数学完备吗?

叶卓杨[1]*

摘要

本文为数理逻辑课 [1] 的综述性总结。其中包含：数理逻辑的历史、一阶逻辑与一阶逻辑的完备性，局限性、ZFC 公理体系，Register-machine 模型、停机问题、哥德尔不完备性定理。本文尝试用尽可能通俗的语言介绍定理，定理的意义，以及定理的证明思路，略去繁琐的细节。希望能对读者有所启迪。

关键词

逻辑　证明　完备性

[1] 上海交通大学致远学院　[2] 致远学院科学与技术协会　* 邮箱：yezhuoyang@sjtu.edu.cn

1. 历史 [2]

17 世纪的德国数学家莱布尼兹一直有一个梦想：用一个统一的符号系统，以及一套操纵这些符号的方法来记录人的知识与思想。他所发明的微积分符号就一直被人们沿用至今。在一封给友人的信中他提到：

代数部分的秘密就在于文字，也就是说在于恰当地使用符号表达式地技艺。

莱布尼兹认为宇宙的演化是被上帝决定的，所有规律都可以被符号演算所反映。一个多世纪后，英国数学家布尔第一次用数学符号表示逻辑运算，创立布尔代数。

德国数学家弗雷格对数学秉持逻辑主义的观点，他认为算数，微积分，乃至一切数学都可与看作是逻辑的分支。他用毕生精力如果将逻辑方法引入算数基础，为此撰写了两卷关于算数基础的巨著。然而即将发表这部作品之前，他收到罗素的来信，毕生经历毁于一旦：他的逻辑体系的前提会导致矛盾。这个矛盾就是著名的罗素悖论。

20 世纪有一群被称为直觉主义的数学家，包括克罗内克，布劳威尔在内。他们否认数学的非构造性证明。比如：一个无限的集合中排中律是否正确?

为了捍卫数学证明的合理性，反驳直觉主义的观点，希尔伯特提出了元数学纲领：一致性有待证明的公理将被包含在一个形式逻辑系统之内，而证明仅仅是有限数目的符号的一种排列而已。只要证明形式逻辑是一致的，由于形式逻辑依赖的正是包含排中律的经典逻辑，数学证明一定是合理的。他在数学家大会上将其列为 20 世纪的重要数学难题之一。

然而令人震惊的是，一位年轻的数学家哥德尔迅速把这条纲领送入坟墓：哥德尔证明了形式化数学公理的一致性不可被证明，希尔伯特的纲领是不可能完成的。哥德尔不完备性定理彻底颠覆了人们对数理逻辑的认识。

后文将在以下几个方面进行介绍性论述：数学公理是如何一步一步形式化的? 哥德尔定理的具体内容，以及形式化后的数学公理最后是怎样自掘坟墓。

2. 一阶逻辑与完备性定理 [3]

不妨以群为例：为了刻画群这个数学结构，我们首先必须定义群的乘法运算 $\circ$，群的单位元 e。某个群中的所有元素必须满足 1. 结合律。2. 具有可逆元。两个性质。这两条性质又可以叫做群公理。群的所有性质都可以由这两条公理出发，以严格的数学推理得到。

引入大家早已熟知的关系符号：全称量词，析取合取量词群的几条公理用符号可以写作

$\forall v_0 \exists v_1 v_0 \circ v_1 \equiv e \ \forall v_0 \forall v_1 \forall v_2 (v_0 \circ v_1) \circ v_2 \equiv v_0 \circ (v_1 \circ v_2)$ 现在的问题是

1. 上述一组符号语句是否完整地刻画了群的结构?
2. 上述一组符号语句是否能包含比群结构更多的内涵?

