# The Teichmüller–Tukey Lemma

Vithor Lindermann Kraisch
Department of Mobility Engineering
Federal University of Santa Catarina, Joinville, Brazil

Luiz Gustavo Cordeiro
Department of Mobility Engineering
Federal University of Santa Catarina, Joinville, Brazil

September 23, 2026

**Abstract**

This entry formalizes the Teichmüller–Tukey lemma in Isabelle/HOL: every nonempty family of sets of finite character has a member that is maximal under inclusion. Instead of deriving the result from Zorn's lemma, which is already available in Isabelle/HOL, the development follows the direct choice-function construction of Sun and Yu, originally formulated in Morse–Kelley set theory and checked in Coq. The Isabelle proof makes explicit the choice argument that turns a fixed point of the construction into a maximal member.

## Contents



## 1 The Teichmüller–Tukey Lemma

**theory** *Teichmuller-Tukey-Lemma*
  **imports** *Main*
**begin**

The Teichmüller–Tukey lemma states that every nonempty family of sets of finite character has a member that is maximal under inclusion. Although the result follows from Zorn's lemma, the development below formalizes the direct choice-function construction presented by Sun and Yu [2]. Their proof is formulated in Morse–Kelley set theory and checked in Coq; here it is adapted to polymorphic HOL sets.

This result was developed as infrastructure for a separate formalization of first-order logic following Shoenfield [1].

### 1.1 Families of finite character

**definition** *Maximal* :: $'a\ set \Rightarrow 'a\ set\ set \Rightarrow bool$ **where**
 $Maximal\ A\ J \equiv A \in J \wedge (\forall B{\in}J.\ \neg\ A \subset B)$

**definition** *Nest* :: $'a\ set\ set \Rightarrow bool$ **where**
 $Nest\ A \equiv \forall x{\in}A.\ \forall y{\in}A.\ x \subseteq y \vee y \subseteq x$

**definition** *Finite-character* :: $'a\ set\ set \Rightarrow bool$ **where**
 $Finite\text{-}character\ J \equiv \forall \Delta.\ \Delta \in J \longleftrightarrow (\forall \delta{\subseteq}\Delta.\ finite\ \delta \longrightarrow \delta \in J)$

A family of finite character is downward closed. It is also closed under the union of a nest. Nonemptiness is needed for the empty nest: it ensures that the empty set belongs to the family.

**proposition** *property-fin-char*:
 **assumes** $Finite\text{-}character\ J$ **and** $J \neq \{\}$
 **shows** $\forall A{\in}J.\ \forall B{\subseteq}A.\ B \in J$
  **and** $\forall G.\ Nest\ G \wedge G \subseteq J \longrightarrow \bigcup G \in J$
**proof** −
 **show** $\forall A{\in}J.\ \forall B{\subseteq}A.\ B \in J$
  **using** *Finite-character-def assms(1)* **by** (*meson subset-trans*)
**next**
 **show** $\forall G.\ Nest\ G \wedge G \subseteq J \longrightarrow \bigcup G \in J$
 **proof** (*rule allI*, *rule impI*)
  **fix** $G$
  **assume** $B$: $Nest\ G \wedge G \subseteq J$
  **have** $C$: $\bigwedge g.\ set\ g \subseteq \bigcup G \Longrightarrow g \neq [] \Longrightarrow \exists X{\in}G.\ set\ g \subseteq X$
  **proof** −
   **fix** $g$
   **show** $set\ g \subseteq \bigcup G \Longrightarrow g \neq [] \Longrightarrow \exists X{\in}G.\ set\ g \subseteq X$
   **proof** (*induction g rule*: *measure-induct*[*of length*])
    **case** (*1 g*)
    **then show** *?case*
    **proof** *cases*
     **assume** $g = [] \vee (\exists a.\ g = [a])$
     **then show** *?case* **using** *1.prems(1,2)* **by** *fastforce*
    **next**
     **assume** $\neg\ (g = [] \vee (\exists a.\ g = [a]))$
     **then obtain** $a$ **and** $xs$ **where** $axs$: $g = a \# xs \wedge xs \neq []$
      **by** (*metis list.exhaust*)
     **from** *this 1(1−2)* **have**
      $(\exists X{\in}G.\ \{a\} \subseteq X) \wedge (\exists X{\in}G.\ set\ xs \subseteq X)$
      **by** *auto*

