University of Benghazi
Faculty of Science
Department of Mathematics

Finite Topological Spaces with Maple

A dissertation Submitted to the Department of Mathematics in Partial Fulfillment of the requirements for the degree of Master of Science in Mathematics

By

Taha Guma el turki

Supervisor

Prof. Kahtan H.Alzubaidy

Benghazi –Libya

2013/2014

Dedication

For the sake of science and progress in my country new Libya .

Taha

Acknowledgements

I don’t find words articulate enough to express my gratitude for the help and grace that Allah almighty has bestowed upon me.

I would like to express my greatest thanks and full gratitude to my supervisor Prof. Kahtan H. Al zubaidy for his invaluable assistance patient guidance and constant encouragement during the preparation of the thesis.

Also, I would like to thank the department of Mathematics for all their efforts advice and every piece of knowledge they offered me to achieve the accomplishment of writing this thesis.

Finally , I express my appreciation and thanks to my family for the constant support.

Contents

Appendix-Maple

Abstract

In this thesis , we discuss the properties of finite topological spaces and their different properties from other topological spaces including Alexandroff spaces . And also we create and apply some Maple programming procedures to compute the special points of a set in a space and the number of topological spaces of a given cardinality .

Introduction

The finite topological spaces have recently captured topologists's attention. Since digital processing and image processing start of finite sets of observations and seek to understand pictures that emerge from notion of nearness of points .There was a brief early flurry of beautiful mathematical works on this subject.Two independent papers, by McCord [2] and Stong [7] , both published in 1966 ,are especially interesting . In this thesis we will work through them , also through a lecture note by May J.P [1] , published on the internet in 2008.

Chapter one starts with the definition of finite topological spaces and minimal basic open sets . And also we discuss the preorder relation and its relation with separation axioms , then we discuss the different properties of finite topological spaces in continuity and homeomorphisms , compactness , connectivity and path connectivity .

In chapter two we study Alexandroff spaces we show how to construct new Alexandroff spaces from given ones and discuss some of the very important properties of Alexandroff spaces .

In chapter three we create procedures of Maple 15 programming to compute the six special points and the topologies and T_0 topologies on a finite set . Also procedures to find minimal bases and connected components of a finite space .

There is appendix to introduce the most important information of maple program and its packages of commands .

The list of used references is put at the end of the thesis . An abstract in Arabic is provided also .

Two papers have been extracted from this thesis and published on the following links:

Maple in Finite Topological Spaces – Special Points .
Kahtan H.Alzubaidy, Taha Guma El turki

http://www.maplesoft.com/applications/view.aspx?SID=145571 ,(April 2013)

Maple in Finite Topological Spaces –Connectedness .
Taha Guma El turki , Kahtan H.Alzubaidy,

http://www.maplesoft.com/applications/view.aspx?SID=150631 ,(August2013)

Chapter Zero

Preliminaries

<u>Partially Ordered Sets</u>

<u>Definition 0.1:</u>

A partially ordered set (poset) (A , ≤) consist :

a non empty set A and a binary relation ≤ on A such that

for all a , b , $c \in$ A :

(i) ≤ is reflexive *i.e.*, $a \leq a$.

(ii) ≤ is anti symmetric *i.e.*, if $a \leq b$ and $b \leq a$, then $a = b$.

(iii) ≤ is transitive *i.e* ., if $a \leq b$ and $b \leq c$, then $a \leq c$ [5 , p. 2] .

$x \leq y$ is read as x precedes (contained in) y or y dominate (contains) x .

<u>Definition 0.2:</u>

A partial order relation ≤ is called a totally order (or linear order or chain) if for any a , $b \in$ A , we have either $a \leq b$ or $b \leq a$ [5 , p. 2] .

<u>Examples 0.1:</u>

(i) $\mathbb{R}$ with ≤ (magnitude) ($\mathbb{R}$, ≤) is a poset in fact it's a chain .

(ii) $\mathcal{A}$ is a family of sets with the inclusion ⊆ , ($\mathcal{A}$, ⊆) is a poset

(iii) $\mathbb{Z}^+$ with division $|$ is a poset , but it is not a totally ordered set ,

since $3 , 7 \in \mathbb{Z}^+$ and neither $3 \nmid 7$ nor $7 \nmid 3$.

<u>Definition 0.3</u>:

A preorder or quasiorder on a non empty set A is a binary relation that is reflexive and transitive [5 , p. 3] .

<u>Definition 0.4</u>:

An equivalence relation on a non empty set A is a binary relation that is reflexive, symmetric (*i.e*., if $x \leq y$, then $y \leq x$) and transitive. If $x \in$ A , then the set [x] of elements of A that equivalent to x is called an equivalence class of x [5 , p. 2] .

<u>Definition 0.5:</u>

A binary relation $<$ on a non empty set A is called a strict order if for all $a , b , c \in$ A , we have :

(i) $<$ is irreflexive *i.e* ., $a \nless a$.

(ii) $<$ is transitive *i.e* ., if $a < b$ and $b < c$, then $a < c$ [5 , p. 4] .

Relationships between $<$ and $\leq$:

(i) $a \leq b$ iff $a < b$ or $a = b$.

(ii) $a < b$ iff $a \leq b$ and $a \neq b$ [5 , p. 4].

Remark:

The inverse order of an order $\leq$ is denoted by $\geq$.

$\geq$ is defined as follows : $a \geq b$ iff $b \leq a$.

Definition 0.6:

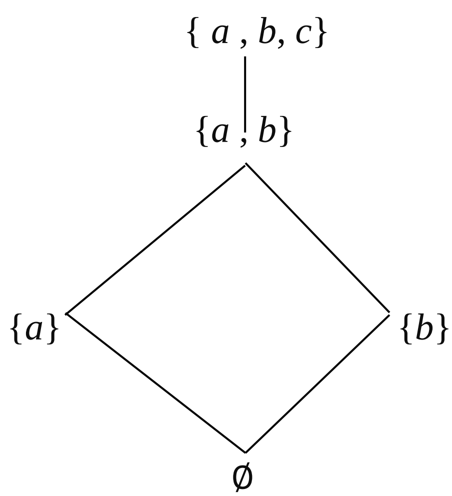
Let (A , $\leq$) be a poset . Then y covers x in A , denoted by $x \sqsubset y$, if $x < y$ and no element in A lies strictly between x and y , that is , if $x \leq z \leq y$ then $x = z$ or $y = z$.

If $x \sqsubset y$, or $x = y$, we write $x \sqsubseteq y$ [5 , p. 4] .

For a finite poset A , the covering relation uniquely determines the order on A , since $x \leq y$, if and only if there is a finite sequence of elements of A such that , $x \sqsubset p_1 \sqsubset p_2 \sqsubset p_3 \sqsubset . . . \sqsubset p_n \sqsubset y$ .Small finite posets are often described with a diagram called a Hasse diagram , which is a graph whose nodes are labeled with the elements of the poset and whose edges indicate the covering relation .This is illustrated in the following example .

Example 0.2:

Figure (1) shows the Hasse diagram of the poset $P = \{\emptyset , \{a\} , \{b\} , \{a , b\},\{a , b ,c\} \}$ under inclusion .



Definition 0.7:

If (A , $\leq$) is a poset and B $\subseteq$ A , then (B , $\leq$) is a poset it is called a subposet of a poset A .

Definition 0.8:

Let (A , $\leq_1$) and (B , $\leq_2$) be two posets , suppose that C = A $\times$ B , then C can be made a poset such that :

(i) ***Product order :***

$(a , b) \leq (c , d)$ iff $a \leq_1 c$ and $b \leq_2 d$. $\leq$ is a partially order on C .

Remark:

If $\leq_1$ and $\leq_2$ are total orders , then $\leq$ may not be total .

(ii) ***Lexicographical order (dictionary order):***

$(a , b) \leq (c , d)$ iff $a <_1 c$ or $(a = c$ and $b \leq_2 d)$. $\leq$ is partially order on C [5 , p. 4] .

Remark:

if $\leq_1$ and $\leq_2$ are total orders , then $\leq$ is a total order .

Definition 0.9 :

Let (A , $\leq_1$) and (B , $\leq_2$) be two posets. A function f : A $\longrightarrow$ B is order preserving if $x \leq_1 y$ implies $f(x) \leq_2 f(y)$ where $x , y \in$ A .

Definition 0.10:

Two posets (A , $\leq_1$) and (B , $\leq_2$) are called order isomorphic , if there exists a one-to-one onto function f: A $\longrightarrow$ B such that f and f^{-1} are order-preserving .Also f is called order isomorphism [5 , p.13].

Definition 0.11:

Let A be a poset .

(i) An element $s \in$ A is called smallest element of A , if $s \leq x$ for all $x \in$ A .

(ii) An element g $\in$ A is called largest element of A , if $x \leq g$ for all $x \in$ A .

(iii) An element $m \in$ A is called minimal element of A , if there is no $x \in$ A such that $x < m$.

i.e., if there is $x \in$ A such that $x \leq m$, then $x = m$.

(iv) An element g $\in$ A is called maximal element of A , if there is no $x \in$ A such that g $< x$.

i.e., if there is $x \in$ A such that $g \leq x$, then $x = g$ [5 , p. 6] .

Definition 0.12:

Let (A , $\leq$) be a poset and B $\subseteq$ A .

(i) An element $u \in$ A is an upper bound of B if $x \leq u$ for any $x \in$ B .

The least upper bound or Supremum of B is an upper bound which precedes every other upper bound of B . It is denoted by sup.(B) . *i.e.*, sup.(B) $\leq u$ for each upper bound u of B .

sup.(B) is the smallest upper bound of B .

(ii) An element $l \in$ A is an lower bound of B if $l \leq x$ for any $x \in$ B .

The greatest lower bound or Infimum of B is a lower bound

which dominates every other lower bound of B . It is denoted by inf.(B).

i.e., $l \leq$ inf.(B) for each lower bound l of B .

inf.(B) is the greatest lower bound of B .

Lemma 0.1 :

Smallest and greatest elements are unique , if they exists .

Definition 0.13:

A poset (A , $\leq$) is called complete , if for every B $\subseteq$ A , inf.(B) and sup.(B) both exist .

Lattices :

Definition 0.14:

A poset L is a lattice if for every a , $b \in$ L both sup.{a , b} and inf .{a , b} exist in L [5, p .53] .

Notation:

sup.{ a , b} = $a \vee b$ and Inf .{ a , b } = $a \wedge b$

$\vee$ is called join and $\wedge$ is called meet [5 , p .53] .

Examples 0.3:

(i) ($\mathbb{R}$, $\leq$) is a lattice .

Since for any $x , y \in \mathbb{R}$, $x \vee y = \max\{ x , y \}$ and $x \wedge y = \min \{x , y\}$

(ii) if A is a set , then $(2^A, \subseteq)$ is a lattice .

For any $X, Y \subseteq A$, we have $X \vee Y = X \cup Y$ and $X \wedge Y = X \cap Y$.

(iii) $(\mathbb{Z}^+, \mid)$ is a lattice

$x \wedge y = (x , y)$ greatest common divisor

$x \vee y = [x , y]$ least common multiple .

<u>Definition 0.15:</u>

A poset L is a complete lattice if L has arbitrary joins and arbitrary meets.

Topological Spaces

In this section we will introduce the basic definitions that are related to topological spaces and also discuss the most important topological properties

<u>Definition 0.16:</u>

A topology on a set X consist of a collection τ of subsets of X called (open sets of X) .With the following properties :

(i) $\emptyset \in \tau$ and $X \in \tau$.

(ii) if $O_1 , O_2 \in \tau$, then $O_1 \cap O_2 \in \tau$, and

(iii) if $O_\alpha \in \tau$ for any $\alpha \in \mathbf{J}$, then $\bigcup_{\alpha \in J} O_\alpha \in \tau$.

(ii) and (iii) mean that the collection τ is closed under finite intersections and arbitrary unions [4 , p .1] .

<u>Examples 0.4:</u>

(i) ***<u>Discrete Topology</u>:***

If X is a set , then $\delta = 2^X = \{ O : O \subseteq X \}$ is a topology on X , it is called discrete topology . It's the largest topology on X .

(ii) ***Indiscrete Topology :***

If X is a set , then τ={∅ , X} is topology on X , it is called indiscrete topology , it's the smallest topology on X .

(iii) ***Cofinite Topology :***

If X is a set , τ= { O ⊆ X : X − O is finite} ∪ {ϕ}, τ is a topology on X is called cofinite topology ∅ ∈ τ and X ∈ τ, since X−X= ∅ is finite .

Let O_1 , O_2 ∈ τ, then X − O_1 and X – O_1 are finite .

X – (O_1 ∩ O_2) = (X − O_1) ∪ (X − O_2) which is finite and hence

O_1 ∩ O_2 ∈ τ. Let Oα ∈ τ for any α ∈ **J** , then X − Oα is finite

for any α ∈ **J** . $X - \bigcup_{\alpha \in J} O\alpha = \bigcap_{\alpha \in J} (X - O\alpha)$,which is finite and hence

$\bigcup_{\alpha \in J} O\alpha \in \tau$.

(iv) ***Metric Topology :***

If (X , d) is a metric space then

τ_d = { O ⊆ X : ∀ a ∈ O , ∃ $\epsilon > 0$, *s.t* $a \in B_d(a\,;\epsilon)$ ⊆ O }

is the metric topology on X induced by d . For example the usual metric topology on ℝ is

$\mathcal{E}$ = {O ⊆ ℝ : ∀ a ∈ O , ∃ $\epsilon > 0$, *s.t* $a \in (a - \epsilon , a + \epsilon)$ ⊆ O }.

Definition 0.17:

An ordered topological space is a triple (X $,\tau, \leq$) where (X $,\tau$) is a topological space and (X , $\leq$) is a totally order set .

Remark:

Infinite intersection of open sets may not be open set .Take the usual real line ($\mathbb{R}$, $\mathcal{E}$) . The sets $\left(\frac{-1}{n}, \frac{1}{n}\right) : n \in \mathbb{Z}^{+}$ are open in ($\mathbb{R}$, $\mathcal{E}$) ,

but $\bigcap_{n=1}^{\infty} \left(\frac{-1}{n}, \frac{1}{n}\right) = \{0\}$ is not open in ($\mathbb{R}$, $\mathcal{E}$) .

Definition 0.18:

β is a base for a topology τ if :

(i) $\beta \subseteq \tau$

(ii) $\tau = \{ \cup \beta' \mid \beta' \subseteq \beta \}$ [4, p.12].

Examples 0.5:

(i) $\{\{x\} : x \in X\}$ is basis for the discrete topology on X .

(ii) {X} is basis for the indiscrete topology on X .

(iii) The set of open balls $\{B_d(a, \epsilon) : a \in X$ and $\epsilon > 0 \}$ is basis for the metric topology on X .

***Theorem 0.1* [4]:**

Let X be a set and $\beta \subseteq 2^X$, then β is a basis for unique topology on X iff :

(i) For each $x \in X$ there exist , $B \in \beta$ such that $x \in B \subseteq X$.

(ii) For any B_1 , $B_2 \in \beta$ if $x \in B_1 \cap B_2$, then there is $B_3 \in \beta$ such that

$x \in B_3 \subseteq B_1 \cap B_2$.

