The Stone-Čech Compactification

Mike Stannett

June 5, 2024

Contents

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

definition *arbitrary-unions-of* :: *'a set set* ⇒ *'a set set*
  where *arbitrary-unions-of* $S = \{ (\bigcup F) \mid F .\ F \subseteq S \}$

lemma *generator-imp-arbitrary-union*:
  shows $S \subseteq$ *arbitrary-unions-of* S
  ⟨*proof*⟩

lemma *finite-intersections-container*:
  shows $\forall\ s \in$ *finite-intersections-of* S . $\bigcup S \cap s = s$
  ⟨*proof*⟩

lemma *generator-imp-finite-intersection*:
  shows $S \subseteq$ *finite-intersections-of* S
  ⟨*proof*⟩

lemma *finite-intersections-equiv*:
  shows (*finite' intersection-of* $(\lambda x.\ x \in S)$) $U \longleftrightarrow U \in$ *finite-intersections-of* S
  ⟨*proof*⟩

lemma *arbitrary-unions-equiv*:
  shows (*arbitrary union-of* $(\lambda\ x\ .\ x \in S)$) $U \longleftrightarrow U \in$ *arbitrary-unions-of* S
  ⟨*proof*⟩

Supplementary information about topological bases and the topologies they generate

definition *base-generated-on-by* :: *'a set* ⇒ *'a set set* ⇒ *'a set set*
  where *base-generated-on-by X S* = { *X* ∩ *s* | *s* . *s* ∈ *finite-intersections-of S*}

definition *opens-generated-on-by* :: *'a set* ⇒ *'a set set* ⇒ *'a set set*
  where *opens-generated-on-by X S* = *arbitrary-unions-of* (*base-generated-on-by X S*)

definition *base-generated-by* :: *'a set set* ⇒ *'a set set*
  where *base-generated-by S* = *finite-intersections-of S*

definition *opens-generated-by* :: *'a set set* ⇒ *'a set set*
  where *opens-generated-by S* = *arbitrary-unions-of* (*base-generated-by S*)

lemma *generators-are-basic*:
  shows *S* ⊆ *base-generated-by S*
  ⟨*proof*⟩

lemma *basics-are-open*:
  shows *base-generated-by S* ⊆ *opens-generated-by S*
  ⟨*proof*⟩

lemma *generators-are-open*:
  shows *S* ⊆ *opens-generated-by S*
  ⟨*proof*⟩

lemma *generated-topspace*:
  assumes *T* = *topology-generated-by S*
  shows *topspace T* = ⋃ *S*
  ⟨*proof*⟩

lemma *base-generated-by-alt*:
  shows *base-generated-by S* = *base-generated-on-by* (⋃ *S*) *S*
  ⟨*proof*⟩

lemma *opens-generated-by-alt*:
  shows *opens-generated-by S* = *arbitrary-unions-of* (*finite-intersections-of S*)
  ⟨*proof*⟩

lemma *opens-generated-unfolded*:
  shows *opens-generated-by S* = {⋃ *A* | *A* . *A* ⊆ { ⋂ *B* | *B* . *finite' B* ∧ *B* ⊆ *S*}}
  ⟨*proof*⟩

lemma *opens-eq-generated-topology*:
  shows *openin* (*topology-generated-by S*) *U* ⟷ *U* ∈ *opens-generated-by S*
⟨*proof*⟩

1 C^*-embedding

abbreviation *continuous-from-to*
:: $'a$ *topology* ⇒ $'b$ *topology* ⇒ ($'a$ ⇒ $'b$) *set* (*cts*[-, -])
where *continuous-from-to X Y* ≡ { *f* . *continuous-map X Y f* }

abbreviation *continuous-from-to-extensional*
:: $'a$ *topology* ⇒ $'b$ *topology* ⇒ ($'a$ ⇒ $'b$) *set* (cts_E[-, -])
where *continuous-from-to-extensional X Y* ≡ (*topspace X* $\rightarrow_E$ *topspace Y*) ∩ *cts*[*X*, *Y*]

