Peter Černo

Incorporating Delta Rules

About

- Suppose we have a **sample computation** for (delta) **clearing restarting automata**.
- Suppose that the inferred automaton accepts some **wrong words**.
- There are two ways how to **restrict** the resulting **inferred language**:
  - We can **increase** k – the **length of contexts**.
  - We can **change the sample computation** by **incorporating** some **delta rules**.

Example 1

- Consider the following **sample computation**:

  $¢\ ababababababab\underline{a}b\ \$ \vdash_M ¢\ abababababab\underline{a}babb\ \$ \vdash_M$

  $¢\ abababa\underline{a}babbabb\ \$ \vdash_M ¢\ ab\underline{a}babbabbabb\ \$ \vdash_M$

  $¢\ abbabbabbab\underline{b}b\ \$ \vdash_M ¢\ abbabbab\underline{b}bab\ \$ \vdash_M$

  $¢\ abbab\underline{b}babab\ \$ \vdash_M ¢\ a\underline{b}bababab\ \$ \vdash_M$

  $¢\ abababa\underline{a}b\ \$ \vdash_M ¢\ ab\underline{a}babb\ \$ \vdash_M$

  $¢\ abbab\underline{b}b\ \$ \vdash_M ¢\ a\underline{b}bab\ \$ \vdash_M$

  $¢\ ab\underline{a}b\ \$ \vdash_M ¢\ a\underline{b}b\ \$ \vdash_M ¢\ \underline{ab}\ \$ \vdash_M ¢\ \lambda\ \$$ ***accept***.

- We can collect **15 reductions**.

Example 1

- For $k = 4$ we get the following **instructions**:
  - $(\{¢ab, abab\}, \underline{a}, \{b\$, babb\})$,
  - $(\{¢a, abba\}, \underline{b}, \{b\$, bab\$, baba\})$,
  - $(¢, \underline{ab}, \$)$.
- For the resulting **4-cl-RA-automaton M** the following holds:
  $L(M) \cap \{(ab)^n \mid n>0\} = \{(ab)^{2^m} \mid m \geq 0\}$.
- However, this **does not work** for **smaller k**.

Example 1

- For $k = 3$ we get the following **instructions**:
  - $(\{¢ab, bab\}, \underline{a}, \{b\$, bab\})$,
  - $(\{¢a, bba\}, \underline{b}, \{b\$, bab\})$,
  - $(¢, \underline{ab}, \$)$.
- Now the resulting ***3-cl-RA*-automaton *M*** accepts the **wrong** word ***ababab*** :
  - $ab\underline{a}bab \vdash_M ab\underline{b}ab \vdash_M ab\underline{a}b \vdash_M a\underline{b}b \vdash_M \underline{ab} \vdash_M \lambda$.
- We **blame the instruction** $(¢ab, \underline{a}, bab)$ which **caused the reduction**: $ab\underline{a}bab \vdash_M abbab$.

Idea

- **Why?** The instruction reduced the **wrong** word ***ababab*** to the **correct** word ***abbab***.
  - ***abbab*** is **ok** – it is in the **sample computation**.
- **Idea**: Let us replace the instruction:
  - $(\{¢ab, bab\}, \underline{a}, \{b\$, bab\})$

  by the **delta instruction**:
  - $(\{¢ab, bab\}, \underline{a} \rightarrow \Delta, \{b\$, bab\})$.
- How does the sample computation **change**?

Idea

- The **modified sample computation** is now:

$¢\, ababababababab\underline{a}b\ \$ \vdash_M ¢\, abababababab\underline{a}bab\Delta b\ \$ \vdash_M$

$¢\, abababab\underline{a}bab\Delta bab\Delta b\ \$ \vdash_M ¢\, ab\underline{a}bab\Delta bab\Delta bab\Delta b\ \$ \vdash_M$

$¢\, ab\Delta bab\Delta bab\Delta ba\underline{b\Delta} b\ \$ \vdash_M ¢\, ab\Delta bab\Delta ba\underline{b\Delta} bab\ \$ \vdash_M$

$¢\, ab\Delta ba\underline{b\Delta} bababab\ \$ \vdash_M ¢\, a\underline{b\Delta} bababab\ \$ \vdash_M$

$¢\, abababab\underline{a}b\ \$ \vdash_M ¢\, ab\underline{a}bab\Delta b\ \$ \vdash_M$

$¢\, ab\Delta ba\underline{b\Delta} b\ \$ \vdash_M ¢\, a\underline{b\Delta} bab\ \$ \vdash_M$

$¢\, ab\underline{a}b\ \$ \vdash_M ¢\, a\underline{b\Delta} b\ \$ \vdash_M ¢\, \underline{ab}\ \$ \vdash_M ¢\, \lambda\ \$$ ***accept***.

- Unfortunately, it **does not help** us if $\boldsymbol{k = 3}$.

Idea

- Consider the following **sample reduction**:
  - $ab\underline{a}bab\Delta b \vdash_M ab\Delta bab\Delta b$
- As you can see, **this reduction also works** with the word ***ababab*** :
  - $aba\underline{b}ab \vdash_M ab\Delta bab$

  because the length of contexts is $\boldsymbol{k = 3}$.
- Is there any **other way** how to **modify** the instruction $(\{¢ab, bab\}, \underline{a}, \{b\$, bab\})$?
- **Yes, it is**.

