Complete Non-Orders and Fixed Points

Akihisa Yamada and Jérémy Dubut

March 17, 2025

Abstract

We develop an Isabelle/HOL library of order-theoretic concepts, such as various completeness conditions and fixed-point theorems. We keep our formalization as general as possible: we reprove several well-known results about complete orders, often with only antisymmetry or attractivity, a mild condition implied by either antisymmetry or transitivity. In particular, we generalize various theorems ensuring the existence of a quasi-fixed point of monotone maps over complete relations, and show that the set of (quasi-)fixed points is itself complete. This result generalizes and strengthens theorems of Knaster–Tarski, Bourbaki–Witt, Kleene, Markowsky, Pataraia, Mashburn, Bhatta–George, and Stouti–Maaden.

Contents

1 Introduction

The main driving force towards mechanizing mathematics using proof assistants has been the reliability they offer, exemplified prominently by [10], [12], [15], etc. In this work, we utilize another aspect of proof assistants: they are also engineering tools for developing mathematical theories.

Fixed-point theorems are important in computer science, such as in denotational semantics [20] and in abstract interpretation [7], as they allow the definition of semantics of loops and recursive functions. The Knaster–Tarski theorem [23] shows that any monotone map $f : A \to A$ over complete lattice $(A, \sqsubseteq)$ has a fixed point, and the set of fixed points forms also a complete lattice. The result was generalized in various ways: Markowsky [16] showed a corresponding result for *chain-complete* posets. The proof uses the Bourbaki–Witt theorem [6], stating that any inflationary map over a chain-complete poset has a fixed point. The original proof of the latter is non-elementary in the sense that it relies on ordinals and Hartogs' theorem. Pataraia [18] gave an elementary proof that monotone maps over *pointed directed-complete* poset has a fixed point. Fixed points are studied also for *pseudo-orders* [21], relaxing transitivity. Stouti and Maaden [22] showed that every monotone map over a complete pseudo-order has a (least) fixed point. Markowsky's result was also generalized to *weak* chain-complete pseudo-orders by Bhatta and George [4, 5].

Another line of order-theoretic fixed points is the *iterative* approach. Kantorovitch showed that for ω*-continuous* map f over a complete lattice,[1] the iteration $\bot, f\ \bot, f^2\ \bot, \ldots$ converges to a fixed point [14, Theorem I]. Tarski [23] also claimed a similar result for a *countably distributive* map over a *countably complete Boolean algebra*. Kleene's fixed-point theorem states that, for *Scott-continuous* maps over pointed directed-complete posets, the iteration converges to the least fixed point. Finally, Mashburn [17] proved a version for ω-continuous maps over ω-complete posets, which covers Kantorovitch's, Tarski's and Kleene's claims.

In particular, we provide the following:

- Several *locales* that help organizing the different order-theoretic conditions, such as reflexivity, transitivity, antisymmetry, and their combination, as well as concepts such as connex and well-related sets, analogues of chains and well-ordered sets in a non-ordered context.

- Existence of fixed points: We provide two proof schemes to prove that monotone or inflationary mapping $f : A \to A$ over a complete related set $\langle A, \sqsubseteq \rangle$ has a *quasi-fixed point* $f\ x \sim x$, meaning $x \sqsubseteq f\ x \wedge f\ x \sqsubseteq x$, for various notions of completeness. The first one, similar to the original proof by Tarski [23], does not require any ordering assumptions, but relies on completeness with respect to all subsets. The second one, inspired by a *constructive* approach by Grall [11], is a proof scheme based on the notion of derivations. Here we demand antisymmetry (to avoid the necessity of the axiom of choice), but can be instantiated to *well-complete* sets, a generalization of weak chain-completeness. This also allows us to generalize Bourbaki–Witt theorem [6] to pseudo-orders.

- Completeness of the set of fixed points: if $(A, \sqsubseteq)$ satisfies a mild condition, which we call *attractivity* and which is implied by either transitivity or antisymmetry, then the set of quasi-fixed points inherits the completeness class from $(A, \sqsubseteq)$, if it is at least well-complete. The result instantiates to the full completeness (generalizing Knaster–Tarski and [22]), directed-completeness [18], chain-completeness [16], and weak chain-completeness [5].

[1] More precisely, he assumes a conditionally complete lattice defined over vectors and that $\bot \sqsubseteq f\ \bot$ and $f\ v' \sqsubseteq v'$. Hence f, which is monotone, is a map over the complete lattice $\{v \mid \bot \sqsubseteq v \sqsubseteq v'\}$.

- Iterative construction: For an ω-continuous map over an ω-complete related set, we show that suprema of $\{f^n \perp \mid n \in \mathbb{N}\}$ are quasi-fixed points. Under attractivity, the quasi-fixed points obtained from this scheme are precisely the least quasi-fixed points of f. This generalizes Mashburn's result, and thus ones by Kantorovitch, Tarski and Kleene.

We remark that all these results would have required much more effort than we spent (if possible at all), if we were not with the aforementioned smart assistance by Isabelle. Our workflow was often the following: first we formalize existing proofs, try relaxing assumptions, see where proof breaks, and at some point ask for a counterexample.

Concerning Isabelle formalization, one can easily find several formalizations of complete partial orders or lattices in Isabelle?s standard library. They are, however, defined on partial orders, either in form of classes or locales, and thus not directly reusable for non-orders. Nevertheless we tried to make our formalization compatible with the existing ones, and various correspondences are ensured.

This archive is the third version of this work. The first version has been published in the conference paper [24]. The second version has been published in the journal paper [8]. The third version is a restructuration of the second version for future formalizations, including [25].

2 Binary Relations

We start with basic properties of binary relations.

theory *Binary-Relations*
imports

Main
begin

unbundle *lattice-syntax*

lemma *conj-iff-conj-iff-imp-iff*: $Trueprop\ (x \wedge y \longleftrightarrow x \wedge z) \equiv (x \Longrightarrow (y \longleftrightarrow z))$
⟨*proof*⟩

lemma *conj-imp-eq-imp-imp*: $(P \wedge Q \Longrightarrow PROP\ R) \equiv (P \Longrightarrow Q \Longrightarrow PROP\ R)$
⟨*proof*⟩

lemma *tranclp-trancl*: $r^{++} = (\lambda x\ y.\ (x,y) \in \{(a,b).\ r\ a\ b\}^{+})$
⟨*proof*⟩

lemma *tranclp-id*[*simp*]: $transp\ r \Longrightarrow tranclp\ r = r$
⟨*proof*⟩

lemma *transp-tranclp*[*simp*]: $transp\ (tranclp\ r)$ ⟨*proof*⟩

lemma *funpow-dom*: f ‘ $A \subseteq A \Longrightarrow (f\widehat{\ \ }\widehat{\ \ }n)$ ‘ $A \subseteq A$ ⟨*proof*⟩

lemma *image-subsetD*: f ‘ $A \subseteq B \Longrightarrow a \in A \Longrightarrow f\ a \in B$ ⟨*proof*⟩

Below we introduce an Isabelle-notation for $\{\ldots x \ldots \mid x \in X\}$.

syntax
-range :: *'a* ⇒ *idts* ⇒ *'a set* (‹(*1*{- /|./ -})›)
-image :: *'a* ⇒ *pttrn* ⇒ *'a set* ⇒ *'a set* (‹(*1*{- /|./ (-/ ∈ -)})›)
syntax-consts
-range ⇌ *range* **and**
-image ⇌ *image*
translations
{*e* |. *p*} ⇌ *CONST range* (λ*p*. *e*)
{*e* |. *p* ∈ *A*} ⇌ *CONST image* (λ*p*. *e*) *A*

lemma *image-constant*:
assumes $\bigwedge i.\ i \in I \Longrightarrow f\ i = y$
shows f ‘ I = (*if* $I = \{\}$ *then* $\{\}$ *else* $\{y\}$)
⟨*proof*⟩

2.1 Various Definitions

Here we introduce various definitions for binary relations. The first one is our abbreviation for the dual of a relation.

abbreviation(*input*) *dual* (‹(-$^-$)› [*1000*] *1000*) **where** $r^-\ x\ y \equiv r\ y\ x$

lemma *conversep-is-dual*[*simp*]: *conversep* = *dual* ⟨*proof*⟩

lemma *dual-inf*: $(r \sqcap s)^- = r^- \sqcap s^-$ ⟨*proof*⟩

Monotonicity is already defined in the library, but we want one restricted to a domain.

lemmas *monotone-onE* = *monotone-on-def*[*unfolded atomize-eq*, *THEN iffD1*, *elim-format*, *rule-format*]

lemma *monotone-on-dual*: *monotone-on X r s f* ⟹ *monotone-on X* r^- s^- *f*
⟨*proof*⟩

lemma *monotone-on-id*: *monotone-on X r r id*
⟨*proof*⟩

lemma *monotone-on-cmono*: *A* ⊆ *B* ⟹ *monotone-on B* ≤ *monotone-on A*
⟨*proof*⟩

Here we define the following notions in a standard manner

The symmetric part of a relation:

definition *sympartp* **where** *sympartp r x y* ≡ *r x y* ∧ *r y x*

lemma *sympartpI*[*intro*]:
  fixes r (**infix** ‹⊑› *50*)
  assumes $x \sqsubseteq y$ **and** $y \sqsubseteq x$ **shows** *sympartp* (⊑) x y
  ⟨*proof*⟩

lemma *sympartpE*[*elim*]:
  fixes r (**infix** ‹⊑› *50*)
  assumes *sympartp* (⊑) x y **and** $x \sqsubseteq y \Longrightarrow y \sqsubseteq x \Longrightarrow$ *thesis* **shows** *thesis*
  ⟨*proof*⟩

lemma *sympartp-dual*: *sympartp* r^- = *sympartp* r
  ⟨*proof*⟩

lemma *sympartp-eq*[*simp*]: *sympartp* (=) = (=) ⟨*proof*⟩

lemma *sympartp-sympartp*[*simp*]: *sympartp* (*sympartp* r) = *sympartp* r ⟨*proof*⟩

lemma *reflclp-sympartp*[*simp*]: (*sympartp* r)$^{==}$ = *sympartp* $r^{==}$ ⟨*proof*⟩

definition *equivpartp* r x $y \equiv x = y \lor r\ x\ y \land r\ y\ x$

lemma *sympartp-reflclp-equivp*[*simp*]: *sympartp* $r^{==}$ = *equivpartp* r ⟨*proof*⟩

lemma *equivpartI*[*simp*]: *equivpartp* r x x
  and *sympartp-equivpartpI*: *sympartp* r x $y \Longrightarrow$ *equivpartp* r x y
  and *equivpartpCI*[*intro*]: ($x \neq y \Longrightarrow$ *sympartp* r x y) $\Longrightarrow$ *equivpartp* r x y
  ⟨*proof*⟩

lemma *equivpartpE*[*elim*]:
  assumes *equivpartp* r x y
    and $x = y \Longrightarrow$ *thesis*
    and $r\ x\ y \Longrightarrow r\ y\ x \Longrightarrow$ *thesis*
  shows *thesis*
  ⟨*proof*⟩

lemma *equivpartp-eq*[*simp*]: *equivpartp* (=) = (=) ⟨*proof*⟩

lemma *sympartp-equivpartp*[*simp*]: *sympartp* (*equivpartp* r) = (*equivpartp* r)
  and *equivpartp-equivpartp*[*simp*]: *equivpartp* (*equivpartp* r) = (*equivpartp* r)
  and *equivpartp-sympartp*[*simp*]: *equivpartp* (*sympartp* r) = (*equivpartp* r)
  ⟨*proof*⟩

lemma *equivpartp-dual*: *equivpartp* r^- = *equivpartp* r
  ⟨*proof*⟩

The asymmetric part:

definition *asympartp* r x $y \equiv r\ x\ y \land \neg\ r\ y\ x$

lemma *asympartpE*[*elim*]:
fixes *r* (**infix** ‹⊑› *50*)
shows *asympartp* (⊑) *x y* ⟹ (*x* ⊑ *y* ⟹ ¬*y* ⊑ *x* ⟹ *thesis*) ⟹ *thesis*
⟨*proof*⟩

lemmas *asympartpI*[*intro*] = *asympartp-def*[*unfolded atomize-eq*, *THEN iffD2*, *unfolded conj-imp-eq-imp-imp*, *rule-format*]

lemma *asympartp-eq*[*simp*]: *asympartp* (=) = *bot* ⟨*proof*⟩

lemma *asympartp-sympartp* [*simp*]: *asympartp* (*sympartp r*) = *bot*
and *sympartp-asympartp* [*simp*]: *sympartp* (*asympartp r*) = *bot*
⟨*proof*⟩

lemma *asympartp-dual*: *asympartp* r^- = (*asympartp r*)$^-$ ⟨*proof*⟩

Restriction to a set:

definition *Restrp* (**infixl** ‹↾› *60*) **where** (*r* ↾ *A*) *a b* ≡ *a* ∈ *A* ∧ *b* ∈ *A* ∧ *r a b*

lemmas *RestrpI*[*intro*!] = *Restrp-def*[*unfolded atomize-eq*, *THEN iffD2*, *unfolded conj-imp-eq-imp-imp*]
lemmas *RestrpE*[*elim*!] = *Restrp-def*[*unfolded atomize-eq*, *THEN iffD1*, *elim-format*, *unfolded conj-imp-eq-imp-imp*]

lemma *Restrp-simp*[*simp*]: *a* ∈ *A* ⟹ *b* ∈ *A* ⟹ (*r* ↾ *A*) *a b* ⟷ *r a b* ⟨*proof*⟩

lemma *Restrp-UNIV*[*simp*]: *r* ↾ *UNIV* ≡ *r* ⟨*proof*⟩

lemma *Restrp-Restrp*[*simp*]: *r* ↾ *A* ↾ *B* ≡ *r* ↾ *A* ∩ *B* ⟨*proof*⟩

lemma *sympartp-Restrp*[*simp*]: *sympartp* (*r* ↾ *A*) ≡ *sympartp r* ↾ *A*
⟨*proof*⟩

Relational images:

definition *Imagep* (**infixr** ‹‘‘‘› *59*) **where** *r* ‘‘‘ *A* ≡ {*b*. ∃ *a* ∈ *A*. *r a b*}

lemma *Imagep-Image*: *r* ‘‘‘ *A* = {(*a*,*b*). *r a b*} ‘‘ *A*
⟨*proof*⟩

lemma *in-Imagep*: *b* ∈ *r* ‘‘‘ *A* ⟷ (∃ *a* ∈ *A*. *r a b*) ⟨*proof*⟩

lemma *ImagepI*: *a* ∈ *A* ⟹ *r a b* ⟹ *b* ∈ *r* ‘‘‘ *A* ⟨*proof*⟩

lemma *subset-Imagep*: *B* ⊆ *r* ‘‘‘ *A* ⟷ (∀ *b*∈*B*. ∃ *a*∈*A*. *r a b*)
⟨*proof*⟩

Bounds of a set:

definition *bound X* (⊑) *b* ≡ ∀ *x* ∈ *X*. *x* ⊑ *b* **for** *r* (**infix** ‹⊑› *50*)

lemma
fixes r (**infix** ‹$\sqsubseteq$› *50*)
shows *boundI*[*intro*!]: $(\bigwedge x.\ x \in X \Longrightarrow x \sqsubseteq b) \Longrightarrow$ *bound* X $(\sqsubseteq)$ b
and *boundE*[*elim*]: *bound* X $(\sqsubseteq)$ $b \Longrightarrow ((\bigwedge x.\ x \in X \Longrightarrow x \sqsubseteq b) \Longrightarrow$ *thesis*$) \Longrightarrow$ *thesis*
and *boundD*: *bound* X $(\sqsubseteq)$ $b \Longrightarrow a \in X \Longrightarrow a \sqsubseteq b$
⟨*proof*⟩

lemma *bound-empty*: *bound* {} = ($\lambda r\ x.$ *True*) ⟨*proof*⟩

lemma *bound-cmono*: **assumes** $X \subseteq Y$ **shows** *bound* $Y \leq$ *bound* X
⟨*proof*⟩

lemmas *bound-subset* = *bound-cmono*[*THEN le-funD*, *THEN le-funD*, *THEN le-boolD*, *folded atomize-imp*]

lemma *bound-un*: *bound* $(A \cup B)$ = *bound* $A \sqcap$ *bound* B
⟨*proof*⟩

lemma *bound-insert*[*simp*]:
fixes r (**infix** ‹$\sqsubseteq$› *50*)
shows *bound* (*insert* x X) $(\sqsubseteq)$ $b \longleftrightarrow x \sqsubseteq b \wedge$ *bound* X $(\sqsubseteq)$ b ⟨*proof*⟩

lemma *bound-cong*:
assumes $A = A'$
and $b = b'$
and $\bigwedge a.\ a \in A' \Longrightarrow le\ a\ b' = le'\ a\ b'$
shows *bound* A *le* b = *bound* A' le' b'
⟨*proof*⟩

lemma *bound-subsel*: $le \leq le' \Longrightarrow$ *bound* A *le* $\leq$ *bound* A le'
⟨*proof*⟩

Extreme (greatest) elements in a set:

definition *extreme* X $(\sqsubseteq)$ $e \equiv e \in X \wedge (\forall x \in X.\ x \sqsubseteq e)$ **for** r (**infix** ‹$\sqsubseteq$› *50*)

lemma
fixes r (**infix** ‹$\sqsubseteq$› *50*)
shows *extremeI*[*intro*]: $e \in X \Longrightarrow (\bigwedge x.\ x \in X \Longrightarrow x \sqsubseteq e) \Longrightarrow$ *extreme* X $(\sqsubseteq)$ e
and *extremeD*: *extreme* X $(\sqsubseteq)$ $e \Longrightarrow e \in X$ *extreme* X $(\sqsubseteq)$ $e \Longrightarrow (\bigwedge x.\ x \in X \Longrightarrow x \sqsubseteq e)$
and *extremeE*[*elim*]: *extreme* X $(\sqsubseteq)$ $e \Longrightarrow (e \in X \Longrightarrow (\bigwedge x.\ x \in X \Longrightarrow x \sqsubseteq e) \Longrightarrow$ *thesis*$) \Longrightarrow$ *thesis*
⟨*proof*⟩

lemma
fixes r (**infix** ‹$\sqsubseteq$› *50*)
shows *extreme-UNIV*[*simp*]: *extreme UNIV* $(\sqsubseteq)$ $t \longleftrightarrow (\forall x.\ x \sqsubseteq t)$ ⟨*proof*⟩

lemma *extreme-iff-bound*: *extreme X r e* $\longleftrightarrow$ *bound X r e* $\wedge$ *e* $\in$ *X* ⟨*proof*⟩

lemma *extreme-imp-bound*: *extreme X r x* $\Longrightarrow$ *bound X r x* ⟨*proof*⟩

lemma *extreme-inf*: *extreme X* (*r* $\sqcap$ *s*) *x* $\longleftrightarrow$ *extreme X r x* $\wedge$ *extreme X s x* ⟨*proof*⟩

lemma *extremes-equiv*: *extreme X r b* $\Longrightarrow$ *extreme X r c* $\Longrightarrow$ *sympartp r b c* ⟨*proof*⟩

lemma *extreme-cong*:
  assumes *A* = *A'*
    and *b* = *b'*
    and $\bigwedge$*a*. *a* $\in$ *A'* $\Longrightarrow$ *b'* $\in$ *A'* $\Longrightarrow$ *le a b'* = *le' a b'*
  shows *extreme A le b* = *extreme A' le' b'*
  ⟨*proof*⟩

lemma *extreme-subset*: *X* $\subseteq$ *Y* $\Longrightarrow$ *extreme X r x* $\Longrightarrow$ *extreme Y r y* $\Longrightarrow$ *r x y* ⟨*proof*⟩

lemma *extreme-subrel*:
  le $\leq$ *le'* $\Longrightarrow$ *extreme A le* $\leq$ *extreme A le'* ⟨*proof*⟩

Now suprema and infima are given uniformly as follows. The definition is restricted to a given set.

definition
  extreme-bound A ($\sqsubseteq$) *X* $\equiv$ *extreme* {*b* $\in$ *A*. *bound X* ($\sqsubseteq$) *b*} ($\sqsubseteq$)$^-$ **for** *r* (**infix** ‹$\sqsubseteq$› *50*)

lemmas *extreme-boundI-extreme* = *extreme-bound-def*[*unfolded atomize-eq*, *THEN fun-cong*, *THEN iffD2*]

lemmas *extreme-boundD-extreme* = *extreme-bound-def*[*unfolded atomize-eq*, *THEN fun-cong*, *THEN iffD1*]

context
  fixes *A* :: *'a set* **and** *less-eq* :: *'a* $\Rightarrow$ *'a* $\Rightarrow$ *bool* (**infix** ‹$\sqsubseteq$› *50*)
begin

lemma *extreme-boundI*[*intro*]:
  assumes $\bigwedge$*b*. *bound X* ($\sqsubseteq$) *b* $\Longrightarrow$ *b* $\in$ *A* $\Longrightarrow$ *s* $\sqsubseteq$ *b* **and** $\bigwedge$*x*. *x* $\in$ *X* $\Longrightarrow$ *x* $\sqsubseteq$ *s* **and** *s* $\in$ *A*
  shows *extreme-bound A* ($\sqsubseteq$) *X s*
  ⟨*proof*⟩

lemma *extreme-boundD*:
  assumes *extreme-bound A* ($\sqsubseteq$) *X s*
  shows *x* $\in$ *X* $\Longrightarrow$ *x* $\sqsubseteq$ *s*
    and *bound X* ($\sqsubseteq$) *b* $\Longrightarrow$ *b* $\in$ *A* $\Longrightarrow$ *s* $\sqsubseteq$ *b*
    and *extreme-bound-in*: *s* $\in$ *A*

⟨*proof*⟩

lemma *extreme-boundE*[*elim*]:
assumes *extreme-bound* A (⊑) X s
and $s \in A \Longrightarrow$ *bound* X (⊑) $s \Longrightarrow (\bigwedge b.$ *bound* X (⊑) $b \Longrightarrow b \in A \Longrightarrow s \sqsubseteq b)$ $\Longrightarrow$ *thesis*
shows *thesis*
⟨*proof*⟩

lemma *extreme-bound-imp-bound*: *extreme-bound* A (⊑) X $s \Longrightarrow$ *bound* X (⊑) s
⟨*proof*⟩

lemma *extreme-imp-extreme-bound*:
assumes *Xs*: *extreme* X (⊑) s **and** *XA*: $X \subseteq A$ **shows** *extreme-bound* A (⊑) X s
⟨*proof*⟩

lemma *extreme-bound-subset-bound*:
assumes *XY*: $X \subseteq Y$
and *sX*: *extreme-bound* A (⊑) X s
and *b*: *bound* Y (⊑) b **and** *bA*: $b \in A$
shows $s \sqsubseteq b$
⟨*proof*⟩

lemma *extreme-bound-subset*:
assumes *XY*: $X \subseteq Y$
and *sX*: *extreme-bound* A (⊑) X *sX*
and *sY*: *extreme-bound* A (⊑) Y *sY*
shows *sX* ⊑ *sY*
⟨*proof*⟩

lemma *extreme-bound-iff*:
extreme-bound A (⊑) X $s \longleftrightarrow s \in A \wedge (\forall c \in A. (\forall x \in X. x \sqsubseteq c) \longrightarrow s \sqsubseteq c) \wedge (\forall x \in X. x \sqsubseteq s)$
⟨*proof*⟩

lemma *extreme-bound-empty*: *extreme-bound* A (⊑) {} $x \longleftrightarrow$ *extreme* A $(\sqsubseteq)^-$ x
⟨*proof*⟩

lemma *extreme-bound-singleton-refl*[*simp*]:
extreme-bound A (⊑) $\{x\}$ $x \longleftrightarrow x \in A \wedge x \sqsubseteq x$ ⟨*proof*⟩

lemma *extreme-bound-image-const*:
$x \sqsubseteq x \Longrightarrow I \neq \{\} \Longrightarrow (\bigwedge i.\ i \in I \Longrightarrow f\ i = x) \Longrightarrow x \in A \Longrightarrow$ *extreme-bound* A (⊑) $(f\ `\ I)$ x
⟨*proof*⟩

lemma *extreme-bound-UN-const*:
$x \sqsubseteq x \Longrightarrow I \neq \{\} \Longrightarrow (\bigwedge i\ y.\ i \in I \Longrightarrow P\ i\ y \longleftrightarrow x = y) \Longrightarrow x \in A \Longrightarrow$

extreme-bound A (⊑) ($\bigcup i{\in}I$. $\{y.\ P\ i\ y\}$) x
⟨*proof*⟩

lemma *extreme-bounds-equiv*:
assumes *s*: *extreme-bound* A (⊑) X s **and** s': *extreme-bound* A (⊑) X s'
shows *sympartp* (⊑) s s'
⟨*proof*⟩

lemma *extreme-bound-squeeze*:
assumes *XY*: $X \subseteq Y$ **and** *YZ*: $Y \subseteq Z$
and *Xs*: *extreme-bound* A (⊑) X s **and** *Zs*: *extreme-bound* A (⊑) Z s
shows *extreme-bound* A (⊑) Y s
⟨*proof*⟩

lemma *bound-closed-imp-extreme-bound-eq-extreme*:
assumes *closed*: $\forall\, b \in A$. *bound* X (⊑) $b \longrightarrow b \in X$ **and** *XA*: $X \subseteq A$
shows *extreme-bound* A (⊑) X = *extreme* X (⊑)
⟨*proof*⟩

end

lemma *extreme-bound-cong*:
assumes $A = A'$
and $X = X'$
and $\bigwedge a\ b.\ a \in A' \Longrightarrow b \in A' \Longrightarrow$ *le* a $b \longleftrightarrow$ *le*′ a b
and $\bigwedge a\ b.\ a \in X' \Longrightarrow b \in A' \Longrightarrow$ *le* a $b \longleftrightarrow$ *le*′ a b
shows *extreme-bound* A *le* X s = *extreme-bound* A *le*′ X s
⟨*proof*⟩