为了回答这两个问题，一阶逻辑被划分为语义，语法两个层面。语义上讲：如果我把 $\circ$ 这个符号解释成群的乘法，把符号解释成构成群的集合中的元素，e 解释成群中的单位元，那么上述符号语句就是我们熟知的自然语言的另一种表述，它自然能刻画群的结构。从语法上讲，为了分析这个语句的内涵，势必要建立一套符合逻辑的语法规则，如果在该语

法规则下所有由公理出发的结论都与群的性质相符，那么第二个问题的回答就是肯定的。

语义上，一阶逻辑对一个数学结构的刻画分为以下几个部分

1. 一套语法规则，什么样的项 (s-term) 或表达式 (s-formula) 符合要求。
2. 一套推理规则
3. 一组公理语句
4. 一个 S-结构 (s-structure), 规定了变量所在的数域，如何解释所有的函数符号，关系符号，常数符号。
5. 对没有全全称量词限定的变量（自由变量）在域上赋值。
6. 一个对 S-结构的解释。

一套公式可能存在多种解释，我们称正确的解释为 model。

2.1 一阶谓词的推理系统

一阶逻辑的一个重要意义是将传统自然语言的数学证明转化为能用机器完成的对字符串的操作。比如利用矛盾进行反证的规则可以表述为：如果一个公理系统在某个条件不成立的时候既能推出另一个命题的肯定，也能推出另一个命题的否定，那么该条件一定不成立。推理系统里，这条规则写为：

$$\Gamma \quad \neg\phi \quad \psi \tag{1}$$

$$\Gamma \quad \neg\phi \quad \neg\psi \tag{2}$$

$$\frac{\Gamma \qquad\qquad}{\Gamma \qquad\quad \phi} \tag{3}$$

类似的规则还有很多条。当我们说一条推理规则是正确的时候，意思是对它进行任意的可能的解释，在该解释下只要前提成立，那么推理所得的结论成立。所以推理规则（语法）独立于解释（语义），反映的是推理过程的结构。我们把语义上的推理符号记为 $\models$, 一阶逻辑的推理符号记为 $\vdash$, 一组公理命题集合 Φ 能通过数学推理得到的公式集合记为 $\Phi^{\vdash}$，能用机器进行一阶逻辑运算推理得到的公式集合记为 $\Phi^{\vdash}$。我们把机器对公式进行推理的过程称 sequent calculus。

2.2 语义证明 = 语法证明吗? $\Phi^{\vdash} = \Phi^{\vDash}$?

所有推理系统的推理规则必定保证过程的正确性 (correctness)，但是反之是否所有正确的推理都能由推理规则导得呢？答案是肯定的，但是并不显然，需要由一阶逻辑的完备性保证。引入以下两个定义

1. 一套公式集合是自洽的 (consistent) 当且仅当存在它无法通过推理系统证明的命题。
2. 一套公式是可被满足的 (sastisfiable) 当且仅当存在某个解释，该解释下公式集合的所有命题是正确的。

很容易证明，只要所有自洽的公式集合一定可被满足，那么所有一阶逻辑中能被数学证明的命题一定可以用机器的 sequent calculus 一步一步证明。所以我们只需要对任意一组公自洽的公式集合找到一个正确的解释即可。教材中给出了一个最自然的“项模型”解释，该解释把变量集合组成的所有项划分为若干等价类，同一等价类对应同一个解释。两个项属于同一等价类当且仅当机器能够推出它们相等。

有一件事情值得注意：$\neg\Phi \vdash \psi$ 并不一定意味着 $\Phi \vdash \neg\psi$。只有性质很强的，自洽的 Φ 才能满足任给一个语句，要么 Sequent calculus 能推出它的肯定，不然就一定能推出它的否定，我们称这样 Φ negation complete。

可以证明，当公式集合 negation complete, 且只要存在 $\exists x\phi$ 这样的语句，就一定能找到某个项对应这个 x（contain witness), 那么在项模型下这组公式成立，也即这组公式可被满足。通过适当扩张公式集合，引入额外的常数符号，可以让任意公式集合满足以上两个要求，最终证明一阶逻辑的完备性[1]。