abbreviation *continuous-maps-from-to-shared-where* ::
$'a$ *topology* ⇒ ($'b$ *topology* ⇒ *bool*) ⇒ ($'a$ ⇒ $'b$) *set* ⇒ *bool* (*cts′-on* - *to′-shared* -)
where *continuous-maps-from-to-shared-where X P*
≡ (λ *fs* . (∃ *Y* . *P Y* ∧ *fs* ⊆ *cts*[*X*,*Y*]))

definition *dense-in* :: $'a$ *topology* ⇒ $'a$ *set* ⇒ $'a$ *set* ⇒ *bool*
where *dense-in T A B* ≡ *T closure-of A* = *B*

lemma *dense-in-closure*:
assumes *dense-in T A B*
shows *dense-in* (*subtopology T B*) *A B*
⟨*proof*⟩
abbreviation *dense-embedding* :: $'a$ *topology* ⇒ $'b$ *topology* ⇒ ($'a$ ⇒ $'b$) ⇒ *bool*
where *dense-embedding small big f* ≡ (*embedding-map small big f*)
∧ *dense-in big* (*f*‘*topspace small*) (*topspace big*)

lemma *continuous-maps-on-dense-subset*:
assumes (*cts-on X to-shared Hausdorff-space*) {*f*,*g*}
and *dense-in X D* (*topspace X*)
and ∀ *x* ∈ *D* . *f x* = *g x*
shows ∀ *x* ∈ *topspace X* . *f x* = *g x*
⟨*proof*⟩

lemma *continuous-map-on-dense-embedding*:
assumes (*cts-on X to-shared Hausdorff-space*) {*f*,*g*}
and *dense-embedding D X e*
and ∀ *d* ∈ *topspace D* . (*f o e*) *d* = (*g o e*) *d*
shows ∀ *x* ∈ *topspace X* . *f x* = *g x*
⟨*proof*⟩

definition *range′* :: $'a$ *topology* ⇒ ($'a$ ⇒ *real*) ⇒ *real set*
where *range′ X f* = *euclideanreal closure-of* (*f* ‘ *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*
  $\langle proof \rangle$

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

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

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

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* $\equiv (\exists\ e\ .\ c\text{-}embedding\ S\ X\ e)$

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

lemma *bounded-range-iff-fbounded*:
  assumes $f \in C\ X$
  shows *bounded* (*f* ‘ *topspace X*) $\longleftrightarrow$ *fbounded f X*
(**is** *?lhs* $\longleftrightarrow$ *?rhs*)
$\langle proof \rangle$

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

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

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

definition *fmax* :: ($'a \Rightarrow real$) $\Rightarrow$ ($'a \Rightarrow real$) $\Rightarrow$ ($'a \Rightarrow real$)
  where *fmax f g* = (λ *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* $\in C\ X$
  ⟨*proof*⟩

lemma *fmax-cts*:
  assumes ($f \in C\ X$) $\wedge$ ($g \in C\ X$)
  shows *fmax f g* $\in C\ X$
  ⟨*proof*⟩

lemma *fmid-cts*:
  assumes ($f \in C\ X$) $\wedge$ ($m \in C\ X$) $\wedge$ ($M \in C\ X$)
  shows *fmid f m M* $\in C\ X$
  ⟨*proof*⟩

lemma *fconst-cts*:
  shows *fconst v* $\in C\ X$
  ⟨*proof*⟩

lemma *fbound-cts*:
  assumes $f \in C\ X$
  shows *fbound f m M* $\in C\ X$
  ⟨*proof*⟩

