

# Formalizing Results on Directed Sets

Akihisa Yamada and Jérémy Dubut

June 6, 2026

## Abstract

Directed sets are of fundamental interest in domain theory and topology. In this paper, we formalize some results on directed sets in Isabelle/HOL, most notably: under the axiom of choice, a poset has a supremum for every directed set if and only if it does so for every chain; and a function between such posets preserves suprema of directed sets if and only if it preserves suprema of chains. The known pen-and-paper proofs of these results crucially use uncountable transfinite sequences, which are not directly implementable in Isabelle/HOL. We show how to emulate such proofs by utilizing Isabelle/HOL's ordinal and cardinal library. Thanks to the formalization, we relax some conditions for the above results.

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Preliminaries</b>	<b>3</b>
2.1	Connecting Predicate-Based and Set-Based Relations . . . . .	3
2.2	Missing Lemmas . . . . .	5
<b>3</b>	<b>Iwamura's lemma</b>	<b>7</b>
3.1	Uncountable Case . . . . .	8
3.2	Countable Case . . . . .	11
<b>4</b>	<b>Directed Completeness and Scott-Continuity</b>	<b>14</b>

## 1 Introduction

A *directed set* is a set  $D$  equipped with a binary relation  $\sqsubseteq$  such that any finite subset  $X \subseteq D$  has an upper bound in  $D$  with respect to  $\sqsubseteq$ . The property is often equivalently stated that  $D$  is non-empty and any two elements  $x, y \in D$  have a bound in  $D$ , assuming that  $\sqsubseteq$  is transitive (as in posets).

Directed sets find uses in various fields of mathematics and computer science. In topology (see for example the textbook [7]), directed sets are used

to generalize the set of natural numbers: sequences  $\mathbb{N} \rightarrow A$  are generalized to *nets*  $D \rightarrow A$ , where  $D$  is an arbitrary directed set. For example, the usual result on metric spaces that continuous functions are precisely functions that preserve limits of sequences can be generalized in general topological spaces as: the continuous functions are precisely functions that preserve limits of nets. In domain theory [1], key ingredients are *directed-complete posets*, where every directed subset has a supremum in the poset, and *Scott-continuous functions* between posets, that is, functions that preserve suprema of directed sets. Thanks to their fixed-point properties (which we have formalized in Isabelle/HOL in a previous work [5]), directed-complete posets naturally appear in denotational semantics of languages with loops or fixed-point operators (see for example Scott domains [11, 13]). Directed sets also appear in reachability and coverability analyses of transition systems through the notion of ideals, that is, downward-closed directed sets. They allow effective representations of objects, making forward and backward analysis of well-structured transition systems – such as Petri nets – possible (see e.g., [6]).

Apparently milder generalizations of natural numbers are chains (totally ordered sets) or even well-ordered sets. In the mathematics literature, the following results are known (assuming the axiom of choice):

**Theorem 1** ([4]) *A poset is directed-complete if (and only if) it has a supremum for every non-empty well-ordered subset.*

**Theorem 2** ([9]) *Let  $f$  be a function between posets, each of which has a supremum for every non-empty chain. If  $f$  preserves suprema of non-empty chains, then it is Scott-continuous.*

The pen-and-paper proofs of these results use induction on cardinality, where the finite case is merely the base case. The core of the proof is a technical result called Iwamura’s Lemma [8], where the countable case is merely an easy case, and the main part heavily uses transfinite sequences indexed by uncountable ordinals.

To formalize these results in Isabelle/HOL we extensively use the existing library for ordinals and cardinals [3], but we needed some delicate work in emulating the pen-and-paper proofs. In Isabelle/HOL, or any proof assistant based on higher-order logic (HOL), it is not possible to have a datatype for arbitrarily large ordinals; hence, it is not possible to directly formalize transfinite sequences. We show how to emulate transfinite sequences using the ordinal and cardinal library [3]. As far as the authors know, our work is the first to mechanize the proof of Theorems 1 and 2, as well as Iwamura’s Lemma. We prove the two theorems for quasi-ordered sets, relaxing antisymmetry, and strengthen Theorem 2 so that chains are replaced by well-ordered sets and conditions on the codomain are completely dropped.

**Related Work** Systems based on Zermelo-Fraenkel set theory, such as Mizar [2] and Isabelle/ZF [10], have more direct support for ordinals and cardinals and should pose less challenge in mechanizing the above results. Nevertheless, a part of our contribution is in demonstrating that the power of (Isabelle/)HOL is strong enough to deal with uncountable transfinite sequences.

Except for the extra care for transfinite sequences, our proof of Iwamura’s Lemma is largely based on the original proof from [8]. Markowsky presented a proof of Theorem 1 using Iwamura’s Lemma [9, Corollary 1]. While he took a minimal-counterexample approach, we take a more constructive approach to build a well-ordered set of suprema. This construction was crucial to be reused in the proof of Theorem 2, which Markowsky claimed without a proof [9]. Another proof of Theorem 1 can be found in [4], without using Iwamura’s Lemma, but still crucially using transfinite sequences.

This work has been published in the conference paper [14].

## 2 Preliminaries

### 2.1 Connecting Predicate-Based and Set-Based Relations

**theory** *Well-Order-Connection*

**imports**

*Main*

*Complete-Non-Orders.Well-Relations*

**begin**

**lemma** *refl-on-relation-of*: *refl-on A (relation-of r A)  $\longleftrightarrow$  reflexive A r*

**by** (*auto simp: refl-on-def reflexive-def relation-of-def*)

**lemma** *trans-relation-of*: *trans (relation-of r A)  $\longleftrightarrow$  transitive A r*

**by** (*auto simp: trans-def relation-of-def transitive-def*)

**lemma** *preorder-on-relation-of*: *preorder-on A (relation-of r A)  $\longleftrightarrow$  quasi-ordered-set A r*

**proof** (*rule iffI*)

**assume** *preorder-on A (relation-of r A)*

**thus** *quasi-ordered-set A r*

**by** (*simp add: preorder-on-def refl-on-relation-of trans-relation-of quasi-ordered-set-def del: Order-Relation.trans-relation-of*)

**next**

**assume** *quasi-ordered-set A r*

**thus** *preorder-on A (relation-of r A)*

**by** (*metis (no-types, lifting) mem-Collect-eq*

*order-on-defs(1) prod.simps(2) quasi-ordered-set-def*

*refl-on-relation-of[of A r] relation-of-def subrelI*

*trans-relation-of[of r A]*)

**qed**

**lemma** *antisym-relation-of*:  $\text{antisym (relation-of } r \ A) \longleftrightarrow \text{antisymmetric } A \ r$   
**by** (*auto simp: antisym-def relation-of-def antisymmetric-def*)

**lemma** *partial-order-on-relation-of*:  
 $\text{partial-order-on } A \ (\text{relation-of } r \ A) \longleftrightarrow \text{partially-ordered-set } A \ r$   
**by** (*auto simp add: partial-order-on-def preorder-on-relation-of antisym-relation-of quasi-ordered-set-def partially-ordered-set-def simp del: Order-Relation.antisym-relation-of*)

