Regulated Pushdown Automata

Alexander Meduna

Faculty of Information Technology

Brno University of Technology

Brno, Czech Republic, Europe

Fundamental References

- **Meduna Alexander, Kolář Dušan:**
Regulated Pushdown Automata, ***Acta Cybernetica*, Vol. 2000, No. 4, p. 653-664**

- **Meduna Alexander, Kolář Dušan:**
One-Turn Regulated Pushdown Automata and Their Reduction, *Fundamenta Informatica*, Vol. 2002, No. 16, p. 399-405

Inspiration: Regulated Grammars

- **Grammar G:**

1. $S \rightarrow AC$
2. $A \rightarrow aAb$
3. $A \rightarrow ab$
4. $C \rightarrow Cc$
5. $C \rightarrow c$

- $\Xi = \{1\}\{24\}^*\{35\}$

Regulated Grammars 1/2

- **Grammar G:**

  1. $S \rightarrow AC$

  2. $A \rightarrow aAb$

  3. $A \rightarrow ab$

  4. $C \rightarrow Cc$

  5. $C \rightarrow c$

$\Xi = \{1\}\{24\}^*\{35\}$

- **Without Ξ, G generates *aabbccc*:**

$S \Rightarrow AC$ [1]
$\Rightarrow aAbC$ [2]
$\Rightarrow aAbCc$ [4]
$\Rightarrow aabbCc$ [3]
$\Rightarrow aabbCcc$ [4]
$\Rightarrow aabbccc$ [5]

$$L(G) = \{a^n b^n c^m : n, m \geq 1\}$$

Regulated Grammars 2/2

- with Ξ, G does not generate ***aabbcccc***, because

$$124345 \notin \Xi = \{1\}\{24\}^*\{35\}$$

- with Ξ, G generates ***aabbcc***:

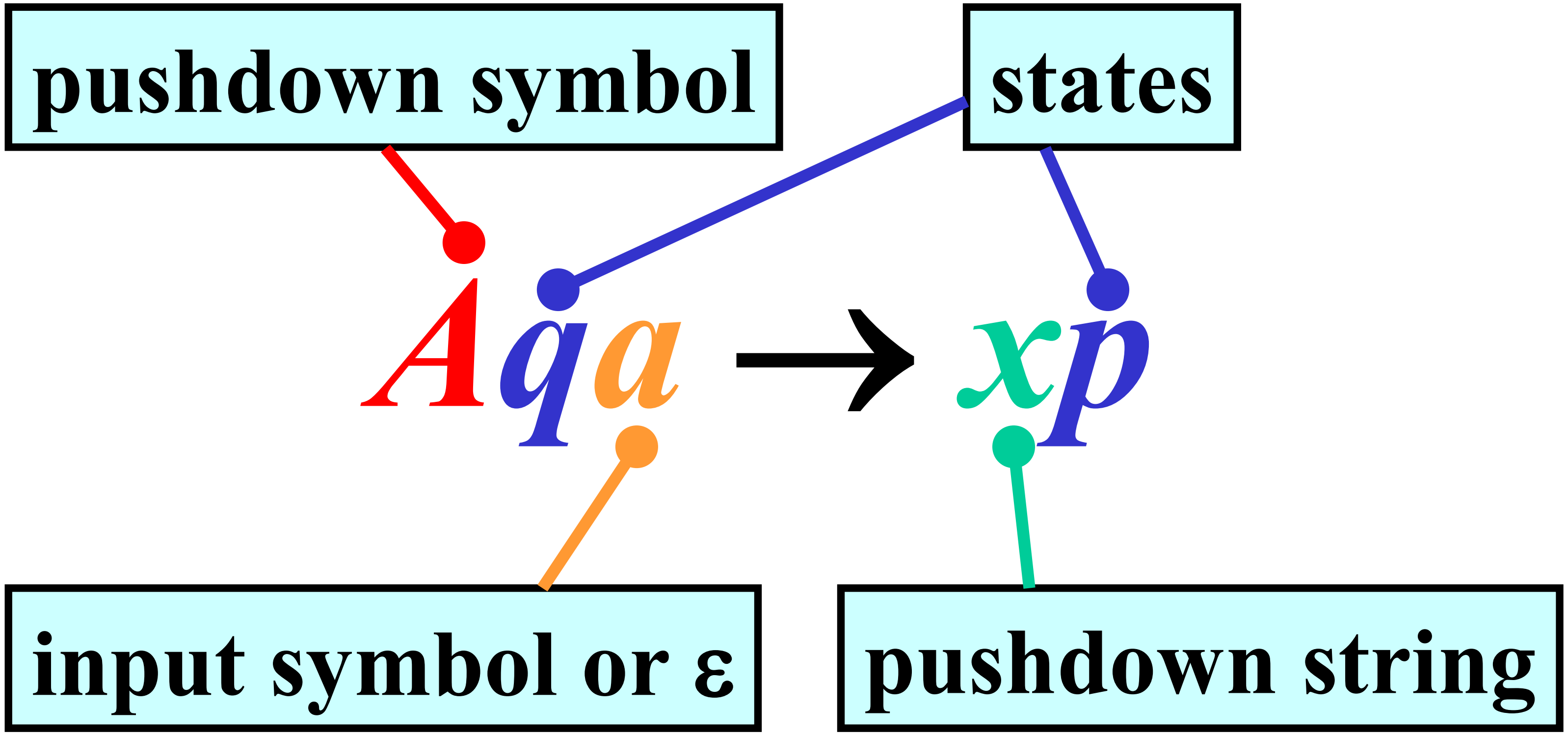
$$
\begin{aligned}
S &\Rightarrow AC & [1] \\
&\Rightarrow aAbC & [2] \\
&\Rightarrow aAbCc & [4] \\
&\Rightarrow aabbCc & [3] \\
&\Rightarrow aabbcc & [5]
\end{aligned}
$$

