PUSHDOWN AUTOMATA

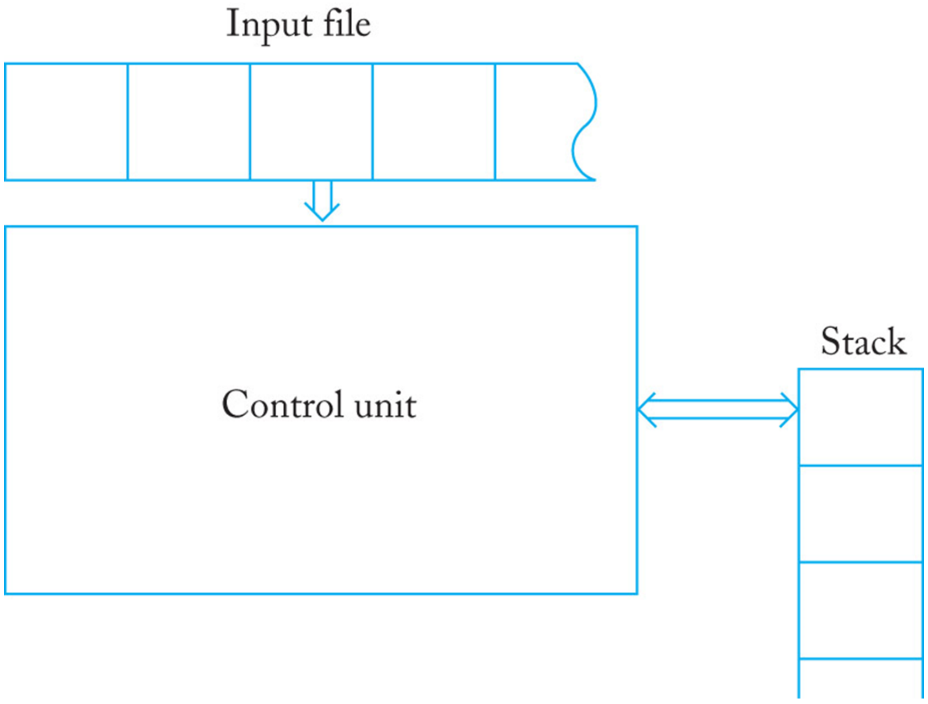
Chap. 7

Learning Objectives

- Describe the components of a *Nondeterministic PushDown Automaton (NPDA).*
- State whether an input string is *accepted* by a NPDA.
- Construct a pushdown automaton to accept a specific language.
- Given a Context-Free Grammar in Greibach Normal Form, construct the corresponding PushDown Automaton.
- Describe the differences between *Deterministic* and *Nondeterministic Pushdown Automata*: DPDA vs. NPDA.
- Describe the differences between Deterministic and general Context-Free Languages: DCFL vs. *CFL*.

Nondeterministic Pushdown Automata

- A *pushdown automaton* is a model of computation designed to process *context-free languages*.
- Pushdown automata use a *stack* as storage mechanism.



Nondeterministic Pushdown Automata

- Definition 7.1: A *Nondeterministic PushDown Automaton* (NPDA) is defined by the septuple

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

  - Q is a finite set of *states* of the control unit,
  - Σ is an *input alphabet*,
  - Γ is a finite set of symbols, called the *stack alphabet*,
  - $\delta: Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to \wp(Q \times \Gamma^*)$ is a transition function,
  - $q_0 \in Q$ is an *initial state*,
  - $z \in \Gamma$ is a *stack start symbol (optional)*,
  - $F \subseteq Q$ is a set of *final states*.
- Input to the transition function δ consists of a triple consisting of a state, input symbol (or λ), and the symbol at the top of stack.
- Output of δ consists of a new state and new top of stack.
- Transitions can be used to model common stack operations.

Example: NDPA Transitions

- <u>Example 7.1:</u> the sample transition rule:

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

- According to this rule, when the control unit is in state q_1, the input symbol is a, and the top of the stack is b, two moves are possible:
  - (q_2, cd): the new state is q_2 and the symbols '*cd*' *replace* '*b*' on the stack.
  - (q_3, λ): the new state is q_3 and '*b*' is simply *removed* from the stack.
- If a particular transition is not defined, the corresponding (state, symbol, stack top) configuration represents a *dead* state.

Example: Nondeterministic PDA

- Example 7.2: Consider the NPDA, M, where

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

  with initial state q_0 and transition function given by:

  $\delta(q_0, a, 0) = \{ (q_1, 10), (q_3, \lambda) \}$

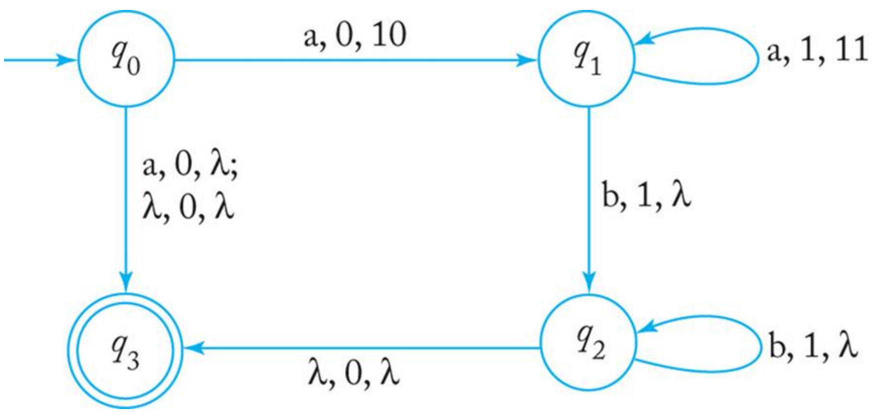
  $\delta(q_0, \lambda, 0) = \{ (q_3, \lambda) \}$: pop 0

