Unranked tree rewriting and effective closures of languages

Florent Jacquemard[1] Michael Rusinowitch[2]

[1]Ircam, INRIA Paris-Rocquencourt

[2]LORIA, INRIA Nancy

June 27, 2013

IFIP WG 1.6 meeting – Eindhoven

Term and Unranked Tree Rewriting

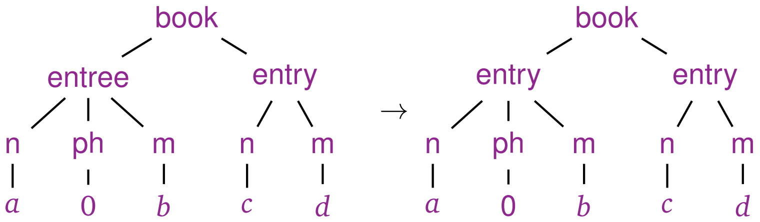
ranked	unranked
finite ranked signature $\Sigma = \{\bot, \top : 0, \neg : 1, \vee, \wedge : 2\}$	finite alphabet $\Sigma = \{\bot, \top, \neg, \vee, \wedge\}$
term := $a($term$_1, \ldots,$ term$_n)$ $a \in \Sigma_n$ $\wedge(t_1, \wedge(t_2, t_3))$	tree := $a($hedge$)$ $a \in \Sigma$ hedge := tree* $\wedge(t_1\ t_2\ t_3)$
rewrite rules	
terms $\times$ terms	hedges $\times$ hedges
substitutions	
variables $\rightarrow$ terms	variables $\rightarrow$ hedges

Unranked Tree Rewriting:
[Löding Spelten 07 MFCS], [Touili 07 VECOS]

Unranked Tree Rewriting — Node renaming

$$\text{entree}(x) \rightarrow \text{entry}(x)$$

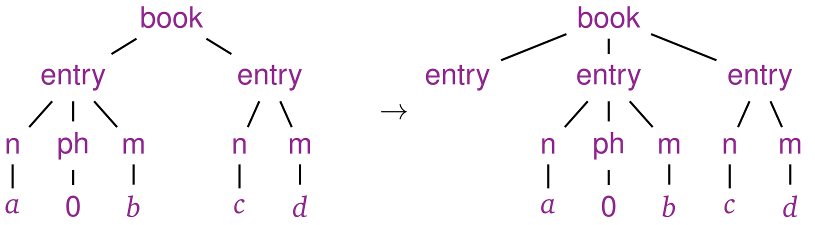
- the rule can be applied to any node labeled by entree
- the variable x ins instantiated by a finite sequence of trees (hedge)



Unranked Tree Rewriting

Insert first

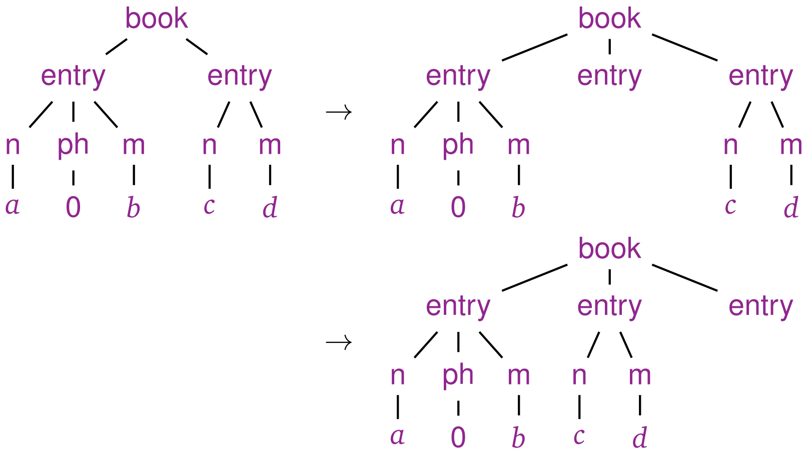
$$\mathsf{book}(x) \rightarrow \mathsf{book}(\mathsf{entry}\; x)$$



