

Decreasing-Diagrams

Harald Zankl

May 26, 2024

Abstract

This theory contains a formalization of decreasing diagrams showing that any locally decreasing abstract rewrite system is confluent. We consider the valley (van Oostrom, TCS 1994) and the conversion version (van Oostrom, RTA 2008) and closely follow the original proofs. As an application we prove Newman’s lemma.

A description of this formalization is available in [3].

Contents

1	Decreasing Diagrams	1
1.1	Valley Version	1
1.1.1	Appendix	1
1.1.2	Multisets	3
1.1.3	Lexicographic maximum measure	6
1.1.4	Labeled Rewriting	9
1.1.5	Application: Newman’s Lemma	12
1.2	Conversion Version	12

1 Decreasing Diagrams

```
theory Decreasing-Diagrams imports HOL-Library.Multiset Abstract-Rewriting.Abstract-Rewriting  
begin
```

1.1 Valley Version

This section follows [1].

1.1.1 Appendix

interaction of multisets with sets

```
definition diff :: 'a multiset  $\Rightarrow$  'a set  $\Rightarrow$  'a multiset  
where diff M S = filter-mset ( $\lambda x. x \notin S$ ) M
```

definition *intersect* :: 'a multiset \Rightarrow 'a set \Rightarrow 'a multiset

where *intersect* $M\ S = \text{filter-mset } (\lambda x. x \in S)\ M$

notation

diff (infixl -s 800) and

intersect (infixl \cap s 800)

lemma *count-diff* [simp]:

$\text{count } (M -s A)\ a = \text{count } M\ a * \text{of-bool } (a \notin A)$
(proof)

lemma *set-mset-diff* [simp]:

$\text{set-mset } (M -s A) = \text{set-mset } M - A$
(proof)

lemma *diff-eq-singleton-imp*:

$M -s A = \{\#a\# \} \implies a \in (\text{set-mset } M - A)$
(proof)

lemma *count-intersect* [simp]:

$\text{count } (M \cap s A)\ a = \text{count } M\ a * \text{of-bool } (a \in A)$
(proof)

lemma *set-mset-intersect* [simp]:

$\text{set-mset } (M \cap s A) = \text{set-mset } M \cap A$
(proof)

lemma *diff-from-empty*: $\{\#\} -s S = \{\#\}$ (proof)

lemma *diff-empty*: $M -s \{\#\} = M$ (proof)

lemma *submultiset-implies-subset*: assumes $M \subseteq\# N$ shows $\text{set-mset } M \subseteq \text{set-mset } N$

(proof)

lemma *subset-implies-remove-empty*: assumes $\text{set-mset } M \subseteq S$ shows $M -s S = \{\#\}$

(proof)

lemma *remove-empty-implies-subset*: assumes $M -s S = \{\#\}$ shows $\text{set-mset } M \subseteq S$ (proof)

lemma *lemmaA-3-8*: $(M + N) -s S = (M -s S) + (N -s S)$ (proof)

lemma *lemmaA-3-9*: $(M -s S) -s T = M -s (S \cup T)$ (proof)

lemma *lemmaA-3-10*: $M = (M \cap s S) + (M -s S)$ (proof)

lemma *lemmaA-3-11*: $(M -s T) \cap s S = (M \cap s S) -s T$ (proof)

1.1.2 Multisets

Definition 2.5(1)

definition $ds :: 'a\ rel \Rightarrow 'a\ set \Rightarrow 'a\ set$
where $ds\ r\ S = \{y . \exists x \in S. (y,x) \in r\}$

definition $dm :: 'a\ rel \Rightarrow 'a\ multiset \Rightarrow 'a\ set$
where $dm\ r\ M = ds\ r\ (set\mset\ M)$

definition $dl :: 'a\ rel \Rightarrow 'a\ list \Rightarrow 'a\ set$
where $dl\ r\ \sigma = ds\ r\ (set\ \sigma)$

notation

ds (**infixl** \downarrow_s 900) **and**
 dm (**infixl** \downarrow_m 900) **and**
 dl (**infixl** \downarrow_l 900)

missing but useful

lemma $ds\text{-}ds\text{-}subsepeq\text{-}ds$: **assumes** $t: trans\ r$ **shows** $ds\ r\ (ds\ r\ S) \subseteq ds\ r\ S$ $\langle proof \rangle$

from PhD thesis of van Oostrom

lemma $ds\text{-}monotone$: **assumes** $S \subseteq T$ **shows** $ds\ r\ S \subseteq ds\ r\ T$ $\langle proof \rangle$

lemma $subset\text{-}imp\text{-}ds\text{-}subset$: **assumes** $trans\ r$ **and** $S \subseteq ds\ r\ T$ **shows** $ds\ r\ S \subseteq ds\ r\ T$
 $\langle proof \rangle$

Definition 2.5(2)

strict order (mult) is used from Multiset.thy

definition $mult\text{-}eq :: 'a\ rel \Rightarrow 'a\ multiset\ rel$ **where**
 $mult\text{-}eq\ r = (mult1\ r)^*$

definition $mul :: 'a\ rel \Rightarrow 'a\ multiset\ rel$ **where**
 $mul\ r = \{(M,N).\exists I\ J\ K. M = I + K \wedge N = I + J \wedge set\mset\ K \subseteq dm\ r\ J \wedge J \neq \{\#\}\}$

definition $mul\text{-}eq :: 'a\ rel \Rightarrow 'a\ multiset\ rel$ **where**
 $mul\text{-}eq\ r = \{(M,N).\exists I\ J\ K. M = I + K \wedge N = I + J \wedge set\mset\ K \subseteq dm\ r\ J\}$

lemma $in\text{-}mul\text{-}eqI$:

assumes $M = I + K\ N = I + J\ set\mset\ K \subseteq r\ \downarrow_m\ J$
shows $(M, N) \in mul\text{-}eq\ r$
 $\langle proof \rangle$

lemma $downset\text{-}intro$:

assumes $\forall k \in set\mset\ K. \exists j \in set\mset\ J. (k,j) \in r$ **shows** $set\mset\ K \subseteq dm\ r\ J$
 $\langle proof \rangle$

lemma *downset-elim*:

assumes *set-mset* $K \subseteq dm\ r\ J$ **shows** $\forall k \in set\text{-}mset\ K. \exists j \in set\text{-}mset\ J. (k,j) \in r$
<proof>

to closure-free representation

lemma *mult-eq-implies-one-or-zero-step*:

assumes *trans* r **and** $(M,N) \in mult\text{-}eq\ r$ **shows** $\exists I\ J\ K. N = I + J \wedge M = I + K \wedge set\text{-}mset\ K \subseteq dm\ r\ J$
<proof>

from closure-free representation

lemma *one-step-implies-mult-eq*: **assumes** *trans* r **and** *set-mset* $K \subseteq dm\ r\ J$ **shows** $(I+K, I+J) \in mult\text{-}eq\ r$
<proof>

lemma *mult-is-mul*: **assumes** *trans* r **shows** $mult\ r = mul\ r$ *<proof>*

lemma *mult-eq-is-mul-eq*: **assumes** *trans* r **shows** $mult\text{-}eq\ r = mul\text{-}eq\ r$ *<proof>*