Definition 0.19:

Let (X ,τ) be a topological space , ξ is a subbasis for τ on X if :

(i) $\xi \subseteq \tau$.

(ii) Finite intersection of members of ξ form a basis for τ .

Members of ξ are called subbasic open sets [4, p .15] .

Remark:

Let $O \in \tau$ for any $x \in O$, there are $S_1 , S_2, \ldots , S_n \in \xi$ *s.t* $x \in \bigcap_{i=1}^{n} S_i \subseteq O$.

Theorem 0.2:

Let X be a set and $\xi \subseteq 2^X$, ξ forms a subbasis for a topology on X ,

moreover this topology is unique and it is the smallest topology contains ξ.

Sets in Spaces

Definition 0.20:

Let X be a space and $A \subseteq X$, a point $x \in X$ is called an interior point of A if there exist an open set O such that $x \in O \subseteq A$. The set of all interior point of A is denoted by A°.

Theorem 0.3:

1- $A^{\circ} \subseteq A$.

2- A° is open .

3- A is open iff $A = A^{\circ}$.

4- $(A^{\circ})^{\circ} = A^{\circ}$.

5- if $A \subseteq B$, then $A^{\circ} \subseteq B^{\circ}$.

6- $(A \cap B)^{\circ} = A^{\circ} \cap B^{\circ}$.

7- $A^{\circ} \cup B^{\circ} \subseteq (A \cup B)^{\circ}$ [4 , p. 6] .

Definition 0.21:

Let X be a space and $A \subseteq X$, a point $x \in X$ is called a closure point of A if for any open set $O \ni x$, we have that $A \cap O \neq \emptyset$ .The set of all closure points of A is denoted by $\overline{A}$.

Theorem 0.4:

1- $A \subseteq \overline{A}$.

2- $\overline{A}$ is closed .

3- A is closed iff $A = \overline{A}$.

4- $\overline{\overline{A}} = \overline{A}$.

5- if $A \subseteq B$, then $\overline{A} \subseteq \overline{B}$.

6- $\overline{(A \cap B)} \subseteq \overline{A} \cap \overline{B}$.

7- $\overline{(A \cup B)} = \overline{A} \cup \overline{B}$.

Definition 0.22:

Let X be a space and $A \subseteq X$, a point $x \in X$ is called a boundary point of A if for any open set $O \ni x$,we have that $A \cap O \neq \emptyset$ and $O \cap (X - A) \neq \emptyset$. The set of all boundary points of A is denoted by ∂A .

Theorem 0.5 :

1- $\partial A = \overline{A} \cap \overline{(X - A)}$.

2- $\partial A = \partial (X - A)$.

3- ∂A is closed .

4- $\overline{A} = A \cup \partial A$.

5- A is closed iff $\partial A \subseteq A$.

6- $\partial A = \overline{A} - A^{\circ}$.

Definition 0.23:

Let X be a space and $A \subseteq X$, a point $x \in X$ is called a limit point of A if for any open set $O \ni x$, we have that $(O - \{x\}) \cap A \neq \emptyset$.

The set of all limit points of A is denoted by A'.

Theorem 0.6 :

1 - $\overline{A} = A \cup A'$.

2- A is closed iff $A' \subseteq A$.

3 - if $A \subseteq B$, then $A' \subseteq B'$.

4 - $(A \cup B)' = A' \cup B'$.

5 - $(A \cap B)' \subseteq A' \cap B'$.

Definition 0.24:

Let X be a space and $A \subseteq X$, a point $x \in X$ is called an exterior point of A if there exist an open set $O \ni x$ such that $O \subseteq (X - A)$. The set of all exterior points of A is denoted by A^{ext} .

Theorem 0.7 :

1- A^{ext}, A°, ∂A are pair wise disjoint and $A^{\circ} \cup \partial A \cup A^{ext} = X$

2- $A^{ext} = (X - A)^{\circ}$ and thus A^{ext} is open [4 , p .11] .

Definition 0.25:

Let X be a space and A ⊆ X , a point $x \in X$ is called isolated point of A if there exist an open set O such that $O \cap A = \{x\}$. The set of all isolated points of A is denoted by A^{iso}.

Theorem 0.8 :

1) $A^{iso} \subseteq A$.

2) $A^{iso} \cap A' = \emptyset$.

3) $A^{iso} = A - A'$.

Separation Axioms

Definition 0.26:

A topological space (X , τ) is called T_0 -space if for every two distinct points $x , y \in X$, there is an open set O , such that either :

$x \in O$ and $y \notin O$ or $x \notin O$ and $y \in O$.

Examples 0.6 :

(i) Right ray topology over R is T_0 -space

$\tau_{\text{right}} = \{(a , \infty) : a \in \mathbb{R} \} \cup \{\emptyset , \mathbb{R}\}$.

(ii) **Sierpiński** space is T_0 -space .

$X = \{ a , b\}$, $\tau = \{\emptyset , \{a\} , X \}$.

(iii) Indiscrete space is not T_0 –space .

Theorem 0.9:

X is T_0–space iff for any $a , b \in X$; $\overline{\{a\}} = \overline{\{b\}}$, implies that $a = b$

Proof :

If X is T_0 -space , let $a \neq b$, then $a \notin \overline{\{b\}}$, but $a \in \overline{\{a\}}$, and hence $\overline{\{a\}} \neq \overline{\{b\}}$.

Conversely

Let $a \neq b$, then $\overline{\{a\}} \neq \overline{\{b\}}$, Take $O = X - \overline{\{a\}}$, then O is open and $O \ni b$, $O \not\ni a$.

Definition 0.27:

A topological space (X , τ) is called T_1- space if for every two distinct points $x , y \in X$, there exist two open sets O_1 and O_2 , such that $x \in O_1$, $y \notin O_1$ and $x \notin O_2$, $y \in O_2$.

Examples 0.7:

(i) The cofinite topology $\tau=\{O \subseteq X : X - O \text{ is finite}\} \cup \{\emptyset\}$.

Let $x \neq y$ in X , Take $O_1 = \{x\}^c$, and $O_2 = \{y\}^c$, then O_1 and O_2 are open and $y \in O_1$, $x \notin O_1$ and $y \notin O_2$, $x \in O_2$.

(ii) The metric topological space (X, T_d) is T_1- space

let $x \neq y$ in X , Take $O_1 = B_d(x, \frac{\epsilon}{2})$, and $O_2 = B_d(y, \frac{\epsilon}{2})$,where

$\epsilon = d(x, y) > 0$, $x \in O_1$, $y \notin O_1$ and $x \notin O_2$, $y \in O_2$.

Theorem 0.10:

A space (X, τ) is T_1- space iff for every $x \in X$, $\{x\}$ is closed .

Corollary 0.1:

A space (X, τ) is T_1- space iff every finite subset is closed .

Definition 0.28:

A topological space (X, τ) is called Hausdorff space (T_2- space) if for every two distinct points $x, y \in X$, there exist two open sets O_1 , O_2 , such that $x \in O_1$ and $y \in O_2$ and $O_1 \cap O_2 = \emptyset$.

Examples 0.8:

(i) Metric topology is T_2- space

Let $x, y \in X$ with $x \neq y$ and let $\epsilon = d(x, y) > 0$.

Take $O_1 = B_d(x, \frac{\epsilon}{2})$ and $O_2 = B_d(y, \frac{\epsilon}{2})$;

Then $x \in O_1$ and $y \in O_2$ and $O_1 \cap O_2 = \emptyset$.

(ii) Infinite cofinite topology is not T_2

$x, y \in X$; $x \neq y$, suppose that X is T_2- space .Then there exist two open sets O_1 and O_2 , such that :

$O_1 \ni x$,$O_2 \ni y$ and $O_1 \cap O_2 = \emptyset$.

Then $O_1^c \cup O_2^c = X$, but O_1^c and O_2^c are finite sets , then X is finite ,

hence X is not T_2- space .

Definition 0.29:

A topological space (X , τ) is called regular if for every point $x \in X$ and closed subset $F \subseteq X$ with $x \notin F$ there are two open sets O_1 , O_2 such that $x \in O_1$ and $F \subseteq O_2$ and $O_1 \cap O_2 = \emptyset$.

(X , τ) is called T_3- space iff its regular space and T_1 - space .

Examples 0.9:

(i) $X = \{a , b, c\}$, $\tau = \{\emptyset , \{a\} , \{b , c\} , X \}$

is regular space but is not T_1- space since $\{b\}$ is not closed ,

then (X , τ) is not T_3 -space .

(ii) Metric topological space (X ,τ_d) is regular and T_3- space

let $x \in X$, $F \subseteq X$ is closed set , and $x \notin F$.

Take $\epsilon = d(x, F) = \min\{d(x, y): y \in F\}$, and take $O_1 = B_d(x, \frac{\epsilon}{4})$ and

$O_2 = \bigcup_{y \in F} B_d(y, F)$, then O_1 and O_2 are open sets and $x \in O_1$ and $F \subseteq O_2$

and $O_1 \cap O_2 = \emptyset$, then (X, τ) is T_3- space .

Definition 0.30:

A topological space (X, τ) is called Normal space if for any two disjoint closed sets F_1 and F_2 of X , there are two open sets O_1 and O_2 such that $F_1 \subseteq O_1$, $F_2 \subseteq O_2$ and $O_1 \cap O_2 = \emptyset$.

A topological space is called T_4- space if it is Normal and T_1 - space .

Examples 0.10:

(i) $X = \{a, b, c, d\}$, $\tau = \{\emptyset, \{a, b\}, \{c, d\}, X\}$

(X, τ) is Normal , but not T_1- space since $\{a\}$ is not closed .

(ii) Metric topological space is T_4- space .

Continuous Functions and Homeomorphisms

Definition 0.31:

Let X and Y be two spaces A function $f: X \longrightarrow Y$ is continuous if for for every open set O in Y, $f^{-1}(O)$ is open in X .

A continuous function is called " map " [4 , p .31].

Theorem 0.11* [4]*:

f : X $\longrightarrow$Y is continuous iff $\overset{-1}{f}$(E) is closed in X for any closed set E in Y .

Definition 0.32:

A homeomorphism (Topological transformation) is a bijective map and its inverse is a map [4, p.35] .

Notation:

If there is a homeomorphism from X onto Y, we say that X and Y are homeomorphic or topologically equivalent , it is denoted by X $\cong$ Y.

Theorem 0.12:

$\cong$ is an equivalent relation on spaces .

Definition 0.33:

If (X , τ_1 , $\leq_1$) and (Y ,τ_2 , $\leq_2$) ordered topological spaces then a map f: X $\longrightarrow$Y is an order-homeomorphism if it is an order isomorphism of posets and a homeomorphism of topological spaces [5 , p .218].

Theorem (The pasting Lemma) 0.13:

Let X = A $\cup$ B , where A and B are closed in X , let f : A $\longrightarrow$ X and g:B $\longrightarrow$Y be continuous . IF $f(x) = g(x)$ for every $x \in$ A $\cap$ B , then f and g combine to give a continuous function h:X $\longrightarrow$ Y , defined by setting $h(x) = f(x)$ if $x \in$ A , and $h(x) = g(x)$ if $x \in$ B [3 , p. 108].

Note:

Theorem 0.13 is also hold if A and B are both open sets .

Notation:

If X and Y are two topological spaces , let

$Y^X = \{ f \mid f : X \longrightarrow Y ,\ f$ is continuous function $\}$.

<u>Compactness</u>

<u>Definition 0.34:</u>

A space X is called compact space if every open cover of X can be reduced to a finite subcover .

i.e. , if $X = \bigcup_{\alpha \in J} O_\alpha$, then there exist a finite subcover $\{O_{\alpha_i}\}_{i=1}^{n}$ such tahat $X = \bigcup_{i=1}^{n} O_{\alpha_i}$ [4 , p .139] .

<u>Examples 0.11:</u>

(i) Any Indiscrete space is compact .

(ii)Any closed bounded subset of $\mathbb{R}^n$ is compact (Hiene-Borle Theorem).

<u>Theorem (Tychonov) 0.14</u> **<u>[4]</u>*****<u>:</u>***

Let (X_α , τ_α) be a topological spaces for any $\alpha \in$ **J** , then $\prod_{\alpha \in J} X_\alpha$ is compact iff (X_α , τ_α) is compact topological space for each $\alpha \in$ **J** .

Corollary 0.2:

Compactness is a topological property.

Definition 0.35:

A topological space X is locally compact space if for any $x \in$ X there is compact neighborhood of x [4 , p .154] .

Examples 0.12:

(i) $\mathbb{R}^{\boldsymbol{n}}$ is locally compact space .

(ii) Any compact space is locally compact .

Connectivity and Path Connectivity

Definition 0.36:

O_1 and O_2 Separate a topological space X if :

(i) O_1 and O_2 are both open .

(ii) O_1 and O_2 are both non-empty .

(iii) $O_1 \cap O_2 = \emptyset$.

(iv) $O_1 \cup O_2 = X$ [4 , p .119].

Definition 0.37:

A space X is connected if there do not exist non-empty proper sets O_1 and O_2 which separate X [4 , p .119] .

Examples 0.13:

The following spaces are connected .

(i) Sierpinski space.

(ii) Any indiscrete space.

(iii) The usual real line ($\mathbb{R}$, $\mathcal{E}$) .

Note:

If X is not connected , we generally say that X is disconnected .

Example 0.14:

The discrete topology over a set with more than one point is disconnected **.**

***Theorem 0.15* [4]:**

Let (X_α , τ_α) be a topological spaces for any $\alpha \in$ **J**, then $\prod_{\alpha \in J} X_\alpha$ is connected iff (X_α , τ_α) is connected for each $\alpha \in$ **J** .

Corollary 0.3

Connectedness is topological property.

Definition 0.38:

Let $x \in X$ .Then $C_x = \bigcup\{A |\, x \in A \subseteq X$ and A is connected$\}$, C_x is called the connected component of x .

Examples 0.15:

(i) In the discrete space $C_x = \{x\}$ for any $x \in X$.

(ii) In the usual real line ($\mathbb{R}$, $\mathcal{E}$) C_x = $\mathbb{R}$ for any $x \in \mathbb{R}$.

Definition 0.39:

A topological space X is said to be locally connected at x if for every neighborhood O of x , there is a connected neighborhood U of x contained in O. If X is locally connected at each of its points then it is said to be locally connected [3 , p.161] .

Example 0.16:

Each interval in the usual real line is locally connected .

Definition 0.40:

A function p: I⟶X is called a path in X if p is continuous [4 , p .131] .

Definition 0.41:

A map p: I⟶X is called a path from x to y in X if p is a path in X and $p(0) = x$ and $p(1) = y$, x is called initial point and y is called terminal point of p [4 , p. 131] .

Definition 0.42:

A loop p based at $x \in$ X is a path in X if $p(0) = p(1)= x$, *i.e.*, the beginning point and the terminal point are equal [3, p. 326].

Definition 0.43:

A space X is path connected if for any x , $y \in$ X there is a path joining them in X [3 , p. 155] .