Unranked Tree Rewriting

Insert into

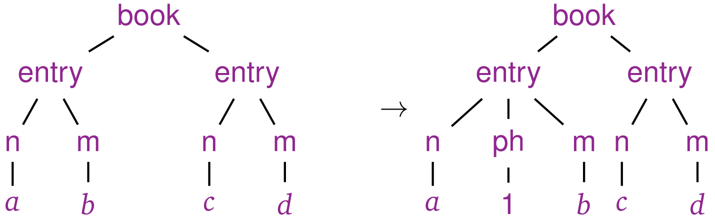
$$\mathsf{book}(x_1\, x_2) \to \mathsf{book}(x_1\ \mathsf{entry}\ x_2)$$



Unranked Tree Rewriting

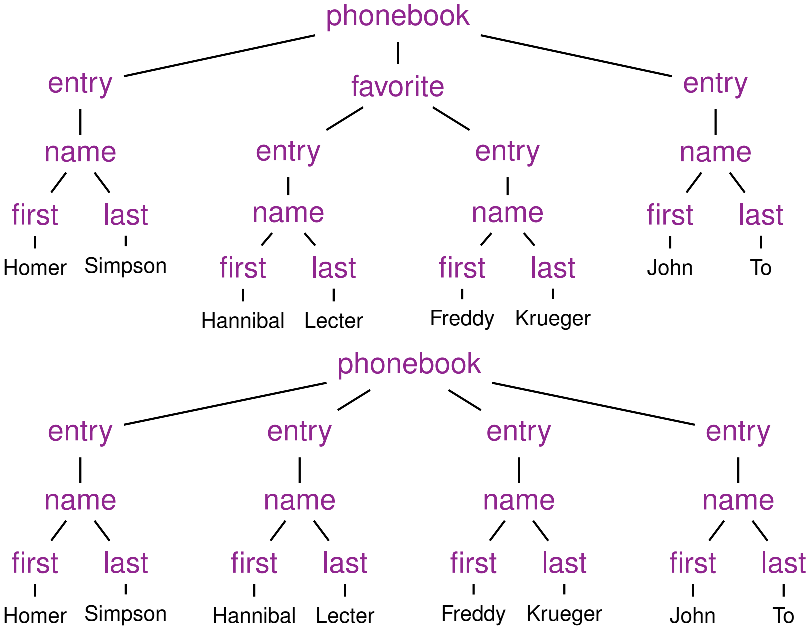
Insert after

$$n(x) \to n(x)\ ph(1)$$



Unranked Tree Rewriting

favorite$(x) \to x$



Collapsing Unranked Tree Rewriting Rule

$$\text{favorite}(x) \to x$$

”delete a single node labeled by favorite”
”move the trees in the sequence of children x up to the position of the deleted node.”

- useful for constructing security views of documents

Motivations and Rewrite Closure

Motivations

Analysis of programs and protocols

- Tree Regular Tree Model Checking

XML processing and verification

- transformations (XSLT), static type checking
- update primitives (XQuery UF), reachability
- consistency of R/W access control policies

Rhythm trees

- tree structured representation of music notation
- simplification of rhythms, decision of equivalences.

Verification of Infinite State Systems

Tree Regular Tree Model Checking [Abdulla et al 2002 CAV]:

- configurations are represented by trees,
- transitions by rewrite rules / tree transducers,
- verification by reachability analysis.

$$\mathscr{R}^*(L_{\mathsf{init}}) \cap L_{\mathsf{error}} = \emptyset$$

higher-order functional programs : [Jones, Andersen 2007], [Kochems Ong 2011 RTA] (collecting semantics)

multithreaded recursive programs:

- term model: [Seidl 2009 IIA] [Bouajjani et al 2000], [Genet, Tong 2001], [Genet, Rusu 2010] .
- unranked tree model: [Bouajjani Touili 2005 RTA].

Static Typechecking

[Milo Suciu Vianu 03 JCSS]

$\mathscr{T}$: tree transducer rewrite system (tree transformation model)

Typechecking:
$\mathscr{T}$ always converts valid input data from a tree set L_{in}
into valid output data from a tree set L_{out}

forward
rewrite closure

$$\mathscr{T}(L_{\text{in}}) \subseteq L_{\text{out}}$$

decision procedures

backward
rewrite closure

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

composition (Boolean closure)

Consistency of R/W Access Control Policies

[Fundulaki Maneth], [Bravo et al, ACCOn] atomic r/w access (updates) modeled by rewrite rules

An ACP is defined by two rewrite systems:

- $\mathscr{R}_+$: authorized operations,
- $\mathscr{R}_-$: forbidden operations.