**lemma** *total-on-relation-of*:  $\text{total-on } A \ (\text{relation-of } r \ A) \longleftrightarrow \text{semiconnex } A \ r$   
**by** (*auto simp: total-on-def relation-of-def semiconnex-def*)

**lemma** *linear-order-on-relation-of*:  
**shows**  $\text{linear-order-on } A \ (\text{relation-of } r \ A) \longleftrightarrow \text{total-ordered-set } A \ r$   
**by** (*auto simp: linear-order-on-def partial-order-on-relation-of total-on-relation-of total-ordered-set-def total-quasi-ordered-set-def partially-ordered-set-def connex-iff-semiconnex-reflexive*)

**lemma** *relation-of-sub-Id*:  $(\text{relation-of } r \ A - \text{Id}) = \text{relation-of } (\lambda x \ y. r \ x \ y \wedge x \neq y) \ A$   
**by** (*auto simp: relation-of-def*)

**lemma** (*in antisymmetric*) *asymptp-iff-weak-neq*:  
**shows**  $x \in A \implies y \in A \implies \text{asymptp } (\sqsubseteq) \ x \ y \longleftrightarrow x \sqsubseteq y \wedge x \neq y$   
**by** (*auto intro!: asymptpI antisym*)

**lemma** *wf-relation-of*:  $\text{wf (relation-of } r \ A) = \text{well-founded } A \ r$   
**apply** (*simp add: wf-eq-minimal relation-of-def well-founded-iff-ex-extremal Ball-def*)  
**by** (*metis (no-types, opaque-lifting) equals0I insert-Diff insert-not-empty subsetI subset-iff*)

**lemma** *well-order-on-relation-of*:  
**shows**  $\text{well-order-on } A \ (\text{relation-of } r \ A) \longleftrightarrow \text{well-ordered-set } A \ r$   
**by** (*auto simp: well-order-on-def linear-order-on-relation-of relation-of-sub-Id wf-relation-of well-ordered-iff-well-founded-total-ordered antisymmetric.asymptp-iff-weak-neq total-ordered-set-def cong: well-founded-cong*)

**lemma** (*in connex*) *Field-relation-of*:  $\text{Field (relation-of } (\sqsubseteq) \ A) = A$   
**by** (*auto simp: Field-def relation-of-def*)

**lemma** (*in well-ordered-set*) *Well-order-relation-of*:  
**shows**  $\text{Well-order (relation-of } (\sqsubseteq) \ A)$   
**by** (*auto simp: Field-relation-of well-order-on-relation-of well-ordered-set-axioms*)

**lemma** *in-relation-of*:  $(x, y) \in \text{relation-of } r \ A \longleftrightarrow x \in A \wedge y \in A \wedge r \ x \ y$   
**by** (*simp add: relation-of-def*)

**lemma** *relation-of-triv*: *relation-of*  $(\lambda x y. (x,y) \in r)$  *UNIV* = *r*  
**by** (*auto simp: relation-of-def*)

**lemma** *Restr-eq-relation-of*: *Restr* *R* *A* = *relation-of*  $(\lambda x y. (x,y) \in R)$  *A*  
**by** (*auto simp: relation-of-def*)

**theorem** *ex-well-order*:  $\exists r. \text{well-ordered-set } A \ r$

**proof** –

**from** *well-order-on* **obtain** *R* **where** *R*: *well-order-on* *A* *R* **by** *auto*

**then have** *well-order-on* *A* (*Restr* *R* *A*)

**by** (*simp add: well-order-on-Field[OF R] Restr-Field*)

**then show** *?thesis* **by** (*auto simp: Restr-eq-relation-of well-order-on-relation-of*)

**qed**

**end**

**theory** *Directed-Completeness*

**imports**

*Complete-Non-Orders.Continuity*

*Well-Order-Connection*

*HOL-Cardinals.Cardinals*

*HOL-Library.FuncSet*

**begin**

## 2.2 Missing Lemmas

**no-notation** *disj* (**infixr**  $\langle | \rangle$  30)

**lemma** *Sup-funpow-mono*:

**fixes** *f* :: 'a :: *complete-lattice*  $\Rightarrow$  'a

**assumes** *mono*: *mono* *f*

**shows** *mono*  $(\bigsqcup i. f \hat{\sim} i)$

**by** (*intro monoI, auto intro!: Sup-mono dest: funpow-mono[OF mono]*)

**lemma** *iso-imp-compat*:

**assumes** *iso*: *iso* *r* *r'* *f* **shows** *compat* *r* *r'* *f*

**by** (*simp add: compat-def iso iso-forward*)

**lemma** *iso-inv-into*:

**assumes** *ISO*: *iso* *r* *r'* *f*

**shows** *iso* *r'* *r* (*inv-into* (*Field* *r*) *f*)

**using** *assms* **unfolding** *iso-def*

**using** *bij-betw-inv-into inv-into-Field-embed-bij-betw* **by** *blast*

**lemmas** *iso-imp-compat-inv-into* = *iso-imp-compat*[*OF iso-inv-into*]

**lemma** *infinite-iff-natLeq*: *infinite* *A*  $\longleftrightarrow$  *natLeq*  $\leq_o$   $|A|$

**using** *infinite-iff-natLeq-ordLeq* **by** *blast*

As we cannot formalize transfinite sequences directly, we take the fol-

lowing approach: We just use  $A$  as the index set, and instead of the ordering on ordinals, we take the well-order that is chosen by the cardinality library to denote  $|A|$ .

**definition** *well-order-of* ( $\langle\langle'(\preceq\cdot)'\rangle\rangle$ ) [0]1000) **where**  $(\preceq_A) x y \equiv (x, y) \in |A|$

**abbreviation** *well-order-le* ( $\langle\langle\preceq\cdot\rangle\rangle$ ) [51,0,51]50) **where**  $x \preceq_A y \equiv (\preceq_A) x y$

**abbreviation** *well-order-less* ( $\langle\langle\prec\cdot\rangle\rangle$ ) [51,0,51]50) **where**  $x \prec_A y \equiv \text{asymptpt}(\preceq_A) x y$

**lemmas** *well-order-ofI* = *well-order-of-def*[*unfolded atomize-eq*, *THEN iffD2*]

**lemmas** *well-order-ofD* = *well-order-of-def*[*unfolded atomize-eq*, *THEN iffD1*]

**lemma** *carrier*: **assumes**  $x \preceq_A y$  **shows**  $x \in A$  **and**  $y \in A$   
**using** *assms* **by** (*auto dest!*: *well-order-ofD dest!*: *FieldI1 FieldI2*)

**lemma** *relation-of[simp]*: *relation-of*  $(\preceq_A) A = |A|$   
**by** (*auto simp!*: *relation-of-def well-order-of-def dest!*: *FieldI1 FieldI2*)

**interpretation** *well-order-of*: *well-ordered-set*  $A (\preceq_A)$   
**apply** (*fold well-order-on-relation-of*)  
**by** *auto*

Thanks to the well-order theorem, one can have a sequence  $\{A_\alpha\}_{\alpha < |A|}$  of subsets of  $A$  that satisfies the following three conditions:

- cardinality:  $|A_\alpha| < |A|$  for every  $\alpha < |A|$ ,
- monotonicity:  $A_\alpha \subseteq A_\beta$  whenever  $\alpha \leq \beta < |A|$ , and
- range: if  $A$  is infinite,  $A = \bigcup_{\alpha < |A|} A_\alpha$ .

