Chapter 3. NP完全性理论

1

不可解问题

- 是否对于每个问题都有解决它的算法?
- 自然地，人们会想到另外一个问题：会不会所有的问题都可以找到渐进时间复杂度为多项式级(Polynomial Time)的算法呢?
- 答案是否定的。有些问题甚至根本不可能找到一个正确的算法来，这称之为“不可解问题”(Undecidable Decision Problem)。
- 例：
  - Hamilton回路
  - 问题是这样的：给你一个图，问你能否找到一条经过每个顶点一次且恰好一次(不遗漏也不重复)最后又走回来的路(满足这个条件的路径叫做Hamilton回路);
  - 这个问题现在还没有找到多项式级的算法。事实上，这个问题就是我们后面要说的NPC问题。

§3 NP完全性理论——计算机科学的局限性

- 可解性：问题及其可解性可用函数和可计算性来代替
- 可计算性理论：研究计算的一般性质的数学理论，它通过建立计算的数学模型(例如抽象计算机)，精确区分哪些是可计算的，哪些是不可计算的。
- 可计算函数：能够在抽象计算机上编出程序计算其值的函数。这样就可以讨论哪些函数是可计算的，哪些函数是不可计算的。
- Church-Turing论题：若一函数在某个合理的计算模型上可计算，则它在图灵机上也是可计算的。
- C-T论题结论：可计算性不依赖于计算模型
- 不可计算性：很多问题和函数是无法用具有有穷描述的过程完成计算的

3

不可计算问题：停机问题

- 停机问题：能否写一个程序正确判定输入给它的任何一个程序是否会停机?

设程序halts(P,X)总是正确地判定程序P在其输入X上是否停机:若停机，则返回yes；否则死循环，返回no。设另有一程序:

```
diagonal(Y){
    a: if  halts(Y,Y)  then
           goto a;
       else  halt;
}
```

- 功能：若halts断定当程序Y用其自身Y作为输入时Y停机,则diagonal(Y)死循环；否则它停机

diagonal(diagonal)是否停机？ 不可判定

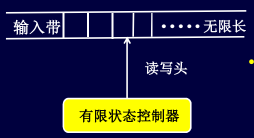
它停机当且仅当halts(diagonal,diagonal)返回否，也就是：

diagonal停机当且仅当它自己不停机，矛盾！

即：halts(P,X)并不存在，停机问题是不可解的！

4

图灵机



- 有单带、多带等变种，计算能力等价

- 依据控制器的状态和读写头所读字符，决定执行以下3个操作之一、之二或全部:

1)改变有限状态控制器中的状态;

2)读写头在相应的方格中写一符号;

3)读写头左、右移一格或不动。

- 确定型图灵机DTM：若对任一个状态和符号，要执行的动作都是唯一的
- 非确定型图灵机NTM：执行的动作是有穷多个可供选择

5

§ 2.3 NP完全性理论 (阅读课本P617)

- P类问题：一类问题的集合，对其中的任一问题，都存在一个确定型图灵机M和一个多项式p，对于该问题的任何(编码)长度为n的实例，M都能在p(n)步内，给出对该实例的回答(Caution: Trick)。即：多项式时间内可解的问题

- NP类问题(Nondeterministic Polynomial):一类问题的集合，对其中的任一问题，都存在一个非确定型图灵机M和一个多项式p，对于该问题的任何(编码)长度为n的实例，M都能在p(n)步内，给出对该实例的回答。

若NTM在每一步都恰有2步可供选择，则回答实例需考察$2^{p(n)}$种不同的可能性。

存在多项式时间的算法吗?

多项式时间内可验证问题 (指验证其解的正确性)

6

§ 2.3 NP完全性理论

- 之所以要定义NP问题，是因为通常只有NP问题才可能找到多项式的算法
- 我们不会指望一个连多项式地验证一个解都不行的问题存在一个解决它的多项式级的算法
- 很显然，所有的P类问题都是NP问题。也就是说，能多项式时间内解决一个问题，必然能在多项式时间验证一个问题的解——既然可以在polynomial时间内获得问题的正确解，那么验证任意给定的解也只需要比较一下即可！

§ 2.3 NP完全性理论

- 关键问题：是否所有的NP问题都是P类问题？
- 所有对NP问题的研究都集中在一个问题上，即究竟是否有P=NP？
- 目前为止这个问题还“啃不动”。但是，一个总的趋势、一个大方向是有的。人们普遍认为，P=NP不成立，也就是说，多数人相信，存在至少一个不可能有多项式级复杂度的算法的NP问题。

