Chapter 1
Formal Pushdown Automata
Formal Definition and View

Lecture Preprint *Formal Pushdown Automata* on the 28th April 2009

Dušan Kolář, Faculty of Information Technology, Brno University of Technology

1.1

Overview

Aim of the Lecture

1. Define pushdown automaton in a formal way
2. Top-down and bottom-up analysis of context-free languages
3. One-turn pushdown automata
4. Atomic pushdown automata

1.2

Contents

1.3

1 Introduction

What is a *pushdown*?

Merriam-Webster
A store of data (as in a computer) from which the most recently stored item must be the first retrieved—called also *pushdown list* or *pushdown stack*.

We stick to this definition!

1.4

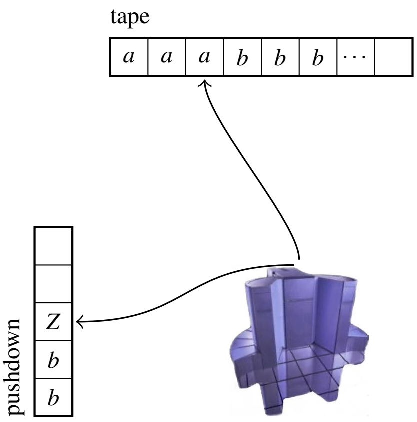
Why do we need them?

- Finite automata handle regular languages only
- Syntax of programming languages is context-free
  - Program text is of arbitrary length
  - Repeated nested bracket-like structures(e.g. arithmetic expressions)
- Pushdown automata offer infinite memory(even if LIFO structure)

1.5

2 Formal Definition of PA

Informal View



1.6

2.1 Definition

Definition of Pushdown Automaton

Formal Definition of Pushdown Automaton
A pushdown automaton is a seven-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, where

- Q — finite *set of states*
- Σ — finite *input alphabet*
- Γ — finite *alphabet of pushdown symbols*
- δ — mapping $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to 2^{Q \times \Gamma^*}$ *transition function*
- $q_0 \in Q$ — *starting/initial state*
- $Z_0 \in \Gamma$ — *start symbol* on the pushdown
- $F \subseteq Q$ — *set of final states*

1.7

Configuration

Definition of Configuration
A *configuration* of PA M is a triple $(q, w, \alpha) \in Q \times \Sigma^* \times \Gamma^*$, where

- q — current state of the finite control
- w — unused input; the first symbol of w is under the reading head; if $w = \varepsilon$, then it is assumed that all the input has been read
- α — contents of the pushdown; the leftmost symbol is the topmost symbol; if $\alpha = \varepsilon$, then the pushdown is assumed to be empty

1.8

Move / Transition

Move/Transition Definition
A *move/transition* by M is represented by the binary relation $\vdash_M$ (or $\vdash$ if M is understood) on configurations. We write

$$(q, aw, Z\alpha) \vdash (q', w, \gamma\alpha)$$

if $(q', \gamma) \in \delta(q, a, Z)$ for any $q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, $w \in \Sigma^*$, $Z \in \Gamma$, and $\alpha \in \Gamma^*$.

We can define relations $\vdash^i, i \geq 0, \vdash^*$, and $\vdash^+$ in the traditional way. $\vdash^*$ and $\vdash^+$ stand for reflexive-transitive and transitive closures of $\vdash$, respectively.

1.9

Final Configuration

- Initial configuration: (q_0, w, Z_0), $w \in \Sigma^*$
- *Note: from the definition, it follows that there is no move possible if the pushdown is empty.*
- Final configuration, Ξ, $\Xi \in \{1, 2, 3\}$:
  1. Empty pushdown: $(q, \varepsilon, \varepsilon)$, $q \in Q$
  2. Final state: $(q_F, \varepsilon, \alpha)$, $q_F \in F$, $\alpha \in \Gamma^*$
  3. Final state and empty pushdown: $(q_F, \varepsilon, \varepsilon)$, $q_F \in F$

1.10

Accepted Language

- A word w, $w \in \Sigma^*$, is accepted by M if it makes transitions $(q_0, w, Z_0) \vdash^* \Xi$, where $\Xi \in \{1, 2, 3\}$ is the selected final configuration.

Language accepted by M
$L(M) = \{w \mid (q_0, w, Z_0) \vdash^* \Xi\}$, where $\Xi \in \{1, 2, 3\}$ is some (but fixed) of the final configurations.

1.11

2.2 Examples

Example 1 ($a^n b^n$, $n \geq 0$) — Definition

$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$
$Q = \{q_0\}$ $\Sigma = \{a, b\}$ $\Gamma = \{a, Z_0\}$ $F = \{q_0\}$
δ: $\delta(q_0, a, Z_0) \to \{(q_0, Z_0 a)\}$
$\delta(q_0, b, a) \to \{(q_0, \varepsilon)\}$
$\delta(q_0, \varepsilon, Z_0) \to \{(q_0, \varepsilon)\}$

1.12

Example 1 ($a^n b^n$, $n \geq 0$) — Acceptance

$(q_0, aaabbb, Z_0)$	// $\delta(q_0, a, Z_0) \to (q_0, Z_0 a)$
$\vdash (q_0, aabbb, Z_0 a)$	// $\delta(q_0, a, Z_0) \to (q_0, Z_0 a)$
$\vdash (q_0, abbb, Z_0 aa)$	// $\delta(q_0, a, Z_0) \to (q_0, Z_0 a)$
$\vdash (q_0, bbb, Z_0 aaa)$	// $\delta(q_0, \varepsilon, Z_0) \to (q_0, \varepsilon)$
$\vdash (q_0, bbb, aaa)$	// $\delta(q_0, b, a) \to (q_0, \varepsilon)$
$\vdash (q_0, bb, aa)$	// $\delta(q_0, b, a) \to (q_0, \varepsilon)$
$\vdash (q_0, b, a)$	// $\delta(q_0, b, a) \to (q_0, \varepsilon)$
$\vdash (q_0, \varepsilon, \varepsilon)$	