lemma $mul\text{-}eq\ r = (mul\ r)^=$ *<proof>*

useful properties on multisets

lemma *mul-eq-reflexive*: $(M,M) \in mul\text{-}eq\ r$ *<proof>*

lemma *mul-eq-trans*: **assumes** *trans* r **and** $(M,N) \in mul\text{-}eq\ r$ **and** $(N,P) \in mul\text{-}eq\ r$ **shows** $(M,P) \in mul\text{-}eq\ r$
<proof>

lemma *mul-eq-singleton*: **assumes** $(M, \{\#\alpha\#\}) \in mul\text{-}eq\ r$ **shows** $M = \{\#\alpha\#\}$
 $\vee set\text{-}mset\ M \subseteq dm\ r\ \{\#\alpha\#\}$ *<proof>*

lemma *mul-and-mul-eq-imp-mul*: **assumes** *trans* r **and** $(M,N) \in mul\ r$ **and** $(N,P) \in mul\text{-}eq\ r$ **shows** $(M,P) \in mul\ r$
<proof>

lemma *mul-eq-and-mul-imp-mul*: **assumes** *trans* r **and** $(M,N) \in mul\text{-}eq\ r$ **and** $(N,P) \in mul\ r$ **shows** $(M,P) \in mul\ r$
<proof>

lemma *wf-mul*: **assumes** *trans* r **and** *wf* r **shows** *wf* $(mul\ r)$
<proof>

lemma *remove-is-empty-imp-mul*: **assumes** $M \text{ --}s\ dm\ r\ \{\#\alpha\#\} = \{\#\}$ **shows** $(M, \{\#\alpha\#\}) \in mul\ r$ *<proof>*

Lemma 2.6

lemma *lemma2-6-1-set*: $ds\ r\ (S \cup T) = ds\ r\ S \cup ds\ r\ T$
<proof>

lemma *lemma2-6-1-list*: $dl\ r\ (\sigma @ \tau) = dl\ r\ \sigma \cup dl\ r\ \tau$

<proof>

lemma *lemma2-6-1-multiset*: $dm\ r\ (M + N) = dm\ r\ M \cup dm\ r\ N$

<proof>

lemma *lemma2-6-1-diff*: $(dm\ r\ M) - ds\ r\ S \subseteq dm\ r\ (M -s S)$

<proof>

missing but useful

lemma *dl-monotone*: $dl\ r\ (\sigma @ \tau) \subseteq dl\ r\ (\sigma @ \tau' @ \tau)$ *<proof>*

Lemma 2.6.2

lemma *lemma2-6-2-a*: **assumes** $t: trans\ r$ **and** $M \subseteq \# N$ **shows** $(M, N) \in mul\text{-}eq\ r$ *<proof>*

lemma *mul-eq-not-equal-imp-elt*:

assumes $(M, N) \in mul\text{-}eq\ r$ **and** $y \in set\text{-}mset\ M - set\text{-}mset\ N$ **shows** $\exists z \in set\text{-}mset\ N. (y, z) \in r$ *<proof>*

lemma *lemma2-6-2-b*: **assumes** $trans\ r$ **and** $(M, N) \in mul\text{-}eq\ r$ **shows** $dm\ r\ M \subseteq dm\ r\ N$ *<proof>*

Lemma 2.6.3

lemma *ds-trans-contrapos*: **assumes** $t: trans\ r$ **and** $x \notin ds\ r\ S$ **and** $(x, y) \in r$ **shows** $y \notin ds\ r\ S$

<proof>

lemma *dm-max-elt*: **assumes** $i: irrefl\ r$ **and** $t: trans\ r$ **shows** $x \in dm\ r\ M \implies \exists y \in set\text{-}mset\ (M -s dm\ r\ M). (x, y) \in r$

<proof>

lemma *dm-subset*: **assumes** $i: irrefl\ r$ **and** $t: trans\ r$ **shows** $dm\ r\ M \subseteq dm\ r\ (M -s dm\ r\ M)$

<proof>

lemma *dm-eq*: **assumes** $i: irrefl\ r$ **and** $t: trans\ r$ **shows** $dm\ r\ M = dm\ r\ (M -s dm\ r\ M)$

<proof>

lemma *lemma2-6-3*: **assumes** $t: trans\ r$ **and** $i: irrefl\ r$ **and** $(M, N) \in mul\text{-}eq\ r$ **shows** $\exists I' J' K'. N = I' + J' \wedge M = I' + K' \wedge J' \cap \# K' = \{\#\} \wedge set\text{-}mset\ K' \subseteq dm\ r\ J'$

<proof>

Lemma 2.6.4

lemma *lemma2-6-4*: **assumes** $t: trans\ r$ **and** $N \neq \{\#\}$ **and** $set\text{-}mset\ M \subseteq dm\ r\ N$ **shows** $(M, N) \in mul\ r$ *<proof>*

lemma *lemma2-6-5-a*: **assumes** $t: \text{trans } r$ **and** $ds \ r \ S \subseteq S$ **and** $(M, N) \in \text{mul-eq } r$
shows $(M -s \ S, N -s \ S) \in \text{mul-eq } r$
 $\langle \text{proof} \rangle$

lemma *lemma2-6-5-a'*: **assumes** $t: \text{trans } r$ **and** $(M, N) \in \text{mul-eq } r$ **shows** $(M -s \ ds \ r \ S, N -s \ ds \ r \ S) \in \text{mul-eq } r$
 $\langle \text{proof} \rangle$

Lemma 2.6.6

lemma *lemma2-6-6-a*: **assumes** $t: \text{trans } r$ **and** $(M, N) \in \text{mul-eq } r$ **shows** $(Q + M, Q + N) \in \text{mul-eq } r$ $\langle \text{proof} \rangle$

lemma *add-left-one*:

assumes $\exists I \ J \ K. \text{add-mset } q \ N = I + J \wedge \text{add-mset } q \ M = I + K \wedge (J \cap \#K = \{\#\}) \wedge \text{set-mset } K \subseteq dm \ r \ J$
shows $\exists I2 \ J \ K. N = I2 + J \wedge M = I2 + K \wedge \text{set-mset } K \subseteq dm \ r \ J$ $\langle \text{proof} \rangle$

lemma *lemma2-6-6-b-one* :

assumes $\text{trans } r$ **and** $\text{irrefl } r$ **and** $(\text{add-mset } q \ M, \text{add-mset } q \ N) \in \text{mul-eq } r$
shows $(M, N) \in \text{mul-eq } r$
 $\langle \text{proof} \rangle$

lemma *lemma2-6-6-b'* : **assumes** $\text{trans } r$ **and** $i: \text{irrefl } r$ **and** $(Q + M, Q + N) \in \text{mul-eq } r$
shows $(M, N) \in \text{mul-eq } r$ $\langle \text{proof} \rangle$

lemma *lemma2-6-9*: **assumes** $t: \text{trans } r$ **and** $(M, N) \in \text{mul } r$ **shows** $(Q + M, Q + N) \in \text{mul } r$ $\langle \text{proof} \rangle$

Lemma 2.6.7

lemma *lemma2-6-7-a*: **assumes** $t: \text{trans } r$ **and** $\text{set-mset } Q \subseteq dm \ r \ N - dm \ r \ M$ **and** $(M, N) \in \text{mul-eq } r$
shows $(Q + M, N) \in \text{mul-eq } r$ $\langle \text{proof} \rangle$

missing?; similar to lemma_2.6.2?