Definition 0.44:

If p is a path in X from x into y , and if g is a path in X from y to z , we define the composition $p * g$ of p and g to be the path h given by the equation

$$h(t) = \begin{cases} p(2t) & \text{for all } t \in [0 , \frac{1}{2}] \\ g(2t-1) & \text{for ll } t \in [\frac{1}{2} , 1] \end{cases} .$$

The function h is well defined and continuous by pasting lemma .

Example 0.17:

($\mathbb{R}$, $\mathcal{E}$) is path connected space .

Remark:

A connected space may not be path connected [4 , p .131 . Example 2.4].

***Theorem 0.16* [4]:**

Let X be path connected space then X is connected .

***Theorem 0.17* [4]:**

Let (X_α , τ_α) be a topological spaces for each $\alpha \in \mathbf{J}$ .Then $\prod_{\alpha \in J} X_\alpha$ is path connected iff (X_α , τ_α) is path connected for each $\alpha \in \mathbf{J}$.

Corollary 0.4 :

Path connectedness is a topological property.

Definition 0.45:

Let $x \in X$ .Then $H_x = \bigcup\{A \mid x \in A \subseteq X$ and A is path connected$\}$, H_x is called the path connected component of x [4, p. 134].

Definition 0.46:

A topological space X is said to be locally path connected at x

if for every neighborhood O of x , there is a path connected

neighborhood U of x contained in O. If X is locally path connected at each of its points then it is said to be locally path connected [3, p. 161]

Example 0.18:

(i) $\mathbb{R}^n$ is locally path connected [3, p .161] .

Quotient Spaces

Definition 0.47:

Let X and Y be a topological spaces, let $q : X \longrightarrow Y$ be a surjective

map . The map q is said to be a quotient map, provided a subset O of Y

is open in Y if $q^{-1}(O)$ is open in X [3, p .135] .

<u>Definition 0.48:</u>

If X is a space and A is a set , $p : X \longrightarrow A$ a surjective map , then there exists exactly one topology τ on A relative to which p is a quotient map , it is called the quotient topology induced by p [3 , p .136] .

<u>Definition 0.49:</u>

Let X be a topological space , and let X^* be a partition of X into disjoint subsets whose union is X . Let $p : X \longrightarrow X^*$ be a surjective map that carries each point of X to the element of X^* containing it .In the quotient topology induced by p , the space X^* is called a quotient space of X [3, p .136].

Chapter One

Finite Topological Spaces

In this chapter we introduce the very important properties of finite topological spaces that are different from the general topological spaces. There was a brief early flurry of beautiful mathematical works on this subject .Two independent papers , by J.P.May and Stong [1 , 7] , [7] published in 1966 , are especially interesting . We will work through them and also we create some computer procedures applications related to finite topological spaces in chapter three .

Basic Definitions

Now , if X is finite , then 2^X is finite and hence any topology τ on X will consist only finitely many open sets and hence we can introduce the following definition .

Definition 1.1:

A topological space for which the underlying point set X is finite is called a finite topological space . Finite topological space can be redefine by the following conditions:

(i) $\emptyset \in \tau$ and $X \in \tau$.

(ii) if $O_1 , O_2 \in \tau$, then $O_1 \cup O_2 \in \tau$, and

(iii) if $O_1 , O_2 \in \tau$, then $O_1 \cap O_2 \in \tau$ [1 , p.1] .

Note:

A finite topological space is a complete lattice .

Definition 1.2:

Let X be a finite topological space . For $x \in$ X , define

$U_x = \bigcap \{ O \subseteq X : O$ is open and $O \ni x \}$

U_x called minimal basic open set [1 , p.2] .

Example 1.1:

X = { a , b , c } & τ = { ∅ , {a} , {a , b} , {a , c } , X } . Then

$U_a = \{a\} \cap \{a , b\} \cap \{a , c\} \cap X = \{a\}$.

$U_b = \{a , b\} \cap X = \{a , b\}$.

$U_c = \{a , c\} \cap X = \{a , c\}$.

Definition1.3:

Let $\leq$ be relation on X defined by $x \leq y$ in X if $x \in U_y$ or , equivalently ,$U_x \subseteq U_y$,write $x < y$ if the inclusion is proper [1 , p.2] .

Lemma 1.1:

The set of open sets U_x is basis for X . Indeed , it is the unique minimal basis for X .

Proof :

Let $\boldsymbol{\mu}$ be the set of all U_x , then for any $x \in$ X there is $U_x \ni x$ and hence

$X = \bigcup_{x \in X} U_x$ *i.e.*, $\boldsymbol{\mu}$ cover X .

Let $x, y, z \in X$, if $z \in U_x \cap U_y$, then $z \in U_x$ and $z \in U_y$.Which implies that $U_z \subseteq U_x$ and

$U_z \subseteq U_y$, hence $z \in U_z \subseteq U_x \cap U_y$.

Now suppose that $\boldsymbol{\zeta}$ is another minimal basis , let $C \in \boldsymbol{\zeta}$ such that

$x \in C \subseteq U_x$, then $C = U_x$ since U_x is the smallest open set contain x ,

so that $U_x \in \boldsymbol{\zeta}$ for all $x \in X$ and hence $\boldsymbol{\zeta} = \boldsymbol{\mu}$.

Lemma 1.2:

A set β of non-empty subsets of X is the minimal base for a topology iff

(i) Members of β cover X .

(ii) The intersection of any two sets in β is a union of some sets in β .

(iii) If $B\alpha \in \beta$ for $\alpha \in \Delta$ and $\bigcup_{\alpha \in \Delta} B\alpha \in \beta$, then $\bigcup_{\alpha \in \Delta} B\alpha = B\acute{\alpha}$ for some $\acute{\alpha} \in \Delta$.

Proof :

Conditions (i) and (ii) are equivalent to saying that β is a basis , for (iii).
Suppose that β is a minimal basis , then $U_x \in \beta$ for all $x \in X$, and if

$U_x = \bigcup_{y \in U_x} U_y$, then $U_y \subseteq U_x$ for all $y \in U_x$ (1) .

Also we have that $x \in \bigcup_{y \in U_x} U_y$ which implies $U_x \subseteq U_y$ for some $y \in U_x$ (2) .

From (1) and (2) we get that $\bigcup_{y \in U_x} U_y = U_y$ for certain $y \in U_x$.

Conversely

Let $\boldsymbol{\mu}$ be a minimal basis and let $B \in \beta$, then $B = \bigcup_{x \in B} U_x$ for some $U_x \in \boldsymbol{\mu}$

(B is open) . And by condition (iii) $B = U_x$ for a certain $x \in B$, then B is a minimal basic open set and hence β is minimal basis .

Separation Properties

Lemma 1.3:

The relation $\leq$ is a preorder . It is a partial order iff X is T_0-space .

Proof :

The first statement is clear. For the second suppose that $(X , \leq)$ is a poset . Let $x \neq y$, then $x \not\leq y$ or $y \not\leq x$ which implies $x \notin U_y$ or $y \notin U_x$ then there exist an open set $U_y \ni y$ and $U_y \not\ni x$ or an open set $U_x \ni x$ and $U_x \not\ni y$ thus X is T_0- space .

Conversely

Suppose that X is T_0- space , let $x \leq y$ and $y \leq x$, then $U_x \subseteq U_y$ and $U_y \subseteq U_x$ which gives $U_x = U_y$, hence we must have that $x = y$.

Proposition 1.1:

For a finite set X , the topologies on X are in bijective correspondence with the reflexive and transitive relations $\leq$ on X . The topology corresponding to $\leq$ is T_0 if and only if the relation $\leq$ is a partial order[1, p.3].

Lemma 1.4:

Finite T_1- space is discrete space.

Proof :

Suppose that $X = \{ x_1 , x_2 ,.., x_n \}$ is a finite T_1 - space , take $x \in X$ then $\{ x \}^c$ must be finite subset of X . Then $\{ x \}^c$ is closed and hence any single point set $\{x\}$ is open , thus topology is discrete .

Remark:

Finite T_2 ,T_3 ,T_4 , spaces are obviously discrete .

Theorem 1.1:

Finite T_0- space has at least one closed single point .

Proof:

By using mathematical induction

if $|X| = 1$, that is $X=\{x\}$ the result is true . Assume that the result is true for $|X| = n - 1$. Now let $|X| = n$, let $A \subseteq X$, such that $|A| = n - 1$, then A is T_0- space by induction and there is a point $p \in A$ such that $\{ p\}$ is closed set in A , then $\{ p\} = A \cap F$ for some closed set F in X ,

then we will have $F=\{ p \}$ or $F=\{ p , x_n \}$ if $F=\{ p\}$ then it is done .
Hence we may assume that $F= \{ p , x_n \}$, let O_1 be open set in X such

that $p \in O_1$ and $x_n \notin O_1$.Then $O_1^c \cap F = \{x_n\}$ is closed in X.

Let O_2 be open set in X such that $p \notin O_2$ and $x_n \in O_2$, then $O_2^c \cap F=\{ p \}$ closed in X .

Remark:

There are infinite T_0- spaces which do not have any closed single point , for example right ray topology over $\mathbb{R}$.

Remark :

Non – discrete finite space can also be Normal .

<u>Example 1.2:</u>

Excluded point topology on any finite set . Let $X = \{x_1 , x_2 , x_3\}$

define a topology τ on X by $\tau = \{ O \subseteq X : x_3 \notin O\} \cup \{X\}$.

Then τ is the Excluded point topology on X ,

$\tau= \{\emptyset ,\{ x_1 \} ,\{ x_2 \} , \{x_1 , x_2\} , X\}$. It's clear that (X , τ) is Normal .

The only disjoint closed sets are $\emptyset$ and X and they are separated by themselves .

Number of topologies on a finite set

Topologies on a finite set are in one-to-one correspondence with preorders on the set, and T_0 topologies are in one-to-one correspondence with partial orders . Therefore the number of topologies on a finite set is equal to the number of preorders and the number of T_0 topologies is equal to the number of partial orders . The table below lists the number of distinct (T_0) topologies on a set with *n* elements. It also lists the number of inequivalent (i.e. nonhomeomorphic) topologies.

n	**Distinct topologies**	**Distinct T_0 topologies**	**Inequivalent topologies**	**Inequivalent T_0 opologies**
0	1	1	1	1
1	1	1	1	1
2	4	3	3	2
3	29	19	9	5
4	355	219	33	16
5	6942	4231	139	63
6	209527	130023	718	318
7	9535241	6129859	4535	2045
8	642779354	431723379	35979	16999
9	63260289423	44511042511	363083	183231
10	8977053873043	6611065248783	4717687	2567284
OEIS	A000798	A001035	A001930	A000112

Let *T*(*n*) denote the number of distinct topologies on a set with *n* points. There is no known simple formula to compute *T*(*n*) for arbitrary *n*. The Online Encyclopedia of Integer Sequences presently lists *T*(*n*) for ≤18.

The number of distinct T_0 topologies on a set with n points , denoted $T_0(n)$, is related to $T(n)$ by the formula $T(n) = \sum_{k=0}^{n} S(n,k)\, T_0(n, k)$ [10] .

where $S(n , k)$ is Stirling number of the second kind which is the the number of ways to partition a set of n labelled objects into k non empty unlabelled subsets

$$S(n , k) = \frac{1}{k!}\sum_{j=0}^{k}(-1)^{j} \binom{k}{j} (k-j)^{n} \quad [11] .$$

Continuous Functions and Homeomorphisms

Lemma 1.5 :

A function $f : X \longrightarrow Y$ is continuous iff it is order preserving *i.e.* , if $x \leq y$ in X then $f(x) \leq f(y)$ in Y .

Proof :

Let f be a continuous function , suppose that $x \leq y$ in X , then $x \in U_y \subseteq f^{-1}(U_{f(y)})$ and thus $f(x) \in U_{f(y)}$ which implies that $f(x) \leq f(y)$.

Conversely

Let O be open set in Y if $f(y) \in O$, then $U_{f(y)} \subseteq O$. If $x \in U_y$, then $x \leq y$ and thus $f(x) \leq f(y)$ and $f(x) \in U_{f(y)} \subseteq O$. So that $x \in U_y \subseteq f^{-1}(O)$, then $f^{-1}(O) = \bigcup_{y \in f^{-1}(O)} U_y$, therefore f is continuous .

Lemma1.6:

A map $f: X \longrightarrow X$ is a homeomorphism iff f is either one -to- one or onto .

Proof:

One-to-one and onto are equivalent by finiteness , since f is one-to-one

$A \rightarrow f(A)$ defines one-to-one correspondence $g : 2^X \longrightarrow 2^X$. If $g(A) \in \tau$,

$f(A)$ is open and by continuity and one-to-one nature of f, A is open. Since

τ is finite and $\tau \subseteq g(\tau)$, g gives one-to-one correspondence

$\tau \longrightarrow \tau$ .Thus A open implies $f(A)$ open and hence f is a homeomorphism

Conversely

If f is a homeomorphism then f is one – to –one and onto .

Note:

In infinite topological spaces Lemma 1.6 is not held and we will discuss the following example to explain that .

<u>Example 1.3</u>:

Let X be the set of integers with the topology τ created by declaring a set to be open if either it is a subset of $\mathbb{N}$ or it is the entire set $\mathbb{Z}$ *i.e.*,

$\tau = \{O : O \subseteq \mathbb{N}\} \cup \{\mathbb{Z}\}$ .Then the map $f : X \longrightarrow X$ such that $f(x) = x - 1$,

is a continuous bijection from X to itself , but it is not a homeomorphism

since the image of $\mathbb{N}$ is not open .

Compactness

1- Every finite space is compact .

2- A compact discrete space is a finite space .

Proof :

Let $X = \bigcup_{i \in I} \{ x_i \}$ *i.e.,* $\{\{x_i\}\}_{i \in I}$ is an open cover of X .Since X is compact , then $X=\bigcup_{i=1}^{n} \{x_i\}$ and hence $X = \{x_1 , x_2 , . . . , x_n\}$.

Definition 1.3 :

A space X is called smally compact if every open subset is locally compact [1 , p .9] .

Theorem 1.2 :

Every finite space is smally compact space .

Proof :

Since every open subset is compact (since it's finite) then every open subset is locally compact .

Definition 1.4:

If Y is a finite space , then the point wise ordering $\leq$ on Y^X is given by $f \leq g$ if $f(x) \leq g(x)$ for all $x \in X$ [1 , p . 9] .

Definition 1.5:

A compact open topology (C-O-topology) on Y^X is a topology in which the subbasis are the sets $W(C, O) = \{ f / f(C) \subseteq O \}$ where C is compact in X and O is open in Y [1 , p. 9] .

Lemma 1.7 :

If X and Y are finite spaces , then

$\bigcap \{O \subseteq Y^X : O \text{ is open and } O \ni g\} = \{ f / f \leq g \}$.

Proof :

Let $V_g = \bigcap \{O \subseteq Y : O \text{ is open and } O \ni g\}$ and $Z_g = \{ f / f \leq g \}$ and let $x \in X$, let $f \in V_g$. Since $g \in W(\{x\}, U_{g(x)})$, $W(\{x\}, U_{g(x)})$ is open w. r. t C-O-topology then $f \in W(\{x\}, U_{g(x)})$, so $f(x) \in U_{g(x)}$, there for $f \leq g$, then $V_g \subseteq Z_g$.