1.13

Example 1 ($a^n b^n$, $n \geq 0$) — Failure

$(q_0, aabbb, Z_0)$	// $\delta(q_0, a, Z_0) \to (q_0, Z_0 a)$
$\vdash (q_0, abbb, Z_0 a)$	// $\delta(q_0, a, Z_0) \to (q_0, Z_0 a)$
$\vdash (q_0, bbb, Z_0 aa)$	// $\delta(q_0, \varepsilon, Z_0) \to (q_0, \varepsilon)$
$\vdash (q_0, bbb, aa)$	// $\delta(q_0, b, a) \to (q_0, \varepsilon)$
$\vdash (q_0, bb, a)$	// $\delta(q_0, b, a) \to (q_0, \varepsilon)$
$\vdash (q_0, b, \varepsilon)$	// no rule to be applied

1.14

Example 2 ($ww^R, w \in \{a,b\}^+$) — Definition

$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$
$Q = \{q_0, q_1, q_F\}$ $\Sigma = \{a, b\}$ $\Gamma = \{a, b, Z_0\}$ $F = \{q_F\}$
δ: $\delta(q_0, a, Z_0) \to \{(q_0, aZ_0)\}$
$\delta(q_0, b, Z_0) \to \{(q_0, bZ_0)\}$
$\delta(q_0, a, a) \to \{(q_0, aa), (q_1, \varepsilon)\}$
$\delta(q_0, a, b) \to \{(q_0, ab)\}$
$\delta(q_0, b, a) \to \{(q_0, ba)\}$
$\delta(q_0, b, b) \to \{(q_0, bb), (q_1, \varepsilon)\}$
$\delta(q_1, b, b) \to \{(q_1, \varepsilon)\}$
$\delta(q_1, a, a) \to \{(q_1, \varepsilon)\}$
$\delta(q_1, \varepsilon, Z_0) \to \{(q_F, \varepsilon)\}$

1.15

Example 2 ($ww^R, w \in \{a,b\}^+$) — Acceptance

$(q_0, abbaabba, Z_0)$ // $\delta(q_0, a, Z_0) \to (q_0, aZ_0)$
$\vdash (q_0, bbaabba, aZ_0)$ // $\delta(q_0, b, a) \to (q_0, ba)$
$\vdash (q_0, baabba, baZ_0)$ // $\delta(q_0, b, b) \to (q_0, bb)$
$\vdash (q_0, aabba, bbaZ_0)$ // $\delta(q_0, a, b) \to (q_0, ab)$
$\vdash (q_0, abba, abbaZ_0)$ // $\delta(q_0, a, a) \to (q_1, \varepsilon)$
$\vdash (q_1, bba, bbaZ_0)$ // $\delta(q_1, b, b) \to (q_1, \varepsilon)$
$\vdash (q_1, ba, baZ_0)$ // $\delta(q_1, b, b) \to (q_1, \varepsilon)$
$\vdash (q_1, a, aZ_0)$ // $\delta(q_1, a, a) \to (q_1, \varepsilon)$
$\vdash (q_1, \varepsilon, Z_0)$ // $\delta(q_1, \varepsilon, Z_0) \to (q_F, \varepsilon)$
$\vdash (q_F, \varepsilon, \varepsilon)$

1.16

Example 2 ($ww^R, w \in \{a,b\}^+$) — Failure

$(q_0, abbaaaba, Z_0)$ // $\delta(q_0, a, Z_0) \to (q_0, aZ_0)$
$\vdash (q_0, bbaaaba, aZ_0)$ // $\delta(q_0, b, a) \to (q_0, ba)$
$\vdash (q_0, baaaba, baZ_0)$ // $\delta(q_0, b, b) \to (q_0, bb)$
$\vdash (q_0, aaaba, bbaZ_0)$ // $\delta(q_0, a, b) \to (q_0, ab)$
$\vdash (q_0, aaba, abbaZ_0)$ // $\delta(q_0, a, a) \to (q_1, \varepsilon)$
$\vdash (q_1, aba, bbaZ_0)$ // no rule to be applied

1.17

3 Syntax Analysis with PA

Syntax Analysis—General View

How to construct a PA?

- Directly
- From a context-free grammar

Syntax Analysis
Virtual reconstruction of a particular derivation tree for the given grammar and sentence of the language.

Note: to make difference between end of the input tape and ignoring the input symbol we will use \$ *to represent end of the input tape (empty input) and* ε *to denote that we do not read the symbol under the reading head and the reading head is not moved.* 1.18

3.1 Top-Down Analysis

Top-Down Analysis
Virtual reconstruction of the derivation tree starts from the starting non-terminal of the respective context-free grammar.

Construction
Let $G = (N, T, P, S)$ be a context-free grammar then construct $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ this way:

- $Q = \{q_0, q, q_F\}$

- $\Sigma = T \cup \{\$\}$
- $\Gamma = \{Z_0\} \cup T \cup N, \quad \{Z_0, \$\} \cap (T \cup N) = \emptyset$
- $F = \{q_F\}$
- For all $A \to \alpha \in P, A \in N, \alpha \in (N \cup T)^*$
  add $(q, \varepsilon, A) \to \{(q, \alpha)\}$ to δ
  For all $a \in T$ add $(q, a, a) \to \{(q, \varepsilon)\}$ to δ
  Add $(q_0, \varepsilon, Z_0) \to \{(q, SZ_0)\}$ to δ
  Add $(q, \$, Z_0) \to \{(q_F, \varepsilon)\}$ to δ

1.19

Example—Construction

CF Grammar
Let $G = (N, T, P, S)$, where: $N = \{S, E\}$, $T = \{\ (,), +, i\ \}$, $P = \{\ S \to SS \mid E,\ E \to E + E \mid i \mid (E)\ \}$.

