

The Stone-Čech Compactification

Mike Stannett

February 6, 2026

Contents

1	C^*-embedding	4
2	Weak topologies	7
2.1	Tychonov spaces carry the weak topology induced by $C^*(X)$	9
2.2	A topology is a weak topology if it admits a continuous function set that separates points from closed sets	10
2.3	A product topology is the weak topology induced by its projections if the projections separate points from closed sets. . .	11
2.4	Evaluation is an embedding for weak topologies	12
3	Compactification	12
3.1	Definition	12
3.2	Example: The Alexandroff compactification of a non-compact locally-compact Hausdorff space	13
3.3	Example: The closure of a subset of a compact space	13
3.4	Example: A compact space is a compactification of itself . . .	13
3.5	Example: A closed non-trivial real interval is a compactification of its interior	13
4	The Stone-Čech compactification of a Tychonov space	13
4.1	Definition of βX	15
4.2	βX is a compactification of X	15
4.3	Evaluation is a C^* -embedding of X into βX	16
4.4	The Stone-Čech Extension Property: Any continuous map from X to a compact Hausdorff space K extends uniquely to a continuous map from βX to K	17

Building on parts of HOL-Analysis, we provide mathematical components for work on the Stone-Čech compactification. The main concepts covered are: C^* -embedding, weak topologies and compactification, focusing in particular on the Stone-Čech compactification of an arbitrary Tychonov space X . Many of the proofs given here derive from those of Willard (*General*

Topology, 1970, Addison-Wesley) and Walker (*The Stone-Čech Compactification*, 1974, Springer-Verlag).

Using traditional topological proof strategies we define the evaluation and projection functions for product spaces, and show that product spaces carry the weak topology induced by their projections whenever those projections separate points both from each other and from closed sets.

In particular, we show that the evaluation map from an arbitrary Tychonov space X into βX is a dense C^* -embedding, and then verify the Stone-Čech Extension Property: any continuous map from X to a compact Hausdorff space K extends uniquely to a continuous map from βX to K .

theory *Stone-Cech*

imports *HOL.Topological-Spaces*

HOL.Set

HOL-Analysis.Urysohn

begin

Concrete definitions of finite intersections and arbitrary unions, and their relationship to the *Analysis.Abstract_Topology* versions.

definition *finite-intersections-of* :: 'a set set \Rightarrow 'a set set

where *finite-intersections-of* $S = \{ (\bigcap F) \mid F . F \subseteq S \wedge \text{finite}' F \}$

definition *arbitrary-unions-of* :: 'a set set \Rightarrow 'a set set

where *arbitrary-unions-of* $S = \{ (\bigcup F) \mid F . F \subseteq S \}$

lemma *generator-imp-arbitrary-union*:

shows $S \subseteq \text{arbitrary-unions-of } S$

<proof>

lemma *finite-intersections-container*:

shows $\forall s \in \text{finite-intersections-of } S . \bigcup S \cap s = s$

<proof>

lemma *generator-imp-finite-intersection*:

shows $S \subseteq \text{finite-intersections-of } S$

<proof>

lemma *finite-intersections-equiv*:

shows $(\text{finite}' \text{ intersection-of } (\lambda x . x \in S)) U \longleftrightarrow U \in \text{finite-intersections-of } S$

<proof>

lemma *arbitrary-unions-equiv*:

shows $(\text{arbitrary union-of } (\lambda x . x \in S)) U \longleftrightarrow U \in \text{arbitrary-unions-of } S$

<proof>

Supplementary information about topological bases and the topologies they

generate

definition *base-generated-on-by* :: 'a set \Rightarrow 'a set set \Rightarrow 'a set set
where *base-generated-on-by* $X S = \{ X \cap s \mid s . s \in \text{finite-intersections-of } S \}$

definition *opens-generated-on-by* :: 'a set \Rightarrow 'a set set \Rightarrow 'a set set
where *opens-generated-on-by* $X S = \text{arbitrary-unions-of } (\text{base-generated-on-by } X S)$

definition *base-generated-by* :: 'a set set \Rightarrow 'a set set
where *base-generated-by* $S = \text{finite-intersections-of } S$

definition *opens-generated-by* :: 'a set set \Rightarrow 'a set set
where *opens-generated-by* $S = \text{arbitrary-unions-of } (\text{base-generated-by } S)$

lemma *generators-are-basic*:
shows $S \subseteq \text{base-generated-by } S$
(*proof*)

lemma *basics-are-open*:
shows $\text{base-generated-by } S \subseteq \text{opens-generated-by } S$
(*proof*)

lemma *generators-are-open*:
shows $S \subseteq \text{opens-generated-by } S$
(*proof*)

lemma *generated-topospace*:
assumes $T = \text{topology-generated-by } S$
shows $\text{topospace } T = \bigcup S$
(*proof*)

lemma *base-generated-by-alt*:
shows $\text{base-generated-by } S = \text{base-generated-on-by } (\bigcup S) S$
(*proof*)

lemma *opens-generated-by-alt*:
shows $\text{opens-generated-by } S = \text{arbitrary-unions-of } (\text{finite-intersections-of } S)$
(*proof*)

lemma *opens-generated-unfolded*:
shows $\text{opens-generated-by } S = \{ \bigcup A \mid A . A \subseteq \{ \bigcap B \mid B . \text{finite}' B \wedge B \subseteq S \} \}$
(*proof*)

lemma *opens-eq-generated-topology*:
shows $\text{openin } (\text{topology-generated-by } S) U \iff U \in \text{opens-generated-by } S$
(*proof*)

