p, \lambda, u),\ p \in F,\ u \in \Gamma^*\}$$

In words, the language accepted by M is the set of all strings that can put M into a final state at the end of the string. The final stack content u is irrelevant to this definition of acceptance.

- 补充（见习题19）：M还有另一种接受语言的方式，用空栈接受的语言

$$N(M) = \{w \mid (q_0, w, Z_0) \vdash^*_M (p, \lambda, \lambda)\}$$

读入完输入字符串后，栈符号为空

任意CFL均存在使用空栈方式接受此语言的PDA，也存在使用终态方式接受

eg. PDA M=({q}, T, V, δ, q, S, Φ)

19. An alternative to Definition 7.2 for language acceptance is to require the stack to be empty when the end of the input string is reached. Formally, an npda M is said to accept the language $N(M)$ by empty stack if

$$N(M) = \left\{w \in \Sigma^* : (q_0, w, z) \overset{*}{\vdash}_M (p, \lambda, \lambda)\right\},$$

where p is any element in Q. Show that this notion is effectively equivalent to Definition 7.2, in the sense that for any npda M, there exists an npda $\widehat{M}$ such that $L(M) = N\left(\widehat{M}\right)$, and vice versa.

EXAMPLE 7.4

Construct an npda for the language

$$L = \left\{w \in \{a, b\}^* : n_a(w) = n_b(w)\right\}.$$

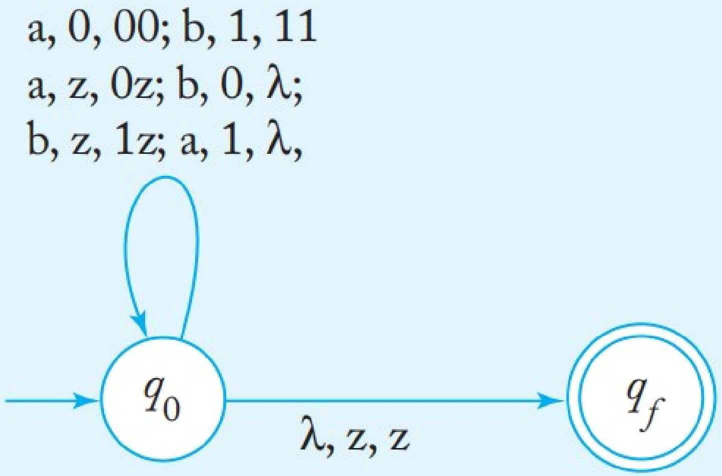


FIGURE 7.3

In processing the string *baab*, the npda makes the moves

$$\begin{aligned}(q_0, baab, z) &\vdash (q_0, aab, 1z) \vdash (q_0, ab, z)\\ &\vdash (q_0, b, 0z) \vdash (q_0, \lambda, z) \vdash (q_f, \lambda, z)\end{aligned}$$

and hence the string is accepted.

EXAMPLE 7.5

To construct an npda for accepting the language

$$L = \left\{ ww^R : w \in \{a, b\}^+ \right\},$$

A solution to the problem is given by $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$, where

$$\begin{aligned} Q &= \{q_0, q_1, q_2\}, \\ \Sigma &= \{a, b\}, \\ \Gamma &= \{a, b, z\}, \\ F &= \{q_2\}. \end{aligned}$$

q_0——记录阶段
q_1——匹配阶段

The transition function can be visualized as having several parts: a set to push w on the stack,

$$\begin{aligned} \delta(q_0, a, a) &= \{(q_0, aa)\}, \\ \delta(q_0, b, a) &= \{(q_0, ba)\}, \\ \delta(q_0, a, b) &= \{(q_0, ab)\}, \\ \delta(q_0, b, b) &= \{(q_0, bb)\}, \\ \delta(q_0, a, z) &= \{(q_0, az)\} \\ \delta(q_0, b, z) &= \{(q_0, bz)\} \end{aligned}$$

$$\begin{aligned} \delta(q_0, \lambda, a) &= \{(q_1, a)\}, \\ \delta(q_0, \lambda, b) &= \{(q_1, b)\}, \end{aligned}$$

$$\begin{aligned} \delta(q_1, a, a) &= \{(q_1, \lambda)\}, \\ \delta(q_1, b, b) &= \{(q_1, \lambda)\}, \end{aligned}$$

