The Kuratowski Closure-Complement Theorem

Peter Gammie and Gianpaolo Gioiosa

August 16, 2018

Contents

1 Introduction 1

2 Interiors and unions 1

3 Additional facts about the rationals and reals 3

4 Kuratowski’s result 3

5 A corollary of Kuratowski’s result 7

6 Chagrov’s result 8

   6.1 Discrete spaces 10
   6.2 Partition spaces 10
   6.3 Extremally disconnected and open unresolvable spaces 12
   6.4 Extremally disconnected spaces 15
   6.5 Open unresolvable spaces 17
   6.6 Kuratowski spaces 18
   6.7 Chagrov’s theorem 20

References 20

1 Introduction

We discuss a topological curiosity discovered by Kuratowski (1922): the fact that the number of distinct operators on a topological space generated by compositions of closure and complement never exceeds 14, and is exactly 14 in the case of \( \mathbb{R} \). In addition, we prove a theorem due to Chagrov (1982) that classifies topological spaces according to the number of such operators they support.

Kuratowski’s result, which is exposited in Whitty (2015) and Chapter 7 of Chamberland (2015), has already been treated in Mizar — see Bagińska and Grabowski (2003) and Grabowski (2004). To the best of our knowledge, we are the first to mechanize Chagrov’s result.

Our work is based on a presentation of Kuratowski’s and Chagrov’s results by Gardner and Jackson (2008). We begin with some preliminary facts pertaining to the relationship between interiors of unions and unions of interiors (§2) and the relationship between \( \mathbb{Q} \) and \( \mathbb{R} \) (§3). We then prove Kuratowski’s result (§4) and the corollary that at most 7 distinct operators on a topological space can be generated by compositions of closure and interior (§5). Finally, we prove Chagrov’s result (§6).

2 Interiors and unions

definition

\[
\text{boundary} :: 'a::topological\_space \Rightarrow 'a \text{ set}
\]

where

\[
\text{boundary} \ X = \text{closure} \ X - \text{interior} \ X
\]
lemma boundary_empty:
  shows boundary {} = {}

unfolding boundary_def by simp

definition
  exterior :: 'a::topological_space set ⇒ 'a set
where
  exterior X = − (interior X ∪ boundary X)

lemma interior_union_boundary:
  shows interior (X ∪ Y) = interior X ∪ interior Y
        ⇐⇒ boundary X ∩ boundary Y ⊆ boundary (X ∪ Y) (is (?lhs1 = ?lhs2) ⇐⇒ ?rhs)
proof
(rule iffI[OF subset_antisym[OF subsetI]])
assume ?lhs1 = ?lhs2 then show ?rhs by (force simp: boundary_def)
next
fix x
assume ?rhs and x ∈ ?lhs1
have x ∈ ?lhs2 if x /∈ interior X x /∈ interior Y
proof (cases x ∈ boundary X ∩ boundary Y)
case True with ⟨?rhs⟩ ⟨x ∈ ?lhs1⟩ show ?thesis by (simp add: boundary_def subset_iff)
next
case False then consider (X) x /∈ boundary X | (Y) x /∈ boundary Y by blast
then show ?thesis
proof cases
  case X
  from X ⟨x /∈ interior X⟩ have x ∈ exterior X by (simp add: exterior_def)
  from ⟨x /∈ boundary X⟩ ⟨x ∈ exterior X⟩ ⟨x /∈ interior X⟩
  obtain U where open U U ⊆ − X x ∈ U
  by (metis ComplI DiffI boundary_def closure_interior interior_subset open_interior)
  from ⟨x ∈ interior (X ∪ Y)⟩ obtain U′ where open U′ U′ ⊆ X ∪ Y x ∈ U′
  by (meson interiorE)
  from ⟨U ⊆ − X⟩ ⟨U′ ⊆ X ∪ Y⟩ have U ∩ U′ ⊆ Y by blast
  with ⟨x /∈ interior Y⟩ (open U′) ⟨open U⟩ ⟨x ∈ U′⟩ ⟨x ∈ U⟩ show ?thesis
  by (meson IntI interiorI open_Int)
next
  case Y
  from Y ⟨x /∈ interior Y⟩ have x ∈ exterior Y by (simp add: exterior_def)
  from ⟨x /∈ boundary Y⟩ ⟨x ∈ exterior Y⟩ ⟨x /∈ interior Y⟩
  obtain U where open U U ⊆ − Y x ∈ U
  by (metis ComplI DiffI boundary_def closure_interior interior_subset open_interior)
  from ⟨x ∈ interior (X ∪ Y)⟩ obtain U′ where open U′ U′ ⊆ X ∪ Y x ∈ U′
  by (meson interiorE)
  from ⟨U ⊆ − Y⟩ ⟨U′ ⊆ X ∪ Y⟩ have U ∩ U′ ⊆ X by blast
  with ⟨x /∈ interior X⟩ (open U′) ⟨open U⟩ ⟨x ∈ U′⟩ ⟨x ∈ U⟩ show ?thesis
  by (meson IntI interiorI open_Int)
qed

lemma interior_union_closed_intervals:
  fixes a :: 'a::ordered_euclidean_space
  assumes b < c
  shows interior {(a..b) ∪ {c..d}} = interior {a..b} ∪ interior {c..d}
using assms by (subst interior_union_boundary; auto simp: boundary_def)
3 Additional facts about the rationals and reals

lemma Rat_real_limpt:
  fixes x :: real
  shows x islimpt Q
proof (rule islimptI)
  fix T assume x ∈ T open T
  then obtain e where 0 < e and ball: |x' - x| < e → x' ∈ T for x' by (auto simp: open_real)
  from (0 < e) obtain q where x < real_of_rat q ∧ real_of_rat q < x + e using of_rat_dense by force
  with ball show ∃y ∈ Q. y ∈ T ∧ y ≠ x by force
qed

lemma Rat_closure:
  shows closure Q = (UNIV :: real set)
unfolding closure_def using Rat_real_limpt by blast

lemma Rat_interval_closure:
  fixes x :: real
  assumes x < y
  shows closure ({x..<y} ∩ Q) = {x..y}
using assms by (metis (no_types, lifting) Rat_closure closure_closure closure_greaterThanLessThan closure_mono inf_left1 inf_top.right_neq open_Int_closure_subset open_real_greaterThanLessThan subset antisym)

lemma Rat_not_open:
  fixes T :: real set
  assumes x < y
  assumes T ≠ {}
  shows ¬T ⊆ Q
using assms by (simp add: countable_rat open_minus_countable_subset subset_eq)

lemma Irrat_dense_in_real:
  fixes x :: real
  assumes x < y
  shows ∃r ∈ −Q. x < r ∧ r < y
using assms Rat_not_open[where T={x..<y}] by force

lemma closed_interval_Int_compl:
  fixes x :: real
  assumes x < y
  assumes y < z
  shows −{x..y} ∩ −{y..z} = − {x..z}
using assms by auto

