Term Rewriting with Prefix Context Constraints and Bottom-Up Strategies

Florent Jacquemard[1] Yoshiharu Kojima[2] Masahiko Sakai[2]

[1]Ircam, INRIA Paris

[2]Nagoya University

CADE 25 August 4, 2015 Berlin

Summary

- term rewriting rules extended with context constraints (for node selection)
- properties
  - reachability
  - regular tree model checking
  - preservation of regularity
- bottom-up rewrite strategy for decidability

Rewriting and Verification

for rewrite systems $\mathscr{R}$ modeling:

- the transitions of a distributed system [Bouajjani, Touili 2005]
- a communication protocol [Bouajjani et al 2006]
- a security protocol [Genet et al 2009]
- the evaluation of a functional program [Jones, Andersen 2007]
- transformation of rhythms in common western music notation 

$\rightarrow$ 

Reachability Analysis

$$t_{\text{source}} \xrightarrow[\mathscr{R}]{*} t_{\text{target}}$$

Regular Model Checking

composition (Boolean closure)

decision procedures (emptiness)

$$\mathscr{R}^*(L_{\mathsf{in}}) \cap L_{\mathsf{err}} = \emptyset$$

forward closure
of infinite (but regular) configuration set

Static Typechecking [Milo Suciu Vianu 03 JCSS]

The rewrite system $\mathscr{R}$ models a tree transformation.

Verify that the iteration of $\mathscr{R}$ always converts valid input data from a tree set L_{in} into valid output data from a tree set L_{out}.

forward
rewrite closure

$$\mathscr{R}^*(L_{\text{in}}) \subseteq L_{\text{out}}$$

decision procedures

backward
rewrite closure

$$L_{\text{in}} \cap (\mathscr{R}^{-1})^*(\overline{L_{\text{out}}}) = \emptyset$$

composition (Boolean closure)

Rewrite Closure & Tree Automata

if the closure $\mathscr{R}^*(L)$ is effectively regular when L regular (regularity preservation)

- reduce regular model checking or typechecking to tree automata decision problems

$$\mathscr{R}^*(L_{\mathsf{in}}) \cap L_{\mathsf{err}} = \emptyset$$

if not

- upper approximations [Genet et al]
- extend the tree automata model [Rusinowitch et al]
- apply rewrite strategies

Standard Term Rewriting Systems (TRS)

Finite set of rewrite rules of the form $\ell \to r$.

$$a(x) \to b(x)$$

renames a into b.

It can be applied at any position matching $a(x)$
i.e. labelled with a.

Prefix Controlled Term Rewriting Systems (pCTRS)

unary signatures (word rewriting)

$$\Sigma^* b \Sigma^* : a(x) \to b(x)$$

can be applied at any position π matching $a(x)$
such that there is at least one occurrence of b above π.

$$ab\underline{a}c \to abbc$$

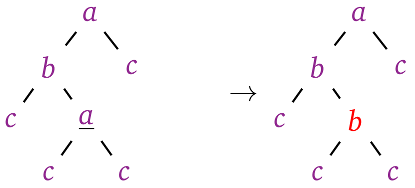
$$aaac \not\to aabc$$

Prefix Controlled Term Rewriting Systems (*p*CTRS)

binary signature (term rewriting)

$$(\Sigma \times \mathbb{N})^* \, \langle b, 2 \rangle \, (\Sigma \times \mathbb{N})^* : a(x_1, x_2) \to b(x_1, x_2)$$

can be applied at any position π labelled with a
such that the path from root to π contains at least one occurrence of b and goes right afterwards.



pCTRS: definition

rules of the form $L : \ell \to r$ with

- $L \subseteq \mathscr{D}ir(\Sigma)^*$, regular
- $\mathscr{D}ir(\Sigma) = \{\langle g, i\rangle \mid g \in \Sigma, 0 < i \leq arity(g)\}$
- ℓ, r terms over Σ and variables

rewrite relation: $t[\ell\sigma]_\pi \xrightarrow[\mathscr{R}]{} t[r\sigma]_\pi$ if $path(t,\pi) \in L$

$$path(g(t_1,\dots,t_n),\varepsilon) = \varepsilon$$
$$path(g(t_1,\dots,t_n), i\cdot\pi) = \langle g, i\rangle \cdot path(t_i,\pi) \quad \text{(with } 1 \leq i \leq n\text{)}$$

Related tree transformation formalisms

- standard TRS are a particular case of pCTRS:
  rules of the form $\mathscr{D}ir(\Sigma)^* : \ell \to r$
- Context Sensitive TRS
  [Futatsugi, Goguen, Jouannaud, Meseguer OBJ2 1985]
  [Lucas 1998]
  are particular cases of *p*CTRS
- XQuery update (W3C)
  with XPath node selection

Linear and Flat pCTRS

[Nagaya, Toyama 2002]

Right-linear and right-flat (uncontrolled) TRS effectively preserve regularity.