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

The sequence of moves in accepting $abba$ is

$$\begin{aligned} (q_0, abba, z) &\vdash (q_0, bba, az) \vdash (q_0, ba, baz) \\ &\vdash (q_1, ba, baz) \vdash (q_1, a, az) \vdash (q_1, \lambda, z) \vdash (q_2, z). \end{aligned}$$

补 PDA示例

$L=\{ww^R|w\in\{a,b\}^*\}$的PDA

$\delta(q_0, a, Z_0)=\{(q_0, AZ_0)\}$

$\delta(q_0, b, Z_0)=\{(q_0, BZ_0)\}$

$\delta(q_0, a, B)=\{(q_0, AB)\}$

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

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

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

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

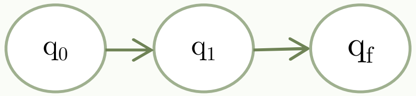
$\delta(q_1, b, B)=\{(q_1, \lambda)\}$

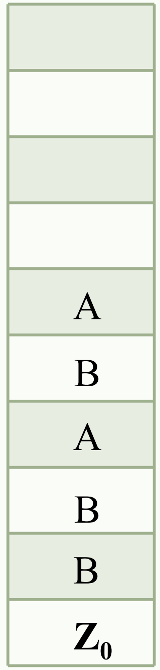
$\delta(q_1, \lambda, Z_0)=\{(q_f, \lambda)\}$

$\delta(q_0, \lambda, Z_0)=\{(q_f, \lambda)\}$ #专用于接受λ

识别句子：

b b a b a a b a b b





PUSHDOWN AUTOMATA AND CONTEXT-FREE LANGUAGES

- Every context-free language can be accepted by a corresponding NPDA.
- The central idea is to implement a leftmost derivation for any string in the language.

CFG->NPDA

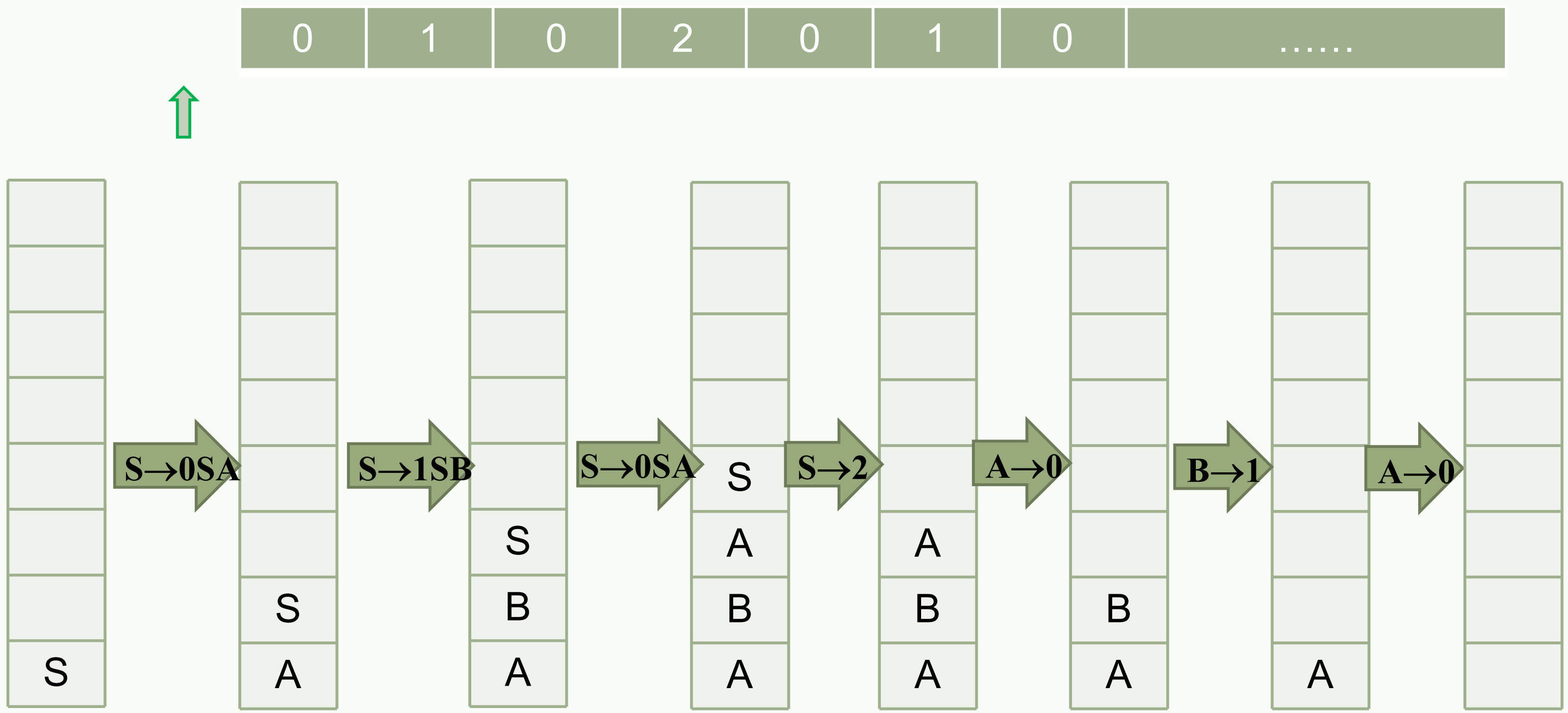
- Assumption(前提):
  - The language is generated by a grammar in Greibach Normal Form
- Construction of NPDA from a GNF grammar:
  - Represents derivation by keeping variables of the sentential form's right part on its stack.
  - Use the leftmost terminal to matches the input symbol.
  - Starts with the initial symbol on the stack.
  - To simulate production $A \rightarrow ax$, the stack top must have variable A and the input symbol as terminal a.
  - The stack top variable is removed and replaced with the variable string x.

