

Termination Restricted to Right-Forward Closures

René Thiemann¹, Dieter Hofbauer², Ulysse Le Huitouze³, and
Johannes Waldmann⁴

¹University of Innsbruck, Austria

²ASW Saarland, Germany

³University of Rennes, France

⁴HTWK Leipzig, Germany

August 26, 2026

Abstract

In this AFP entry we formalize Dershowitz’ theorem that termination of a term rewrite system (TRS) starting from arbitrary terms is equivalent to termination starting from terms in the right-forward closures of right-hand sides, provided that the TRS is right-linear or orthogonal [1]. Our proof deviates from the original one in that no reorderings of steps in infinite derivations are required, making it more precise in its argumentation. We also integrate a later result that one can weaken orthogonality to locally confluent overlay TRSs [2].

In order to arrive at these statements, we had to formalize two further results: Following Gramlich, we prove that for locally confluent overlay TRSs the notions of termination and innermost termination coincide [3]; and we verify that narrowing with a right-linear TRS preserves linearity of terms.

For further details, we refer to our FSCD 2026 paper [4].

Contents

1	Rewriting with the Innermost Strategy	2
2	Gramlich’s Criterion to Prove Termination by Innermost Termination	3
3	Linearity Preservation by Unification	4
4	Narrowing and Linearity Preservation	6
4.1	Preparations for Narrowing	6
4.2	Narrowing Relations	7

1 Rewriting with the Innermost Strategy

```
theory Innermost-Rewriting
  imports First-Order-Rewriting.Trs
begin
```

```
inductive-set inn-rstep :: ('f, 'v) trs  $\Rightarrow$  ('f, 'v) term rel for R
  where inn-rstep[intro]:  $(l, r) \in R \implies \text{set } (\text{args } (l \cdot \sigma)) \subseteq \text{NF-trs } R \implies (C\langle l \cdot \sigma \rangle, C\langle r \cdot \sigma \rangle) \in \text{inn-rstep } R$ 
```

```
lemma inn-rstep-rstep: inn-rstep R  $\subseteq$  rstep R
  <proof>
```

```
lemma inn-rstep-ctxt: assumes  $(s, t) \in \text{inn-rstep } R$ 
  shows  $(C\langle s \rangle, C\langle t \rangle) \in \text{inn-rstep } R$ 
  <proof>
```

```
lemma ctxt-closed-inn-rstep[simp, intro]: ctxt.closed (inn-rstep R)
  <proof>
```

```
inductive-set inn-nrrstep :: ('f, 'v) trs  $\Rightarrow$  ('f, 'v) term rel for R
  where inn-nrrstep[intro]:  $(l, r) \in R \implies \text{set } (\text{args } (l \cdot \sigma)) \subseteq \text{NF-trs } R \implies C \neq \text{Hole} \implies (C\langle l \cdot \sigma \rangle, C\langle r \cdot \sigma \rangle) \in \text{inn-nrrstep } R$ 
```

```
lemma inn-nrrstep-nrrstep: inn-nrrstep R  $\subseteq$  nrrstep R
  <proof>
```

```
lemma inn-nrrstep-inn-rstep: inn-nrrstep R  $\subseteq$  inn-rstep R
  <proof>
```

```
lemma inn-nrrstep-ctxt: assumes  $(s, t) \in \text{inn-nrrstep } R$ 
  shows  $(C\langle s \rangle, C\langle t \rangle) \in \text{inn-nrrstep } R$ 
  <proof>
```

```
lemma inn-rstep-ctxt-inn-nrrstep: assumes  $(s, t) \in \text{inn-rstep } R$ 
  and  $C \neq \text{Hole}$ 
  shows  $(C\langle s \rangle, C\langle t \rangle) \in \text{inn-nrrstep } R$ 
  <proof>
```

```
lemma NF-inn-rstep-rstep: NF (inn-rstep R) = NF (rstep R)
  <proof>
```

```
lemma NF-inn-nrrstep-nrrstep: NF (inn-nrrstep R) = NF (nrrstep R)
  <proof>
```

inductive-set *inn-rrstep* :: (*'f*, *'v*) *trs* $\Rightarrow$ (*'f*, *'v*) *term rel* **for** *R*
where *inn-rrstep*[*intro*]: (*l*, *r*) $\in R \Rightarrow \text{set } (\text{args } (l \cdot \sigma)) \subseteq \text{NF-trs } R \Rightarrow (l \cdot \sigma,
r \cdot \sigma) \in inn-rrstep R$