lemma *lemma2-6-8*: **assumes** $t: \text{trans } r$ **and** $S \subseteq T$ **shows** $(M -s \ T, M -s \ S) \in \text{mul-eq } r$ $\langle \text{proof} \rangle$

1.1.3 Lexicographic maximum measure

Def 3.1: lexicographic maximum measure

fun *lexmax* :: 'a rel \Rightarrow 'a list \Rightarrow 'a multiset **where**
 $\text{lexmax } r \ [] = \{\#\}$
 $|\ \text{lexmax } r \ (\alpha \#\sigma) = \{\#\alpha\#\} + (\text{lexmax } r \ \sigma -s \ ds \ r \ \{\alpha\})$

notation

lexmax (-|-| [1000] 1000)

lemma *lexmax-singleton*: $r|\alpha| = \{\#\alpha\}$ *<proof>*

Lemma 3.2

Lemma 3.2(1)

lemma *lemma3-2-1*: **assumes** t : *trans* r **shows** $r \downarrow m r|\sigma| = r \downarrow l \sigma$ *<proof>*

Lemma 3.2(2)

lemma *lemma3-2-2*: $r|\sigma@tau| = r|\sigma| + (r|tau| -s r \downarrow l \sigma)$ *<proof>*

Definition 3.3

definition $D :: 'a \text{ rel} \Rightarrow 'a \text{ list} \Rightarrow 'a \text{ list} \Rightarrow 'a \text{ list} \Rightarrow 'a \text{ list} \Rightarrow \text{bool}$ **where**
 $D r \tau \sigma \sigma' \tau' = ((r|\sigma@tau'|, r|tau| + r|\sigma|) \in \text{mul-eq } r$
 $\wedge (r|tau@sigma'|, r|tau| + r|\sigma|) \in \text{mul-eq } r)$

lemma *D-eq*: **assumes** *trans* r **and** *irrefl* r **and** $D r \tau \sigma \sigma' \tau'$
shows $(r|tau'| -s dl r \sigma, r|tau|) \in \text{mul-eq } r$ **and** $(r|sigma'| -s dl r \tau, r|sigma|) \in \text{mul-eq } r$
<proof>

lemma *D-inv*: **assumes** *trans* r **and** *irrefl* r **and** $(r|tau'| -s dl r \sigma, r|tau|) \in \text{mul-eq } r$
and $(r|sigma'| -s dl r \tau, r|sigma|) \in \text{mul-eq } r$

shows $D r \tau \sigma \sigma' \tau'$
<proof>

lemma *D*: **assumes** *trans* r **and** *irrefl* r
shows $D r \tau \sigma \sigma' \tau' = ((r|tau'| -s dl r \sigma, r|tau|) \in \text{mul-eq } r$
 $\wedge (r|sigma'| -s dl r \tau, r|sigma|) \in \text{mul-eq } r)$
<proof>

lemma *mirror-D*: **assumes** *trans* r **and** *irrefl* r **and** $D r \tau \sigma \sigma' \tau'$ **shows** $D r \sigma$
 $\tau \tau' \sigma'$
<proof>

Proposition 3.4

definition $LD-1' :: 'a \text{ rel} \Rightarrow 'a \Rightarrow 'a \Rightarrow 'a \text{ list} \Rightarrow 'a \text{ list} \Rightarrow 'a \text{ list} \Rightarrow \text{bool}$
where $LD-1' r \beta \alpha \sigma 1 \sigma 2 \sigma 3 =$
 $(\text{set } \sigma 1 \subseteq ds r \{\beta\} \wedge \text{length } \sigma 2 \leq 1 \wedge \text{set } \sigma 2 \subseteq \{\alpha\} \wedge \text{set } \sigma 3 \subseteq ds r \{\alpha, \beta\})$

definition $LD' :: 'a \text{ rel} \Rightarrow 'a \Rightarrow 'a$
 $\Rightarrow 'a \text{ list} \Rightarrow 'a \text{ list} \Rightarrow 'a \text{ list} \Rightarrow 'a \text{ list} \Rightarrow 'a \text{ list} \Rightarrow 'a \text{ list} \Rightarrow \text{bool}$
where $LD' r \beta \alpha \sigma 1 \sigma 2 \sigma 3 \tau 1 \tau 2 \tau 3 = (LD-1' r \beta \alpha \sigma 1 \sigma 2 \sigma 3 \wedge LD-1' r \alpha$
 $\beta \tau 1 \tau 2 \tau 3)$

auxiliary properties on multisets

lemma *lexmax-le-multiset*: **assumes** t : *trans* r **shows** $r|\sigma| \subseteq\# \text{mset } \sigma$ *<proof>*

lemma split: assumes $LD-1' r \beta \alpha \sigma 1 \sigma 2 \sigma 3$ shows $\sigma 2 = [] \vee \sigma 2 = [\alpha]$
 ⟨proof⟩

lemma proposition3-4-step: assumes $trans r$ and $irrefl r$ and $LD-1' r \beta \alpha \sigma 1 \sigma 2 \sigma 3$
 shows $(r|\sigma 1 @ \sigma 2 @ \sigma 3| -s (dm r \{\#\beta\#\}), r|[\alpha]|) \in mul-eq r$ ⟨proof⟩

lemma proposition3-4:
 assumes $t: trans r$ and $i: irrefl r$ and $ld: LD' r \beta \alpha \sigma 1 \sigma 2 \sigma 3 \tau 1 \tau 2 \tau 3$
 shows $D r [\beta] [\alpha] (\sigma 1 @ \sigma 2 @ \sigma 3) (\tau 1 @ \tau 2 @ \tau 3)$
 ⟨proof⟩

lemma lexmax-decompose: assumes $\alpha \in \# r|\sigma|$ shows $\exists \sigma 1 \sigma 3. (\sigma = \sigma 1 @ [\alpha] @ \sigma 3 \wedge \alpha \notin dl r \sigma 1)$
 ⟨proof⟩

lemma lexmax-elt: assumes $trans r$ and $x \in (set \sigma)$ and $x \notin set-mset r|\sigma|$
 shows $\exists y. (x, y) \in r \wedge y \in set-mset r|\sigma|$ ⟨proof⟩

lemma lexmax-set: assumes $trans r$ and $set-mset r|\sigma| \subseteq r \downarrow_s S$ shows $set \sigma \subseteq r \downarrow_s S$ ⟨proof⟩

lemma drop-left-mult-eq:
 assumes $trans r$ and $irrefl r$ and $(N+M, M) \in mul-eq r$ shows $N = \{\#\}$ ⟨proof⟩

generalized to lists

lemma proposition3-4-inv-lists:
 assumes $t: trans r$ and $i: irrefl r$ and $k: (r|\sigma| -s r \downarrow_l \beta, \{\#\alpha\#\}) \in mul-eq r$ (is $(?M, -) \in -$)
 shows $\exists \sigma 1 \sigma 2 \sigma 3. ((\sigma = \sigma 1 @ \sigma 2 @ \sigma 3) \wedge set \sigma 1 \subseteq dl r \beta \wedge length \sigma 2 \leq 1 \wedge set \sigma 2 \subseteq \{\alpha\}) \wedge set \sigma 3 \subseteq dl r (\alpha \# \beta)$ ⟨proof⟩

