

覆盖性质的特征理论 一般拓扑学论题之一

蒋继光

序 言

本书介绍基本覆盖性质的特征理论, 重心是仿紧空间和次亚紧空间的刻画. 所谓基本覆盖性质还包括亚紧性, 次仿紧性, 正规覆盖, 可缩性和强仿紧性等. 某类覆盖性质的特征或刻画 (characterizations) 是指与该性质的原始定义等价的命题. 这些特征命题一般比原始定义更弱. 自从 A.H.Stone [1948] 建立 T_2 仿紧性的重合定理以来, 寻找仿紧空间及广义仿紧空间的各种各样的特征的努力一直延续至今. 发现不少美妙定理与精巧技术, 形成系统理论. 本书试图叙述这一领域的主要成果, 全部给予证明. 仿紧空间的刻画分别在节 6 与节 10 介绍, 节 6 介绍不附加分离公理的仿紧性的刻画. 节 10 则介绍包含 T_2 仿紧空间为子类的 λ 完满正规空间的刻画. 作为它的另一子类, 正规 λ 强仿紧空间, 在节 11 中介绍. 节 4 介绍次亚紧空间的刻画. 这三节是特征理论的重心. 节 7 介绍正规覆盖的刻画, 包含了点集拓扑学发展早期的一些好结果. 节 8 介绍集体正规空间与可缩空间的刻画. 这两节的内容是基本的, 不仅有其自身的意义, 也是节 10 中定理证明需要引用的. 节 9 与节 5 分别介绍次仿紧与亚仿紧空间的刻画. 每种覆盖性质有一类型刻画, 我们称之为分解定理, 即这种刻画表现为比它较弱的两种拓朴性质之和 (或两类较弱空间类的交). 其中一类是可扩型空间, 另一类我们称之为弱覆盖性质. 我们在节 2 与节 3 中分别介绍这些空间需要的知识. 其中也有值得学习和欣赏的好结果.

有关覆盖性质的特征理论的已有文献, 我愿推荐: [Bur84], [Jun80], [Gao2008] 和 [Yas89].

本书假定读书了解点集拓扑学的基础知识, 如 [Eng77] 的前 5 章, 或 [Gao2008] 的前 6 章. 集合论只需了解基数与序数的一般性质.

本书的参考文献只限于本书所引用者. 我在此向每一位作者谨致敬意和谢忱. 最后, 我对部分作者被引用的工作 (他们引入的概念和建立的定理的) 做了一个索引. 按照每一位作者被引用论文的最早发表年份为序排列他们的姓名. 这个索引有助于了解特征理论的发展历史.

蒋继光

于成都四川大学竹林村

乙未暮春

目 录

序言

第一章 预备	1
1.1 记号、术语与基本事实	1
1.2 空间与覆盖	1
第二章 Python 入门	2
2.1 关于 Python	2
2.2 安装 Python 开发环境软件集 Anaconda	2
2.3 初识 Spyder	4
2.4 初识 Jupyter	5
2.5 常用进位制	5
习题	5
第三章 Python 应用	8
3.1 Jieba	8
3.2 Some graphs	8
3.3 A List	10
名词索引	12

第一章 预备

覆盖性质有一类型的刻画是把它们分解成另两类较弱的拓扑性质的和. 其中一类是可扩型性质, 另一类是较弱的其它覆盖性质. 例如, 一个拓扑空间是仿紧的当且仅当它是可扩的和次亚紧的. 我们把这种类型的刻画叫分解定理. 本章第 2 节介绍用于各种分解方式来定理的弱覆盖性质. 第 3 节则介绍用作另一分解因子的可扩型性质. 它们的应用将在后面的章节中介绍.

1.1 记号、术语与基本事实

III

1.2 空间与覆盖

第二章 Python 入门

2.1 关于 Python

不犯错者一事无成.

Не ошибается тот, кто ничего не делает.