4 Kuratowski’s result

We prove that at most 14 distinct operators can be generated by compositions of closure and complement. For convenience, we give these operators short names and try to avoid pointwise reasoning. We treat the interior operator at the same time.

declare o_apply[simp del]

definition C :: 'a::topological_space set ⇒ 'a set where C X = − X

definition K :: 'a::topological_space set ⇒ 'a set where K X = closure X

definition I :: 'a::topological_space set ⇒ 'a set where I X = interior X
The inductive set $CK$ captures all operators that can be generated by compositions of $C$ and $K$. We shallowly embed the operators; that is, we identify operators up to extensional equality.

**inductive** $CK :: ('a::topological_space set) \Rightarrow 'a set \Rightarrow bool$
We show, using the following subset of $\mathbb{IR}$ (an example taken from Rusin (2001)) as a witness, that there exist the inductive set $CK_nf$ captures the normal forms for the 14 distinct operators.

The inductive set $CK_nf$ captures the normal forms for the 14 distinct operators.

declare $CK_nf.intros[intro!]

lemma $CK_nf.intros[intro!]:$

by (metis $CK_nf.intros(1)$ $CK_nf.intros(3)$ $C.C$)

That each operator generated by compositions of $C$ and $K$ is extensionally equivalent to one of the normal forms captured by $CK_nf$ is demonstrated by means of an induction over the construction of $CK_nf$ and an appeal to the facts proved above.

theorem $CK_nf$: $CK_f \leftrightarrow CK_nf f$

proof(rule iffI)

assume $CK_f$ then show $CK_nf f$

by induct

(also $CK_nf_cases$; clarsimp simp: id_def[symmetric] $C.C$ $K.K$ $CKKCKKCK_KCKKCKK$ o_assoc; simp add: o_assoc[symmetric]; clarsimp simp: $C.C$ $K.K$ $CKKCKKCKKCKK$ o_assoc

| blast)+

next

assume $CK_nf f$ then show $CK_f$ by induct (auto simp: id_def[symmetric])

qed

theorem $CK_card$: $\text{card}\{f.\ CK_f\} \leq 14$

by (auto simp: $CK_nf$ $CK_nf_set$ card_insert intro!: le_trans[OF card_Diff1_le])

We show, using the following subset of $\mathbb{IR}$ (an example taken from Rusin (2001)) as a witness, that there exist topological spaces on which all 14 operators are distinct.

definition $RRR :: \text{real set}$
where
\[ RRR = \{0,<..1\} \cup \{1,<..2\} \cup \{3\} \cup \{(5,<..7) \cap \mathbb{Q}\} \]

The following facts allow the required proofs to proceed by simp:

**lemma RRR_closure:**
- shows closure RRR = \{0..2\} \cup \{3\} \cup \{5..7\}

**unfolding RRR_def by (force simp: closure_insert Rat_interval_closure)**

**lemma RRR_interior:**
- interior RRR = \{0,<..1\} \cup \{1,<..2\} (is \ ?lhs = \ ?rhs)

**proof (rule equalityI[OF subsetI subsetI])**
- fix x assume x \in \ ?lhs
- then obtain T where open T and x \in T and T \subseteq RRR by (blast elim: interiorE)
- then obtain e where 0 < e and ball x e \subseteq T by (blast elim!: openE)
- from (x \in T) (0 < e) (ball x e \subseteq T) (T \subseteq RRR) have False if x = 3
  - using that unfolding RRR_def ball_def
    - by (auto dest!: subsetD[where c=\min (3 + e/2) 4] simp: dist_real_def)

moreover
from Irrat_dense_in_real[where x=x and y=x + e/2] (0 < e)
obtain r where r \in \mathbb{Q} \land x < r \land r < x + e / 2 by auto
with (x \in T) (ball x e \subseteq T) (T \subseteq RRR) have False if x \in \{5,<..7\} \cap \mathbb{Q}
  - using that unfolding RRR_def ball_def
    - by (force simp: dist_real_def dest: subsetD[where c=r])

moreover note (x \in interior RRR)
ultimately show x \in \ ?rhs
  - unfolding RRR_def by (auto dest: subsetD[OF interior_subset])

next
- fix x assume x \in \ ?rhs
- then show x \in \ ?lhs
  - unfolding RRR_def interior_def by (auto intro: open_real_greaterThanLessThan)

qed

**lemma RRR_interior_closure[simplified]:**
- shows interior (\{0::real..2\} \cup \{3\} \cup \{5..7\}) = \{0,<..2\} \cup \{5..<7\} (is \ ?lhs = \ ?rhs)

**proof –**
- have \ ?lhs = interior (\{0..2\} \cup \{5..7\})
  - by (metis (no_types, lifting) Un_assoc Un_commute closed_Un closed_eucl_atLeastAtMost interior_closed_Un_empty_interior_simple)
- also have ... = \ ?rhs
  - by (simp add: interior_union_closed_intervals)
- finally show \ ?thesis .

qed

The operators can be distinguished by testing which of the points in \{1,2,3,4,6\} belong to their results.

