Minimal, Maximal, Least, and Greatest Elements w.r.t. Restricted Ordering

Martin Desharnais

October 30, 2024

Abstract

This entry provides small, reusable, theories that specify the concepts of minimal, maximal, least, and greatest elements in sets, final sets, and final multisets. The concepts are uniformly specified as predicates parametrized by a binary relation. The binary relation is only required to be an ordering on the elements of the concrete collection considered. This is useful when working with a partial ordering, but some assumption or invariant proves that the ordering is total on all elements of the considered set.

Contents

1 Definitions **1**

2 Conversions **2**

3 Existence **3**

4 Uniqueness **3**

5 Existence of unique element **4**

6 Transformations **4**

7 Minimal and maximal elements **5**
- 7.1 Conversions . . . 5
- 7.2 Existence . . . 5
- 7.3 Miscellaneous . . . 5

8 Least and greatest elements **6**
- 8.1 Conversions . . . 6
- 8.2 Uniqueness . . . 7
- 8.3 Existence . . . 7

9 Hide stuff **7**

10 Integration in type classes **7**

11 Minimal and maximal elements **9**
11.1 Conversions 9
11.2 Existence 9
11.3 Non-existence 10
11.4 Miscellaneous 10

12 Least and greatest elements **10**
12.1 Conversions 10
12.2 Uniqueness 11
12.3 Existence 11
12.4 Nonexistence 11
12.5 Miscellaneous 12

13 Hide stuff **12**

14 Integration in type classes **12**

15 Minimal and maximal elements **14**
15.1 Conversions 15
15.2 Existence 15
15.3 Miscellaneous 16
15.4 Nonuniqueness 16

16 Least and greatest elements **17**
16.1 Conversions 17
16.2 Uniqueness 18
16.3 Miscellaneous 18

17 Examples of duplicate handling in set and multiset definitions **20**

18 Hide stuff **21**

19 Integration in type classes **21**

theory *Relation-Reachability*
imports *Main*
begin

1 Definitions

When a binary relation hold for two values, i.e., $R\ x\ y$, we say that x reaches y and, conversely, that y is reachable by x.

definition *non-reachable-wrt* **where**
non-reachable-wrt $R\ X\ x \longleftrightarrow x \in X \wedge (\forall\, y \in X - \{x\}.\ \neg\ (R\ y\ x))$

definition *non-reaching-wrt* **where**
non-reaching-wrt $R\ X\ x \longleftrightarrow x \in X \wedge (\forall\, y \in X - \{x\}.\ \neg\ (R\ x\ y))$

definition *reaching-all-wrt* **where**
reaching-all-wrt $R\ X\ x \longleftrightarrow x \in X \wedge (\forall\, y \in X - \{x\}.\ R\ x\ y)$

definition *reachable-by-all-wrt* **where**
reachable-by-all-wrt $R\ X\ x \longleftrightarrow x \in X \wedge (\forall\, y \in X - \{x\}.\ R\ y\ x)$

2 Conversions

lemma *non-reachable-wrt-iff*:
non-reachable-wrt $R\ X\ x \longleftrightarrow x \in X \wedge (\forall\, y \in X.\ y \neq x \longrightarrow \neg\ R\ y\ x)$
⟨*proof*⟩

lemma *non-reaching-wrt-iff*:
non-reaching-wrt $R\ X\ x \longleftrightarrow x \in X \wedge (\forall\, y \in X.\ y \neq x \longrightarrow \neg\ R\ x\ y)$
⟨*proof*⟩

lemma *reaching-all-wrt-iff*:
reaching-all-wrt $R\ X\ x \longleftrightarrow x \in X \wedge (\forall\, y \in X.\ y \neq x \longrightarrow R\ x\ y)$
⟨*proof*⟩

lemma *reachable-by-all-wrt-iff*:
reachable-by-all-wrt $R\ X\ x \longleftrightarrow x \in X \wedge (\forall\, y \in X.\ y \neq x \longrightarrow R\ y\ x)$
⟨*proof*⟩

lemma *non-reachable-wrt-filter-iff*:
non-reachable-wrt $R\ \{y \in X.\ P\ y\}\ x \longleftrightarrow x \in X \wedge P\ x \wedge (\forall\, y \in X - \{x\}.\ P\ y \longrightarrow \neg\ R\ y\ x)$
⟨*proof*⟩

lemma *non-reachable-wrt-conversep*[*simp*]:
non-reachable-wrt $R^{-1-1} =$ *non-reaching-wrt* R
⟨*proof*⟩

lemma *non-reaching-wrt-conversep*[*simp*]:
non-reaching-wrt $R^{-1-1} =$ *non-reachable-wrt* R
⟨*proof*⟩

lemma *reaching-all-wrt-conversep*[*simp*]:
reaching-all-wrt $R^{-1-1} =$ *reachable-by-all-wrt* R
⟨*proof*⟩

lemma *reachable-by-all-wrt-conversep*[*simp*]:
reachable-by-all-wrt $R^{-1-1} =$ *reaching-all-wrt* R

⟨*proof*⟩

lemma *non-reachable-wrt-eq-reaching-all-wrt*:
assumes *asym*: *asymp-on X R* **and** *tot*: *totalp-on X R*
shows *non-reachable-wrt R X* = *reaching-all-wrt R X*
⟨*proof*⟩

lemma *non-reaching-wrt-eq-reachable-by-all-wrt*:
assumes *asym*: *asymp-on X R* **and** *tot*: *totalp-on X R*
shows *non-reaching-wrt R X* = *reachable-by-all-wrt R X*
⟨*proof*⟩

lemma *non-reachable-wrt-reflclp*[*simp*]:
non-reachable-wrt $R^{==}$ = *non-reachable-wrt R*
⟨*proof*⟩

lemma *non-reaching-wrt-reflclp*[*simp*]:
non-reaching-wrt $R^{==}$ = *non-reaching-wrt R*
⟨*proof*⟩

lemma *reaching-all-wrt-reflclp*[*simp*]:
reaching-all-wrt $R^{==}$ = *reaching-all-wrt R*
⟨*proof*⟩

lemma *reachable-by-all-wrt-reflclp*[*simp*]:
reachable-by-all-wrt $R^{==}$ = *reachable-by-all-wrt R*
⟨*proof*⟩

3 Existence

lemma *ex-non-reachable-wrt*:
transp-on A R ⟹ *asymp-on A R* ⟹ *finite A* ⟹ $A \neq \{\}$ ⟹ ∃ *m*. *non-reachable-wrt R A m*
⟨*proof*⟩

lemma *ex-non-reaching-wrt*:
transp-on A R ⟹ *asymp-on A R* ⟹ *finite A* ⟹ $A \neq \{\}$ ⟹ ∃ *m*. *non-reaching-wrt R A m*
⟨*proof*⟩

lemma *ex-reaching-all-wrt*:
transp-on A R ⟹ *totalp-on A R* ⟹ *finite A* ⟹ $A \neq \{\}$ ⟹ ∃ *g*. *reaching-all-wrt R A g*
⟨*proof*⟩

lemma *ex-reachable-by-all-wrt*:
transp-on A R ⟹ *totalp-on A R* ⟹ *finite A* ⟹ $A \neq \{\}$ ⟹ ∃ *g*. *reachable-by-all-wrt R A g*

