# Endliche Automaten als Terminations-Zertifikate

Jörg Endrullis (Uni Leipzig)
Alfons Geser (HTWK Leipzig)
Dieter Hofbauer (Kassel)
*Johannes Waldmann* (HTWK Leipzig)
Hans Zantema (TU Eindhoven)



## Wortersetzungssysteme (SRS)

$R$ ist Wortersetzungssystem
(string rewriting system = SRS) über Alphabet $\Sigma$:

$$R \subseteq \Sigma^* \times \Sigma^*$$

Beispiel (Geser/Zantema 96):

$$R = \{bc \to dc, bd \to db, ad \to abb\}.$$

definiert Ableitungsrelation auf $\Sigma^*$:

$$\to_R = \{(xly, xry) \mid x, y \in \Sigma^*, (l, r) \in R\}.$$

$abbbc \to_R abbdc \to_R abdbc \to_R adbbc \to_R abbbbc \to_R \ldots$



## Eigenschaften von SRS

Wortersetzung ist Berechnungsmodell. Fragen:

- Termination (es exist. keine unendliche Ableitung)
- Erreichbarkeit ($\exists x \in X, y \in Y : x \to_R^* y$)

Motivation: Programm-Analyse

- Programm soll für alle Eingaben halten,
- Programmzustand soll nie fehlerhaft sein.

Modell ist Turing-vollständig → alle interessanten Eigenschaften sind unentscheidbar → betrachte (syntaktisch) eingeschränkte Klassen von SRS (vergleiche Grammatiken der Chomsky-Hierarchie)



## Termination von SRS

- $ab \to bc$ betrachte Anzahl der $a$
- $ab \to ba$ Interpretation $a = \lambda x.2x, b = \lambda x.x + 1$
- $ab \to baa$ Interpretation
- $aa \to aba$ selbsteinbettend → keine Interpretation
- $aabb \to bbbaaa$
- $1000 \to 0001110$
- $\{aa \to bc, bb \to ac, cc \to ab\}$

eines der Systeme ist nichtterminierend
für eines ist Termination offen.



## Plan des Vortrags

Endliche Automaten als Terminationszertifikate

- was ist und wie funktioniert das Zertifikat?
  - Überdeckungen von Ersetzungssystemen
  - lokale Termination von Ersetzungssystemen
  - kompatible Automaten
- wie konstruiert man das Zertifikat?
  - ein Zerlegungssatz für *deleting* Systeme
  - effiziente Implementierung



## 1. Zertifikate



## Überdeckungen

Ziel: zu (Ersetzungs-)Relation $R$ auf $U$
konstruiere Relation $R'$ auf $U'$,
aus Termination von $R'$ folgt Termination von $R$.
Definition: $R'$ auf $U'$ ist *Überdeckung* von $R$ auf $U$ *vermittels* $(\mathrm{lift} : U \to U', \mathrm{base} : U' \to U)$, falls

- $\mathrm{lift} \circ \mathrm{base} = \mathrm{id}|_U$
- $(\mathrm{base} \circ R) \subseteq (R' \circ \mathrm{base})$

jede Folge von $R$-Schritten wird durch gleichlange Folge von $R'$-Schritte überdeckt.



## Höhen-Annotationen

wählen $U = \Sigma^*$ und $U' = \Sigma'^*$ mit $\Sigma' = \Sigma \times \mathbb{N}$.
benutze Morphismen
$\mathrm{base}(a, h) = a, \mathrm{height}(a, h) = h, \mathrm{lift}_h(a) = (a, h)$.
für SRS $R$ über $\Sigma$ definiere SRS über $\Sigma \times \mathbb{N}$ durch

$$\mathrm{match}(R) = \{l' \to \mathrm{lift}_{h+1}(r) \mid (l \to r) \in R, h = \min_{a' \in l'} \mathrm{height}\, a'\}$$

Beispiel: $\mathrm{match}(aabb \to bbbaaa) = \{a_0a_0b_0b_0 \to b_1b_1b_1a_1a_1a_1, \ldots, a_3a_1b_0b_2 \to b_1b_1b_1a_1a_1a_1, \ldots\}$
Fakt: $\mathrm{match}(R)$ ist Überdeckung von $R$.

