停机问题与判定问题
(Halting Problem & Decision Problem)

熊 明
mingshone@163.com

South China Normal University

主要内容

1. 通用图灵机
2. 停机问题
3. 判定问题

主要内容

通用图灵机 (UTM)

- 在 1936 年那篇著名的论文中，图灵特别地给出了这样一个图灵机（程序）：
- 这个程序能模拟任何图灵机（程序）的任何执行情况！
- 这样的图灵机被认为“通用图灵机”（universal Turing machine）

6. *The universal computing machine.*

It is possible to invent a single machine which can be used to compute any computable sequence. If this machine $\mathfrak{U}$ is supplied with a tape on the beginning of which is written the S.D of some computing machine $\mathcal{M}$,

SER. 2. VOL. 42. NO. 2144. R

通用图灵机 (UTM)

- 在 1936 年那篇著名的论文中，图灵特别地给出了这样一个图灵机（程序）:
- 这个程序能模拟任何图灵机（程序）的任何执行情况!
- 这样的图灵机被认为"通用图灵机"（universal Turing machine）

6. *The universal computing machine.*

It is possible to invent a single machine which can be used to compute any computable sequence. If this machine $\mathfrak{U}$ is supplied with a tape on the beginning of which is written the S.D of some computing machine $\mathcal{M}$,

SER. 2. VOL. 42. NO. 2144. R

通用图灵机 (UTM)

- 在 1936 年那篇著名的论文中，图灵特别地给出了这样一个图灵机（程序）：
- 这个程序能模拟任何图灵机（程序）的任何执行情况！
- 这样的图灵机被认为"通用图灵机"(universal Turing machine)

6. *The universal computing machine.*

It is possible to invent a single machine which can be used to compute any computable sequence. If this machine $\mathfrak{U}$ is supplied with a tape on the beginning of which is written the S.D of some computing machine $\mathcal{M}$,

SER. 2. VOL. 42. NO. 2144. R

通用图灵机 (UTM)

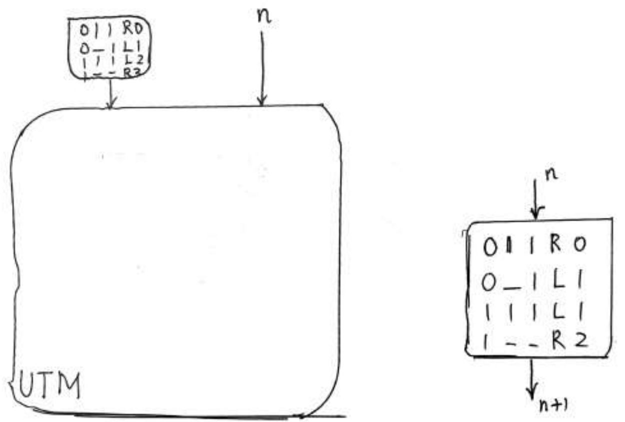
- 在 1936 年那篇著名的论文中，图灵特别地给出了这样一个图灵机（程序）：
- 这个程序能模拟任何图灵机（程序）的任何执行情况！
- 这样的图灵机被认为“通用图灵机”（universal Turing machine）

6. *The universal computing machine.*

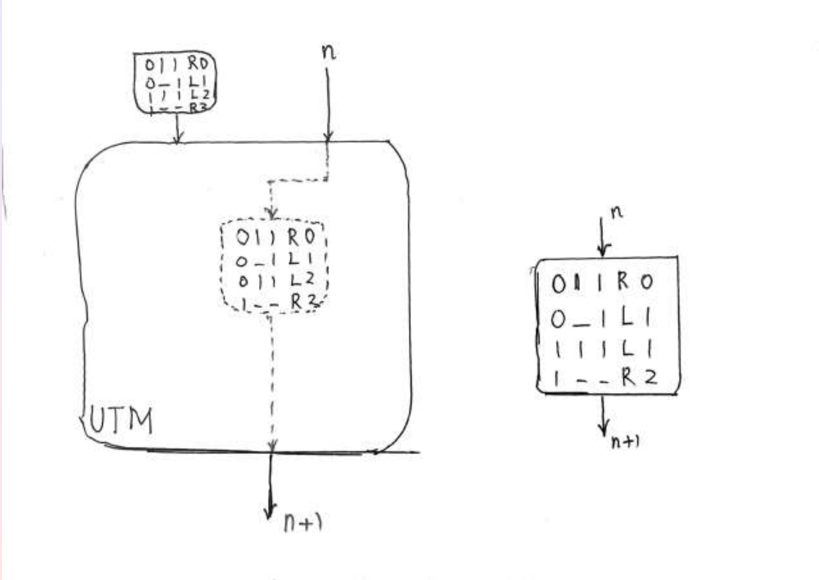
It is possible to invent a single machine which can be used to compute any computable sequence. If this machine $\mathfrak{U}$ is supplied with a tape on the beginning of which is written the S.D of some computing machine $\mathcal{M}$,