  $\delta(q_1, a, 1) = \{ (q_1, 11) \}$: push 1

  $\delta(q_1, b, 1) = \{ (q_2, \lambda) \}$

  $\delta(q_2, b, 1) = \{ (q_2, \lambda) \}$

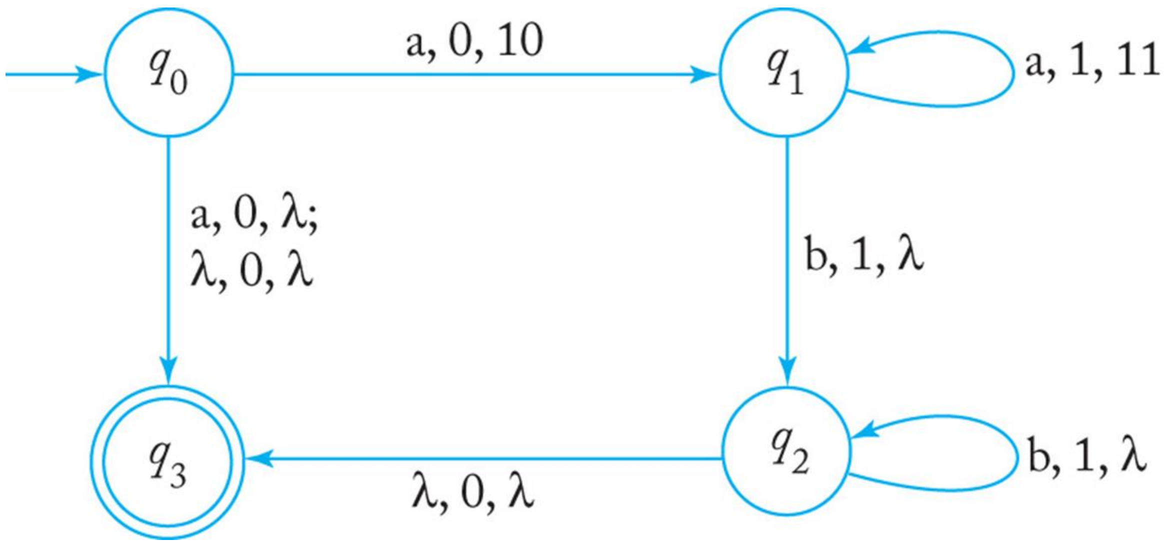
  $\delta(q_2, \lambda, 0) = \{ (q_3, \lambda) \}$



- As long as the control unit is in q_1, a *'1'* is pushed onto the stack when an *'a'* is read.
- The first *'b'* causes control to shift to q_2, which removes a symbol from the stack whenever a *'b'* is read.
- $L = \{a^n b^n \mid n \geq 0\} \cup \{a\}$

Transition Graphs

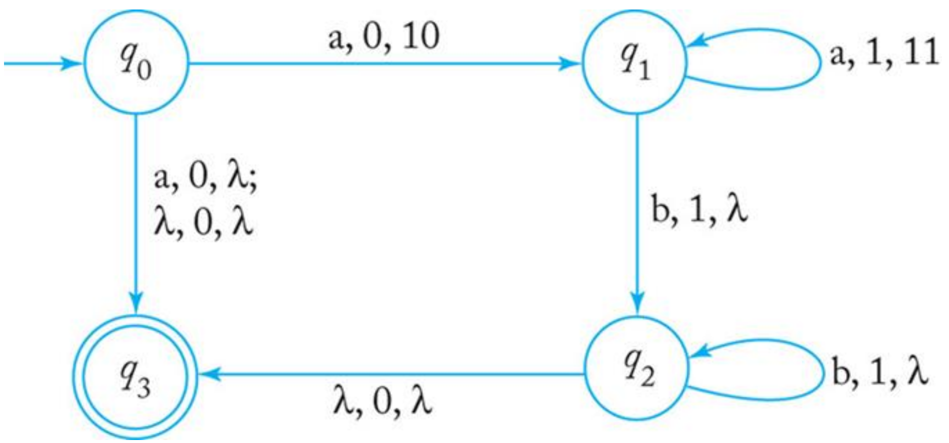
- In the transition graph for a NPDA, each edge is labeled with *the input symbol, the stack top*, and *the string that replaces the top of the stack*
- The transition graph of the NPDA in Example 7.2:



Instantaneous Descriptions

- To trace the operation of a NPDA, we must keep track of the *current state* of the control unit, the *stack contents*, and the *unread part of the input string*.
- An *instantaneous description* is a triplet (q, w, u) that describes state, unread input symbols, and stack contents (with the top as the *leftmost* symbol)
- A move is denoted by the symbol $\vdash$
- A partial trace of the NPDA in Example 7.2 with input string *ab* is

$$(q_0, ab, 0)$$
$$\vdash (q_1, b, 10)$$
$$\vdash (q_2, \lambda, \lambda 0)$$
$$\vdash (q_3, \lambda, \lambda)$$



The Language Accepted by a PDA

- Definition 7.2: Let $M=(Q, \Sigma, \Gamma, \delta, q_0, z, F)$ be a NDPA.

  The *language accepted* by M is

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

- i.e. the language accepted by a NPDA is the set of all strings that cause the NPDA to halt in a final state, at the end of string (~~with an empty stack~~).
- The final contents of the stack are irrelevant.
- As was the case with nondeterministic automata, the string is accepted if *any of the computations* cause the NDPA to *halt in a final state*.
- The NPDA in Ex. 7.2 accepts the language

  $$L(M) = \{a^n b^n \mid n \geq 0\} \cup \{ a \}.$$

Example: The Language Accepted by a PDA

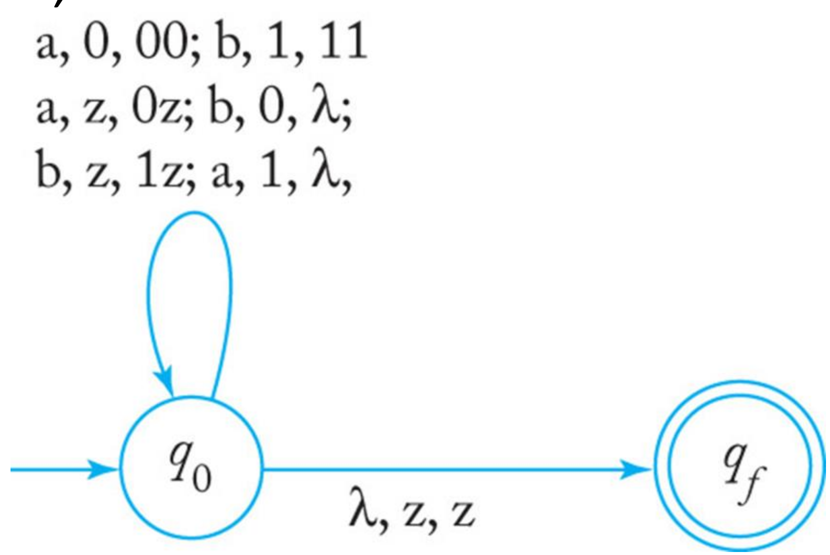
- Example 7.4: Construct an NPDA for the language

  $L = \{w \in \{a, b\}^* \mid n_a(w) = n_b(w)\}$

- Idea:

  Push a counter symbol, say 0, into the stack whenever an '*a*' is read, then pop one counter symbol from the stack whenever a '*b*' is read.

- Problem: stack underflow due to more popping '*b*' than pushing '*a*' if there is a prefix of *w* with more '*b*'s than '*a*'.
- Solution: Use a negative counter symbol, say 1, for counting the matching '*b*'s against '*a*'s.



- $w = baab$: $(q_0, baab, z)$

  $\vdash (q_0, aab, 1z) \vdash (q_0, ab, z)$

  $\vdash (q_0, b, 0z) \vdash (q_0, \lambda, z) \vdash (q_f, \lambda, z)$.

  So, *w* is accepted.

