Controlled Term Rewriting

Florent Jacquemard[1], Yoshiharu Kojima[2,3], and Masahiko Sakai[2]

[1]INRIA Sacray, LSV

[2]Nagoya University

[3]Research Fellow of JSPS

FroCoS2011, Saarbrücken, 7, October, 2011

Strategies for TRS

Analysis of term rewriting under strategies with tree automata techniques

Innermost is analogous of *call by value*, [Rety Vuotto JSC 05], [Gascon, Godoy, Jacquemard WRS 08]

Outermost is analogous of *call by name*, [Rety Vuotto JSC 05]

Context-sensitive is used for `if...then..else` [Kojima Sakai RTA 08]

Bottom-up [Durand Senizergues Sylvestre RTA 07,10,11]

Node selection explicit definition of rewrite positions
e.g. with XPath in XML transformation languages
[Jacquemard Rusinowitch PPDP 10]

- XQuery Update Facility, W3C recommendation.
- Analysis of Access Control Policies.

Results

- Formalization of controlled term rewriting systems (CntTRSs)
- Showing decidable or undecidable properties about reachability or model checking problem for CntTRSs.
  - Monotonic CntTRSs,
  - Prefix CntTRSs,
  - Non-monotonic CntTRSs, and
  - Recursive prefix CntTRSs.

Results

- Formalization of controlled term rewriting systems (CntTRSs)
- Showing decidable or undecidable properties about reachability or model checking problem for CntTRSs.
    - Monotonic CntTRSs,
    - Prefix CntTRSs,
    - Non-monotonic CntTRSs, and
    - Recursive prefix CntTRSs.

Tree automaton (TA)

The automaton that accepts trees (terms).

Example (TA accepting $f(a,b)$)

TA $\mathcal{A} = \langle Q, F, \Delta \rangle$, $\mathcal{L}(\mathcal{A}) = \{f(a,b)\}$

- F(final states): $\{q_f\}$
- Δ(transition rules): $\{a \to q_1, b \to q_2, f(q_1, q_2) \to q_f\}$

$$f(a,b) \underset{\Delta}{\longrightarrow} f(q_1, b) \underset{\Delta}{\longrightarrow} f(q_1, q_2) \underset{\Delta}{\longrightarrow} q_f \in F$$

Controlled term rewriting systems (CntTRSs)

A controlled rewrite rule $\mathcal{A} : l \rightarrow r$ is made of

- a Selection Automaton (SA) $\mathcal{A}$ and
- a rewrite rule $l \rightarrow r$

Was introduced for string rewriting, see survey [Senizergues 93] and [Dassow, Paun, Sallomaa 97].

Definition (Selection automaton (Gottlob, Koch 04))

Quadruple $\langle Q, F, S, \Delta \rangle$ where $\langle Q, F, \Delta \rangle$ is TA and $S \subseteq Q$.

Example

$$R = \begin{cases} \mathcal{A}_1 & : & a \to c \\ \mathcal{A}_2 & : & b \to c \\ \mathcal{A}_3 & : & f(x,y) \to g(x,y) \end{cases}$$

where

$$\begin{array}{rcl} & & \quad Q \qquad\qquad\quad F \qquad S \qquad\qquad\qquad\qquad \Delta \\ \mathcal{A}_1 & = & \langle \{q_1, q_2, q_f\}, \{q_f\}, \{q_1\}, \{a \to q_1, b \to q_2, f(q_1, q_2) \to q_f\} \rangle \\ \mathcal{A}_2 & = & \langle \{q_1, q_2, q_f\}, \{q_f\}, \{q_2\}, \{a \to q_1, b \to q_2, g(q_1, q_2) \to q_f\} \rangle \\ \mathcal{A}_3 & = & \langle \{q_1, q_2, q_f\}, \{q_f\}, \{q_f\}, \{c \to q_1, b \to q_2, f(q_1, q_2) \to q_f\} \rangle \end{array}$$

$$f(a,b) \xrightarrow[R]{} f(c,b) \xrightarrow[R]{} g(c,b) \xrightarrow[R]{} g(c,c).$$

$$f(a,b) \xrightarrow[\mathcal{A}_1]{} f(q_1,b) \xrightarrow[\mathcal{A}_1]{} f(q_1,2) \xrightarrow[\mathcal{A}_1]{} q_f$$