补 构造PDA 模拟GNF最左派生

$L=\{w2w^T \mid w\in \{0,1\}^*\}$

分析：句子0102010的最左派生和我们希望相应的PDA M的动作。

G_2：$S\to 2|0SA|1SB$
$A\to 0$
$B\to 1$



S ⇒0SA ⇒01SBA ⇒010SABA ⇒0102ABA⇒01020BA⇒010201A ⇒0102010

EXAMPLE 7.6

Construct a pda that accepts the language generated by a grammar with productions

$$S \to aSbb|a.$$

We first transform the grammar into Greibach normal form, changing the productions to

$$\begin{aligned} S &\to aSA|a, \\ A &\to bB, \\ B &\to b. \end{aligned}$$

The corresponding automaton will have three states $\{q_0, q_1, q_2\}$, with initial state q_0 and final state q_2. First, the start symbol S is put on the stack by

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

$$\delta(q_1, a, S) = \{(q_1, SA), (q_1, \lambda)\}.$$

In an analogous manner, the other productions give

$$\begin{aligned} \delta(q_1, b, A) &= \{(q_1, B)\}, \\ \delta(q_1, b, B) &= \{(q_1, \lambda)\}. \end{aligned}$$

The appearance of the stack start symbol on top of the stack signals the completion of the derivation and the pda is put into its final state by

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

NPDA Construction for λ-free Context-Free Languages

- **THEOREM 7.1**

  For any context-free language L, there exists an npda M such that

  $$L = L(M).$$

- Construction step1:
  - For any λ-free context-free language L, a corresponding context-free grammar in Greibach normal form exists, denoted as $G = (V, T, S, P)$
  - We construct an NPDA $M = (q_0, q_1, q_f, T, V \cup z, \delta, q_0, z, q_f)$ that simulates leftmost derivations of G.
  - In this NPDA, the stack retains the unprocessed part of the sentential form, aligning its terminal prefix with the input string's corresponding prefix.
  - The input alphabet of M matches G's terminals, and the stack alphabet includes G's variables.

NPDA Construction for λ-free Context-Free Languages

- Construction step2:

$$\delta\ (q_0, \lambda, z)\ =\ \{(q_1, Sz)\}\ , \tag{7.1}$$

  - After the move (7.1) , the stack contains the start symbol S of the derivation. (The stack start symbol z is a marker to allow us to detect the end of the derivation.)

$$(q_1, u)\ \in\ \delta\ (q_1, a, A), \tag{7.2}$$

  whenever $A\ \rightarrow\ au$ is in P.

  - (7.2) reads input a and removes the variable A from the stack, replacing it with u. In this way it generates the transitions that allow the pda to simulate all derivations.

$$\delta\ (q_1, \lambda, z)\ =\ \{(q_f\ , z)\}\ , \tag{7.3}$$

  - (7.3) get M into a final state.

Proof of L(M)= L(G) is omitted here.