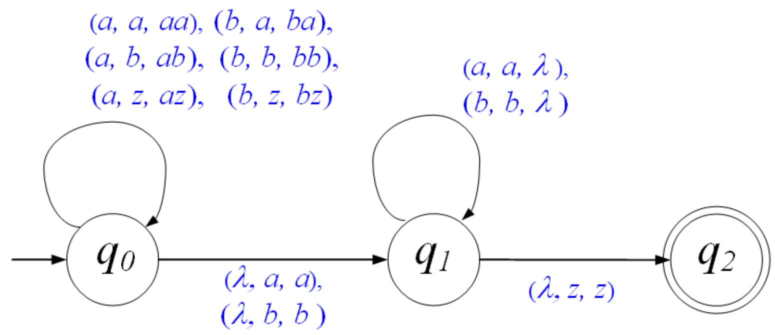
Example: Even length Palindrome(cont.)

- Example 7.5: Construct an NPDA for the language

$$L = \{ww^R \mid w \in \{a, b\}^+\}$$

- Idea: The symbols are retrieved from a stack in the reverse order of their insertion. When reading the 1st part of the string, push the symbols on the stack. For the 2nd part, compare the current input symbol with the top of the stack, continuing as long as they match. Since symbols are retrieved in the reverse order of their insertion, a complete match will be achieved iff the input is of the form ww^R.
- Difficulty: How to find the middle of the string?
- Solution: Guess the middle point and switch the state at that point.

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



Pushdown Automata (PDA) and Context-Free Languages (CFL)

1. **PDA for CFL (PDA ← CFG of CFL)**

- Theorem 7.1: $\forall$CFL L, $\exists$ NDPA M s.t. L(M) = L.

  For any CFL L, there exists a NDPA that accepts L.

- Assume that the language L is generated by a CFG in Greibach Normal Form, the constructive proof provides an algorithm that can be used to build the corresponding NPDA.

- The resulting NDPA simulates grammar derivations by keeping variables on the stack while making sure that the input symbol matches the terminal on the right side of the production.

Part 1: Construction of a NPDA from a Grammar in Greibach Normal Form

- The NPDA has Q = { q_0, q_1, q_F }, input alphabet equal to the grammar terminal symbols (Σ=T), and stack alphabet equal to the grammar variables (Γ=V).

  i.e. NPDA M = {Q, T, V$\cup${z}, δ, q_0, z, {q_f})
- The transition function contains the following:
  - A rule that pushes *S* on the stack and switches control to q_1 without consuming input: $\delta(q_0, \lambda, z) = (q_1, Sz)$ eq.7.1
  - For every production of the form A $\rightarrow$ *aX*,

    a transition rule $\delta(q_1, a, A) = (q_1, X)$ eq.7.2
  - A rule that switches the control unit to the final state when there is *no more input* and the *stack is empty*

    (or a stack start symbol).: $\delta(q_1,\lambda, z) = \{(q_f, z)\}$ eq.7.3

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

Part 2: Proof of L(M) = L(G)

$\rightarrow$) <u>Claim</u>: If $\forall w \in L(G)$, then $\forall w \in L(M)$, i.e. M accepts any $w \in L(G)$.

Consider the partial leftmost derivation

$$S^* \Rightarrow a_1 a_2 \ldots a_n A_1 A_2 \ldots A_m \Rightarrow a_1 a_2 \ldots a_n b B_1 \ldots B_k A_2 \ldots A_m.$$

If M is to simulate this derivation, then after reading $a_1 a_2 \cdots a_n$, the stack must contain $A_1 A_2 \cdots A_m$.

To take the next step in the derivation, G must have a production

$$A_1 \rightarrow b B_1 \cdots B_k.$$

The construction of M is such that, then M has a transition rule in which

$$(q_1, B_1 \cdots B_k) \in \delta\,(q_1, b, A_1),$$

so that the stack now contains $B_1 \cdots B_k A_2 \cdots A_m$ after having read $a_1 a_2 \cdots a_n b$.

A simple *induction argument on the number of steps* in the derivation shows that if $S \Rightarrow^* w$, then

$$(q_1, w, Sz) ^*\vdash (q_1, \lambda, z).$$

Using (7.1) and (7.3) we have

$$(q_0, w, z) \vdash^{7.1} (q_1, w, Sz) \vdash^* (q_1, \lambda, z) \vdash^{7.3} (q_f, \lambda, z).$$

so that $L\,(G) \subseteq L\,(M)$.

Part 2: Proof of L(M) = L(G)

$\leftarrow$) If $\forall w \in L(M)$, then $\forall w \in L(G)$,

i.e. CFG G s.t. $L(G)=L$ generates any w that is accepted by M.

Let $w \in L(M)$. Then, by definition, $(q_0, w, z) \vdash^* (q_f, \lambda, u)$.

But there is only one way[7.1] to get from q_0 to q_1 and only one way[7.3] from q_1 to q_f.

Therefore, we must have $(q_0, \lambda w, z) \vdash (q_1, w, Sz) \vdash^* (q_1, \lambda, z) \vdash (q_f, \lambda, z)$

Now let us write $w = a_1 a_2 a_3 \cdots a_n$. Then the first step in

$(q_1, a_1 a_2 a_3 \cdots a_n, Sz) \vdash^* (q_1, \lambda, \lambda z))$[7.4]

must be a rule of the form (7.2) to get

$(q_1, a_1 a_2 a_3 \cdots a_n, Sz) \vdash (q_1, a_2 a_3 \cdots a_n, u_1 z)$. Note: $\delta(q_1, a, A) = (q_1, X)$[7.2]

Then, the grammar has a rule of the form $S \rightarrow a_1 u_1$, so that $S \Rightarrow a_1 u_1$.

Repeating this, writing $u_1 = A u_2$, we have

$(q_1, a_2 a_3 \cdots a_n, A u_2 z) \vdash (q_1, a_3 \cdots a_n, u_3 u_2 z)$,

implying that $A \rightarrow a_2 u_3$ is in the grammar and that $S \Rightarrow^* a_1 a_2 u_3 u_2$.

(i.e. $S \Rightarrow a_1 u_1 (= a_1 A u_2 u_3 u_2) \Rightarrow a_1 a_2 u_3 u_2$).

This makes it clear at any point the stack contents (excluding z) are identical with the unmatched part of the sentential form, so that (7.4) implies $S^* \Rightarrow a_1 a_2 \cdots a_n \lambda$.

In consequence, $L(M) \subseteq L(G)$, if $\lambda \notin L$.

- If $\lambda \in L$, we add to the NDPA M the transition $\delta(q_0, \lambda, z) = \{(q_f, z)\}$ so the empty string λ is also accepted.

Example: Construction of a NPDA from a CFG

- Example 7.6: Construct a PDA that accepts the language by a grammar below.

$$S \to aSbb \mid a$$

1. Transform the grammar into Greibach Normal Form

$$S \to aSA \mid a\lambda, \qquad A \to bB, \qquad B \to b\lambda$$

2. The corresponding NPDA has $Q = \{ q_0, q_1, q_f \}$ with initial state q_0 and final state q_f.

- The start symbol S is placed on the stack with the transition

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

- The grammar productions are simulated with the transitions

$$\delta(q_1, a, S) = \{ (q_1, SA), (q_1, \lambda) \}$$
$$\delta(q_1, b, A) = \{ (q_1, B) \}$$
$$\delta(q_1, b, B) = \{ (q_1, \lambda) \}$$

- A final transition places the control unit in its final state when the stack is empty (i.e. with the stack start symbol).