⟨*proof*⟩

lemma *not-ex-greatest-element-doubleton-if*:
assumes $x \neq y$ **and** $\neg\ R\ x\ y$ **and** $\neg\ R\ y\ x$
shows $\nexists g.$ *reachable-by-all-wrt* R $\{x, y\}$ g
⟨*proof*⟩

4 Uniqueness

lemma *Uniq-non-reachable-wrt*:
totalp-on X $R \Longrightarrow \exists_{\leq 1} x.$ *non-reachable-wrt* R X x
⟨*proof*⟩

lemma *Uniq-non-reaching-wrt*:
totalp-on X $R \Longrightarrow \exists_{\leq 1} x.$ *non-reaching-wrt* R X x
⟨*proof*⟩

lemma *Uniq-reaching-all-wrt*:
asymp-on X $R \Longrightarrow \exists_{\leq 1} x.$ *reaching-all-wrt* R X x
⟨*proof*⟩

lemma *Uniq-reachable-by-all-wrt*:
asymp-on X $R \Longrightarrow \exists_{\leq 1} x.$ *reachable-by-all-wrt* R X x
⟨*proof*⟩

5 Existence of unique element

lemma *ex1-reaching-all-wrt*:
transp-on X $R \Longrightarrow$ *asymp-on* X $R \Longrightarrow$ *totalp-on* X $R \Longrightarrow$ *finite* $X \Longrightarrow X \neq \{\} \Longrightarrow$
$\exists!x.$ *reaching-all-wrt* R X x
⟨*proof*⟩

lemma *ex1-reachable-by-all-wrt*:
transp-on X $R \Longrightarrow$ *asymp-on* X $R \Longrightarrow$ *totalp-on* X $R \Longrightarrow$ *finite* $X \Longrightarrow X \neq \{\} \Longrightarrow$
$\exists!x.$ *reachable-by-all-wrt* R X x
⟨*proof*⟩

6 Transformations

lemma *non-reachable-wrt-insert-wrtI*:
assumes
trans: *transp-on* (*insert* y X) R **and**
asym: *asymp-on* (*insert* y X) R **and**
R y x **and**
x-non-reachable: *non-reachable-wrt* R X x
shows *non-reachable-wrt* R (*insert* y X) y

⟨*proof*⟩

end
theory *Min-Max-Least-Greatest-Set*
imports
Relation-Reachability
begin

7 Minimal and maximal elements

If the binary relation is a strict partial order, then non-reachability corresponds to minimality and non-reaching correspond to maximality.

definition *is-minimal-in-set-wrt* :: $('a \Rightarrow 'a \Rightarrow bool) \Rightarrow 'a\ set \Rightarrow 'a \Rightarrow bool$ **where**
$transp\text{-}on\ X\ R \Longrightarrow asymp\text{-}on\ X\ R \Longrightarrow is\text{-}minimal\text{-}in\text{-}set\text{-}wrt\ R\ X = non\text{-}reachable\text{-}wrt\ R\ X$

definition *is-maximal-in-set-wrt* :: $('a \Rightarrow 'a \Rightarrow bool) \Rightarrow 'a\ set \Rightarrow 'a \Rightarrow bool$ **where**
$transp\text{-}on\ X\ R \Longrightarrow asymp\text{-}on\ X\ R \Longrightarrow is\text{-}maximal\text{-}in\text{-}set\text{-}wrt\ R\ X = non\text{-}reaching\text{-}wrt\ R\ X$

context
fixes $X\ R$
assumes
trans: *transp-on* $X\ R$ **and**
asym: *asymp-on* $X\ R$
begin

7.1 Conversions

lemma *is-minimal-in-set-wrt-iff*:
$is\text{-}minimal\text{-}in\text{-}set\text{-}wrt\ R\ X\ x \longleftrightarrow x \in X \wedge (\forall\, y \in X.\ y \neq x \longrightarrow \neg\ R\ y\ x)$
⟨*proof*⟩

lemma *is-maximal-in-set-wrt-iff*:
$is\text{-}maximal\text{-}in\text{-}set\text{-}wrt\ R\ X\ x \longleftrightarrow x \in X \wedge (\forall\, y \in X.\ y \neq x \longrightarrow \neg\ R\ x\ y)$
⟨*proof*⟩

7.2 Existence

lemma *ex-minimal-in-set-wrt*:
$finite\ X \Longrightarrow X \neq \{\} \Longrightarrow \exists\, x.\ is\text{-}minimal\text{-}in\text{-}set\text{-}wrt\ R\ X\ x$
⟨*proof*⟩

lemma *ex-maximal-in-set-wrt*:
$finite\ X \Longrightarrow X \neq \{\} \Longrightarrow \exists\, m.\ is\text{-}maximal\text{-}in\text{-}set\text{-}wrt\ R\ X\ m$
⟨*proof*⟩

end

7.3 Miscellaneous

lemma *is-minimal-in-set-wrt-filter-iff*:
fixes X R
assumes *trans*: *transp-on* $\{y \in X.\ P\ y\}$ R **and** *asym*: *asymp-on* $\{y \in X.\ P\ y\}$ R
shows *is-minimal-in-set-wrt* R $\{y \in X.\ P\ y\}$ $x \longleftrightarrow x \in X \wedge P\ x \wedge (\forall y \in X - \{x\}.\ P\ y \longrightarrow \neg\ R\ y\ x)$
⟨*proof*⟩

lemma *is-minimal-in-set-wrt-insertI*:
assumes
trans: *transp-on* (*insert* y X) R **and**
asym: *asymp-on* (*insert* y X) R **and**
R y x **and**
x-min: *is-minimal-in-set-wrt* R X x
shows *is-minimal-in-set-wrt* R (*insert* y X) y
⟨*proof*⟩

8 Least and greatest elements

If the binary relation is a strict total ordering, then an element reaching all others is a least element and an element reachable by all others is a greatest element.

definition *is-least-in-set-wrt* :: $('a \Rightarrow 'a \Rightarrow bool) \Rightarrow 'a\ set \Rightarrow 'a \Rightarrow bool$ **where**
transp-on X $R \Longrightarrow$ *asymp-on* X $R \Longrightarrow$ *totalp-on* X $R \Longrightarrow$
is-least-in-set-wrt R X = *reaching-all-wrt* R X

definition *is-greatest-in-set-wrt* :: $('a \Rightarrow 'a \Rightarrow bool) \Rightarrow 'a\ set \Rightarrow 'a \Rightarrow bool$ **where**
transp-on X $R \Longrightarrow$ *asymp-on* X $R \Longrightarrow$ *totalp-on* X $R \Longrightarrow$
is-greatest-in-set-wrt R X = *reachable-by-all-wrt* R X

context
fixes X R
assumes
trans: *transp-on* X R **and**
asym: *asymp-on* X R **and**
tot: *totalp-on* X R
begin