一阶逻辑的完备性告诉我们 $\Phi^{\vdash} = \Phi^{\vDash}$，所以对一阶逻辑下数学证明的研究等效于在一阶逻辑下 sequent-calculus 的研究，数学证明的局限性即 sequent-calculus 的局限性。

2.3 一阶逻辑的局限性

并不是所有数学结构都能用有限的一阶逻辑公式描述。比如现在要用逻辑公式刻画一个群，满足所有群元素的阶 (order) 必须是有限整数。由于这个限制条件太宽泛了，只能把公式写成 $\forall x(x \equiv e \vee x \circ x \equiv e \vee (x \circ x) \circ x \equiv e \vee ...)$。二阶逻辑可以描述这样的

[1] 扩张的过程需要用到集合论中的 Zorn's Lemma

涉及无穷多种性质的数学结构。比如著名的皮亚诺算数公理为

$$\begin{aligned}&\forall x\neg\sigma x\equiv 0\\&\forall x\forall y(\sigma x\equiv\sigma y\to x\equiv y)\\&\forall X((X0\wedge\forall x(Xx\to X\sigma x))\to\forall yXy)\end{aligned}\tag{4}$$

公理中 σ 符号是“后继”的意思，每个自然数都有唯一的后继。公理中的第三条本质上就是数学归纳，$\forall X$ 表示任意的一元关系。尽管归纳法符合我们对自然数结构的认知，但是” 任意一元关系” 这个量词无法用一阶逻辑表示，因为一阶逻辑中关系是有限可数的，且不可以加量词。

有几种方式可以将一阶逻辑扩充为二阶逻辑，比如在一个语句当中允许无穷多析取量词出现，或者定义一个新的量词 Q 表示“存在无穷多了”。或者把关系纳入域的范围，可以对关系符号加量词。

尽管二阶逻辑表达能力强于一阶逻辑，且形式更接近自然语言，但是二阶逻辑不存在一个完备的推理系统。

3. ZFC 公理系统-数学大厦的基础

虽然上文提到一阶逻辑在表述某些数学结构的时候会遇到困难, 对结构的刻画能力有限，但事实上所有数学都可以建立一阶公理的基础上，只不过语法更加复杂，且刻画能力不如高阶逻辑。由于一阶段逻辑只有一个固定的域，一个精妙的想法是把所有数学对象看作一个”数学对象域”(universe), 在这个域中包含比较简单的数学对象：一个数，空间中的一个点。也包含较为复杂的数学对象：数的集合，函数，结构，甚至是整个拓扑空间。

任何复杂的数学对象实际上都可以用集合来刻画，比如我们可以用归纳的方式定义自然数：0 是空集合，每一个后继元（successor）都是由前一个元素与一个额外的空集合组成的新集合。由于数本身就是集合，所有描述数与数的关系，数集与数集的关系都变为集合与集合的关系。又如任何的 n 元关系都可以表述为由 n 个元素组成的集合的集合。

基于以上论述，只要能完整地描述集合，那么所有数学结构，不管多么复杂，都可以从集合结构派生而来，确定了集合的基础就等于确定了数学的基础。

集合的公理系统就是著名的 ZFC 公理系统，最早由 Zermelo,Fraenkel,Skolem 提出。ZFC 公理的成员以及表述为：

1. EXT(the axiom of extensionality)：任意两个集合，只要所有元素（集合）要么同时出现在这两个集合中，要么同时不出现在两个集合中，那么这两个集合相等
2. SEP(separation axioms): 任意给一个集合 x 与一个性质 P，存在某个集合由 x 中所有满足性质 P 的元素组成。
3. PAIR(the pair set axiom)：任给一对元素 x,y, 存在一个集合 (pair)x,y 由两个元素组成
4. SUM(the sum set axiom): 任给一个集合，存在一个集合由该集合的所有子集的并集组成。
5. POW(the prower set axiom)：任给一个集合, 该集合的幂集（所有子集组成的集合）存在。
6. INF(the axiom of infinity)：存在一个无穷集合。
7. AC(the axiom of choice)：任给一个非空的，由不相交元素对组成的集合，存在一个集合恰好由所有元素对的其中之一个元素组成。
8. REP(replacement axioms)：如果对于一组确定的集合参数 $x_1,\cdots,x_n$ 一个公式 $\phi(x,y,x_1,\cdots,x_n)$ 确定一个由 $x\to y$ 的映射，那么任何一个集合在该映射下的像集也是一个集合。

