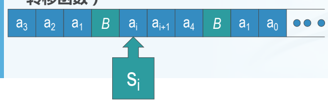
离散数学：形式语言与自动机：图灵机

陈斌 北京大学地球与空间科学学院 gischen@pku.edu.cn

图灵机基本结构

- 一条**分格**的**无限长**的**纸带**，每格可容纳一个字符
- 一个**读写头**，可以在纸带上**移动**，**读出**当前格子的字符，**重写**格子上的内容，改变自己的内部**状态**
- 一系列关于读写头动作的**规则**（程序、状态转移函数）



输入、计算和输出

- **纸带**上的内容（包括空白B）是计算前的**输入**和计算完成后的**输出**
- 计算从读写头的**初始状态**s_0、**初始位置**和**输入**的纸带格局开始
- 按照规则进行读写头的**移动**、纸带字符的**修改**，直到进入**停机**状态（$S_H/S_Y/S_N$）
- 在停机状态时纸带的内容就是输出

状态转移函数：计算规则

- 状态转移函数：$\delta:Q\times\Gamma\rightarrow Q\times\Gamma\times\{L, R, N\}$
- 每个五元组为一条规则：$q=<s_i,a_k,a_l,s_j,d>$
- $s_i,s_j\in S$（内部状态），$a_k,a_l\in A$（纸带字符），$d\in\{L,R,N\}$（左移，右移，不动）
- 表示如果读写头**当前状态**为s_i，读到**当前格子**字符为a_k，则：

① **改写**格子内容为a_l

② **转变**状态为s_j

③ **进行**d指定的动作（读写头左移、右移或不动）

计算规则的限制

- 首先，规则数量**有限**
- 其次，保证动作的**确定性**：任意两条规则前两项不能相同
- 最后，没有规则第一项是$S_H/S_Y/S_N$，保证这三个状态是**停机**状态

形式语言与图灵机

- 当$W\subseteq A^*$，将W中的任意字符串w作为输入，
- M都停止在**接受**状态S_Y
- 而A-W中的所有字符串，M停止在**拒绝**状态S_N，或者**永不停机**
- 称M**接受/识别**语言W=L(M)
- 形式语言L能被图灵机M识别，当且仅当L是一个0型形式语言

离散数学：形式语言与自动机：图灵机例子

陈斌 北京大学地球与空间科学学院 gischen@pku.edu.cn

图灵机例子：$\{a^m b^m | m \geq 0\}$

- 思路：将a和b一一对消，如果最后剩下空白则接受，否则拒绝
- <s_0, a, B, s_1, R>：初始碰到a，消去，s_1，右移
- <s_1, a, a, s_1, R>：消去1个a的状态，继续右移，找最后一个b
- <s_1, b, b, s_1, R>：继续右移
- <s_1, B, B, s_2, L>：右移到头状态s_2，回移
- <s_2, b, B, s_3, L>：如果有b，消去，进入左移状态s_3

图灵机例子：$\{a^m b^m | m \geq 0\}$

- <s_3, b, b, s_3, L>：左移
- <s_3, a, a, s_3, L>：左移
- <s_3, B, B, s_0, R>：左移到头变初始状态，右移看下个字符
- <s_0, B, B, s_Y, N>：a,b都能一一消完，则接受
- <s_0, b, b, s_N, R>：b多了，或者在a前，拒绝
- <s_2, a, a, s_N, R>：a多了，或者在b后，拒绝
- <s_2, B, B, s_N, R>：a多了，拒绝

图灵机仿真运行：aaabbb

图灵机仿真运行：aab

图灵机例子：加法

- 首先，表示自然数：
- 用n+1个连续的1表示自然数n
- 0＝1，1＝11，2＝111，3＝1111......
- 其次，表示二元参数：
- 两个输入的自然数之间用B隔开
- 0，2＝1B111
- f(m,n)=m+n：即字符串的连接操作
- f(m,n)=m*n：即将m重复copy n次

离散数学：形式语言与自动机：图灵机变种

陈斌 北京大学地球与空间科学学院 gischen@pku.edu.cn

设施扩展的图灵机

- 具有k条纸带的多带图灵机
- k条纸带，k个读写头
- 状态转移函数$\delta:Q\times\Gamma^k\rightarrow Q\times\Gamma^k\times\{L, R, N\}^k$
- 双向无界的图灵机
- 左边、右边都无限制增长