Bounded and bounding functions

lemma *fconst-bounded*:
  shows *fbounded* (*fconst v*) *X*
  ⟨*proof*⟩

lemma *fmin-bounded-below*:
  assumes *fbounded-below f X* $\wedge$ *fbounded-below g X*
  shows *fbounded-below* (*fmin f g*) *X*
⟨*proof*⟩

lemma *fmax-bounded-above*:
  assumes *fbounded-above f X* ∧ *fbounded-above g X*
  shows *fbounded-above* (*fmax f g*) *X*
⟨*proof*⟩

lemma *fmid-bounded*:
  assumes *fbounded m X* ∧ *fbounded M X*
  shows *fbounded* (*fmid f m M*) *X*
⟨*proof*⟩

lemma *fbound-bounded*:
  shows *fbounded* (*fbound f m M*) *X*
  ⟨*proof*⟩

Members of $C^*(X)$

lemma *fconst-cstar*:
  shows *fconst v* ∈ *C*∗ *X*
  ⟨*proof*⟩

lemma *fbound-cstar*:
  assumes *f* ∈ *C X*
  shows *fbound f m M* ∈ *C*∗ *X*
  ⟨*proof*⟩

lemma *cstar-nonempty*:
  shows {} ≠ *C*∗ *X*
  ⟨*proof*⟩

2 Weak topologies

definition *funcset-types* :: *'a set* ⇒ (*'b* ⇒ *'a* ⇒ *'c*) ⇒ (*'b* ⇒ *'c topology*) ⇒ *'b set* ⇒ *bool*
  where *funcset-types S F T I* = (∀ *i* ∈ *I* . *F i* ∈ *S* → *topspace* (*T i*))

lemma *cstar-types*:
  shows *funcset-types* (*topspace X*) (*cstar-id X*) (λ*f* ∈ *C*∗ *X* . *euclideanreal*) (*C*∗ *X*)
  ⟨*proof*⟩

lemma *cstar-types-restricted*:
  shows *funcset-types* (*topspace X*) (*cstar-id X*)
      (λ*f* ∈ *C*∗ *X*. (*subtopology euclideanreal* (*range' X f*))) (*C*∗ *X*)
⟨*proof*⟩

definition *inverse'* :: (*'a* ⇒ *'b*) ⇒ *'a set* ⇒ *'b set* ⇒ *'a set*
where *inverse' f source target* = { *x* ∈ *source* . *f x* ∈ *target* }

lemma *inverse′-alt*:
  shows *inverse′ f s t* = (*f* −‘ *t*) ∩ *s*
  ⟨*proof*⟩

definition *open-sets-induced-by-func* :: (′*a* ⇒ ′*b*) ⇒ ′*a set* ⇒ ′*b topology* ⇒ ′*a set set*
  where *open-sets-induced-by-func f source T*
    = { (*inverse′ f source V*) | *V* . *openin T V* ∧ *f* ∈ *source* → *topspace T*}

definition *weak-generators* :: ′*a set* ⇒ (′*b* ⇒ ′*a* ⇒ ′*c*) ⇒ (′*b* ⇒ ′*c topology*) ⇒ ′*b set* ⇒ ′*a set set*
  where *weak-generators source funcs tops index*
    = ⋃ { *open-sets-induced-by-func* (*funcs i*) *source* (*tops i*) | *i*. *i* ∈ *index* }

definition *weak-base* :: ′*a set* ⇒ (′*b* ⇒ ′*a* ⇒ ′*c*) ⇒ (′*b* ⇒ ′*c topology*) ⇒ ′*b set* ⇒ ′*a set set*
  where *weak-base source funcs tops index* = *base-generated-by* (*weak-generators source funcs tops index*)

definition *weak-opens* :: ′*a set* ⇒ (′*b* ⇒ ′*a* ⇒ ′*c*) ⇒ (′*b* ⇒ ′*c topology*) ⇒ ′*b set* ⇒ ′*a set set*
  where *weak-opens source funcs tops index* = *opens-generated-by* (*weak-generators source funcs tops index*)

definition *weak-topology* :: ′*a set* ⇒ (′*b* ⇒ ′*a* ⇒ ′*c*) ⇒ (′*b* ⇒ ′*c topology*) ⇒ ′*b set* ⇒ ′*a topology*
  where *weak-topology source funcs tops index*
    = *topology-generated-by* (*weak-generators source funcs tops index*)