The following serves the purpose.

**definition** *Pre* ( $\langle\langle\prec\cdot\rangle\rangle$ ) [1000]1000) **where**  $A_\prec a \equiv \{b \in A. b \prec_A a\}$

**lemma** *Pre-eq-underS*:  $A_\prec a = \text{underS } |A| a$   
**by** (*auto simp!*: *Pre-def underS-def well-order-ofD carrier well-order-of.antisym dest!*: *well-order-ofI*)

**lemma** *Pre-card*: **assumes**  $a \in A$  **shows**  $|A_\prec a| < |A|$   
**by** (*auto simp!*: *Pre-eq-underS aA intro!*: *card-of-underS[OF card-of-Card-order]*)

**lemma** *Pre-carrier*:  $A_\prec a \subseteq A$  **by** (*auto simp!*: *Pre-def*)

**lemma** *Pre-mono*: *monotone-on*  $A (\preceq_A) (\subseteq) (A_\prec)$   
**by** (*auto intro!*: *monotone-onI simp!*: *Pre-def dest!*: *well-order-of.asym-trans well-order-of.asym.irrefl*)

**lemma** *extreme-imp-finite*:  
**assumes**  $e$ : *extreme*  $A (\preceq_A) e$  **shows** *finite*  $A$

```

proof (rule ccontr)
  assume inf: infinite A
  from e have eA:  $e \in A$  by auto
  from e have  $A = \{a \in A. a \preceq_A e\}$  by auto
  also have  $\dots - \{e\} = A_{\prec} e$ 
    using eA by (auto simp: Pre-def dest: well-order-of.asympartp-iff-weak-peq)
  finally have AeP:  $A - \{e\} = \dots$ 
  have infinite ( $A - \{e\}$ ) using infinite-remove[OF inf].
  with AeP have infP: infinite ( $A_{\prec} e$ ) by simp
  have  $A = \text{insert } e (A_{\prec} e)$  using eA by (fold AeP, auto)
  also have  $|\dots| = o |A_{\prec} e|$  using infinite-card-of-insert[OF infP].
  finally have  $|A_{\prec} e| = o |A|$  using ordIso-symmetric by auto
  with Pre-card[OF eA] not-ordLess-ordIso
  show False by auto
qed

```

```

lemma infinite-imp-ex-Pre:
  assumes inf: infinite A and xA:  $x \in A$  shows  $\exists y \in A. x \in A_{\prec} y$ 
proof -
  from inf
  have  $\neg \text{extreme } A (\preceq_A) x$  by (auto dest!: extreme-imp-finite)
  with xA obtain y where yA:  $y \in A$  and  $\neg y \preceq_A x$  by auto
  with xA have  $x \prec_A y$  by (auto simp: well-order-of.not-weak-iff asympartpI)
  with yA show ?thesis by (auto simp: Pre-def xA)
qed

```

```

lemma infinite-imp-Un-Pre: assumes inf: infinite A shows  $\bigcup (A_{\prec} \text{ ` } A) = A$ 
proof (safe)
  fix x assume xA:  $x \in A$ 
  show  $y \in A_{\prec} x \implies y \in A$  for y using Pre-carrier[of A x] by auto
  from infinite-imp-ex-Pre[OF inf xA]
  show  $x \in \bigcup (A_{\prec} \text{ ` } A)$  by (auto simp: Pre-def)
qed

```

### 3 Iwamura's lemma

As the proof involves a number of (inductive) definitions, we build a locale for collecting those definitions and lemmas.

```

locale Iwamura-proof = related-set +
  assumes dir: directed-set A ( $\sqsubseteq$ )
begin

```

Inside this locale, a related set  $(A, \sqsubseteq)$  is fixed and assumed to be directed. The proof starts with declaring, using the axiom of choice, a function  $f$  that chooses a bound  $f X \in A$  for every finite subset  $X \subseteq A$ . This function can be formalized using the SOME construction:

```

definition f where  $f X \equiv \text{SOME } z. z \in A \wedge \text{bound } X (\sqsubseteq) z$ 

```

**lemma** *assumes*  $XA: X \subseteq A$  **and**  $Xfin: finite\ X$   
*shows*  $f\text{-carrier}: f\ X \in A$  **and**  $f\text{-bound}: bound\ X\ (\sqsubseteq)\ (f\ X)$   
*using*  $directed\text{-set}D[OF\ dir\ XA\ Xfin, unfolded\ Bex\text{-}def, THEN\ someI\text{-}ex]$   
*by*  $(auto\ simp: f\text{-}def)$

### 3.1 Uncountable Case

Actually, the main part of the proof of Iwamura's Lemma is about monotonically expanding an infinite subset (in particular  $A_\alpha$ ) of  $A$  into a directed one, without changing the cardinality. To this end, Iwamura's original proof introduces a function  $F: PowA \rightarrow PowA$  that expands a set with upper bounds of *all finite subsets*. This approach is different from Markowsky's reproof (based on [12]) which uses nested transfinite induction to extend a set one element after another.

**definition**  $F$  **where**  $F\ X \equiv X \cup f\ 'Fpow\ X$

**lemma**  $F\text{-carrier}: X \subseteq A \implies F\ X \subseteq A$   
**and**  $F\text{-infl}: X \subseteq F\ X$   
**and**  $F\text{-fin}: finite\ X \implies finite\ (F\ X)$   
*by*  $(auto\ simp: F\text{-}def\ Fpow\text{-}def\ f\text{-}carrier)$