## Höhen-Annotationen für $\{\underline{aabb} \to bbbaaa\}$

$$\begin{array}{rl}
 & \underline{a_0a_0b_0b_0}\ a_0b_0b_0\ a_0b_0b_0\ a_0b_0b_0 \\
\to & b_1^3a_1a_1\ \underline{a_1\ a_0b_0b_0}\ a_0b_0b_0\ a_0b_0b_0 \\
\to & \\
\to^2 & \\
\to^2 & \\
\to^2 & \\
\to^2 & \\
\to & \\
\to &
\end{array}$$



## Höhen-Annotationen für $\{\underline{aabb} \to bbbaaa\}$

$$\begin{array}{rl}
 & \underline{a_0a_0b_0b_0}\ a_0b_0b_0\ a_0b_0b_0\ a_0b_0b_0 \\
\to & b_1^3a_1a_1\ \underline{a_1\ a_0b_0b_0}\ a_0b_0b_0\ a_0b_0b_0 \\
\to & b_1^3\ \underline{a_1a_1\ b_1b_1}\ b_1a_1a_1\ \underline{a_1\ a_0b_0b_0}\ a_0b_0b_0 \\
\to^2 & \\
\to^2 & \\
\to^2 & \\
\to^2 & \\
\to & \\
\to &
\end{array}$$



## Höhen-Annotationen für $\{\underline{aabb} \to bbbaaa\}$

$$\begin{array}{rl}
 & \underline{a_0a_0b_0b_0}\ a_0b_0b_0\ a_0b_0b_0\ a_0b_0b_0 \\
\to & b_1^3a_1a_1\ \underline{a_1\ a_0b_0b_0}\ a_0b_0b_0\ a_0b_0b_0 \\
\to & b_1^3\ \underline{a_1a_1\ b_1b_1}\ b_1a_1a_1\ \underline{a_1\ a_0b_0b_0}\ a_0b_0b_0 \\
\to^2 & b_1^3\ b_2^3a_2a_2a_2\ b_1\ \underline{a_1a_1\ b_1b_1}\ b_1a_1a_1\ \underline{a_1\ a_0b_0b_0} \\
\to^2 & \\
\to^2 & \\
\to^2 & \\
\to & \\
\to &
\end{array}$$



## Höhen-Annotationen für $\{\underline{aabb} \to bbbaaa\}$

$$\begin{array}{rl}
 & \underline{a_0a_0b_0b_0}\ a_0b_0b_0\ a_0b_0b_0\ a_0b_0b_0 \\
\to & b_1^3a_1a_1\ \underline{a_1\ a_0b_0b_0}\ a_0b_0b_0\ a_0b_0b_0 \\
\to & b_1^3\ \underline{a_1a_1\ b_1b_1}\ b_1a_1a_1\ \underline{a_1\ a_0b_0b_0}\ a_0b_0b_0 \\
\to^2 & b_1^3\ b_2^3a_2a_2a_2\ b_1\ \underline{a_1a_1\ b_1b_1}\ b_1a_1a_1\ \underline{a_1\ a_0b_0b_0} \\
\to^2 & b_1^3\ b_2^3a_2\ \underline{a_2a_2b_1\ b_2}\ b_2b_2a_2a_2a_2\ b_1\ \underline{a_1a_1\ b_1b_1}\ b_1a_1^3 \\
\to^2 & \\
\to^2 & \\
\to & \\
\to &
\end{array}$$



