Rewrite Closure and CF Hedge Automata

Florent Jacquemard[1] Michael Rusinowitch[2]

[1]Ircam, INRIA Paris-Rocquencourt

[2]LORIA, INRIA Nancy

April 4, 2013

LATA 2013 – Bilbao

Static Typechecking [Milo Suciu Vianu 03 JCSS]

Consider a tree transformation modeled by a rewrite system $\mathscr{R}$.

Typechecking:
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

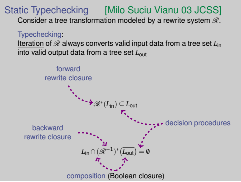
$$\mathscr{R}^*(L_{\text{in}}) \subseteq L_{\text{out}}$$

decision procedures

backward
rewrite closure

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

composition (Boolean closure)



- hence TA are also useful for XML reasoning tasks
- like for instance the problem of typechecking.
- It is the verification that a tree transformations T defined by a program in a language like XSLT always converts valid input data into valid output data,
- input and output types are defined by tree automata
- and if the forward closure by T of input data L_{in} is regular, then type checking is reduced to the problem of inclusion for TA (decidable in EXPTIME)
- if this is not the case, like in the cited paper, a second option is to consider the backward closure of the complement of the output data L_{out}, and, if it is regular, typechecking is reduced to testing the emptiness of the intersection with the input type L_{in}.

Consistency of R/W Access Control Policies

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 document 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

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

- reduce typechecking and local inconsistency
  to tree automata decision problems

if not

- approximate
- extend the tree automata model

Rewrite Closure & Tree Automata

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

- reduce typechecking and local inconsistency to tree automata decision problems

if not

- approximate
- extend the tree automata model

- plan: introduce new model automata for unranked trees, given 2 results of automata construction for rewrite closure

- here: we consider unranked - but start with case of ranked trees (better known and simpler)

(parenthesis)

The Case of Ranked Trees

The Case of Ranked Trees

Σ is a ranked alphabet, every symbol has a fixed arity

A Term Rewriting System $\mathscr{R}$ is a finite set of rewrite rules $\ell \to r$, where ℓ and r are ranked trees (terms) with variables.

Rewriting:

- pattern matching of ℓ and replacement by r
- substitution of variables by ranked trees

Rewrite relation denoted $s \xrightarrow[\mathscr{R}]{} t$,
reflexive transitive closure $s \xrightarrow[\mathscr{R}]{*} t$.

Rewrite closure of a term set L: $\mathscr{R}^*(L) = \{t \mid \exists s \in L, s \xrightarrow[\mathscr{R}]{*} t\}$

Ranked Tree Automata

Ranked Tree Automata

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

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

Language

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

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

Rewrite closure of Ranked Tree Automata Languages

Ranked tree automata completion:

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

For linear and right-flat TRS $\mathscr{R}$:
superposition of $\mathscr{R}$'s rules into $\mathscr{A}$'s transitions

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

$$\begin{tikzcd}
a(b(q_1), c(q_2)) \arrow[r, "*", "\Delta_i"'] \arrow[d, "*"', "\mathscr{R}"] & q \\
c(q_2, q_1) \arrow[ur, dashed, "*", "\Delta_{i+1}"'] &
\end{tikzcd}$$

Application to static analysis of functional programs:
Tree Regular Model Checking [Bouajjani et al 2002]

(end of parenthesis)

Unranked Trees

Unranked Ordered Trees and Rewriting

Σ unranked alphabet

hedge	=	finite sequence of unranked trees (possibly ε)
unranked tree	=	variable
		a(hedge) with $a \in \Sigma$

Rewrite System: finite set of rewrite rules $\ell \to r$,
where ℓ and r are hedges with variables.

Rewriting:

- pattern matching and replacement
- replacement of variables by hedges

Hedge Automata and CF Hedge Automata

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

- alphabet Σ,
- finite state set Q,
- final states $F \subseteq Q$,
- set Δ of transitions $a(L) \to p$ with
  - $a \in \Sigma$,
  - $p \in Q$,
  - L:

HA	[Murata 00]:	L is a regular language over Q^*,
CF-HA	[Ohsaki et al 01]:	L is a CF language over Q^*.