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

Example: Construction of a NPDA from a CFG

- <u>Example 7.7</u>: Consider the grammar G

$$S \to aA, \quad A \to aABC \mid bB \mid a, \quad B \to b, \quad C \to c$$

1. The grammar is already GNF
2. The corresponding NPDA M has $Q = \{ q_0, q_1, q_f \}$ with initial state q_0 and final state q_f.

- The transition from the start state q_0 & start symbol S and the transition to the final state q_f with the empty stack (or with z)

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

- The grammar productions are simulated with the transitions

$$\delta(q_1, a, S) = \{ (q_1, A)\},$$
$$\delta(q_1, a, A) = \{ (q_1, ABC), (q_1, \lambda)\}, \quad \delta(q_1, b, A) = \{ (q_1, B)\}$$
$$\delta(q_1, b, B) = \{ (q_1, \lambda)\}, \qquad \delta(q_1, c, C) = \{ (q_1, \lambda)\}$$

e.g.) $w = aaabc \in L(M)$, i.e. $(q_0, w, z) \vdash^* (q_f, \lambda, z)$?

$$(q_0, aaabc, z) \vdash (q_1, aaabc, Sz) \vdash (q_1, aabc, Az) \vdash (q_1, abc, ABCz)$$
$$\vdash (q_1, bc, BCz) \vdash (q_1, c, Cz) \vdash (q_1, \lambda, z) \vdash (q_f, \lambda, z),$$

i.e. in G, $S \Rightarrow^* aaabc$: $S \Rightarrow aA \Rightarrow aaABC \Rightarrow aaaBC \Rightarrow aaabC \Rightarrow aaabc$.

PDA and CFL

2. CFG for PDA (PDA → CFG)

A derivation in Grammar G:

terminals variables

$$S \Rightarrow \cdots \Rightarrow abc\ldots ABC\ldots \Rightarrow \ldots \Rightarrow abc\ldots$$

Input processed Stack contents

in NPDA M

PDA and CFL

2. CFG for PDA (PDA → CFG)

<u>Assumption:</u>

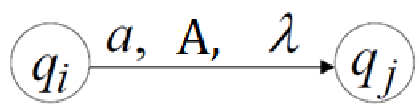
(1A) It has a *single final state* q_f

(1B) The stack is empty when it accepts the input.

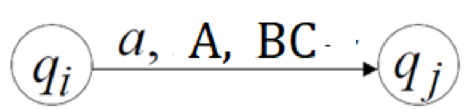
(2) With $a \in \Sigma \cup \{\lambda\}$, all transitions must have the form

$$\delta(q_i, a, A) = \{c_1, c_2, .., c_n\}$$

where $c_i=(q_j, \lambda)$[7.5]



or $c_i=(q_j, BC)$ [7.6].



i.e. each move either decreases (pop)

or increases (push) the stack content by a single symbol.

PDA and CFL

2. CFG for PDA (PDA → CFG)

Idea: Suppose that we can find a grammar

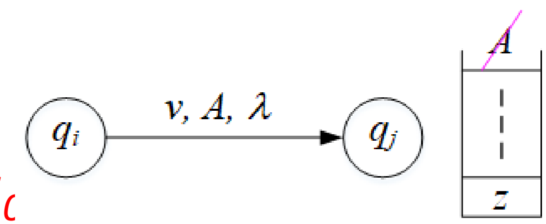
whose variables are of the form $(q_i A q_j)$ and

whose productions are s.t. $(q_i A q_j) \Rightarrow^* v$,

iff the NPDA removes A from the stack

while reading v and going from state q_i to state q_j.

(Removal of A from the stack means that
the stack doesn't change below A and
the symbol below A is at the top of the stack.)



If such a grammar is found and if we choose $(q$ … l,
then $(q_0 z q_f) \Rightarrow^* w$ iff

the NPDA removes z (i.e. empty stack) while reading w

and going from q_0 to q_f.

-- This is how NPDA accepts w!! Thus,

the language generated by the grammar

= the language accepted by NPDA.

PDA and CFL

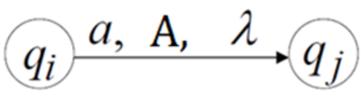
2. CFG for PDA (cont.)

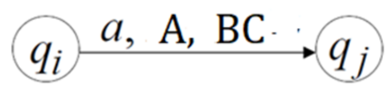
To construct such grammar,

let's examine the different types of transitions by the NPDA.

- $\delta(q_i, a, A) = (q_j, \lambda)$[7.5] involves an immediate erase of A.

  So, the grammar will have a corresponding production $(q_i A q_j) \rightarrow a$.





- $\delta(q_i, a, A) = (q_j, BC)$[7.6]) generate the set of rules $(q_i A q_k) \rightarrow a(q_j B q_l)(q_l C q_k)$,

  where q_k , q_l take on all possible states in Q

~~i.e. To erase A, first replace A with BC, while reading an *a* and going from state q_i to q_j. Subsequently, we go from q_j to q_l, erasing B, then from q_l to q_k, erasing C.~~

To replace A with BC, remove A while reading an *a* and going from state q_i to q_k.

➔ push C, going to q_k to q_l ➔ push B, going to q_l to q_j .

In the last step, perhaps too much rules were be added as there may be some states q_l that are unreachable from q_j while erasing B. It's true, but it doesn't affect the grammar. The resulting variables (q_jBq_l) are useless and don't affect the language accepted by the grammar. → They, (q_jBq_l)s, may be eliminated.

- Finally, as a start variable we take $(q_0 z q_f)$, where q_f is the single final state of the NPDA.

PDA and CFL

<u>Theorem 7.2:</u> If L = L(M) for some NPDA M, then L is a CFL.
Proof) Let M = (Q, Σ, Γ, δ, q_0, z, $\{q_f\}$), satisfying Assumptions 1AB & 2.
Construct the grammar $G = (V, T, S, P)$ with
$T = \Sigma$ and V consisting of elements of the form $(q_i c q_j)$, $c \in \Gamma$.
Let's show that the grammar so obtained is
s.t. $\forall q_i, q_j \in Q$, $A \in \Gamma$, $X \in \Gamma^*$, $u,v \in \Sigma^*$,

$$(q_i, uv, AX) \vdash_M^* (q_j, v, X)^{7.7}$$

implies $(q_i A q_j) \Rightarrow_G^* u$, and vice versa.

The 1st part is to show that: whenever the NPDA is s.t. 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. -- It's easy to see since the grammar was explicitly constructed to do this.

Let's use an induction on the number of moves to make it precise.