## Höhen-Annotationen für $\{\underline{aabb} \to bbbaaa\}$

$$\begin{array}{rl}
 & \underline{a_0a_0b_0b_0}\ a_0b_0b_0\ a_0b_0b_0\ a_0b_0b_0 \\
\to & b_1^3a_1a_1\ \underline{a_1\ a_0b_0b_0}\ a_0b_0b_0\ a_0b_0b_0 \\
\to & b_1^3\ \underline{a_1a_1\ b_1b_1}\ b_1a_1a_1\ \underline{a_1\ a_0b_0b_0}\ a_0b_0b_0 \\
\to^2 & b_1^3\ b_2^3a_2a_2a_2\ b_1\ \underline{a_1a_1\ b_1b_1}\ b_1a_1a_1\ \underline{a_1\ a_0b_0b_0} \\
\to^2 & b_1^3\ b_2^3a_2\ \underline{a_2a_2b_1\ b_2}\ b_2b_2a_2a_2a_2\ b_1\ \underline{a_1a_1\ b_1b_1}\ b_1a_1^3 \\
\to^2 & b_1^3\ b_2^3a_2\ b_2^3a_2\ \underline{a_2a_2\ b_2b_2}\ a_2\ \underline{a_2a_2b_1\ b_2}\ b_2^2a_2^3\ b_1a_1^3 \\
\to^2 & \\
\to & \\
\to &
\end{array}$$



## Höhen-Annotationen für $\{\underline{aabb} \to bbbaaa\}$

$$\begin{array}{rl}
 & \underline{a_0a_0b_0b_0}\ a_0b_0b_0\ a_0b_0b_0\ a_0b_0b_0 \\
\to & b_1^3a_1a_1\ \underline{a_1\ a_0b_0b_0}\ a_0b_0b_0\ a_0b_0b_0 \\
\to & b_1^3\ \underline{a_1a_1\ b_1b_1}\ b_1a_1a_1\ \underline{a_1\ a_0b_0b_0}\ a_0b_0b_0 \\
\to^2 & b_1^3\ b_2^3a_2a_2a_2\ b_1\ \underline{a_1a_1\ b_1b_1}\ b_1a_1a_1\ \underline{a_1\ a_0b_0b_0} \\
\to^2 & b_1^3\ b_2^3a_2\ \underline{a_2a_2b_1\ b_2}\ b_2b_2a_2a_2a_2\ b_1\ \underline{a_1a_1\ b_1b_1}\ b_1a_1^3 \\
\to^2 & b_1^3\ b_2^3a_2\ b_2^3a_2\ \underline{a_2a_2\ b_2b_2}\ a_2\ \underline{a_2a_2b_1\ b_2}\ b_2^2a_2^3\ b_1a_1^3 \\
\to^2 & b_1^3\ b_2^3a_2\ b_2^3a_2\ b_3^3a_3a_3\ \underline{a_3\ a_2\ b_2b_2}\ b_2a_2^3\ b_2^2a_2^3\ b_1a_1^3 \\
\to & \\
\to &
\end{array}$$



## Höhen-Annotationen für $\{\underline{aabb} \to bbbaaa\}$

$$\begin{array}{rl}
 & \underline{a_0a_0b_0b_0}\ a_0b_0b_0\ a_0b_0b_0\ a_0b_0b_0 \\
\to & b_1^3a_1a_1\ \underline{a_1\ a_0b_0b_0}\ a_0b_0b_0\ a_0b_0b_0 \\
\to & b_1^3\ \underline{a_1a_1\ b_1b_1}\ b_1a_1a_1\ \underline{a_1\ a_0b_0b_0}\ a_0b_0b_0 \\
\to^2 & b_1^3\ b_2^3a_2a_2a_2\ b_1\ \underline{a_1a_1\ b_1b_1}\ b_1a_1a_1\ \underline{a_1\ a_0b_0b_0} \\
\to^2 & b_1^3\ b_2^3a_2\ \underline{a_2a_2b_1\ b_2}\ b_2b_2a_2a_2a_2\ b_1\ \underline{a_1a_1\ b_1b_1}\ b_1a_1^3 \\
\to^2 & b_1^3\ b_2^3a_2\ b_2^3a_2\ \underline{a_2a_2\ b_2b_2}\ a_2\ \underline{a_2a_2b_1\ b_2}\ b_2^2a_2^3\ b_1a_1^3 \\
\to^2 & b_1^3\ b_2^3a_2\ b_2^3a_2\ b_3^3a_3a_3\ \underline{a_3\ a_2\ b_2b_2}\ b_2a_2^3\ b_2^2a_2^3\ b_1a_1^3 \\
\to & b_1^3\ b_2^3a_2\ b_2^3a_2\ b_3^3\ \underline{a_3a_3\ b_3b_3}\ b_3a_3^3\ b_2a_2^3\ b_2^2a_2^3\ b_1a_1^3 \\
\to &
\end{array}$$



