The Knowledge

Callum Cassidy-Nolan*

November 9, 2021

*cuppajoeman.com

Dedicated to the proofs that were left as exercises to the readers

Contents

1 Analysis **8**
- 1.1 Limits . . . 9
- 1.2 Differentiation . . . 12
- 1.3 Sequences . . . 16
- 1.4 Multi Variable . . . 16

2 Linear Algebra **17**
- 2.1 Vectors . . . 17
- 2.2 Linear Transformations . . . 18
- 2.3 Matrices . . . 19

3 First Order Logic **21**
- 3.1 Languages . . . 21
- 3.2 Deductions . . . 28
- 3.3 Completeness . . . 29
- 3.4 Incompleteness . . . 33

4 Topology **37**
- 4.1 Topological Spaces and Continuous Functions . . . 37
  - 4.1.1 Basis for a Topology . . . 37
  - 4.1.2 The Subspace Topology . . . 37
  - 4.1.3 The Product Topology . . . 38
  - 4.1.4 The Metric Topology . . . 39
- 4.2 Connectedness and Compactness . . . 42
  - 4.2.1 Compact Spaces . . . 45

5 Computer Science **46**

6 Graph Theory **47**

7 Probability **48**
- 7.1 Bayesian Interference . . . 48

8 Programming **49**

List of Theorems

1.0.1 Triangle Inequality 8
1.2.1 Chain Rule 12
3.3.1 Completeness Theorem 33
4.1.1 Basis for the Box Topology 38
4.1.2 Basis for the Product Topology 39

List of Propositions

1.1.1 Limit of Constant 9
1.1.2 Constant in Limit 9
1.1.3 Sum of Limits 10
1.2.1 Little O and Differentiability Equivalence 13
2.1.1 Algebraic Def Implies Geometric Def of Dot Product 17
3.1.1 Double Negation Equivalence 26
3.3.1 Contradiction has no Model 29
3.3.2 Implying a Contradiction Means no Model 30
3.3.3 Logical Contradiction 30
3.4.1 Negation of a Sigma Formula is a Pi Formula 34
4.1.1 Epsilon Balls Form A Basis 40
4.2.1 Connected Implies Closure Connected 44
6.0.1 Maximum Number of Edges 47

List of Lemmas

1.0.1 Absolute Value is Equal to max 8
3.2.1 Universal connection to Variable Assignment Function 28
3.3.1 Constant Replacement Removes It 31
3.3.2 Constant Extension still Consistent 32
4.1.1 Epsilon Ball Contains Another 40
4.2.1 Covering Yields Finite Covering if and only if Compact 45

List of Definitions

1.1.1 Real Valued Limit . . . 9
1.3.1 Bounded Sequence . . . 16
1.4.1 Directional Derivative . . . 16
2.1.1 Algebraic Dot Product . . . 17
2.1.2 Geometric Dot Product . . . 17
2.2.1 Matrix for a Linear Transformation . . . 18
2.3.1 Matrix Multiplication . . . 19
3.1.1 First-Order Language . . . 21
3.1.2 Language of Number Theory . . . 21
3.1.3 Term . . . 22
3.1.4 Terms of a Language . . . 22
3.1.5 Formula . . . 22
3.1.6 Formulas of a Language . . . 22
3.1.7 Subformula . . . 22
3.1.8 Scope . . . 22
3.1.9 A Term's Variable Replaced by a Term . . . 23
3.1.10 A Formulas Variable Replaced by a Term . . . 23
3.1.11 A Term is Substitutable for a Variable . . . 24
3.1.12 Free Variable in a Formula . . . 24
3.1.13 Variable Assignment Function . . . 25
3.1.14 Term Assignment Function . . . 25
3.1.15 Interpretation of a Term . . . 25
3.1.16 Satisfaction . . . 26
3.1.17 L-Structure . . . 26
3.1.18 Equivalent Formulas . . . 26
3.1.19 A Literal Structure . . . 27
3.1.20 Restricted L-Structure . . . 27
3.2.1 Deduction of a Formula . . . 28
3.2.2 Logical Axioms . . . 28
3.3.1 Consistent Set of L Formulas . . . 29
3.4.1 Bounded Quantifier Abbreviations . . . 33
3.4.2 Sigma Formula . . . 34
3.4.3 Pi Formula . . . 34
3.4.4 Representable Set . . . 35
3.4.5 Weakly Representable Set . . . 36
3.4.6 Total Function . . . 36
3.4.7 Partial Function . . . 36
3.4.8 Representable Function . . . 36
3.4.9 Definable Set . . . 36
4.1.1 Basis . . . 37

4.1.2 Subspace Topology 37
4.1.3 Product Topology 38
4.1.4 Box Topology 38
4.1.5 R Omega 39
4.1.6 A metric 39
4.1.7 Epsilon Ball 40
4.1.8 Metric Topology 41
4.1.9 Bounded Subset of a Metric Space 41
4.2.1 Connected Space 42
4.2.2 Totally Disconnected 44
4.2.3 Covering 45
4.2.4 Compact Space 45
5.0.1 Big-O 46
5.0.2 Little-o 46
7.0.1 Conditional Independence 48

List of Examples

4.1.1 Discrete Metric 39
4.1.2 Function on R omega 41
4.2.1 Closed and Bounded, not Compact 42
4.2.2 R omega Connected? 43

Preface

This book contains knowledge that that me or my peers have obtained, the purpose is to explain things fundamentally and in full detail so that someone who has never touched the subject may be able to understand it. It will focus on conveying the ideas that are involved in synthesizing the new knowledge with less of a focus on the results themselves.

It differs from a normal textbook in that it is open source and will fall under more continuous development rather than having editions that periodically come out. It also differs in the sense that it welcomes other users to improve the book.

Here is a link to the book: https://raw.githubusercontent.com/cuppajoeman/knowledge-book/main/build/book.pdf

Structure of book

The book is partitioned into different sections based on the domain it is involved with. There may be shared definitions and theorems throughout the chapters, but in general it will start more elementary and get more advanced.

Knowledge

In this book you will find many results, will will characterize them as being one of the following

- Theorems - Results that are of importantance and who's proof is not easily found (maybe using a novel idea)
- Propositions - Results of less importance who's proof could be constructed without a novel idea
- Lemmas - Results that are technical intermediate steps which has no standing as an independent result on first observation *
- Corollaries - Results which follow readily from an existing result of greater importance

Recommendations

By now you might know that in order to actually get better at mathematics you have to engage with it. This book may be used as a reference at times, but I highly recommend trying to re-prove statements or coming up with your own ideas before instantly looking at the solutions.

*But sometimes they escape, as their usage becomes more than just an intermediate step, as Zorn's or Fatou's Lemmas did

About the companion website

The website[†] for this file contains:

- A link to (freely downlodable) latest version of this document.
- Link to download LaTeXsource for this document.

Acknowledgements

- A special word of thanks to professors who wanted to make sure I understood and learned as much as possible Alfonso Gracia-Saz[‡], Jean-Baptiste Campesato[§], Valentine Chiche-Lapierre and Gal Gross[¶]
- Thanks to Z-Module, riv, PlanckWalk, franciman, qergle from #math on https://libera.chat/.

[†] https://github.com/cuppajoeman/knowledge-book
[‡] https://www.math.toronto.edu/cms/alfonso-memorial/
[§] https://math.univ-angers.fr/~campesato/
[¶] https://www.galgr.com/

Contributing

Contributions to the project are very welcome, let's delve into how to get started with this.

Here is a list of things that can be worked on:

- Content Based
  - Adding Definitions, Theorems, ...
  - Finishing TODO's
  - Formatting of the book
- Structural Layout of Project
  - Organization
  - Simplyfing the existing structure of the directories
  - Making scripts which set up new structures
- External
  - Adding explanatory content to help onboard new users
  - Getting others involved
  - Creating infrastructure to support users (Github discussions)

Example

I will describe how to do this in detail in the below paragraphs, here is the TLDR of them. Let's say you have a theorem you'd like to add, here is a quick outline of what you have to do. Fork the project, find the structure that your theorem belongs in, one possible structure could be first order logic, then go ahead and copy the blank `theorem.tex` file to `my_new_theorem.tex`, after that write up your theorem, include it in that structures content file, compile the project and verify it's there, then push to your forked repository and create a pull request against the main repository on github.

Setup for Contributing

In order to contribute to this project you will need to understand the basics of git, LaTeXand the structure of the project.

Git

If you aren't familiar with what git is, I recommend you watch the videos on the following page https://git-scm.com/doc, and remember the link to the docs for future reference.

Next you can make a github account on github.com, the way you will get a copy of the project is by forking the project. To do that visit https://github.com/cuppajoeman/knowledge-book and click fork in the top right corner, once you've forked it we can now download it to our computer.

If you have access to a terminal then you can do the following:

```
user@machine:~# git clone <link gotten from your forked repo>
```

From here on out you will make changes to the project and make commits using git. Once you're happy with the changes you've made, you can save your git commits online by running

```
user@machine:~# git push
```

Otherwise you can download https://desktop.github.com/. And after you've made edits to the book, you can push, then finally then go to your forked version of the project and make a pull request into the main project. To do this visit your forked version of the project on github and navigate to the pull requests tab, then make sure the base repository is `cuppajoeman/knowledge-book` and add relevant info explaining the pull request. Once someone has reviewed your changes it will be merged into the main project.

LaTeX

This document is entirely written in LaTeXwhich is a language which lets us format mathematics and make figures easily.

If this system is entirely new to you, then don't worry, you'll be able to pick up most of what you need by looking at examples being used in this project, otherwise take a look at http://www.docs.is.ed.ac.uk/skills/documents/3722/3722-2014.pdf.

To be able to LaTeXup and running, it is dependent on the operating system you are using, but the one thing that remains invariant is that you will need a way to compile it and a way to edit it.

Compilers

1. https://miktex.org/ - windows
2. https://tug.org/mactex/ - mac
3. https://www.tug.org/texlive/ - linux

- Note this will probably be available in your distributions package repositories.

Editors

1. https://tug.org/texworks/ - minimal
2. https://tug.org/texworks/, https://www.xm1math.net/texmaker/ - more fully featured
3. vim, emacs, other terminal based editors
    - Allows you to be the most efficient

Once you've obtained an editor, if it has some type of built-in compiler then you will want to change the output directory to be the build directory. This makes sure that when you compile the document that the root directory of the project isn't filled with irrelevant files. For each editor it will be different, but the general setup will be to get to some sort of configuration panel (or file) and to change the editors compile command to this: `pdflatex -output-directory=build`. Some editors might have a built in mode for this, for example with tex maker one can tick the "use build directory for output files", for users who use vim, there is a built in option for vimtex like so:

```
let g:vimtex_compiler_latexmk = {
            \ 'build_dir' : 'build',
            \}
```

Communication

Feel free to ask any question on the discord server: https://discord.gg/ReceZrGuN6, it will also be a place for general discussion. Another option we can look at is github discussions.