**lemma**  $F\text{-card}: assumes\ inf: infinite\ X\ shows\ |F\ X| =_o\ |X|$

**proof** –

**have**  $|f\ 'Fpow\ X| \leq_o\ |Fpow\ X|$  **using**  $card\text{-of}\text{-}image$ .

**thm**  $card\text{-of}\text{-}Fpow\text{-infinite}$

**also have**  $|Fpow\ X| =_o\ |X|$  **using**  $card\text{-of}\text{-}Fpow\text{-infinite}[OF\ inf]$ .

**finally have**  $|f\ 'Fpow\ X| \leq_o\ |X|$ .

**with**  $inf$  **show**  $?thesis$  **by**  $(auto\ simp: F\text{-}def)$

**qed**

**lemma**  $F\text{-mono}: mono\ F$

**proof** $(intro\ monoI)$

**show**  $X \subseteq Y \implies F\ X \subseteq F\ Y$  **for**  $X\ Y$

**using**  $Fpow\text{-}mono[of\ X\ Y]$  **by**  $(auto\ simp: F\text{-}def)$

**qed**

**lemma**  $Fn\text{-carrier}: X \subseteq A \implies (F\ \overset{\sim}{\sim} n)\ X \subseteq A$

**and**  $Fn\text{-infl}: X \subseteq (F\ \overset{\sim}{\sim} n)\ X$

**and**  $Fn\text{-fin}: finite\ X \implies finite\ ((F\ \overset{\sim}{\sim} n)\ X)$

**and**  $Fn\text{-card}: infinite\ X \implies |(F\ \overset{\sim}{\sim} n)\ X| =_o\ |X|$

**proof**  $(atomize(full), induct\ n)$

**case**  $(Suc\ n)$

**define**  $Y$  **where**  $Y \equiv (F\ \overset{\sim}{\sim} n)\ X$

**then have**  $*$ :  $(F\ \overset{\sim}{\sim} Suc\ n)\ X = F\ Y$  **by**  $auto$

**from**  $Suc[folded\ Y\text{-}def]$

**have**  $infinite\ X \implies infinite\ Y \wedge |Y| =_o\ |X|$

**and**  $finite\ X \implies finite\ Y$

**and**  $X \subseteq Y$

**and**  $X \subseteq A \implies Y \subseteq A$  **by** (*auto simp: Y-def*)  
**with**  $F\text{-carrier}[of\ Y]$   $F\text{-infl}[of\ Y]$   $F\text{-card}[of\ Y]$   $F\text{-fin}[of\ Y]$   
**show**  $?case$  **by** (*unfold \*, auto del:subsetI dest:ordIso-transitive*)  
**qed** *auto*

**lemma**  $Fn\text{-mono1}$ :  $i \leq j \implies (F \overset{\sim}{\sim} i) X \subseteq (F \overset{\sim}{\sim} j) X$  **for**  $i\ j$   
**using**  $Fn\text{-infl}[of\ (F \overset{\sim}{\sim} i) X\ j-i]$   $funpow\text{-add}[of\ j-i\ i\ F]$   
**by** *auto*

We take the  $\omega$ -iteration of the monotone function  $F$ , namely:

**definition**  $Flim\ (\langle F^\omega \rangle)$  **where**  $F^\omega X \equiv \bigcup i. (F \overset{\sim}{\sim} i) X$

**lemma**  $Flim\text{-mono}$ : *mono*  $F^\omega$   
**proof** –  
**have**  $F^\omega = (\bigsqcup\ range\ ((\overset{\sim}{\sim})\ F))$  **by** (*auto simp: Flim-def*)  
**with**  $Sup\text{-funpow}\text{-mono}[OF\ F\text{-mono}]$   
**show**  $?thesis$  **by** *auto*  
**qed**

**lemma**  $Flim\text{-infl}$ :  $X \subseteq F^\omega X$   
**using**  $Fn\text{-infl}$  **by** (*auto simp: Flim-def*)

**lemma**  $Flim\text{-carrier}$ : **assumes**  $X \subseteq A$  **shows**  $F^\omega X \subseteq A$   
**using**  $Fn\text{-carrier}[OF\ assms]$  **by** (*auto simp: Flim-def*)

**lemma**  $Flim\text{-directed}$ : **assumes**  $X \subseteq A$  **shows** *directed-set*  $(F^\omega X)$   $(\sqsubseteq)$   
**proof** (*safe intro!: directed-setI*)  
**fix**  $Y$  **assume**  $YC$ :  $Y \subseteq F^\omega X$  **and**  $finY$ : *finite*  $Y$   
**from**  $finY\ YC$  **have**  $\exists i. Y \subseteq (F \overset{\sim}{\sim} i) X$   
**proof** (*induct*)  
**case** *empty*  
**then show**  $?case$  **by** *auto*  
**next**  
**case** (*insert y Y*)  
**then obtain**  $i\ j$  **where**  $Yi$ :  $Y \subseteq (F \overset{\sim}{\sim} i) X$  **and**  $y \in (F \overset{\sim}{\sim} j) X$  **by** (*auto simp: Flim-def*)  
**with**  $Fn\text{-mono1}[OF\ max.cobounded1[of\ i\ j],\ of\ X]$   $Fn\text{-mono1}[OF\ max.cobounded2[of\ j\ i],\ of\ X]$   
**show**  $?case$  **by** (*auto intro!: exI[of - max i j]*)  
**qed**  
**then obtain**  $i$  **where**  $Yi$ :  $Y \subseteq (F \overset{\sim}{\sim} i) X$  **by** *auto*  
**with**  $Fn\text{-carrier}[OF\ assms]$  **have**  $YA$ :  $Y \subseteq A$  **by** *auto*  
**from**  $Yi\ finY$  **have**  $f\ Y \in (F \overset{\sim}{\sim} Suc\ i) X$  **by** (*auto simp: F-def Fpow-def*)  
**then have**  $f\ Y \in F^\omega X$  **by** (*auto simp: Flim-def simp del: funpow.simps*)  
**with**  $f\text{-bound}[OF\ YA\ finY]$   
**show**  $\exists z \in F^\omega X. bound\ Y\ (\sqsubseteq)\ z$  **by** *auto*  
**qed**

**lemma**  $Flim\text{-card}$ : **assumes** *infinite*  $X$  **shows**  $|F^\omega X| = o\ |X|$

**proof** –  
**from** *assms* **have**  $\text{nat}X: |UNIV :: \text{nat set}| \leq_o |X|$  **by** (*simp add: infinite-iff-card-of-nat*)  
**have**  $|F^\omega X| \leq_o |X|$   
**apply** (*unfold Flim-def, rule card-of-UNION-ordLeq-infinite[OF assms natX]*)  
**using** *Fn-card[OF assms] ordIso-imp-ordLeq*  
**by** *auto*  
**with** *Flim-infl* **show**  $|F^\omega X| =_o |X|$  **by** (*simp add: ordIso-iff-ordLeq*)  
**qed**

**lemma** *Flim-fin*: **assumes** *finite X* **shows**  $|F^\omega X| \leq_o \text{natLeq}$   
**proof** –  
**have**  $|F^\omega X| \leq_o |UNIV :: \text{nat set}|$   
**apply** (*unfold Flim-def*)  
**apply** (*rule card-of-UNION-ordLeq-infinite*)  
**by** (*auto simp: Fn-fin[OF assms] intro!: ordLess-imp-ordLeq*)  
**then show** *?thesis* **using** *card-of-nat ordLeq-ordIso-trans* **by** *auto*  
**qed**

**lemma** *mono-uncountable*: *monotone-on A* ( $\preceq_A$ ) ( $\subseteq$ ) ( $F^\omega \circ A_{\prec}$ )  
**using** *monotone-on-o[OF Flim-mono Pre-mono]*  
**by** (*auto simp: o-def*)