SER. 2. VOL. 42. NO. 2144. R

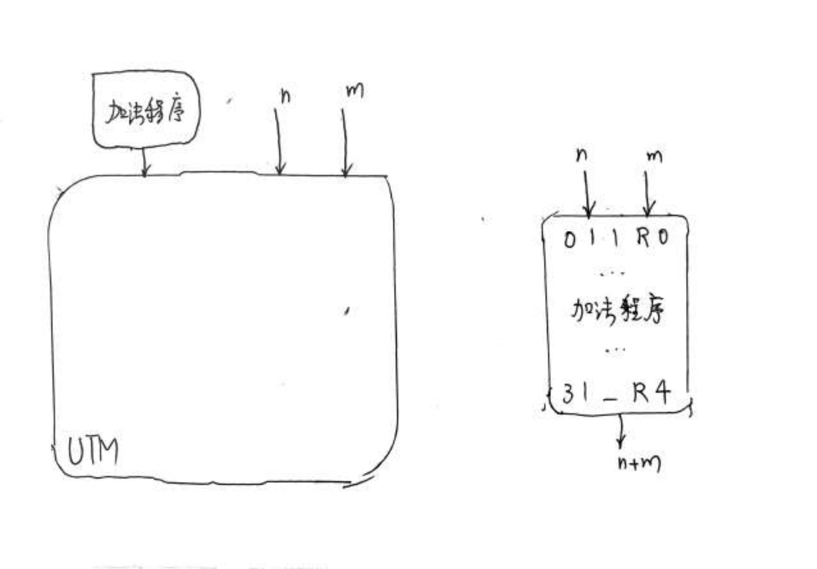
通用图灵机图示



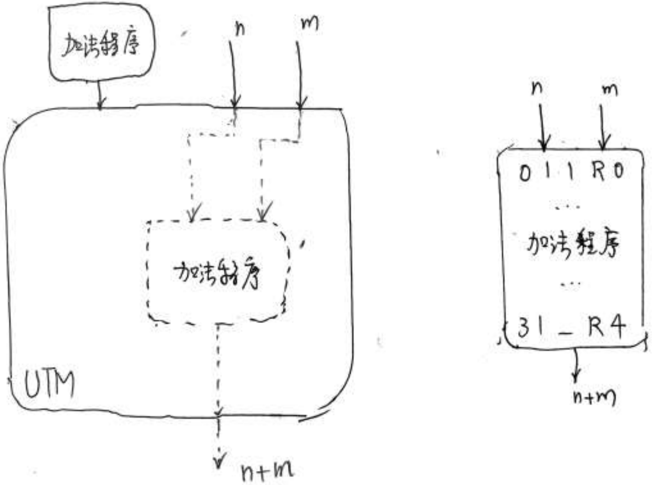
通用图灵机图示


0 1 1 R 0
0 _ 1 L 1
1 1 1 L 2
1 _ _ R 3
n
0 1 1 R 0
0 _ 1 L 1
0 1 1 L 2
1 _ _ R 3
UTM
n+1
n
0 1 1 R 0
0 _ 1 L 1
1 1 1 L 1
1 _ _ R 2
n+1