Content Based

To understand how to add content to the project, the best step is to understand fundamentally what a structure is.

A **structure** is a directory which has the following :

- content.tex
    - This file includes all the content in the other directories of this directory
- definitions
- theorems

- lemmas
- lemmas
- corollaries
- sub_structures
  - This directory contains other **structures**

Note that the other directories; definitions, theorems, lemmas, lemmas, corollaries are non-recursive directories that contain content (Let's call them content directories from now on) see here for exactly what these directories contain.

If you want to add a new structure, the best thing to do is to verify with other members of the project if it warrants it's own structure, otherwise it can be added as a substructure of an existing one.

Supposing that you are on a unix based operating system, then here is how one could create a new structure:

Manually

```
user@machine:~# cp -r structure new_structure
user@machine:~# cd new_structure
user@machine:~# mv content.tex new_structure.tex
user@machine:~# nvim new_structure.tex
```

Otherwise if you're adding a new theorem, it could be:

```
user@machine:~# cd existing_structure/theorems
user@machine:~# cp theorem.tex my_new_theorem.tex
user@machine:~# nvim my_new_theorem.tex
user@machine:~# ...
user@machine:~#
    git add -A && git commit -m "add my new theorem" && git push
```

Using the Script

Alternatively for a more streamlined experience we can use some of the scripts we have for creating new content. (Note that right now this script is only for creating new content files, but eventually will create new structures as well)

To get started make sure you have `fzf` installed and the python package `pyperclip`.

First to learn about the script run:

```
user@machine:~# python scripts/main.py -h
```

And once you have read this we may create a new theorem like

```
user@machine:~# python scripts/main.py t "My new Theorem" -c
```

Which asks you for which structure the theorem should belong to and creates the file for us, the -c option copies `input{`*`path to new theorem`*`/my_new_theorem}` to the clipboard so that it may be included in the main content file associated with the structure.

With gui

If you're on windows or another operating system without access to terminal, you can always take the directory `structure` and copy it to a new name. To make new content file, go to the revelent content directory and copy the blank bootsrap file, that is if you wanted to make a new definition you can go to the structures definitions folder and then copy `definition.tex` to `my_new_definition.tex` and start working on it.

Chapter 1: Analysis

Theorem 1.0.1: Triangle Inequality

Let $a, b \in \mathbb{R}$ *then*

$$|a+b| \leq |a| + |b|$$

Proof

- $a + b \leq |a| + b \leq |a| + |b| \Leftrightarrow a + b \leq |a| + |b|$
- $-(a+b) = -a - b \leq |a| - b \leq |a| + |b| \Leftrightarrow -(a+b) \leq |a| + |b|$
  - Which are both derived by the fact that $|x| = \max(x, -x) \geq x, -x$
- Finally we know that $|a+b|$ is equal to the max of $a + b, -(a+b)$so no matter which one it is, we have:

$$|a+b| \leq |a| + |b|$$

■

Note that the triangle equality is intuitively telling us that the shortest distance between two points is a straight line

Lemma 1.0.1: Absolute Value is Equal to max

$\forall x \in \mathbb{R}$

$$|x| = \max(x, -x)$$

Proof

- If $x \geq 0$ then $|x| = x$ and $\max(x, -x) = x$
- If $x < 0$ then $|x| = -x$ and $\max(x, -x) = -x$ since $x < 0 \leftrightarrow -x > 0$ and therefore $-x > x$

■

1.1 Limits

Proposition 1.1.1: Limit of Constant

Let $f(x) = \alpha \in \mathbb{R}$ be a constant function, then

$$\lim_{x \to a} f(x) = \alpha$$

Proof

- Let $\varepsilon \in \mathbb{R}^+$, let δ be fixed as any real number, then assume that $\forall x \in \text{dom}(f)$ that $0 < |x - a| < \delta$, we will show that $\left|f(x) - \alpha\right| < \varepsilon$.
- Recall that $f(x) = \alpha$ for all $x \in \text{dom}(f)$, therefore $\left|f(x) - \alpha\right| = |\alpha - \alpha| = 0 < \varepsilon$ as needed.

■

Notice that in the above proof that we didn't use our hypothesis in the proof of the consequent, which makes sense because no matter which interval you look in the function is constant there, thus it doesn't depend on the hypothesis at all.

Definition 1.1.1: Real Valued Limit

Suppose f is a real valued function, then we say that the limit of f at a is L and write $\lim_{x \to a} f(x) = L$ when the following holds:

$$\forall \varepsilon \in \mathbb{R}^+, \exists \delta \in \mathbb{R}^+ \text{ such that } \forall x \in \text{dom}(f), 0 < |x - a| < \delta \Rightarrow \left|f(x) - L\right| < \varepsilon$$

Proposition 1.1.2: Constant in Limit

Assume that the following limit exists $\lim_{x \to a} f(x)$ and define L to be it's value, then for any $c \in \mathbb{R}$

$$\lim_{x \to a} \left[cf(x)\right] = c \lim_{x \to a} f(x)$$

Proof

- If $c = 0$ then using the fact that the limit of a constant is that constant itself, we have:

$$\lim_{x \to a} \left[0 f(x)\right] = \lim_{x \to a} 0 = 0 = 0 \lim_{x \to a} f(x)$$

- If $c \neq 0$, we must prove that for any $\varepsilon_c \in \mathbb{R}^{>0}$ there exists δ_c such that for

all $x_c \in \text{dom}(f)$

$$|x_c - a| \leq \delta_c \Rightarrow \left|cf(x_c) - cL\right| \leq \varepsilon_c$$

- Notice that if we were to let ε in the orignal definition be equal to $\frac{\varepsilon_c}{|c|}$ then we could multiply the equation after the implication on both sides by $|c|$ (so that we can absorb it into the absolute value).
- Let $\varepsilon_c \in \mathbb{R}^{>0}$ since the original limit holds for any epsilon, bind ε to $\frac{\varepsilon_c}{|c|}$ and we get δ such that for all $x \in \text{dom}(f)$, the following holds:

$$|x - a| \leq \delta \Rightarrow \left|f(x) - L\right| \leq \frac{\varepsilon_c}{|c|} \qquad (\alpha)$$

- Take $\delta_c = \delta$, let $x_c \in \text{dom}(f)$ and bind x in the original definition to x_c, and assume that $|x_c - a| \leq \delta_c$, because of our choice for δ_c we satisfy α's hypothesis with x replaced by x_c and we get

$$\left|f(x_c) - L\right| \leq \frac{\varepsilon_c}{c} \Leftrightarrow |c|\left|f(x) - L\right| \leq \varepsilon_c$$

- Since for any $a, b \in \mathbb{R}$ we have $|ab| = |a|\,|b|$ we can conclude with distributivity in $\mathbb{R}$ that

$$\left|cf(x_c) - cL\right| \leq \varepsilon_c$$

As required.

■

Proposition 1.1.3: Sum of Limits

If $\lim_{x\to a} f(x) = L$ *and* $\lim_{x\to a} g(x) = M$ *both exist then*

$$\lim_{x\to a}[f(x) + g(x)] = L + M$$

Proof

- Suppose antecedent holds, so we have existsance of the limits $\lim_{x\to a} f(x) = L$ and $\lim_{x\to a} g(x) = M$, now we'd like to show that the limit $\lim_{x\to a}[f(x) + g(x)]$ equals $L + M$
- Let $\varepsilon \in \mathbb{R}^{+}$, now from the fact that the two prior limits exist we can substi-

tute in $\frac{\varepsilon}{2}$ into them, to get δ_L and δ_M respectively, such that

$$\forall x \in \mathrm{dom}\,(f)\,, 0 < |x-a| < \delta_L \Rightarrow \left|f\,(x) - L\right| < \frac{\varepsilon}{2}$$

and

$$\forall x \in \mathrm{dom}\,(g)\,, 0 < |x-a| < \delta_M \Rightarrow \left|g\,(x) - M\right| < \frac{\varepsilon}{2}$$

- Take $\delta = \min(\delta_L, \delta_M)$, so that we may utilize the above inequalities as we assume that $\forall x \in \mathrm{dom}\,(f+g)$ that $0 < |x-a| < \delta$, and proceed to show that $\Big|\big(f\,(x) + g\,(x)\big) - (L+M)\Big| < \varepsilon$, since our x is bounded by the min of the two deltas from the other limits, we utilize the inequalities in tandem with the triangle equality to obtain:

$$\Big|\big(f\,(x) + g\,(x)\big) - (M+L)\Big| = \Big|\big(f\,(x) - L\big) + \big(g\,(x) - M\big)\Big| \leq \left|f\,(x) - L\right| + \left|g\,(x) - M\right| < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon$$

■

1.2 Differentiation

Theorem 1.2.1: Chain Rule

Given two functions f and g where g is differentiable at the point $\overline{x}$ and f differentiable at the point $\overline{y} = g(\overline{x})$ then

$$(f \circ g)'(\overline{x}) = f'\left(g(\overline{x})\right) \cdot g(\overline{x})$$

Proof

- We define the following two new functions

$$v(h) = \frac{g(\overline{x}+h) - g(\overline{x})}{h} - g'(\overline{x}) \text{ and } w(k) = \frac{f(\overline{y}+k) - f(\overline{y})}{k} - f'(\overline{y})$$

α : Note that $\lim_{h\to 0} v(h) = 0$ and $\lim_{k\to 0} w(k) = 0$, and that we can re-arrange the above to:

$$\Big(v(h) + g'(\overline{x})\Big)\cdot h + g(\overline{x}) = g(\overline{x}+h) \text{ and } \Big(w(k) + f'(\overline{y})\Big)\cdot k + f(\overline{y}) = f(\overline{y}+k)$$

- Now it follows that

$$\begin{aligned} f\left(g(\overline{x}+h)\right) &= f\left(\Big(v(h) + g'(\overline{x})\Big)\cdot h + g(\overline{x})\right) \\ &= f\left(g(\overline{x}) + \Big(v(h) + g'(\overline{x})\Big)\cdot h\right) \\ &= f\left(\overline{y} + \Big(v(h) + g'(\overline{x})\Big)\cdot h\right) \\ &= \left(w\left(\Big(v(h) + g'(\overline{x})\Big)\cdot h\right) + f'(\overline{y})\right)\cdot\Big(v(h) + g'(\overline{x})\Big)\cdot h + f(\overline{y}) \end{aligned}$$

- Now recall that we are interested in $(f \circ g)'(\overline{x})$, so we're going to have to look at:

$$\lim_{h\to 0} \frac{f\left(g(\overline{x}+h)\right) - f\left(g(\overline{x})\right)}{h}$$

- Notice that the term on the left of the sum in the numerator is something we already know, and therefore it becomes:

$$\lim_{h\to 0} \frac{\left(w\left(\Big(v(h) + g'(\overline{x})\Big)\cdot h\right) + f'(\overline{y})\right)\cdot\Big(v(h) + g'(\overline{x})\Big)\cdot h + f(\overline{y}) - f\left(g(\overline{x})\right)}{h}$$