8.1 Conversions

lemma *is-least-in-set-wrt-iff*:
is-least-in-set-wrt R X $x \longleftrightarrow x \in X \wedge (\forall y \in X.\ y \neq x \longrightarrow R\ x\ y)$
⟨*proof*⟩

lemma *is-greatest-in-set-wrt-iff*:
is-greatest-in-set-wrt R X $x \longleftrightarrow x \in X \wedge (\forall y \in X.\ y \neq x \longrightarrow R\ y\ x)$

⟨*proof*⟩

lemma *is-minimal-in-set-wrt-eq-is-least-in-set-wrt*:
is-minimal-in-set-wrt R X = *is-least-in-set-wrt R X*
⟨*proof*⟩

lemma *is-maximal-in-set-wrt-eq-is-greatest-in-set-wrt*:
is-maximal-in-set-wrt R X = *is-greatest-in-set-wrt R X*
⟨*proof*⟩

8.2 Uniqueness

lemma *Uniq-is-least-in-set-wrt*:
$\exists_{\leq 1} x$. *is-least-in-set-wrt R X x*
⟨*proof*⟩

lemma *Uniq-is-greatest-in-set-wrt*:
$\exists_{\leq 1} x$. *is-greatest-in-set-wrt R X x*
⟨*proof*⟩

8.3 Existence

lemma *ex-least-in-set-wrt*:
finite X $\Longrightarrow$ $X \neq \{\}$ $\Longrightarrow$ $\exists\, x$. *is-least-in-set-wrt R X x*
⟨*proof*⟩

lemma *ex-greatest-in-set-wrt*:
finite X $\Longrightarrow$ $X \neq \{\}$ $\Longrightarrow$ $\exists\, x$. *is-greatest-in-set-wrt R X x*
⟨*proof*⟩

lemma *ex1-least-in-set-wrt*:
finite X $\Longrightarrow$ $X \neq \{\}$ $\Longrightarrow$ $\exists!\, x$. *is-least-in-set-wrt R X x*
⟨*proof*⟩

lemma *ex1-greatest-in-set-wrt*:
finite X $\Longrightarrow$ $X \neq \{\}$ $\Longrightarrow$ $\exists!\, x$. *is-greatest-in-set-wrt R X x*
⟨*proof*⟩

end

9 Hide stuff

We restrict the public interface to ease future internal changes.

hide-fact *is-minimal-in-set-wrt-def is-maximal-in-set-wrt-def*
hide-fact *is-least-in-set-wrt-def is-greatest-in-set-wrt-def*

10 Integration in type classes

abbreviation (**in** *order*) *is-minimal-in-set* **where**
is-minimal-in-set ≡ *is-minimal-in-set-wrt* (<)

abbreviation (**in** *order*) *is-maximal-in-set* **where**
is-maximal-in-set ≡ *is-maximal-in-set-wrt* (<)

lemmas (**in** *order*) *is-minimal-in-set-iff* =
is-minimal-in-set-wrt-iff[*OF transp-on-less asymp-on-less*]

lemmas (**in** *order*) *is-maximal-in-set-iff* =
is-maximal-in-set-wrt-iff[*OF transp-on-less asymp-on-less*]

lemmas (**in** *order*) *ex-minimal-in-set* =
ex-minimal-in-set-wrt[*OF transp-on-less asymp-on-less*]

lemmas (**in** *order*) *ex-maximal-in-set* =
ex-maximal-in-set-wrt[*OF transp-on-less asymp-on-less*]

lemmas (**in** *order*) *is-minimal-in-set-filter-iff* =
is-minimal-in-set-wrt-filter-iff[*OF transp-on-less asymp-on-less*]

abbreviation (**in** *linorder*) *is-least-in-set* **where**
is-least-in-set ≡ *is-least-in-set-wrt* (<)

abbreviation (**in** *linorder*) *is-greatest-in-set* **where**
is-greatest-in-set ≡ *is-greatest-in-set-wrt* (<)

lemmas (**in** *linorder*) *is-least-in-set-iff* =
is-least-in-set-wrt-iff[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *is-greatest-in-set-iff* =
is-greatest-in-set-wrt-iff[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *is-minimal-in-set-eq-is-least-in-set* =
is-minimal-in-set-wrt-eq-is-least-in-set-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *is-maximal-in-set-eq-is-greatest-in-set* =
is-maximal-in-set-wrt-eq-is-greatest-in-set-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *Uniq-is-least-in-set*[*intro*] =
Uniq-is-least-in-set-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *Uniq-is-greatest-in-set*[*intro*] =
Uniq-is-greatest-in-set-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *ex-least-in-set* =
ex-least-in-set-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *ex-greatest-in-set* =
ex-greatest-in-set-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *ex1-least-in-set* =
ex1-least-in-set-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *ex1-greatest-in-set* =
ex1-greatest-in-set-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

end
theory *Min-Max-Least-Greatest-FSet*
imports
Min-Max-Least-Greatest-Set
HOL−Library.FSet
begin

11 Minimal and maximal elements

definition *is-minimal-in-fset-wrt* :: $('a \Rightarrow 'a \Rightarrow bool) \Rightarrow 'a\ fset \Rightarrow 'a \Rightarrow bool$ **where**
transp-on (*fset X*) *R* $\Longrightarrow$ *asymp-on* (*fset X*) *R* $\Longrightarrow$
is-minimal-in-fset-wrt R X = *is-minimal-in-set-wrt R* (*fset X*)

definition *is-maximal-in-fset-wrt* :: $('a \Rightarrow 'a \Rightarrow bool) \Rightarrow 'a\ fset \Rightarrow 'a \Rightarrow bool$ **where**
transp-on (*fset X*) *R* $\Longrightarrow$ *asymp-on* (*fset X*) *R* $\Longrightarrow$
is-maximal-in-fset-wrt R X = *is-maximal-in-set-wrt R* (*fset X*)

context
fixes *X R*
assumes
trans: *transp-on* (*fset X*) *R* **and**
asym: *asymp-on* (*fset X*) *R*
begin

11.1 Conversions

lemma *is-minimal-in-fset-wrt-iff*:
is-minimal-in-fset-wrt R X x $\longleftrightarrow$ *x* $|\in|$ *X* $\wedge$ *fBall X* ($\lambda y.\ y \neq x \longrightarrow \neg\ R\ y\ x$)
⟨*proof*⟩

lemma *is-maximal-in-fset-wrt-iff*:
is-maximal-in-fset-wrt R X x $\longleftrightarrow$ *x* $|\in|$ *X* $\wedge$ *fBall X* ($\lambda y.\ y \neq x \longrightarrow \neg\ R\ x\ y$)
⟨*proof*⟩

11.2 Existence

lemma *ex-minimal-in-fset-wrt*:
$X \neq \{||\} \Longrightarrow \exists\, m.$ *is-minimal-in-fset-wrt* $R\ X\ m$
⟨*proof*⟩

lemma *ex-maximal-in-fset-wrt*:
$X \neq \{||\} \Longrightarrow \exists\, m.$ *is-maximal-in-fset-wrt* $R\ X\ m$
⟨*proof*⟩

end

11.3 Non-existence

lemma *not-is-minimal-in-fset-wrt-fempty*[*simp*]: $\bigwedge R\ x. \neg$ *is-minimal-in-fset-wrt* R $\{||\}\ x$
⟨*proof*⟩

lemma *not-is-maximal-in-fset-wrt-fempty*[*simp*]: $\bigwedge R\ x. \neg$ *is-maximal-in-fset-wrt* $R\ \{||\}\ x$
⟨*proof*⟩

