1963年全国数理逻輯专业学术会议论文选集

中国电子学会电子計算机专业委員会編

国防工业出版社

1963年全国数理邏輯专业学术会議論文选集

中国电子学会电子計算机专业委員会編

国防工業出版社

内 容 簡 介

本文集系汇集1963年10月在西安举行的第三次全国計算技术經驗交流会的数理邏輯組上所发表的論文而成。其中包括：多值邏輯，算法論，証明論，数学基础，自动机理論，程序理論等論文及論文摘要，共23篇。

本文集可供数理邏輯，自动机理論，程序理論等方面的研究人員和教师参考。

1963年全国数理邏輯专业学术会議論文选集
中国电子学会电子計算机专业委員会編
*
国防工業出版社 出版
北京市书刊出版业營业許可証出字第074号
新华书店北京发行所发行　　各地新华书店經售
国防工业出版社印刷厂印装
*
787×1092 1/16　印張 8 1/8　187千字
1965年9月第一版　1965年9月第一次印刷　印数：0,001—3,400册
統一书号：15034·869　　定价：(科八)1.70元

目　录

論　文

摘　要

对于数理邏輯和計算技术的一些看法●(代序)

胡 世 华

§1 数理邏輯是怎样一門科学？它研究些什么問題？

数理邏輯是一門用数学的方法来研究推理規律的科学。研究推理規律就是研究前提和結論的关系，研究从哪些前提可以推出哪些結論，不能推出哪些結論，等等。

数理邏輯在研究推理規律的时候，自然要研究推理的方法。从前提推出結論来有沒有机械的方法或者計算的方法（算法）？換言之，能不能够为某些推理过程建立一套規則，按照这些規則一步一步地机械地进行推理，把結論推出？或者像数学中的加减乘除那样，可以按照确定的方法机械地把結論“算”出来？这样就把数理邏輯的研究引导到对于机械性、計算、可計算性等問題的研究上去了。由于这种研究，形成了数理邏輯研究中的一个方向，这就是算法理論。算法理論不仅在数理邏輯里，就是在整个数学中也是一个重要的方向。

由于对于机械性，計算，可計算性进行数学的研究，需要建立相应的数学概念，并从而建立“計算机”的概念。“計算机”在数理邏輯里是一个有严格数学定义的概念。有了这样的概念，“机械性”、“計算”、“可計算性”等概念也就可以随之建立，因之对于机械性、計算、可計算性可以进行数学的研究。

数理邏輯开始于邏輯問題的研究，进一步以数学中的邏輯問題，以計算、可計算性，以机械性为自己的研究課題。在計算技术的迅速发展中，数理邏輯与計算技术相結合，显示出对計算技术的作用和意义。

§2 計算技术促进数理邏輯的进一步发展

在电子計算机出世之前，已經存在着与計算机理論研究有关的两种理論，一种是网絡理論，另一种是1936年数理邏輯学家图林在建立起数学的“計算机”和“通用計算机”概念之后而开始发展的图林机器理論[1],[2]。在計算技术实践的启示和概括下，在这两种理論的基础上发展出自动机理論。自动机理論是一种計算机理論，它与算法理論密切結合在一起，二者的研究范圍是难以划分的。这种研究对于計算机的功能（計算机能做些什么，不能做什么）問題，結构（机器如何由机器元素組織起来）問題的研究，对于計算机的功能設計、邏輯設計、建立計算机的新的体系，提出新的設計思想、設計方法是有意义的。

在自动机理論研究中包括两种傾向。一种是着重研究自动机的功能特点，可以称为自动机的功能理論或自动机的算法理論。这是一种特殊形式的算法理論。在这一次会議上宣讀的論文中，陶仁驥的論文就是一篇这种傾向的論文。在陶仁驥的論文中給出了一个“自动

● 作者应会議組織者的要求，在大会上作了一篇介紹数理邏輯及其有关問題的“綜合报告”。本文是根据那篇报告整理而成的。

机"的概念，这是一个反映現代数字計算机整机功能的数学模型。所給出的概念能够比較直接和自然地反映現实的机器，对于可能实現的机器体系也提供了十分寬广的設想和探索的可能性，对于新的設計思想的提出是有意义的。

自动机和算法的一般理論，目前正处于理論建立和发展的早期。理論已經建立起来，还不完善，发展得还不够，因之对于計算技术的实际作用还不显著。然而理論的进一步发展，对于計算机新体系的建立，新的設計思想的提出是可以預期的。

自动机研究中的另一种傾向是研究自动机的結构特点。从研究对象来說，它是对于一种特殊的自动机，有穷自动机的研究。这一傾向的研究也是重要的，有的时候甚至是更加重要的。自动机的功能和結构两种傾向的研究互相促进，互相关連；前者联系計算机的功能設計問題，后者联系計算机的邏輯設計問題。在有穷自动机、邏輯設計的研究中，代数方法是很重要的。这一傾向的研究的重要性，在于对計算机的各个部分(例如运算控制部分)的功能，从它的結构方面来作整体性的分析和理解。这就是說，对于机器的这一部分能进行的活动，进行全面的代数刻划。由之，我們可以知道所需要的性能是否已經具备。这些問題的解决意味着不是单凭經驗，而是凭理論指导以从事邏輯設計工作。这一傾向的研究对邏輯設計是有指导性的。提到会議上来的吉林大学王湘浩同志所領导的这一方面的工作是值得重視的。

