LIMITED CONTEXT RESTARTING AUTOMATA AND MCNAUGHTON FAMILIES OF LANGUAGES

Friedrich Otto

Peter Černo, František Mráz

Introduction

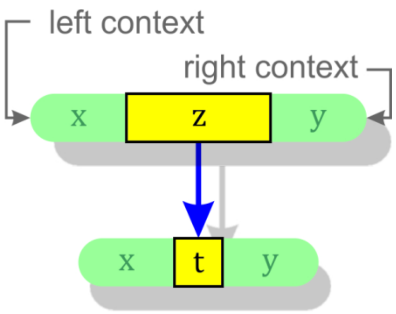
- **Part I**: Introduction,
- **Part II**: **Clearing and Δ-Clearing** Restarting Automata,
- **Part III**: **Limited Context** Restarting Automata,
- **Part IV**: **Confluent** Limited Context Restarting Automata,
- **Part V**: Concluding Remarks.

Part I: Introduction

- **Restarting Automata**:
  - Model for the linguistic technique of ***analysis by reduction***.
  - Many different types have been defined and studied intensively.
- **Analysis by Reduction**:
  - Method for checking [non-]correctness of a sentence.
  - Iterative application of simplifications.
  - Until the input cannot be simplified anymore.
- **Restricted Models**:
  - Clearing, Δ-Clearing and Δ^*-Clearing Restarting Automata,
  - Limited Context Restarting Automata.

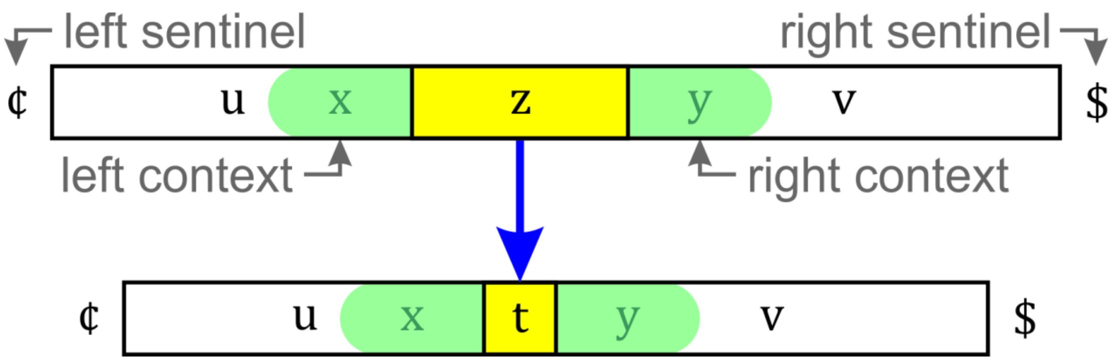
Part II: Clearing Restarting Automata

- Let k be a *nonnegative integer*.
- ***k* – context rewriting system** (*k-CRS*)
- **Is a triple** $M = (\Sigma, \Gamma, I)$:
  - Σ … ***input alphabet***, $¢, \$ \notin \Sigma$,
  - Γ … ***working alphabet***, $\Gamma \supseteq \Sigma$,
  - I … finite set of ***instructions*** $(x, z \to t, y)$:
    - $x \in \{¢, \lambda\}.\Gamma^*$, $|x| \leq k$ **(left context)**
    - $y \in \Gamma^*.\{\lambda, \$\}$, $|y| \leq k$ **(right context)**
    - $z \in \Gamma^+, z \neq t \in \Gamma^*$.
  - $¢$ and $\$$ … ***sentinels***.



Rewriting

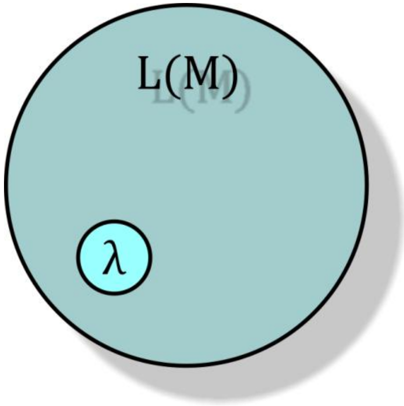
- $uzv \vdash_M utv$ **iff** $\exists\,(x, z \to t, y) \in I$:
- x is a ***suffix*** of $¢.u$ **and** y is a ***prefix*** of $v.\$$.



- $L(M) = \{w \in \Sigma^* \,/\, w \vdash^*_M \lambda\}$.
- $L_C(M) = \{w \in \Gamma^* \,/\, w \vdash^*_M \lambda\}$.

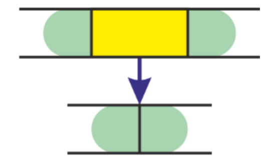
Empty Word

- **Note**: For every *k-CRS M*: $\lambda \vdash^*_M \lambda$, hence $\lambda \in L(M)$.
- Whenever we say that a *k-CRS M* ***recognizes a language*** *L*, we always mean that $L(M) = L \cup \{\lambda\}$.
- We simply ***ignore the empty word*** in this setting.

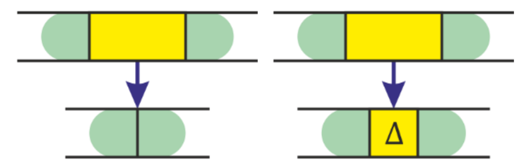


Clearing Restarting Automata

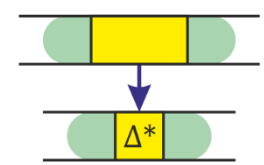
- **k – Clearing Restarting Automaton** (k-cl-RA)
  - Is a k-CRS $M = (\Sigma, \Sigma, I)$ such that:
  - For each $(x, z \to t, y) \in I$: $z \in \Sigma^+, t = \lambda$.