11.4 Miscellaneous

lemma *is-minimal-in-fset-wrt-ffilter-iff*:
assumes
tran: *transp-on* (*fset* (*ffilter* $P\ X$)) R **and**
asym: *asymp-on* (*fset* (*ffilter* $P\ X$)) R
shows *is-minimal-in-fset-wrt* R (*ffilter* $P\ X$) $x \longleftrightarrow$
$(x\ |\in|\ X \wedge P\ x \wedge$ *fBall* $(X - \{|x|\})\ (\lambda y.\ P\ y \longrightarrow \neg\ R\ y\ x))$
⟨*proof*⟩

lemma *is-minimal-in-fset-wrt-finsertI*:
assumes *trans*: *transp-on* (*fset* (*finsert* $y\ X$)) R **and** *asym*: *asymp-on* (*fset* (*finsert* $y\ X$)) R
shows $R\ y\ x \Longrightarrow$ *is-minimal-in-fset-wrt* $R\ X\ x \Longrightarrow$ *is-minimal-in-fset-wrt* R (*finsert* $y\ X$) y
⟨*proof*⟩

12 Least and greatest elements

definition *is-least-in-fset-wrt* :: $('a \Rightarrow 'a \Rightarrow bool) \Rightarrow 'a\ fset \Rightarrow 'a \Rightarrow bool$ **where**
transp-on (*fset* X) $R \Longrightarrow$ *asymp-on* (*fset* X) $R \Longrightarrow$ *totalp-on* (*fset* X) $R \Longrightarrow$
is-least-in-fset-wrt $R\ X =$ *is-least-in-set-wrt* R (*fset* X)

definition *is-greatest-in-fset-wrt* :: $('a \Rightarrow 'a \Rightarrow bool) \Rightarrow 'a\ fset \Rightarrow 'a \Rightarrow bool$ **where**
transp-on (*fset* X) $R \Longrightarrow$ *asymp-on* (*fset* X) $R \Longrightarrow$ *totalp-on* (*fset* X) $R \Longrightarrow$
is-greatest-in-fset-wrt $R\ X =$ *is-greatest-in-set-wrt* R (*fset* X)

context

fixes X R
assumes
$trans$: $transp\text{-}on\ (fset\ X)\ R$ **and**
$asym$: $asymp\text{-}on\ (fset\ X)\ R$ **and**
tot: $totalp\text{-}on\ (fset\ X)\ R$
begin

12.1 Conversions

lemma *is-least-in-fset-wrt-iff*:
$is\text{-}least\text{-}in\text{-}fset\text{-}wrt\ R\ X\ x \longleftrightarrow x \mid\in\mid X \wedge fBall\ X\ (\lambda y.\ y \neq x \longrightarrow R\ x\ y)$
⟨*proof*⟩

lemma *is-greatest-in-fset-wrt-iff*:
$is\text{-}greatest\text{-}in\text{-}fset\text{-}wrt\ R\ X\ x \longleftrightarrow x \mid\in\mid X \wedge fBall\ X\ (\lambda y.\ y \neq x \longrightarrow R\ y\ x)$
⟨*proof*⟩

lemma *is-minimal-in-fset-wrt-eq-is-least-in-fset-wrt*:
$is\text{-}minimal\text{-}in\text{-}fset\text{-}wrt\ R\ X = is\text{-}least\text{-}in\text{-}fset\text{-}wrt\ R\ X$
⟨*proof*⟩

lemma *is-maximal-in-fset-wrt-eq-is-greatest-in-fset-wrt*:
$is\text{-}maximal\text{-}in\text{-}fset\text{-}wrt\ R\ X = is\text{-}greatest\text{-}in\text{-}fset\text{-}wrt\ R\ X$
⟨*proof*⟩

12.2 Uniqueness

lemma *Uniq-is-least-in-fset-wrt*[*intro*]:
$\exists_{\leq 1} x.\ is\text{-}least\text{-}in\text{-}fset\text{-}wrt\ R\ X\ x$
⟨*proof*⟩

lemma *Uniq-is-greatest-in-fset-wrt*[*intro*]:
$\exists_{\leq 1} x.\ is\text{-}greatest\text{-}in\text{-}fset\text{-}wrt\ R\ X\ x$
⟨*proof*⟩

12.3 Existence

lemma *ex-least-in-fset-wrt*:
$X \neq \{||\} \Longrightarrow \exists\, x.\ is\text{-}least\text{-}in\text{-}fset\text{-}wrt\ R\ X\ x$
⟨*proof*⟩

lemma *ex-greatest-in-fset-wrt*:
$X \neq \{||\} \Longrightarrow \exists\, x.\ is\text{-}greatest\text{-}in\text{-}fset\text{-}wrt\ R\ X\ x$
⟨*proof*⟩

lemma *ex1-least-in-fset-wrt*:
$X \neq \{||\} \Longrightarrow \exists !x.\ is\text{-}least\text{-}in\text{-}fset\text{-}wrt\ R\ X\ x$
⟨*proof*⟩

lemma *ex1-greatest-in-fset-wrt*:

$X \neq \{||\} \Longrightarrow \exists!x.\ is\text{-}greatest\text{-}in\text{-}fset\text{-}wrt\ R\ X\ x$
⟨*proof*⟩

end

12.4 Nonexistence

lemma *not-is-least-in-fset-wrt-fempty*[*simp*]: $\bigwedge R\ x.\ \neg\ is\text{-}least\text{-}in\text{-}fset\text{-}wrt\ R\ \{||\}\ x$
⟨*proof*⟩

lemma *not-is-greatest-in-fset-wrt-fempty*[*simp*]: $\bigwedge R\ x.\ \neg\ is\text{-}greatest\text{-}in\text{-}fset\text{-}wrt\ R\ \{||\}\ x$
⟨*proof*⟩

12.5 Miscellaneous

lemma *is-least-in-ffilter-wrt-iff*:
assumes
trans: *transp-on* (*fset* (*ffilter P X*)) *R* **and**
asym: *asymp-on* (*fset* (*ffilter P X*)) *R* **and**
tot: *totalp-on* (*fset* (*ffilter P X*)) *R*
shows *is-least-in-fset-wrt R* (*ffilter P X*) *x* ⟷
(x |∈| $X \wedge P\ x \wedge$ *fBall X* ($\lambda y.\ y \neq x \longrightarrow P\ y \longrightarrow R\ x\ y$))
⟨*proof*⟩

lemma *is-least-in-ffilter-wrt-swap-predicate*:
assumes
trans: *transp-on* (*fset X*) *R* **and**
asym: *asymp-on* (*fset X*) *R* **and**
tot: *totalp-on* (*fset X*) *R*
assumes
y-least: *is-least-in-fset-wrt R* (*ffilter P X*) *y* **and**
same-on-prefix: $\bigwedge x.\ x$ |∈| $X \Longrightarrow R^{==}\ x\ y \Longrightarrow P\ x \longleftrightarrow Q\ x$
shows *is-least-in-fset-wrt R* (*ffilter Q X*) *y*
⟨*proof*⟩

lemma *ex-is-least-in-ffilter-wrt-iff*:
assumes
trans: *transp-on* (*fset* (*ffilter P X*)) *R* **and**
asym: *asymp-on* (*fset* (*ffilter P X*)) *R* **and**
tot: *totalp-on* (*fset* (*ffilter P X*)) *R*
shows ($\exists x.$ *is-least-in-fset-wrt R* (*ffilter P X*) *x*) ⟷ ($\exists x$ |∈| $X.\ P\ x$)
⟨*proof*⟩