Consequence: reachability and RMC are decidable.

Linear and flat pCTRS do not preserve regularity.

$$\mathscr{R} = \begin{array}{lrlll lrlll} (1) & c^*: & c(x) & \to & a'(x) & (2) & c^*a'a^*b^*: & d(x) & \to & b'(x) \\ (3) & c^*: & a'(x) & \to & a(x) & (4) & c^*a^*b^*: & b'(x) & \to & b(x) \end{array}$$

$$\mathscr{R}^*\left(c^*d^*\right) \cap a^*b^* = \{a^nb^m \mid n \geq m\}\text{, not regular}$$

$$c\underline{c}dd \xrightarrow[1]{} ca'\underline{d}d \xrightarrow[2]{} c\underline{a'}b'd \xrightarrow[3]{} ca\underline{b'}d \xrightarrow[4]{} \underline{c}abd \xrightarrow[1]{} a'ab\underline{d} \xrightarrow[2]{} \underline{a'}abb' \xrightarrow[3]{} aab\underline{b'} \xrightarrow[4]{} aabb$$

Generalizable to $\{a^nb^mc^p \mid n \geq m \geq p\}$...

Left–(Linear and Flat) pCTRS

uncontrolled left–(linear and flat) TRS:

- = inverse-monadic
- = production rules of CF tree grammars
- ▸ transform CF tree languages into CF tree languages (reachability and RMC decidable)

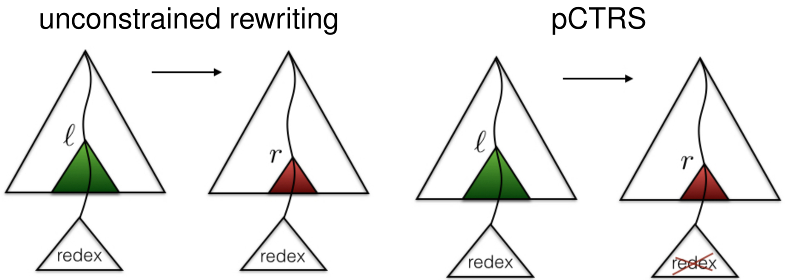
pCTRS simulate CS grammars $\mathscr{G}$ in Pentonnen Normal form:

$\mathscr{G}$	$\mathscr{R}$
$A := BC$	$(\mathrm{NT} \cup \Sigma)^* : \quad A(x) \to B(C(x))$
$AB := AC$	$(\mathrm{NT} \cup \Sigma)^* A : \quad B(x) \to C(x)$
$A := a$	$(\mathrm{NT} \cup \Sigma)^* : \quad A(x) \to a(x)$
$A := \varepsilon$	$(\mathrm{NT} \cup \Sigma)^* : \quad A(x) \to x$

- ▸ reachability undecidable
- ▸ without *collapsing* rules $A(x) \to x$
  reachability PSPACE-complete and RMC undecidable

Sum up bad news

- good decidability properties for (left–) linear and flat uncontrolled TRS
- no good for *p*CTRS!
- simulation of TM or LBA steps: rewriting in both directions
  $c\underline{c}dd \xrightarrow[1]{} ca'\underline{d}d \xrightarrow[2]{} c\underline{a'}b'd \xrightarrow[3]{} ca\underline{b'}d \xrightarrow[4]{} \underline{c}abd \xrightarrow[1]{} a'ab\underline{d} \xrightarrow[2]{} \underline{a'}abb' \xrightarrow[3]{} aab\underline{b'} \xrightarrow[4]{} aab$



- restriction to bottom-up rewriting

Bottom-Up Rewrite Strategy

based on marked terms [Durand, Sénizergues 2007]
= terms over $\Sigma \cup \overline{\Sigma}$, with $\overline{\Sigma} = \{\bar{g} \mid g \in \Sigma\}$.

for t term over Σ:

$\bar{t}$ over $\Sigma \cup \overline{\Sigma}$ is a partially marked copy of t,

$\widetilde{t}$ over $\overline{\Sigma}$ is the fully marked copy of t.

bottom-up rewrite relation:

$$\bar{t}[\overline{\ell\sigma}]_\pi \xrightarrow[\mathscr{R}]{bu} \bar{t}[r\,\widetilde{\sigma}]_\pi$$

for $L : \ell \to r \in \mathscr{R}$ if $path(t,\pi) \in L$ and the root symbol of $\overline{\ell}$ is in Σ.

