



南京大學

研究生畢業論文
(申請碩士專業學位)

論文題目 一種使用Hammer砸碎Nut的方法

作者姓名 張三

學科、專業 軟件工程

研究方向 某某研究主題

指導教師 李四 职称

20xx 年 x 月 xx 日

学 号 : MFxxxxxxx
论文答辩日期 : 20xx 年 x 月 xx 日
指 导 教 师 : (签字)



Using a Hammer to Crack a Nut

By

San Zhang

Supervised by

Advisor Title **Si Li**

A Thesis

Submitted to the Software Institute

and the Graduate School

of Nanjing University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Engineering

Software Institute

May 20xx

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目： 一种使用Hammer砸碎Nut的方法
软件工程 专业 20xx 级硕士生姓名： 张三
指导教师（姓名、职称）： 李四 职称

摘 要

这部分是中文摘要。

注意：本模板使用的是PDFLaTeX编译的，这一编译的好处在于速度快，并能直接引用pdf格式的图形。

以下展示列举（无编号）：

1. 贡献1。
2. 贡献2。
3. 贡献3。

关键词： 中文， 关键， 字

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Using a Hammer to Crack a Nut

SPECIALIZATION: Software Engineering

POSTGRADUATE: San Zhang

MENTOR: Advisor Title Si Li

Abstract

This is English abstract.

以下展示的是圆点列举（无编号）做些修改：

- First Contribution.
- Second Contribution.
- Third Contribution.

Keywords: English, Keywords

目 录

目录	v
第一章 标题	1
1.1 这是节标题	1
1.1.1 这是小节标题	1
第二章 正文	3
2.1 正文书写的小技巧	3
2.2 一些正文中的标记	3
2.3 注意软换行的使用	3
第三章 表格	5
3.1 表格与表格引用的基本概念	5
3.2 基本表格	5
3.3 表格单元跨列	5
3.4 表格单元跨行	6
3.5 表格与图形位置	6
第四章 图形	7
4.1 基本图形	7
4.2 引用代码	8
第五章 公式	9
5.1 公式5.1与论证	9
5.2 公式5.2与论证	9
5.3 公式5.3与论证	10
第六章 算法	11
第七章 论文引用	13

参考文献	15
简历与科研成果	17
致谢	19

表 格

3.1	实验系统中函数调用与数据依赖的交集.....	5
3.2	错误率与函数特征之间的关联	6
3.3	五个实验系统概述	6

插图

4.1	以含错误的RTM为输入的五五个系统上三个实验（Call，Data， Call+Data）的错误率（Incorrectness）	7
4.2	VoD系统中的代码片段	8
7.1	在dblp上下载Bibtex	13
7.2	Bibtex详细信息	14

第一章 标题

这是章节标题。注：一般而言，标题不要比小节标题更小，即不要出现1.2.3.4这种标题（本模板支持此类标题，即Subsubsection）。

1.1 这是节标题

1.1.1 这是小节标题

第二章 正文

2.1 正文书写的小技巧

CTeX自带的pdf浏览器，双击每段文字之后会自动回到WinEdt的编辑位置。

只有间隔一个明显的换行才会自然段分段。

因此，建议把一个自然段中的每句话都单独作为一行。这样的好处是，每次双击一句话，都可以回到WinEdt编辑器具体的一行。如果编辑时也按照自然段组织，则双击时会返回到一大段，不能定位到具体位置。不便于快速定位到出现问题的地方。

2.2 一些正文中的标记

斜体

加粗

代码元素格式

居中，左右对齐同理。

这里展示脚注。¹

一个小建议，中文后直接跟上述格式标记（包含各种引用）可能会出现一些问题。因此，在中文字和格式标记的斜杠之间加入一个波浪号是一个常用的习惯。双波浪线等价于一个强制空格，有时比键盘输入的空格要好用。

2.3 注意软换行的使用

论文一般会引用代码，本模板建议将代码声明为 `class.this()` 格式。在引用代码时，较长的函数名会导致函数名超出文本边界的情况，因此可以考虑手动进行软换行，请参考以下例子。

“图XX 展示了从AquaLush 系统中抽取的函数调用依赖示例，其中 `UIController.buildLogScrn()` 是为了实现新功能 “the control panel shows log message” 而在新版本中添加的函数。”