PA Construction
Let $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, where:

- $Q = \{q_0, q, q_F\}$, $F = \{q_F\}$
- $\Sigma = T \cup \{\$\} = \{\ (,), +, i, \$\ \}$
- $\Gamma = \{Z_0\} \cup T \cup N = \{S, E, (,), +, i, Z_0\}$
- $\delta(q, \$, Z_0) \to \{(q_F, \varepsilon)\}$
  $\delta(q_0, \varepsilon, Z_0) \to \{(q, SZ_0)\}$
  $\delta(q, \varepsilon, S) \to \{(q, SS), (q, E)\}$
  $\delta(q, \varepsilon, E) \to \{(q, E+E), (q, i), (q, (E))\}$
  $\delta(q, (, () \to \{(q, \varepsilon)\}$
  $\delta(q,),)) \to \{(q, \varepsilon)\}$
  $\delta(q, +, +) \to \{(q, \varepsilon)\}$
  $\delta(q, i, i) \to \{(q, \varepsilon)\}$

1.20

Example—Analysis

$(q_0, i+i(i), Z_0)$	// $\delta(q_0, \varepsilon, Z_0) \to (q, SZ_0)$
$\vdash (q, i+i(i), SZ_0)$	// $\delta(q, \varepsilon, S) \to (q, SS)$
$\vdash (q, i+i(i), SSZ_0)$	// $\delta(q, \varepsilon, S) \to (q, E)$
$\vdash (q, i+i(i), ESZ_0)$	// $\delta(q, \varepsilon, E) \to (q, E+E)$
$\vdash (q, i+i(i), E+ESZ_0)$	// $\delta(q, \varepsilon, E) \to (q, i)$
$\vdash (q, i+i(i), i+ESZ_0)$	// $\delta(q, i, i) \to (q, \varepsilon)$
$\vdash (q, +i(i), +ESZ_0)$	// $\delta(q, +, +) \to (q, \varepsilon)$
$\vdash (q, i(i), ESZ_0)$	// $\delta(q, \varepsilon, E) \to (q, i)$
$\vdash (q, i(i), iSZ_0)$	// $\delta(q, i, i) \to (q, \varepsilon)$
$\vdash (q, (i), SZ_0)$	// $\delta(q, \varepsilon, S) \to (q, E)$
$\vdash (q, (i), EZ_0)$	// $\delta(q, \varepsilon, E) \to (q, (E))$
$\vdash (q, (i), (E)Z_0)$	// $\delta(q, (, () \to (q, \varepsilon)$
$\vdash (q, i), E)Z_0)$	// $\delta(q, \varepsilon, E) \to (q, i)$
$\vdash (q, i), i)Z_0)$	// $\delta(q, i, i) \to (q, \varepsilon)$
$\vdash (q,),)Z_0)$	// $\delta(q,),)) \to (q, \varepsilon)$
$\vdash (q, \$, Z_0)$	// $\delta(q, \$, Z_0) \to (q_F, \varepsilon)$
$\vdash (q_F, \$, \varepsilon)$	

1.21

Example—Derivation Reconstruction

Grammar "rules" used in the following order:

1. $S \to SS$
2. $S \to E$
3. $E \to E + E$
4. $E \to i$
5. $E \to i$
6. $S \to E$
7. $E \to (E)$
8. $E \to i$

It was **left derivation**, thus we get:

$S \Rightarrow SS \Rightarrow ES \Rightarrow E+ES \Rightarrow i+ES \Rightarrow i+iS \Rightarrow \quad \Rightarrow i+iE \Rightarrow i+i(E) \Rightarrow i+i(i)$ 1.22

3.2 Bottom-Up Analysis

Definitions

Bottom-Up Analysis
Reconstruction of the derivation tree starts from the symbols of the input sentence up to the starting non-terminal of the respective grammar.

Definition of Extended PA
An extended pushdown automaton is a seven-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, where Q, Σ, Γ, q_0, Z_0, and F have the same meaning as for pushdown automaton.

δ is defined as mapping $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^* \rightarrow 2^{Q \times \Gamma^*}$

1.23

Construction
Let $G = (N, T, P, S)$ be a context-free grammar then construct $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ this way:

- $Q = \{q_0, q_1, q_F\}$
- $\Sigma = T \cup \{\$\}$
- $\Gamma = \{Z_0\} \cup T \cup N$
- $F = \{q_F\}$
- For all $A \rightarrow \alpha \in P, A \in N, \alpha \in (N \cup T)^*$
  add $(q_0, \varepsilon, \alpha^R) \rightarrow \{(q_0, A)\}$ to δ
  For all $a \in T$ and all $z \in \Gamma$ add $(q_0, a, z) \rightarrow \{(q_0, az)\}$ to δ
  Add $(q_0, \varepsilon, S) \rightarrow \{(q_1, \varepsilon)\}$ to δ
  Add $(q_1, \$, Z_0) \rightarrow \{(q_F, \varepsilon)\}$ to δ

1.24

Example—Construction

CF Grammar
Let $G = (N, T, P, S)$, where: $N = \{S, E\}$, $T = \{\ (,), +, i\ \}$, $P = \{\ S \rightarrow SS \mid E,\ E \rightarrow E + E \mid i \mid (E)\ \}$.