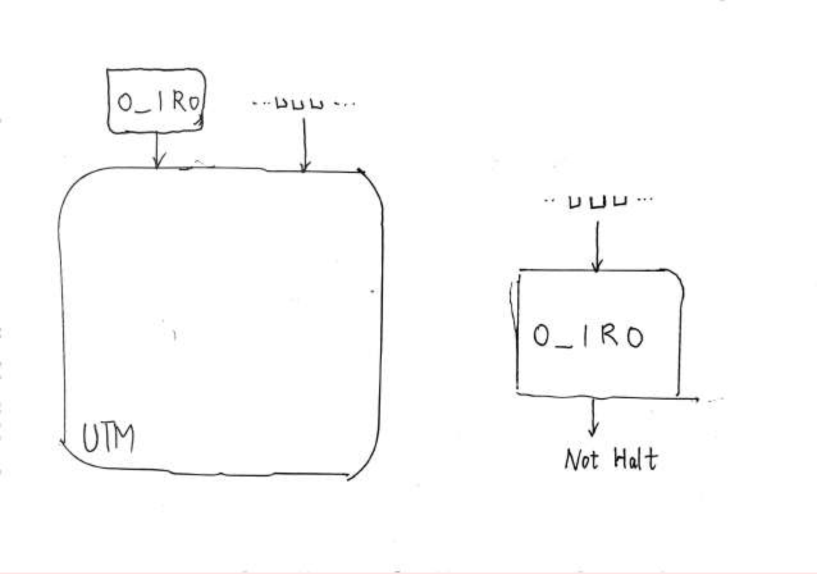
通用图灵机图示



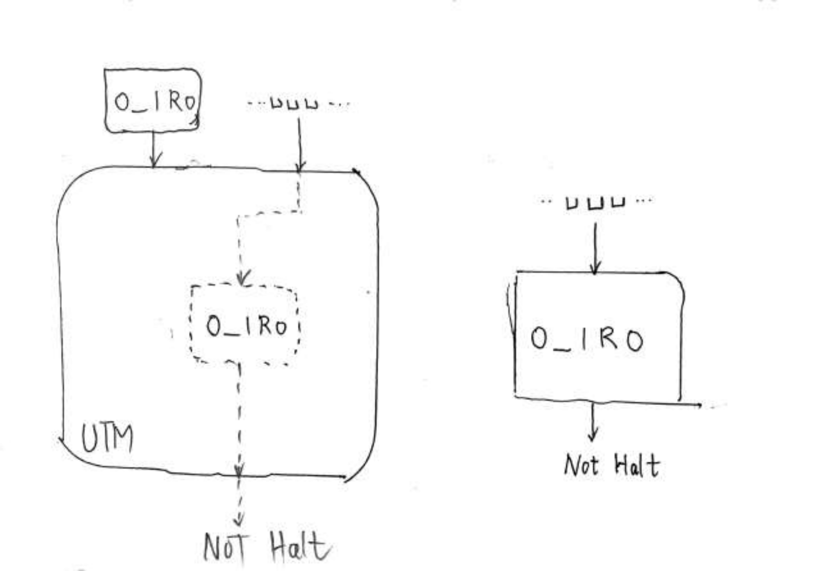
通用图灵机图示



通用图灵机图示



通用图灵机图示



通用图灵机的重要性

- 通用图灵机奠定了“存储程序电脑”的思想基础。
- 每个通用图灵机可看做是一个操作系统！

主要内容

1. 通用图灵机
2. 停机问题
3. 判定问题

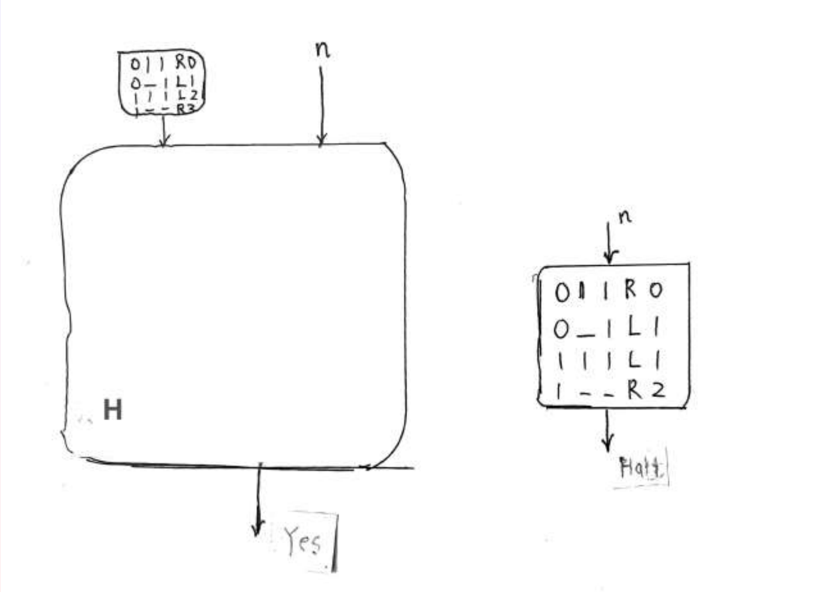
停机问题

是否存在一台图灵机，使用它可以判断任何一台图灵机在某个输入下停机与否?