**lemma** *card-uncountable*:  
**assumes**  $aA: a \in A$  **and**  $unc: \text{natLeq} <_o |A|$   
**shows**  $|F^\omega (A_{\prec} a)| <_o |A|$   
**proof** (*cases finite (A\_{\prec} a)*)  
**case** *True*  
**note** *Flim-fin[OF this]*  
**also note** *unc*  
**finally show** *?thesis*  
**using** *unc not-ordLess-ordIso* **by** *auto*  
**next**  
**case** *False*  
**note** *Flim-card[OF this]*  
**also note** *Pre-card[OF aA]*  
**finally show** *?thesis* **using** *unc not-ordLess-ordIso* **by** *auto*  
**qed**

**lemma** *in-I-uncountable*:  
**assumes**  $aA: a \in A$  **and**  $inf: \text{infinite } A$   
**shows**  $\exists a' \in A. a \in F^\omega (A_{\prec} a')$   
**using** *infinite-imp-ex-Pre[OF inf aA] Flim-infl*  
**by** *auto*

**lemma** *carrier-uncountable*:  
**shows**  $F^\omega (A_{\prec} a) \subseteq A$   
**using** *Flim-carrier[OF Pre-carrier]*  
**by** *auto*

**lemma** *range-uncountable*: **assumes** *inf*: *infinite A* **shows**  $\bigcup ((F^\omega \circ A_\prec) \text{ ` } A) = A$

**proof** (*safe intro!*: *subset-antisym*)

**fix** *a* **assume** *aA*:  $a \in A$

**from** *infinite-imp-ex-Pre*[*OF inf aA*] *Flim-infl*

**show**  $a \in \bigcup ((F^\omega \circ A_\prec) \text{ ` } A)$  **by** *auto*

**show**  $x \in (F^\omega \circ A_\prec) a \implies x \in A$  **for** *x*

**using** *carrier-uncountable* **by** *auto*

**qed**

**lemma** *infl-uncountable*:

**assumes** *aA*:  $a \in A$  **and** *bA*:  $b \in A$  **and** *ab*:  $a \prec_A b$

**shows**  $a \in F^\omega (A_\prec b)$

**using** *assms Flim-infl*[*of A\_\prec b*]

**by** (*auto simp: Pre-def*)

### 3.2 Countable Case

**context**

**assumes** *countable*:  $|A| =_o \text{ natLeq}$

**begin**

The assumption above means that there exists an order-isomorphism between  $(\mathbb{N}, \leq)$  and  $(A, \preceq_A)$ .

**definition** *seq* ::  $\text{nat} \Rightarrow 'a$  **where** *seq*  $\equiv \text{SOME } f. \text{ iso natLeq } |A| f$

**lemma** *seq-iso*: *iso natLeq |A| seq*

**apply** (*unfold seq-def*)

**apply** (*rule someI-ex*[*of iso natLeq |A|*])

**using** *countable*[*THEN ordIso-symmetric*]

**apply** (*unfold ordIso-def*) **by** *auto*

**lemma** *seq-bij-betw*: *bij-betw seq UNIV A*

**using** *seq-iso* **by** (*auto simp: iso-def Field-natLeq*)

This means that  $A$  has been indexed by  $\mathbb{N}$ .

**lemma** *range-seq*: *range seq = A*

**using** *seq-bij-betw bij-betw-imp-surj-on* **by** *force*

**lemma** *seq-mono*: *monotone*  $(\leq)$   $(\preceq_A)$  *seq*

**using** *iso-imp-compat*[*OF seq-iso*]

**by** (*auto intro!*: *monotoneI well-order-ofI simp: compat-def natLeq-def*)

**lemma** *inv-seq-mono*: *monotone-on*  $A$   $(\preceq_A)$   $(\leq)$  (*inv seq*)

**using** *iso-imp-compat-inv-into*[*OF seq-iso*]

**unfolding** *Field-natLeq*

**by** (*auto intro!*: *monotone-onI simp: natLeq-def compat-def well-order-of-def*)

We turn the sequence into a sequence of directed subsets of  $A$ :

**fun** *Seq* ::  $\text{nat} \Rightarrow 'a$  **set** **where**

$Seq\ 0 = \{f\ \{\}\}$   
 $| Seq\ (Suc\ n) = Seq\ n \cup \{seq\ n, f\ (Seq\ n \cup \{seq\ n\})\}$

**lemma** *seq-n-in-Seq-n*:  $seq\ n \in Seq\ (Suc\ n)$  **by** *auto*

**lemma** *Seq-finite*:  $finite\ (Seq\ n)$   
**by** (*induction n*) *auto*

**lemma** *Seq-card*:  $|Seq\ n| <_o\ |A|$   
**using** *countable Seq-finite* **by** (*simp add: ordIso-natLeq-infinite1*)

**lemma** *Seq-carrier*:  $Seq\ n \subseteq A$   
**proof**(*induction n*)  
  **case**  $0$   
  **show** *?case* **by** (*auto intro!: f-carrier*)  
**next**  
  **case** (*Suc n*)  
  **with** *range-seq* **have** *sgA*:  $Seq\ n \cup \{seq\ n\} \subseteq A$  **by** *auto*  
  **from** *Seq-finite f-carrier[OF sgA]*  
  **have**  $f\ (Seq\ n \cup \{seq\ n\}) \in A$  **by** *auto*  
  **with** *sgA* **show** *?case* **by** *auto*  
**qed**

**lemma** *Seq-range*:  $\bigcup (range\ Seq) = A$   
**proof** (*intro equalityI*)  
  **from** *Seq-carrier* **show**  $\bigcup (range\ Seq) \subseteq A$  **by** *auto*  
  **show**  $A \subseteq \bigcup (range\ Seq)$   
  **proof**  
  **fix** *a* **assume** *aA*:  $a \in A$   
  **with** *seq-bij-betw* **obtain** *n* **where**  $a = seq\ n$   
  **by** (*metis bij-betw-inv-into-right*)  
  **with** *seq-n-in-Seq-n* **show**  $a \in \bigcup (range\ Seq)$  **by** (*auto intro!: exI[of - Suc n]*)  
  **qed**  
**qed**