上面談到一些与計算机設計有关的数学研究，下面我們来介紹一点与使用計算机有关的数理邏輯和有关理論的研究，主要是在解决特定脑力劳动自动化中所要进行的研究。

数值計算是一种脑力劳动，研究数值計算規律的科学是計算方法。除了数值計算之外，邏輯推理也是一种很大量的脑力劳动，这种脑力劳动的机械化，自动化規律的研究的一个主要基础是数理邏輯。

在計算技术的发展中提出的一个十分尖銳的問題：机器的速度、数量是按数量級增长的，然而为通用計算机解題作准备工作的計算数学工作人員的数目难于按数量級上升，这就大大地限制了計算机应用的可能性。为解决这个矛盾，便产生了一門叫做程序自动化的学科。

程序設計和程序自动化現在还是一門經驗的学科，但正在迅速地奠定自己的理論基础，逐步向精确化过渡。也就是說，正在形成一門所謂程序理論的学科（它至少应包含一种适于处理要在机器上实現的算法的算法理論，亦应包含有关程序設計和程序自动化实踐中提出的其他理論問題，例如形式語言的理論、語法理論、語义理論、形式語言的翻譯等）。程序理論和数理邏輯有极密切的关系。程序理論不很好建立，程序自动化問題便不能从根本上得到完滿的解决。

論文集中董韞美、李开德的論文，作为形式語言的一种語法理論，提出了具有判定合式公式和分解合式公式的通用算法的一类很广的語法模型，并給出了判定算法的最佳估值。工作的意义是显然的。

在計算工作中往往有些問題需要进行大量繁重的解析演算。其中有些工作不一定难，然而十分容易出錯，却可以交給机器去做。

程序自动化和解析演算自动化的实現可以大大減輕机器解題的准备工作。

数学研究工作中数学定理的証明占用了极为大量的时間和精力。用計算机来协助数学

工作者进行定理証明的研究，在数理邏輯研究中提供了丰富的成果，使我們知道这个問題解决的可能性和局限性。看来前景是光明的。在一定范圍之內，机器可以成为人从事数学定理証明工作中的有力工具。

§3 数理邏輯与数学

数理邏輯与数学分析、代数、几何一起，都是数学中的基本研究。然而，数理邏輯对于数学这一門科学本身的发展，又有自己另外的任务。

数理邏輯对于数学来說，它还要研究数学理論本身的邏輯結构。它研究：数学理論是按什么規律构造起来的？数学概念、命題之間存在着怎样的邏輯关系？哪些命題可以証明，哪些命題不能証明？等等。数学的邏輯是数理邏輯研究中的一个相当重要的課題。

哥德尔于 1931 年的数理邏輯研究[3],[4]中，发現了数学中的“潜在的絕境”，証明了存在着形式地不可証明的数学“定理”。另一方面，判定問題的研究中又給出了那样的数学命題类，其中的命題是否是一个数学定理存在着确定的判定方法。这种研究对于数学工作者的工作是有一定的具有一般性的指导意义的。由于数理邏輯的这种研究，弄清楚了一些数学历史上的十分困难的問題，从而得到了解决。例如几十年来所不能解决的困难問題如貝恩賽德問題、图埃的字的問題，在数理邏輯所发展的方法指引之下得到了順利的解决。数理邏輯已經成为一般数学研究中的一种方法和工具了。事实証明，在一定的情况下，这种方法和工具是有力的。

数学中的初等部分，通过数理邏輯的研究被人們理解得比較深入了。初等代数和初等几何的判定問題已經解决[5]，初等代数和初等几何中定理的証明已經給出了确定的方法。所以，对于数学教育工作者，具有一定的数理邏輯素养看来是有益的。

数理邏輯的基本內容，在一般数学教育——包括綜合性大学的数学专业和师范院校的数学专业的数学教育中，应当受到重視而給以一定分量的安排。

不少同志注意到数理邏輯与計算技术、自动化的关系，由之认为数理邏輯是与尖端技术联系的，这是正确的。然而也有同志以为数理邏輯只与尖端技术有关，与一般数学的关系不大，这是不正确的。

§4 科学的数学化和数学的机器化

計算机在許多方面引起广泛和深远的影响，这里想簡单地談一些它在科学研究中的影响和作用。

由于电子計算机的出現，使得許多原来很少使用数学的科学領域都使用数学了，在很大的程度上扩大了数学的应用范圍。有人称这一現象为“科学的数学化”。科学在数学化的过程中，出現了用計算机解决問題的可能性，由之，計算方法的研究成为十分迫切和重要。另一方面，数理邏輯在这件事情中也起着它的作用。

从一堆数据、已知的信息形成概念，构成理論体系，包括許多邏輯問題。在科学的数学化中，要从已知信息形成数学的概念，构成数学理論体系，許多邏輯問題是可以用数学方法来处理的，并从而用机器对数据进行邏輯加工。这样，可以大大簡化科学研究的某些环节的工作，縮短研究工作的周期，提高研究工作的效率。

在自然科学、技术科学的理論研究中，除了大量的数值計算之外，还有大量的邏輯推理活动，由于电子計算机的应用而可以自动化，从而提高理論研究的效率。这一研究方向的初步目标是科学技术情报工作的自动化，例如应用計算机作文献檢索，作文摘；进一步的目标是用計算机进行演繹、归納推理，根据已知情报进行邏輯推理和作出一定的判断。数理邏輯用数学的工具和方法，研究演繹、归納推理，提供了这一方向的研究方法和途徑。