PA Construction
Let $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, where:

- $Q = \{q_0, q_1, q_F\}$, $F = \{q_F\}$
- $\Sigma = T \cup \{\$\} = \{\ (,), +, i, \$\ \}$
- $\Gamma = \{Z_0\} \cup T \cup N = \{S, E, (,), +, i, Z_0\}$
- $\delta(q_0, \varepsilon, S) \rightarrow \{(q_1, \varepsilon)\}$
  $\delta(q_1, \$, Z_0) \rightarrow \{(q_F, \varepsilon)\}$
  $\delta(q_0, \varepsilon, SS) \rightarrow \{(q_0, S)\}$
  $\delta(q_0, \varepsilon, E) \rightarrow \{(q_0, S)\}$
  $\delta(q_0, \varepsilon, E + E) \rightarrow \{(q_0, E)\}$
  $\delta(q_0, \varepsilon, i) \rightarrow \{(q_0, E)\}$
  $\delta(q_0, \varepsilon,)E() \rightarrow \{(q_0, E)\} \ldots$

1.25

Example—Construction

PA Construction

- $\ldots\ \delta(q_0, (, Z_0) \rightarrow \{(q_0, (Z_0)\}$
  $\delta(q_0,), Z_0) \rightarrow \{(q_0,)Z_0)\}$
  $\delta(q_0, +, Z_0) \rightarrow \{(q_0, +Z_0)\}$
  $\delta(q_0, i, Z_0) \rightarrow \{(q_0, iZ_0)\}$
  $\delta(q_0, (, S) \rightarrow \{(q_0, (S)\}$
  $\delta(q_0,), S) \rightarrow \{(q_0,)S)\}$
  $\delta(q_0, +, S) \rightarrow \{(q_0, +S)\}$
  $\delta(q_0, i, S) \rightarrow \{(q_0, iS)\}$
  $\delta(q_0, (, E) \rightarrow \{(q_0, (E)\}$
  $\delta(q_0,), E) \rightarrow \{(q_0,)E)\}$
  $\delta(q_0, +, E) \rightarrow \{(q_0, +E)\}$
  $\delta(q_0, i, E) \rightarrow \{(q_0, iE)\} \ldots$

$\ldots$ for all $z \in \Gamma$

1.26

Example—Analysis

$(q_0, i+i(i), Z_0)$	// $\delta(q_0, i, Z_0) \rightarrow (q_0, iZ_0)$
$\vdash (q_0, +i(i), iZ_0)$	// $\delta(q_0, \varepsilon, i) \rightarrow (q_0, E)$
$\vdash (q_0, +i(i), EZ_0)$	// $\delta(q_0, +, E) \rightarrow (q_0, +E)$
$\vdash (q_0, i(i), +EZ_0)$	// $\delta(q_0, i, +) \rightarrow (q_0, i+)$
$\vdash (q_0, (i), i+EZ_0)$	// $\delta(q_0, \varepsilon, i) \rightarrow (q_0, E)$
$\vdash (q_0, (i), E+EZ_0)$	// $\delta(q_0, \varepsilon, E+E) \rightarrow (q_0, E)$
$\vdash (q_0, (i), EZ_0)$	// $\delta(q_0, \varepsilon, E) \rightarrow (q_0, S)$
$\vdash (q_0, (i), SZ_0)$	// $\delta(q_0, (, S) \rightarrow (q_0, (S)$
$\vdash (q_0, i), (SZ_0)$	// $\delta(q_0, i, () \rightarrow (q_0, i()$
$\vdash (q_0,), i(SZ_0)$	// $\delta(q_0, \varepsilon, i) \rightarrow (q_0, E)$
$\vdash (q_0,), E(SZ_0)$	// $\delta(q_0,), E) \rightarrow (q_0,)E)$
$\vdash (q_0, \$,)E(SZ_0)$	// $\delta(q_0, \varepsilon,)E() \rightarrow (q_0, E)$
$\vdash (q_0, \$, ESZ_0)$	// $\delta(q_0, \varepsilon, E) \rightarrow (q_0, S)$
$\vdash (q_0, \$, SSZ_0)$	// $\delta(q_0, \varepsilon, SS) \rightarrow (q_0, S)$
$\vdash (q_0, \$, SZ_0)$	// $\delta(q_0, \varepsilon, S) \rightarrow (q_1, \varepsilon)$
$\vdash (q_1, \$, Z_0)$	// $\delta(q_1, \$, Z_0) \rightarrow (q_F, \varepsilon)$
$\vdash (q_F, \$, \varepsilon)$	

1.27

Example—Derivation Reconstruction

Grammar “rules” used in the following order:

1. $E \rightarrow i$
2. $E \rightarrow i$
3. $E \rightarrow E+E$
4. $S \rightarrow E$
5. $E \rightarrow i$
6. $E \rightarrow (E)$
7. $S \rightarrow E$
8. $S \rightarrow SS$

It was **right derivation** in reverse order, thus we get:

$S \Rightarrow SS \Rightarrow SE \Rightarrow S(E) \Rightarrow S(i) \Rightarrow E(i) \Rightarrow \quad \Rightarrow E+E(i) \Rightarrow E+i(i) \Rightarrow i+i(i)$ 1.28

4 Deterministic/Non-Deterministic PA

Importance of Determinism

Non-deterministic analysis of context-free languages

- Possible
- With backtracking
- Inefficient

Deterministic analysis of context-free languages

- Possible for certain, but sufficiently large subset of context-free languages
- Without backtracking
- Efficient

1.29

4.1 Non-Deterministic PA

Non-Deterministic Pushdown Automata

Presented definition matches non-deterministic pushdown automata

Where is the non-determinism?