It is

- inconsistent if one rule of $\mathscr{R}_-$ can be simulated through a sequence of rules of $\mathscr{R}_+$.
- locally inconsistent for a tree t if there exists u such that $t \xrightarrow[\mathscr{R}_-]{} u$ and $t \xrightarrow[\mathscr{R}_+]{*} u$, i.e. $\mathscr{R}_-^1(t) \cap \mathscr{R}_+^*(t) \neq \emptyset$.

Rewrite Closure & Tree Automata

When $\mathscr{R}^*(L)$ or $(\mathscr{R}^{-1})^*(L)$ is effectively regular (for L regular)

- RTMC, typechecking, local inconsistency reduce to tree automata decision problems

  [Milo Suciu Vianu 03 JCSS], [Tosawa 2001]

Otherwise

- approximate

  [Touili Bouajani RTA 05], [Genet, Rusu 2010]...
- extend the tree automata model

Term Rewriting

Ranked Tree Automata

Ranked Tree Automata

$\mathscr{A} = \langle \Sigma, Q, F, \Delta \rangle$ with

- Σ ranked alphabet,
  every symbol has a fixed arity
- Q finite state set,
- $F \subseteq Q$ final states,
- Δ set of transitions $a(q_1, \ldots, q_n) \to q$
  - $a \in \Sigma$, a of arity n
  - $q_1, \ldots, q_n, q \in Q$

Consider Δ as a TRS over $\Sigma \uplus Q$.

Language

$$L(\mathscr{A}, q) = \{t \mid t \xrightarrow[\Delta]{*} q\}$$

$$L(\mathscr{A}) = \bigcup_{q \in F} L(\mathscr{A}, q)$$

Regular sets of terms = ranked tree automata languages

Regularity Preservation

> **[Salomaa 1988]**
>
> linear and right-flat rewrite rules preserve regular languages.

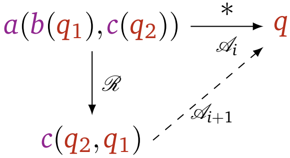
i.e. for all linear and right-flat TRS $\mathscr{R}$,
the forward closure $\mathscr{R}^*(L)$ of a regular language L is regular.

Ranked tree automata completion:

- given $\mathscr{A} = \langle \Sigma, Q, F, \Delta_0 \rangle$ and $\mathscr{R}$ over Σ,
- compute $\mathscr{A}^*$ such that $L(\mathscr{A}^*) = \mathscr{R}^*\big(L(\mathscr{A})\big)$.

By superposition of $\mathscr{A}$'s transitions into $\mathscr{R}$'s rules

for $a(b(x_1), c(x_2)) \to c(x_2, x_1) \in \mathscr{R}$



Also for right-linear and right-flat TRS [Nagaya, Toyama 2002].

Unranked Tree Rewriting

Hedge Automata

Hedge Automata

[Murata 2000]

$\mathscr{A} = (\Sigma, Q, F, \Delta)$ with

- Σ ranked alphabet,
  every symbol has a fixed arity
- Q finite state set,
- $F \subseteq Q$ final states,
- Δ set of transitions $a(L) \rightarrow q$
  - $a \in \Sigma$, $q \in Q$,
  - L regular language over Q^*.

Δ represents the (possible infinite) rewrite system

$$\Delta_\infty = \{a(q_1 \ldots q_n) \rightarrow q \mid a(L) \rightarrow q \in \Delta, q_1 \ldots q_n \in L\}$$

Language

$$L(\mathscr{A}, q) = \{t \mid t \xrightarrow[\Delta_\infty]{*} q\}$$

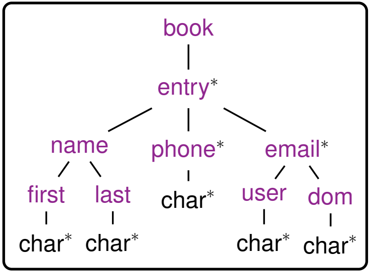
$$L(\mathscr{A}) = \bigcup_{q \in F} L(\mathscr{A}, q)$$