Conversely

Let $f \leq g$ Take $W(C, O) \ni g$ then $g(x) \in O$ for some $x \in C$

since $f(x) \leq g(x)$ then $f(x) \in U_{g(x)}$ and hence $f(x)$ is in any open set containing g and hence $Z_g \subseteq V_g$.

Connectivity and Path Connectivity

Lemma 1.8:

If $x \leq y$ in X . Then there is a path connecting x and y .

Proof :

Define P: I $\longrightarrow$ X by $P(t) = x$ for all $t \in [0, 1)$ and $P(1) = y$, let O be open in X if $x \in$ O and $y \notin$ O then P^{-1}(O) = $[0, 1)$ which is open in I w.r.t usual topology if $x, y \in$O then P^{-1}(O) = I , if $y \in$ O , then $x \in$ O ($x \in U_y$ since $x \leq y$) then P^{-1}(O) = I , if $x \notin$ O , then P^{-1}(O)=∅ and hence P is continuous .

Lemma 1.9:

Each U_x is path connected .

Proof :

Let $y_1, y_2 \in U_x$ then $y_1 \leq x$ and $y_2 \leq x$ (def of U_x) then by Lemma 1.8 there are two paths $p: y_1 \longrightarrow x$ and $q: y_2 \longrightarrow x$

Now , if we take the inverse path of q which denoted by q^{-1}, in the following path composition

defined by

$$p * q^{-1}(t) = \begin{cases} p(2t) & \text{for all } t \in [0, \frac{1}{2}] \\ q^{-1}(2t-1) & \text{for ll } t \in [\frac{1}{2}, 1] \end{cases} .$$

Which is continuous by pasting lemma and connecting y_1 and y_2

and hence U_x is path connected .

Lemma 1.10:

If X is a finite connected space and $x , y \in$ X , then there is either an increasing or decreasing sequence of points $\{p_i\}_{i=1}^{s}$ connecting x and y ($i.e.$, $x = p_1 \leq p_2 \leq \ldots \leq p_s = y$ or $x = p_1 \geq p_2 \geq \ldots \geq p_s = y$) .

Proof :

Let $x \in$ X and let O be a proper open subset of X containing x suppose that O = { $y \in$ X : y connecting x by some sequence $\{p_i\}_{i=1}^{s}$ } .

Now if $\{p_i\}_{i=1}^{s}$ is decreasing sequence , then $y \in U_x \subseteq$ O for all $y \in$ O which is implies that $U_x =$ O , if $'y \notin$ O , then neither is any point of $U_{'y}$, then O^c is open and hence O is clopen , since X is connected then
we must have that X = O , similarly if $\{p_i\}_{i=1}^{s}$ is increasing sequence of points .

Lemma 1.11:

Every finite space X is locally path connected .

Proof :

Let $x \in$ X suppose that O is an open set such that $x \in$ O then $x \in U_x \subseteq$ O where U_x is path connected by Lemma 1.9 and hence X is locally path connected .

Theorem 1.3 :

A connected finite space is path connected space .

Proof :

By Lemma 1.8 and Lemma 1.10.

Chapter Two
Alexandroff Space

In this chapter we study spaces that have topologies which satisfy a stronger condition . Namely, arbitrary intersections of open sets are open with this restriction , we lose important spaces such as Euclidean spaces, but the specialized spaces in turn display interesting properties .

Basic Definitions:

Definition 2.1:

Let X be a topological space , then X is an Alexandroff space if arbitrary intersections of open sets are open [2 , p. 465] .

Note:

We will denoted to Alexandroff space by A- space [1 , p.5] .

Lemma 2.1 :

Any finite space is an A-space.

Proof:

It's clear by property (iii) in the Definition of finite topological spaces in the previous chapter .

Lemma 2.2:

Any discrete topological space is an A-space .

Proof :

Let $\{ O\alpha \}_{\alpha \in J}$ be a family of open sets , let $x \in \bigcap_{\alpha \in J} O\alpha$, then $x \in O\alpha$ for all

$\alpha \in$ **J ,** and then $x \in \{x\} \subseteq O\alpha$ for all $\alpha \in$ **J** , and then

$x \in \{x\} \subseteq \bigcap_{\alpha \in J} O\alpha$, and hence $\bigcap_{\alpha \in J} O\alpha$ is open .

Some examples of A-spaces .

<u>Example 2.1:</u> (*Disjoint Minimal Open Neighborhoods*)

Take $X = \mathbb{R} \setminus \mathbb{Z}$ and $\beta = \{ (n , n+1) : n \in \mathbb{Z} \}$. Then X is an Alexandroff space with $U_x = (n , n+1)$ where $n < x < n+1$. For any two minimal open neighborhoods $U_x \neq U_y$ we have that $U_x \cap U_y = \emptyset$.

<u>Lemma 2.3:</u>

Let X be a metric space , then X is an A-space iff X has the discrete topology .

Proof :

Let $x \in X$, then the open balls $B_d(x , \frac{1}{n})$ with radius $\frac{1}{n}$, and centre x

$n \in \mathbb{N}$ are open in X , since X is an A-space $\bigcap_{n=1}^{\infty} B_d(x , \frac{1}{n})$ is an open set ,

But by the properties of metric space we have that $\bigcap_{n=1}^{\infty} B_d(x , \frac{1}{n}) = \{x\}$,

so we have shown that singletons are open , hence X has the discrete topology.

Conversely

The reverse direction follows from Lemma 2.2.

<u>Theorem 2.1 :</u>

X is an A-space iff each point in X has minimal basic open set .

Proof:

Suppose that X is an A-space , let $x \in$ X then $U_x = \bigcap\{O \subseteq X : O$ is open - and $x \in O\}$ is an open set , since X is an A- space .

Conversely

Suppose that each point $x \in$ X has minimal basic open set U_x . Consider an arbitrary intersection of open sets $V = \bigcap_{\alpha \in \Delta} O\alpha$, where each $O\alpha$ is open in X if $V = \emptyset$, then we are done . But if $V \neq \emptyset$, then pick $x \in V$ and then $x \in O\alpha$ for all $\alpha \in \Delta$ and hence $U_x \subseteq O\alpha$ for all $\alpha \in \Delta$, since U_x is the minimal basic open set at x , therefore $x \in U_x \subseteq V$, and hence V is open .

<u>Theorem 2.2 :</u>

If β is a collection of subsets of X such that for each $x \in$ X there is a minimal set $m(x) \in \beta$ with $m(x) \ni x$, then β is a basis for a topology on X and X is an A- space with this topology , In addition $U_x = m(x)$.

Proof:

It's clear that members of β cover X , suppose that B_1 , $B_2 \in \beta$, and

$x \in B_1 \cap B_2$, since $m(x)$ is minimal set containing x , so we have

$m(x) \subseteq B_1$, and $m(x) \subseteq B_2$, and hence $x \in m(x) \subseteq B_1 \cap B_2$, so β is basis for

topology on X , to show that X is an A-space with this basis , let $x \in$ X and

O be an open set in X such that O $\ni x$, then $O = \bigcup_{\alpha \in \Delta} B_\alpha$, where $B_\alpha \in \beta$, then

$x \in B_\alpha$ for some $\alpha \in \Delta$, there is $m(x) \subseteq B_\alpha \subseteq O$, hence $m(x)$ is minimal basic open set containing x , therefore X is an A-space and $U_x = m(x)$ [6 ,p. 2] .

<u>Example 2.2</u>: (An Alexandroff Topology on $\mathbb{R}^n$)

Take X to be $\mathbb{R}^n$ and let $\beta = \{\overline{B(0,r)} : r \in \mathbb{R}_+ \cup \{0\}\}$. Note that $\overline{B(0,r)}$ is the closed ball with center 0 and radius r and that $\overline{B(0,0)} = \{0\}$. If $x \in$ X then $\overline{B(0,|x|)}$ is a minimal set in β containing x . β is a basis for an Alexandroff topology on X .

<u>Theorem 2.3:</u>

If B is a subspace of an A-space X then B is an A-space .

Proof :

Let $x \in$ B and suppose that U is an open set in B with $x \in$ U , then

$U = B \cap O$, where O is open in X , this mean that $U_x \subseteq O$, so that

$B \cap U_x \subseteq B \cap O = U$, hence B is an A-space by Theorem 2.1 .

Theorem 2.4 :

If X and Y are A-spaces , then X×Y is also an A-space .

Proof:

X×Y has as basis $\beta = \{U \times V$: U is open in X and V is open in Y$\}$, let

$(x , y) \in X\times Y$, then $U_x \times U_y \in \beta$, and then claim that this is a minimal set

in β containing (x , y) . If $(x , y) \in U \times V \in \beta$, then $x \in U$ and $y \in V$,
so $U_x \subseteq U$ and $U_y \subseteq V$. Therefore $U_x \times U_y \subseteq U \times V$ and hence by

Theorem 2.2 X×Y is an A-space .

Separation Properties

Theorem 2.5:

X is a Hausdorff A-space iff for any $x \neq y$ in X we have $U_x \cap U_y = \emptyset$.

Proof:

Suppose that X is a Hausdorff A-space and let $x \neq y$ in X , then there are

two open sets U and V such that $U \ni x$ and $V \ni y$ and $U \cap V = \emptyset$,

since $U_x \subseteq U$ and $U_y \subseteq V$, hence $U_x \cap U_y = \emptyset$.

Conversely

This is trivial , suppose for any $x \neq y$ in X we have $U_x \ni x$ and $U_y \ni y$,

such that $U_x \cap U_y = \emptyset$, then X is a Hausdorff A-space .

Corollary 2.1:

X is a Hausdorff A-space iff X is discrete .

Proof:

Suppose that X is Hausdroff , then we claim that $U_x = \{x\}$ to see this

suppose $y \in U_x$, then $U_y \subseteq U_x$ and thus $U_x \cap U_y = U_y$, and since $U_y \neq \emptyset$,

then by Theorem 2.5 we must have $y = x$ and have $\{x\}$ is open in X , so X is discrete .

Conversely

If X is discrete , then it is Hausdroff space .

Continuous Functions and Homeomorphisms

Note:

A continuous image of an A-space may not be an A-space [6 , p.7] .

Example 2.3:

Let $X = \mathbb{N}$ with discrete topology and Let $Y = \mathbb{Q}$ with subspace topology

from ($\mathbb{R}$, ε) . Pick a bijection $f : \mathbb{N} \longrightarrow \mathbb{Q}$, then f is continuous , since

the domain X is discrete but $f(\mathbb{N}) = \mathbb{Q}$ is not an A-space [6 , p.7] .

Theorem 2.6 :

Let f: X ⟶ Y be an open and continuous function if X is an A-space , then so is f(X) . In addition if $y \in f$(X) then $U_y = f(U_x)$, where $f(x) = y$.

Proof:

Let $y \in f$(X) and let $x \in$ X such that $f(x) = y$, since U_x is open in X , then

$f(U_x)$ is open in f(X) , because f is open function , suppose that $y \in$ O for

some open set O in f(X) , then $x \in f^{-1}(O)$ in X , where

$f^{-1}(O)$ is open in X , since f is continuous , and we have $U_x \subseteq f^{-1}(O)$,

therefore $f(U_x) \subseteq$ O and hence y has a minimal basic open set ,

then by Theorem 2.1 f(X) is an A-space with $f(U_x) = U_y$ [6 , p . 7] .

Corollary 2.2:

If X is homeomorphic to Y and X is an A-space then so is Y .

Proof :

If a function f is homeomorphism between two spaces X and Y , then f is open and continuous with f(X) =Y and by Theorem 2.6 Y is an A-space

Compactness

Theorem 2.7 :

If X is an Alexandroff space , then U_x is compact for all $x \in$ X .

Proof:

Let $\{O_\alpha\}_{\alpha \in \Delta}$ be an open cover of U_x. Then $x \in O\alpha$ for some $\alpha \in \Delta$. So we must have $U_x \subseteq O\alpha$. Hence, $\{O\alpha\}$ is a finite subcover of $\{O_\alpha\}_{\alpha \in \Delta}$.

<u>Quotient Spaces</u>

<u>Theorem 2.8:</u>

If X is an A-space, then the quotient space $X/\sim$ is also an A-space.

Proof :

Let q: $X \longrightarrow X/\sim$ be the quotient map consider the arbitrary

$\bigcap_{\alpha \in \Delta} O\alpha$ of open sets in $X/\sim$ we have $q^{-1}(\bigcap_{\alpha \in \Delta} O\alpha) = \bigcap_{\alpha \in \Delta} q^{-1}(O\alpha)$

Now $q^{-1}(O\alpha)$ is open in X for all $\alpha \in \Delta$ because q is the quotient map, hence $\bigcap_{\alpha \in \Delta} q^{-1}(O\alpha)$ is open in X and therefore $\bigcap_{\alpha \in \Delta} O\alpha$ is open in $X/\sim$ by definition of quotient topology.

Chapter Three

Finite Topological Spaces with Maple

In this chapter we create procedures of Maple 15 to do computations of a lot of issues are related to finite topological spaces .

<u>The Procedures Used in Finite Topological Spaces :</u>

The following procedures have been improved:

1 - A procedure to get all possible intersections of a given subasis (S) .

Basis (S) ;

2 - A procedure to generate a Topology by a basis(B) .

Topology (B) ;

3 - A procedures to check if (T) is a topology over X or not .

(i) CloseIntersection(T) ; (ii) CloseUnion(T) ; (iii) IsTopology(T) ;

4 - A procedure to find the clopen sets of the topology (T) .

CO (X , T) ; [8] .

5 - A procedure to find the closed sets of the topology (T) .

CLO (X , T) ; [8] .

6- A procedure to obtain the relative topology on subset of X .

subspace (A , X , T) ; [8] .

7- A procedure to check if a given topology is connected .

isConn(X,T) ; [8] .

8- A procedure to find the connected components of a given point .

K(,X,T); [8] .

9- A procedure to check if a Topology is totally Disconnected .

isTotDisc(X,T);[8].

10 –A procedure to check if a Topology is T_0-spaces .

isT0(X,T); [8] .

11- A procedure to check if a given topology is T_1 – space .

isT_1 (X , T) ; [8] .

12 - A procedure to check if two spaces are homeomorphic or not .

ishomeo:=proc(t1,t2); [9] .

13- A procedure to find all inequivalent topologies on a finite set .

ishomeo:=proc(t1,t2); [9] .

The following procedures have been created in our study:

14 - A procedure to find the minimal basic open set at given point .

minbasic(,X,T);

15 - A procedure to find the minimal basis of a given space X .
minibasis(X,T);

16 - A procedure to find the connected components of a given space .

ALLCC(X,T);

17 - A procedure to check if a given point is a limit point or not .

IsLimitPoint(, A , X ,T) ;

18 - A procedure to find all limit points of given subset of X .

LimitPoints (A , X , T) ;

19 - A procedure to find the closure points of a given subset of X derived from the limit points .

ClosurePoints(A , X ,T) ;

20 - A procedure to find the boundary points of a given subset of X derived from the limit points .

BoundaryPoints(A , X ,T) ;

21 - A procedure to find the interior points of a given subset of X derived from the limit points .