**lemma** *Seq-extremed*:  
  **assumes** *refl*:  $reflexive\ A\ (\sqsubseteq)$  **shows**  $extremed\ (Seq\ n)\ (\sqsubseteq)$   
**proof** –  
  **interpret** *reflexive* **using** *refl*.  
  **show** *?thesis*  
  **proof**(*induction n*)  
  **case**  $0$   
  **show** *?case* **by** (*auto intro!: extremedI extremeI f-carrier*)  
**next**  
  **case** (*Suc n*)  
  **show** *?case*  
  **proof** (*intro extremedI extremeI*)  
  **show**  $f\ (Seq\ n \cup \{seq\ n\}) \in Seq\ (Suc\ n)$  **by** *auto*  
  **fix** *x* **assume** *xssn*:  $x \in Seq\ (Suc\ n)$

```

show  $x \sqsubseteq f (Seq\ n \cup \{seq\ n\})$ 
proof(cases  $x \in Seq\ n \cup \{seq\ n\}$ )
  case True
    with  $f\text{-bound}[of\ Seq\ n \cup \{seq\ n\}]$   $range\text{-}seq\ Seq\text{-}finite[of\ n]$ 
       $Seq\text{-}carrier[of\ n]$ 
    show ?thesis by (auto simp: bound-def)
  next
    case False
    with xssn have  $x = f (Seq\ n \cup \{seq\ n\})$  by auto
    from  $range\text{-}seq\ Seq\text{-}finite[of\ n]$   $Seq\text{-}carrier[of\ n]$ 
    show ?thesis by (auto simp: x intro!: f-carrier)
  qed
qed
qed
qed

```

**lemma** *Seq-directed*: **assumes** *refl*: *reflexive*  $A (\sqsubseteq)$  **shows** *directed-set*  $(Seq\ n) (\sqsubseteq)$   
**using** *Seq-extremed*[*OF refl*] **by** (*simp add: directed-set-iff-extremed*[*OF Seq-finite*])

**lemma** *range-countable*:  $\bigcup ((Seq \circ inv\ seq) \text{ ` } A) = A$   
**apply** (*fold image-comp*)  
**apply** (*unfold bij-betw-imp-surj-on*[*OF bij-betw-inv-into*[*OF seq-bij-betw*]])  
**using** *Seq-range*.

**lemma** *Seq-mono*: *mono*  $Seq$   
**proof** (*intro monoI*)  
**show**  $n \leq m \implies Seq\ n \subseteq Seq\ m$  **for**  $n\ m$  **by** (*induct rule:inc-induct, auto*)  
**qed**

**lemma** *mono-countable*: *monotone-on*  $A (\preceq_A) (\subseteq) (Seq \circ inv\ seq)$   
**by** (*rule monotone-on-o*[*OF Seq-mono inv-seq-mono*]) *auto*

**lemma** *infl-countable*:  
**assumes**  $aA$ :  $a \in A$  **and**  $bA$ :  $b \in A$  **and**  $ab$ :  $a \prec_A b$   
**shows**  $a \in Seq (inv\ seq\ b)$   
**proof**–  
**from**  $aA$  *seq-bij-betw seq-n-in-Seq-n*  
**have**  $a : a \in Seq (Suc (inv\ seq\ a))$  **by** (*simp add: bij-betw-inv-into-right*)  
**from**  $ab$  **have**  $inv\ seq\ a < inv\ seq\ b$   
**by** (*metis (mono-tags, lifting) aA well-order-of.asympartp-iff-weak-neq bA range-seq inv-seq-mono inv-into-injective not-le-imp-less ord.mono-onD verit-la-disequality*)  
**then have**  $Suc (inv\ seq\ a) \leq inv\ seq\ b$  **by** *auto*  
**from**  $a\ monoD$ [*OF Seq-mono this*] **have**  $a \in Seq (inv\ seq\ b)$  **by** *auto*  
**then show** *?thesis* **by** *auto*  
**qed**

**end**

To match the types, we use the inverse *inv seq* of the isomorphism *isaseq*. We define the final *I* as follows:

**definition**  $I$  where  $I \equiv \text{if } |A| =_o \text{ natLeq then Seq} \circ \text{inv seq else } F^\omega \circ A \prec$

**lemma**  $I$ -carrier:  $I a \subseteq A$   
**using**  $\text{Seq-carrier carrier-uncountable}$  **by** ( $\text{auto simp: } I\text{-def}$ )

**lemma**  $I$ -directed: **assumes**  $\text{reflexive } A (\sqsubseteq)$  **shows**  $\text{directed-set } (I a) (\sqsubseteq)$   
**using**  $\text{Seq-directed}[OF - \text{assms}] \text{Flim-directed}[OF \text{Pre-carrier}]$   
**by** ( $\text{auto simp: } I\text{-def}$ )

**lemma**  $I$ -mono:  $\text{monotone-on } A (\preceq_A) (\subseteq) I$   
**by** ( $\text{auto simp: mono-uncountable mono-countable } I\text{-def}$ )

**lemma**  $I$ -card:  
**assumes**  $\text{inf: infinite } A$  **and**  $aA: a \in A$   
**shows**  $|I a| <_o |A|$   
**proof** ( $\text{cases } |A| =_o \text{ natLeq}$ )  
**case**  $\text{True}$   
**with**  $\text{Seq-finite}[OF \text{this}]$  **show**  $?thesis$  **by** ( $\text{simp add: } I\text{-def inf}$ )  
**next**  
**case**  $\text{False}$   
**with**  $\text{inf have natLeq } <_o |A|$   
**by** ( $\text{auto simp: infinite-iff-natLeq ordLeq-iff-ordLess-or-ordIso ordIso-symmetric}$ )  
**from**  $\text{card-uncountable}[OF aA \text{this}]$  **show**  $?thesis$  **by** ( $\text{auto simp: } I\text{-def } F$ )  
**qed**

**lemma**  $I$ -range: **assumes**  $\text{inf: infinite } A$  **shows**  $\bigcup (I'A) = A$   
**using**  $\text{range-uncountable}[OF \text{inf}] \text{range-countable}$  **by** ( $\text{auto simp: } I\text{-def}$ )

**lemma**  $I$ -inft: **assumes**  $a \in A$   $b \in A$   $a \prec_A b$  **shows**  $a \in I b$   
**using**  $\text{inft-countable inft-uncountable assms}$  **by** ( $\text{auto simp: } I\text{-def}$ )

**end**

Now we close the locale *Iwamura-proof* and state the final result in the global scope.

**theorem** (**in**  $\text{reflexive}$ )  $\text{Iwamura}$ :  
**assumes**  $\text{dir: directed-set } A (\sqsubseteq)$  **and**  $\text{inf: infinite } A$   
**shows**  $\exists I. (\forall a \in A. \text{directed-set } (I a) (\sqsubseteq) \wedge |I a| <_o |A|) \wedge$   
 $\text{monotone-on } A (\preceq_A) (\subseteq) I \wedge \bigcup (I'A) = A$   
**proof** –  
**interpret**  $\text{Iwamura-proof}$  **using**  $\text{dir}$  **by**  $\text{unfold-locales}$   
**show**  $?thesis$  **using**  $\text{I-mono } I\text{-card}[OF \text{inf}] \text{I-directed } I\text{-range}[OF \text{inf}]$   
**by** ( $\text{auto intro!: exI[of - } I]$ )  
**qed**