**thus** *?case*
**using** *B Nest-def axs*
**by** (*metis dual-order.trans insert-subset list.simps*(*15*))
**qed**
**qed**
**qed**
**thus** $\bigcup G \in J$
**using** *assms*(*1*−*2*) *B C Finite-character-def*
**by** (*smt* (*z3*) *ex-in-conv finite-list set-empty subset-eq list.set*(*1*,*1*,*1*,*1*) *subsetI subset-antisym*)
**qed**
**qed**

## 1.2 The choice extension

**fun** *enlarge* :: $'a\ set \Rightarrow 'a\ set\ set \Rightarrow 'a\ set$ **where**
$enlarge\ F\ J = \{x.\ x \in \bigcup J \wedge F{\cup}\{x\} \in J\}$

**proposition** *enlarge-subset*: $F \in J \Longrightarrow F \subseteq enlarge\ F\ J$
**using** *UnionI insert-absorb* **by** *force*

**fun** $\mathcal{X}$*-choice* :: $'a\ set \Rightarrow 'a\ set\ set \Rightarrow ('a\ set \Rightarrow 'a) \Rightarrow 'a\ set$ **where**
$\mathcal{X}$*-choice* $F\ J\ c =$
(*if* $enlarge\ F\ J - F = \{\}$ *then* $F$ *else* $F{\cup}\{c\ (enlarge\ F\ J - F)\}$)

For a member $F$ of $J$, *enlarge* $F\ J$ contains the elements that can be adjoined to $F$ while staying in $J$. The operation $\mathcal{X}$*-choice* $F\ J\ c$ adjoins an element chosen outside $F$, unless no such element exists.

The fixed-point argument below makes explicit a detail that is only implicit in Proof 4.7 of the source article. From $\mathcal{X}$*-choice* $F\ J\ c = F$ one may conclude *enlarge* $F\ J = F$ only by using the choice property of $c$: if *enlarge* $F\ J - F$ were nonempty, its chosen element would lie both outside and inside $F$. This is the step that turns a fixed point of the choice extension into a maximal member.

**lemma** $\mathcal{X}$*-implies-TTL*:
**assumes** *Finite-character J*
**and** $\exists F{\in}J.\ \exists c.\ ((\forall Y.\ Y \neq \{\} \longrightarrow c\ Y \in Y) \wedge \mathcal{X}\text{-}choice\ F\ J\ c = F)$
**shows** $\exists \Delta{\in}J.\ Maximal\ \Delta\ J$
**proof** −
**from** *assms*(*2*) **obtain** $\Delta$ **and** $c$ **where**
$\Delta c$: $\Delta \in J \wedge (\forall Y.\ Y \neq \{\} \longrightarrow c\ Y \in Y) \wedge \mathcal{X}\text{-}choice\ \Delta\ J\ c = \Delta$
**by** *blast*
**hence** *ig*: $enlarge\ \Delta\ J = \Delta$
**by** (*metis DiffD2 Un-upper2* $\mathcal{X}$*-choice.elims diff-shunt enlarge-subset insert-subset order-antisym-conv*)
**have** *Maximal* $\Delta\ J$
**proof** (*rule ccontr*)
**assume** $\neg$ *Maximal* $\Delta\ J$

**from** *this* $\Delta c$ **obtain** $B$ **where** $B$: $B \in J \wedge \Delta \subset B$
**using** *Maximal-def* **by** *meson*
**then obtain** $d$ **where** $d$: $d \in B \wedge d \notin \Delta$ **by** *blast*
**hence** $d \in$ *enlarge* $\Delta$ $J$
**using** *assms*(*1*) *Finite-character-def* $B$ *property-fin-char*(*1*) **by** *fastforce*
**thus** *False* **using** *ig* $d$ **by** *blast*
**qed**
**thus** $\exists\, \Delta{\in}J.$ *Maximal* $\Delta$ $J$ **using** $\Delta c$ **by** *blast*
**qed**

## 1.3 The least closed subclass

Following the notation of the source proof, a *t-subclass* contains the empty set and is closed under both choice extensions and unions of nests. Intersecting all t-subclasses gives the least such subclass.