集合与集合之间的关系可以归结为 $\in$ 关系，以上公理全部可以用一阶逻辑语言（记为 $L^{\in}$）描述。

更重要的一点：包含 ZFC 公理系统 ($L^{\in}$) 在内的所有一阶逻辑符号，语句，公式，以及一阶逻辑的推理关系也都可以用集合的语言描述，也就是说一阶逻辑可以刻画一阶逻辑自身！后文讲指出，正是一阶逻辑的这个特点导致了它的不完备性。

4. 停机问题的不可判定性

考虑这样一个命题：是否存在一个程序 P，P 能判断任意的程序是否在无输入的情况下最终停机? 这个问题是计算理论中著名的停机问题 (halting probrom)。停机问题是不可判定的，简要证明如下：如果停机问题可由 P 判定，任意给一个程序 W，我们可以简单修改 P，使得如果 W 不停机，P 陷入死循环，反之 P 停机输出某个结果。那么当 P 输入 P 自身时，如果 P 停机，那么 P 应该陷入死循环，如果 P 不停机，那么 P 将输出某个字符串，这是一个

明显的悖论，所以不存在这样的程序 P。后文将指出，哥德尔通过证明一阶逻辑中数学公理是否完备取决于停机问题是否可判定，得出一阶逻辑的不完备定理。

4.1 Register-machine

由丘奇-图灵定理，所有的程序都可以由对一种叫做 Register-mashine 的机器操作，以及相应的程序语言模拟，这个程序语言的语法结构有点像 Basic 语言。Register-mashine 由一组寄存器 $R_1, R_2 \cdots, R_n$ 组成，每个寄存器当中存储着一些从有限字母表 A 中选取的一些字符。如果第 L 条语句想要给某个寄存器中加入一个字符 a，程序语句写为

$$\text{L Let } R_i = R_i + a_j$$

类似的语句还有停机语句 Halt，打印某个寄存器中的字符语句 PRINT，转移语句 GOTO，条件语句 IF 等等。

在这里 Register-mashine 主要用来解决两个问题

1. 对于满足什么性质的可数字符串集 Φ，我们能写一个程序 P 将 Φ 中的元素按某个顺序一一枚举出来？(具有这样性质的 Φ 称为 R-enumerable)。由所以有限字符组成的集合一定满足这个性质，因为我们将所有字符串按其字典序排列然后以此输出。

2. 是否存在一个程序 P，P 能在有限时间内判定任意一个字符串 ψ 是否属于 Φ（具有这样性质的 Φ 称为 R-decidable）

5. 哥德尔不完备性定理

为了研究停机问题，哥德尔将所有程序进行编码。

把 Register machine 中存储的字符所在的字母表 A 扩充为

$$B := A \cup \{A, B, C, \cdots, X, Y, Z\} \cup$$
$$\{0, 1, \cdots, 8, 9\} \cup \{=, +, -, \square, \S\}$$

其中 $\square$ 代表空串，ξ 用来断句。比如程序

$$\begin{gathered} 0 \text{ LET } R_1 = R_1 - a_0 \\ 1 \text{ PRINT} \\ 2 \text{ HALT} \end{gathered}$$