lemma *weak-topology-alt*:
  shows *openin* (*weak-topology S F T I*) *U* ⟷ *U* ∈ *weak-opens S F T I*
  ⟨*proof*⟩

lemma *weak-generators-exist-for-each-point-and-axis*:
  assumes *x* ∈ *S*
and *funcset-types S F T I*
and *i* ∈ *I*
and *b* = *inverse′* (*F i*) *S* (*topspace* (*T i*))
and *F i* ∈ *S* → *topspace* (*T i*)
shows *x* ∈ *b* ∧ *b* ∈ *weak-generators S F T I*
⟨*proof*⟩

lemma *weak-generators-topspace*:
  assumes *W* = *weak-topology S F T I*
  shows *topspace W* = ⋃ (*weak-generators S F T I*)
  ⟨*proof*⟩

lemma *weak-topology-topspace*:
assumes $W = \textit{weak-topology}\ S\ F\ T\ I$
and $\textit{funcset-types}\ S\ F\ T\ I$
shows $(I = \{\} \longrightarrow \textit{topspace}\ W = \{\}) \wedge (I \neq \{\} \longrightarrow \textit{topspace}\ W = S)$
⟨*proof*⟩

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

lemma *opens-generate-opens*:
assumes $\forall\ b \in S\ .\ \textit{openin}\ T\ b$
shows $\forall\ U \in \textit{opens-generated-by}\ S\ .\ \textit{openin}\ T\ U$
⟨*proof*⟩

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

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

lemma *funcset-types-on-empty*:
shows $\textit{funcset-types}\ \{\}\ F\ T\ I$
⟨*proof*⟩

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

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

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

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 $U \subseteq$ *topspace euclideanreal*
shows *inverse′ f* (*topspace X*) $U =$ *inverse′ f* (*topspace X*) ($U \cap$ *range′ X f*)
⟨*proof*⟩

lemma *weak-restricted-topology-eq-weak*:
shows *weak-topology* (*topspace X*) (*cstar-id X*) ($\lambda f \in C{*}\ X$. *euclideanreal*) ($C{*}$ X)
  = *weak-topology* (*topspace X*) (*cstar-id X*) ($\lambda f \in C{*}\ X$. *subtopology euclideanreal* (*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*
  $= (\forall\ x \in$ *topspace X* $.\ \forall\ y \in$ *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*
  $= (\forall\ x\ .\ \forall\ A\ .$ *closedin X A* $\wedge\ x \in$ (*topspace X* $-\ A$)
  $\longrightarrow (\exists\ i \in I\ .\ F\ i\ x \notin (T\ i)$ *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 $\forall\ i \in I$. *continuous-map X* $(T\ i)\ (F\ i)$
and $W =$ *weak-topology* (*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* ⇒ (*'b* ⇒ *'a* ⇒ *'c*) ⇒ *'b set* ⇒ *'a* ⇒ *'b* ⇒ *'c*
where *evaluation-map X F I* = (λ *x* ∈ *topspace X* . (λ *i* ∈ *I* . *F i x*))

definition *product-projection* :: (*'a* ⇒ *'b topology*) ⇒ *'a set* ⇒ *'a* ⇒ (*'a* ⇒ *'b*) ⇒ *'b*
where *product-projection T I* = (λ *i* ∈ *I* . (λ *p* ∈ *topspace* (*product-topology T I*) . *p i*))

lemma *product-projection*:
shows ∀ *i* ∈ *I* . ∀ *p* ∈ *topspace* (*product-topology T I*) . *product-projection T I i p* = *p i*
⟨*proof*⟩

lemma *evaluation-then-projection*:
assumes ∀ *i* ∈ *I* . *F i* ∈ *topspace X* → *topspace* (*T i*)
shows ∀ *i* ∈ *I* . ∀ *x* ∈ *topspace X* . ((*product-projection T I i*) *o* (*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* = *product-topology T I*
and *F* = (λ *i* ∈ *I* . *product-projection T I i*)
shows ∀ *i*∈*I*. *continuous-map P* (*T i*) (*F i*)
⟨*proof*⟩