俄国谚语

Python 语言的最初设计者和主要架构师吉多·范罗苏姆 (Guido van Rossum), 荷兰人, 1982 年在阿姆斯特丹大学获得数学和计算机科学硕士学位。据说 1989 年的圣诞节期间, 为了打发时间而开发一个新的脚本解释器 (Script Interpreter), 以替代使用 Unix shell 和 C 语言进行系统管理。后来 Python 成了一种广泛使用的解释型、高级和通用的编程语言, 支持多种编程范型, 包括函数式、指令式、结构化、面向对象和反射式编程。它拥有动态类型系统和垃圾回收功能, 能够自动管理内存使用, 并且其本身拥有一个巨大而广泛的标准库。本节提到的名词术语, 如果初次接触的话, 可能有些不太好理解。这里作些解释, 但是就本书的使用来说, 完全可以忽略, 而不影响我们做要做的事情。

解释型 (interpreting) 意味着, 写好的、人可读代码直接一行一行地被执行。而在 C/C++、Java 等语言里, 编译器 (Compiler) 先把相关的代码文件编译 (compile) 成可执行的、机器可读而人不可读的二进制指令文件, 然后被执行或解释。

2.2 安装 Python 开发环境软件集 Anaconda

Python 在 MacOS 和 Unix/Linux 操作系统是默认安装好的, 但版本一般是 2.x, 且没有图形用户界面, 只适合在终端命令行上进行交互运行或脚本运行。微软视窗没有默认安装 Python。

本书采用 Python 3.x。虽然可访问 <https://www.python.org/downloads/> 安装 IDLE (Integrated Development and Learning Environment, 集成开发学习环境), 但是这个安装可省去。这个 IDLE 是终端命令行式的, 无图形用户界面。而本书将在有图形用户界面的 IDE (Integrated Development Environment, 集成开发环境) 里操作。

Python 有好几种免费的, 且功能卓越的 IDE 软件, 其中 Spyder 和 Jupyter 都包括在内 Anaconda。Spyder 提供一个优秀的开发平台和用户界面, 进行 Python 的代码编辑、交互运行和图形绘制。Jupyter 也提供这样的开发平台, 但它以互联网浏览器为用户界面, 背后

由一个本地网络服务器引擎在支持。其它很受欢迎并且为软件开发人员广泛使用的免费 IDE 有 Microsoft Visual Studio Code (<https://code.visualstudio.com>) 和 IntelliJ (Community 版本) (<https://www.jetbrains.com>)。这两种 IDE 支持除了 Python 以外的其它语言，如 Java, JavaScript, Node.js 和 C++ 等，但 Python 以及它的一些常用包和库需要另行安装和配置。Jetbrain 开发的 IntelliJ 和 Pycharm 都非常好。但 IntelliJ Ultimate 版和 PyCharm Professional 版都需要付费购买许可证，而免费的 IntelliJ Community 版和 PyCharm Community 版不支持本书要使用的一些 Python 库。本书主要使用 Spyder，为开拓眼见我们也使用 Jupyter。