非确定性图灵机

- 计算规则的前两项可以相同，机器任意选择一个分支进行
- 但规则必须一致，禁止一个分支导致接受而另一个分支导致拒绝
- 每个上述的变种图灵机M，都存在单带图灵机M'，满足L(M)=L(M')
- 也就是，所有这些图灵机变种的计算能力都相同，都可以归结到等价的确定性单纸带单向无界图灵机

离散数学：形式语言与自动机：识别与判定

陈斌 北京大学地球与空间科学学院 gischen@pku.edu.cn

识别与枚举

- 识别：
- 属于语言的串在有限步被接受，不属于语言的串被拒绝或者永不停机
- 枚举：
- 枚举生成所有属于语言的串，对于无限集合语言，这个生成过程可能永不停止
- 可识别 = 可枚举

判定

- 判定：
- 属于语言的串被接受，且不属于语言的串被拒绝，不存在永不停机的可能。
- 只有L和L的补语言都是图灵可识别的情况下，L才是图灵可判定的。
- 存在语言是图灵机可识别但不可判定的
- 和停机问题相关

离散数学：形式语言与自动机：哥德尔编码

陈斌 北京大学地球与空间科学学院 gischen@pku.edu.cn

哥德尔编码Gödel Numbering

- 将任何形式语言L**编码**为自然数的集合
- 将语言上的运算**变换**为自然数的运算
- 形式系统的问题**变换**为数论问题
- 还记得命题演算、谓词演算？

哥德尔编码过程

- 首先定义函数p(n)=第n个素数（n>0）
- 我们知道：素数有无限个，
- 而且任何正整数都能唯一地写成素数的乘积
- $L \subseteq A^*$，$A=\{a_1,a_2,\ldots a_n\}$，任意 $w=a_{m1}a_{m2}..a_{mk} \in L$
- 其中mi是w中第i个字符在字母表里的序号

$$G(w)=\prod_{i=1}^{k} p(i)^{m_i}$$

哥德尔编码例子

- A={a,b,c,d}
- G('accdbdd')
- $=2^1\times3^3\times5^3\times7^4\times11^2\times13^4\times17^4$
- =4,677,894,230,215,956,750
- 重要结论：如果字母表A是**可数**的（可以对应自然数的子集），则A上的所有形式语言L都是**可数**的

离散数学：形式语言与自动机：通用图灵机

陈斌 北京大学地球与空间科学学院 gischen@pku.edu.cn

程序和通用计算机

- 每个程序执行特定的计算，实现特定的功能
- 但我们的计算机可以**执行**所有的程序，并得到相同的结果
- **通用计算机**是一种特别的算法程序
- 将**程序**和程序的**输入**作为其**输入**，输出程序的输出

通用图灵机

- 每个图灵机计算特定的函数，识别特定的语言
- 能否有一个通用图灵机，能模拟任意图灵机的执行，对相应的输入得到相应的结果呢？

对语言和机器进行编码

- 图灵机的编码
- 图灵机的定义包含有限字符集A、有限状态集S和读写头的运动规则
- 对图灵机的描述就是一个A∪S∪{L,R,N}上的字符串
- 将这个字符串进行哥德尔编码，得到图灵机的哥德尔数G(M)
- 输入、输出的编码
- 对图灵机的输入、输出是A上的字符串，也可以进行哥德尔编码

通用图灵机Universal Turing Machine

- 通用图灵机U存在
- 接受两个输入整数a, k
- a是图灵机的哥德尔数
- k是图灵机输入的哥德尔数
- 计算结果是M(a)在输入I(k)下的输出结果的哥德尔数G(o)
- U所计算的函数是U:N×N→N
- 有趣的是，U本身也可以进行哥德尔编码，作为输入参数在U中进行计算

离散数学：形式语言与自动机：停机问题

陈斌 北京大学地球与空间科学学院 gischen@pku.edu.cn

停机问题

- 给定一个算法和相应的输入，判定这个算法是否能够完成？（或进入死循环永不能完成？）
- 程序正确性证明，包括了给一段程序，判断这段程序是否会进入死循环
- 由于算法都可以用图灵机描述，所以问题转变为，给定一个图灵机和相应的输入，判定是否停机？
- 对于特定的算法，我们大概是可以判定的，但是否存在对一般的算法进行判定的方法？
- 注意：这个方法本身也应是一个算法，否则没有意义