1 C^* -embedding

abbreviation *continuous-from-to*

$:: 'a \text{ topology} \Rightarrow 'b \text{ topology} \Rightarrow ('a \Rightarrow 'b) \text{ set} (\langle \text{cts}[-, -] \rangle)$
where *continuous-from-to* $X Y \equiv \{ f . \text{continuous-map } X Y f \}$

abbreviation *continuous-from-to-extensional*

$:: 'a \text{ topology} \Rightarrow 'b \text{ topology} \Rightarrow ('a \Rightarrow 'b) \text{ set} (\langle \text{cts}_E[-, -] \rangle)$
where *continuous-from-to-extensional* $X Y \equiv (\text{topspace } X \rightarrow_E \text{topspace } Y) \cap \text{cts}[X, Y]$

abbreviation *continuous-maps-from-to-shared-where* $::$

$'a \text{ topology} \Rightarrow ('b \text{ topology} \Rightarrow \text{bool}) \Rightarrow ('a \Rightarrow 'b) \text{ set} \Rightarrow \text{bool} (\langle \text{cts}'\text{-on } - \text{ to}'\text{-shared } - \rangle)$
where *continuous-maps-from-to-shared-where* $X P$
 $\equiv (\lambda fs . (\exists Y . P Y \wedge fs \subseteq \text{cts}[X, Y]))$

definition *dense-in* $:: 'a \text{ topology} \Rightarrow 'a \text{ set} \Rightarrow 'a \text{ set} \Rightarrow \text{bool}$

where *dense-in* $T A B \equiv T \text{closure-of } A = B$

lemma *dense-in-closure*:

assumes *dense-in* $T A B$
shows *dense-in* $(\text{subtopology } T B) A B$
 $\langle \text{proof} \rangle$

abbreviation *dense-embedding* $:: 'a \text{ topology} \Rightarrow 'b \text{ topology} \Rightarrow ('a \Rightarrow 'b) \Rightarrow \text{bool}$

where *dense-embedding* $\text{small big } f \equiv (\text{embedding-map } \text{small big } f) \wedge \text{dense-in } \text{big } (f' \text{topspace } \text{small}) (\text{topspace } \text{big})$

lemma *continuous-maps-on-dense-subset*:

assumes $(\text{cts-on } X \text{ to-shared Hausdorff-space}) \{f, g\}$
and *dense-in* $X D (\text{topspace } X)$
and $\forall x \in D . f x = g x$
shows $\forall x \in \text{topspace } X . f x = g x$
 $\langle \text{proof} \rangle$

lemma *continuous-map-on-dense-embedding*:

assumes $(\text{cts-on } X \text{ to-shared Hausdorff-space}) \{f, g\}$
and *dense-embedding* $D X e$
and $\forall d \in \text{topspace } D . (f \circ e) d = (g \circ e) d$
shows $\forall x \in \text{topspace } X . f x = g x$
 $\langle \text{proof} \rangle$

definition *range'* $:: 'a \text{ topology} \Rightarrow ('a \Rightarrow \text{real}) \Rightarrow \text{real set}$

where *range'* $X f = \text{euclideanreal closure-of } (f ' \text{topspace } X)$

abbreviation *fbounded-below* :: ('a \Rightarrow real) \Rightarrow 'a topology \Rightarrow bool
where *fbounded-below* f X \equiv (\exists m . \forall y \in *topspace* X . f y \geq m)

abbreviation *fbounded-above* :: ('a \Rightarrow real) \Rightarrow 'a topology \Rightarrow bool
where *fbounded-above* f X \equiv (\exists M . \forall y \in *topspace* X . f y \leq M)

abbreviation *fbounded* :: ('a \Rightarrow real) \Rightarrow 'a topology \Rightarrow bool
where *fbounded* f X \equiv (\exists m M . \forall y \in *topspace* X . m \leq f y \wedge f y \leq M)

lemma *fbounded-iff*:
shows *fbounded* f X \longleftrightarrow *fbounded-below* f X \wedge *fbounded-above* f X
<proof>

abbreviation *c-of* :: 'a topology \Rightarrow ('a \Rightarrow real) set (\langle C(-) \rangle)
where C(X) \equiv { f . *continuous-map* X *euclideanreal* f }