lemma *inn-rrstep-rrstep*: *inn-rrstep R* $\subseteq$ *rrstep R*
 $\langle \text{proof} \rangle$

lemma *inn-rrstep-inn-rstep*: *inn-rrstep R* $\subseteq$ *inn-rstep R*
 $\langle \text{proof} \rangle$

lemma *inn-rstep-iff-inn-rrstep-or-inn-nrrstep*: *inn-rstep R* = (*inn-rrstep R* $\cup$ *inn-nrrstep R*)
 $\langle \text{proof} \rangle$

lemma *inn-rstep-cases*[*consumes 1*, *case-names root nonroot*]:
 $\llbracket (s, t) \in \text{inn-rstep } R; (s, t) \in \text{inn-rrstep } R \Rightarrow P; (s, t) \in \text{inn-nrrstep } R \Rightarrow P \rrbracket$
 $\Rightarrow P$
 $\langle \text{proof} \rangle$

lemma *inn-nrrstep-args*:
assumes (*s*, *t*) $\in \text{inn-nrrstep } R$
shows $\exists f \text{ ss ts. } s = \text{Fun } f \text{ ss} \wedge t = \text{Fun } f \text{ ts} \wedge \text{length ss} = \text{length ts}$
 $\wedge (\exists j < \text{length ss. } (ss!j, ts!j) \in \text{inn-rstep } R \wedge (\forall i < \text{length ss. } i \neq j \longrightarrow ss!i = ts!i))$
 $\langle \text{proof} \rangle$

lemma *Tinf-imp-SN-inn-nrrstep*: **assumes** *t* $\in \text{Tinf } (\text{inn-rstep } R)$
shows *SN-on* (*inn-nrrstep R*) {*t*}
 $\langle \text{proof} \rangle$

lemma *Tinf-inn-rstep-imp-first-root-step*: **assumes** *s* $\in \text{Tinf } (\text{inn-rstep } R)$
shows $\exists t \text{ u. } (s, t) \in (\text{inn-nrrstep } R) \hat{*} \wedge (t, u) \in \text{inn-rrstep } R \wedge \neg \text{SN-on}$
(*inn-rstep R*) {*u*}
 $\langle \text{proof} \rangle$

end

2 Gramlich's Criterion to Prove Termination by Innermost Termination

Gramlich showed that for locally confluent overlay TRSs, innermost termination and termination coincide. We formalize this result in this theory.

```

theory Gramlich-Innermost-Switch
  imports
    First-Order-Rewriting.Critical-Pairs
    Innermost-Rewriting
  begin

lemma WCR-on-rstep-imp-WCR-on-nrrstep: assumes WCR-on (rstep R) {t. SN-on
(rstep R) {t}}
  shows WCR-on (nrrstep R) {t. SN-on (nrrstep R) {t}}
  <proof>

lemma SN-innermost-switch-locally-confluent-overlay-local:
  assumes WCR: WCR-on (rstep R) {t. SN-on (rstep R) {t}}
  and overlay:  $\bigwedge l\ r. (False, l, r) \notin \text{critical-pairs ren } R\ R$ 
  and wf: wf-trs R
shows SN-on (inn-rstep R) = SN-on (rstep R)
  <proof>

lemma SN-innermost-switch-locally-confluent-overlay:
  assumes WCR: WCR-on (rstep R) {t. SN-on (rstep R) {t}}
  and overlay:  $\bigwedge l\ r. (False, l, r) \notin \text{critical-pairs ren } R\ R$ 
  and wf: wf-trs R
shows SN (inn-rstep R) = SN (rstep R)
  <proof>

end

```

3 Linearity Preservation by Unification

```

theory Linear-Unification
  imports
    First-Order-Terms.Unification-More
  begin

```

A sufficient criterion to ensure that $t \cdot \sigma$ is linear.

```

lemma linear-term-subst: linear-term t
   $\implies (\bigwedge x. x \in \text{vars-term } t \implies \text{linear-term } (\sigma\ x))$ 
   $\implies (\bigwedge x\ y. x \in \text{vars-term } t \implies y \in \text{vars-term } t \implies x \neq y \implies \text{vars-term } (\sigma\ x)$ 
 $\cap \text{vars-term } (\sigma\ y) = \{\})$ 
   $\implies \text{linear-term } (t \cdot \sigma)$ 
  <proof>