- By noting that $f(\overline{y}) = f\left(g(\overline{x})\right)$, then cancelling out the h we can see that the above simplifies to

$$\lim_{h\to 0}\left(w\left(\left(v(h)+g'(\overline{x})\right)\cdot h\right)+f'(\overline{y})\right)\cdot\left(v(h)+g'(\overline{x})\right)$$

- By α on v and w with $k = g(\overline{x})\cdot h$, we have that the above limit is:

$$\lim_{h\to 0}\left(w\left(\left(g'(\overline{x})\right)\cdot h\right)+f'(\overline{y})\right)\cdot\left(g'(\overline{x})\right) = \lim_{h\to 0} f'(\overline{y})\cdot g'(\overline{x}) = \boxed{f'\left(g(\overline{x})\right)\cdot g'(\overline{x})}$$

■

The chain rule is sometimes written as $\frac{df}{dx}(x) = \frac{df}{dg}(x)\frac{dg}{dx}(x)$, but notice that this extends liebniz notation as we only know that $\frac{df}{dx} = f'(x)$, and it is not defined for when we take the derivative with respect to another function. This motivates the definition

$$\frac{df}{dg}(x) = \frac{df}{dx}\left(g(x)\right)$$

Proposition 1.2.1: Little O and Differentiability Equivalence

A function f is differentiable at the point $\overline{x}$ if and only if there is a number $\alpha \in \mathbb{R}$ for any $h \in \mathbb{R}$ we have

$$f(\overline{x}+h) = f(\overline{x}) + h\alpha + E(h)$$

Where $E \in o(h)$

Proof

- $\Rightarrow$
  - Assume that f is differentiable, therefore we have

$$f'(\overline{x}) = \lim_{h\to 0}\frac{f(\overline{x}+h)-f(x)}{h}$$

  - Equivalently:

$$\lim_{h\to 0}\frac{f(\overline{x}+h)-f(x)}{h} - f'(\overline{x}) = 0$$

  - Since $\lim_{h\to 0} f'(\overline{x}) = f'(\overline{x})$ as it's a constant, we may use the limit law

to deduce:

$$\lim_{h\to 0}\left(\frac{f(\overline{x}+h)-f(x)}{h}-f'(\overline{x})\right)=0\Leftrightarrow\lim_{h\to 0}\frac{f(\overline{x}+h)-f(\overline{x})-hf'(\overline{x})}{h}=0$$

- That is to say that $f(\overline{x}+h)-f(\overline{x})-hf'(\overline{x})\in o(h)$, set $E(h)=f(\overline{x}+h)-f(\overline{x})-hf'(\overline{x})$ then we have that

$$f(\overline{x}+h)=f(\overline{x})+hf'(\overline{x})+E(h)$$

where we've taken $\alpha=f'(\overline{x})$

- $\Leftarrow$
  - Assume that there is a number $\alpha\in\mathbb{R}$ and $E\in o(h)$, so that for any $h\in\mathbb{R}$ we have:

$$f(\overline{x}+h)=f(\overline{x})+h\alpha+E(h)\Leftrightarrow l(h)=f(\overline{x}+h)-f(\overline{x})-h\alpha$$

thus $f(\overline{x}+h)-f(\overline{x})-h\alpha\in o(h)$.

  - So then by definition of little-o, we get:

$$\lim_{h\to 0}\frac{f(\overline{x}+h)-f(\overline{x})-h\alpha}{h}=0$$

  - Since $\lim_{h\to 0}\alpha=\alpha$ we may add it to both sides using the limit law on the left to obtain:

$$\lim_{h\to 0}\left(\frac{f(\overline{x}+h)-f(\overline{x})-h\alpha}{h}+\alpha\right)=\alpha\Leftrightarrow\lim_{h\to 0}\frac{f(\overline{x}+h)-f(\overline{x})}{h}=\alpha$$

Meaning that $\alpha=f'(\overline{x})$ and that f is differentiable

■

Notice that the number α is equal to a limit and that limits are unique, therefore the solution $\alpha=f'(\overline{x})$ is the unique solution to the above proposition. Also since this it the unique solution, we know that there is no other real besides $f'(\overline{x})$ where this holds, therefore $f'(\overline{x})$ is the best linear approximation to f at the point $\overline{x}$.

Finally notice that if you solve for $E(h)$ and then recall that $E(h) \in o(h)$ we have:

$$\lim_{h \to 0} \frac{\overbrace{\overbrace{f(\overline{x}+h)}^{\text{actual value}} - \overbrace{\left(f(\overline{x}) + hf'(\overline{x})\right)}^{\text{linear estimation}}}^{\text{error}}}{h} = 0$$

the error is going to zero at a rate which is faster than linear, as if the error were decreasing at a linear rate the limit would evaluate to some non-zero constant.

1.3 Sequences

Definition 1.3.1: Bounded Sequence

A sequence $\{x_n\}$ *is said to be bounded if* $\exists M \in \mathbb{R}$ *such that* $|x_n| \leq M$ *for all* $n \in \mathbb{N}$

1.4 Multi Variable

Definition 1.4.1: Directional Derivative

Let $U \subseteq \mathbb{R}^n$ *be open,* $f : U \to \mathbb{R}$ *and* $\vec{a} \in U$ *and* $\vec{v} \in \mathbb{R}^n$

$$\partial_{\vec{v}} f\left(\vec{a}\right) \stackrel{\mathtt{D}}{=} \lim_{t \to 0 \in \mathbb{R}} \frac{f\left(\vec{a} + t\vec{v}\right) - f\left(\vec{a}\right)}{t}$$

Chapter 2: Linear Algebra

2.1 Vectors

Definition 2.1.1: Algebraic Dot Product

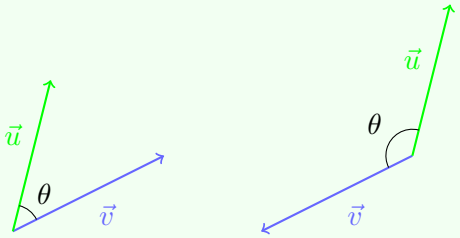
Let $u_1, u_2, \ldots, u_{n-1}, u_n$ and $v_1, v_2, \ldots, v_{n-1}, v_n$ denote the components of $\vec{u}$ and $\vec{v}$ respectively, then we have:

$$\vec{v} \cdot \vec{u} \overset{\texttt{D}}{=} \sum_{i=1}^{n} v_i u_i$$

Definition 2.1.2: Geometric Dot Product

Let $\vec{u}, \vec{v} \in \mathbb{R}^n$ and let θ be the angle between the two (the one in the range $[0, \pi]$), then we have the dot product:

$$\vec{v} \cdot \vec{u} \overset{\texttt{D}}{=} \|v\| \, \|u\| \cos(\theta)$$



Proposition 2.1.1: Algebraic Def Implies Geometric Def of Dot Product

$$\sum_{i=1}^{n} v_i u_i = \|v\| \, \|u\| \cos(\theta)$$

Proof

- Consider two vectors $\vec{x}, \vec{y} \in \mathbb{R}^2$, we know that from the polar coordinate system we can represent them as

$$\vec{x} = \left(\|x\| \cos(\theta_x), \|x\| \sin(\theta_x)\right) \quad \text{and} \quad \vec{y} = \left(\|y\| \cos\left(\theta_y\right), \|y\| \sin\left(\theta_y\right)\right)$$

- The the algebraic definition implies the following

$$\vec{x} \cdot \vec{y} = \|x\| \, \|y\| \cos(\theta_x) \cos\left(\theta_y\right) + \|x\| \, \|y\| \sin(\theta_x) \sin\left(\theta_y\right)$$

Which is equal to:

$$\|x\| \|y\| \left(\cos\left(\theta_x\right) \cos\left(\theta_y\right) + \sin\left(\theta_x\right) \sin\left(\theta_y\right)\right)$$

- Then we can recall the sin angle difference formula to obtain:

$$\vec{x} \cdot \vec{y} = \|x\| \|y\| \left(\cos\left(\theta_x - \theta_y\right)\right)$$

- And just noting that $\theta_x - \theta_y$ is the angle between the vectors $\vec{x}$ and $\vec{y}$ as needed.

■

2.2 Linear Transformations

Definition 2.2.1: Matrix for a Linear Transformation

Let $T : \mathbb{R}^m \to \mathbb{R}^n$ be a linear transformation, then a matrix $\boldsymbol{M}$ which satisfies the following equation for all $\vec{v} \in \mathbb{R}^m$

$$\left[T\vec{v}\right]_{\mathcal{E}'} = \boldsymbol{M}[\vec{v}]_{\mathcal{E}}$$

is known as a matrix from the linear transformation T. Here $\mathcal{E}$ is the standard basis for $\mathbb{R}^m$ and $\mathcal{E}'$ is the standard basis for $\mathbb{R}^n$.

2.3 Matrices

Definition 2.3.1: Matrix Multiplication

If $\mathbf{A}$ *is an* $m \times n$ *matrix and* $\mathbf{B}$ *is an* $n \times p$ *matrix,*

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{np} \end{pmatrix}$$

the matrix product $\mathbf{C} = \mathbf{AB}$ *(denoted without multiplication signs or dots) is defined to be the* $m \times p$ *matrix*

$$\mathbf{C} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mp} \end{pmatrix}$$

such that

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj} = \sum_{k=1}^{n} a_{ik}b_{kj}$$

for $i = 1, \ldots, m$ *and* $j = 1, \ldots, p$.

```python
def multiply_matrices(matrix_A, matrix_B):
  """
  Given two matrices, we compute the matrix multiplication of the two of them

  Supposing that matrix_A is MxN and that matrix_B is NxK
  then the resulting matrix is M x K (1)
  """

  transpose_matrix_B = transpose_matrix(matrix_B)

  num_rows_of_matrix_A = len(matrix_A) # This is M
  num_columns_of_matrix_A = len(matrix_A[0]) #This is N

  num_rows_of_matrix_B = len(matrix_B) #this is N
  num_columns_of_matrix_B = len(matrix_B[0])# This is K

  resulting_matrix = []

```

```
19    #initialize the matir xto be all zeros
20    # then we will fill in the entries with the correct values
21    for i in range(num_rows_of_matrix_A):
22      blank_row = []
23      for j in range(num_columns_of_matrix_B):
24        blank_row.append(0)
25      resulting_matrix.append(blank_row)
26    # Notice that the matrix now has dimenisions `num_rowsof_matrix_A` x `
        num_columns_of_matrix_B`
27    # Or equivalently M x K , as we noted by  (1)
28
29    for row_index in range(num_rows_of_matrix_A):
30      for column_index in range(num_rows_of_matrix_B):
31        row_from_matrix_A = matrix_A[row_index]
32        transposed_column_from_matrix_B = transpose_matrix_B[column_index]
33        resulting_matrix[row_index][column_index] = dot_product(row_from_matrix_A,
        transposed_column_from_matrix_B)
34
35    return resulting_matrix
36
37
38  def transpose_matrix(matrix):
39    """Given a matrix turn it into a new matrix where it's rows are equal to it's columns
        """
40    rows = len(matrix)
41    columns = len(matrix[0]) # assuming matrix is non-empty
42
43    transpose_matrix = []
44    for i in range(columns):
45      temp_new_row = []
46      for j in range(rows):
47        temp_new_row.append(matrix[j][i])
48      transpose_matrix.append(temp_new_row)
49
50    return transpose_matrix
51
52
53
54  def dot_product(v1, v2):
55    dot_product = 0
56    for i in range(len(row)):
57      dot_product += v1[i] * v2[i]
58    return dot_product
```

