

简要叙述可计算性与计算复杂性的发展历程,并谈谈你对这门课的认识

20 世纪 30 年代开始,人们开始研究计算的意义.1931 年,赫尔布兰德研究了递归函数,并向哥德尔叙述了自己的成果,同年,丘奇和他的两个学生提出 λ 演算,并提出所有由 λ 运算定义的函数都是可计算函数.

1934 年,哥德尔提出了"赫尔布兰德-哥德尔递归函数"理论,使用递归函数对可计算函数进行了自己的定义.

1936 年,图灵提出了图灵机的概念,并注意到了丘奇和哥德尔的工作,在 1937 年,图灵证明了图灵机可计算函数, λ 可定义函数和一般递归函数是等价的,至此,可计算性领域的多种理论联系在了一起,后来数学家波斯特也提出了自己的对于可计算函数的理论,而这一理论与图灵等人的也是相一致的.

1947 年,苏联数学家马尔科夫发表<算法论>,首次提出了算法这个概念,但是他同以前定义的递归函数和可计算函数都是等价的.随着这些观点的深入人心,判定问题成为了数理逻辑的一个重要分支,今天的可计算理论也基本形成了.人们第一次认识到,有一些问题是找不到算法解的,也没办法在有限步骤内判断他是不是无解的.

虽然在过去的几十年里我们对计算复杂性理论做了大量的研究,但是我们仍不知道,是什么使得某些问题难以计算,有的问题容易计算.目前比较重要的成果是,发现了一个按照计算难度给问题分类的完美体系,按照这个理论,我们能判断出一个问题是不是难以计算的.

可计算理论给出了目前计算机的计算能力的上限,如果我们设计一门程序设计语言,那么他的计算能力是有上限的,我们可以通过计算理论的相关方法论证这门语言是具备完整的计算表达能力的,能够表达所有的可计算函数,那么这样的语言就是可用的了.对于计算机体系结构的创新也难以突破图灵机等理论给出的能力上限,这也是量子计算机等研究领域的一个关键问题,而现实中更普遍的问题是计算复杂性问题,往往决定我们是否能解决一个问题的关键因素就是复杂性,分析算法的时间和空间复杂性,对于改进算法的性能十分重要,高性能的算法使我们一直想要的,特别是在机器学习,大数据等热点背景下,往往潜藏着超高的计算复杂度,如何提高算法的性能是决定其方法是否具有实际价值的关键所在.

对于日常的科研和工作而言,掌握计算复杂性理论对于提高算法设计能力有着重要的意义,而掌握可计算性理论和理论中提出的各种方法和思路,对于解决问题,提出自己的观点并证明自己的观点等科研能力也有着重要作用.