可以在 B 的字符串集合 (记为 B^*) 中编码为

$$0\text{LETR1=R1-}a_0\S\text{PRINT}\S\text{2HALT}$$

如果上述编码在 B 的字典序中排第 n 个，又可将其一一映射为一个新的，更简便的字符串 $\xi_P = a_0 \cdots a_0 (a_0$ 出现 n 次)。这样的 ξ_P 就是程序 P 的哥德尔编码数。定义集合 $\Pi := \{\xi_P | P$是一个 A 上的程序$\}$。任意一个有限长度字符串是一个语法正确的程序 (即是否对应一个哥德尔编码) 可以通过暴力枚举判定。但是由于停机问题的不可判定性，可以构造一个不可判定的字符串集合 $\Pi'_{\text{halt}} = \{\xi_P | P$是 A 上的程序且 P 输入空串会停机

$$P : \square \to \text{Halt}\}$$

接下来我们回到一阶逻辑。有了 Π'_{halt} 一定不可判定的结论，哥德尔证明了在一阶逻辑下的永真，不含自由变元的命题也不可判定 ($\{\phi \in L_0^{S\infty} | \models \phi\}$)。证明的简要思路是对任意的程序 P，一定可以构造一组一阶逻辑公式完整的刻画程序运行的所有步骤，在这个基础上一定可以构造一个命题，这个命题的含义就是 P 会停机，假如这个命题可以被判定，那么 P 能不能停机也随之被判定，导致矛盾。

5.1 自然数理论不可由机器判定

目前超级计算机的运算能力已经达到大约每秒 10^{16} 次浮点数运算了，有了以上的铺垫，你会问这样的问题：随着计算能力的迅猛发展，既然所有数学的公理可以完全用一阶逻辑刻画，一阶逻辑又有完备性定理，有没有可能依靠计算机解决所有人类目前无法证明的数学猜想呢？

所有数学问题最终可以归结为自然数的问题。对于自然数上的算数，上文已给出了皮亚诺公理，那么我们问另一个问题：能否用机器判定任意一个命题是否能用皮亚诺公理证明？。

我们知道哥德巴赫猜想表述为：任意大于 2 的偶数都可以表示成两个素数之和。很容易将这个命题写为一阶逻辑的语句。如果以上问题的回答是肯定的，那么我们可以写一个程序，这个程序可以告诉我们哥德巴赫猜想到底能不能被证明出来。

为了更清晰的表述这个问题，先给出以下定义：

1. 一个属于 S-结构字符串集合 $T \subset L_0^S$ 被称为一个理论，当且仅当它可以被满足，且所有能被 T 证明的公式已经在 T 中。自然数结构的理论被称为算数 (arithmetic)，记为 Th($\Re$)。

2. 如果一个理论能简化为一系列能够被机器判定的语句的集合 Φ，称其可被 R-公理化 (R-axiomatizable)（Th($\Re$)=$\Phi^{\vdash}$）。

3. 如果对任意的命题，一个理论总能证明其肯定或否定，称其为完全理论 (complete)。这个定义类似于证明一阶逻辑完备性时的所要求的 Negation-complete。

可以证明如果一个理论完全且可被 R-公理化，或者一个理论可枚举且完全，那么所有该理论中的命题就可以被判定。

自然数理论是否可被判定的问题归结为：Th($\Re$) 是否可判定（R-decidable)？

然而答案是否定的。一定存在某些一阶逻辑下的自然数命题，机器即不能判定它是否可以被证明，也不能判定它是否不可以被证明，有点接近直觉主义的观点。

证明思路也是把问题转化为停机问题：对任意一个程序 P，构造一个自然数结构下的一阶逻辑语句，自然数结构下这个语句成立当且仅当这个程序停机。根据反证法，如果自然数理论可判定，那么所有程序是否停机也可以判定。证明中最大的难点是如何用自然数结构下的语句描述程序寄存器的某一个状态。哥德尔通过 β 函数解决了这个困难，这里略过。

所以我们得出结论：自然数理论的命题在一阶逻辑下不可判定。从另一个角度说，对于自然数，我们不可能找出一组有限的公理，使得我们可以设计出某个程序，在这个公理下把所有自然数结构中正确的命题一条一条罗列出来。皮亚诺公理一定不能完整地描述自然数，即 $\Phi_{\mathrm{PA}}^{\vdash} \subsetneq \mathrm{Th}(\Re)$。

哥德巴赫猜想能不能证明我们并不知道。

5.2 数学完备吗?

