电子科技大学

University of Electronic Science and Technology of China

学 士 学 位 论 文

BACHELOR DISSERTATION

论文题目进化树结构对比算法的研究

学生姓名 彭云

学　　号 2011060050019

专　　业 计算机科学与技术

学　　院 计算机科学与工程学院

指导教师 肖鸣宇

指导单位 电子科技大学

2015年3月13日

摘 要

J. H. Wilkinson[?]建立了非奇异矩阵的逆是矩阵元素的连续函数的理论。G. W. Stewart[?]推出了矩阵 的广义逆 的连续性。为了得到Drazin逆的连续性，本文先给出了$M-$矩阵、$H-$矩阵类的逆的连续性。Campbell和Meyer[?] 也给出了Drazin 逆的连续性性质，但没有给出明显的边界。

Drazin逆对扰动是很不稳定的。然而，在某种特定的扰动条件下，矩阵$(A+E)^D$与A^D的接近程度能够得到量化且也能得到明显的相对误差边界。基于Drazin逆的不同形式，很多科学家和学者从事这一方面的研究。U. G. Rothblum 给出的Drazin逆的以下的表达式：

$$A^D = (A-H)^{-1}(I-H) = (I-H)(A-H)^{-1}$$

其中$H = I - AA^D = I - A^DA$.基于这个表达式,我们在本文中也给出了$\|(A+E)^D - A^D\|_2/\|A^D\|$和$\|(A+E)^\sharp - A^D\|_2/\|A^D\|_2$的范数估计，并与前人的成果进行了比较。

关键词：$M-$矩阵，$H-$矩阵，Drazin逆，Pseudo-Drazin逆，条件数

ABSTRACT

The theory that the inverse of a nonsingular matrix is continuous function of the elements of the matrix was established by J. H. Wilkinson[?]. The continuity of the generalized inverse A^+ of a matrix A was introduced by G. W. Stewart[?]. In this paper, at first, the continuity of the special matrices inverse, such that $M-$matrices and $H-$matrices, respectively, are provided. Campbell and Meyer[?] also established the continuity properties of Drazin inverse, but the explicit bound was not given.

The Drazin inverse is unstable with respect to perturbation. However, under some specific perturbation , the closeness of the matrices $(A+E)^D$ and A^D can be proved and the explicit bound the relation error can also be obtained. Based on the different representations of Drazin inverse, many scientists and scholars have worked it research. U. G. Rothblum gave the following representation of Drazin inverse:

$$A^D = (A-H)^{-1}(I-H) = (I-H)(A-H)^{-1}$$

where $H = I - AA^D = I - A^DA$. Based on the representation, we also obtain the norm estimate of $\|(A+E)^D - A^D\|_2/\|A^D\|$ and $\|(A+E)^\sharp - A^D\|_2/\|A^D\|_2$ and compare with the precedent results.

Keywords: $M-$matrices, $H-$matrices, Drazin inverse, Pseudo-Drazin inverse, Condition number

目 录

第1章 引言 1
1.1 研究背景 1
1.2 研究意义 1
1.3 近似算法 1
1.4 参数算法 1
第2章 问题建模和描述 2
2.1 基本定义 2
2.2 问题描述 4
2.3 相关研究 5
2.3.1 近似算法 5
2.3.2 参数算法 5
2.3.3 启发式算法 6
2.3.4 Whidden的参数算法[10]概述 6
第3章 算法思想与设计 8
3.1 改进思路 8
3.2 详细步骤 8
参考文献 9
致 谢 11
附录 A 附录章 12
A.1 附录节 12
附录 B 附录另一章 13
B.1 附录另一章的一节 13
外文资料原文 14
外文资料译文 16

第1章 引言

1.1 研究背景

分子系统学在生物学中是一个非常重要的研究领域，其中进化树，又称系统发生树，是用来研究一群物种进化历史的标准模型，是分子系统学中不可缺少的重要工具。由于生物进化中可能产生的杂交、基因水平转移、基因重组等网状事件，导致一个物种的基因可能来源于多个祖先。多年来对生物进化历史的研究发现，杂交事件分别在植物、鸟类、鱼类的种群中均有发生。自然的杂交事件在哺乳动物，甚至是灵长类动物中都被发现过。研究显示，25%的植物和10%的动物，尤其是年轻的物种，都与杂交事件相关。已有许多方法能够从一组物种中构建起他们对应的进化树，但由于进化过程中网络事件的存在，一组相同物种基于不同基因的分析可能产生不同的进化树，反之，通过对比这些进化树的相似性是帮助人们发现这些网络事件的重要手段。许多计算生物学家都对此问题非常感兴趣，他们常用的衡量标准有子树剪切再接距离（subtree prune and regraft distance, rSPR）和杂交数（hybridization number, HN）两种。Baroni 等人证明了rSPR距离为进化过程中网状事件的数目给出了一个下限。