**definition**
\[ test :: (real set \Rightarrow real set) \Rightarrow bool \text{ list} \]

**where**
\[ test f \equiv map (\lambda x. \ x \in f \ \text{RRR}) \ [1,2,3,4,6] \]

**lemma RRR_test:**
- assumes f RRR = g RRR
- shows test f = test g

**unfolding test_def using assms by simp**

**lemma nf_RRR:**
- shows
test id = [False, False, True, False, True]
test C = [True, True, False, True, False]
test K = [True, True, True, False, True]
test \((K \circ C)\) = [True, True, True, True, True]
test \((C \circ K)\) = [False, False, False, False, False]
test \((C \circ K \circ C)\) = [False, False, False, False, False]
test \((K \circ C \circ K)\) = [True, False, False, True, False]
test \((C \circ K \circ C \circ K)\) = [False, False, False, False, False]
test \((K \circ C \circ K \circ C)\) = [True, True, True, True, True]
test \((K \circ C \circ K \circ C \circ K)\) = [False, False, False, False, False]
test \((C \circ K \circ C \circ K \circ C \circ K)\) = [True, False, False, True, False]

unfolding test_def C_def K_def

by (simp_all add: RRR_closure RRR_interior RRR_interior_closure closure_complement closed_interval_Int_compl o_apply)

(simp_all add: RRR_def)

theorem CK_nf_real_card:
  shows card \((\lambda f . f \ RRR) \cdot \{f . \ CK\_nf\ f\}\) = 14
by (simp add: CK_nf_set) ((subst card_insert_disjoint; auto dest!: RRR_test simp: nf_RRR_id_def[symmetric])[I])

theorem CK_real_card:
  shows card \(\{f::\text{real set} \Rightarrow \text{real set}. \ CK\ f\}\) = 14 (is ?lhs = ?rhs)
proof (rule antisym[OF CK_card])
  show ?lhs \leq ?rhs
  unfolding CK_nf
  by (rule le_trans[OF eq_imp_le[OF CK_nf_real_card[symmetric] card_image_le]])

qed

5 A corollary of Kuratowski’s result

We show that it is a corollary of CK_real_card that at most 7 distinct operators on a topological space can be generated by compositions of closure and interior. In the case of \(\mathbb{R}\), exactly 7 distinct operators can be so generated.

inductive IK :: ('a::topological_space_set \Rightarrow 'a set) \Rightarrow bool where
  IK_id 
| IK I 
| IK K 
| \[ IK f; IK g \] \Rightarrow IK (f \circ g)

inductive IK_nf :: ('a::topological_space_set \Rightarrow 'a set) \Rightarrow bool where
  IK_nf_id 
| IK_nf I 
| IK_nf K 
| IK_nf (I \circ K) 
| IK_nf (K \circ I) 
| IK_nf (I \circ K \circ I) 
| IK_nf (K \circ I \circ K)

declare IK.intros[intro!]

declare IK_nf.intros[intro!]

lemma IK_nf_set:
  \(\{f . \ IK\_nf\ f\}\) = \{id, I, K, I \circ K, K \circ I, I \circ K \circ I, K \circ I \circ K\}
by (auto simp: IK_nf.simps)

theorem IK_nf:
  IK f ↔ IK_nf f
proof (rule iffI)
  assume IK f then show IK_nf f
    by (induct elim: IK_nf_cases; clarsimp simp: id_def[symmetric] o_assoc; simp add: I_I K_K o_assoc[symmetric]; clarsimp simp: K_I K_I I_K_I o_assoc)
  next
  assume IK_nf f then show IK f by (induct blast)+
qed

theorem IK_card:
  shows card {f. IK f} ≤ 7
by (auto simp: IK_nf IK_nf_set card_insert intro: le_trans[OF card_Diff1_le])

theorem IK_real_card:
  shows card ((λ f. f RRR ' {f . IK_nf f})) = 7
by (simp add: IK_nf_set) ((subst card_insert_disjoint; auto dest!: RRR_test simp: nf_RRR I_K id_def[symmetric] o_assoc)[1]+)

definition CK_nf_pos :: ('a::topological_space set ⇒ 'a set) ⇒ bool where
  CK_nf_pos id |
  CK_nf_pos I |
  CK_nf_pos K |
  CK_nf_pos (I o K) |
  CK_nf_pos (K o I) |
  CK_nf_pos (I o K o I) |
  CK_nf_pos (K o I o K)

declare CK_nf_pos.intros[intro]

lemma CK_nf_pos_set:
  shows {f . CK_nf_pos f} = {id, I, K, I o K, K o I, I o K o I, K o I o K}
by (auto simp: CK_nf_pos.simps)

definition CK_nf_neg :: ('a::topological_space set ⇒ 'a set) ⇒ bool where

6 Chagrov’s result

Chagrov’s theorem, which is discussed in Section 2.1 of Gardner and Jackson (2008), states that the number of distinct operators on a topological space that can be generated by compositions of closure and complement is one of 2, 6, 8, 10 or 14.

We begin by observing that the set of normal forms CK_nf can be split into two disjoint sets, CK_nf_pos and CK_nf_neg, which we define in terms of interior and closure.
lemma \( CK_{\text{nf\_pos\_neg\_disjoint}} \):

\[
\begin{align*}
\text{assumes } CK_{\text{nf\_pos}} f \\
\text{assumes } CK_{\text{nf\_neg}} g \\
\text{shows } f \neq g 
\end{align*}
\]

using assms unfolding \( CK_{\text{nf\_neg\_def}} \)
by (clarsimp simp: \( CK_{\text{nf\_pos\_simps}} \); elim disjE; metis comp_def I_def K_def Compl_iff closure_UNIV interior_UNIV id_apply)

lemma \( CK_{\text{nf\_pos\_neg\_CK\_nf}} \):

\[
CK_{\text{nf}} f \longleftrightarrow CK_{\text{nf\_pos}} f \lor CK_{\text{nf\_neg}} f 
\]

proof (rule iffI)

assume \( \text{?lhs} \) then show \( \text{?rhs} \)
unfolding \( CK_{\text{nf\_neg\_def}} \)
by (rule \( CK_{\text{nf\_cases}} \); metis (no_types, lifting) \( CK_{\text{nf\_pos\_simps}} \) C C I K I comp_id o_assoc)

next

assume \( \text{?rhs} \) then show \( \text{?lhs} \)
unfolding \( CK_{\text{nf\_neg\_def}} \)
by (auto elim!: \( CK_{\text{nf\_pos\_cases}} \) simp: I K C C o_assoc)

qed

We now focus on \( CK_{\text{nf\_pos}} \). In particular, we show that its cardinality for any given topological space is one of 1, 3, 4, 5 or 7.