lemma proposition3-4-inv-step:
 assumes $t: trans r$ and $i: irrefl r$ and $k: (r|\sigma| -s r \downarrow_l [\beta], \{\#\alpha\#\}) \in mul-eq r$ (is $(?M, -) \in -$)
 shows $\exists \sigma 1 \sigma 2 \sigma 3. ((\sigma = \sigma 1 @ \sigma 2 @ \sigma 3) \wedge LD-1' r \beta \alpha \sigma 1 \sigma 2 \sigma 3)$
 ⟨proof⟩

lemma proposition3-4-inv:
 assumes $t: trans r$ and $i: irrefl r$ and $D r [\beta] [\alpha] \sigma \tau$
 shows $\exists \sigma 1 \sigma 2 \sigma 3 \tau 1 \tau 2 \tau 3. (\sigma = \sigma 1 @ \sigma 2 @ \sigma 3 \wedge \tau = \tau 1 @ \tau 2 @ \tau 3 \wedge LD' r \beta \alpha \sigma 1 \sigma 2 \sigma 3 \tau 1 \tau 2 \tau 3)$
 ⟨proof⟩

Lemma 3.5

lemma lemma3-5-1:

assumes t : *trans* r and *irrefl* r and $D r \tau \sigma \sigma' \tau'$ and $D r v \sigma' \sigma'' v'$
shows $(\text{lexmax } r (\tau @ v @ \sigma'), \text{lexmax } r (\tau @ v) + \text{lexmax } r \sigma) \in \text{mul-eq } r$ *<proof>*

lemma claim1: **assumes** t : *trans* r and $D r \tau \sigma \sigma' \tau'$
shows $(r|\sigma@v| + ((r|v'| -s r \downarrow l (\sigma@v')) \cap s r \downarrow l \tau), r|\sigma| + r|\tau|) \in \text{mul-eq } r$ (**is** $(?F+?H, ?G) \in -$)
<proof>

lemma step3: **assumes** t : *trans* r and $D r \tau \sigma \sigma' \tau'$
shows $r \downarrow l (\sigma@v) \supseteq (r \downarrow m (r|\sigma'| + r|\tau|))$ *<proof>*

lemma claim2: **assumes** t : *trans* r and $D r \tau \sigma \sigma' \tau'$
shows $((r|v'| -s r \downarrow l (\sigma@v')) -s r \downarrow l \tau, (r|v'| -s r \downarrow l \sigma') -s r \downarrow l \tau) \in \text{mul-eq } r$
(is $(?L, ?R) \in -$)
<proof>

lemma lemma3-5-2: **assumes** *trans* r and *irrefl* r and $D r \tau \sigma \sigma' \tau'$ and $D r v \sigma' \sigma'' v'$
shows $(r|(\sigma @ \tau' @ v')|, r|\sigma| + r|(\tau@v)|) \in \text{mul-eq } r$
<proof>

lemma lemma3-5: **assumes** *trans* r and *irrefl* r and $D r \tau \sigma \sigma' \tau'$ and $D r v \sigma' \sigma'' v'$
shows $D r (\tau@v) \sigma \sigma'' (\tau'@v')$
<proof>

lemma step2: **assumes** *trans* r and $\tau \neq []$ **shows** $(M \cap s dl r \tau, \text{lexmax } r \tau) \in \text{mul } r$ *<proof>*

Lemma 3.6

lemma lemma3-6: **assumes** t : *trans* r and ne : $\tau \neq []$ and D : $D r \tau \sigma \sigma' \tau'$
shows $(r|\sigma'| + r|v|, r|\sigma| + r|\tau@v|) \in \text{mul } r$ (**is** $(?L, ?R) \in -$) *<proof>*

lemma lemma3-6-v: **assumes** *trans* r and *irrefl* r and $\sigma \neq []$ and $D r \tau \sigma \sigma' \tau'$
shows $(r|\tau'| + r|v|, r|\tau| + r|\sigma@v|) \in \text{mul } r$
<proof>

1.1.4 Labeled Rewriting

Theorem 3.7

type-synonym (a, b) *lars* = $(a \times b \times a)$ *set*

type-synonym (a, b) *seq* = $(a \times (b \times a))$ *list*

inductive-set $seq :: (a, b)$ *lars* \Rightarrow (a, b) *seq set for ars*

where $(a, []) \in seq ars$

$| (a, \alpha, b) \in ars \Rightarrow (b, ss) \in seq ars \Rightarrow (a, (\alpha, b) \# ss) \in seq ars$

definition $lst :: (a, b)$ *seq* $\Rightarrow a$

where $lst ss =$ (*if* $snd ss = []$ *then* $fst ss$ *else* $snd (last (snd ss))$)

results on seqs

lemma *seq-tail1*: **assumes** $seq: (s, x\#xs) \in seq\ lars$

shows $(snd\ x, xs) \in seq\ lars$ **and** $(s, fst\ x, snd\ x) \in lars$ **and** $lst\ (s, x\#xs) = lst\ (snd\ x, xs)$

<proof>

lemma *seq-chop*: **assumes** $(s, ss@ts) \in seq\ ars$ **shows** $(s, ss) \in seq\ ars$ $(lst\ (s, ss), ts) \in seq\ ars$ *<proof>*

lemma *seq-concat-helper*:

assumes $(s, ls) \in seq\ ars$ **and** $ss2 \in seq\ ars$ **and** $lst\ (s, ls) = fst\ ss2$

shows $(s, ls@snd\ ss2) \in seq\ ars \wedge (lst\ (s, ls@snd\ ss2) = lst\ ss2)$

<proof>

lemma *seq-concat*:

assumes $ss1 \in seq\ ars$ **and** $ss2 \in seq\ ars$ **and** $lst\ ss1 = fst\ ss2$

shows $(fst\ ss1, snd\ ss1@snd\ ss2) \in seq\ ars$ **and** $(lst\ (fst\ ss1, snd\ ss1@snd\ ss2) = lst\ ss2)$

<proof>

diagrams

definition *diagram* :: $('a, 'b)\ lars \Rightarrow ('a, 'b)\ seq \times ('a, 'b)\ seq \times ('a, 'b)\ seq \times ('a, 'b)\ seq \Rightarrow bool$

where *diagram* $ars\ d = (let\ (\tau, \sigma, \sigma', \tau') = d\ in\ \{\sigma, \tau, \sigma', \tau'\} \subseteq seq\ ars \wedge$
 $fst\ \sigma = fst\ \tau \wedge lst\ \sigma = fst\ \tau' \wedge lst\ \tau = fst\ \sigma' \wedge lst\ \sigma' = lst\ \tau')$

definition *labels* :: $('a, 'b)\ seq \Rightarrow 'b\ list$

where *labels* $ss = map\ fst\ (snd\ ss)$

definition *D2* :: $'b\ rel \Rightarrow ('a, 'b)\ seq \times ('a, 'b)\ seq \times ('a, 'b)\ seq \times ('a, 'b)\ seq \Rightarrow bool$

where *D2* $r\ d = (let\ (\tau, \sigma, \sigma', \tau') = d\ in\ D\ r\ (labels\ \tau)\ (labels\ \sigma)\ (labels\ \sigma')\ (labels\ \tau'))$