Regular sets of unranked tree = HA languages

$\equiv$ regular term set via binary encodings

Hedge Automata and XML Typing

DTD



corresponding HA

$$
\begin{array}{rcl}
\text{book}(q_{\text{e}}{}^*) & \rightarrow & q_{\text{b}} \\
\text{entry}(q_{\text{n}}\, q_{\text{h}}{}^*\, q_{\text{m}}{}^*) & \rightarrow & q_{\text{e}} \\
\text{name}(q_{\text{f}}\, q_{\text{l}}) & \rightarrow & q_{\text{n}} \\
\text{first}(p^*) & \rightarrow & q_{\text{f}} \\
\text{last}(p^*) & \rightarrow & q_{\text{l}} \\
\text{phone}(p^*) & \rightarrow & q_{\text{h}} \\
\text{email}(q_{\text{u}}\, q_{\text{d}}) & \rightarrow & q_{\text{m}} \\
\text{user}(p^*) & \rightarrow & q_{\text{u}} \\
\text{dom}(p^*) & \rightarrow & q_{\text{d}} \\
\text{a} & \rightarrow & q \\
\text{b} & \rightarrow & q \\
\vdots & &
\end{array}
$$

Hedge Automata: Main Properties

- Boolean closures of recognized languages
- membership $t \in L(\mathscr{A})$ is decidable
  - PTIME when horizontal languages are presented by NFAs
  - NP-complete when horizontal languages are presented by alternating automata
- emptiness: $L(\mathscr{A}) = \emptyset$ is decidable
  - PTIME when horizontal languages are presented by NFAs
  - PSPACE-complete when horizontal languages are presented by alternating automata

HA Preservation

[JR 2008 RTA]

inverse CF rewrite rules $\ell \to a(x)$, $x \in vars(\ell)$ preserve HA.

$$\mathsf{book}\big(\mathsf{entry}(\mathsf{name}(y)\, y_1)\ \mathsf{entry}(\mathsf{name}(y)\, y_2)\ x\big) \to \mathsf{book}(x)$$

exponential construction (needs determinization)

non-trivial combination

- [Touili 07] (unranked trees)
  HA completion, for linear HRS, approximative.
- [Nagaya Toyama 99] (terms)
  TA completion, non-left-linear TRS.

HA Preservation

[JR 2008 RTA]

inverse CF rewrite rules $\ell \to a(x)$, $x \in vars(\ell)$ preserve HA.

exponential construction (needs determinization)

- $\mathscr{A}_0 :=$ determinization of the given HA (subset constr.)
- complete according to this schema

if $\ell\sigma \xrightarrow[\mathscr{A}_i]{*} S$ and $\ell\sigma \xrightarrow{\mathscr{R}} a(x\sigma)$

replace	$a(L') \to S'$	$\in \mathscr{A}_i$
by	$a(L' \cap x\sigma) \to S' \cup S$	$\in \mathscr{A}_{i+1}$
and	$a(L' \setminus x\sigma) \to S'$	$\in \mathscr{A}_{i+1}$

with substitution $\sigma : vars(\ell) \to 2^{(2^Q)^*}$

- invariant: determinism.
- fixpoint: rewrite closure of $L(\mathscr{A}_0)$.

Parametrized Rewrite Systems

Given a fixed HA $\mathscr{B}$ with state set Q
parametrized rewrite rule : symbols of Q allowed in leaves of rhs

$$
\begin{array}{rcll}
\mathsf{book}(x) & \to & \mathsf{book}(q_{\mathsf{e}}\, x) & \text{(insert first)} \\
\mathsf{book}(x_1\, x_2) & \to & \mathsf{book}(x_1\, q_{\mathsf{e}}\, x_2) & \text{(insert into)} \\
\mathsf{name}(x) & \to & \mathsf{name}(x)\, q_{\mathsf{h}} & \text{(insert after)} \\
\mathsf{name}(x) & \to & q_1 \ldots q_n & \text{(replace/delete)}, n \geq 0
\end{array}
$$

semantics of parametrized rewrite system $\mathscr{R}$:

- (possibly infinite) rewrite system $\mathscr{R}/\mathscr{B}$ obtained by replacement of q in rhs by a (ground) tree in $L(\mathscr{B}, q)$.
- Different occurrences of q can be replaced by different trees.

see also [Gilleron 91 STACS], [Löding 02 STACS]

Forward and Backward Closure of Update Primitives

[JR 2010 PPDP]