- **k – Δ – Clearing Restarting Automaton** (k-Δ-cl-RA)
  - Is a k-CRS $M = (\Sigma, \Gamma, I)$ such that:
  - $\Gamma = \Sigma \cup \{\Delta\}$ where Δ is a new symbol, and
  - For each $(x, z \to t, y) \in I$: $z \in \Gamma^+, t \in \{\lambda, \Delta\}$.

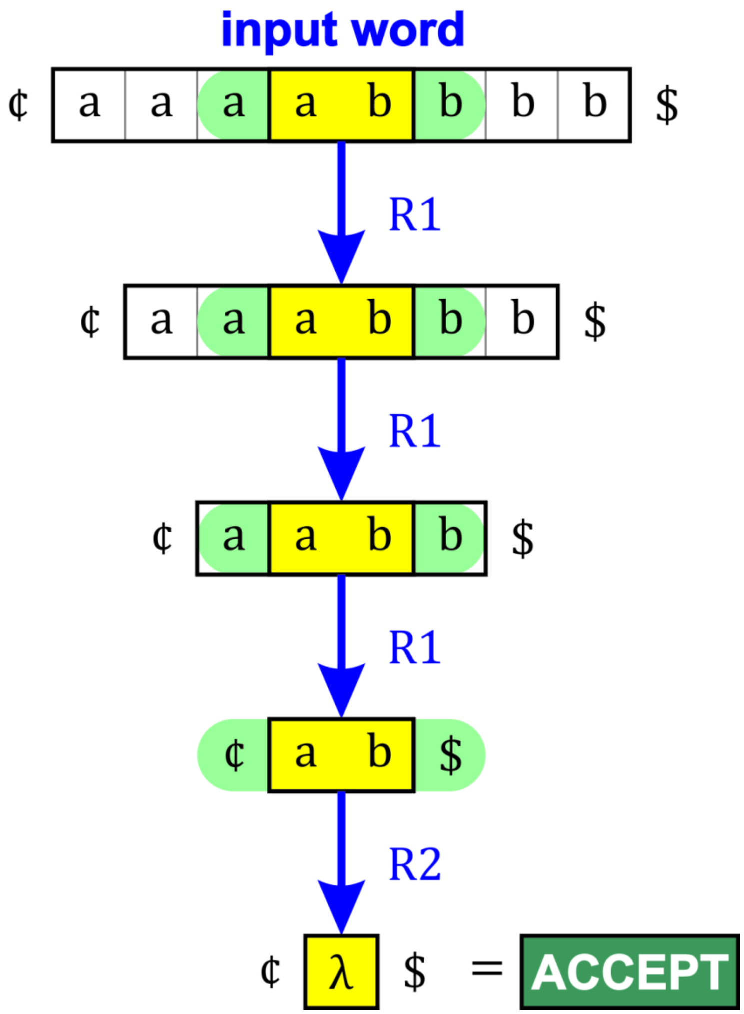


- **k – Δ^* – Clearing Restarting Automaton** (k-Δ^*-cl-RA)
  - Is a k-CRS $M = (\Sigma, \Gamma, I)$ such that:
  - $\Gamma = \Sigma \cup \{\Delta\}$ where Δ is a new symbol, and
  - For each $(x, z \to t, y) \in I$: $z \in \Gamma^+, t = \Delta^i, 0 \leq i \leq |z|$.



Example 1

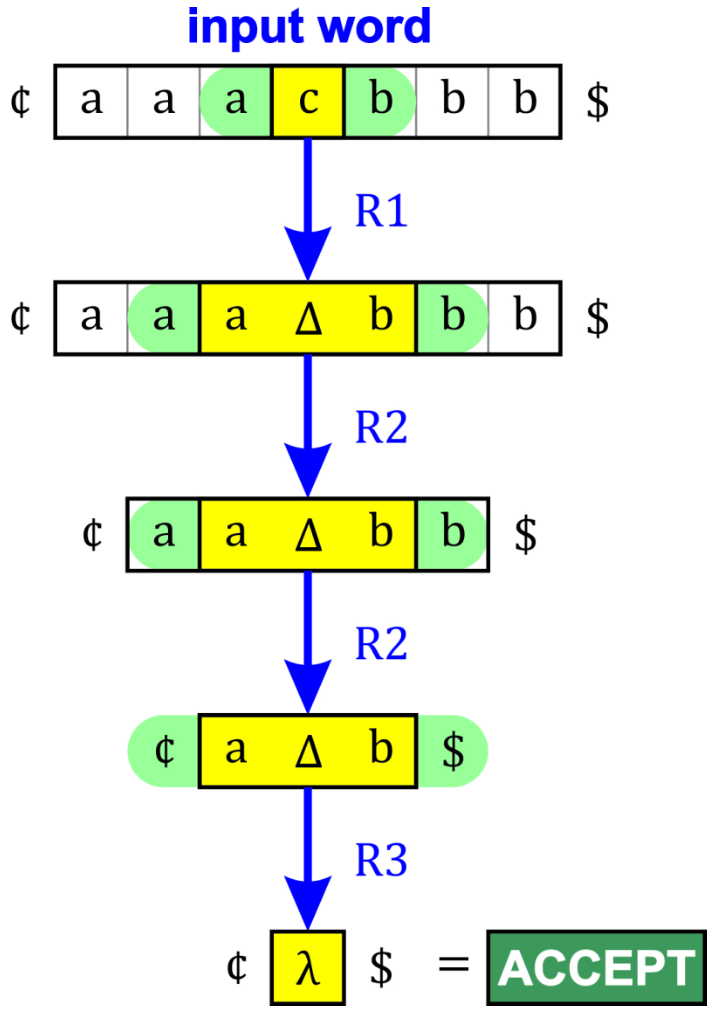
- $L_1 = \{a^n b^n \mid n > 0\} \cup \{\lambda\}$:
- 1-cl-RA $M = (\{a, b\}, I)$,
- **Instructions** I are:
  - $R1 = (a, \underline{ab} \rightarrow \lambda, b)$,
  - $R2 = (¢, \underline{ab} \rightarrow \lambda, \$)$.