and $12435 \in \Xi$

$$L(G, \Xi) = \{a^n b^n c^n : n \geq 1\}$$

PDA: Notation

- **A PDA is based on a finite set of rules of the form:**



New Concept: Regulated PDAs

- **PDA M:**

1. $Ssa \rightarrow Sas$
2. $asa \rightarrow aas$
3. $asb \rightarrow q$
4. $aqb \rightarrow q$
5. $Sqc \rightarrow Sq$
6. $Sqc \rightarrow f$

- $\Xi = \{12^m34^n5^n6 : m, n \geq 0\}$

Regulated PDAs 1/2

- **PDA *M*:**

  1. $Ssa \rightarrow Sas$
  2. $asa \rightarrow aas$
  3. $asb \rightarrow q$
  4. $aqb \rightarrow q$
  5. $Sqc \rightarrow Sq$
  6. $Sqc \rightarrow f$

$\Xi = \{12^m34^n5^n6 : m, n \geq 0\}$

- **Without Ξ, *M* accepts *aabbccc*:**

$Ssaabbccc$
$\Rightarrow Sasabbccc$ [1]
$\Rightarrow Saasbbccc$ [2]
$\Rightarrow Saqbccc$ [3]
$\Rightarrow Sqccc$ [4]
$\Rightarrow Sqcc$ [5]
$\Rightarrow Sqc$ [5]
$\Rightarrow f$ [6]

$$L(M) = \{a^n b^n c^m : n, m \geq 1\}$$

Regulated PDAs 2/2

- with Ξ, M does not accept ***aabbccc*** because

$$1234556 \notin \Xi = \{12^m34^n5^n6 : m, n \geq 0\}$$

- with Ξ, M accepts ***aabbcc***:

$$\begin{aligned} Ssaabbcc &\Rightarrow Sasabbcc && [1] \\ &\Rightarrow Saasbbcc && [2] \\ &\Rightarrow Saqbcc && [3] \\ &\Rightarrow Sqcc && [4] \\ &\Rightarrow Sqc && [5] \\ &\Rightarrow f && [6] \end{aligned}$$

and $123456 \in \Xi$

$$L(M, \Xi) = \{a^nb^nc^n : n \geq 1\}$$

Gist: Regulated PDAs

- Consider a pushdown automaton, *M*, and control language, Ξ.
- *M* accepts a string, *x*, if and only if Ξ contains a control string according to which *M* makes a sequence of moves so it reaches a final configuration after reading *x*.

