011094，中国科学技术大学，2020年春季学期

数理逻辑讲义

陈小平
计算机科学与技术学院

图片来自朋友圈

3.6 可计算性

回顾：K_N可表示函数

❖ **定义1**（K_N可表示函数）一个k元函数g是K_N可表示的，如果存在一个含k+1个自由变元的K_N公式$p(x_1, \ldots, x_{k+1})$，使得对任意对$p(x_1, \ldots, x_{k+1})$中x_{k+1}自由的项u及$n_1, \ldots, n_k, n_{k+1} \in \mathbf{N}$有

1. 如果$g(n_1, \ldots, n_k) = n_{k+1}$则$\vdash_{K_N} p(\underline{n_1}, \ldots, \underline{n_k}, \underline{n_{k+1}})$；
2. 如果$g(n_1, \ldots, n_k) \neq n_{k+1}$则$\vdash_{K_N} \neg p(\underline{n_1}, \ldots, \underline{n_k}, \underline{n_{k+1}})$；
3. $\vdash_{K_N} p(\underline{n_1}, \ldots, \underline{n_k}, u) \rightarrow u = \underline{g(n_1, \ldots, n_k)}$.

❖ **注释** “大部分”数论函数不是K_N可表示的。但是，可计算的数论函数都是K_N可表示的。什么是可计算函数？

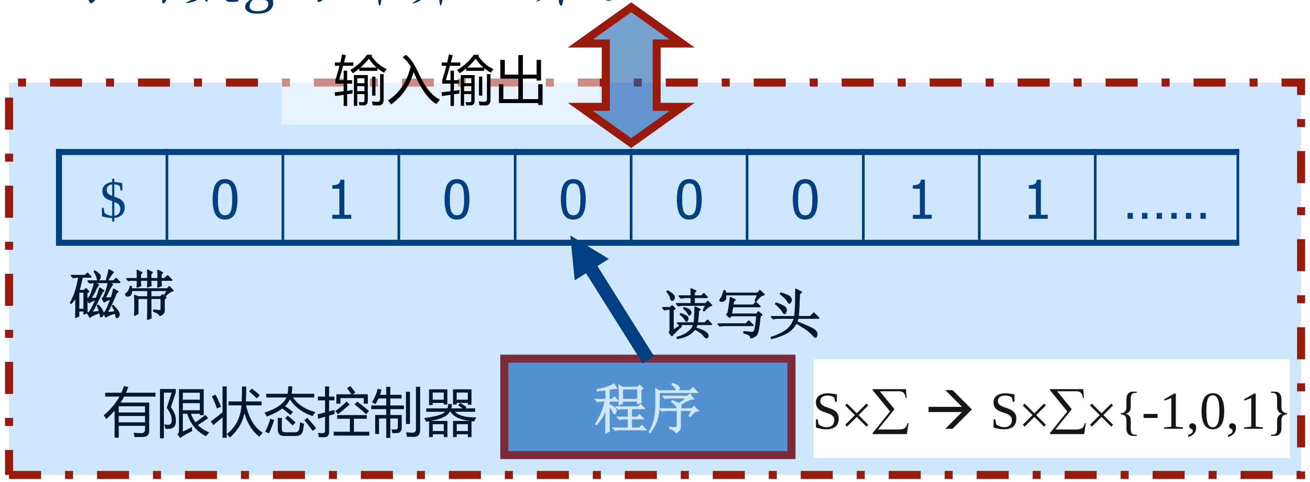
回顾：递归函数

- **定义5（递归函数）** 三个基本函数及由它们经有限次应用三个规则生成的函数称为(一般)递归函数。
- **基本函数** 零函数z，z(n) = 0；后继函数s，s(n) = n+1；投影函数p_i^k，$p_i^k(n_1, \ldots, n_k) = n_i, i=1, \ldots, k$。
- **规则** 复合规则、递归规则、μ算子。
- **观察** 三个基本函数是能行可计算的，三个规则的应用保持能行可计算性，所以一般递归函数是能行可计算的。
- **问题** K_N可表示函数、递归函数和可计算函数有什么关系？