Revised Idea

- It is **not enough** to replace a **single** letter a :
  - $(\{¢ab, bab\}, \underline{a} \rightarrow \Delta, \{b\$, bab\})$.
- We have **two simple choices** for **two** letters:
  - $(\{¢ab, bab\}, \underline{ab} \rightarrow \Delta, \{\$, ab\})$,
  - $(\{¢a, ba\}, \underline{ba} \rightarrow \Delta, \{b\$, bab\})$.
- We **move** one letter **from the context to the rule** and thus extend the **context horizon**.
- In the following we show how does the sample computation **change** in both cases.

Case 1

- If we use $(\{¢ab, bab\}, \underline{ab} \rightarrow \Delta, \{\$, ab\})$ we get the following **new sample computation**:

$¢ abababababababab\underline{ab}\ \$ \vdash_M ¢ abababababab\underline{ab}ab\Delta\ \$ \vdash_M$

$¢ abababab\underline{ab}ab\Delta ab\Delta\ \$ \vdash_M ¢ ab\underline{ab}ab\Delta ab\Delta ab\Delta\ \$ \vdash_M$

$¢ ab\Delta ab\Delta ab\Delta ab\underline{\Delta}\ \$ \vdash_M ¢ ab\Delta ab\Delta ab\underline{\Delta} ab\ \$ \vdash_M$

$¢ ab\Delta ab\underline{\Delta} abab\ \$ \vdash_M ¢ ab\underline{\Delta} ababab\ \$ \vdash_M$

$¢ abababab\underline{ab}\ \$ \vdash_M ¢ ab\underline{ab}ab\Delta\ \$ \vdash_M$

$¢ ab\Delta ab\underline{\Delta}\ \$ \vdash_M ¢ ab\underline{\Delta} ab\ \$ \vdash_M ¢ ab\underline{ab}\ \$ \vdash_M$

$¢ ab\underline{\Delta}\ \$ \vdash_M ¢ \underline{ab}\ \$ \vdash_M ¢ \lambda\ \$$ **accept**.

Case 1

- For $k = 3$ we get the following **instructions**:
  - $(\{¢ab, \Delta ab\}, \underline{\Delta}, \{\$, ab\$, aba\})$,
  - $(\{¢ab, bab\}, \underline{ab} \rightarrow \Delta, \{\$, ab\Delta\})$,
  - $(¢, \underline{ab}, \$)$.
- For the resulting ***3-cl-RA*-automaton M** the following holds:
  $L(M) = \{(ab)^{2^m} \mid m \geq 0\} \cup \{\lambda\}$.
- **Note** that the symbol Δ **codes** the letter b from the **original sample computation**.

Case 2

- If we use $(\{¢a, ba\}, \underline{ba} \rightarrow \Delta, \{b\$, bab\})$ we get the following **new sample computation**:

$¢ ababababababa\underline{ba}b\ \$ \vdash_M ¢ abababababa\underline{ba}ba\Delta b\ \$ \vdash_M$

$¢ ababa\underline{ba}ba\Delta ba\Delta b\ \$ \vdash_M ¢ a\underline{ba}ba\Delta ba\Delta ba\Delta b\ \$ \vdash_M$

$¢ a\Delta ba\Delta ba\Delta ba\underline{\Delta}b\ \$ \vdash_M ¢ a\Delta ba\Delta ba\underline{\Delta}bab\ \$ \vdash_M$

$¢ a\Delta ba\underline{\Delta}babab\ \$ \vdash_M ¢ a\underline{\Delta}bababab\ \$ \vdash_M$

$¢ ababa\underline{ba}b\ \$ \vdash_M ¢ a\underline{ba}ba\Delta b\ \$ \vdash_M$

$¢ a\Delta ba\underline{\Delta}b\ \$ \vdash_M ¢ a\underline{\Delta}bab\ \$ \vdash_M ¢ a\underline{ba}b\ \$ \vdash_M$

$¢ a\underline{\Delta}b\ \$ \vdash_M ¢ \underline{ab}\ \$ \vdash_M ¢ \lambda\ \$$ ***accept***.

Case 2

- For $k = 3$ we get the following **instructions**:
  - $(\{¢a, \Delta ba\}, \underline{\Delta}, \{b\$, bab\})$,
  - $(\{¢a, aba\}, \underline{ba} \rightarrow \Delta, \{b\$, ba\Delta\})$,
  - $(¢, \underline{ab}, \$)$.
- For the resulting ***3-cl-RA*-automaton M** the same formula holds:
  $L(M) = \{(ab)^{2^m} \mid m \geq 0\} \cup \{\lambda\}$.
- **Again** the symbol Δ **codes** the letter b from the **original sample computation**.

Some Remarks

- Note that the for $k < 3$ the previous algorithm **does not work**.
- It is also **not obvious** how to replace **more letters** or even **more instructions** without disturbing the sample computation.
- The question is whether we can do these replacements in some **automated way**?
- Is this idea applicable to **other examples**?