Definition: Regulated PDA 1/4

A pushdown automaton is a 7-tuple

$$M = (Q, \Sigma, \Omega, R, s, S, F), \text{ where}$$

- Q is a *finite set of states*,
- Σ is an *input alphabet*,
- Ω is a *pushdown alphabet*,
- R is a *finite set of rules* of the form:

$$Apa \rightarrow wq, \text{ where}$$

$$A \in \Omega, p,q \in Q, a \in \Sigma \cup \{\varepsilon\}, w \in \Omega^*$$

- $s \in Q$ is the *start state*
- $S \in \Omega$ is the *start symbol*
- $F \subseteq Q$ is a set of *final states*

Definition: Regulated PDA 2/4

- Let Ψ be an alphabet of *rule labels*. Let every rule $Apa \rightarrow wq$ be labeled with a unique $\rho \in \Psi$ as

$$\rho.\ Apa \rightarrow wq.$$

- A configuration of M, χ, is any string from $\Omega^* Q \Sigma^*$

- For every $x \in \Omega^*$, $y \in \Sigma^*$, and $\rho.\ Apa \rightarrow wq \in R$, M makes a move from configuration $xApay$ to configuration $xwqy$ according to ρ, written as

$$xApay \Rightarrow xwqy\ [\rho]$$

Definition: Regulated PDA 3/4

- Let χ be any configuration of *M*. *M* makes *zero moves* from χ to χ *according to* ε, written as

$$\chi \Rightarrow^0 \chi\ [\varepsilon]$$

- Let there exist a sequence of configurations $\chi_0, \chi_1, ..., \chi_n$ for some $n \geq 1$ such that $\chi_{i-1} \Rightarrow \chi_i\ [\rho_i]$, where $\rho_i \in \Psi$, for $i = 1,...,n$, then *M* makes *n moves from* χ_0 *to* χ_n *according to* $[\rho_1 \dots \rho_n]$, written as

$$\chi_0 \Rightarrow^n \chi_n\ [\rho_1 ... \rho_n]$$

Definition: Regulated PDA 3/4

- If for some $n \geq 0$, $\chi_0 \Rightarrow^n \chi_n$ $[\rho_1 \ldots \rho_n]$, we write

$$\chi_0 \Rightarrow^* \chi_n \; [\rho_1 \ldots \rho_n]$$

- Let Ξ be a *control language* over Ψ, that is, $\Xi \subseteq \Psi^*$. With Ξ, M accepts its language, $L(M, \Xi)$, as

$$L(M, \Xi) = \{w\colon w \in \Sigma^*, Ssw \Rightarrow^* f\,[\sigma], \sigma \in \Xi\}$$

Language Families

- ***LIN*** - the family of linear languages
- ***CF*** - the family of context-free languages
- ***RE*** - the family of recursively enumerable languages

- ***RPD(REG)*** - the family of languages accepted by PDAs regulated by regular languages
- ***RPD(LIN)*** - the family of languages accepted by PDAs regulated by linear languages

Theorem 1 and its Proof 1/2

$RPD(REG) = CF$

Proof:

I. $CF \subseteq RPD(REG)$ is clear.

II. $RPD(REG) \subseteq CF$**:**

- Let $L = L(M, \Xi)$,

  PDA (M) — **Regular language** (Ξ)

- Let $\Xi = L(G)$, G - regular grammar based on rules: $A \rightarrow aB$, $A \rightarrow a$

Theorem 1 and its Proof 2/2

Transform *M* regulated by Ξ to a *PDA N* as follows:

---

1) for every $a.Cqb \rightarrow xp$ from M and every $A \rightarrow aB$ from G, add $C\langle qA\rangle b \rightarrow x\langle pB\rangle$ to N

---

2) for every $a.Cqb \rightarrow xp$ from M and every $A \rightarrow a$ from G, add $C\langle qA\rangle b \rightarrow x\langle pf\rangle$ to N

New symbol

---

3) The set of final states in N:

$\{\langle pf\rangle: p \text{ is a final state in } M\}$

Theorem 2

RPD(*LIN*) = *RE*

Proof:

- See [**Meduna Alexander, Kolář Dušan: Regulated Pushdown Automata**, ***Acta Cybernetica***,**Vol. 2000, No. 4, p. 653-664**]