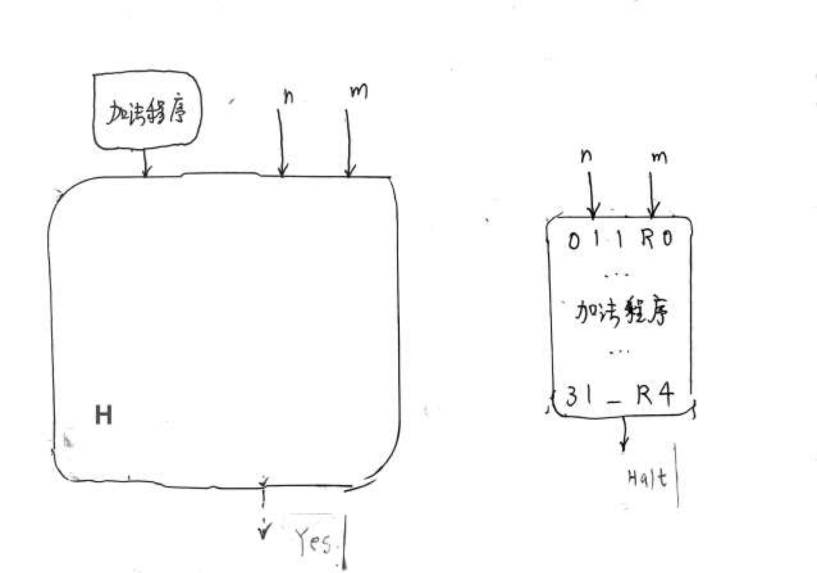
即：是否存在图灵机 H，对其输入图灵机（的程序或编码）x，输入为 y，

- 如果图灵机 x 对输入 y 停机，则 H 输出 Yes（或者 1）;
- 如果图灵机 x 对输入 y 不停机，则 H 输出 No（或者 0）。

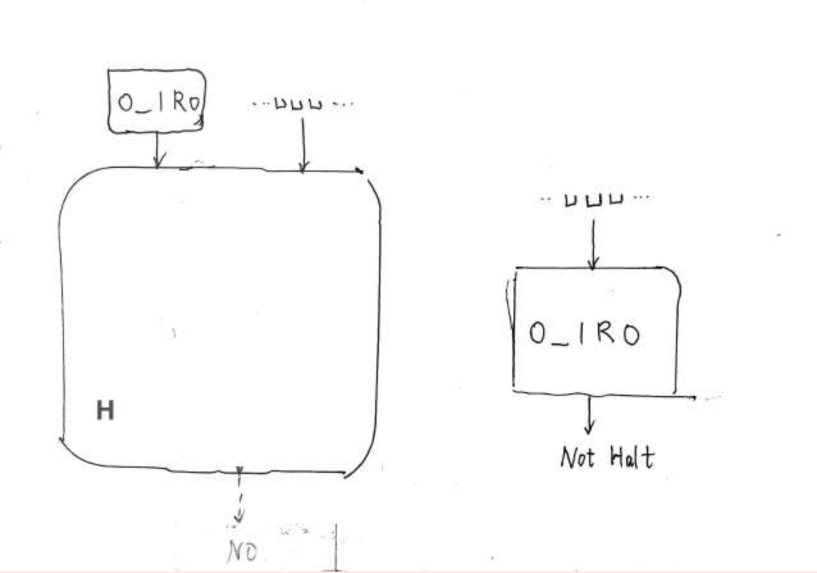
停机问题图示



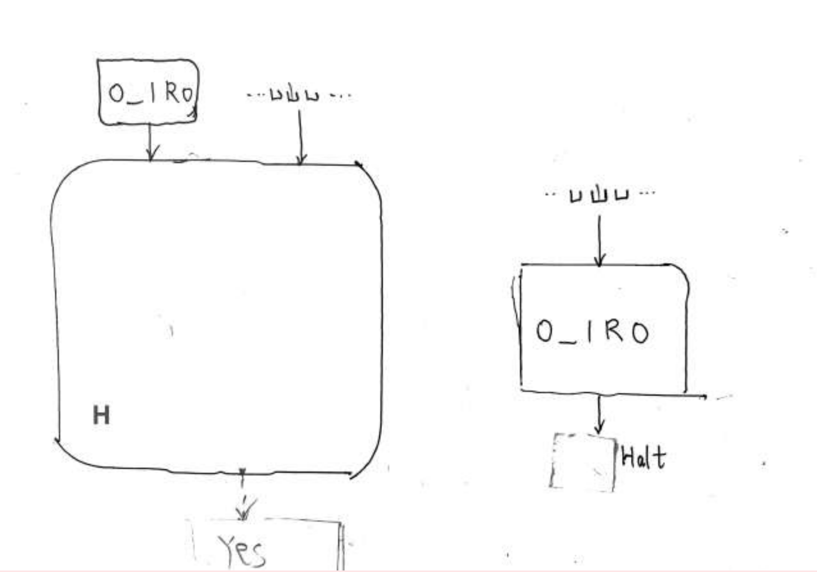
停机问题图示



停机问题图示

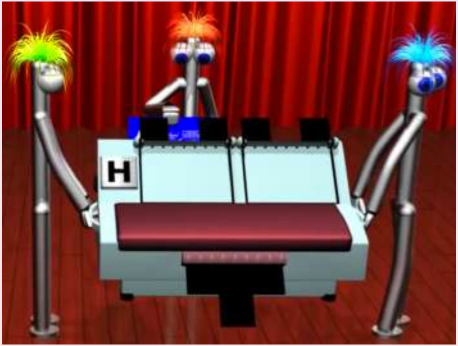


停机问题图示

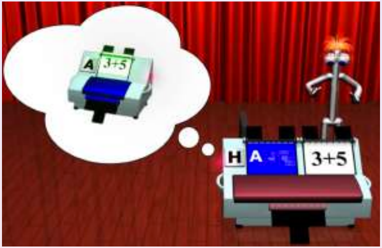


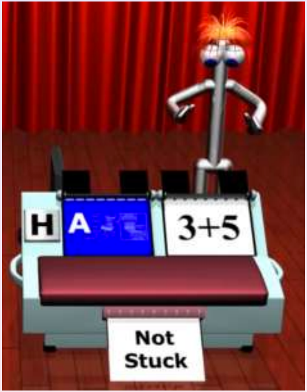
停机问题不可解

- 没有图灵机能用于判定停机问题!
- 证明思想：假设存在这样的图灵机，设这个图灵机为 H。




H
A
3+5


A
3+5
H
A
3+5


H
A
3+5
Not
Stuck

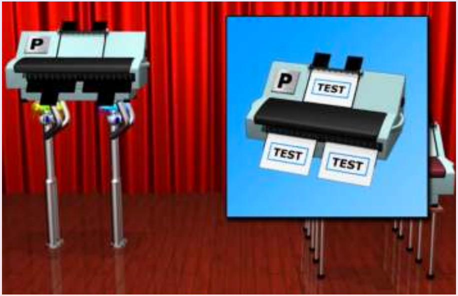

H
A
Stuck

构造过程

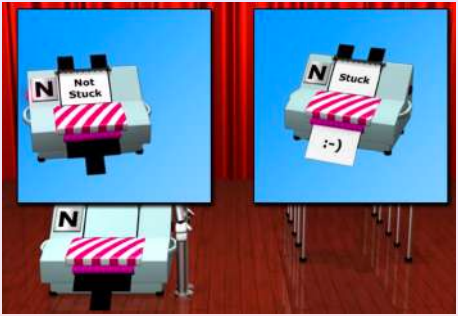
需要两台图灵机作为辅助：

- 复制机
- 反转机

复制机

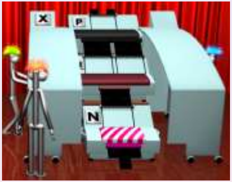


反转机

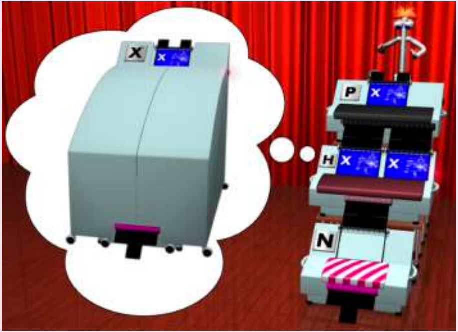


- 若输入为 stuck，则停机并输出笑脸。
- 若输入为 not stuck，则不停机。

组合



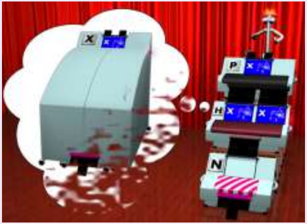
判断