The derivation $s \xrightarrow[\mathscr{R}]{*} t$ is called bottom-up if $s \xrightarrow[\mathscr{R}]{bu}{}^{*}\ \bar{t}$ for some $\bar{t}$

Bottom-Up Rewriting: Examples

definition: $\bar{t}[\overline{\ell\sigma}]_\pi \xrightarrow[\mathscr{R}]{bu} \bar{t}[r\,\widetilde{\sigma}]_\pi$ with root symbol of $\bar{\ell}$ not marked.

$$\mathscr{R}_1 = \{\varepsilon : h(x) \to g(x),\ \langle h, 1\rangle : a \to b\}$$

$$h(a) \xrightarrow[\mathscr{R}_1]{bu} h(b) \xrightarrow[\mathscr{R}_1]{bu} g(\bar{b})$$

(i.e. $h(a) \xrightarrow[\mathscr{R}_1]{*} g(b)$ is bottom-up).

$$\mathscr{R}_2 = \{\varepsilon : h(x) \to g(x),\ \langle g, 1\rangle : a \to b\}$$

The derivation $h(a) \xrightarrow[\mathscr{R}_2]{*} g(b)$ is not bottom-up:

It holds that $h(a) \xrightarrow[\mathscr{R}_2]{bu} g(\bar{a})$ but not $g(\bar{a}) \xrightarrow[\mathscr{R}_2]{bu} g(\bar{b})$ ($\bar{a} \notin \Sigma$)

and not $h(a) \xrightarrow[\mathscr{R}_2]{bu} h(b)$.

Main Result

Bottom-up rewriting with linear and right shallow *p*CTRS effectively preserves regularity.

principle of the proof: tree automata completion

Given a linear and right-shallow *p*CTRS $\mathscr{R}$

and a tree automaton $\mathscr{A}$

1. combine $\mathscr{A}$ with the automata defining the control languages in rewrite rules of $\mathscr{R}$
(Cartesian product)
2. saturate the product automaton by superposition with the rewrite rules of $\mathscr{R}$

Tree Automata Completion: Example 1

Given $\mathscr{R}_1$ = $\{ \quad \varepsilon : h(x) \to g(x), \quad \langle h, 1\rangle : a \to b \}$

control NFA $\qquad$ control NFA

$v_0 \qquad v_1 \xrightarrow{\langle h,1\rangle} v_2$

Given $\mathscr{A}_1$ recognizing $\{h(a)\}$ with states q_a and q (final)
and transitions $a \to q_a$ and $h(q_a) \to q$

1. start with $\mathscr{A}$ with transitions:

$$
\begin{array}{rcll}
a & \to & \langle q_a, s\rangle & \text{for all } s \subseteq \{v_0, v_1, v_2\} \\
h\big(\langle q_a, \{v_2\}\rangle\big) & \to & \langle q, s\rangle & \text{for all } s \text{ with } \{v_1\} \subseteq s \subseteq \{v_0, v_1, v_2\} \\
h\big(\langle q_a, \emptyset\rangle\big) & \to & \langle q, s\rangle & \text{for all } s \subseteq \{v_0, v_2\}
\end{array}
$$

2. add transitions (superposition of $\mathscr{R}$ into $\mathscr{A}$):

$$
\begin{array}{rcll}
b & \to & \langle q_a, s\rangle & \text{for all } s \text{ with } \{v_2\} \subseteq s \subseteq \{v_0, v_1, v_2\} \\
g\big(\langle q_a, \{v_2\}\rangle\big) & \to & \langle q, s\rangle & \text{for all } s \text{ with } \{v_0, v_1\} \subseteq s \subseteq \{v_0, v_1, v_2\} \\
g\big(\langle q_a, \emptyset\rangle\big) & \to & \langle q, s\rangle & \text{for all } s \text{ with } \{v_0\} \subseteq s \subseteq \{v_0, v_2\}
\end{array}
$$

bottom-up $\mathscr{R}^*\big(\{h(a)\}\big)$: $\qquad g(a) \xrightarrow[\mathscr{A}]{} g\big(\langle q_a, \{v_2\}\rangle\big) \xrightarrow[\mathscr{A}_*]{} \langle q, \{v_0, v_1\}\rangle$

and $\qquad g(b) \xrightarrow[\mathscr{A}_*]{} g\big(\langle q_a, \{v_2\}\rangle\big) \xrightarrow[\mathscr{A}_*]{} \langle q, \{v_0, v_1\}\rangle$