13 Hide stuff

We restrict the public interface to ease future internal changes.

hide-fact *is-minimal-in-fset-wrt-def is-maximal-in-fset-wrt-def*

hide-fact *is-least-in-fset-wrt-def is-greatest-in-fset-wrt-def*

14 Integration in type classes

abbreviation (**in** *order*) *is-minimal-in-fset* **where**
is-minimal-in-fset ≡ *is-minimal-in-fset-wrt* (<)

abbreviation (**in** *order*) *is-maximal-in-fset* **where**
is-maximal-in-fset ≡ *is-maximal-in-fset-wrt* (<)

lemmas (**in** *order*) *is-minimal-in-fset-iff* =
is-minimal-in-fset-wrt-iff[*OF transp-on-less asymp-on-less*]

lemmas (**in** *order*) *is-maximal-in-fset-iff* =
is-maximal-in-fset-wrt-iff[*OF transp-on-less asymp-on-less*]

lemmas (**in** *order*) *ex-minimal-in-fset* =
ex-minimal-in-fset-wrt[*OF transp-on-less asymp-on-less*]

lemmas (**in** *order*) *ex-maximal-in-fset* =
ex-maximal-in-fset-wrt[*OF transp-on-less asymp-on-less*]

lemmas (**in** *order*) *is-minimal-in-fset-ffilter-iff* =
is-minimal-in-fset-wrt-ffilter-iff[*OF transp-on-less asymp-on-less*]

lemmas (**in** *order*) *is-minimal-in-fset-finsertI* =
is-minimal-in-fset-wrt-finsertI[*OF transp-on-less asymp-on-less*]

abbreviation (**in** *linorder*) *is-least-in-fset* **where**
is-least-in-fset ≡ *is-least-in-fset-wrt* (<)

abbreviation (**in** *linorder*) *is-greatest-in-fset* **where**
is-greatest-in-fset ≡ *is-greatest-in-fset-wrt* (<)

lemmas (**in** *linorder*) *is-least-in-fset-iff* =
is-least-in-fset-wrt-iff[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *is-greatest-in-fset-iff* =
is-greatest-in-fset-wrt-iff[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *is-minimal-in-fset-eq-is-least-in-fset* =
is-minimal-in-fset-wrt-eq-is-least-in-fset-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *is-maximal-in-fset-eq-is-greatest-in-fset* =
is-maximal-in-fset-wrt-eq-is-greatest-in-fset-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *Uniq-is-least-in-fset*[*intro*] =
Uniq-is-least-in-fset-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *Uniq-is-greatest-in-fset*[*intro*] =
Uniq-is-greatest-in-fset-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *ex-least-in-fset* =
ex-least-in-fset-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *ex-greatest-in-fset* =
ex-greatest-in-fset-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *ex1-least-in-fset* =
ex1-least-in-fset-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *ex1-greatest-in-fset* =
ex1-greatest-in-fset-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *is-least-in-ffilter-iff* =
is-least-in-ffilter-wrt-iff[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *ex-is-least-in-ffilter-iff* =
ex-is-least-in-ffilter-wrt-iff[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *is-least-in-ffilter-swap-predicate* =
is-least-in-ffilter-wrt-swap-predicate[*OF transp-on-less asymp-on-less totalp-on-less*]

end
theory *Min-Max-Least-Greatest-Multiset*
imports
Relation-Reachability
Min-Max-Least-Greatest-Set
HOL−Library.Multiset
HOL−Library.Multiset-Order
begin

15 Minimal and maximal elements

definition *is-minimal-in-mset-wrt* :: (*'a* ⇒ *'a* ⇒ *bool*) ⇒ *'a multiset* ⇒ *'a* ⇒ *bool*
where
transp-on (*set-mset X*) *R* ⟹ *asymp-on* (*set-mset X*) *R* ⟹
is-minimal-in-mset-wrt R X = *is-minimal-in-set-wrt R* (*set-mset X*)

definition *is-maximal-in-mset-wrt* :: (*'a* ⇒ *'a* ⇒ *bool*) ⇒ *'a multiset* ⇒ *'a* ⇒ *bool*
where
transp-on (*set-mset X*) *R* ⟹ *asymp-on* (*set-mset X*) *R* ⟹
is-maximal-in-mset-wrt R X = *is-maximal-in-set-wrt R* (*set-mset X*)

definition *is-strictly-minimal-in-mset-wrt* :: (*'a* ⇒ *'a* ⇒ *bool*) ⇒ *'a multiset* ⇒ *'a*

$\Rightarrow$ *bool* **where**
transp-on (*set-mset X*) *R* $\Longrightarrow$ *asymp-on* (*set-mset X*) *R* $\Longrightarrow$
is-strictly-minimal-in-mset-wrt R X x $\longleftrightarrow$ *x* $\in\#$ *X* $\wedge$ ($\forall$ *y* $\in\#$ *X* $-$ {# *x* #}. $\neg$ ($R^{==}$ *y x*))

definition *is-strictly-maximal-in-mset-wrt* :: ('*a* $\Rightarrow$ '*a* $\Rightarrow$ *bool*) $\Rightarrow$ '*a multiset* $\Rightarrow$ '*a* $\Rightarrow$ *bool* **where**
transp-on (*set-mset X*) *R* $\Longrightarrow$ *asymp-on* (*set-mset X*) *R* $\Longrightarrow$
is-strictly-maximal-in-mset-wrt R X x $\longleftrightarrow$ *x* $\in\#$ *X* $\wedge$ ($\forall$ *y* $\in\#$ *X* $-$ {# *x* #}. $\neg$ ($R^{==}$ *x y*))

context
fixes *X R*
assumes
trans: *transp-on* (*set-mset X*) *R* **and**
asym: *asymp-on* (*set-mset X*) *R*
begin