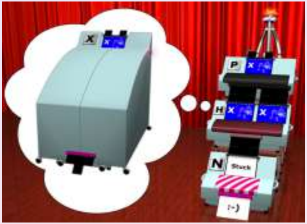


第一种情况

对图灵机 X 输入 X，假设会停机，则……

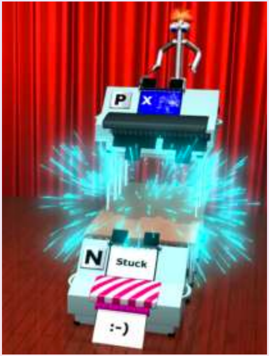
第二种情况





对图灵机 X 输入 X，假设不会停机，则……

结论



图灵机 H 不存在，停机问题不可解。相关视频：

☞ 动画通俗解释-停机问题不可解

主要内容

1. 通用图灵机
2. 停机问题
3. 判定问题

Hilbert 的 Entscheidungsproblem

1928 年，Hilbert 提出如下 Entscheidungsproblem （判定问题）：[1]

是否存在一种**能行方法**（或算法、机械方法）用于判定一阶公式是否有效，使得当输入此公式时，

- 若此公式是有效时，此程序回答“YES”;
- 若此公式是无效时，此程序回答“NO”?

[1] David Hilbert and Wilhelm Ackermann (1928). Grundzüge der theoretischen Logik (Principles of Mathematical Logic). Springer-Verlag.

回答此问题的主要思路

- **能行方法**的数学模型（比如，图灵机）
- 寻找此数学模型中的程序终止（停机）问题，把此归约为上述判定问题

Entscheidungsproblem 的再表述

是否存在一个图灵机（程序）用于判定一阶公式是否有效，使得当输入此公式时，

- 若此公式是有效时，此程序回答“YES”;
- 若此公式是无效时，此程序回答“NO”?

问题的归约

图灵机 x 在输入为 y 时停机，当且仅当下面的推理是有效的：

- 前提：
  - （图灵机的）初始格局是 $\_q_0\underbrace{11\ldots1}_{y+1\text{ 次}}\_$ （初始状态是 q_0，且读写头指向 q_0 右侧的那个符号，此处是 1）