Proof (cont.) Let's use an induction on the number of moves to make it precise.

1. (1a) Consider a *single step* in the *derivation* such as

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

Using the corresponding transition for the NPDA, $\delta(q_i, a, A)=\{(q_j, BC), \ldots\}$ [7.8], we see that the A can be removed from the stack, BC put on, reading a, with the control unit going from state q_i to q_j.

(1b) Similarly, if $(q_i A q_j) \Rightarrow_G a$,
then there must be a corresponding transition $\delta(q_i, a, A) = \{(q_j, \lambda)\}$ [7.10] .

2. From this, we see that the *sentential forms* derived from $(q_i A q_j)$ define a sequence of possible configurations of the NPDA by which eq.(7.7) can be achieved.

Note that $(q_i A q_j) \Rightarrow a(q_j B q_l)(q_l C q_k)$ might be possible for some $(q_j B q_l)(q_l C q_k)$ for which there is no corresponding transition of the form (7.8) & (7.10).

In such case, at least one of the variables on the right will be useless.

For all sentential forms leading to a terminal string, the given argument holds.

3. If we apply the conclusion to $(q_0, w, z) \vdash_M^* (q_f, \lambda, \lambda)$,

$$(q_0, w, z) \vdash_M^* (q_f, \lambda, \lambda), \text{ iff } (q_0 z q_f) \Rightarrow^* w.$$

Consequently, L(M) = L(G).

i.e. If L = L(M) for some NPDA M, then $\exists$ CFG G, L(G) = L, so L is CFL.

Example: CFG for PDA

Example 7.8: A NPDA with the transitions:

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

Since the assumption 2 is not satisfied, let's introduce a new state q_3 and an intermediate step in which we first remove A from the stack then replace it in the next move.

$\delta(q_0,a,z)=\{(q_0, Az)\}^{(1)}$, $\delta(q_3,\lambda,z)=\{(q_0,Az)\}^{(2)}$, $\delta(q_0,a,A)=\{(q_3,\lambda)\}^{(3)}$,

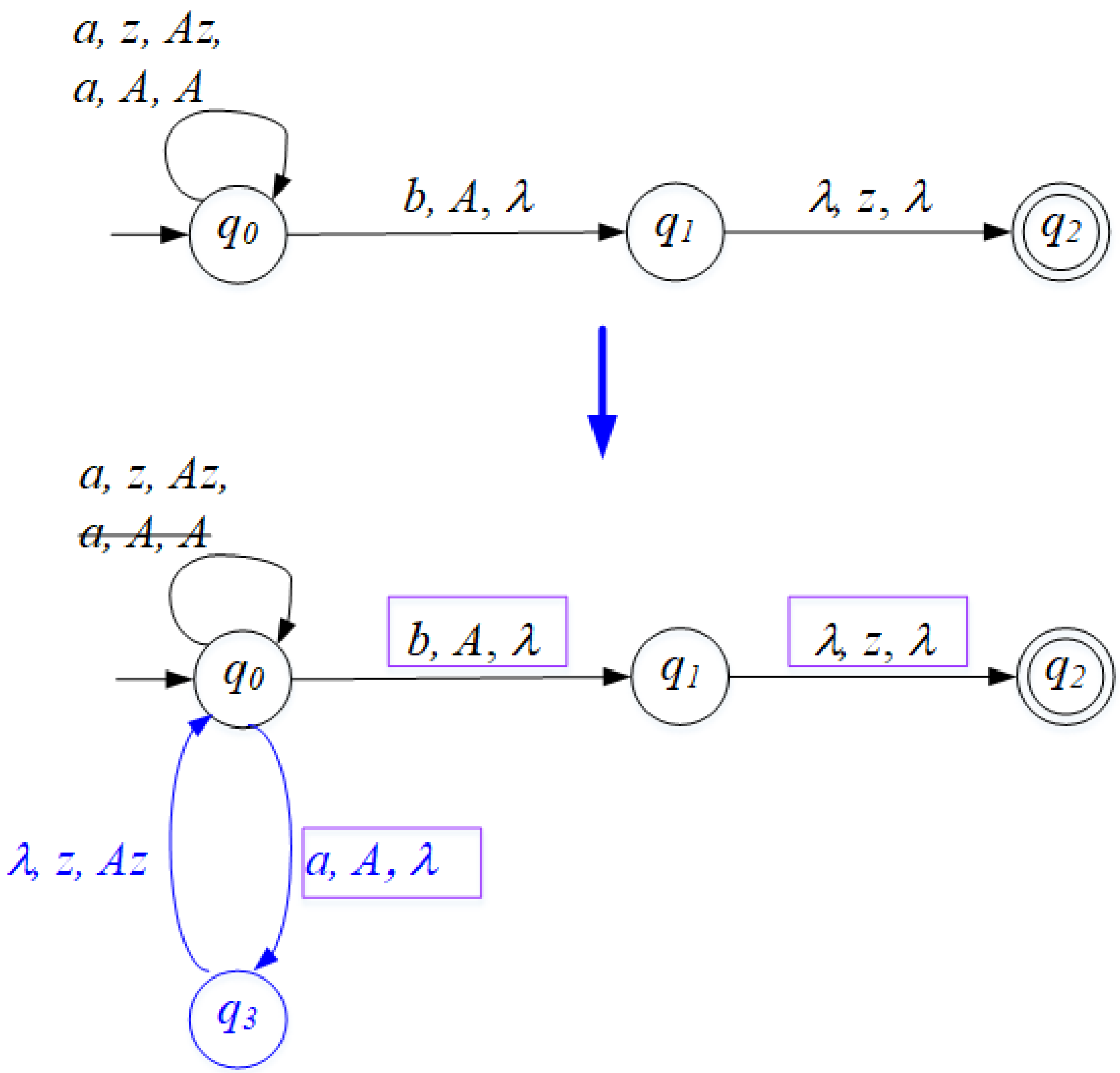
$\delta(q_0,b,A)=\{(q_1,\lambda)\}^{(4)}$, $\delta(q_0,\lambda,z)=\{(q_2,\lambda)\}^{(5)}$.

The transitions (3) – (5) with λ are of the form $c_i = (q_j, \lambda)$, they yield the corresponding productions in G: $(q_0Aq_3)^{(3)} \to a$, $(q_0Aq_1)^{(4)} \to b$, $(q_0zq_2)^{(5)} \to \lambda$.

Reminder: $\delta(q_i, a, A) = (q_j, BC)^{7.6}$ generates rules $(q_iAq_k) \to a(q_jBq_l)(q_lCq_k)$.