abbreviation *cstar-of* :: 'a topology \Rightarrow ('a \Rightarrow real) set (\langle C*(-) \rangle)
where C* X \equiv { f | f . f \in *c-of* X \wedge *fbounded* f X }

definition *cstar-id* :: 'a topology \Rightarrow ('a \Rightarrow real) \Rightarrow 'a \Rightarrow real
where *cstar-id* X = (λ f \in C* X . f)

abbreviation *c-embedding* :: 'a topology \Rightarrow 'b topology \Rightarrow ('a \Rightarrow 'b) \Rightarrow bool
where *c-embedding* S X e \equiv *embedding-map* S X e \wedge
 $(\forall$ fS \in C(S) . \exists fX \in C(X) . \forall x \in *topspace* S . fS x =
fX (e x))

abbreviation *cstar-embedding* :: 'a topology \Rightarrow 'b topology \Rightarrow ('a \Rightarrow 'b) \Rightarrow bool
where *cstar-embedding* S X e \equiv *embedding-map* S X e \wedge
 $(\forall$ fS \in C*(S) . \exists fX \in C*(X) . \forall x \in *topspace* S . fS x
= fX (e x))

definition *c-embedded* :: 'a topology \Rightarrow 'b topology \Rightarrow bool
where *c-embedded* S X \equiv (\exists e . *c-embedding* S X e)

definition *cstar-embedded* :: 'a topology \Rightarrow 'b topology \Rightarrow bool
where *cstar-embedded* S X \equiv (\exists e . *cstar-embedding* S X e)

lemma *bounded-range-iff-fbounded*:
assumes f \in C X
shows *bounded* (f ' *topspace* X) \longleftrightarrow *fbounded* f X
(is ?lhs \longleftrightarrow ?rhs)
<proof>

Combinations of functions in C(X) and C*(X)

abbreviation *fconst* :: real \Rightarrow 'a \Rightarrow real
where *fconst* v \equiv (λ x . v)

definition $fmin :: ('a \Rightarrow real) \Rightarrow ('a \Rightarrow real) \Rightarrow ('a \Rightarrow real)$
where $fmin\ f\ g = (\lambda\ x.\ min\ (f\ x)\ (g\ x))$

definition $fmax :: ('a \Rightarrow real) \Rightarrow ('a \Rightarrow real) \Rightarrow ('a \Rightarrow real)$
where $fmax\ f\ g = (\lambda\ x.\ max\ (f\ x)\ (g\ x))$

definition $fmid :: ('a \Rightarrow real) \Rightarrow ('a \Rightarrow real) \Rightarrow ('a \Rightarrow real) \Rightarrow 'a \Rightarrow real$
where $fmid\ f\ m\ M = fmax\ m\ (fmin\ f\ M)$

definition $fbound :: ('a \Rightarrow real) \Rightarrow real \Rightarrow real \Rightarrow 'a \Rightarrow real$
where $fbound\ f\ m\ M = fmid\ f\ (fconst\ m)\ (fconst\ M)$

lemma $fmin-cts$:
assumes $(f \in C\ X) \wedge (g \in C\ X)$
shows $fmin\ f\ g \in C\ X$
 $\langle proof \rangle$

lemma $fmax-cts$:
assumes $(f \in C\ X) \wedge (g \in C\ X)$
shows $fmax\ f\ g \in C\ X$
 $\langle proof \rangle$

lemma $fmid-cts$:
assumes $(f \in C\ X) \wedge (m \in C\ X) \wedge (M \in C\ X)$
shows $fmid\ f\ m\ M \in C\ X$
 $\langle proof \rangle$

lemma $fconst-cts$:
shows $fconst\ v \in C\ X$
 $\langle proof \rangle$

lemma $fbound-cts$:
assumes $f \in C\ X$
shows $fbound\ f\ m\ M \in C\ X$
 $\langle proof \rangle$

Bounded and bounding functions

lemma $fconst-bounded$:
shows $fbounded\ (fconst\ v)\ X$
 $\langle proof \rangle$

lemma $fmin-bounded-below$:
assumes $fbounded-below\ f\ X \wedge fbounded-below\ g\ X$
shows $fbounded-below\ (fmin\ f\ g)\ X$
 $\langle proof \rangle$

lemma *fmax-bounded-above*:
assumes *fbounded-above f X* \wedge *fbounded-above g X*
shows *fbounded-above (fmax f g) X*
 \langle *proof* \rangle

lemma *fmid-bounded*:
assumes *fbounded m X* \wedge *fbounded M X*
shows *fbounded (fmid f m M) X*
 \langle *proof* \rangle

lemma *fbound-bounded*:
shows *fbounded (fbound f m M) X*
 \langle *proof* \rangle