Language

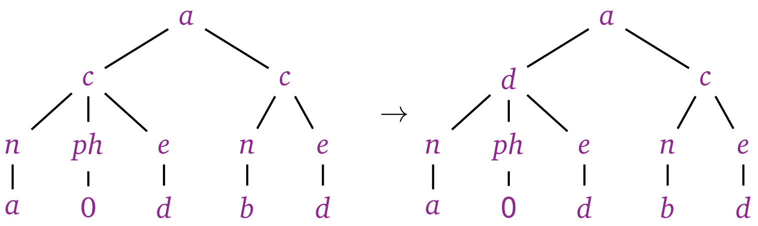
$$L(\mathscr{A}, q) = \{t \mid t \xrightarrow[\Delta]{*} q\}$$

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

Regular set of unranked tree = HA language.

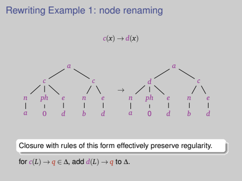
Rewriting Example 1: node renaming

$$c(x) \to d(x)$$



Closure with rules of this form effectively preserve regularity.

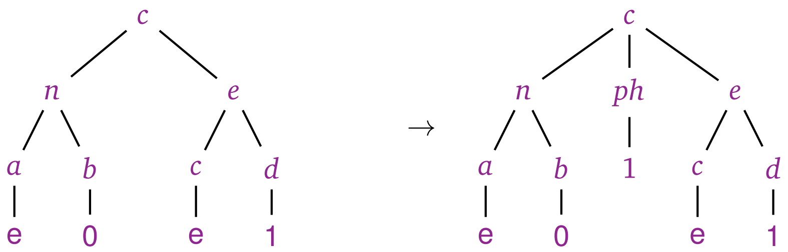
for $c(L) \to q \in \Delta$, add $d(L) \to q$ to Δ.



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

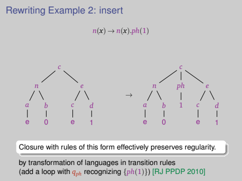
Rewriting Example 2: insert

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



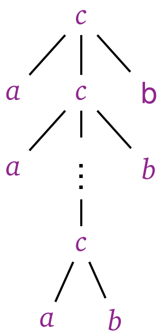
Closure with rules of this form effectively preserves regularity.

by transformation of languages in transition rules
(add a loop with q_{ph} recognizing $\{ph(1)\}$) [RJ PPDP 2010]



- the right hand side of this rule is an hedge of length 2 (not a tree)

Rewriting Example 3: collapse

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

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

The rewrite closure is a CF-HA language.

also for inserts $\mathscr{R} = \{c(x) \to c(a.x.b)\}$
and $\mathscr{R} = \{c_0(x) \to c_1(a.x),\ c_1(x) \to c_0(x.b)\}$

Rewriting Example 4: T-patterns

$$\mathscr{R} = \left\{ \begin{array}{lcllcl} p_0(x) & \to & a.p_1(x), & p_1(x) & \to & p_2(x).c, \\ p_2(x) & \to & p_0(b(x)), & p_2(x) & \to & b(x) \end{array} \right\}$$

$$\begin{array}{ll} p_0 & \xrightarrow[\mathscr{R}]{*} a.p_1 \xrightarrow[\mathscr{R}]{*} a.p_2.c \xrightarrow[\mathscr{R}]{*} a.p_0(b).c \xrightarrow[\mathscr{R}]{*} a.a.p_1(b).c \\ & \xrightarrow[\mathscr{R}]{*} a.a.p_2(b).c.c \xrightarrow[\mathscr{R}]{*} a.a.p_0(b(b)).c.c \xrightarrow[\mathscr{R}]{*} \ldots \end{array}$$

$\mathscr{R}^*(\{p_0\})$ without p_0, p_1, p_2 is the set of trees

$$\begin{array}{c} a \ldots a.\ b\ .c \ldots c \\ | \\ \vdots \\ | \\ b \\ | \\ b \end{array}$$

with the same number of a, b and c.

The rewrite closure is not a CF-HA language.