- **Note**:
  - We **did not use** Δ.

Example 2

- $L_2 = \{a^n c b^n \mid n > 0\} \cup \{\lambda\}$:
- *1-Δ-cl-RA* $M = (\{a, b, c\}, I)$,
- **Instructions** I are:
  - $R1 = (a, \underline{c} \to \Delta, b)$,
  - $R2 = (a, \underline{a\Delta b} \to \Delta, b)$,
  - $R3 = (¢, \underline{a\Delta b} \to \lambda, \$)$.



- **Note**:
  - We **must use Δ**.

Clearing Restarting Automata

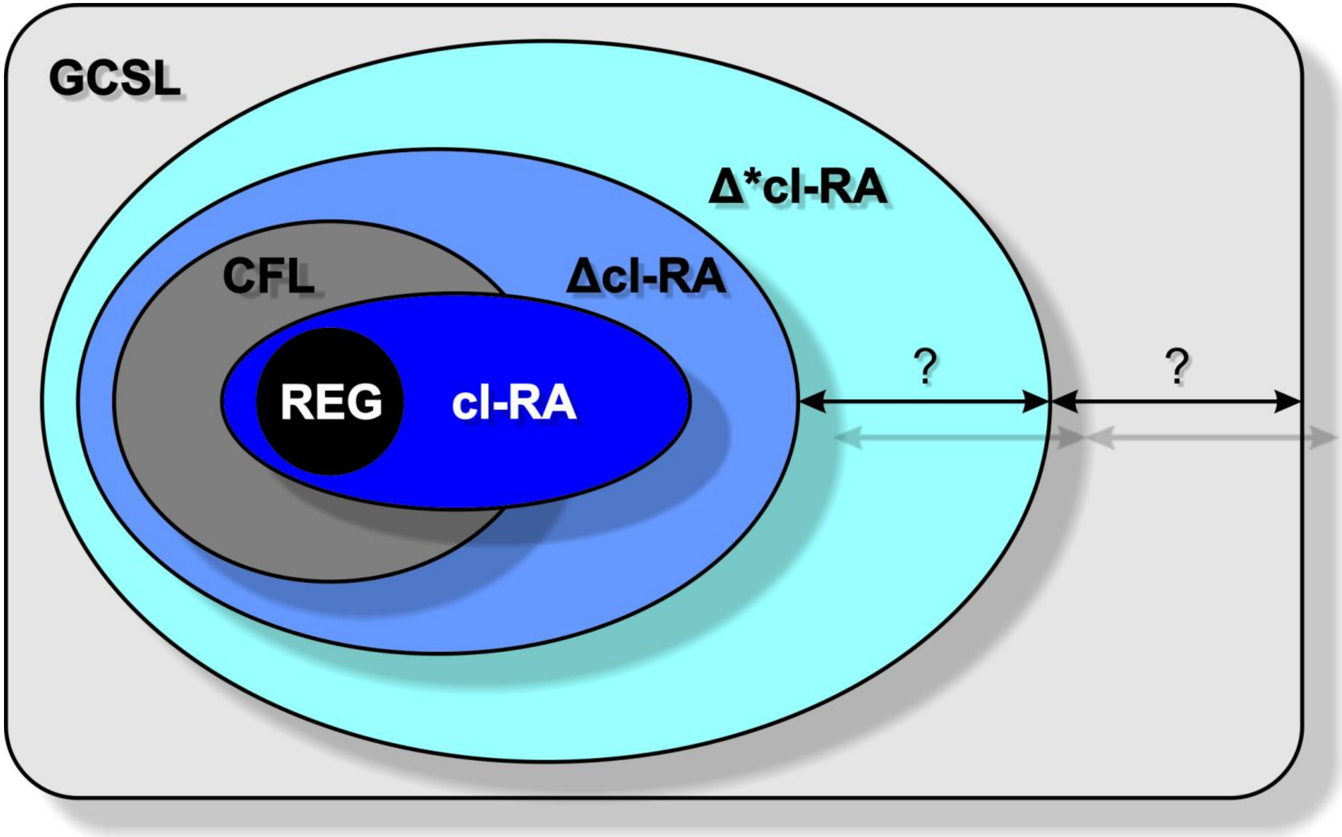
- **Clearing Restarting Automata**:
  - Accept **all regular** and even **some non-context-free** languages.
  - They do **not** accept **all context-free** languages ($\{a^n c b^n \mid n > 0\}$).
- **Δ-Clearing and Δ*-Clearing Restarting Automata**:
  - Accept **all context-free** languages.
  - The exact expressive power remains open.
- Here we establish an **upper bound** by showing that **Clearing**, **Δ-** and **Δ*-Clearing Restarting Automata** only accept languages that are **growing context-sensitive** *[Dahlhaus, Warmuth]*.