Members of $C^*(X)$

lemma *fconst-cstar*:
shows *fconst v* $\in C^* X$
 \langle *proof* \rangle

lemma *fbound-cstar*:
assumes *f* $\in C X$
shows *fbound f m M* $\in C^* X$
 \langle *proof* \rangle

lemma *cstar-nonempty*:
shows $\{\}$ $\neq C^* X$
 \langle *proof* \rangle

2 Weak topologies

definition *funcset-types* :: *'a set* \Rightarrow (*'b* \Rightarrow *'a* \Rightarrow *'c*) \Rightarrow (*'b* \Rightarrow *'c topology*) \Rightarrow *'b set*
 \Rightarrow *bool*
where *funcset-types S F T I* = (\forall *i* $\in I$. *F i* $\in S \rightarrow$ *topspace (T i)*)

lemma *cstar-types*:
shows *funcset-types (topspace X) (cstar-id X) ($\lambda f \in C^* X$. euclideanreal)* ($C^* X$)
 \langle *proof* \rangle

lemma *cstar-types-restricted*:
shows *funcset-types (topspace X) (cstar-id X)*
 $(\lambda f \in C^* X$. *subtopology euclideanreal (range' X f))*) ($C^* X$)
 \langle *proof* \rangle

definition *inverse'* :: (*'a* \Rightarrow *'b*) \Rightarrow *'a set* \Rightarrow *'b set* \Rightarrow *'a set*
where *inverse' f source target* = { *x* \in *source* . *f x* \in *target* }

lemma *inverse'-alt:*

shows $\text{inverse}' f s t = (f -' t) \cap s$
<proof>

definition *open-sets-induced-by-func* :: $('a \Rightarrow 'b) \Rightarrow 'a \text{ set} \Rightarrow 'b \text{ topology} \Rightarrow 'a \text{ set set}$

where *open-sets-induced-by-func* $f \text{ source } T$
 $= \{ (\text{inverse}' f \text{ source } V) \mid V . \text{openin } T V \wedge f \in \text{source} \rightarrow \text{topspace } T \}$

definition *weak-generators* :: $'a \text{ set} \Rightarrow ('b \Rightarrow 'a \Rightarrow 'c) \Rightarrow ('b \Rightarrow 'c \text{ topology}) \Rightarrow 'b \text{ set} \Rightarrow 'a \text{ set set}$

where *weak-generators* $\text{source funcs tops index}$
 $= \bigcup \{ \text{open-sets-induced-by-func } (funcs \ i) \text{ source } (tops \ i) \mid i . i \in \text{index} \}$

definition *weak-base* :: $'a \text{ set} \Rightarrow ('b \Rightarrow 'a \Rightarrow 'c) \Rightarrow ('b \Rightarrow 'c \text{ topology}) \Rightarrow 'b \text{ set} \Rightarrow 'a \text{ set set}$

where *weak-base* $\text{source funcs tops index} = \text{base-generated-by } (\text{weak-generators } \text{source funcs tops index})$

definition *weak-opens* :: $'a \text{ set} \Rightarrow ('b \Rightarrow 'a \Rightarrow 'c) \Rightarrow ('b \Rightarrow 'c \text{ topology}) \Rightarrow 'b \text{ set} \Rightarrow 'a \text{ set set}$

where *weak-opens* $\text{source funcs tops index} = \text{opens-generated-by } (\text{weak-generators } \text{source funcs tops index})$

definition *weak-topology* :: $'a \text{ set} \Rightarrow ('b \Rightarrow 'a \Rightarrow 'c) \Rightarrow ('b \Rightarrow 'c \text{ topology}) \Rightarrow 'b \text{ set} \Rightarrow 'a \text{ topology}$

where *weak-topology* $\text{source funcs tops index}$
 $= \text{topology-generated-by } (\text{weak-generators } \text{source funcs tops index})$

lemma *weak-topology-alt:*

shows $\text{openin } (\text{weak-topology } S F T I) U \iff U \in \text{weak-opens } S F T I$
<proof>

lemma *weak-generators-exist-for-each-point-and-axis:*

assumes $x \in S$
and $\text{funcset-types } S F T I$
and $i \in I$
and $b = \text{inverse}' (F \ i) S (\text{topspace } (T \ i))$
and $F \ i \in S \rightarrow \text{topspace } (T \ i)$
shows $x \in b \wedge b \in \text{weak-generators } S F T I$
<proof>

lemma *weak-generators-topspace:*

assumes $W = \text{weak-topology } S F T I$
shows $\text{topspace } W = \bigcup (\text{weak-generators } S F T I)$
<proof>

lemma *weak-topology-topospace*:
assumes $W = \text{weak-topology } S F T I$
and $\text{funcset-types } S F T I$
shows $(I = \{\}) \longrightarrow \text{topospace } W = \{\}) \wedge (I \neq \{\}) \longrightarrow \text{topospace } W = S)$
 $\langle \text{proof} \rangle$