若只安装 Spyder，可访问 <https://www.spyder-ide.org>。

若只安装 Jupyter，可访问 <https://www.jupyter.org>。

我们建议安装 Anaconda Navigator，以便可同时使用 Spyder 和 Jupyter 及其它。Anaconda 的网址是：

<https://www.anaconda.com>。

需要注意的是，Anaconda 个人版本（Individual Edition）是免费的，而其它版本则不是。个人版本从

<https://www.anaconda.com/products/individual>。

下载安装。安装好后，打开 Anaconda Navigator 所见如图 2.2.1。

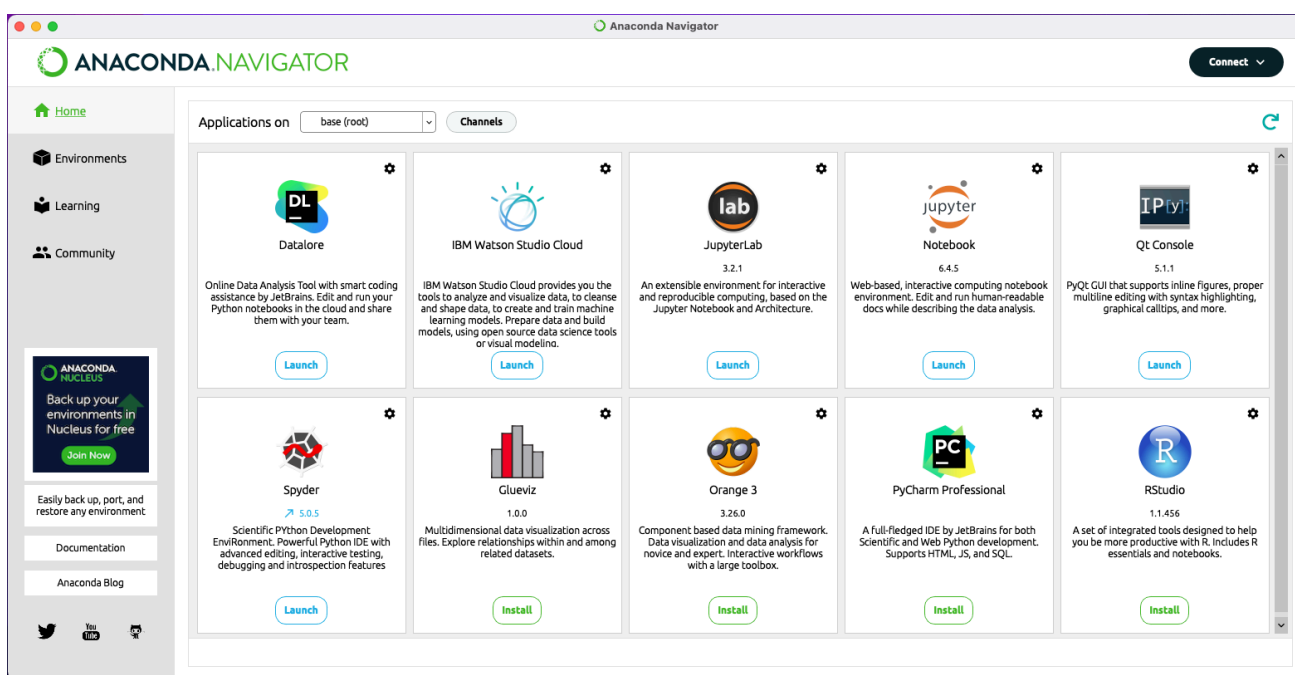


图 2.2.1: Anaconda Navigator

```
conda install -c conda-forge imagemagick
```

```
conda install -c conda-forge jieba
```

```
conda install -c conda-forge wordcloud
```