EXAMPLE 7.7

Consider the grammar

$$\begin{aligned}
S &\to aA, \\
A &\to aABC \,|bB|\, a, \\
B &\to b, \\
C &\to c.
\end{aligned}$$

Since the grammar is already in Greibach normal form, we can use the construction in the previous theorem immediately. In addition to rules

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

and

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

the pda will also have transition rules

$$\begin{aligned}
\delta\,(q_1, a, S) &= \{(q_1, A)\}\,, \\
\delta\,(q_1, a, A) &= \{(q_1, ABC)\,, (q_1, \lambda)\}\,, \\
\delta\,(q_1, b, A) &= \{(q_1, B)\}\,, \\
\delta\,(q_1, b, B) &= \{(q_1, \lambda)\}\,, \\
\delta\,(q_1, c, C) &= \{(q_1, \lambda)\}\,.
\end{aligned}$$

The sequence of moves made by M in processing $aaabc$ is

$$\begin{aligned}
(q_0, aaabc, z) &\vdash (q_1, aaabc, Sz) \\
&\vdash (q_1, aabc, Az) \\
&\vdash (q_1, abc, ABCz) \\
&\vdash (q_1, bc, BCz) \\
&\vdash (q_1, c, Cz) \\
&\vdash (q_1, \lambda, z) \\
&\vdash (q_f, \lambda, z)\,.
\end{aligned}$$

This corresponds to the derivation

$$S \Rightarrow aA \Rightarrow aaABC \Rightarrow aaaBC \Rightarrow aaabC \Rightarrow aaabc.$$

Other CFG->NPDA Methods

- For example, for productions of the form

$$A \rightarrow Bx,$$

we remove A from the stack and replace it with Bx, but consume no input symbol.

- For productions of the form

$$A \rightarrow abCx,$$

we must first match the ab in the input against a similar string in the stack and then replace A with Cx.

PDA->CFG(1)

- Reverse the process in Theorem 7.1 so that the grammar simulates the moves of the pda? No,Not that simple.
- Assume
  1. It has a single final state q_f that is entered if and only if the stack is empty;
  2. With $a \in \Sigma \cup \{\lambda\}$, all transitions must have the form $\delta(q_i, a, A) = \{c_1, c_2, ..., c_n\}$, where

$$c_i = (q_j, \lambda), \tag{7.5}$$

or

$$c_i = (q_j, BC). \tag{7.6}$$

That is, each move either increases or decreases the stack content by a single symbol.

PDA->CFG(2)

- Construction
  - A grammar whose variables are of the form $(q_i A q_j)$ and whose productions are such that

    $$(q_i A q_j) \overset{*}{\Rightarrow} v,$$

    If and only if the npda erases A from the stack while reading v and going from state q_i to state q_j.
  - “Erasing” here means that A and its effects (i.e., all the 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 $(q0zqf\ )$ as its start symbol, then

    $$(q_0 z q_f\ ) \overset{*}{\Rightarrow} w$$

PDA->CFG(3)

- To construct a grammar that satisfies these conditions, we examine the different types of transitions that can be made by the npda.
- Since (7.5) involves an immediate erasure of A, the grammar will have a corresponding production

$$(q_i A q_j) \to a.$$

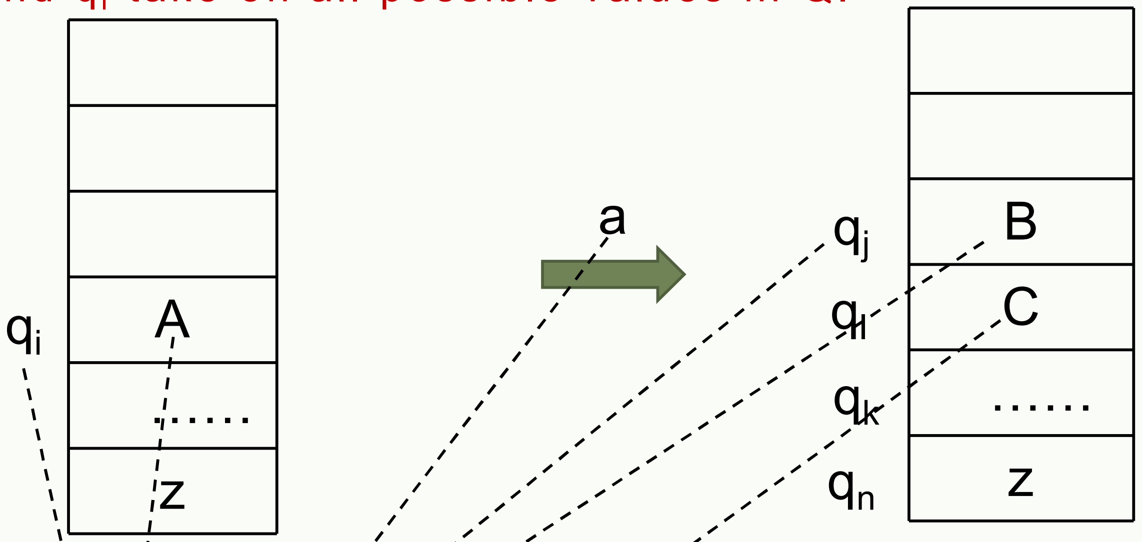
$\delta(q_i, a, A) = \{(q_j, \lambda), \ldots\}$

