

Reducing Rewrite Properties to Properties on Ground Terms*

Alexander Lochmann

September 13, 2023

Abstract

This AFP entry relates important rewriting properties between the set of terms and the set of ground terms induced by a given signature. The properties considered are confluence, strong/local confluence, the normal form property, unique normal forms with respect to reduction and conversion, commutation, conversion equivalence, and normalization equivalence.

Contents

1	Introduction	3
2	Preliminaries	4
2.1	Additional operations on terms and positions	4
2.1.1	Linearity	4
2.1.2	Positions induced by contexts, by variables and by given subterms	4
2.1.3	Replacing functions symbols that aren't specified in the signature by variables	4
2.1.4	Replace term at a given position in contexts	5
2.1.5	Multihole context closure of a term relation as inductive set	5
2.2	Destruction and introduction of <i>all-ctxt-closed</i>	5
2.3	Lemmas for <i>poss</i> and ordering of positions	5
2.4	Lemmas for $(-)$ and <i>replace-term-at</i>	6
2.5	<i>term-to-sig</i> invariants and distributions	8
2.6	Misc	8

*Supported by FWF (Austrian Science Fund) projects P30301.

3	Rewriting	10
3.1	Basic rewrite definitions	10
3.1.1	Rewrite steps with implicit signature declaration (en- coded in the type)	10
3.1.2	Restrict relations to terms induced by a given signature	11
3.1.3	Rewriting under a given signature/restricted to ground terms	11
3.1.4	Rewriting sequences involving a root step	11
3.2	Monotonicity laws	11
3.3	Introduction, elimination, and destruction rules for <i>sig-step</i> , <i>rstep</i> , <i>rrstep</i> , <i>srrstep</i> , and <i>srstep</i>	12
3.3.1	Transitive and reflexive closure distribution over <i>sig-step</i>	13
3.3.2	Distributivity laws	14
3.4	Substitution closure of <i>srstep</i>	15
3.5	Context closure of <i>srstep</i>	16
3.6	Removing/Replacing constants in a rewrite sequence that do not appear in the rewrite system	17
3.6.1	Removal lemma applied to various rewrite relations	18
4	Confluence related rewriting properties	20
4.1	Confluence related ARS properties	20
4.2	Signature closure of relation to model multihole context closure	20
4.3	Specific results about rewriting under a linear variable-separated system	26
4.4	Specific results about rewriting under a ground system	31
4.5	funas	32
5	Reducing Rewrite Properties to Properties on Ground Terms over Left-Linear Right-Ground Systems	33
6	LLRG results	33
6.1	Specialized for monadic signature	34
7	Reducing Rewrite Properties to Properties on Ground Terms over Ground Systems	35
8	Reducing Rewrite Properties to Properties on Ground Terms over Linear Variable-Separated Systems	37
9	Linear-variable separated results	38
9.1	Specialized for monadic signature	41

1 Introduction

Rewriting is an abstract model of computation. Among other things, it studies important properties including the following:

CR:	$\forall s \forall t \forall u (s \rightarrow^* t \wedge s \rightarrow^* u \implies t \downarrow u)$	confluence
SCR:	$\forall s \forall t \forall u (s \rightarrow t \wedge s \rightarrow u \implies \exists v (t \rightarrow^= v \wedge u \rightarrow^* v))$	strong confluence
WCR:	$\forall s \forall t \forall u (s \rightarrow t \wedge s \rightarrow u \implies t \downarrow u)$	local confluence
NFP:	$\forall s \forall t \forall u (s \rightarrow^* t \wedge s \rightarrow^! u \implies t \rightarrow^! u)$	normal form property
UNR:	$\forall s \forall t \forall u (s \rightarrow^! t \wedge s \rightarrow^! u \implies t = u)$	unique normal forms with respect to reduction
UNC:	$\forall t \forall u (t \leftrightarrow^* u \wedge \text{NF}(t) \wedge \text{NF}(u) \implies t = u)$	unique normal forms with respect to conversion

We also consider the following properties involving two TRSs \mathcal{R} and \mathcal{S} :

COM:	$\forall s \forall t \forall u (s \rightarrow_{\mathcal{R}}^* t \wedge s \rightarrow_{\mathcal{S}}^* u \implies \exists v (t \rightarrow_{\mathcal{S}}^* v \wedge u \rightarrow_{\mathcal{R}}^* v))$	commutation
CE:	$\forall s \forall t (s \leftrightarrow_{\mathcal{R}}^* t \iff s \leftrightarrow_{\mathcal{S}}^* t)$	conversion equivalence
NE:	$\forall s \forall t (s \rightarrow_{\mathcal{R}}^! t \iff s \rightarrow_{\mathcal{S}}^! t)$	normalization equivalence

An interesting observation is that for each of these properties there exists a rewrite system that satisfies the property when restricted to ground terms but not when arbitrary terms are allowed. Consider the left-linear right-ground TRS \mathcal{R} consisting of the rules

$$a \rightarrow b \qquad f(a, x) \rightarrow b \qquad f(b, b) \rightarrow b$$

over the signature $\mathcal{F} = \{a, b, f\}$. It is ground-confluent because every ground term in $\mathcal{T}(\mathcal{F})$ rewrites to b . Confluence does not hold; the term $f(a, x)$ rewrites to the different normal forms b and $f(b, x)$.

In this AFP entry, properties on arbitrary terms are reduced to the corresponding properties on ground terms, for left-linear right-ground rewrite systems and for linear variable-separated systems. To do this, I formalized fundamental term rewriting operations that include the root step and the one step rewriting relations. Also, I added definitions for conversion equivalence, normalization equivalence, strong confluence and the normal form property extending the list of important rewriting properties of the AFP entry ‘‘Abstract Rewriting’’ [1].

Rewrite sequences that contain a root step play an important role in the formalization. The table of contents should give the reader a good overview of the content of this entry.

2 Preliminaries

theory *Terms-Positions*
imports *Regular-Tree-Relations.Ground-Terms*
begin

2.1 Additional operations on terms and positions

2.1.1 Linearity

fun *linear-term* :: ('f, 'v) term \Rightarrow bool **where**
linear-term (Var -) = True |
linear-term (Fun f ts) = (is-partition (map vars-term ts) \wedge ($\forall t \in \text{set } ts.$ *linear-term* t))
abbreviation *linear-sys* $\mathcal{R} \equiv \forall (l, r) \in \mathcal{R}. \text{linear-term } l \wedge \text{linear-term } r$

2.1.2 Positions induced by contexts, by variables and by given subterms

definition *possc* C = {p | p t. p \in poss C(t)}
definition *varposs* s = {p | p. p \in poss s \wedge is-Var (s |- p)}
definition *poss-of-term* u t = {p. p \in poss t \wedge t |- p = u}

2.1.3 Replacing functions symbols that aren't specified in the signature by variables

definition *funas-rel* $\mathcal{R} = (\bigcup (l, r) \in \mathcal{R}. \text{funas-term } l \cup \text{funas-term } r)$

fun *term-to-sig* **where**
term-to-sig \mathcal{F} v (Var x) = Var x
| *term-to-sig* \mathcal{F} v (Fun f ts) =
(if (f, length ts) \in \mathcal{F} then Fun f (map (term-to-sig \mathcal{F} v) ts) else Var v)

fun *ctxt-well-def-hole-path* **where**
ctxt-well-def-hole-path \mathcal{F} Hole \longleftrightarrow True
| *ctxt-well-def-hole-path* \mathcal{F} (More f ss C ts) \longleftrightarrow (f, Suc (length ss + length ts)) \in $\mathcal{F} \wedge \text{ctxt-well-def-hole-path } \mathcal{F} C$

fun *inv-const-ctxt* **where**
inv-const-ctxt \mathcal{F} v Hole = Hole
| *inv-const-ctxt* \mathcal{F} v ((More f ss C ts))
= (More f (map (term-to-sig \mathcal{F} v) ss) (inv-const-ctxt \mathcal{F} v C) (map (term-to-sig \mathcal{F} v) ts))

fun *inv-const-ctxt'* **where**
inv-const-ctxt' \mathcal{F} v Hole = Var v
| *inv-const-ctxt'* \mathcal{F} v ((More f ss C ts))
= (if (f, Suc (length ss + length ts)) \in \mathcal{F} then Fun f (map (term-to-sig \mathcal{F} v) ss @ inv-const-ctxt' \mathcal{F} v C # map (term-to-sig \mathcal{F} v) ts) else Var v)

2.1.4 Replace term at a given position in contexts

fun *replace-term-context-at* :: ('f, 'v) *ctxt* \Rightarrow *pos* \Rightarrow ('f, 'v) *term* \Rightarrow ('f, 'v) *ctxt*
 (-[- \leftarrow -]_C [1000, 0] 1000) **where**
replace-term-context-at \square *p u* = \square
 | *replace-term-context-at* (More *f ss C ts*) (*i # ps*) *u* =
 (if *i* < length *ss* then More *f* (*ss*[*i* := (*ss* ! *i*)[*ps* \leftarrow *u*]]) *C ts*
 else if *i* = length *ss* then More *f ss* (*replace-term-context-at* *C ps u*) *ts*
 else More *f ss C* (*ts*[(*i* - Suc (length *ss*)) := (*ts* ! (*i* - Suc (length *ss*)))] [*ps* \leftarrow *u*]))

abbreviation *constT c* \equiv *Fun c* []

2.1.5 Multihole context closure of a term relation as inductive set

definition *all-ctxt-closed* **where**

all-ctxt-closed *F r* \iff (\forall *f ts ss*. (*f*, length *ss*) \in *F* \implies length *ts* = length *ss* \longrightarrow
 (\forall *i*. *i* < length *ts* \longrightarrow (*ts* ! *i*, *ss* ! *i*) \in *r*) \longrightarrow
 (\forall *i*. *i* < length *ts* \longrightarrow funas-term (*ts* ! *i*) \cup funas-term (*ss* ! *i*) \subseteq *F*) \longrightarrow (*Fun*
f ts, *Fun f ss*) \in *r*) \wedge
 (\forall *x*. (*Var x*, *Var x*) \in *r*)

2.2 Destruction and introduction of *all-ctxt-closed*

lemma *all-ctxt-closedD*: *all-ctxt-closed* *F r* \implies (*f*, length *ss*) \in *F* \implies length *ts* =
 length *ss*

\implies [\wedge *i*. *i* < length *ts* \implies (*ts* ! *i*, *ss* ! *i*) \in *r*]
 \implies [\wedge *i*. *i* < length *ts* \implies funas-term (*ts* ! *i*) \subseteq *F*]
 \implies [\wedge *i*. *i* < length *ts* \implies funas-term (*ss* ! *i*) \subseteq *F*]
 \implies (*Fun f ts*, *Fun f ss*) \in *r*
 <proof>

lemma *trans-ctxt-sig-imp-all-ctxt-closed*: **assumes** *tran*: *trans r*

and *refl*: \wedge *t*. funas-term *t* \subseteq *F* \implies (*t*, *t*) \in *r*

and *ctxt*: \wedge *C s t*. funas-ctxt *C* \subseteq *F* \implies funas-term *s* \subseteq *F* \implies funas-term *t* \subseteq
F \implies (*s*, *t*) \in *r* \implies (*C* < *s*>, *C* < *t*>) \in *r*

shows *all-ctxt-closed* *F r* <proof>

2.3 Lemmas for *poss* and ordering of positions

lemma *subst-poss-mono*: *poss s* \subseteq *poss* (*s* \cdot σ)

<proof>

lemma *par-pos-prefix* [*simp*]:

(*i* # *p*) \perp (*i* # *q*) \implies *p* \perp *q*

<proof>

lemma *pos-diff-itself* [*simp*]: *p* $-_p$ *p* = []

<proof>

lemma *pos-les-eq-append-diff* [simp]:

$$p \leq_p q \implies p @ (q -_p p) = q$$

⟨proof⟩

lemma *pos-diff-append-itself* [simp]: $(p @ q) -_p p = q$

⟨proof⟩

lemma *poss-pos-diffI*:

$$p \leq_p q \implies q \in \text{poss } s \implies q -_p p \in \text{poss } (s \mid\!-\! p)$$

⟨proof⟩

lemma *less-eq-poss-append-itself* [simp]: $p \leq_p (p @ q)$

⟨proof⟩

lemma *poss-ctxt-apply* [simp]:

$$\text{hole-pos } C @ p \in \text{poss } C \langle s \rangle \longleftrightarrow p \in \text{poss } s$$

⟨proof⟩

lemma *pos-replace-at-pres*:

$$p \in \text{poss } s \implies p \in \text{poss } s[p \leftarrow t]$$

⟨proof⟩

lemma *par-pos-replace-pres*:

$$p \in \text{poss } s \implies p \perp q \implies p \in \text{poss } s[q \leftarrow t]$$

⟨proof⟩

lemma *poss-of-termE* [elim]:

assumes $p \in \text{poss-of-term } u \ s$
and $p \in \text{poss } s \implies s \mid\!-\! p = u \implies P$
shows P ⟨proof⟩

lemma *poss-of-term-Cons*:

$$i \# p \in \text{poss-of-term } u \ (Fun \ f \ ts) \implies p \in \text{poss-of-term } u \ (ts \ ! \ i)$$

⟨proof⟩

lemma *poss-of-term-const-ctxt-apply*:

assumes $p \in \text{poss-of-term } (constT \ c) \ C \langle s \rangle$
shows $p \perp (\text{hole-pos } C) \vee (\text{hole-pos } C) \leq_p p$ ⟨proof⟩

2.4 Lemmas for $(\mid\!-\!)$ and *replace-term-at*

lemma *subt-at-append-dist*:

$$p @ q \in \text{poss } s \implies s \mid\!-\! (p @ q) = (s \mid\!-\! p) \mid\!-\! q$$

⟨proof⟩

lemma *ctxt-apply-term-subt-at-hole-pos* [simp]:

$$C \langle s \rangle \mid\!-\! (\text{hole-pos } C @ q) = s \mid\!-\! q$$

⟨proof⟩

lemma *subst-subt-at-dist*:

$$p \in \text{poss } s \implies s \cdot \sigma \mid\!-\! p = s \mid\!-\! p \cdot \sigma$$

<proof>

lemma *replace-term-at-subt-at-id* [simp]: $s[p \leftarrow (s \mid\!-\! p)] = s$

<proof>

lemma *replace-term-at-same-pos* [simp]:

$$s[p \leftarrow u][p \leftarrow t] = s[p \leftarrow t]$$

<proof>

lemma *subt-at-vars-term*:

$$p \in \text{poss } s \implies s \mid\!-\! p = \text{Var } x \implies x \in \text{vars-term } s$$

<proof>

lemma *linear-term-varposs-subst-replace-term*:

$$\begin{aligned} & \text{linear-term } s \implies p \in \text{varposs } s \implies p \leq_p q \implies \\ & (s \cdot \sigma)[q \leftarrow u] = s \cdot (\lambda x. \text{if } \text{Var } x = s \mid\!-\! p \text{ then } (\sigma x)[q \!-\!_p p \leftarrow u] \text{ else } (\sigma x)) \end{aligned}$$

<proof>

lemma *par-hole-pos-replace-term-context-at*:

$$p \perp \text{hole-pos } C \implies C\langle s \rangle[p \leftarrow u] = (C[p \leftarrow u]_C)\langle s \rangle$$