1.2 研究意义

1.3 近似算法

1.4 参数算法

第2章 问题建模和描述

2.1 基本定义

本节将给出在研究进化树结构的过程中所需要的一些重要概念的定义。

定义 2.1.1 **有根二叉进化树**（简称进化树）是一棵叶节点被集合X中的元素所标记的满二叉树，即除叶节点没有子节点外，其余节点都有且仅有两个子节点，记作T。将T的所有边的集合记为E_T。将X称作该进化树的**标识集**。

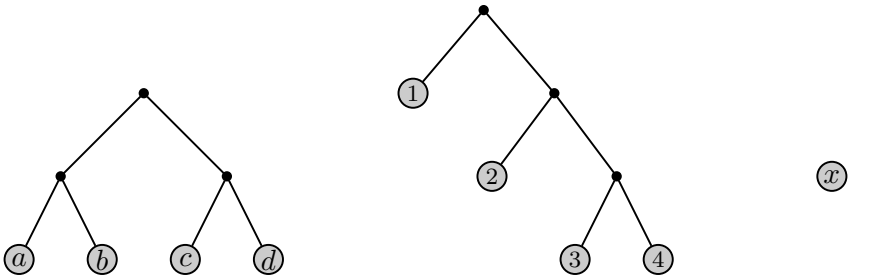

图 2–1 有根二叉进化树示例

定义 2.1.2 如果两棵进化树具有相同的标识集，并且**同构**，则认为这两棵进化树相等，记作$T_1 = T_2$。

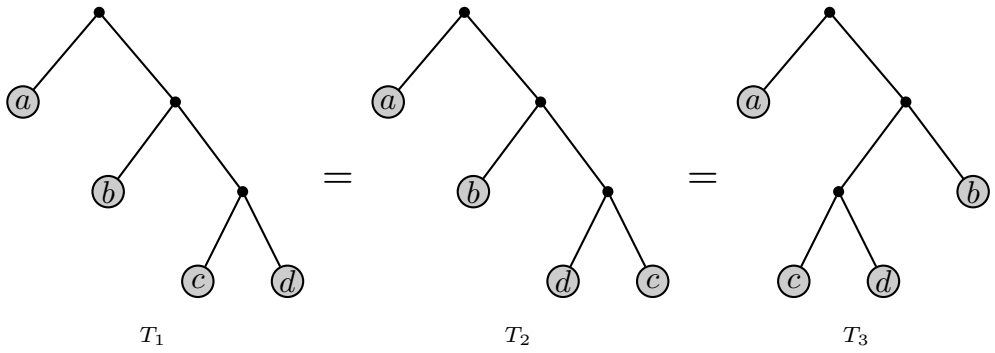

图 2–2 相等的3棵进化树

定义 2.1.3 将一棵二叉树删去所有只有一个子节点的内部节点以及所有未被标记的叶节点，使其变成一棵进化树的操作称为**收缩**。

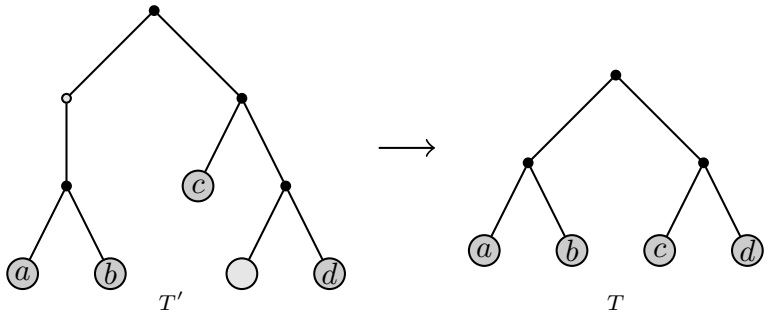

图 2–3 将T'收缩得到T

定义 2.1.4 **子树剪切再接**（rooted subtree prune and regraft，简称rSPR）对于一棵进化树T，剪去任意节点x的父边e_x得到一棵以x为根的子树t_x，将t_x嫁接到余下子树$T-t_x$的一条边上，并对操作后的树进行一次**收缩**，获得一棵新的进化树，该过程称为一次**rSPR操作**。对于两棵进化树T_1,T_2，将其中一棵树转化为另一棵所需的最少rSPR操作次数称为T_1,T_2的**rSPR距离**，记作 $d(T_1,T_2)$。

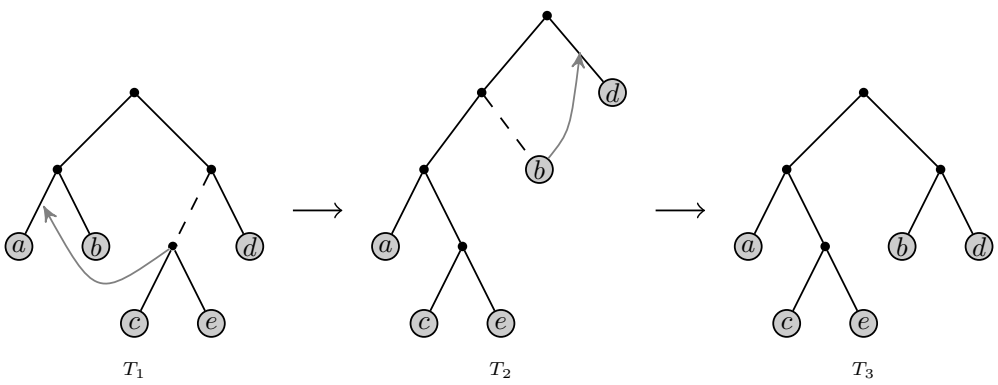

图 2–4 将T_1转化为T_3需要至少2次SPR操作，$d(T_1,T_3)=2$