```

Unification of two var disjoint terms where one of them is linear results in a partially linear substitution and linear terms

definition vars-mset-left :: $((f, 'v)\text{term} \times (f, 'v)\text{term}) \text{ multiset} \Rightarrow 'v \text{ multiset}$ **where**
 vars-mset-left m = sum-mset (image-mset (vars-term-ms o fst) m)

definition *vars-mset-right* :: $((f, 'v)term \times (f, 'v)term) \text{ multiset} \Rightarrow 'v \text{ multiset}$
where

vars-mset-right *m* = *sum-mset* (*image-mset* (*vars-term-ms* *o snd*) *m*)

definition *linear-mset* :: $'a \text{ multiset} \Rightarrow \text{bool}$ **where**

linear-mset *m* = $(\forall x. \text{count } m \ x \leq 1)$

lemma *count-sum-mset-image-mset*:

count (*sum-mset* (*image-mset* *f m*)) *x* = *sum-mset* (*image-mset* $(\lambda a. \text{count } (f \ a) \ x) \ m$)
 <proof>

lemma *linear-vars-term-ms*: *linear-mset* (*vars-term-ms* *t*) = *linear-term* *t*

<proof>

lemma *linear-term-count*: **assumes** *linear-term* *t*

shows *count* (*vars-term-ms* *t*) *x* ≤ 1

<proof>

lemma *linear-term-Var-subst*: *linear-term* (*t* · (*Var* *o r*)) $\Longrightarrow$ *linear-term* *t*

<proof>

lemma *vars-mset-right-add[simp]*: *vars-mset-right* (*add-mset* *p E*) = *vars-term-ms* (*snd* *p*) + *vars-mset-right* *E*

<proof>

lemma *right-linear-var-disjoint-mgu-mset*: **fixes** *E* :: $((f, 'v)term \times (f, 'v)term) \text{ multiset}$

and *u* :: $(f, 'v)term$

assumes *set-mset* (*vars-mset-left* *E*) $\cap$ *set-mset* (*vars-mset-right* *E*) = {}

and *linear-mset* (*vars-mset-right* *E*)

and *is-mgu* σ (*set-mset* *E*)

and *vars-term* *u* $\cap$ *set-mset* (*vars-mset-right* *E*) = {}

and *linear-term* *u*

shows *linear-term* (*u* · σ)

<proof>

lemma *right-linear-var-disjoint-mgu*: **fixes** *s t* :: $(f, 'v)term$

assumes *disj*: *vars-term* *s* $\cap$ *vars-term* *t* = {}

and *lin*: *linear-term* *t*

and *mgu*: *is-mgu* σ {(*s*, *t*)}

and *linu*: *linear-term* *u*

and *disju*: *vars-term* *u* $\cap$ *vars-term* *t* = {}

shows *linear-term* (*u* · σ)

<proof>

Corollary: Unification of two linear var disjoint terms results in a linear

substitution and linear unified terms.

lemma *linear-var-disjoint-is-mgu*: **fixes** $s\ t :: ('f, 'v)\text{term}$
assumes *disj*: $\text{vars-term } s \cap \text{vars-term } t = \{\}$
and *lin*: $\text{linear-term } s\ \text{linear-term } t$
and *mgu*: $\text{is-mgu } \sigma\ \{(s, t)\}$
shows $\text{vars-term } u \cap \text{vars-term } t = \{\} \implies \text{linear-term } u \implies \text{linear-term } (u \cdot \sigma)$
and $\text{vars-term } u \cap \text{vars-term } s = \{\} \implies \text{linear-term } u \implies \text{linear-term } (u \cdot \sigma)$
and $\text{linear-term } (s \cdot \sigma)\ \text{linear-term } (t \cdot \sigma)$
and $\text{linear-term } (\sigma\ x)$
 $\langle \text{proof} \rangle$
end

4 Narrowing and Linearity Preservation

In this theory we define narrowing and show that narrowing of a linear term with a right-linear TRS results in a linear term again. Moreover, some further results on narrowing are formalized.

theory *Linear-Narrowing*
imports
Automatic-Refinement.Misc
Linear-Unification
First-Order-Rewriting.Trs
begin