Controlled term rewriting systems (CntTRS)

Example

$$R = \begin{cases} \mathcal{A}_1 & : \quad a \to c \\ \mathcal{A}_2 & : \quad b \to c \\ \mathcal{A}_3 & : \quad f(x,y) \to g(x,y) \end{cases}$$

where

$$\begin{array}{lcl} & & \quad Q \qquad\qquad\quad F \qquad S \qquad\qquad\qquad\qquad \Delta \\ \mathcal{A}_1 & = & \langle \{q_1, q_2, q_f\}, \{q_f\}, \{q_1\}, \{a \to q_1, b \to q_2, f(q_1, q_2) \to q_f\} \rangle \\ \mathcal{A}_2 & = & \langle \{q_1, q_2, q_f\}, \{q_f\}, \{q_2\}, \{a \to q_1, b \to q_2, g(q_1, q_2) \to q_f\} \rangle \\ \mathcal{A}_3 & = & \langle \{q_1, q_2, q_f\}, \{q_f\}, \{q_f\}, \{c \to q_1, b \to q_2, f(q_1, q_2) \to q_f\} \rangle \end{array}$$

$$f(a,b) \xrightarrow[R]{} f(c,b) \xrightarrow[R]{} g(c,b) \xrightarrow[R]{} g(c,c).$$

$$f(a,b) \xrightarrow[\mathcal{A}_1]{} f(q_1,b) \xrightarrow[\mathcal{A}_1]{} f(q_1,q_2) \xrightarrow[\mathcal{A}_1]{} q_f.$$

Controlled term rewriting systems (CntTRS)

Example

$$R = \begin{cases} \mathcal{A}_1 & : & a \rightarrow c \\ \mathcal{A}_2 & : & b \rightarrow c \\ \mathcal{A}_3 & : & f(x,y) \rightarrow g(x,y) \end{cases}$$

where

$$\begin{array}{lcl} & & \quad Q \qquad\qquad\quad F \qquad S \qquad\qquad\qquad\qquad \Delta \\ \mathcal{A}_1 & = & \langle \{q_1, q_2, q_f\}, \{q_f\}, \{q_1\}, \{a \rightarrow q_1, b \rightarrow q_2, f(q_1, q_2) \rightarrow q_f\} \rangle \\ \mathcal{A}_2 & = & \langle \{q_1, q_2, q_f\}, \{q_f\}, \{q_2\}, \{a \rightarrow q_1, b \rightarrow q_2, g(q_1, q_2) \rightarrow q_f\} \rangle \\ \mathcal{A}_3 & = & \langle \{q_1, q_2, q_f\}, \{q_f\}, \{q_f\}, \{c \rightarrow q_1, b \rightarrow q_2, f(q_1, q_2) \rightarrow q_f\} \rangle \end{array}$$

$$f(a,b) \xrightarrow[R]{} f(c,b) \xrightarrow[R]{} g(c,b) \xrightarrow[R]{} g(c,c).$$

$$f(c,b) \xrightarrow[\mathcal{A}_3]{} f(q_1,b) \xrightarrow[\mathcal{A}_3]{} f(q_1,q_2) \xrightarrow[\mathcal{A}_3]{} q_f.$$

Controlled term rewriting systems (CntTRS)