15.1 Conversions

lemma *is-minimal-in-mset-wrt-iff*:
is-minimal-in-mset-wrt R X x $\longleftrightarrow$ *x* $\in\#$ *X* $\wedge$ ($\forall$ *y* $\in\#$ *X*. *y* $\neq$ *x* $\longrightarrow$ $\neg$ *R y x*)
⟨*proof*⟩

lemma *is-minimal-in-mset-wrt R X x* $\longleftrightarrow$ *x* $\in\#$ *X* $\wedge$ ($\forall$ *y* $\in\#$ *X*. $\neg$ *R y x*)
⟨*proof*⟩

lemma *is-maximal-in-mset-wrt-iff*:
is-maximal-in-mset-wrt R X x $\longleftrightarrow$ *x* $\in\#$ *X* $\wedge$ ($\forall$ *y* $\in\#$ *X*. *y* $\neq$ *x* $\longrightarrow$ $\neg$ *R x y*)
⟨*proof*⟩

lemma *is-maximal-in-mset-wrt R X x* $\longleftrightarrow$ *x* $\in\#$ *X* $\wedge$ ($\forall$ *y* $\in\#$ *X*. $\neg$ *R x y*)
⟨*proof*⟩

lemma *is-strictly-minimal-in-mset-wrt-iff*:
is-strictly-minimal-in-mset-wrt R X x $\longleftrightarrow$ *x* $\in\#$ *X* $\wedge$ ($\forall$ *y* $\in\#$ *X*$-$ {# *x* #}. $\neg$ $R^{==}$ *y x*)
⟨*proof*⟩

lemma *is-strictly-maximal-in-mset-wrt-iff*:
is-strictly-maximal-in-mset-wrt R X x $\longleftrightarrow$ *x* $\in\#$ *X* $\wedge$ ($\forall$ *y* $\in\#$ *X*$-$ {# *x* #}. $\neg$ $R^{==}$ *x y*)
⟨*proof*⟩

lemma *is-minimal-in-mset-wrt-if-is-strictly-minimal-in-mset-wrt*:
is-strictly-minimal-in-mset-wrt R X x $\Longrightarrow$ *is-minimal-in-mset-wrt R X x*
⟨*proof*⟩

lemma *is-maximal-in-mset-wrt-if-is-strictly-maximal-in-mset-wrt*:

is-strictly-maximal-in-mset-wrt R X x $\Longrightarrow$ *is-maximal-in-mset-wrt R X x*
⟨*proof*⟩

15.2 Existence

lemma *ex-minimal-in-mset-wrt*:
$X \neq \{\#\} \Longrightarrow \exists\, m.$ *is-minimal-in-mset-wrt R X m*
⟨*proof*⟩

lemma *ex-maximal-in-mset-wrt*:
$X \neq \{\#\} \Longrightarrow \exists\, m.$ *is-maximal-in-mset-wrt R X m*
⟨*proof*⟩

15.3 Miscellaneous

lemma *explode-maximal-in-mset-wrt*:
assumes *max*: *is-maximal-in-mset-wrt R X x*
obtains n :: *nat* **where** *replicate-mset* (*Suc n*) $x + \{\#y \in\# X.\ y \neq x\#\} = X$
⟨*proof*⟩

lemma *explode-strictly-maximal-in-mset-wrt*:
assumes *max*: *is-strictly-maximal-in-mset-wrt R X x*
shows *add-mset x* $\{\#y \in\# X.\ y \neq x\#\} = X$
⟨*proof*⟩

end

lemma *is-minimal-in-filter-mset-wrt-iff*:
assumes
tran: *transp-on* (*set-mset* (*filter-mset P X*)) *R* **and**
asym: *asymp-on* (*set-mset* (*filter-mset P X*)) *R*
shows *is-minimal-in-mset-wrt R* (*filter-mset P X*) *x* $\longleftrightarrow$
$(x \in\# X \wedge P\ x \wedge (\forall\, y \in\# X - \{\#x\#\}.\ P\ y \longrightarrow \neg\ R\ y\ x))$
⟨*proof*⟩

lemma *multp-if-maximal-of-lhs-is-less*:
assumes
trans: *transp R* **and**
asym: *asymp-on* (*set-mset M1*) *R* **and**
tot: *totalp-on* (*set-mset M1* $\cup$ *set-mset M2*) *R* **and**
x1 $\in\#$ *M1* **and** *x2* $\in\#$ *M2* **and**
is-maximal-in-mset-wrt R M1 x1 **and** *R x1 x2*
shows *multp R M1 M2*
⟨*proof*⟩

15.4 Nonuniqueness

lemma
fixes x :: $'a$ **and** y :: $'a$
assumes $x \neq y$

shows
not-Uniq-is-minimal-in-mset-if-two-distinct-elements:
$\neg\ (\forall\,(R :: 'a \Rightarrow 'a \Rightarrow bool)\ (X :: 'a\ multiset).$
transp-on (*set-mset* X) $R \longrightarrow$ *asymp-on* (*set-mset* X) $R \longrightarrow$
$(\exists_{\leq 1} x.$ *is-minimal-in-mset-wrt* $R\ X\ x))$ **and**
not-Uniq-is-maximal-in-mset-wrt-if-two-distinct-elements:
$\neg\ (\forall\,(R :: 'a \Rightarrow 'a \Rightarrow bool)\ (X :: 'a\ multiset).$
transp-on (*set-mset* X) $R \longrightarrow$ *asymp-on* (*set-mset* X) $R \longrightarrow$
$(\exists_{\leq 1} x.$ *is-maximal-in-mset-wrt* $R\ X\ x))$ **and**
not-Uniq-is-strictly-minimal-in-mset-if-two-distinct-elements:
$\neg\ (\forall\,(R :: 'a \Rightarrow 'a \Rightarrow bool)\ (X :: 'a\ multiset).$
transp-on (*set-mset* X) $R \longrightarrow$ *asymp-on* (*set-mset* X) $R \longrightarrow$
$(\exists_{\leq 1} x.$ *is-strictly-minimal-in-mset-wrt* $R\ X\ x))$ **and**
not-Uniq-is-strictly-maximal-in-mset-wrt-if-two-distinct-elements:
$\neg\ (\forall\,(R :: 'a \Rightarrow 'a \Rightarrow bool)\ (X :: 'a\ multiset).$
transp-on (*set-mset* X) $R \longrightarrow$ *asymp-on* (*set-mset* X) $R \longrightarrow$
$(\exists_{\leq 1} x.$ *is-strictly-maximal-in-mset-wrt* $R\ X\ x))$
⟨*proof*⟩

16 Least and greatest elements

definition *is-least-in-mset-wrt* :: $('a \Rightarrow 'a \Rightarrow bool) \Rightarrow 'a\ multiset \Rightarrow 'a \Rightarrow bool$
where
transp-on (*set-mset* X) $R \Longrightarrow$ *asymp-on* (*set-mset* X) $R \Longrightarrow$ *totalp-on* (*set-mset* X) $R \Longrightarrow$
is-least-in-mset-wrt $R\ X\ x \longleftrightarrow x \in\# X \wedge (\forall\, y \in\# X - \{\#x\#\}.\ R\ x\ y)$

definition *is-greatest-in-mset-wrt* :: $('a \Rightarrow 'a \Rightarrow bool) \Rightarrow 'a\ multiset \Rightarrow 'a \Rightarrow bool$
where
transp-on (*set-mset* X) $R \Longrightarrow$ *asymp-on* (*set-mset* X) $R \Longrightarrow$ *totalp-on* (*set-mset* X) $R \Longrightarrow$
is-greatest-in-mset-wrt $R\ X\ x \longleftrightarrow x \in\# X \wedge (\forall\, y \in\# X - \{\#x\#\}.\ R\ y\ x)$

context
fixes $X\ R$
assumes
trans: *transp-on* (*set-mset* X) R **and**
asym: *asymp-on* (*set-mset* X) R **and**
tot: *totalp-on* (*set-mset* X) R
begin