Clearing Restarting Automata

- **Theorem**: $\mathcal{L}(\Delta^*\text{-}cl\text{-}RA) \subseteq GCSL$.
- **Proof.**
  - Let $M = (\Sigma, \Gamma, I)$ be a $k\text{-}\Delta^*\text{-}cl\text{-}RA$ for some $k \geq 0$.
  - Let $\Omega = \Gamma \cup \{¢, \$, Y\}$, where Y is a *new letter*.
  - Let $S(M)$ be the following ***string-rewriting system*** over Ω:

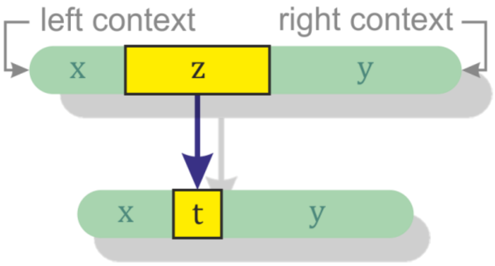
    $$S(M) = \{ xzy \rightarrow xty \mid (x, z \rightarrow t, y) \in I \} \cup \{ ¢\$ \rightarrow Y \}.$$
  - Let g be a ***weight function***: $g(\Delta) = 1$ and $g(a) = 2$ for all $a \neq \Delta$.
- **Claim**: $L(M)$ coincides with the ***McNaughton language*** [Beaudry, Holzer, Niemann, Otto] specified by $(S(M), ¢, \$, Y)$.
- As $S(M)$ is a ***finite weight-reducing system***, it follows that the ***McNaughton language*** $L(M)$ is a ***growing context-sensitive language***, that is, $L(M) \in GCSL$. ■

Clearing Restarting Automata



Part III: Limited Context RA

- **Limited Context Restarting Automaton** (*lc-RA*):
  - Is defined exactly as ***Context Rewriting Systems***, except that:
  - There is ***no upper bound*** k **on the length of contexts**.
  - The **instructions** are usually written as: $(x / z \to t / y)$.
  - There is a ***weight function*** g such that $g(z) > g(t)$ for all instructions $(x / z \to t / y)$ of the automaton.



Limited Context Restarting Automata

- **Restricted types**: *lc-RA* $M = (\Sigma, \Gamma, I)$ is of **type**:
  - $\mathcal{R}_0'$, if I is an arbitrary finite set of (***weight-reducing***) instructions,
  - $\mathcal{R}_1'$, if $|t| \leq 1$,
  - $\mathcal{R}_2'$, if $|t| \leq 1$, $x \in \{¢, \lambda\}$, $y \in \{\lambda, \$\}$,
  - $\mathcal{R}_3'$, if $|t| \leq 1$, $x \in \{¢, \lambda\}$, $y = \$$,

    for all $(x / z \rightarrow t / y) \in I$.
- **Moreover**, *lc-RA* $M = (\Sigma, \Gamma, I)$ is of **type**:
  - $\mathcal{R}_0$, ($\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}_3$, respectively) if it is of type:
  - $\mathcal{R}_0'$, ($\mathcal{R}_1'$, $\mathcal{R}_2'$, $\mathcal{R}_3'$, respectively) and all instructions of M are ***length-reducing*** (i.e. $|z| > |t|$ for all $(x / z \rightarrow t / y) \in I$).
- We use the **notation** *lc-RA*$[\mathcal{R}_i']$, *lc-RA*$[\mathcal{R}_i]$ to denote the corresponding **class** of the **restricted** *lc-RA* s.

$lc\text{-}RA[\mathcal{R}_0']$ and $lc\text{-}RA[\mathcal{R}_0]$

- **Theorem**: $\mathcal{L}(lc\text{-}RA[\mathcal{R}_0']) = \mathcal{L}(lc\text{-}RA[\mathcal{R}_0]) = GCSL$.
- **Proof**.
  - For each $lc\text{-}RA\ M = (\Sigma, \Gamma, I)$ we can **associate** a ***finite weight-reducing string-rewriting system*** $S(M)$ such that $L(M)$ is the ***McNaughton language*** specified by the four-tuple $(S(M), ¢, \$, Y)$.

$$S(M) = \{\, xzy \rightarrow xty \mid (x \mid z \rightarrow t \mid y) \in I \,\} \cup \{\, ¢\$ \rightarrow Y \,\}.$$

  - **It follows that** $L(M) \in GCSL$.
  - **On the other hand**, **each** growing context-sensitive language is accepted by an $lc\text{-}RA[\mathcal{R}_0]$. ■

$lc\text{-}RA[\mathcal{R}_1']$

- **Theorem**: $\mathcal{L}(lc\text{-}RA[\mathcal{R}_1']) = GCSL$.
- **Proof**.
  - Let $G = (N, T, S, P)$ be a ***weight-increasing context-sensitive grammar***. By taking:
  - $I(G) \quad = \quad \{ (u \mid x \rightarrow A \mid v) \mid (uAv \rightarrow uxv) \in P \} \cup$
    $\{ (¢ \mid r \rightarrow \lambda \mid \$) \mid (S \rightarrow r) \in P \}$,
  - we obtain an $lc\text{-}RA[\mathcal{R}_1']$ $M(G) = (T, N \cup T, I(G))$ such that
  - $L(M(G)) = L(G) \cup \{\lambda\}$.
  - The class of languages generated by ***weight-increasing*** *context-sensitive grammars*, which is known as the class $ACSL$ ***(acyclic context-sensitive languages)***, coincides with the class $GCSL$ *[Niemann, Woinowski]*.
  - Thus, $\mathcal{L}(lc\text{-}RA[\mathcal{R}_1']) \supseteq GCSL$. ■

lc-RA$[\mathcal{R}_1]$

- **Theorem**: $\mathcal{L}(lc\text{-}RA[\mathcal{R}_1]) = GACSL$.
- **Proof**.
  - Let $lc\text{-}RA\ M = (\Sigma, \Gamma, I)$ be of type $\mathcal{R}_1$.
    - For all $(x\,/\,z \to t\,/\,y) \in I$: $|z| > |t|$ and $|t| \leq 1$.
  - **Lemma**: It is possible to obtain an ***equivalent*** $lc\text{-}RA\ M$ such that:
    - For all $(x\,/\,z \to t\,/\,y) \in I$: $|z| > |t|$ and $|t| = 1$ **if** $x \neq ¢$ or $y \neq \$$.
  - From ***string-rewriting system***: $R = \{\, xty \to xzy \mid (x\,/\,z \to t\,/\,y) \in I \,\}$,
  - We construct a ***length-increasing context-sensitive grammar*** :
  - $G = (\Gamma, \Sigma, S, R)$ such that $L(G) = ¢ . L(M) . \$$.
  - The class of languages generated by ***length-increasing context-sensitive grammars*** is known as the class $GACSL$ (***growing acyclic context-sensitive languages***). $GACSL \subseteq ACSL = GCSL$.
  - $¢ . L(M) . \$ \in GACSL$, i.e. $L(M) \in GACSL$ *[Buntrock]*. **Similarly** $\supseteq$. ■

lc-RA[$\mathcal{R}_2$'] and *lc-RA*[$\mathcal{R}_2$]