CF^{2}HA

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

- alphabet Σ,
- finite state set Q,
- final states $F \subseteq Q$,
- set Δ of transitions of the form,

$$\begin{array}{rcll} p_1(x_1)\dots p_n(x_n) & \rightarrow & q(x_1\dots x_n) & \text{horizontal transitions,} \\ p_1\big(p_2(x)\big) & \rightarrow & q(x) & \text{vertical transitions.} \end{array}$$

where $p_1, \dots, p_n \in Q \cup \Sigma$, $q \in Q$ and $n \geq 0$

equivalent form:

$$\begin{array}{rcl} p_1(\delta_1)\dots p_n(\delta_n) & \rightarrow & q(\delta_1\dots\delta_n) \\ p_1(p_2(\delta_1)) & \rightarrow & q(\delta_1) \end{array} \qquad n > 0$$

where every δ_i is either a variable x_i or ε.

CF^{2}HA: T-patterns

$$\begin{array}{c} a \ldots a.\ b\ .c \ldots c \\ | \\ \vdots \\ | \\ b \\ | \\ b \end{array}$$

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

$$\Delta = \left\{ \begin{array}{ll} b(x_1) \to q_2(x_1), & a.q_1(x_2) \to q_0(x_2), \\ q_2(x_1).c \to q_1(x_1), & q_0(b(x)) \to q_2(x) \end{array} \right\}$$

$\mathrm{CF}^2\mathrm{HA} \supset$ HA and CF-HA

CF-HA with variable-free transitions

$$\begin{array}{rcl} q_1 \ldots q_n & \to & q \\ a(q_1) & \to & q_2 \end{array}$$

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

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

$$\begin{array}{rcl} \varepsilon & \to & q_\mathsf{h} \\ q_\mathsf{h}.q_\mathsf{v} & \to & q'_\mathsf{h} \\ a(q_\mathsf{h}) & \to & q_\mathsf{v} \end{array}$$

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 for CF^2HA.

PSPACE procedure

Emptiness is decidable in PTIME for CF^2HA.

State marking with 2 marks.

Rewrite Closure under

Inverse-Monadic and 1-childvar Rewrite Systems

Inverse-Monadic and 1-childvar Rewrite Systems

Rewrite closure of $\mathrm{CF}^2\mathrm{HA}$ is $\mathrm{CF}^2\mathrm{HA}$ for rules of the form $a(x) \to r$ where x

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

example: T-patterns

$$\mathscr{R} = \left\{ \begin{array}{lllllll} p_0(x) & \to & a.p_1(x), & \quad & p_1(x) & \to & p_2(x).c, \\ p_2(x) & \to & p_0(b(x)), & \quad & p_2(x) & \to & b(x) \end{array} \right\}$$

Inverse-Monadic and 1-childvar Rewrite Systems

Rewrite closure of CF^{2}HA is CF^{2}HA for rules of the form $a(x) \to r$ where x only var., has 1 occ. and no siblings in r.

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

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

and transitions

$$\begin{array}{rcll}
\underline{p_1}(x_1)\ldots\underline{p_n}(x_n) & \to & p(x_1\ldots x_n) & \mid p_1(x_1)\ldots p_n(x_n) \to p(\bar{x}) \in \Delta_{\mathsf{in}} \\
\underline{p_1}(\underline{p_2}(x)) & \to & p(x) & \mid p_1(p_2(x)) \to p(x) \in \Delta_{\mathsf{in}}
\end{array}$$

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

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

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 CF^{2}HA.

Rewrite Closure under Update Rewrite Systems

Update Rewrite Rules

We assume a fixed HA $\mathscr{A} = \langle \Sigma, Q, F, \Delta \rangle$.

