Open Induction

Mizuhito Ogawa Christian Sternagel*

September 19, 2015

Abstract

A proof of the open induction schema based on [1].

Contents

1 Open Induction 1
1.1 (Greatest) Lower Bounds and Chains . . . . . . . . . . . . . 1
1.2 Open Properties . . . . . . . . . . . . . . . . . . . . . . . . 3
1.3 Downward Completeness . . . . . . . . . . . . . . . . . . . . 3
1.4 The Open Induction Schema . . . . . . . . . . . . . . . . . 3
1.5 Universal Open Induction Schemas . . . . . . . . . . . . . . 4
1.6 Type Class of Downward Complete Orders . . . . . . . . . 5

1 Open Induction

theory Open-Induction
imports
  Main
  ../Well-Quasi-Orders/Restricted-Predicates
begin

1.1 (Greatest) Lower Bounds and Chains

A set $B$ has the lower bound $x$ w.r.t. to the order $P::'a ⇒ 'a ⇒ bool$ iff $x$ is less than or equal to every element of $B$.

definition lb where
  lb P B x ≡ ∀ y∈B. P x y

A set $B$ has the greatest lower bound $x$ (w.r.t. $P$) iff $x$ is a lower bound and less than or equal to every other lower bound of $B$.

definition glb where

*The research was partly funded by the Austrian Science Fund (FWF): J3202.
\[
\text{glb } P B x \equiv \text{lb } P B x \land (\forall y. \text{lb } P B y \rightarrow P y x)
\]

A subset \( C \) of \( A \) is a **chain on** \( A \) (w.r.t. \( P \)) iff for all pairs of elements of \( C \), one is less than or equal to the other one.

**definition chain-on** where  
\[
\text{chain-on } P C A \equiv C \subseteq A \land (\forall x \in C. \forall y \in C. P x y \lor P y x)
\]

A chain \( M \) on \( A \) (w.r.t. \( P \)) is a **maximal chain** iff there is no chain on \( A \) that is a superset of \( M \).

**definition max-chain-on** where  
\[
\text{max-chain-on } P M A \equiv \text{chain-on } P M A \land (\forall C. \text{chain-on } P C A \land M \subseteq C \rightarrow M = C)
\]

**lemma chain-onI** [Pure.intro]:  
\[
C \subseteq A \Rightarrow (\exists x y. [x \in C ; y \in C] \Rightarrow P x y \lor P y x) \Rightarrow \text{chain-on } P C A
\]

**lemma chain-on-subset**:  
\[
A \subseteq B \Rightarrow \text{chain-on } P C A \Rightarrow \text{chain-on } P C B
\]

**lemma chain-on-imp-subset**:  
\[
\text{chain-on } P C A \Rightarrow C \subseteq A \text{ by (simp add: chain-on-def)}
\]

**lemma chain-on-Union**:  
\[
\text{assumes } C \in \text{chains } \{ C. \text{chain-on } P C A \} \text{ (is } C \in \text{ chains } ?A) \text{ shows } \text{chain-on } P (\bigcup C) A
\]

**proof**  
from assms have \( C \subseteq ?A \) and  

* [rule-format]: \( \forall x \in C. \forall y \in C. x \subseteq y \lor y \subseteq x \)  
  by (auto simp: chain-on-def chain-subset-def)  

then show \( \bigcup C \subseteq A \) unfolding chain-on-def by blast  

fix \( x y \) assume \( x \in \bigcup C \) and \( y \in \bigcup C \)  
then obtain \( X Y \)  
where \( X \in C \) and \( Y \in C \) and \( x \in X \) and \( y \in Y \) by auto  
with \( C \subseteq ?A \) have \( X \subseteq A \) and \( Y \subseteq A \)  
and \( \text{chain-on } P X A \) and \( \text{chain-on } P Y A \) unfolding chain-on-def by auto  
with \( x \in X \) and \( y \in Y \) show \( P x y \lor P y x \)  
using * [OF \( X \in C \), \( Y \in C \)]  
unfolding chain-on-def by blast  

qed

**lemma chain-on-glb**:  
\[
\text{assumes } \text{go-on } P A
\]
shows \( \text{chain-on } P C A \Rightarrow C \neq \{\} \Rightarrow \text{glb } P C x \Rightarrow x \in A \Rightarrow y \in A \Rightarrow P y x \Rightarrow \text{chain-on } P \{y \cup C\} A
\]
using \( \text{go-on-imp-reflp-on } [\text{OF assms, unfolded reflp-on-def, rule-format, of } y] \)  
and \( \text{go-on-imp-transp-on } [\text{OF assms, unfolded transp-on-def}] \)
unfolding chain-on-def glb-def lb-def by blast

2
1.2 Open Properties

**definition open-on** where

\[
open-on\ P\ Q\ A \equiv \\
\forall\ C.\ chain-on\ P\ C\ A \land C \neq \emptyset \land (\exists x \in A.\ \text{glb} P\ C\ x \land Q\ x) \rightarrow (\exists y \in C.\ Q\ y)
\]

**lemma open-on-glb**: 

\[
\llbracket\ \text{chain-on} P\ C\ A;\ C \neq \emptyset;\ open-on\ P\ Q\ A;\ \forall x \in C.\ \neg Q\ x;\ x \in A;\ \text{glb} P\ C\ x\rrbracket \implies \neg Q\ x
\]

by (auto simp: open-on-def)