Chapter 3: First Order Logic

3.1 Languages

Definition 3.1.1: First-Order Language

A first order language $\mathcal{L}$ is an infinite collection of distinct symbols, no one of which is properly contained in another, sparated into the following cateogories:

1. *Parentheses: (,).*
2. *Connectives: $\vee, \neg$.*
3. *Quantifier: $\forall$.*
4. *Variables, one for each positive integer n: $v_1, v_2, \ldots, v_n, \ldots$ The set of variable symbols will be denoted $\mathcal{V}$.*
5. *Equality symbol: $=$.*
6. *Constant symbols: Some set of zero or more symbols. This set will be denoted as $\mathcal{C}$*
7. *Function symbols: For each positive integer n, some set of zero or more n-ary function symbols as $\mathcal{F}$.*
8. *Relation symbols: For each positive integer n, some set of zero or more n-ary relation symbols as $\mathcal{R}$.*

We may denote a language $\mathcal{L}$ as $(\mathcal{C}, \mathcal{F}, \mathcal{R})$ as these are the only parts of a the language which may vary (note that we may have different variables, but their usage and the fact that there are a countably infinite amount of them makes it so that we don't have to specify them). Also note that we may construct string, or ordered tuples of elements from a language, so for example we may write "$(v_1 = \vee$", and say that "$($" $\in$ "$(v_1 = \vee$", when we want to reference a string we use an equality symbol and a variable which aren't in the language so we may write $\gamma :\equiv (v_1 = \vee$ and say that $(\in \gamma$ (notice that we dropped the use of quotations as it should be clear now that these are strings).

Definition 3.1.2: Language of Number Theory

We define

$$\mathcal{L}_{NT} \stackrel{\text{D}}{=} \{\mathcal{C} = \{0\}, \mathcal{F} = \{S, +, \cdot, E\}, \mathcal{R} = \{<\}\}$$

Definition 3.1.3: Term

If $\mathcal{L}$ is a language, a term of $\mathcal{L}$ is a nonempty finite string t of symbols from $\mathcal{L}$ such that either:

1. *t is a variable, or*
2. *t is a constant symbol, or*
3. *$t :\equiv ft_1t_2\dots t_n$, where f is an n-ary function symbol of $\mathcal{L}$ and each of the t_i is a term of $\mathcal{L}$.*

Definition 3.1.4: Terms of a Language

We denote the set of all terms of a language as $\mathfrak{T}(\mathcal{L})$

Definition 3.1.5: Formula

If $\mathcal{L}$ is a first-order language, a formula of $\mathcal{L}$ is a nonempty finite string ϕ of symbols from $\mathcal{L}$ such that either:

1. *$\phi :\equiv= t_1t_2$, where t_1 and t_2 are terms of $\mathcal{L}$, or*
2. *$\phi :\equiv Rt_1t_2\dots t_n$, where R is an n-ary relation symbol of $\mathcal{L}$ and $t_1, t_2, \dots, t_n$ are all terms of $\mathcal{L}$, or*
3. *$\phi :\equiv (\neg\alpha)$, where α is a formula of $\mathcal{L}$, or*
4. *$\phi :\equiv (\alpha \vee \beta)$, where α and β are formulas of $\mathcal{L}$, or*
5. *$\phi :\equiv (\forall v)(\alpha)$, where v is a variable and α is a formula of $\mathcal{L}$.*

Definition 3.1.6: Formulas of a Language

We denote the set of all formula of a langauge $\mathcal{L}$ as $\mathfrak{F}(\mathcal{L})$

Definition 3.1.7: Subformula

A formula ϕ is a subformula of a formula ψ if ϕ appears as a substring of ψ

Definition 3.1.8: Scope

If $(\forall v)(\alpha)$ is a subformula of ϕ , we will say that the scope of the quantifier $\forall$ is α. If there are multiple $\forall$ in ϕ then each occurrence of the quantifier will have its own scope.

Definition 3.1.9: A Term's Variable Replaced by a Term

Suppose that u is a term, x is a variable, and t is a term. We define the term u_t^x (read "u with x replaced by t") as follows:

1. *If u is a variable not equal to x, then u_t^x is u.*
2. *If u is x, then u_t^x is t.*
3. *If u is a constant symbol, then u_t^x is u.*
4. *If $u :\equiv fu_1u_2 \dots u_n$, where f is an n-ary function symbol and the u_i are terms, then u_t^x is $f\,(u_1)_t^x\,(u_2)_t^x \dots (u_n)_t^x$*

Definition 3.1.10: A Formulas Variable Replaced by a Term

Suppose that ϕ is an $\mathcal{L}$-formula, t is a term, and x is a variable. We define the formula ϕ_t^x (read " ϕ with x replaced by t ") as follows:

1. *If $\phi :\equiv= u_1u_2$, then ϕ_t^x is $= (u_1)_t^x\,(u_2)_t^x$.*
2. *If $\phi :\equiv Ru_1u_2 \dots u_n$, then ϕ_t^x is $R\,(u_1)_t^x\,(u_2)_t^x \dots (u_n)_t^x$.*
3. *If $\phi :\equiv \neg(\alpha)$, then ϕ_t^x is $\neg\left(\alpha_t^x\right)$.*
4. *If $\phi :\equiv (\alpha \vee \beta)$, then ϕ_t^x is $\left(\alpha_t^x \vee \beta_t^x\right)$.*
5. *If $\phi :\equiv (\forall y)(\alpha)$, then*

$$\phi_t^x = \begin{cases} \phi & \text{if } x \text{ is } y \\ (\forall y)\left(\alpha_t^x\right) & \text{otherwise} \end{cases}$$

- In the fourth clause of the definition above and in the first two clauses of the next definition, the parentheses are not really there. Because $u_{1_t^x}$ is hard to read so the parentheses have been added in the interest of readability.
- If we let t be $g(c)$ and we let u be $f(x, y) + h(z, x, g(x))$, then u_t^x is

$$f(g(c), y) + h(z, g(c), g(g(c)))$$

Definition 3.1.11: A Term is Substitutable for a Variable

Suppose that ϕ is an $\mathcal{L}$-formula, t is a term, and x is a variable. We say that t is substitutable for x in ϕ if

1. *ϕ is atomic, or*
2. *$\phi :\equiv \neg(\alpha)$ and t is substitutable for x in α, or*
3. *$\phi :\equiv (\alpha \vee \beta)$ and t is substitutable for x in both α and β, or*
4. *$\phi :\equiv (\forall y)(\alpha)$ and either*
   - *x is not free in ϕ, or*
   - *y does not occur in t and t is substitutable for x in α.*

To understand the motivation behind the fourth clause, consider the formula:

$$\phi :\equiv (\forall x)\,(\forall y)\,x = y$$

Then one might want to say that $\left((\forall y)\,x = y\right)^x_t$ where t is any term

Definition 3.1.12: Free Variable in a Formula

Suppose that v is a variable and ϕ is a formula. We will say that v is free in ϕ if

1. *ϕ is atomic and v occurs in (is a symbol in) ϕ, or*
2. *$\phi :\equiv (\neg\alpha)$ and v is free in α, or*
3. *$\phi :\equiv (\alpha \vee \beta)$ and v is free in at least one of α or β, or*
4. *$\phi :\equiv (\forall u)(\alpha)$ and v is not u and v is free in α.*

Examples

- Thus, if we look at the formula

  $$\forall v_2 \neg\,(\forall v_3)\,\left(v_1 = S\,(v_2) \vee v_3 = v_2\right)$$

  the variable v_1 is free whereas the variables v_2 and v_3 are not free.

- A slightly more complicated example is

  $$\left(\forall v_1 \forall v_2\,(v_1 + v_2 = 0)\right) \vee v_1 = S(0)$$

  – In this formula, v_1 is free whereas v_2 is not free. Especially when a formula is presented informally, you must be careful about the scope of the quantifiers and the placement of parentheses.

Notes

- We will have occasion to use the informal notation $\forall x \phi(x)$. This will mean that ϕ is a formula and x is among the free variables of ϕ.
- If we then write $\phi(t)$, where t is an $\mathcal{L}$-term, that will denote the formula obtained by taking ϕ and replacing each occurrence of the variable x with the term t.

Definition 3.1.13: Variable Assignment Function

If $\mathfrak{A}$ is an $\mathcal{L}$-structure, a variable assignment function $s \in \operatorname{func}(\mathcal{V}, A)$ assigns variables to elements of the universe.

Definition 3.1.14: Term Assignment Function

Suppose that $\mathfrak{A}$ is an $\mathcal{L}$-structure and s is a variable assignment function into $\mathfrak{A}$. The function $\bar{s}$, called the term assignment function generated by s, is the function with domain consisting of the set of $\mathcal{L}$-terms and codomain A defined recursively as follows:

1. *If t is a variable, $\bar{s}(t) = s(t)$.*
2. *If t is a constant symbol c, then $\bar{s}(t) = c^{\mathfrak{A}}$.*
3. *If $t :\equiv f t_1 t_2 \dots t_n$, then $\bar{s}(t) = f^{\mathfrak{A}}\left(\bar{s}(t_1), \bar{s}(t_2), \dots, \bar{s}(t_n)\right)$.*

Definition 3.1.15: Interpretation of a Term

Let $t \in \mathfrak{T}(\mathcal{L})$, $\mathfrak{A}$ an $\mathcal{L}$-Structure, and $s \in \operatorname{func}(\mathcal{V}, A)$, then the interpretation of t is $\overline{s}(t)$

Definition 3.1.16: Satisfaction

Suppose that $\mathfrak{A}$ is an $\mathcal{L}$-structure, ϕ is an $\mathcal{L}$-formula, and $s : Vars \to A$ is an assignment function. We will say that $\mathfrak{A}$ satisfies ϕ with assignment s, and write $\mathfrak{A} \models \phi[s]$, in the following circumstances:

1. *If $\phi :\equiv= t_1 t_2$ and $\bar{s}(t_1)$ is the same element of the universe A as $\bar{s}(t_2)$, or*
2. *If $\phi :\equiv Rt_1 t_2 \dots t_n$ and $\left(\bar{s}(t_1), \bar{s}(t_2), \dots, \bar{s}(t_n)\right) \in R^{\mathfrak{A}}$, or*
3. *If $\phi :\equiv (\neg\alpha)$ and it's false that $\mathfrak{A} \models \alpha[s]$ or,*
4. *If $\phi :\equiv (\alpha \vee \beta)$ and $\mathfrak{A} \models \alpha[s]$, or $\mathfrak{A} \models \beta[s]$ (or both), or*
5. *If $\phi :\equiv (\forall x)(\alpha)$ and, for each element a of A, $\mathfrak{A} \models \alpha[s(x \mid a)]$.*

If Γ is a set of $\mathcal{L}$-formulas, we say that $\mathfrak{A}$ satisfies Γ with assignment s, and write $\mathfrak{A} \models \Gamma[s]$ if for each $\gamma \in \Gamma, \mathfrak{A} \models \gamma[s]$.