图灵机停机问题

- 是否有算法能够判定某个图灵机M在输入I下是否停机？
- 即Halt(M,I)=True iif M在I处停机，False iif M在I处不停机
- 答案是否定的，不存在这样的算法
- 停机问题是不可计算问题

停机问题证明

- 假设存在这样的停机判定算法，它对应的图灵机是Halt(a,k)，a是要判定的图灵机编码，k是输入的编码
- 我们构造这么一个图灵机Trouble(a)：
- function Trouble(a)
- if **Halt(a, a)** = False then
-     return True
- else
-     Loop Forever

停机问题证明

- Trouble图灵机自然也可以编码，记为t，我们来看看把t作为Trouble输入会怎么样？
- Trouble(t)=? ......矛盾？矛盾！
- 如果Trouble(t)停机返回了
- 也就是Halt(t,t)=False，但这样是说Trouble(t)不停机，矛盾
- 如果Trouble(t)不停机：
- 也就是Halt(t,t)返回了True，但这样却是说Trouble(t)能停机，矛盾

停机问题的意义

- 消除矛盾的唯一途径就是否定假设，即**不存在**能一般性地判定图灵机停机的算法
- 人的**智能**就能解决停机问题么？至少目前还不能：
- 对于太长的算法，或者输入太复杂的算法，无法解决
- 对于一些短而简单的算法，也不能（一些数论难题，比如：寻找大于给定N的孪生素数）
- 启示：算法无法解决所有问题

离散数学：形式语言与自动机：哥德尔不完备定理

陈斌 北京大学地球与空间科学学院 gischen@pku.edu.cn

哥德尔不完备定理

- 任何包含自然数定义的形式系统都是不完全的，也就是存在不能证明为真也不能证明为假的命题
- 重点是**证明**，一个开始于公理，结束于此命题的公理-定理有限序列
- 也就是，系统可能存在一些命题，它们是定理，却无法给出证明序列
- 证明和计算
- 可证明 v.s. 判断计算停机
- 不可判定的命题 v.s. 不可解的停机问题

哥德尔不完备定理的证明

- 首先对形式系统进行哥德尔编码
- 将符号系统用自然数编号
- 所有的命题和谓词看成是符号系统上的**字符串**
- 可以得到所有命题和谓词的哥德尔数
- **命题公式序列**也进行编码：
- $GN=2^{GN1}*3^{GN2}*5^{GN3}$，其中有3个公式：
- Statement 1 (GN1)
- Statement 2 (GN2)
- Statement 3 (GN3)

哥德尔不完备定理的证明

- 这样： “**一个序列是一个命题的证明**”
- 我们可以写谓词R(v,x)，x是序列的哥德尔数，v是命题的哥德尔数
- 我们看公式：$\forall x \neg R(v,x)$：表示“**v不可证明**”
- 这个谓词公式对应的哥德尔数记做GN4
- 考察：$\forall x \neg R(GN4,x)$：表示“**本命题不可证明**”
- 这是一个命题，具有确定的真值，但是否可证明呢？
- 如果可证明，那么就是证明了一个假命题，系统不一致了
- 如果不可证明，那就是一个真命题，是一个定理，却不能证明，系统不完备

离散数学：形式语言与自动机：不可判定问题

陈斌 北京大学地球与空间科学学院 gischen@pku.edu.cn

两个不同领域的不可判定问题

- 数理逻辑 vs 计算理论
- 得到相似的结论：
- 没有一个“万能”的公理系统，在其中能够证明所有的数学真理，而排除谬误
- 没有一个“万能”的算法，能够判定所有的算法是可以完成，还是陷入无限循环
- 形式化的缺陷？人类思维的局限？
- 两个问题的共同点在于“自指、层次缠绕”
- 课程之初所提到的版画艺术及音乐

知识的边界？

- 人类推理和知识的极限：
- 只通过形式系统不能达到所有的真理？
- 终极理论的命运：是否有统一描述宇宙的理论？
- 人工智能的极限：机器的智能能否超越人类思维？
- 通过对复杂系统的研究，人们认识到层次的重要性
- 生命游戏：简单的规则，却导致复杂的行为
- 生命和自组织现象