lemma *lemma3-5-d*: **assumes** *diagram* $ars\ (\tau, \sigma, \sigma', \tau')$ **and** *diagram* $ars\ (v, \sigma', \sigma'', v')$
shows *diagram* $ars\ ((fst\ \tau, snd\ \tau@snd\ v), \sigma, \sigma'', (fst\ \tau'), snd\ \tau'@snd\ v')$ *<proof>*

lemma *lemma3-5-d-v*: **assumes** *diagram* $ars\ (\tau, \sigma, \sigma', \tau')$ **and** *diagram* $ars\ (\tau', v, v', \tau'')$
shows *diagram* $ars\ (\tau, (fst\ \sigma, snd\ \sigma@snd\ v), (fst\ \sigma', snd\ \sigma'@snd\ v'), \tau'')$ *<proof>*

lemma *lemma3-5'*: **assumes** *trans* r **and** *irrefl* r **and** *D2* $r\ (\tau, \sigma, \sigma', \tau')$ **and** *D2* $r\ (v, \sigma', \sigma'', v')$

shows *D2* $r\ ((fst\ \tau, snd\ \tau@snd\ v), \sigma, \sigma'', (fst\ \tau'), snd\ \tau'@snd\ v')$

<proof>

lemma *lemma3-5'-v*: **assumes** *trans* r **and** *irrefl* r **and** *D2* $r\ (\tau, \sigma, \sigma', \tau')$ **and** *D2* $r\ (\tau', v, v', \tau'')$

shows *D2* $r\ (\tau, (fst\ \sigma, snd\ \sigma@snd\ v), (fst\ \sigma', snd\ \sigma'@snd\ v'), \tau'')$ *<proof>*

lemma trivial-diagram: assumes $\sigma \in \text{seq ars}$ shows diagram ars $(\sigma, (\text{fst } \sigma, []), (\text{lst } \sigma, []), \sigma)$
 ⟨proof⟩

lemma trivial-D2: assumes $\sigma \in \text{seq ars}$ shows $D2\ r\ (\sigma, (\text{fst } \sigma, []), (\text{lst } \sigma, []), \sigma)$
 ⟨proof⟩

definition DD :: ('a,'b) lars \Rightarrow 'b rel \Rightarrow ('a,'b) seq \times ('a,'b) seq \times ('a,'b) seq \times ('a,'b) seq \Rightarrow bool
 where $DD\ \text{ars}\ r\ d = (\text{diagram ars } d \wedge D2\ r\ d)$

lemma lemma3-5-DD: assumes trans r and irrefl r and $DD\ \text{ars}\ r\ (\tau, \sigma, \sigma', \tau')$ and $DD\ \text{ars}\ r\ (v, \sigma', \sigma'', v')$
 shows $DD\ \text{ars}\ r\ ((\text{fst } \tau, \text{snd } \tau @ \text{snd } v), \sigma, \sigma'', (\text{fst } \tau'), \text{snd } \tau' @ \text{snd } v')$
 ⟨proof⟩

lemma lemma3-5-DD-v: assumes trans r and irrefl r and $DD\ \text{ars}\ r\ (\tau, \sigma, \sigma', \tau')$ and $DD\ \text{ars}\ r\ (\tau', v, v', \tau')$
 shows $DD\ \text{ars}\ r\ (\tau, (\text{fst } \sigma, \text{snd } \sigma @ \text{snd } v), (\text{fst } \sigma', \text{snd } \sigma' @ \text{snd } v'), \tau')$
 ⟨proof⟩

lemma trivial-DD: assumes $\sigma \in \text{seq ars}$ shows $DD\ \text{ars}\ r\ (\sigma, (\text{fst } \sigma, []), (\text{lst } \sigma, []), \sigma)$
 ⟨proof⟩

lemma mirror-DD: assumes trans r and irrefl r and $DD\ \text{ars}\ r\ (\tau, \sigma, \sigma', \tau')$ shows $DD\ \text{ars}\ r\ (\sigma, \tau, \tau', \sigma')$
 ⟨proof⟩

well-foundedness of rel r

definition measure :: 'b rel \Rightarrow ('a,'b) seq \times ('a,'b) seq \Rightarrow 'b multiset
 where $\text{measure}\ r\ P = r|\text{labels } (\text{fst } P)| + r|\text{labels } (\text{snd } P)|$

definition pex :: 'b rel \Rightarrow (('a,'b) seq \times ('a,'b) seq) rel
 where $\text{pex}\ r = \{(P1, P2). (\text{measure}\ r\ P1, \text{measure}\ r\ P2) \in \text{mul}\ r\}$

lemma wfi: assumes $\text{relr} = \text{pex}\ r$ and $\neg \text{wf } (\text{relr})$ shows $\neg \text{wf } (\text{mul}\ r)$ ⟨proof⟩

lemma wf: assumes trans r and $\text{wf}\ r$ shows $\text{wf } (\text{pex}\ r)$ ⟨proof⟩

main result

definition peak :: ('a,'b) lars \Rightarrow ('a,'b) seq \times ('a,'b) seq \Rightarrow bool
 where $\text{peak ars } p = (\text{let } (\tau, \sigma) = p \text{ in } \{\tau, \sigma\} \subseteq \text{seq ars} \wedge \text{fst } \tau = \text{fst } \sigma)$

definition local-peak :: ('a,'b) lars \Rightarrow ('a,'b) seq \times ('a,'b) seq \Rightarrow bool
 where $\text{local-peak ars } p = (\text{let } (\tau, \sigma) = p \text{ in } \text{peak ars } p \wedge \text{length } (\text{snd } \tau) = 1 \wedge \text{length } (\text{snd } \sigma) = 1)$

proof of Theorem 3.7

lemma *LD-imp-D*: **assumes** *trans r and wf r and* $\forall P. (\text{local-peak ars } P \longrightarrow (\exists \sigma' \tau'. DD \text{ ars } r (fst P, snd P, \sigma', \tau')))$

and *peak ars P shows* $(\exists \sigma' \tau'. DD \text{ ars } r (fst P, snd P, \sigma', \tau'))$ *<proof>*

CR with unlabeled

definition *unlabel* :: $('a, 'b) \text{ lars} \Rightarrow 'a \text{ rel}$

where *unlabel ars* = $\{(a, c). \exists b. (a, b, c) \in \text{ars}\}$

lemma *step-imp-seq*: **assumes** $(a, b) \in (\text{unlabel ars})$

shows $\exists ss \in \text{seq ars}. fst ss = a \wedge lst ss = b$ *<proof>*

lemma *steps-imp-seq*: **assumes** $(a, b) \in (\text{unlabel ars})^*$

shows $\exists ss \in \text{seq ars}. fst ss = a \wedge lst ss = b$ *<proof>*

lemma *step-imp-unlabeled-step*: **assumes** $(a, b, c) \in \text{ars}$ **shows** $(a, c) \in (\text{unlabel ars})$

<proof>

lemma *seq-imp-steps*:

assumes $ss \in \text{seq ars}$ **and** $fst ss = a$ **and** $lst ss = b$ **shows** $(a, b) \in (\text{unlabel ars})^*$

<proof>

lemma *seq-vs-steps*: **shows** $(a, b) \in (\text{unlabel ars})^* = (\exists ss. fst ss = a \wedge lst ss = b \wedge ss \in \text{seq ars})$