2.3 初识 Spyder

然后单击 Spyder 下面的 Launch 就会打开 Spyder。单击 Jupyter 下面的 Launch 就会打开 Spyder 打开 Spyder

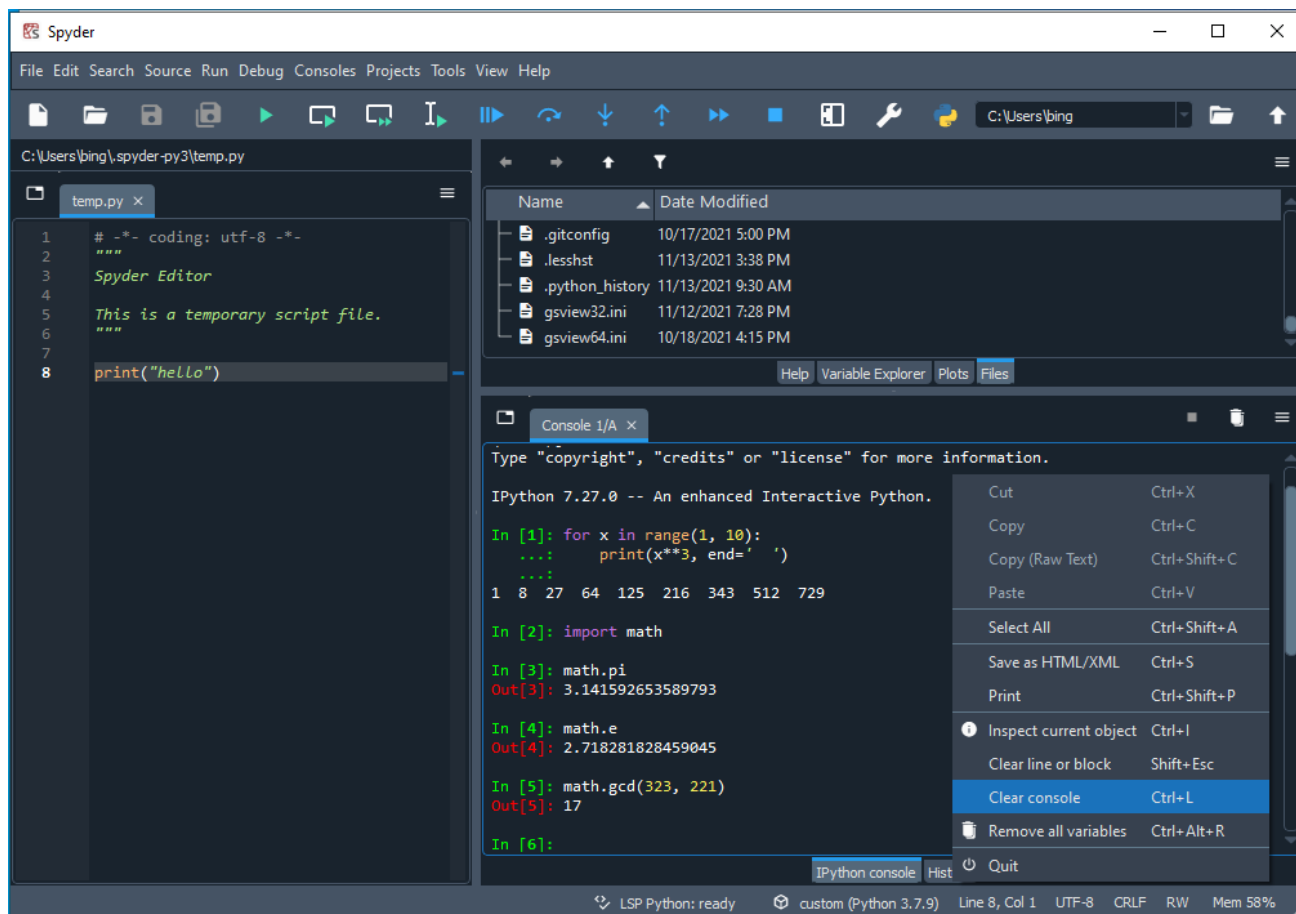


图 2.3.1: Spyder IDE

the Specifying Colors tutorial; the matplotlib.colors API; the Color Demo.

Text enclosed 'inside' `\texttt{verbatim}` environment
is printed directly
and all `\LaTeX{}` commands are ignored. 下面的

```

1 import re
2
3 x = "100 以内有 25 个质数。"
4 x += "不实际计算的话,51和91很容易被误判为是素数。"
5 y = re.sub("质数", "素数", x)
6 z = x.replace("素数", "质数")
7 print(y)
8 print(z)
9 print(x)
10
11 ###
12 import math

```

```

13
14 print("hello")
15 print(re.sub("This", "That", "This_is_not"))
16 print(math.pi)