Example

$$R = \begin{cases} \mathcal{A}_1 & : \quad a \to c \\ \mathcal{A}_2 & : \quad b \to c \\ \mathcal{A}_3 & : \quad f(x,y) \to g(x,y) \end{cases}$$

where

$$\begin{array}{lcl} & & \quad Q \qquad\qquad\quad F \qquad S \qquad\qquad\qquad\qquad \Delta \\ \mathcal{A}_1 & = & \langle \{q_1, q_2, q_f\}, \{q_f\}, \{q_1\}, \{a \to q_1, b \to q_2, f(q_1,q_2) \to q_f\} \rangle \\ \mathcal{A}_2 & = & \langle \{q_1, q_2, q_f\}, \{q_f\}, \{q_2\}, \{a \to q_1, b \to q_2, g(q_1,q_2) \to q_f\} \rangle \\ \mathcal{A}_3 & = & \langle \{q_1, q_2, q_f\}, \{q_f\}, \{q_f\}, \{c \to q_1, b \to q_2, f(q_1,q_2) \to q_f\} \rangle \end{array}$$

$$f(a,b) \xrightarrow[R]{} f(c,b) \xrightarrow[R]{} g(c,b) \xrightarrow[R]{} g(c,c).$$

$$g(c,b) \xrightarrow[\mathcal{A}_2]{} g(q_1,b) \xrightarrow[\mathcal{A}_2]{} g(q_1,q_2) \xrightarrow[\mathcal{A}_2]{} q_f.$$

Controlled term rewriting systems (CntTRSs)

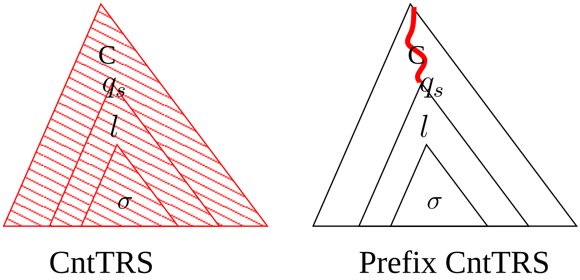
Definition (Rewriting by CntTRS)

$s \xrightarrow[R]{} t$ by the rewrite rule $\mathcal{A} : l \to r$ for a CntTRS R if

$$s = C[l\sigma]_p, t = C[r\sigma]_p \text{ and } C[l\sigma]_p \xrightarrow[\mathcal{A}]{*} C[q]_p \xrightarrow[\mathcal{A}]{*} q^f.$$

where $q \in S$ and $q^f \in F$.

Prefix CntTRSs (pCntTRSs)



Red part in the above figures represent controlled part.

- CntTRS can control whole term.
- Prefix CntTRS can only control prefix.

Generality

(Context-sensitive TRS $\leq$) Prefix CntTRS $\leq$ CntTRS.

Results

- Formalization of controlled term rewriting systems (CntTRSs)
- Showing decidable or undecidable properties about reachability or model checking problem for CntTRSs.
  - Monotonic CntTRSs,
  - Prefix CntTRSs,
  - Non-monotonic CntTRSs, and
  - Recursive prefix CntTRSs.

Decision problems

Reachability problem

- Input : Terms s, t and CntTRSs R.
- Output : Yes if $s \xrightarrow[R]{*} t$, No otherwise.

Regular model-checking problem

- $R^*(L) = \{t \mid \exists s \in L. s \xrightarrow[R]{*} t\}$
- Input : Regular tree language $L_{\mathtt{in}}, L_{\mathtt{err}}$ and CntTRSs R.
- Output : Yes $R^*(L_{\mathtt{in}}) \cap L_{\mathtt{err}} \neq \emptyset$, No otherwise.
  Note that $R^*(L_{\mathtt{in}})$ is the rewrite closure of $L_{\mathtt{in}}$ by R.

Classes of CntTRSs

Definition (Context-free (CF))

A rule $\mathcal{A} : l \to r$ is *context-free* if

- l is of the form $f(x_1, \ldots, x_n)$ where $x_1, \ldots, x_n$ are distinct.

A CntTRS R is *context-free* if each rule of R is context-free.

Definition (Monotonic)

A rule $\mathcal{A} : l \to r$ is *monotonic* if

- $l \to r$ is of the form $C[x_1, \ldots, x_n] \to D[x_1, \ldots, x_n]$ where $x_1, \ldots, x_n$ are distinct variables and $|C| \leq |D|$.