NPC问题

- 人们如此坚信P≠NP是有原因的，就是在研究NP问题的过程中找出了一类非常特殊的NP问题叫做NP-完全问题，也即所谓的NPC问题。C是英文单词“完全”的第一个字母。正是NPC问题的存在，使人们相信P≠NP。
- 为了说明NPC问题，我们先引入一个概念——归约(Reducibility)

归约

- 一个问题A可以归约为问题B的含义即是，可以用问题B的解法解决问题A，或者说，问题A可以“变成”问题B。
- “问题A可归约为问题B”有一个重要的直观意义：B的时间复杂度高于或者等于A的时间复杂度。也就是说，问题A不比问题B难。
  - 这很容易理解。既然问题A能用问题B来解决，倘若B的时间复杂度比A的时间复杂度还低了，那A的算法就可以改进为B的算法，两者的时间复杂度还是相同。

归约

- 归约具有一项重要的性质：归约具有传递性。如果问题A可归约为问题B，问题B可归约为问题C，则问题A一定可归约为问题C。
- 归约的标准概念：如果能找到这样一个变化法则，对任意一个程序A的输入，都能按这个法则变换成程序B的输入，使两程序的输出相同，那么我们说，问题A可归约为问题B。
- 我们所说的“可归约”是指的可“多项式地”归约(Polynomial-time Reducible)，即变换输入的方法是能在多项式的时间里完成的。归约的过程只有用多项式的时间完成才有意义。

P、NP及NPC类问题

- 多一归约
  - 假设L_1和L_2是两个判定问题，f将L_1的每个实例I变换成L_2的实例f(I)。若对L_1的每个实例I，I的答案为“是”当且仅当f(I)是L_2的答案为“是”的实例，则称f是从L_1到L_2的多一归约，记作：$L_1 \leq_m L_2$(传递关系)
  - 直观意义：将求解L_1的问题转换为求解L_2的问题，而问题L_1不会难于L_2
- 多项式时间多一归约：若f是多项式时间可计算，则上述归约称为多项式时间多一归约，也称多项式时间变换。记作：

$$L_1 \leq_m^p L_2$$

12

P、NP及NPC类问题

- NPC问题：对于一个(判定性)问题q，若
  (1) q∈NP
  (2) NP中任一问题均可多项式时间多一归约到q
  则称问题q为NP-完全的(NP-complete，NPC)
- NP-hard问题：若问题q仅满足条件(2)而不一定满足条件(1)，则问题q称为NP-难的(NP-hard)。显然：NPC⊆NP-hard (NP-hard是一个更大的集合)
- NPC和NP-hard关系
  NP-hard问题至少跟NPC问题一样难。
  NPC问题肯定是NP-hard的，但反之不一定

例：停机问题是NP-hard而非NPC的！

∵该问题不可判定，即无任何算法(无论何复杂度)求解该问题
∴该问题∉NP。但是
可满足问题SAT $\le_p$停机问题

13

P、NP及NPC类问题

- NP=?P
  ∵确定型图灵机是非确定型图灵机的特例，∴P⊆NP
  是否有NP⊆P？即是否NP=P？

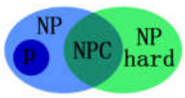
美国麻省的Clay数学研究所于2000年5月24日在巴黎法兰西学院宣布：对七个"千年数学难题"中的每一个均悬赏100万美元，
而问题NP=?P位列其首：

1. P问题对NP问题
2. 霍奇猜想
3. 庞加莱猜想(2002.11-2003.7，俄罗斯数学家佩雷尔曼在3篇论文预印本中证明了几何化猜想，2006被授予菲尔兹奖)
4. 黎曼假设
5. 杨-米尔斯存在性和质量缺口
6. 纳维叶-斯托克斯方程的存在性与光滑性
7. 贝赫和斯维讷通-戴尔猜想

14

P、NP及NPC类问题

- P、NP、NPC和NP-hard之关系
  NPC是NP中最难的问题，但是NP-hard至少与NPC一样难



- 如何证明问题q是NP-hard或是NPC的？
  若已知q'∈NPC或q'∈NPH，且q' $\le_p$ q，则q∈NPH；若进一步有q∈NP，则q∈NPC。
  即：要证q是NPH的，只要找到1个已知的NPC或NPH问题q'，然后将q'多项式归约到q即可。若还能验证q∈NP，则q是NPC的。
  ∵ NP中任意问题均可多项式归约到q'，由于$\le_p$有传递性
  ∴他们也都能多项式归约到q，由定义可知q是NPH的