Maximal or Minimal

definition *extremal* X (⊑) $x \equiv x \in X \wedge (\forall\, y \in X.\ x \sqsubseteq y \longrightarrow y \sqsubseteq x)$ **for** r (**infix** ‹⊑› *50*)

context
fixes r :: $'a \Rightarrow\ 'a \Rightarrow$ *bool* (**infix** ‹⊑› *50*)
begin

lemma *extremalI*:
assumes $x \in X$ $\bigwedge y.\ y \in X \Longrightarrow x \sqsubseteq y \Longrightarrow y \sqsubseteq x$
shows *extremal* X (⊑) x
⟨*proof*⟩

lemma *extremalE*:
assumes *extremal* X (⊑) x
and $x \in X \Longrightarrow (\bigwedge y.\ y \in X \Longrightarrow x \sqsubseteq y \Longrightarrow y \sqsubseteq x) \Longrightarrow$ *thesis*
shows *thesis*
⟨*proof*⟩

lemma *extremalD*:

assumes *extremal* X $(\sqsubseteq)$ x **shows** $x \in X$ $y \in X \Longrightarrow x \sqsubseteq y \Longrightarrow y \sqsubseteq x$
⟨*proof*⟩

end

context
fixes *ir* (**infix** ‹⪯› *50*) **and** *r* (**infix** ‹⊑› *50*) **and** *I f*
assumes *mono*: *monotone-on* I $(\preceq)$ $(\sqsubseteq)$ f
begin

lemma *monotone-image-bound*:
assumes $X \subseteq I$ **and** $b \in I$ **and** *bound* X $(\preceq)$ b
shows *bound* $(f$ ‘ $X)$ $(\sqsubseteq)$ $(f\ b)$
⟨*proof*⟩

lemma *monotone-image-extreme*:
assumes *e*: *extreme* I $(\preceq)$ e
shows *extreme* $(f$ ‘ $I)$ $(\sqsubseteq)$ $(f\ e)$
⟨*proof*⟩

end

context
fixes *ir* :: $'i \Rightarrow\ 'i \Rightarrow$ *bool* (**infix** ‹⪯› *50*)
and *r* :: $'a \Rightarrow\ 'a \Rightarrow$ *bool* (**infix** ‹⊑› *50*)
and *f* **and** *A* **and** *e* **and** *I*
assumes *fIA*: f ‘ $I \subseteq A$
and *mono*: *monotone-on* I $(\preceq)$ $(\sqsubseteq)$ f
and *e*: *extreme* I $(\preceq)$ e
begin

lemma *monotone-extreme-imp-extreme-bound*:
extreme-bound A $(\sqsubseteq)$ $(f$ ‘ $I)$ $(f\ e)$
⟨*proof*⟩

lemma *monotone-extreme-extreme-boundI*:
$x = f\ e \Longrightarrow$ *extreme-bound* A $(\sqsubseteq)$ $(f$ ‘ $I)$ x
⟨*proof*⟩

end

2.2 Locales for Binary Relations

We now define basic properties of binary relations, in form of *locales* [13, 2].

2.2.1 Syntactic Locales

The following locales do not assume anything, but provide infix notations for relations.

locale *less-eq-syntax* =
  fixes *less-eq* :: $'a \Rightarrow 'a \Rightarrow bool$ (**infix** ‹⊑› *50*)

locale *less-syntax* =
  fixes *less* :: $'a \Rightarrow 'a \Rightarrow bool$ (**infix** ‹⊏› *50*)

locale *equivalence-syntax* =
  fixes *equiv* :: $'a \Rightarrow 'a \Rightarrow bool$ (**infix** ‹∼› *50*)
begin

abbreviation *equiv-class* (‹$[-]_\sim$›) **where** $[x]_\sim \equiv \{ y.\ x \sim y \}$

end

Next ones introduce abbreviations for dual etc. To avoid needless constants, one should be careful when declaring them as sublocales.

locale *less-eq-dualize* = *less-eq-syntax*
begin

abbreviation (*input*) *greater-eq* (**infix** ‹⊒› *50*) **where** $x \sqsupseteq y \equiv y \sqsubseteq x$

end

locale *less-eq-symmetrize* = *less-eq-dualize*
begin

abbreviation *sym* (**infix** ‹∼› *50*) **where** $(\sim) \equiv$ *sympartp* $(\sqsubseteq)$
abbreviation *equiv* (**infix** ‹(≃)› *50*) **where** $(\simeq) \equiv$ *equivpartp* $(\sqsubseteq)$

end

locale *less-eq-asymmetrize* = *less-eq-symmetrize*
begin

abbreviation *less* (**infix** ‹⊏› *50*) **where** $(\sqsubset) \equiv$ *asympartp* $(\sqsubseteq)$
abbreviation *greater* (**infix** ‹⊐› *50*) **where** $(\sqsupset) \equiv (\sqsubset)^-$

lemma *asym-cases*[*consumes 1*, *case-names asym sym*]:
  assumes $x \sqsubseteq y$ **and** $x \sqsubset y \Longrightarrow$ *thesis* **and** $x \sim y \Longrightarrow$ *thesis*
  shows *thesis*
  ⟨*proof*⟩

end

locale *less-dualize* = *less-syntax*
begin

abbreviation (*input*) *greater* (**infix** ‹⊐› *50*) **where** $x \sqsupset y \equiv y \sqsubset x$

end

locale *related-set* =
  fixes A :: *'a set* **and** *less-eq* :: *'a* ⇒ *'a* ⇒ *bool* (**infix** ‹⊑› *50*)

2.2.2 Basic Properties of Relations

In the following we define basic properties in form of locales.

Reflexivity restricted on a set:

locale *reflexive* = *related-set* +
  assumes *refl*[*intro*]: $x \in A \Longrightarrow x \sqsubseteq x$
begin

lemma *eq-implies*: $x = y \Longrightarrow x \in A \Longrightarrow x \sqsubseteq y$ ⟨*proof*⟩

lemma *reflexive-subset*: $B \subseteq A \Longrightarrow$ *reflexive* B (⊑) ⟨*proof*⟩

lemma *extreme-singleton*[*simp*]: $x \in A \Longrightarrow$ *extreme* $\{x\}$ (⊑) $y \longleftrightarrow x = y$ ⟨*proof*⟩

lemma *extreme-bound-singleton*: $x \in A \Longrightarrow$ *extreme-bound* A (⊑) $\{x\}$ x ⟨*proof*⟩

lemma *extreme-bound-cone*: $x \in A \Longrightarrow$ *extreme-bound* A (⊑) $\{a \in A.\ a \sqsubseteq x\}$ x
⟨*proof*⟩

end

lemmas *reflexiveI*[*intro*!] = *reflexive.intro*

lemma *reflexiveE*[*elim*]:
  assumes *reflexive* A r **and** $(\bigwedge x.\ x \in A \Longrightarrow r\ x\ x) \Longrightarrow$ *thesis* **shows** *thesis*
  ⟨*proof*⟩

lemma *reflexive-cong*:
  $(\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b) \Longrightarrow$ *reflexive* A $r \longleftrightarrow$ *reflexive*
A r'
  ⟨*proof*⟩

locale *irreflexive* = *related-set* A (⊏) **for** A **and** *less* (**infix** ‹⊏› *50*) +
  assumes *irrefl*: $x \in A \Longrightarrow \neg\ x \sqsubset x$
begin

lemma *irreflD*[*simp*]: $x \sqsubset x \Longrightarrow \neg x \in A$ ⟨*proof*⟩

lemma *implies-not-eq*: $x \sqsubset y \Longrightarrow x \in A \Longrightarrow x \neq y$ ⟨*proof*⟩

lemma *Restrp-irreflexive*: *irreflexive* *UNIV* ((⊏)↾A)
  ⟨*proof*⟩

lemma *irreflexive-subset*: $B \subseteq A \Longrightarrow$ *irreflexive* B $(\sqsubset)$ ⟨*proof*⟩

end

lemmas *irreflexiveI*[*intro*!] = *irreflexive.intro*

lemma *irreflexive-cong*:
$(\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b) \Longrightarrow$ *irreflexive* $A\ r \longleftrightarrow$ *irreflexive* $A\ r'$
⟨*proof*⟩

context *reflexive* **begin**

interpretation *less-eq-asymmetrize*⟨*proof*⟩

lemma *asympartp-irreflexive*: *irreflexive* A $(\sqsubset)$ ⟨*proof*⟩

end

locale *transitive* = *related-set* +
assumes *trans*[*trans*]: $x \sqsubseteq y \Longrightarrow y \sqsubseteq z \Longrightarrow x \in A \Longrightarrow y \in A \Longrightarrow z \in A \Longrightarrow x \sqsubseteq z$
begin

lemma *Restrp-transitive*: *transitive UNIV* $((\sqsubseteq)\restriction A)$
⟨*proof*⟩

lemma *bound-trans*[*trans*]: *bound* X $(\sqsubseteq)$ $b \Longrightarrow b \sqsubseteq c \Longrightarrow X \subseteq A \Longrightarrow b \in A \Longrightarrow c \in A \Longrightarrow$ *bound* X $(\sqsubseteq)$ c
⟨*proof*⟩

lemma *extreme-bound-mono*:
assumes *XY*: $\forall x{\in}X.\ \exists y{\in}Y.\ x \sqsubseteq y$ **and** *XA*: $X \subseteq A$ **and** *YA*: $Y \subseteq A$
and *sX*: *extreme-bound* A $(\sqsubseteq)$ X sX
and *sY*: *extreme-bound* A $(\sqsubseteq)$ Y sY
shows $sX \sqsubseteq sY$
⟨*proof*⟩

lemma *transitive-subset*:
assumes *BA*: $B \subseteq A$ **shows** *transitive* B $(\sqsubseteq)$
⟨*proof*⟩

lemma *asympartp-transitive*: *transitive* A (*asympartp* $(\sqsubseteq)$)
⟨*proof*⟩

lemma *reflclp-transitive*: *transitive* A $(\sqsubseteq)^{==}$
⟨*proof*⟩

The symmetric part is also transitive, but this is done in the later semi-attractive locale

end

lemmas *transitiveI* = *transitive.intro*

lemma *transitive-ball*[*code*]:
transitive A (⊑) ⟷ (∀ *x* ∈ *A*. ∀ *y* ∈ *A*. ∀ *z* ∈ *A*. *x* ⊑ *y* ⟶ *y* ⊑ *z* ⟶ *x* ⊑ *z*)
for *less-eq* (**infix** ‹⊑› *50*)
⟨*proof*⟩

lemma *transitive-cong*:
assumes *r*: ⋀*a b*. *a* ∈ *A* ⟹ *b* ∈ *A* ⟹ *r a b* ⟷ *r′ a b* **shows** *transitive A r* ⟷ *transitive A r′*
⟨*proof*⟩

lemma *transitive-empty*[*intro*!]: *transitive* {} *r* ⟨*proof*⟩

lemma *tranclp-transitive*: *transitive A* (*tranclp r*)
⟨*proof*⟩

locale *symmetric* = *related-set A* (∼) **for** *A* **and** *equiv* (**infix** ‹∼› *50*) +
assumes *sym*[*sym*]: *x* ∼ *y* ⟹ *x* ∈ *A* ⟹ *y* ∈ *A* ⟹ *y* ∼ *x*
begin

lemma *sym-iff*: *x* ∈ *A* ⟹ *y* ∈ *A* ⟹ *x* ∼ *y* ⟷ *y* ∼ *x*
⟨*proof*⟩

lemma *Restrp-symmetric*: *symmetric UNIV* ((∼)↾*A*)
⟨*proof*⟩

lemma *symmetric-subset*: *B* ⊆ *A* ⟹ *symmetric B* (∼)
⟨*proof*⟩

end

lemmas *symmetricI*[*intro*] = *symmetric.intro*

lemma *symmetric-cong*:
(⋀*a b*. *a* ∈ *A* ⟹ *b* ∈ *A* ⟹ *r a b* ⟷ *r′ a b*) ⟹ *symmetric A r* ⟷ *symmetric A r′*
⟨*proof*⟩

lemma *symmetric-empty*[*intro*!]: *symmetric* {} *r* ⟨*proof*⟩

global-interpretation *sympartp*: *symmetric UNIV sympartp r*
rewrites ⋀*r*. *r* ↾ *UNIV* ≡ *r*
and ⋀*x*. *x* ∈ *UNIV* ≡ *True*
and ⋀*P1*. (*True* ⟹ *P1*) ≡ *Trueprop P1*
and ⋀*P1 P2*. (*True* ⟹ *PROP P1* ⟹ *PROP P2*) ≡ (*PROP P1* ⟹ *PROP P2*)

⟨*proof*⟩

lemma *sympartp-symmetric*: *symmetric A* (*sympartp r*) ⟨*proof*⟩

locale *antisymmetric* = *related-set* +
assumes *antisym*: $x \sqsubseteq y \Longrightarrow y \sqsubseteq x \Longrightarrow x \in A \Longrightarrow y \in A \Longrightarrow x = y$
begin

interpretation *less-eq-symmetrize*⟨*proof*⟩

lemma *sym-iff-eq-refl*: $x \in A \Longrightarrow y \in A \Longrightarrow x \sim y \longleftrightarrow x = y \wedge y \sqsubseteq y$ ⟨*proof*⟩

lemma *equiv-iff-eq*[*simp*]: $x \in A \Longrightarrow y \in A \Longrightarrow x \simeq y \longleftrightarrow x = y$ ⟨*proof*⟩

lemma *extreme-unique*: $X \subseteq A \Longrightarrow$ *extreme X* $(\sqsubseteq)$ $x \Longrightarrow$ *extreme X* $(\sqsubseteq)$ $y \longleftrightarrow x = y$
⟨*proof*⟩

lemma *ex-extreme-iff-ex1*:
$X \subseteq A \Longrightarrow$ *Ex* (*extreme X* $(\sqsubseteq)$) $\longleftrightarrow$ *Ex1* (*extreme X* $(\sqsubseteq)$) ⟨*proof*⟩

lemma *ex-extreme-iff-the*:
$X \subseteq A \Longrightarrow$ *Ex* (*extreme X* $(\sqsubseteq)$) $\longleftrightarrow$ *extreme X* $(\sqsubseteq)$ (*The* (*extreme X* $(\sqsubseteq)$))
⟨*proof*⟩

lemma *eq-The-extreme*: $X \subseteq A \Longrightarrow$ *extreme X* $(\sqsubseteq)$ $x \Longrightarrow x =$ *The* (*extreme X* $(\sqsubseteq)$)
⟨*proof*⟩

lemma *Restrp-antisymmetric*: *antisymmetric UNIV* $((\sqsubseteq)\restriction A)$
⟨*proof*⟩

lemma *antisymmetric-subset*: $B \subseteq A \Longrightarrow$ *antisymmetric B* $(\sqsubseteq)$
⟨*proof*⟩

end

lemmas *antisymmetricI*[*intro*] = *antisymmetric.intro*

lemma *antisymmetric-cong*:
$(\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b) \Longrightarrow$ *antisymmetric A r* $\longleftrightarrow$ *antisymmetric A r'*
⟨*proof*⟩

lemma *antisymmetric-empty*[*intro*!]: *antisymmetric* {} *r* ⟨*proof*⟩

lemma *antisymmetric-union*:
fixes *less-eq* (**infix** ‹$\sqsubseteq$› *50*)
assumes *A*: *antisymmetric A* $(\sqsubseteq)$ **and** *B*: *antisymmetric B* $(\sqsubseteq)$

and AB: $\forall\, a \in A.\ \forall\, b \in B.\ a \sqsubseteq b \longrightarrow b \sqsubseteq a \longrightarrow a = b$
shows *antisymmetric* $(A \cup B)$ $(\sqsubseteq)$
⟨*proof*⟩

The following notion is new, generalizing antisymmetry and transitivity.

locale *semiattractive* = *related-set* +
assumes *attract*: $x \sqsubseteq y \Longrightarrow y \sqsubseteq x \Longrightarrow y \sqsubseteq z \Longrightarrow x \in A \Longrightarrow y \in A \Longrightarrow z \in A \Longrightarrow x \sqsubseteq z$
begin

interpretation *less-eq-symmetrize*⟨*proof*⟩

lemma *equiv-order-trans*[*trans*]:
assumes *xy*: $x \simeq y$ **and** *yz*: $y \sqsubseteq z$ **and** *x*: $x \in A$ **and** *y*: $y \in A$ **and** *z*: $z \in A$
shows $x \sqsubseteq z$
⟨*proof*⟩

lemma *equiv-transitive*: *transitive* A $(\simeq)$
⟨*proof*⟩

lemma *sym-order-trans*[*trans*]:
assumes *xy*: $x \sim y$ **and** *yz*: $y \sqsubseteq z$ **and** *x*: $x \in A$ **and** *y*: $y \in A$ **and** *z*: $z \in A$
shows $x \sqsubseteq z$
⟨*proof*⟩

interpretation *sym*: *transitive* A $(\sim)$
⟨*proof*⟩

lemmas *sym-transitive* = *sym.transitive-axioms*

lemma *extreme-bound-quasi-const*:
assumes *C*: $C \subseteq A$ **and** *x*: $x \in A$ **and** *C0*: $C \neq \{\}$ **and** *const*: $\forall\, y \in C.\ y \sim x$
shows *extreme-bound* A $(\sqsubseteq)$ C x
⟨*proof*⟩

lemma *extreme-bound-quasi-const-iff*:
assumes *C*: $C \subseteq A$ **and** *x*: $x \in A$ **and** *y*: $y \in A$ **and** *C0*: $C \neq \{\}$ **and** *const*: $\forall\, z \in C.\ z \sim x$
shows *extreme-bound* A $(\sqsubseteq)$ C $y \longleftrightarrow x \sim y$
⟨*proof*⟩

lemma *Restrp-semiattractive*: *semiattractive* *UNIV* $((\sqsubseteq)\restriction A)$
⟨*proof*⟩

lemma *semiattractive-subset*: $B \subseteq A \Longrightarrow$ *semiattractive* B $(\sqsubseteq)$
⟨*proof*⟩

end

lemmas *semiattractiveI* = *semiattractive.intro*

lemma *semiattractive-cong*:
  assumes *r*: $\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b$
  shows *semiattractive* $A\ r \longleftrightarrow$ *semiattractive* $A\ r'$ (**is** $?l \longleftrightarrow ?r$)
⟨*proof*⟩

lemma *semiattractive-empty*[*intro*!]: *semiattractive* {} *r*
  ⟨*proof*⟩

locale *attractive* = *semiattractive* +
  assumes *semiattractive* $A\ (\sqsubseteq)^-$
begin

interpretation *less-eq-symmetrize*⟨*proof*⟩

sublocale *dual*: *semiattractive* $A\ (\sqsubseteq)^-$
  rewrites $\bigwedge r.$ *sympartp* $(r \restriction A) \equiv$ *sympartp* $r \restriction A$
    and $\bigwedge r.$ *sympartp* (*sympartp* r) $\equiv$ *sympartp* r
    and *sympartp* $((\sqsubseteq) \restriction A)^- \equiv (\sim) \restriction A$
    and *sympartp* $(\sqsubseteq)^- \equiv (\sim)$
    and *equivpartp* $(\sqsubseteq)^- \equiv (\simeq)$
  ⟨*proof*⟩

lemma *order-equiv-trans*[*trans*]:
  assumes *xy*: $x \sqsubseteq y$ **and** *yz*: $y \simeq z$ **and** *x*: $x \in A$ **and** *y*: $y \in A$ **and** *z*: $z \in A$
  shows $x \sqsubseteq z$
  ⟨*proof*⟩

lemma *order-sym-trans*[*trans*]:
  assumes *xy*: $x \sqsubseteq y$ **and** *yz*: $y \sim z$ **and** *x*: $x \in A$ **and** *y*: $y \in A$ **and** *z*: $z \in A$
  shows $x \sqsubseteq z$
  ⟨*proof*⟩

lemma *extreme-bound-sym-trans*:
  assumes *XA*: $X \subseteq A$ **and** *Xx*: *extreme-bound* $A\ (\sqsubseteq)\ X\ x$
    and *xy*: $x \sim y$ **and** *yA*: $y \in A$
  shows *extreme-bound* $A\ (\sqsubseteq)\ X\ y$
⟨*proof*⟩

interpretation *Restrp*: *semiattractive UNIV* $(\sqsubseteq)\restriction A$ ⟨*proof*⟩
interpretation *dual.Restrp*: *semiattractive UNIV* $(\sqsubseteq)^-\restriction A$ ⟨*proof*⟩

lemma *Restrp-attractive*: *attractive UNIV* $((\sqsubseteq)\restriction A)$
  ⟨*proof*⟩

lemma *attractive-subset*: $B \subseteq A \Longrightarrow$ *attractive* $B\ (\sqsubseteq)$
  ⟨*proof*⟩

end

lemmas *attractiveI* = *attractive.intro*[*OF* - *attractive-axioms.intro*]

lemma *attractive-cong*:
  assumes *r*: $\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b$
  shows $attractive\ A\ r \longleftrightarrow attractive\ A\ r'$
  ⟨*proof*⟩

lemma *attractive-empty*[*intro*!]: *attractive* {} *r*
  ⟨*proof*⟩

context *antisymmetric* **begin**

sublocale *attractive*
  ⟨*proof*⟩

end

context *transitive* **begin**

sublocale *attractive*
  rewrites $\bigwedge r.\ sympartp\ (r \upharpoonright A) \equiv sympartp\ r \upharpoonright A$
    and $\bigwedge r.\ sympartp\ (sympartp\ r) \equiv sympartp\ r$
    and $sympartp\ (\sqsubseteq)^{-} \equiv sympartp\ (\sqsubseteq)$
    and $(sympartp\ (\sqsubseteq))^{-} \equiv sympartp\ (\sqsubseteq)$
    and $(sympartp\ (\sqsubseteq) \upharpoonright A)^{-} \equiv sympartp\ (\sqsubseteq) \upharpoonright A$
    and $asympartp\ (asympartp\ (\sqsubseteq)) = asympartp\ (\sqsubseteq)$
    and $asympartp\ (sympartp\ (\sqsubseteq)) = bot$
    and $asympartp\ (\sqsubseteq) \upharpoonright A = asympartp\ ((\sqsubseteq) \upharpoonright A)$
  ⟨*proof*⟩

end

2.3 Combined Properties

Some combinations of the above basic properties are given names.