## 4 Directed Completeness and Scott-Continuity

**abbreviation**  $\text{nonempty } A \equiv \text{if } A = \{\} \text{ then } \perp \text{ else } \top$

**lemma** (in *quasi-ordered-set*) *directed-completeness-lemma*:  
**fixes** *leB* (infix  $\langle \sqsubseteq \rangle$  50)  
**assumes** *comp*: (nonempty  $\sqcap$  well-related-set)–complete *A* ( $\sqsubseteq$ ) **and** *dir*: di-  
rected-set *D* ( $\sqsubseteq$ ) **and** *DA*:  $D \subseteq A$   
**shows**  $\exists s$ . extreme-bound *A* ( $\sqsubseteq$ ) *D* *s*  
**and** well-related-set–continuous *A* ( $\sqsubseteq$ ) *B* ( $\sqsubseteq$ ) *f*  $\implies$   
 $D \neq \{\}$   $\implies$  extreme-bound *A* ( $\sqsubseteq$ ) *D* *t*  $\implies$  extreme-bound *B* ( $\sqsubseteq$ ) (*f* ‘ *D*) (*f*  
*t*)  
**proof** (atomize(full), insert wf-ordLess *dir* *DA*, induct  $|D|$  arbitrary: *D* *t* rule:  
wf-induct-rule)  
**interpret** less-eq-symmetrize.  
**case** less  
**note** *this*(1)  
**note** *IH* = *this*[*THEN* *conjunct1*]  
**and** *IH2* = *this*[*THEN* *conjunct2*, rule-format]  
**note** *DA* =  $\langle D \subseteq A \rangle$   
**interpret** *D*: quasi-ordered-set *D* ( $\sqsubseteq$ ) **using** quasi-ordered-subset[*OF* *DA*].  
**note** *dir* =  $\langle$ directed-set *D* ( $\sqsubseteq$ ) $\rangle$   
**show** ?*case*  
**proof**(*cases* finite *D*)  
**case** True  
**from** directed-set-iff-extremed[*OF* True] *dir*  
**obtain** *d* **where** *dD*:  $d \in D$  **and** *exd*: extreme *D* ( $\sqsubseteq$ ) *d* **by** (auto simp: ex-  
tremed-def)  
**then** have *dd*:  $d \sqsubseteq d$  **by** (auto simp: extreme-def)  
**show** ?*thesis*  
**proof**(intro *conjI* allI *impI* *exI*[of - *d*])  
**from** extreme-imp-extreme-bound[*OF* *exd* *DA*]  
**show** *exbd*: extreme-bound *A* ( $\sqsubseteq$ ) *D* *d* **by** auto  
**assume** *f*: well-related-set–continuous *A* ( $\sqsubseteq$ ) *B* ( $\sqsubseteq$ ) *f*  
**and** *Dt*: extreme-bound *A* ( $\sqsubseteq$ ) *D* *t* **and** *D0*:  $D \neq \{\}$   
**from** *f*[*THEN* continuous-carrier*D*] **have** *fAB*:  $f$  ‘ *A*  $\subseteq$  *B* **by** auto  
**from** *Dt* **have** *tA*:  $t \in A$  **by** auto  
**show** extreme-bound *B* ( $\sqsubseteq$ ) (*f* ‘ *D*) (*f* *t*)  
**proof** (safe intro!: extreme-boundI)  
**from** *fAB* *tA* **show**  $f$  *t*  $\in$  *B* **by** auto  
**fix** *x* **assume** *xD*:  $x \in D$   
**from** *xD* *Dt* **have** *xt*:  $x \sqsubseteq t$  **by** auto  
**have** monotone-on *A* ( $\sqsubseteq$ ) ( $\sqsubseteq$ ) *f*  
**by** (auto intro!: continuous-imp-monotone-on[*OF* *f*] pair-well-related)  
**from** monotone-on*D*[*OF* *this*] *xD* *DA* *tA* *xt*  
**show**  $f$  *x*  $\sqsubseteq$  *f* *t* **by** (auto simp: bound-empty extreme-def)  
**next**  
**fix** *b* **assume** bound (*f* ‘ *D*) ( $\sqsubseteq$ ) *b* **and** *bB*:  $b \in B$   
**with** *dD* **have** *fdb*:  $f$  *d*  $\sqsubseteq$  *b* **by** auto  
**from** *Dt* *exbd* **have** *dt*:  $d \sim t$  **by** (auto simp: extreme-bound-iff)  
**from** *dD* *DA* **have** *dA*:  $d \in A$  **by** auto  
**with** extreme-bound-sym-trans[*OF* - extreme-bound-singleton[*OF* *dA*] *dt* *tA*]  
**have** extreme-bound *A* ( $\sqsubseteq$ )  $\{d\}$  *t* **by** auto

```

    from dD DA f[THEN continuousD, OF well-related-singleton-refl - - this]
    have exfdt: extreme-bound B ( $\sqsubseteq$ ) {f d} (f t) by auto
    from fdb bB exfdt show f t  $\sqsubseteq$  b by auto
  qed
next
case inf: False
from D.Iwamura[OF dir inf]
obtain I where Imono: monotone-on D ( $\preceq_D$ ) ( $\sqsubseteq$ ) I
and Icard:  $\forall a \in D. |I a| < o |D|$ 
and Idir:  $\forall a \in D. \text{directed-set } (I a)$  ( $\sqsubseteq$ )
and Irange:  $\bigcup (I ' D) = D$ 
by auto
have  $\forall d \in D. \exists s. \text{extreme-bound } A$  ( $\sqsubseteq$ ) (I d) s
proof safe
  fix d assume dD:  $d \in D$ 
  with Irange DA have IdA:  $I d \subseteq A$  by auto
  with IH Icard Idir dD range DA
  show  $\exists s. \text{extreme-bound } A$  ( $\sqsubseteq$ ) (I d) s by auto
qed
from bchoice[OF this]
obtain s where s:  $\bigwedge d. d \in D \implies \text{extreme-bound } A$  ( $\sqsubseteq$ ) (I d) (s d) by auto
then have sDA:  $s ' D \subseteq A$  by auto
have smono: monotone-on D ( $\preceq_D$ ) ( $\sqsubseteq$ ) s
proof (intro monotone-onI)
  fix x y assume xD:  $x \in D$  and yD:  $y \in D$  and xy:  $x \preceq_D y$ 
  show  $s x \sqsubseteq s y$ 
  apply (rule extreme-bound-subset[OF monotone-onD[OF Imono xD yD xy],
of A])
  using s xD yD by auto
qed
from well-order-of.monotone-image-well-related[OF this]
have wsD: well-related-set (s ' D) ( $\sqsubseteq$ ).
from inf have sD0: nonempty (s ' D) ( $\sqsubseteq$ ) by auto
from completeD[OF comp sDA] wsD sD0
obtain x where x: extreme-bound A ( $\sqsubseteq$ ) (s ' D) x by auto
show ?thesis
proof (intro conjI allI impI exI[of - x])
  show Dx: extreme-bound A ( $\sqsubseteq$ ) D x
  proof (intro smono exI[of - x] extreme-boundI)
    from x show xA:  $x \in A$  by auto
    fix d assume dD:  $d \in D$ 
    with Irange obtain d' where d'D:  $d' \in D$  and  $d \in I d'$  by auto
    with s have 1:  $d \sqsubseteq s d'$  by auto
    from x d'D have 2:  $\dots \sqsubseteq x$  by auto
    from trans[OF 1 2] show  $d \sqsubseteq x$  using dD sDA d'D DA xA by auto
  next
  fix b assume bA:  $b \in A$  and Db: bound D ( $\sqsubseteq$ ) b
  have bound (s ' D) ( $\sqsubseteq$ ) b