4.1 Preparations for Narrowing

definition *subst-term-mset* :: $('f, 'v)\text{term} \Rightarrow ('f, 'v)\text{subst} \Rightarrow ('f, 'v)\text{term multiset}$
where $\text{subst-term-mset } t\ \sigma = \text{image-mset } \sigma\ (\text{vars-term-ms } t)$

lemma *vars-term-ms-subst-compose-split*:
 $(\delta\ \#\ (\sum x \in \#xs. \text{vars-term-ms } (\mu\ x)), (\mu \circ_s \delta)\ \#\ xs) \in (\text{mult } \{\triangleleft\})^=$
 $\wedge (\text{Ball } (\mu\ \text{'set-mset } xs)\ \text{is-Var} \vee (\delta\ \#\ (\sum x \in \#xs. \text{vars-term-ms } (\mu\ x)), (\mu \circ_s$
 $\delta)\ \#\ xs) \in \text{mult } \{\triangleleft\})$
 $(\text{is } - \in ?R^= \wedge -)$
 $\langle \text{proof} \rangle$

lemma *linear-term-vars-term-ms*: **assumes** $\text{vars-term } r \subseteq \text{vars-term } l$
and $\text{linear-term } r$
shows $\text{vars-term-ms } r \subseteq \# \text{vars-term-ms } l$
 $\langle \text{proof} \rangle$

lemma *vars-term-ms-map-vars-term[simp]*: $\text{vars-term-ms } (\text{map-vars-term } f\ t) = f$
 $\#\ \text{vars-term-ms } t$
 $\langle \text{proof} \rangle$

lemma *linear-term-map-inj-on*: **assumes** $\text{linear-term } (\text{map-vars-term } f\ t)$
shows $\text{inj-on } f\ (\text{vars-term } t)$
 $\langle \text{proof} \rangle$

4.2 Narrowing Relations

context

fixes $ren :: 'v :: infinite\ renaming2$

begin

definition

$narrows-r-p-s :: ('f, 'v) trs \Rightarrow ('f, 'v) rule \Rightarrow pos \Rightarrow ('f, 'v)subst \Rightarrow ('f, 'v)trs$

where

$narrows-r-p-s\ R\ r\ p\ \mu \equiv \{(s, t). \exists\ \mu2. p \in poss\ s \wedge is-Fun\ (s \mid -\ p) \wedge r \in R \wedge$
 $mgu-vd\ ren\ (s \mid -\ p)\ (fst\ r) = Some\ (\mu, \mu2)$
 $\wedge t = replace-at\ (s \cdot \mu)\ p\ (snd\ r \cdot \mu2)\}$

definition

$narrow-step-s :: ('f, 'v) trs \Rightarrow ('f, 'v)subst \Rightarrow ('f, 'v)trs$ **where**
 $narrow-step-s\ R\ \mu = \{st \mid st\ lr\ p. st \in narrows-r-p-s\ R\ lr\ p\ \mu\}$

definition $narrow-step :: ('f, 'v) trs \Rightarrow ('f, 'v)trs$ **where**

$narrow-step\ R = \{st \mid st\ lr\ p\ \mu. st \in narrows-r-p-s\ R\ lr\ p\ \mu\}$

lemma $narrow-step-to-s: narrow-step\ R = \bigcup\ (range\ (narrow-step-s\ R))$

$\langle proof \rangle$

theorem $right-linear-rule-narrowing: fixes\ R :: ('f, 'v)trs$

assumes $linr: linear-term\ r$

and $lins: linear-term\ s$

and $narr: (s, t) \in narrows-r-p-s\ R\ (l, r)\ p\ \mu$

shows $linear-term\ t$

$\langle proof \rangle$

theorem $right-linear-rule-narrow-step:$

assumes $\bigwedge\ lr. lr \in R \implies linear-term\ (snd\ lr)$

and $lins: linear-term\ s$

and $narr: (s, t) \in narrow-step\ R$

shows $linear-term\ t$

$\langle proof \rangle$

lemma $narrowing-right-linear-one-step-simulation-r-p-s: fixes\ R :: ('f, 'v)trs$

assumes $(s \cdot \sigma, t) \in rstep-r-p-s\ R\ (l, r)\ p\ \mu$

and $lins: linear-term\ s$

and $linr: linear-term\ r$

and $var-cond: vars-term\ r \subseteq vars-term\ l$

shows $p \notin fun-poss\ s \wedge$

$(\exists\ \delta. t = s \cdot \delta \wedge (subst-term-mset\ s\ \delta, subst-term-mset\ s\ \sigma) \in mult1$