locale *asymmetric* = *related-set* *A* (⊏) **for** *A* **and** *less* (**infix** ‹⊏› *50*) +
  assumes *asym*: $x \sqsubset y \Longrightarrow y \sqsubset x \Longrightarrow x \in A \Longrightarrow y \in A \Longrightarrow False$
begin

sublocale *irreflexive*
  ⟨*proof*⟩

lemma *antisymmetric-axioms*: *antisymmetric* *A* (⊏)
  ⟨*proof*⟩

lemma *Restrp-asymmetric*: *asymmetric* *UNIV* ((⊏)↾*A*)
  ⟨*proof*⟩

lemma *asymmetric-subset*: $B \subseteq A \Longrightarrow$ *asymmetric* B ($\sqsubset$)
⟨*proof*⟩

end

lemmas *asymmetricI* = *asymmetric.intro*

lemma *asymmetric-iff-irreflexive-antisymmetric*:
fixes *less* (**infix** ‹$\sqsubset$› *50*)
shows *asymmetric* A ($\sqsubset$) $\longleftrightarrow$ *irreflexive* A ($\sqsubset$) $\wedge$ *antisymmetric* A ($\sqsubset$) (**is** *?l* $\longleftrightarrow$ *?r*)
⟨*proof*⟩

lemma *asymmetric-cong*:
assumes r: $\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b$
shows *asymmetric* $A\ r \longleftrightarrow$ *asymmetric* $A\ r'$
⟨*proof*⟩

lemma *asymmetric-empty*: *asymmetric* {} r
⟨*proof*⟩

locale *quasi-ordered-set* = *reflexive* + *transitive*
begin

lemma *quasi-ordered-subset*: $B \subseteq A \Longrightarrow$ *quasi-ordered-set* B ($\sqsubseteq$)
⟨*proof*⟩

end

lemmas *quasi-ordered-setI* = *quasi-ordered-set.intro*

lemma *quasi-ordered-set-cong*:
assumes r: $\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b$
shows *quasi-ordered-set* $A\ r \longleftrightarrow$ *quasi-ordered-set* $A\ r'$
⟨*proof*⟩

lemma *quasi-ordered-set-empty*[*intro*!]: *quasi-ordered-set* {} r
⟨*proof*⟩

lemma *rtranclp-quasi-ordered*: *quasi-ordered-set* A (*rtranclp* r)
⟨*proof*⟩

locale *near-ordered-set* = *antisymmetric* + *transitive*
begin

interpretation *Restrp*: *antisymmetric UNIV* ($\sqsubseteq$)$\restriction A$ ⟨*proof*⟩
interpretation *Restrp*: *transitive UNIV* ($\sqsubseteq$)$\restriction A$ ⟨*proof*⟩

lemma *Restrp-near-order*: *near-ordered-set UNIV* $((\sqsubseteq)\restriction A)$⟨*proof*⟩

lemma *near-ordered-subset*: $B \subseteq A \Longrightarrow$ *near-ordered-set* B $(\sqsubseteq)$
⟨*proof*⟩

end

lemmas *near-ordered-setI* = *near-ordered-set.intro*

lemma *near-ordered-set-cong*:
assumes r: $\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b$
shows *near-ordered-set* $A\ r \longleftrightarrow$ *near-ordered-set* $A\ r'$
⟨*proof*⟩

lemma *near-ordered-set-empty*[*intro*!]: *near-ordered-set* $\{\}\ r$
⟨*proof*⟩

locale *pseudo-ordered-set* = *reflexive* + *antisymmetric*
begin

interpretation *less-eq-symmetrize*⟨*proof*⟩

lemma *sym-eq*[*simp*]: $x \in A \Longrightarrow y \in A \Longrightarrow x \sim y \longleftrightarrow x = y$
⟨*proof*⟩

lemma *extreme-bound-singleton-eq*[*simp*]: $x \in A \Longrightarrow$ *extreme-bound* A $(\sqsubseteq)$ $\{x\}$ y $\longleftrightarrow x = y$
⟨*proof*⟩

lemma *eq-iff*: $x \in A \Longrightarrow y \in A \Longrightarrow x = y \longleftrightarrow x \sqsubseteq y \wedge y \sqsubseteq x$ ⟨*proof*⟩

lemma *extreme-order-iff-eq*: $e \in A \Longrightarrow$ *extreme* $\{x \in A.\ x \sqsubseteq e\}$ $(\sqsubseteq)$ $s \longleftrightarrow e = s$
⟨*proof*⟩

lemma *pseudo-ordered-subset*: $B \subseteq A \Longrightarrow$ *pseudo-ordered-set* B $(\sqsubseteq)$
⟨*proof*⟩

end

lemmas *pseudo-ordered-setI* = *pseudo-ordered-set.intro*

lemma *pseudo-ordered-set-cong*:
assumes r: $\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b$
shows *pseudo-ordered-set* $A\ r \longleftrightarrow$ *pseudo-ordered-set* $A\ r'$
⟨*proof*⟩

lemma *pseudo-ordered-set-empty*[*intro*!]: *pseudo-ordered-set* $\{\}\ r$
⟨*proof*⟩

locale *partially-ordered-set* = *reflexive* + *antisymmetric* + *transitive*
begin

sublocale *pseudo-ordered-set* + *quasi-ordered-set* + *near-ordered-set* ⟨*proof*⟩

lemma *partially-ordered-subset*: $B \subseteq A \Longrightarrow$ *partially-ordered-set* B $(\sqsubseteq)$
⟨*proof*⟩

end

lemmas *partially-ordered-setI* = *partially-ordered-set.intro*

lemma *partially-ordered-set-cong*:
assumes *r*: $\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b$
shows *partially-ordered-set* $A\ r \longleftrightarrow$ *partially-ordered-set* $A\ r'$
⟨*proof*⟩

lemma *partially-ordered-set-empty*[*intro*!]: *partially-ordered-set* $\{\}\ r$
⟨*proof*⟩

locale *strict-ordered-set* = *irreflexive* + *transitive* A $(\sqsubset)$
begin

sublocale *asymmetric*
⟨*proof*⟩

lemma *near-ordered-set-axioms*: *near-ordered-set* A $(\sqsubset)$
⟨*proof*⟩

interpretation *Restrp*: *asymmetric* *UNIV* $(\sqsubset)\restriction A$ ⟨*proof*⟩
interpretation *Restrp*: *transitive* *UNIV* $(\sqsubset)\restriction A$ ⟨*proof*⟩

lemma *Restrp-strict-order*: *strict-ordered-set* *UNIV* $((\sqsubset)\restriction A)$⟨*proof*⟩

lemma *strict-ordered-subset*: $B \subseteq A \Longrightarrow$ *strict-ordered-set* B $(\sqsubset)$
⟨*proof*⟩

end

lemmas *strict-ordered-setI* = *strict-ordered-set.intro*

lemma *strict-ordered-set-cong*:
assumes *r*: $\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b$
shows *strict-ordered-set* $A\ r \longleftrightarrow$ *strict-ordered-set* $A\ r'$
⟨*proof*⟩

lemma *strict-ordered-set-empty*[*intro*!]: *strict-ordered-set* $\{\}\ r$
⟨*proof*⟩

locale *tolerance* = *symmetric* + *reflexive A* (∼)
begin

lemma *tolerance-subset*: $B \subseteq A \Longrightarrow$ *tolerance B* (∼)
⟨*proof*⟩

end

lemmas *toleranceI* = *tolerance.intro*

lemma *tolerance-cong*:
assumes r: $\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b$
shows *tolerance A r* ⟷ *tolerance A r′*
⟨*proof*⟩

lemma *tolerance-empty*[*intro*!]: *tolerance* {} *r* ⟨*proof*⟩

global-interpretation *equiv*: *tolerance UNIV equivpartp r*
rewrites $\bigwedge r.\ r \restriction UNIV \equiv r$
and $\bigwedge x.\ x \in UNIV \equiv True$
and $\bigwedge P1.\ (True \Longrightarrow P1) \equiv Trueprop\ P1$
and $\bigwedge P1\ P2.\ (True \Longrightarrow PROP\ P1 \Longrightarrow PROP\ P2) \equiv (PROP\ P1 \Longrightarrow PROP\ P2)$
⟨*proof*⟩

locale *partial-equivalence* = *symmetric* +
assumes *transitive A* (∼)
begin

sublocale *transitive A* (∼)
rewrites *sympartp* (∼)↾*A* ≡ (∼)↾*A*
and *sympartp* ((∼)↾*A*) ≡ (∼)↾*A*
⟨*proof*⟩

lemma *partial-equivalence-subset*: $B \subseteq A \Longrightarrow$ *partial-equivalence B* (∼)
⟨*proof*⟩

end

lemmas *partial-equivalenceI* = *partial-equivalence.intro*[*OF - partial-equivalence-axioms.intro*]

lemma *partial-equivalence-cong*:
assumes r: $\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b$
shows *partial-equivalence A r* ⟷ *partial-equivalence A r′*
⟨*proof*⟩

lemma *partial-equivalence-empty*[*intro*!]: *partial-equivalence* {} *r*
⟨*proof*⟩

locale *equivalence* = *symmetric* + *reflexive* A (∼) + *transitive* A (∼)
begin

sublocale *tolerance* + *partial-equivalence* + *quasi-ordered-set* A (∼)⟨*proof*⟩

lemma *equivalence-subset*: $B \subseteq A \Longrightarrow$ *equivalence* B (∼)
⟨*proof*⟩

end

lemmas *equivalenceI* = *equivalence.intro*

lemma *equivalence-cong*:
assumes r: $\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b$
shows *equivalence* $A\ r \longleftrightarrow$ *equivalence* $A\ r'$
⟨*proof*⟩

Some combinations lead to uninteresting relations.

context
fixes r :: $'a \Rightarrow 'a \Rightarrow$ *bool* (**infix** ‹⋈› *50*)
begin

proposition *reflexive-irreflexive-is-empty*:
assumes r: *reflexive* A (⋈) **and** *ir*: *irreflexive* A (⋈)
shows $A = \{\}$
⟨*proof*⟩

proposition *symmetric-antisymmetric-imp-eq*:
assumes s: *symmetric* A (⋈) **and** *as*: *antisymmetric* A (⋈)
shows $(\bowtie)\restriction A \le (=)$
⟨*proof*⟩

proposition *nontolerance*:
shows *irreflexive* A (⋈) ∧ *symmetric* A (⋈) ⟷ *tolerance* A $(\lambda x\ y.\ \neg\ x \bowtie y)$
⟨*proof*⟩

proposition *irreflexive-transitive-symmetric-is-empty*:
assumes *irr*: *irreflexive* A (⋈) **and** *tr*: *transitive* A (⋈) **and** *sym*: *symmetric* A (⋈)
shows $(\bowtie)\restriction A$ = *bot*
⟨*proof*⟩

end

2.4 Totality

locale *semiconnex* = *related-set* - (⊏) + *less-syntax* +
assumes *semiconnex*: $x \in A \Longrightarrow y \in A \Longrightarrow x \sqsubset y \lor x = y \lor y \sqsubset x$
begin

lemma *cases*[*consumes 2*, *case-names less eq greater*]:
assumes $x \in A$ **and** $y \in A$ **and** $x \sqsubset y \Longrightarrow P$ **and** $x = y \Longrightarrow P$ **and** $y \sqsubset x \Longrightarrow P$
shows P ⟨*proof*⟩

lemma *neqE*:
assumes $x \in A$ **and** $y \in A$
shows $x \neq y \Longrightarrow (x \sqsubset y \Longrightarrow P) \Longrightarrow (y \sqsubset x \Longrightarrow P) \Longrightarrow P$
⟨*proof*⟩

lemma *semiconnex-subset*: $B \subseteq A \Longrightarrow$ *semiconnex* B $(\sqsubset)$
⟨*proof*⟩

end

lemmas *semiconnexI*[*intro*] = *semiconnex.intro*

Totality is negated antisymmetry [19, Proposition 2.2.4].

proposition *semiconnex-iff-neg-antisymmetric*:
fixes *less* (**infix** ‹$\sqsubset$› *50*)
shows *semiconnex* A $(\sqsubset) \longleftrightarrow$ *antisymmetric* A $(\lambda x\ y.\ \neg\ x \sqsubset y)$ (**is** *?l* $\longleftrightarrow$ *?r*)
⟨*proof*⟩

lemma *semiconnex-cong*:
assumes r: $\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b$
shows *semiconnex* $A\ r \longleftrightarrow$ *semiconnex* $A\ r'$
⟨*proof*⟩

locale *semiconnex-irreflexive* = *semiconnex* + *irreflexive*
begin

lemma *neq-iff*: $x \in A \Longrightarrow y \in A \Longrightarrow x \neq y \longleftrightarrow x \sqsubset y \lor y \sqsubset x$ ⟨*proof*⟩

lemma *semiconnex-irreflexive-subset*: $B \subseteq A \Longrightarrow$ *semiconnex-irreflexive* B $(\sqsubset)$
⟨*proof*⟩

end

lemmas *semiconnex-irreflexiveI* = *semiconnex-irreflexive.intro*

lemma *semiconnex-irreflexive-cong*:
assumes r: $\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b$
shows *semiconnex-irreflexive* $A\ r \longleftrightarrow$ *semiconnex-irreflexive* $A\ r'$
⟨*proof*⟩

locale *connex* = *related-set* +
assumes *comparable*: $x \in A \Longrightarrow y \in A \Longrightarrow x \sqsubseteq y \lor y \sqsubseteq x$
begin

interpretation *less-eq-asymmetrize*⟨*proof*⟩

sublocale *reflexive* ⟨*proof*⟩

lemma *comparable-cases*[*consumes 2*, *case-names le ge*]:
assumes $x \in A$ **and** $y \in A$ **and** $x \sqsubseteq y \Longrightarrow P$ **and** $y \sqsubseteq x \Longrightarrow P$ **shows** P
⟨*proof*⟩

lemma *comparable-three-cases*[*consumes 2*, *case-names less eq greater*]:
assumes $x \in A$ **and** $y \in A$ **and** $x \sqsubset y \Longrightarrow P$ **and** $x \sim y \Longrightarrow P$ **and** $y \sqsubset x \Longrightarrow P$ **shows** P
⟨*proof*⟩

lemma
assumes *x*: $x \in A$ **and** *y*: $y \in A$
shows *not-iff-asym*: $\neg x \sqsubseteq y \longleftrightarrow y \sqsubset x$
and *not-asym-iff*: $\neg x \sqsubset y \longleftrightarrow y \sqsubseteq x$
⟨*proof*⟩

lemma *connex-subset*: $B \subseteq A \Longrightarrow$ *connex* B $(\sqsubseteq)$
⟨*proof*⟩

interpretation *less-eq-asymmetrize*⟨*proof*⟩

end

lemmas *connexI*[*intro*] = *connex.intro*

lemmas *connexE* = *connex.comparable-cases*

lemma *connex-empty*: *connex* {} *A* ⟨*proof*⟩

context
fixes *less-eq* :: $'a \Rightarrow 'a \Rightarrow$ *bool* (**infix** ‹⊑› *50*)
begin

lemma *connex-iff-semiconnex-reflexive*: *connex* A $(\sqsubseteq) \longleftrightarrow$ *semiconnex* A $(\sqsubseteq) \wedge$ *reflexive* A $(\sqsubseteq)$
(**is** $?c \longleftrightarrow ?t \wedge ?r$)
⟨*proof*⟩

lemma *chain-connect*: *Complete-Partial-Order.chain* r $A \equiv$ *connex* A r
⟨*proof*⟩

lemma *connex-union*:
assumes *connex* X $(\sqsubseteq)$ **and** *connex* Y $(\sqsubseteq)$ **and** $\forall x \in X.\ \forall y \in Y.\ x \sqsubseteq y \vee y \sqsubseteq x$
shows *connex* $(X \cup Y)$ $(\sqsubseteq)$
⟨*proof*⟩

end

lemma *connex-cong*:
  assumes r: $\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b$
  shows $connex\ A\ r \longleftrightarrow connex\ A\ r'$
  ⟨*proof*⟩

locale *total-pseudo-ordered-set* = *connex* + *antisymmetric*
begin

sublocale *pseudo-ordered-set* ⟨*proof*⟩

lemma *not-weak-iff*:
  assumes x: $x \in A$ **and** y: $y \in A$ **shows** $\neg\ y \sqsubseteq x \longleftrightarrow x \sqsubseteq y \wedge x \neq y$
⟨*proof*⟩

lemma *total-pseudo-ordered-subset*: $B \subseteq A \Longrightarrow$ *total-pseudo-ordered-set* B $(\sqsubseteq)$
  ⟨*proof*⟩

interpretation *less-eq-asymmetrize*⟨*proof*⟩

interpretation *asympartp*: *semiconnex-irreflexive* A $(\sqsubset)$
⟨*proof*⟩

lemmas *asympartp-semiconnex* = *asympartp.semiconnex-axioms*
lemmas *asympartp-semiconnex-irreflexive* = *asympartp.semiconnex-irreflexive-axioms*

end

lemmas *total-pseudo-ordered-setI* = *total-pseudo-ordered-set.intro*

lemma *total-pseudo-ordered-set-cong*:
  assumes r: $\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b$
  shows *total-pseudo-ordered-set* $A\ r \longleftrightarrow$ *total-pseudo-ordered-set* $A\ r'$
  ⟨*proof*⟩

locale *total-quasi-ordered-set* = *connex* + *transitive*
begin

sublocale *quasi-ordered-set* ⟨*proof*⟩

lemma *total-quasi-ordered-subset*: $B \subseteq A \Longrightarrow$ *total-quasi-ordered-set* B $(\sqsubseteq)$
  ⟨*proof*⟩

end

lemmas *total-quasi-ordered-setI* = *total-quasi-ordered-set.intro*

lemma *total-quasi-ordered-set-cong*:
  assumes r: $\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b$
  shows *total-quasi-ordered-set* $A\ r \longleftrightarrow$ *total-quasi-ordered-set* $A\ r'$
  ⟨*proof*⟩

locale *total-ordered-set* = *total-quasi-ordered-set* + *antisymmetric*
begin

sublocale *partially-ordered-set* + *total-pseudo-ordered-set* ⟨*proof*⟩

lemma *total-ordered-subset*: $B \subseteq A \Longrightarrow$ *total-ordered-set* B $(\sqsubseteq)$
  ⟨*proof*⟩

lemma *weak-semiconnex*: *semiconnex* A $(\sqsubseteq)$
  ⟨*proof*⟩

interpretation *less-eq-asymmetrize*⟨*proof*⟩

end

lemmas *total-ordered-setI* = *total-ordered-set.intro*[*OF total-quasi-ordered-setI*]

lemma *total-ordered-set-cong*:
  assumes r: $\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b$
  shows *total-ordered-set* $A\ r \longleftrightarrow$ *total-ordered-set* $A\ r'$
  ⟨*proof*⟩

lemma *monotone-connex-image*:
  fixes *ir* (**infix** ‹$\preceq$› *50*) **and** r (**infix** ‹$\sqsubseteq$› *50*)
  assumes *mono*: *monotone-on* I $(\preceq)$ $(\sqsubseteq)$ f **and** *connex*: *connex* I $(\preceq)$
  shows *connex* $(f$ ‘ $I)$ $(\sqsubseteq)$
⟨*proof*⟩

2.5 Order Pairs

We pair a relation (weak part) with a well-behaving “strict” part. Here no assumption is put on the “weak” part.

locale *compatible-ordering* =
  related-set + *irreflexive* +
  assumes *strict-implies-weak*: $x \sqsubset y \Longrightarrow x \in A \Longrightarrow y \in A \Longrightarrow x \sqsubseteq y$
  assumes *weak-strict-trans*[*trans*]: $x \sqsubseteq y \Longrightarrow y \sqsubset z \Longrightarrow x \in A \Longrightarrow y \in A \Longrightarrow z \in A \Longrightarrow x \sqsubset z$
  assumes *strict-weak-trans*[*trans*]: $x \sqsubset y \Longrightarrow y \sqsubseteq z \Longrightarrow x \in A \Longrightarrow y \in A \Longrightarrow z \in A \Longrightarrow x \sqsubset z$
begin

The following sequence of declarations are in order to obtain fact names in a manner similar to the Isabelle/HOL facts of orders.

The strict part is necessarily transitive.

sublocale *strict*: *transitive* A $(\sqsubset)$
⟨*proof*⟩

sublocale *strict-ordered-set* A $(\sqsubset)$ ⟨*proof*⟩

thm *strict.trans asym irrefl*

lemma *Restrp-compatible-ordering*: *compatible-ordering UNIV* $((\sqsubseteq)\restriction A)$ $((\sqsubset)\restriction A)$
⟨*proof*⟩

lemma *strict-implies-not-weak*: $x \sqsubset y \Longrightarrow x \in A \Longrightarrow y \in A \Longrightarrow \neg\ y \sqsubseteq x$
⟨*proof*⟩

lemma *weak-implies-not-strict*:
assumes *xy*: $x \sqsubseteq y$ **and** [*simp*]: $x \in A$ $y \in A$
shows $\neg y \sqsubset x$
⟨*proof*⟩

lemma *compatible-ordering-subset*: **assumes** $X \subseteq A$ **shows** *compatible-ordering* X $(\sqsubseteq)$ $(\sqsubset)$
⟨*proof*⟩

end

context *transitive* **begin**

interpretation *less-eq-asymmetrize*⟨*proof*⟩

lemma *asym-trans*[*trans*]:
shows $x \sqsubset y \Longrightarrow y \sqsubseteq z \Longrightarrow x \in A \Longrightarrow y \in A \Longrightarrow z \in A \Longrightarrow x \sqsubset z$
and $x \sqsubseteq y \Longrightarrow y \sqsubset z \Longrightarrow x \in A \Longrightarrow y \in A \Longrightarrow z \in A \Longrightarrow x \sqsubset z$
⟨*proof*⟩

lemma *asympartp-compatible-ordering*: *compatible-ordering* A $(\sqsubseteq)$ $(\sqsubset)$
⟨*proof*⟩

end

locale *reflexive-ordering* = *reflexive* + *compatible-ordering*

locale *reflexive-attractive-ordering* = *reflexive-ordering* + *attractive*

locale *pseudo-ordering* = *pseudo-ordered-set* + *compatible-ordering*
begin

sublocale *reflexive-attractive-ordering*⟨*proof*⟩

end

locale *quasi-ordering* = *quasi-ordered-set* + *compatible-ordering*
begin

sublocale *reflexive-attractive-ordering*⟨*proof*⟩

lemma *quasi-ordering-subset*: **assumes** $X \subseteq A$ **shows** *quasi-ordering* X (⊑) (⊏)
⟨*proof*⟩

end

context *quasi-ordered-set* **begin**

interpretation *less-eq-asymmetrize*⟨*proof*⟩

lemma *asympartp-quasi-ordering*: *quasi-ordering* A (⊑) (⊏)
⟨*proof*⟩

end

locale *partial-ordering* = *partially-ordered-set* + *compatible-ordering*
begin

sublocale *quasi-ordering* + *pseudo-ordering*⟨*proof*⟩

lemma *partial-ordering-subset*: **assumes** $X \subseteq A$ **shows** *partial-ordering* X (⊑) (⊏)
⟨*proof*⟩

end

context *partially-ordered-set* **begin**

interpretation *less-eq-asymmetrize*⟨*proof*⟩

lemma *asympartp-partial-ordering*: *partial-ordering* A (⊑) (⊏)
⟨*proof*⟩

end

locale *total-quasi-ordering* = *total-quasi-ordered-set* + *compatible-ordering*
begin

sublocale *quasi-ordering*⟨*proof*⟩

lemma *total-quasi-ordering-subset*: **assumes** $X \subseteq A$ **shows** *total-quasi-ordering* X (⊑) (⊏)
⟨*proof*⟩

end

context *total-quasi-ordered-set* **begin**

interpretation *less-eq-asymmetrize*⟨*proof*⟩

lemma *asympartp-total-quasi-ordering*: *total-quasi-ordering* A (⊑) (⊏)
⟨*proof*⟩

end

Fixing the definition of the strict part is very common, though it looks restrictive to the author.