3.6 可计算性

❖ 标准图灵机模型(1936)的运行

1. 待计算函数g自变量的值输入到磁带上；
2. 从初始状态运行事先存贮的计算g的程序；
3. 运行到终止状态时停机，磁带上的内容即为函数g的计算结果。



◆ 有限状态控制器/程序

根据：

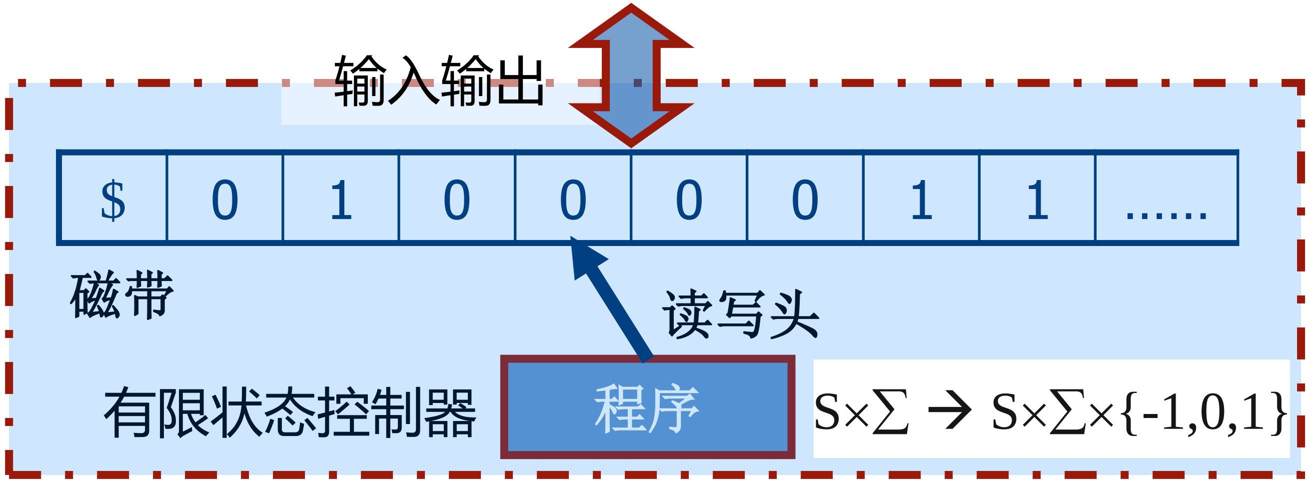
1. 当前状态；
2. 当前单元中的符号

决定：

1. 进入下一个状态(可以和当前状态相同)；
2. 改写当前单元中的符号(可与原符号相同)；
3. 向左或右移动一个单元或停留在原单元。

3.6 可计算性

❖ 例 计算阶乘函数g(m)=m!当m=8的值。采用二进制表示，将100输入到磁带上，前4个单元的内容为\$100，其余单元里的内容都是符号B（代表空白）。程序运行终止时磁带上内容是\$1001110110000000B……即为g(8)的值。



❖ 标准图灵机的等价模型
- 双向无限带图灵机；
- 多带和多维图灵机；
- 2符号图灵机；…

❖ 通用图灵机

3.6 可计算性

❖ 车赤-图灵论题(Church-Turing Thesis) 一个函数是可计算的，当且仅当该函数是图灵机可计算的。

❖ 记号 所有图灵机可计算的函数的集合记为TM；所有递归函数的集合记为REC；所有K_N可表示函数的集合记为REP。

❖ 定理 REC = REP。

◆ 证明 略。

❖ 定理 REC = TM。

◆ 证明 略。

❖ 推论 REC = REP = TM。

3.6 可计算性

- **相似性定理(洪加威)** 包括图灵机在内的12种计算模型相互等价，并且可在多项式时间内相互模拟。
- **注释** “可计算”是一个直观概念，无法证明它与图灵机的等价性/不等价性。经过长期研究，车赤-图灵机论题被大量等价性结果的证明和不成功计算模型的识别所确认——凡公认合理的计算模型都被证明与图灵机等价，凡不与图灵机等价的计算模型都被公认为不合理/不充分。因此，图灵机被国际学术界普遍接受为电子数字计算机的理论模型。