- **Theorem**: $\mathcal{L}$(*lc-RA*[$\mathcal{R}_2$']) = $\mathcal{L}$(*lc-RA*[$\mathcal{R}_2$]) = *CFL*.
- **Proof**.
  - Let *lc-RA M = (Σ, Γ, I)* be of type $\mathcal{R}_2$'.
  - For all $(x \,/\, z \to t \,/\, y) \in I$: $|t| \leq 1$, $x \in \{¢, \lambda\}$, $y \in \{\lambda, \$\}$.
  - We split $R(M) = \{ xzy \to xty \mid (x \,/\, z \to t \,/\, y) \in I \}$ into ***4 subsystems***:

(a) $R_{bif} = \{ ¢x\$ \to ¢y\$ \mid (¢ \mid x \to y \mid \$) \in I \}$, the *bifix rules* of $R(M)$,
(b) $R_{pre} = \{ ¢x \to ¢y \mid (¢ \mid x \to y \mid \lambda) \in I \}$, the *prefix rules* of $R(M)$,
(c) $R_{suf} = \{ x\$ \to y\$ \mid (\lambda \mid x \to y \mid \$) \in I \}$, the *suffix rules* of $R(M)$,
(d) $R_{inf} = \{ x \to y \mid (\lambda \mid x \to y \mid \lambda) \in I \}$, the *infix rules* of $R(M)$.

  - Take $A(M) = \{ \alpha \in \Gamma^* \mid ¢\alpha\$ \in \mathrm{dom}(R_{bif}) \text{ and } ¢\alpha\$ \Rightarrow^*_{R(M)} ¢\$ \}$
  - Then ***A(M)*** is a ***finite set***. Let $R' = R_{pre} \cup R_{suf} \cup R_{inf}$. Then ***L(M) =***

$\{ w \in \Sigma^* \mid ¢w\$ \Rightarrow^*_{R(M)} ¢\$ \} = \{ w \in \Sigma^* \mid \exists \alpha \in A(M) \cup \{\lambda\} : ¢w\$ \Rightarrow^*_{R'} ¢\alpha\$ \}$

lc-RA[$\mathcal{R}_2$'] and *lc-RA*[$\mathcal{R}_2$]

- **Proof**. (Continued).
  - Consider a ***mixed rewriting system***: $P(M) = P_{pre} \cup P_{suf} \cup P_{inf}$
  - *Prefix-rewriting system*: $P_{pre} = \{ x \to y \mid (¢x \to ¢y) \in R_{pre} \}$
  - *Suffix-rewriting system*: $P_{suf} = \{ x \to y \mid (x\$ \to y\$) \in R_{suf} \}$
  - *String-rewriting system*: $P_{inf} = R_{inf}$
  - The **rules** of a ***prefix-rewriting system*** (***suffix-rewriting system***) **are only applied to** the ***prefix*** (***suffix***) of a word.
  - **Apparently**: $L(M) = \nabla^*_{P(M)}(A(M) \cup \{\lambda\}) \cap \Sigma^*$
  - As *P(M)* only contains ***generalized monadic rules***, it follows that the language *L(M)* is ***context-free*** *[Leupold, Otto]*.
  - Moreover, it is easy to obtain from a given ***context-free grammar*** an equivalent ***lc-RA M = (Σ, Γ, I)*** of the type $\mathcal{R}_2$.
  - Thus we have: $CFL \subseteq \mathcal{L}(lc\text{-}RA[\mathcal{R}_2]) \subseteq \mathcal{L}(lc\text{-}RA[\mathcal{R}_2']) \subseteq CFL$. ■

$lc\text{-}RA[\mathcal{R}_3']$ and $lc\text{-}RA[\mathcal{R}_3]$

- **Theorem**: $\mathcal{L}(lc\text{-}RA[\mathcal{R}_3']) = \mathcal{L}(lc\text{-}RA[\mathcal{R}_3]) = REG$.
- **Proof.**
  - Let $lc\text{-}RA\ M = (\Sigma, \Gamma, I)$ be of type $\mathcal{R}_3'$.
  - For all $(x\,/\,z \to t\,/\,y) \in I$: $|t| \leq 1$, $x \in \{¢, \lambda\}$, $y = \$$.
  - We split $R(M) = \{\, xzy \to xty \mid (x\,/\,z \to t\,/\,y) \in I \,\}$ into **2 subsystems**:

(a) $R_{bif} = \{\, ¢x\$ \to ¢y\$ \mid (¢ \mid x \to y \mid \$) \in I \,\}$, the *bifix rules* of $R(M)$,
(b) $R_{suf} = \{\, x\$ \to y\$ \mid (\lambda \mid x \to y \mid \$) \in I \,\}$, the *suffix rules* of $R(M)$.

