

目录

1	定理证明	3
1.1	tactics	3
1.2	More tactics	4
1.2.1	generalize dependent	4
2	关于学习	7
2.1	影响学习效果的因素	7
2.2	能力	8
2.3	如何提高能力	8
2.4	性格	9
2.5	对性格的改造	9
3	TEX	11
3.1	Box	11
3.2	Glue	11
3.3	Macro	11
3.4	文档 & 帮助	12
3.5	实现	12
4	Lisp	13
5	ϵmacs	15
5.1	Abbrev mode	15
5.2	speed bar	15
5.3	bookmark	15

Chapter 1

定理证明

1.1 tactics

目前学过并且用了的 tactics 有：

- 归纳法
- 运算化简 (`simpl`)
- 运用已知的结论、中间结论、前提，进行改写 (`rewrite`). 这一步感觉是最难的，有几点需要弄清楚：
 - 有那些结论可用？
 - 改写哪一部分？
 - 改写的时候，要避免 free variable capture 的问题
- 分类讨论 (`destruct`)

还有一些方法是没有用到的，比如反证法，比如逻辑连接词的使用。比如怎么证明 $3 \neq 4$ ？

```
Theorem three_noteq_four: (3 =? 4) = false.
```

```
Proof.
```

```
  simpl. reflexivity.
```

```
□.
```

1.2 More tactics

新学习了一些 tactics, 包括

- `discriminate`
- `injection`
- `apply` 对有 \forall 限定的变量, 进行替换。有 `with` 变体
- `generalize dependent`

然后, 很多 tactic 都有 `in` 变体.

有一些 tactic 只是用于简化证明, 并没有引入什么新内容的, 比如 `symmetry`.

1.2.1 generalize dependent

需要先 `intros` 所有量词, 然后再按照希望的顺序, 重新泛化部分量词。通过一个例子来说明。

```

Theorem nth\_error\_after\_last:  $\forall$  (n : nat) (X : Type) (l : list X),
  length l = n  $\rightarrow$ 
  nth\_error l n = None.
Proof.
  intros n X l.
  generalize dependent n.
  (* Why? We need to induction on l, but doing so,
     will first intros n automatically, and so n in the premise is not so
     general *)
  induction l as [| h t].
  - (* For empty list *)
    simpl. reflexivity.
  - (* For non empty list, l = x :: t *)
    simpl.
    induction n as [| n'].
    + (* n can't be 0 *)
      simpl. intros eq1. discriminate eq1.
    + (* n = S n' *)
      simpl. intros eq2.
```

```
injection eq2. intros eq3. apply IHt. apply eq3.
```

□.

再来一个例子

```
Theorem implies_implies_and: ∀ P Q R : Prop,  
  ((P → Q) → R) → (P → (Q ∧ R)).
```

这个结论其实是错的。考虑,如果 $P=\text{True}, Q=\text{False}$,那么 $(P \rightarrow Q) = \text{False}$, 左边 $\text{False} \rightarrow R$ 恒为真。然而 $Q = \text{False} \rightarrow (Q \wedge R) = \text{False}$, 所以右边为假。因此结论不成立。

Chapter 2

关于学习

2.1 影响学习效果的因素

- 天赋

- 经验

这里的经验是有效经验。如果一个经验重复了 10000 次，那么也只是
一个有效经验。所以重复的、无意义的重复劳动，要尽量避免。哪怕
是失败的经验，只要不是冗余的，也是宝贵的。另外，多读一些经典
的书籍，也是快速积累有效经验的捷径。要经常自我反省，总结。

- 反馈

要接受有效的反馈，需要

- 保持真诚开放的心态。不要因为情绪的因素，而选择性忽视事实，
忽视反馈。
- 敏锐的观察，以及对效果的评估。能否将反馈量化/可视化？
- 考察重要的因子。有些影响反馈的因子，自己是很难察觉的，有
点类似思维的盲点。这时候最好是跟别人交流，别人能帮你拓宽
视野，指出一些关键的影响因子。