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

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

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 $\mathscr{A}_{\mathsf{in}} = \langle \Sigma, Q_{\mathsf{in}}, F_{\mathsf{in}}, \Delta_{\mathsf{in}} \rangle$

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

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)	$a_n(x) \to x$	$\{q^{a_1 \ldots a_n} \to q_{a_1 \ldots a_n} \mid q^{a_1 \ldots a_n} \in Q\}$

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)	$a_n(x) \to x$	$\{q^{a_1 \ldots a_n} \to q_{a_1 \ldots a_n} \mid q^{a_1 \ldots a_n} \in Q\}$

principle of proof:
- construction of a new transition set for the rewrite closure,
  by iterative completion of Δ_{in} according to $\mathscr{R}$
- insertion rules depend on the label of the current node,

$\Rightarrow$ the automaton must check the label in order to be sure that an insert was possible
- there can be interleavings of (ren) and (ins) of $\neq$ kinds (e.g. child and sibling), $\neq$ (ins) checked at $\neq$ levels of bottom-up computation

$\Rightarrow$ store the renaming chains in states.
- $\mathscr{R}$ loop-free $\Rightarrow$ length of every renaming chain is bounded by $|\mathscr{R}|$.
- 2 modes for states:

– *push* mode for $q^{a_1 \ldots a_n}$: for ins sibling (performed first)

– *pop* mode for $q_{a_1 \ldots a_n}$: for ins child (performed second)

Synchronized rename and insert

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

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

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

Conclusion

- decidable model CF^2HA of unranked tree recognizer extending hedge automata
- captures the rewrite closure under
  - linear, inverse-monadic, 1-childvar rewrite systems
  - update (parametric) rewrite systems

Perspectives

- intersection with regular tree languages
- counting constraints on horizontal and vertical paths
- closure under (maximal) parallel rewriting [Solimando, Delzanno, Guerrini Games 2012]

T

Rule based ACPs

[Fundulaki Maneth][Bravo et al, ACCOn]

An access control policy (ACP) given by two finite sets of rules

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

example

- $\mathscr{R}_+ = \left\{ \begin{array}{rcl} \mathsf{addressbook}(x) & \rightarrow & \mathsf{addressbook}(p_{\mathsf{ec}}\, x), \\ \mathsf{card}(x) & \rightarrow & () \end{array} \right\}$
  - user can insert card with name, delete card.
- $\mathscr{R}_- = \{\mathsf{name}(x) \rightarrow p_{\mathsf{n}}\}$
  - user cannot change a name.

Closure under Update Rewrite Systems

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

normalized CF-HA $\langle \Sigma, Q, Q^{\mathsf{f}}, \Delta \rangle$ for all $a \in \Sigma$ and $q \in Q$, there exists one unique $q^a \in Q$ such that $a(q^a) \to q \in \Delta$, and moreover, q^a does neither occur in a left hand side of an horizontal transition of Δ nor in a right hand side of a vertical transition of Δ.

$$Q = P \cup \{ q_a \mid q^a \in P_{\mathsf{in}} \} \cup \left\{ \begin{array}{c} q^{a_1 \ldots a_n} \\ q_{a_1 \ldots a_n} \end{array} \middle| \begin{array}{l} q \in P \setminus P_{\mathsf{in}}, n \geq 2, \\ a_1, \ldots, a_n \text{ is a renaming chain} \end{array} \right\}$$

Closure under Update Rewrite Systems

$$\Delta_0 = \Gamma_h \quad \cup \quad \{q_{a_1} \to q \mid q_{a_1} \in Q\}$$
$$\qquad\quad \cup \quad \{a_n\left(q^{a_1\ldots a_n}\right) \to q_{a_1\ldots a_n} \mid q^{a_1\ldots a_n}, q_{a_1\ldots a_n} \in Q, n \geq 1\}$$

where Γ_h is the subset of horizontal transitions of Γ.

	$\mathscr{R}$ 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)	$a_n(x) \to a_n(uxv)$	$\{u\, q^{a_1\ldots a_n}\, v \to q^{a_1\ldots a_n} \mid q^{a_1\ldots a_n} \in Q\}$
(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)	$a_n(x) \to b\left(a_n(x)\right)$	$\{b\left(q_{a_1\ldots a_n}\right) \to q_{a_1\ldots a_n} \mid q_{a_1\ldots a_n} \in Q\}$
(rpl)	$a_n(x) \to u$	$\{u \to q_{a_1\ldots a_n} \mid q_{a_1\ldots a_n} \in Q\}$
(del)	$a_n(x) \to x$	$\{q^{a_1\ldots a_n} \to q_{a_1\ldots a_n} \mid q^{a_1\ldots a_n} \in Q\}$