InteriorPoints(A , X , T) ;

22 - A procedure to find the Exterior points of a given subset of X .

InteriorPoints(X-A , X ,T) ;

23 - A procedure to find the isolated points of a given subset of X by the definition .

IsolatedPoints(A , X ,T) ;

24 - A procedure to find the isolated points of given subset of X

derived from limit points .

IsolatedPoints2 (A , X , T) ;

25 - A procedure to find all topologies on a given set X .

AllTop(T);

26- A procedure to find all T_0 spaces on a given set X .

ALLT0(ALLTopologies);

27- A procedure to find all inequivalent T_0 topologies .

ishomeo:=proc(t1,t2);

The Implementations :

```
>restart;

with(combinat):

> #(1) A procedure to get all possible intersections of
 a given subasis(S).
 Basis:=proc(S)
local s,U,B;
if `subset`(S,powerset(X))then
U:=S;
for s in S do
U:=U union map(`intersect`,U,s);od;
B:=U union {X};#to add empty intersection;
else false;fi;
end:
```

```
#(2) A procedure to generate a Topology by a bsis(B).
> Topology:=proc(B)
local b,U,t;
U:=Basis(S);
for b in Basis(S) do
U:=U union map(`union`,U,b);
od;
t:=U union {{}};#to add the empty union;
end:
#(3)A procedures to check if T is a topology over X or not.
> CloseIntersection:=proc(T)
local A,U;
U:=T;
for A in T do
U:=U union map(`intersect`,U,A);
od;
if U=T then U; else CloseIntersection(U); fi;
end:
> CloseUnion:=proc(T)
local A,U;U:=T;
for A in T do
U:=U union map(`union`,U,A);
od;
if U=T then U; else CloseUnion(U);
fi;
end:
> IsTopology:=proc(T)

 CloseIntersection(T)=T and CloseUnion(T)=T
and member({},T) and member(X,T)
and `subset`(T,powerset(X));
end:
#(4)A procedure to find the clopen sets of the topology(T).
>CO:=proc(X,T)
local A,W;W:={};
for A in T do
if member(X minus A,T)then W:=W union{A};fi;
od;W;end:
```

```
#(5)A procedure to find the closed sets of the topology(T).
> CLO:=proc(X,T)
{seq(X  minus T[i],i=1..nops(T))};
end:

#(6) A procedure to obtain the relative topology on a subset
of X.

> subspace:=proc(A,X,T)
if `subset`(A,X) then
map2(`intersect`,A,T);
else false;
fi;
end:

#(7)A procedure to check that if the topology is connected.
> isConn:=proc(X,T)
evalb(CO(X,T)={X,{}});
end:

#(8)A procedure to find the connected components of
 a given point.
> K:=proc(x,X,T)
local i,S,SK;
if `member`(x,X)
then SK:={};
S:=map2(`union`,{x},powerset(X));
for i to nops(S) do
if isConn(S[i],subspace(S[i],X,T)) then SK:=SK union
S[i];fi;od; SK ;else flase;fi;end:

#(9)A procedure to check if a Topology is Totaly
Disconnected.
> isTotDisc:=proc(X,T)
local i;
for i to nops(X) do
if not(K(X[i],X,T)={X[i]}) then RETURN(false)fi;
od;RETURN(true);end:
```

```
#(10)A procedure to check if agiven topology is T0-space.

> isT0:=proc(X,T)
local x,y,O,test;
if nops(X)=1 then  true ;else
for x in X do
for y in X minus{x} do
for O in T do
test:=evalb((member(x,O)and not(member(y,O)))or (member(y,O)
and not(member(x,O))));
if test then break; fi;
od:
if not(test) then break;fi;
od:
if not(test) then break;fi;
od:
test;
fi;
end:
#(11) A procedure to check if a given topology is T1-space.
> isT1:=proc(X,T)
`subset`({seq({X[i]},i=1..nops(X))},CLO(X,T));
end:
#(14) A procedure to find the minimal basic open set for
  a certain point x .
minbasic:= proc(x,X,T)
local i,O,COUNT;
COUNT:={};
if member(x,X) then
for O in T do
if member(x,O)then COUNT:=COUNT union {O};
else COUNT:=COUNT;
fi;
od;
COUNT;
COUNT[1];
else false;
fi;
end:
```

```
#(15) A procedure to find the minimal basis of a given
  topological space.
minibasis:=proc(X,T)
local x,minimalbasis:={};
for x in X do
minimalbasis:=minimalbasis union {minbasic(x,X,T)};
od;
minimalbasis;end:

#(16) procedure to find the connected components of
  a given space .
> ALLCC:=proc(X,T)
local x,CC;
CC:={};
for x in X do
CC:=CC union {K(x,X,T)};
od;
CC;
end:
#(17)A procedure to check if a given point is a limit
   point or not.
>IsLimitPoint:=proc(x,A,X,T)
local i,o,L,Omx,O;
O:={};
L:={};
if member(x,X)= true then
for i to nops(T) do
if (member(x,T[i]))then O:= O union {T[i]};
else O:=O;
fi;
od;
Omx:={seq(O[i] minus {x},i=1..nops(O))};
for o in  Omx do
if ((o intersect A) <> {}) then L:=L union {x};
else L:={};
break;
fi;
od;
L;
```

```
else false;
fi;
end:
```

#(18)A procedure to find limit points of a given subset of X;

```
> LimitPoints:=proc(A,X,T)
local x,LI;
LI:={};
if `subset`(A,X)= true then
for x in X do
if IsLimitPoint(x,A,X,T) <> {} then LI:=LI union {x};
else LI:=LI;
fi;
od;
LI;
else false ;
fi;
end:
```

#(19)A procedure to find the closure points of a given subset of X derived from limit points .

```
> ClosurePoints:=proc(A,X,T)
A union LimitPoints(A,X,T);
end:
```

#(20)A procedure to find the boundary points of a given subset of X derived from limit points.

```
>BoundaryPoints:=proc(A,X,T)
ClosurePoints(A,X,T)intersect ClosurePoints(X minus A,X,T);
end:
```

#(21)A procedure to find the interior points of a given subset of X derived from limit points .

```
InteriorPoints:=proc(A,X,T)
ClosurePoints(A,X,T)minus BoundaryPoints(A,X,T);
end:
```

```
#(22)A procedure to find the Exterior points of a given
  subset of X derived from limit points.
> ExteriorPoints:= proc(A,X,T)
InteriorPoints(X minus A,A,X,T);
 end:
#(23)A procedure to find the isolated points of a given
subset by definition.

IsolatedPoints:=proc(A,X,T)
local O,x,iso;
iso:={};
for x in X do
for O in T do
if( member(x,O) and O intersect A ={x}) then iso:=iso union
{x};
fi;od;
od;iso; end:

#(24)A procedure to find all isolated points of a given
 subset of X derived from limit points.

> IsolatedPoints2:=proc(A,X,T)
A minus LimitPoints(A,X,T);
end:

# Examples:-
```

Indiscrete Space :

```
> X:={a,b,c} ;
```

$$X := \{a, b, c\}$$

```
> S:={{}};
```

$$S := \{\{\ \}\}$$

```
> B:=Basis(S);
```

$$B := \{\{\ \}, \{a, b, c\}\}$$

```
> T:=Topology(B);
```

$$T := \{\{\ \}, \{a, b, c\}\}$$

```
> IsTopology(T);
```

true

```
> A:={a};
```

$A := \{a\}$

```
> CLOPEN:=CO(X,T);
```

$CLOPEN := \{\{\,\}, \{a, b, c\}\}$

```
> CLOSED:=CLO(X,T);
```

$CLOSED := \{\{\,\}, \{a, b, c\}\}$

```
> subspace(A,X,T);
```

$\{\{\,\}, \{a\}\}$

```
> isConn(X,T);
```

true

```
> Cx:=K(b,X,T);
```

$Cx := \{a, b, c\}$

```
> isTotDisc(X,T);
```

false

```
> isT0(X,T);
```

false

```
> isT1(X,T);
```

false

```
> Ux:=minbasic(a,X,T);
```

$Ux := \{a, b, c\}$

```
> minimalbasis:=minibasis(X,T);
```

$minimalbasis := \{\{a, b, c\}\}$

```
> ALLCONNECTED_COMPONENTS:=ALLCC(X,T);
```

$ALLCONNECTED\_COMPONENTS := \{\{a, b, c\}\}$

```
> print(`The Number of the connected components is`
,nops(ALLCONNECTED_COMPONENTS));
```

The Number of the connected components is, 1

```
> IsLimitPoint(a,A,X,T);
```

$\{\,\}$

```
> LimitPoints(A,X,T);
```

$\{b, c\}$

```
> ClosurePoints(A,X,T);
```
$$\{a, b, c\}$$

```
> BoundaryPoints(A,X,T);
```
$$\{a, b, c\}$$

```
> InteriorPoints(A,X,T);
```
$$\{\,\}$$

```
> ExteriorPoints(A,X,T);
```
$$\{\,\}$$

```
> IsolatedPoints(A,X,T);
```
$$\{a\}$$

```
> IsolatedPoints2(A,X,T);
```
$$\{a\}$$

Discrete Space :

```
# Examples :-

> X:={a,b,c,d} ;
```
$$X := \{a, b, c, d\}$$

```
> S:={{a},{b},{c},{d}};
```
$$S := \{\{a\}, \{b\}, \{c\}, \{d\}\}$$

```
> B:=Basis(S);
```
$$B := \{\{\,\}, \{a\}, \{b\}, \{c\}, \{d\}, \{a, b, c, d\}\}$$

```
> T:=Topology(B);
```
$$T := \{\{\,\}, \{a\}, \{b\}, \{c\}, \{d\}, \{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}, \{a, b, c\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\}, \{a, b, c, d\}\}$$

```
>IsTopology(T);
```
true

```
> A:={a,c};
```
$$A := \{a, c\}$$

```
> CLOPEN:=CO(X,T);
```
$$CLOPEN := \{\{\,\}, \{a\}, \{b\}, \{c\}, \{d\}, \{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}, \{a, b, c\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\}, \{a, b, c, d\}\}$$

```
> CLOSED:=CLO(X,T);
```
$$CLOSED := \{\{\,\}, \{a\}, \{b\}, \{c\}, \{d\}, \{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}, \{a, b, c\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\}, \{a, b, c, d\}\}$$

```
> subspace(A,X,T);
```

$\{\{\,\}, \{a\}, \{c\}, \{a, c\}\}$

```
> isConn(X,T);
```

false

```
> Cx:=K(b,X,T);
```

$Cx := \{b\}$

```
> isTotDisc(X,T);
```

true

```
>isT0(X,T);
```

true

```
> isT1(X,T);
```

true

```
> Ux:=minbasic(a,X,T);
```

$Ux := \{a\}$

```
> minimalbasis:=minibasis(X,T);
```

$minimalbasis := \{\{a\}, \{b\}, \{c\}, \{d\}\}$

```
> ALLCONNECTED_COMPONENTS:=ALLCC(X,T);
```

$ALLCONNECTED\_COMPONENTS := \{\{a\}, \{b\}, \{c\}, \{d\}\}$

```
> print(`The Number of the connected components is`
,nops(ALLCONNECTED_COMPONENTS));
```

The Number of the connected components is, 4

```
> IsLimitPoint(a,A,X,T);
```

$\{\,\}$

```
> LimitPoints(A,X,T);
```

$\{\,\}$

```
> ClosurePoints(A,X,T);
```

$\{a, c\}$

```
> BoundaryPoints(A,X,T);
```

$\{\,\}$

```
> InteriorPoints(A,X,T);
```

$\{a, c\}$

```
>ExteriorPoints(A,X,T);
```

$\{b,d\}$

```
> IsolatedPoints(A,X,T);
```

$\{a, c\}$

```
> IsolatedPoints2(A,X,T);
```

$\{a, c\}$

#Sierpinski:

```
#Example:

> X:={a,b};
```

$X := \{a, b\}$

```
> S:={{b}};
```

$S := \{\{b\}\}$

```
> B:=Basis(S);
```

$B := \{\{b\}, \{a, b\}\}$

```
> T:=Topology(B);
```

$T := \{\{\ \}, \{b\}, \{a, b\}\}$

```
> IsTopology(T);
```

true

```
>  A:={b};
```

$A := \{b\}$

```
> CLOPEN:=CO(X,T);
```

$CLOPEN := \{\{\ \}, \{a, b\}\}$

```
> CLOSED:=CLO(X,T);
```

$CLOSED := \{\{\ \}, \{a\}, \{a, b\}\}$

```
> subspace(A,X,T);
```

$\{\{\ \}, \{b\}\}$

```
> isConn(X,T);
```

true

```
> Cx:=K(b,X,T);
```

$Cx := \{a, b\}$

```
> isTotDisc(X,T);
```

false

```
>isT0(X,T);
```

true

```
> isT1(X,T);
```

false

```
> Ux:=minbasic(a,X,T);
```

$Ux := \{a, b\}$

```
> minimalbasis:=minibasis(X,T);
```

$minimalbasis := \{\{b\}, \{a, b\}\}$

```
> ALLCONNECTED_COMPONENTS:=ALLCC(X,T);
```

$ALLCONNECTED\_COMPONENTS := \{\{a, b\}\}$

```
> print(`The Number of the connected components is`
,nops(ALLCONNECTED_COMPONENTS));
```

The Number of the connected components is, 1

```
> IsLimitPoint(b,A,X,T);
```

$\{\}$

```
>LimitPoints(A,X,T);
```

$\{a\}$

```
>ClosurePoints(A,X,T);
```

$\{a, b\}$

```
> BoundaryPoints(A,X,T);
```

$\{a\}$

```
> InteriorPoints(A,X,T);
```

$\{b\}$

```
> ExteriorPoints(A,X,T);
```

$\{\}$

```
> IsolatedPoints(A,X,T);
```

$\{b\}$

```
> IsolatedPoints2(A,X,T);
```

$\{b\}$

General Topological Space :

```
>#Examples:-
> X:={a,b,c,d} ;
```

$X := \{a, b, c, d\}$

```
> T:={{},{a},{a,b},{c},{a,c},{a,b,c},X };
```

$T := \{\{\ \}, \{a\}, \{c\}, \{a, b\}, \{a, c\}, \{a, b, c\}, \{a, b, c, d\}\}$

```
> IsTopology(T);
```

true

```
> A:={a,c};
```

$A := \{a, c\}$

```
> CLOPEN:=CO(X,T);
```

$CLOPEN := \{\{\ \}, \{a, b, c, d\}\}$

```
> CLOSED:=CLO(X,T);
```

$CLOSED := \{\{\ \}, \{d\}, \{b, d\}, \{c, d\}, \{a, b, d\}, \{b, c, d\}, \{a, b, c, d\}\}$

```
> subspace(A,X,T);
```