```

```

proof safe
  fix d assume dD:  $d \in D$ 
  from dD Db Irange have bound (I d) ( $\sqsubseteq$ ) b by auto
  with s dD bA show  $s d \sqsubseteq b$  by auto
qed
with x bA show  $x \sqsubseteq b$  by auto
qed
assume f: well-related-set-continuous  $A (\sqsubseteq) B (\preceq) f$ 
  and Dt: extreme-bound  $A (\sqsubseteq) D t$  and D0:  $D \neq \{\}$ 
from Dt have tA:  $t \in A$  by auto
have fmono: monotone-on  $A (\sqsubseteq) (\preceq) f$ 
  by (auto intro!: continuous-imp-monotone-on[OF f] pair-well-related)
show extreme-bound  $B (\preceq) (f ' D) (f t)$ 
proof (safe intro!: extreme-boundI)
  from f tA show  $f t \in B$  by auto
  fix d assume dD:  $d \in D$ 
  from dD Dt have dt:  $d \sqsubseteq t$  by auto
from dD Dt DA show  $f d \preceq f t$  by (auto intro!: monotone-onD[OF fmono])
next
  fix b assume fDb: bound  $(f ' D) (\preceq) b$  and bB:  $b \in B$ 
  from Dx Dt have  $x \sim t$  by (auto intro!: sympartI elim!: extreme-boundE)
  with extreme-bound-sym-trans[OF sDA x this tA]
  have extreme-bound  $A (\sqsubseteq) (s ' D) t$  by auto
  from f[THEN continuousD, OF wsD - sDA this] D0
  have ft: extreme-bound  $B (\preceq) (f ' s ' D) (f t)$  by auto
  have bound  $(f ' s ' D) (\preceq) b$ 
  proof (safe)
    fix d assume dD:  $d \in D$ 
    from Irange dD have IdD:  $I d \subseteq D$  by auto
    with DA have IdA:  $I d \subseteq A$  by auto
    from directed-setD[OF Idir[rule-format, OF dD], of  $\{\}$ ]
    have Idne:  $I d \neq \{\}$  by auto
    have fsd: extreme-bound  $B (\preceq) (f ' I d) (f (s d))$ 
      apply (rule IH2[OF - - IdA f Idne s[OF dD]])
      using Icard Idir dD by auto
    from IdD have  $f ' I d \subseteq f ' D$  by auto
    from bound-subset[OF this fDb] fsd bB
    show  $f (s d) \preceq b$  by auto
  qed
with ft bB show  $f t \preceq b$  by auto
qed
qed
qed
qed

```

The next Theorem corresponds to Proposition 5.9 of [4], without anti-symmetry on  $A$ .

**theorem** (*in* *quasi-ordered-set*) *well-complete-iff-directed-complete*:  
*(nonempty  $\sqcap$  well-related-set)-complete*  $A (\sqsubseteq) \longleftrightarrow$  *directed-set-complete*  $A (\sqsubseteq)$

```

(is ?l  $\longleftrightarrow$  ?r)
proof (intro iffI)
  show ?l  $\implies$  ?r
    by (auto intro!: completeI dest!: directed-completeness-lemma(1))
  assume r: ?r
  show ?l
    apply (rule complete-subclass[OF r])
    using well-related-set.directed-set
    by auto
qed

```

The next Theorem corresponds to Corollary 3 of [9] without any assumptions on the codomain  $B$  and without antisymmetry on the domain  $A$ .

```

theorem (in quasi-ordered-set)
  fixes leB (infix  $\langle \trianglelefteq \rangle$  50)
  assumes comp: (nonempty  $\sqcap$  well-related-set)–complete A ( $\sqsubseteq$ )
  shows well-related-set–continuous A ( $\sqsubseteq$ ) B ( $\trianglelefteq$ )  $f \longleftrightarrow$  directed-set–continuous
  A ( $\sqsubseteq$ ) B ( $\trianglelefteq$ )  $f$ 
  (is ?l  $\longleftrightarrow$  ?r)
proof (intro iffI)
  assume l: ?l
  show ?r
    using continuous-carrierD[OF l]
    using directed-completeness-lemma(2)[OF comp - - l]
    by (auto intro!: continuousI)
next
  assume r: ?r
  show ?l
    apply (rule continuous-subclass[OF - r])
    using well-related-set.directed-set by auto
qed

end

```

## References

- [1] S. Abramsky and A. Jung. *Domain Theory*. Number III in Handbook of Logic in Computer Science. Oxford University Press, 1994.
- [2] G. Bancerek. The ordinal numbers. *Journal of Formalized Mathematics*, 1, 1989.
- [3] J. C. Blanchette, A. Popescu, and D. Traytel. Cardinals in Isabelle/HOL. In G. Klein and R. Gamboa, editors, *Interactive Theorem Proving - 5th International Conference, ITP 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17*,

2014. *Proceedings*, volume 8558 of *Lecture Notes in Computer Science*, pages 111–127. Springer, 2014.
- [4] P. M. Cohn. *Universal Algebra*. Harper & Row, 1965.
- [5] J. Dubut and A. Yamada. Fixed point theorems for non-transitive relations. *Log. Methods Comput. Sci.*, 18(1), 2022.
- [6] A. Finkel and J. Goubault-Larrecq. Forward Analysis for WSTS, Part I: Completions. In S. Albers and J.-Y. Marion, editors, *26th International Symposium on Theoretical Aspects of Computer Science*, volume 3 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 433–444, Dagstuhl, Germany, 2009. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [7] J. Goubault-Larrecq. *Non-Hausdorff Topology and Domain Theory: Selected Topics in Point-Set Topology*, volume 22 of *New Mathematical Monographs*. Cambridge University Press, 2013.
- [8] T. Iwamura. A lemma on directed sets. In *Zenkoku Shijo Sugaku Danwakai*, number 262, pages 107–111, 1944. in Japanese.
- [9] G. Markowsky. Chain-complete posets and directed sets with applications. *Algebra Universalis*, 6:53–68, 1976.
- [10] L. C. Paulson and K. Grabczewski. Mechanizing set theory. *J. Autom. Reason.*, 17(3):291–323, 1996.
- [11] D. Scott. Outline of a Mathematical Theory of Computation. Technical Report PRG02, OUCL, 1970.
- [12] L. A. Skorniakov. *Complemented modular lattices and regular rings*. Oliver & Boyd, 1964.
- [13] G. Winskel. *The Formal Semantics of Programming Languages: An Introduction*. Foundations of Computing. The MIT Press, 1993.
- [14] 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.