lemma *product-topology-eq-weak-topology*:
assumes *P* = *product-topology T I*
and *F* = (λ *i* ∈ *I* . *product-projection T I i*)
and *W* = *weak-topology* (*topspace P*) *F T I*
and *funcset-types* (*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 $\forall\ y\ .\ (y \in f\ \text{`}\ S) \longleftrightarrow (\exists\ x \in S\ .\ y = f\ x)$
$\langle proof \rangle$

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*)
$\langle proof \rangle$

2.4 Evaluation is an embedding for weak topologies

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

3 Compactification

3.1 Definition

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

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

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

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

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 $\{a$::*real* $<..<$ $b\}$ = *interior* $\{a..b\}$
⟨*proof*⟩

lemma *open-interval-closure*:
shows (a < (b::*real*)) $\longrightarrow$ $\{a$ .. $b\}$ = *closure* $\{a$ <..< $b\}$
⟨*proof*⟩

lemma *closed-interval-compactification*:
assumes (a::*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-Čech 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 *topspace* $X \neq \{\}$
shows *range'* $X\, f \neq \{\}$
⟨*proof*⟩

lemma *cstar-range-nonempty*:
assumes $f \in C{*}\ X$
and *topspace* $X \neq \{\}$
shows *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) ($\lambda f \in C{*}\ X$. *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* ⇒ (*'a* ⇒ *real*) ⇒ *real topology*
where *scT* X = ($\lambda\ f \in C{*}\ X$. *subtopology euclideanreal* (*range'* $X\ f$))

definition *scT-full* :: *'a topology* ⇒ (*'a* ⇒ *real*) ⇒ *real topology*
where *scT-full* X = ($\lambda\ f \in C{*}\ X$. *euclideanreal*)

definition *scProduct* :: *'a topology* ⇒ ((*'a* ⇒ *real*) ⇒ *real*) *topology*
where *scProduct* X = *product-topology* (*scT* X) ($C{*}\ X$)

definition *scProject* :: *'a topology* ⇒ (*'a* ⇒ *real*) ⇒ ((*'a* ⇒ *real*) ⇒ *real*) ⇒ *real*
where *scProject* X = *product-projection* (*scT* X) ($C{*}\ X$)

definition *scEmbed* :: *'a topology* ⇒ *'a* ⇒ (*'a* ⇒ *real*) ⇒ *real*
where *scEmbed* X = *evaluation-map* X (*cstar-id* X) ($C{*}\ X$)

lemma *scT-images-compact-Hausdorff*:
shows $\forall\ f \in C{*}\ X$. *compact-Hausdorff* (*scT* $X\ f$)
⟨*proof*⟩

lemma *scT-images-bounded*:
shows $\forall\ f \in C{*}\ X$. *bounded* (*topspace* (*scT* $X\ f$))
⟨*proof*⟩

lemma *scProduct-compact-Hausdorff*:
shows *compact-Hausdorff* (*scProduct X*)
⟨*proof*⟩

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

lemma *tych-space-weak*:
assumes *tych-space X*
shows X = *weak-topology* (*topspace X*) (*cstar-id X*) (*scT X*) ($C*$ X)
⟨*proof*⟩

4.1 Definition of βX

definition *scEmbeddedCopy* :: *'a topology* ⇒ ((*'a* ⇒ *real*) ⇒ *real*) *set*
where *scEmbeddedCopy X* = *scEmbed X* ' *topspace X*

definition *scCompactification* :: *'a topology* ⇒ ((*'a* ⇒ *real*) ⇒ *real*) *topology* (β -)
where *scCompactification X*
= *subtopology* (*scProduct X*) ((*scProduct X*) *closure-of* (*scEmbeddedCopy X*))

lemma *sc-topspace*:
shows *topspace* (β X) = (*scProduct X*) *closure-of* (*scEmbeddedCopy X*)
⟨*proof*⟩

lemma *scProject'*:
shows ∀ *f* ∈ $C*$ X . ∀ *p* ∈ *topspace* (β X) . *scProject X f p* = *p f*
⟨*proof*⟩