## Höhen-Annotationen für $\{\underline{aabb} \to bbbaaa\}$

$$\begin{array}{rl}
 & \underline{a_0a_0b_0b_0}\ a_0b_0b_0\ a_0b_0b_0\ a_0b_0b_0 \\
\to & b_1^3a_1a_1\ \underline{a_1\ a_0b_0b_0}\ a_0b_0b_0\ a_0b_0b_0 \\
\to & b_1^3\ \underline{a_1a_1\ b_1b_1}\ b_1a_1a_1\ \underline{a_1\ a_0b_0b_0}\ a_0b_0b_0 \\
\to^2 & b_1^3\ b_2^3a_2a_2a_2\ b_1\ \underline{a_1a_1\ b_1b_1}\ b_1a_1a_1\ \underline{a_1\ a_0b_0b_0} \\
\to^2 & b_1^3\ b_2^3a_2\ \underline{a_2a_2b_1\ b_2}\ b_2b_2a_2a_2a_2\ b_1\ \underline{a_1a_1\ b_1b_1}\ b_1a_1^3 \\
\to^2 & b_1^3\ b_2^3a_2\ b_2^3a_2\ \underline{a_2a_2\ b_2b_2}\ a_2\ \underline{a_2a_2b_1\ b_2}\ b_2^2a_2^3\ b_1a_1^3 \\
\to^2 & b_1^3\ b_2^3a_2\ b_2^3a_2\ b_3^3a_3a_3\ \underline{a_3\ a_2\ b_2b_2}\ b_2a_2^3\ b_2^2a_2^3\ b_1a_1^3 \\
\to & b_1^3\ b_2^3a_2\ b_2^3a_2\ b_3^3\ \underline{a_3a_3\ b_3b_3}\ b_3a_3^3\ b_2a_2^3\ b_2^2a_2^3\ b_1a_1^3 \\
\to & b_1^3\ b_2^3a_2\ b_2^3a_2\ b_3^3\ b_4^3a_4^3\ b_3a_3^3\ b_2a_2^3\ b_2^2a_2^3\ b_1a_1^3
\end{array}$$

## Höhen-Annotationen für $\{\underline{aabb} \to bbbaaa\}$

$$
\begin{array}{ll}
 & \underline{a_0a_0b_0b_0}\ a_0b_0b_0\ a_0b_0b_0\ a_0b_0b_0 \\
\to & b_1^3a_1a_1\ \underline{a_1\ a_0b_0b_0}\ a_0b_0b_0\ a_0b_0b_0 \\
\to & b_1^3\ \underline{a_1a_1\ b_1b_1}\ b_1a_1a_1\ \underline{a_1\ a_0b_0b_0}\ a_0b_0b_0 \\
\to^2 & b_1^3\ b_2^3a_2a_2a_2\ b_1\ \underline{a_1a_1\ b_1b_1}\ b_1a_1a_1\ \underline{a_1\ a_0b_0b_0} \\
\to^2 & b_1^3\ b_2^3a_2\ \underline{a_2a_2b_1\ b_2}\ b_2b_2a_2a_2a_2\ b_1\ \underline{a_1a_1\ b_1b_1}\ b_1a_1^3 \\
\to^2 & b_1^3\ b_2^3a_2\ b_2^3a_2\ \underline{a_2a_2\ b_2b_2}\ a_2\ \underline{a_2a_2b_1\ b_2}\ b_2^2a_2^3\ b_1a_1^3 \\
\to^2 & b_1^3\ b_2^3a_2\ b_2^3a_2\ b_3^3a_3a_3\ \underline{a_3\ a_2\ b_2b_2}\ b_2a_2^3\ b_2^2a_2^3\ b_1a_1^3 \\
\to & b_1^3\ b_2^3a_2\ b_2^3a_2\ b_3^3\ \underline{a_3a_3\ b_3b_3}\ b_3a_3^3\ b_2a_2^3\ b_2^2a_2^3\ b_1a_1^3 \\
\to & b_1^3\ b_2^3a_2\ b_2^3a_2\ b_3^3\ b_4^3a_4^3\ b_3a_3^3\ b_2a_2^3\ b_2^2a_2^3\ b_1a_1^3
\end{array}
$$

Höhen $> 4$ kommen nicht vor $\Rightarrow$ System terminiert.