- Productions of type (7.6) generate the set of rules

$$(q_i A q_k) \to a\,(q_j B q_l)\,(q_l C q_k),$$

$\delta(q_i, a, A) = \{(q_j, BC), \ldots\}$

where q_k and q_l take on all possible values in Q.



$$(q_i A q_k) \to a(q_j B q_l)(q_l C q_k)$$

EXAMPLE 7.8

Consider the npda with transitions

$$\begin{aligned}
\delta\left(q_0, a, z\right) &= \left\{\left(q_0, Az\right)\right\},\\
\delta\left(q_0, a, A\right) &= \left\{\left(q_0, A\right)\right\},\\
\delta\left(q_0, b, A\right) &= \left\{\left(q_1, \lambda\right)\right\},\\
\delta\left(q_1, \lambda, z\right) &= \left\{\left(q_2, \lambda\right)\right\}.
\end{aligned}$$

Using q_0 as the initial state and q_2 as the final state, the npda satisfies condition 1 above, but not 2. To satisfy the latter, we introduce a new state q_3 and an intermediate step in which we first remove the A from the stack, then replace it in the next move. The new set of transition rules is

$$\begin{aligned}
\delta\left(q_0, a, z\right) &= \left\{\left(q_0, Az\right)\right\},\\
\delta\left(q_3, \lambda, z\right) &= \left\{\left(q_0, Az\right)\right\},\\
\delta\left(q_0, a, A\right) &= \left\{\left(q_3, \lambda\right)\right\},\\
\delta\left(q_0, b, A\right) &= \left\{\left(q_1, \lambda\right)\right\},\\
\delta\left(q_1, \lambda, z\right) &= \left\{\left(q_2, \lambda\right)\right\}.
\end{aligned}$$

The last three transitions are of the form (7.5) so that they yield the corresponding productions

$$\left(q_0 A q_3\right) \rightarrow a, \quad \left(q_0 A q_1\right) \rightarrow b, \quad \left(q_1 z q_2\right) \rightarrow \lambda.$$

Consider the npda with transitions

$$\begin{aligned}
\delta(q_0, a, z) &= \{(q_0, Az)\}, \\
\delta(q_0, a, A) &= \{(q_0, A)\}, \\
\delta(q_0, b, A) &= \{(q_1, \lambda)\}, \\
\delta(q_1, \lambda, z) &= \{(q_2, \lambda)\}.
\end{aligned}$$

From the first two transitions we get the set of productions

$$\begin{aligned}
(q_0 z q_0) &\to a(q_0 A q_0)(q_0 z q_0) | a(q_0 A q_1)(q_1 z q_0) | \\
&\quad a(q_0 A q_2)(q_2 z q_0) | a(q_0 A q_3)(q_3 z q_0), \\
(q_0 z q_1) &\to a(q_0 A q_0)(q_0 z q_1) | a(q_0 A q_1)(q_1 z q_1) | \\
&\quad a(q_0 A q_2)(q_2 z q_1) | a(q_0 A q_3)(q_3 z q_1),
\end{aligned}$$

$$\begin{aligned}
(q_0 z q_2) &\to a(q_0 A q_0)(q_0 z q_2) | a(q_0 A q_1)(q_1 z q_2) | \\
&\quad a(q_0 A q_2)(q_2 z q_2) | a(q_0 A q_3)(q_3 z q_2), \\
(q_0 z q_3) &\to a(q_0 A q_0)(q_0 z q_3) | a(q_0 A q_1)(q_1 z q_3) | \\
&\quad a(q_0 A q_2)(q_2 z q_3) | a(q_0 A q_3)(q_3 z q_3),
\end{aligned}$$