Definition 3.1.17: L-Structure

Fix a language $\mathcal{L}$. An $\mathcal{L}$-structure $\mathfrak{A}$ is a nonempty set A, called the universe of $\mathfrak{A}$, together with:

1. *For each constant symbol c of $\mathcal{L}$, an element $c^{\mathfrak{A}}$ of A,*
2. *For each n-ary function symbol f of $\mathcal{L}$, a function $f^{\mathfrak{A}} : A^n \to A$, and*
3. *For each n-ary relation symbol R of $\mathcal{L}$, an n-ary relation $R^{\mathfrak{A}}$ on A (i.e., a subset of A^n).*

We may denote an $\mathcal{L}$-Structure as follows

$$\left(A, \mathcal{C}^{\mathfrak{A}} \stackrel{\text{D}}{=} \left\{c^{\mathfrak{A}} : c \in \mathcal{C}\right\}, \mathcal{F}^{\mathfrak{A}} = \left\{f^{\mathfrak{A}} : f \in \mathcal{F}\right\}, \mathcal{R}^{\mathfrak{A}} = \left\{R^{\mathfrak{A}} : R \in \mathcal{R}\right\}\right)$$

Definition 3.1.18: Equivalent Formulas

We say that two formulas ϕ and ψ are equivalent when for any $\mathcal{L}$-Structure $\mathfrak{A}$ and assignment function s we have that

$$\mathfrak{A} \models \phi[s] \Leftrightarrow \mathfrak{A} \models \psi[s]$$

Proposition 3.1.1: Double Negation Equivalence

Let ϕ be a formula, then ϕ is equivalent to $\neg(\neg\phi)$

Proof

Let $\mathfrak{A}$ be $\mathcal{L}$-Structure, and let s be an assignment function, let's consider what it means for $\mathfrak{A} \models \neg(\neg\phi)$

$$\begin{aligned}\mathfrak{A} \models \neg(\neg\phi) &\Leftrightarrow \text{ it's false that } \mathfrak{A} \models \neg\phi \\ &\Leftrightarrow \text{ it's false that, it's false that } \mathfrak{A} \models \phi \\ &\Leftrightarrow \text{ it's true that } \mathfrak{A} \models \phi \\ &\Leftrightarrow \mathfrak{A} \models \phi\end{aligned}$$

And we know that if it's false that it's false, then it's true, because something is either true or false. Thus we've shown that $\mathfrak{A} \models \neg(\neg\phi)$ if and only if $\mathfrak{A} \models \phi$. ■

Definition 3.1.19: A Literal Structure

Let $\mathcal{L} = (\mathcal{V}, \mathcal{C}, \mathcal{F}, \mathcal{R})$ be a language, and $X \subseteq \mathfrak{F}(\mathcal{L})$, then a literal structure is:

$$\mathfrak{L}_X \overset{\mathtt{D}}{=} (X, \mathcal{C}, \mathcal{F}, \mathcal{R})$$

That is to say, our interpretation of any constant, function or relation symbol is that symbol itself.

Definition 3.1.20: Restricted L-Structure

Suppose $\mathcal{L} = (\mathcal{V}, \mathcal{C}, \mathcal{F}, \mathcal{R})$ and $\mathcal{L}' = (\mathcal{V}', \mathcal{C}', \mathcal{F}', \mathcal{R}')$ are languages and $\mathfrak{A}' = \left(A', \mathcal{C}'^{\mathfrak{A}'}, \mathcal{F}'^{\mathfrak{A}'}, \mathcal{R}'^{\mathfrak{A}'}\right)$ be an $\mathcal{L}'$-Structure, then we define

$$\mathfrak{A}' \restriction_{\mathcal{L}} \overset{\mathtt{D}}{=} \left(A', \left\{c^{\mathfrak{A}'} : c \in \mathcal{C}\right\}, \left\{f^{\mathfrak{A}'} : f \in \mathcal{F}\right\}, \left\{R^{\mathfrak{A}'} : R \in \mathcal{R}\right\}\right)$$

In other words we have just created an $\mathcal{L}$-Structure $\mathfrak{A}' \restriction_{\mathcal{L}}$ which intereprets everything the same was as $\mathfrak{A}'$ but it is now defined for $\mathcal{L}$.

3.2 Deductions

Definition 3.2.1: Deduction of a Formula

Suppose that Σ is a collection of $\mathcal{L}$-formulas and D is a finite sequence $(\phi_1, \phi_2, \ldots, \phi_n)$ of $\mathcal{L}$-formulas. We will say that D is a deduction from Σ if for each $i, 1 \leq i \leq n$, either

1. *$\phi_i \in \Lambda$ (ϕ_i is a logical axiom), or*
2. *$\phi_i \in \Sigma$ (ϕ_i is a nonlogical axiom), or*
3. *There is a rule of inference (Γ, ϕ_i) such that $\Gamma \subseteq \{\phi_1, \phi_2, \ldots, \phi_{i-1}\}$.*

If there is a deduction from Σ, the last line of which is the formula ϕ, we will call this a deduction from Σ of ϕ, and write $\Sigma \vdash \phi$.

Definition 3.2.2: Logical Axioms

Given a first order language $\mathcal{L}$ the set Λ of logical axioms is the collection of all $\mathcal{L}-$formulas that fall into one of the following categories:

$$x = x \text{ for each variable } x$$
$$\left[(x_1 = y_1) \wedge (x_2 = y_2) \wedge \cdots \wedge (x_n = y_n)\right] \rightarrow \left(f(x_1, x_2, \ldots, x_n) = f(y_1, y_2, \ldots, y_n)\right)$$
$$\left[(x_1 = y_1) \wedge (x_2 = y_2) \wedge \cdots \wedge (x_n = y_n)\right] \rightarrow \left(R(x_1, x_2, \ldots, x_n) \rightarrow R(y_1, y_2, \ldots, y_n)\right)$$
$$(\forall x \phi) \rightarrow \phi_t^x \text{ if } t \text{ is substitutable for } x \text{ in } \phi$$
$$\phi_t^x \rightarrow (\exists x \phi) \text{ if } t \text{ is substitutable for } x \text{ in } \phi$$

To refer to them easily we label them by moving down the above list E1, E2, E3, Q1, Q2

Lemma 3.2.1: Universal connection to Variable Assignment Function

$\Sigma \vdash \theta$ *if and only if* $\Sigma \vdash \forall x \theta$

Proof

- $\Rightarrow$
  - Suppose that $\Sigma \vdash \theta$, therefore we have a deduction $\mathcal{D}$ of θ, then the

proof

$$\begin{array}{rr} \mathcal{D} & \\ \Big[\big(\forall y\,(y=y)\big) \wedge \neg\big(\forall y\,(y=y)\big)\Big] \to \theta & \text{(taut. PC)} \\ \Big[\big(\forall y\,(y=y)\big) \wedge \neg\big(\forall y\,(y=y)\big)\Big] \to (\forall x)\,\theta & \text{(QR)} \\ (\forall x)\,\theta & \text{(PC)} \end{array}$$

- $\Leftarrow$
  - Suppose that $\Sigma \vdash \forall x\theta$, so we have a deduction of it, call it $\mathcal{D}$, then the following deduction suffices

$$\begin{array}{c} \mathcal{D} \\ \forall x\theta \\ \forall x\theta \to \theta^x_x \\ \theta^x_x \end{array}$$

■

3.3 Completeness

Definition 3.3.1: Consistent Set of L Formulas

Let Σ be a set of $\mathcal{L}$-formulas. We will say that Σ is inconsistent if there is a deduction from Σ of $[(\forall x)x = x] \wedge \neg[(\forall x)x = x]$. We say that Σ is consistent if it is not inconsistent.

Proposition 3.3.1: Contradiction has no Model

There is no $\mathcal{L}$-Structure $\mathfrak{A}$ such that $\mathfrak{A} \models \bot$

Proof

Suppose there was a model of $\bot$, that would mean that we have an $\mathcal{L}$-Structure $\mathfrak{A}$ such that $\mathfrak{A} \models \bot$. Recall that $\bot :\equiv \big[(\forall x)\,x = x\big] \wedge \neg\big[(\forall x)\,x = x\big]$, so then we would have to have $\mathfrak{A} \models (\forall x)\,x = x$ and also not have $\mathfrak{A} \models (\forall x)\,x = x$ which is a contradiction. ■

Proposition 3.3.2: Implying a Contradiction Means no Model

Let Σ be a set of sentences, then if

$$\Sigma \models \bot$$

then Σ has no model

Proof

- Recall that $\Sigma \models \bot$ means that for any $\mathcal{L}$-Structure $\mathfrak{A}$ we have that

  $$\mathfrak{A} \models \Sigma \text{ implies } \mathfrak{A} \models \bot$$

  but recall that $\mathfrak{A} \models \bot$ is always false, therefore for the implication to hold we require that $\mathfrak{A} \models \Sigma$ to be false, which means Σ has no model as $\mathfrak{A}$ was arbitrary.

■

Proposition 3.3.3: Logical Contradiction

Let Σ be a set of sentences, and let ϕ be a sentence, then

$$\Sigma \models \phi \textit{ implies } \Sigma \cup \{\neg\phi\} \models \bot$$

Proof

- Suppose that $\Sigma \models \phi$ that means for any $\mathcal{L}$-Structure $\mathfrak{A}$ if $\mathfrak{A} \models \Sigma$ then $\mathfrak{A} \models \phi$ we want to show that $\Sigma \cup \{\neg\phi\} \models \bot$, recall that by the previous theorem that means that $\Sigma \cup \{\neg\phi\}$ has no model.
- Suppose for the sake of contradiction that $\Sigma \cup \{\neg\phi\}$ has a model $\mathfrak{B}$ then certainly $\mathfrak{B} \models \Sigma$ and by assuption we have that $\mathfrak{B} \models \phi$ but also we know that $\mathfrak{B} \models \neg\phi$ which is a contradiction, so therefore $\Sigma \cup \{\neg\phi\}$ has no model and therefore

  $$\Sigma \cup \{\neg\phi\} \models \bot$$

  as needed.