上面提到的由于計算技术的发展而促进的脑力劳动的自动化，只是可能的自动化中的一部分，这件事无疑将对科学发展起巨大的推进作用，从而引起科学发展的革命性变化。周恩来总理在1956年的时候就对这件事作过明确的評价了。他讲到，由于現代科学研究，由于电子計算机的出現“已經可以开始有条件地代替一部分特定的脑力劳动，就像其他机器代替体力劳动一样，从而大大提高了自动化技术的水平。这些最新的成就，使人类面临着一个新的科学技术和工业革命的前夕。”就它的意义来說，远远超过蒸汽和电的出現而产生的工业革命（《社会主义教育課程的閱讀文件汇編》第二編，第967頁）。

在科学数学化和数学机器化的过程中，数理邏輯的研究是重要的。

§5 一些看法和建議

在电子計算机的研究、試制工作中，技术問題是带有根本性的問題，这是显然的。除此之外，有关理論問題的研究同样是带有根本性的，同样是重要的。无論在机器的設計制造或者在机器的使用方面，都需要我們来进行大量的理論研究工作，这些工作有待于数学工作者的参与和承担。

簡单地回顾一下計算机的发展历史，可以进一步闡明上述观点。我們不去考察計算机的創始人巴斯格（1623—1662）和莱布尼茲（1646—1716）（他們都是数学家、理論家）的思想，他們离开我們太遙远了。我們从十九世紀三十年代看起，看看理論和技术这两方面是如何影响計算机的研制的。自然，生产、科学技术发展的客观需要是計算机发展的巨大动力；生产和科学技术如何推动計算技术的发展，則不是我們所要討論的問題。

巴貝治于1833年已經有了比較完善的通用計算机的概念。我們可以說，在本世紀五十年代中运轉的所有电子計算机，在体系組織方面的做法，巴貝治都曾經想到过。可是巴貝治的机器一直到他1871年逝世都未能做成；他的儿子继承他的工作，一直到二十世紀初也未能最后把机器制成。由于技术条件的不成熟，巴貝治的机器的理想还不能成为現实。巴氏的机器在技术条件成熟起来之后，才一步一步地实現，这就是1937年用继电器制成的MK1和以后的MK2。到第一架电子計算机出世为止，計算机理論一直是走在技术之前的。

第一架电子計算机ENIAC是为了計算彈道而設計的专用电子計算机。使用中发現它有更大的用处，由于馮·諾依曼的参与工作，系統地扩大了机器的用途。由之，由馮·諾依曼的創議而研制了第一架通用电子計算机EDVAC；同时他与波克斯和哥尔德斯丁一起，对机器的邏輯体系、程序的基本問題进行了研究[6]，为現在通常称之为馮·諾依曼設計思想的机器奠定了基础。由之在不太长的时間之內，很快由EDVAC的启示而研制了EDSAC和ACE。計算机的研究不久就发展成为一門独立的科学。

从这簡单的历史回顾中，我們可以得到两点启示：

第一，如果技术条件不成熟，尽管像巴貝治已經有了比較完善的通用計算机的概念，还是不能把机器实际地制造出来。沒有技术就沒有机器。

第二，沒有理論的先行，机器也造不出来。技术条件成熟了，要不是已經有了一个完整的計算机的概念（像巴貝治的概念），MK1和MK2这样的机器也是造不出的。通用电子計算机的体系是由馮·諾依曼和波克斯等理論家的参与工作而开始建立的。布斯在他的书[7]中說：“当无綫电工业的技术发展使无綫电真空管的大量生产成为可能时，应用这些装置于計算机器的制造的道路就已經打开了。这一步这样迟才实現，在今天回顾起来是令人詫异的。因为計算机中所需要的真空管的性能远比无綫电中所需要的要簡单。”其实这里沒有什么值得詫异之处。理論沒有与技术結合，机器自然是做不出来的。馮·諾依曼，波克斯等人参加了工作，带去了理論，計算机就迅速发展起来了。这說明了在技术条件已經成熟的前提下，沒有理論的指导，机器也是做不出来的。

在五十年代里，技术有了很大的进步，一下子把所有的理論儲备都用光了。古典分析中的数値計算方法不够用了，开关网絡理論不够用了，馮·諾依曼的設計思想不够用了。理論大大地落后于技术了。这一情况在有些科学工作者的思想中得到了歪曲的反映，产生了忽視理論的傾向。这一情况在实际机器上的反映是像STRETCH这样的机器的研制，基本上是老体系，用器材堆积起来，花費很多也沒有做出与付出的代价相称的好机器。

从現在外国的一般情况看，理論一般說是落后于实际的。进入六十年代之后，情况虽然略有好轉，有些方面稍为赶上了一些，然而还是很不够的。有些学者意識到理論的重要，因而认为体系、理論的研究将成为計算技术发展的重点。自然，这些說法是否正确，是有待于討論的。