## Lokal terminierende SRS

Definition. Ein SRS $R$ über $\Sigma$ heißt *lokal terminierend*, wenn jede Einschränkung von $R$ auf eine endliche Signatur terminiert.
Satz: Für jedes $R$ ist $\operatorname{match}(R)$ lokal terminierend.
Beweis: ignoriere Buchstaben, betrachte nur Höhen.
größte vorkommende Höhe sei $c$,
längste rechte Regelseite hat $w$ Buchstaben.
Gewichtsfunktion $f(h) = (w+1)^{c-h}$
liefert streng mononoton fallende Interpretation:
aus $(l \to r) \in \operatorname{match}(R)$
folgt $h = \min \operatorname{height} l < \min \operatorname{height} r$,
also $f(l) \ge (w+1)^{c-h} > w \cdot (w+1)^{c-(h+1)} \ge f(r)$.



## Kompatible Automaten

Definition: Endlicher Automat $A$
heißt kompatibel mit SRS $R$ für Sprache $L$, falls

- $L \subseteq L(A)$
- für alle $(l \to r) \in R$ und $p \xrightarrow{l}_A q$ gilt $p \xrightarrow{r}_A q$.

Fakt. Falls $A$ kompatibel mit $R$ für $L$,
dann $\to_R^* (L) \subseteq L(A)$
Fakt. ... und Signatur von $\to_R^* (L)$ ist endlich.
Fakt. „$A$ ist kompatibel mit $R$ für $L \in$ REG“
ist entscheidbar.



## Kompatible Automaten als Zertifikate

Satz: Wenn

- $R'$ über $\Sigma'$ eine Überdeckung von $R$ über $\Sigma$ vermittels $(\mathrm{lift}, \mathrm{base})$
- und $R'$ lokal terminierend
- und $A$ ein Automat kompatibel mit $R'$ und $\Sigma'^*$,

dann ist $R$ terminierend.



## Kompatibilität mit $\operatorname{match}(R)$

Fakt. $\operatorname{match}(R)$ ist lokal terminierende Überdeckung von $R$ vermittels $(\mathrm{lift}_0, \mathrm{base})$.
Ein mit $\operatorname{match}(R)$ und $\mathrm{lift}_0(\Sigma^*)$ kompatibler Automat $A$ zertifiziert, daß $R$ *match-bounded* ist:
$\max \operatorname{height}(\to_{\operatorname{match}(R)}^* (\mathrm{lift}_0(\Sigma^*))) < \infty$.
Beispiel: $\{aa \to aba\}$ ist *match-bounded* durch 2, es gibt einen Zertifikat-Automaten mit 5 Zuständen.



## Match-boundedness: Implementierungen

Die Konstruktion solcher Automaten ist implementiert in den Programmen

- Matchbox [Waldmann 2003] (exakt und ineffizient),
- Torpa [Zantema 2004] (effizient, aber approximiert),
- Aprove [Giesl et al 2004],
- Jambox [Endrullis 2005] (effizient *und* exakt).

Leistungsvergleich bei *Termination Competition*
(Torpa und Aprove implementieren weitere Terminations-Verfahren.)



## 2. Konstruktion



## Ein Zerlegungssatz (I)

gegeben Alphabet $\Sigma$, definiere neue Zeichen

$$\overrightarrow{\Sigma} = \{\overrightarrow{a} \mid a \in \Sigma\}, \quad \overleftarrow{\Sigma} = \{\overleftarrow{a} \mid a \in \Sigma\}$$

als formale Inverse der Zeichen aus $\Sigma$:

$$E_\Sigma = \{\overrightarrow{a}\,a \to \epsilon, \quad a\,\overleftarrow{a} \to \epsilon \quad | \quad a \in \Sigma\}$$

Setze fort zu $\overrightarrow{a_1 \dots a_n} = \overrightarrow{a_n} \dots \overrightarrow{a_1}$, $\overleftarrow{a_1 \dots a_n} = \overleftarrow{a_n} \dots \overleftarrow{a_1}$.
Dann $\overrightarrow{x}\,x \to_E^* \epsilon, x\,\overleftarrow{x} \to_E^* \epsilon$.