$\{\{\ \}, \{a\}, \{c\}, \{a, c\}\}$

```
> isConn(X,T);
```

true

```
> Cx:=K(b,X,T);
```

$Cx := \{a, b, c, d\}$

```
> isTotDisc(X,T);
```

false

```
>isT0(X,T);
```

true

```
> isT1(X,T);
```

false

```
> Ux:=minbasic(a,X,T);
```

$Ux := \{a\}$

```
> minimalbasis:=minibasis(X,T);
```

$minimalbasis := \{\{a\}, \{c\}, \{a, b\}, \{a, b, c, d\}\}$

```
> ALLCONNECTED_COMPONENTS:=ALLCC(X,T);
```

$ALLCONNECTED\_COMPONENTS := \{\{a, b, c, d\}\}$

```
> print(`The Number of the connected components is`
,nops(ALLCONNECTED_COMPONENTS));
```

The Number of the connected components is, 1

```
> IsLimitPoint(d,A,X,T);
```

$\{d\}$

```
> LimitPoints(A,X,T);
```

$\{b, d\}$

```
> ClosurePoints(A,X,T);
```

$\{a, b, c, d\}$

```
> BoundaryPoints(A,X,T);
```

$\{b, d\}$

```
> InteriorPoints(A,X,T);
```

$\{a, c\}$

```
> ExteriorPoints(A,X,T);
```

$\{\}$

```
> IsolatedPoints(A,X,T);
```

$\{a, c\}$

```
> IsolatedPoints2(A,X,T);
```

$\{a, c\}$

```
> restart;

with(combinat):
#(12) A procedure to check if two spaces are homeomorphic
  or not .
```

```
> newjob:=proc(t,p)
local u:
> #apply a permutation p of the elements of a space to the
sets in a topology t.
 {seq(subs({seq(X[i]=p[i],i=1..nops(X))} ,u) , u = t)}:
end:
> ishomeo:=proc(t1,t2) #check to see if two spaces are
homeomorphic
local answer, p:
 #we can first check for some trivial invariants, such as.
 if nops(t1) <> nops(t2) then return(false) fi:
 answer:=false:
 for p in P do
if {op(newjob(t1,p))}={op(t2)} then answer:=true: break:
fi:od:
 #Note that we have to compare _sets_ rather than _lists_!.
 answer:
end:
> #1-Example;
> X:={a,b,c,d};
```

$$X := \{a, b, c, d\}$$

```
> P:=permute(X);
```

$$P := [[a, b, c, d], [a, b, d, c], [a, c, b, d], [a, c, d, b], [a, d, b, c], [a, d, c, b], [b, a, c, d], [b, a, d, c], [b, c, a, d], [b, c, d, a], [b, d, a, c], [b, d, c, a], [c, a, b, d], [c, a, d, b], [c, b, a, d], [c, b, d, a], [c, d, a, b], [c, d, b, a], [d, a, b, c], [d, a, c, b], [d, b, a, c], [d, b, c, a], [d, c, a, b], [d, c, b, a]]$$

```
> t1:={{},{a},{a,b},X};
```

$$t1 := \{\{\ \}, \{a\}, \{a, b\}, \{a, b, c, d\}\}$$

```
> t2:={{},{a,b},X};
```

$$t2 := \{\{\ \}, \{a, b\}, \{a, b, c, d\}\}$$

```
> ishomeo(t1,t2);
```

$$false$$

```
#2-Example;
> X:={a,b,c,d};
```

$$X := \{a, b, c, d\}$$

```
> P:=permute(X):
> t1:={{},{a,b},{c,d},X};
t2:={{},{a},{b,c,d},X};
```

```
X:=[op(X)]:
```

$$t1 := \{\{\,\}, \{a, b\}, \{c, d\}, \{a, b, c, d\}\}$$

$$t2 := \{\{\,\}, \{a\}, \{b, c, d\}, \{a, b, c, d\}\}$$

```
> ishomeo(t1,t2);
```

false

```
> restart;
with(combinat):
>  X:={a};
```

$$X := \{a\}$$

```
> Y:=powerset(X);
```

$$Y := \{\{\,\}, \{a\}\}$$

```
> Z:=Y minus{{},X};
```

$$Z := \{\,\}$$

```
> W:=powerset(Z);
```

$$W := \{\{\,\}\}$$

```
> T:={seq(w union{{},X},w=W)};
```

$$T := \{\{\{\,\}, \{a\}\}\}$$

```
> print(`there are `, nops(T), `candidate collection of
subsets of X!`);
```

there are , 1, *candidate collection of subsets of X!*

```
> #(25)A procedure to find all topologies on a given set X.
AllTop:=proc(T)
local i,O,A,U,B,c;
B:={};
for O in T do
U:=O;
for A in O do
U:=U union map(`intersect`,U,A);
od;
U;
for c in U do
U:=U union map(`union`,U,c);
od;
U;
if U=O then B:=B union {O}; else B:=B;
fi;
```

```
od;
B;
end:
ALLTopologies:=AllTop(T);
```

$ALLTopologies := \{\{\{\,\}, \{a\}\}\}$

```
> print(`There are`,nops(ALLTopologies),`topologies on a set
of`,nops(X),`points`):
```

There are, 1, *topologies on a set of*, 1, *points*

```
>#(10)A procedure to check if agiven topology is T0-space .
isT0:=proc(X,T)
local x,y,O,test;
if nop(X)=1 then true ; else
for x in X do
for y in X minus{x} do
for O in T do
test:=evalb((member(x,O)and not(member(y,O)))or (member(y,O)
and not(member(x,O))));
if test then break; fi;
od:
if not(test) then break;fi;
od:
if not(test) then break;fi;
od:
fi;
test;
end:
> #(26) A procedure to find all T0-spaces on a given set X.
> ALLT0:=proc(ALLTopologies)
local i,T,T0S;
T0S:={};
if nops(X)=1 then T0S:=ALLTopologies;else
for T in ALLTopologies do
if isT0(X,T)=true then T0S:= T0S union {T} ; else T0S:=T0S;
fi;
od;
fi;
T0S;
end:
> ALLT0_Topologies:=ALLT0(ALLTopologies);
```

$ALLT0\_Topologies := \{\{\{\ \}, \{a\}\}\}$

```
> print(`there are` ,nops(ALLT0_Topologies),`T0-spaces on set
with`,nops(X),`points`);
```

there are, 1, *T0-spaces on set with*, 1, *points*

```
> #(13) A procedure to find all inequivalent topologies on a
given set X .
#Let's tidy them up by size.
>
>  bigger:=proc(t1,t2) if nops(t1) < nops(t2) then true else
false fi:end:
>
>  #Apply this to all the elements in each topology, and to
the set of all top's.
>
>  ALLTopologies:=sort([seq( sort([op(t)],bigger)
,t=ALLTopologies)],bigger):
>
>  #Now think about homeomorphisms, i.e., permutations
preserving open sets.
>  P:=permute(X):
>
>  X:=[op(X)]:
>  newjob:=proc(t, p) local u:
>
>  #apply a permutation p of the elements of a space to the
sets in a topology t.
>   {seq(subs({seq(X[i]=p[i],i=1..nops(X))} ,u), u = t)}:end:
>  ishomeo:=proc(t1,t2) #check to see if two spaces are
homeomorphic
>  local answer,p:
>   #we can first check for some trivial invariants,such as.
>    if nops(t1) <> nops(t2) then return(false) fi:
>   answer:=false:
>   for p in P do if {op(newjob(t1,p))}={op(t2)} then
answer:=true: break: fi:od:
>   #Note that we have to compare _sets_ rather than_lists_!.
>   answer:> end:>
Types:=[]:
```

```
>  for t in ALLTopologies do
>  isnew:=true:
>  for u in Types do
>  if ishomeo(t,u) then isnew:=false: break:fi:
>  od:
>  if isnew = true then Types:=[op(Types),t] fi:
>  od:
>  print(`There are`,nops(Types),`homeomorphism types of
topologies among them`):

>  for t in Types do lprint(t) od:
>  #quit.
> #...or keep playing with these sets.
>                    There are, 1, homeomorphism types of topologies among them
> [{}, {a}]

> #A procedure to find all Homeomorphosm types of
   T0-spaces on a given set X;

 #Let's tidy them up by size.
 bigger:=proc(t1,t2) if nops(t1) < nops(t2) then true else
false fi:end:
>
>  #Apply this to all the elements in each topology, and to
the set of all top's.
>
>  ALLT0_Topologies:=sort([seq( sort([op(t)],bigger)
,t=ALLT0_Topologies)],bigger):
>
>  #Now think about homeomorphisms, i.e., permutations
preserving open sets.
>  P:=permute(X):
>
>  X:=[op(X)]:
>  newjob:=proc(t, p) local u:
>
>  #apply a permutation p of the elements of a space to the
sets in a topology t.
>{seq(subs({seq(X[i]=p[i],i=1..nops(X))} ,u) ,u =t)}: end:
```

```
> ishomeo:=proc(t1,t2) #check to see if two spaces are
homeomorphic
 local answer, p:
  #we can first check for some trivial invariants, such as.
   if nops(t1) <> nops(t2) then return(false) fi:
  answer:=false:
  for p in P do if {op(newjob(t1,p))}={op(t2)} then
answer:=true: break: fi:od:
  #Note that we have to compare _sets_ rather than_lists_!.
  answer: end:
 Types:=[]:
 for t in ALLT0_Topologies do
  isnew:=true:
 for u in Types do
   if ishomeo(t,u) then isnew:=false: break:fi:
  od:
  if isnew = true then Types:=[op(Types),t] fi:
 od:
 print(`There are`,nops(Types),`T0 homeomorphism types of
topologies among them`):
 for t in Types do lprint(t) od:
#quit.
#...or keep playing with these sets.
```

There are, 1, *T0 homeomorphism types of topologies among them*

```
[{}, {a}]
> restart;
with(combinat):
>  X:={a,b};
```

$$X := \{a, b\}$$

```
> Y:=powerset(X);
```

$$Y := \{\{\,\}, \{a\}, \{b\}, \{a, b\}\}$$

```
> Z:=Y minus{{},X};
```

$$Z := \{\{a\}, \{b\}\}$$

```
> W:=powerset(Z);
```

$$W := \{\{\,\}, \{\{a\}\}, \{\{b\}\}, \{\{a\}, \{b\}\}\}$$

```
> T:={seq(w union{{},X},w=W)};
```

$$T := \{\{\{\,\}, \{a, b\}\}, \{\{\,\}, \{a\}, \{a, b\}\}, \{\{\,\}, \{b\}, \{a, b\}\}, \{\{\,\}, \{a\}, \{b\}, \{a, b\}\}\}$$

```
> print(`there are `, nops(T), `candidate collection of
subsets of X!`);
```

there are , 4, *candidate collection of subsets of X!*

```
> #(25)A procedure to find all topologies on a given set X.
AllTop:=proc(T)
local i,O,A,U,B,c;
B:={};
for O in T do
U:=O;
for A in O do
U:=U union map(`intersect`,U,A);
od;
U;
for c in U do
U:=U union map(`union`,U,c);
od;
U;
if U=O then B:=B union {O}; else B:=B;
fi;
od;
B;
end:
ALLTopologies:=AllTop(T);
```

$$ALLTopologies := \{\{\{\,\}, \{a, b\}\}, \{\{\,\}, \{a\}, \{a, b\}\}, \{\{\,\}, \{b\}, \{a, b\}\}, \{\{\,\}, \{a\}, \{b\}, \{a, b\}\}\}$$

```
> print(`There are`,nops(ALLTopologies),`topologies on a set
of`,nops(X),`points`):
```

There are, 4, *topologies on a set of*, 2, *points*

```
> #A procedure to check if agiven topology is T0-space .
isT0:=proc(X,T)
local x,y,O,test;
if nop(X)=1 then true ; else
for x in X do
for y in X minus{x} do
for O in T do
test:=evalb((member(x,O)and not(member(y,O)))or (member(y,O)
and not(member(x,O))));
if test then break;fi;od:
```

```
if not(test) then break;fi;
od:
if not(test) then break;fi;
od:
fi;
test;
end:
> #(26) A procedure to find all T0-spaces on a given set X.
> ALLT0:=proc(ALLTopologies)
local i,T,T0S;
T0S:={};
if nops(X)=1 then T0S:=ALLTopologies;else
for T in ALLTopologies do
if isT0(X,T)=true then T0S:= T0S union {T} ; else T0S:=T0S;
fi;
od;
fi;
T0S;
end:
> ALLT0_Topologies:=ALLT0(ALLTopologies);
```

$$ALLT0\_Topologies := \{\{\{\,\}, \{a\}, \{a, b\}\}, \{\{\,\}, \{b\}, \{a, b\}\}, \{\{\,\}, \{a\}, \{b\}, \{a, b\}\}\}$$

```
> print(`there are` ,nops(ALLT0_Topologies),`T0-spaces on set with`,nops(X),`points`);
```

there are, 3, *T0-spaces on set with*, 2, *points*

```
> #(13) A procedure to find all inequivalent topologies on a given set X .
#Let's tidy them up by size.
>  bigger:=proc(t1,t2) if nops(t1) < nops(t2) then true else false fi:end:
>  #Apply this to all the elements in each topology, and to the set of all top's.
>  ALLTopologies:=sort([seq( sort([op(t)],bigger) ,t=ALLTopologies)],bigger):
>
>  #Now think about homeomorphisms, i.e., permutations preserving open sets.
>  P:=permute(X):
```

```
>
>  X:=[op(X)]:
>  newjob:=proc(t, p) local u:
>
>  #apply a permutation p of the elements of a space to the
sets in a topology t.
>
>   {seq(subs({seq(X[i]=p[i],i=1..nops(X))} ,u), u = t)}:end:
>  ishomeo:=proc(t1,t2) #check to see if two spaces are
homeomorphic
>  local answer,p:
>   #we can first check for some trivial invariants,such as.
>    if nops(t1) <> nops(t2) then return(false) fi:
>   answer:=false:
>   for p in P do if {op(newjob(t1,p))}={op(t2)} then
answer:=true: break: fi:od:
>   #Note that we have to compare _sets_ rather than
_lists_!.
>   answer:
>  end:
>  Types:=[]:
>  for t in ALLTopologies do
>  isnew:=true:
>  for u in Types do
>  if ishomeo(t,u) then isnew:=false: break:fi:
>  od:
>  if isnew = true then Types:=[op(Types),t] fi:
>  od:
>  print(`There are`,nops(Types),`homeomorphism types of
topologies among them`):
> for t in Types do lprint(t) od:
>
>  #quit.
> #...or keep playing with these sets.
>
```