- $\delta(q, a, z) \rightarrow \{(q', \alpha), (q'', \beta), \dots\}$ — multiple choice
- $\delta(q, \varepsilon, z) \rightarrow \{\dots\}$ — ε matches always

All examples presented so far demonstrated non-deterministic pushdown automata 1.30

4.2 Deterministic PA

Deterministic Pushdown Automata

Formal Definition of Deterministic Pushdown Automaton
A deterministic pushdown automaton is a seven-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, where

- Q — finite *set of states*
- Σ — finite *input alphabet*
- Γ — finite *alphabet of pushdown symbols*
- δ — mapping $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow Q \times \Gamma^*$ *transition function*
- $q_0 \in Q$ — *starting/initial state*
- $Z_0 \in \Gamma$ — *start symbol* on the pushdown
- $F \subseteq Q$ — *set of final states*

Configuration, move, final configuration, and accepted language defined the same way 1.31

Deterministic Pushdown Automata

Is the definition sufficient to ensure determinism?

No!

- Eliminated multiple choices,
- but the other option must be eliminated by **proper construction**.
  Compare: $\delta(q, a, Z) \rightarrow (p, u)$ versus $\delta(q, \varepsilon, Z) = (p', u')$

1.32

Deterministic Pushdown Automata

Is it possible to find deterministic automaton for every non-deterministic one?

Unfortuntely, **no!**

- Principal reasons
- Ambiguity of grammar used for automata construction

Let us try ... 1.33

4.3 Examples

Example 1' ($a^n b^n$, $n \geq 0$) — Introduction

Possible solution features:

- Separation of empty and non-empty input cases
- State for reading of *a*s with duplication to the pushdown
- State for reading of *b*s when removing symbols from pushdown to check that number of both is equal
- Separete final state — final configuration 3 (empty pushdown & final state)

1.34

Example 1' ($a^n b^n$, $n \geq 0$) — Definition

$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$
$Q = \{q_0, q_1, q_2, q_F\}$ $\Sigma = \{a, b, \$\}$ $\Gamma = \{Z_0\}$ $F = \{q_F\}$
δ: $\delta(q_0, a, Z_0) \rightarrow (q_1, Z_0Z_0)$
$\delta(q_1, a, Z_0) \rightarrow (q_1, Z_0Z_0)$
$\delta(q_1, b, Z_0) \rightarrow (q_2, \varepsilon)$
$\delta(q_2, b, Z_0) \rightarrow (q_2, \varepsilon)$
$\delta(q_2, \$, Z_0) \rightarrow (q_F, \varepsilon)$
$\delta(q_0, \$, Z_0) \rightarrow (q_F, \varepsilon)$

1.35

Example 1' ($a^n b^n$, $n \geq 0$) — Acceptance

$$\begin{array}{ll}
(q_0, aaabbb, Z_0) & /\!/\ \delta(q_0,a,Z_0) \to (q_1, Z_0Z_0) \\
\quad \vdash (q_1, aabbb, Z_0Z_0) & /\!/\ \delta(q_1,a,Z_0) \to (q_1, Z_0Z_0) \\
\quad \vdash (q_1, abbb, Z_0Z_0Z_0) & /\!/\ \delta(q_1,a,Z_0) \to (q_1, Z_0Z_0) \\
\quad \vdash (q_1, bbb, Z_0Z_0Z_0Z_0) & /\!/\ \delta(q_1,b,Z_0) \to (q_2, \varepsilon) \\
\quad \vdash (q_2, bb, Z_0Z_0Z_0) & /\!/\ \delta(q_2,b,Z_0) \to (q_2, \varepsilon) \\
\quad \vdash (q_2, b, Z_0Z_0) & /\!/\ \delta(q_2,b,Z_0) \to (q_2, \varepsilon) \\
\quad \vdash (q_2, \$, Z_0) & /\!/\ \delta(q_2,\$,Z_0) \to (q_F, \varepsilon) \\
\quad \vdash (q_F, \$, \varepsilon) &
\end{array}$$

1.36

Example 1' ($a^n b^n$, $n \geq 0$) — Failure

$$\begin{array}{ll}
(q_0, aabbb, Z_0) & /\!/\ \delta(q_0,a,Z_0) \to (q_1, Z_0Z_0) \\
\quad \vdash (q_1, abbb, Z_0Z_0) & /\!/\ \delta(q_1,a,Z_0) \to (q_1, Z_0Z_0) \\
\quad \vdash (q_1, bbb, Z_0Z_0Z_0) & /\!/\ \delta(q_1,b,Z_0) \to (q_2, \varepsilon) \\
\quad \vdash (q_2, bb, Z_0Z_0) & /\!/\ \delta(q_2,b,Z_0) \to (q_2, \varepsilon) \\
\quad \vdash (q_2, b, Z_0) & /\!/\ \delta(q_2,b,Z_0) \to (q_2, \varepsilon) \\
\quad \vdash (q_2, \$, \varepsilon) & /\!/\ \text{no rule to be applied}
\end{array}$$

1.37

Example 1' ($a^n b^n$, $n \geq 0$) — Discussion

Creation possible

No algorithm to transform the non-deterministic PA

Other solutions may work as well *What modification can be done?* *Can we reduce number of states?*

A non-ambiguous grammar exists $S \to \varepsilon \mid aSb$ Other features: LL(1) 1.38

Example 2' ($ww^R, w \in \{a,b\}^+$) — Discussion

No deterministic PA exists

Reason: it is impossible to denote the midpoint without additional information deterministically.