## Ein Zerlegungssatz (II)

Gegeben SRS $R$ über $\Sigma$. Für jede Regel $(l \to r) \in R$ zerlege $l = xay$ mit $x \in \Sigma^*, a \in \Sigma, y \in \Sigma^*$ und definiere

$$S = \{a \to \overleftarrow{x}\, r\, \overrightarrow{y} \mid (xay \to r) \in R\}$$

Satz: $\to_R^* = (\to_S^* \circ \to_E^*) \cap (\Sigma^* \times \Sigma^*)$.
Beweis:

- $l = xay \to_S x\overleftarrow{x}\, r\, \overrightarrow{y} y \to_E^* \epsilon r \epsilon$
- man kann alle $S$-Schritte vor alle $E$-Schritte ziehen

$R$ is beliebig, $S$ ist kontextfrei, $E$ ist monadisch.
Idee: für gewisse $R$ ist $S$ terminierend,
beschreibt also eine endliche Substitution.



## Deleting SRS

Für $>$ strikter Anteil einer Prä-ordnung $\geq$ auf $\Sigma$:
Definition: $R$ ist $>$-deleting, falls

$$\forall (l \to r) \in R : \exists a \in l : \forall b \in r : a > b.$$

Satz: Für deleting $R$ existieren $S, E$ über $\Gamma \supseteq \Sigma$ mit

- $S \subseteq \Sigma \times \Gamma^*$ ($S$ ist kontextfrei)
- $S$ ist terminierend
- $E \subseteq \Gamma^* \times \{\epsilon\}$ ($E$ ist speziell monadisch)
- $\to_R^* = (\to_S^* \circ \to_E^*) \cap (\Sigma^* \times \Sigma^*)$.



## Deleting SRS: Eigenschaften

$\to_R^* = (\to_S^* \circ \to_E^*) \cap (\Sigma^* \times \Sigma^*)$
Folgerungen:

- $\to_R^*$ ist effektiv REG-erhaltend:
  $\forall L \in \text{REG} : \to_R^* (L) \in \text{REG}$
- $\to_R^{-*}$ ist effektiv CF-erhaltend
- $\to_R$ ist terminierend
- $\text{Inf}(\to_R^-) \in$ REG (effektiv)



## Match-bounded SRS

Fakt: $\text{match}(R)$ ist deleting
bzgl. Ordnung $a_h > a_i \iff h < i$.
Satz: wenn $R$ match-bounded, dann

- $\to_R^*$ ist effektiv REG-erhaltend:
  $\forall L \in \text{REG} : \to_R^* (L) \in \text{REG}$
- $\to_R^{-*}$ ist effektiv CF-erhaltend
- $\to_R$ ist terminierend
- $\text{Inf}(\to_R^-) \in$ REG (effektiv)



## Konstruktionen: Nachfolgermenge

- Gegeben: $R$ über $\Sigma$, $L \in \text{REG}, c \in \mathbb{N}$.
- Gesucht: $\to_{\text{match}_{\leq c}(R)}^* (\text{lift}_0\, L)$.

Naiver Algorithmus (fürchterlich ineffizient)

- konstruiere Zerlegung $S', E'$
  mit $\to_{\text{match}_c(R)}^* = (\to_{S'}^* \circ \to_{E'}^*) \cap (\Sigma'^* \times \Sigma'^*)$
- konstruiere (Automaten für) $\text{lift}_0\, L$ (Morphismus)
- konstruiere $\to_{S'}^* (\text{lift}_0\, L)$ (endliche Substitution)
- konstruiere $\to_{E'}^* (\to_{S'}^* (\text{lift}_0\, L))$ (monad. Abschluß)
- lösche Kanten $\notin \Sigma'$



## Konstruktionen: Nachfolgermenge (II)

Fakt: der so konstruierte Automat ist kompatibel mit $\text{match}(R)$ und $L$.