locale *strict-quasi-ordering* = *quasi-ordered-set* + *less-syntax* +
assumes *strict-iff*: $x \in A \Longrightarrow y \in A \Longrightarrow x \sqsubset y \longleftrightarrow x \sqsubseteq y \wedge \neg y \sqsubseteq x$
begin

sublocale *compatible-ordering*
⟨*proof*⟩

end

locale *strict-partial-ordering* = *strict-quasi-ordering* + *antisymmetric*
begin

sublocale *partial-ordering*⟨*proof*⟩

lemma *strict-iff-neq*: $x \in A \Longrightarrow y \in A \Longrightarrow x \sqsubset y \longleftrightarrow x \sqsubseteq y \wedge x \neq y$
⟨*proof*⟩

end

locale *total-ordering* = *reflexive* + *compatible-ordering* + *semiconnex* A (⊏)
begin

sublocale *semiconnex-irreflexive* ⟨*proof*⟩

sublocale *connex*
⟨*proof*⟩

lemma *not-weak*:
assumes $x \in A$ **and** $y \in A$ **shows** $\neg\ x \sqsubseteq y \longleftrightarrow y \sqsubset x$
⟨*proof*⟩

lemma *not-strict*: $x \in A \Longrightarrow y \in A \Longrightarrow \neg\ x \sqsubset y \longleftrightarrow y \sqsubseteq x$
⟨*proof*⟩

sublocale *strict-partial-ordering*
⟨*proof*⟩

sublocale *total-ordered-set*⟨*proof*⟩

context
fixes s
assumes s: $\forall x \in A.\ x \sqsubset s \longrightarrow (\exists z \in A.\ x \sqsubset z \wedge z \sqsubset s)$ **and** sA: $s \in A$
begin

lemma *dense-weakI*:
assumes *bound*: $\bigwedge x.\ x \sqsubset s \Longrightarrow x \in A \Longrightarrow x \sqsubseteq y$ **and** yA: $y \in A$
shows $s \sqsubseteq y$
⟨*proof*⟩

lemma *dense-bound-iff*:
assumes bA: $b \in A$ **shows** *bound* $\{x{\in}A.\ x \sqsubset s\}$ $(\sqsubseteq)$ $b \longleftrightarrow s \sqsubseteq b$
⟨*proof*⟩

lemma *dense-extreme-bound*:
extreme-bound A $(\sqsubseteq)$ $\{x \in A.\ x \sqsubset s\}$ s
⟨*proof*⟩

end

lemma *ordinal-cases*[*consumes 1*, *case-names suc lim*]:
assumes aA: $a \in A$
and *suc*: $\bigwedge p.$ *extreme* $\{x \in A.\ x \sqsubset a\}$ $(\sqsubseteq)$ $p \Longrightarrow$ *thesis*
and *lim*: *extreme-bound* A $(\sqsubseteq)$ $\{x \in A.\ x \sqsubset a\}$ $a \Longrightarrow$ *thesis*
shows *thesis*
⟨*proof*⟩

end

context *total-ordered-set* **begin**

interpretation *less-eq-asymmetrize*⟨*proof*⟩

lemma *asympartp-total-ordering*: *total-ordering* A $(\sqsubseteq)$ $(\sqsubset)$
⟨*proof*⟩

end

2.6 Functions

definition *pointwise* $I\ r\ f\ g \equiv \forall i \in I.\ r\ (f\ i)\ (g\ i)$

lemmas *pointwiseI* = *pointwise-def*[*unfolded atomize-eq*, *THEN iffD2*, *rule-format*]

lemmas *pointwiseD*[*simp*] = *pointwise-def*[*unfolded atomize-eq*, *THEN iffD1*, *rule-format*]

lemma *pointwise-cong*:
  assumes $r = r'$ $\bigwedge i.\ i \in I \Longrightarrow f\ i = f'\ i$ $\bigwedge i.\ i \in I \Longrightarrow g\ i = g'\ i$
  shows *pointwise* $I\ r\ f\ g$ = *pointwise* $I\ r'\ f'\ g'$
  ⟨*proof*⟩

lemma *pointwise-empty*[*simp*]: *pointwise* $\{\} = \top$ ⟨*proof*⟩

lemma *dual-pointwise*[*simp*]: (*pointwise* $I\ r)^{-}$ = *pointwise* $I\ r^{-}$
  ⟨*proof*⟩

lemma *pointwise-dual*: *pointwise* $I\ r^{-}\ f\ g \Longrightarrow$ *pointwise* $I\ r\ g\ f$ ⟨*proof*⟩

lemma *pointwise-un*: *pointwise* $(I \cup J)\ r$ = *pointwise* $I\ r \sqcap$ *pointwise* $J\ r$
  ⟨*proof*⟩

lemma *pointwise-unI*[*intro*!]: *pointwise* $I\ r\ f\ g \Longrightarrow$ *pointwise* $J\ r\ f\ g \Longrightarrow$ *pointwise* $(I \cup J)\ r\ f\ g$
  ⟨*proof*⟩

lemma *pointwise-bound*: *bound* F (*pointwise* $I\ r$) $f \longleftrightarrow (\forall i \in I.$ *bound* $\{f\ i\ |.\ f \in F\}\ r\ (f\ i))$
  ⟨*proof*⟩

lemma *pointwise-extreme*:
  shows *extreme* F (*pointwise* $X\ r$) $e \longleftrightarrow e \in F \wedge (\forall x \in X.$ *extreme* $\{f\ x\ |.\ f \in F\}\ r\ (e\ x))$
  ⟨*proof*⟩

lemma *pointwise-extreme-bound*:
  fixes r (**infix** ‹$\sqsubseteq$› *50*)
  assumes F: $F \subseteq \{f.\ f\ `\ X \subseteq A\}$
  shows *extreme-bound* $\{f.\ f\ `\ X \subseteq A\}$ (*pointwise* $X\ (\sqsubseteq)$) $F\ s \longleftrightarrow$
    $(\forall x \in X.$ *extreme-bound* $A\ (\sqsubseteq)\ \{f\ x\ |.\ f \in F\}\ (s\ x))$ (**is** $?p \longleftrightarrow ?a$)
⟨*proof*⟩

lemma *dual-pointwise-extreme-bound*:
  extreme-bound FA (*pointwise* $X\ r)^{-}\ F$ = *extreme-bound FA* (*pointwise* $X\ r^{-}$) F
  ⟨*proof*⟩

lemma *pointwise-monotone-on*:
  fixes *less-eq* (**infix** ‹$\sqsubseteq$› *50*) **and** *prec-eq* (**infix** ‹$\preceq$› *50*)
  shows *monotone-on* $I\ (\preceq)$ (*pointwise* $A\ (\sqsubseteq)$) $f \longleftrightarrow$
   $(\forall a \in A.$ *monotone-on* $I\ (\preceq)\ (\sqsubseteq)\ (\lambda i.\ f\ i\ a))$ (**is** $?l \longleftrightarrow ?r$)
⟨*proof*⟩

lemmas *pointwise-monotone* = *pointwise-monotone-on*[*of UNIV*]

lemma (**in** *reflexive*) *pointwise-reflexive*: *reflexive* $\{f.\ f\ `\ I \subseteq A\}$ (*pointwise* $I\ (\sqsubseteq)$)
  ⟨*proof*⟩

lemma (**in** *irreflexive*) *pointwise-irreflexive*:
assumes *I0*: $I \neq \{\}$ **shows** *irreflexive* $\{f.\ f\ `\ I \subseteq A\}$ (*pointwise* I $(\sqsubset)$)
⟨*proof*⟩

lemma (**in** *semiattractive*) *pointwise-semiattractive*: *semiattractive* $\{f.\ f\ `\ I \subseteq A\}$ (*pointwise* I $(\sqsubseteq)$)
⟨*proof*⟩

lemma (**in** *attractive*) *pointwise-attractive*: *attractive* $\{f.\ f\ `\ I \subseteq A\}$ (*pointwise* I $(\sqsubseteq)$)
⟨*proof*⟩

Antisymmetry will not be preserved by pointwise extension over restricted domain.

lemma (**in** *antisymmetric*) *pointwise-antisymmetric*:
antisymmetric $\{f.\ f\ `\ I \subseteq A\}$ (*pointwise* I $(\sqsubseteq)$)
⟨*proof*⟩

lemma (**in** *transitive*) *pointwise-transitive*: *transitive* $\{f.\ f\ `\ I \subseteq A\}$ (*pointwise* I $(\sqsubseteq)$)
⟨*proof*⟩

lemma (**in** *quasi-ordered-set*) *pointwise-quasi-order*:
quasi-ordered-set $\{f.\ f\ `\ I \subseteq A\}$ (*pointwise* I $(\sqsubseteq)$)
⟨*proof*⟩

lemma (**in** *compatible-ordering*) *pointwise-compatible-ordering*:
assumes *I0*: $I \neq \{\}$
shows *compatible-ordering* $\{f.\ f\ `\ I \subseteq A\}$ (*pointwise* I $(\sqsubseteq)$) (*pointwise* I $(\sqsubset)$)
⟨*proof*⟩

2.7 Relating to Classes

In Isabelle 2020, we should declare sublocales in class before declaring dual sublocales, since otherwise facts would be prefixed by "dual.dual."

context *ord* **begin**

abbreviation *least* **where** *least* $X \equiv$ *extreme* X $(\lambda x\ y.\ y \leq x)$

abbreviation *greatest* **where** *greatest* $X \equiv$ *extreme* X $(\leq)$

abbreviation *supremum* **where** *supremum* $X \equiv$ *least* (*Collect* (*bound* X $(\leq)$))

abbreviation *infimum* **where** *infimum* $X \equiv$ *greatest* (*Collect* (*bound* X $(\lambda x\ y.\ y \leq x)$))

lemma *supremumI*: *bound* X $(\leq)$ $s \Longrightarrow (\bigwedge b.$ *bound* X $(\leq)$ $b \Longrightarrow s \leq b) \Longrightarrow$ *supremum* X s

and *infimumI*: *bound X* (≥) *i* ⟹ (⋀*b*. *bound X* (≥) *b* ⟹ *b* ≤ *i*) ⟹ *infimum X i*
⟨*proof*⟩

lemma *supremumE*: *supremum X s* ⟹
(*bound X* (≤) *s* ⟹ (⋀*b*. *bound X* (≤) *b* ⟹ *s* ≤ *b*) ⟹ *thesis*) ⟹ *thesis*
and *infimumE*: *infimum X i* ⟹
(*bound X* (≥) *i* ⟹ (⋀*b*. *bound X* (≥) *b* ⟹ *b* ≤ *i*) ⟹ *thesis*) ⟹ *thesis*
⟨*proof*⟩

lemma *extreme-bound-supremum*[*simp*]: *extreme-bound UNIV* (≤) = *supremum* ⟨*proof*⟩
lemma *extreme-bound-infimum*[*simp*]: *extreme-bound UNIV* (≥) = *infimum* ⟨*proof*⟩

lemma *Least-eq-The-least*: *Least P* = *The* (*least* {*x*. *P x*})
⟨*proof*⟩

lemma *The-least-eq-Least*: *The* (*least X*) = *Least* (λ*x*. *x* ∈ *X*)
⟨*proof*⟩

lemma *least-imp-infimum*: **assumes** *least X x* **shows** *infimum X x*
⟨*proof*⟩

lemma *least-LeastI-ex1*:
assumes *ex1*: ∃!*x*. *least* {*x*. *P x*} *x*
shows *least* {*x*. *P x*} (*LEAST x*. *P x*)
⟨*proof*⟩

end
context *order* **begin**

lemma *Greatest-eq-The-greatest*: *Greatest P* = *The* (*greatest* {*x*. *P x*})
⟨*proof*⟩

lemma *The-greatest-eq-Greatest*: *The* (*greatest X*) = *Greatest* (λ*x*. *x* ∈ *X*)
⟨*proof*⟩

lemma *greatest-imp-supremum*: **assumes** *greatest X x* **shows** *supremum X x*
⟨*proof*⟩

lemma *greatest-GreatestI-ex1*:
assumes *ex1*: ∃!*x*. *greatest* {*x*. *P x*} *x*
shows *greatest* {*x*. *P x*} (*GREATEST x*. *P x*)
⟨*proof*⟩

end

lemma *Ball-UNIV*[*simp*]: *Ball UNIV* = *All* ⟨*proof*⟩
lemma *Bex-UNIV*[*simp*]: *Bex UNIV* = *Ex* ⟨*proof*⟩

lemma *pointwise-UNIV-le*[*simp*]: *pointwise UNIV* (≤) = (≤) ⟨*proof*⟩
lemma *pointwise-UNIV-ge*[*simp*]: *pointwise UNIV* (≥) = (≥) ⟨*proof*⟩

lemma *fun-supremum-iff*: *supremum F e* ⟷ (∀ *x*. *supremum* {*f x* |. *f* ∈ *F*} (*e x*))
⟨*proof*⟩

lemma *fun-infimum-iff*: *infimum F e* ⟷ (∀ *x*. *infimum* {*f x* |. *f* ∈ *F*} (*e x*))
⟨*proof*⟩

class *reflorder* = *ord* + **assumes** *reflexive-ordering UNIV* (≤) (<)
begin

sublocale *order*: *reflexive-ordering UNIV*
rewrites ⋀*x*. *x* ∈ *UNIV* ≡ *True*
and ⋀*X*. *X* ⊆ *UNIV* ≡ *True*
and ⋀*r*. *r* ↾ *UNIV* ≡ *r*
and ⋀*P*. *True* ∧ *P* ≡ *P*
and *Ball UNIV* ≡ *All*
and *Bex UNIV* ≡ *Ex*
and *sympartp* (≤)$^-$ ≡ *sympartp* (≤)
and ⋀*P1*. (*True* ⟹ *PROP P1*) ≡ *PROP P1*
and ⋀*P1*. (*True* ⟹ *P1*) ≡ *Trueprop P1*
and ⋀*P1 P2*. (*True* ⟹ *PROP P1* ⟹ *PROP P2*) ≡ (*PROP P1* ⟹ *PROP P2*)
⟨*proof*⟩

end

We should have imported locale-based facts in classes, e.g.:

thm *order.trans order.strict.trans order.refl order.irrefl order.asym order.extreme-bound-singleton*

class *attrorder* = *ord* +
assumes *reflexive-attractive-ordering UNIV* (≤) (<)
begin

We need to declare subclasses before sublocales in order to preserve facts for superclasses.

subclass *reflorder*
⟨*proof*⟩

sublocale *order*: *reflexive-attractive-ordering UNIV*
rewrites ⋀*x*. *x* ∈ *UNIV* ≡ *True*
and ⋀*X*. *X* ⊆ *UNIV* ≡ *True*
and ⋀*r*. *r* ↾ *UNIV* ≡ *r*
and ⋀*P*. *True* ∧ *P* ≡ *P*
and *Ball UNIV* ≡ *All*

and *Bex UNIV* ≡ *Ex*
and *sympartp* $(\leq)^-$ ≡ *sympartp* $(\leq)$
and $\bigwedge$*P1*. (*True* ⟹ *PROP P1*) ≡ *PROP P1*
and $\bigwedge$*P1*. (*True* ⟹ *P1*) ≡ *Trueprop P1*
and $\bigwedge$*P1 P2*. (*True* ⟹ *PROP P1* ⟹ *PROP P2*) ≡ (*PROP P1* ⟹ *PROP P2*)
⟨*proof*⟩

end

thm *order.extreme-bound-quasi-const*

class *psorder* = *ord* + **assumes** *pseudo-ordering UNIV* $(\leq)$ $(<)$
begin

subclass *attrorder*
⟨*proof*⟩

sublocale *order*: *pseudo-ordering UNIV*
rewrites $\bigwedge$*x*. *x* ∈ *UNIV* ≡ *True*
and $\bigwedge$*X*. *X* ⊆ *UNIV* ≡ *True*
and $\bigwedge$*r*. *r* ↾ *UNIV* ≡ *r*
and $\bigwedge$*P*. *True* ∧ *P* ≡ *P*
and *Ball UNIV* ≡ *All*
and *Bex UNIV* ≡ *Ex*
and *sympartp* $(\leq)^-$ ≡ *sympartp* $(\leq)$
and $\bigwedge$*P1*. (*True* ⟹ *PROP P1*) ≡ *PROP P1*
and $\bigwedge$*P1*. (*True* ⟹ *P1*) ≡ *Trueprop P1*
and $\bigwedge$*P1 P2*. (*True* ⟹ *PROP P1* ⟹ *PROP P2*) ≡ (*PROP P1* ⟹ *PROP P2*)
⟨*proof*⟩

end

class *qorder* = *ord* + **assumes** *quasi-ordering UNIV* $(\leq)$ $(<)$
begin

subclass *attrorder*
⟨*proof*⟩

sublocale *order*: *quasi-ordering UNIV*
rewrites $\bigwedge$*x*. *x* ∈ *UNIV* ≡ *True*
and $\bigwedge$*X*. *X* ⊆ *UNIV* ≡ *True*
and $\bigwedge$*r*. *r* ↾ *UNIV* ≡ *r*
and $\bigwedge$*P*. *True* ∧ *P* ≡ *P*
and *Ball UNIV* ≡ *All*
and *Bex UNIV* ≡ *Ex*
and *sympartp* $(\leq)^-$ ≡ *sympartp* $(\leq)$
and $\bigwedge$*P1*. (*True* ⟹ *PROP P1*) ≡ *PROP P1*

and $\bigwedge P1.\ (True \Longrightarrow P1) \equiv Trueprop\ P1$
and $\bigwedge P1\ P2.\ (True \Longrightarrow PROP\ P1 \Longrightarrow PROP\ P2) \equiv (PROP\ P1 \Longrightarrow PROP\ P2)$
⟨*proof*⟩

lemmas [*intro*!] = *order.quasi-ordered-subset*

end

class *porder* = *ord* + **assumes** *partial-ordering UNIV* $(\leq)$ $(<)$
begin

interpretation *partial-ordering UNIV*
⟨*proof*⟩

subclass *psorder*⟨*proof*⟩

subclass *qorder*⟨*proof*⟩

sublocale *order*: *partial-ordering UNIV*
rewrites $\bigwedge x.\ x \in UNIV \equiv True$
and $\bigwedge X.\ X \subseteq UNIV \equiv True$
and $\bigwedge r.\ r \upharpoonright UNIV \equiv r$
and $\bigwedge P.\ True \wedge P \equiv P$
and $Ball\ UNIV \equiv All$
and $Bex\ UNIV \equiv Ex$
and $sympartp\ (\leq)^{-} \equiv sympartp\ (\leq)$
and $\bigwedge P1.\ (True \Longrightarrow PROP\ P1) \equiv PROP\ P1$
and $\bigwedge P1.\ (True \Longrightarrow P1) \equiv Trueprop\ P1$
and $\bigwedge P1\ P2.\ (True \Longrightarrow PROP\ P1 \Longrightarrow PROP\ P2) \equiv (PROP\ P1 \Longrightarrow PROP\ P2)$
⟨*proof*⟩

end

class *linqorder* = *ord* + **assumes** *total-quasi-ordering UNIV* $(\leq)$ $(<)$
begin

interpretation *total-quasi-ordering UNIV*
⟨*proof*⟩

subclass *qorder*⟨*proof*⟩

sublocale *order*: *total-quasi-ordering UNIV*
rewrites $\bigwedge x.\ x \in UNIV \equiv True$
and $\bigwedge X.\ X \subseteq UNIV \equiv True$
and $\bigwedge r.\ r \upharpoonright UNIV \equiv r$
and $\bigwedge P.\ True \wedge P \equiv P$
and $Ball\ UNIV \equiv All$

and *Bex UNIV* ≡ *Ex*
and *sympartp* (≤)$^{-}$ ≡ *sympartp* (≤)
and ⋀*P1*. (*True* ⟹ *PROP P1*) ≡ *PROP P1*
and ⋀*P1*. (*True* ⟹ *P1*) ≡ *Trueprop P1*
and ⋀*P1 P2*. (*True* ⟹ *PROP P1* ⟹ *PROP P2*) ≡ (*PROP P1* ⟹ *PROP P2*)
⟨*proof*⟩

lemmas *asympartp-le* = *order.not-iff-asym*[*symmetric*, *abs-def*]

end

Isabelle/HOL's *preorder* belongs to *qorder*, but not vice versa.

context *preorder* **begin**

The relation (<) is defined as the antisymmetric part of (≤).

lemma [*simp*]:
shows *asympartp-le*: *asympartp* (≤) = (<)
and *asympartp-ge*: *asympartp* (≥) = (>)
⟨*proof*⟩

interpretation *strict-quasi-ordering UNIV* (≤) (<)
⟨*proof*⟩

subclass *qorder*⟨*proof*⟩

sublocale *order*: *strict-quasi-ordering UNIV*
rewrites ⋀*x*. *x* ∈ *UNIV* ≡ *True*
and ⋀*X*. *X* ⊆ *UNIV* ≡ *True*
and ⋀*r*. *r* ↾ *UNIV* ≡ *r*
and ⋀*P*. *True* ∧ *P* ≡ *P*
and *Ball UNIV* ≡ *All*
and *Bex UNIV* ≡ *Ex*
and *sympartp* (≤)$^{-}$ ≡ *sympartp* (≤)
and ⋀*P1*. (*True* ⟹ *PROP P1*) ≡ *PROP P1*
and ⋀*P1*. (*True* ⟹ *P1*) ≡ *Trueprop P1*
and ⋀*P1 P2*. (*True* ⟹ *PROP P1* ⟹ *PROP P2*) ≡ (*PROP P1* ⟹ *PROP P2*)
⟨*proof*⟩

end

context *order* **begin**

interpretation *strict-partial-ordering UNIV* (≤) (<)
⟨*proof*⟩

subclass *porder*⟨*proof*⟩

sublocale *order*: *strict-partial-ordering UNIV*
  rewrites $\bigwedge x.\ x \in UNIV \equiv True$
    and $\bigwedge X.\ X \subseteq UNIV \equiv True$
    and $\bigwedge r.\ r \upharpoonright UNIV \equiv r$
    and $\bigwedge P.\ True \wedge P \equiv P$
    and $Ball\ UNIV \equiv All$
    and $Bex\ UNIV \equiv Ex$
    and $sympartp\ (\leq)^{-} \equiv sympartp\ (\leq)$
    and $\bigwedge P1.\ (True \Longrightarrow PROP\ P1) \equiv PROP\ P1$
    and $\bigwedge P1.\ (True \Longrightarrow P1) \equiv Trueprop\ P1$
    and $\bigwedge P1\ P2.\ (True \Longrightarrow PROP\ P1 \Longrightarrow PROP\ P2) \equiv (PROP\ P1 \Longrightarrow PROP\ P2)$
  ⟨*proof*⟩

end

context *order* **begin**

lemma *ex-greatest-iff-Greatest*:
  $Ex\ (greatest\ X) \longleftrightarrow greatest\ X\ (Greatest\ (\lambda x.\ x \in X))$
  ⟨*proof*⟩

lemma *greatest-imp-supremum-Greatest*:
  $greatest\ X\ x \Longrightarrow supremum\ X\ (Greatest\ (\lambda x.\ x \in X))$
  ⟨*proof*⟩

end

Isabelle/HOL's *linorder* is equivalent to our locale *total-ordering*.

context *linorder* **begin**

subclass *linqorder* ⟨*proof*⟩

sublocale *order*: *total-ordering UNIV*
  rewrites $\bigwedge x.\ x \in UNIV \equiv True$
    and $\bigwedge X.\ X \subseteq UNIV \equiv True$
    and $\bigwedge r.\ r \upharpoonright UNIV \equiv r$
    and $\bigwedge P.\ True \wedge P \equiv P$
    and $Ball\ UNIV \equiv All$
    and $Bex\ UNIV \equiv Ex$
    and $sympartp\ (\leq)^{-} \equiv sympartp\ (\leq)$
    and $\bigwedge P1.\ (True \Longrightarrow PROP\ P1) \equiv PROP\ P1$
    and $\bigwedge P1.\ (True \Longrightarrow P1) \equiv Trueprop\ P1$
    and $\bigwedge P1\ P2.\ (True \Longrightarrow PROP\ P1 \Longrightarrow PROP\ P2) \equiv (PROP\ P1 \Longrightarrow PROP\ P2)$
  ⟨*proof*⟩

end

Tests: facts should be available in the most general classes.

thm *order.strict.trans*[**where** *'a='a::reflorder*]
thm *order.extreme-bound-quasi-const*[**where** *'a='a::attrorder*]
thm *order.extreme-bound-singleton-eq*[**where** *'a='a::psorder*]
thm *order.trans*[**where** *'a='a::qorder*]
thm *order.comparable-cases*[**where** *'a='a::linqorder*]
thm *order.cases*[**where** *'a='a::linorder*]

2.8 Declaring Duals

sublocale *reflexive* ⊆ *sym*: *reflexive A sympartp* (⊑)
rewrites *sympartp* (⊑)$^-$ ≡ *sympartp* (⊑)
and ⋀*r*. *sympartp* (*sympartp r*) ≡ *sympartp r*
and ⋀*r*. *sympartp r* ↾ *A* ≡ *sympartp* (*r* ↾ *A*)
⟨*proof*⟩

sublocale *quasi-ordered-set* ⊆ *sym*: *quasi-ordered-set A sympartp* (⊑)
rewrites *sympartp* (⊑)$^-$ = *sympartp* (⊑)
and *sympartp* (*sympartp* (⊑)) = *sympartp* (⊑)
⟨*proof*⟩

At this point, we declare dual as sublocales. In the following, "rewrites" eventually cleans up redundant facts.