lemma *weak-opens-nhood-base*:
assumes $W = \text{weak-topology } S F T I$
and $\text{openin } W U$
and $x \in U$
shows $\exists b \in \text{weak-base } S F T I . x \in b \wedge b \subseteq U$
 $\langle \text{proof} \rangle$

lemma *opens-generate-opens*:
assumes $\forall b \in S . \text{openin } T b$
shows $\forall U \in \text{opens-generated-by } S . \text{openin } T U$
 $\langle \text{proof} \rangle$

lemma *weak-topology-is-weakest*:
assumes $W = \text{weak-topology } S F T I$
and $\text{funcset-types } S F T I$
and $\text{topospace } X = \text{topospace } W$
and $\forall i \in I . \text{continuous-map } X (T i) (F i)$
and $\text{openin } W U$
shows $\text{openin } X U$
 $\langle \text{proof} \rangle$

lemma *weak-generators-continuous*:
assumes $W = \text{weak-topology } S F T I$
and $\text{funcset-types } S F T I$
and $i \in I$
shows $\text{continuous-map } W (T i) (F i)$
 $\langle \text{proof} \rangle$

lemma *funcset-types-on-empty*:
shows $\text{funcset-types } \{\} F T I$
 $\langle \text{proof} \rangle$

lemma *weak-topology-on-empty*:
assumes $W = \text{weak-topology } \{\} F T I$
shows $\forall U . \text{openin } W U \longleftrightarrow U = \{\}$
 $\langle \text{proof} \rangle$

2.1 Tychonov spaces carry the weak topology induced by $C^*(X)$

abbreviation *tych-space* :: 'a topology \Rightarrow bool
where $\text{tych-space } X \equiv \text{t1-space } X \wedge \text{completely-regular-space } X$

abbreviation *compact-Hausdorff* :: 'a topology \Rightarrow bool
where *compact-Hausdorff* X \equiv *compact-space* X \wedge *Hausdorff-space* X

lemma *compact-Hausdorff-imp-tych*:
assumes *compact-Hausdorff* K
shows *tych-space* K
 ⟨*proof*⟩

lemma *tych-space-imp-Hausdorff*:
assumes *tych-space* X
shows *Hausdorff-space* X
 ⟨*proof*⟩

lemma *cstar-range-restricted*:
assumes $f \in C^* X$
and $U \subseteq \text{topspace } \text{euclideanreal}$
shows $\text{inverse}' f (\text{topspace } X) U = \text{inverse}' f (\text{topspace } X) (U \cap \text{range}' X f)$
 ⟨*proof*⟩

lemma *weak-restricted-topology-eq-weak*:
shows $\text{weak-topology } (\text{topspace } X) (\text{cstar-id } X) (\lambda f \in C^* X . \text{euclideanreal}) (C^* X)$
 $= \text{weak-topology } (\text{topspace } X) (\text{cstar-id } X) (\lambda f \in C^* X . \text{subtopology } \text{euclideanreal } (\text{range}' X f)) (C^* X)$
 ⟨*proof*⟩

2.2 A topology is a weak topology if it admits a continuous function set that separates points from closed sets

definition *funcset-separates-points* :: 'a topology \Rightarrow ('b \Rightarrow 'a \Rightarrow 'c) \Rightarrow 'b set \Rightarrow bool
where *funcset-separates-points* X F I
 $= (\forall x \in \text{topspace } X . \forall y \in \text{topspace } X . x \neq y \longrightarrow (\exists i \in I . F i x \neq F i y))$

definition *funcset-separates-points-from-closed-sets* ::
 'a topology \Rightarrow ('b \Rightarrow 'a \Rightarrow 'c) \Rightarrow ('b \Rightarrow 'c topology) \Rightarrow 'b set \Rightarrow bool
where *funcset-separates-points-from-closed-sets* X F T I
 $= (\forall x . \forall A . \text{closedin } X A \wedge x \in (\text{topspace } X - A)$
 $\longrightarrow (\exists i \in I . F i x \notin (T i) \text{closure-of } (F i ' A)))$

lemma *funcset-separates-points-from-closed-sets-imp-weak*:
assumes *funcset-separates-points-from-closed-sets* X F T I
and $\forall i \in I . \text{continuous-map } X (T i) (F i)$
and $W = \text{weak-topology } (\text{topspace } X) F T I$
and *funcset-types* (*topspace* X) F T I
shows $X = W$

<proof>

The canonical functions on a product space: evaluation and projection

definition *evaluation-map* :: 'a topology \Rightarrow ('b \Rightarrow 'a \Rightarrow 'c) \Rightarrow 'b set \Rightarrow 'a \Rightarrow 'b \Rightarrow 'c

where *evaluation-map* X F I = ($\lambda x \in \text{topspace } X . (\lambda i \in I . F i x)$)

definition *product-projection* :: ('a \Rightarrow 'b topology) \Rightarrow 'a set \Rightarrow 'a \Rightarrow ('a \Rightarrow 'b) \Rightarrow 'b

where *product-projection* T I = ($\lambda i \in I . (\lambda p \in \text{topspace } (\text{product-topology } T I) . p i)$)

lemma *product-projection*:

shows $\forall i \in I . \forall p \in \text{topspace } (\text{product-topology } T I) . \text{product-projection } T I i p = p i$