16.1 Conversions

lemma *is-least-in-mset-wrt-iff*:
is-least-in-mset-wrt $R\ X\ x \longleftrightarrow x \in\# X \wedge (\forall\, y \in\# X - \{\#x\#\}.\ R\ x\ y)$
⟨*proof*⟩

lemma *is-greatest-in-mset-wrt-iff*:
is-greatest-in-mset-wrt $R\ X\ x \longleftrightarrow x \in\# X \wedge (\forall\, y \in\# X - \{\#x\#\}.\ R\ y\ x)$

⟨*proof*⟩

lemma *is-minimal-in-mset-wrt-if-is-least-in-mset-wrt*[*intro*]:
is-least-in-mset-wrt R X x ⟹ *is-minimal-in-mset-wrt R X x*
⟨*proof*⟩

lemma *is-maximal-in-mset-wrt-if-is-greatest-in-mset-wrt*[*intro*]:
is-greatest-in-mset-wrt R X x ⟹ *is-maximal-in-mset-wrt R X x*
⟨*proof*⟩

lemma *is-strictly-minimal-in-mset-wrt-iff-is-least-in-mset-wrt*:
is-strictly-minimal-in-mset-wrt R X = *is-least-in-mset-wrt R X*
⟨*proof*⟩

lemma *is-strictly-maximal-in-mset-wrt-iff-is-greatest-in-mset-wrt*:
is-strictly-maximal-in-mset-wrt R X = *is-greatest-in-mset-wrt R X*
⟨*proof*⟩

16.2 Uniqueness

lemma *Uniq-is-minimal-in-mset-wrt*[*intro*]:
$\exists_{\leq 1} x$. *is-minimal-in-mset-wrt R X x*
⟨*proof*⟩

lemma *Uniq-is-maximal-in-mset-wrt*[*intro*]:
$\exists_{\leq 1} x$. *is-maximal-in-mset-wrt R X x*
⟨*proof*⟩

lemma *Uniq-is-least-in-mset-wrt*[*intro*]:
$\exists_{\leq 1} x$. *is-least-in-mset-wrt R X x*
⟨*proof*⟩

lemma *Uniq-is-greatest-in-mset-wrt*[*intro*]:
$\exists_{\leq 1} x$. *is-greatest-in-mset-wrt R X x*
⟨*proof*⟩

16.3 Miscellaneous

lemma *is-least-in-mset-wrt-iff-is-minimal-and-count-eq-one*:
is-least-in-mset-wrt R X x ⟷ *is-minimal-in-mset-wrt R X x* ∧ *count X x* = *1*
⟨*proof*⟩

lemma *is-greatest-in-mset-wrt-iff-is-maximal-and-count-eq-one*:
is-greatest-in-mset-wrt R X x ⟷ *is-maximal-in-mset-wrt R X x* ∧ *count X x* = *1*
⟨*proof*⟩

lemma *count-ge-2-if-minimal-in-mset-wrt-and-not-least-in-mset-wrt*:
assumes *is-minimal-in-mset-wrt R X x* **and** ¬ *is-least-in-mset-wrt R X x*
shows *count X x* ≥ *2*

⟨*proof*⟩

lemma *count-ge-2-if-maximal-in-mset-wrt-and-not-greatest-in-mset-wrt*:
assumes *is-maximal-in-mset-wrt R X x* **and** ¬ *is-greatest-in-mset-wrt R X x*
shows *count X x* ≥ *2*
⟨*proof*⟩

lemma *explode-greatest-in-mset-wrt*:
assumes *max*: *is-greatest-in-mset-wrt R X x*
shows *add-mset x* {#*y* ∈# *X*. *y* ≠ *x*#} = *X*
⟨*proof*⟩

end

lemma *multp*$_{HO}$*-if-maximal-wrt-less-that-maximal-wrt*:
assumes
trans: *transp-on* (*set-mset M1* ∪ *set-mset M2*) *R* **and**
asym: *asymp-on* (*set-mset M1* ∪ *set-mset M2*) *R* **and**
tot: *totalp-on* (*set-mset M1* ∪ *set-mset M2*) *R* **and**
x1-maximal: *is-maximal-in-mset-wrt R M1 x1* **and**
x2-maximal: *is-maximal-in-mset-wrt R M2 x2* **and**
R x1 x2
shows *multp*$_{HO}$ *R M1 M2*
⟨*proof*⟩

lemma *multp*$_{DM}$*-if-maximal-wrt-less-that-maximal-wrt*:
assumes
trans: *transp-on* (*set-mset M1* ∪ *set-mset M2*) *R* **and**
asym: *asymp-on* (*set-mset M1* ∪ *set-mset M2*) *R* **and**
tot: *totalp-on* (*set-mset M1* ∪ *set-mset M2*) *R* **and**
x1-maximal: *is-maximal-in-mset-wrt R M1 x1* **and**
x2-maximal: *is-maximal-in-mset-wrt R M2 x2* **and**
R x1 x2
shows *multp*$_{DM}$ *R M1 M2*
⟨*proof*⟩

lemma *multp-if-maximal-wrt-less-that-maximal-wrt*:
assumes
trans: *transp-on* (*set-mset M1* ∪ *set-mset M2*) *R* **and**
asym: *asymp-on* (*set-mset M1* ∪ *set-mset M2*) *R* **and**
tot: *totalp-on* (*set-mset M1* ∪ *set-mset M2*) *R* **and**
x1-maximal: *is-maximal-in-mset-wrt R M1 x1* **and**
x2-maximal: *is-maximal-in-mset-wrt R M2 x2* **and**
R x1 x2
shows *multp R M1 M2*
⟨*proof*⟩

lemma *multp$_{HO}$-if-same-maximal-wrt-and-count-lt*:
assumes
trans: *transp-on* (*set-mset M1* ∪ *set-mset M2*) *R* **and**
asym: *asymp-on* (*set-mset M1* ∪ *set-mset M2*) *R* **and**
tot: *totalp-on* (*set-mset M1* ∪ *set-mset M2*) *R* **and**
x1-maximal: *is-maximal-in-mset-wrt R M1 x* **and**
x2-maximal: *is-maximal-in-mset-wrt R M2 x* **and**
count M1 x < *count M2 x*
shows *multp$_{HO}$ R M1 M2*
⟨*proof*⟩

lemma *multp-if-same-maximal-wrt-and-count-lt*:
assumes
trans: *transp-on* (*set-mset M1* ∪ *set-mset M2*) *R* **and**
asym: *asymp-on* (*set-mset M1* ∪ *set-mset M2*) *R* **and**
tot: *totalp-on* (*set-mset M1* ∪ *set-mset M2*) *R* **and**
x1-maximal: *is-maximal-in-mset-wrt R M1 x* **and**
x2-maximal: *is-maximal-in-mset-wrt R M2 x* **and**
count M1 x < *count M2 x*
shows *multp R M1 M2*
⟨*proof*⟩

lemma *less-than-maximal-wrt-if-multp$_{HO}$*:
assumes
trans: *transp-on* (*set-mset M1* ∪ *set-mset M2*) *R* **and**
asym: *asymp-on* (*set-mset M2*) *R* **and**
tot: *totalp-on* (*set-mset M2*) *R* **and**
x2-maximal: *is-maximal-in-mset-wrt R M2 x2* **and**
multp$_{HO}$ R M1 M2 **and**
x1 ∈# *M1*
shows $R^{==}$ *x1 x2*
⟨*proof*⟩