sublocale *reflexive* ⊆ *dual*: *reflexive A* (⊑)$^-$
rewrites *sympartp* (⊑)$^-$ ≡ *sympartp* (⊑)
and ⋀*r*. *sympartp* (*r* ↾ *A*) ≡ *sympartp r* ↾ *A*
and (⊑)$^-$ ↾ *A* ≡ ((⊑) ↾ *A*)$^-$
⟨*proof*⟩

context *attractive* **begin**

interpretation *less-eq-symmetrize*⟨*proof*⟩

sublocale *dual*: *attractive A* (⊒)
rewrites *sympartp* (⊒) = (∼)
and *equivpartp* (⊒) ≡ (≃)
and ⋀*r*. *sympartp* (*r* ↾ *A*) ≡ *sympartp r* ↾ *A*
and ⋀*r*. *sympartp* (*sympartp r*) ≡ *sympartp r*
and (⊑)$^-$ ↾ *A* ≡ ((⊑) ↾ *A*)$^-$
⟨*proof*⟩

end

context *irreflexive* **begin**

sublocale *dual*: *irreflexive A* (⊏)$^-$
rewrites (⊏)$^-$ ↾ *A* ≡ ((⊏) ↾ *A*)$^-$
⟨*proof*⟩

end

sublocale *transitive* $\subseteq$ *dual*: *transitive* A $(\sqsubseteq)^-$
rewrites $(\sqsubseteq)^- \restriction A \equiv ((\sqsubseteq) \restriction A)^-$
and *sympartp* $(\sqsubseteq)^-$ = *sympartp* $(\sqsubseteq)$
and *asympartp* $(\sqsubseteq)^-$ = (*asympartp* $(\sqsubseteq))^-$
⟨*proof*⟩

sublocale *antisymmetric* $\subseteq$ *dual*: *antisymmetric* A $(\sqsubseteq)^-$
rewrites $(\sqsubseteq)^- \restriction A \equiv ((\sqsubseteq) \restriction A)^-$
and *sympartp* $(\sqsubseteq)^-$ = *sympartp* $(\sqsubseteq)$
⟨*proof*⟩

context *antisymmetric* **begin**

lemma *extreme-bound-unique*:
extreme-bound A $(\sqsubseteq)$ X $x \Longrightarrow$ *extreme-bound* A $(\sqsubseteq)$ X $y \longleftrightarrow x = y$
⟨*proof*⟩

lemma *ex-extreme-bound-iff-ex1*:
Ex (*extreme-bound* A $(\sqsubseteq)$ X) $\longleftrightarrow$ *Ex1* (*extreme-bound* A $(\sqsubseteq)$ X)
⟨*proof*⟩

lemma *ex-extreme-bound-iff-the*:
Ex (*extreme-bound* A $(\sqsubseteq)$ X) $\longleftrightarrow$ *extreme-bound* A $(\sqsubseteq)$ X (*The* (*extreme-bound* A $(\sqsubseteq)$ X))
⟨*proof*⟩

end

sublocale *semiconnex* $\subseteq$ *dual*: *semiconnex* A $(\sqsubset)^-$
rewrites *sympartp* $(\sqsubset)^-$ = *sympartp* $(\sqsubset)$
⟨*proof*⟩

sublocale *connex* $\subseteq$ *dual*: *connex* A $(\sqsubseteq)^-$
rewrites *sympartp* $(\sqsubseteq)^-$ = *sympartp* $(\sqsubseteq)$
⟨*proof*⟩

sublocale *semiconnex-irreflexive* $\subseteq$ *dual*: *semiconnex-irreflexive* A $(\sqsubset)^-$
rewrites *sympartp* $(\sqsubset)^-$ = *sympartp* $(\sqsubset)$
⟨*proof*⟩

sublocale *pseudo-ordered-set* $\subseteq$ *dual*: *pseudo-ordered-set* A $(\sqsubseteq)^-$
rewrites *sympartp* $(\sqsubseteq)^-$ = *sympartp* $(\sqsubseteq)$
⟨*proof*⟩

sublocale *quasi-ordered-set* $\subseteq$ *dual*: *quasi-ordered-set* A $(\sqsubseteq)^-$
rewrites *sympartp* $(\sqsubseteq)^-$ = *sympartp* $(\sqsubseteq)$
⟨*proof*⟩

sublocale *partially-ordered-set* ⊆ *dual*: *partially-ordered-set* A $(\sqsubseteq)^-$
rewrites *sympartp* $(\sqsubseteq)^- =$ *sympartp* $(\sqsubseteq)$
⟨*proof*⟩

sublocale *total-pseudo-ordered-set* ⊆ *dual*: *total-pseudo-ordered-set* A $(\sqsubseteq)^-$
rewrites *sympartp* $(\sqsubseteq)^- =$ *sympartp* $(\sqsubseteq)$
⟨*proof*⟩

sublocale *total-quasi-ordered-set* ⊆ *dual*: *total-quasi-ordered-set* A $(\sqsubseteq)^-$
rewrites *sympartp* $(\sqsubseteq)^- =$ *sympartp* $(\sqsubseteq)$
⟨*proof*⟩

sublocale *compatible-ordering* ⊆ *dual*: *compatible-ordering* A $(\sqsubseteq)^-$ $(\sqsubset)^-$
rewrites *sympartp* $(\sqsubseteq)^- =$ *sympartp* $(\sqsubseteq)$
⟨*proof*⟩

lemmas(**in** *qorder*) [*intro*!] = *order.dual.quasi-ordered-subset*

sublocale *reflexive-ordering* ⊆ *dual*: *reflexive-ordering* A $(\sqsubseteq)^-$ $(\sqsubset)^-$
rewrites *sympartp* $(\sqsubseteq)^- =$ *sympartp* $(\sqsubseteq)$
⟨*proof*⟩

sublocale *reflexive-attractive-ordering* ⊆ *dual*: *reflexive-attractive-ordering* A $(\sqsubseteq)^-$ $(\sqsubset)^-$
rewrites *sympartp* $(\sqsubseteq)^- =$ *sympartp* $(\sqsubseteq)$
⟨*proof*⟩

sublocale *pseudo-ordering* ⊆ *dual*: *pseudo-ordering* A $(\sqsubseteq)^-$ $(\sqsubset)^-$
rewrites *sympartp* $(\sqsubseteq)^- =$ *sympartp* $(\sqsubseteq)$
⟨*proof*⟩

lemma (**in** *psorder*) *least-Least*:
fixes X :: *'a set*
shows *Ex* (*least* X) ⟷ *least* X (*LEAST* x. $x \in X$)
⟨*proof*⟩

sublocale *quasi-ordering* ⊆ *dual*: *quasi-ordering* A $(\sqsubseteq)^-$ $(\sqsubset)^-$
rewrites *sympartp* $(\sqsubseteq)^- =$ *sympartp* $(\sqsubseteq)$
⟨*proof*⟩

sublocale *partial-ordering* ⊆ *dual*: *partial-ordering* A $(\sqsubseteq)^-$ $(\sqsubset)^-$
rewrites *sympartp* $(\sqsubseteq)^- =$ *sympartp* $(\sqsubseteq)$
⟨*proof*⟩

sublocale *total-quasi-ordering* ⊆ *dual*: *total-quasi-ordering* A $(\sqsubseteq)^-$ $(\sqsubset)^-$
rewrites *sympartp* $(\sqsubseteq)^- =$ *sympartp* $(\sqsubseteq)$
⟨*proof*⟩

sublocale *total-ordering* ⊆ *dual*: *total-ordering* A $(\sqsubseteq)^-$ $(\sqsubset)^-$
rewrites *sympartp* $(\sqsubseteq)^-$ = *sympartp* $(\sqsubseteq)$
⟨*proof*⟩

sublocale *strict-quasi-ordering* ⊆ *dual*: *strict-quasi-ordering* A $(\sqsubseteq)^-$ $(\sqsubset)^-$
rewrites *sympartp* $(\sqsubseteq)^-$ = *sympartp* $(\sqsubseteq)$
⟨*proof*⟩

sublocale *strict-partial-ordering* ⊆ *dual*: *strict-partial-ordering* A $(\sqsubseteq)^-$ $(\sqsubset)^-$
rewrites *sympartp* $(\sqsubseteq)^-$ = *sympartp* $(\sqsubseteq)$
⟨*proof*⟩

sublocale *total-ordering* ⊆ *dual*: *total-ordering* A $(\sqsubseteq)^-$ $(\sqsubset)^-$
rewrites *sympartp* $(\sqsubseteq)^-$ = *sympartp* $(\sqsubseteq)$
⟨*proof*⟩

lemma(**in** *antisymmetric*) *monotone-extreme-imp-extreme-bound-iff*:
fixes *ir* (**infix** ‹⪯› *50*)
assumes f ‘ $C \subseteq A$ **and** *monotone-on* C $(\preceq)$ $(\sqsubseteq)$ f **and** i: *extreme* C $(\preceq)$ i
shows *extreme-bound* A $(\sqsubseteq)$ $(f$ ‘ $C)$ $x \longleftrightarrow f\ i = x$
⟨*proof*⟩

2.9 Instantiations

Finally, we instantiate our classes for sanity check.

instance *nat* :: *linorder* ⟨*proof*⟩

Pointwise ordering of functions are compatible only if the weak part is transitive.

instance *fun* :: (*type*,*qorder*) *reflorder*
⟨*proof*⟩

instance *fun* :: (*type*,*qorder*) *qorder*
⟨*proof*⟩

instance *fun* :: (*type*,*porder*) *porder*
⟨*proof*⟩

end
theory *Well-Relations*
imports *Binary-Relations*
begin

3 Well-Relations

A related set $\langle A, \sqsubseteq\rangle$ is called *topped* if there is a "top" element $\top \in A$, a greatest element in A. Note that there might be multiple tops if $(\sqsubseteq)$ is not antisymmetric.

definition *extremed A r* ≡ ∃ *e*. *extreme A r e*

lemma *extremedI*: *extreme A r e* ⟹ *extremed A r*
⟨*proof*⟩

lemma *extremedE*: *extremed A r* ⟹ (⋀*e*. *extreme A r e* ⟹ *thesis*) ⟹ *thesis*
⟨*proof*⟩

lemma *extremed-imp-ex-bound*: *extremed A r* ⟹ *X* ⊆ *A* ⟹ ∃ *b* ∈ *A*. *bound X r b*
⟨*proof*⟩

locale *well-founded* = *related-set* - (⊏) + *less-syntax* +
assumes *induct*[*consumes 1*, *case-names less*, *induct set*]:
a ∈ *A* ⟹ (⋀*x*. *x* ∈ *A* ⟹ (⋀*y*. *y* ∈ *A* ⟹ *y* ⊏ *x* ⟹ *P y*) ⟹ *P x*) ⟹ *P a*
begin

sublocale *asymmetric*
⟨*proof*⟩

lemma *prefixed-Imagep-imp-empty*:
assumes *a*: *X* ⊆ ((⊏) ``` *X*) ∩ *A* **shows** *X* = {}
⟨*proof*⟩

lemma *nonempty-imp-ex-extremal*:
assumes *QA*: *Q* ⊆ *A* **and** *Q*: *Q* ≠ {}
shows ∃ *z* ∈ *Q*. ∀ *y* ∈ *Q*. ¬ *y* ⊏ *z*
⟨*proof*⟩

interpretation *Restrp*: *well-founded UNIV* (⊏)↾*A*
rewrites ⋀*x*. *x* ∈ *UNIV* ≡ *True*
and (⊏)↾*A*↾*UNIV* = (⊏)↾*A*
and ⋀*P1*. (*True* ⟹ *PROP P1*) ≡ *PROP P1*
and ⋀*P1*. (*True* ⟹ *P1*) ≡ *Trueprop P1*
and ⋀*P1 P2*. (*True* ⟹ *PROP P1* ⟹ *PROP P2*) ≡ (*PROP P1* ⟹ *PROP P2*)
⟨*proof*⟩

lemmas *Restrp-well-founded* = *Restrp.well-founded-axioms*
lemmas *Restrp-induct*[*consumes 0*, *case-names less*] = *Restrp.induct*

interpretation *Restrp.tranclp*: *well-founded UNIV* $((\sqsubset)\restriction A)^{++}$
rewrites ⋀*x*. *x* ∈ *UNIV* ≡ *True*
and $((\sqsubset)\restriction A)^{++} \restriction UNIV = ((\sqsubset)\restriction A)^{++}$
and $(((\sqsubset)\restriction A)^{++})^{++} = ((\sqsubset)\restriction A)^{++}$
and ⋀*P1*. (*True* ⟹ *PROP P1*) ≡ *PROP P1*
and ⋀*P1*. (*True* ⟹ *P1*) ≡ *Trueprop P1*
and ⋀*P1 P2*. (*True* ⟹ *PROP P1* ⟹ *PROP P2*) ≡ (*PROP P1* ⟹ *PROP P2*)

⟨*proof*⟩

lemmas *Restrp-tranclp-well-founded* = *Restrp.tranclp.well-founded-axioms*
lemmas *Restrp-tranclp-induct*[*consumes 0*, *case-names less*] = *Restrp.tranclp.induct*

end

context
  fixes A :: *'a set* **and** *less* :: $'a \Rightarrow 'a \Rightarrow$ *bool* (**infix** ‹⊏› *50*)
begin

lemma *well-foundedI-pf*:
  assumes *pre*: $\bigwedge X.\ X \subseteq A \Longrightarrow X \subseteq ((\sqsubset)\ ``\ X) \cap A \Longrightarrow X = \{\}$
  shows *well-founded* A $(\sqsubset)$
⟨*proof*⟩

lemma *well-foundedI-extremal*:
  assumes *a*: $\bigwedge X.\ X \subseteq A \Longrightarrow X \neq \{\} \Longrightarrow \exists x \in X.\ \forall y \in X.\ \neg\ y \sqsubset x$
  shows *well-founded* A $(\sqsubset)$
⟨*proof*⟩

lemma *well-founded-iff-ex-extremal*:
  well-founded A $(\sqsubset) \longleftrightarrow (\forall X \subseteq A.\ X \neq \{\} \longrightarrow (\exists x \in X.\ \forall z \in X.\ \neg\ z \sqsubset x))$
  ⟨*proof*⟩

end

lemma *well-founded-cong*:
  assumes *r*: $\bigwedge a\ b.\ a \in A \Longrightarrow b \in A \Longrightarrow r\ a\ b \longleftrightarrow r'\ a\ b$
    and *A*: $\bigwedge a\ b.\ r'\ a\ b \Longrightarrow a \in A \longleftrightarrow a \in A'$
    and *B*: $\bigwedge a\ b.\ r'\ a\ b \Longrightarrow b \in A \longleftrightarrow b \in A'$
  shows *well-founded* A $r \longleftrightarrow$ *well-founded* A' r'
⟨*proof*⟩

lemma *wfP-iff-well-founded-UNIV*: *wfP* $r \longleftrightarrow$ *well-founded UNIV* r
  ⟨*proof*⟩

lemma *well-founded-empty*[*intro*!]: *well-founded* $\{\}$ r
  ⟨*proof*⟩

lemma *well-founded-singleton*:
  assumes $\neg r\ x\ x$ **shows** *well-founded* $\{x\}$ r
  ⟨*proof*⟩

lemma *well-founded-Restrp*[*simp*]: *well-founded* A $(r{\restriction}B) \longleftrightarrow$ *well-founded* $(A \cap B)$ r (**is** *?l* $\longleftrightarrow$ *?r*)
⟨*proof*⟩

lemma *Restrp-tranclp-well-founded-iff*:

fixes *less* (**infix** ‹$\sqsubset$› *50*)
shows *well-founded UNIV* $((\sqsubset) \upharpoonright A)^{++}$ $\longleftrightarrow$ *well-founded A* ($\sqsubset$) (**is** *?l* $\longleftrightarrow$ *?r*)
⟨*proof*⟩

lemma (**in** *well-founded*) *well-founded-subset*:
assumes $B \subseteq A$ **shows** *well-founded B* ($\sqsubset$)
⟨*proof*⟩

lemma *well-founded-extend*:
fixes *less* (**infix** ‹$\sqsubset$› *50*)
assumes *A*: *well-founded A* ($\sqsubset$)
assumes *B*: *well-founded B* ($\sqsubset$)
assumes *AB*: $\forall a \in A.\ \forall b \in B.\ \neg b \sqsubset a$
shows *well-founded* $(A \cup B)$ ($\sqsubset$)
⟨*proof*⟩

lemma *closed-UN-well-founded*:
fixes *r* (**infix** ‹$\sqsubset$› *50*)
assumes *XX*: $\forall X{\in}XX.$ *well-founded X* ($\sqsubset$) $\wedge$ $(\forall x{\in}X.\ \forall y{\in}\bigcup XX.\ y \sqsubset x \longrightarrow y \in X)$
shows *well-founded* $(\bigcup XX)$ ($\sqsubset$)
⟨*proof*⟩

lemma *well-founded-cmono*:
assumes r': $r' \leq r$ **and** *wf*: *well-founded A r*
shows *well-founded A* r'
⟨*proof*⟩

locale *well-founded-ordered-set* = *well-founded* + *transitive* - ($\sqsubset$)
begin

sublocale *strict-ordered-set*⟨*proof*⟩

interpretation *Restrp*: *strict-ordered-set UNIV* $(\sqsubset)\upharpoonright A$ + *Restrp*: *well-founded UNIV* $(\sqsubset)\upharpoonright A$
⟨*proof*⟩

lemma *Restrp-well-founded-order*: *well-founded-ordered-set UNIV* $((\sqsubset)\upharpoonright A)$⟨*proof*⟩

lemma *well-founded-ordered-subset*: $B \subseteq A \Longrightarrow$ *well-founded-ordered-set B* ($\sqsubset$)
⟨*proof*⟩

end

lemmas *well-founded-ordered-setI* = *well-founded-ordered-set.intro*

lemma *well-founded-ordered-set-empty*[*intro*!]: *well-founded-ordered-set* {} *r*
⟨*proof*⟩

locale *well-related-set* = *related-set* +
assumes *nonempty-imp-ex-extreme*: $X \subseteq A \Longrightarrow X \neq \{\} \Longrightarrow \exists e.$ *extreme* X $(\sqsubseteq)^{-}$ e
begin

sublocale *connex*
⟨*proof*⟩

lemmas *connex* = *connex-axioms*

interpretation *less-eq-asymmetrize*⟨*proof*⟩

sublocale *asym*: *well-founded* A $(\sqsubset)$
⟨*proof*⟩

lemma *well-related-subset*: $B \subseteq A \Longrightarrow$ *well-related-set* B $(\sqsubseteq)$
⟨*proof*⟩

lemma *monotone-image-well-related*:
fixes *leB* (**infix** ‹⊴› *50*)
assumes *mono*: *monotone-on* A $(\sqsubseteq)$ $(\trianglelefteq)$ f **shows** *well-related-set* $(f$ ‘ $A)$ $(\trianglelefteq)$
⟨*proof*⟩

end

sublocale *well-related-set* ⊆ *reflexive* ⟨*proof*⟩

lemmas *well-related-setI* = *well-related-set.intro*

lemmas *well-related-iff-ex-extreme* = *well-related-set-def*

lemma *well-related-set-empty*[*intro*!]: *well-related-set* {} *r*
⟨*proof*⟩

context
fixes *less-eq* :: $'a \Rightarrow 'a \Rightarrow$ *bool* (**infix** ‹⊑› *50*)
begin

lemma *well-related-iff-neg-well-founded*:
well-related-set A $(\sqsubseteq)$ $\longleftrightarrow$ *well-founded* A $(\lambda x\ y.\ \neg\ y \sqsubseteq x)$
⟨*proof*⟩

lemma *well-related-singleton-refl*:
assumes $x \sqsubseteq x$ **shows** *well-related-set* $\{x\}$ $(\sqsubseteq)$
⟨*proof*⟩

lemma *closed-UN-well-related*:
assumes *XX*: $\forall X \in XX.$ *well-related-set* X $(\sqsubseteq)$ $\wedge$ $(\forall x \in X.\ \forall y \in \bigcup XX.\ \neg x \sqsubseteq y$

$\longrightarrow y \in X$)
shows *well-related-set* ($\bigcup XX$) ($\sqsubseteq$)
⟨*proof*⟩

end

lemma *well-related-extend*:
fixes *r* (**infix** ‹$\sqsubseteq$› *50*)
assumes *well-related-set* A ($\sqsubseteq$) **and** *well-related-set* B ($\sqsubseteq$) **and** $\forall a \in A. \forall b \in B. a \sqsubseteq b$
shows *well-related-set* ($A \cup B$) ($\sqsubseteq$)
⟨*proof*⟩

lemma *pair-well-related*:
fixes *less-eq* (**infix** ‹$\sqsubseteq$› *50*)
assumes $i \sqsubseteq i$ **and** $i \sqsubseteq j$ **and** $j \sqsubseteq j$
shows *well-related-set* $\{i, j\}$ ($\sqsubseteq$)
⟨*proof*⟩

locale *pre-well-ordered-set* = *semiattractive* + *well-related-set*
begin

interpretation *less-eq-asymmetrize*⟨*proof*⟩

sublocale *transitive*
⟨*proof*⟩

sublocale *total-quasi-ordered-set*⟨*proof*⟩

end

lemmas *pre-well-ordered-iff-semiattractive-well-related* =
pre-well-ordered-set-def[*unfolded atomize-eq*]

lemma *pre-well-ordered-set-empty*[*intro*!]: *pre-well-ordered-set* {} *r*
⟨*proof*⟩

lemma *pre-well-ordered-iff*:
pre-well-ordered-set A $r \longleftrightarrow$ *total-quasi-ordered-set* A $r \wedge$ *well-founded* A (*asympartp* *r*)
(**is** *?p* $\longleftrightarrow$ *?t* $\wedge$ *?w*)
⟨*proof*⟩

lemma (**in** *semiattractive*) *pre-well-ordered-iff-well-related*:
assumes *XA*: $X \subseteq A$
shows *pre-well-ordered-set* X ($\sqsubseteq$) $\longleftrightarrow$ *well-related-set* X ($\sqsubseteq$) (**is** *?l* $\longleftrightarrow$ *?r*)
⟨*proof*⟩

lemma *semiattractive-extend*:

fixes *r* (**infix** ‹⊑› *50*)
assumes *A*: *semiattractive A* (⊑) **and** *B*: *semiattractive B* (⊑)
and *AB*: ∀ *a* ∈ *A*. ∀ *b* ∈ *B*. *a* ⊑ *b* ∧ ¬ *b* ⊑ *a*
shows *semiattractive* (*A* ∪ *B*) (⊑)
⟨*proof*⟩

lemma *pre-well-order-extend*:
fixes *r* (**infix** ‹⊑› *50*)
assumes *A*: *pre-well-ordered-set A* (⊑) **and** *B*: *pre-well-ordered-set B* (⊑)
and *AB*: ∀ *a* ∈ *A*. ∀ *b* ∈ *B*. *a* ⊑ *b* ∧ ¬ *b* ⊑ *a*
shows *pre-well-ordered-set* (*A*∪*B*) (⊑)
⟨*proof*⟩

lemma (**in** *well-related-set*) *monotone-image-pre-well-ordered*:
fixes *leB* (**infix** ‹⊑′′› *50*)
assumes *mono*: *monotone-on A* (⊑) (⊑′) *f*
and *image*: *semiattractive* (*f* ‘ *A*) (⊑′)
shows *pre-well-ordered-set* (*f* ‘ *A*) (⊑′)
⟨*proof*⟩

locale *well-ordered-set* = *antisymmetric* + *well-related-set*
begin

sublocale *pre-well-ordered-set*⟨*proof*⟩

sublocale *total-ordered-set*⟨*proof*⟩

lemma *well-ordered-subset*: *B* ⊆ *A* ⟹ *well-ordered-set B* (⊑)
⟨*proof*⟩

sublocale *asym*: *well-founded-ordered-set A asympartp* (⊑)
⟨*proof*⟩

end

lemmas *well-ordered-iff-antisymmetric-well-related* = *well-ordered-set-def*[*unfolded atomize-eq*]

lemma *well-ordered-set-empty*[*intro*!]: *well-ordered-set* {} *r*
⟨*proof*⟩

lemma (**in** *antisymmetric*) *well-ordered-iff-well-related*:
assumes *XA*: *X* ⊆ *A*
shows *well-ordered-set X* (⊑) ⟷ *well-related-set X* (⊑) (**is** *?l* ⟷ *?r*)
⟨*proof*⟩

context
fixes *A* :: ′*a set* **and** *less-eq* :: ′*a* ⇒ ′*a* ⇒ *bool* (**infix** ‹⊑› *50*)
begin

context
assumes *A*: ∀ *a* ∈ *A*. ∀ *b* ∈ *A*. *a* ⊑ *b*
begin

interpretation *well-related-set A* (⊑)
⟨*proof*⟩

lemmas *trivial-well-related* = *well-related-set-axioms*

lemma *trivial-pre-well-order*: *pre-well-ordered-set A* (⊑)
⟨*proof*⟩

end

interpretation *less-eq-asymmetrize*⟨*proof*⟩

lemma *well-ordered-iff-well-founded-total-ordered*:
well-ordered-set A (⊑) ⟷ *total-ordered-set A* (⊑) ∧ *well-founded A* (⊏)
⟨*proof*⟩

end

context
fixes *less-eq* :: ′*a* ⇒ ′*a* ⇒ *bool* (**infix** ‹⊑› *50*)
begin

lemma *well-order-extend*:
assumes *A*: *well-ordered-set A* (⊑) **and** *B*: *well-ordered-set B* (⊑)
and *ABa*: ∀ *a* ∈ *A*. ∀ *b* ∈ *B*. *a* ⊑ *b* ⟶ *b* ⊑ *a* ⟶ *a* = *b*
and *AB*: ∀ *a* ∈ *A*. ∀ *b* ∈ *B*. *a* ⊑ *b*
shows *well-ordered-set* (*A*∪*B*) (⊑)
⟨*proof*⟩

interpretation *singleton*: *antisymmetric* {*a*} (⊑) **for** *a* ⟨*proof*⟩

lemmas *singleton-antisymmetric*[*intro*!] = *singleton.antisymmetric-axioms*

lemma *singleton-well-ordered*[*intro*!]: *a* ⊑ *a* ⟹ *well-ordered-set* {*a*} (⊑)
⟨*proof*⟩

lemma *closed-UN-well-ordered*:
assumes *anti*: *antisymmetric* (⋃ *XX*) (⊑)
and *XX*: ∀ *X*∈*XX*. *well-ordered-set X* (⊑) ∧ (∀ *x*∈*X*. ∀ *y*∈⋃ *XX*. ¬ *x* ⊑ *y* ⟶ *y* ∈ *X*)
shows *well-ordered-set* (⋃ *XX*) (⊑)
⟨*proof*⟩

end

lemma (**in** *well-related-set*) *monotone-image-well-ordered*:
  fixes *leB* (**infix** ‹⊑′′› *50*)
  assumes *mono*: *monotone-on A* (⊑) (⊑′) *f*
    and *image*: *antisymmetric* (*f* ‘ *A*) (⊑′)
  shows *well-ordered-set* (*f* ‘ *A*) (⊑′)
  ⟨*proof*⟩