¹数字列举和圆点列举见摘要部分

第三章 表格

表格是LaTeX中少数没有Word好用的功能。但word的表格依然存在行间距的问题，而LaTeX也有简洁美观，相对易用（相对）的三线表。

3.1 表格与表格引用的基本概念

表格的编号和表目录都是自动生成并持续编号的，无需人工修改。只要对表格有标注（label），则在正文中引用该表的label，就可以随时保持最新编号。

注意：如果一个新表格加入，并被引用，编辑器将需要连续编译两次到三次，才能完成全部标题、引用和目录的更新。可以理解为第一次编译引入新表格，此时还不知表格引用位置的具体编号，需要留待第二次编译完成。而有可能第三次编译才将表格信息写入开头的表目录。类似的情况也会出现在图形和论文引用这两部分，其中尤以论文引用部分最为奇特，详见相关章节。

3.2 基本表格

表 3.1（这里是一个表引用!）是一个简单的三线表，双击表格可以在编辑界面内见到具体设置。

具体解释一下表格的设置：第一个table体内首先先声明标记位置以及字体大小；随后声明表格对齐方式；其次描述表标题；之后进入具体的表内容（tabular，此时还要声明表格单元中的内容如何对齐）；依次画出三线并填充内容；如果表格内容较多，可以相应的加入横线来划分（hline）；之后退出tabular；最后给表起名以实现全局引用，并退出表格。

表 3.1: 实验系统中函数调用与数据依赖的交集

	Call	Data	Overlap
VoD	222	899	66
GanttProject	5560	24243	1042
jHotDraw	3943	14555	893

3.3 表格单元跨列

表 3.2展示如何实现表格单元跨列。

表 3.2: 错误率与函数特征之间的关联

	Parameters		Return Value		Is Constructor	
	with	without	with	without	with	without
VoD	8.99%	9.20%	6.10%	9.51%	9.43%	8.46%
GanttProject	9.53%	6.05%	8.43%	6.71%	5.14%	8.09%
jHotDraw	4.40%	3.89%	4.36%	3.88%	2.91%	4.39%

3.4 表格单元跨行

表 3.3展示如何实现表格单元跨行（Average Number那一行）。此外，本表格的字体尺寸为scriptsize，比上一个表格的footnotesize要更小。

表 3.3: 五个实验系统概述

	VoD	Chess	GanttProject	jHotDraw	iTrust
Version	-	0.1.0	2.0.9	7.2	13.0
Programming Language	Java	Java	Java	Java	Java
KLOC	3.6	7.2	45	72	43
Executed methods	165	316	2741	1755	250
Evaluated requirements	12	7	17	21	34
Average Number of Methods	45	173	387	121	12
Implementing a Requirement	(9-148)	(23-288)	(78-815)	(1-555)	(1-33)
Size of the golden RTM	1980	2212	46597	36855	8500
Requirement traces	534	1210	6584	2547	353
Random Chance of guessing	0.5-7.5%	1-13%	0.2-1.7%	0.003-1.5%	0.01-0.4%
Method Call Dependencies	210	439	4830	3848	319
Method Data Dependencies	905	976	30452	17316	5329

3.5 表格与图形位置

常用选项[htbp]是浮动格式：

『h』当前位置。将图形放置在正文文本中给出该图形环境的地方。如果本页所剩的页面不够，这一参数将不起作用。

『t』顶部。将图形放置在页面的顶部。

『b』底部。将图形放置在页面的底部。

『p』浮动页。将图形放置在一只允许有浮动对象的页面上。

一般使用[htb]这样的组合，只用[h]是没有用的。这样组合的意思就是LaTeX会尽量满足排在前面的浮动格式，就是h-t-b这个顺序，让排版的效果尽量好。图形章节会有更多位置符号的例子。

第四章 图形

4.1 基本图形

相对于表格而言，LaTeX中的图形就简单多了，需要注意的是本模板推荐将所有图形都转化为pdf，具体内容参见图 4.1。该图形放在本模板的本地文件夹FIGs中。图 4.1是将Excel的五个子图形排布在一个ppt页面上，之后保存为pdf文件，最终得到的图形可以保证是矢量图。