The proof consists of exhibiting normal forms for the operators supported by each of six classes of topological spaces. These are sublattices of the following lattice of \( CK_{\text{nf\_pos}} \) operators:

\[
\begin{align*}
I \leq (id :: 'a::topological\_space \Rightarrow 'a set) \\
\text{id} \leq (K :: 'a::topological\_space \Rightarrow 'a set) \\
I \leq I \circ K \circ I (I :: 'a::topological\_space \Rightarrow 'a set) \\
I \circ K \circ I \leq I \circ (K :: 'a::topological\_space \Rightarrow 'a set) \\
I \circ K \circ I \leq K \circ (I :: 'a::topological\_space \Rightarrow 'a set) \\
I \circ K \leq K \circ I (K :: 'a::topological\_space \Rightarrow 'a set) \\
K \circ I \leq K \circ (I :: 'a::topological\_space \Rightarrow 'a set) \\
K \circ I \circ K \leq (K :: 'a::topological\_space \Rightarrow 'a set) 
\end{align*}
\]

unfolding I_def K_def
by (simp_all add: interior_subset closure_subset interior_maximal closure_mono o_apply interior_mono K_I_K_subseteq_K le_fun)

We define the six classes of topological spaces in question, and show that they are related by inclusion in the following way (as shown in Figure 2.3 of Gardner and Jackson (2008)):
6.1 Discrete spaces

**Definition**
\[
\text{discrete } (X : 'a::topological\_space\ set) \leftrightarrow I = (id : 'a set \Rightarrow 'a set)
\]

**Lemma discrete_eqs**:
- **Assumes** \(\text{discrete } (X : 'a::topological\_space\ set)\)
- **Shows**
  \[I = (id : 'a set \Rightarrow 'a set)
  \]
  \[K = (id : 'a set \Rightarrow 'a set)
  \]
- **Using** \(\text{assms unfolding discrete\_def by (auto simp: C\_C K\_I)}\)

**Lemma discrete_card**:
- **Fixes** \(X : 'a::topological\_space\ set\)
- **Assumes** \(\forall Y : 'a set. \text{open } Y\)
- **Shows** \(\text{discrete } X\)
- **Using** \(\text{assms unfolding discrete\_def I\_def interior\_def islimpt\_def by (auto simp: fun\_eq\_iff)}\)

6.2 Partition spaces

**Definition**
\[
\text{part } (X : 'a::topological\_space\ set) \leftrightarrow K \circ I = (I : 'a set \Rightarrow 'a set)
\]

**Lemma discrete_part**:
- **Assumes** \(\text{discrete } X\)
- **Shows** \(\text{part } X\)
- **Using** \(\text{assms unfolding discrete\_def part\_def by (simp add: C\_C K\_I)}\)

**Lemma part_eqs**:
- **Assumes** \(\text{part } (X : 'a::topological\_space\ set)\)
- **Shows**
  \[K \circ I = (I : 'a set \Rightarrow 'a set)
  \]
  \[I \circ K = (K : 'a set \Rightarrow 'a set)
  \]
- **Using** \(\text{assms unfolding part\_def by (assumption, metis (no_types, hide_lams) I\_I K\_I o\_assoc)}\)
A partition space is a topological space whose basis consists of the empty set and the equivalence classes of points of the space induced by some equivalence relation \( R \) on the underlying set of the space. Equivalently, a partition space is one in which every open set is closed. Thus, for example, the class of partition spaces includes every topological space whose open sets form a boolean algebra.

**datatype** part_witness = a | b | c

**lemma** part_witness_UNIV:
  shows \( \text{UNIV} = \text{set} \{a, b, c\} \)
  using part_witness.exhaust by auto

**lemmas** part_witness_pow = subset_subseqs[OF subset_trans[OF subset_UNIV Set.equalityD1[OF part_witness_UNIV]]]

**lemmas** part_witness_Compl = Compl_eq_Diff_UNIV[where 'a=part_witness, unfolded part_witness_UNIV, simplified]

**instantiation** part_witness :: topological_space

**begin**

**definition** open_part_witness X \( \leftrightarrow \) \( X \in \{\{\}, \{a\}, \{b, c\}, \{a, b, c\}\} \)

**lemma** part_witness_ball:
  \((\forall s \in S. s \in \{\{\}, \{a\}, \{b, c\}, \{a, b, c\}\}) \leftrightarrow S \subseteq \text{set} \{\{\}, \{a\}, \{b, c\}, \{a, b, c\}\}\)
  by simp blast

**lemmas** part_witness_subsets_pow = subset_subseqs[OF iffD1[OF part_witness_ball]]

**instance** proof standard
  fix K :: part_witness set set
  assume \( \forall S \in K. \text{open} S \text{ then show open } (\bigcup K) \)
  unfolding open_part_witness_def
  by (drule part_witness_subsets_pow; clarsimp; elim disjE; simp add: insert_commute)

**qed** (auto simp: open_part_witness_def part_witness_UNIV)

**end**

**lemma** part_witness_interior_sims:
  shows
  \( \text{interior} \{a\} = \{a\} \)
  \( \text{interior} \{b\} = \{\} \)
  \( \text{interior} \{c\} = \{\} \)
  \( \text{interior} \{a, b\} = \{a\} \)
  \( \text{interior} \{a, c\} = \{a\} \)
  \( \text{interior} \{b, c\} = \{b, c\} \)
  \( \text{interior} \{a, b, c\} = \{a, b, c\} \)
  unfolding interior_def open_part_witness_def by auto