$((rstep\ R)^{\wedge-1}))$

$\vee$

$p \in fun-poss\ s \wedge$

$(\exists\ \mu1\ \mu2\ \delta\ u. t = u \cdot \delta \wedge linear-term\ u \wedge (s, u) \in narrows-r-p-s\ R\ (l, r)\ p$

$\mu1 \wedge \sigma = \mu1 \circ_s \delta \wedge \mu = \mu2 \circ_s \delta$

$\wedge (\text{subst-term-mset } u \ \delta, \text{subst-term-mset } s \ \sigma) \in (\text{mult } \{\triangleleft\})^=)$
 $\langle \text{proof} \rangle$

lemma *narrowing-right-linear-one-step-simulation*: **fixes** $R :: ('f, 'v)\text{trs}$
assumes $(s \cdot \sigma, t) \in \text{rstep } R$
and $\text{lin}s$: *linear-term* s
and *right-lin*: $\bigwedge \text{lr}. \text{lr} \in R \implies \text{linear-term } (\text{snd } \text{lr})$
and *var-cond*: $\bigwedge \text{lr}. \text{lr} \in R \implies \text{vars-term } (\text{snd } \text{lr}) \subseteq \text{vars-term } (\text{fst } \text{lr})$
shows $(\exists \delta. t = s \cdot \delta \wedge (\text{subst-term-mset } s \ \delta, \text{subst-term-mset } s \ \sigma) \in \text{mult1 } ((\text{rstep } R)^{\wedge-1}))$
 $\vee (\exists \mu \ \delta \ u. t = u \cdot \delta \wedge \text{linear-term } u \wedge (s, u) \in \text{narrow-step-s } R \ \mu \wedge \sigma = \mu \circ_s \delta)$
 δ
 $\wedge (\text{subst-term-mset } u \ \delta, \text{subst-term-mset } s \ \sigma) \in (\text{mult } \{\triangleleft\})^=)$
 $\langle \text{proof} \rangle$

lemma *rstep-instance-imp-narrows-r-p-s*: **assumes** $\text{step}: (s \cdot \sigma, t) \in \text{rstep-r-p-s } R$
 $r \ p \ \tau$
and $\text{not}\sigma$: $p \in \text{poss } s \text{ is-Fun } (s \mid - \ p)$
shows $\exists \mu \ \mu_2 \ \delta \ u. (s, u) \in \text{narrows-r-p-s } R \ r \ p \ \mu \wedge s \cdot \sigma = s \cdot \mu \cdot \delta \wedge t = u \cdot \delta \wedge \sigma = \mu \circ_s \delta \wedge \tau = \mu_2 \circ_s \delta$
 $\langle \text{proof} \rangle$

lemma *narrows-r-p-s-imp-rstep-r-p-s*: **assumes** $\text{narr}: (s, t) \in \text{narrows-r-p-s } R \ r \ p$
 μ
shows $\exists \delta. (s \cdot \mu, t) \in \text{rstep-r-p-s } R \ r \ p \ \delta$
 $\langle \text{proof} \rangle$

lemma *narrows-r-p-s-imp-rstep*: **assumes** $\text{narr}: (s, t) \in \text{narrows-r-p-s } R \ r \ p \ \mu$
shows $(s \cdot \mu, t) \in \text{rstep } R$
 $\langle \text{proof} \rangle$

lemma *narrow-step-s-to-rstep*: $(s, t) \in \text{narrow-step-s } R \ \mu \implies (s \cdot \mu, t) \in \text{rstep } R$
 $\langle \text{proof} \rangle$

lemma *rstep-imp-narrows-r-p-s*: **assumes** $\text{step}: (s, t) \in \text{rstep-r-p-s } R \ r \ p \ \tau$
and wf : *wf-trs* R
shows $\exists \mu \ \delta \ u. (s, u) \in \text{narrows-r-p-s } R \ r \ p \ \mu \wedge s = s \cdot \mu \cdot \delta \wedge t = u \cdot \delta \wedge u = t \cdot \mu \wedge \mu \circ_s \delta = \text{Var}$
 $\langle \text{proof} \rangle$