理論与技术两方面，沒有哪一方面“太强調”的問題，强調是不会过头的。倒是强調了一方面，忽視了另一方面，却是不对的。两方面中的任何一方面都不能代替另一方面。

前面我們多談了一些数理邏輯，这是由于会議的組織者要求介紹数理邏輯，由于第三分組本来是一个拟議很久的数理邏輯学术会議，也由于个人专业知識的局限和准备时間的仓促，来不及安排更多的內容。其实，計算机的理論研究所需要的数学岂止是我們所談到的这些呢。代数不用說了，除此之外，概率論、数理統計、数論、几何、拓朴、集論等都会大有用武之地。从計算机研究中提出的数学問題是十分广泛和深刻的，也难于想像得出哪一門数学在計算机研究中不能发揮其作用。数学工作在計算技术的发展中所起的作用也是带有根本性的。世界上第一架計算机、第一架通用計算机、第一架通用电子計算机的体系都是出自数学工作者之手；自然，沒有工程設計，机器也是制造不出来的。在計算技术的进一步发展中，所需要的数学将是无穷无尽的。随便怎么多的数学工作者参加到計算机的研究工作中来，都是不会嫌多的。我国的計算技术事业是我国科学技术发展中的重要的一环，而数学工作者在其中正可以大大地显示他們的才能，把精力貢献于計算技术，在我国社会主义建設事业中，在反对帝国主义、現代修正主义和各国反动派的革命斗爭中作出貢献。

不参加計算机研究的数学研究人員对于計算机也不能不有所了解。在机器生产水平的不断提高，数学工作的机器化的进步中，仅仅借助于紙和笔的数学研究将成为历史陈迹。从我国电子工业的发展状況，从机器数学的发展状况看，这将不是非常遙远的事情。

最后，願乘此机会向各地負責数学教学和研究的組織、行政工作的同志們提出一点建議，希望考虑。

目前数理邏輯的队伍还是太小，力量太薄弱。从計算技术和科学的发展，从数学中各分支学科的平衡发展看，都远不能满足需要。几年来，数学队伍进行了必要的調整、巩固，今后应当尽可能在进一步巩固的基础上逐步进行充实和提高，使数理邏輯队伍能够稳步发展。1958年大跃进以来，在党的三面紅旗的光輝照耀下，数理邏輯开始形成一支队伍，有一批青年經受了一定的数理邏輯的訓练。我們应当尽可能使他們在数理邏輯工作中坚持下去，不断前进，以便他們能够逐步承担这方面不断发展的教学、研究工作的任务。

在上面的讲話中，我們讲到了論文集中和計算技术关系較密切的論文的一部分。文集中还包括数理邏輯基本理論方面的論文，也都是有意义的研究成果。我們在这里无意对所有論文作出全面的学术評价。目前，我們的工作还比較少，然而，第一次数理邏輯的专业会議毕竟作为中国电子学会第三次全国計算技术經驗交流会議的一个組成部分而举行了。有了第一次，就会有第二次、第三次。可喜的是，我們見到在这一方面的青年正在茁壮地成长起来。我相信我們在党的三面紅旗的光輝照耀下，在自力更生、奋发图强的方針指引下，我們事业的发展将是比較迅速的。

以上所讲的可能有錯，希望同志們批評指正。

参考文献

〔1〕 A. M. Turing, On computable numbers with an application to the entscheidungsproblem. Proceedings of the London Mathematical Society, Ser. 2, vol. 42 (1936～1937), 230～265; Correction, ibid., vol. 43 (1937), 544～546.

〔2〕 M. Davis. Computability and Unsolvability, 1958.

〔3〕 K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. Monatschefte für Mathematik und Physik, vol. 38 (1931), 173～198.

〔4〕 S. C. Kleene. Introduction to metamathematics, 1952.

〔5〕 A. 塔尔斯基, J. C. C. 麦克經賽, 初等代数与几何的判定法, 1951; 科学出版社1959.

〔6〕 A. Burks, H. H. Goldstine and J. von Neumann, Preliminary discussion of the logical design of an electronic computing instrument, 1947.

〔7〕 A. D. 布斯, K. H. V. 布斯, 快速电子計算机, 1956; 科学出版社1958.

多种类递归算法

递归算法論IV

胡 世 华

引 言

在数理邏輯和証明論的研究中需要把直观数学中的“可計算性”或“能行性”精确化、数学化，建立相应的数学概念，由此产生了一系列的企图反映直观的可計算性、能行性的数学概念。如文献〔1〕中所讲，其中重要的有Skolem-Gödel-Herband-Kleene(1923—1936)的“递归函数”，Church 的“λ-轉换演算”(1935—1936)，Turing 的“計算机”(1936)和 Марков 的“正規算法”(1951)等。与建立这些概念的同时，发展了相应的理論，如递归函数論，图林机器理論等。我們称这种理論为能行性理論(見文献〔1〕§1)。这些概念，如所周知，在一定意义下是等价的。从当时建立这些概念和理論的目的看，难于說它們有什么原則的差别，也不大能說它們之中那一个比其他的更好。在解决某一問題时，可能某一概念、某一理論工具更为方便一些。例如，在解决字的問題时（見文献〔4〕，〔5〕），看来正規算法的概念和理論工具是比較方便和自然的。