Tree Automata Completion: Example 2

Given $\mathscr{R}_2$ = $\{\quad \varepsilon : h(x) \to g(x), \quad \langle g, 1\rangle : a \to b\}$

control NFA $\qquad$ control NFA

$v_0 \qquad v_1 \xrightarrow{\langle g,1\rangle} v_2$

Given $\mathscr{A}_2$ recognizing $\{h(a)\}$ with states q_a and q (final)
and transitions $a \to q_a$ and $h(q_a) \to q$

1. start with $\mathscr{A}$ with transitions:

$$\begin{array}{rcll} a & \to & \langle q_a, s\rangle & \text{for all } s \subseteq \{v_0, v_1, v_2\} \\ h\big(\langle q_a, \emptyset\rangle\big) & \to & \langle q, s\rangle & \text{for all } s \subseteq \{v_0, v_1, v_2\} \end{array}$$

2. add transitions (superposition of $\mathscr{R}$ into $\mathscr{A}$):

$$\begin{array}{rcll} b & \to & \langle q_a, s\rangle & \text{for all } s \text{ with } \{v_2\} \subseteq s \subseteq \{v_0, v_1, v_2\} \\ g\big(\langle q_a, \emptyset\rangle\big) & \to & \langle q, s\rangle & \text{for all } s \text{ with } \{v_0\} \subseteq s \subseteq \{v_0, v_2\} \end{array}$$

bottom-up $\mathscr{R}^*\big(\{h(a)\}\big)$: $\quad g(a) \xrightarrow[\mathscr{A}]{} g\big(\langle q_a, \emptyset\rangle\big) \xrightarrow[\mathscr{A}_*]{} \langle q, \{v_0, v_1\}\rangle$

and not $\quad h(b) \xrightarrow[\mathscr{A}_*]{} h\big(\langle q_a, s\rangle\big)$ if $v_2 \in s$, stuck!

and not $\quad g(b) \xrightarrow[\mathscr{A}_*]{} g\big(\langle q_a, s\rangle\big)$ if $v_2 \in s$, stuck!

Main Result and Consequences

Bottom-up rewriting with linear and right shallow pCTRS effectively preserves regularity.

Corollary 1

Reachability and RMC wrt bottom-up rewriting are decidable for linear and right shallow pCTRS.

Consequences: left-linear and right-ground *p*CTRS

Corollary 2

Reachability and RMC are decidable for left-linear and right-ground *p*CTRS.

proof: because every rewrite sequence with a right-ground *p*CTRS is bottom-up.

Reachability is PSPACE-hard for ground *p*CTRS.

proof: reduction of the intersection emptiness problem for regular string languages $L_1, \ldots, L_n$ ($n \geq 2$) over Σ.

$$\begin{array}{rcll} \mathscr{R} & = & \{\Sigma^* : \sharp_1 \to a(\sharp_1) \mid a \in \Sigma\} & \cup \quad \{L_i : \sharp_i \to \sharp_{i+1} \mid 1 \leq i \leq n\} \\ & \cup & \{\Sigma^* : a(\sharp_{n+1}) \to \sharp_{n+1} \mid a \in \Sigma\} & \end{array}$$

It holds that $\sharp_1 \xrightarrow[\mathscr{R}]{*} \sharp_{n+1}$ iff $L_1 \cap \cdots \cap L_n \neq \emptyset$

Conclusion

- prefix Controlled Term Rewriting Systems
- decidability of Reachability and Regular Model Checking:
  - wrt bottom-up rewriting for linear and right shallow *p*CTRS
  - for left-linear and right-ground *p*CTRS

Perspectives

- for right–(linear and shallow) *p*CTRS?
- over-approximating constructions handling larger classes of *p*CTRS's
- unranked tree rewriting

Thank You!