**definition** *tSubclass* :: $'a$ *set set* $\Rightarrow$ $'a$ *set set* $\Rightarrow$ ($'a$ *set* $\Rightarrow$ $'a$) $\Rightarrow$ *bool* **where**
*tSubclass* $G$ $J$ $c$ $\equiv$
$G \subseteq J \wedge \{\} \in G \wedge (\forall\, F{\in}G.\ \mathcal{X}\text{-}choice\ F\ J\ c \in G) \wedge$
$(\forall\, L{\subseteq}G.\ Nest\ L \longrightarrow \bigcup L \in G)$

**definition** *inter-subclass* :: $'a$ *set set* $\Rightarrow$ ($'a$ *set* $\Rightarrow$ $'a$) $\Rightarrow$ $'a$ *set set* **where**
*inter-subclass* $J$ $c$ $\equiv$ $\bigcap\{G.\ tSubclass\ G\ J\ c\}$

**definition** $\mu$*-function* :: $'a$ *set* $\Rightarrow$ $'a$ *set set* $\Rightarrow$ ($'a$ *set* $\Rightarrow$ $'a$) $\Rightarrow$ $'a$ *set set* **where**
$\mu$*-function* $C$ $J$ $c$ $\equiv$
$\{A{\in}inter\text{-}subclass\ J\ c.\ A \subseteq C \vee C \subseteq A\}$

**definition** *total-subclass* :: $'a$ *set set* $\Rightarrow$ ($'a$ *set* $\Rightarrow$ $'a$) $\Rightarrow$ $'a$ *set set* **where**
*total-subclass* $J$ $c$ $\equiv$
$\{C{\in}inter\text{-}subclass\ J\ c.\ \mu\text{-}function\ C\ J\ c = inter\text{-}subclass\ J\ c\}$

**definition** $\nu$*-function* :: $'a$ *set* $\Rightarrow$ $'a$ *set set* $\Rightarrow$ ($'a$ *set* $\Rightarrow$ $'a$) $\Rightarrow$ $'a$ *set set* **where**
$\nu$*-function* $D$ $J$ $c$ $\equiv$
$\{A{\in}inter\text{-}subclass\ J\ c.\ A \subseteq D \vee \mathcal{X}\text{-}choice\ D\ J\ c \subseteq A\}$

**proposition** *property-*$\mathcal{X}$: $F \in J \Longrightarrow F \subseteq \mathcal{X}\text{-}choice\ F\ J\ c$
**by** *auto*

**proposition** *property-inter-subclass*:
**assumes** *Finite-character* $J$ **and** $J \neq \{\}$
**and** $\forall\, Y.\ Y \neq \{\} \longrightarrow c\ Y \in Y$
**shows** *tSubclass* (*inter-subclass* $J$ $c$) $J$ $c$
**and** $\forall\, G.\ tSubclass\ G\ J\ c \longrightarrow inter\text{-}subclass\ J\ c \subseteq G$
**proof** $-$
**have** *JSubJ*: *tSubclass* $J$ $J$ $c$
**using** *tSubclass-def Nest-def assms*(*1*$-$*3*) *property-fin-char*(*2*)
**by** (*smt* (*verit*, *ccfv-threshold*) *Diff-iff Sup-empty* $\mathcal{X}$*-choice.simps*
*dual-order.refl empty-iff empty-subsetI enlarge.simps mem-Collect-eq*)
**show** *tSubclass* (*inter-subclass* $J$ $c$) $J$ $c$