<proof>

lemma *par-pos-replace-term-at*:

$$p \in \text{poss } s \implies p \perp q \implies s[q \leftarrow t] \mid\!-\! p = s \mid\!-\! p$$

<proof>

lemma *less-eq-subt-at-replace*:

$$p \in \text{poss } s \implies p \leq_p q \implies s[q \leftarrow t] \mid\!-\! p = (s \mid\!-\! p)[q \!-\!_p p \leftarrow t]$$

<proof>

lemma *greater-eq-subt-at-replace*:

$$p \in \text{poss } s \implies q \leq_p p \implies s[q \leftarrow t] \mid\!-\! p = t \mid\!-\! (p \!-\!_p q)$$

<proof>

lemma *replace-subterm-at-itself* [simp]:

$$s[p \leftarrow (s \mid\!-\! p)[q \leftarrow t]] = s[p \text{ @ } q \leftarrow t]$$

<proof>

lemma *hole-pos-replace-term-at* [simp]:

$$\text{hole-pos } C \leq_p p \implies C\langle s \rangle[p \leftarrow u] = C\langle s[p \!-\!_p \text{hole-pos } C \leftarrow u] \rangle$$

<proof>

lemma *ctxt-of-pos-term-apply-replace-at-ident*:

$$\text{assumes } p \in \text{poss } s$$

shows $(\text{ctxt-at-pos } s \ p)\langle t \rangle = s[p \leftarrow t]$
 $\langle \text{proof} \rangle$

lemma *ctxt-apply-term-replace-term-hole-pos* [simp]:
 $C\langle s \rangle[\text{hole-pos } C \ @ \ q \leftarrow u] = C\langle s[q \leftarrow u] \rangle$
 $\langle \text{proof} \rangle$

lemma *ctxt-apply-subst-at-hole-pos* [simp]: $C\langle s \rangle \mid\text{- hole-pos } C = s$
 $\langle \text{proof} \rangle$

lemma *subst-at-imp-supteq'*:
assumes $p \in \text{poss } s$ **and** $s \mid\text{-} p = t$ **shows** $s \supseteq t$ $\langle \text{proof} \rangle$

lemma *subst-at-imp-supteq*:
assumes $p \in \text{poss } s$ **shows** $s \supseteq s \mid\text{-} p$
 $\langle \text{proof} \rangle$

2.5 term-to-sig invariants and distributions

lemma *funas-term-term-to-sig* [simp]: *funas-term* (*term-to-sig* $\mathcal{F} \ v \ t$) $\subseteq \mathcal{F}$
 $\langle \text{proof} \rangle$

lemma *term-to-sig-id* [simp]:
funas-term $t \subseteq \mathcal{F} \implies \text{term-to-sig } \mathcal{F} \ v \ t = t$
 $\langle \text{proof} \rangle$

lemma *term-to-sig-subst-sig* [simp]:
funas-term $t \subseteq \mathcal{F} \implies \text{term-to-sig } \mathcal{F} \ v \ (t \cdot \sigma) = t \cdot (\lambda x. \text{term-to-sig } \mathcal{F} \ v \ (\sigma \ x))$
 $\langle \text{proof} \rangle$

lemma *funas-ctxt-ctxt-inv-const-ctxt-ind* [simp]:
funas-ctxt $C \subseteq \mathcal{F} \implies \text{inv-const-ctxt } \mathcal{F} \ v \ C = C$
 $\langle \text{proof} \rangle$

lemma *term-to-sig-ctxt-apply* [simp]:
 $\text{ctxt-well-def-hole-path } \mathcal{F} \ C \implies \text{term-to-sig } \mathcal{F} \ v \ C\langle s \rangle = (\text{inv-const-ctxt } \mathcal{F} \ v \ C)\langle \text{term-to-sig } \mathcal{F} \ v \ s \rangle$
 $\langle \text{proof} \rangle$

lemma *term-to-sig-ctxt-apply'* [simp]:
 $\neg \text{ctxt-well-def-hole-path } \mathcal{F} \ C \implies \text{term-to-sig } \mathcal{F} \ v \ C\langle s \rangle = \text{inv-const-ctxt}' \mathcal{F} \ v \ C$
 $\langle \text{proof} \rangle$

lemma *funas-ctxt-ctxt-well-def-hole-path*:
funas-ctxt $C \subseteq \mathcal{F} \implies \text{ctxt-well-def-hole-path } \mathcal{F} \ C$
 $\langle \text{proof} \rangle$

2.6 Misc

lemma *funas-term-subst-at*:

$(f, n) \in \text{funas-term } t \implies (\exists p \text{ ts. } p \in \text{poss } t \wedge t \mid\text{- } p = \text{Fun } f \text{ ts} \wedge \text{length } \text{ts} = n)$
 ⟨proof⟩

lemma *finite-poss*: *finite* (poss *s*)
 ⟨proof⟩

lemma *finite-varposs*: *finite* (varposs *s*)
 ⟨proof⟩

lemma *ground-linear* [*simp*]: *ground* *t* \implies *linear-term* *t*
 ⟨proof⟩

declare *ground-substI*[*intro*, *simp*]

lemma *ground-ctxt-substI*:
 $(\bigwedge x. x \in \text{vars-ctxt } C \implies \text{ground } (\sigma x)) \implies \text{ground-ctxt } (C \cdot_c \sigma)$
 ⟨proof⟩

lemma *funas-ctxt-subst-apply-ctxt*:
 $\text{funas-ctxt } (C \cdot_c \sigma) = \text{funas-ctxt } C \cup (\bigcup (\text{funas-term } \sigma \text{ vars-ctxt } C))$
 ⟨proof⟩

lemma *varposs-Var*[*simp*]:
 $\text{varposs } (\text{Var } x) = \{\}\{\}$
 ⟨proof⟩

lemma *varposs-Fun*[*simp*]:
 $\text{varposs } (\text{Fun } f \text{ ts}) = \{ i \# p \mid i p. i < \text{length } \text{ts} \wedge p \in \text{varposs } (\text{ts } ! i) \}$
 ⟨proof⟩

lemma *vars-term-varposs-iff*:
 $x \in \text{vars-term } s \iff (\exists p \in \text{varposs } s. s \mid\text{- } p = \text{Var } x)$
 ⟨proof⟩

lemma *vars-term-empty-ground*:
 $\text{vars-term } s = \{\} \implies \text{ground } s$
 ⟨proof⟩

lemma *ground-subst-apply*: *ground* *t* $\implies t \cdot \sigma = t$
 ⟨proof⟩

lemma *varposs-imp-poss*:
 $p \in \text{varposs } s \implies p \in \text{poss } s$ ⟨proof⟩

lemma *varposs-empty-ground*:
 $\text{varposs } s = \{\} \iff \text{ground } s$
 ⟨proof⟩

lemma *funas-term-subterm-atI* [*intro*]:

$p \in \text{poss } s \implies \text{funas-term } s \subseteq \mathcal{F} \implies \text{funas-term } (s \mid\!-\! p) \subseteq \mathcal{F}$
 ⟨proof⟩

lemma *varposs-ground-replace-at*:

$p \in \text{varposs } s \implies \text{ground } u \implies \text{varposs } s[p \leftarrow u] = \text{varposs } s - \{p\}$
 ⟨proof⟩

lemma *funas-term-replace-at-upper*:

$\text{funas-term } s[p \leftarrow t] \subseteq \text{funas-term } s \cup \text{funas-term } t$
 ⟨proof⟩

lemma *funas-term-replace-at-lower*:

$p \in \text{poss } s \implies \text{funas-term } t \subseteq \text{funas-term } (s[p \leftarrow t])$
 ⟨proof⟩

lemma *poss-of-term-possI* [intro!]:

$p \in \text{poss } s \implies s \mid\!-\! p = u \implies p \in \text{poss-of-term } u \text{ } s$
 ⟨proof⟩

lemma *poss-of-term-replace-term-at*:

$p \in \text{poss } s \implies p \in \text{poss-of-term } u \text{ } s[p \leftarrow u]$
 ⟨proof⟩

lemma *constT-nfunas-term-poss-of-term-empty*:

$(c, 0) \notin \text{funas-term } t \iff \text{poss-of-term } (\text{constT } c) \text{ } t = \{\}$
 ⟨proof⟩

lemma *poss-of-term-poss-emptyD*:

assumes $\text{poss-of-term } u \text{ } s = \{\}$
shows $p \in \text{poss } s \implies s \mid\!-\! p \neq u$ ⟨proof⟩

lemma *possc-subt-at-ctxt-apply*:

$p \in \text{possc } C \implies p \perp \text{hole-pos } C \implies C\langle s \rangle \mid\!-\! p = C\langle t \rangle \mid\!-\! p$
 ⟨proof⟩

end

3 Rewriting

theory *Rewriting*

imports *Terms-Positions*

begin

3.1 Basic rewrite definitions

3.1.1 Rewrite steps with implicit signature declaration (encoded in the type)

inductive-set *rrstep* :: $(\iota, \iota) \text{ term rel} \Rightarrow (\iota, \iota) \text{ term rel}$ for \mathcal{R} where

[intro]: $(l, r) \in \mathcal{R} \implies (l \cdot \sigma, r \cdot \sigma) \in rstep \mathcal{R}$

inductive-set $rstep :: ('f, 'v) \text{ term rel} \Rightarrow ('f, 'v) \text{ term rel}$ for \mathcal{R} where
 $(s, t) \in rstep \mathcal{R} \implies (C\langle s \rangle, C\langle t \rangle) \in rstep \mathcal{R}$

3.1.2 Restrict relations to terms induced by a given signature

definition $sig\text{-step } \mathcal{F} \mathcal{R} = Restr \mathcal{R} (Collect (\lambda s. \text{funas-term } s \subseteq \mathcal{F}))$

3.1.3 Rewriting under a given signature/restricted to ground terms

abbreviation $srrstep \mathcal{F} \mathcal{R} \equiv sig\text{-step } \mathcal{F} (rstep \mathcal{R})$

abbreviation $srstep \mathcal{F} \mathcal{R} \equiv sig\text{-step } \mathcal{F} (rstep \mathcal{R})$

abbreviation $gsrstep \mathcal{F} \mathcal{R} \equiv Restr (sig\text{-step } \mathcal{F} (rstep \mathcal{R})) (Collect \text{ground})$

3.1.4 Rewriting sequences involving a root step

abbreviation (input) $relto :: 'a \text{ rel} \Rightarrow 'a \text{ rel} \Rightarrow 'a \text{ rel}$ where
 $relto R S \equiv S^* \circ R \circ S^*$

definition $srsteps\text{-with-root-step } \mathcal{F} \mathcal{R} \equiv relto (sig\text{-step } \mathcal{F} (rstep \mathcal{R})) (srstep \mathcal{F} \mathcal{R})$

3.2 Monotonicity laws

lemma $Restr\text{-mono}: Restr r A \subseteq r \langle \text{proof} \rangle$

lemma $Restr\text{-trancl-mono-set}: (Restr r A)^+ \subseteq A \times A$
 $\langle \text{proof} \rangle$

lemma $rstep\text{-rstep-mono}: rstep \mathcal{R} \subseteq rstep \mathcal{R}$
 $\langle \text{proof} \rangle$

lemma $sig\text{-step-mono}$:
 $\mathcal{F} \subseteq \mathcal{G} \implies sig\text{-step } \mathcal{F} \mathcal{R} \subseteq sig\text{-step } \mathcal{G} \mathcal{R}$
 $\langle \text{proof} \rangle$

lemma $sig\text{-step-mono2}$:
 $\mathcal{R} \subseteq \mathcal{L} \implies sig\text{-step } \mathcal{F} \mathcal{R} \subseteq sig\text{-step } \mathcal{F} \mathcal{L}$
 $\langle \text{proof} \rangle$

lemma $srrstep\text{-monp}$:
 $\mathcal{F} \subseteq \mathcal{G} \implies srrstep \mathcal{F} \mathcal{R} \subseteq srrstep \mathcal{G} \mathcal{R}$
 $\langle \text{proof} \rangle$

lemma $srstep\text{-monp}$:
 $\mathcal{F} \subseteq \mathcal{G} \implies srstep \mathcal{F} \mathcal{R} \subseteq srstep \mathcal{G} \mathcal{R}$
 $\langle \text{proof} \rangle$

lemma $srsteps\text{-monp}$:

$\mathcal{F} \subseteq \mathcal{G} \implies (\text{srstep } \mathcal{F} \mathcal{R})^+ \subseteq (\text{srstep } \mathcal{G} \mathcal{R})^+$
 ⟨proof⟩

lemma *srsteps-eq-monp*:

$\mathcal{F} \subseteq \mathcal{G} \implies (\text{srstep } \mathcal{F} \mathcal{R})^* \subseteq (\text{srstep } \mathcal{G} \mathcal{R})^*$
 ⟨proof⟩

lemma *srsteps-with-root-step-sig-mono*:

$\mathcal{F} \subseteq \mathcal{G} \implies \text{srsteps-with-root-step } \mathcal{F} \mathcal{R} \subseteq \text{srsteps-with-root-step } \mathcal{G} \mathcal{R}$
 ⟨proof⟩

3.3 Introduction, elimination, and destruction rules for *sig-step*, *rstep*, *rrstep*, *srrstep*, and *srstep*

lemma *sig-stepE* [*elim, consumes 1*]:

$(s, t) \in \text{sig-step } \mathcal{F} \mathcal{R} \implies [(s, t) \in \mathcal{R} \implies \text{funas-term } s \subseteq \mathcal{F} \implies \text{funas-term } t \subseteq \mathcal{F} \implies P] \implies P$
 ⟨proof⟩

lemma *sig-stepI* [*intro*]:

$\text{funas-term } s \subseteq \mathcal{F} \implies \text{funas-term } t \subseteq \mathcal{F} \implies (s, t) \in \mathcal{R} \implies (s, t) \in \text{sig-step } \mathcal{F} \mathcal{R}$
 ⟨proof⟩

lemma *rrstep-subst* [*elim, consumes 1*]:

assumes $(s, t) \in \text{rrstep } \mathcal{R}$
obtains $l r \sigma$ **where** $(l, r) \in \mathcal{R} \ s = l \cdot \sigma \ t = r \cdot \sigma$ ⟨proof⟩

lemma *rstep-imp-C-s-r*:

assumes $(s, t) \in \text{rstep } \mathcal{R}$
shows $\exists C \sigma l r. (l, r) \in \mathcal{R} \wedge s = C \langle l \cdot \sigma \rangle \wedge t = C \langle r \cdot \sigma \rangle$ ⟨proof⟩

lemma *rstep-imp-C-s-r'* [*elim, consumes 1*]:

assumes $(s, t) \in \text{rstep } \mathcal{R}$
obtains $C l r \sigma$ **where** $(l, r) \in \mathcal{R} \ s = C \langle l \cdot \sigma \rangle \ t = C \langle r \cdot \sigma \rangle$ ⟨proof⟩

lemma *rrstep-basicI* [*intro*]:

$(l, r) \in \mathcal{R} \implies (l, r) \in \text{rrstep } \mathcal{R}$
 ⟨proof⟩

lemma *rstep-ruleI* [*intro*]:

$(l, r) \in \mathcal{R} \implies (l, r) \in \text{rstep } \mathcal{R}$
 ⟨proof⟩

lemma *rstepI* [*intro*]:

$(l, r) \in \mathcal{R} \implies s = C \langle l \cdot \sigma \rangle \implies t = C \langle r \cdot \sigma \rangle \implies (s, t) \in \text{rstep } \mathcal{R}$
 ⟨proof⟩

lemma *rstep-substI* [intro]:

$$(s, t) \in \text{rstep } \mathcal{R} \implies (s \cdot \sigma, t \cdot \sigma) \in \text{rstep } \mathcal{R}$$

<proof>

lemma *rstep-ctxtI* [intro]:

$$(s, t) \in \text{rstep } \mathcal{R} \implies (C\langle s \rangle, C\langle t \rangle) \in \text{rstep } \mathcal{R}$$

<proof>

lemma *srrstepD*:

$$(s, t) \in \text{srrstep } \mathcal{F} \mathcal{R} \implies (s, t) \in \text{rrstep } \mathcal{R} \wedge \text{funas-term } s \subseteq \mathcal{F} \wedge \text{funas-term } t \subseteq \mathcal{F}$$

<proof>

lemma *srstepD*:

$$(s, t) \in (\text{srstep } \mathcal{F} \mathcal{R}) \implies (s, t) \in \text{rstep } \mathcal{R} \wedge \text{funas-term } s \subseteq \mathcal{F} \wedge \text{funas-term } t \subseteq \mathcal{F}$$

<proof>

lemma *srstepsD*:

$$(s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^+ \implies (s, t) \in (\text{rstep } \mathcal{R})^+ \wedge \text{funas-term } s \subseteq \mathcal{F} \wedge \text{funas-term } t \subseteq \mathcal{F}$$

<proof>

3.3.1 Transitive and reflexive closure distribution over *sig-step*

lemma *funas-rel-converse*:

$$\text{funas-rel } \mathcal{R} \subseteq \mathcal{F} \implies \text{funas-rel } (\mathcal{R}^{-1}) \subseteq \mathcal{F} \text{ } \langle \text{proof} \rangle$$

lemma *rstep-term-to-sig-r*:

assumes $(s, t) \in \text{rstep } \mathcal{R}$ **and** $\text{funas-rel } \mathcal{R} \subseteq \mathcal{F}$ **and** $\text{funas-term } s \subseteq \mathcal{F}$
shows $(s, \text{term-to-sig } \mathcal{F} v t) \in \text{rstep } \mathcal{R}$

<proof>

lemma *rstep-term-to-sig-l*:

assumes $(s, t) \in \text{rstep } \mathcal{R}$ **and** $\text{funas-rel } \mathcal{R} \subseteq \mathcal{F}$ **and** $\text{funas-term } t \subseteq \mathcal{F}$
shows $(\text{term-to-sig } \mathcal{F} v s, t) \in \text{rstep } \mathcal{R}$

<proof>

lemma *rstep-trancl-sig-step-r*:

assumes $(s, t) \in (\text{rstep } \mathcal{R})^+$ **and** $\text{funas-rel } \mathcal{R} \subseteq \mathcal{F}$ **and** $\text{funas-term } s \subseteq \mathcal{F}$
shows $(s, \text{term-to-sig } \mathcal{F} v t) \in (\text{srstep } \mathcal{F} \mathcal{R})^+$ *<proof>*

lemma *rstep-trancl-sig-step-l*:

assumes $(s, t) \in (\text{rstep } \mathcal{R})^+$ **and** $\text{funas-rel } \mathcal{R} \subseteq \mathcal{F}$ **and** $\text{funas-term } t \subseteq \mathcal{F}$
shows $(\text{term-to-sig } \mathcal{F} v s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^+$ *<proof>*

lemma *rstep-srstepI* [intro]:

$$\text{funas-rel } \mathcal{R} \subseteq \mathcal{F} \implies \text{funas-term } s \subseteq \mathcal{F} \implies \text{funas-term } t \subseteq \mathcal{F} \implies (s, t) \in \text{rstep } \mathcal{R} \implies (s, t) \in \text{srstep } \mathcal{F} \mathcal{R}$$

<proof>

lemma *rsteps-srstepsI* [*intro*]:

funas-rel $\mathcal{R} \subseteq \mathcal{F} \implies \text{funas-term } s \subseteq \mathcal{F} \implies \text{funas-term } t \subseteq \mathcal{F} \implies (s, t) \in (\text{rstep } \mathcal{R})^+ \implies (s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^+$

<proof>

lemma *rsteps-eq-srsteps-eqI* [*intro*]:

funas-rel $\mathcal{R} \subseteq \mathcal{F} \implies \text{funas-term } s \subseteq \mathcal{F} \implies \text{funas-term } t \subseteq \mathcal{F} \implies (s, t) \in (\text{rstep } \mathcal{R})^* \implies (s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^*$

<proof>

lemma *rsteps-eq-relcomp-srsteps-eq-relcompI* [*intro*]:

assumes *funas-rel* $\mathcal{R} \subseteq \mathcal{F}$ *funas-rel* $\mathcal{S} \subseteq \mathcal{F}$

and *funas*: *funas-term* $s \subseteq \mathcal{F}$ *funas-term* $t \subseteq \mathcal{F}$

and *steps*: $(s, t) \in (\text{rstep } \mathcal{R})^* \ O (\text{rstep } \mathcal{S})^*$

shows $(s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^* \ O (\text{srstep } \mathcal{F} \mathcal{S})^*$

<proof>

3.3.2 Distributivity laws

lemma *rstep-smycl-dist*:

$(\text{rstep } \mathcal{R})^{\leftrightarrow} = \text{rstep } (\mathcal{R}^{\leftrightarrow})$

<proof>

lemma *sig-step-symcl-dist*:

$(\text{sig-step } \mathcal{F} \mathcal{R})^{\leftrightarrow} = \text{sig-step } \mathcal{F} (\mathcal{R}^{\leftrightarrow})$

<proof>

lemma *srstep-symcl-dist*:

$(\text{srstep } \mathcal{F} \mathcal{R})^{\leftrightarrow} = \text{srstep } \mathcal{F} (\mathcal{R}^{\leftrightarrow})$

<proof>

lemma *Restr-smycl-dist*:

$(\text{Restr } \mathcal{R} \mathcal{A})^{\leftrightarrow} = \text{Restr } (\mathcal{R}^{\leftrightarrow}) \mathcal{A}$

<proof>

lemmas *rew-symcl-inwards* = *rstep-smycl-dist* *sig-step-symcl-dist* *srstep-symcl-dist* *Restr-smycl-dist*

lemmas *rew-symcl-outwards* = *rew-symcl-inwards*[*symmetric*]

lemma *rstep-converse-dist*:

$(\text{rstep } \mathcal{R})^{-1} = \text{rstep } (\mathcal{R}^{-1})$

<proof>

lemma *srrstep-converse-dist*:

$(\text{srrstep } \mathcal{F} \mathcal{R})^{-1} = \text{srrstep } \mathcal{F} (\mathcal{R}^{-1})$

<proof>

lemma *sig-step-converse-rstep*:
 $(srstep \mathcal{F} \mathcal{R})^{-1} = sig\text{-step } \mathcal{F} ((rstep \mathcal{R})^{-1})$
 ⟨proof⟩

lemma *srstep-converse-dist*:
 $(srstep \mathcal{F} \mathcal{R})^{-1} = srstep \mathcal{F} (\mathcal{R}^{-1})$
 ⟨proof⟩

lemma *Restr-converse*: $(Restr \mathcal{R} A)^{-1} = Restr (\mathcal{R}^{-1}) A$
 ⟨proof⟩

lemmas *rew-converse-inwards* = *rstep-converse-dist* *srrstep-converse-dist* *sig-step-converse-rstep*
srstep-converse-dist *Restr-converse* *tranc1-converse*[*symmetric*] *rtranc1-converse*[*symmetric*]

lemmas *rew-converse-outwards* = *rew-converse-inwards*[*symmetric*]

lemma *sig-step-rsteps-dist*:
 $funas\text{-rel } \mathcal{R} \subseteq \mathcal{F} \implies sig\text{-step } \mathcal{F} ((rstep \mathcal{R})^+) = (srstep \mathcal{F} \mathcal{R})^+$
 ⟨proof⟩

lemma *sig-step-rsteps-eq-dist*:
 $funas\text{-rel } \mathcal{R} \subseteq \mathcal{F} \implies sig\text{-step } \mathcal{F} ((rstep \mathcal{R})^+) \cup Id = (srstep \mathcal{F} \mathcal{R})^*$
 ⟨proof⟩

lemma *sig-step-conversion-dist*:
 $(srstep \mathcal{F} \mathcal{R})^{\leftrightarrow*} = (srstep \mathcal{F} (\mathcal{R}^{\leftrightarrow}))^*$
 ⟨proof⟩

lemma *gsrstep-conversion-dist*:
 $(gsrstep \mathcal{F} \mathcal{R})^{\leftrightarrow*} = (gsrstep \mathcal{F} (\mathcal{R}^{\leftrightarrow}))^*$
 ⟨proof⟩

lemma *sig-step-grstep-dist*:
 $gsrstep \mathcal{F} \mathcal{R} = sig\text{-step } \mathcal{F} (Restr (rstep \mathcal{R}) (Collect\ ground))$
 ⟨proof⟩

3.4 Substitution closure of *srstep*

lemma *srstep-subst-closed*:
 assumes $(s, t) \in srstep \mathcal{F} \mathcal{R} \wedge x. funas\text{-term } (\sigma x) \subseteq \mathcal{F}$
 shows $(s \cdot \sigma, t \cdot \sigma) \in srstep \mathcal{F} \mathcal{R}$ ⟨proof⟩

lemma *srsteps-subst-closed*:
 assumes $(s, t) \in (srstep \mathcal{F} \mathcal{R})^+ \wedge x. funas\text{-term } (\sigma x) \subseteq \mathcal{F}$
 shows $(s \cdot \sigma, t \cdot \sigma) \in (srstep \mathcal{F} \mathcal{R})^+$ ⟨proof⟩

lemma *srsteps-eq-subst-closed*:
 assumes $(s, t) \in (srstep \mathcal{F} \mathcal{R})^* \wedge x. funas\text{-term } (\sigma x) \subseteq \mathcal{F}$
 shows $(s \cdot \sigma, t \cdot \sigma) \in (srstep \mathcal{F} \mathcal{R})^*$ ⟨proof⟩

lemma *srsteps-eq-subst-relcomp-closed*:

assumes $(s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^* O (\text{srstep } \mathcal{F} \mathcal{S})^* \wedge x. \text{funas-term } (\sigma x) \subseteq \mathcal{F}$
shows $(s \cdot \sigma, t \cdot \sigma) \in (\text{srstep } \mathcal{F} \mathcal{R})^* O (\text{srstep } \mathcal{F} \mathcal{S})^*$
 $\langle \text{proof} \rangle$

3.5 Context closure of *srstep*

lemma *srstep-ctxt-closed*:

assumes *funas-ctxt* $C \subseteq \mathcal{F}$ **and** $(s, t) \in \text{srstep } \mathcal{F} \mathcal{R}$
shows $(C\langle s \rangle, C\langle t \rangle) \in \text{srstep } \mathcal{F} \mathcal{R}$ $\langle \text{proof} \rangle$

lemma *srsteps-ctxt-closed*:

assumes *funas-ctxt* $C \subseteq \mathcal{F}$ **and** $(s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^+$
shows $(C\langle s \rangle, C\langle t \rangle) \in (\text{srstep } \mathcal{F} \mathcal{R})^+$ $\langle \text{proof} \rangle$

lemma *srsteps-eq-ctxt-closed*:

assumes *funas-ctxt* $C \subseteq \mathcal{F}$ **and** $(s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^*$
shows $(C\langle s \rangle, C\langle t \rangle) \in (\text{srstep } \mathcal{F} \mathcal{R})^*$ $\langle \text{proof} \rangle$

lemma *sig-steps-join-ctxt-closed*:

assumes *funas-ctxt* $C \subseteq \mathcal{F}$ **and** $(s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^\downarrow$
shows $(C\langle s \rangle, C\langle t \rangle) \in (\text{srstep } \mathcal{F} \mathcal{R})^\downarrow$ $\langle \text{proof} \rangle$

The following lemma shows that every rewrite sequence either contains a root step or is root stable

lemma *nsrsteps-with-root-step-step-on-args*:

assumes $(s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^+ (s, t) \notin \text{srsteps-with-root-step } \mathcal{F} \mathcal{R}$
shows $\exists f \text{ ss ts. } s = \text{Fun } f \text{ ss} \wedge t = \text{Fun } f \text{ ts} \wedge \text{length } \text{ss} = \text{length } \text{ts} \wedge$
 $(\forall i < \text{length } \text{ts. } (\text{ss} ! i, \text{ts} ! i) \in (\text{srstep } \mathcal{F} \mathcal{R})^*)$ $\langle \text{proof} \rangle$

lemma *rstep-to-pos-replace*:

assumes $(s, t) \in \text{rstep } \mathcal{R}$
shows $\exists p \text{ l } r \sigma. p \in \text{poss } s \wedge (l, r) \in \mathcal{R} \wedge s \mid\!-\! p = l \cdot \sigma \wedge t = s[p \leftarrow r \cdot \sigma]$
 $\langle \text{proof} \rangle$

lemma *pos-replace-to-rstep*:

assumes $p \in \text{poss } s (l, r) \in \mathcal{R}$
and $s \mid\!-\! p = l \cdot \sigma t = s[p \leftarrow r \cdot \sigma]$
shows $(s, t) \in \text{rstep } \mathcal{R}$
 $\langle \text{proof} \rangle$

end

theory *Replace-Constant*

imports *Rewriting*

begin

3.6 Removing/Replacing constants in a rewrite sequence that do not appear in the rewrite system

lemma *funas-term-const-subst-conv*:

$(c, 0) \notin \text{funas-term } l \iff \neg (l \supseteq \text{constT } c)$
 $\langle \text{proof} \rangle$

lemma *fresh-const-single-step-replace*:

assumes *lin*: linear-sys \mathcal{R} **and** *fresh*: $(c, 0) \notin \text{funas-rel } \mathcal{R}$
and *occ*: $p \in \text{poss-of-term } (\text{constT } c) s$ **and** *step*: $(s, t) \in \text{rstep } \mathcal{R}$
shows $(s[p \leftarrow u], t) \in \text{rstep } \mathcal{R} \vee$
 $(\exists q. q \in \text{poss-of-term } (\text{constT } c) t \wedge (s[p \leftarrow u], t[q \leftarrow u]) \in \text{rstep } \mathcal{R})$
 $\langle \text{proof} \rangle$

lemma *fresh-const-steps-replace*:

assumes *lin*: linear-sys \mathcal{R} **and** *fresh*: $(c, 0) \notin \text{funas-rel } \mathcal{R}$
and *occ*: $p \in \text{poss-of-term } (\text{constT } c) s$ **and** *steps*: $(s, t) \in (\text{rstep } \mathcal{R})^+$
shows $(s[p \leftarrow u], t) \in (\text{rstep } \mathcal{R})^+ \vee$
 $(\exists q. q \in \text{poss-of-term } (\text{constT } c) t \wedge (s[p \leftarrow u], t[q \leftarrow u]) \in (\text{rstep } \mathcal{R})^+)$
 $\langle \text{proof} \rangle$

lemma *remove-const-lhs-steps*:

assumes *lin*: linear-sys \mathcal{R} **and** *fresh*: $(c, 0) \notin \text{funas-rel } \mathcal{R}$
and *const*: $(c, 0) \notin \text{funas-term } t$
and *pos*: $p \in \text{poss-of-term } (\text{constT } c) s$
and *steps*: $(s, t) \in (\text{rstep } \mathcal{R})^+$
shows $(s[p \leftarrow u], t) \in (\text{rstep } \mathcal{R})^+ \langle \text{proof} \rangle$