定义 2.1.5 **进化树森林**（简称森林）是由若干棵进化树组成的集合。记作$F=\{t_1,t_2,...,t_n\}$。将一棵进化树T删去若干条边E，并对每棵独立的子树进行一次收缩操作得到的森林F，记作$F=T-E$。同理，将一个森林F删去若干条边E，并对每棵独立的子树进行一次收缩操作得到的森林F'，记作$F'=F-E$。

定义 2.1.6　最大一致森林（maximum-agreement forest，简称MAF）对于两棵进化树T_1, T_2，若存在$E_1 \subset E_{T_1}$，$E_2 \subset E_{T_2}$，使得$F = T_1 - E_1$并且$F = T_2 - E_2$，我们称F为T_1, T_2的**一致森林**。将T_1, T_2的所有一致森林中，包含最少进化树个数的森林称为**最大一致森林**，其包含的进化树个数记为$m(T_1, T_2)$。

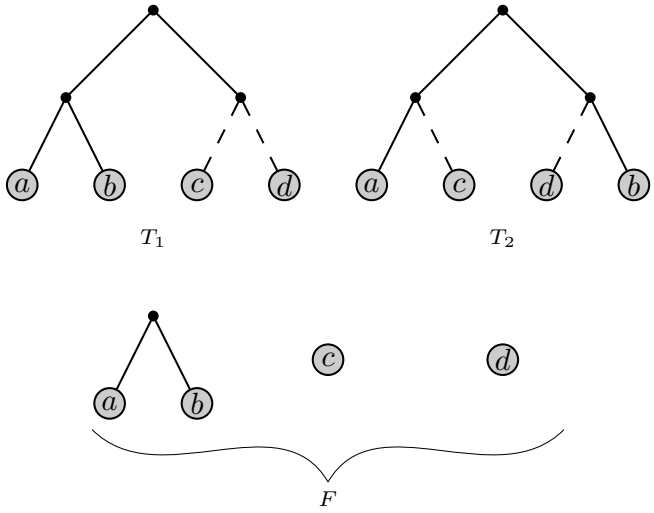

图 2–5 将T_1和T_2删去对应边后获得最大一致森林F，$m(T_1, T_2) = 3$

Bordewich和Semple[1]已经指出两棵进化树的rSPR距离与MAF的大小存在着等价关系，他们证明了如下定理：

定理 2.1.1　对于两棵具有相同标识集X的进化树T_1, T_2，存在

$$d(T_1, T_2) = m(T_1, T_2) - 1$$

2.2 问题描述

根据定理2.1.1我们可以将求解两棵进化树的rSPR距离转化为求解它们的最大一致森林的最优化问题，$d(T_1, T_2)$也可等价为将T_1, T_2转化为它们的MAF所需要删除的最少边数。而因为该问题是NP完全问题，我们可以将其进一步转化为可以用固定参数算法求解的判定性问题，问题的具体描述如下：