Evaluation densely embeds Tychonov X in βX

lemma *dense-embedding-scEmbed*:
assumes *tych-space X*
shows *dense-embedding X* (β X) (*scEmbed X*)
⟨*proof*⟩

4.2 βX is a compactification of X

lemma *scCompactification-compact-Hausdorff*:
assumes *tych-space X*
shows *compact-Hausdorff* (β X)
⟨*proof*⟩

lemma *scCompactification-is-compactification-via-scEmbed*:
assumes *tych-space X*
shows *compactification-via* (*scEmbed X*) X (β X)
⟨*proof*⟩

lemma *scCompactification-is-compactification*:
assumes *tych-space X*

shows *compactification X* $(\beta\ X)$
⟨*proof*⟩

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

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

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

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

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

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

lemma *cstar-embedding-scEmbed*:
assumes *tych-space X*
shows *cstar-embedding X* $(\beta\ X)$ (*scEmbed X*)
⟨*proof*⟩

A compact Hausdorff space is its own Stone-Cech compactification

lemma *scCompactification-of-compact-Hausdorff*:
assumes *compact-Hausdorff X*
shows *homeomorphic-map X* $(\beta\ X)$ (*scEmbed X*)
⟨*proof*⟩

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 $\forall\ g \in C{*}\ K\ .\ (g\ o\ f) \in C{*}\ X$
⟨*proof*⟩

proposition *scEmbed-range*:
assumes *tych-space X*
and $x \in$ *topspace X*
shows *scEmbed X x* $\in$ *topspace* $(\beta\ X)$
⟨*proof*⟩

proposition *scEmbed-range'*:
assumes *tych-space X*
and $x \in$ *topspace X*
shows *scEmbed X x* $\in$ *topspace* (*scProduct X*)
⟨*proof*⟩

proposition *scProjection*:
shows $\forall\ f \in C{*}\ X.\ \forall\ p \in$ *topspace* (*scProduct X*) . *scProject X f p* $=$ *p f*
⟨*proof*⟩

proposition *scProjections-continuous*:
shows $\forall\ f \in C{*}\ X$. *continuous-map* (*scProduct X*) (*scT X f*) (*scProject X f*)
⟨*proof*⟩

proposition *continuous-embedding-inverse*:
assumes *embedding-map X Y e*
shows $\exists\ e'$. *continuous-map* (*subtopology Y* (*e* ‘ *topspace X*)) *X* $e' \wedge (\forall\ x \in$ *topspace X* . $e'\ (e\ x) = x)$
⟨*proof*⟩

lemma *scExtension-exists*:
assumes *tych-space X*
and *compact-Hausdorff K*
shows $\forall\ f \in cts[X,K]\ .\ \exists\ F \in cts[\beta\ X,\ K]\ .\ (\forall\ x \in$ *topspace X* . *F* (*scEmbed X x*) $=$ *f x*)
⟨*proof*⟩

lemma *scExtension-unique*:
assumes $F \in cts[\beta\ X,\ K] \wedge (\forall\ x \in$ *topspace X* . *F* (*scEmbed X x*) $=$ *f x*)
and *compact-Hausdorff K*
shows $(\forall\ G\ .\ G \in cts[\beta\ X,\ K] \wedge (\forall\ x \in$ *topspace X* . *G* (*scEmbed X x*) $=$ *f x*)
$\longrightarrow (\forall\ p \in$ *topspace* $(\beta\ X)\ .\ F\ p = G\ p))$
⟨*proof*⟩

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

end

References

[Wal74] Russell C. Walker. *The Stone-Čech Compactification*. Springer-Verlag, 1974.

[Wil70] Stephen Willard. *General Topology*. Addison-Wesley, 1970.