Now we can show that we may remove a constant substitution

definition *const-replace-closed where*

const-replace-closed $c U = (\forall s t u p.$
 $p \in \text{poss-of-term } (\text{constT } c) s \implies (s, t) \in U \implies$
 $(\exists q. q \in \text{poss-of-term } (\text{constT } c) t \wedge (s[p \leftarrow u], t[q \leftarrow u]) \in U) \vee (s[p \leftarrow u],$
 $t) \in U)$

lemma *const-replace-closedD*:

assumes *const-replace-closed* $c U$ $p \in \text{poss-of-term } (\text{constT } c) s$ $(s, t) \in U$
shows $(s[p \leftarrow u], t) \in U \vee (\exists q. q \in \text{poss-of-term } (\text{constT } c) t \wedge (s[p \leftarrow u], t[q \leftarrow u]) \in U) \langle \text{proof} \rangle$

lemma *const-replace-closedI*:

assumes $\bigwedge s t u p. p \in \text{poss-of-term } (\text{constT } c) s \implies (s, t) \in U \implies$
 $(\exists q. q \in \text{poss-of-term } (\text{constT } c) t \wedge (s[p \leftarrow u], t[q \leftarrow u]) \in U) \vee (s[p \leftarrow u],$
 $t) \in U$
shows *const-replace-closed* $c U \langle \text{proof} \rangle$

abbreviation *const-subst* :: $'f \Rightarrow 'v \Rightarrow ('f, 'v) \text{Term.term where}$

const-subst $c \equiv (\lambda x. \text{Fun } c \ [])$

lemma *lin-fresh-rstep-const-replace-closed*:

linear-sys $\mathcal{R} \implies (c, 0) \notin \text{funas-rel } \mathcal{R} \implies \text{const-replace-closed } c \text{ (rstep } \mathcal{R})$
 ⟨proof⟩

lemma *const-replace-closed-symcl*:
 $\text{const-replace-closed } c \ U \implies \text{const-replace-closed } c \ (U^=)$
 ⟨proof⟩

lemma *const-replace-closed-trancl*:
 $\text{const-replace-closed } c \ U \implies \text{const-replace-closed } c \ (U^+)$
 ⟨proof⟩

lemma *const-replace-closed-rtrancl*:
 $\text{const-replace-closed } c \ U \implies \text{const-replace-closed } c \ (U^*)$
 ⟨proof⟩

lemma *const-replace-closed-relcomp*:
 $\text{const-replace-closed } c \ U \implies \text{const-replace-closed } c \ V \implies \text{const-replace-closed } c \ (U \ O \ V)$
 ⟨proof⟩

const-replace-closed allow the removal of a fresh constant substitution

lemma *const-replace-closed-remove-subst-lhs*:
assumes *repl*: $\text{const-replace-closed } c \ U$
and *const*: $(c, 0) \notin \text{funas-term } t$
and *steps*: $(s \cdot \text{const-subst } c, t) \in U$
shows $(s, t) \in U$ ⟨proof⟩

3.6.1 Removal lemma applied to various rewrite relations

lemma *remove-const-subst-step-lhs*:
assumes *lin*: *linear-sys* \mathcal{R} **and** *fresh*: $(c, 0) \notin \text{funas-rel } \mathcal{R}$
and *const*: $(c, 0) \notin \text{funas-term } t$
and *step*: $(s \cdot \text{const-subst } c, t) \in (\text{rstep } \mathcal{R})$
shows $(s, t) \in (\text{rstep } \mathcal{R})$
 ⟨proof⟩

lemma *remove-const-subst-steps-lhs*:
assumes *lin*: *linear-sys* \mathcal{R} **and** *fresh*: $(c, 0) \notin \text{funas-rel } \mathcal{R}$
and *const*: $(c, 0) \notin \text{funas-term } t$
and *steps*: $(s \cdot \text{const-subst } c, t) \in (\text{rstep } \mathcal{R})^+$
shows $(s, t) \in (\text{rstep } \mathcal{R})^+$
 ⟨proof⟩

lemma *remove-const-subst-steps-eq-lhs*:
assumes *lin*: *linear-sys* \mathcal{R} **and** *fresh*: $(c, 0) \notin \text{funas-rel } \mathcal{R}$
and *const*: $(c, 0) \notin \text{funas-term } t$
and *steps*: $(s \cdot \text{const-subst } c, t) \in (\text{rstep } \mathcal{R})^*$
shows $(s, t) \in (\text{rstep } \mathcal{R})^*$ ⟨proof⟩

lemma *remove-const-subst-steps-rhs*:

assumes *lin*: linear-sys \mathcal{R} **and** *fresh*: $(c, 0) \notin \text{funas-rel } \mathcal{R}$
and *const*: $(c, 0) \notin \text{funas-term } s$
and *steps*: $(s, t \cdot \text{const-subst } c) \in (\text{rstep } \mathcal{R})^+$
shows $(s, t) \in (\text{rstep } \mathcal{R})^+$
 $\langle \text{proof} \rangle$

lemma *remove-const-subst-steps-eq-rhs*:
assumes *lin*: linear-sys \mathcal{R} **and** *fresh*: $(c, 0) \notin \text{funas-rel } \mathcal{R}$
and *const*: $(c, 0) \notin \text{funas-term } s$
and *steps*: $(s, t \cdot \text{const-subst } c) \in (\text{rstep } \mathcal{R})^*$
shows $(s, t) \in (\text{rstep } \mathcal{R})^*$
 $\langle \text{proof} \rangle$

Main lemmas

lemma *const-subst-eq-ground-eq*:
assumes $s \cdot \text{const-subst } c = t \cdot \text{const-subst } d$ $c \neq d$
and $(c, 0) \notin \text{funas-term } t$ $(d, 0) \notin \text{funas-term } s$
shows $s = t$ $\langle \text{proof} \rangle$

lemma *remove-const-subst-steps*:
assumes linear-sys \mathcal{R} **and** $(c, 0) \notin \text{funas-rel } \mathcal{R}$ **and** $(d, 0) \notin \text{funas-rel } \mathcal{R}$
and $c \neq d$ $(c, 0) \notin \text{funas-term } t$ $(d, 0) \notin \text{funas-term } s$
and $(s \cdot \text{const-subst } c, t \cdot \text{const-subst } d) \in (\text{rstep } \mathcal{R})^*$
shows $(s, t) \in (\text{rstep } \mathcal{R})^*$
 $\langle \text{proof} \rangle$

lemma *remove-const-subst-relcomp-lhs*:
assumes *sys*: linear-sys \mathcal{R} linear-sys \mathcal{S}
and *fr*: $(c, 0) \notin \text{funas-rel } \mathcal{R}$ **and** *fs*: $(c, 0) \notin \text{funas-rel } \mathcal{S}$
and *funas*: $(c, 0) \notin \text{funas-term } t$
and *seq*: $(s \cdot \text{const-subst } c, t) \in (\text{rstep } \mathcal{R})^* \ O (\text{rstep } \mathcal{S})^*$
shows $(s, t) \in (\text{rstep } \mathcal{R})^* \ O (\text{rstep } \mathcal{S})^*$ $\langle \text{proof} \rangle$

lemma *remove-const-subst-relcomp-rhs*:
assumes *sys*: linear-sys \mathcal{R} linear-sys \mathcal{S}
and *fr*: $(c, 0) \notin \text{funas-rel } \mathcal{R}$ **and** *fs*: $(c, 0) \notin \text{funas-rel } \mathcal{S}$
and *funas*: $(c, 0) \notin \text{funas-term } s$
and *seq*: $(s, t \cdot \text{const-subst } c) \in (\text{rstep } \mathcal{R})^* \ O (\text{rstep } \mathcal{S})^*$
shows $(s, t) \in (\text{rstep } \mathcal{R})^* \ O (\text{rstep } \mathcal{S})^*$
 $\langle \text{proof} \rangle$

lemma *remove-const-subst-relcomp*:
assumes *sys*: linear-sys \mathcal{R} linear-sys \mathcal{S}
and *fr*: $(c, 0) \notin \text{funas-rel } \mathcal{R}$ $(d, 0) \notin \text{funas-rel } \mathcal{R}$
and *fs*: $(c, 0) \notin \text{funas-rel } \mathcal{S}$ $(d, 0) \notin \text{funas-rel } \mathcal{S}$
and *diff*: $c \neq d$ **and** *funas*: $(c, 0) \notin \text{funas-term } t$ $(d, 0) \notin \text{funas-term } s$
and *seq*: $(s \cdot \text{const-subst } c, t \cdot \text{const-subst } d) \in (\text{rstep } \mathcal{R})^* \ O (\text{rstep } \mathcal{S})^*$

shows $(s, t) \in (rstep \mathcal{R})^* O (rstep \mathcal{S})^*$
 ⟨proof⟩

end

4 Confluence related rewriting properties

theory *Rewriting-Properties*

imports *Rewriting*

Abstract-Rewriting.Abstract-Rewriting

begin

4.1 Confluence related ARS properties

definition *SCR-on* $r A \equiv (\forall a \in A. \forall b c. (a, b) \in r \wedge (a, c) \in r \longrightarrow (\exists d. (b, d) \in r^= \wedge (c, d) \in r^*))$

abbreviation *SCR* $:: 'a \text{ rel} \Rightarrow \text{bool}$ **where** *SCR* $r \equiv \text{SCR-on } r \text{ UNIV}$

definition *NFP-on* $:: 'a \text{ rel} \Rightarrow 'a \text{ set} \Rightarrow \text{bool}$ **where**

NFP-on $r A \longleftrightarrow (\forall a \in A. \forall b c. (a, b) \in r^* \wedge (a, c) \in r^! \longrightarrow (b, c) \in r^*)$

abbreviation *NFP* $:: 'a \text{ rel} \Rightarrow \text{bool}$ **where** *NFP* $r \equiv \text{NFP-on } r \text{ UNIV}$

definition *CE-on* $:: 'a \text{ rel} \Rightarrow 'a \text{ rel} \Rightarrow 'a \text{ set} \Rightarrow \text{bool}$ **where**

CE-on $r s A \longleftrightarrow (\forall a \in A. \forall b. (a, b) \in r^{\leftrightarrow*} \longleftrightarrow (a, b) \in s^{\leftrightarrow*})$

abbreviation *CE* $:: 'a \text{ rel} \Rightarrow 'a \text{ rel} \Rightarrow \text{bool}$ **where** *CE* $r s \equiv \text{CE-on } r s \text{ UNIV}$

definition *NE-on* $:: 'a \text{ rel} \Rightarrow 'a \text{ rel} \Rightarrow 'a \text{ set} \Rightarrow \text{bool}$ **where**

NE-on $r s A \longleftrightarrow (\forall a \in A. \forall b. (a, b) \in r^! \longleftrightarrow (a, b) \in s^!)$

abbreviation *NE* $:: 'a \text{ rel} \Rightarrow 'a \text{ rel} \Rightarrow \text{bool}$ **where** *NE* $r s \equiv \text{NE-on } r s \text{ UNIV}$

4.2 Signature closure of relation to model multihole context closure

lemma *all-ctxt-closed-sig-rsteps* [intro]:

fixes $\mathcal{R} :: ('f, 'v) \text{ term rel}$

shows *all-ctxt-closed* $\mathcal{F} ((srstep \mathcal{F} \mathcal{R})^*)$ (**is** *all-ctxt-closed* - ($?R^*$))

⟨proof⟩

lemma *sigstep-trancl-funass*:

$(s, t) \in (srstep \mathcal{F} \mathcal{S})^* \Longrightarrow s \neq t \Longrightarrow \text{funas-term } s \subseteq \mathcal{F}$

$(s, t) \in (srstep \mathcal{F} \mathcal{S})^* \Longrightarrow s \neq t \Longrightarrow \text{funas-term } t \subseteq \mathcal{F}$

⟨proof⟩

lemma *srrstep-to-srstep*:

$(s, t) \in \text{srrstep } \mathcal{F} \mathcal{R} \Longrightarrow (s, t) \in \text{srstep } \mathcal{F} \mathcal{R}$

$\langle \text{proof} \rangle$

lemma *srsteps-with-root-step-srstepsD*:

$(s, t) \in \text{srsteps-with-root-step } \mathcal{F} \mathcal{R} \implies (s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^+$
 $\langle \text{proof} \rangle$

lemma *srsteps-with-root-step-srsteps-eqD*:

$(s, t) \in \text{srsteps-with-root-step } \mathcal{F} \mathcal{R} \implies (s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^*$
 $\langle \text{proof} \rangle$

lemma *symcl-srstep-conversion*:

$(s, t) \in \text{srstep } \mathcal{F} (\mathcal{R}^{\leftrightarrow}) \implies (s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^{\leftrightarrow*}$
 $\langle \text{proof} \rangle$

lemma *symcl-srsteps-conversion*:

$(s, t) \in (\text{srstep } \mathcal{F} (\mathcal{R}^{\leftrightarrow}))^* \implies (s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^{\leftrightarrow*}$
 $\langle \text{proof} \rangle$

lemma *NF-srstep-args*:

assumes $\text{Fun } f \text{ } ss \in \text{NF } (\text{srstep } \mathcal{F} \mathcal{R}) \text{ funas-term } (\text{Fun } f \text{ } ss) \subseteq \mathcal{F} \text{ } i < \text{length } ss$
shows $ss ! i \in \text{NF } (\text{srstep } \mathcal{F} \mathcal{R})$
 $\langle \text{proof} \rangle$

lemma *all-ctxt-closed-srstep-conversions [simp]*:

$\text{all-ctxt-closed } \mathcal{F} ((\text{srstep } \mathcal{F} \mathcal{R})^{\leftrightarrow*})$
 $\langle \text{proof} \rangle$

lemma *NFP-stepD*:

$\text{NFP } r \implies (a, b) \in r^* \implies (a, c) \in r^* \implies c \in \text{NF } r \implies (b, c) \in r^*$
 $\langle \text{proof} \rangle$

lemma *NE-symmetric*: $\text{NE } r \text{ } s \implies \text{NE } s \text{ } r$

$\langle \text{proof} \rangle$

lemma *CE-symmetric*: $\text{CE } r \text{ } s \implies \text{CE } s \text{ } r$

$\langle \text{proof} \rangle$

Reducing the quantification over rewrite sequences for properties *CR* ...
to rewrite sequences containing at least one root step

lemma *all-ctxt-closed-sig-reflE*:

$\text{all-ctxt-closed } \mathcal{F} \mathcal{R} \implies \text{funas-term } t \subseteq \mathcal{F} \implies (t, t) \in \mathcal{R}$
 $\langle \text{proof} \rangle$

lemma *all-ctxt-closed-relcomp [intro]*:

$(\bigwedge s t. (s, t) \in \mathcal{R} \implies s \neq t \implies \text{funas-term } s \subseteq \mathcal{F} \wedge \text{funas-term } t \subseteq \mathcal{F}) \implies$
 $(\bigwedge s t. (s, t) \in \mathcal{S} \implies s \neq t \implies \text{funas-term } s \subseteq \mathcal{F} \wedge \text{funas-term } t \subseteq \mathcal{F}) \implies$
 $\text{all-ctxt-closed } \mathcal{F} \mathcal{R} \implies \text{all-ctxt-closed } \mathcal{F} \mathcal{S} \implies \text{all-ctxt-closed } \mathcal{F} (\mathcal{R} \circ \mathcal{S})$
 ⟨proof⟩

abbreviation $\text{prop-to-rel } P \equiv \{(s, t) \mid s t. P s t\}$

abbreviation $\text{prop-mctxt-cl } \mathcal{F} P \equiv \text{all-ctxt-closed } \mathcal{F} (\text{prop-to-rel } P)$

lemma prop-mctxt-cl-Var :

$\text{prop-mctxt-cl } \mathcal{F} P \implies P (\text{Var } x) (\text{Var } x)$
 ⟨proof⟩

lemma $\text{prop-mctxt-cl-refl-on}$:

$\text{prop-mctxt-cl } \mathcal{F} P \implies \text{funas-term } t \subseteq \mathcal{F} \implies P t t$
 ⟨proof⟩

lemma $\text{prop-mctxt-cl-reflcl-on}$:

$\text{prop-mctxt-cl } \mathcal{F} P \implies \text{funas-term } s \subseteq \mathcal{F} \implies P s s$
 ⟨proof⟩

lemma $\text{reduction-relations-to-root-step}$:

assumes $\bigwedge s t. (s, t) \in \text{srsteps-with-root-step } \mathcal{F} \mathcal{R} \implies P s t$
and cl: $\text{prop-mctxt-cl } \mathcal{F} P$
and well: $\text{funas-term } s \subseteq \mathcal{F} \text{funas-term } t \subseteq \mathcal{F}$
and steps: $(s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^*$
shows $P s t$ ⟨proof⟩

abbreviation $\text{comp-rrstep-rel } \mathcal{F} \mathcal{R} \mathcal{S} \equiv \text{srsteps-with-root-step } \mathcal{F} \mathcal{R} \circ ((\text{srstep } \mathcal{F} \mathcal{S})^* \cup (\text{srstep } \mathcal{F} \mathcal{R})^* \circ \text{srsteps-with-root-step } \mathcal{F} \mathcal{S})$

abbreviation $\text{comp-rrstep-rel}' \mathcal{F} \mathcal{R} \mathcal{S} \equiv \text{srsteps-with-root-step } \mathcal{F} \mathcal{R} \circ ((\text{srstep } \mathcal{F} \mathcal{S})^+ \cup (\text{srstep } \mathcal{F} \mathcal{R})^+ \circ \text{srsteps-with-root-step } \mathcal{F} \mathcal{S})$

lemma $\text{reduction-join-relations-to-root-step}$:

assumes $\bigwedge s t. (s, t) \in \text{comp-rrstep-rel } \mathcal{F} \mathcal{R} \mathcal{S} \implies P s t$
and cl: $\text{prop-mctxt-cl } \mathcal{F} P$
and well: $\text{funas-term } s \subseteq \mathcal{F} \text{funas-term } t \subseteq \mathcal{F}$
and steps: $(s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^* \circ ((\text{srstep } \mathcal{F} \mathcal{S})^* \circ \text{srsteps-with-root-step } \mathcal{F} \mathcal{S})$
shows $P s t$ ⟨proof⟩

definition $\text{commute-redp } \mathcal{F} \mathcal{R} \mathcal{S} s t \iff (s, t) \in ((\text{srstep } \mathcal{F} \mathcal{S})^* \circ ((\text{srstep } \mathcal{F} \mathcal{R})^{-1})^*)$

declare *subsetI*[rule del]

lemma *commute-redp-mctxt-cl*:

prop-mctxt-cl \mathcal{F} (*commute-redp* \mathcal{F} \mathcal{R} \mathcal{S})

\langle *proof* \rangle

declare *subsetI*[intro!]

lemma *commute-rrstep-intro*:

assumes $\bigwedge s t. (s, t) \in \text{comp-rrstep-rel}' \mathcal{F} (\mathcal{R}^{-1}) \mathcal{S} \implies \text{commute-redp } \mathcal{F} \mathcal{R} \mathcal{S}$
 $s t$

shows *commute* (*srstep* \mathcal{F} \mathcal{R}) (*srstep* \mathcal{F} \mathcal{S})

\langle *proof* \rangle

lemma *commute-to-rrstep*:

assumes *commute* (*srstep* \mathcal{F} \mathcal{R}) (*srstep* \mathcal{F} \mathcal{S})

shows $\bigwedge s t. (s, t) \in \text{comp-rrstep-rel } \mathcal{F} (\mathcal{R}^{-1}) \mathcal{S} \implies \text{commute-redp } \mathcal{F} \mathcal{R} \mathcal{S} s t$
 \langle *proof* \rangle

lemma *CR-Aux*:

assumes $\bigwedge s t. (s, t) \in (\text{srstep } \mathcal{F} (\mathcal{R}^{-1}))^* \mathcal{O} \text{ srsteps-with-root-step } \mathcal{F} \mathcal{R} \implies$
commute-redp $\mathcal{F} \mathcal{R} \mathcal{R} s t$

shows $\bigwedge s t. (s, t) \in \text{comp-rrstep-rel } \mathcal{F} (\mathcal{R}^{-1}) \mathcal{R} \implies \text{commute-redp } \mathcal{F} \mathcal{R} \mathcal{R} s t$
 \langle *proof* \rangle

lemma *CR-rrstep-intro*:

assumes $\bigwedge s t. (s, t) \in (\text{srstep } \mathcal{F} (\mathcal{R}^{-1}))^+ \mathcal{O} \text{ srsteps-with-root-step } \mathcal{F} \mathcal{R} \implies$
commute-redp $\mathcal{F} \mathcal{R} \mathcal{R} s t$

shows *CR* (*srstep* \mathcal{F} \mathcal{R})

\langle *proof* \rangle

lemma *CR-to-rrstep*:

assumes *CR* (*srstep* \mathcal{F} \mathcal{R})

shows $\bigwedge s t. (s, t) \in \text{comp-rrstep-rel } \mathcal{F} (\mathcal{R}^{-1}) \mathcal{R} \implies \text{commute-redp } \mathcal{F} \mathcal{R} \mathcal{R} s t$
 \langle *proof* \rangle

definition *NFP-redp* **where**

NFP-redp $\mathcal{F} \mathcal{R} s t \iff t \in \text{NF } (\text{srstep } \mathcal{F} \mathcal{R}) \longrightarrow (s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^*$

lemma *prop-mctxt-cl-NFP-redp*:

prop-mctxt-cl \mathcal{F} (*NFP-redp* \mathcal{F} \mathcal{R})

\langle *proof* \rangle

lemma *NFP-rrstep-intro*:

assumes $\bigwedge s t. (s, t) \in \text{comp-rrstep-rel}' \mathcal{F} (\mathcal{R}^{-1}) \mathcal{R} \implies \text{NFP-redp } \mathcal{F} \mathcal{R} s t$

shows *NFP* (*srstep* \mathcal{F} \mathcal{R})

\langle *proof* \rangle

lemma *NFP-lift-to-conversion*:

assumes *NFP* r $(s, t) \in (r^{\leftrightarrow})^*$ **and** $t \in \text{NF } r$

shows $(s, t) \in r^*$ \langle *proof* \rangle

lemma *NFP-to-rrstep*:

assumes *NFP* (*srstep* \mathcal{F} \mathcal{R})

shows $\bigwedge s t. (s, t) \in \text{srsteps-with-root-step } \mathcal{F} (\mathcal{R}^{\leftrightarrow}) \implies \text{NFP-redp } \mathcal{F} \mathcal{R} s t$
 $\langle \text{proof} \rangle$

definition *UN-redp* $\mathcal{F} \mathcal{R} s t \iff s \in \text{NF} (\text{srstep } \mathcal{F} \mathcal{R}) \wedge t \in \text{NF} (\text{srstep } \mathcal{F} \mathcal{R})$
 $\longrightarrow s = t$

lemma *prop-mctxt-cl-UN-redp*:

prop-mctxt-cl \mathcal{F} (*UN-redp* \mathcal{F} \mathcal{R})

$\langle \text{proof} \rangle$

lemma *UNC-rrstep-intro*:

assumes $\bigwedge s t. (s, t) \in \text{srsteps-with-root-step } \mathcal{F} (\mathcal{R}^{\leftrightarrow}) \implies \text{UN-redp } \mathcal{F} \mathcal{R} s t$

shows *UNC* (*srstep* \mathcal{F} \mathcal{R})

$\langle \text{proof} \rangle$

lemma *UNC-to-rrstep*:

assumes *UNC* (*srstep* \mathcal{F} \mathcal{R})

shows $\bigwedge s t. (s, t) \in \text{srsteps-with-root-step } \mathcal{F} (\mathcal{R}^{\leftrightarrow}) \implies \text{UN-redp } \mathcal{F} \mathcal{R} s t$

$\langle \text{proof} \rangle$

lemma *UNF-rrstep-intro*:

assumes $\bigwedge t u. (t, u) \in \text{comp-rrstep-rel}' \mathcal{F} (\mathcal{R}^{-1}) \mathcal{R} \implies \text{UN-redp } \mathcal{F} \mathcal{R} t u$

shows *UNF* (*srstep* \mathcal{F} \mathcal{R})

$\langle \text{proof} \rangle$

lemma *UNF-to-rrstep*:

assumes *UNF* (*srstep* \mathcal{F} \mathcal{R})

shows $\bigwedge s t. (s, t) \in \text{comp-rrstep-rel } \mathcal{F} (\mathcal{R}^{-1}) \mathcal{R} \implies \text{UN-redp } \mathcal{F} \mathcal{R} s t$

$\langle \text{proof} \rangle$

lemma *CE-rrstep-intro*:

assumes $\bigwedge s t. (s, t) \in \text{srsteps-with-root-step } \mathcal{F} (\mathcal{R}^{\leftrightarrow}) \implies (s, t) \in (\text{srstep } \mathcal{F} \mathcal{S})^{\leftrightarrow*}$

and $\bigwedge s t. (s, t) \in \text{srsteps-with-root-step } \mathcal{F} (\mathcal{S}^{\leftrightarrow}) \implies (s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^{\leftrightarrow*}$

shows *CE* (*srstep* \mathcal{F} \mathcal{R}) (*srstep* \mathcal{F} \mathcal{S})

$\langle \text{proof} \rangle$

lemma *CE-to-rrstep*:

assumes *CE* (*srstep* \mathcal{F} \mathcal{R}) (*srstep* \mathcal{F} \mathcal{S})

shows $\bigwedge s t. (s, t) \in \text{srsteps-with-root-step } \mathcal{F} (\mathcal{R}^{\leftrightarrow}) \implies (s, t) \in (\text{srstep } \mathcal{F} \mathcal{S})^{\leftrightarrow*}$

$\bigwedge s t. (s, t) \in \text{srsteps-with-root-step } \mathcal{F} (\mathcal{S}^{\leftrightarrow}) \implies (s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^{\leftrightarrow*}$

$\langle \text{proof} \rangle$

definition *NE-redp* **where**

NE-redp $\mathcal{F} \mathcal{R} \mathcal{S} s t \iff t \in \text{NF} (\text{srstep } \mathcal{F} \mathcal{R}) \longrightarrow t \in \text{NF} (\text{srstep } \mathcal{F} \mathcal{R}) \longrightarrow (s, t) \in (\text{srstep } \mathcal{F} \mathcal{S})^*$

lemma *prop-mctxt-cl-NE-redp*:
prop-mctxt-cl \mathcal{F} (*NE-redp* \mathcal{F} \mathcal{R} \mathcal{S})
 ⟨*proof*⟩

lemma *NE-rrstep-intro*:
 assumes $\bigwedge s t. (s, t) \in \text{srsteps-with-root-step } \mathcal{F} \mathcal{R} \implies \text{NE-redp } \mathcal{F} \mathcal{R} \mathcal{S} s t$
 and $\bigwedge s t. (s, t) \in \text{srsteps-with-root-step } \mathcal{F} \mathcal{S} \implies \text{NE-redp } \mathcal{F} \mathcal{S} \mathcal{R} s t$
 and $\text{NF } (\text{srstep } \mathcal{F} \mathcal{R}) = \text{NF } (\text{srstep } \mathcal{F} \mathcal{S})$
 shows $\text{NE } (\text{srstep } \mathcal{F} \mathcal{R}) (\text{srstep } \mathcal{F} \mathcal{S})$
 ⟨*proof*⟩

lemma *NE-to-rrstep*:
 assumes $\text{NE } (\text{srstep } \mathcal{F} \mathcal{R}) (\text{srstep } \mathcal{F} \mathcal{S})$
 shows $\bigwedge s t. (s, t) \in \text{srsteps-with-root-step } \mathcal{F} \mathcal{R} \implies \text{NE-redp } \mathcal{F} \mathcal{R} \mathcal{S} s t$
 $\bigwedge s t. (s, t) \in \text{srsteps-with-root-step } \mathcal{F} \mathcal{S} \implies \text{NE-redp } \mathcal{F} \mathcal{S} \mathcal{R} s t$
 ⟨*proof*⟩

lemma *NE-NF-eq*:
 $\text{NE } \mathcal{R} \mathcal{S} \implies \text{NF } \mathcal{R} = \text{NF } \mathcal{S}$
 ⟨*proof*⟩

abbreviation $\text{SCRp } \mathcal{F} \mathcal{R} t u \equiv \exists v. (t, v) \in (\text{srstep } \mathcal{F} \mathcal{R})^\# \wedge (u, v) \in (\text{srstep } \mathcal{F} \mathcal{R})^*$

lemma *SCR-rrstep-intro*:
 assumes $\bigwedge s t u. (s, t) \in \text{sig-step } \mathcal{F} (\text{rrstep } \mathcal{R}) \implies (s, u) \in \text{srstep } \mathcal{F} \mathcal{R} \implies \text{SCRp } \mathcal{F} \mathcal{R} t u$
 and $\bigwedge s t u. (s, t) \in \text{srstep } \mathcal{F} \mathcal{R} \implies (s, u) \in \text{sig-step } \mathcal{F} (\text{rrstep } \mathcal{R}) \implies \text{SCRp } \mathcal{F} \mathcal{R} t u$
 shows $\text{SCR } (\text{srstep } \mathcal{F} \mathcal{R})$
 ⟨*proof*⟩

lemma *SCE-to-rrstep*:
 assumes $\text{SCR } (\text{srstep } \mathcal{F} \mathcal{R})$
 shows $\bigwedge s t u. (s, t) \in \text{sig-step } \mathcal{F} (\text{rrstep } \mathcal{R}) \implies (s, u) \in \text{srstep } \mathcal{F} \mathcal{R} \implies \text{SCRp } \mathcal{F} \mathcal{R} t u$
 $\bigwedge s t u. (s, t) \in \text{srstep } \mathcal{F} \mathcal{R} \implies (s, u) \in \text{sig-step } \mathcal{F} (\text{rrstep } \mathcal{R}) \implies \text{SCRp } \mathcal{F} \mathcal{R} t u$
 ⟨*proof*⟩

lemma *WCR-rrstep-intro*:
 assumes $\bigwedge s t u. (s, t) \in \text{sig-step } \mathcal{F} (\text{rrstep } \mathcal{R}) \implies (s, u) \in \text{srstep } \mathcal{F} \mathcal{R} \implies (t, u) \in (\text{srstep } \mathcal{F} \mathcal{R})^\downarrow$
 shows $\text{WCR } (\text{srstep } \mathcal{F} \mathcal{R})$
 ⟨*proof*⟩

end

theory *Rewriting-LLRG-LV-Mondaic*
imports *Rewriting*
Replace-Constant
begin

4.3 Specific results about rewriting under a linear variable-separated system

lemma *card-varposs-ground*:

$\text{card } (\text{varposs } s) = 0 \iff \text{ground } s$
 $\langle \text{proof} \rangle$

lemma *poss-of-term-subst-apply-varposs*:

assumes $p \in \text{poss-of-term } (\text{constT } c) (s \cdot \sigma) (c, 0) \notin \text{funas-term } s$
shows $\exists q. q \in \text{varposs } s \wedge q \leq_p p \langle \text{proof} \rangle$

lemma *poss-of-term-hole-poss*:

assumes $p \in \text{poss-of-term } t C\langle s \rangle$ **and** $\text{hole-pos } C \leq_p p$
shows $p \dashv_p \text{hole-pos } C \in \text{poss-of-term } t s \langle \text{proof} \rangle$

lemma *remove-const-subst-from-match*:

assumes $s \cdot \text{const-subst } c = C\langle l \cdot \sigma \rangle (c, 0) \notin \text{funas-term } l \text{ linear-term } l$
shows $\exists D \tau. s = D\langle l \cdot \tau \rangle \langle \text{proof} \rangle$

definition *llrg* $\mathcal{R} \iff (\forall (l, r) \in \mathcal{R}. \text{linear-term } l \wedge \text{ground } r)$

definition *lv* $\mathcal{R} \iff (\forall (l, r) \in \mathcal{R}. \text{linear-term } l \wedge \text{linear-term } r \wedge \text{vars-term } l \cap \text{vars-term } r = \{\})$

definition *monadic* $\mathcal{F} \iff (\forall (f, n) \in \mathcal{F}. n \leq \text{Suc } 0)$

— NF of ground terms

lemma *ground-NF-srstep-gsrstep*:

$\text{ground } s \implies s \in \text{NF } (\text{srstep } \mathcal{F} \mathcal{R}) \implies s \in \text{NF } (\text{gsrstep } \mathcal{F} \mathcal{R})$
 $\langle \text{proof} \rangle$

lemma *NF-to-fresh-const-subst-NF*:

assumes *lin*: $\text{linear-sys } \mathcal{R}$ **and** *fresh-const*: $(c, 0) \notin \text{funas-rel } \mathcal{R} \text{ funas-rel } \mathcal{R} \subseteq \mathcal{F}$

and *nf-f*: $\text{funas-term } s \subseteq \mathcal{F} \ s \in \text{NF } (\text{srstep } \mathcal{F} \mathcal{R})$

shows $s \cdot \text{const-subst } c \in \text{NF } (\text{gsrstep } \mathcal{H} \mathcal{R})$

$\langle \text{proof} \rangle$

lemma *fresh-const-subst-NF-pres*:

assumes *fresh-const*: $(c, 0) \notin \text{funas-rel } \mathcal{R}$ $\text{funas-rel } \mathcal{R} \subseteq \mathcal{F}$

and *nf-f*: $\text{funas-term } s \subseteq \mathcal{F}$ $\mathcal{F} \subseteq \mathcal{H}$ $(c, 0) \in \mathcal{H}$ $s \cdot \text{const-subst } c \in \text{NF} (\text{gsrstep } \mathcal{H} \mathcal{R})$

shows $s \in \text{NF} (\text{srrstep } \mathcal{F} \mathcal{R})$

<proof>

lemma *linear-sys-gNF-eq-NF-eq*:

assumes *lin*: $\text{linear-sys } \mathcal{R}$ $\text{linear-sys } \mathcal{S}$

and *well*: $\text{funas-rel } \mathcal{R} \subseteq \mathcal{F}$ $\text{funas-rel } \mathcal{S} \subseteq \mathcal{F}$

and *fresh*: $(c, 0) \notin \text{funas-rel } \mathcal{R}$ $(c, 0) \notin \text{funas-rel } \mathcal{S}$

and *lift*: $\mathcal{F} \subseteq \mathcal{H}$ $(c, 0) \in \mathcal{H}$

and *nf*: $\text{NF} (\text{gsrstep } \mathcal{H} \mathcal{R}) = \text{NF} (\text{gsrstep } \mathcal{H} \mathcal{S})$

shows $\text{NF} (\text{srrstep } \mathcal{F} \mathcal{R}) = \text{NF} (\text{srrstep } \mathcal{F} \mathcal{S})$

<proof>

lemma *gsrsteps-to-srsteps*:

$(s, t) \in (\text{gsrstep } \mathcal{F} \mathcal{R})^+ \implies (s, t) \in (\text{srrstep } \mathcal{F} \mathcal{R})^+$

<proof>

lemma *gsrsteps-eq-to-srsteps-eq*:

$(s, t) \in (\text{gsrstep } \mathcal{F} \mathcal{R})^* \implies (s, t) \in (\text{srrstep } \mathcal{F} \mathcal{R})^*$

<proof>

lemma *gsrsteps-to-rsteps*:

$(s, t) \in (\text{gsrstep } \mathcal{F} \mathcal{R})^+ \implies (s, t) \in (\text{rstep } \mathcal{R})^+$

<proof>

lemma *gsrsteps-eq-to-rsteps-eq*:

$(s, t) \in (\text{gsrstep } \mathcal{F} \mathcal{R})^* \implies (s, t) \in (\text{rstep } \mathcal{R})^*$

<proof>

lemma *gsrsteps-eq-relcomp-srsteps-relcompD*:

$(s, t) \in (\text{gsrstep } \mathcal{F} \mathcal{R})^* \text{ } O (\text{gsrstep } \mathcal{F} \mathcal{S})^* \implies (s, t) \in (\text{srrstep } \mathcal{F} \mathcal{R})^* \text{ } O (\text{srrstep } \mathcal{F} \mathcal{S})^*$

<proof>

lemma *gsrsteps-eq-relcomp-to-rsteps-relcomp*:

$(s, t) \in (\text{gsrstep } \mathcal{F} \mathcal{R})^* \text{ } O (\text{gsrstep } \mathcal{F} \mathcal{S})^* \implies (s, t) \in (\text{rstep } \mathcal{R})^* \text{ } O (\text{rstep } \mathcal{S})^*$

<proof>

lemma *ground-srsteps-gsrsteps*:

assumes *ground* s *ground* t

and $(s, t) \in (\text{srrstep } \mathcal{F} \mathcal{R})^+$

shows $(s, t) \in (\text{gsrstep } \mathcal{F} \mathcal{R})^+$

<proof>

lemma *ground-srsteps-eq-gsrsteps-eq*:

assumes *ground s ground t*
and $(s, t) \in (\text{srstep } \mathcal{F} \ \mathcal{R})^*$
shows $(s, t) \in (\text{gsrstep } \mathcal{F} \ \mathcal{R})^*$
<proof>

lemma *srsteps-eq-relcomp-gsrsteps-relcomp*:

assumes $(s, t) \in (\text{srstep } \mathcal{F} \ \mathcal{R})^* \ O \ (\text{srstep } \mathcal{F} \ \mathcal{S})^*$
and *ground s ground t*
shows $(s, t) \in (\text{gsrstep } \mathcal{F} \ \mathcal{R})^* \ O \ (\text{gsrstep } \mathcal{F} \ \mathcal{S})^*$
<proof>

lemma *llrg-ground-rhs*:

$\text{llrg } \mathcal{R} \implies (l, r) \in \mathcal{R} \implies \text{ground } r$
<proof>

lemma *llrg-rrsteps-groundness*:

assumes $\text{llrg } \mathcal{R}$ **and** $(s, t) \in (\text{srrstep } \mathcal{F} \ \mathcal{R})$
shows *ground t* *<proof>*

lemma *llrg-rsteps-pres-groundness*:

assumes $\text{llrg } \mathcal{R}$ *ground s*
and $(s, t) \in (\text{srstep } \mathcal{F} \ \mathcal{R})^*$
shows *ground t* *<proof>*

lemma *llrg-srsteps-with-root-step-ground*:

assumes $\text{llrg } \mathcal{R}$ **and** $(s, t) \in \text{srsteps-with-root-step } \mathcal{F} \ \mathcal{R}$
shows *ground t* *<proof>*

lemma *llrg-srsteps-with-root-step-inv-ground*:

assumes $\text{llrg } \mathcal{R}$ **and** $(s, t) \in \text{srsteps-with-root-step } \mathcal{F} \ (\mathcal{R}^{-1})$
shows *ground s* *<proof>*

lemma *llrg-funas-term-step-pres*:

assumes $\text{llrg } \mathcal{R}$ **and** $(s, t) \in (\text{rstep } \mathcal{R})$
shows $\text{funas-term } t \subseteq \text{funas-rel } \mathcal{R} \cup \text{funas-term } s$
<proof>

lemma *llrg-funas-term-steps-pres*:

assumes $\text{llrg } \mathcal{R}$ **and** $(s, t) \in (\text{rstep } \mathcal{R})^*$
shows $\text{funas-term } t \subseteq \text{funas-rel } \mathcal{R} \cup \text{funas-term } s$
<proof>

lemma *monadic-ground-ctxt-apply*:

$\text{monadic } \mathcal{F} \implies \text{funas-ctxt } C \subseteq \mathcal{F} \implies \text{ground } r \implies \text{ground } C\langle r \rangle$
<proof>

lemma *llrg-monadic-rstep-pres-groundness*:

assumes *llrg* \mathcal{R} *monadic* \mathcal{F}
and $(s, t) \in \text{srstep } \mathcal{F} \ \mathcal{R}$
shows *ground* t $\langle \text{proof} \rangle$

lemma *llrg-monadic-rsteps-groundness*:

assumes *llrg* \mathcal{R} *monadic* \mathcal{F}
and $(s, t) \in (\text{srstep } \mathcal{F} \ \mathcal{R})^+$
shows *ground* t $\langle \text{proof} \rangle$

fun *monadic-term* **where**

monadic-term $(\text{Var } x) = \text{True}$
| *monadic-term* $(\text{Fun } f \ \square) = \text{True}$
| *monadic-term* $(\text{Fun } f \ ts) = (\text{length } ts = \text{Suc } 0 \wedge \text{monadic-term } (\text{hd } ts))$

fun *monadic-get-leave* **where**

monadic-get-leave $(\text{Var } x) = (\text{Var } x)$
| *monadic-get-leave* $(\text{Fun } f \ \square) = \text{Fun } f \ \square$
| *monadic-get-leave* $(\text{Fun } f \ ts) = \text{monadic-get-leave } (\text{hd } ts)$

fun *monadic-replace-leave* **where**

monadic-replace-leave t $(\text{Var } x) = t$
| *monadic-replace-leave* t $(\text{Fun } f \ \square) = t$
| *monadic-replace-leave* t $(\text{Fun } f \ ts) = \text{Fun } f \ [\text{monadic-replace-leave } t \ (\text{hd } ts)]$

lemma *monadic-replace-leave-undo-const-subst*:

assumes *monadic-term* s
shows *monadic-replace-leave* $(\text{monadic-get-leave } s) (s \cdot \text{const-subst } c) = s$ $\langle \text{proof} \rangle$

lemma *monadic-replace-leave-context*:

assumes *monadic-term* $C \langle s \rangle$
shows *monadic-replace-leave* t $C \langle s \rangle = C \langle \text{monadic-replace-leave } t \ s \rangle$ $\langle \text{proof} \rangle$

lemma *monadic-replace-leave-subst*:

assumes *monadic-term* $(s \cdot \sigma) \neg \text{ground } s$
shows *monadic-replace-leave* t $(s \cdot \sigma) = s \cdot (\lambda x. \text{monadic-replace-leave } t \ (\sigma \ x))$
 $\langle \text{proof} \rangle$

lemma *monadic-sig*:

monadic $\mathcal{F} \implies (f, \text{length } ts) \in \mathcal{F} \implies \text{length } ts \leq \text{Suc } 0$
 $\langle \text{proof} \rangle$

lemma *monadic-sig-funas-term-mt*:

monadic $\mathcal{F} \implies \text{funas-term } s \subseteq \mathcal{F} \implies \text{monadic-term } s$
 $\langle \text{proof} \rangle$

lemma *monadic-term-const-pres [intro]*:

monadic-term $s \implies \text{monadic-term } (s \cdot \text{const-subst } c)$

<proof>

lemma *remove-const-lv-mondaic-step-lhs:*

assumes *lv: lv* \mathcal{R} **and** *fresh: (c, 0) \notin funas-rel* \mathcal{R}

and *mon: monadic* \mathcal{F}

and *step: (s · const-subst c, t) \in (srstep* \mathcal{F} \mathcal{R})

shows $(s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})$

<proof>

lemma *remove-const-lv-mondaic-step-rhs:*

assumes *lv: lv* \mathcal{R} **and** *fresh: (c, 0) \notin funas-rel* \mathcal{R}

and *mon: monadic* \mathcal{F}

and *step: (s, t · const-subst c) \in (srstep* \mathcal{F} \mathcal{R})

shows $(s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})$

<proof>

lemma *remove-const-lv-mondaic-steps-lhs:*

assumes *lv: lv* \mathcal{R} **and** *fresh: (c, 0) \notin funas-rel* \mathcal{R}

and *mon: monadic* \mathcal{F}

and *steps: (s · const-subst c, t) \in (srstep* \mathcal{F} $\mathcal{R})^+$

shows $(s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^+$

<proof>

lemma *remove-const-lv-mondaic-steps-rhs:*

assumes *lv: lv* \mathcal{R} **and** *fresh: (c, 0) \notin funas-rel* \mathcal{R}

and *mon: monadic* \mathcal{F}

and *steps: (s, t · const-subst c) \in (srstep* \mathcal{F} $\mathcal{R})^+$

shows $(s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^+$

<proof>

lemma *remove-const-lv-mondaic-steps:*

assumes *lv: lv* \mathcal{R} **and** *fresh: (c, 0) \notin funas-rel* \mathcal{R}

and *mon: monadic* \mathcal{F}

and *steps: (s · const-subst c, t · const-subst c) \in (srstep* \mathcal{F} $\mathcal{R})^+$

shows $(s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^+$

<proof>

lemma *lv-root-step-idep-subst:*

assumes *lv* \mathcal{R}

and $(s, t) \in \text{srrstep } \mathcal{F} \mathcal{R}$

and *well: $\bigwedge x. \text{funas-term } (\sigma x) \subseteq \mathcal{F} \bigwedge x. \text{funas-term } (\tau x) \subseteq \mathcal{F}$*

shows $(s \cdot \sigma, t \cdot \tau) \in \text{srrstep } \mathcal{F} \mathcal{R}$