<proof>

lemma *evaluation-then-projection*:

assumes $\forall i \in I . F i \in \text{topspace } X \rightarrow \text{topspace } (T i)$

shows $\forall i \in I . \forall x \in \text{topspace } X . ((\text{product-projection } T I i) o (\text{evaluation-map } X F I)) x = F i x$

<proof>

2.3 A product topology is the weak topology induced by its projections if the projections separate points from closed sets.

lemma *projections-continuous*:

assumes $P = \text{product-topology } T I$

and $F = (\lambda i \in I . \text{product-projection } T I i)$

shows $\forall i \in I . \text{continuous-map } P (T i) (F i)$

<proof>

lemma *product-topology-eq-weak-topology*:

assumes $P = \text{product-topology } T I$

and $F = (\lambda i \in I . \text{product-projection } T I i)$

and $W = \text{weak-topology } (\text{topspace } P) F T I$

and *funcset-types* ($\text{topspace } P$) F T I

and *funcset-separates-points-from-closed-sets* P F T I

shows $P = W$

<proof>

Reducing the domain and minimising the range of continuous functions, and related results concerning weak topologies.

lemma *continuous-map-reduced*:

assumes *continuous-map* X Y f

shows *continuous-map* (*subtopology* X S) (*subtopology* Y (f'S)) (*restrict* f S)

<proof>

lemma *inj-on-imp*:
assumes *inj-on f S*
shows $\forall y . (y \in f \text{' } S) \longleftrightarrow (\exists x \in S . y = f x)$
 $\langle \text{proof} \rangle$

lemma *injection-on-intersection*:
assumes *inj-on f S*
and $B \neq \{\}$
and $\forall b \in B . b \subseteq S$
shows $f \text{' } (\bigcap B) = \bigcap \{ f \text{' } b \mid b . b \in B \}$
(is ?lhs = ?rhs)
 $\langle \text{proof} \rangle$

2.4 Evaluation is an embedding for weak topologies

lemma *evaluation-is-embedding*:
assumes $X = \text{weak-topology } (\text{topspace } X) F T I$
and $P = \text{product-topology } T I$
and $\text{funcset-types } (\text{topspace } X) F T I$
and $\text{funcset-separates-points } X F I$
shows $\text{embedding-map } X P (\text{evaluation-map } X F I)$
 $\langle \text{proof} \rangle$

3 Compactification

3.1 Definition

lemma *embedding-map-id*:
assumes $S \subseteq \text{topspace } X$
shows $\text{embedding-map } (\text{subtopology } X S) X \text{ id}$
 $\langle \text{proof} \rangle$

definition *compactification-via* :: $('a \Rightarrow 'b) \Rightarrow 'a \text{ topology} \Rightarrow 'b \text{ topology} \Rightarrow \text{bool}$
where $\text{compactification-via } f X K \equiv \text{compact-space } K \wedge \text{dense-embedding } X K f$

definition *compactification* :: $'a \text{ topology} \Rightarrow 'b \text{ topology} \Rightarrow \text{bool}$
where $\text{compactification } X K = (\exists f . \text{compactification-via } f X K)$

lemma *compactification-compactification-via*:
assumes $\text{compactification-via } f X K$
shows $\text{compactification } X K$
 $\langle \text{proof} \rangle$

3.2 Example: The Alexandroff compactification of a non-compact locally-compact Hausdorff space

lemma *Alexandroff-is-compactification-via-Some:*

assumes \neg compact-space $X \wedge$ Hausdorff-space $X \wedge$ locally-compact-space X

shows compactification-via Some X (Alexandroff-compactification X)

<proof>

3.3 Example: The closure of a subset of a compact space

lemma *compact-closure-is-compactification:*

assumes compact-space K

and $S \subseteq$ topspace K

shows compactification-via id (subtopology K S) (subtopology K (K closure-of S))

<proof>

3.4 Example: A compact space is a compactification of itself

lemma *compactification-of-compact:*

assumes compact-space K

shows compactification-via id K K

<proof>

3.5 Example: A closed non-trivial real interval is a compactification of its interior

lemma *closed-interval-interior:*

shows $\{a::\text{real} <..< b\} =$ interior $\{a..b\}$

<proof>

lemma *open-interval-closure:*

shows $(a < (b::\text{real})) \longrightarrow \{a .. b\} =$ closure $\{a <..< b\}$