在計算技术的发展中，对能行性理論提出了新的要求。

人們在工程技术工作和科学研究工作中往往找到或发現某种借以进行計算的一套一套直观的規則、法則。这种計算法則之为計算法則，并不一定有一个通过严格数学定义的准則，然而人們却可以根据这种直观的法則对于給出的数据进行計算，算出所要的結果，达到一定的目的。我們可以称这样一套法則为一个直观算法。計算技术要求解决这样的問題，使机器来实現原来由人根据直观算法进行的計算。自然，先决的問題是，如何把直观算法精确化，以便我們有可能使机器按照經过精确化的算法进行計算。这就要求我們有一套理論工具，使我們能够自然而直接地描述直观算法，使精确化之后的算法不失它所反映的直观算法的特点，同时我們又可以对之作数学的处理、轉换和加工。在文献〔1〕中建立的递归算法的概念和企图发展的理論，就是在这种意图之下的产物。作者覺得，以往的概念和理論，至少在它們已經存在的形式之下是有不足之处的，它們并不能自然而直接地表达直观算法。

在文献〔1〕，〔2〕，〔3〕中我們显示了，或者，企图显示用递归算法可以比較自然地表达正規算法，图林机器等。然而这三篇文章中处理問題时运用了一类称为核函数的递归函数。这是一个由归納定义确定的函数类，它在文章中起了主要的作用，甚至可以說，文章主要就是讲核函数的，显示运用核函数工具可以构造其他形式的算法，并且令我們想到直观算法也可以在一定意义上較为自然地用核函数（加上少数其他函数）来表达。文章沒有显示出如何用递归算法直接描述直观算法。正因为如此，作者听到这样一些評論，例如說，文章

中讲到可以把 Марков 的正規算法理論，图林机器理論簡捷而自然地包容到遞归算法論的理論体系中来，这并不很清楚。在我們还沒有充分发展这一理論时，这样的感覚是可以理解的。

我們在本文中引进了“多种类遞归算法”的概念，可以比較明确地看出，各种能行性理論确实是可以很簡捷而自然地包容于遞归算法論的理論体系之中的。

作者相信，由于“多种类遞归算法”概念的引入，可以比較清楚地显示，遞归算法論的理論工具是可以自然而直接地描述直观算法的。“多种类遞归算法”概念的引进是遞归算法論发展中的应有的一步。理論有待于进一步发展；在理論的进一步发展中，我們要求于理論的那些特点将会得到更加充分的显示。

“多种类遞归算法”中的“多种类”字样取自数理邏輯中的現成的名詞“多种类謂詞演算”，“多种类理論的邏輯”等，可参看A.Schmidt[6],[7]，王浩[8],[9]的論文。

作者在文献〔1〕§2中說过，遞归算法論是把通常的遞归函数論由自然数推广到一个有穷自由基的自由半群的理論；遞归算法論中的“遞归函数”可以看作是自然数的“遞归函数”(“部分遞归函数”)概念的推广。早在1958年L.Kalmar訪問我国时，他与作者談到R.Peter把“遞归函数概念推广到脉絡的思想。在文献〔1〕,〔2〕,〔3〕三文发表之后，作者和他的同事們又見到 Peter[10], V. Vučković[11],[12], J.McCarthy[13],[14], G.Asser[15]等的文章。今年，作者接到 H.B.Curry 的信，說到他在文献〔16〕中已有把“遞归函数”推广的思想。所举各学者的工作都是对“遞归函数”概念的推广。

以下，在§4中証明了多种类遞归算法的主要定理之后，在§5和§6两节中举了两个例子，借以說明如何不用核函数而直接以多种类遞归算法表达图林机器和正規算法。这两节內容的选擇也是为了便于使讀者可以与〔1〕中§5和§6两节相对照。

§1 遞归可枚举集

在本文中，我們以$\mathfrak{A}$表示字母表$\mathscr{A}$中所有字构成的集，空字$\odot$也在$\mathfrak{A}$中。

在自然数的遞归函数論中，称一个由一遞归全函数枚举的集，即一遞归全函数的值域，为一遞归可枚举集。E.L.Post[17]把空集也算作遞归可枚举集。我們采用 Post 的定义：一自然数集$\mathfrak{B}$称为遞归可枚举集，当且仅当，$\mathfrak{B}$是空集或者$\mathfrak{B}$是一遞归全函数的值域。在自然数的遞归函数論中有

輔助定理 1 $\mathfrak{B}$是一遞归可枚举集，当且仅当，$\mathfrak{B}$是一遞归(全或偏)函数的定义域。

证 以下的証明中使用了 Kleene[5]的結果和符号。

当$\mathfrak{B}$是空集时，輔助定理1是显然成立的，故下面的証明中可以假設$\mathfrak{B}$是非空集。

(部分) 遞归函数 f 可以写作

$$f(x)=U(\mu y T_1(e,x,y))$$

其中 e 为仅由 f 所确定的常数。令

$$\varphi(x)=\begin{cases}(x)_0 & \text{当 } T_1(e,(x)_0,(x)_1)\\ (\mu y T_1(e,(y)_0,(y)_1))_0 & \text{其他情形}\end{cases}$$