**lemma** part_witness_part:
  fixes X :: part_witness set
  shows part X
  proof —
have closure (interior Y) = interior Y for Y :: part_witness set
using part_witness_pow[where X=Y]
by (auto simp: closure_interior part_witness_interior_simps part_witness_Compl insert_Diff_if)
then show ?thesis
  unfolding part_def I_def K_def by (simp add: o_def)
qed

lemma part_witness_not_discrete:
  fixes X :: part_witness set
  shows ¬discrete X
unfolding discrete_def I_def
by (clarsimp simp: o_apply fun_eq_iff exI[where x={b}] part_witness_interior_simps)

lemma part_witness_card:
  shows card {f. CK_nf_pos (f::part_witness set ⇒ part_witness set)} = 3
by (rule part_not_discrete_card[OF part_witness_part part_witness_not_discrete])

6.3 Extremally disconnected and open unresolvable spaces

definition
ed_ou (X :: 'a::topological_space set) ⌦→ I o K o (I :: 'a set ⇒ 'a set)

lemma discrete_ed_ou:
  assumes discrete X
  shows ed_ou X
using assms unfolding discrete_def ed_ou_def by simp

lemma ed_ou_eqs:
  assumes ed_ou (X :: 'a::topological_space set)
  shows I o K o I = K o (I :: 'a set ⇒ 'a set)
       K o I o K = K o (I :: 'a set ⇒ 'a set)
       I o K = K o (I :: 'a set ⇒ 'a set)
using assms unfolding ed_ou_def by (metis I I K K o_assoc)+

lemma ed_ou_neqs:
  assumes ed_ou (X :: 'a::topological_space set)
  assumes ¬discrete X
  shows I ≠ (K :: 'a set ⇒ 'a set)
       I ≠ K o (I :: 'a set ⇒ 'a set)
       K ≠ K o (I :: 'a set ⇒ 'a set)
       I ≠ (id :: 'a set ⇒ 'a set)
       K ≠ (id :: 'a set ⇒ 'a set)
using assms CK_nf_pos_lattice[where 'a='a]
unfolding ed_ou_def discrete_def
by (metis (no_types, lifting) C.C I.K K.I comp_id o_assoc antisym)+

lemma ed_ou_not_discrete_card:
  assumes ed_ou (X :: 'a::topological_space set)
  assumes ¬discrete X
  shows card {f. CK_nf_pos (f::'a set ⇒ 'a set)} = 4
using ed_ou_eqs[OF (ed_ou X)] ed_ou_neqs[OF assms]
by (subst CK_nf_pos_set) (subst card_insert_disjoint; (auto)[1])+

We consider an example extremally disconnected and open unresolvable topological space.

datatype ed_ou_witness = a | b | c | d | e
lemma ed_ou_witness_UNIV:
  shows UNIV = set [a, b, c, d, e]
using ed_ou_witness.exhaust by auto

lemmas ed_ou_witness_pow = subset_subseqs[OF subset_trans[OF subset_UNIV Set.equalityD1[OF ed_ou_witness_UNIV]]]

lemmas ed_ou_witness_Compl = Compl_eq_Diff_UNIV[where 'a=ed_ou_witness, unfolded ed_ou_witness_UNIV, simplified]

instance ed_ou_witness :: finite
  by standard (simp add: ed_ou_witness_UNIV)

instantiation ed_ou_witness :: topological_space
begin

inductive open_ed_ou_witness :: ed_ou_witness set ⇒ bool where
  open_ed_ou_witness {}
| open_ed_ou_witness {a}
| open_ed_ou_witness {b}
| open_ed_ou_witness {c}
| open_ed_ou_witness {d}
| open_ed_ou_witness {e}
| open_ed_ou_witness {a, b}
| open_ed_ou_witness {a, c}
| open_ed_ou_witness {a, d}
| open_ed_ou_witness {a, e}
| open_ed_ou_witness {b, c}
| open_ed_ou_witness {b, d}
| open_ed_ou_witness {b, e}
| open_ed_ou_witness {c, d}
| open_ed_ou_witness {c, e}
| open_ed_ou_witness {d, e}
| open_ed_ou_witness {a, b, c}
| open_ed_ou_witness {a, b, d}
| open_ed_ou_witness {a, b, e}
| open_ed_ou_witness {a, c, d}
| open_ed_ou_witness {a, c, e}
| open_ed_ou_witness {a, d, e}
| open_ed_ou_witness {b, c, d}
| open_ed_ou_witness {b, c, e}
| open_ed_ou_witness {b, d, e}
| open_ed_ou_witness {c, d, e}

declare open_ed_ou_witness.intros[intro!]

lemma ed_ou_witness_inter:
  fixes S :: ed_ou_witness set
  assumes open S
  assumes open T
  shows open (S ∩ T)
using assms by (auto elim!: open_ed_ou_witness.cases)

lemma ed_ou_witness_union:
  fixes X :: ed_ou_witness set set
  assumes ∀x∈X. open x
  shows open (∪X)
using finite[of X] assms
by (induct, force)
  (clarsimp; elim open_ed_ou_witness.cases; simp add: open_ed_ou_witness.simps subset_insertI2 insert_commute; mesi Union_empty_conv)

instance
by standard (auto simp: ed_ou_witness_UNIV intro: ed_ou_witness_inter ed_ou_witness_union)

end
lemma ed_ou_witness_interior_simps:
  shows
  interior \{a\} = \{a\}
  interior \{b\} = \{b\}
  interior \{c\} = \{\}
  interior \{d\} = \{\}
  interior \{e\} = \{e\}
  interior \{a, b\} = \{a, b\}
  interior \{a, c\} = \{a, c\}
  interior \{a, d\} = \{a\}
  interior \{a, e\} = \{a, e\}
  interior \{b, c\} = \{b\}
  interior \{b, d\} = \{b, d\}
  interior \{b, e\} = \{b, e\}
  interior \{c, d\} = \{\}
  interior \{c, e\} = \{e\}
  interior \{d, e\} = \{e\}
  interior \{a, b, c\} = \{a, b, c\}
  interior \{a, b, d\} = \{a, b, d\}
  interior \{a, b, e\} = \{a, b, e\}
  interior \{a, c, d\} = \{a, c\}
  interior \{a, c, e\} = \{a, c, e\}
  interior \{a, d, e\} = \{a, e\}
  interior \{b, c, d\} = \{b, d\}
  interior \{b, c, e\} = \{b, e\}
  interior \{b, d, e\} = \{b, d, e\}
  interior \{c, d, e\} = \{e\}
  interior \{a, b, c, d\} = \{a, b, c, d\}
  interior \{a, b, c, e\} = \{a, b, c, e\}
  interior \{a, b, d, e\} = \{a, b, d, e\}
  interior \{a, c, d, e\} = \{a, c, d, e\}
  interior \{b, c, d, e\} = \{b, c, d, e\}