A non-ambiguous grammar exists though $S \to \varepsilon \mid aSa \mid bSb$ Other features: can't be LL(1)

Question: can a deterministic PA be defined for language $wcw^R, w \in \{a,b\}^+$? 1.39

Example 3 — Introduction

Definition based on grammar description of the language: $P = \{\ S \to SS \mid E,\ E \to E+E \mid i \mid (E)\ \}$.

Problem: ambiguous grammar
Consider: $i+i+i$
$S \Rightarrow E \Rightarrow E_1 + E_2 \Rightarrow E_{1.1} + E_{1.2} + E_2 \Rightarrow \ldots$
$S \Rightarrow E \Rightarrow E_1 + E_2 \Rightarrow E_1 + E_{2.1} + E_{2.2} \Rightarrow \ldots$

We can make it non-ambiguous: $P = \{S \to ES',\ S' \to \varepsilon \mid ES',\ E \to (E)E' \mid iE',\ E' \to \varepsilon \mid +E\}$

Does it help? Not for human... but for machine, maybe. We need a modification of PA. 1.40

5 Modifications of PA

Modifications of Pushdown Automata

Purpose:

- new features to make our life easier
- limit features to study properties and simplify analysis
- extend features (regulated PA)

Selected modifications:

- Non-Head-Moving PA
- One-Turn PA
- Atomic PA

1.41

5.1 Non-Head-Moving PA

Head Does Not Move During Reading

Limiting feature: when recognizing symbol under the reading head, the head has to move.

Formal modification of transition function:
$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to Q \times \Gamma^* \times \{\mathscr{R}, \mathscr{M}\}$
$\mathscr{R}$ stays for reading the symbol under the reading head only
$\mathscr{M}$ stays for read&move (traditional behaviour)

Notation:
$\delta(q,a,z) \to (q', \alpha)_R$
$\delta(q,a,z) \to (q', \alpha)_M$ we write $\delta(q,a,z) \to (q', \alpha)$ if clear
where $q, q' \in Q$, $a \in \Sigma$, $z \in \Gamma$, $\alpha \in \Gamma^*$

Generally non-deterministic variant: $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to 2^{Q \times \Gamma^* \times \{\mathscr{R}, \mathscr{M}\}}$ 1.42

Example 1" ($a^n b^n$, $n \geq 0$) — Definition

$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$
$Q = \{q_0\}$ $\Sigma = \{a, b, \$\}$ $\Gamma = \{a, b, Z_0\}$ $F = \{q_0\}$
δ: $\delta(q_0, a, Z_0) \to (q_0, aZ_0b)_R$
$\delta(q_0, b, Z_0) \to (q_0, \varepsilon)_R$
$\delta(q_0, a, a) \to (q_0, \varepsilon)$
$\delta(q_0, b, b) \to (q_0, \varepsilon)$
$\delta(q_0, \$, Z_0) \to (q_0, \varepsilon)$

Final configuration 3. 1.43

Example 1" ($a^n b^n$, $n \geq 0$) — Acceptance

$(q_0, aaabbb, Z_0)$	// $\delta(q_0, a, Z_0) \to (q_0, aZ_0b)_R$
$\vdash (q_0, aaabbb, aZ_0b)$	// $\delta(q_0, a, a) \to (q_0, \varepsilon)$
$\vdash (q_0, aabbb, Z_0b)$	// $\delta(q_0, a, Z_0) \to (q_0, aZ_0b)_R$
$\vdash (q_0, aabbb, aZ_0bb)$	// $\delta(q_0, a, a) \to (q_0, \varepsilon)$
$\vdash (q_0, abbb, Z_0bb)$	// $\delta(q_0, a, Z_0) \to (q_0, aZ_0b)_R$
$\vdash (q_0, abbb, aZ_0bbb)$	// $\delta(q_0, a, a) \to (q_0, \varepsilon)$
$\vdash (q_0, bbb, Z_0bbb)$	// $\delta(q_0, b, Z_0) \to (q_0, \varepsilon)_R$
$\vdash (q_0, bbb, bbb)$	// $\delta(q_0, b, b) \to (q_0, \varepsilon)$
$\vdash (q_0, bb, bb)$	// $\delta(q_0, b, b) \to (q_0, \varepsilon)$
$\vdash (q_0, b, b)$	// $\delta(q_0, b, b) \to (q_0, \varepsilon)$
$\vdash (q_0, \$, \varepsilon)$	

1.44

Example 1" ($a^n b^n$, $n \geq 0$) — Failure

$(q_0, aabbb, Z_0)$	// $\delta(q_0, a, Z_0) \to (q_0, aZ_0b)_R$
$\vdash (q_0, aabbb, aZ_0b)$	// $\delta(q_0, a, a) \to (q_0, \varepsilon)$
$\vdash (q_0, abbb, Z_0b)$	// $\delta(q_0, a, Z_0) \to (q_0, aZ_0b)_R$
$\vdash (q_0, abbb, aZ_0bb)$	// $\delta(q_0, a, a) \to (q_0, \varepsilon)$
$\vdash (q_0, bbb, Z_0bb)$	// $\delta(q_0, b, Z_0) \to (q_0, \varepsilon)_R$
$\vdash (q_0, bbb, bb)$	// $\delta(q_0, b, b) \to (q_0, \varepsilon)$
$\vdash (q_0, bb, b)$	// $\delta(q_0, b, b) \to (q_0, \varepsilon)$
$\vdash (q_0, b, \varepsilon)$	// no rule to be applied

1.45

Example 3' — Definition

$P = \{S \to ES',\ S' \to \varepsilon \mid ES',\ E \to (E)E' \mid iE',\ E' \to \varepsilon \mid +E\}$

$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$
$Q = \{q_0\}$ $\Sigma = \{+, i, (,), \$\}$ $\Gamma = \{+, i, (,), E, S', E', Z_0\}$ $F = \{q_0\}$
δ: $\delta(q_0, (, Z_0) \to (q_0, ES')_R$
$\delta(q_0, i, Z_0) \to (q_0, ES')_R$
$\delta(q_0, (, E) \to (q_0, (E)E')_R$
$\delta(q_0, i, E) \to (q_0, iE')_R$
$\delta(q_0, (, S') \to (q_0, ES')_R$
$\delta(q_0, i, S') \to (q_0, ES')_R$
$\delta(q_0, \$, S') \to (q_0, \varepsilon)_R$
...