3.1 Relating to Classes

locale *well-founded-quasi-ordering* = *quasi-ordering* + *well-founded*
begin

lemma *well-founded-quasi-ordering-subset*:
  assumes $X \subseteq A$ **shows** *well-founded-quasi-ordering X* (⊑) (⊏)
  ⟨*proof*⟩

end

class *wf-qorder* = *ord* +
  assumes *well-founded-quasi-ordering UNIV* (≤) (<)
begin

interpretation *well-founded-quasi-ordering UNIV*
  ⟨*proof*⟩

subclass *qorder* ⟨*proof*⟩

sublocale *order*: *well-founded-quasi-ordering UNIV*
  rewrites $\bigwedge x.\ x \in UNIV \equiv True$
    and $\bigwedge X.\ X \subseteq UNIV \equiv True$
    and $\bigwedge r.\ r \restriction UNIV \equiv r$
    and $\bigwedge P.\ True \wedge P \equiv P$
    and *Ball UNIV* ≡ *All*
    and *Bex UNIV* ≡ *Ex*
    and *sympartp* $(\leq)^{-}$ ≡ *sympartp* (≤)
    and $\bigwedge P1.\ (True \Longrightarrow PROP\ P1) \equiv PROP\ P1$
    and $\bigwedge P1.\ (True \Longrightarrow P1) \equiv Trueprop\ P1$
    and $\bigwedge P1\ P2.\ (True \Longrightarrow PROP\ P1 \Longrightarrow PROP\ P2) \equiv (PROP\ P1 \Longrightarrow PROP\ P2)$
  ⟨*proof*⟩

end

context *wellorder* **begin**

subclass *wf-qorder*
  ⟨*proof*⟩

sublocale *order*: *well-ordered-set UNIV*
rewrites $\bigwedge x.\ x \in UNIV \equiv True$
and $\bigwedge X.\ X \subseteq UNIV \equiv True$
and $\bigwedge r.\ r \upharpoonright UNIV \equiv r$
and $\bigwedge P.\ True \wedge P \equiv P$
and $Ball\ UNIV \equiv All$
and $Bex\ UNIV \equiv Ex$
and $sympartp\ (\leq)^{-} \equiv sympartp\ (\leq)$
and $\bigwedge P1.\ (True \Longrightarrow PROP\ P1) \equiv PROP\ P1$
and $\bigwedge P1.\ (True \Longrightarrow P1) \equiv Trueprop\ P1$
and $\bigwedge P1\ P2.\ (True \Longrightarrow PROP\ P1 \Longrightarrow PROP\ P2) \equiv (PROP\ P1 \Longrightarrow PROP\ P2)$
⟨*proof*⟩

end

thm *order.nonempty-imp-ex-extreme*

3.2 omega-Chains

definition *omega-chain A r* $\equiv \exists f :: nat \Rightarrow 'a.\ monotone\ (\leq)\ r\ f \wedge range\ f = A$

lemma *omega-chainI*:
fixes $f :: nat \Rightarrow 'a$
assumes $monotone\ (\leq)\ r\ f\ range\ f = A$ **shows** *omega-chain A r*
⟨*proof*⟩

lemma *omega-chainE*:
assumes *omega-chain A r*
and $\bigwedge f :: nat \Rightarrow 'a.\ monotone\ (\leq)\ r\ f \Longrightarrow range\ f = A \Longrightarrow thesis$
shows *thesis*
⟨*proof*⟩

lemma (**in** *transitive*) *local-chain*:
assumes *CA*: $range\ C \subseteq A$
shows $(\forall i{::}nat.\ C\ i \sqsubseteq C\ (Suc\ i)) \longleftrightarrow monotone\ (<)\ (\sqsubseteq)\ C$
⟨*proof*⟩

lemma *pair-omega-chain*:
assumes *r a a r b b r a b* **shows** *omega-chain* $\{a,b\}$ *r*
⟨*proof*⟩

Every omega-chain is a well-order.

lemma *omega-chain-imp-well-related*:
fixes *less-eq* (**infix** ‹⊑› *50*)
assumes *A*: *omega-chain A* $(\sqsubseteq)$ **shows** *well-related-set A* $(\sqsubseteq)$
⟨*proof*⟩

lemma (**in** *semiattractive*) *omega-chain-imp-pre-well-ordered*:
assumes *omega-chain A* $(\sqsubseteq)$ **shows** *pre-well-ordered-set A* $(\sqsubseteq)$

⟨*proof*⟩

lemma (**in** *antisymmetric*) *omega-chain-imp-well-ordered*:
assumes *omega-chain A* (⊑) **shows** *well-ordered-set A* (⊑)
⟨*proof*⟩

3.2.1 Relation image that preserves well-orderedness.

definition *well-image f A* (⊑) *fa fb* ≡
∀ *a b*. *extreme* {*x*∈*A*. *fa* = *f x*} (⊑)⁻ *a* ⟶ *extreme* {*y*∈*A*. *fb* = *f y*} (⊑)⁻ *b* ⟶
a ⊑ *b*
for *less-eq* (**infix** ‹⊑› *50*)

lemmas *well-imageI* = *well-image-def*[*unfolded atomize-eq*, *THEN iffD2*, *rule-format*]
lemmas *well-imageD* = *well-image-def*[*unfolded atomize-eq*, *THEN iffD1*, *rule-format*]

lemma (**in** *pre-well-ordered-set*)
well-image-well-related: *pre-well-ordered-set* (*f*‘*A*) (*well-image f A* (⊑))
⟨*proof*⟩

end
theory *Directedness*
imports *Binary-Relations Well-Relations*
begin

Directed sets:

locale *directed* =
fixes *A* **and** *less-eq* (**infix** ‹⊑› *50*)
assumes *pair-bounded*: $x \in A \Longrightarrow y \in A \Longrightarrow \exists z \in A.\ x \sqsubseteq z \wedge y \sqsubseteq z$

lemmas *directedI*[*intro*] = *directed.intro*

lemmas *directedD* = *directed-def*[*unfolded atomize-eq*, *THEN iffD1*, *rule-format*]

context
fixes *less-eq* :: *'a* ⇒ *'a* ⇒ *bool* (**infix** ‹⊑› *50*)
begin

lemma *directedE*:
assumes *directed A* (⊑) **and** $x \in A$ **and** $y \in A$
and $\bigwedge z.\ z \in A \Longrightarrow x \sqsubseteq z \Longrightarrow y \sqsubseteq z \Longrightarrow$ *thesis*
shows *thesis*
⟨*proof*⟩

lemma *directed-empty*[*simp*]: *directed* {} (⊑) ⟨*proof*⟩

lemma *directed-union*:
assumes *dX*: *directed X* (⊑) **and** *dY*: *directed Y* (⊑)
and *XY*: $\forall x{\in}X.\ \forall y{\in}Y.\ \exists z \in X \cup Y.\ x \sqsubseteq z \wedge y \sqsubseteq z$

shows *directed* $(X \cup Y)$ $(\sqsubseteq)$
⟨*proof*⟩

lemma *directed-extend*:
assumes X: *directed* X $(\sqsubseteq)$ **and** Y: *directed* Y $(\sqsubseteq)$ **and** XY: $\forall x{\in}X.\ \forall y{\in}Y.\ x \sqsubseteq y$
shows *directed* $(X \cup Y)$ $(\sqsubseteq)$
⟨*proof*⟩

end

sublocale *connex* $\subseteq$ *directed*
⟨*proof*⟩

lemmas(**in** *connex*) *directed* = *directed-axioms*

lemma *monotone-directed-image*:
fixes *ir* (**infix** ‹$\preceq$› *50*) **and** *r* (**infix** ‹$\sqsubseteq$› *50*)
assumes *mono*: *monotone-on* I $(\preceq)$ $(\sqsubseteq)$ f **and** *dir*: *directed* I $(\preceq)$
shows *directed* $(f \text{ ‘ } I)$ $(\sqsubseteq)$
⟨*proof*⟩

definition *directed-set* A $(\sqsubseteq) \equiv \forall X \subseteq A.$ *finite* $X \longrightarrow (\exists b \in A.$ *bound* X $(\sqsubseteq)$ $b)$
for *less-eq* (**infix** ‹$\sqsubseteq$› *50*)

lemmas *directed-setI* = *directed-set-def*[*unfolded atomize-eq*, *THEN iffD2*, *rule-format*]
lemmas *directed-setD* = *directed-set-def*[*unfolded atomize-eq*, *THEN iffD1*, *rule-format*]

lemma *directed-imp-nonempty*:
fixes *less-eq* (**infix** ‹$\sqsubseteq$› *50*)
shows *directed-set* A $(\sqsubseteq) \Longrightarrow A \neq \{\}$
⟨*proof*⟩

lemma *directedD2*:
fixes *less-eq* (**infix** ‹$\sqsubseteq$› *50*)
assumes *dir*: *directed-set* A $(\sqsubseteq)$ **and** *xA*: $x \in A$ **and** *yA*: $y \in A$
shows $\exists z \in A.\ x \sqsubseteq z \land y \sqsubseteq z$
⟨*proof*⟩

lemma *monotone-directed-set-image*:
fixes *ir* (**infix** ‹$\preceq$› *50*) **and** *r* (**infix** ‹$\sqsubseteq$› *50*)
assumes *mono*: *monotone-on* I $(\preceq)$ $(\sqsubseteq)$ f **and** *dir*: *directed-set* I $(\preceq)$
shows *directed-set* $(f \text{ ‘ } I)$ $(\sqsubseteq)$
⟨*proof*⟩

lemma *directed-set-iff-extremed*:
fixes *less-eq* (**infix** ‹$\sqsubseteq$› *50*)

assumes *Dfin*: *finite D*
shows *directed-set D* (⊑) ⟷ *extremed D* (⊑)
⟨*proof*⟩

lemma (**in** *transitive*) *directed-iff-nonempty-pair-bounded*:
directed-set A (⊑) ⟷ *A* ≠ {} ∧ (∀ *x*∈*A*. ∀ *y*∈*A*. ∃ *z*∈*A*. *x* ⊑ *z* ∧ *y* ⊑ *z*)
(**is** *?l* ⟷ - ∧ *?r*)
⟨*proof*⟩

lemma (**in** *transitive*) *directed-set-iff-nonempty-directed*:
directed-set A (⊑) ⟷ *A* ≠ {} ∧ *directed A* (⊑)
⟨*proof*⟩

lemma (**in** *well-related-set*) *finite-sets-extremed*:
assumes *fin*: *finite X* **and** *X0*: *X* ≠ {} **and** *XA*: *X* ⊆ *A*
shows *extremed X* (⊑)
⟨*proof*⟩

lemma (**in** *well-related-set*) *directed-set*:
assumes *A0*: *A* ≠ {} **shows** *directed-set A* (⊑)
⟨*proof*⟩

lemma *prod-directed*:
fixes *leA* (**infix** ‹$\sqsubseteq_A$› *50*) **and** *leB* (**infix** ‹$\sqsubseteq_B$› *50*)
assumes *dir*: *directed X* (*rel-prod* ($\sqsubseteq_A$) ($\sqsubseteq_B$))
shows *directed* (*fst* ‘ *X*) ($\sqsubseteq_A$) **and** *directed* (*snd* ‘ *X*) ($\sqsubseteq_B$)
⟨*proof*⟩

class *dir* = *ord* + **assumes** *directed UNIV* (≤)
begin

sublocale *order*: *directed UNIV* (≤)
rewrites ⋀*x*. *x* ∈ *UNIV* ≡ *True*
and ⋀*X*. *X* ⊆ *UNIV* ≡ *True*
and ⋀*r*. *r* ↾ *UNIV* ≡ *r*
and ⋀*P*. *True* ∧ *P* ≡ *P*
and *Ball UNIV* ≡ *All*
and *Bex UNIV* ≡ *Ex*
and *sympartp* $(\leq)^{-}$ ≡ *sympartp* (≤)
and ⋀*P1*. (*True* ⟹ *PROP P1*) ≡ *PROP P1*
and ⋀*P1*. (*True* ⟹ *P1*) ≡ *Trueprop P1*
and ⋀*P1 P2*. (*True* ⟹ *PROP P1* ⟹ *PROP P2*) ≡ (*PROP P1* ⟹ *PROP P2*)
⟨*proof*⟩

end

class *filt* = *ord* +
assumes *directed UNIV* (≥)

begin

sublocale *order.dual*: *directed UNIV* (≥)
  rewrites $\bigwedge x.\ x \in UNIV \equiv True$
    and $\bigwedge X.\ X \subseteq UNIV \equiv True$
    and $\bigwedge r.\ r \upharpoonright UNIV \equiv r$
    and $\bigwedge P.\ True \wedge P \equiv P$
    and *Ball UNIV* ≡ *All*
    and *Bex UNIV* ≡ *Ex*
    and *sympartp* $(\leq)^{-} \equiv$ *sympartp* (≤)
    and $\bigwedge P1.\ (True \Longrightarrow PROP\ P1) \equiv PROP\ P1$
    and $\bigwedge P1.\ (True \Longrightarrow P1) \equiv Trueprop\ P1$
    and $\bigwedge P1\ P2.\ (True \Longrightarrow PROP\ P1 \Longrightarrow PROP\ P2) \equiv (PROP\ P1 \Longrightarrow PROP\ P2)$
  ⟨*proof*⟩

end

subclass (**in** *linqorder*) *dir*⟨*proof*⟩

subclass (**in** *linqorder*) *filt*⟨*proof*⟩

thm *order.directed-axioms*[**where** *'a* = *'a* ::*dir*]

thm *order.dual.directed-axioms*[**where** *'a* = *'a* ::*filt*]

end

4 Completeness of Relations

Here we formalize various order-theoretic completeness conditions.

theory *Complete-Relations*
  imports *Well-Relations Directedness*
begin

4.1 Completeness Conditions

Order-theoretic completeness demands certain subsets of elements to admit suprema or infima.

definition *complete* (‹-—*complete*›[*999*]*1000*) **where**
$\mathcal{C}$—*complete A* (⊑) ≡ ∀ *X* ⊆ *A*. $\mathcal{C}$ *X* (⊑) ⟶ (∃ *s*. *extreme-bound A* (⊑) *X s*) **for** *less-eq* (**infix** ‹⊑› *50*)

lemmas *completeI* = *complete-def*[*unfolded atomize-eq*, *THEN iffD2*, *rule-format*]
lemmas *completeD* = *complete-def*[*unfolded atomize-eq*, *THEN iffD1*, *rule-format*]
lemmas *completeE* = *complete-def*[*unfolded atomize-eq*, *THEN iffD1*, *rule-format*, *THEN exE*]

lemma *complete-cmono*: $CC \le DD \Longrightarrow DD{-}complete \le CC{-}complete$
⟨*proof*⟩

lemma *complete-subclass*:
fixes *less-eq* (**infix** ‹⊑› *50*)
assumes $\mathcal{C}{-}complete\ A\ (\sqsubseteq)$ **and** $\forall X \subseteq A.\ \mathcal{D}\ X\ (\sqsubseteq) \longrightarrow \mathcal{C}\ X\ (\sqsubseteq)$
shows $\mathcal{D}{-}complete\ A\ (\sqsubseteq)$
⟨*proof*⟩

lemma *complete-empty*[*simp*]: $\mathcal{C}{-}complete\ \{\}\ r \longleftrightarrow \neg\ \mathcal{C}\ \{\}\ r$ ⟨*proof*⟩

context
fixes *less-eq* :: $'a \Rightarrow 'a \Rightarrow bool$ (**infix** ‹⊑› *50*)
begin

Toppedness can be also seen as a completeness condition, since it is equivalent to saying that the universe has a supremum.

lemma *extremed-iff-UNIV-complete*: $extremed\ A\ (\sqsubseteq) \longleftrightarrow (\lambda X\ r.\ X = A){-}complete\ A\ (\sqsubseteq)$ (**is** $?l \longleftrightarrow ?r$)
⟨*proof*⟩

The dual notion of topped is called "pointed", equivalently ensuring a supremum of the empty set.

lemma *pointed-iff-empty-complete*: $extremed\ A\ (\sqsubseteq) \longleftrightarrow (\lambda X\ r.\ X = \{\}){-}complete\ A\ (\sqsubseteq)^-$
⟨*proof*⟩

Downward closure is topped.

lemma *dual-closure-is-extremed*:
assumes *bA*: $b \in A$ **and** $b \sqsubseteq b$
shows $extremed\ \{a \in A.\ a \sqsubseteq b\}\ (\sqsubseteq)$
⟨*proof*⟩

Downward closure preserves completeness.

lemma *dual-closure-is-complete*:
assumes *A*: $\mathcal{C}{-}complete\ A\ (\sqsubseteq)$ **and** *bA*: $b \in A$
shows $\mathcal{C}{-}complete\ \{x{\in}A.\ x \sqsubseteq b\}\ (\sqsubseteq)$
⟨*proof*⟩

interpretation *less-eq-dualize*⟨*proof*⟩

Upward closure preserves completeness, under a condition.

lemma *closure-is-complete*:
assumes *A*: $\mathcal{C}{-}complete\ A\ (\sqsubseteq)$ **and** *bA*: $b \in A$
and *Cb*: $\forall X \subseteq A.\ \mathcal{C}\ X\ (\sqsubseteq) \longrightarrow bound\ X\ (\sqsupseteq)\ b \longrightarrow \mathcal{C}\ (X \cup \{b\})\ (\sqsubseteq)$
shows $\mathcal{C}{-}complete\ \{x{\in}A.\ b \sqsubseteq x\}\ (\sqsubseteq)$
⟨*proof*⟩

lemma *biclosure-is-complete*:
assumes A: $\mathcal{C}$−*complete* A (⊑) **and** aA: $a \in A$ **and** bA: $b \in A$ **and** ab: $a \sqsubseteq b$
and Ca: $\forall X \subseteq A.\ \mathcal{C}\ X$ (⊑) $\longrightarrow$ *bound* X (⊒) $a \longrightarrow \mathcal{C}\ (X \cup \{a\})$ (⊑)
shows $\mathcal{C}$−*complete* $\{x{\in}A.\ a \sqsubseteq x \wedge x \sqsubseteq b\}$ (⊑)
⟨*proof*⟩

end

One of the most well-studied notion of completeness would be the semi-lattice condition: every pair of elements x and y has a supremum $x \sqcup y$ (not necessarily unique if the underlying relation is not antisymmetric).

definition *pair-complete* ≡ ($\lambda X\ r.\ \exists\, x\ y.\ X = \{x,y\}$)−*complete*

lemma *pair-completeI*:
assumes $\bigwedge x\ y.\ x \in A \Longrightarrow y \in A \Longrightarrow \exists\, s.$ *extreme-bound* $A\ r\ \{x,y\}\ s$
shows *pair-complete* $A\ r$
⟨*proof*⟩

lemma *pair-completeD*:
assumes *pair-complete* $A\ r$
shows $x \in A \Longrightarrow y \in A \Longrightarrow \exists\, s.$ *extreme-bound* $A\ r\ \{x,y\}\ s$
⟨*proof*⟩

context
fixes *less-eq* :: $'a \Rightarrow 'a \Rightarrow$ *bool* (**infix** ‹⊑› *50*)
begin

lemma *pair-complete-imp-directed*:
assumes *comp*: *pair-complete* A (⊑) **shows** *directed* A (⊑)
⟨*proof*⟩

end

lemma (**in** *connex*) *pair-complete*: *pair-complete* A (⊑)
⟨*proof*⟩

The next one assumes that every nonempty finite set has a supremum.

abbreviation *finite-complete* ≡ ($\lambda X\ r.$ *finite* $X \wedge X \neq \{\}$)−*complete*

lemma *finite-complete-le-pair-complete*: *finite-complete* ≤ *pair-complete*
⟨*proof*⟩

The next one assumes that every nonempty bounded set has a supremum. It is also called the Dedekind completeness.

abbreviation *conditionally-complete* A ≡ ($\lambda X\ r.\ \exists\, b \in A.$ *bound* $X\ r\ b \wedge X \neq \{\}$)−*complete* A

lemma *conditionally-complete-imp-nonempty-imp-ex-extreme-bound-iff-ex-bound*:

assumes *comp*: *conditionally-complete* A r
assumes $X \subseteq A$ **and** $X \neq \{\}$
shows $(\exists\, s.\ \textit{extreme-bound}\ A\ r\ X\ s) \longleftrightarrow (\exists\, b{\in}A.\ \textit{bound}\ X\ r\ b)$
⟨*proof*⟩

The ω-completeness condition demands a supremum for an ω-chain.

Directed completeness is an important notion in domain theory [1], asserting that every nonempty directed set has a supremum. Here, a set X is *directed* if any pair of two elements in X has a bound in X.

definition $\textit{directed-complete} \equiv (\lambda X\ r.\ \textit{directed}\ X\ r \wedge X \neq \{\})\textit{−complete}$

lemma *monotone-directed-complete*:
assumes *comp*: *directed-complete* A r
assumes *fI*: f ‘ $I \subseteq A$ **and** *dir*: *directed* I *ri* **and** *I0*: $I \neq \{\}$ **and** *mono*: *monotone-on* I *ri* r f
shows $\exists\, s.\ \textit{extreme-bound}\ A\ r\ (f\ \text{‘}\ I)\ s$
⟨*proof*⟩

lemma (**in** *reflexive*) *dual-closure-is-directed-complete*:
assumes *comp*: *directed−complete* A $(\sqsubseteq)$ **and** *bA*: $b \in A$
shows *directed−complete* $\{X \in A.\ b \sqsubseteq X\}$ $(\sqsubseteq)$
⟨*proof*⟩

The next one is quite complete, only the empty set may fail to have a supremum. The terminology follows [3], although there it is defined more generally depending on a cardinal α such that a nonempty set X of cardinality below α has a supremum.

abbreviation $\textit{semicomplete} \equiv (\lambda X\ r.\ X \neq \{\})\textit{−complete}$

lemma *semicomplete-nonempty-imp-extremed*:
$\textit{semicomplete}\ A\ r \Longrightarrow A \neq \{\} \Longrightarrow \textit{extremed}\ A\ r$
⟨*proof*⟩

lemma *connex-dual-semicomplete*: $\textit{semicomplete}\ \{C.\ \textit{connex}\ C\ r\}\ (\supseteq)$
⟨*proof*⟩

4.2 Pointed Ones

The term ‘pointed’ refers to the dual notion of toppedness, i.e., there is a global least element. This serves as the supremum of the empty set.

lemma *complete-sup*: $(CC \sqcup CC')\textit{−complete}\ A\ r \longleftrightarrow CC\textit{−complete}\ A\ r \wedge CC'\textit{−complete}\ A\ r$
⟨*proof*⟩

lemma *pointed-directed-complete*:
$\textit{directed−complete}\ A\ r \longleftrightarrow \textit{directed-complete}\ A\ r \wedge \textit{extremed}\ A\ r^{-}$
⟨*proof*⟩

"Bounded complete" refers to pointed conditional complete, but this notion is just the dual of semicompleteness. We prove this later.

abbreviation *bounded-complete A* ≡ (λ*X r*. ∃ *b*∈*A*. *bound X r b*)−*complete A*

4.3 Relations between Completeness Conditions

context
fixes *less-eq* :: *'a* ⇒ *'a* ⇒ *bool* (**infix** ‹⊑› *50*)
begin

interpretation *less-eq-dualize*⟨*proof*⟩

Pair-completeness implies that the universe is directed. Thus, with directed completeness implies toppedness.

proposition *directed-complete-pair-complete-imp-extremed*:
assumes *dc*: *directed-complete A* (⊑) **and** *pc*: *pair-complete A* (⊑) **and** *A*: *A* ≠ {}
shows *extremed A* (⊑)
⟨*proof*⟩

Semicomplete is conditional complete and topped.

proposition *semicomplete-iff-conditionally-complete-extremed*:
assumes *A*: *A* ≠ {}
shows *semicomplete A* (⊑) ⟷ *conditionally-complete A* (⊑) ∧ *extremed A* (⊑) (**is** *?l* ⟷ *?r*)
⟨*proof*⟩

proposition *complete-iff-pointed-semicomplete*:
⊤−*complete A* (⊑) ⟷ *semicomplete A* (⊑) ∧ *extremed A* (⊒) (**is** *?l* ⟷ *?r*)
⟨*proof*⟩

Conditional completeness only lacks top and bottom to be complete.

proposition *complete-iff-conditionally-complete-extremed-pointed*:
⊤−*complete A* (⊑) ⟷ *conditionally-complete A* (⊑) ∧ *extremed A* (⊑) ∧ *extremed A* (⊒)
⟨*proof*⟩

If the universe is directed, then every pair is bounded, and thus has a supremum. On the other hand, supremum gives an upper bound, witnessing directedness.

proposition *conditionally-complete-imp-pair-complete-iff-directed*:
assumes *comp*: *conditionally-complete A* (⊑)
shows *pair-complete A* (⊑) ⟷ *directed A* (⊑) (**is** *?l* ⟷ *?r*)
⟨*proof*⟩

end

4.4 Duality of Completeness Conditions

Conditional completeness is symmetric.

context fixes *less-eq* :: $'a \Rightarrow 'a \Rightarrow bool$ (**infix** ‹⊑› *50*)
begin

interpretation *less-eq-dualize*⟨*proof*⟩

lemma *conditionally-complete-dual*:
assumes *comp*: *conditionally-complete A* (⊑) **shows** *conditionally-complete A* (⊒)
⟨*proof*⟩

Full completeness is symmetric.

lemma *complete-dual*:
⊤−*complete A* (⊑) ⟹ ⊤−*complete A* (⊒)
⟨*proof*⟩

Now we show that bounded completeness is the dual of semicompleteness.

lemma *bounded-complete-iff-pointed-conditionally-complete*:
assumes *A*: $A \neq \{\}$
shows *bounded-complete A* (⊑) ⟷ *conditionally-complete A* (⊑) ∧ *extremed A* (⊒)
⟨*proof*⟩

proposition *bounded-complete-iff-dual-semicomplete*:
bounded-complete A (⊑) ⟷ *semicomplete A* (⊒)
⟨*proof*⟩

lemma *bounded-complete-imp-conditionally-complete*:
assumes *bounded-complete A* (⊑) **shows** *conditionally-complete A* (⊑)
⟨*proof*⟩

Completeness in downward-closure:

lemma *conditionally-complete-imp-semicomplete-in-dual-closure*:
assumes *A*: *conditionally-complete A* (⊑) **and** *bA*: $b \in A$
shows *semicomplete* $\{a \in A.\ a \sqsubseteq b\}$ (⊑)
⟨*proof*⟩

end

Completeness in intervals:

lemma *conditionally-complete-imp-complete-in-interval*:
fixes *less-eq* (**infix** ‹⊑› *50*)
assumes *comp*: *conditionally-complete A* (⊑) **and** *aA*: $a \in A$ **and** *bA*: $b \in A$
and *aa*: $a \sqsubseteq a$ **and** *ab*: $a \sqsubseteq b$
shows ⊤−*complete* $\{x \in A.\ a \sqsubseteq x \wedge x \sqsubseteq b\}$ (⊑)
⟨*proof*⟩

lemmas *connex-bounded-complete* = *connex-dual-semicomplete*[*folded bounded-complete-iff-dual-semicomplete*]

4.5 Completeness Results Requiring Order-Like Properties

Above results hold without any assumption on the relation. This part demands some order-like properties.