## Konstruktionen: Beschränktheit

Schranke ist gegeben:

- Gegeben: $R$ über $\Sigma$, $L \in \text{REG}, c \in \mathbb{N}$.
- Frage: ist $R$ match-bounded für $L$ durch $c$?

Algorithmus:

- konstruiere $A$ für $\to_{\text{match}_{c+1}(R)}^* (\text{lift}_0\, L)$
- antworte JA, falls $A$ keine nützliche Kante enthält, die mit $\Sigma \times (c+1)$ beschriftet ist, sonst NEIN



## Konstruktion: Beschränktheit (II)

Schranke ist nicht gegeben:

- Gegeben: $R$ über $\Sigma$, $L \in$ REG.
- Frage: ist $R$ match-bounded für $L$?

Semi-Algorithmus: voriger Algorithmus für $c = 0, 1, 2, \ldots$
(Entscheidbarkeit ist offen.)

## Effiziente Algorithmen: bessere Inverse

Für $R = \{aa \to aba\}$ enthält $R' = \text{match}(R)$ u. a.

$$a_0a_1 \to a_1b_1a_1, a_0a_2 \to a_1b_1a_1, a_0a_3 \to a_1b_1a_1, \ldots$$

mit gemeinsamen rechten Seiten. Benutze Zerlegung

$$\begin{aligned} S' &= \{\text{lift}_h(a) \to \text{lift}_h(\overleftarrow{x})\ \text{lift}_{h+1}(r)\ \text{lift}_h(\overrightarrow{y}) \\ &\quad \mid (xay \to r) \in R, h \in \mathbb{N}\} \end{aligned}$$

mit „erweiterten“ Inversen

$$E'_\Sigma = \{\overrightarrow{a_i}a_j \to \epsilon, \quad a_j\overleftarrow{a_i} \to \epsilon \quad \mid \quad a \in \Sigma, i \le j\}.$$

Satz. $\to^*_{\text{match}(R)} = (\to^*_{S'} \circ \to^*_{E'}) \cap ((\Sigma \times \mathbb{N})^* \times (\Sigma \times \mathbb{N})^*)$.



## Effiziente Algorithmen: Erreichbarkeit

In $\to^*_{E'} (\to^*_{S'} (\text{lift}_0\, L))$: konstruiere von vornherein nur erreichbare Kanten.
Konstruiere Folge $(A_0, A_1, \ldots)$ von Automaten mit $L(A_0) = \text{lift}_0(L)$ und

- falls $(l' \to r') \in \text{match}_c(R)$
  mit $p \xrightarrow{l'}_{A_k} q$ und nicht $(p \xrightarrow{r'}_{A_k} q)$,
- dann wähle Zerlegung $l' = x'a'y'$
  mit $h = \text{height}\, a'$ minimal in $l'$ und $p \xrightarrow{x'} s \xrightarrow{a'} t \xrightarrow{y'} q$

$A_{k+1} = A_k$ plus neuer Pfad $s \xrightarrow{\text{lift}_h(\overleftarrow{x})\ \text{lift}_{h+1}(r)\ \text{lift}_h(\overrightarrow{y})} t$
und Abschluß unter Regeln aus $E'$



## Zusammenfassung

- Anregung:
  change-bounded rewriting (Ravikumar 96)
  $\{a_2b_1b_4 \to b_3a_2a_5, \ldots\}$
  nur für längenerhaltende Systeme
- deleting/match-bounded Theorie publiziert in DLT03, TCS, MFCS03, AAECC, JAR, WST03, RTA04;
- inzwischen vier unabhängige Implementierungen



## Ausblick

- match bounds für Termersetzung (TRS)
  - welches sind die richtigen Höhen-Annotationen?
  - welches ist der richtige Zerlegungssatz (welche TRS sind *deleting*)?
  - wie behandelt man Nichtlinearitäten?
  - Implementierung (Matchbox): top/match-heights benutzen und Zertifikat-Automaten (halbwegs) intelligent raten, WST04, RTA05
- *hot topic*: Beziehungen zu gewichteten Automaten