There are, 3, *homeomorphism types of topologies among them*

```
[{}, {a, b}]
[{}, {b}, {a, b}]
[{}, {b}, {a}, {a, b}]
```

```
> #(27)A procedure to find all Homeomorphosm types of  T0-
spaces on a given set X;
 #Let's tidy them up by size.
 bigger:=proc(t1,t2) if nops(t1) < nops(t2) then true else
false fi:end:
>
>  #Apply this to all the elements in each topology, and to
the set of all top's.
>  ALLT0_Topologies:=sort([seq( sort([op(t)],bigger)
,t=ALLT0_Topologies)],bigger):
>
>  #Now think about homeomorphisms, i.e., permutations
preserving open sets.
>  P:=permute(X):
>  X:=[op(X)]:
>  newjob:=proc(t, p) local u:
>  #apply a permutation p of the elements of a space to the
sets in a topology t.
>
>  {seq(subs({seq(X[i]=p[i],i=1..nops(X))} ,u) ,u =t)}:
> end:
> ishomeo:=proc(t1,t2) #check to see if two spaces are
homeomorphic
 local answer, p:
  #we can first check for some trivial invariants, such as.
   if nops(t1) <> nops(t2) then return(false) fi:
  answer:=false:
  for p in P do if {op(newjob(t1,p))}={op(t2)} then
answer:=true: break: fi:od:
  #Note that we have to compare _sets_ rather than _lists_!.
  answer:end:
 Types:=[]:
 for t in ALLT0_Topologies do
  isnew:=true:
 for u in Types do
   if ishomeo(t,u) then isnew:=false: break:fi:
  od:
  if isnew = true then Types:=[op(Types),t] fi:
 od:
```

```
 print(`There are`,nops(Types),`T0 homeomorphism types of
topologies among them`):
 for t in Types do lprint(t) od:
#quit.
#...or keep playing with these sets.
```

There are, 2, *T0 homeomorphism types of topologies among them*

```
>[{}, {b}, {a, b}]
 [{}, {b}, {a}, {a, b}]
> restart;
with(combinat):
>  X:={a,b,c};
```

$X := \{a, b, c\}$

```
> Y:=powerset(X);
```

$Y := \{\{\}, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$

```
> Z:=Y minus{{},X};
```

$Z := \{\{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}\}$

```
> W:=powerset(Z);
```

$W := \{\{\}, \{\{a\}\}, \{\{b\}\}, \{\{c\}\}, \{\{a, b\}\}, \{\{a, c\}\}, \{\{b, c\}\}, \{\{a\}, \{b\}\}, \{\{a\}, \{c\}\}, \{\{a\}, \{a, b\}\}, \{\{a\}, \{a, c\}\}, \{\{a\}, \{b, c\}\}, \{\{b\}, \{c\}\}, \{\{b\}, \{a, b\}\}, \{\{b\}, \{a, c\}\}, \{\{b\}, \{b, c\}\}, \{\{c\}, \{a, b\}\}, \{\{c\}, \{a, c\}\}, \{\{c\}, \{b, c\}\}, \{\{a, b\}, \{a, c\}\}, \{\{a, b\}, \{b, c\}\}, \{\{a, c\}, \{b, c\}\}, \{\{a\}, \{b\}, \{c\}\}, \{\{a\}, \{b\}, \{a, b\}\}, \{\{a\}, \{b\}, \{a, c\}\}, \{\{a\}, \{b\}, \{b, c\}\}, \{\{a\}, \{c\}, \{a, b\}\}, \{\{a\}, \{c\}, \{a, c\}\}, \{\{a\}, \{c\}, \{b, c\}\}, \{\{a\}, \{a, b\}, \{a, c\}\}, \{\{a\}, \{a, b\}, \{b, c\}\}, \{\{a\}, \{a, c\}, \{b, c\}\}, \{\{b\}, \{c\}, \{a, b\}\}, \{\{b\}, \{c\}, \{a, c\}\}, \{\{b\}, \{c\}, \{b, c\}\}, \{\{b\}, \{a, b\}, \{a, c\}\}, \{\{b\}, \{a, b\}, \{b, c\}\}, \{\{b\}, \{a, c\}, \{b, c\}\}, \{\{c\}, \{a, b\}, \{a, c\}\}, \{\{c\}, \{a, b\}, \{b, c\}\}, \{\{c\}, \{a, c\}, \{b, c\}\}, \{\{a, b\}, \{a, c\}, \{b, c\}\}, \{\{a\}, \{b\}, \{c\}, \{a, b\}\}, \{\{a\}, \{b\}, \{c\}, \{a, c\}\}, \{\{a\}, \{b\}, \{c\}, \{b, c\}\}, \{\{a\}, \{b\}, \{a, b\}, \{a, c\}\}, \{\{a\}, \{b\}, \{a, b\}, \{b, c\}\}, \{\{a\}, \{b\}, \{a, c\}, \{b, c\}\}, \{\{a\}, \{c\}, \{a, b\}, \{a, c\}\}, \{\{a\}, \{c\}, \{a, b\}, \{b, c\}\}, \{\{a\}, \{c\}, \{a, c\}, \{b, c\}\}, \{\{a\}, \{a, b\}, \{a, c\}, \{b, c\}\}, \{\{b\}, \{c\}, \{a, b\}, \{a, c\}\}, \{\{b\}, \{c\}, \{a, b\}, \{b, c\}\}, \{\{b\}, \{c\}, \{a, c\}, \{b, c\}\}, \{\{b\}, \{a, b\}, \{a, c\}, \{b, c\}\}, \{\{c\}, \{a, b\}, \{a, c\}, \{b, c\}\}, \{\{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}\}, \{\{a\}, \{b\}, \{c\}, \{a, b\}, \{b, c\}\}, \{\{a\}, \{b\}, \{c\}, \{a, c\}, \{b, c\}\}, \{\{a\}, \{b\}, \{a, b\}, \{a, c\}, \{b, c\}\}, \{\{a\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}\}, \{\{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}\}, \{\{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}\}\}$

```
 T:={seq(w union{{},X},w=W)};
```

T := {{{ }, {a, b, c}}, {{ }, {a}, {a, b, c}}, {{ }, {b}, {a, b, c}}, {{ },
{c}, {a, b, c}}, {{ }, {a, b}, {a, b, c}}, {{ }, {a, c}, {a, b, c}}, {{ },
{b, c}, {a, b, c}}, {{ }, {a}, {b}, {a, b, c}}, {{ }, {a}, {c}, {a, b,
c}}, {{ }, {a}, {a, b}, {a, b, c}}, {{ }, {a}, {a, c}, {a, b, c}}, {{ },
{a}, {b, c}, {a, b, c}}, {{ }, {b}, {c}, {a, b, c}}, {{ }, {b}, {a, b},
{a, b, c}}, {{ }, {b}, {a, c}, {a, b, c}}, {{ }, {b}, {b, c}, {a, b, c}},
{{ }, {c}, {a, b}, {a, b, c}}, {{ }, {c}, {a, c}, {a, b, c}}, {{ }, {c},
{b, c}, {a, b, c}}, {{ }, {a, b}, {a, c}, {a, b, c}}, {{ }, {a, b}, {b, c},
{a, b, c}}, {{ }, {a, c}, {b, c}, {a, b, c}}, {{ }, {a}, {b}, {c}, {a, b,
c}}, {{ }, {a}, {b}, {a, b}, {a, b, c}}, {{ }, {a}, {b}, {a, c}, {a, b,
c}}, {{ }, {a}, {b}, {b, c}, {a, b, c}}, {{ }, {a}, {c}, {a, b}, {a, b,
c}}, {{ }, {a}, {c}, {a, c}, {a, b, c}}, {{ }, {a}, {c}, {b, c}, {a, b,
c}}, {{ }, {a}, {a, b}, {a, c}, {a, b, c}}, {{ }, {a}, {a, b}, {b, c}, {a,
b, c}}, {{ }, {a}, {a, c}, {b, c}, {a, b, c}}, {{ }, {b}, {c}, {a, b}, {a,
b, c}}, {{ }, {b}, {c}, {a, c}, {a, b, c}}, {{ }, {b}, {c}, {b, c}, {a, b,
c}}, {{ }, {b}, {a, b}, {a, c}, {a, b, c}}, {{ }, {b}, {a, b}, {b, c}, {a,
b, c}}, {{ }, {b}, {a, c}, {b, c}, {a, b, c}}, {{ }, {c}, {a, b}, {a, c},
{a, b, c}}, {{ }, {c}, {a, b}, {b, c}, {a, b, c}}, {{ }, {c}, {a, c}, {b,
c}, {a, b, c}}, {{ }, {a, b}, {a, c}, {b, c}, {a, b, c}}, {{ }, {a}, {b},
{c}, {a, b}, {a, b, c}}, {{ }, {a}, {b}, {c}, {a, c}, {a, b, c}}, {{ },
{a}, {b}, {c}, {b, c}, {a, b, c}}, {{ }, {a}, {b}, {a, b}, {a, c}, {a, b,
c}}, {{ }, {a}, {b}, {a, b}, {b, c}, {a, b, c}}, {{ }, {a}, {b}, {a, c},
{b, c}, {a, b, c}}, {{ }, {a}, {c}, {a, b}, {a, c}, {a, b, c}}, {{ }, {a},
{c}, {a, b}, {b, c}, {a, b, c}}, {{ }, {a}, {c}, {a, c}, {b, c}, {a, b,
c}}, {{ }, {a}, {a, b}, {a, c}, {b, c}, {a, b, c}}, {{ }, {b}, {c}, {a,
b}, {a, c}, {a, b, c}}, {{ }, {b}, {c}, {a, b}, {b, c}, {a, b, c}}, {{ },
{b}, {c}, {a, c}, {b, c}, {a, b, c}}, {{ }, {b}, {a, b}, {a, c}, {b, c},
{a, b, c}}, {{ }, {c}, {a, b}, {a, c}, {b, c}, {a, b, c}}, {{ }, {a}, {b},
{c}, {a, b}, {a, c}, {a, b, c}}, {{ }, {a}, {b}, {c}, {a, b}, {b, c}, {a,
b, c}}, {{ }, {a}, {b}, {c}, {a, c}, {b, c}, {a, b, c}}, {{ }, {a}, {b},
{a, b}, {a, c}, {b, c}, {a, b, c}}, {{ }, {a}, {c}, {a, b}, {a, c}, {b,
c}, {a, b, c}}, {{ }, {b}, {c}, {a, b}, {a, c}, {b, c}, {a, b, c}}, {{ },
{a}, {b}, {c}, {a, b}, {a, c}, {b, c}, {a, b, c}}}

```
> print(`there are `, nops(T), `candidate collection of
subsets of X!`);
```

there are , 64, *candidate collection of subsets of X!*

```
> #(25)A procedure to find all topologies on a given set X.
AllTop:=proc(T)
```

```
local i,O,A,U,B,c;
B:={};
for O in T do
U:=O;
for A in O do
U:=U union map(`intersect`,U,A);
od;
U;
for c in U do
U:=U union map(`union`,U,c);
od;
U;
if U=O then B:=B union {O}; else B:=B;
fi;
od;
B;
end:
ALLTopologies:=AllTop(T);
```

$ALLTopologies := \{\{\{\ \}, \{a, b, c\}\}, \{\{\ \}, \{a\}, \{a, b, c\}\}, \{\{\ \}, \{b\}, \{a, b, c\}\}, \{\{\ \}, \{c\}, \{a, b, c\}\}, \{\{\ \}, \{a, b\}, \{a, b, c\}\}, \{\{\ \}, \{a, c\}, \{a, b, c\}\}, \{\{\ \}, \{b, c\}, \{a, b, c\}\}, \{\{\ \}, \{a\}, \{a, b\}, \{a, b, c\}\}, \{\{\ \}, \{a\}, \{a, c\}, \{a, b, c\}\}, \{\{\ \}, \{a\}, \{b, c\}, \{a, b, c\}\}, \{\{\ \}, \{b\}, \{a, b\}, \{a, b, c\}\}, \{\{\ \}, \{b\}, \{a, c\}, \{a, b, c\}\}, \{\{\ \}, \{b\}, \{b, c\}, \{a, b, c\}\}, \{\{\ \}, \{c\}, \{a, b\}, \{a, b, c\}\}, \{\{\ \}, \{c\}, \{a, c\}, \{a, b, c\}\}, \{\{\ \}, \{c\}, \{b, c\}, \{a, b, c\}\}, \{\{\ \}, \{a\}, \{b\}, \{a, b\}, \{a, b, c\}\}, \{\{\ \}, \{a\}, \{c\}, \{a, c\}, \{a, b, c\}\}, \{\{\ \}, \{a\}, \{a, b\}, \{a, c\}, \{a, b, c\}\}, \{\{\ \}, \{b\}, \{c\}, \{b, c\}, \{a, b, c\}\}, \{\{\ \}, \{b\}, \{a, b\}, \{b, c\}, \{a, b, c\}\}, \{\{\ \}, \{c\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}, \{\{\ \}, \{a\}, \{b\}, \{a, b\}, \{a, c\}, \{a, b, c\}\}, \{\{\ \}, \{a\}, \{b\}, \{a, b\}, \{b, c\}, \{a, b, c\}\}, \{\{\ \}, \{a\}, \{c\}, \{a, b\}, \{a, c\}, \{a, b, c\}\}, \{\{\ \}, \{a\}, \{c\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}, \{\{\ \}, \{b\}, \{c\}, \{a, b\}, \{b, c\}, \{a, b, c\}\}, \{\{\ \}, \{b\}, \{c\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}, \{\{\ \}, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}\}$

```
> print(`There are`,nops(ALLTopologies),`topologies on a set of`,nops(X),`points`):
```

There are, 29, *topologies on a set of*, 3, *points*

```
> #A procedure to check if agiven topology is T0-space .
isT0:=proc(X,T)
local x,y,O,test;
```

```
if nop(X)=1 then true ; else
for x in X do
for y in X minus{x} do
for O in T do
test:=evalb((member(x,O)and not(member(y,O)))or (member(y,O)
and not(member(x,O))));
if test then break; fi;
od:if not(test) then break;fi;
od:
if not(test) then break;fi;
od:
fi;
test;
end:
> #(26) A procedure to find all T0-spaces on a given set X.
> ALLT0:=proc(ALLTopologies)
local i,T,T0S;
T0S:={};
if nops(X)=1 then T0S:=ALLTopologies;else
for T in ALLTopologies do
if isT0(X,T)=true then T0S:= T0S union {T} ; else T0S:=T0S;
fi;
od;
fi;
T0S;
end:
> ALLT0_Topologies:=ALLT0(ALLTopologies);
```

$$ALLT0\_Topologies := \{\{\{\,\}, \{a\}, \{a, b\}, \{a, b, c\}\}, \{\{\,\}, \{a\}, \{a, c\}, \{a, b, c\}\}, \{\{\,\}, \{b\}, \{a, b\}, \{a, b, c\}\}, \{\{\,\}, \{b\}, \{b, c\}, \{a, b, c\}\}, \{\{\,\}, \{c\}, \{a, c\}, \{a, b, c\}\}, \{\{\,\}, \{c\}, \{b, c\}, \{a, b, c\}\}, \{\{\,\}, \{a\}, \{b\}, \{a, b\}, \{a, b, c\}\}, \{\{\,\}, \{a\}, \{c\}, \{a, c\}, \{a, b, c\}\}, \{\{\,\}, \{a\}, \{a, b\}, \{a, c\}, \{a, b, c\}\}, \{\{\,\}, \{b\}, \{c\}, \{b, c\}, \{a, b, c\}\}, \{\{\,\}, \{b\}, \{a, b\}, \{b, c\}, \{a, b, c\}\}, \{\{\,\}, \{c\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}, \{\{\,\}, \{a\}, \{b\}, \{a, b\}, \{a, c\}, \{a, b, c\}\}, \{\{\,\}, \{a\}, \{b\}, \{a, b\}, \{b, c\}, \{a, b, c\}\}, \{\{\,\}, \{a\}, \{c\}, \{a, b\}, \{a, c\}, \{a, b, c\}\}, \{\{\,\}, \{a\}, \{c\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}, \{\{\,\}, \{b\}, \{c\}, \{a, b\}, \{b, c\}, \{a, b, c\}\}, \{\{\,\}, \{b\}, \{c\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}, \{\{\,\}, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}\}$$

```
>print(`there are` ,nops(ALLT0_Topologies),`T0-spaces on set
                   with`,nops(X),`points`);
```

there are, 19, *T0-spaces on set with*, 3, *points*

```
> #(13) A procedure to find all inequivalent topologies on a
given set X .
#Let's tidy them up by size.
>bigger:=proc(t1,t2) if nops(t1) < nops(t2) then true else
false fi:end:
>
>  #Apply this to all the elements in each topology, and to
the set of all top's.
>
ALLTopologies:=sort([seq(ort([op(t)],bigger),t=ALLTopologies)
],bigger):
>#Now think about homeomorphisms, i.e., permutations
preserving open sets.
>  P:=permute(X):
>  X:=[op(X)]:
>  newjob:=proc(t, p) local u:
>
>  #apply a permutation p of the elements of a space to the
sets in a topology t.
>
>   {seq(subs({seq(X[i]=p[i],i=1..nops(X))} ,u), u=t)}:end:
>  ishomeo:=proc(t1,t2) #check to see if two spaces are
homeomorphic
>  local answer,p:
>   #we can first check for some trivial invariants,such as.
>    if nops(t1) <> nops(t2) then return(false) fi:
>   answer:=false:
>   for p in P do if {op(newjob(t1,p))}={op(t2)} then
answer:=true: break: fi:od:
>   #Note that we have to compare _sets_ rather than
_lists_!.
>   answer:
>  end:
>Types:=[]:
```

```
>  for t in ALLTopologies do
>  isnew:=true:
>  for u in Types do
>  if ishomeo(t,u) then isnew:=false: break:fi:
>  od:
>  if isnew = true then Types:=[op(Types),t] fi:
>  od:
>  print(`There are`,nops(Types),`homeomorphism types of
topologies among them`):
>
>  for t in Types do lprint(t) od:
>
>  #quit.
> #...or keep playing with these sets.
```