unfolding interior_def by safe (clarsimp simp: open_ed_ou_witness.simps; blast)+

lemma ed_ou_witness_not_discrete:
  fixes X :: ed_ou_witness_set
  shows ¬discrete X
unfolding discrete_def I_def using ed_ou_witness_interior_simps by (force simp: fun_eq_iff)

lemma ed_ou_witness_ed_ou:
  fixes X :: ed_ou_witness_set
  shows ed_ou X
unfolding ed_ou_def I_def K_def
proof(clarsimp simp: o_apply fun_eq_iff)
  fix x :: ed_ou_witness_set
  from ed_ou_witness_pow[of x]
  show interior (closure x) = closure (interior x)
    by (simp; elim disjE; simp add: closure_interior ed_ou_witness_interior_simps ed_ou_witness_Compl insert_Diff_if)
qed

lemma ed_ou_witness_card:
  shows \{f. CK_nf_pos (f::ed_ou_witness_set \Rightarrow ed_ou_witness_set)\} = 4
by (rule ed_ou_not_discrete_card[OF ed_ou_witness_ed_ou ed_ou_witness_not_discrete])
6.4 Extremally disconnected spaces

definition
extremally_disconnected (X :: 'a::topological_space set) ←→ K ⊖ I ⊖ K = I ⊖ (K :: 'a set ⇒ 'a set)

lemma ed_ou_part_extremally_disconnected:
  assumes ed_ou X
  assumes part X
  shows extremally_disconnected X
using assms unfolding extremally_disconnected_def ed_ou_def part_def by simp

lemma extremally_disconnected_eqs:
  fixes X :: 'a::topological_space set
  assumes extremally_disconnected X
  shows I ⊖ K ⊖ I = K ⊖ I ⊖ K = I ⊖ (K :: 'a set ⇒ 'a set)
using assms unfolding extremally_disconnected_def by (metis K_I_K_I)+

lemma extremally_disconnected_not_part_not_ed_ou_card:
  fixes X :: 'a::topological_space set
  assumes extremally_disconnected X
  assumes ¬part X
  assumes ¬ed_ou X
  shows card {f. CK_nf_pos (f::'a set ⇒ 'a set)} = 5
using extremally_disconnected_eqs[of extremally_disconnected X] CK_nf_pos_lattice[where 'a='a] assms(2,3)
unfolding part_def ed_ou_def
by (simp add: CK_nf_pos_set C_C I_K K_o_assoc card_insert_if; metis (no_types) C_C K_I id_comp o_assoc)

Any topological space having an infinite underlying set and whose topology consists of the empty set and every cofinite subset of the underlying set is extremally disconnected. We consider an example such space having a countably infinite underlying set.

datatype 'a cofinite = cofinite 'a

instantiation cofinite :: (type) topological_space
begin

definition open_cofinite = (λX::'a cofinite set. finite (−X) ∨ X = {})

instance
by standard (auto simp: open_cofinite_def uminus_Sup)
end

lemma cofinite_closure_finite:
  fixes X :: 'a cofinite set
  assumes finite X
  shows closure X = X
using assms by (simp add: closed_open open_cofinite_def)

lemma cofinite_closure_infinite:
  fixes X :: 'a cofinite set
  assumes infinite X
  shows closure X = UNIV
using assms by (metis Compl_empty_eq closure_subset double_compl finite_subset interior_complement open_cofinite_def open_interior)

lemma cofinite_interior_finite:
fixes $X :: \\text{'}a \text{ cofinite set}\\$
assumes $\text{finite } X\\$
assumes $\text{infinite (UNIV::'}a \text{ cofinite set)}\\$
shows $\text{interior } X = \{\}\\$
using $\text{assms cofinite}\_\text{closure}\_\text{infinite}[\text{where } X=-X]$ by $(\text{simp add: interior}\_\text{closure})$

lemma $\text{cofinite}\_\text{interior}\_\text{infinite}$:
fixes $X :: \\text{'}a \text{ cofinite set}\\$
assumes $\text{infinite } X\\$
assumes $\text{infinite } (-X)\\$
shows $\text{interior } X = \{\}\\$
using $\text{assms cofinite}\_\text{closure}\_\text{infinite}[\text{where } X=-X]$ by $(\text{simp add: interior}\_\text{closure})$

abbreviation $\text{evens :: nat cofinite set} \equiv \{\text{cofinite } n \mid n. \exists i. n=2+i\}$

lemma $\text{evens}\_\text{infinite}$:
shows $\text{infinite } \text{evens}$
proof $(\text{rule iffD2} [\text{OF infinite iff countable subset}, \text{rule exI}, \text{rule conjI})$
let $?f = \lambda n::\text{nat}. \text{cofinite } (2+n)$
show inj $?f$ by $(\text{auto intro: inj}\_\text{onI})$
show range $?f \subseteq \text{evens}$ by auto
qed

lemma $\text{cofinite}\_\text{nat}\_\text{infinite}$:
shows $\text{infinite } (-\text{evens})$ using $\text{evens}\_\text{infinite} \text{ finite}\_\text{Diff2}$ by fastforce

lemma $\text{evens}\_\text{Compl}\_\text{infinite}$:
shows $\text{infinite } (-\text{evens})$
proof $(\text{rule iffD2} [\text{OF infinite iff countable subset}, \text{rule exI}, \text{rule conjI})$
let $?f = \lambda n::\text{nat}. \text{cofinite } (2+n+1)$
show inj $?f$ by $(\text{auto intro: inj}\_\text{onI})$
show range $?f \subseteq -\text{evens}$ by clarsimp presburger
qed

lemma $\text{evens}\_\text{closure}$:
shows $\text{closure } \text{evens} = \text{UNIV}$
using $\text{evens}\_\text{infinite} \text{ by (rule cofinite}\_\text{closure}\_\text{infinite)}$