<proof>

lemma *lv-srsteps-with-root-step-idep-subst:*

assumes *lv* \mathcal{R}

and $(s, t) \in \text{srsteps-with-root-step } \mathcal{F} \mathcal{R}$

and well: $\bigwedge x. \text{funas-term } (\sigma x) \subseteq \mathcal{F} \bigwedge x. \text{funas-term } (\tau x) \subseteq \mathcal{F}$
shows $(s \cdot \sigma, t \cdot \tau) \in \text{srrsteps-with-root-step } \mathcal{F} \mathcal{R} \langle \text{proof} \rangle$

end
theory *Rewriting-GTRS*
imports *Rewriting*
Replace-Constant
begin

4.4 Specific results about rewriting under a ground system

abbreviation *ground-sys* $\mathcal{R} \equiv (\forall (s, t) \in \mathcal{R}. \text{ground } s \wedge \text{ground } t)$

lemma *srrstep-ground:*
assumes *ground-sys* \mathcal{R}
and $(s, t) \in \text{srrstep } \mathcal{F} \mathcal{R}$
shows $\text{ground } s \text{ ground } t \langle \text{proof} \rangle$

lemma *srstep-pres-ground-l:*
assumes *ground-sys* \mathcal{R} *ground* s
and $(s, t) \in \text{srstep } \mathcal{F} \mathcal{R}$
shows $\text{ground } t \langle \text{proof} \rangle$

lemma *srstep-pres-ground-r:*
assumes *ground-sys* \mathcal{R} *ground* t
and $(s, t) \in \text{srstep } \mathcal{F} \mathcal{R}$
shows $\text{ground } s \langle \text{proof} \rangle$

lemma *srsteps-pres-ground-l:*
assumes *ground-sys* \mathcal{R} *ground* s
and $(s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^+$
shows $\text{ground } t \langle \text{proof} \rangle$

lemma *srsteps-pres-ground-r:*
assumes *ground-sys* \mathcal{R} *ground* t
and $(s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^+$
shows $\text{ground } s \langle \text{proof} \rangle$

lemma *srsteps-eq-pres-ground-l:*
assumes *ground-sys* \mathcal{R} *ground* s
and $(s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^*$
shows $\text{ground } t \langle \text{proof} \rangle$

lemma *srsteps-eq-pres-ground-r:*
assumes *ground-sys* \mathcal{R} *ground* t
and $(s, t) \in (\text{srstep } \mathcal{F} \mathcal{R})^*$
shows $\text{ground } s \langle \text{proof} \rangle$

lemma *srsteps-with-root-step-ground*:
assumes *ground-sys* \mathcal{R}
and $(s, t) \in \text{srsteps-with-root-step } \mathcal{F} \ \mathcal{R}$
shows *ground* s *ground* t $\langle \text{proof} \rangle$

4.5 funas

lemma *srrstep-funas*:
assumes *ground-sys* \mathcal{R}
and $(s, t) \in \text{srrstep } \mathcal{F} \ \mathcal{R}$
shows *funas-term* $s \subseteq \text{funas-rel } \mathcal{R}$ *funas-term* $t \subseteq \text{funas-rel } \mathcal{R}$ $\langle \text{proof} \rangle$

lemma *srstep-funas-l*:
assumes *ground-sys* \mathcal{R}
and $(s, t) \in \text{srstep } \mathcal{F} \ \mathcal{R}$
shows *funas-term* $t \subseteq \text{funas-term } s \cup \text{funas-rel } \mathcal{R}$ $\langle \text{proof} \rangle$

lemma *srstep-funas-r*:
assumes *ground-sys* \mathcal{R}
and $(s, t) \in \text{srstep } \mathcal{F} \ \mathcal{R}$
shows *funas-term* $s \subseteq \text{funas-term } t \cup \text{funas-rel } \mathcal{R}$ $\langle \text{proof} \rangle$

lemma *srsteps-funas-l*:
assumes *ground-sys* \mathcal{R}
and $(s, t) \in (\text{srstep } \mathcal{F} \ \mathcal{R})^+$
shows *funas-term* $t \subseteq \text{funas-term } s \cup \text{funas-rel } \mathcal{R}$ $\langle \text{proof} \rangle$

lemma *srsteps-funas-r*:
assumes *ground-sys* \mathcal{R}
and $(s, t) \in (\text{srstep } \mathcal{F} \ \mathcal{R})^+$
shows *funas-term* $s \subseteq \text{funas-term } t \cup \text{funas-rel } \mathcal{R}$ $\langle \text{proof} \rangle$

lemma *srsteps-eq-funas-l*:
assumes *ground-sys* \mathcal{R}
and $(s, t) \in (\text{srstep } \mathcal{F} \ \mathcal{R})^*$
shows *funas-term* $t \subseteq \text{funas-term } s \cup \text{funas-rel } \mathcal{R}$ $\langle \text{proof} \rangle$

lemma *srsteps-eq-funas-r*:
assumes *ground-sys* \mathcal{R}
and $(s, t) \in (\text{srstep } \mathcal{F} \ \mathcal{R})^*$
shows *funas-term* $s \subseteq \text{funas-term } t \cup \text{funas-rel } \mathcal{R}$ $\langle \text{proof} \rangle$

lemma *srsteps-with-root-step-funas*:
assumes *ground-sys* \mathcal{R}
and $(s, t) \in \text{srsteps-with-root-step } \mathcal{F} \ \mathcal{R}$
shows *funas-term* $s \subseteq \text{funas-rel } \mathcal{R}$ *funas-term* $t \subseteq \text{funas-rel } \mathcal{R}$
 $\langle \text{proof} \rangle$

end

5 Reducing Rewrite Properties to Properties on Ground Terms over Left-Linear Right-Ground Systems

```

theory Ground-Reduction-on-LLRG
  imports
    Rewriting-Properties
    Rewriting-LLRG-LV-Mondaic
begin

```

```

lemma llrg-linear-sys:
  llrg  $\mathcal{R} \implies$  linear-sys  $\mathcal{R}$ 
  <proof>

```

6 LLRG results

```

lemma llrg-commute:
  assumes sig: funas-rel  $\mathcal{R} \subseteq \mathcal{F}$  funas-rel  $\mathcal{S} \subseteq \mathcal{F}$ 
  and fresh:  $(c, 0) \notin \mathcal{F}$ 
  and llrg: llrg  $\mathcal{R}$  llrg  $\mathcal{S}$ 
  and com: commute (gsrstep (insert  $(c, 0)$   $\mathcal{F}$ )  $\mathcal{R}$ ) (gsrstep (insert  $(c, 0)$   $\mathcal{F}$ )  $\mathcal{S}$ )
  shows commute (srsstep  $\mathcal{F}$   $\mathcal{R}$ ) (srsstep  $\mathcal{F}$   $\mathcal{S}$ )
  <proof>

```

```

lemma llrg-CR:
  assumes sig: funas-rel  $\mathcal{R} \subseteq \mathcal{F}$ 
  and fresh:  $(c, 0) \notin \mathcal{F}$ 
  and llrg: llrg  $\mathcal{R}$ 
  and com: CR (gsrstep (insert  $(c, 0)$   $\mathcal{F}$ )  $\mathcal{R}$ )
  shows CR (srsstep  $\mathcal{F}$   $\mathcal{R}$ )
  <proof>

```

```

lemma llrg-SCR:
  assumes sig: funas-rel  $\mathcal{R} \subseteq \mathcal{F}$  and fresh:  $(c, 0) \notin \mathcal{F}$ 
  and llrg: llrg  $\mathcal{R}$ 
  and scr: SCR (gsrstep (insert  $(c, 0)$   $\mathcal{F}$ )  $\mathcal{R}$ )
  shows SCR (srsstep  $\mathcal{F}$   $\mathcal{R}$ )
  <proof>

```

```

lemma llrg-WCR:
  assumes sig: funas-rel  $\mathcal{R} \subseteq \mathcal{F}$  and fresh:  $(c, 0) \notin \mathcal{F}$ 
  and llrg: llrg  $\mathcal{R}$ 
  and wcr: WCR (gsrstep (insert  $(c, 0)$   $\mathcal{F}$ )  $\mathcal{R}$ )
  shows WCR (srsstep  $\mathcal{F}$   $\mathcal{R}$ )
  <proof>

```

lemma *llrg-UNF*:
assumes *sig*: *funas-rel* $\mathcal{R} \subseteq \mathcal{F}$ **and** *fresh*: $(c, 0) \notin \mathcal{F}$
and *llrg*: *llrg* \mathcal{R}
and *unf*: *UNF* (*gsrstep* (*insert* ($c, 0$) \mathcal{F}) \mathcal{R})
shows *UNF* (*sstep* \mathcal{F} \mathcal{R})
 \langle *proof* \rangle

lemma *llrg-NFP*:
assumes *sig*: *funas-rel* $\mathcal{R} \subseteq \mathcal{F}$ **and** *fresh*: $(c, 0) \notin \mathcal{F}$
and *llrg*: *llrg* \mathcal{R}
and *nfp*: *NFP* (*gsrstep* (*insert* ($c, 0$) \mathcal{F}) \mathcal{R})
shows *NFP* (*sstep* \mathcal{F} \mathcal{R})
 \langle *proof* \rangle

lemma *llrg-NE-aux*:
assumes $(s, t) \in$ *ssteps-with-root-step* \mathcal{F} \mathcal{R}
and *sig*: *funas-rel* $\mathcal{R} \subseteq \mathcal{F}$ *funas-rel* $\mathcal{S} \subseteq \mathcal{F}$ **and** *fresh*: $(c, 0) \notin \mathcal{F}$
and *llrg*: *llrg* \mathcal{R} *llrg* \mathcal{S}
and *ne*: *NE* (*gsrstep* (*insert* ($c, 0$) \mathcal{F}) \mathcal{R}) (*gsrstep* (*insert* ($c, 0$) \mathcal{F}) \mathcal{S})
shows *NE-redp* \mathcal{F} \mathcal{R} \mathcal{S} s t
 \langle *proof* \rangle

lemma *llrg-NE*:
assumes *sig*: *funas-rel* $\mathcal{R} \subseteq \mathcal{F}$ *funas-rel* $\mathcal{S} \subseteq \mathcal{F}$ **and** *fresh*: $(c, 0) \notin \mathcal{F}$
and *llrg*: *llrg* \mathcal{R} *llrg* \mathcal{S}
and *ne*: *NE* (*gsrstep* (*insert* ($c, 0$) \mathcal{F}) \mathcal{R}) (*gsrstep* (*insert* ($c, 0$) \mathcal{F}) \mathcal{S})
shows *NE* (*sstep* \mathcal{F} \mathcal{R}) (*sstep* \mathcal{F} \mathcal{S})
 \langle *proof* \rangle

6.1 Specialized for monadic signature

lemma *monadic-commute*:
assumes *llrg* \mathcal{R} *llrg* \mathcal{S} *monadic* \mathcal{F}
and *com*: *commute* (*gsrstep* \mathcal{F} \mathcal{R}) (*gsrstep* \mathcal{F} \mathcal{S})
shows *commute* (*sstep* \mathcal{F} \mathcal{R}) (*sstep* \mathcal{F} \mathcal{S})
 \langle *proof* \rangle

lemma *monadic-CR*:
assumes *llrg* \mathcal{R} *monadic* \mathcal{F}
and *CR* (*gsrstep* \mathcal{F} \mathcal{R})
shows *CR* (*sstep* \mathcal{F} \mathcal{R}) \langle *proof* \rangle

lemma *monadic-SCR*:
assumes *sig*: *funas-rel* $\mathcal{R} \subseteq \mathcal{F}$ *monadic* \mathcal{F}
and *llrg*: *llrg* \mathcal{R}

and *scr*: $SCR (gsrstep \mathcal{F} \mathcal{R})$
shows $SCR (srstep \mathcal{F} \mathcal{R})$
 <proof>

lemma *monadic-WCR*:
assumes *sig*: $funas-rel \mathcal{R} \subseteq \mathcal{F}$ *monadic* \mathcal{F}
and *llrg*: $llrg \mathcal{R}$
and *wcr*: $WCR (gsrstep \mathcal{F} \mathcal{R})$
shows $WCR (srstep \mathcal{F} \mathcal{R})$
 <proof>

lemma *monadic-UNF*:
assumes *llrg* \mathcal{R} *monadic* \mathcal{F}
and *unf*: $UNF (gsrstep \mathcal{F} \mathcal{R})$
shows $UNF (srstep \mathcal{F} \mathcal{R})$
 <proof>

lemma *monadic-UNC*:
assumes *llrg* \mathcal{R} *monadic* \mathcal{F}
and *well*: $funas-rel \mathcal{R} \subseteq \mathcal{F}$
and *unc*: $UNC (gsrstep \mathcal{F} \mathcal{R})$
shows $UNC (srstep \mathcal{F} \mathcal{R})$
 <proof>

lemma *monadic-NFP*:
assumes *llrg* \mathcal{R} *monadic* \mathcal{F}
and *nfp*: $NFP (gsrstep \mathcal{F} \mathcal{R})$
shows $NFP (srstep \mathcal{F} \mathcal{R})$
 <proof>

end

7 Reducing Rewrite Properties to Properties on Ground Terms over Ground Systems

theory *Ground-Reduction-on-GTRS*
imports
 Rewriting-Properties
 Rewriting-GTRS
 Rewriting-LLRG-LV-Mondaic
begin

lemma *ground-sys-nf-eq-lift*:
fixes $\mathcal{R} :: ('f, 'v)$ *term rel*
assumes *gtrs*: $ground-sys \mathcal{R}$ $ground-sys \mathcal{S}$

and $nf: NF (gsrstep \mathcal{F} \mathcal{R}) = NF (gsrstep \mathcal{F} \mathcal{S})$
shows $NF (srstep \mathcal{F} \mathcal{R}) = NF (srstep \mathcal{F} \mathcal{S})$
 $\langle proof \rangle$

lemma *ground-sys-inv*:
 $ground\text{-}sys \mathcal{R} \implies ground\text{-}sys (\mathcal{R}^{-1}) \langle proof \rangle$

lemma *ground-sys-symcl*:
 $ground\text{-}sys \mathcal{R} \implies ground\text{-}sys (\mathcal{R}^{\leftrightarrow}) \langle proof \rangle$

lemma *ground-sys-comp-rrstep-rel'-ground*:
assumes $ground\text{-}sys \mathcal{R} \ ground\text{-}sys \mathcal{S}$
and $(s, t) \in comp\text{-}rrstep\text{-}rel' \mathcal{F} \mathcal{R} \mathcal{S}$
shows $ground\ s \ ground\ t$
 $\langle proof \rangle$

lemma *GTRS-commute*:
assumes $ground\text{-}sys \mathcal{R} \ ground\text{-}sys \mathcal{S}$
and $com: commute (gsrstep \mathcal{F} \mathcal{R}) (gsrstep \mathcal{F} \mathcal{S})$
shows $commute (srstep \mathcal{F} \mathcal{R}) (srstep \mathcal{F} \mathcal{S})$
 $\langle proof \rangle$

lemma *GTRS-CR*:
assumes $ground\text{-}sys \mathcal{R}$
and $CR (gsrstep \mathcal{F} \mathcal{R})$
shows $CR (srstep \mathcal{F} \mathcal{R}) \langle proof \rangle$