A CntTRS R is *monotonic* if each rule of R is monotonic.

Definition (flat)

A rule $\mathcal{A} : l \to r$ is *flat* if

- the depth of l and r are zero or one.

A CntTRS R is *flat* if each rule of R is flat.

Results

- Formalization of controlled term rewriting systems (CntTRSs)
- Showing decidable or undecidable properties about reachability or model checking problem for CntTRSs.
  - Monotonic CntTRSs,
  - Prefix CntTRSs,
  - Non-monotonic CntTRSs, and
  - Recursive prefix CntTRSs.

Context-sensitive tree grammar (CSTG)

Definition (Tree grammar)

A *tree grammar* is a quadruple $\langle N, \Sigma, S, P \rangle$ that generates a tree language.

- Production rules are term rewriting rules over $N \cup \Sigma$.

Definition (Context-sensitive tree grammar)

Context-sensitive tree grammar (CSTG) is a tree grammar where

- each production rule is monotonic term rewriting rule,
- membership is PSPACE-complete, and
- emptiness is undecidable.

Results for monotonic CntTRSs

Theorem

Given a context-sensitive tree grammar (CSTG) G and a monotonic CntTRS R, we can construct a CSTG generating $R^*(L(G))$.

Corollary

Reachability is PSPACE-complete for monotonic CntTRSs.

Unary signature : Symbols of arity 1 and one constant $\perp$.

Proposition

Reachability is NLINSPACE-complete and regular model checking is undecidable for monotonic flat CntTRSs over unary signatures

Monotonic CntTRS

Theorem

Given a context-sensitive tree grammar (CSTG) G and a monotonic CntTRS R, we can construct a CSTG generating $R^*(L)$.

Proof (sketch).

(a) Simulate rewrite rules (without control)

- to add a symbol $\langle a \rangle$ to N for each $a \in \Sigma$, and
- if $\mathcal{A} : l \to r \in R$ then add $\langle l \rangle \to \langle r \rangle$ to P where $\langle l \rangle$ can be obtained by replacing each symbol f in l by $\langle f \rangle$.

Since $l \to r$ is monotonic, $\langle l \rangle \to \langle r \rangle$ is also monotonic.

Monotonic CntTRS

Proof (sketch).

(b) Simulate control of SA

- to add symbols of the form $\langle f, q\rangle$ to N where $f \in \Sigma$ and $q \in Q$, and
- to add the production rule $\langle f\rangle(\langle f_1, q_1\rangle(\overline{x_1}), \ldots, \langle f_n, q_n\rangle(\overline{x_n})) \to \langle f, q\rangle(\langle f_1, q_1\rangle(\overline{x_1}), \ldots, \langle f_n, q_n\rangle(\overline{x_n}))$ for $f(q_1, \ldots, q_n) \to q \in \Delta$.

where Q and Δ are the set of states and transition rules of SA $\mathcal{A}$, respectively.

□

Monotonic flat CntTRS

Proposition

Reachability is NLINSPACE-complete and regular model checking is undecidable for monotonic flat CntTRS over unary signatures.

Proof (Undecidability).

By reducing emptiness of a linear bounded automaton (LBA) to regular model checking of monotonic flat CntTRSs.

Monotonic flat CntTRS

Proof (Undecidability).

Every configuration of LBA $\mathcal{M}$ is represented by
$\|{:}a_1 \ldots a_{j-1} a_j^p a_{j+1} \ldots a_n{:}\|$.
For the transition $\theta = ((a, p) \to (b, p', left))$ we associate the following rules:

$$\begin{array}{rcl}
\|{:}\Gamma^* c\underline{a^p}\Gamma^*{:}\| & : & a^p(x) \to \langle a^p, \theta\rangle(x) \\
\|{:}\Gamma^* \underline{c}\langle a^p, \theta\rangle\Gamma^*{:}\| & : & c(x) \to \langle c, \theta\rangle(x) \\
\|{:}\Gamma^* \langle c, \theta\rangle\underline{\langle a^p, \theta\rangle}\Gamma^*{:}\| & : & \langle a^p, \theta\rangle(x) \to b(x) \\
\|{:}\Gamma^* \underline{\langle c, \theta\rangle}b\Gamma^*{:}\| & : & \langle c, \theta\rangle(x) \to c^{p'}(x)
\end{array}$$

where Γ is the input alphabet of $\mathcal{M}$.
$R^*(\|{:}S\Gamma^*{:}\|) \cap (\|{:}F\Gamma^*{:}\|) =$ iff $L(\mathcal{M}) =$ where S and F are start and final symbol of $\mathcal{M}$, respectively. □

Results

- Formalization of controlled term rewriting systems (CntTRSs)
- Showing decidable or undecidable properties about reachability or model checking problem for CntTRSs.
  - Monotonic CntTRSs,
  - Prefix CntTRSs,
  - Non-monotonic CntTRSs, and
  - Recursive prefix CntTRSs.

CF non-collapsing pCntTRSs

Proposition

For all CS tree language L over an unary signature Σ, there exists a CF non-collapsing pCntTRS R s.t. $L = R^*(\{c\}) \cap \mathcal{T}(\Sigma)$

Proof.

Every CS language can be generated by a CS grammar with rules of the forms $A \to BC, AB \to AC, A \to a$ ([Penttonen 74]).
CS grammar $G = \langle N, \Sigma, S, P \rangle$ is simulated by the following pCntTRS R:

$$R = \left\{ \begin{array}{rcrcl} \underline{c} & : & c & \to & S(\bot) \\ (N \cup \Sigma)^* \underline{A} & : & A(x) & \to & B(C(x)) \\ (N \cup \Sigma)^* A\underline{B} & : & B(x) & \to & C(x) \\ (N \cup \Sigma)^* \underline{A} & : & A(x) & \to & a(x) \end{array} \right.$$

□

CF non-collapsing pCntTRSs

Corollary

Reachability is PSPACE-complete and regular model checking is undecidable for CF-non-collapsing pCntTRS over unary signatures.

Results

- Formalization of controlled term rewriting systems (CntTRSs)
- Showing decidable or undecidable properties about reachability or model checking problem for CntTRSs.
  - Monotonic CntTRSs,
  - Prefix CntTRSs,
  - Non-monotonic CntTRSs, and
  - Recursive prefix CntTRSs.

Results for non-monotonic CntTRSs

Proposition

Reachability is undecidable for flat CntTRS over unary signatures.

Proposition

Reachability is undecidable for ground CntTRSs.

Theorem

Reachability is decidable in PSPACE for flat non-collapsing pCntTRSs over unary signatures.

Flat CntTRSs

Proposition

Reachability is undecidable for flat CntTRS over unary signatures.

Proof.

Monotonic flat CntTRS simulate LBA.
Non-monotonic flat CntTRSs simulate LBA + tape end = TM.

Reducing halting problem of TM.
Tape end is simulated by the following rules:

- $:\| \rightarrow \flat:\|$ and $\flat:\| \rightarrow :\|$ where $\flat$ is the blank symbol of TM.

□

Ground CntTRSs

Proposition

Reachability is undecidable for ground CntTRSs.

Proof.

- Simulating Turing Machine.
- By representing a word $a_1 \cdots a_n$ as right combs $f(a_1, f(\cdots f(a_n, \perp)))$.
- Simulate TM transitions by controlled rewrite rules of the form $\mathcal{A} : a \to a'$ where $\mathcal{A}$ controls the neighborhood of a.

□

Flat non-collapsing pCntTRSs

Theorem

Reachability is decidable in PSPACE for flat non-collapsing pCntTRS over unary signatures.

Proof.

We show the claim: $u \xrightarrow[R]{*} v$ iff $u = u_0 \xrightarrow[R]{} u_1 \xrightarrow[R]{} \cdots \xrightarrow[R]{} u_k = v$ with $|u_0|, \ldots, |u_k| \leq max(|u|, |v|)$.