■

Lemma 3.3.1: Constant Replacement Removes It

Let ϕ be an $\mathcal{L}$ formula, then for any $c \in \phi \cap \mathcal{C}$ and any $t \in \mathfrak{T}(\mathcal{L})$ where $c \notin t$ then

$$c \notin \phi_t^c$$

Proof

The proof is by structural induction on ϕ. Since formulas are constructed from terms, it must be shown on terms first:

- Base Case
  - Suppose $u :\equiv x$ where x is some variable, then $x_u^c :\equiv x$ and so $c \notin x$, because $\mathcal{C} \cap \mathcal{V} = \varnothing$
  - If u is a constant but $c \notin u$ then the subtitution doesn't change anything as above, but if $u :\equiv c$ then $u_t^c :\equiv t$ and so $c \notin u_t^c$
- Induction Step
  - Let $t_1, t_2, \ldots, t_{n-1}, t_n$ be terms where the claim holds, we will show the claim holds on
    $$f t_1, t_2, \ldots, t_{n-1}, t_n$$
  - Note that $\left(f t_1, t_2, \ldots, t_{n-1}, t_n\right)_t^c :\equiv f\left(t_1\right)_t^c, \left(t_2\right)_t^c, \ldots, \left(t_{n-1}\right)_t^c, \left(t_n\right)_t^c$, and that $c \notin f$, therefore $c \notin f\left(t_1\right)_t^c, \left(t_2\right)_t^c, \ldots, \left(t_{n-1}\right)_t^c, \left(t_n\right)_t^c :\equiv f\left(t_1\right)_t^c, \left(t_2\right)_t^c, \ldots, \left(t_{n-1}\right)_t^c, \left(t_n\right)_t^c$ as needed.

Now for the formulas

- Base Case
  - For atomic formulas, we know that they are created out of terms, and symbols which don't change under replacement (namely $=$ and relation symbols) therefore since the claim holds on terms, and that the newly added symbols don't change and our induction hypothesis holds for α , β the claim holds.
- Induction Step
  - Suppose that the claim holds on formulas α, β then it also holds for $\neg\alpha$ and $\alpha \vee \beta$ since the new characters added to the strings are $\vee$ and $\neg$ which don't change under replacement
  - When we have $(\forall x)\, \phi$ we can be sure that $x :\not\equiv c$ as $x \in \mathcal{V}$ and $c \in \mathcal{C}$ and therefore by the definition of replacement we have that $\left((\forall x)\, \alpha\right)_t^c :\equiv (\forall x)\, \alpha_t^c$ and then by the induction hypothesis we have that $c \notin (\forall x)\, \alpha$

■

Lemma 3.3.2: Constant Extension still Consistent

If Σ is a consistent set of $\mathcal{L}$-sentences and $\mathcal{L}'$ is an extension by constants of $\mathcal{L}$, then Σ is consistent when viewed as a set of $\mathcal{L}'$-sentences.

Proof

- Suppose for the sake of contradiction that Σ is not consistent when viewed as a set of $\mathcal{L}'$-sentences, so $\Sigma \vdash \bot$, considering the set of all deductions of $\bot$ from Σ we may find a deduction $\mathcal{D}$ which uses the least number of the newly added constants by the well ordering principle, let this number be $n \in \mathbb{N}$ and that $n > 0$ or else we would have a deduction from $\mathcal{L}$ of $\bot$ which would be a contradiction.
- Let v be a variable that isn't used in $\mathcal{D}$, note that we can do this because there are infinitely many variables, but only finitely many of them can be used in a deduction and let c be one of the newly added constants which is used in $\mathcal{D}$ and let $\mathcal{D}_v^c$ be created where for each line $\phi \in \mathcal{D}$ we replace it with ϕ_v^c , and note that the last line of $\mathcal{D}_v$ is still $\bot$, at this point we don't know if $\mathcal{D}_v$ is a deduction, so we have to check that it is.
  - Note that if it is a deduction, then we've arrived at a contradiction since we have a deduction of $\bot$ with a number of newly added constants which is less than n (and n is minimal).
- If ϕ is an equality axiom or an element of Σ then $\phi_v :\equiv \phi$ because equality axioms only contain variables and not constants, also Σ is a set of $\mathcal{L}$ sentences and so it can't contain any of the new constants. Therefore equality axioms and any sentence from Σ is not modified by this procedure, leaving them as valid steps in the deduction.
- If ϕ is $(\forall x)\, \theta \to \theta_t^x$ then ϕ_v is $(\forall x)\, \theta_v \to (\theta_v)_{t_v}^x$, to see why we use t_v as well as θ_v try $\theta :\equiv c = x$ and $t :\equiv c + 3$

■

Notice that if we write $\Sigma \models \bot$ it means that for any $\mathcal{L}$-Structure $\mathfrak{A}$ if $\mathfrak{A} \models \Sigma$ then $\mathfrak{A} \models \bot$ by the above discussion that forces $\mathfrak{A} \models \Sigma$ to be false, and therefore Σ has no model.

Theorem 3.3.1: Completeness Theorem

Suppose that Σ is a set of $\mathcal{L}$-formulas, where $\mathcal{L}$ is a countable langauge and ϕ is an $\mathcal{L}$-formula. If $\Sigma \models \phi$, then $\Sigma \vdash \phi$.

Setup

- *We start by assuming that $\Sigma \models \phi$, we must show that $\Sigma \vdash \phi$.*
- *If ϕ is not a sentence then we can always prove ϕ' which is the same as ϕ with all of it's variables bound*
  - *We can do that by appending $\left(\forall x_f\right)$ where each x_f is a free varaible of ϕ to the front of ϕ*
- *Therefore we will prove it for all sentences ϕ*

3.4 Incompleteness

Definition 3.4.1: Bounded Quantifier Abbreviations

If x is a variable that does not occur in the term t, let us agree to use the following abbreviations:

$$(\forall x < t)\phi \text{ means } \forall x(x < t \to \phi)$$
$$(\forall x \leq t)\phi \text{ means } \forall x((x < t \vee x = t) \to \phi)$$
$$(\exists x < t)\phi \text{ means } \exists x(x < t \wedge \phi)$$
$$(\exists x \leq t)\phi \text{ means } \exists x((x < t \vee x = t) \wedge \phi)$$

These abbreviations will constitute the set of bounded quantifiers.

Definition 3.4.2: Sigma Formula

The collection of Σ-formulas is defined as the smallest set of $\mathcal{L}_{NT}$ formulas such that:

1. *Every atomic formula is a Σ-formula.*
2. *Every negation of an atomic formula is a Σ-formula.*
3. *If α and β are Σ-formulas, then $\alpha \wedge \beta$ and $\alpha \vee \beta$ are both Σ-formulas.*
4. *If α is a Σ-formula, and x is a variable that does not occur in the term t, then the following are Σ-formulas: $(\forall x < t)\alpha, (\forall x \leq t)\alpha$, $(\exists x < t)\alpha, (\exists x \leq t)\alpha$*
5. *If α is a Σ-formula and x is a variable, then $(\exists x)\alpha$ is a Σ-formula.*

Definition 3.4.3: Pi Formula

The collection of Π-formulas is the smallest set of $\mathcal{L}_{NT}$ formulas such that:

1. *Every atomic formula is a Π-formula.*
2. *Every negation of an atomic formula is a $\Pi-$formula.*
3. *If α and β are $\Pi-$formulas, then $\alpha \wedge \beta$ and $\alpha \vee \beta$ are both $\Pi-$formulas.*
4. *If α is a Π-formula, and x is a variable that does not occur in the term t, then the following are $\Pi-$formulas: $(\forall x < t)\alpha, (\forall x \leq t)\alpha$ $(\exists x < t)\alpha, (\exists x \leq t)\alpha$*
5. *If α is a Π-formula and x is a variable, then $(\forall x)\alpha$ is a Π-formula.*

Proposition 3.4.1: Negation of a Sigma Formula is a Pi Formula

Suppose σ is a Σ-formula, then $\neg\sigma$ is equivalent a Π-formula

Proof

We will proceed by induction on the complexity of the formula, in other words, we will do structural induction

- Base Case
  - Suppose ϕ is an atomic Σ-formula, that means that it is an atomic formula in the sense of a normal formula, therefore it's negation is the negation of an atomic formula and is therefore a Π-formula
  - Suppose $\phi :\equiv \neg\alpha$ where α is an atomic formula then by the fact that double negation "cancels" we see that ϕ is equivalent to α which is an atomic formula, and therefore α is a Π-formula

- Induction Step
  - Suppose α and β are Σ-formulas where the claim holds
    * $\phi :\equiv \alpha \vee \beta$, then $\neg\phi$ is equivalent to $\neg\alpha \wedge \neg\beta$ and therefore by our induction hypothesis we know that $\neg\alpha$ and $\neg\beta$ are equivent to some Π-formulas π_α, π_β then ϕ is equivalent to $\pi_\alpha \wedge \pi_\beta$ which is a Π-formula by the third clause.
    * $\phi :\equiv \alpha \wedge \beta$ then $\neg\phi$ is equivalent to $\neg\alpha \vee \neg\beta$ and by the same argument above, we can see that this is a Π-formula
    * $\phi :\equiv (\forall x < t)\,\alpha$, then it's negation is $(\exists x)\,(x < t \wedge \neg\alpha)$ by our induction hypothesis we know that $\neg\alpha$ is equivalent to some Π-formula $\pi_{\neg\alpha}$ so $(\exists x)\,(x < t \wedge \neg\alpha)$is equivalent to $(\exists x)\,(x < t \wedge \pi_{\neg\alpha}) \equiv (\exists x < t)\,\pi_{\neg\alpha}$ and therefore we may say that ϕ is equivalent to a Π-formula.
    * Suppose $\phi :\equiv (\forall x \leq t)$ then $\neg\phi$ is equivalent to $(\exists x) \neg \big((x < t \vee x = t) \to \alpha\big) :\equiv (\exists x) \neg \big(\neg\,(x < t \vee x = t) \vee \alpha\big)$ which is equivalent to $(\exists x)\,\big((x < t \vee x = t) \wedge \neg\alpha\big) :\equiv (\exists x \leq t)\,\neg\alpha$ and since $\neg\alpha$ is equivalent to some Π-formula, $\pi_{\neg\alpha}$ by our inductive hypothesis, we know that $(\exists x \leq t)\,\pi_{\neg\alpha}$ is a Π-formula.
    * Suppose $\phi :\equiv (\exists x < t)\,\alpha$ then the negation is equivalent to $(\forall x) \neg \big(\neg\,(x < t) \vee \neg\alpha\big) :\equiv (\forall x)\,(x < t \to \neg\alpha) :\equiv (\forall x < t)\,\neg\alpha$, and following similar logic as above we can see that $(\forall x < t)\,\neg\alpha$ is a Π-formula
    * $\phi :\equiv (\exists x \leq t)\,\alpha$ the negation is equivalent to $(\forall x)\,\Big(\neg\,\big((x < t) \vee (x = t) \vee \neg\alpha\big)\Big) :\equiv (\forall x)\,\big((x < t) \vee (x = t)\big) \to \neg\alpha :\equiv (\forall x \leq t)\,\neg\alpha$ as needed.