3.6 可计算性

❖ **哥德尔数/编码** 上述部分结果和哥德尔不完备性定理的证明需要通过哥德尔编码/哥德尔数，将K_N公式和公式序列映射为自然数。

1. K_N符号u的哥德尔数g(u)规定如下：

u	′	+	×	¬	→	∀	=	0	x_i (i=1,2, …)
g(u)	1	3	5	7	9	11	13	15	15+2i

g(u)将K_N的所有符号映射为奇自然数，且不同符号对应的自然数也不同。

3.6 可计算性

❖ 哥德尔数/编码

2. K_N符号串$u_0u_1\ldots u_k$的哥德尔数：

$$g(u_0u_1\ldots u_k) = 2^{g(u_0)}3^{g(u_1)}\ldots p_{k+1}{}^{g(u_k)}。$$

其中2、3、…、p_{k+1}是第1到k+1个素数。

注意：每个K_N符号的哥德尔数为奇数，而每个K_N符号串(默认不含空串)的哥德尔数为偶数，但幂指数都是奇数。

◆因此，任何一个K_N符号的哥德尔数与任何K_N符号串的哥德尔数是不同的。

3.6 可计算性

❖ 哥德尔数/编码

3. K_N符号串序列$S_0, S_1, \ldots, S_k$的哥德尔数：

$$g(S_0, S_1, \ldots, S_k) = 2^{g(S_0)}3^{g(S_1)}\ldots p_{k+1}{}^{g(S_k)}。$$

其中2、3、…、p_{k+1}是第1到k+1个素数。

注意：每个K_N符号串序列(默认不含空序列)的哥德尔数都是偶数，而且所有幂指数也都是偶数。

◆因此，任何K_N符号串的哥德尔数与任何K_N符号串序列的哥德尔数是不同的。

3.6 可计算性

❖ 观察

1. K_N符号、符号串和符号串序列三者的哥德尔数可以通过初等数论的计算加以区分。
2. 根据哥德尔编码，不同的K_N符号有不同的哥德尔数；根据素分解唯一性定理，不同的K_N符号串有不同的哥德尔数，不同的K_N符号串序列有不同的哥德尔数。
3. 未必每一个偶素数都是一个K_N符号串的哥德尔数；例如，假设$14=2^1 3^0 5^0 7^1$代表一个符号串$u_0u_1u_2u_3$，则其中符号u_1和u_2均无定义，所以14不可能是一个K_N符号串的哥德尔数。

3.6 可计算性

❖ **命题** 下列9个集合是递归的：

1. $\{g(u) \mid u$是K_N项$\}$；

◆ **注释** 记此集为A，则A的特征函数$C_A(x)$是递归函数，即对任何自然数x：(1)当x是一个K_N项u的哥德尔数g(u)时，$C_A(x)$的计算结果为1；(2)当x不是一个K_N项u的哥德尔数g(u)时，$C_A(x)$的计算结果为0。$C_A(x)$的递归性保证有限时间内完成计算。

◆ **证明** 用递归函数分解出x的幂指数并分析它们代表的符号串。

❖ **观察** $C_A(x)$是计算机编译技术诞生之前出现的、用递归函数编写的词法解析器。

3.6 可计算性

2. $\{g(u) \mid u$是K_N公式$\}$；/此集B的特征函数$C_B(x)$是递归函数/
3. $\{g(S) \mid S$是K_N公式序列$\}$；
4. $\{g(p) \mid p$是Ki公理$\}$，$i=1,2,3,4,5$；
5. $\{g(p) \mid p$是Ei公设$\}$，$i=1,2,3$；
6. $\{g(p) \mid p$是Ni公设$\}$，$i=1,2,3,4,5,6,7$；
7. $\{(n_1, n_2, n_3) \mid n_1=g(p), n_2=g(p \to q), n_3=g(q)\}$；
8. $\{(n_1, n_2) \mid n_1=g(p), n_2=g(\forall x\, p)\}$；
9. $\{(n_1, n_2) \mid n_1=g(p), n_2=g(S)$, S是$p$的一个$K_N$证明$\}$。