lemma *GTRS-SCR*:
assumes $gtrs: ground\text{-}sys \mathcal{R}$
and $scr: SCR (gsrstep \mathcal{F} \mathcal{R})$
shows $SCR (srstep \mathcal{F} \mathcal{R})$
 $\langle proof \rangle$

lemma *GTRS-WCR*:
assumes $gtrs: ground\text{-}sys \mathcal{R}$
and $wcr: WCR (gsrstep \mathcal{F} \mathcal{R})$
shows $WCR (srstep \mathcal{F} \mathcal{R})$
 $\langle proof \rangle$

lemma *GTRS-UNF*:
assumes $gtrs: ground\text{-}sys \mathcal{R}$
and $unf: UNF (gsrstep \mathcal{F} \mathcal{R})$
shows $UNF (srstep \mathcal{F} \mathcal{R})$
 $\langle proof \rangle$

lemma *GTRS-UNC*:
assumes $gtrs: ground\text{-}sys \mathcal{R}$

and *unc*: $UNC (gsrstep \mathcal{F} \mathcal{R})$
shows $UNC (srstep \mathcal{F} \mathcal{R})$
 ⟨*proof*⟩

lemma *GTRS-NFP*:
assumes *ground-sys* \mathcal{R}
and *nfp*: $NFP (gsrstep \mathcal{F} \mathcal{R})$
shows $NFP (srstep \mathcal{F} \mathcal{R})$
 ⟨*proof*⟩

lemma *GTRS-NE-aux*:
assumes $(s, t) \in srsteps\text{-with-root-step } \mathcal{F} \mathcal{R}$
and *gtrs*: *ground-sys* \mathcal{R} *ground-sys* \mathcal{S}
and *ne*: $NE (gsrstep \mathcal{F} \mathcal{R}) (gsrstep \mathcal{F} \mathcal{S})$
shows $NE\text{-redp } \mathcal{F} \mathcal{R} \mathcal{S} s t$
 ⟨*proof*⟩

lemma *GTRS-NE*:
assumes *gtrs*: *ground-sys* \mathcal{R} *ground-sys* \mathcal{S}
and *ne*: $NE (gsrstep \mathcal{F} \mathcal{R}) (gsrstep \mathcal{F} \mathcal{S})$
shows $NE (srstep \mathcal{F} \mathcal{R}) (srstep \mathcal{F} \mathcal{S})$
 ⟨*proof*⟩

lemma *gtrs-CE-aux*:
assumes *step*: $(s, t) \in srsteps\text{-with-root-step } \mathcal{F} (\mathcal{R}^{\leftrightarrow})$
and *gtrs*: *ground-sys* \mathcal{R} *ground-sys* \mathcal{S}
and *ce*: $CE (gsrstep \mathcal{F} \mathcal{R}) (gsrstep \mathcal{F} \mathcal{S})$
shows $(s, t) \in (srstep \mathcal{F} \mathcal{S})^{\leftrightarrow*}$
 ⟨*proof*⟩

lemma *gtrs-CE*:
assumes *gtrs*: *ground-sys* \mathcal{R} *ground-sys* \mathcal{S}
and *ce*: $CE (gsrstep \mathcal{F} \mathcal{R}) (gsrstep \mathcal{F} \mathcal{S})$
shows $CE (srstep \mathcal{F} \mathcal{R}) (srstep \mathcal{F} \mathcal{S})$
 ⟨*proof*⟩

end

8 Reducing Rewrite Properties to Properties on Ground Terms over Linear Variable-Separated Systems

theory *Ground-Reduction-on-LV*
imports

Rewriting-Properties
Rewriting-LLRG-LV-Mondaic

begin

lemma *lv-linear-sys*: $lv \mathcal{R} \implies linear\text{-}sys \mathcal{R}$
 $\langle proof \rangle$

lemma *comp-rrstep-rel'-sig-mono*:
 $\mathcal{F} \subseteq \mathcal{G} \implies comp\text{-}rrstep\text{-}rel' \mathcal{F} \mathcal{R} \mathcal{S} \subseteq comp\text{-}rrstep\text{-}rel' \mathcal{G} \mathcal{R} \mathcal{S}$
 $\langle proof \rangle$

lemma *srsteps-eqD*: $(s, t) \in (srstep \mathcal{F} \mathcal{R})^* \implies (s, t) \in (rstep \mathcal{R})^*$
 $\langle proof \rangle$

9 Linear-variable separated results

locale *open-terms-two-const-lv* =
fixes $\mathcal{R} :: ('f, 'v) \text{ term rel}$ **and** $\mathcal{F} \ c \ d$
assumes $lv: lv \mathcal{R}$ **and** $sig: funas\text{-}rel \ \mathcal{R} \subseteq \mathcal{F}$
and $fresh: (c, 0) \notin \mathcal{F} \ (d, 0) \notin \mathcal{F}$
and $diff: c \neq d$
begin

abbreviation $\mathcal{H} \equiv insert \ (c, 0) \ (insert \ (d, 0) \ \mathcal{F})$

abbreviation $\sigma_c \equiv const\text{-}subst \ c$

abbreviation $\sigma_d \equiv const\text{-}subst \ d$

lemma *sig-mono*: $\mathcal{F} \subseteq \mathcal{H} \ \langle proof \rangle$

lemma *fresh-sym-c*: $(c, 0) \notin funas\text{-}rel \ \mathcal{R} \ \langle proof \rangle$

lemma *fresh-sym-d*: $(d, 0) \notin funas\text{-}rel \ \mathcal{R} \ \langle proof \rangle$

lemma *fresh-sym-c-inv*: $(c, 0) \notin funas\text{-}rel \ (\mathcal{R}^{-1}) \ \langle proof \rangle$

lemma *fresh-sym-d-inv*: $(d, 0) \notin funas\text{-}rel \ (\mathcal{R}^{-1}) \ \langle proof \rangle$

lemmas *all-fresh* = *fresh-sym-c* *fresh-sym-d* *fresh-sym-c-inv* *fresh-sym-d-inv*

lemma *sig-inv*: $funas\text{-}rel \ (\mathcal{R}^{-1}) \subseteq \mathcal{F} \ \langle proof \rangle$

lemma *lv-inv*: $lv \ (\mathcal{R}^{-1}) \ \langle proof \rangle$

lemma *well-subst*:

$\bigwedge x. funas\text{-}term \ ((const\text{-}subst \ c) \ x) \subseteq \mathcal{H}$

$\bigwedge x. funas\text{-}term \ ((const\text{-}subst \ d) \ x) \subseteq \mathcal{H}$

$\langle proof \rangle$

lemma *srsteps-with-root-step-to-grsteps*:

assumes $(s, t) \in srsteps\text{-}with\text{-}root\text{-}step \ \mathcal{F} \ \mathcal{R}$

shows $(s \cdot \sigma_c, t \cdot \sigma_d) \in (gsrstep \ \mathcal{H} \ \mathcal{R})^*$

$\langle proof \rangle$

lemma *comp-rrstep-rel'-to-grsteps*:

assumes $(s, t) \in \text{comp-rrstep-rel}' \mathcal{F} (\mathcal{R}^{-1}) \mathcal{R}$
shows $(s \cdot \sigma_c, t \cdot \sigma_d) \in (\text{gsrstep } \mathcal{H} (\mathcal{R}^{-1}))^* O (\text{gsrstep } \mathcal{H} \mathcal{R})^*$
 ⟨proof⟩

lemma *gsrsteps-eq-to-srsteps*:
assumes $(s \cdot \sigma_c, t \cdot \sigma_d) \in (\text{gsrstep } \mathcal{H} \mathcal{R})^*$
and *funas*: *funas-term* $s \subseteq \mathcal{F}$ *funas-term* $t \subseteq \mathcal{F}$
shows $(s, t) \in (\text{srrstep } \mathcal{F} \mathcal{R})^*$
 ⟨proof⟩

lemma *convert-NF-to-GNF*:
funas-term $t \subseteq \mathcal{F} \implies t \in \text{NF} (\text{srrstep } \mathcal{F} \mathcal{R}) \implies t \cdot \sigma_c \in \text{NF} (\text{gsrstep } \mathcal{H} \mathcal{R})$
funas-term $t \subseteq \mathcal{F} \implies t \in \text{NF} (\text{srrstep } \mathcal{F} \mathcal{R}) \implies t \cdot \sigma_d \in \text{NF} (\text{gsrstep } \mathcal{H} \mathcal{R})$
 ⟨proof⟩

lemma *convert-GNF-to-NF*:
funas-term $t \subseteq \mathcal{F} \implies t \cdot \sigma_c \in \text{NF} (\text{gsrstep } \mathcal{H} \mathcal{R}) \implies t \in \text{NF} (\text{srrstep } \mathcal{F} \mathcal{R})$
funas-term $t \subseteq \mathcal{F} \implies t \cdot \sigma_d \in \text{NF} (\text{gsrstep } \mathcal{H} \mathcal{R}) \implies t \in \text{NF} (\text{srrstep } \mathcal{F} \mathcal{R})$
 ⟨proof⟩

lemma *lv-CR*:
assumes *cr*: $\text{CR} (\text{gsrstep } \mathcal{H} \mathcal{R})$
shows $\text{CR} (\text{srrstep } \mathcal{F} \mathcal{R})$
 ⟨proof⟩

lemma *lv-WCR*:
assumes *wcr*: $\text{WCR} (\text{gsrstep } \mathcal{H} \mathcal{R})$
shows $\text{WCR} (\text{srrstep } \mathcal{F} \mathcal{R})$
 ⟨proof⟩

lemma *lv-NFP*:
assumes *nfp*: $\text{NFP} (\text{gsrstep } \mathcal{H} \mathcal{R})$
shows $\text{NFP} (\text{srrstep } \mathcal{F} \mathcal{R})$
 ⟨proof⟩

lemma *lv-UNF*:
assumes *unf*: $\text{UNF} (\text{gsrstep } \mathcal{H} \mathcal{R})$
shows $\text{UNF} (\text{srrstep } \mathcal{F} \mathcal{R})$
 ⟨proof⟩

lemma *lv-UNC*:
assumes *unc*: $\text{UNC} (\text{gsrstep } \mathcal{H} \mathcal{R})$
shows $\text{UNC} (\text{srrstep } \mathcal{F} \mathcal{R})$
 ⟨proof⟩

lemma *lv-SCR*:

assumes *scr*: $SCR (gsrstep \mathcal{H} \mathcal{R})$
shows $SCR (srstep \mathcal{F} \mathcal{R})$
 $\langle proof \rangle$

end

locale *open-terms-two-const-lv-two-sys* =
open-terms-two-const-lv \mathcal{R}
for $\mathcal{R} :: ('f, 'v) \text{ term rel} +$
fixes $\mathcal{S} :: ('f, 'v) \text{ term rel}$
assumes *lv-S*: $lv \mathcal{S}$ **and** *sig-S*: $funas-rel \mathcal{S} \subseteq \mathcal{F}$
begin

lemma *fresh-sym-c-S*: $(c, 0) \notin funas-rel \mathcal{S} \langle proof \rangle$

lemma *fresh-sym-d-S*: $(d, 0) \notin funas-rel \mathcal{S} \langle proof \rangle$

lemma *lv-commute*:

assumes *com*: $commute (gsrstep \mathcal{H} \mathcal{R}) (gsrstep \mathcal{H} \mathcal{S})$
shows $commute (srstep \mathcal{F} \mathcal{R}) (srstep \mathcal{F} \mathcal{S})$
 $\langle proof \rangle$

lemma *lv-NE*:

assumes *ne*: $NE (gsrstep \mathcal{H} \mathcal{R}) (gsrstep \mathcal{H} \mathcal{S})$
shows $NE (srstep \mathcal{F} \mathcal{R}) (srstep \mathcal{F} \mathcal{S})$
 $\langle proof \rangle$

end

— CE is special as it only needs one additional constant therefore not included in the locale

lemma *lv-CE-aux*:

assumes $(s, t) \in srsteps-with-root-step \mathcal{F} (\mathcal{R}^{\leftrightarrow})$
and *sig*: $funas-rel \mathcal{R} \subseteq \mathcal{F} \text{ funas-rel } \mathcal{S} \subseteq \mathcal{F}$
and *fresh*: $(c, 0) \notin \mathcal{F}$ **and** *const*: $(a, 0) \in \mathcal{F}$
and *lv*: $lv \mathcal{R} \text{ lv } \mathcal{S}$
and *ce*: $CE (gsrstep (insert (c, 0) \mathcal{F}) \mathcal{R}) (gsrstep (insert (c, 0) \mathcal{F}) \mathcal{S})$
shows $(s, t) \in (srstep \mathcal{F} \mathcal{S})^{\leftrightarrow*}$
 $\langle proof \rangle$

lemma *lv-CE*:

assumes *sig*: $funas-rel \mathcal{R} \subseteq \mathcal{F} \text{ funas-rel } \mathcal{S} \subseteq \mathcal{F}$
and *fresh*: $(c, 0) \notin \mathcal{F}$ **and** *const*: $(a, 0) \in \mathcal{F}$
and *lv*: $lv \mathcal{R} \text{ lv } \mathcal{S}$
and *ce*: $CE (gsrstep (insert (c, 0) \mathcal{F}) \mathcal{R}) (gsrstep (insert (c, 0) \mathcal{F}) \mathcal{S})$

shows $CE (srstep \mathcal{F} \mathcal{R}) (srstep \mathcal{F} \mathcal{S})$
<proof>

9.1 Specialized for monadic signature

lemma *lv-NE-aux*:

assumes $(s, t) \in srsteps\text{-with-root-step } \mathcal{F} \mathcal{R}$ **and** *fresh*: $(c, 0) \notin \mathcal{F}$
and *sig*: $funas\text{-rel } \mathcal{R} \subseteq \mathcal{F} \text{ funas-rel } \mathcal{S} \subseteq \mathcal{F}$
and *lv*: $lv \mathcal{R} \text{ lv } \mathcal{S}$
and *mon*: *monadic* \mathcal{F}
and *ne*: $NE (gsrstep (insert (c, 0) \mathcal{F}) \mathcal{R}) (gsrstep (insert (c, 0) \mathcal{F}) \mathcal{S})$
shows $NE\text{-redp } \mathcal{F} \mathcal{R} \mathcal{S} s t$
<proof>

lemma *lv-NE*:

assumes *sig*: $funas\text{-rel } \mathcal{R} \subseteq \mathcal{F} \text{ funas-rel } \mathcal{S} \subseteq \mathcal{F}$
and *mon*: *monadic* \mathcal{F} **and** *fresh*: $(c, 0) \notin \mathcal{F}$
and *lv*: $lv \mathcal{R} \text{ lv } \mathcal{S}$
and *ne*: $NE (gsrstep (insert (c, 0) \mathcal{F}) \mathcal{R}) (gsrstep (insert (c, 0) \mathcal{F}) \mathcal{S})$
shows $NE (srstep \mathcal{F} \mathcal{R}) (srstep \mathcal{F} \mathcal{S})$
<proof>

end

References

- [1] C. Sternagel and R. Thiemann. Abstract rewriting. *Archive of Formal Proofs*, June 2010. <https://isa-afp.org/entries/Abstract-Rewriting.html>, Formal proof development.