```

2.4 初识 Jupyter

然后单击 Spyder 下面的 Launch 就会打开 Spyder。单击 Jupyter 下面的 Launch 就会打开 Spyder 打开 Spyder

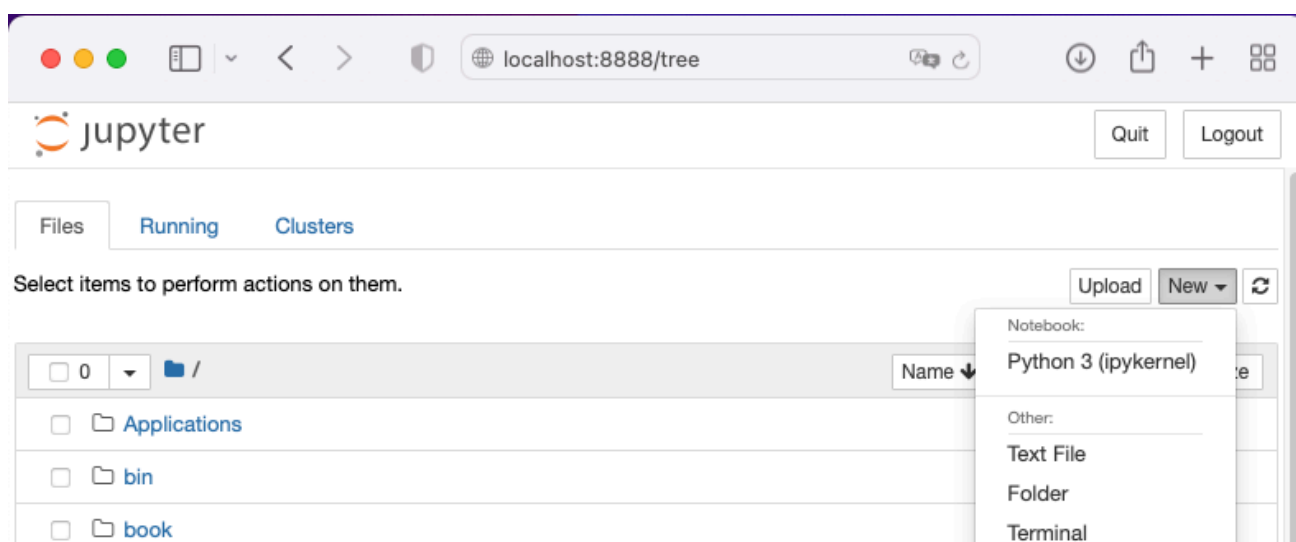


图 2.4.1: Jupyter 网页界面

the Specifying Colors tutorial; the matplotlib.colors API; the Color Demo.

2.5 常用进位制

处理字的符的 Python 程序

Python 程序 xxx 生成希腊的字母如下:

从 Unicode 表中 Python 程序 xxx 生成部分常用数学运算速度符号如果下:

字符串处理

习 题

1. 把下列值代入函数 `int()` 后的结果是什么?

- (a) 4.5; (b) 4.4; (c) -4.5; (d) -4.4;
 (e) `True`; (f) `False`; (g) `true`; (h) `false`;
 (i) `5 == 1+4`; (j) `6 < 5`; (k) `6 > -1.2`; (l) `'3.14159'`;

2. 把下列各对值代入函数 `int()` 后的结果是什么?


```
1  ##### 生成一组 Unicode 值在 m 和 n (包括) 之间的字符列表。
2  ## 输入: Unicode 起始值 m 和终止值 (包括) n。
3  ## 输出: 所定范围内相关应字符的列表。
4  ##### Unicode 的各种字符表可见 https://www.unicode.org/charts/
5  """"
6  ## 用遍历方法生成的, 不推荐。
7  def charactersFromTo(m, n):
8      result = []
9      for x in range(m, n+1):
10         result.append(chr(x))
11     return result
12 """"
13 def charactersFromTo(m, n):
14     return list(map(lambda x: chr(x), list(range(m, n+1))))
15
16
17 ##### 汉字表。
18 def chineseCharacters():
19     return charactersFromTo(0x4E00, 0x9FFF)
20
21
22 ##### 大写希腊字母表: 0x0391 ~ 0x03A9。
23 def upperCaseGreekAlphabet():
24     return charactersFromTo(0x0391, 0x03A9)
25
26
27 ##### 小写希腊字母表: 0x03B1 ~ 0x03C9。
28 def lowerCaseGreekAlphabet():
29     return charactersFromTo(0x03B1, 0x03C9)
30
31
32 ##### 大写俄语字母表: 0x0410 ~ 0x042F。
33 def upperCaseCyrillicAlphabet():
34     return charactersFromTo(0x0410, 0x042F)
35
36
37 ##### 小写俄语字母表: 0x0430 ~ 0x044F。
38 def lowerCaseCyrillicAlphabet():
39     return charactersFromTo(0x0430, 0x044F)
40
41
42 ##### 数学运费算符号表: 0x2200 ~ 0x22FF。
43 def mathOperators():
44     return charactersFromTo(0x2200, 0x22FF)
45
46
47 #print(chineseCharacters())
48 print(" ".join(upperCaseGreekAlphabet()))
49 print(" ".join(lowerCaseGreekAlphabet()))
50
51 print(upperCaseCyrillicAlphabet())
52 print(lowerCaseCyrillicAlphabet())
53 print(" ".join(mathOperators()))
54 print(mathOperators())
```