It is well known that in a semilattice, i.e., a pair-complete partial order, every finite nonempty subset of elements has a supremum. We prove the result assuming transitivity, but only that.

lemma (**in** *transitive*) *pair-complete-iff-finite-complete*:
pair-complete A ($\sqsubseteq$) $\longleftrightarrow$ *finite-complete* A ($\sqsubseteq$) (**is** *?l* $\longleftrightarrow$ *?r*)
⟨*proof*⟩

Gierz et al. [9] showed that a directed complete partial order is semicomplete if and only if it is also a semilattice. We generalize the claim so that the underlying relation is only transitive.

proposition(**in** *transitive*) *semicomplete-iff-directed-complete-pair-complete*:
semicomplete A ($\sqsubseteq$) $\longleftrightarrow$ *directed-complete* A ($\sqsubseteq$) $\wedge$ *pair-complete* A ($\sqsubseteq$) (**is** *?l* $\longleftrightarrow$ *?r*)
⟨*proof*⟩

The last argument in the above proof requires transitivity, but if we had reflexivity then x itself is a supremum of $\{x\}$ (see ⟦*reflexive ?A ?less-eq*; *?x* $\in$ *?A*⟧ $\Longrightarrow$ *extreme-bound ?A ?less-eq* {*?x*} *?x*) and so $x \sqsubseteq s$ would be immediate. Thus we can replace transitivity by reflexivity, but then pair-completeness does not imply finite completeness. We obtain the following result.

proposition (**in** *reflexive*) *semicomplete-iff-directed-complete-finite-complete*:
semicomplete A ($\sqsubseteq$) $\longleftrightarrow$ *directed-complete* A ($\sqsubseteq$) $\wedge$ *finite-complete* A ($\sqsubseteq$) (**is** *?l* $\longleftrightarrow$ *?r*)
⟨*proof*⟩

4.6 Relating to Classes

Isabelle's class *complete-lattice* is $\top$−*complete*.

lemma (**in** *complete-lattice*) $\top$−*complete UNIV* ($\leq$)
⟨*proof*⟩

4.7 Set-wise Completeness

lemma *Pow-extreme-bound*: $X \subseteq$ *Pow* $A \Longrightarrow$ *extreme-bound* (*Pow* A) ($\subseteq$) X ($\bigcup X$)
⟨*proof*⟩

lemma *Pow-complete*: $\mathcal{C}$−*complete* (*Pow* A) ($\subseteq$)
⟨*proof*⟩

lemma *directed-directed-Un*:
assumes *ch*: $XX \subseteq \{X.\ directed\ X\ r\}$ **and** *dir*: *directed* XX $(\subseteq)$
shows *directed* $(\bigcup XX)$ r
⟨*proof*⟩

lemmas *directed-connex-Un* = *directed-directed-Un*[*OF* - *connex.directed*]

lemma *directed-sets-directed-complete*:
assumes *cl*: $\forall DC.\ DC \subseteq AA \longrightarrow (\forall X{\in}DC.\ directed\ X\ r) \longrightarrow (\bigcup DC) \in AA$
shows *directed−complete* $\{X \in AA.\ directed\ X\ r\}$ $(\subseteq)$
⟨*proof*⟩

lemma *connex-directed-Un*:
assumes *ch*: $CC \subseteq \{C.\ connex\ C\ r\}$ **and** *dir*: *directed* CC $(\subseteq)$
shows *connex* $(\bigcup CC)$ r
⟨*proof*⟩

lemma *connex-is-directed-complete*: *directed−complete* $\{C.\ C \subseteq A \wedge connex\ C\ r\}$ $(\subseteq)$
⟨*proof*⟩

lemma (**in** *well-ordered-set*) *well-ordered-set-insert*:
assumes *aA*: *total-ordered-set* (*insert a A*) $(\sqsubseteq)$
shows *well-ordered-set* (*insert a A*) $(\sqsubseteq)$
⟨*proof*⟩

The following should be true in general, but here we use antisymmetry to avoid the axiom of choice.

lemma (**in** *antisymmetric*) *pointwise-connex-complete*:
assumes *comp*: *connex−complete* A $(\sqsubseteq)$
shows *connex−complete* $\{f.\ f\ `\ X \subseteq A\}$ (*pointwise* X $(\sqsubseteq)$)
⟨*proof*⟩

Our supremum/infimum coincides with those of Isabelle's *complete-lattice*.

lemma *complete-UNIV*: $\top$*−complete* (*UNIV*::'*a*::*complete-lattice set*) $(\leq)$
⟨*proof*⟩

context
fixes X :: '*a* :: *complete-lattice set*
begin

lemma *supremum-Sup*: *supremum* X $(\bigsqcup X)$
⟨*proof*⟩

lemmas *Sup-eq-The-supremum* = *order.dual.eq-The-extreme*[*OF supremum-Sup*]

lemma *supremum-eq-Sup*: *supremum* $X\ x \longleftrightarrow \bigsqcup X = x$
⟨*proof*⟩

lemma *infimum-Inf*:
  shows *infimum* X ($\bigsqcap X$)
⟨*proof*⟩

lemmas *Inf-eq-The-infimum* = *order.eq-The-extreme*[*OF infimum-Inf*]

lemma *infimum-eq-Inf*: *infimum* $X\ x \longleftrightarrow \bigsqcap X = x$
  ⟨*proof*⟩

end

end

theory *Fixed-Points*
  imports *Complete-Relations Directedness*
begin

5 Existence of Fixed Points in Complete Related Sets

The following proof is simplified and generalized from Stouti–Maaden [22]. We construct some set whose extreme bounds – if they exist, typically when the underlying related set is complete – are fixed points of a monotone or inflationary function on any related set. When the related set is attractive, those are actually the least fixed points. This generalizes [22], relaxing reflexivity and antisymmetry.

locale *fixed-point-proof* = *related-set* +
  fixes f
  assumes f: f ‘ $A \subseteq A$
begin

sublocale *less-eq-asymmetrize*⟨*proof*⟩

definition AA **where** $AA \equiv$
  $\{X.\ X \subseteq A \wedge f$ ‘ $X \subseteq X \wedge (\forall Y\ s.\ Y \subseteq X \longrightarrow$ *extreme-bound* $A\ (\sqsubseteq)\ Y\ s \longrightarrow s \in X)\}$

lemma AA-I:
  $X \subseteq A \Longrightarrow f$ ‘ $X \subseteq X \Longrightarrow (\bigwedge Y\ s.\ Y \subseteq X \Longrightarrow$ *extreme-bound* $A\ (\sqsubseteq)\ Y\ s \Longrightarrow s \in X) \Longrightarrow X \in AA$
  ⟨*proof*⟩

lemma AA-E:
  $X \in AA \Longrightarrow$
  $(X \subseteq A \Longrightarrow f$ ‘ $X \subseteq X \Longrightarrow (\bigwedge Y\ s.\ Y \subseteq X \Longrightarrow$ *extreme-bound* $A\ (\sqsubseteq)\ Y\ s \Longrightarrow s \in X) \Longrightarrow$ *thesis*$) \Longrightarrow$ *thesis*

⟨*proof*⟩

definition C **where** $C \equiv \bigcap AA$

lemma *A-AA*: $A \in AA$ ⟨*proof*⟩

lemma *C-AA*: $C \in AA$
⟨*proof*⟩

lemma *CA*: $C \subseteq A$ ⟨*proof*⟩

lemma *fC*: f ‘ $C \subseteq C$ ⟨*proof*⟩

context
fixes c **assumes** *Cc*: *extreme-bound* A $(\sqsubseteq)$ C c
begin

private lemma *cA*: $c \in A$ ⟨*proof*⟩ **lemma** *cC*: $c \in C$ ⟨*proof*⟩ **lemma** *fcC*: $f\ c \in C$ ⟨*proof*⟩ **lemma** *fcA*: $f\ c \in A$ ⟨*proof*⟩

lemma *qfp-as-extreme-bound*:
assumes *infl-mono*: $\forall x \in A.\ x \sqsubseteq f\ x \vee (\forall y \in A.\ y \sqsubseteq x \longrightarrow f\ y \sqsubseteq f\ x)$
shows $f\ c \sim c$
⟨*proof*⟩

lemma *extreme-qfp*:
assumes *attract*: $\forall q \in A.\ \forall x \in A.\ f\ q \sim q \longrightarrow x \sqsubseteq f\ q \longrightarrow x \sqsubseteq q$
and *mono*: *monotone-on* A $(\sqsubseteq)$ $(\sqsubseteq)$ f
shows *extreme* $\{q \in A.\ f\ q \sim q \vee f\ q = q\}$ $(\sqsupseteq)$ c
⟨*proof*⟩

end

lemma *ex-qfp*:
assumes *comp*: $CC{-}complete$ A $(\sqsubseteq)$ **and** C: CC C $(\sqsubseteq)$
and *infl-mono*: $\forall a \in A.\ a \sqsubseteq f\ a \vee (\forall b \in A.\ b \sqsubseteq a \longrightarrow f\ b \sqsubseteq f\ a)$
shows $\exists s \in A.\ f\ s \sim s$
⟨*proof*⟩

lemma *ex-extreme-qfp-fp*:
assumes *comp*: $CC{-}complete$ A $(\sqsubseteq)$ **and** C: CC C $(\sqsubseteq)$
and *attract*: $\forall q \in A.\ \forall x \in A.\ f\ q \sim q \longrightarrow x \sqsubseteq f\ q \longrightarrow x \sqsubseteq q$
and *mono*: *monotone-on* A $(\sqsubseteq)$ $(\sqsubseteq)$ f
shows $\exists c.$ *extreme* $\{q \in A.\ f\ q \sim q \vee f\ q = q\}$ $(\sqsupseteq)$ c
⟨*proof*⟩

lemma *ex-extreme-qfp*:
assumes *comp*: $CC{-}complete$ A $(\sqsubseteq)$ **and** C: CC C $(\sqsubseteq)$
and *attract*: $\forall q \in A.\ \forall x \in A.\ f\ q \sim q \longrightarrow x \sqsubseteq f\ q \longrightarrow x \sqsubseteq q$

and *mono*: *monotone-on* A (⊑) (⊑) f
shows $\exists\, c.$ *extreme* $\{q \in A.\ f\ q \sim q\}$ (⊒) c
⟨*proof*⟩

end

context
fixes *less-eq* :: $'a \Rightarrow 'a \Rightarrow$ *bool* (**infix** ‹⊑› *50*) **and** A :: $'a$ *set* **and** f
assumes f: f ‘ $A \subseteq A$
begin

interpretation *less-eq-symmetrize*⟨*proof*⟩
interpretation *fixed-point-proof* A (⊑) f ⟨*proof*⟩

theorem *complete-infl-mono-imp-ex-qfp*:
assumes *comp*: ⊤−*complete* A (⊑) **and** *infl-mono*: $\forall\, a{\in}A.\ a \sqsubseteq f\ a \vee (\forall\, b{\in}A.\ b \sqsubseteq a \longrightarrow f\ b \sqsubseteq f\ a)$
shows $\exists\, s{\in}A.\ f\ s \sim s$
⟨*proof*⟩

end

corollary (**in** *antisymmetric*) *complete-infl-mono-imp-ex-fp*:
assumes *comp*: ⊤−*complete* A (⊑) **and** f: f ‘ $A \subseteq A$
and *infl-mono*: $\forall\, a{\in}A.\ a \sqsubseteq f\ a \vee (\forall\, b{\in}A.\ b \sqsubseteq a \longrightarrow f\ b \sqsubseteq f\ a)$
shows $\exists\, s \in A.\ f\ s = s$
⟨*proof*⟩

context *semiattractive* **begin**

interpretation *less-eq-symmetrize*⟨*proof*⟩

theorem *complete-mono-imp-ex-extreme-qfp*:
assumes *comp*: ⊤−*complete* A (⊑) **and** f: f ‘ $A \subseteq A$
and *mono*: *monotone-on* A (⊑) (⊑) f
shows $\exists\, s.$ *extreme* $\{p \in A.\ f\ p \sim p\}$ (⊑) s
⟨*proof*⟩

end

corollary (**in** *antisymmetric*) *complete-mono-imp-ex-extreme-fp*:
assumes *comp*: ⊤−*complete* A (⊑) **and** f: f ‘ $A \subseteq A$
and *mono*: *monotone-on* A (⊑) (⊑) f
shows $\exists\, s.$ *extreme* $\{s \in A.\ f\ s = s\}$ $(\sqsubseteq)^-$ s
⟨*proof*⟩

6 Fixed Points in Well-Complete Antisymmetric Sets

In this section, we prove that an inflationary or monotone map over a well-complete antisymmetric set has a fixed point.

In order to formalize such a theorem in Isabelle, we followed Grall's [11] elementary proof for Bourbaki–Witt and Markowsky's theorems. His idea is to consider well-founded derivation trees over A, where from a set $C \subseteq A$ of premises one can derive $f\ (\bigsqcup C)$ if C is a chain. The main observation is as follows: Let D be the set of all the derivable elements; that is, for each $d \in D$ there exists a well-founded derivation whose root is d. It is shown that D is a chain, and hence one can build a derivation yielding $f\ (\bigsqcup D)$, and $f\ (\bigsqcup D)$ is shown to be a fixed point.

lemma *bound-monotone-on*:
assumes *mono*: *monotone-on* $A\ r\ s\ f$ **and** *XA*: $X \subseteq A$ **and** *aA*: $a \in A$ **and** *rXa*: *bound* $X\ r\ a$
shows *bound* $(f`X)\ s\ (f\ a)$
⟨*proof*⟩

context *fixed-point-proof* **begin**

To avoid the usage of the axiom of choice, we carefully define derivations so that any derivable element determines its lower set. This led to the following definition:

definition *derivation* $X \equiv X \subseteq A \land$ *well-ordered-set* $X\ (\sqsubseteq) \land$
$(\forall x \in X.$ *let* $Y = \{y \in X.\ y \sqsubset x\}$ *in*
$(\exists y.$ *extreme* $Y\ (\sqsubseteq)\ y \land x = f\ y) \lor$
$f\ `\ Y \subseteq Y \land$ *extreme-bound* $A\ (\sqsubseteq)\ Y\ x)$

lemma *empty-derivation*: *derivation* $\{\}$ ⟨*proof*⟩

lemma assumes *derivation* P
shows *derivation-A*: $P \subseteq A$ **and** *derivation-well-ordered*: *well-ordered-set* $P\ (\sqsubseteq)$
⟨*proof*⟩

lemma *derivation-cases*[*consumes 2*, *case-names suc lim*]:
assumes *derivation* X **and** $x \in X$
and $\bigwedge Y\ y.\ Y = \{y \in X.\ y \sqsubset x\} \Longrightarrow$ *extreme* $Y\ (\sqsubseteq)\ y \Longrightarrow x = f\ y \Longrightarrow$ *thesis*
and $\bigwedge Y.\ Y = \{y \in X.\ y \sqsubset x\} \Longrightarrow f\ `\ Y \subseteq Y \Longrightarrow$ *extreme-bound* $A\ (\sqsubseteq)\ Y\ x$ $\Longrightarrow$ *thesis*
shows *thesis*
⟨*proof*⟩

definition *derivable* $x \equiv \exists X.$ *derivation* $X \land x \in X$

lemma *derivableI*[*intro?*]: *derivation* $X \Longrightarrow x \in X \Longrightarrow$ *derivable* x ⟨*proof*⟩

lemma *derivableE*: *derivable* $x \Longrightarrow (\bigwedge P.$ *derivation* $P \Longrightarrow x \in P \Longrightarrow$ *thesis*$) \Longrightarrow$ *thesis*
⟨*proof*⟩

lemma *derivable-A*: *derivable* $x \Longrightarrow x \in A$ ⟨*proof*⟩

lemma *UN-derivations-eq-derivable*: $(\bigcup\{P.$ *derivation* $P\}) = \{x.$ *derivable* $x\}$
⟨*proof*⟩

end

locale *fixed-point-proof2* = *fixed-point-proof* + *antisymmetric* +
assumes *derivation-infl*: $\forall X\ x\ y.$ *derivation* $X \longrightarrow x \in X \longrightarrow y \in X \longrightarrow x \sqsubseteq y \longrightarrow x \sqsubseteq f\ y$
and *derivation-f-refl*: $\forall X\ x.$ *derivation* $X \longrightarrow x \in X \longrightarrow f\ x \sqsubseteq f\ x$
begin

lemma *derivation-lim*:
assumes *P*: *derivation* P **and** *fP*: f ‘ $P \subseteq P$ **and** *Pp*: *extreme-bound* A $(\sqsubseteq)$ P p
shows *derivation* $(P \cup \{p\})$
⟨*proof*⟩

lemma *derivation-suc*:
assumes *P*: *derivation* P **and** *Pp*: *extreme* P $(\sqsubseteq)$ p **shows** *derivation* $(P \cup \{f\ p\})$
⟨*proof*⟩

lemma *derivable-closed*:
assumes *x*: *derivable* x **shows** *derivable* $(f\ x)$
⟨*proof*⟩

The following lemma is derived from Grall's proof. We simplify the claim so that we consider two elements from one derivation, instead of two derivations.

lemma *derivation-useful*:
assumes *X*: *derivation* X **and** *xX*: $x \in X$ **and** *yX*: $y \in X$ **and** *xy*: $x \sqsubset y$
shows $f\ x \sqsubseteq y$
⟨*proof*⟩

Next one is the main lemma of this section, stating that elements from two possibly different derivations are comparable, and moreover the lower one is in the derivation of the upper one. The latter claim, not found in Grall's proof, is crucial in proving that the union of all derivations is well-related.

lemma *derivations-cross-compare*:
assumes *X*: *derivation* X **and** *Y*: *derivation* Y **and** *xX*: $x \in X$ **and** *yY*: $y \in Y$
shows $(x \sqsubset y \wedge x \in Y) \vee x = y \vee (y \sqsubset x \wedge y \in X)$
⟨*proof*⟩

sublocale *derivable*: *well-ordered-set* $\{x.\ derivable\ x\}$ $(\sqsubseteq)$
⟨*proof*⟩

lemma *pred-unique*:
 assumes X: *derivation* X **and** xX: $x \in X$
 shows $\{z \in X.\ z \sqsubset x\} = \{z.\ derivable\ z \wedge z \sqsubset x\}$
⟨*proof*⟩

The set of all derivable elements is itself a derivation.

lemma *derivation-derivable*: *derivation* $\{x.\ derivable\ x\}$
 ⟨*proof*⟩

Finally, if the set of all derivable elements admits a supremum, then it is a fixed point.

context
 fixes p
 assumes p: *extreme-bound* A $(\sqsubseteq)$ $\{x.\ derivable\ x\}$ p
begin

lemma *sup-derivable-derivable*: *derivable* p
 ⟨*proof*⟩ **lemmas** *sucp* = *sup-derivable-derivable*[*THEN derivable-closed*]

lemma *sup-derivable-prefixed*: $f\ p \sqsubseteq p$ ⟨*proof*⟩

lemma *sup-derivable-postfixed*: $p \sqsubseteq f\ p$
 ⟨*proof*⟩

lemma *sup-derivable-qfp*: $f\ p \sim p$
 ⟨*proof*⟩

lemma *sup-derivable-fp*: $f\ p = p$
 ⟨*proof*⟩

end

end

The assumptions are satisfied by monotone functions.

context *fixed-point-proof* **begin**

context
 assumes *ord*: *antisymmetric* A $(\sqsubseteq)$
begin

interpretation *antisymmetric* ⟨*proof*⟩

context
 assumes *mono*: *monotone-on* A $(\sqsubseteq)$ $(\sqsubseteq)$ f
begin

interpretation *fixed-point-proof2*
⟨*proof*⟩

lemmas *mono-imp-fixed-point-proof2* = *fixed-point-proof2-axioms*

corollary *mono-imp-sup-derivable-fp*:
  assumes *p*: *extreme-bound* A (⊑) {*x*. *derivable* *x*} *p*
  shows $f\ p = p$
  ⟨*proof*⟩

lemma *mono-imp-sup-derivable-lfp*:
  assumes *p*: *extreme-bound* A (⊑) {*x*. *derivable* *x*} *p*
  shows *extreme* $\{q \in A.\ f\ q = q\}$ (⊒) *p*
⟨*proof*⟩

lemma *mono-imp-ex-least-fp*:
  assumes *comp*: *well-related-set−complete* A (⊑)
  shows $\exists p.$ *extreme* $\{q \in A.\ f\ q = q\}$ (⊒) *p*
⟨*proof*⟩

end

end

end

Bourbaki-Witt Theorem on well-complete pseudo-ordered set:

theorem (**in** *pseudo-ordered-set*) *well-complete-infl′-imp-ex-fp*:
  assumes *comp*: *well-related-set−complete* A (⊑)
    and *f*: f ‘ $A \subseteq A$ **and** *infl*: $\forall x \in A.\ \forall y \in A.\ x \sqsubseteq y \longrightarrow x \sqsubseteq f\ y$
  shows $\exists p \in A.\ f\ p = p$
⟨*proof*⟩

Bourbaki-Witt Theorem on posets:

corollary (**in** *partially-ordered-set*) *well-complete-infl-imp-ex-fp*:
  assumes *comp*: *well-related-set−complete* A (⊑)
    and *f*: f ‘ $A \subseteq A$ **and** *infl*: $\forall x \in A.\ x \sqsubseteq f\ x$
  shows $\exists p \in A.\ f\ p = p$
⟨*proof*⟩

7 Completeness of (Quasi-)Fixed Points

We now prove that, under attractivity, the set of quasi-fixed points is complete.

definition *setwise* **where** *setwise* $r\ X\ Y \equiv \forall x{\in}X.\ \forall y{\in}Y.\ r\ x\ y$

lemmas *setwiseI*[*intro*] = *setwise-def*[*unfolded atomize-eq*, *THEN iffD2*, *rule-format*]

lemmas *setwiseE*[*elim*] = *setwise-def*[*unfolded atomize-eq*, *THEN iffD1*, *elim-format*, *rule-format*]

context *fixed-point-proof* **begin**

abbreviation *setwise-less-eq* (**infix** ‹$\sqsubseteq^s$› *50*) **where** $(\sqsubseteq^s) \equiv$ *setwise* $(\sqsubseteq)$