  - Now we take only the *suffix-rewriting system* $P(M) = P_{suf}$, where:
  - $P_{suf} = \{\, y \to x \mid (x\$ \to y\$) \in R_{suf} \,\}$
  - **Apparently**: $L(M) = \Delta^*_{P(M)}(A(M) \cup \{\lambda\}) \cap \Sigma^*$ is ***regular***.
  - Again, it is easy to obtain from a given ***regular grammar*** an equivalent $lc\text{-}RA\ M = (\Sigma, \Gamma, I)$ of the type $\mathcal{R}_3$.
  - Thus we have: $REG \subseteq \mathcal{L}(lc\text{-}RA[\mathcal{R}_3]) \subseteq \mathcal{L}(lc\text{-}RA[\mathcal{R}_3']) \subseteq REG$. ■

Limited Context Restarting Automata

- **Hierarchy of Language Classes**:

$$
\begin{array}{ccccc}
\mathcal{L}(\text{lc-R}[\mathcal{R}_0]) = & \mathcal{L}(\text{lc-R}[\mathcal{R}'_0]) & = & \text{GCSL} \\
 & \| & & \uparrow ? \\
 & \mathcal{L}(\text{lc-R}[\mathcal{R}'_1]) & & \\
 & \uparrow ? & & \\
 & \mathcal{L}(\text{lc-R}[\mathcal{R}_1]) & = & \text{GACSL} \\
 & & & \uparrow \\
\mathcal{L}(\text{lc-R}[\mathcal{R}_2]) = & \mathcal{L}(\text{lc-R}[\mathcal{R}'_2]) & = & \text{CFL} \\
 & & & \uparrow \\
\mathcal{L}(\text{lc-R}[\mathcal{R}_3]) = & \mathcal{L}(\text{lc-R}[\mathcal{R}'_3]) & = & \text{REG}
\end{array}
$$

Part IV: Confluent lc-RA

- Since *lc-RA M* is a ***nondeterministic*** device, it is ***difficult*** to decide the membership in $L(M)$.
- Here we are interested in *lc-RA* $M = (\Sigma, \Gamma, I)$ for which ***all computations*** from $¢\, w\, \$$ lead to $¢\, \$$, **if** $w \in L(M)$.
- The *reduction relation* $\vdash_M$ corresponds to the *single-step reduction relation* $\Rightarrow_{R(M)}$ induced by the *string-rewriting system* $R(M) = \{ xzy \rightarrow xty \mid (x \mid z \rightarrow t \mid y) \in I \}$ on $¢\, \Gamma^*\, \$$.
- As it is ***undecideable*** whether $R(M)$ is confluent on the congruence class $[¢\, \$]_{R(M)}$, we consider only ***confluence***.
- An *lc-RA* $M = (\Sigma, \Gamma, I)$ is called ***confluent*** if the corresponding *string-rewriting system* $R(M)$ is ***confluent***.
- We use the ***prefix*** ***con-*** to denote ***confluent*** ***lc-RA***.

$lc\text{-}RA[con\text{-}\mathcal{R}_0']$ and $lc\text{-}RA[con\text{-}\mathcal{R}_0]$

- **Theorem**: $\mathcal{L}(lc\text{-}RA[con\text{-}\mathcal{R}_0']) = \mathcal{L}(lc\text{-}RA[con\text{-}\mathcal{R}_0]) = CRL$.
- **Proof**.
  - For each $lc\text{-}RA[con\text{-}\mathcal{R}_0']$ $M = (\Sigma, \Gamma, I)$ **:** $S(M) = R(M) \cup \{ ¢\$ \rightarrow Y \}$ is a ***finite weight-reducing string-rewriting system*** that is ***confluent***.
  - $L(M)$ is the **McNaughton language** specified by $(S(M), ¢, \$, Y)$, i.e.
  - $L(M)$ is a ***Church-Rosser language*** *[McNaughton, Narendran, Otto]*.
  - **On the other hand**, **each** ***Church-Rosser language*** L is accepted by a ***length-reducing deterministic two-pushdown automaton*** A *[Niemann, Otto]*.
  - Based on A it is possible to construct a ***confluent*** $lc\text{-}RA$ of type $\mathcal{R}_0$ recognizing the language L. ■

$lc\text{-}RA[con\text{-}\mathcal{R}_3']$ and $lc\text{-}RA[con\text{-}\mathcal{R}_3]$

- **Theorem**: $\mathcal{L}(lc\text{-}RA[con\text{-}\mathcal{R}_3']) = \mathcal{L}(lc\text{-}RA[con\text{-}\mathcal{R}_3]) = REG$.
- **Proof**.
  - **Apparently**, $\mathcal{L}(lc\text{-}RA[con\text{-}\mathcal{R}_3']) \subseteq \mathcal{L}(lc\text{-}RA[\mathcal{R}_3']) = REG$.
  - **Conversely**, if $L \subseteq \Sigma^*$ is ***regular*** then there exists *DFA* $A = (Q, \Sigma, q_0, F, \delta)$ that accepts L^R. We define *lc-RA* $M = (\Sigma, \Sigma \cup Q, I)$, where $I =$

$$\{ (¢ \mid ab \to q \mid \lambda) \mid \delta(q_0, ab) = q \} \cup \{ (¢ \mid qa \to q' \mid \lambda) \mid \delta(q, a) = q' \} \cup$$
$$\{ (¢ \mid q \to \lambda \mid \$) \mid q \in F \} \cup \{ (¢ \mid a \to \lambda \mid \$) \mid a \in \Sigma \cap L^R \}.$$

  - **It is easy to see** that $L(M) = L^R$, and that the ***string-rewriting system*** $R(M)$ is ***confluent***. By taking $M' = (\Sigma, \Sigma \cup Q, I')$, where:

$$I' = \{ (\lambda \mid u^R \to v^R \mid \$) \mid (¢ \mid u \to v \mid \lambda) \in I \} \cup$$
$$\{ (¢ \mid u^R \to v^R \mid \$) \mid (¢ \mid u \to v \mid \$) \in I \},$$