给定两个具有相同标识集X的进化树T_1, T_2，以及一个参数k。
判断 $d(T_1, T_2) \leq k$ 是否成立？

2.3 相关研究

尽管rSPR距离在生物学上具有十分重要的意义，但计算它却被证明是NP难的。因此研究针对该问题的近似算法，参数算法以及一些启发式算法成为了计算rSPR距离的主要手段。在介绍本文提出的参数算法之前，先回顾之前针对此问题各种算法的研究进展。

2.3.1 近似算法

自从Jotun Hein等人在[2]中引入MAF的概念以来，MAF已经成为解决该问题最有效的工具，不少以此为基础的近似算法得以提出。Hein首先在[2]中提出了一个近似比为3的近似算法，但却被Rodrigues在[3]中证明在某些情况下其近似比不可能小于4。同时，Rodrigues也提出了一些修正，宣称修正后的算法的近似比为3。然而，Bonet在[4]中给出的反例证明了[2]和[3]中的算法近似比实际上均为5，并且可以在线性时间复杂度下实现。Bordewich在[5]中提出了一个正确的近似比为3的算法，然而是以时间复杂度增加到$O(n^5)$为代价①。在[3]中提出的第二个近似比为3的算法的时间复杂度为$O(n^2)$。最终，线性时间复杂度下近似比为3的算法由Whidden在[6]中提出。

2.3.2 参数算法

在生物进化过程中网状事件发生的次数远远小于物种的数量，因此固定参数算法往往是计算rSPR距离精确值的最好途径。将两棵进化树的rSPR距离作为参数k的参数算法便是一个很有前景的研究方法。以此为基础，Bordewich在[5]中提出了一个时间复杂度为$O(4^k \cdot k^5 + n^3)$参数算法。之后，Hallett，Allen，Bonet等人针对该问题的相关问题也有许多有意义的研究。[7][8][9]Whidden在[10]中提出的算法可以说是该问题在参数算法上的一个突破，该算法的时间复杂度为$O(2.42^k n)$，并且从实验结果可以看出其性能远优于先前的算法，有能力处理具有更多节点数更大rSPR距离的进化树对比问题。随后，Chen和Wang在[11]中对Whidden的算法中最差的情况进行了许多改进，最后获得了一个时间复杂度为$O(2.34^k n)$的参数算法，但其改进方法分类较多，比较复杂，因此算法实现难度很大。本文也是基于相同的思想，但使用相对更简单的方法得到了更好的改进效果。

① 适当地改进数据结构可以将时间复杂度降低到$O(n^4)$

2.3.3 启发式算法

许多关于SPR距离的启发式算法在近年也得以提出，并开发了相关的软件。其中，Hallett和Lagergen开发的程序LatTrans[12]针对一些特殊的rSPR操作进行建模，只考虑进化树只可能在两种情况下相异，在这种特殊的条件下，它的时间复杂度可以减少到$O(2^k n^2)$。Macleod开发的HorizStory[13]可以计算多叉树的SPR距离，但只考虑SPR操作对象是只有一个叶节点的子树。SPRdist[14]和TreeSAT[15]是两个计算rSPR距离精确值的软件，他们分别把计算MAF的问题转化为整数线性规划问题（integer linear programming, ILP）和可满足性问题（satisfiability problem, SAT），然后利用求解对应问题的有效手段来获得rSPR距离的解。但根据实验结果，它们的性能均不能与Whidden所提出的算法相比。[10]