lemma *rstep-imp-narrows-s*: **assumes** $\text{step}: (s, t) \in \text{rstep } R$
and wf : *wf-trs* R
shows $\exists \mu \ \delta \ u. (s, u) \in \text{narrow-step-s } R \ \mu \wedge s = s \cdot \mu \cdot \delta \wedge t = u \cdot \delta \wedge u = t \cdot \mu \wedge \mu \circ_s \delta = \text{Var}$
 $\langle \text{proof} \rangle$

lemma *rstep-imp-narrows*: **assumes** $\text{step}: (s, t) \in \text{rstep } R$
and wf : *wf-trs* R

shows $\exists \mu \delta u. (s, u) \in \text{arrow-step } R \wedge t = u \cdot \delta \wedge u = t \cdot \mu$
 $\langle \text{proof} \rangle$

lemma narrow-instance-pos-in-subst: assumes
 $\text{narr}: (t \cdot \gamma, s) \in \text{narrows-r-p-s } (R :: ('f, 'v)\text{trs}) (l, r) \ p \ \mu$
and $\text{pos}: p \notin \text{fun-poss } t$
and $\text{vars}: \text{vars-term } r \subseteq \text{vars-term } l$
and $\text{lin}: \text{linear-term } (t :: ('f, 'v)\text{term}) \text{ linear-term } r$
shows $\exists \gamma. t \cdot \gamma = s$
 $\langle \text{proof} \rangle$

lemma narrow-instance-pos-in-term: fixes $t :: ('f, 'v)\text{term}$
assumes $\text{narr}: (t, s) \in \text{narrows-r-p-s } R (l, r) \ p \ \mu 1$
and $\text{narr-inst}: (t \cdot \gamma, u) \in \text{narrows-r-p-s } R (l, r) \ p \ \mu 2$
shows $\exists \delta. s \cdot \delta = u \wedge \gamma \circ_s \mu 2 = \mu 1 \circ_s \delta$
 $\langle \text{proof} \rangle$

lemma SN-on-narrows-imp-SN-on-rstep: fixes $R :: ('f, 'v)\text{trs}$
assumes $\text{SN-on } (\text{arrow-step } R) \ \{s\}$
and $\text{wf}: \text{wf-trs } R$
shows $\text{SN-on } (\text{rstep } R) \ \{s\}$
 $\langle \text{proof} \rangle$

lemma exists-narrow-steps-to-infinite-rsteps: fixes $R :: ('f, 'v)\text{trs}$
defines $Q \equiv \lambda (n :: \text{nat}) \ s \ (\sigma :: ('f, 'v)\text{subst}) \ u \ (\delta :: ('f, 'v)\text{subst}).$
 $(\exists \mu. (s, u) \in \text{arrow-step-s } R \ \mu \wedge \sigma = \mu \circ_s \delta)$
assumes $\text{wf}: \text{wf-trs } R$
and $\text{single-steps}: \bigwedge n \ s \ \sigma. P \ n \ s \ \sigma \implies \exists u \ \delta. P \ (\text{Suc } n) \ u \ \delta \wedge Q \ n \ s \ \sigma \ u \ \delta$
and $P0: P \ 0 \ r \ \sigma$
shows $\neg \text{SN-on } (\text{arrow-step } R) \ \{r\}$
 $\neg \text{SN-on } (\text{rstep } R) \ ((\text{arrow-step } R)^* \ \{r\})$
 $\langle \text{proof} \rangle$

end
end

5 Dershowitz' Right-Forward Closures for Proving Termination

Right-Forward Closures (RFC) are defined, and it is shown that termination on all terms is equivalent to termination starting on RFC for TRSs that are right-linear, or for overlay TRSs that are locally confluent.

theory Right-Forward-Closure
imports
 Linear-Narrowing
 $\text{First-Order-Rewriting.Critical-Pairs}$