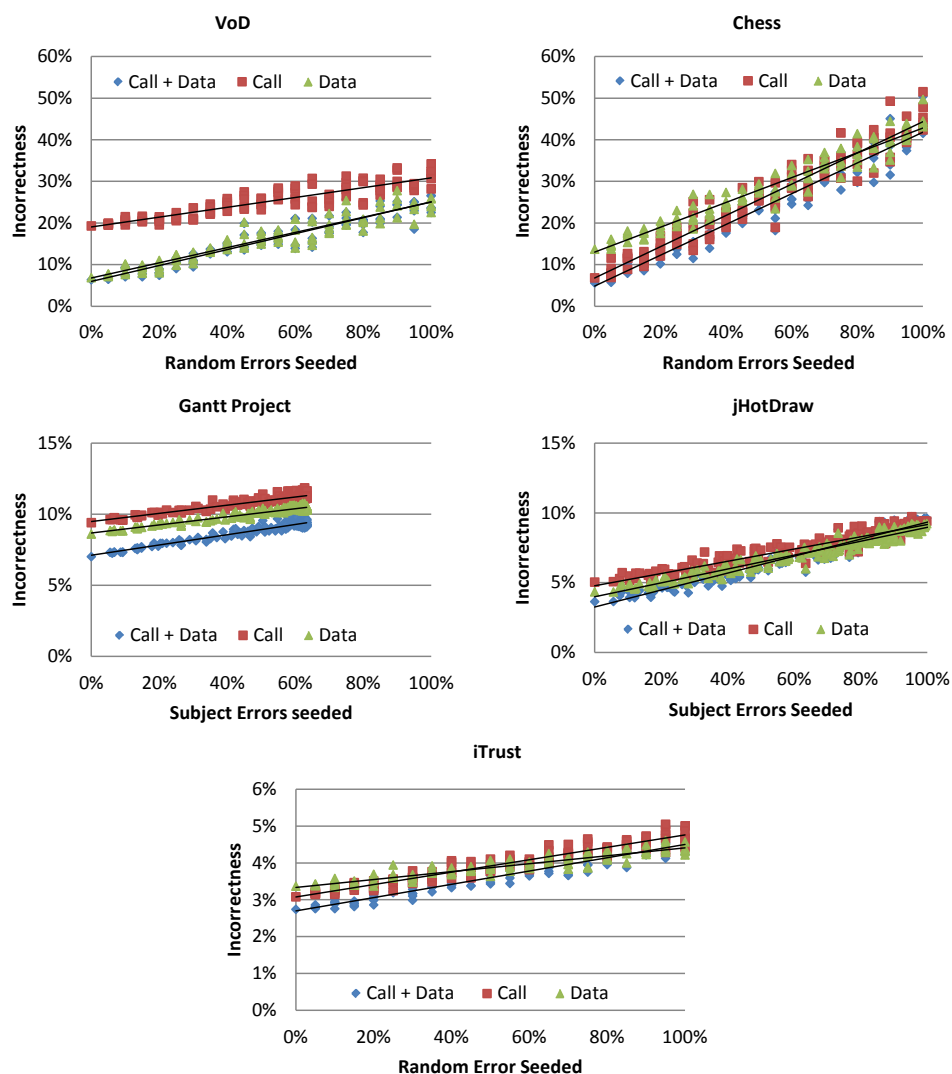


图 4.1: 以含错误的RTM为输入的五個系統上三個實驗（Call，Data，Call+Data）的錯誤率（Incorrectness）

注意：不要删除FIGs下面的njulogo和njuname这两个文件，这是论文封面的校徽和手写体南大校名。

4.2 引用代码

```
class VODClient{
    public final void init(){
        ...
        server = new ServerReq( "127.0.0.1", s);
        server.connect();
        listframe = new ListFrame(server, this);
    }
}

class ListFrame{
    public ListFrame(ServerReq serverReq,
        VODClient vODClient){
        ...
        ser = serverReq;
        parent = vODClient;
        ...}
    void buttonControl3_actionPerformed(...) {
        ...
        String s = listControll1.getSelectedItem();
        if (s != null){
            Movie movie = ser.getmovie(s);
        }
        ...}
}
```

