# #-Rewriting Systems

## and An Infinite Hierarchy Resulting from Them

# Based upon

**Křivka, Z., Meduna, A., Schönecker, R.:**

Generation of Languages by Rewriting Systems that Resemble Automata,

In: *IJFCS* Vol. 17, No. 5, 2006

 **meduna@fit.vutbr.cz**

**Brno University of Technology, Czech Republic**

# Contents

# #-Rewriting Systems in Formal Language Theory

- Language-defining models
- Pure rewriting systems
- Between automata and grammars:
  have states but generate languages

# Concept

***#-Rewriting System*** is based on the rules of the form

$$p_m\# \rightarrow q\, x_0 \# x_1 \ldots \# x_n$$

**by which the system makes a computational step $\Rightarrow$ as**

$m$**th #**

$$\downarrow$$

$$(p, \ldots \# y_{m-1} \# y_m \# y_{m+1} \cdots) \Rightarrow$$

$$(q, \ldots \# y_{m-1} x_0 \# x_1 \ldots \# x_n y_m \# y_{m+1} \cdots)$$

# Definition 1/2

***#-Rewriting System*** (***#RS***) is a quadruple

$$H = (Q, \Sigma, s, R), \text{ where}$$

- $Q$—finite set of *states*,
- $\Sigma$—*alphabet*, $\#$ $\in \Sigma$ is called a *bounder*,
- $s \in Q$—*start state*,
- $R$—*finite set of rules* of the form

$$\boldsymbol{p}_{\boldsymbol{m}}\# \rightarrow \boldsymbol{q}\boldsymbol{x}$$

where $\boldsymbol{p}$, $\boldsymbol{q} \in Q$, $\boldsymbol{m}$ is a positive integer, $\boldsymbol{x} \in \Sigma^*$.

# Definition 2/2

***Configuration***: $(q, x)$, $q \in Q$, $x \in \Sigma^*$

---

***Computational step***:

$$(p, u\#v) \Rightarrow (q, uxv) \; [p_m\# \rightarrow qx \in R],$$

where the number of #s in $u$ is $m - 1$,

$$p, q \in Q, u, x, v \in \Sigma^*.$$

***Generated language***:

$$L(H) = \{w \in (\Sigma - \#)^* \mathbf{:} \, (s, \#) \Rightarrow^* (q, w) \text{ in } H, q \in Q\}.$$

# Example: #*RS*

**#*RS H*:**

[1]. $s\ {}_1\# \rightarrow p\ \#\#$

[2]. $p\ {}_1\# \rightarrow q\ a\#b$

[3]. $q\ {}_2\# \rightarrow p\ \#c$

[4]. $p\ {}_1\# \rightarrow f\ ab$

[5]. $f\ {}_1\# \rightarrow f\ c$

***H* accepts *aabbcc*:**

# Example: #*RS*

**#*RS H*:**

[1]. $s\ {}_1\# \rightarrow p\ \#\#$

[2]. $p\ {}_1\# \rightarrow q\ a\#b$

[3]. $q\ {}_2\# \rightarrow p\ \#c$

[4]. $p\ {}_1\# \rightarrow f\ ab$

[5]. $f\ {}_1\# \rightarrow f\ c$

***H* accepts *aabbcc*:**

$$
\begin{array}{lll}
 & (s, \#) & \\
\Rightarrow & (p, \#\#) & [1] \\
\Rightarrow & (q, a\#b\#) & [2] \\
\Rightarrow & (p, a\#bc\#) & [3] \\
\Rightarrow & (f, aabbc\#) & [4] \\
\Rightarrow & (f, aabbcc) & [5]
\end{array}
$$

$$\boxed{L(H) = \{a^n b^n c^n : n \geq 1\}}$$

# Finite index of #*RS*

#-Rewriting systems of ***index k***:

$\Rightarrow$ over configurations with $\boldsymbol{k}$ or fewer #s

$\boldsymbol{\#RS_k}$ – the language family generated by #*RS*s of index $k$

**Example**: Index $k = 2$:

1. $(p, a\#a\#b) \Rightarrow (q, aa\#aa\#b)$ $[p_1\# \rightarrow qa\#a \in R]$

**OK**

2. $(p, a\#a\#b) \cancel{\Rightarrow} (q, a\#aa\#\#bb)$ $[p_2\# \rightarrow qa\#\#b \in R]$

**INCORRECT**

# Example: #*RS* of finite index

**#*RS H*:**

[1]. $s\ _1\# \rightarrow p\ \#\#$

[2]. $p\ _1\# \rightarrow q\ a\#b$

[3]. $q\ _2\# \rightarrow p\ \#c$

[4]. $p\ _1\# \rightarrow f\ ab$

[5]. $f\ _1\# \rightarrow f\ c$

***H* accepts *aabbcc*:**

| | |
|---|---|
| $(s, \#)$ | |
| $\Rightarrow (p, \#\#)$ | [1] |
| $\Rightarrow (q, a\#b\#)$ | [2] |
| $\Rightarrow (p, a\#bc\#)$ | [3] |
| $\Rightarrow (f, aabbc\#)$ | [4] |
| $\Rightarrow (f, aabbcc)$ | [5] |

***H* is of index 2.**

$$L(H) = \{a^n b^n c^n : n \geq 1\} \in \#RS_2$$

# Main Result: An Infinite Hierarchy

**Theorem: $\#\boldsymbol{RS}_k \subset \#\boldsymbol{RS}_{k+1}$, for all $\boldsymbol{k} \geq 1$.**

---

**Proof**:

makes use of programmed grammars ($PG$) of index $k$

# Proof: Programmed Grammars

***Programmed Grammar*** (***PG***) is a modification of context-free grammar based on the rules of the form:

$$r\text{: } A \to x,\ W_r$$

- $r$: $A \to x$ is a context-free rule labeled by $r$,
- $W_r$—finite set of rule labels

***Derivation step*** ($\Rightarrow$):

after the application of rule $r$,

a rule from $W_r$ has to be applied

# Proof: Finite index of *PG*

Programmed grammars of ***index k***:

- $\Rightarrow$ over sentential forms with ***k*** or fewer occurrences of nonterminals.