2.3.4 Whidden的参数算法[10]概述

Whidden的算法所解决的正是本文在第2.2节中所提出的问题。设$MAF(T_1, F_2, k)$为针对此问题的判定函数，T_1代表第一棵进化树，F_2代表T_2删除某个边集后所得的森林，k是一个非负整数。若F_2能够在删除至多k条边后变成T_1, T_2的最大一致森林，则$MAF(T_1, F_2, k)$返回$true$，否则返回$false$。初始时，$F_2 = T_2$。求解rSPR距离，只需要从0开始枚举k的值，直到$MAF(T_1, T_2, k)$返回$true$。因为时间复杂度是关于k的指数，所以相对于计算$MAF(T_1, T_2, k)$，计算rSPR距离只会在时间复杂度的常数上有所增加。算法采用递归的思想，MAF函数的具体步骤如下：

1. 如果$k < 0$，返回$false$
2. 如果在T_1中存在一对兄弟叶节点a, b，并且它们在F_2中的对应节点也是兄弟叶节点，那么就合并a, b，然后把它们的在T_1, F_2中的父节点作为对应的叶节点。重复此步骤，直至没有节点可以合并。
3. 如果在F_2中存在只有一个节点的子树x，则将x从T_1, F_2中移除。重复此步骤，直至没有节点可以移除。此时如果产生新的可以合并的兄弟叶节点，则转至2，否则继续。
4. 如果F_2为空，返回$true$
5. 任意选择T_1中的一对兄弟叶节点a, b（注意到此时a, b在F_2中一定不是兄弟叶节点），分三种情况讨论（如图2–6）：

(a) 若a,b在F_2中属于不同的连通分量。可以证明[16]，a的父边e_a和b的父边e_b两条边中至少有一条需要删除。对于两种情况下修改后的$F_2' = F_2 - \{e_a\}$和$F_2' = F_2 - \{e_b\}$，分别调用两次$MAF(T_1, F_2', k-1)$。只要有任意一次结果为$true$，则返回$true$，否则返回$false$。

(b) 若a,b在F_2中连通，并且a,b之间有且只有一个悬挂节点p。可以证明[16]，一定存在最优解$E \subset E_{T_2}$，使得$T_2 - E$是T_1, T_2的MAF，并且$e_p \in E$。因此，只需要直接修改F_2，得到$F_2' = F_2 - \{e_p\}$。若$MAF(T_1, F_2', k-1)$结果为$true$，则返回$true$，否则返回$false$。

(c) 若a,b在F_2中连通，并且a,b之间有至少两个悬挂节点。设a,b之间所有悬挂节点的集合为$P = \{p_1, p_2, ..., p_n\}$。可以证明[16]，一定存在最优解$E \subset E_{T_2}$，使得$T_2 - E$是$T_1, T_2$的MAF，并且$e_a \in E$或者$e_b \in E$或者对于所有$1 \le i \le n$，有$e_{p_i} \in E$。因此，分别修改$F_2$得到$F_2' = F_2 - \{e_a\}$、$F_2' = F_2 - \{e_b\}$和$F_2' = F_2 - E_P$（$E_P$为P中所有节点对应的父边的集合），然后分别调用两次$MAF(T_1, F_2', k-1)$和一次$MAF(T_1, F_2', k-n)$。只要有任意一次结果为$true$，则返回$true$，否则返回$false$。