■

Definition 3.4.4: Representable Set

A set $A \subseteq \mathbb{N}^k$ is said to be representable (in N) if there is an $\mathcal{L}_{NT}$-formula $\phi(\underline{x})$ such that

$$\begin{array}{ll} \forall a \in A & N \vdash \phi(\bar{a}) \\ \forall \underline{b} \notin A & N \vdash \neg\phi(\bar{b}) \end{array}$$

In this case we will say that the formula ϕ represents the set A.

Definition 3.4.5: Weakly Representable Set

A set $A \subseteq \mathbb{N}^k$ is said to be weakly representable (in N) if there is an $\mathcal{L}_{NT}$-formula $\phi(x)$ such that

$$\begin{array}{ll} \forall a \in A & N \vdash \phi(\bar{a}) \\ \forall b \notin A & N \nvdash \phi(\bar{b}) \end{array}$$

In this case we will say that the formula ϕ weakly represents the set A.

Definition 3.4.6: Total Function

Suppose that $A \subseteq \mathbb{N}^k$ and suppose that $f : A \to \mathbb{N}$. If $A = \mathbb{N}^k$ we will say that f is a total function

Definition 3.4.7: Partial Function

Suppose that $A \subsetneq \mathbb{N}^k$ and suppose that $f : A \to \mathbb{N}$, then we will call f a partial function.

Definition 3.4.8: Representable Function

Suppose that $f : \mathbb{N}^k \to \mathbb{N}$ is a total function. We will say that f is a representable function (in N) if there is an $\mathcal{L}_{NT}$ formula $\phi\left(x_1, \ldots, x_{k+1}\right)$ such that, for all $a_1, a_2, \ldots a_{k+1} \in \mathbb{N}$

- *If $f\left(a_1, \ldots, a_k\right) = a_{k+1}$, then $N \vdash \phi\left(\overline{a_1}, \ldots, \overline{a_{k+1}}\right)$*
- *If $f\left(a_1, \ldots, a_k\right) \neq a_{k+1}$, then $N \vdash \neg\phi\left(\overline{a_1}, \ldots, \overline{a_{k+1}}\right)$*

Definition 3.4.9: Definable Set

We will say that a set $A \subseteq \mathbb{N}^k$ is definable if there is a formula $\phi(x)$ such that

$$\begin{array}{ll} \forall a \in A & \mathfrak{N} \models \phi(\bar{a}) \\ \forall \underline{\downarrow b} \notin A & \mathfrak{N} \models \neg\phi(\bar{b}) \end{array}$$

In this case, we will say that ϕ defines the set A.

Corollary 3.4.1: I

$A \subseteq \mathbb{N}^k$ is definable by a Δ formula, then it is representable

Chapter 4: Topology

4.1 Topological Spaces and Continuous Functions

4.1.1 Basis for a Topology

Definition 4.1.1: Basis

If X is a set, a basis for a topology on X is a collection $\mathcal{B}$ of subsets of X (called basis elements) such that

1. *For each $x \in X$, there is at least one basis element B containing x.*
2. *If x belongs to the intersection of two basis elements B_1 and B_2, then there is a basis element B_3 containing x such that $B_3 \subset B_1 \cap B_2$.*

If $\mathcal{B}$ satisfies these two conditions, then we define the topology $\mathcal{T}$ generated by $\mathcal{B}$ as follows: A subset U of X is said to be open in X (that is, to be an element of $\mathcal{T}$) if for, each $x \in U$, there is a basis element $B \in \mathcal{B}$ such that $x \in B$ and $B \subset U$. Note that each basis element is itself an element of $\mathcal{T}$.

4.1.2 The Subspace Topology

Definition 4.1.2: Subspace Topology

Let X be a topological space with topology $\mathcal{T}$. If Y is a subset of X, the collection

$$\mathcal{T}_Y = \{Y \cap U \mid U \in \mathcal{T}\}$$

is a topology on Y, called the subspace topology. With this topology, Y is called a subspace of X; its open sets consist of all intersections of open sets of X with Y.

4.1.3 The Product Topology

Definition 4.1.3: Product Topology

Let $\mathcal{S}_\beta$ denote the collection

$$\mathcal{S}_\beta = \left\{ \pi_\beta^{-1}\left(U_\beta\right) \mid U_\beta \text{ open in } X_\beta \right\}$$

and let $\mathcal{S}$ denote the union of these collections,

$$\mathcal{S} = \bigcup_{\beta \in J} \mathcal{S}_\beta$$

The topology generated by the subbasis $\mathcal{S}$ is called the product topology. In this topology $\prod_{\alpha \in J} X_\alpha$ is called a product space.

Definition 4.1.4: Box Topology

Let $\{X_\alpha\}_{\alpha \in J}$ be an indexed family of topological spaces. Let us take as a basis for a topology on the product space

$$\prod_{\alpha \in J} X_\alpha$$

the collection of all sets of the form

$$\prod_{\alpha \in J} U_\alpha$$

where U_α is open in X_α, for each $\alpha \in J$. The topology generated by this basis is called the box topology.

Theorem 4.1.1: Basis for the Box Topology

Suppose the topology on each space X_α is given by a basis $\mathcal{B}_\alpha$. The collection of all sets of the form

$$\prod_{\alpha \in \mathcal{J}} B_\alpha$$

where $B_\alpha \in \mathcal{B}_\alpha$ for each α, will serve as a basis for the box topology on $\prod_{\alpha \in \mathcal{J}} X_\alpha$.

Theorem 4.1.2: Basis for the Product Topology

Suppose the topology on each space X_α is given by a basis $\mathcal{B}_\alpha$. The collection of all sets of the form

$$\prod_{\alpha\in\mathcal{J}} B_\alpha$$

where $B_\alpha \in \mathcal{B}_\alpha$ for finitely many indices α and $B_\alpha = X_\alpha$ for all the remaining indices, will serve as a basis for the product topology $\prod_{\alpha\in\mathcal{J}} X_\alpha$.

Definition 4.1.5: R Omega

$\mathbb{R}^\omega$, the countably infinite product of $\mathbb{R}$ with itself. Recall that

$$\mathbb{R}^\omega = \prod_{n\in\mathbb{N}} X_n$$

with $X_n = \mathbb{R}$ for each n

4.1.4 The Metric Topology

Definition 4.1.6: A metric

A metric on a set X is a function

$$d : X \times X \to \mathbb{R}$$

having the following properties:

1. *$d(x, y) \geq 0$ for all $x, y \in X$; equality holds if and only if $x = y$.*
2. *$d(x, y) = d(y, x)$ for all $x, y \in X$.*
3. *Triangle Inequality: $d(x, y) + d(y, z) \geq d(x, z)$, for all $x, y, z \in X$.*

Example 4.1.1: Discrete Metric

$d : X \times X \to \mathbb{R}$ given by

$$d(x, y) = \begin{cases} 0 & x = y \\ 1 & \text{otherwise} \end{cases}$$

Definition 4.1.7: Epsilon Ball

Given $\epsilon > 0$, consider the set

$$B_d(x, \epsilon) = \{y \mid d(x, y) < \epsilon\}$$

of all points y whose distance from x is less than ϵ. It is called the ϵ-ball centered at $\boldsymbol{x}$. Sometimes we omit the metric d from the notation and write this ball simply as $B(x, \epsilon)$, when no confusion will arise.

Lemma 4.1.1: Epsilon Ball Contains Another

Let $x \in X$ then for every $B(x, \varepsilon)$ there is $y \in B(x, \varepsilon)$ and $\delta \in \mathbb{R}^+$ such that

$$B(y, \delta) \subseteq B(x, \varepsilon)$$

Proof

- Let $y \in B(x, \varepsilon)$ we claim $B(y, \delta)$ where $\delta = \varepsilon - d(x, y)$ works.
- Note $B(y, \delta) \subseteq B(x, \varepsilon)$, since given any $z \in B(y, \delta)$ we have that $d(y, z) < \varepsilon - d(x, y)$, re-arranging gives us $d(x, y) + d(y, z) < \varepsilon$ and by the triangle inequality we get $d(x, z) < \varepsilon$ therefore $z \in B(x, \varepsilon)$

■

Proposition 4.1.1: Epsilon Balls Form A Basis

The collection $\mathcal{B}$ of all ϵ-balls $B_d(x, \epsilon)$, for $x \in X$ and $\epsilon > 0$, is a basis for a topology on X

Proof

- Let $x \in X$ then clearly $x \in B_d(x, \varepsilon)$ works for any $\varepsilon \in \mathbb{R}^+$
- Let $B_1, B_2 \in \mathcal{B}$ and and let $y \in B_1 \cap B_2$, for each of B_1 and B_2 we have δ_1 and δ_2 respectively with $B(y, \delta_1) \subseteq B_1$ and $B(y, \delta_2) \subseteq B_2$, by taking $\delta = \min(\delta_1, \delta_2)$ then we have $B(y, \delta) \subseteq B_1 \cap B_2$ as needed.

■

Definition 4.1.8: Metric Topology

If d is a metric on the set X, then the collection of all ϵ-balls $B_d(x, \epsilon)$, for $x \in X$ and $\epsilon > 0$, is a basis for a topology on X, called the metric topology induced by d.

Definition 4.1.9: Bounded Subset of a Metric Space

Let X be a metric space with metric d. A subset A of X is said to be bounded if there is some number $M \in \mathbb{R}$ such that

$$d(a_1, a_2) \leq M$$

for every pair of points $a_1, a_2 \in A$

Example 4.1.2: Function on R omega

Consider a function $h : \mathbb{R}^\omega \to \mathbb{R}^\omega$ defined by

$$h(x_1, x_2, \ldots) = (\alpha_1 x_1, \alpha_2 x_2, \ldots)$$

with $\alpha_1, \alpha_2, \ldots \in \mathbb{R} \setminus \{0\}$

1. *Is h continuous, when $\mathbb{R}^\omega$ is given the product topology?*
2. *Is h continuous, when $\mathbb{R}^\omega$ is given the box topology?*
3. *Is h continuous, when $\mathbb{R}^\omega$ is given the uniform topology?*

Proof

1. • Take an arbitrary basis element from the product topology, that is:

$$\prod_{\alpha \in \mathcal{J}} B_\alpha$$

where $B_\alpha \in \mathcal{B}_\mathbb{R}$ for finitely many indices α and $B_\alpha = \mathbb{R}$ for all the remaining indices.

α : Now note the following

– The inverse image of each of these intervals is a scaled version of itself therefore it is still an interval of $\mathbb{R}$, and it is therefore still open with respect to $\mathbb{R}$

– The inverse image of $\mathbb{R}$ under any scaling is still $\mathbb{R}$

• Therefore $h^{-1}\left(\prod_{\alpha \in \mathcal{J}} B_\alpha\right)$ is a product of finitely many intervals, and infinitely many $\mathbb{R}$'s and so it is open with respect to $\mathbb{R}^\omega$ as needed.