$\boldsymbol{P_k}$ – the language family defined by programmed grammars of index $k$

# Example: *PG*

***PG G*:**

1: $S \rightarrow ABC$, $\{2, 5\}$

2: $A \rightarrow aA$, $\{3\}$

3: $B \rightarrow bB$, $\{4\}$

4: $C \rightarrow cC$, $\{2, 5\}$

5: $A \rightarrow a$, $\{6\}$

6: $B \rightarrow b$, $\{7\}$

7: $C \rightarrow c$, $\emptyset$

***G* generates *aabbcc*:**

# Example: *PG*

***PG G*:**

1: $S \rightarrow ABC, \{2, 5\}$

2: $A \rightarrow aA, \{3\}$

3: $B \rightarrow bB, \{4\}$

4: $C \rightarrow cC, \{2, 5\}$

5: $A \rightarrow a, \{6\}$

6: $B \rightarrow b, \{7\}$

7: $C \rightarrow c, \emptyset$

***G* generates *aabbcc*:**

| | |
|---|---|
| $S$ | |
| $\Rightarrow ABC$ | [1] |
| $\Rightarrow aABC$ | [2] |
| $\Rightarrow aAbBC$ | [3] |
| $\Rightarrow aAbBcC$ | [4] |
| $\Rightarrow aabBcC$ | [5] |
| $\Rightarrow aabbcC$ | [6] |
| $\Rightarrow aabbcc$ | [7] |

# Example: *PG*

***PG G*:**

1: $S \to ABC$, {2, 5}

2: $A \to aA$, {3}

3: $B \to bB$, {4}

4: $C \to cC$, {2, 5}

5: $A \to a$, {6}

6: $B \to b$, {7}

7: $C \to c$, $\varnothing$

***G* generates *aabbcc*:**

$$
\begin{array}{ll}
S & \\
\Rightarrow ABC & [1] \\
\Rightarrow aABC & [2] \\
\Rightarrow aAbBC & [3] \\
\Rightarrow aAbBcC & [4] \\
\Rightarrow aabBcC & [5] \\
\Rightarrow aabbcC & [6] \\
\Rightarrow aabbcc & [7]
\end{array}
$$

$$L(G) = \{a^n b^n c^n : n \geq 1\} \in P_3$$

# Proof: $\boldsymbol{P_k = \#RS_k}$, $k \geq 1$

## $\boldsymbol{P_k \subseteq \#RS_k}$:

Let $G$ be a $PG$ of index $k$. Construct a $\#RS$ $H$ of index $k$, so $H$ simulates derivation step

$$\boldsymbol{a\underline{A}bBc \Rightarrow_G adXYbBc\ [p\!: A \rightarrow dXY, \{q, o\}] \Rightarrow_G \ldots\ [q]}$$

**as**

$$\boldsymbol{(\langle \underline{A}B, p\rangle, a\underline{\#}b\#c) \Rightarrow_H (\langle XYB, q\rangle, ad\#\#b\#c)}$$

$$\boldsymbol{[\langle \underline{A}B, p\rangle\ {}_1\# \rightarrow \langle XYB, q\rangle\ d\#\#]}$$

# Proof: $\#RS_k = P_k$, $k \geq 1$

## $\#RS_k \subseteq P_k$:

Let $H$ be a $\#RS$ of index $k$. Construct a $PG$ $G$ of index $k$, so $G$ simulates a computational step

$$(p, a\#b\#c) \Rightarrow_H (q, aa\#bb\#c)\ [p_1\# \rightarrow q\ a\#b]$$

**as**

$$a\langle p, 1, 2\rangle b\langle p, 2, 2\rangle c$$

1) Renumbering: $\Rightarrow_G a\langle q'', 1, 2\rangle b\langle p, 2, 2\rangle c$

$\Rightarrow_G a\langle q'', 1, 2\rangle b\langle q', 2, 2\rangle c$

2) Rewriting: $\Rightarrow_G aa\langle q', 1, 2\rangle bb\langle q', 2, 2\rangle c$

3) Finalization: $\Rightarrow_G aa\langle q, 1, 2\rangle bb\langle q', 2, 2\rangle c$

$\Rightarrow_G aa\langle q, 1, 2\rangle bb\langle q, 2, 2\rangle c$

# Proof: $\#\boldsymbol{RS}_k \subset \#\boldsymbol{RS}_{k+1}$, $k \geq 1$

**Recall that:**

- $\boldsymbol{P}_k \subset \boldsymbol{P}_{k+1}$, for all $k \geq 1$

---

**As $\boldsymbol{P}_k = \#\boldsymbol{RS}_k$, for all $k \geq 1$, we have**

**Theorem: $\#\boldsymbol{RS}_k \subset \#\boldsymbol{RS}_{k+1}$, for all $k \geq 1$.**

---

# Future Investigation

- Determinism
- Unlimited index
- Other variants:
    - Right-linear
    - Context-sensitive
    - Parallel

**Discussion**