Gramlich-Innermost-Switch
begin

context
fixes $ren :: 'v :: infinite\ renaming2$
begin

definition *right-forw-closure* $:: ('f, 'v)trs \Rightarrow ('f, 'v)term\ set$ **where**
right-forw-closure $R = (narrow-step\ ren\ R)^* \text{ `` } rhss\ R$

lemma *right-forw-closure-SN-main*: **fixes** $R :: ('f, 'v)trs$
assumes *right-lin*: $\bigwedge lr. lr \in R \Longrightarrow linear-term\ (snd\ lr)$
and *wf*: $wf-trs\ R$
and *nSN*: $\neg SN\ (rstep\ R)$
shows $\neg SN-on\ (rstep\ R)\ (right-forw-closure\ R)$
 $\langle proof \rangle$

theorem *right-linear-SN-rstep-RFC*:
assumes *wf*: $wf-trs\ R$
and *rlin*: $\bigwedge lr. lr \in R \Longrightarrow linear-term\ (snd\ lr)$
shows $SN\ (rstep\ R) \longleftrightarrow SN-on\ (rstep\ R)\ (right-forw-closure\ R)$ (**is** $?A = ?B$)
 $\langle proof \rangle$

context
fixes $R :: ('f, 'v)trs$
assumes *wf*: $wf-trs\ R$
and *WCR*: $WCR\ (rstep\ R)$
and *overlay*: $\bigwedge l\ r. (False, l, r) \notin critical-pairs\ ren\ R\ R$
begin

lemma *WCRO-not-SN-imp-non-terminating-innermost-rhs-instance*:
assumes $\neg SN\ (rstep\ R)$
shows $\exists r\ \sigma. r \in rhss\ R \wedge \neg SN-on\ (inn-rstep\ R)\ \{r \cdot \sigma\} \wedge \sigma \text{ `` } vars-term\ r \subseteq NF\ (rstep\ R)$
 $\langle proof \rangle$

lemma *WCRO-one-step-simulation-by-narrowing*: **fixes** $s :: ('f, 'v)term$
assumes *nSN*: $\neg SN-on\ (inn-rstep\ R)\ \{s \cdot \sigma\}$
and *NF-sigma*: $\sigma \text{ `` } vars-term\ s \subseteq NF\ (rstep\ R)$
shows $\exists u\ \mu\ \delta. (s, u) \in narrow-step-s\ ren\ R\ \mu$
 $\wedge \neg SN-on\ (inn-rstep\ R)\ \{u \cdot \delta\}$
 $\wedge \delta \text{ `` } vars-term\ u \subseteq NF\ (rstep\ R)$
 $\wedge \sigma = \mu \circ_s \delta$
 $\langle proof \rangle$

lemma *WCRO-not-SN-rstep-imp-not-SN-rstep*: **assumes** $\neg SN \ (rstep \ R)$
shows $\neg SN\text{-on} \ (rstep \ R)$ (*right-forw-closure* R)
 $\langle proof \rangle$
end

theorem *WCRO-SN-rstep-RFC*: **assumes** $wf: wf\text{-trs} \ R$
and *WCR*: $WCR \ (rstep \ R)$
and *overlay*: $\bigwedge l \ r. (False, l, r) \notin critical\text{-pairs} \ ren \ R \ R$
shows $SN \ (rstep \ R) \longleftrightarrow SN\text{-on} \ (rstep \ R)$ (*right-forw-closure* R) (**is** $?A = ?B$)
 $\langle proof \rangle$
end
end

References

- [1] N. Dershowitz. Termination of linear rewriting systems (preliminary version). In S. Even and O. Kariv, editors, *Automata, Languages and Programming, 8th Colloquium, Acre (Akko), Israel, July 13-17, 1981. Proceedings*, volume 115 of *Lecture Notes in Computer Science*, pages 448–458. Springer, 1981.
- [2] N. Dershowitz and C. Hoot. Natural termination. *Theor. Comput. Sci.*, 142(2):179–207, 1995.
- [3] B. Gramlich. Relating innermost, weak, uniform and modular termination of term rewriting systems. In A. Voronkov, editor, *Logic Programming and Automated Reasoning, International Conference, LPAR '92, St. Petersburg, Russia, July 15-20, 1992. Proceedings*, volume 624 of *Lecture Notes in Computer Science*, pages 285–296. Springer, 1992.
- [4] R. Thiemann, D. Hofbauer, U. Le Huitouze, and J. Waldmann. New and Formalized Proofs for Right-Forward Closures and Core Matrix Interpretations. In F. Pfenning, editor, *11th International Conference on Formal Structures for Computation and Deduction (FSCD 2026)*, volume 378 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 32:1–32:19, Dagstuhl, Germany, 2026. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.