From the transitions (1) $\delta(q_0,a,z)=\{(q_0, Az)\}^{(1)}$, we get the set of productions in G

- $(q_0zq_0) \to a(q_0Aq_0)(q_0zq_0) \mid a(q_0Aq_1)(q_1zq_0) \mid a(q_0Aq_2)(q_2zq_0) \mid a(q_0Aq_3)(q_3zq_0)$,
- $(q_0zq_1) \to a(q_0Aq_0)(q_0zq_1) \mid a(q_0Aq_1)(q_1zq_1) \mid a(q_0Aq_2)(q_2zq_1) \mid a(q_0Aq_3)(q_3zq_1)$,
- $(q_0zq_2) \to a(q_0Aq_0)(q_0zq_2) \mid a(q_0Aq_1)(q_1zq_2) \mid a(q_0Aq_2)(q_2zq_2) \mid a(q_0Aq_3)(q_3zq_2)$,
- $(q_0zq_3) \to a(q_0Aq_0)(q_0zq_3) \mid a(q_0Aq_1)(q_1zq_3) \mid a(q_0Aq_2)(q_2zq_3) \mid a(q_0Aq_3)(q_3zq_3)$.


a, z, Az,
a, A, A
b, A, λ
λ, z, λ
q0
q1
q2
a, z, Az,
~~a, A, A~~
b, A, λ
λ, z, λ
q0
q1
q2
λ, z, Az
a, A, λ
q3

Example: CFG for PDA

Example 7.8 (cont.): A NPDA with the transitions:

Reminder: $\delta(q_i, a, A) = (q_j, BC)$[7.6]) generates $(q_i A q_k) \rightarrow a(q_j B q_l)(q_l C q_k)$.

From the transitions (2) $\delta(q_3,\lambda,z)=\{(q_0,Az)\}^{(2)}$, we get the set of productions in G

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

Simplification: A variable that doesn't occur on the left side of any production must be useless → eliminate them: $(q_0 A q_0)$, $(q_0 A q_2)$.

From the transition of NPDA,

there is no path $q_1 \rightsquigarrow q_0$, $q_1 \rightsquigarrow q_1$, $q_1 \rightsquigarrow q_3$, $q_2 \rightsquigarrow q_2$,

so their associated variables are also useless.

➔ Eliminate those useless productions.

Example 7.8 (cont.): A NPDA with the transitions:

Useless Variables: (q_0Aq_0), (q_0Aq_2).

Useless productions associated with $q_1 \rightsquigarrow q_0,\ q_1 \rightsquigarrow q_1,\ q_1 \rightsquigarrow q_3,\ q_2 \rightsquigarrow q_2,$

1) $(q_0Aq_3) \to a,$

2) $(q_0Aq_1) \to b,$

3) $(q_1zq_2) \to \lambda.$

4) $(q_0zq_0) \to a\cancel{(q_0Aq_0})(q_0zq_0) \mid a(q_0Aq_1)(\cancel{q_1zq_0}) \mid a(\cancel{q_0Aq_2})(q_2zq_0) \mid a(q_0Aq_3)(q_3zq_0)$,

5) $(q_0zq_1) \to a(\cancel{q_0Aq_0})(q_0zq_1) \mid a(q_0Aq_1)(\cancel{q_1zq_1}) \mid a(\cancel{q_0Aq_2})(q_2zq_1) \mid a(q_0Aq_3)(q_3zq_1)$,

6) $(q_0zq_2) \to a(\cancel{q_0Aq_0})(q_0zq_2) \mid a(q_0Aq_1)(q_1zq_2) \mid a(\cancel{q_0Aq_2})(\cancel{q_2zq_2}) \mid a(q_0Aq_3)(q_3zq_2)$,

7) $(q_0zq_3) \to a(\cancel{q_0Aq_0})(q_0zq_3) \mid a(q_0Aq_1)(\cancel{q_1zq_3}) \mid a(\cancel{q_0Aq_2})(q_2zq_3) \mid a(q_0Aq_3)(q_3zq_3)$.

8) $(q_3zq_0) \to \cancel{(q_0Aq_0)}(q_0zq_0) \mid (q_0Aq_1)(\cancel{q_1zq_0}) \mid (\cancel{q_0Aq_2})(q_2zq_0) \mid (q_0Aq_3)(q_3zq_0)$,

9) $(q_3zq_1) \to \cancel{(q_0Aq_0)}(q_0zq_1) \mid (q_0Aq_1)(\cancel{q_1zq_1}) \mid (\cancel{q_0Aq_2})(q_2zq_1) \mid (q_0Aq_3)(q_3zq_1)$,

10) $(q_3zq_2) \to \cancel{(q_0Aq_0)}(q_0zq_2) \mid (q_0Aq_1)(q_1zq_2) \mid (\cancel{q_0Aq_2})(\cancel{q_2zq_2}) \mid (q_0Aq_3)(q_3zq_2)$,

11) $(q_3zq_3) \to \cancel{(q_0Aq_0)}(q_0zq_3) \mid (q_0Aq_1)(\cancel{q_1zq_3}) \mid (\cancel{q_0Aq_2})(q_2zq_3) \mid (q_0Aq_3)(q_3zq_3)$.

with the start variable $\mathbf{(q_0zq_2)}$

Example 7.8 (cont.): A NPDA with the transitions:

i.e.

1) $(q_0 A q_3) \rightarrow a,$

2) $(q_0 A q_1) \rightarrow b,$

3) $(q_1 z q_2) \rightarrow \lambda.$

4) $(q_0 z q_0) \rightarrow \cancel{a(q_0 A q_0)(q_0 z q_0)} \mid \cancel{a(q_0 A q_1)(q_1 z q_0)} \mid \cancel{a(q_0 A q_2)(q_2 z q_0)} \mid a(q_0 A q_3)(q_3 z q_0)$,

5) $(q_0 z q_1) \rightarrow \cancel{a(q_0 A q_0)(q_0 z q_1)} \mid \cancel{a(q_0 A q_1)(q_1 z q_1)} \mid \cancel{a(q_0 A q_2)(q_2 z q_1)} \mid a(q_0 A q_3)(q_3 z q_1)$,

6) $(q_0 z q_2) \rightarrow \cancel{a(q_0 A q_0)(q_0 z q_2)} \mid a(q_0 A q_1)(q_1 z q_2) \mid \cancel{a(q_0 A q_2)(q_2 z q_2)} \mid a(q_0 A q_3)(q_3 z q_2)$,

7) $(q_0 z q_3) \rightarrow \cancel{a(q_0 A q_0)(q_0 z q_3)} \mid \cancel{a(q_0 A q_1)(q_1 z q_3)} \mid \cancel{a(q_0 A q_2)(q_2 z q_3)} \mid a(q_0 A q_3)(q_3 z q_3).$

8) $(q_3 z q_0) \rightarrow \cancel{(q_0 A q_0)(q_0 z q_0)} \mid \cancel{(q_0 A q_1)(q_1 z q_0)} \mid \cancel{(q_0 A q_2)(q_2 z q_0)} \mid (q_0 A q_3)(q_3 z q_0)$,

9) $(q_3 z q_1) \rightarrow \cancel{(q_0 A q_0)(q_0 z q_1)} \mid \cancel{(q_0 A q_1)(q_1 z q_1)} \mid \cancel{(q_0 A q_2)(q_2 z q_1)} \mid (q_0 A q_3)(q_3 z q_1)$,

10) $(q_3 z q_2) \rightarrow \cancel{(q_0 A q_0)(q_0 z q_2)} \mid (q_0 A q_1)(q_1 z q_2) \mid \cancel{(q_0 A q_2)(q_2 z q_2)} \mid (q_0 A q_3)(q_3 z q_2)$,

11) $(q_3 z q_3) \rightarrow \cancel{(q_0 A q_0)(q_0 z q_3)} \mid \cancel{(q_0 A q_1)(q_1 z q_3)} \mid \cancel{(q_0 A q_2)(q_2 z q_3)} \mid (q_0 A q_3)(q_3 z q_3)$.

with the start variable $\boldsymbol{(q_0 z q_2)}$

Example 7.8 (cont.):

More elimination of the useless variables and their associated productions:

Note that a variable is *useless* if:

- No terminal strings can be derived from the variable:
- The variable symbol can not be reached from S.