$a(x)$	$\rightarrow$	$b(x)$	ren
$a(x)$	$\rightarrow$	$a(p\, x)$	$\text{ins}_{\text{first}}$
$a(x)$	$\rightarrow$	$a(x\, p)$	ins_{last}
$a(x_1\, x_2)$	$\rightarrow$	$a(x_1\, p\, x_2)$	ins_{into}
$a(x)$	$\rightarrow$	q_1	rpl_1
$a(x)$	$\rightarrow$	ε	del

preserve HA
polynomial construction

$a(x)$	$\rightarrow$	$p\, a(x)$	$\text{ins}_{\text{before}}$
$a(x)$	$\rightarrow$	$a(x)\, p$	$\text{ins}_{\text{after}}$
$a(x)$	$\rightarrow$	$q_1 \dots q_n$	rpl
$a(x)$	$\rightarrow$	x	dels

do not preserve HA

inverse-preserve HA
exponential construction

Forward and Backward Closure of Update Primitives

HA preservation: by transformation of the horizontal languages (NFA) L in rules $a(L) \rightarrow q$.

ex: for $a(x) \rightarrow b(x) \in \mathscr{R}$ and $a(L) \rightarrow q \in \mathscr{A}$, add $b(L) \rightarrow q$ to $\mathscr{A}$.

ex: for $a(x) \rightarrow a(p\,x) \in \mathscr{R}$, and $a(L) \rightarrow q \in \mathscr{A}$, add a loop labeled with p on the initial state of L.

HA inverse-preservation: automata completion as before.

Unranked Tree Rewriting

CF Hedge Automata

Non preservation of HA

linear & flat rewrite rules do not preserve HA.

RE rewrite closure

$$g(x\,q\,a\,y) \to g(x\,b\,q'\,y)$$

CF rewrite closure (simultaneously insert first and last)

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

CF rewrite closure (simultaneously insert and rename)

$$c_0(x) \to c_1(a\,x),\; c_1(x) \to c_0(x\,b)$$

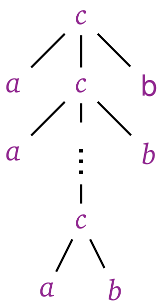
flat and right-ground rewrite rules do not preserve HA.

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

collapsing rewrite rules do not preserve HA.

Non preservation of HA (collapse)

collapsing rewrite rules do not preserve HA.

$\mathscr{R} = \{c(x) \to x\}, L_{\text{in}} = \{$  $\}$ (regular)

$\mathscr{R}^*(L_{in}) \cap a^* \, b^* = \{a^n \, b^n \mid n \geq 0\}$

The rewrite closure is a CF-HA language.

- all these examples are in contrast with the case of terms.
- an extension of HA is needed.

CF Hedge Automata [de la Higuera PhD], [Ohsaki 01 CSL]

$\mathscr{A} = (\Sigma, Q, F, \Delta)$ with

- Σ ranked alphabet,
  every symbol has a fixed arity
- Q finite state set,
- $F \subseteq Q$ final states,
- Δ set of transitions $a(L) \to q$
  - $a \in \Sigma$, $q \in Q$,
  - L is a CF language over Q^*.

Δ represents the (possible infinite) rewrite system

$$\Delta_\infty = \{a(q_1 \ldots q_n) \to q \mid a(L) \to q \in \Delta, q_1 \ldots q_n \in L\}$$

HA $\equiv$ ranked tree automata

CF-HA $\equiv$ ranked tree automata modulo A

CF Hedge Automata: Main Properties

- closure of recognized languages under union
- no closure under intersection and complementation
- closure under intersection with HA

- membership $t \in L(\mathscr{A})$ is decidable in PTIME
- emptiness: $L(\mathscr{A}) = \emptyset$ is decidable in PTIME
  (when horizontal languages are presented by CFG)

Forward Closure of Update Primitives [JR 2010 PPDP]

$a(x) \to b(x)$	ren		
$a(x) \to a(q\,x)$	$\text{ins}_{\text{first}}$	$a(x) \to p\,a(x)$	$\text{ins}_{\text{before}}$
$a(x) \to a(x\,q)$	ins_{last}	$a(x) \to a(x)\,q$	$\text{ins}_{\text{after}}$
$a(x_1\,x_2) \to a(x_1\,q\,x_2)$	ins_{into}		
$a(x) \to q_1$	rpl_1	$a(x) \to q_1 \ldots q_n$	rpl
$a(x) \to \varepsilon$	del	$a(x) \to x$	dels

preserve HA

preserve CF-HA
polynomial construction

inverse-preserve HA
exponential construction

Verification of Inconsistency for Rule Based ACP

An ACP $\langle \mathscr{R}_+, \mathscr{R}_- \rangle$ is locally inconsistent for t
if there exists u such that $t \xrightarrow[\mathscr{R}_-]{} u$ and $t \xrightarrow[\mathscr{R}_+]{*} u$.

equivalently $\mathscr{R}_-^1(t) \cap \mathscr{R}_+^*(t) \neq \emptyset$.