**proof** (*auto simp only*: *tSubclass-def*[*of inter-subclass J c J c*])
  **show** $\{\} \in$ *inter-subclass J c*
    **using** *inter-subclass-def tSubclass-def*
    **by** (*metis* (*no-types*, *lifting*) *Inter-iff mem-Collect-eq*)
  **next**
    **have** $\forall\, G\, F.$ *tSubclass G J c* $\longrightarrow F \in G \longrightarrow \mathcal{X}$*-choice F J c* $\in G$
      **using** *tSubclass-def* **by** *blast*
    **thus** $\bigwedge F.\ F \in$ *inter-subclass J c* $\Longrightarrow \mathcal{X}$*-choice F J c* $\in$ *inter-subclass J c*
      **using** *inter-subclass-def* **by** *blast*
  **next**
    **have** $\forall\, G\, L.$ *tSubclass G J c* $\longrightarrow L \subseteq G \longrightarrow$ *Nest L* $\longrightarrow \bigcup L \in G$
      **using** *tSubclass-def* **by** *metis*
    **thus** $\bigwedge L.\ L \subseteq$ *inter-subclass J c* $\Longrightarrow$ *Nest L* $\Longrightarrow \bigcup L \in$ *inter-subclass J c*
      **using** *inter-subclass-def*[*of J c*] **by** *blast*
  **qed** (*auto simp add*: *JSubJ inter-subclass-def*)
**qed** (*auto simp add*: *inter-subclass-def*)

## 1.4 The least t-subclass is a Nest

The next three lemmas correspond to Steps I–III of Section 4.1 in the source. The auxiliary subclasses $\mu$*-function C J c* and $\nu$*-function D J c* isolate the members comparable with a given set and allow comparability to be propagated through the choice extension.

**lemma** *step1*:
  **assumes** *Finite-character J* **and** $J \neq \{\}$
    **and** $\forall\, Y.\ Y \neq \{\} \longrightarrow c\ Y \in Y$
    **and** $D \in$ *total-subclass J c*
  **shows** *tSubclass* ($\nu$*-function D J c*) *J c*
**proof** −
  **show** *?thesis*
  **proof** (*auto simp only*: *tSubclass-def*[*of* $\nu$*-function D J c J c*])
    **have** $\nu$*-function D J c* $\subseteq$ *inter-subclass J c*
      **using** $\nu$*-function-def* **by** *blast*
    **thus** $\bigwedge x.\ x \in \nu$*-function D J c* $\Longrightarrow x \in J$
      **using** *tSubclass-def property-inter-subclass*(*1*) *assms* **by** (*metis in-mono*)
  **next**
    **fix** *F*
    **assume** $F \in \nu$*-function D J c*
    **then have** *F*: $F \in$ *inter-subclass J c* $\wedge$
        ($F \subseteq D \vee \mathcal{X}$*-choice D J c* $\subseteq F$)
      **using** $\nu$*-function-def* **by** *blast*
    **show** $\mathcal{X}$*-choice F J c* $\in \nu$*-function D J c*
    **proof** *cases*
      **assume** *c1*: $\mathcal{X}$*-choice D J c* $\subseteq F$
      **then show** *?thesis*
        **using** *F* $\nu$*-function-def assms*(*1*−*3*) *property-*$\mathcal{X}$
          *property-inter-subclass*(*1*) *tSubclass-def*
        **by** (*smt* (*verit*) *in-mono mem-Collect-eq subset-trans*)
    **next**