Wir müssen wissen, wir werden wissen. 我们终将知道，我们必将知道 —— **David Hilbert**

之前的所有不完备性，不可判定性的问题的证明都是基于停机问题不可判定。哥德尔最早的证明虽然思路有所不同，但是本质上殊途同归，都是基于“自引用命题”(Self-Referential Statements) 所导致的悖论。停机问题矛盾的根源在于它无法判定自己是否停机，撒谎者悖论“我现在不在说谎”这句话的矛盾也在于它描述的对象就是自己，后文也将指出：一阶逻辑不完备的根源在于“一阶逻辑是完备的”这个命题可以被一阶逻辑描述。

对任意一组命题 Φ 引入定义” 可被表示” (representable)

给定一组自然数 $n_0, \cdots, n_r$，一个自然数上的关系 $R(n_0, \cdots, n_r)$ 可被表示，当且仅当存在存在一个命题 $\phi_R(n_0, \cdots, n_r)$，如果 R 成立，则 $\Phi \vdash \phi_R$，反之 $\Phi \vdash \neg\phi_R$

任给一个自然数上的 r 元函数 F，F 可被表示，当且仅当存在一个一阶逻辑公式 $\phi_F(v_0, \cdots, v_r)$，使得任给自然数集 $n_0, \cdots, n_r$ 只要 $F(n_0, \cdots, n_{r-1}) = n_r$ 那么 $\Phi \vdash \phi_F(n_0, \cdots, n_{r-1}, n_r)$，反之 $\Phi \vdash \neg\phi_F(n_0, \cdots, n_{r-1}, n_r)$。

如果自然数结构上所有可被判定的关系和可被计算的函数都可被表示，称该公式集合 Φ 允许表示 (allows representations)

“允许表示”是一个非常强的性质。由一阶逻辑的完备性定理[2]可以证明 Th($\Re$) 允许表示。由于前文提到的 Register-mashine 的操作过程完全可以用皮亚诺算数公理下的一阶逻辑语句描述，“允许表示”范畴下只讨论 Register-machine 能够判断的关系和计算的函数，这个判断和计算的过程本质上就是在做逻辑推理，所以皮亚诺算数公理 $\Phi_{\mathrm{PA}}^{\vdash}$ 也具有“允许表示”的性质。

前文曾提到过哥德尔编码把任意一个程序映射为字典序，也就是到自然数的满射。用同样的方式也可以把任意自然数结构中一阶逻辑的命题 ϕ 映射为自然数 n^{ϕ}，所以能用自然数结构研究一阶逻辑命题本身。

哥德尔证明在所有“允许表示”的自然数一阶逻辑体系中的不动点定理：任给一个含有自由变元的语句 $\psi \in L_1^{\mathrm{Sar}}$，存在一个语句 ϕ，使得 $\Phi \vdash \phi \leftrightarrow \psi(n^{\phi})$。

这个 ψ 可以理解成语句是否具有某种性质，所以不动点定理告诉我们对于任何一阶逻辑语句的某个性质，都可以由 sequent-calculus 得到某个语句，这个语句将该性质等效于另一个语句。如果这个性质的内涵就是：我不可以被机器证明。那么就形成了悖论。

我们现在考虑这样一件事：对于任意一个语句 ϕ，ϕ 是否能用机器证明（$\phi \in \Phi^{\vdash}$?）这个一元关系在 Φ 中可表示吗？我们利用反证法，如果该性质可被某个语句 $\chi(v_0)$ 表示，根据” 可表示” 的定义，对任意的语句 α：

$\Phi \vdash \neg\chi(n^{\alpha})$ 当且仅当 $\Phi \vdash \alpha$ 不成立。但是由于不动点定理，$\neg\psi$ 就是“不可被机器证明”，一定有一个语句 ϕ 使得