由于

$$(\exists y)[\varphi(y)=x]\Longleftrightarrow x\in\delta f\,(f\text{ 的定义域})$$

所以，φ 是一全函数，并且枚举 f 的定义域。反之，已知 φ 为枚举集$\mathfrak{B}$的全函数，則可找一递归函数 f，使 $\delta f=\mathfrak{B}$，如下：

$$f(x)=\mu y[\varphi(y)=x]\quad .$$

因为

$$f(x)\text{有定义}\Longleftrightarrow(\exists y)[\varphi(y)=x]$$

证毕。

当然，我們也可以像在自然数的递归函数論中那样在递归算法論中定义递归可枚举集为空集或递归全函数的值域。輔助定理 1 也是成立的，証明类似于自然数的递归函数論中的証明，故从略。由此可見，我們完全可以这样来定义递归可枚举集：$\mathfrak{B}\subset\mathfrak{A}$是一递归可枚举集，当且仅当，$\mathfrak{B}$是一$\mathscr{A}$中递归函数的定义域。这样，我們說$\mathfrak{B}$是$\mathscr{A}$中的递归可枚举集。这里我們所定义的递归可枚举集是$\mathfrak{A}$的子集。显然，我們可以很容易地把定义推广到$\mathfrak{A}$的 n 度笛卡儿空間

$$\mathfrak{A}^n=\underbrace{\mathfrak{A}\times\cdots\times\mathfrak{A}}_{n}$$

的子集：$\mathfrak{B}\subset\mathfrak{A}^n$ 是一递归可枚举集，当且仅当，有一 n 元的$\mathscr{A}$中的递归函数 f，$\mathfrak{B}$是 f 的定义域。这样，我們仍称$\mathfrak{B}$是一$\mathscr{A}$中的递归可枚举集。

在結束本节时，我們提出下面的一項注記：設$\mathfrak{B}$是由一递归函数 f 所枚举的不空集，則$\mathfrak{B}$是一递归可枚举集。这里沒有假定 f 是一全函数。这是因为，由輔助定理 1，f 的定义域是一递归可枚举集，因此有一递归全函数 g 枚举它，令

$$h(x)=_{df}f(g(x))$$

則显然 $h(x)$ 是一递归全函数，而且$\mathfrak{B}$也是由 h 来枚举的。这就是說，一个由递归（全或偏）函数枚举的不空集，总可以由一递归全函数来枚举它。由此可見，上面討論中的“全”字是可以省去的。結合輔助定理 1，“递归可枚举集”，“递归函数的定义域”和“递归函数的值域”这三个概念是等价的。

§2 多种类递归算法概念

在文献〔1〕中，我們依 Kleene[5] 称一等式系統 Γ 的末一等式的左方第一个字母（这个字母是一函数字母）为主函数字母。在技术上，这样引进主函数字母的概念有一定的不方便之处。为了理論发展的便利，我們把递归算法的定义作以下的改变。設 Γ 是一等式的有穷序列，f_i 是 n_i 元函数字母，$i=1,\cdots,m$；又設任何$\mathscr{A}$中字 $a_1,\cdots,a_{n_i}$，如果有一个据 Γ 对 $f_i\langle a_1\,\mathbf{7}\cdots\mathbf{7}\,a_{n_i}\rangle$ 的計算，則計算結果总是唯一的。則我們称这样一个 Γ 連同 $\boldsymbol{R}_1,\boldsymbol{R}_2$ 为一$\mathscr{A}$中的递归算法，特别是一以 $f_1,\cdots,f_m$ 为主函数字母的递归算法。递归函数的定义可以和原来的相同，或者把定义中的“Γ 的主函数字母 f”改为“Γ 的主函数字母中有 f”，借以表示 Γ 的主函数字母可以不止 f 一个。这样改变有两点方便。第一，Γ 的主函数字母不一定要是末一等式的第一个字母，在具体书写递归算法时可以比較方便。第二，一个递归算法可以不止定义一个递归函数，而是可以定义一个有穷序列的递归函数。事实上，任何有穷个在同一字母表上的递归函数 $f_1,\cdots,f_m$，可以有一个递归算法 Γ 定义它們。如果 Γ_i 定义 $f_i,i=$

$1,\cdots,m$，則于 $i\neq k$ 时，可以使 Γ_i 与Γ_k没有共同的函数字母。令

$$\Gamma=\Gamma_1,\ \cdots,\ \Gamma_m$$

則Γ就同时定义 $f_1,\ \cdots,\ f_m$。

为了容易讲得清楚，下面在引进多种类递归算法概念时我們分作两步。

第一步

按§1中所规定，$\mathfrak{A}$为$\mathscr{A}$中所有字的集。

設

$$\mathfrak{A}_i\subset\mathfrak{A}\qquad i=1,\ \cdots,\ m$$

$$\alpha=(\mathfrak{A}_1,\ \cdots,\ \mathfrak{A}_m)$$

即α是由$\mathfrak{A}_1,\ \cdots,\mathfrak{A}_m$构成的有序$m$元組。把变元字母分为$m$类，第$i$（$i=1,\ \cdots,m$）类中的变元字母記作$x_i$。仿文献[1]中的$R_1$，$R_2$我們定义下面的两条推理規則：

R_α 設e_1是一等式，x_i是一在e_1中出現的变元字母，$1\leq i\leq m$，$a_i\epsilon\mathfrak{A}_i$；又設$e_2$是由$e_1$把在其中出現的$x_i$代入$a_i$而得。我們說$e_2$是由$e_1$經施行$R_1^\alpha$而得。

R_2^α 同R_2。

現在我們仿文献[1]中“計算”和“递归算法”的定义来定义“α-計算”和“α-递归算法”。

設Γ是一等式的有穷序列，f是一n元函数字母。令

1) $$e_1,\ \cdots,\ e_r$$

是一等式序列，其中每一e_i，$i=1,\cdots,r$，或者是一Γ中的等式，或者是从1）中e_i之前的某一e_j，$j<i$，經R_1^α而得，或者是从1）中e_i之前的$e_j,e_k,j,k<i$，經R_2^α而得，而e_r具有