  - We obtain a ***confluent lc-RA*** of type $\mathcal{R}_3$ that accepts L. ■

lc-RA[con-$\mathcal{R}_2$'] and *lc-RA[con-$\mathcal{R}_2$]*

- For other classes we have ***no characterization results***.
- We have only some ***preliminary results***.
- **<u>Lemma</u>**: *$\mathcal{L}$(lc-RA[con-$\mathcal{R}_2$']) ⊆ DCFL ∩ DCFL^R*.
- **Proof Idea**.
  - Consider the ***leftmost derivation***, which can be realized by a ***deterministic pushdown automaton***. ■
- **<u>Lemma</u>**: The ***deterministic context-free language***

$$L_u = \{ ca^n b^n c \mid n \geq 1 \} \cup \{ da^m b^{2m} d \mid m \geq 1 \}$$

- Is ***not accepted*** by ***any*** *lc-RA[con-$\mathcal{R}_2$']*.
- **<u>Note</u>**: Both L_u and L_u^R are ***DLIN* languages**.
- **<u>Corollary</u>**: *$\mathcal{L}$(lc-RA[con-$\mathcal{R}_2$']) ⊂ DCFL ∩ DCFL^R*.

lc-RA[con-$\mathcal{R}_2$'] and *lc-RA[con-$\mathcal{R}_2$]*

- **Lemma**: The ***nonlinear language*** $\{ a^n b^n c^m d^m \mid n, m \geq 1 \}$ is accepted by a ***confluent*** *lc-RA* ***of type*** $\mathcal{R}_2$.
- **Corollary**: The class of languages accepted by ***confluent*** *lc-RA* ***of type*** $\mathcal{R}_2$' is **incomparable** to *DLIN* and *LIN*.
- These results also hold for the class of languages that are accepted by *lc-RA[con-$\mathcal{R}_2$]*.
- The exact relationship of these classes of languages to the class of confluent *[generalized]* monadic McNaughton languages *[Leupold, Otto]* remains open.

lc-RA[con-$\mathcal{R}_1$'] and *lc-RA[con-$\mathcal{R}_1$]*

- **Lemma**: The language $L_{expo5} = \{ a^{5^n} \mid n \geq 0 \}$ is accepted by an *lc-RA[con-$\mathcal{R}_1$]*.
- **Proof**. Take $\Sigma = \{a\}$, $\Gamma = \{a, b, A, B, C, D\}$, and $M = (\Sigma, \Gamma, I)$, where I:

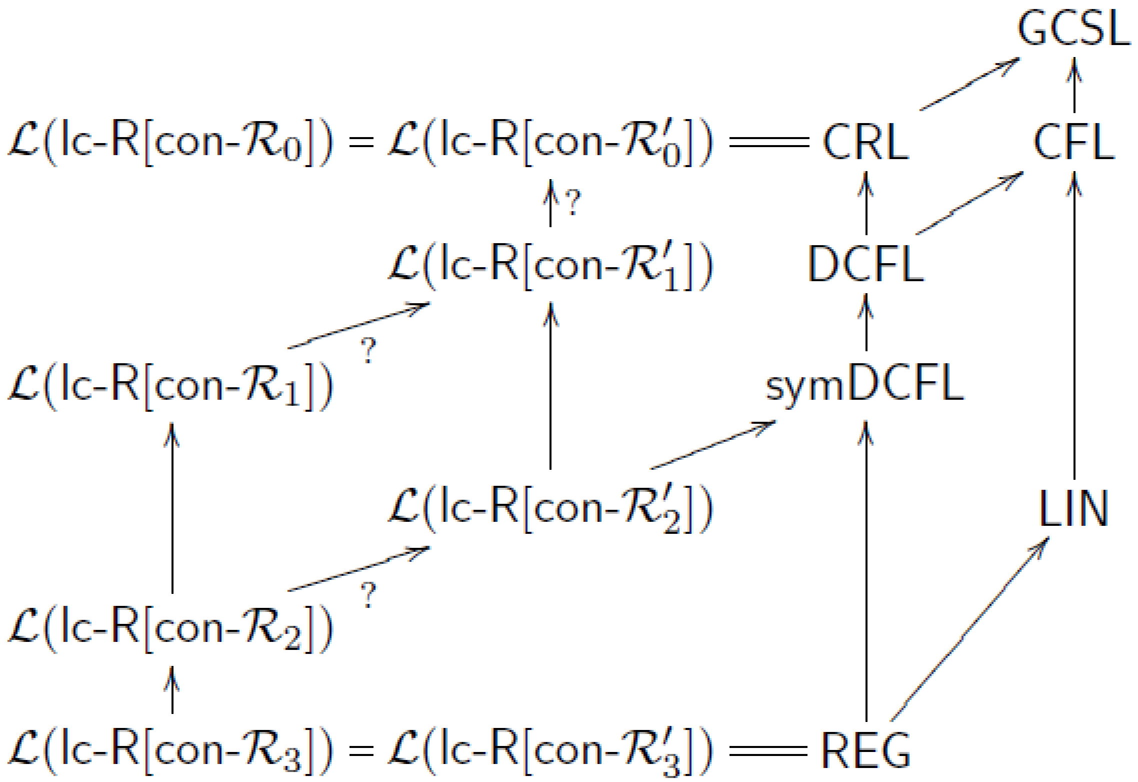
	I	**R**(M)		
(1)	$(¢a^2 \mid a^2 \to B \mid a)$	$¢a^5$	$\to$	$¢a^2Ba$
(2)	$(¢ \mid a^2 \to b \mid Ba)$	$¢a^2Ba$	$\to$	$¢bBa$
(3)	$(Aa^2 \mid a^2 \to B \mid a)$	Aa^5	$\to$	Aa^2Ba
(4)	$(\lambda \mid Aa^2 \to b \mid Ba)$	Aa^2Ba	$\to$	bBa
(5)	$(b \mid Ba \to A \mid \lambda)$	bBa	$\to$	bA
(6)	$(\lambda \mid A \to \lambda \mid \$)$	$A\$$	$\to$	$\$$
(7)	$(\lambda \mid b^2 \to C \mid b^3\$)$	$b^5\$$	$\to$	$Cb^3\$$
(8)	$(Cb \mid b^2 \to a \mid \$)$	$Cb^3\$$	$\to$	$Cba\$$
(9)	$(\lambda \mid Cb \to D \mid a)$	Cba	$\to$	Da
(10)	$(\lambda \mid b^2 \to C \mid b^3D)$	b^5D	$\to$	Cb^3D
(11)	$(Cb \mid b^2D \to a \mid \lambda)$	Cb^3D	$\to$	Cba
(12)	$(¢ \mid D \to \lambda \mid \lambda)$	$¢D$	$\to$	$¢$
(13)	$(¢ \mid a \to \lambda \mid \$)$	$¢a\$$	$\to$	$¢\$$
(14)	$(¢ \mid b \to \lambda \mid \$)$	$¢b\$$	$\to$	$¢\$$