第三章 Python 应用

3.1 Jieba

```
gb2312
import urllib.request import urllib.parse
url = 'http://xh.5156edu.com' f = urllib.request.urlopen(url) txt = f.read().decode('gb2312')
print(txt)
```

GB 18030, 全称《信息技术中文编码字符集》，是中华人民共和国国家标准所规定的变长多字节字符集。

1、jieba 库基本介绍 (1)、jieba 库概述 jieba 是优秀的中文分词第三方库 - 中文文本需要通过分词获得单个的词语 - jieba 是优秀的中文分词第三方库，需要额外安装 - jieba 库提供三种分词模式，最简单只需掌握一个函数 (2)、jieba 分词的原理 Jieba 分词依靠中文词库 - 利用一个中文词库，确定汉字之间的关联概率 - 汉字间概率大的组成词组，形成分词结果 - 除了分词，用户还可以添加自定义的词组 jieba 库使用说明 (1)、jieba 分词的三种模式精确模式、全模式、搜索引擎模式

精确模式：把文本精确的切分开，不存在冗余单词 - 全模式：把文本中所有可能的词语都扫描出来，有冗余 - 搜索引擎模式：在精确模式基础上，对长词再次切分 (2)、jieba 库常用函数

```
cut cutforsearch lcut addword
```

3.2 Some graphs

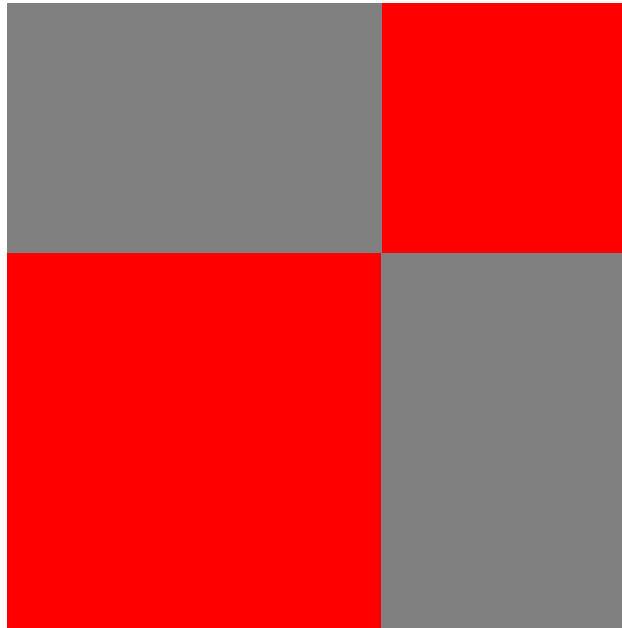
Postscript Language Reference, 3rd Edition^{PSLF3}

<https://www.adobe.com/content/dam/acom/en/devnet/actionsript/articles/PLRM.pdf>

3.2.1 一些数学公式的几何解释

$$(a + b)^2 = a^2 + 2ab + b^2$$

例 3.2.1. n 边形的内角和是 $180(n - 2)$ 度，所以正五边形的内角和是 $180 \cdot 3 = 540$ 度，而单个内角则是 $540/5 = 108$ 度。