**assume** $\neg$ $\mathcal{X}$*-choice D J c* $\subseteq$ *F*
**hence** *b*: $\mathcal{X}$*-choice F J c* $\subseteq$ *D* $\vee$ *D* $\subseteq$ $\mathcal{X}$*-choice F J c*
**using** *assms F* $\mu$*-function-def tSubclass-def property-inter-subclass*(*1*) *total-subclass-def*
**by** (*smt* (*verit*, *del-insts*) *mem-Collect-eq*)
**show** *?thesis*
**proof** (*rule ccontr*)
**assume** *hyp*: $\mathcal{X}$*-choice F J c* $\notin$ $\nu$*-function D J c*
**then have** $\neg$ $\mathcal{X}$*-choice F J c* $\subseteq$ *D*
**using** *assms*(*1*−*3*) $\nu$*-function-def F tSubclass-def property-inter-subclass*(*1*)
**by** (*metis* (*no-types*, *lifting*) *mem-Collect-eq*)
**from** *this b* **have** *D* $\subset$ $\mathcal{X}$*-choice F J c* **by** *blast*
**hence** *D* $\subset$ *F* $\vee$ *D* $\subset$ *F*$\cup${*c* (*enlarge F J* − *F*)}
**by** (*metis* $\mathcal{X}$*-choice.simps*)
**hence** $\mathcal{X}$*-choice D J c* $\subseteq$ $\mathcal{X}$*-choice F J c*
**using** *F* ‹$\neg$ $\mathcal{X}$*-choice D J c* $\subseteq$ *F*›
**by** (*metis Un-insert-right insert-subsetI leD order-class.order-eq-iff psubset-insert-iff sup-bot-right*)
**thus** *False*
**using** *hyp* $\nu$*-function-def F assms*(*1*−*3*) *property-inter-subclass*(*1*) *tSubclass-def*
**by** (*metis* (*no-types*, *lifting*) *mem-Collect-eq*)
**qed**
**qed**
**next**
**fix** *L*
**assume** *hyp1*: *L* $\subseteq$ $\nu$*-function D J c* **and** *hyp2*: *Nest L*
**then have** *UL*: $\bigcup$ *L* $\in$ *inter-subclass J c*
**using** $\nu$*-function-def*[*of D J c*] *property-inter-subclass*(*1*) *assms*(*1*−*3*) *tSubclass-def*
**by** (*smt* (*verit*) *mem-Collect-eq subset-eq*)
**have** $\bigcup$ *L* $\subseteq$ *D* $\vee$ $\mathcal{X}$*-choice D J c* $\subseteq$ $\bigcup$ *L*
**proof** *cases*
**assume** $\forall$ *l*$\in$*L*. *l* $\subseteq$ *D*
**then show** *?thesis* **using** *UL* $\nu$*-function-def* **by** *blast*
**next**
**assume** $\neg$ ($\forall$ *l*$\in$*L*. *l* $\subseteq$ *D*)
**from** *this hyp1* **show** *?thesis* **using** $\nu$*-function-def*[*of D J c*] **by** *blast*
**qed**
**thus** $\bigcup$ *L* $\in$ $\nu$*-function D J c*
**using** $\nu$*-function-def*[*of D J c*] *UL* **by** *blast*
**qed** (*simp add*: $\nu$*-function-def inter-subclass-def tSubclass-def*)
**qed**

**lemma** *step2*:
**assumes** *Finite-character J* **and** *J* $\neq$ {}
**and** $\forall$ *Y*. *Y* $\neq$ {} $\longrightarrow$ *c Y* $\in$ *Y*
**and** *D* $\in$ *total-subclass J c*

**shows** $\mathcal{X}$*-choice D J c* ∈ *total-subclass J c*
**proof** −
**from** *assms* **have** *inter-subclass J c* ⊆ *ν-function D J c*
**using** *step1 property-inter-subclass*(*2*) **by** *metis*
**moreover have** *ν-function D J c* ⊆ *μ-function* ($\mathcal{X}$*-choice D J c*) *J c*
**using** *μ-function-def ν-function-def*
**by** (*smt* (*verit*, *best*) *Un-upper1* $\mathcal{X}$*-choice.simps mem-Collect-eq subsetD subsetI*)
**moreover have** ... ⊆ *inter-subclass J c* **using** *μ-function-def* **by** *blast*
**ultimately have**
*μ-function* ($\mathcal{X}$*-choice D J c*) *J c* = *inter-subclass J c*
**by** *order*
**thus** *?thesis*
**using** *assms total-subclass-def*[*of J c*] *property-inter-subclass*[*of J c*]
*tSubclass-def*[*of inter-subclass J c J c*] **by** *simp*
**qed**