**lemma max-chain-on-exists**: 

\[
\exists M.\ max-chain-on\ P\ M\ A
\]

**proof**

- let \( ?S = \{ C.\ chain-on P\ C\ A \} \)
- have \( \bigwedge C.\ C \in \text{chains} ?S \implies \bigcup C \in ?S \)
  using chain-on-Union and chain-on-imp-subset by blast
- with Zorn-Lemma [of \( ?S \)]
  obtain \( M \) where \( M \in ?S \) and \( \ast: \forall z \in ?S.\ M \subseteq z \implies z = M \) by blast
- then have \( M \subseteq A \) and \( \text{chain-on} P\ M\ A \) by (auto dest: chain-on-imp-subset)
- moreover \{ 
  - fix \( C \) assume \( chain-on P\ C\ A \) and \( M \subseteq C \)
  - with \( \ast \) have \( M = C \)
  - using chain-on-imp-subset [OF \( \langle chain-on P\ C\ A \rangle \)]
  - by blast \}
- ultimately show \( ?\text{thesis} \) by (auto simp: max-chain-on-def)

qed

1.3 Downward Completeness

An order \( P \) is **downward-complete** on \( A \) iff every non-empty chain on \( A \) has a greatest lower bound in \( A \).

**definition dc-on** where

\[
dc-on\ P\ A \equiv \forall C.\ chain-on\ P\ C\ A \land C \neq \emptyset \rightarrow (\exists x \in A.\ \text{glb} P\ C\ x)
\]

1.4 The Open Induction Schema

**lemma open-induct-on** [consumes 4]:

- assumes \( \text{go-on} P\ A \) and \( dc-on\ P\ A \) and \( open-on\ P\ Q\ A \)
- and \( x \in A \)
- and \( \text{ind}: \bigwedge x.\ [x \in A;\ \bigwedge y.\ [y \in A;\ strict P\ y\ x] \implies Q\ y] \implies Q\ x \)
- shows \( Q\ x \)

**proof** (rule ccontr)

- note refl = qo-on-imp-reflp-on [OF \( \langle qo-on P\ A \rangle \), unfolded reflp-on-def, rule-format]
- assume \( \neg Q\ x \)
- let \( ?A = \{ x \in A.\ \neg Q\ x \} \)
- from max-chain-on-exists [of \( P ?A \)] obtain \( M \) where
  chain: \( \text{chain-on} P\ M\ ?A \) and
max: \( \bigwedge C. \) chain-on \( P \ C \ ?A \implies M \subseteq C \implies M = C \) by (auto simp: max-chain-on-def)

from chain have \( M \subseteq ?A \) by (auto simp: chain-on-imp-subset)
show False
proof (cases \( M = \{\} \))
assume \( M = \{\} \)
moreover have chain-on \( P \ \{x\} \ ?A \)
using refl and \( \{x \in A\} \) and \( \neg Q x \) by (simp add: chain-on-def)
ultimately show False using max by blast
next
assume \( M \neq \{\} \)
have \( ?A \subseteq A \) by blast
with chain have chain-on \( P \ M A \)
using chain-on-subset by blast
moreover with \( \langle \text{dc-on } PA \rangle \) and \( \langle \{M \neq \{\}\} \rangle \) obtain \( m \) where
\( m \in A \) and \( \text{glb } P M m \)
unfolding dc-on-def by auto
ultimately have \( \neg Q m \) and \( m \in ?A \) using max by blast

1.5 Universal Open Induction Schemas

Open induction on quasi-orders (i.e., preorder).

lemma (in preorder) dc-open-induct [consumes 2]:
assumes \( \langle \text{dc-on } (op \leq) \rangle \ \text{UNIV} \)
and \( \langle \text{open-on } (op \leq) \rangle \ \text{Q \ UNIV} \)
and \( \bigwedge x. (\bigwedge y. y < x \implies Q y) \implies Q x \)
shows \( Q x \)
proof —
have \( \text{go-on } (op \leq) \ \text{UNIV} \)
unfolding \textit{go-on-UNIV-conv}
unfolding \textit{less-le-not-le [symmetric]} ..
moreover have \textit{dc-on (op \leq) UNIV by fact}
ultimately show \( Q \times \)
using \textit{assms and open-induct-on [of \( \text{op} \leq \text{UNIV \ Q} \)]}
unfolding \textit{less-le-not-le by blast}

\textit{qed}

1.6 Type Class of Downward Complete Orders

\textbf{class} \textit{dcorder} = \textit{preorder}
\textbf{assumes} \textit{dc: \[\text{chain-on (op \leq) C \ \text{UNIV}; \ C \neq \{\}] \Rightarrow (\exists x. \text{glb (op \leq) C} \ x)\]}
\textbf{begin}

\textbf{lemma} \textit{dc-on-UNIV}: \textit{dc-on (op \leq) UNIV}
using \textit{dc unfolding} \textit{dc-on-def by blast}

\textbf{Open induction on downward-complete orders.}
\textbf{lemmas} \textit{open-induct [consumes 1]} = \textit{dc-open-induct [OF dc-on-UNIV]}

\textbf{end}
end

\textbf{References}