[JR 2010 PPDP]

Local inconsistency is decidable in PTIME for update rules.

proof:

- compute a HA recognizing $\tau_- = \{u \mid t \xrightarrow[\mathscr{R}_-]{} u\}$
- compute a CF-HA recognizing $\tau_+ = \mathscr{R}_+^*\big(\{t\}\big)$
- check that $\tau_- \cap \tau_+ \neq \emptyset$.

CF-HA Preservation

[JR 2008 RTA]

restricted CF rewrite rules $a(x) \to r$ with

1. r linear
2. x at depth ≤ 1 in r

preserve CF-HA.

$$a(x) \to b(x), \quad a(x) \to a(\text{entry}(\text{name}(\text{Homer})), x)$$

polynomial construction
finite transformation of CF grammars for horizontal language.

Non-preservation of CF-HA

flat and non-linear CF rewrite rules do not preserve CF-HA .

$$\text{with } g(x) \to g(x\,x) \quad g(a) \xrightarrow{*} g(a^{2^k})$$

non-shallow CF rewrite rules do not preserve CF-HA .

$$\text{with } a(x) \to b(a(x\,e)) \quad a(c) \xrightarrow{*} \underbrace{b(\ldots b}_{n}(a(c\,e^n)))$$

Unranked Tree Rewriting

CF^2 Hedge Automata

CF^{2}HA

transitions are rewrite rules!

$\mathscr{A} = (\Sigma, Q, F, \Delta)$ with

- Σ ranked alphabet,
  every symbol has a fixed arity
- Q finite state set,
- $F \subseteq Q$ final states,
- Δ set of transitions of the form,

$$\begin{array}{rcl} a & \rightarrow & q \\ p(x) & \rightarrow & q(x) \\ p_1(x)\, p_2 & \rightarrow & q(x) \\ p_1\, p_2(x) & \rightarrow & q(x) \\ p_1\big(p_2(x)\big) & \rightarrow & q(x) \end{array}$$

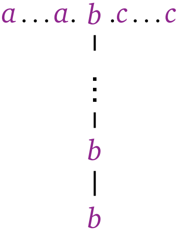
horizontal transitions

vertical transitions

where $p, p_1, p_2 \in Q \cup \Sigma$, $q \in Q$.

equivalently, x can be set to ε.

CF^{2}HA: T-patterns



$\langle\{a,b,c\},\{q_0,q_1,q_2\},\{q_0\},\Delta\rangle$ with

$$\Delta = \left\{ \begin{array}{rcl rcl} b(x) & \rightarrow & q_2(x), & a\, q_1(x) & \rightarrow & q_0(x), \\ q_2(x)\, c & \rightarrow & q_1(x), & q_0(b(x)) & \rightarrow & q_2(x) \end{array} \right\}$$

...not a CF-HA language.

$CF^2HA \supset HA$ and CF-HA

CF-HA with variable-free transitions

$$\begin{aligned} q_1\, q_2 &\rightarrow q \\ a(q_1) &\rightarrow q_2 \\ a &\rightarrow q \end{aligned}$$

where $a \in \Sigma$ and q_1, q_2, q are states.

HA with $Q = Q_{\mathsf{h}} \uplus Q_{\mathsf{v}}$, transitions:

$$\begin{aligned} q_{\mathsf{h}}\, q_{\mathsf{v}} &\rightarrow q'_{\mathsf{h}} \\ a(q_{\mathsf{h}}) &\rightarrow q_{\mathsf{h}} \mid q_{\mathsf{v}} \\ a &\rightarrow q_{\mathsf{h}} \mid q_{\mathsf{v}} \end{aligned}$$

where $a \in \Sigma$, $q_{\mathsf{h}}, q'_{\mathsf{h}} \in Q_{\mathsf{h}}$, $q_{\mathsf{v}} \in Q_{\mathsf{v}}$.