More useless variables: A, B, D, E, F, K.

1) $(q_0Aq_3) \to a,$

2) $(q_0Aq_1) \to b,$

3) $(q_1zq_2) \to \lambda.$

4) $\cancel{(q_0zq_0) \to a(q_0Aq_3)(q_3zq_0)},$

5) $\cancel{(q_0zq_1) \to a(q_0Aq_3)(q_3zq_1)},$

6) $(q_0zq_2) \to a(q_0Aq_1)(q_1zq_2) \,|\, a(q_0Aq_3)(q_3zq_2),$

7) $\cancel{(q_0zq_3) \to a(q_0Aq_3)(q_3zq_3)}.$

8) $\cancel{(q_3zq_0) \to (q_0Aq_3)(q_3zq_0)},$

9) $\cancel{(q_3zq_1) \to (q_0Aq_3)(q_3zq_1)},$

10) $(q_3zq_2) \to (q_0Aq_1)(q_1zq_2) \,|\, (q_0Aq_3)(q_3zq_2),$

11) $\cancel{(q_3zq_3) \to (q_0Aq_3)(q_3zq_3)}.$

with the start variable $\boldsymbol{(q_0zq_2)}$

Rename the variables.

$X \to a$

$Y \to b$

$Z \to \lambda$

$\cancel{A \to aXE}$

$\cancel{B \to aXF}$

$S \to aYZ \,|\, aXH$

$\cancel{D \to aAK}$

$\cancel{E \to AE}$

$\cancel{F \to AF}$

$H \to YZ \,|\, XH$

$\cancel{K \to XK}$

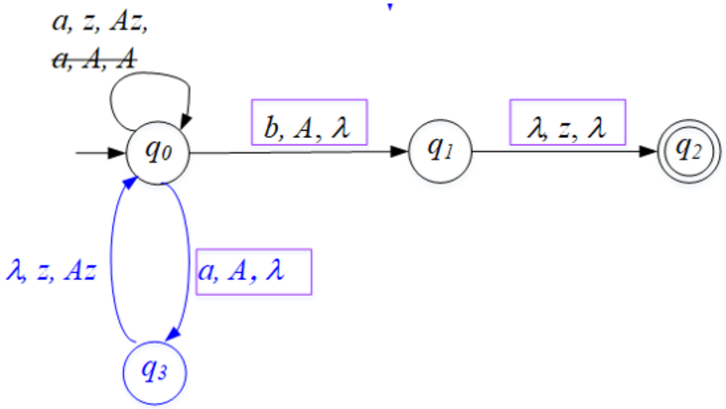
$L(G) = \{aa^*b\}$

Example: Generation of a CFG for PDA

- Example 7.9: The string, w=*aab*, is accepted by the NPDA in Ex.7.8, with successive configurations.

$(q_0,aab,z) \vdash (q_0,ab,Az) \vdash (q_3,b,z) \vdash (q_0,b,Az) \vdash (q_1,\lambda,z) \vdash (q_2,\lambda,\lambda).$

The corresponding derivation with G is:



$$
\begin{aligned}
(q_0 z q_2) &\Rightarrow^{6)} a(q_0 A q_3)(q_3 z q_2) \\
&\Rightarrow^{1)} aa(q_3 z q_2) \\
&\Rightarrow^{10)} aa(q_0 A q_1)(q_1 z q_2) \\
&\Rightarrow^{2)} aab(q_1 z q_2) \\
&\Rightarrow^{3)} aab\lambda = aab.
\end{aligned}
$$

So, $(q_0,aab,z) \vdash^* (q_2,\lambda,\lambda)$ in NPDA ➔ $(q_0zq_2) \Rightarrow^* aab$ in CFG.

i.e. $(q_0,w, z) \vdash_M^* (q_f,\lambda,\lambda)$, iff $(q_0zq_f) \Rightarrow^* w$.

Summary: Generation of a CFG for PDA

A derivation in Grammar G:

$$S \Rightarrow \cdots \Rightarrow \overbrace{abc\ldots}^{\text{terminals}}\overbrace{ABC\ldots}^{\text{variables}} \Rightarrow \ldots \Rightarrow abc\ldots$$

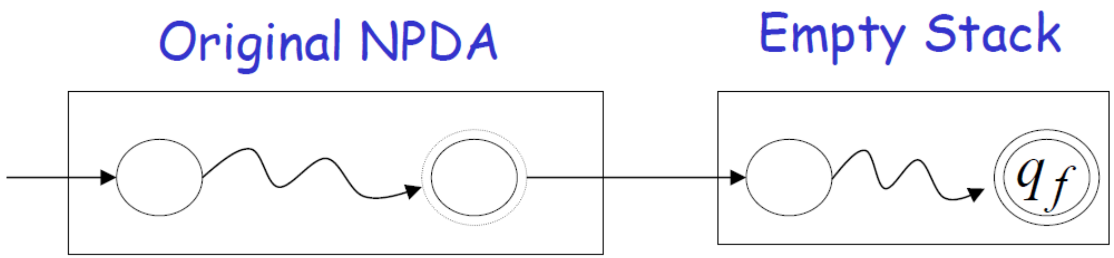
Input processed

Stack contents

in NPDA M

Needs Modification:

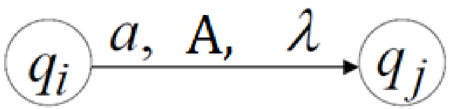
- 1. NPDA
  - It has a single final state q_f.
  - It empties the stack when it accepts the input.