$\Phi \vdash \neg\chi(n^{\phi})$ 当且仅当 $\Phi \vdash \phi$ 成立，构成矛盾。

[2]这个完备性定理值得是语义证明等价于语法证明，与后文的哥德尔不完备性定理不是一回事

由于 $\Phi^{\vdash}$ 这个性质不可被表示，可以得到哥德尔第一不完备性定理: 如果公式集 Φ 自洽，可判定，且允许表示，那么一定存在一个自然数结构上的语句 ϕ,$\Phi \vdash \phi$，$\Phi \vdash \neg\phi$ 都不成立。

接下来可以开始讨论数学本身是否完备这个终极目标了。上文曾指出：数学的根基是集合论，集合论的根基是 ZFC 公理系统，ZFC 公理系统可以用一阶逻辑表示，所以数学是否完备取决于一阶逻辑是否完备。到目前为止，一阶逻辑已经遇到了不可克服的困难：机器不可判定一阶逻辑下的永真命题与自然数理论，甚至对任意性质良好的公式集合，都一定存在某些不可证明也不可证伪的命题。接下来的哥德尔不完备性定理更可谓是对一阶逻辑的致命一击：我们甚至无法证明我们知不知道数学本身是自洽的体系。

一阶逻辑下一个基于皮亚诺算数的公理体系 Φ 是否自洽 (consistent) 取决于是否有它无法证明的命题。不妨取这个命题 ψ 为 $\neg 0 \equiv 0$，这个命题显然是伪命题，我们只要能证明 $\Phi \vdash \psi$ 不成立，那么就证明了 Φ 自洽。假设 Φ 是一个性质良好的公理系统，即能被机器判定又允许表示 (事实上 Th($\Re$),ZFC 都满足)。一定存在一个语句 $\mathrm{Der}(x)$，其含义是 x 对应的语句能够在 Φ 中由 sequent calculus 推得。

语句 $\mathrm{Consis}_{\Phi} := \neg\mathrm{Der}_{\Phi}(n^{\neg 0\equiv 0})$ 就是"Φ 是否自洽" 的表述。

由不动点定理，一定存在语句 ϕ 使得 $\Phi \vdash \phi \leftrightarrow \neg\mathrm{Der}(n^{\phi})$ 于是 Φ 自洽的另一个必要条件是 $\Phi \vdash \phi$ 不成立。可以证明 $\Phi \vdash \mathrm{Consis}_{\Phi} \rightarrow \neg\mathrm{Der}(n^{\phi})$

哥德尔第二不完备性定理：如果 Φ 自洽且可被判定，$\Phi_{\mathrm{PA}} \subset \Phi$，那么 $\Phi \vdash \mathrm{Consis}_{\Phi}$ 不成立

哥德尔第二不完备性定理的含义是：我们不能在一个数学理论的语境下证明一个数学理论是不是完备的。

上述证明基于皮亚诺公理，由于自然数可以用集合定义，可以把皮亚诺公理中的后继 σ 符号改写为 $\in$，ZFC 公理系统中亦可以完成两条不完备性定理的证明。

如果 ZFC 公理系统自洽，那么 $\mathrm{ZFC} \vdash \mathrm{Consis}_{\mathrm{ZFC}}$ 不成立。数学的基石竟然是不完备的！我们无法用数学方法证明数学的自洽性。

6. 结论

哥德尔不完备性定理说明从公理出发的数学是不完备的，我们无法用数学证明的方法证明数学证明不会导致矛盾。另一方面，哥德尔不完备性定理也说明机器无法取代人的智慧，很多未解数学难题依然需要靠人类艰苦卓绝的努力才有希望解决。

参考文献

[1] Teaching of yijia chen.

[2] 马丁戴维斯. 逻辑的引擎.

[3] W.Thomas. H.-D.Ebbinghaus, J.Flum. Mathematical logic,second edition.