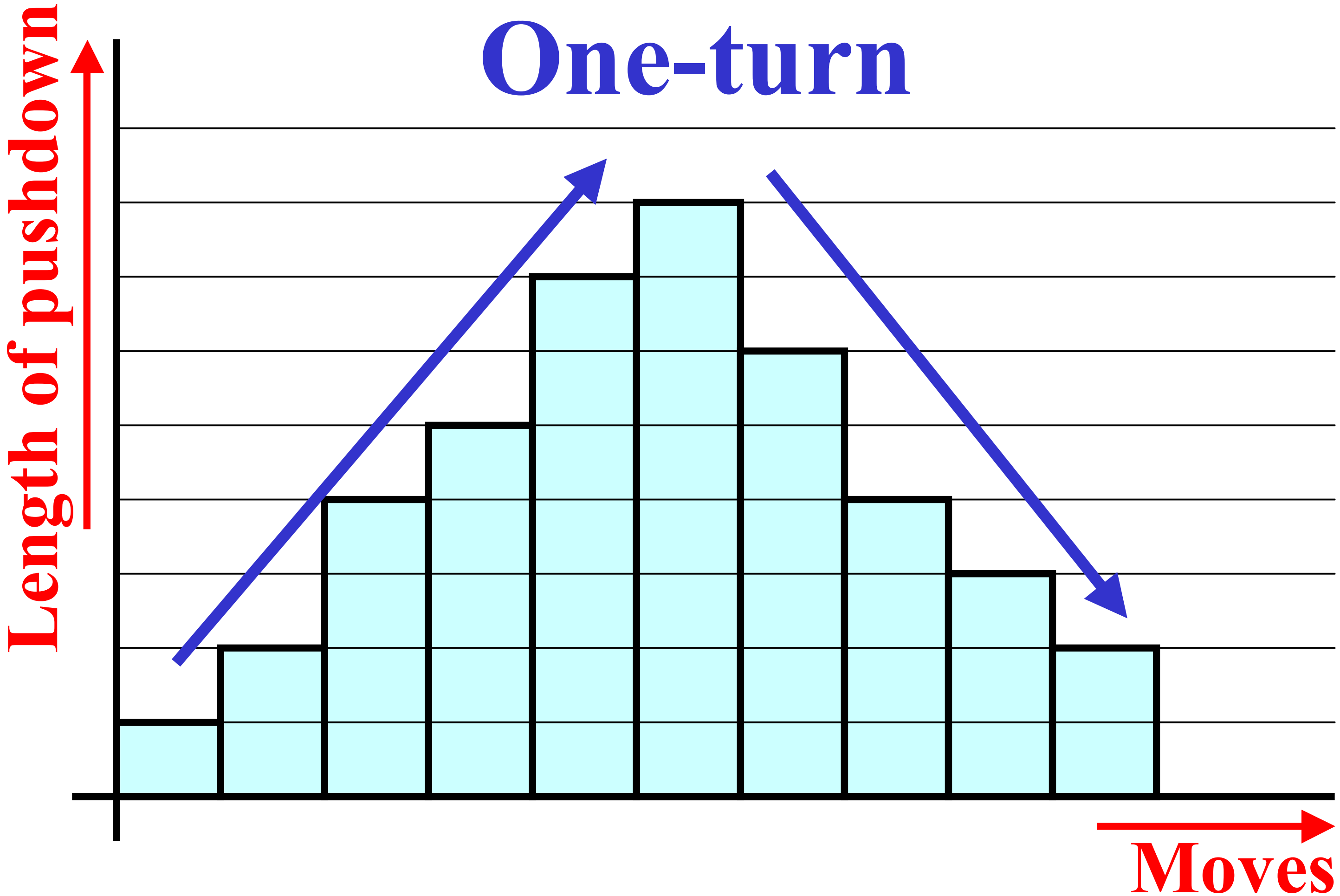
Simplification of RPDAs 1/2

I. 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 during any computation starting from an initial configuration.

One-Turn PDA: Illustration



Simplification of RPDAs 2/2

II. During a move, an *atomic* regulated PDA changes a state and, in addition, performs exactly one of the following actions:

1. pushes a symbol onto the pushdown
2. pops a symbol from the pushdown
3. reads an input symbol

Theorem 3

- **Every $L \in RE$ is accepted by an atomic one-turn PDA regulated by Ξ, where $\Xi \in LIN$.**

Proof:

- See [**Meduna Alexander, Kolář Dušan: One-Turn Regulated Pushdown Automata and Their Reduction, *Fundamenta Informatica*,Vol. 2002, No. 16, p. 399-405**]

End