<proof>

lemma *D-imp-CR*: **assumes** $\forall P. (\text{peak ars } P \longrightarrow (\exists \sigma' \tau'. DD \text{ ars } r (fst P, snd P, \sigma', \tau')))$ **shows** *CR (unlabel ars)* *<proof>*

definition *LD* :: $'b \text{ set} \Rightarrow 'a \text{ rel} \Rightarrow \text{bool}$

where *LD L ars* = $(\exists (r::('b \text{ rel})) (lrs::('a, 'b) \text{ lars}). (\text{ars} = \text{unlabel lrs}) \wedge \text{trans } r \wedge \text{wf } r \wedge (\forall P. (\text{local-peak lrs } P \longrightarrow (\exists \sigma' \tau'. (DD \text{ lrs } r (fst P, snd P, \sigma', \tau'))))))$

lemma *sound*: **assumes** *LD L ars* **shows** *CR ars*

<proof>

1.1.5 Application: Newman's Lemma

lemma *measure*:

assumes *lab-eq*: $lrs = \{(a, c, b). c = a \wedge (a, b) \in \text{ars}\}$ **and** $(s, (\alpha, t) \# ss) \in \text{seq lrs}$ **shows** $\text{set } (\text{labels } (t, ss)) \subseteq ds ((\text{ars}^+)^{-1}) \{\alpha\}$ *<proof>*

lemma *newman*: **assumes** *WCR ars* **and** *SN ars* **shows** *CR ars* *<proof>*

1.2 Conversion Version

This section follows [2].

auxiliary results on multisets

lemma *mul-eq-add-right*: $(M, M+P) \in \text{mul-eq } r \langle \text{proof} \rangle$

lemma *mul-add-right*: **assumes** $(M, N) \in \text{mul } r$ **shows** $(M, N+P) \in \text{mul } r \langle \text{proof} \rangle$

lemma *mul-eq-and-ds-imp-ds*:

assumes $t: \text{trans } r$ **and** $(M, N) \in \text{mul-eq } r$ **and** $\text{set-mset } N \subseteq \text{ds } r \ S$

shows $\text{set-mset } M \subseteq \text{ds } r \ S \langle \text{proof} \rangle$

lemma *lemma2-6-2-set*: **assumes** $S \subseteq T$ **shows** $\text{ds } r \ S \subseteq \text{ds } r \ T \langle \text{proof} \rangle$

lemma *leq-imp-subseteq*: **assumes** $M \subseteq\# N$ **shows** $\text{set-mset } M \subseteq \text{set-mset } N \langle \text{proof} \rangle$

lemma *mul-add-mul-eq-imp-mul*: **assumes** $(M, N) \in \text{mul } r$ **and** $(P, Q) \in \text{mul-eq } r$ **shows** $(M+P, N+Q) \in \text{mul } r \langle \text{proof} \rangle$

labeled conversion

type-synonym $('a, 'b) \text{ conv} = ('a \times ((\text{bool} \times 'b \times 'a) \text{ list}))$

inductive-set $\text{conv} :: ('a, 'b) \text{ lars} \Rightarrow ('a, 'b) \text{ conv set for ars}$

where $(a, []) \in \text{conv ars}$

| $(a, \alpha, b) \in \text{ars} \Longrightarrow (b, \text{ss}) \in \text{conv ars} \Longrightarrow (a, (\text{True}, \alpha, b) \# \text{ss}) \in \text{conv ars}$

| $(b, \alpha, a) \in \text{ars} \Longrightarrow (b, \text{ss}) \in \text{conv ars} \Longrightarrow (a, (\text{False}, \alpha, b) \# \text{ss}) \in \text{conv ars}$

definition $\text{labels-conv} :: ('a, 'b) \text{ conv} \Rightarrow 'b \text{ list}$

where $\text{labels-conv } c = \text{map } (\lambda q. (\text{fst } (\text{snd } q))) (\text{snd } c)$

definition $\text{measure-conv} :: 'b \text{ rel} \Rightarrow ('a, 'b) \text{ conv} \Rightarrow 'b \text{ multiset}$

where $\text{measure-conv } r \ c = \text{lexmax } r \ (\text{labels-conv } c)$

fun $\text{lst-conv} :: ('a, 'b) \text{ conv} \Rightarrow 'a$

where $\text{lst-conv } (s, []) = s$

| $\text{lst-conv } (s, (d, \alpha, t) \# \text{ss}) = \text{lst-conv } (t, \text{ss})$

definition $\text{local-diagram1} :: ('a, 'b) \text{ lars} \Rightarrow ('a, 'b) \text{ seq} \Rightarrow ('a, 'b) \text{ seq} \Rightarrow ('a, 'b) \text{ seq}$

$\Rightarrow ('a, 'b) \text{ seq} \Rightarrow ('a, 'b) \text{ seq} \Rightarrow \text{bool}$

where $\text{local-diagram1 ars } \beta \ \alpha \ \sigma 1 \ \sigma 2 \ \sigma 3 =$

$(\text{local-peak ars } (\beta, \alpha) \wedge \{\sigma 1, \sigma 2, \sigma 3\} \subseteq \text{seq ars} \wedge \text{lst } \beta = \text{fst } \sigma 1 \wedge \text{lst } \sigma 1 = \text{fst } \sigma 2 \wedge \text{lst } \sigma 2 = \text{fst } \sigma 3)$

definition $\text{LDD1} :: ('a, 'b) \text{ lars} \Rightarrow 'b \text{ rel} \Rightarrow ('a, 'b) \text{ seq} \Rightarrow ('a, 'b) \text{ seq} \Rightarrow ('a, 'b) \text{ seq}$

$\Rightarrow ('a, 'b) \text{ seq} \Rightarrow ('a, 'b) \text{ seq} \Rightarrow \text{bool}$

where $\text{LDD1 ars } r \ \beta \ \alpha \ \sigma 1 \ \sigma 2 \ \sigma 3 = (\text{local-diagram1 ars } \beta \ \alpha \ \sigma 1 \ \sigma 2 \ \sigma 3 \wedge$

$\text{LD-1}' r \ (\text{hd } (\text{labels } \beta)) \ (\text{hd } (\text{labels } \alpha)) \ (\text{labels } \sigma 1) \ (\text{labels } \sigma 2) \ (\text{labels } \sigma 3))$

definition $\text{LDD} :: ('a, 'b) \text{ lars} \Rightarrow 'b \text{ rel} \Rightarrow ('a, 'b) \text{ seq} \times ('a, 'b) \text{ seq} \times ('a, 'b) \text{ seq} \times$