CF^2HA: properties

Membership is decidable in PTIME for CF^2HA.

dynamic programming procedure (CKY like)

Emptiness is decidable in PTIME for CF^2HA.

State marking with 2 marks.

CF^2HA Preservation

[JR 2013 LATA]

CF rewrite rule $a(x) \to r$ where x

- is the only variable in r,
- has at most 1 occurrence in r,
- has no siblings in r.

preserve CF^2HA.

example: T-patterns

$$\mathscr{R} = \left\{ \begin{array}{lllllll} q_0(x) & \to & a\, q_1(x), & & q_1(x) & \to & q_2(x)\, c, \\ q_2(x) & \to & q_0(b(x)), & & q_2(x) & \to & b(x) \end{array} \right\}$$

$$\begin{array}{ll} q_0 & \xrightarrow[\mathscr{R}]{*} a\, q_1 \xrightarrow[\mathscr{R}]{*} a\, q_2\, c \xrightarrow[\mathscr{R}]{*} a\, q_0(b).c \xrightarrow[\mathscr{R}]{*} a\, a\, q_1(b)\, c \\ & \xrightarrow[\mathscr{R}]{*} a\, a\, q_2(b)\, c\, c \xrightarrow[\mathscr{R}]{*} a\, a\, q_0(b(b))\, c\, c \xrightarrow[\mathscr{R}]{*} \ldots \end{array}$$

CF^{2}HA Preservation

Given a CF^{2}HA $\mathscr{A}_{\text{in}} = \langle \Sigma, Q_{\text{in}}, F_{\text{in}}, \Delta_{\text{in}} \rangle$,
we construct a CF^{2}HA $\mathscr{A}$ with state set

$$Q = Q_{\text{in}} \uplus \{\underline{h} \mid h \text{ non-var subhedge of a rhs of } \mathscr{R}\} \uplus \{\underline{a} \mid a \in \Sigma\} \uplus \{q\}$$

and transitions

$$\begin{array}{lcll}
\underline{q_1}(x_1)\ldots\underline{q_n}(x_n) & \to & p(x_1 \ldots x_n) & \mid q_1(x_1)\ldots q_n(x_n) \to p(\bar{x}) \in \Delta_{\text{in}} \\
\underline{q_1}(\underline{q_2}(x)) & \to & p(x) & \mid q_1(q_2(x)) \to p(x) \in \Delta_{\text{in}}
\end{array}$$

$$\begin{array}{lcll}
\underline{t}(x)\ \underline{h} & \to & \underline{t\,h}(x) & \mid x \in t, \underline{t\,h} \in Q \\
\underline{t}(x)\ \underline{h} & \to & q(x) & \mid x \in t, \underline{t\,h} \notin Q \\
\underline{t}\ \underline{h}(x) & \to & \underline{t\,h}(x) & \mid x \notin t, \underline{t\,h} \in Q \\
\underline{t}\ \underline{h}(x) & \to & q(x) & \mid x \notin t, \underline{t\,h} \notin Q \\
\underline{h}(x) & \to & \underline{a}(x) & \mid a(x) \to h \in \mathscr{R}
\end{array}$$

$$\begin{array}{lcll}
a(x) & \to & \underline{a}(x) & \\
a(\underline{h}(x)) & \to & \underline{a(h)}(x) & \mid \underline{a(h)} \in Q \\
a(\underline{h}(x)) & \to & \underline{a}(x) & \mid \underline{a(h)} \notin Q \\
a(q(x)) & \to & \underline{a}(x) &
\end{array}$$

1-childvar Condition

$$\mathscr{R} = \{a(x) \to c\, a(e\, x\, g)\, d\}$$

$$\mathscr{R}^*\big(\{a\}\big) = \{c^n\, a(e^n\, g^n)\, d^n \mid n \geq 1\}$$

seemingly not $\mathrm{CF}^2\mathrm{HA}$.

Forward Closure of Ext^d Update Primitives [JR 2013 LATA]

Given a fixed HA $\mathscr{B}$ with state set Q

$a(x)$	$\rightarrow$	$b(x)$		node renaming	(ren)
$a(x)$	$\rightarrow$	$a(u_1\, x\, u_2)$	$u_1, u_2 \in Q^*$	insertion of child nodes	(ins.c)
$a(x)$	$\rightarrow$	$v_1\, a(x)\, v_2$	$v_1, v_2 \in Q^*$	insertion of sibling nodes	(ins.s)
$a(x)$	$\rightarrow$	$b\big(a(x)\big)$		insertion of a parent node	(ins.p)
$a(x)$	$\rightarrow$	u	$u \in Q^*$	node replacement recursive deletion	(rpl)
$a(x)$	$\rightarrow$	x		node deletion	(del)

preserve CF-HA.