17 Examples of duplicate handling in set and multiset definitions

lemma
fixes *M* :: *nat multiset*
defines *M* ≡ {#*0*, *0*, *1*, *1*, *2*, *2*#}
shows
is-minimal-in-set-wrt (<) (*set-mset M*) *0*
is-minimal-in-mset-wrt (<) *M 0*
is-least-in-set-wrt (<) (*set-mset M*) *0*
∄ *y*. *is-least-in-mset-wrt* (<) *M y*
⟨*proof*⟩

lemma
fixes *x y* :: *'a* **and** *M* :: *'a multiset*

defines $M \equiv \{\#x,\ y,\ y\#\}$
defines $R \equiv \lambda\text{- -}.\ \mathit{False}$
assumes $x \neq y$
shows
$\mathit{is\text{-}maximal\text{-}in\text{-}mset\text{-}wrt}\ R\ M\ x$
$\mathit{is\text{-}maximal\text{-}in\text{-}mset\text{-}wrt}\ R\ M\ y$
$\mathit{is\text{-}strictly\text{-}maximal\text{-}in\text{-}mset\text{-}wrt}\ R\ M\ x$
$\neg\ \mathit{is\text{-}strictly\text{-}maximal\text{-}in\text{-}mset\text{-}wrt}\ R\ M\ y$
⟨*proof*⟩

18 Hide stuff

We restrict the public interface to ease future internal changes.

hide-fact *is-minimal-in-mset-wrt-def is-maximal-in-mset-wrt-def*
hide-fact *is-strictly-minimal-in-mset-wrt-def is-strictly-maximal-in-mset-wrt-def*
hide-fact *is-least-in-mset-wrt-def is-greatest-in-mset-wrt-def*

19 Integration in type classes

abbreviation (**in** *order*) *is-minimal-in-mset* **where**
is-minimal-in-mset ≡ *is-minimal-in-mset-wrt* (<)

abbreviation (**in** *order*) *is-maximal-in-mset* **where**
is-maximal-in-mset ≡ *is-maximal-in-mset-wrt* (<)

lemmas (**in** *order*) *is-minimal-in-mset-iff* =
is-minimal-in-mset-wrt-iff[*OF transp-on-less asymp-on-less*]

lemmas (**in** *order*) *is-maximal-in-mset-iff* =
is-maximal-in-mset-wrt-iff[*OF transp-on-less asymp-on-less*]

lemmas (**in** *order*) *ex-minimal-in-mset* =
ex-minimal-in-mset-wrt[*OF transp-on-less asymp-on-less*]

lemmas (**in** *order*) *ex-maximal-in-mset* =
ex-maximal-in-mset-wrt[*OF transp-on-less asymp-on-less*]

lemmas (**in** *order*) *explode-maximal-in-mset* =
explode-maximal-in-mset-wrt[*OF transp-on-less asymp-on-less*]

lemmas (**in** *order*) *explode-strictly-maximal-in-mset* =
explode-strictly-maximal-in-mset-wrt[*OF transp-on-less asymp-on-less*]

lemmas (**in** *order*) *is-minimal-in-filter-mset-iff* =
is-minimal-in-filter-mset-wrt-iff[*OF transp-on-less asymp-on-less*]

abbreviation (**in** *linorder*) *is-least-in-mset* **where**
is-least-in-mset ≡ *is-least-in-mset-wrt* (<)

abbreviation (**in** *linorder*) *is-greatest-in-mset* **where**
is-greatest-in-mset ≡ *is-greatest-in-mset-wrt* (<)

lemmas (**in** *linorder*) *is-least-in-mset-iff* =
is-least-in-mset-wrt-iff[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *is-greatest-in-mset-iff* =
is-greatest-in-mset-wrt-iff[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *is-minimal-in-mset-if-is-least-in-mset*[*intro*] =
is-minimal-in-mset-wrt-if-is-least-in-mset-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *is-maximal-in-mset-if-is-greatest-in-mset*[*intro*] =
is-maximal-in-mset-wrt-if-is-greatest-in-mset-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *Uniq-is-minimal-in-mset*[*intro*] =
Uniq-is-minimal-in-mset-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *Uniq-is-maximal-in-mset*[*intro*] =
Uniq-is-maximal-in-mset-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *Uniq-is-least-in-mset*[*intro*] =
Uniq-is-least-in-mset-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *Uniq-is-greatest-in-mset*[*intro*] =
Uniq-is-greatest-in-mset-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *is-least-in-mset-iff-is-minimal-and-count-eq-one* =
is-least-in-mset-wrt-iff-is-minimal-and-count-eq-one[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *is-greatest-in-mset-iff-is-maximal-and-count-eq-one* =
is-greatest-in-mset-wrt-iff-is-maximal-and-count-eq-one[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *count-ge-2-if-minimal-in-mset-and-not-least-in-mset* =
count-ge-2-if-minimal-in-mset-wrt-and-not-least-in-mset-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *count-ge-2-if-maximal-in-mset-and-not-greatest-in-mset* =
count-ge-2-if-maximal-in-mset-wrt-and-not-greatest-in-mset-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *explode-greatest-in-mset* =
explode-greatest-in-mset-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) $multp_{HO}$*-if-maximal-less-that-maximal* =
$multp_{HO}$*-if-maximal-wrt-less-that-maximal-wrt*[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) $multp_{DM}$*-if-maximal-less-that-maximal* =
$multp_{DM}$*-if-maximal-wrt-less-that-maximal-wrt*[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *multp-if-maximal-less-that-maximal* =
multp-if-maximal-wrt-less-that-maximal-wrt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) $multp_{HO}$*-if-same-maximal-and-count-lt* =
$multp_{HO}$*-if-same-maximal-wrt-and-count-lt*[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *multp-if-same-maximal-and-count-lt* =
multp-if-same-maximal-wrt-and-count-lt[*OF transp-on-less asymp-on-less totalp-on-less*]

lemmas (**in** *linorder*) *less-than-maximal-if-*$multp_{HO}$ =
less-than-maximal-wrt-if-$multp_{HO}$[*OF transp-on-less asymp-on-less totalp-on-less*]

lemma (**in** *linorder*)
assumes*is-greatest-in-mset C L*
shows $C - \{\#L\#\} = \{\#K \in\# C.\ K \neq L\#\}$
⟨*proof*⟩

end