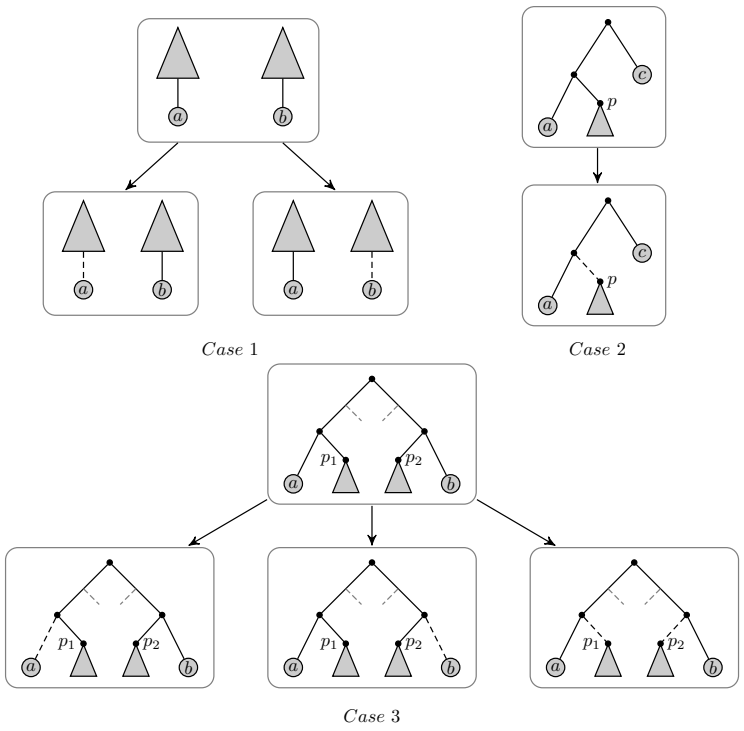

图 2–6 第5步的三种情况

第3章 算法思想与设计

3.1 改进思路

3.2 详细步骤

参考文献

[1] M. Bordewich, C. Semple. On the computational complexity of the rooted subtree prune and regraft distance[J]. Annals of combinatorics, 2005, 8(4):409–423.

[2] J. Hein, T. Jiang, L. Wang, et al. On the complexity of comparing evolutionary trees[J]. Discrete Applied Mathematics, 1996, 71(1):153–169.

[3] E. M. Rodrigues, M.-F. Sagot, Y. Wakabayashi. The maximum agreement forest problem: Approximation algorithms and computational experiments[J]. Theoretical Computer Science, 2007, 374(1):91–110.

[4] M. L. Bonet, K. S. John, R. Mahindru, et al. Approximating subtree distances between phylogenies[J]. Journal of Computational Biology, 2006, 13(8):1419–1434.

[5] M. Bordewich, C. McCartin, C. Semple. A 3-approximation algorithm for the subtree distance between phylogenies[J]. Journal of Discrete Algorithms, 2008, 6(3):458–471.

[6] C. Whidden, N. Zeh. A unifying view on approximation and FPT of agreement forests[M].[S.l.]: Springer, 2009.

[7] M. Hallett, C. Mccartin. A faster FPT algorithm for the maximum agreement forest problem[J]. Theory of Computing Systems, 2007, 41(3):539–550.

[8] B. L. Allen, M. Steel. Subtree transfer operations and their induced metrics on evolutionary trees[J]. Annals of combinatorics, 2001, 5(1):1–15.

[9] M. L. Bonet, K. St John. On the complexity of uSPR distance[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), 2010, 7(3):572–576.

[10] C. Whidden, R. G. Beiko, N. Zeh. Fast FPT algorithms for computing rooted agreement forests: Theory and experiments[M]//Experimental Algorithms.[S.l.]: Springer, 2010:141–153.

[11] Z.-Z. Chen, L. Wang. Faster exact algorithms for hybridization number and rSPR distance[J]. Submitted for, 2012.

[12] M. T. Hallett, J. Lagergren. Efficient algorithms for lateral gene transfer problems[C]//Proceedings of the fifth annual international conference on Computational biology. .[S.l.]: [s.n.] , 2001:149–156.

[13] D. MacLeod, R. L. Charlebois, F. Doolittle, et al. Deduction of probable events of lateral gene transfer through comparison of phylogenetic trees by recursive consolidation and rearrangement[J]. BMC evolutionary biology, 2005, 5(1):27.

[14] M. L. Bonet, K. S. John. Efficiently calculating evolutionary tree measures using SAT[M]//Theory and Applications of Satisfiability Testing-SAT 2009.[S.l.]: Springer, 2009:4–17.

[15] Y. Wu. A practical method for exact computation of subtree prune and regraft distance[J]. Bioinformatics, 2009, 25(2):190–196.

[16] C. Whidden, R. G. Beiko, N. Zeh. Fixed-parameter algorithms for maximum agreement forests[J]. SIAM Journal on Computing, 2013, 42(4):1431–1466.

致 谢

历时将近两个月的时间终于将这篇论文写完，在论文的写作过程中遇到了无数的困难和障碍，都在同学和老师的帮助下度过了。尤其要强烈感谢我的论文指导老师—XX老师，她对我进行了无私的指导和帮助，不厌其烦的帮助进行论文的修改和改进。另外，在校图书馆查找资料的时候，图书馆的老师也给我提供了很多方面的支持与帮助。在此向帮助和指导过我的各位老师表示最中心的感谢!