图 3.2.1: $(a+b)^2 = a^2 + 2ab + b^2$

line spread

line spread dfdsa;fkadf

```

1  %!PS-Adobe-3.0
2  /side 80 def %%定义边长
3
4  /angle { %%定义五角星的一个尖角
5    side neg 0 rlineto
6    currentpoint translate
7    -36 rotate
8    side 0 rlineto } def
9
10 /star {
11   newpath
12   moveto
13   angle %%先定义一个尖角
14   4 { currentpoint translate
15     108 rotate angle
16   } repeat %%然后旋转坐标系后作出其它四个角色
17   closepath } def
18
19 gsave %%保存原始坐标系的信息
20 300 500 star %%描出五星的轮廓路径
21 1 0 0 setrgbcolor %%设置红色
22 fill %%填充绘出红五星，然后轮廓路径就消失
23 grestore
24
25 300 500 star %%重新描出五星的轮廓路径
26 0 setgray %%加黑边
27 stroke
28
29 showpage

```

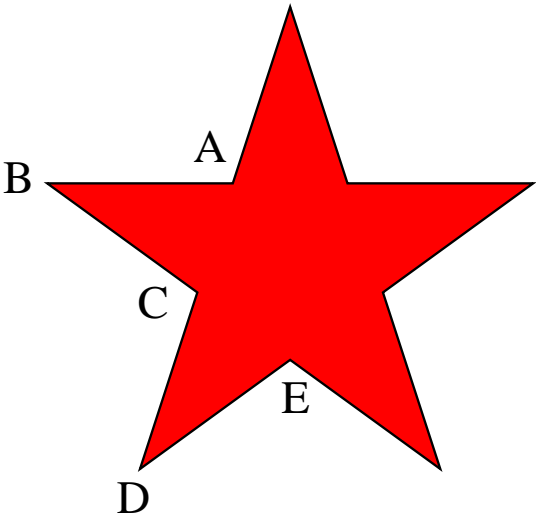


图 3.2.2: 红五星

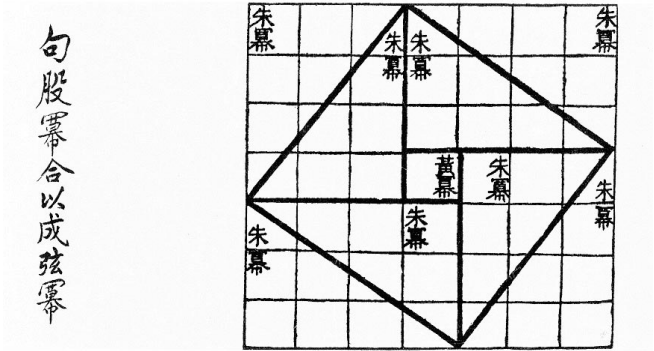


图 3.2.3: 《周髀算经》勾股定理图

图 3.2.4: $(a + b)^2 = a^2 + 2ab + b^2$

3.3 A List

华罗庚数论导引^{HuaL}
钟尔杰数学实验方法^{ZhongEr1}
钟尔杰数值分析讲义^{ZhongEr2}
Is God a Mathematician?^{LivioM}
最后的数学问题^{LivioM2}
菲赫金哥尔茨《微积分学教程》^{FeiH1FeiH2 FeiH3},
Atiyah^{AM69}
Naive Set Theory^{HalmosP}
KelleyJ, General Topology^{KelleyJ}

蒋继光, 一般拓扑学专题选讲 **JiangJi**
[derWaerden]

名词索引

Anaconda, 2

IDE, 集成开发环境, 2

IDLE, 集成开发学习环境, 2

IntelliJ, 3

Jupyter, 2

Spyder, 2

图形用户界面, Graphic User Interface, 2

终端, Terminal, 2

编译器, Compiler, 2

脚本, Script, 2

解释器, Interpreter, 2