15

NP-完全性理论

- Cook的贡献：第一个NPC问题
  史提芬·库克(Stephen Arthur Cook，1939-)NP完全性理论的奠基人，他在1971年论文"The Complexity of Theorem Proving Procedures"中，给出了第一个NP完备的证明，即Cook定理：可满足性问题(Satisfiablity problem)是NP完全问题，亦即SAT∈NPC。且证明了：
  SAT∈P当且仅当P=NP

Cook于1961年获Michigan大学学士学位，1962和1966年分获哈佛大学硕士与博士学位。1966-1970，他在UC Berkeley担任助教授；1970年加盟多伦多大学，现为该校CS和数学系教授，他的论文开启了NP完备性的研究，令该领域于之后的十年成为计算机科学中最活跃和重要的研究。因其在计算复杂性理论方面的贡献，尤其是在奠定NP完全性理论基础上的突出贡献而荣获1982年度的图灵奖。

16

NP-完全性理论

- NP-完全性理论的局限性
  易解问题：可多项式时间内求解的问题
  难解问题：需超多项式时间求解的问题
  NP-完全性理论既没有找到第二类问题的多项式时间的算法，也没有证明这样的算法就不存在，而是证明了这类问题计算难度之等价性(彼此间困难程度相当)。因此，NPC具有如下性质：若其中1个问题多项式可解当且仅当其他所有NPC问题亦多项式可解
- 难解问题与易解问题之相似性
  1) 最短/最长简单路径
  单源最短路径问题：对有向图G，时间O(VE)，P问题
  两点间最长路径：NPC问题，即使所有边上权为1
  2) 欧拉环/哈密尔顿圈(G为无向图或有向图)
  欧拉环：G中有通过每条边恰好一次的环？P，多项式时间可解
  哈氏圈：G中有通过每个顶点恰好1次的圈？NPC

17

NP类问题的求解

- 减少搜索量
  简单算法是穷举搜索，时间为指数
  减少搜索量：分枝限界法，隐枚举法、动态规划等
  可以提高效率，但时间复杂度不变
- 优化问题
  降低优化要求，求近似解，以得到一个多项式时间的算法。即：找寻在容许的时间内得到容许精度的近似最优解的算法
- 近似比：近似算法所能得到的可行解(次优解)与最优解之间的比值

18

关于归约总结

- 减将一个问题归约为另一个问题不存在一劳永逸的方法，一些归约过程极其简单(e.g. 哈密顿回路归约为TSP)，一些归约极其复杂。

注意事项及一些技巧策略：

1.必须满足的形式：将问题X的任意输入转换为关于问题Y的某些输入；

2.利用归约源的限制优势：从哈密顿环问题进行归约比从TSP问题进行归约更为直接。因为TSP问题中，边的权重可以是任意正整数，而不是只可以取1或者0

3.(非常有用！！！)寻找特例：某些NPC恰恰是其他NPC的特例，比如partition problem是knapsack problem的特例，如果你知道问题X是NPC，并且X是Y的特例，那么Y必定也是NPC，为什么？

19

关于归约总结(续)

- 注意事项及一些技巧策略：

3.（非常有用！！！)寻找特例：某些NPC恰恰是其他NPC的特例，比如partition prob是knapsack prob的特例，如果你知道问题X是NPC，并且X是Y的特例，那么Y必定也是NPC，为什么？

因为Y比X更具一般性，问题Y至少与X一样难！！！

4. 寻找合适的规约源：

有的时候，我们会选择跨域归约策略，3-CNF可满足性问题是一个进行跨域归约合适的归约源。既可以规约到团问题(Graph)也可以归约到子集和(knapsack branch)；

在图问题中，如果需要选择部分图，且无需考虑顶点顺序，那么顶点覆盖问题通常是一个合适的归约源

20

关于归约总结(续)

- 注意事项及一些技巧策略：

5. 获取最大收益和最小补偿：

哈密顿环问题的输入图G转化为TSP问题加权图G’时，我们当然可以使用G中出现的边作为TSP问题相应的边。我们对这些边赋予非常小的权重：0，我们利用这些边可以获得巨大的收益。

21