$(('a, 'b) \text{ seq} \times ('a, 'b) \text{ seq} \times ('a, 'b) \text{ seq} \times ('a, 'b) \text{ seq} \times ('a, 'b) \text{ seq} \Rightarrow \text{bool}$

where $\text{LDD ars } r \ d = (\text{let } (\beta, \alpha, \sigma 1, \sigma 2, \sigma 3, \tau 1, \tau 2, \tau 3) = d \text{ in } \text{LDD1 ars } r \ \beta \ \alpha \ \sigma 1 \ \sigma 2 \ \sigma 3 \wedge \text{LDD1 ars } r \ \alpha \ \beta \ \tau 1 \ \tau 2 \ \tau 3 \wedge \text{lst } \sigma 3 = \text{lst } \tau 3)$

definition *local-triangle1* :: ('a,'b) lars \Rightarrow ('a,'b) seq \Rightarrow ('a,'b) seq \Rightarrow ('a,'b) conv \Rightarrow ('a,'b) seq \Rightarrow ('a,'b) conv \Rightarrow bool
where *local-triangle1* ars β α $\sigma 1$ $\sigma 2$ $\sigma 3$ =
(*local-peak* ars (β, α) \wedge $\sigma 2 \in$ seq ars \wedge $\{\sigma 1, \sigma 3\} \subseteq$ conv ars \wedge lst $\beta =$ fst $\sigma 1 \wedge$ lst-conv $\sigma 1 =$ fst $\sigma 2 \wedge$ lst $\sigma 2 =$ fst $\sigma 3$)

definition *LT1* :: ('a,'b) lars \Rightarrow 'b rel \Rightarrow ('a,'b) seq \Rightarrow ('a,'b) seq \Rightarrow ('a,'b) conv \Rightarrow ('a,'b) seq \Rightarrow ('a,'b) conv \Rightarrow bool
where *LT1* ars r β α $\sigma 1$ $\sigma 2$ $\sigma 3$ = (*local-triangle1* ars β α $\sigma 1$ $\sigma 2$ $\sigma 3 \wedge$ LD-1' r (hd (labels β)) (hd (labels α)) (labels-conv $\sigma 1$) (labels $\sigma 2$) (labels-conv $\sigma 3$))

definition *LT* :: ('a,'b) lars \Rightarrow 'b rel \Rightarrow ('a,'b) seq \times ('a,'b) seq \times ('a,'b) conv \times ('a,'b) seq \times ('a,'b) conv \times ('a,'b) conv \times ('a,'b) seq \times ('a,'b) conv \Rightarrow bool
where *LT* ars r t = (let ($\beta, \alpha, \sigma 1, \sigma 2, \sigma 3, \tau 1, \tau 2, \tau 3$) = t in *LT1* ars r β α $\sigma 1$ $\sigma 2$ $\sigma 3 \wedge$ *LT1* ars r α β $\tau 1$ $\tau 2$ $\tau 3 \wedge$ lst-conv $\sigma 3 =$ lst-conv $\tau 3$)

lemma *conv-tail1*: **assumes** conv: (s,(d, α ,t)#xs) \in conv ars
shows (t,xs) \in conv ars **and** d \Longrightarrow (s, α ,t) \in ars **and** \neg d \Longrightarrow (t, α ,s) \in ars **and** lst-conv (s,(d, α ,t)#xs) = lst-conv (t,xs) \langle proof \rangle

lemma *conv-chop*: **assumes** (s,ss1@ss2) \in conv ars **shows** (s,ss1) \in conv ars (lst-conv (s,ss1),ss2) \in conv ars \langle proof \rangle

lemma *conv-concat-helper*:
assumes (s,ls) \in conv ars **and** ss2 \in conv ars **and** lst-conv (s,ls) = fst ss2
shows (s,ls@snd ss2) \in conv ars \wedge (lst-conv (s,ls@snd ss2) = lst-conv ss2) \langle proof \rangle

lemma *conv-concat*:
assumes ss1 \in conv ars **and** ss2 \in conv ars **and** lst-conv ss1 = fst ss2
shows (fst ss1,snd ss1@snd ss2) \in conv ars **and** (lst-conv (fst ss1,snd ss1@snd ss2) = lst-conv ss2) \langle proof \rangle

lemma *conv-concat-labels*:
assumes ss1 \in conv ars **and** ss2 \in conv ars **and** set (labels-conv ss1) \subseteq S **and** set (labels-conv ss2) \subseteq T
shows set (labels-conv (fst ss1,snd ss1@snd ss2)) \subseteq S \cup T \langle proof \rangle

lemma *seq-decompose*:
assumes $\sigma \in$ seq ars **and** labels $\sigma = \sigma 1' @ \sigma 2'$
shows $\exists \sigma 1 \sigma 2. (\{\sigma 1, \sigma 2\} \subseteq$ seq ars $\wedge \sigma =$ (fst $\sigma 1, \text{snd } \sigma 1 @ \text{snd } \sigma 2) \wedge$ lst $\sigma 1 =$ fst $\sigma 2 \wedge$ lst $\sigma 2 =$ lst $\sigma \wedge$ labels $\sigma 1 = \sigma 1' \wedge$ labels $\sigma 2 = \sigma 2')$ \langle proof \rangle

lemma *seq-imp-conv*:
assumes (s,ss) \in seq ars
shows (s,map (λ step. (True,step)) ss) \in conv ars \wedge

$lst\text{-conv } (s, \text{map } (\lambda \text{step.}(True, \text{step})) \text{ ss}) = lst (s, \text{ss}) \wedge$
 $labels (s, \text{ss}) = labels\text{-conv } (s, \text{map } (\lambda \text{step.}(True, \text{step})) \text{ ss})$
 <proof>

fun *conv-mirror* :: ('a,'b) conv \Rightarrow ('a,'b) conv
where *conv-mirror* $\sigma = (\text{let } (s, \text{ss}) = \sigma \text{ in case ss of}$
 $\quad \square \Rightarrow (s, \text{ss})$
 $\quad | x \# xs \Rightarrow \text{let } (d, \alpha, t) = x \text{ in}$
 $\quad \quad (\text{fst } (\text{conv-mirror } (t, xs)), \text{snd } (\text{conv-mirror } (t, xs)) @ [(\neg d, \alpha, s)]))$

lemma *conv-mirror*: **assumes** $\sigma \in \text{conv ars}$
shows *conv-mirror* $\sigma \in \text{conv ars} \wedge$
 $\text{set } (labels\text{-conv } (\text{conv-mirror } \sigma)) = \text{set } (labels\text{-conv } \sigma) \wedge$
 $\text{fst } \sigma = \text{fst } (\text{conv-mirror } \sigma) \wedge$
 $lst\text{-conv } \sigma = \text{fst } (\text{conv-mirror } \sigma)$ <proof>

lemma *DD-subset-helper*:
assumes $t:\text{trans } r$ **and** $(r|\tau @ \sigma', r|\tau + r|\sigma) \in \text{mul-eq } r$ **and** $\text{set-mset } (r|\tau + r|\sigma) \subseteq ds \ r \ S$
shows $\text{set-mset } r|\sigma' \subseteq ds \ r \ S$ <proof>

lemma *DD-subset-ds*:
assumes $t:\text{trans } r$ **and** *DD*: $DD \text{ ars } r (\tau, \sigma, \sigma', \tau')$ **and** $\text{set-mset } (\text{measure } r (\tau, \sigma)) \subseteq ds \ r \ S$ **shows** $\text{set-mset } (\text{measure } r (\sigma', \tau')) \subseteq ds \ r \ S$ <proof>