lemma $\text{evens}\_\text{interior}$:
shows $\text{interior } \text{evens} = \{\}$
using $\text{evens}\_\text{infinite} \text{ evens}\_\text{Compl}\_\text{infinite}$ by $(\text{rule cofinite}\_\text{interior}\_\text{infinite})$

lemma $\text{cofinite}\_\text{not}\_\text{part}$:
fixes $X :: \text{nat cofinite set}\\$
shows $\neg \text{part } X\\$
unfolding part_def I_def K_def
using $\text{cofinite}\_\text{nat}\_\text{infinite}$
bymetis $(\text{no_types}) \text{cofinite}\_\text{closure}\_\text{finite} \text{ cofinite}\_\text{interior}\_\text{finite} \text{ double}\_\text{compl} \text{ finite}\_\text{emptyI} \text{ finite}\_\text{insertI} \text{ insert}\_\text{not}\_\text{empty} \text{ interior}\_\text{closure})$

lemma $\text{cofinite}\_\text{not}\_\text{ed}\_\text{ou}$:
fixes $X :: \text{nat cofinite set}\\$
shows $\neg \text{ed\_ou } X\\$
unfolding ed_ou_def I_def K_def
bymetis $(\text{no_types}) \text{cofinite}\_\text{closure}\_\text{finite} \text{ evens}\_\text{closure} \text{ evens}\_\text{interior} \text{ exI}[\text{where } x=\text{evens}]$
lemma cofinite_extremally_disconnected_aux:
  fixes X :: nat cofinite set
  shows closure (interior (closure X)) ⊆ interior (closure X)
by (metis subsetI closure_closure closure_complement closure_def closure_empty finite_Un interior_eq open_cofinite_def open_interior)

lemma cofinite_extremally_disconnected:
  fixes X :: nat cofinite set
  shows extremally_disconnected X
unfolding extremally_disconnected_def I_def K_def
by (auto simp: fun_eq_iff o_apply dest: subsetD[OF closure_subset subsetD[OF interior_subset subsetD[OF cofinite_extremally_disconnected_aux]])

lemma cofinite_card:
  shows card {f. CK_nf_pos (f::nat cofinite set ⇒ nat cofinite set)} = 5
by (rule extremally_disconnected_not_part_not_ed_ou_card[OF cofinite_extremally_disconnected cofinite_not_part cofinite_not_ed_ou])

6.5 Open unresolvable spaces

definition open_unresolvable (X :: 'a::topological_space) :: 'a set ⇒ 'a set
  where "open_unresolvable X = K ◦ I ◦ K = K ◦ (I :: 'a set ⇒ 'a set)

lemma ed_ou_open_unresolvable:
  assumes ed_ou X
  shows open_unresolvable X
using assms unfolding open_unresolvable_def by (simp add: ed_ou_eqs)

lemma open_unresolvable_eqs:
  assumes open_unresolvable (X :: 'a::topological_space set)
  shows I ◦ K ◦ I = I ◦ (K :: 'a set ⇒ 'a set)
    K ◦ I ◦ K = K ◦ (I :: 'a set ⇒ 'a set)
using assms unfolding open_unresolvable_def by (metis I K I o_assoc; simp)

lemma not_ed_ou_neqs:
  assumes ¬ed_ou (X :: 'a::topological_space set)
  shows I ≠ I ◦ (K :: 'a set ⇒ 'a set)
    K ≠ K ◦ (I :: 'a set ⇒ 'a set)
using assms unfolding ed_ou_def
by (simp_all add: fun_eq_iff I K I_def C_def o_apply)
  (metis (no_types, hide_lams) closure_eq_empty disjoint_eq_subset_Compl double_complement interior_Int interior_complement set_eq_subset)+

lemma open_unresolvable_not_ed_ou_card:
  assumes open_unresolvable (X :: 'a::topological_space set)
  assumes ¬ed_ou X
  shows card {f. CK_nf_pos (f::'a set ⇒ 'a set)} = 5
using open_unresolvable_eqs[OF open_unresolvable X] not_ed_ou_neqs[OF ¬ed_ou X] ¬ed_ou X
unfolding ed_ou_def by (auto simp: CK_nf_pos_set_card_insert_if)

We show that the class of open unresolvable spaces is non-empty by exhibiting an example of such a space.

datatype ou_witness = a | b | c

lemma ou_witness_UNIV:
  shows UNIV = set [a, b, c]
using ou_witness.exhaust by auto
instantiation ou_witness :: topological_space
begin

definition open_ou_witness X ←→ a \notin X \lor X = UNIV

instance
by standard (auto simp: open_ou_witness_def)

end

lemma ou_witness_closure_simps:
shows closure {a} = {a}
closure {b} = {a, b}
closure {c} = {a, c}
closure {a, b} = {a, b}
closure {a, c} = {a, c}
closure {a, b, c} = {a, b, c}
closure {b, c} = {a, b, c}
unfolding closure_def islimpt_def open_ou_witness_def by force+

lemma ou_witness_open_unresolvable:
fixes X :: ou_witness set
shows open_unresolvable X
unfolding open_unresolvable_def I_def K_def
by (clarsimp simp: o_apply fun_eq_iff)
  (metis (no_types, lifting) Compl_iff K_I_K_subseteq_K closure_complement closure_interior closure_mono closure_subset interior_eq interior_maximal open_eq interior_maximal open_unresolvable_def subset_antisym)

lemma ou_witness_not_ed_ou:
fixes X :: ou_witness set
shows \neg ed_ou X
unfolding ed_ou_def I_def K_def
by (clarsimp simp: o_apply fun_eq_iff)
  (metis UNIV_I insert_iff interior_eq open_ou_witness_def singletonD ou_witness.distinct(4,5) ou_witness.simps(2) ou_witness_closure_simps(2))

lemma ou_witness_card:
shows card \{f. CK_nf_pos (f::ou_witness set \Rightarrow ou_witness set)\} = 5
by (rule open_unresolvable_not_ed_ou_card[OF ou_witness_open_unresolvable ou_witness_not_ed_ou])