$lc\text{-}RA[con\text{-}\mathcal{R}_1']$ and $lc\text{-}RA[con\text{-}\mathcal{R}_1]$

- As the language L_{expo5} is ***not context-free***, we obtain:
- **Corollary**: The class of languages accepted by ***confluent*** $lc\text{-}RA$ ***of type*** $\mathcal{R}_1$ is **incomparable** to *CFL*.
- **In particular**, $lc\text{-}RA[con\text{-}\mathcal{R}_1] \supset lc\text{-}RA[con\text{-}\mathcal{R}_2]$.
- These results also hold for the class of languages that are accepted by $lc\text{-}RA[con\text{-}\mathcal{R}_1']$.

Confluent lc-RA

- **Hierarchy of Language Classes**:



Part V: Concluding Remarks

- The class ***GCSL*** forms an ***upper bound*** for ***all types*** of limited context restarting automata considered.
- Under the additional requirement of ***confluence***, the ***Church-Rosser languages*** form an ***upper bound***.
- For the ***most restricted types*** of *lc-RA* we obtain ***regular languages***, both in *confluent* and *non-confluent* case.
- For the ***intermediate systems***, the question for an exact characterization of the corresponding classes of languages ***remains open***.
- For the ***intermediate systems*** it even ***remains open*** whether the ***weight-reducing lc-RA*** are more expressive than the corresponding ***length reducing lc-RA***.

References

- **BASOVNÍK, Learning restricted restarting automata using genetic algorithm**.
  - Master's thesis, Charles University, MFF, Prague, 2010.
- **BASOVNÍK, MRÁZ, Learning limited context restarting automata by genetic algorithms**.
  - In: J. DASSOW, B. TRUTHE (eds.), Theorietag 2011. Otto-von-Guericke-Universität, Magdeburg, 2011, 1-4.
- **BEAUDRY, HOLZER, NIEMANN, OTTO, McNaughton families of languages**.
  - Theoret. Comput. Sci. 290 (2003), 1581-1628.
- **BOOK, OTTO, String-Rewriting Systems**. Springer, New York, 1993.
- **BÜCHI, Regular canonical systems.**
  - Arch. f. Math. Logik Grundlagenf. 6 (1964), 91-111.
- **G. BUNTROCK, Wachsende kontext-sensitive Sprachen.**
  - Habilitationsschrift, Fakultät für Mathematik und Informatik, Universität at Würzburg, 1996.
- **BUNTROCK, OTTO, Growing context-sensitive languages and Church-Rosser languages**.
  - Inform. and Comput. 141 (1998), 1-36.
- **ČERNO, MRÁZ, Clearing restarting automata**. Fund. Inf. 104 (2010), 17-54.
- **ČERNO, MRÁZ, Δ-clearing restarting automata and CFL**.
  - In: G. MAURI, A. LEPORATI (eds.), DLT 2011. LNCS 6795, Springer, Berlin, 2011, 153-164.
- **DAHLHAUS, WARMUTH, Membership for growing context-sensitive grammars is polynomial**.
  - J. Comput. System Sci. 33 (1986), 456-472.
- **HOFBAUER, WALDMANN, Deleting string rewriting systems preserve regularity**.
  - Theoret. Comput. Sci. 327 (2004), 301-317.
- **JANČAR, MRÁZ, PLÁTEK, VÖGEL, Restarting automata**.
  - In: H. REICHEL (ed.), FCT'95. LNCS 965, Springer, Berlin, 1995, 283-292.
- **LEUPOLD, OTTO, On McNaughton families of languages that are specified by some variants of monadic string-rewriting systems**. Fund. Inf. 112 (2011), 219-238.
- **MCNAUGHTON, NARENDRAN, OTTO, Church-Rosser Thue systems and formal languages**.
  - J. Assoc. Comput. Mach. 35 (1988), 324-344.
- **NIEMANN, OTTO, The Church-Rosser languages are the deterministic variants of the growing context-sensitive languages**.
  - Inform. and Comput. 197 (2005), 1-21.
- **NIEMANN, WOINOWSKI, The growing context-sensitive languages are the acyclic context-sensitive languages**.
  - In: W. KUICH, G. ROZENBERG, A. SALOMAA (eds.), DLT 2002 . LNCS 2295, Springer, Berlin, 2002, 197-205.
- **OTTO, On deciding the congruence of a finite string-rewriting system on a given congruence class**.
  - J. Comput. System Sci. 35 (1987), 285-310.
- **OTTO, Restarting automata.**
  - In: Z. ÉSIK, C. MARTIN-VIDE, V. MITRANA (eds.), Recent Advances in Formal Languages and Applications. Studies in Computational Intelligence 25, Springer, Berlin, 2006, 269-303.

Thank You!

- This presentation is available on the following website:
  http://popelka.ms.mff.cuni.cz/cerno/files/otto_cerno_mraz_lcra_presentation.pdf