1.46

Example 3' — Definition

...

$\delta(q_0, +, E') \to (q_0, +E)_R$
$\delta(q_0,), E') \to (q_0, \varepsilon)_R$
$\delta(q_0, (, E') \to (q_0, \varepsilon)_R$
$\delta(q_0, i, E') \to (q_0, \varepsilon)_R$
$\delta(q_0, \$, E') \to (q_0, \varepsilon)_R$
$\delta(q_0,),)) \to (q_0, \varepsilon)$
$\delta(q_0, (, () \to (q_0, \varepsilon)$
$\delta(q_0, i, i) \to (q_0, \varepsilon)$
$\delta(q_0, +, +) \to (q_0, \varepsilon)$

1.47

Example 3' — Analysis

$(q_0, i+i(i), Z_0)$	// $\delta(q_0, i, Z_0) \to (q_0, ES')_R$
$\vdash (q_0, i+i(i), ES')$	// $\delta(q_0, i, E) \to (q_0, iE')_R$
$\vdash (q_0, i+i(i), iE'S')$	// $\delta(q_0, i, i) \to (q_0, \varepsilon)$
$\vdash (q_0, +i(i), E'S')$	// $\delta(q_0, +, E') \to (q_0, +E)_R$
$\vdash (q_0, +i(i), +ES')$	// $\delta(q_0, +, +) \to (q_0, \varepsilon)$
$\vdash (q_0, i(i), ES')$	// $\delta(q_0, i, E) \to (q_0, iE')_R$
$\vdash (q_0, i(i), iE'S')$	// $\delta(q_0, i, i) \to (q_0, \varepsilon)$
$\vdash (q_0, (i), E'S')$	// $\delta(q_0, (, E') \to (q_0, \varepsilon)_R$
$\vdash (q_0, (i), S')$	// $\delta(q_0, (, S') \to (q_0, ES')_R$
$\vdash (q_0, (i), ES')$	// $\delta(q_0, (, E) \to (q_0, (E)E')_R$
$\vdash (q_0, (i), (E)E'S')$	// $\delta(q_0, (, () \to (q_0, \varepsilon)$
$\vdash (q_0, i), E)E'S')$	// $\delta(q_0, i, E) \to (q_0, iE')_R$
$\vdash (q_0, i), iE')E'S')$	// $\delta(q_0, i, i) \to (q_0, \varepsilon)$
$\vdash (q_0,), E')E'S')$	// $\delta(q_0,), E') \to (q_0, \varepsilon)_R$
$\vdash (q_0,),)E'S')$	// $\delta(q_0,),)) \to (q_0, \varepsilon)$
$\vdash (q_0, \$, E'S')$	// $\delta(q_0, \$, E') \to (q_0, \varepsilon)_R$
$\vdash (q_0, \$, S')$	// $\delta(q_0, \$, S') \to (q_0, \varepsilon)_R$
$\vdash (q_0, \$, \varepsilon)$	

1.48

Example 3' — Discussion

Grammar features:

- Non-ambiguous
- LL(1)
- Not trivial to construct

How could we construct the PA?

- By straightforward reasoning from the grammar—**too difficult!**
- By algorithm
  - See Aho, Sethi, Ullman: Compilers ... in References
  - LL(1) grammar is an input
  - Non-moving pushdown automaton necessary

1.49

5.2 One-Turn PA

Definition of One-Turn PA

Definition
Consider two consecutive moves made by a pushdown automaton, M. If during the first move M does not shorten its pushdown and during the second move it does, then M makes a turn during the second move. A pushdown automaton is *one-turn* if it makes no more than one turn with its pushdown during any computation starting from an initial configuration.

Theorem
One-turn pushdown automata characterize a family of linear languages. *Proof*: see Harrison ... in References. 1.50

Discussion

Example of one-turn PA—Example 1'

Can we re-construct Examples 1 and 2 to be one-turn? **YES!**

Do it!

Hint: differentiate extension and shortening of pushdown by states, allow just one turn! 1.51

5.3 Atomic PA

Definition of Atomic PA

Definition
Modify δ as a mapping from $Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\})$. During a move, an *atomic* pushdown automaton changes a state and, in addition, performs exactly one of the following actions:

1. it pushes a symbol onto the pushdowne.g. $\delta(q,\varepsilon,\varepsilon) \to \{(q',a)\}$
2. it pops a symbol from the pushdowne.g. $\delta(q,\varepsilon,a) \to \{(q',\varepsilon)\}$
3. it reads an input symbole.g. $\delta(q,a,\varepsilon) \to \{(q',\varepsilon)\}$