$$\begin{aligned}
(q_3 z q_0) &\to (q_0 A q_0)(q_0 z q_0) | (q_0 A q_1)(q_1 z q_0) | (q_0 A q_2)(q_2 z q_0) | (q_0 A q_3)(q_3 z q_0), \\
(q_3 z q_1) &\to (q_0 A q_0)(q_0 z q_1) | (q_0 A q_1)(q_1 z q_1) | (q_0 A q_2)(q_2 z q_1) | (q_0 A q_3)(q_3 z q_1), \\
(q_3 z q_2) &\to (q_0 A q_0)(q_0 z q_2) | (q_0 A q_1)(q_1 z q_2) | (q_0 A q_2)(q_2 z q_2) | (q_0 A q_3)(q_3 z q_2), \\
(q_3 z q_3) &\to (q_0 A q_0)(q_0 z q_3) | (q_0 A q_1)(q_1 z q_3) | (q_0 A q_2)(q_2 z q_3) | (q_0 A q_3)(q_3 z q_3).
\end{aligned}$$

This looks quite complicated, but can be simplified. A variable that does not occur on the left side of any production must be useless, so we can immediately eliminate $(q_0 A q_0)$ and $(q_0 A q_2)$. Also, by looking at the transition graph of the modified npda, we see that there is no path from q_1 to q_0, from q_1 to q_1, from q_1 to q_3, and from q_2 to q_2, which makes the associated variables also useless. When we eliminate all these useless productions, we are left with the much shorter grammar

$$
\begin{aligned}
(q_0 A q_3) &\rightarrow a, \\
(q_0 A q_1) &\rightarrow b, \\
(q_1 z q_2) &\rightarrow \lambda, \\
(q_0 z q_0) &\rightarrow a(q_0 A q_3)(q_3 z q_0), \\
(q_0 z q_1) &\rightarrow a(q_0 A q_3)(q_3 z q_1), \\
(q_0 z q_2) &\rightarrow a(q_0 A q_1)(q_1 z q_2) | a(q_0 A q_3)(q_3 z q_2), \\
(q_0 z q_3) &\rightarrow a(q_0 A q_3)(q_3 z q_3), \\
(q_3 z q_0) &\rightarrow (q_0 A q_3)(q_3 z q_0), \\
(q_3 z q_1) &\rightarrow (q_0 A q_3)(q_3 z q_1), \\
(q_3 z q_2) &\rightarrow (q_0 A q_1)(q_1 z q_2) | (q_0 A q_3)(q_3 z q_2), \\
(q_3 z q_3) &\rightarrow (q_0 A q_3)(q_3 z q_3),
\end{aligned}
$$

with start variable $(q_0 z q_2)$.

EXAMPLE 7.9

Consider the string $w = aab$. This is accepted by the pda in Example 7.8, with successive configurations

$$\begin{aligned}(q_0, aab, z) &\vdash (q_0, ab, Az)\\ &\vdash (q_3, b, z)\\ \\ &\vdash (q_0, b, Az)\\ &\vdash (q_1, \lambda, z)\\ &\vdash (q_2, \lambda, \lambda).\end{aligned}$$

The corresponding derivation with G is

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

The steps in the proof of the following theorem will be easier to understand if you notice the correspondence between the successive instantaneous descriptions of the pda and the sentential forms in the derivation. The first q_i in the leftmost variable of every sentential form is the current state of the pda, while the sequence of middle symbols is the same as the stack content.

NPDA accepts CFL

- **THEOREM 7.2**

  If $L = L(M)$ for some npda M, then L is a context-free language.

Proof: Assume that $M = (Q, \Sigma, \Gamma, \delta, q_0, z, \{q_f\})$ satisfies conditions 1 and 2 above. We use the suggested construction to get the grammar $G = (V, T, S, P)$, with $T = \Sigma$ and V consisting of elements of the form $(q_i c q_j)$. We will show that the grammar so obtained is such that for all $q_i, q_j, \in Q, A \in \Gamma, X \in \Gamma^*, u, v \in \Sigma^*$,

$$(q_i, uv, AX) \overset{*}{\vdash} (q_j, v, X) \tag{7.7}$$

implies that

$$(q_i A q_j) \overset{*}{\Rightarrow} u,$$

and vice versa.

The first part is to show that, whenever the npda is such that the symbol A and its effects can be removed from the stack while reading u and going from state q_i to q_j, then the variable $(q_i A q_j)$ can derive u. This is not

...

If we now apply the conclusion to

$$(q_0, w, z) \overset{*}{\vdash} (q_f, \lambda, \lambda),$$

we see that this can be so if and only if

$$(q_0 z q_f) \overset{*}{\Rightarrow} w.$$

Consequently $L(M) = L(G)$. ■

END