2) $$f\langle a_1 ⁊ \cdots ⁊ a_n\rangle=a$$

的形式。这样，称1）为一个α-計算，或一个Γ中的α-計算，特别是一个Γ中的对于$\langle a_1 ⁊ \cdots ⁊ a_n\rangle$的$\alpha$-計算，計算的結果为$a$。我們也說从$\Gamma$可以$\alpha$-推演出2）。設$\Gamma$滿足这样的条件：如果存在着一个在$\Gamma$中的对于$f\langle a_1 ⁊ \cdots ⁊ a_n\rangle$的$\alpha$-計算，則这$\alpha$-計算的結果是唯一的；这样，我們称$\Gamma$，連同$R_1^\alpha$和$R_2^\alpha$，为一以$f$为主函数字母的$\mathscr{A}$中的$\alpha$-递归算法。設$\{\mathfrak{A}_1,\ \cdots,\ \mathfrak{A}_m\}$的基数是$k$，即$\{\mathfrak{A}_1,\ \cdots,\ \mathfrak{A}_m\}$中有$k$个互异的集，則也称$\Gamma$为一$k$种种类递归算法。在一$\alpha$-递归算法中，每一变元$x_i$与一$\alpha$中的$\mathfrak{A}_i$相配，此时我們称$x_i$是$\mathfrak{A}_i$中的变元。这里的“$x_i$是$\mathfrak{A}_i$中的变元”是一个数学概念，不能与通常的直观的“变元”概念相混淆。如果怕引起混淆，我們尽可以說“x_i是$\mathfrak{A}_i$中的形式变元”。

設

$$f(x_1,\cdots,x_n)=y$$

是定义于$\mathfrak{A}^n$中的n元函数，$y\epsilon\mathfrak{A}$。又設

$$\mathfrak{A}_i\subset\mathfrak{A}\qquad i=1,\cdots,m$$

$$\alpha=(\mathfrak{A}_1,\cdots,\mathfrak{A}_m)$$

Γ是一有穷的等式序列，f是一n元函数字母。又設

$$f(a_1,\cdots,a_n)=a$$

当且仅当，有一Γ中的对于$f\langle a_1 ⁊ \cdots ⁊ a_n\rangle$的$\alpha$-計算，計算的結果为$a$；設$f(a_1,\cdots,a_n)$无定义，則对于任何$b\epsilon\mathfrak{A}$，不存在$\Gamma$中的对$f\langle a_1 ⁊ \cdots ⁊ a_n\rangle$的$\alpha$-計算，其結果为$b$。这样一个$n$元函数$f$称为一个$\alpha$-递归函数，并說$\Gamma$定义$f$，或說$\Gamma$ α-定义f。我們也說Γ中

的主函数字母 f 表示 f。为了表明一个 α-递归算法，α-递归函数所涉及的字母表是$\mathscr{A}$，我們也如“$\mathscr{A}$中的递归算法”，“$\mathscr{A}$中的递归函数”一样，在用語中加上“$\mathscr{A}$中的”，例如說“$\mathscr{A}$中的 α-递归算法”，“$\mathscr{A}$中的 α-递归函数”等。

显然，上面所引进的概念与文献〔1〕中相应概念的不同之处，仅仅在于把原来的 $\boldsymbol{R}_1$ 改为 $\boldsymbol{R}_1^\alpha$，并在原来的有关名詞之前添上“α-”字样而已。

显然有这样的事实，設

$$\alpha=(\mathfrak{A}_1,\cdots,\mathfrak{A}_m)=(\mathfrak{A},\cdots,\mathfrak{A})$$

则 Γ 是一 α-递归算法，Γ α-定义 f，当且仅当，Γ 是一递归算法，Γ 定义 f。可見，原来文献〔1〕中的递归算法、递归函数是这里引进的 α-递归算法、α-递归函数的显然的特例。

关于多种类递归算法和递归算法之間的关系，有以下的主要定理(将在 §5 中証明)。

定理 1 設 Γ 是字母表$\mathscr{A}$中的 α-递归算法，又設

$$\alpha=(\mathfrak{A}_1,\cdots,\mathfrak{A}_m)$$

式中各 $\mathfrak{A}_i$，$i=1,\cdots,m$ 是$\mathscr{A}$中的递归可枚举集，则由 Γ 可以能行地找到一个$\mathscr{A}$中的递归算法 Γ′，使得 Γ′ 与 Γ 定义相同的递归函数。

第二步

上面讲到的 α-递归算法中的

$$\alpha=(\mathfrak{A}_1,\cdots,\mathfrak{A}_m)$$

中的 $\mathfrak{A}_i$，$i=1,\cdots,m$，是 $\mathfrak{A}$ 的子集：

$$\mathfrak{A}_i\subset\mathfrak{A}$$

現在推广为

$$\mathfrak{A}_i\subset\mathfrak{A}^{m_i}=\underbrace{\mathfrak{A}\times\cdots\times\mathfrak{A}}_{m_i}$$

原来对应于 α 中 $\mathfrak{A}_1,\cdots,\mathfrak{A}_m$ 的变元字母为 $x_1,\cdots,x_m$。現在对应于 α 中 $\mathfrak{A}_1,\cdots,\mathfrak{A}_m$ 的变元字母改为 m 組

$$x_{11},\cdots,x_{1n_1},\cdots,\cdots,x_{m1},\cdots,x_{mn_m}$$

其中第 i 組，即

$$x_{i1},\cdots,x_{in_i}$$

对应于 $\mathfrak{A}_i$。这样的对应是为了陈述 $\boldsymbol{R}_1^\alpha$。現在将 $\boldsymbol{R}_1^\alpha$，$\boldsymbol{R}_2^\alpha$ 陈述如下：