There are, 9, *homeomorphism types of topologies among them*

```
[{}, {a, b, c}]
[{}, {b, c}, {a, b, c}]
[{}, {c}, {a, b, c}]
[{}, {c}, {b, c}, {a, b, c}]
[{}, {c}, {a, b}, {a, b, c}]
[{}, {c}, {b, c}, {a, c}, {a, b, c}]
[{}, {c}, {b}, {b, c}, {a, b, c}]
[{}, {c}, {b}, {b, c}, {a, c}, {a, b, c}]
[{}, {c}, {b}, {a}, {b, c}, {a, c}, {a, b}, {a, b, c}]
```

```
>
> #A procedure to find all Homeomorphosm types of  T0-spaces
on a given set X;
 #Let's tidy them up by size.
 bigger:=proc(t1,t2) if nops(t1) < nops(t2) then true else
false fi:end:
>
>  #Apply this to all the elements in each topology, and to
the set of all top's.
> ALLT0_Topologies:=sort([seq( sort([op(t)],bigger)
,t=ALLT0_Topologies)],bigger):
> #Now think about homeomorphisms, i.e., permutations
preserving open sets.
>  P:=permute(X):
```

```
>
>  X:=[op(X)]:
>  newjob:=proc(t, p) local u:
>  #apply a permutation p of the elements of a space to the
sets in a topology t.
>
>  {seq(subs({seq(X[i]=p[i],i=1..nops(X))} ,u) ,u =t)}:
> end:

> ishomeo:=proc(t1,t2) #check to see if two spaces are
homeomorphic
 local answer, p:
  #we can first check for some trivial invariants, such as.
   if nops(t1) <> nops(t2) then return(false) fi:
  answer:=false:
  for p in P do if {op(newjob(t1,p))}={op(t2)} then
answer:=true: break: fi:od:
  #Note that we have to compare _sets_ rather than_lists_!.
  answer:
  end:
 Types:=[]:
 for t in ALLT0_Topologies do
  isnew:=true:
 for u in Types do
   if ishomeo(t,u) then isnew:=false: break:fi:
  od:
  if isnew = true then Types:=[op(Types),t] fi:
 od:
 print(`There are`,nops(Types),`T0 homeomorphism types of
topologies among them`):
 for t in Types do lprint(t) od:
#quit.
#...or keep playing with these sets.
```

There are, 5, T0 homeomorphism types of topologies among them

```
[{}, {c}, {b, c}, {a, b, c}]
[{}, {c}, {b, c}, {a, c}, {a, b, c}]
[{}, {c}, {b}, {b, c}, {a, b, c}]
[{}, {c}, {b}, {b, c}, {a, c}, {a, b, c}]
[{}, {c}, {b}, {a}, {b, c}, {a, c}, {a, b}, {a, b, c}]
```

Appendix

Maple.15

The Software :

Maple is a commercial computer **algebra** system .It was first developed in 1980 by the symbolic computation group at The University of Waterloo in Waterloo , Ontario , Canada since 1988 , it has been developed and sold commercially by Waterloo Maple Inc (also known as Maple soft)
a Canadian company based in Waterloo , Ontario Canada .The current major version is version 16 which was released in March 2012 .

History:

The first concept of Maple arose from a meeting in November 1980 at The University of Waterloo in Waterloo . Researchers at The University wished to purchase a computer powerful enough to run **Macsyma** , instead , it was decided that they would their own computer algebra system that would be able to run on lower cost computers.

The first limited version appearing in December 1980 with Maple demonstrated first at conferences beginning in 1982 .

The name is conference to Maple`s condition heritage by The end of 1983 , over 50 Universities had copies of Maple installed on their machines .

Releases:

1- Maple 1.0 : January , 1982 .

2- Maple 1.1 : January , 1982 .

3- Maple 2.0 : May , 1982 .

4- Maple 2.1 : Jun , 1982 .

5- Maple 2.15 : August , 1982 .

6- Maple 2.2 : December , 1982 .

7- Maple 3.0 : May , 1982 .

8- Maple 3.1 : October , 1983 .

9- Maple 3.2 : April , 1984 .

10- Maple 3.3 : March , 1985 (first public available version) .

11- Maple 4.0 : April , 1986 .

12- Maple 4.1 : May , 1987.

13- Maple 4.2 : December , 1987 .

14- Maple 4.3 : March , 1989 .

15- Maple V : August , 1990 .

16- Maple V R_2 : November , 1992 .

17- Maple V R_3 : March 15 , 1994 .

18- Maple V R_4 : January , 1996 .

19- Maple V R_5 : November 1 , 1992 .

20- Maple 6 : December 6 , 1999 .

21- Maple 7 : July 1 , 2001 .

22- Maple 8 : April 16 , 2002 .

23- Maple 9 : June 30 , 2003 .

24- Maple 9.5 : April 15 , 2004 .

25- Maple 10 : May 10 , 2005 .

26- Maple 11 : February 21 , 2007 .

27- Maple 12 : May , 2008 .

28- Maple 13 : April , 2009 .

29- Maple 14 : April , 2010 .

30- Maple 14.01 : October 28 , 2010 .

31- Maple 15 : April 13 , 2011 .

32- Maple 15.01 : June 21 , 2011 .

33- Maple 16 : March 28 , 2012 .

34- Maple 16.02 : November 27, 2012 .

Architecture:

Maple is based on small kernel , written in C, which provides the maple language , most functionality is provided by libraries , which come from a variety of sources , many numerical computations are performed by **NAG** numerical libraries , ATLAS libraries , or GMP libraries , most of the libraries are written in the maple language ; these have viewable source code . Different functionality in maple requires numerical data in different formats , symbolic expressions are stored in memory as directed a cyclic graphs , the classic interface is written in C .

Note :

There are general commands and commands in specialized packages.

Packages :

Index of Descriptions for Packages of Library Functions .

Description:

The following packages are available

Algcurves Algebraic Array Tools Audio Tools	CUDA Curve Fitting Database DEtools	Genfunc geom3d geometry gfun

Bits	DifferentialGeometry	Global
Cache		Optimization
	Difforms	
CAD		Graph Theory
	Discrete Transforme	
Codagen		Grid
	DocumentTools	
Code Generation		Groebner
	DynamicSystems	
Code Tools		group
Combinat	ExcelTools	
		hashmest
Combstruct	ExternalCalling	
	File Tools	Heap
Contex Menu	GaussInt	HTTP
	Magma	
	Maplets	Image Tools
Installer Builder		Padic
	MathematicalFunctions	
IntegerRelations	MathML	priqueue
		PDEtools
	Matlab	
IntegrationTools		
Inttrans	MatrixPolynomialAlgebra	Physic
	MmaTranslator	
Large Expression		plots

Library Tools liesymm Linear Algebra LinearFunctionalSystem LinearOperators List Tools Logic LREtools RandomTools RationalNormalForms RealDomain RegularChains RootFinding ScientificConstants	MTM MultiSeries numapporox numtheory Optimization Ore_algera OreTools OrthogonalSeries Orthopoly Student[Calculus1] Student[LinearAlgebra] Student[Multivariate-Calculus] Student[Numerical-Analysis]	Plottoos PolynomialIdeals PolynomialTools powseries priqueue processControl QDifferenceEquatio-ns queue Tolerances Typesetting TypeTools Units VariationalCalculus VectorCalculus

ScientificErrorAnalysis Security Simplex Solde SNAP Sockets SoftwareMetrics SolveTools SpreadStudent	Student[Precalculus] Student[Vector-Calculus] SumTools	Worksheet XMLTools

Packages used in Finite Topological Spaces:

1-*Combinat:*

Combinatorial functions , including commands for calculating permutations and combinations of lists , and partitions of integers .

List of combinat package commands :

Ball	Catpord

Chi	Composition
Decodepart	Eulerian 1
Fibonacci	Graycode
Lastpart	Nextpart
Numbcomp	Numbperm
Permute	Prevpart
Randpart	setpartition
Stirling 2	vectoint
Binomial Choose	Character Conjpart
Encodepart first part multinomial	eulerian 2 inttovec
numbpart	Numbcomb

powerset	Partition
randperm	Randcomb
Subse	Striling 1

2-network :

Description :

A network is represented by a graph consisting of vertices and edges , The edges may be directed , and loops and multiple edges are allowed .

The basic commands in this packages perform the manipulation of the underlying graphs .

List of Networks Packages Commands :

The following is a list of available commands :

acycpoly	addedge	addvertex	allpairs
ancestor	arrivals	bicomponents	charpoly
chrompoly	complement	complete	components

connect	connectivity	contracr	cuntcuts
counttrees	cube	cycle	cyclebase
doughter	degreeseq	delete	departures
diameter	dinic	djspantree	dodecahedron
draw	draw3d	duplicate	edges
ends	eweight	flow	flowpoly
fundcyc	getlabel	girth	graph
graphical	gsimp	gunion	head
icosahedrons	incidence	incident	indgree
induce	isplanar	maxdegree	mincut
mindegree	neighbors	new	octahedron
outdegree	path	Petersen	random

rank	rankpoly	shortpathree	show
shirk	span	spanpoly	spantree
tail	tetrahedron	tuttpoly	vdegree
vertices	void	vweight	

2- plottools :

Description :

The plottools packages contains routines that can generate basic graphical objects for use in plot structures .You can generate and alter plot structures using the commands in this packages .

List of Plottools Package Commands :

arc	arrow	circle	cone
cuboid	curve	cutin	cutout
cylinder	disc	dodecahedron	ellipse

ellipticalArc	hemisphere	hexahedron	hyperbola
icosahedrons	line	octahedron	parallelepiped
pieslice	point	polygon	rectangle
semitorus	sphere	tetrahedron	torus

The commands to alter or examine plot structure are :

getdata	project
rotate	stellate
translate	reflect
homothety	transform
scale	

References:

[1] May , J.P , Finite Topological Spaces(Notes for REU) , (Lecture Notes) ,(2008) .

[2] McCord . M . Singular homology groups and homotopy groups of finite topological spaces , Duke Mathematical Journal 33 , 465-474 , (1996) .

[3] Munkers , R . J, Topology A First Course, Prentice Hall, Massachusetts Institute of Technology , (1975) .

[4] P .V .O ' neil , Fundamental concepts of topology , GORDON and BREACH SCIENCE PUBLISHER HALL , college of William and Mary in Virginia , (1972).

[5] Roman Steven , Lattice and Ordered Sets , Springer Science+ Business Media , (2008) .

[6] Speer , Timothy , A short study of Alexandroff space , New York university , July 2000.

[7] Stong . R . E . Finite topological spaces. Transaction of the American Mathematical Society Vol .123 , No , pp . 325-340 , (Jun ,1966) .

Internet articles:

[8]http://www.maplesoft.com/applications/view.aspx?SID=4122&view=html .

[9] http://www.math.niu.edu/~rusin/known-math/99/top_proc (1999) .

[10] http://en.wikipedia.org/wiki/Finite_topological_space .

[11] http://mathworld.wolfram.com/StirlingNumberoftheSecondKind.html .

خُـــــلاصة

الفضاءات التبولوجية النهائية باستخدام المابل

قد استحوذت الفضاءات التُبولوجية النهائية مُؤخرًا على اهتمام علماء التُبولوجي حيث أن المعالجة الرقمية و الصورية تنطـلق من مفهوم تقارب النقاط وتسعى لفهم النتائج التي تنطلق من هذا المفهوم , توجد مجموعة من الأعمال الرياضية في هذا الموضوع و أهمها ورقتين مستقلتين نُشرتا عام 1966 وهي من الأعمال الرياضية الجميلة ولهما أهمية بشكل خاص , و في هذه الأطروحة سوف نعمل من خلالهما، وأيضا من خلال مُذكرة مُلاحَظات نُشرت على الانترنت في عام 2008.

الفصل الأول بدايةً مع تعريف الفضاءات التوبولوجية النهائية والفئات المفتوحة . الأساسية الصغرى . وأيضا نناقش بديهيات الانفصال، ثم نناقش الخصائص المميزة لهذه الفضاءات من ناحية الاستمرارية والمتشابهات والتراص و الترابط و الترابط المساري . أيضا هناك أمثله تُظهر هذه الخصائص المختلفة في كثير من الحالات.

في الفصل الثاني ندرس فضاءات الكسندروف و أغلب خواصها المميزة وكيفية اشتقاق فضاءات الكسندروفية جديدة من فضاءات الكسندروفية سابقة

في الفصل الثالث نقوم بإنشاء إجراءات مستخدمين برنامج
Maple .15
لحساب الكثير من القضايا المتعلقة بالفراغات التوبولوجية النهائية. مثل حساب النقاط الخاصة و الفضاءات التوبولوجية و فضاءات T_0
الممكن تعريفها لفئة منتهية النقاط . أيضا إجرأت لحساب القواعد الاساسية الصغرى و المركبات المترابطة لفضاء منتهي . الملحق يتضمن وصف للبرنامج وفي نهاية الأطروحة تم وضع قائمة بالمراجع المستخدمة.