图 4.2: VoD系统中的代码片段

这里给出一个代码引用的推荐实践。引用代码时先将代码放入word的文本框中，调整结束后，将该文本框页面另存为pdf文件，之后再作为图形来引用，如图 4.2所示。

第五章 公式

这里直接给出几个较为复杂的公式的例子，可一一进行参照。若有未包含的数学符号或公式格式，请参阅本模板所包含的手册（本地manual文件夹）或百度必应谷歌。

5.1 公式5.1与论证

“从直接代码依赖的角度出发，从一个初始域外的类 C_{out} 出发我们尝试找到一个通往初始域内的类 C_{in} 的路径。一条合法的路径需要满足以下两点要求：（1）这一路径是单向的，即 C_{out} 传递性地到达 C_{in} 或 C_{in} 传递性地到达 C_{out} ；（2）路径中只能包含一个 C_{in} （为了避免重复路径的出现）。为了恰当的估计一条合法路径所代表的交互程度，我们计算路径上所有直接代码依赖的紧密度值的几何平均。我们用如下公式来重新计算给定 C_{out} 的IR值（ IR_{DC} ）：”

$$IR_{DC} = IR_{origin} + (IR_{top} - IR_{origin})^{|PATH|} \sqrt{\prod_{x \in PATH} Closeness_{DC}(x)} \quad (5.1)$$

“其中 IR_{origin} 代表 C_{out} 的初始IR值， IR_{top} 代表 C_{in} 被提升过的IR值， $PATH$ 代表 C_{out} 与 C_{in} 之间的路径内所有的直接代码依赖，而 $Closeness_{DC}(x)$ 则代表每一条直接代码依赖关系的紧密度值。在同一对 C_{out} 和 C_{in} 之间可能存在多条合法路径，我们只保留其中能使 IR_{DC} 值最大的那条路径。”

5.2 公式5.2与论证

“由于IR方法返回的是一个按照IR值大小倒序排列的候选线索列表，因此一种常用的比较IR方法的方式是在不同的查全率水平上比较不同方法之间的精确度，通常用*Precision – Recall*曲线表示。为了进一步衡量IR方法返回结果的整体质量，我们选用了另外两个常用的实验度量： AP （Average Precision）与 MAP （Mean Average Precision）。其中， AP 用于度量全部查询（需求）所检索的相关文档的排序质量，计算方式如下：”

$$AP = \frac{\sum_{r=1}^N (Precision(r) \times isRelevant(r))}{|RelevantDocuments|} \quad (5.2)$$

“其中， r 表示被查询对象（类）在列表中的排序， $Precision(r)$ 表示前 r 个类的准确率。 $isRelevant()$ 为一个二值函数，如果文档是相关的，则返回1，若无关，则返回0。”

5.3 公式5.3与论证

“由此，我们为类数据依赖定义紧密度 $Closeness_{CD}$ 如下：”

$$Closeness_{CD} = \frac{\sum_{x \in \{DT_i \cap DT_j\}} idtf(x)}{\sum_{y \in \{DT_i \cup DT_j\}} idtf(y)} \quad (5.3)$$

“其中 $idtf(x)$ 代表共享数据类型的idf值， DT_i 与 DT_j 的交集代表该数据依赖上的共享数据类型，而 DT_i 与 DT_j 的并集则代表 C_i 和 C_j 在全部代码上所访问的数据类型。 $Closeness_{CD}$ 的取值范围是0到1之间。”

第六章 算法

同样是定义+引用的方式，参见算法 1。本算法已包含大量常用格式，如有未包含的数学符号或格式，请参阅本模板所包含的手册或询问百度必应谷歌。如论文中无需算法则不用强加。

Algorithm 1: 初始需求域外追踪线索的重排

```

1 topIRValue  $\leftarrow$  initialRegion.topIRValue;
2 foreach link in candidateList do
3   if !initialRegion.contains(link.class) then
4     foreach c in initialRegion do
5       pathList  $\leftarrow$  findValidPaths(link.class, c);
6       gMean  $\leftarrow$  0;
7       foreach path in pathList do
8          $gMean \leftarrow \max(\text{GeometricMean}(\text{Closeness}_{DC}(\text{path})), gMean)$ ;
9       link.IRValue  $\leftarrow$  link.IRValue + gMean(topIRValue - link.IRValue);
10      if hasDataDependencies(c, link.class) then
11        link.IRValue  $\leftarrow$  link.IRValue + ClosenessCD(c,
12          link.class)(topIRValue - link.IRValue);
13      if link.IRValue > topIRValue then
14        link.IRValue  $\leftarrow$  topIRValue;
15 candidateList.reorderByIRValue();