(If part) Trivial.

(Only if part) Let $max(|u|, |v|) = M$. Since R is prefix CntTRS,

if

$$\begin{array}{c} u_i = b_1 \cdots b_M \\ \downarrow \\ \vdots \\ \downarrow \\ u_j = b'_1 \cdots b'_M \cdots b'_{n_j} \end{array}$$

then

$$\begin{array}{c} u_i = b_1 \cdots b_M \\ \downarrow \\ \vdots \\ \downarrow \\ u_j[\bot]_M = b'_1 \cdots b'_M \end{array}$$

□

Results

- Formalization of controlled term rewriting systems (CntTRSs)
- Showing decidable or undecidable properties about reachability or model checking problem for CntTRSs.
    - Monotonic CntTRSs,
    - Prefix CntTRSs,
    - Non-monotonic CntTRSs, and
    - Recursive prefix CntTRSs.

An example of recursive pCntTRSs

Example

$$\begin{cases} \mathcal{A}_1 & : & aa & \to & b \\ \mathcal{A}_2 & : & c & \to & d \end{cases} \quad \text{where} \quad \begin{array}{l} L(\mathcal{A}_1) = aa\Sigma^* \\ L(\mathcal{A}_2) = aaaa\Sigma^* \end{array}$$

Then, with recursive control: $aa\underline{aa}c \xrightarrow[R]{} aab\underline{c} \xrightarrow[R]{} aabd$.

- Since $aa \in L(\mathcal{A}_1)$, we have $aa\underline{aa}c \xrightarrow[R]{} aabc$, and
- Since $aaaa \xrightarrow[R]{} aab$ and $aaaa \in L(\mathcal{A}_2)$, we have $aab\underline{c} \xrightarrow[R]{} aabd$.

Recursive pCntTRSs

Definition (Rewriting by recursive pCntTRS)

$s \xrightarrow[R]{} t$ by the rewrite rule $\mathcal{A} : l \to r$ for a recursive pCntTRS R if

$$s = C[l\sigma]_p, t = C[r\sigma]_p \ , \ \exists D.D[x]_p \xrightarrow[R]{*} C[x]_p \wedge D[l\sigma]_p \xrightarrow[\mathcal{A}]{*} D[q_s]_p \xrightarrow[\mathcal{A}]{*} q^f$$

where $q_s \in S$ and $q^f \in F$.

Controlled prefix must belong to the rewrite closure of a given regular language.

A result for recursive pCntTRSs

Theorem

Regular model-checking is decidable in EXPTIME for linear right-shallow recursive pCntTRS.

Proof.

By constructing an alternating tree automaton recognizing $R^*(L)$ for a given recursive pCntTRS R and a tree language L. □

An example of construction of ATA

Example

TA for intial term : $q_5 \xleftarrow{a} q_4 \xleftarrow{a} q_3 \xleftarrow{a} q_2 \xleftarrow{a} q_1 \xleftarrow{c} q_0 \xleftarrow{\perp}$

SA $\mathcal{A}_1$: $u_2 \xleftarrow{a} u_1 \xleftarrow{a} u_s \xleftarrow{\perp,\Sigma} u_0 \xleftarrow{\Sigma} u_0$

SA $\mathcal{A}_2$: $v_4 \xleftarrow{a} v_3 \xleftarrow{a} v_2 \xleftarrow{a} v_1 \xleftarrow{a} v_s \xleftarrow{\perp,\Sigma} v_0 \xleftarrow{\Sigma} u_0$

$$\frac{q_3 \xleftarrow{a} q_2 \xleftarrow{a} q_1 \text{ and } \mathcal{A}_1 : aa \rightarrow b \in R}{q_3 \wedge u_s \xleftarrow{b} q_1}$$

An example of construction of ATA

Example

TA for intial term : $q_5 \xleftarrow{a} q_4 \xleftarrow{a} q_3 \xleftarrow{a} q_2 \xleftarrow{a} q_1 \xleftarrow{c} q_0 \xleftarrow{\perp}$