$\boldsymbol{R}_1^\alpha$ 設 e_1 是一等式，其中出現了第 i 組

$$x_{i1},\cdots,x_{in_i}$$

中的变元字母，設

$$(a_{i1},\cdots,a_{in_i})\in\mathfrak{A}_i$$

又設 e_2 是由 e_1 把在其中出現的 x_{ij}，$j=1,\cdots,n_i$，代入 a_{ij} 而得。我們說 e_2 是由 e_1 經施行 $\boldsymbol{R}_1^\alpha$ 而得。

$\boldsymbol{R}_2^\alpha$ 同 $\boldsymbol{R}_2$。

$\boldsymbol{R}_1^\alpha$ 經过如上的扩充，其它概念如 α-計算、α-递归算法作相应的改变，如何改变是显然的，不贅述。我們也說 $(x_{i1},\cdots,x_{in_i})$ 是 $\mathfrak{A}_i$ 中的变元。

主要定理的陈述如旧，只是其中 α 所表示的已有所推广。

§3 递归算法和多种类递归算法的一种等价的推理关系

設給定了一个字母表$\mathscr{A}$上的递归算法Γ，有一等式的序列

1) $$e_1, \cdots, e_m$$

滿足以下条件：1）中每一e_i，$i=1,\cdots,m$，或者是Γ中的一項，或者是由1）中在其前面的等式經$\boldsymbol{R}_1$而得，或者是由1）中在其前面的等式經$\boldsymbol{R}_2$而得。我們称这样一个1）为以Γ为（形式）**前提的推演**。显然，所有的計算都是推演。那种推演1），其末一項e_m有形式

$$f\langle a_1 \urcorner \cdots \urcorner a_n\rangle = a$$

其中f是一n元函数字母，$a_1,\cdots,a_n$，a都是$\mathscr{A}$中字，則是計算。这里的推演概念是計算概念的推广。設e是一个以Γ为前提的推演的末一項，則称e可以从Γ（形式地）推演出，記作

2) $$\Gamma \vdash e$$

2）的写法符合Kleene[5]的写法。在〔1〕中使用了这个写法，但沒有对它作如上的确切的規定。如果e可以从Γ（形式地）α-推演出，則記作

$$\Gamma \vdash_{\overline{\alpha}} e$$

一个具有形式

$$f\langle a_1 \urcorner \cdots \urcorner a_n\rangle = a$$

的等式，其中f是一n元函数字母，$a_1,\cdots,a_n$，a都是$\mathscr{A}$中字，称为$\mathscr{A}$中的結果等式。如果要表示一个結果等式中的函数字母在Γ中出現，我們也称之为Γ中的結果等式。我們称e是一个$\mathscr{A}$中也是Γ中的結果等式，于不至于引起誤会时，"$\mathscr{A}$中"，"Γ中"等字样可以省略。結果等式中等号左边的項称为簡单函数形式常項。显然，对于一个給定的字母表而言，一个等式e是一个結果等式，当且仅当，e中等号左边是一个簡单函数形式常項，右边是一个$\mathscr{A}$中字。

对照$\boldsymbol{R}_1, \boldsymbol{R}_2$，我們定义以下$\boldsymbol{R}_1^*, \boldsymbol{R}_2^*$两条推理規則：

$\boldsymbol{R}_1^*$ 設e_1是一等式，e_2是由e_1把其中所有出現的变元字母都替換之以$\mathscr{A}$中字而得（如果e_1中沒有变元字母，則e_2就是e_1）的結果等式，則称e_2为由e_1經施行$\boldsymbol{R}_1^*$而得。

$\boldsymbol{R}_2^*$ 設e_1是一具有形式

（1） $$f\langle a_1 \urcorner \cdots \urcorner a_n\rangle = c$$

的等式，$a_1,\cdots,a_n$是$\mathscr{A}$中字，c是一无变元字母的項。又設

（2） $$f_i\langle b_{i1} \urcorner \cdots \urcorner b_{in_i}\rangle = b_i \quad i=1,\cdots,m$$

都是結果等式，其中$f_1\langle b_{11} \urcorner \cdots \urcorner b_{1n_1}\rangle$是$c$中从左起最里面的簡单函数形式常項，$c_1$是由$c$把这个$f_1\langle b_{11} \urcorner \cdots \urcorner b_{1n_1}\rangle$替換之以$b_1$而得。設$c_1,\cdots,c_i\,(i<m)$已經得出，$c_i+1$是由$c_i$把$c_i$中从左起最里面的簡单函数形式常項$f_{i+1}\langle b_{(i+1)1} \urcorner \cdots \urcorner b_{(i+1)n_{i+1}}\rangle$替換之以$b_{i+1}$而得，而$b_m$即是$c_m$，設为$a$，是$\mathscr{A}$中字。我們称結果等式

（3） $$f\langle a_1 \urcorner \cdots \urcorner a_n\rangle = a$$

为由（1）和（2）中各等式經施行$\boldsymbol{R}_2^*$而得，并称（1）为大前提，（2）中各等式为小前提，（3）为結論。