```

第七章 论文引用

此处的论文引用采用的是IEEE格式。建议将被引用的论文全名放入dblp网站（必应谷歌搜索dblp）搜索，之后进入该论文详细信息，如图 7.1所示。

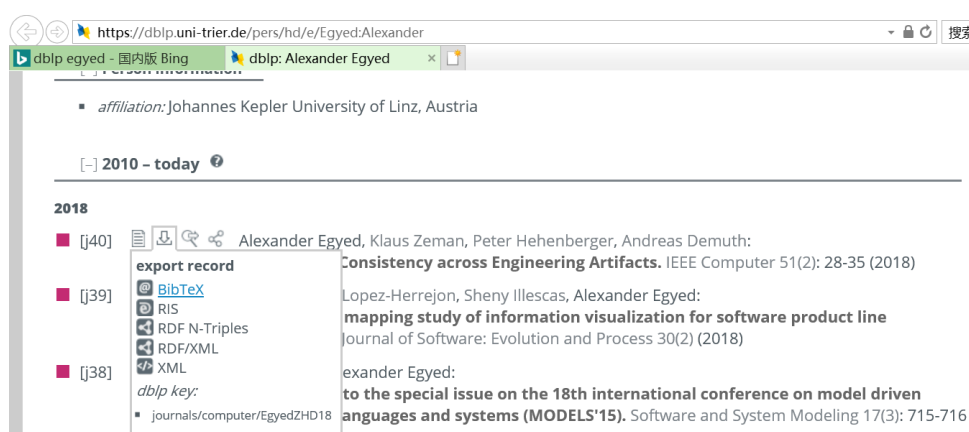


图 7.1: 在dblp上下载Bibtex

点击该链接之后将得到Bibtex信息，如图 7.2所示。打开本地文件夹下的reference.bib文件，完整添加该信息。并在需要引用的位置添加这一引用 [1]。格式为bibtex信息中的开头，例如图中的“DBLP:journals/computer/EgyedZHD18”。

注意：在修改并保存reference.bib文件后，先点击PDFLaTeX旁边的B按钮编译bib文件，之后需要连续使用PDFLaTeX编译三次，直到最后控制台输出的Warnings不再增加，此时才完成一次论文引用的更新。


在bib文件中出现，但并未在论文中被引用的论文不会出现在最后的参考文献中。如果dblp中并未包含你需要的论文，则可以尝试谷歌或百度学术的搜索结果，一般也包含bibtex信息，但可能不完整或不规范。

引用网站链接可以考虑这一格式 [2]。

← → <https://dblp.uni-trier.de/rec/bibtex/journals/computer/EgyedZHD18>

dblp egyed - 国内版 Bing dblp: BibTeX record journa... x

maintained by SCHLOSS DAGSTUHL



dblp
computer science bibliography

BibTeX record journals/computer/EgyedZHD18

> Home > journals/computer/EgyedZHD18

download as .bib file

```
@article{DBLP:journals/computer/EgyedZHD18,  
  author    = {Alexander Egyed and  
              Klaus Zeman and  
              Peter Hehenberger and  
              Andreas Demuth},  
  title     = {Maintaining Consistency across Engineering Artifacts},  
  journal   = {{IEEE} Computer},  
  volume    = {51},  
  number    = {2},  
  pages     = {28--35},  
  year      = {2018},  
  url       = {https://doi.org/10.1109/MC.2018.1451666},  
  doi       = {10.1109/MC.2018.1451666},  
  timestamp = {Tue, 06 Mar 2018 17:05:11 +0100},  
  biburl    = {https://dblp.org/rec/bib/journals/computer/EgyedZHD18},  
  bibsource = {dblp computer science bibliography, https://dblp.org}  
}
```

图 7.2: Bibtex详细信息

参考文献

- [1] Alexander Egyed, Klaus Zeman, Peter Hehenberger, and Andreas Demuth. Maintaining consistency across engineering artifacts. *IEEE Computer*, 51(2):28–35, 2018.
- [2] GanttProject System. <http://www.ganttproject.biz>.

简历与科研成果

基本情况 张三，男，汉族，1993 年 1 月出生，江苏省南京市人。

教育背景

2015.9～2018.6 南京大学软件学院 硕士

2001.9～2015.7 南京大学软件学院 本科

这里是读研期间的成果（实例为受理的专利）

1. 李四，张三，“一种使用Hammer砸碎Nut的方法”，申请号：20xx1018xywz.a，已受理。

致 谢

这里是致谢。一般的感谢顺序：导师，其他指导老师，师兄姐妹、同学，父母和伴侣。