6.6 Kuratowski spaces

definition kuratowski (X :: 'a::topological_space set) ←→

\neg extremally_disconnected X \land \neg open_unresolvable X

A Kuratowski space distinguishes all 7 positive operators.

lemma part_closed_open:
fixes X :: 'a::topological_space set
assumes I \circ K \circ I = (I::'a set \Rightarrow 'a set)
assumes closed X
shows open X
proof(rule Topological_Spaces.openI)
fix x assume x \in X
let ?S = I (-{x})
let ?G = -K ?S
have \( x \in ?G \)

**proof –**

from \( \{ I \circ K \circ I = I \} \) have \( I \ (K \ (I \ ?S)) = ?S \ I \ ?S = ?S \)

unfolding \( I \)\( \)def \( K \)\( \)def by (simp_all add: o_def fun_eq_iff)

then have \( K \ (I \ ?S) \neq \) \( \)UNIV

unfolding \( I \)\( \)def \( K \)\( \)def using interior_subset by fastforce

moreover have \( G \subseteq ?S \lor x \in G \) if open \( G \) for \( G \)

using that unfolding \( I \)\( \)def by (meson interior_maximal subset_CompI singleton)

ultimately show \(?thesis \)

unfolding \( I \)\( \)def \( K \)\( \)def

by clarsimp (metis (no_types, lifting) ComplD Compl_empty_eq closure_interior closure_subset ex_in_conv open_interior_subset_eq)

qed

moreover from \( \{ I \circ K \circ I = I \} \) have open \( ?G \)

unfolding \( I \)\( \)def \( K \)\( \)def by (auto simp: fun_eq_iff o_apply)

moreover have \( ?G \subseteq X \)

proof –

have \( ?G \subseteq K \ ?G \)

unfolding \( K \)\( \)def using closure_subset by fastforce

also from \( \{ I \circ K \circ I = I \} \) have \( \ldots = K \ \{ x \} \)

unfolding \( I \)\( \)def \( K \)\( \)def by (metis closure_interior comp_def double_complement)

also from \( \{ \text{closed} \ X \} \ (x \in X) \) have \( \ldots \subseteq X \)

unfolding \( K \)\( \)def by clarsimp (meson closure_minimal contra_subsetD empty_subsetI insert_subset)

finally show \(?thesis \).

qed

ultimately show \( \exists \ T. \) open \( T \land x \in T \land T \subseteq X \) by blast

qed

**lemma part_I_K_I:**

assumes \( I \circ K \circ I = (I::'a::topological_space set \Rightarrow 'a \ set) \)

shows \( I \circ K = (K::'a \ set \Rightarrow 'a \ set) \)

using interior_open[OF part_closed_open[OF assms closed_closure]] unfolding \( I \)\( \)def \( K \)\( \)def o_def by simp

**lemma part_K_I_I:**

assumes \( I \circ K \circ I = (I::'a::topological_space set \Rightarrow 'a \ set) \)

shows \( K \circ I = (I::'a \ set \Rightarrow 'a \ set) \)

using part_I_K_I[OF assms] assms by simp

**lemma kuratowski_neqs:**

assumes \( \text{kuratowski} \ (X :: 'a::topological_space set) \)

shows \( I \neq I \circ K \circ I \ (I :: 'a \ set \Rightarrow 'a \ set) \)

\( I \circ K \circ I \neq I \circ I \ (I :: 'a \ set \Rightarrow 'a \ set) \)

\( I \circ K \neq K \circ I \circ I (K :: 'a \ set \Rightarrow 'a \ set) \)

\( K \circ I \neq I \circ K \circ I (K :: 'a \ set \Rightarrow 'a \ set) \)

\( K \circ I \neq K \circ I (K :: 'a \ set \Rightarrow 'a \ set) \)

\( K \circ I \neq I \circ K (I :: 'a \ set \Rightarrow 'a \ set) \)

\( I \neq (id :: 'a \ set \Rightarrow 'a \ set) \)

\( K \neq (id :: 'a \ set \Rightarrow 'a \ set) \)

\( I \circ I \neq (id :: 'a \ set \Rightarrow 'a \ set) \)

\( K \circ I \neq (id :: 'a \ set \Rightarrow 'a \ set) \)

\( K \circ K \neq (id :: 'a \ set \Rightarrow 'a \ set) \)

using assms unfolding kuratowski_def extremally_disconnected_def open_unresolvable_def by (metis (no_types, lifting) LK K K K K K K K LK K K I part_I_K_I part_K_I_I o_assoc comp_id)+

**lemma kuratowski_card:**

assumes \( \text{kuratowski} \ (X :: 'a::topological_space set) \)

shows \( \text{card} \ \{ f. \ CK_{nf\ pos} (f::'a \ set \Rightarrow 'a \ set) \} = 7 \)

using CK_nf_pos_lattice[where 'a='a] kuratowski_neqs[OF assms] assms
I \mathbb{R} is a Kuratowski space.

**lemma** kuratowski_reals:

**shows** kuratowski (I \mathbb{R} :: real set)

**unfolding** kuratowski_def extremally_disconnected_def open_unresolvable_def

**by** (rule conjI)


tmetis (no_types, lifting) \simul K list.inject nf_RRR(11) nf_RRR(8) o_assoc,

mtmetis (no_types, lifting) \simul K fun.map_comp list.inject nf_RRR(11) nf_RRR(9))

6.7 Chagrov’s theorem

**theorem** chagrov:

**fixes** X :: 'a::topological_space set

**obtains** discrete X

|¬discrete X ∧ part X
|¬discrete X ∧ ed_ou X
|¬ed_ou X ∧ open_unresolvable X
|¬ed_ou X ∧ ¬part X ∧ extremally_disconnected X
|kuratowski X

**unfolding** kuratowski_def **by** metis

**corollary** chagrov_card:

**shows** card \{f. CK_nf_pos (f::'a::topological_space set ⇒ 'a set)\} ∈ \{1,3,4,5,7\}

**using** discrete_card part_not_discrete_card ed_ou_not_discrete_card open_unresolvable_not_ed_ou_card

extremally_disconnected_not_part_not_ed_ou_card kuratowski_card

**by** (cases rule: chagrov) blast+

References