2. • Given an arbitrary basis element of the box topology we have

$$\prod_{\alpha\in\mathcal{J}} B_\alpha$$

where $B_\alpha \in \mathcal{B}_{\mathbb{R}}$ for each α

• Due to α we know that each of the intervals become new scaled intervals and so $h^{-1}\left(\prod_{\alpha\in\mathcal{J}} B_\alpha\right)$ is a product of open intervals and is therefore open in with respect $\mathbb{R}^\omega$ equipped with the box topology

3. Recall that a basis for the uniform topology are epsilon balls with radius less then 1 if that's the case, then so long as every α_i is greater than 1 then the inverse image scales them to be even smaller, resulting in an open set, otherwise the function h wouldn't be continuous, since all the α_i's are greater than one, it is continouous.

■

4.2 Connectedness and Compactness

Definition 4.2.1: Connected Space

Let X be a topological space. A separation of X is a pair U, V of disjoint nonempty open subsets of X whose union is X. The space X is said to be connected if there does not exist a separation of X.

Notice that U, V are actually clopen, as $X \setminus U = V$ and $X \setminus V = U$ stating that V and U are closed as well.

Example 4.2.1: Closed and Bounded, not Compact

A metric space X and a closed and bounded subspace Y of X that is not compact.

- Consider the set $X = \left\{\frac{1}{n} : n \in \mathbb{N}^+\right\}$, with the discrete metric, it is bounded because the for any two points $a, b \in X, d(a, b) \leq 1$
- Let X be an infinite set and let consider the discrete metric on that set, the metric topology which it induces (call it $\mathcal{T}$) is the discrete topology of X. Therefore if we consider any subset Y of X it is closed, as $X \setminus Y \in \mathcal{T}$ (remember it's the discrete topology). But the open covering $\{\{x\} : x \in X\}$ has no finite subcollection which also covers X.

Example 4.2.2: R omega Connected?

Consider the product, uniform, and box topologies on $\mathbb{R}^\omega$. In which topologies is $\mathbb{R}^\omega$ connected?

Proof

1.
   - Consider the product topology, we will show that $\mathbb{R}^\omega$ is not connected. Consider the set A of bounded sequences in $\mathbb{R}^\omega$, and the complement of A, namely the unbounded sequence of $\mathbb{R}^\omega$ let's label this set as $B = \mathbb{R}^\omega \setminus A$.
     - Note that A is open in the box topology as if we fix any $\varepsilon > 0$ we may define for any $\vec{x} \in \mathbb{R}^\omega$
     $$U_{\vec{x}} \stackrel{\text{D}}{=} (x_1 - \varepsilon, x_1 + \varepsilon) \times (x_2 - \varepsilon, x_2 + \varepsilon) \times \dots$$
     - If $\vec{a}$ is a bounded sequence then U is a set of bounded sequences, as it only ever adds constants to the terms of $\vec{a}$, therefore $\vec{a} \in U_{\vec{a}}$ and $U_{\vec{a}} \subseteq A$, therefore A is open.
     - If $\vec{b}$ is an unbounded sequence then U is a set of unbounded sequences, as adding a constant to every element to an unbounded sequence yields an unbounded sequence. So for every $\vec{b} \in B$ there we have $\vec{b} \in U_{\vec{b}}$ and $U_{\vec{b}} \subseteq B$ so B is open.
   - Therefore A non-trivial set which is both open and closed in $\mathbb{R}^\omega$ and so it is disconnected.

2. Note that $\mathbb{R}^\omega$ is closed with the uniform topology by following the same proof with $\varepsilon = 1$

3.
   - Now if we consider $\mathbb{R}^\omega$ with the product topology we will see that $\mathbb{R}^\omega$ is connected.
   - Let $\mathbb{R}^n_0 \stackrel{\text{D}}{=} \left\{(x_1, x_2, \dots) : \text{ where for } i > n \text{ we have that } x_i = 0\right\}$
   - $\mathbb{R}^n_0$ is homeomorphic to $\mathbb{R}^n$ and since $\mathbb{R}$ is connected, and the finite cartesian product of connected spaces is connected, we get that $\mathbb{R}^n_0$ is connected.
   - We now will show that the closure of $\mathbb{R}^\infty$ is equal to $\mathbb{R}^\omega$ and then by the fact that the closure of a connected set is closed we obtain that $\mathbb{R}^\omega$ is connected.
   - To show that R^ω is equal to the closure of $\mathbb{R}^\infty$ we proceed by using the fact that a point is part of the closure of a set if and only if every basis element intersects it:

– Let $\vec{a} \in \mathbb{R}^\omega$ and let $B = \prod_{i\in\mathbb{N}} B_i$ be some basis element for the product topology with $\vec{a} \in U$, now we have to show this set intersects $\mathbb{R}^\infty$. Since only finitely many of the B_i are equal to basis elements and the rest are equal to $\mathbb{R}$ that means there is some $N \in \mathbb{N}$ such that $\forall j \in \mathbb{N}^{\geq N}, B_j = \mathbb{R}$, therefore the point $\vec{b} = (a_1, a_2, \ldots, a_{N-1}, a_N, 0, 0, 0, \ldots) \in B \cap \mathbb{R}^\infty$

■

Proposition 4.2.1: Connected Implies Closure Connected

Let $A \subseteq X$ be a connected subspace of X then $\overline{A}$ is also connected.

Proof

- Suppose for the sake of contradiction that there is a separation B, C of $\overline{A}$, then $B \cup C = \overline{A}$ and note that that means that $A \subseteq B \cup C$ since $A \subseteq \overline{A}$ so $(B \cup C) \cap A = \overline{A} \cap A = A$.
- Therefore $(B \cap A) \cup (C \cap A) = A$ is a separation of A noting that $B \cap A$ and $C \cap A$ are non-empty because ...

■

Definition 4.2.2: Totally Disconnected

A topological space is totally disconnected if it's only connected subspaces are one-point sets.

Consider $\mathbb{R}_\ell$ if we have $\{a\}$ and $\{b\}$ then the open sets $(-\infty, b)$ and $[b, \infty)$ is a separation of $\mathbb{R}_\ell$, therefore it's disconnected. Similarly for any two points a, b in $\mathbb{Q}$ we have some irrational number r between the two, and thus $(-\infty, r)_\mathbb{Q}, (r, \infty)_\mathbb{Q}$ is a separation thus $\mathbb{Q}$ is totally disconnected under this topology.

Let's look at $\mathbb{R}$ with the fite complement topology. Right off the bat, we note that if a set is finite in $\mathbb{R}$ it's complement must be infinite therefore if $\mathbb{R}$ was completely disconnected it would mean for any singleton sets, we have a separation U, V, but that means that U and V must be infinite, but we then get a contraditiction as $\mathbb{R} = U \cup V$ so $R \setminus U = V$, now since U was open this implies that V is finite, which is a contradiction. This idea may be extended to $\mathbb{R}^2$.

4.2.1 Compact Spaces

Definition 4.2.3: Covering

A collection A of subsets of a space X is said to cover X, or to be a covering of X, if the union of the elements of A is equal to X. It is called an open covering of X if its elements are open subsets of X.

Definition 4.2.4: Compact Space

A space X is said to be compact if every open covering A of X contains a finite subcollection that also covers X.

Lemma 4.2.1: Covering Yields Finite Covering if and only if Compact

Let Y be a subspace of X. Then Y is compact if and only if every covering of Y by sets open in X contains a finite subcollection covering Y.

Chapter 5: Computer Science

Definition 5.0.1: Big-O

Let $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$*, then we define the set*

$$\mathcal{O}(g) \stackrel{\mathtt{D}}{=} \left\{ j : \exists c, B \in R^{+}, \forall n \in \mathbb{N}, n \geq B \to j(n) \leq cg(n) \right\}$$

And we say that f *is in the big-O of* g *when* $f \in \mathcal{O}(g)$ *.*

Definition 5.0.2: Little-o

We define the following set:

$$o(g) = \left\{ j : \lim_{x \to 0} \frac{j(x)}{g(x)} = 0 \right\}$$

We say that f *is in the little-o of* g *when* $f \in o(g)$

Chapter 6: Graph Theory

Proposition 6.0.1: Maximum Number of Edges

Let $G = (V, E)$ be a simple graph such that $|V| = n$ and $|E| = m$, then we have:

$$m \leq \binom{n}{2}$$

Proof

Given $V = \{v_1, v_2, \ldots, v_{n-1}, v_n\}$ an edge could exist between any two vertices. Since there are $\binom{n}{2}$ distinct subsets of V of size 2 we can see that $\binom{n}{2}$ is an upper bound on the edges, thus we have

$$m \leq \binom{n}{2}$$

■

Proof

Consider the adjacency matrix of G at most the matrix could be

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & \ldots & 1 & 1 \\ 1 & 0 & \ldots & 1 & 1 \\ \vdots & & \ddots & & \vdots \\ 1 & 1 & \ldots & 0 & 1 \\ 1 & 1 & \ldots & 1 & 0 \end{bmatrix}$$

which is an $n \times n$ matrix. The reason why is that this means that every two nodes are connected aside from loops (the graph is simple). And note that each 1 in the matrix accounts for edge between two verticies so that $\boldsymbol{a}_{i,j} = 1 \Leftrightarrow v_i v_j \in E$. Also note that $\boldsymbol{A}$ symmetrically overcounds each edges twice as the edge $v_i v_j$ and $v_j v_i$ a the same, therefore sine there are $n^2 - n$ 1's in the matrix, the maximum number of edges is $\frac{n(n-1)}{2} = \binom{n}{2}$. ■

Chapter 7: Probability

Definition 7.0.1: Conditional Independence

Suppose A and B are independent events, then we would want $P(A \mid C) = P(A \mid B \cap C)$:

$$P(A \mid C) = \frac{P(A \cap C)}{P(C)} \text{ and } P(A \mid B \cap C) = \frac{P(A \cap B \cap C)}{P(B \cap C)}$$

So then we want:

$$\begin{gathered} P(A \mid C) = P(A \mid B \cap C) \\ \Updownarrow \\ \frac{P(A \cap C)}{P(C)} = \frac{P(A \cap B \cap C)}{P(B \cap C)} \\ \Updownarrow \\ \frac{P(A \cap B \cap C)}{P(C)} = \frac{P(A \cap C)\,P(B \cap C)}{P(C)\,P(C)} \\ \Updownarrow \\ P(A \cap B \mid C) = P(A \mid C)\,P(B \mid C) \end{gathered}$$

7.1 Bayesian Interference

Chapter 8: Programming

Working in the Shell

Convert all files names to lowercase

```
user@machine:~# for f in ‘find‘; do mv -v "$f" "‘echo $f  tr ’[A-Z]’
    ’[a-z]’‘"; done
```