lemma *conv-imp-valley*:
assumes $t:\text{trans } r$
and *IH*: $!!y . ((y, ((s, [\alpha\text{-step}] @ \varrho\text{-step}), (s, [\beta\text{-step}] @ \nu\text{-step}))) \in \text{pex } r \Longrightarrow \text{peak ars } y$
 $\Longrightarrow \exists \sigma' \tau'. DD \text{ ars } r (\text{fst } y, \text{snd } y, \sigma', \tau')$ (**is** $!!y . ((y, ?P) \in - \Longrightarrow - \Longrightarrow -)$)
and $\delta 1 \in \text{conv ars}$
and $\text{set-mset } (\text{measure-conv } r \ \delta 1) \subseteq dm \ r \ M$
and $(M, \{\#\text{fst } \alpha\text{-step}, \#\text{fst } \beta\text{-step}\}) \in \text{mul-eq } r$
shows $\exists \sigma \tau. (\{\sigma, \tau\} \subseteq \text{seq ars} \wedge \text{fst } \sigma = \text{fst } \delta 1 \wedge \text{fst } \tau = \text{fst } \delta 1 \wedge \text{fst } \sigma = \text{fst } \tau \wedge \text{set-mset } (\text{measure } r (\sigma, \tau)) \subseteq dm \ r \ M)$ <proof>

lemma *labels-multiset*: **assumes** $\text{length } (labels \ \sigma) \leq 1$ **and** $\text{set } (labels \ \sigma) \subseteq \{\alpha\}$
shows $(r|labels \ \sigma, \{\#\alpha\}) \in \text{mul-eq } r$ <proof>

lemma *decreasing-imp-local-decreasing*:
assumes $t:\text{trans } r$ **and** $i:\text{irrefl } r$ **and** *DD*: $DD \text{ ars } r (\tau, \sigma, \sigma', \tau')$ **and** $\text{set } (labels \ \tau) \subseteq ds \ r \ \{\beta\}$
and $\text{length } (labels \ \sigma) \leq 1$ **and** $\text{set } (labels \ \sigma) \subseteq \{\alpha\}$
shows $\exists \sigma 1 \ \sigma 2 \ \sigma 3. (\sigma' = (\text{fst } \sigma 1, \text{snd } \sigma 1 @ \text{snd } \sigma 2 @ \text{snd } \sigma 3) \wedge \text{fst } \sigma 1 = \text{fst } \sigma 2 \wedge \text{fst } \sigma 2 = \text{fst } \sigma 3 \wedge \text{fst } \sigma 3 = \text{fst } \sigma'$
 $\quad \wedge LD\text{-}1' \ r \ \beta \ \alpha \ (labels \ \sigma 1) \ (labels \ \sigma 2) \ (labels \ \sigma 3))$
 $\text{set } (labels \ \tau') \subseteq ds \ r \ (\{\alpha, \beta\})$
 <proof>

lemma *local-decreasing-extended-imp-decreasing*:

assumes *LT1 ars r* (*s*,[*β-step*]) (*s*,[*α-step*]) *γ1 γ2 γ3*
and *t:trans r* **and** *i:irrefl r*
and *IH:!!y . ((y,((s,[β-step]@v-step),(s,[α-step]@ρ-step))) ∈ pex r ⇒ peak ars y*
 $\implies \exists \sigma' \tau'. DD \text{ ars } r \text{ (fst } y, \text{snd } y, \sigma', \tau')$ (**is** !!*y*. ((*y*,?P) ∈ - \implies - \implies -))
shows $\exists \sigma 1 \sigma 2 \sigma 3' \gamma 1'''$. ($\{\sigma 1, \sigma 2, \sigma 3', \gamma 1'''\}$ \subseteq seq ars \wedge
 set (labels $\sigma 1$) \subseteq ds r {fst *β-step*} \wedge length (labels $\sigma 2$) ≤ 1 \wedge set (labels $\sigma 2$) \subseteq
 {fst *α-step*} \wedge set (labels $\sigma 3'$) \subseteq ds r {fst *α-step*,fst *β-step*} \wedge
 set (labels $\gamma 1'''$) \subseteq ds r {fst *α-step*,fst *β-step*} \wedge
 snd *β-step* = fst $\sigma 1$ \wedge lst $\sigma 1$ = fst $\sigma 2$ \wedge lst $\sigma 2$ = fst $\sigma 3'$ \wedge lst $\sigma 3'$ = lst $\gamma 1'''$
 \wedge fst $\gamma 1'''$ = fst $\gamma 3$
 ⟨proof⟩

lemma *LDD-imp-DD*:

assumes *t:trans r* **and** *i:irrefl r* **and** *LDD ars r* (*τ,σ,σ1,σ2,σ3,τ1,τ2,τ3*)
shows $\exists \sigma' \tau'. DD \text{ ars } r \text{ (}\tau, \sigma, \sigma', \tau')$ ⟨proof⟩

lemma *LT-imp-DD*:

assumes *t:trans r*
and *i:irrefl r*
and *IH:!!y . ((y,((s,[β-step]@v-step),(s,[α-step]@ρ-step))) ∈ pex r ⇒ peak ars y*
 $\implies \exists \sigma' \tau'. DD \text{ ars } r \text{ (fst } y, \text{snd } y, \sigma', \tau')$ (**is** !!*y*. ((*y*,?P) ∈ - \implies - \implies -))
and *LT: LT ars r* ((*s*,[*β-step*]),(*s*,[*α-step*]),*γ1,γ2,γ3,δ1,δ2,δ3*)
shows $\exists \kappa \mu. DD \text{ ars } r \text{ ((s,[β-step]),(s,[α-step]),\kappa,\mu)}$
 ⟨proof⟩

lemma *LT-imp-D*: **assumes** *t:trans r* **and** *wf r* **and** $\forall p. (\text{local-peak ars } p \longrightarrow (\exists$
 $\gamma 1 \gamma 2 \gamma 3 \delta 1 \delta 2 \delta 3. LT \text{ ars } r \text{ (fst } p, \text{snd } p, \gamma 1, \gamma 2, \gamma 3, \delta 1, \delta 2, \delta 3)))$
and peak ars *P* **shows** ($\exists \sigma' \tau'. DD \text{ ars } r \text{ (fst } P, \text{snd } P, \sigma', \tau')$) ⟨proof⟩

definition *LD-conv* :: 'b set \Rightarrow 'a rel \Rightarrow bool

where *LD-conv L ars* = ($\exists (r::('b \text{ rel})) (lrs::('a, 'b) \text{ lars}). (\text{ars} = \text{unlabel lrs}) \wedge$
trans r \wedge *wf r* \wedge ($\forall p. (\text{local-peak lrs } p \longrightarrow (\exists \gamma 1 \gamma 2 \gamma 3 \delta 1 \delta 2 \delta 3. LT \text{ lrs } r \text{ (fst}$
 $p, \text{snd } p, \gamma 1, \gamma 2, \gamma 3, \delta 1, \delta 2, \delta 3))))$)

lemma *sound-conv*: **assumes** *LD-conv L ars* **shows** *CR ars*
 ⟨proof⟩

hide-const (open) *D*

hide-const (open) *seq*

hide-const (open) *measure*

hide-fact (open) *split*

end

References

- [1] V. van Oostrom. Confluence by decreasing diagrams. *Theoretical Computer Science*, 126(2):259–280, 1994.
- [2] V. van Oostrom. Confluence by decreasing diagrams – converted. In *Proc. 19th International Conference on Rewriting Techniques and Applications*, volume 5117 of *Lecture Notes in Computer Science*, pages 306–320, 2008.
- [3] H. Zankl. Confluence by decreasing diagrams – formalized. In *Proc. 24th International Conference on Rewriting Techniques and Applications*, number 21 in *Leibniz International Proceedings in Informatics*, pages 352–367, 2013.