<proof>

lemma *closed-interval-compactification:*

assumes $(a::\text{real}) < b$

and open-interval = subtopology euclideanreal $\{a<..<b\}$

and closed-interval = subtopology euclideanreal $\{a..b\}$

shows compactification open-interval closed-interval

<proof>

4 The Stone-Ćech compactification of a Tychonov space

lemma *compact-range':*

assumes $f \in C^* X$

shows compact (range' X f)

<proof>

lemma *c-range-nonempty*:

assumes $f \in C(X)$
and $\text{topspace } X \neq \{\}$
shows $\text{range}' X f \neq \{\}$
<proof>

lemma *cstar-range-nonempty*:

assumes $f \in C^* X$
and $\text{topspace } X \neq \{\}$
shows $\text{range}' X f \neq \{\}$
<proof>

lemma *cstar-separates-tych-space*:

assumes *tych-space* X
shows *funcset-separates-points-from-closed-sets* X (*cstar-id* X) $(\lambda f \in C^* X. \text{euclideanreal})$ $(C^* X)$
 \wedge *funcset-separates-points* X (*cstar-id* X) $(C^* X)$
<proof>

The product topology induced by $C^*(X)$ on a Tychonov space.

definition *scT* :: *'a topology* \Rightarrow (*'a* \Rightarrow *real*) \Rightarrow *real topology*

where *scT* $X = (\lambda f \in C^* X. \text{subtopology euclideanreal } (\text{range}' X f))$

definition *scT-full* :: *'a topology* \Rightarrow (*'a* \Rightarrow *real*) \Rightarrow *real topology*

where *scT-full* $X = (\lambda f \in C^* X. \text{euclideanreal})$

definition *scProduct* :: *'a topology* \Rightarrow ((*'a* \Rightarrow *real*) \Rightarrow *real*) *topology*

where *scProduct* $X = \text{product-topology } (\text{scT } X) (C^* X)$

definition *scProject* :: *'a topology* \Rightarrow (*'a* \Rightarrow *real*) \Rightarrow ((*'a* \Rightarrow *real*) \Rightarrow *real*) \Rightarrow *real*

where *scProject* $X = \text{product-projection } (\text{scT } X) (C^* X)$

definition *scEmbed* :: *'a topology* \Rightarrow *'a* \Rightarrow (*'a* \Rightarrow *real*) \Rightarrow *real*

where *scEmbed* $X = \text{evaluation-map } X (\text{cstar-id } X) (C^* X)$

lemma *scT-images-compact-Hausdorff*:

shows $\forall f \in C^* X. \text{compact-Hausdorff } (\text{scT } X f)$
<proof>

lemma *scT-images-bounded*:

shows $\forall f \in C^* X. \text{bounded } (\text{topspace } (\text{scT } X f))$
<proof>

lemma *scProduct-compact-Hausdorff*:
shows *compact-Hausdorff* (*scProduct* X)
 \langle *proof* \rangle

The Stone-Ćech compactification of a Tychonov space and its extension properties

lemma *tych-space-weak*:
assumes *tych-space* X
shows $X = \text{weak-topology } (\text{topspace } X) (\text{cstar-id } X) (\text{scT } X) (C^* X)$
 \langle *proof* \rangle

4.1 Definition of βX

definition *scEmbeddedCopy* :: $'a \text{ topology} \Rightarrow (('a \Rightarrow \text{real}) \Rightarrow \text{real}) \text{ set}$
where *scEmbeddedCopy* $X = \text{scEmbed } X \text{ ' topspace } X$

definition *scCompactification* :: $'a \text{ topology} \Rightarrow (('a \Rightarrow \text{real}) \Rightarrow \text{real}) \text{ topology } (\langle \beta \text{-} \rangle)$
where *scCompactification* X
 $= \text{subtopology } (\text{scProduct } X) ((\text{scProduct } X) \text{ closure-of } (\text{scEmbeddedCopy } X))$

lemma *sc-topospace*:
shows $\text{topspace } (\beta X) = (\text{scProduct } X) \text{ closure-of } (\text{scEmbeddedCopy } X)$
 \langle *proof* \rangle

lemma *scProject'*:
shows $\forall f \in C^* X . \forall p \in \text{topspace } (\beta X) . \text{scProject } X f p = p f$
 \langle *proof* \rangle

Evaluation densely embeds Tychonov X in βX

lemma *dense-embedding-scEmbed*:
assumes *tych-space* X
shows $\text{dense-embedding } X (\beta X) (\text{scEmbed } X)$
 \langle *proof* \rangle