SA $\mathcal{A}_1$: $u_2 \xleftarrow{a} u_1 \xleftarrow{a} u_s \xleftarrow{\perp,\Sigma} u_0 \xleftarrow{\Sigma} u_0$

SA $\mathcal{A}_2$: $v_4 \xleftarrow{a} v_3 \xleftarrow{a} v_2 \xleftarrow{a} v_1 \xleftarrow{a} v_s \xleftarrow{\perp,\Sigma} v_0 \xleftarrow{\Sigma} u_0$

$$\frac{v_2 \xleftarrow{a} v_1 \xleftarrow{a} v_s \text{ and } \mathcal{A}_1 : aa \to b \in R}{v_2 \wedge u_s \xleftarrow{b} v_s}$$

An example of construction of ATA

Example

TA for intial term : $q_5 \xleftarrow{a} q_4 \xleftarrow{a} q_3 \xleftarrow{a} q_2 \xleftarrow{a} q_1 \xleftarrow{c} q_0 \xleftarrow{\bot}$

SA $\mathcal{A}_1$: $u_2 \xleftarrow{a} u_1 \xleftarrow{a} u_s \xleftarrow{\bot,\Sigma} u_0 \xleftarrow{\Sigma} u_0$

SA $\mathcal{A}_2$: $v_4 \xleftarrow{a} v_3 \xleftarrow{a} v_2 \xleftarrow{a} v_1 \xleftarrow{a} v_s \xleftarrow{\bot,\Sigma} v_0 \xleftarrow{\Sigma} u_0$

$$\frac{q_1 \xleftarrow{c} q_0 \text{ and } \mathcal{A}_2 : c \to d \in R}{q_1 \wedge v_s \xleftarrow{d} q_0}$$

An example of construction of ATA

Example

with these transitions, we have the following runs for 2 descendants of $aaaac$

$$\overset{a}{\left\{\begin{array}{c} q_5 \\ u_2 \end{array}\right\}} \quad \overset{a}{\left\{\begin{array}{c} q_4 \\ u_1 \end{array}\right\}} \quad \overset{b}{\left\{\begin{array}{c} q_3 \\ u_s \end{array}\right\}} \quad \overset{c}{\left\{\begin{array}{c} q_1 \\ u_0 \end{array}\right\}} \quad \overset{\perp}{\left\{\begin{array}{c} q_0 \\ u_0 \end{array}\right\}}$$

$$\overset{a}{\left\{\begin{array}{c} q_5 \\ v_4 \\ u_2 \end{array}\right\}} \quad \overset{a}{\left\{\begin{array}{c} q_4 \\ v_3 \\ u_1 \end{array}\right\}} \quad \overset{b}{\left\{\begin{array}{c} q_3 \\ v_2 \\ u_s \end{array}\right\}} \quad \overset{d}{\left\{\begin{array}{c} q_1 \\ v_s \\ u_0 \end{array}\right\}} \quad \overset{\perp}{\left\{\begin{array}{c} q_0 \\ v_0 \\ u_0 \end{array}\right\}}$$

Conclusion

- Controlled TRS (CntTRS):
  - TRS with selection of rewrite positions using SA.
- Proved decidability or undecidability of reachability and regular model checking for classes CntTRSs.
  - Monotonic CntTRSs,
  - Prefix CntTRSs,
  - Non-monotonic CntTRSs, and
  - Recursive prefix CntTRSs.

Future works

- Reachability is decidable for ground pCntTRS and flat non-collapsing pCntTRS (not unary)?
  - Undecidable for ground CntTRSs.
  - Decidable for flat non-collapsing pCntTRSs over unary signatures
- Generalization of conditional grammars [Dassow et.al. 1997] to trees.

  conditional grammar CF grammars s.t. production sequences can be restricted to a regular language.
- Generalization of CntTRS to unranked trees.
  - XML are represented by unranked trees.