人的精力是有限的，要么是选择做很多事，但是每件事都做的半吊子；要么
选择做重要的事，宁可少做一点，也要争取做好每件事。我选择后一种态度。

2.2 能力

- 记忆

- 观察力

有观察力，意味着能发现不同的行动会导致的微妙的变化，从而在需要精细操作的时候，能够领悟到需要做到什么地步。比如画家需要有出色的色觉，音乐家需要有出色的音感。

- 动手能力

通过观察，知道需要怎么做，这还不足够，还要能确保落实到位。

- 精力

一个人如果经常容易犯困，或者学一会儿就厌倦疲劳，那么就不能长时间投入地去学习。这对于一些复杂的理论，就难以掌握。

- 触类旁通

举一反三，是不是一种能力？存疑。

通过比喻来说明的话，比如知道某分量的菜，需要放多少盐，这是观察力。

知道要放多少盐，但是在操作的时候，有没有手抖，对火候的控制能否做到随心所欲，这就是动手能力。

2.3 如何提高能力

天赋固然重要，但是天赋难以变改，除非以后科技发达，人可以比较自由的对自我进行元编程改造，那时候再来针对“天赋”进行完善。

排除天赋的因素，这里考察人力所能改变的能力，进行探讨。熟能生巧，多练习也可以改进各种能力。以下罗列我认为可能有用的方法。

- 记忆

适当背诵。临睡/早起的时候，进行冥想回忆。

- 观察力

- 动手能力

多练习，针对反馈进行调整。

- 精力

注意锻炼身体。怎么进行有效的休息? 睡觉前不要看太刺激的内容, 保持平静的心境。可以考虑喝点牛奶

- 睡觉前不要看刺激兴奋的内容, 保持平静的心境
- 考虑喝点牛奶
- 居住环境保持舒适干净
- ...

- 触类旁通

需要经常的将新知识与旧经验建立联系。时不时对知识进行梳理, 画下脑图, 做点笔记。

2.4 性格

性格、爱好对学习也有影响。如果某样技艺, 需要花长年累月的练习, 才能练好, 那么就需要练习者对此有强烈的兴趣和爱好, 否则是很难坚持得下来的。这里对兴趣爱好的讨论太宽泛了, 如果具体化, 比如, 喜欢小说电影, 喜欢游山玩水, 喜欢锻炼, 喜欢天马行空的想象, 喜欢创作, 这些喜好对人的影响是怎样的呢, 对学习的影响又是怎样的呢。另, 爱好太宽泛, 就不宜深入。另外还要能抵挡分心的诱惑。比如, 一个人能不能坐得住, 能不能耐住寂寞。一个人要能克制。

野心, 激情, 傲慢, 这些非理性的因素, 对学习的影响是如何的呢?

2.5 对性格的改造

Chapter 3

TEX

2024-10-21, 大致翻阅了一遍 The TEX Book。

这里，最基本的概念有：

- box
- glue
- mode 有 vertical mode, horiz mode, math mode
- macro

3.1 Box

3.2 Glue

3.3 Macro

simple macro:

```
\def\foo#1{#1, #1}  
\def\range#1->#2{#1 $\rightarrow$ #2}
```

`\range{n_1}->{n_∞}`, 效果为: $n_1 \rightarrow n_\infty$

3.4 文档 & 帮助

```
$ texdoc tex
$ texdoc texbytopic
$ texdoc source2e
$ texdoc source3
$ texdoc classes
$ tlmgr search beamer # 查找包
$ sudo tlmgr install <pkg-name>
```

3.5 实现

tex-fpc 是在 knuth 的版本上进行修改, 使之可以通过 `free pascal` 进行编译。下载 tex-fpc, 从 knuth 的 dist 复制 tex.web, mf.web 到 tex-fpc 目录下,

```
$ vi tex.ch # search initex part and shift left
$ tangle tex.web tex.ch # will generate program file: tex.p
$ fpc -FaUnix,sysutils,baseunix tex.p
```

编译 mf 与 tex 类似。然后得到 tex 和 mf 执行文件。但是:

```
$ ./tex
This is TeX-FPC, 4th ed. (no format preloaded)
**\input tex
I can't find TeXformats/plain.fmt!
```

生成 `plain.fmt` 要更麻烦一点, 后面再细看。

Chapter 4

Lisp

Chapter 5

εmacs

5.1 Abbrev mode

local mode

启动 abbrev mode: M-x abbrev-mode

M-x define-mode-abbrev RET abbrev RET exp

M-x define-global-abbrev RET abbrev RET exp

global mode

5.2 speed bar

打开 speed bar: M-x speedbar

5.3 bookmark

For registers, Save positions in register:

C-x r SPC <r>

Record the position and point in register <r> (point-to-register).

C-x r j <r>

Jump to position and buffer saved in register <r>(jump-to-register).

Save text in registers:

```
C-x r s <r>  
copy-to-register  
C-x r i <r>  
insert-register
```

For bookmarks:

```
C-x r m RET  
Set bookmark.  
C-x r b <bookmark> RET  
Jump to bookmark(bookmark-jump).  
C-x r l  
List all bookmarks(list-bookmarks).  
M-x bookmark-save  
Save bookmark values.
```