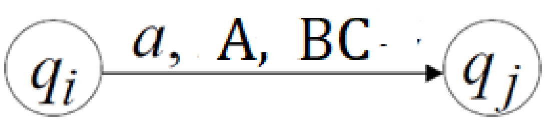


- 2. All transitions of NPDA will have form:

$\delta(q_i, a, A) \rightarrow (q_j, \lambda)$: pop A

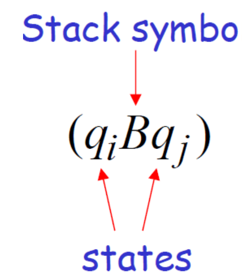
or $\delta(q_i, a, A) \rightarrow (q_j, BC)$: replace A with BC where A, B, C$\in\Gamma$

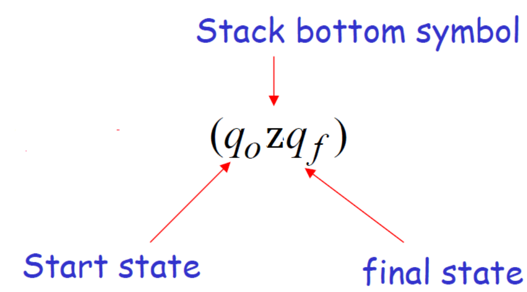




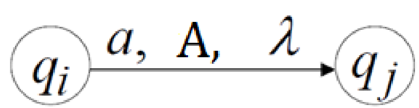
Grammar Construction from NPDA

- In CFG G:
  - Variables: Name them using
    a pair of states and a stack symbol
  - Terminals: $T = \Sigma$, Input symbols of NPDA
  - Start variable: $(q_0 z q_f)$



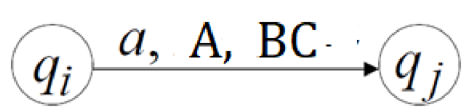


(1) For each transition,



Add production: $(q_i A q_j) \rightarrow a$

(2) For each transition,



Add production

$$(q_i A q_k) \rightarrow a\ (q_j B q_l)\ (q_l C q_k) \quad \forall\ q_k, q_l.$$

Grammar Construction from NPDA

- In general, in the grammar:

$$(q_0 z q_f) \Rightarrow^* w \quad \text{iff} \quad w \in L(NPDA)$$

- By construction of grammar,

$$(q_i A q_j) \Rightarrow^* w$$

if and only if

In the transition going from q_i to q_j, $q_i \rightarrow q_j$

the stack doesn't change below A

and A is removed from stack.

Deterministic Pushdown Automaton (DPDA)

- A *Deterministic PushDown Automaton* (DPDA) never has a choice in its move.
- Definition 7.3: A *Deterministic Pushdown Automaton* (DPDA), $M=(Q, \Sigma, \Gamma, \delta, q_0, z, F)$ is a PDA subject to the restrictions on DPDA transitions that $\forall q \in Q$, $a \in \Sigma \cup \{\lambda\}$ and $b \in \Gamma$,

  1. $\delta(q, a, b)$ contains *at most* one element.
  2. If $\delta(q, \lambda, b) \neq \emptyset$, then $\delta(q, c, b)$ must be empty for $\forall c \in \Sigma$.

  i.e. it means

  (1). for any input symbol and any stack top, *at most one move* (state, stack top) can be made.

  (2). If a λ-move is possible for some configuration, there can be *no input-consuming* alternative *transitions*.
- Unlike the case for NFA & DFA, a λ-transition does not necessarily mean the automaton is nondeterministic.

Example: Deterministic PDA (DPDA)

- Example 7.10: Construct a DPDA to accept the language

$$L = \{ a^n b^n \mid n \geq 0 \}.$$

- The DPDA has Q = { q_0, q_1, q_2 }, input alphabet {a, b}, stack alphabet {0, 1}, z = 0, and q_0 as its initial and final state.
- The transition rules are

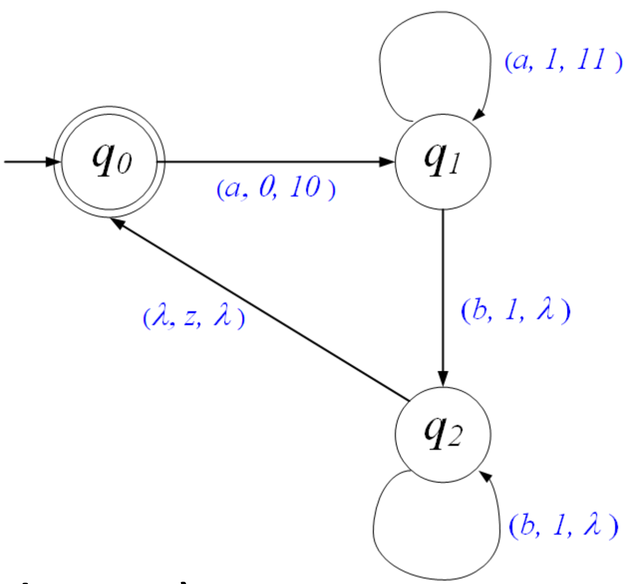
$\delta(q_0, a, 0) = \{ (q_1, 10) \}$

$\delta(q_1, a, 1) = \{ (q_1, 11) \}$

$\delta(q_1, b, 1) = \{ (q_2, \lambda) \}$

$\delta(q_2, b, 1) = \{ (q_2, \lambda) \}$

$\delta(q_2, \lambda, 0) = \{ (q_0, \lambda) \}$



Cf) Example 7.2B of NPDA

Cf) Example 7.5 (even length palindrome)

Deterministic Context-Free Languages

- Definition 7.4: A language L is *Deterministic CFL* if and only if there exists a DPDA M to accept *L, i.e. L = L(M).*
- Sample Deterministic CFL:

  $\{ a^n b^n \mid n \geq 0 \}$

  $\{ wxw^R \mid w \in \{a, b\}^* \}$: odd length palindrome
- $\{ ww^R \mid w \in \{a, b\}^* \}$: even length palindrome ?
- Deterministic and Nondeterministic Pushdown Automata are *not equivalent*: There are some Context-Free Languages for which no DPDA can be built, i.e. there are CFL that are not deterministic . e.g.) even length palindrome.
- DCFL $\subset$ CFK

Deterministic Context-Free Languages

- <u>Example 7.11</u>: Let $L_1=\{ a^nb^n \mid n \geq 0 \}$, $L_2=\{ a^nb^{2n} \mid n \geq 0 \}$.

L_1 is a CFL. With the modification of the argument of L_1,
let's show that L_2 is a CFL.

L = $L_1 \cup L_2$, a union of two CFLs is a CFL as well.

To show it, let $G_1=(V_1,T,S_1,P_1)$ and $G_2=(V_2,T,S_2,P_2)$ be CFG
s.t. $L_1=L(G_1)$ and $L_2=L(G_2)$. Assume that $V_1 \cap V_2=\emptyset$ and $S \notin V_1 \cup V_2$.

Let's combine G_1 *and* G_2, $G = (V_1 \cup V_2 \cup \{S\}, T, S, P)$ where

$$P = P_1 \cup P_2 \cup \{S \rightarrow S_1 \mid S_2\},$$

to generate $L_1 \cup L_2$, *i.e.* $L= L_1 \cup L_2=L(G_1) \cup L(G_2) =L(G)$.

L = L(G) is a CFL. Proof in Chap. 8. But, L is *not Deterministic CFL*.

MORE proof !!!!!!!