1.52

Example 1''' ($a^n b^n$, $n \geq 0$) — Definition

$M = (Q,\Sigma,\Gamma,\delta,q_0,Z_0,F)$
$Q = \{q_0,q_a,q_{a'},q_{a''},q_b,q_F\}$ $\Sigma = \{a,b,\$\}$ $\Gamma = \{b,Z_0\}$ $F = \{q_F\}$
δ: $\delta(q_0,a,\varepsilon) \to (q_a,\varepsilon)$
$\delta(q_a,\varepsilon,Z_0) \to (q_{a'},\varepsilon)$
$\delta(q_{a'},\varepsilon,\varepsilon) \to (q_{a''},b)$
$\delta(q_{a''},\varepsilon,\varepsilon) \to (q_0,Z_0)$
$\delta(q_0,b,\varepsilon) \to (q_b,\varepsilon)$
$\delta(q_b,\varepsilon,Z_0) \to (q_b,\varepsilon)$
$\delta(q_b,\varepsilon,b) \to (q_0,\varepsilon)$
$\delta(q_0,\$,\varepsilon) \to (q_F,\varepsilon)$
$\delta(q_F,\varepsilon,Z_0) \to (q_F,\varepsilon)$

Features: deterministic, atomic $\Rightarrow$ easy to implement 1.53

Example 1''' ($a^n b^n$, $n \geq 0$) — Acceptance

$(q_0,aabb,Z_0)$	// $\delta(q_0,a,\varepsilon) \to (q_a,\varepsilon)$
$\vdash (q_a,abb,Z_0)$	// $\delta(q_a,\varepsilon,Z_0) \to (q_{a'},\varepsilon)$
$\vdash (q_{a'},abb,\varepsilon)$	// $\delta(q_{a'},\varepsilon,\varepsilon) \to (q_{a''},b)$
$\vdash (q_{a''},abb,b)$	// $\delta(q_{a''},\varepsilon,\varepsilon) \to (q_0,Z_0)$
$\vdash (q_0,abb,Z_0b)$	// $\delta(q_0,a,\varepsilon) \to (q_a,\varepsilon)$
$\vdash (q_a,bb,Z_0b)$	// $\delta(q_a,\varepsilon,Z_0) \to (q_{a'},\varepsilon)$
$\vdash (q_{a'},bb,b)$	// $\delta(q_{a'},\varepsilon,\varepsilon) \to (q_{a''},b)$
$\vdash (q_{a''},bb,bb)$	// $\delta(q_{a''},\varepsilon,\varepsilon) \to (q_0,Z_0)$
$\vdash (q_0,bb,Z_0bb)$	// $\delta(q_0,b,\varepsilon) \to (q_b,\varepsilon)$
$\vdash (q_b,b,Z_0bb)$	// $\delta(q_b,\varepsilon,Z_0) \to (q_b,\varepsilon)$
$\vdash (q_b,b,bb)$	// $\delta(q_b,\varepsilon,b) \to (q_0,\varepsilon)$

$\vdash (q_0, b, b)$ // $\delta(q_0, b, \varepsilon) \to (q_b, \varepsilon)$
$\vdash (q_b, \$, b)$ // $\delta(q_b, \varepsilon, b) \to (q_0, \varepsilon)$
$\vdash (q_0, \$, \varepsilon)$ // $\delta(q_0, \$, \varepsilon) \to (q_F, \varepsilon)$
$\vdash (q_F, \$, \varepsilon)$

1.54

Example 1''' ($a^n b^n$, $n \geq 0$) — Failure

(q_0, a, Z_0) // $\delta(q_0, a, \varepsilon) \to (q_a, \varepsilon)$
$\vdash (q_a, \$, Z_0)$ // $\delta(q_a, \varepsilon, Z_0) \to (q_{a'}, \varepsilon)$
$\vdash (q_{a'}, \$, \varepsilon)$ // $\delta(q_{a'}, \varepsilon, \varepsilon) \to (q_{a''}, b)$
$\vdash (q_{a''}, \$, b)$ // $\delta(q_{a''}, \varepsilon, \varepsilon) \to (q_0, Z_0)$
$\vdash (q_0, \$, Z_0 b)$ // $\delta(q_0, \$, \varepsilon) \to (q_F, \varepsilon)$
$\vdash (q_F, \$, Z_0 b)$ // $\delta(q_F, \varepsilon, Z_0) \to (q_F, \varepsilon)$
$\vdash (q_F, \$, b)$ // no rule to be applied

(q_0, b, Z_0) // $\delta(q_0, b, \varepsilon) \to (q_b, \varepsilon)$
$\vdash (q_b, \$, Z_0)$ // $\delta(q_b, \varepsilon, Z_0) \to (q_b, \varepsilon)$
$\vdash (q_b, \$, \varepsilon)$ // no rule to be applied

1.55

Example 1''' ($a^n b^n$, $n \geq 0$) — Discussion

Easy to implement $\not\Leftarrow$ easy to define (algorithmically)

Quite a lot of steps ...

Automaton without movement of reading head is much more efficient ...

Importance in the formal theory

- simplicity of definition
- minimality of description
- ...

1.56

6 Conclusion

Conclusion

We have presented:

- Definition of pushdown automaton
- Connected terms: configuration, transition step, language accepted by PA
- Deterministic and non-deterministic variant
- Some modifications

Main importance in:

- Analysis of context-free languages
- Theoretical computer science

1.57

References

References

Aho, A.V., Ullman, J.D.: *The Theory of Parsing, Translation, and Compiling*, Volume I: Parsing, Prentice-Hall, 1972, ISBN 0-13-914556-7.

Aho, A.V., Sethi, R., Ullman, J.D.: *Compilers, Principles, Techniques, and Tools*, Addison-Wesley, 1986, ISBN 0-201-10194-7.

Merriam Webster, http://www.merriam-webster.com/

Harrison, M.: *Introduction to Formal Language Theory*, Addison Wesley, Reading, 1978.

1.58