感谢这篇论文所涉及到的各位学者。本文引用了数位学者的研究文献，如果没有各位学者的研究成果的帮助和启发，我将很难完成本篇论文的写作。

感谢我的同学和朋友，在我写论文的过程中给予我了很多你问素材，还在论文的撰写和排版灯过程中提供热情的帮助。 由于我的学术水平有限，所写论文难免有不足之处，恳请各位老师和学友批评和指正!

附录 A 附录章

如果将appendix.tex中所有内容删除，最后的论文将不会出现附录。

A.1 附录节

附录 B 附录另一章

B.1 附录另一章的一节

The Name of the Game

1.1 xxx

1.1.1 xxx

1.1.1.1 xxxx

1.2 xxx

1.2.1 xxx

1.2.1.1 xxxx

English words like ‘technology’ stem from a Greek root beginning with the letters $\tau\varepsilon\chi\ldots$; and this same Greek word means *art* as well as technology. Hence the name TEX, which is an uppercase form of $\tau\varepsilon\chi$.TeX (actually TEX), meaning of $\tau\varepsilon\chi$

Insiders pronounce the χ of TEX as a Greek chi, not as an ‘x’, so that TEX rhymes with the word blecchhh. It’s the ‘ch’ sound in Scottish words like *loch* or German words like *ach*; it’s a Spanish ‘j’ and a Russian ‘kh’. When you say it correctly to your computer, the terminal may become slightly moist.

The purpose of this pronunciation exercise is to remind you that TEX is primarily concerned with high-quality technical manuscripts: Its emphasis is on art and technology, as in the underlying Greek word. If you merely want to produce a passably good document—something acceptable and basically readable but not really beautiful—a simpler system will usually suffice. With TEX the goal is to produce the *finest* quality; this requires more attention to detail, but you will not find it much harder to go the extra distance, and you’ll be able to take special pride in the finished product.

On the other hand, it’s important to notice another thing about TEX’s name: The ‘E’ is out of kilter. This logo displaced ‘E’ is a reminder that TEX is about typesetting, and it distinguishes TEX from other system names. In fact, TEX (pronounced *tecks*) is

the admirable *Text EXecutive* processor developed by Honeywell Information Systems. Since these two system names are Bemer, Robert, see TEX, ASCII pronounced quite differently, they should also be spelled differently. The correct way to refer to TEX in a computer file, or when using some other medium that doesn't allow lowering of the 'E', is to type '—TeX—'. Then there will be no confusion with similar names, and people will be primed to pronounce everything properly.

此名有诗意

1.1 xxx

1.1.1 xxx

1.1.1.1 xxxx

1.2 xxx

1.2.1 xxx

1.2.1.1 xxxx

英语单词“technology”来源于以字母$\tau\varepsilon\chi$...开头的希腊词根；并且这个希腊单词除了 technology的意思外也有art的意思。因此，名称TEX是$\tau\varepsilon\chi$的大写格式。

在发音时，TEX的χ的发音与希腊的chi一样，而不是“x”，所以TEX与blecchhh押韵。“ch” 听起来象苏格兰单词中的loch 或者德语单词中的ach；它在西班牙语中是“j”，在俄语中是“kh”。 当你对着计算机正确读出时, 终端屏幕上可能有点雾。

这个发音练习是提醒你，TEX主要处理的是高质量的专业书稿：它的重点在艺术和专业方 面, 就象希腊单词的含义一样。如果你仅仅想得到一个过得去——可读下去但不那么漂亮——的 文书, 那么简单的系统一般就够用了。使用TEX的目的是得到最好的质量；这就要在细节上花功 夫, 但是你不会认为它难到哪里去，并且你会为所完成的作品感到特别骄傲。

另一方面重要的是要注意到与TEX名称有关的另一件事: “E”是错位的。这个偏移“E”的标 识提醒人们，TEX与排版有关，并且把TEX从其它系统的名称区别开来。实际上，TEX(读音为 tecks)是Honeywell Information Systems 的极好的Text EXecutive处理器。因为这两个系统的 名称读音差别很大，所以它们的拼写也不同。在计算机中表明TEX文件的正确方法，或者当所用 的方式无法降低“E”时，就要写作“TeX”。这样, 就与类似的名称不会产生混淆, 并且为人们可以 正确发音提供了条件。