4.2 βX is a compactification of X

lemma *scCompactification-compact-Hausdorff*:
assumes *tych-space* X
shows $\text{compact-Hausdorff } (\beta X)$
 \langle *proof* \rangle

lemma *scCompactification-is-compactification-via-scEmbed*:
assumes *tych-space* X
shows $\text{compactification-via } (\text{scEmbed } X) X (\beta X)$
 \langle *proof* \rangle

lemma *scCompactification-is-compactification*:

assumes *tych-space* X
shows *compactification* X (βX)
 \langle *proof* \rangle

lemma *scEvaluation-range*:
assumes $x \in \text{topspace } X$
and *tych-space* X
shows $(\lambda f \in C^* X . f x) \in \text{topspace } (\text{product-topology } (\text{scT } X) \ C^* X)$
 \langle *proof* \rangle

lemma *scEmbed-then-project*:
assumes $f \in C^* X$
and $x \in \text{topspace } X$
and *tych-space* X
shows $\text{scProject } X f (\text{scEmbed } X x) = f x$
 \langle *proof* \rangle

4.3 Evaluation is a C^* -embedding of X into βX

definition *scExtend* :: $'a \text{ topology} \Rightarrow ('a \Rightarrow \text{real}) \Rightarrow (('a \Rightarrow \text{real}) \Rightarrow \text{real}) \Rightarrow \text{real}$
where $\text{scExtend } X = (\lambda f \in C^* X . \text{restrict } (\text{scProject } X f) (\text{topspace } (\beta X)))$

proposition *scExtend-extends*:
assumes *tych-space* X
shows $\forall f \in C^* X . \forall x \in \text{topspace } X . f x = (\text{scExtend } X f) (\text{scEmbed } X x)$
 \langle *proof* \rangle

lemma *scExtend-extends-cstar*:
assumes *tych-space* X
shows $\forall f \in C^* X . (\forall x \in \text{topspace } X . f x = (\text{scExtend } X f) (\text{scEmbed } X x)) \wedge \text{scExtend } X f \in C^* (\beta X)$
 \langle *proof* \rangle

lemma *cstar-embedding-scEmbed*:
assumes *tych-space* X
shows *cstar-embedding* X (βX) $(\text{scEmbed } X)$
 \langle *proof* \rangle

A compact Hausdorff space is its own Stone-Cech compactification

lemma *scCompactification-of-compact-Hausdorff*:
assumes *compact-Hausdorff* X
shows *homeomorphic-map* X (βX) $(\text{scEmbed } X)$
 \langle *proof* \rangle

4.4 The Stone-Čech Extension Property: Any continuous map from X to a compact Hausdorff space K extends uniquely to a continuous map from βX to K .

proposition *gof-cstar*:

assumes *compact-Hausdorff* K
and *continuous-map* $X K f$
shows $\forall g \in C^* K . (g \circ f) \in C^* X$
<proof>

proposition *scEmbed-range*:

assumes *tych-space* X
and $x \in \text{topspace } X$
shows $\text{scEmbed } X x \in \text{topspace } (\beta X)$
<proof>

proposition *scEmbed-range'*:

assumes *tych-space* X
and $x \in \text{topspace } X$
shows $\text{scEmbed } X x \in \text{topspace } (\text{scProduct } X)$
<proof>

proposition *scProjection*:

shows $\forall f \in C^* X . \forall p \in \text{topspace } (\text{scProduct } X) . \text{scProject } X f p = p f$
<proof>

proposition *scProjections-continuous*:

shows $\forall f \in C^* X . \text{continuous-map } (\text{scProduct } X) (\text{scT } X f) (\text{scProject } X f)$
<proof>

proposition *continuous-embedding-inverse*:

assumes *embedding-map* $X Y e$
shows $\exists e' . \text{continuous-map } (\text{subtopology } Y (e' \text{ topspace } X)) X e' \wedge (\forall x \in \text{topspace } X . e' (e x) = x)$
<proof>

lemma *scExtension-exists*:

assumes *tych-space* X
and *compact-Hausdorff* K
shows $\forall f \in \text{cts}[X, K] . \exists F \in \text{cts}[\beta X, K] . (\forall x \in \text{topspace } X . F (\text{scEmbed } X x) = f x)$
<proof>

lemma *scExtension-unique*:

assumes $F \in \text{cts}[\beta X, K] \wedge (\forall x \in \text{topspace } X . F (\text{scEmbed } X x) = f x)$

and *compact-Hausdorff* K
shows $(\forall G . G \in \text{cts}[\beta X, K] \wedge (\forall x \in \text{topspace } X . G (\text{scEmbed } X x) = f x)$
 $\longrightarrow (\forall p \in \text{topspace } (\beta X) . F p = G p))$
 $\langle \text{proof} \rangle$

lemma *scExtension-property*:
assumes *tych-space* X
and *compact-Hausdorff* K
shows $\forall f \in \text{cts}[X, K] . \exists ! F \in \text{cts}_E[\beta X, K] . (\forall x \in \text{topspace } X . F (\text{scEmbed } X x) = f x)$
 $\langle \text{proof} \rangle$

end

References

- [Wal74] Russell C. Walker. *The Stone-Čech Compactification*. Springer-Verlag, 1974.
- [Wil70] Stephen Willard. *General Topology*. Addison-Wesley, 1970.