The proof uses the CF^{2}HA presentation of CF-HA.

Loop-Free Update Rewrite Systems

$\mathscr{R}$ loop-free if there exists no sequence $a_1, \ldots, a_n$ $(n > 1)$ such that for all $1 \leq i < n$, $a_i(x) \to a_{i+1}(x) \in \mathscr{R}$ and $a_1 = a_n$.

Transformation of an update rewrite system $\mathscr{R}$
into a loop-free update rewrite system $\hat{\mathscr{R}}$
by selecting a representative $\hat{a}$ of a in every a's loop
and suppressing loops.

In the construction of an automaton for $\mathscr{R}^*$,
it is sufficient to consider the rewrite closure by $\hat{\mathscr{R}}$.

Closure under Update Rewrite Systems

Rewrite closure of CF-HA is CF-HA
for every loop-free update rewrite system.

Initialize, given CF-HA $\mathscr{A}_{\text{in}} = \langle \Sigma, Q_{\text{in}}, F_{\text{in}}, \Delta_{\text{in}} \rangle$

$$\begin{aligned}\Delta_0 = \Delta_{\text{in}} \quad & \cup \quad \{q_{a_1} \to q\} \\ & \cup \quad \{a_n(q^{a_1 \ldots a_n}) \to q_{a_1 \ldots a_n} \mid a_1, \ldots, a_n \text{ is a renaming chain}\}\end{aligned}$$

Completion

	$\mathscr{R}$ contains	$\Delta_{i+1} = \Delta_i \cup$
(ren)	$a_n(x) \to b(x)$	$\{q^{a_1 \ldots a_n} \to q^{a_1 \ldots a_n b} \mid q^{a_1 \ldots a_n b} \in Q\}$ $\cup \{q_{a_1 \ldots a_n b} \to q_{a_1 \ldots a_n} \mid q_{a_1 \ldots a_n b} \in Q\}$
(ins.c)	$a_n(x) \to a_n(u\,x\,v)$	$\{u\, q^{a_1 \ldots a_n}\, v \to q^{a_1 \ldots a_n} \mid q^{a_1 \ldots a_n} \in Q\}$
(ins.s)	$a_n(x) \to u\, a_n(x)\, v$	$\{u\, q_{a_1 \ldots a_n}\, v \to q_{a_1 \ldots a_n} \mid q_{a_1 \ldots a_n} \in Q\}$
(ins.p)	$a_n(x) \to b(a_n(x))$	$\{b(q_{a_1 \ldots a_n}) \to q_{a_1 \ldots a_n} \mid q_{a_1 \ldots a_n} \in Q\}$
(rpl)	$a_n(x) \to u$	$\{u \to q_{a_1 \ldots a_n} \mid q_{a_1 \ldots a_n} \in Q\}$
(del)	$a_n(x) \to x$	$\{q^{a_1 \ldots a_n} \to q_{a_1 \ldots a_n} \mid q^{a_1 \ldots a_n} \in Q\}$

Synchronized rename and insert

$$\mathscr{R} = \{a(x) \to c\, a(e\, x\, g)\, d\}$$

$$\mathscr{R}^*\big(\{a\}\big) = \{c^n\, a(e^n\, g^n)\, d^n \mid n \geq 1\}$$

$$\mathscr{R}' = \left\{ \begin{array}{llll} a(x) & \to & c\, a'(x)\, d, & \text{inv-monadic, 1-childvar} \\ & & & \in (\text{ins.s}) + (\text{ren}) \\ a'(x) & \to & a(e\, x\, g) & \notin \text{1-childvar} \\ & & & \in (\text{ins.c}) + (\text{ren}) \end{array} \right\}$$

Conclusion

- decidable models CF-HA, CF^2HA of unranked ordered tree recognizers extending hedge automata
- captures forward/backward rewrite closure under
  - various families of unranked tree rewrite systems with (inverse-) CF rules
  - parametric rewrite systems modeling update primitives

Perspectives

- case of unranked unordered trees
- counting constraints on horizontal and vertical paths

...Thank You!