7.1 Least Quasi-Fixed Points for Attractive Relations.

lemma *attract-mono-imp-least-qfp*:
  assumes *attract*: *attractive* A $(\sqsubseteq)$
    and *comp*: *well-related-set−complete* A $(\sqsubseteq)$
    and *mono*: *monotone-on* A $(\sqsubseteq)$ $(\sqsubseteq)$ f
  shows $\exists\, c.$ *extreme* $\{p \in A.\ f\ p \sim p \lor f\ p = p\}$ $(\sqsupseteq)$ $c \land f\ c \sim c$
⟨*proof*⟩

7.2 General Completeness

lemma *attract-mono-imp-fp-qfp-complete*:
  assumes *attract*: *attractive* A $(\sqsubseteq)$
    and *comp*: *CC−complete* A $(\sqsubseteq)$
    and *wr-CC*: $\forall\, C \subseteq A.$ *well-related-set* C $(\sqsubseteq) \longrightarrow CC\ C\ (\sqsubseteq)$
    and *extend*: $\forall\, X\ Y.\ CC\ X\ (\sqsubseteq) \longrightarrow CC\ Y\ (\sqsubseteq) \longrightarrow X \sqsubseteq^s Y \longrightarrow CC\ (X \cup Y)$
$(\sqsubseteq)$
    and *mono*: *monotone-on* A $(\sqsubseteq)$ $(\sqsubseteq)$ f
    and *P*: $P \subseteq \{x \in A.\ f\ x = x\}$
  shows *CC−complete* $(\{q \in A.\ f\ q \sim q\} \cup P)$ $(\sqsubseteq)$
⟨*proof*⟩

lemma *attract-mono-imp-qfp-complete*:
  assumes *attractive* A $(\sqsubseteq)$
    and *CC−complete* A $(\sqsubseteq)$
    and $\forall\, C \subseteq A.$ *well-related-set* C $(\sqsubseteq) \longrightarrow CC\ C\ (\sqsubseteq)$
    and $\forall\, X\ Y.\ CC\ X\ (\sqsubseteq) \longrightarrow CC\ Y\ (\sqsubseteq) \longrightarrow X \sqsubseteq^s Y \longrightarrow CC\ (X \cup Y)\ (\sqsubseteq)$
    and *monotone-on* A $(\sqsubseteq)$ $(\sqsubseteq)$ f
  shows *CC−complete* $\{p \in A.\ f\ p \sim p\}$ $(\sqsubseteq)$
  ⟨*proof*⟩

lemma *antisym-mono-imp-fp-complete*:
  assumes *anti*: *antisymmetric* A $(\sqsubseteq)$
    and *comp*: *CC−complete* A $(\sqsubseteq)$
    and *wr-CC*: $\forall\, C \subseteq A.$ *well-related-set* C $(\sqsubseteq) \longrightarrow CC\ C\ (\sqsubseteq)$
    and *extend*: $\forall\, X\ Y.\ CC\ X\ (\sqsubseteq) \longrightarrow CC\ Y\ (\sqsubseteq) \longrightarrow X \sqsubseteq^s Y \longrightarrow CC\ (X \cup Y)$
$(\sqsubseteq)$
    and *mono*: *monotone-on* A $(\sqsubseteq)$ $(\sqsubseteq)$ f
  shows *CC−complete* $\{p \in A.\ f\ p = p\}$ $(\sqsubseteq)$
⟨*proof*⟩

end

7.3 Instances

7.3.1 Instances under attractivity

context *attractive* **begin**

interpretation *less-eq-symmetrize*⟨*proof*⟩

Full completeness

theorem *mono-imp-qfp-complete*:
assumes *comp*: $\top-complete\ A\ (\sqsubseteq)$ **and** *f*: $f\ `\ A \subseteq A$ **and** *mono*: *monotone-on* $A\ (\sqsubseteq)\ (\sqsubseteq)\ f$
shows $\top-complete\ \{p \in A.\ f\ p \sim p\}\ (\sqsubseteq)$
⟨*proof*⟩

Connex completeness

theorem *mono-imp-qfp-connex-complete*:
assumes *comp*: $connex-complete\ A\ (\sqsubseteq)$
and *f*: $f\ `\ A \subseteq A$ **and** *mono*: *monotone-on* $A\ (\sqsubseteq)\ (\sqsubseteq)\ f$
shows $connex-complete\ \{p \in A.\ f\ p \sim p\}\ (\sqsubseteq)$
⟨*proof*⟩

Directed completeness

theorem *mono-imp-qfp-directed-complete*:
assumes *comp*: $directed-complete\ A\ (\sqsubseteq)$
and *f*: $f\ `\ A \subseteq A$ **and** *mono*: *monotone-on* $A\ (\sqsubseteq)\ (\sqsubseteq)\ f$
shows $directed-complete\ \{p \in A.\ f\ p \sim p\}\ (\sqsubseteq)$
⟨*proof*⟩

Well Completeness

theorem *mono-imp-qfp-well-complete*:
assumes *comp*: *well-related-set*$-complete\ A\ (\sqsubseteq)$
and *f*: $f\ `\ A \subseteq A$ **and** *mono*: *monotone-on* $A\ (\sqsubseteq)\ (\sqsubseteq)\ f$
shows *well-related-set*$-complete\ \{p \in A.\ f\ p \sim p\}\ (\sqsubseteq)$
⟨*proof*⟩

end

7.3.2 Usual instances under antisymmetry

context *antisymmetric* **begin**

Knaster–Tarski

theorem *mono-imp-fp-complete*:
assumes *comp*: $\top-complete\ A\ (\sqsubseteq)$
and *f*: $f\ `\ A \subseteq A$ **and** *mono*: *monotone-on* $A\ (\sqsubseteq)\ (\sqsubseteq)\ f$
shows $\top-complete\ \{p \in A.\ f\ p = p\}\ (\sqsubseteq)$
⟨*proof*⟩

Markowsky 1976

theorem *mono-imp-fp-connex-complete*:
  assumes *comp*: *connex−complete A* (⊑)
    and *f*: *f ‘ A* ⊆ *A* **and** *mono*: *monotone-on A* (⊑) (⊑) *f*
  shows *connex−complete* {*p* ∈ *A*. *f p* = *p*} (⊑)
⟨*proof*⟩

Pataraia

theorem *mono-imp-fp-directed-complete*:
  assumes *comp*: *directed−complete A* (⊑)
    and *f*: *f ‘ A* ⊆ *A* **and** *mono*: *monotone-on A* (⊑) (⊑) *f*
  shows *directed−complete* {*p* ∈ *A*. *f p* = *p*} (⊑)
⟨*proof*⟩

Bhatta & George 2011

theorem *mono-imp-fp-well-complete*:
  assumes *comp*: *well-related-set−complete A* (⊑)
    and *f*: *f ‘ A* ⊆ *A* **and** *mono*: *monotone-on A* (⊑) (⊑) *f*
  shows *well-related-set−complete* {*p* ∈ *A*. *f p* = *p*} (⊑)
⟨*proof*⟩

end

end
theory *Continuity*
  imports *Complete-Relations*
begin

7.4 Scott Continuity, ω-Continuity

In this Section, we formalize Scott continuity and ω-continuity. We then prove that a Scott continuous map is ω-continuous and that an ω-continuous map is “nearly” monotone.

definition *continuous* (‹-−*continuous*› [*1000*]*1000*) **where**
  𝒞−*continuous A* (⊑) *B* (⊴) *f* ≡
   f ‘ A ⊆ *B* ∧
    (∀ *X s*. 𝒞 *X* (⊑) ⟶ *X* ≠ {} ⟶ *X* ⊆ *A* ⟶ *extreme-bound A* (⊑) *X s* ⟶ *extreme-bound B* (⊴) (*f‘X*) (*f s*))
  for *leA* (**infix** ‹⊑› *50*) **and** *leB* (**infix** ‹⊴› *50*)

lemmas *continuousI*[*intro?*] =
  continuous-def[*unfolded atomize-eq*, *THEN iffD2*, *unfolded conj-imp-eq-imp-imp*, *rule-format*]

lemmas *continuousE* =
  continuous-def[*unfolded atomize-eq*, *THEN iffD1*, *elim-format*, *unfolded conj-imp-eq-imp-imp*, *rule-format*]

lemma
  fixes *prec-eq* (**infix** ‹⪯› *50*) **and** *less-eq* (**infix** ‹⊑› *50*)

assumes $\mathcal{C}$−*continuous* I (⪯) A (⊑) f
shows *continuous-carrierD*[*dest*]: f ' $I \subseteq A$
and *continuousD*: $\mathcal{C}$ X (⪯) $\Longrightarrow X \neq \{\} \Longrightarrow X \subseteq I \Longrightarrow$ *extreme-bound* I (⪯) X $b \Longrightarrow$ *extreme-bound* A (⊑) (f ' X) (f b)
⟨*proof*⟩

lemma *continuous-comp*:
fixes *leA* (**infix** ‹$\sqsubseteq_A$› *50*) **and** *leB* (**infix** ‹$\sqsubseteq_B$› *50*) **and** *leC* (**infix** ‹$\sqsubseteq_C$› *50*)
assumes *KfL*: $\forall X \subseteq A$. $\mathcal{K}$ X ($\sqsubseteq_A$) $\longrightarrow \mathcal{L}$ (f ' X) ($\sqsubseteq_B$)
assumes *f*: $\mathcal{K}$−*continuous* A ($\sqsubseteq_A$) B ($\sqsubseteq_B$) f **and** *g*: $\mathcal{L}$−*continuous* B ($\sqsubseteq_B$) C ($\sqsubseteq_C$) g
shows $\mathcal{K}$−*continuous* A ($\sqsubseteq_A$) C ($\sqsubseteq_C$) ($g \circ f$)
⟨*proof*⟩

lemma *continuous-comp-top*:
fixes *leA* (**infix** ‹$\sqsubseteq_A$› *50*) **and** *leB* (**infix** ‹$\sqsubseteq_B$› *50*) **and** *leC* (**infix** ‹$\sqsubseteq_C$› *50*)
assumes *f*: $\mathcal{K}$−*continuous* A ($\sqsubseteq_A$) B ($\sqsubseteq_B$) f **and** *g*: ⊤−*continuous* B ($\sqsubseteq_B$) C ($\sqsubseteq_C$) g
shows $\mathcal{K}$−*continuous* A ($\sqsubseteq_A$) C ($\sqsubseteq_C$) ($g \circ f$)
⟨*proof*⟩

lemma *id-continuous*:
fixes *leA* (**infix** ‹$\sqsubseteq_A$› *50*)
shows $\mathcal{K}$−*continuous* A ($\sqsubseteq_A$) A ($\sqsubseteq_A$) ($\lambda x.\ x$)
⟨*proof*⟩

lemma *cst-continuous*:
fixes *leA* (**infix** ‹$\sqsubseteq_A$› *50*) **and** *leB* (**infix** ‹$\sqsubseteq_B$› *50*)
assumes $b \in B$ **and** *bb*: $b \sqsubseteq_B b$
shows $\mathcal{K}$−*continuous* A ($\sqsubseteq_A$) B ($\sqsubseteq_B$) ($\lambda x.\ b$)
⟨*proof*⟩

lemma *continuous-cmono*:
assumes *CD*: $\mathcal{C} \leq \mathcal{D}$ **shows** $\mathcal{D}$−*continuous* ≤ $\mathcal{C}$−*continuous*
⟨*proof*⟩

context
fixes *prec-eq* :: ′*i* ⇒ ′*i* ⇒ *bool* (**infix** ‹⪯› *50*) **and** *less-eq* :: ′*a* ⇒ ′*a* ⇒ *bool* (**infix** ‹⊑› *50*)
begin

lemma *continuous-subclass*:
assumes *CD*: $\forall X \subseteq I.\ X \neq \{\} \longrightarrow \mathcal{C}\ X$ (⪯) $\longrightarrow \mathcal{D}\ X$ (⪯) **and** *cont*: $\mathcal{D}$−*continuous* I (⪯) A (⊑) f
shows $\mathcal{C}$−*continuous* I (⪯) A (⊑) f
⟨*proof*⟩

lemma *chain-continuous-imp-well-continuous*:

assumes *cont*: *connex−continuous I* (⪯) *A* (⊑) *f*
shows *well-related-set−continuous I* (⪯) *A* (⊑) *f*
⟨*proof*⟩

lemma *well-continuous-imp-omega-continous*:
assumes *cont*: *well-related-set−continuous I* (⪯) *A* (⊑) *f*
shows *omega-chain−continuous I* (⪯) *A* (⊑) *f*
⟨*proof*⟩

end

abbreviation *scott-continuous I* (⪯) ≡ *directed-set−continuous I* (⪯)
for *prec-eq* (**infix** ‹⪯› *50*)

lemma *scott-continuous-imp-well-continuous*:
fixes *prec-eq* :: *'i* ⇒ *'i* ⇒ *bool* (**infix** ‹⪯› *50*) **and** *less-eq* :: *'a* ⇒ *'a* ⇒ *bool*
(**infix** ‹⊑› *50*)
assumes *cont*: *scott-continuous I* (⪯) *A* (⊑) *f*
shows *well-related-set−continuous I* (⪯) *A* (⊑) *f*
⟨*proof*⟩

lemmas *scott-continuous-imp-omega-continuous* =
scott-continuous-imp-well-continuous[*THEN well-continuous-imp-omega-continous*]

7.4.1 Continuity implies monotonicity

lemma *continuous-imp-mono-refl*:
fixes *prec-eq* (**infix** ‹⪯› *50*) **and** *less-eq* (**infix** ‹⊑› *50*)
assumes *cont*: *𝒞−continuous I* (⪯) *A* (⊑) *f* **and** *xyC*: *𝒞* {*x*,*y*} (⪯)
and *xy*: *x* ⪯ *y* **and** *yy*: *y* ⪯ *y*
and *x*: *x* ∈ *I* **and** *y*: *y* ∈ *I*
shows *f x* ⊑ *f y*
⟨*proof*⟩

lemma *omega-continuous-imp-mono-refl*:
fixes *prec-eq* (**infix** ‹⪯› *50*) **and** *less-eq* (**infix** ‹⊑› *50*)
assumes *cont*: *omega-chain−continuous I* (⪯) *A* (⊑) *f*
and *xx*: *x* ⪯ *x* **and** *xy*: *x* ⪯ *y* **and** *yy*: *y* ⪯ *y*
and *x*: *x* ∈ *I* **and** *y*: *y* ∈ *I*
shows *f x* ⊑ *f y*
⟨*proof*⟩

context *reflexive* **begin**

lemma *continuous-imp-monotone-on*:
fixes *leB* (**infix** ‹⊴› *50*)
assumes *cont*: *𝒞−continuous A* (⊑) *B* (⊴) *f*
and *II*: ∀ *i* ∈ *A*. ∀ *j* ∈ *A*. *i* ⊑ *j* ⟶ *𝒞* {*i*,*j*} (⊑)
shows *monotone-on A* (⊑) (⊴) *f*

⟨*proof*⟩

lemma *well-complete-imp-monotone-on*:
  fixes *leB* (**infix** ‹⊴› *50*)
  assumes *cont*: *well-related-set−continuous* A (⊑) B (⊴) f
  shows *monotone-on* A (⊑) (⊴) f
  ⟨*proof*⟩

end

end
theory *Kleene-Fixed-Point*
  imports *Complete-Relations Continuity*
begin

8 Iterative Fixed Point Theorem

Kleene's fixed-point theorem states that, for a pointed directed complete partial order $\langle A, \sqsubseteq\rangle$ and a Scott-continuous map $f : A \to A$, the supremum of $\{f^n(\bot) \mid n \in \mathbb{N}\}$ exists in A and is a least fixed point. Mashburn [17] generalized the result so that $\langle A, \sqsubseteq\rangle$ is a ω-complete partial order and f is ω-continuous.

In this section we further generalize the result and show that for ω-complete relation $\langle A, \sqsubseteq\rangle$ and for every bottom element $\bot \in A$, the set $\{f^n(\bot) \mid n \in \mathbb{N}\}$ has suprema (not necessarily unique, of course) and, they are quasi-fixed points. Moreover, if (⊑) is attractive, then the suprema are precisely the least quasi-fixed points.

8.1 Existence of Iterative Fixed Points

The first part of Kleene's theorem demands to prove that the set $\{f^n(\bot) \mid n \in \mathbb{N}\}$ has a supremum and that all such are quasi-fixed points. We prove this claim without assuming anything on the relation ⊑ besides ω-completeness and one bottom element.

notation *compower* (‹-^-›[*1000*,*1000*]*1000*)

lemma *monotone-on-funpow*: **assumes** f: f ‘ $A \subseteq A$ **and** *mono*: *monotone-on* A r r f
  shows *monotone-on* A r r (f^n)
⟨*proof*⟩

no-notation *bot* (‹⊥›)

context
  fixes A **and** *less-eq* (**infix** ‹⊑› *50*) **and** *bot* (‹⊥›) **and** f
  assumes *bot*: $\bot \in A$ $\forall q \in A.\ \bot \sqsubseteq q$

assumes *cont*: *omega-chain−continuous* A (⊑) A (⊑) f
begin

interpretation *less-eq-symmetrize*⟨*proof*⟩ **lemma** *f*: f ‘ $A \subseteq A$ ⟨*proof*⟩ **abbreviation**(*input*) *Fn* ≡ {f^n ⊥ |. n :: *nat*}

private lemma *fn-ref*: f^n ⊥ ⊑ f^n ⊥ **and** *fnA*: f^n ⊥ ∈ A
⟨*proof*⟩ **lemma** *FnA*: *Fn* ⊆ A ⟨*proof*⟩ **lemma** *Fn-chain*: *omega-chain Fn* (⊑)
⟨*proof*⟩ **lemma** *Fn*: *Fn* = *range* (λn. f^n ⊥) ⟨*proof*⟩

theorem *kleene-qfp*:
assumes *q*: *extreme-bound* A (⊑) *Fn* q
shows $f\ q \sim q$
⟨*proof*⟩

lemma *ex-kleene-qfp*:
assumes *comp*: *omega-chain−complete* A (⊑)
shows ∃ p. *extreme-bound* A (⊑) *Fn* p
⟨*proof*⟩

8.2 Iterative Fixed Points are Least.

Kleene's theorem also states that the quasi-fixed point found this way is a least one. Again, attractivity is needed to prove this statement.

lemma *kleene-qfp-is-least*:
assumes *attract*: $\forall q \in A.\ \forall x \in A.\ f\ q \sim q \longrightarrow x \sqsubseteq f\ q \longrightarrow x \sqsubseteq q$
assumes *q*: *extreme-bound* A (⊑) *Fn* q
shows *extreme* $\{s \in A.\ f\ s \sim s\}$ (⊒) q
⟨*proof*⟩

lemma *kleene-qfp-iff-least*:
assumes *comp*: *omega-chain−complete* A (⊑)
assumes *attract*: $\forall q \in A.\ \forall x \in A.\ f\ q \sim q \longrightarrow x \sqsubseteq f\ q \longrightarrow x \sqsubseteq q$
assumes *dual-attract*: $\forall p \in A.\ \forall q \in A.\ \forall x \in A.\ p \sim q \longrightarrow q \sqsubseteq x \longrightarrow p \sqsubseteq x$
shows *extreme-bound* A (⊑) *Fn* = *extreme* $\{s \in A.\ f\ s \sim s\}$ (⊒)
⟨*proof*⟩

end

context *attractive* **begin**

interpretation *less-eq-dualize* + *less-eq-symmetrize*⟨*proof*⟩

theorem *kleene-qfp-is-dual-extreme*:
assumes *comp*: *omega-chain−complete* A (⊑)
and *cont*: *omega-chain−continuous* A (⊑) A (⊑) f **and** *bA*: $b \in A$ **and** *bot*: $\forall x \in A.\ b \sqsubseteq x$
shows *extreme-bound* A (⊑) {f^n b |. n :: *nat*} = *extreme* $\{s \in A.\ f\ s \sim s\}$ (⊒)
⟨*proof*⟩

end

corollary(**in** *antisymmetric*) *kleene-fp*:
  assumes *cont*: *omega-chain−continuous* A (⊑) A (⊑) f
    and *b*: $b \in A$ $\forall x \in A$. $b \sqsubseteq x$
    and *p*: *extreme-bound* A (⊑) {f^n b |. n :: *nat*} p
  shows $f\ p = p$
  ⟨*proof*⟩

no-notation *compower* (‹-^-›[*1000*,*1000*]*1000*)

end

References

[1] S. Abramsky and A. Jung. *Domain Theory.* Number III in Handbook of Logic in Computer Science. Oxford University Press, 1994.

[2] C. Ballarin. Interpretation of locales in Isabelle: Theories and proof contexts. In J. M. Borwein and W. M. Farmer, editors, *Proceedings of the 5th International Conference on Mathematical Knowledge Management (MKM 2006)*, volume 4108 of *LNCS*, pages 31–43. Springer Berlin Heidelberg, 2006.

[3] G. M. Bergman. *An Invitation to General Algebra and Universal Constructions.* Springer International Publishing, Cham, 2015.

[4] S. P. Bhatta. Weak chain-completeness and fixed point property for pseudo-ordered sets. *Czechoslovak Mathematical Journal*, 55(2):365–369, 2005.

[5] S. P. Bhatta and S. George. Some fixed point theorems for pseudo ordered sets. *Algebra and Discrete Mathematics*, 11(1):17–22, 2011.

[6] N. Bourbaki. Sur le théorème de zorn. *Archiv der Mathematik*, 2(6):434–437, 1949.

[7] P. Cousot and R. Cousot. Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. In *Proceedings of the 4th ACM Symposium on Principles of Programming Languages (POPL'77)*, pages 238–252. ACM Press, 1977.

[8] J. Dubut and A. Yamada. Fixed Points Theorems for Non-Transitive Relations. *Logical Methods in Computer Science*, 18(1), 2021.

[9] G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. W. Mislove, and D. S. Scott. *Continuous Lattices and Domains.* Cambridge University Press, 2003.

[10] G. Gonthier. Formal proof – the four-color theorem. *Notices of the AMS*, 55(11):1382–1393, 2008.

[11] H. Grall. Proving fixed points. In *Fixed Points in Computer Science 2010*, pages 41–46, 2010.

[12] T. Hales, M. Adams, G. Bauer, T. D. Dang, J. Harrison, H. Le Truong, C. Kaliszyk, V. Magron, S. McLaughlin, T. T. Nguyen, et al. A formal proof of the Kepler conjecture. *Forum of Mathematics, Pi*, 5:e2, 2017.

[13] F. Kammüller. Modular reasoning in Isabelle. In D. McAllester, editor, *Proceedings of the 17th International Conference on Automated Deduction (CADE-17)*, volume 1831 of *LNCS*, pages 99–114. Springer Berlin Heidelberg, 2000.

[14] L. Kantorovitch. The method of successive approximations for functional equations. *Acta Math.*, 71:63–97, 1939.

[15] G. Klein, K. Elphinstone, G. Heiser, J. Andronick, D. Cock, P. Derrin, D. Elkaduwe, K. Engelhardt, R. Kolanski, M. Norrish, T. Sewell, H. Tuch, and S. Wiwood. seL4: Formal verification of an OS kernel. In *Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles (SOSP 2009)*, pages 207–220. ACM, 2009.

[16] G. Markowsky. Chain-complete posets and directed sets with applications. *Algebra Universalis*, 6:53–68, 1976.

[17] J. D. Mashburn. The least fixed point property for omega-chain continuous functions. *Houston Journal of Mathematics*, 9(2):231–244, 1983.

[18] D. Pataraia. A constructive proof of Tarski's fixed-point theorem for dcpo's. Presented in the 65th Peripatetic Seminar on Sheaves and Logic, in Aarhus, Denmark, 1997.

[19] G. Schmidt and T. Ströhlein. *Relations and Graphs: Discrete Mathematics for Computer Scientists.* Springer Berlin Heidelberg, Berlin, Heidelberg, 1993.

[20] D. Scott and C. Strachey. Toward a mathematical semantics for computer languages. Technical Monograph PRG-6, Oxford Programming Research Group, 1971.

[21] H. Skala. Trellis theory. *Algebra Univ.*, 1:218–233, 1971.

[22] A. Stouti and A. Maaden. Fixed points and common fixed points theorems in pseudo-ordered sets. *Proyecciones*, 32(4):409–418, 2013.

[23] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5(2):285–309, 1955.

[24] A. Yamada and J. Dubut. Complete Non-Orders and Fixed Points. In *Proceedings of the 10th International Conference on Interactive Theorem Proving (ITP 2019)*, volume 141 of *Leibniz International Proceedings in Informatics*, pages 30:1–30:16. Leibniz-Zentrum für Informatik, 2019.

[25] A. Yamada and J. Dubut. Formalizing Results on Directed Sets in Isabelle/HOL. In *Proceedings of the fourteenth conference on Interactive Theorem Proving (ITP'23)*, 2023.