**lemma** *step3*:
**assumes** *Finite-character J* **and** *J* ≠ {}
**and** ∀ *Y*. *Y* ≠ {} ⟶ *c Y* ∈ *Y*
**shows** *Nest* (*inter-subclass J c*)
**proof** −
**have** *Nest* (*total-subclass J c*)
**using** *Nest-def*[*of total-subclass J c*] *total-subclass-def*[*of J c*]
*μ-function-def*[*of - J c*] **by** *blast*
**moreover have** *a*: *total-subclass J c* ⊆ *inter-subclass J c*
**using** *total-subclass-def* **by** *auto*
**moreover have** *tSubclass* (*total-subclass J c*) *J c*
**proof** (*auto simp only*: *tSubclass-def*[*of total-subclass J c J c*])
**show** ⋀*x*. *x* ∈ *total-subclass J c* ⟹ *x* ∈ *J*
**using** *a assms property-inter-subclass*[*of J c*] *tSubclass-def* **by** (*meson subsetD*)
**show** {} ∈ *total-subclass J c*
**using** *assms property-inter-subclass*[*of J c*] *μ-function-def*
*tSubclass-def*[*of inter-subclass J c*] *total-subclass-def*[*of J c*]
**by** *fastforce*
**show** ⋀*F*. *F* ∈ *total-subclass J c* ⟹
$\mathcal{X}$*-choice F J c* ∈ *total-subclass J c*
**using** *assms step2* **by** *blast*
**show** ⋀*L*. *L* ⊆ *total-subclass J c* ⟹ *Nest L* ⟹
⋃*L* ∈ *total-subclass J c*
**proof** −
**fix** *L*
**assume** *hyp1*: *L* ⊆ *total-subclass J c* **and** *Nest L*
**then have** ⋃*L* ∈ *inter-subclass J c*
**using** *property-inter-subclass*(*1*) *assms tSubclass-def a*
**by** (*metis subset-trans*)
**moreover from** *hyp1* **have**
∀ *A*∈*inter-subclass J c*. *A* ⊆ ⋃*L* ∨ ⋃*L* ⊆ *A*
**using** *total-subclass-def*[*of J c*] *μ-function-def*[*of - J c*] **by** *blast*

```
      ultimately show ⋃L ∈ total-subclass J c
        using total-subclass-def[of J c] μ-function-def[of ⋃L J c]
        by auto
    qed
  qed
  hence inter-subclass J c ⊆ total-subclass J c
    using inter-subclass-def by blast
  ultimately show ?thesis by simp
qed
```

## 1.5 The fixed point and the main theorem

Since the least t-subclass is a nest, its union belongs to it. Closure under the choice extension then places the extension in the same nest, while the definition of union bounds it from above. Hence the union is a fixed point.

```
lemma step4:
  assumes Finite-character J and J ≠ {}
    and ∀ Y. Y ≠ {} ⟶ c Y ∈ Y
  shows 𝒳-choice (⋃(inter-subclass J c)) J c = ⋃(inter-subclass J c)
  using assms property-𝒳 property-fin-char(2) property-inter-subclass step3
    tSubclass-def
  by (metis Union-upper subset-antisym)

theorem teichmueller-tuckey-lemma:
  fixes J :: 'a set set
  assumes Finite-character J and J ≠ {}
  shows ∃Δ∈J. Maximal Δ J
proof −
  let ?c = λY :: 'a set. SOME y. y ∈ Y
  have choice: ∀ Y :: 'a set. Y ≠ {} ⟶ ?c Y ∈ Y
    by (simp add: some-in-eq)
  have fixed:
    𝒳-choice (⋃(inter-subclass J ?c)) J ?c =
      ⋃(inter-subclass J ?c)
    using assms choice step4 by metis
  have member: ⋃(inter-subclass J ?c) ∈ J
    using assms choice property-fin-char(2) property-inter-subclass(1)
      step3 tSubclass-def by metis
  show ?thesis
    using assms(1) choice fixed member 𝒳-implies-TTL[of J]
    by metis
qed

end
```

# References


[1] J. R. Shoenfield. *Mathematical Logic*. Addison-Wesley, 1967.

[2] T. Sun and W. Yu. Formalization of the axiom of choice and its equivalent theorems. arXiv preprint arXiv:1906.03930v1, 2019. https://arxiv.org/abs/1906.03930v1.