  - （对应于图灵机 x 的指令 $q_iS_jS_ka_lq_m$）对任意的 t，如果在 t 步的格局是 $\ldots q_iS_j\ldots$，那么在 $t+1$ 步的格局就是 $\ldots q_mS_k\ldots$[2]
- 结论：存在这样的 t，在 t 步的格局是 $\ldots q_iS_j\ldots$，并且图灵机 x 的指令没有形如 $q_iS_jS_ka_lq_m$ 的指令。

---

[2] 从第 t 步到第 $t+1$ 步，需要按照动作 a_l 分情况写出前后两种格局的推出关系，这里只明确思想，细节略去。

问题的归约

图灵机 x 在输入为 y 时停机，当且仅当下面的推理是有效的：

- 前提：
  - （图灵机的）初始格局是 $\_q_0\underbrace{11\ldots1}_{y+1\text{ 次}}\_$ （初始状态是 q_0，且读写头指向 q_0 右侧的那个符号，此处是 1）
  - （对应于图灵机 x 的指令 $q_iS_jS_ka_lq_m$）对任意的 t，如果在 t 步的格局是 $\ldots q_iS_j\ldots$，那么在 $t+1$ 步的格局就是 $\ldots q_mS_k\ldots$[2]
- 结论：存在这样的 t，在 t 步的格局是 $\ldots q_iS_j\ldots$，并且图灵机 x 的指令没有形如 $q_iS_jS_ka_lq_m$ 的指令。

[2]从第 t 步到第 $t+1$ 步，需要按照动作 a_l 分情况写出前后两种格局的推出关系，这里只明确思想，细节略去。

问题的归约

图灵机 x 在输入为 y 时停机，当且仅当下面的推理是有效的：

- 前提：
  - （图灵机的）初始格局是 $\_q_0\underbrace{11\ldots1}_{y+1\text{ 次}}\_$ （初始状态是 q_0，且读写头指向 q_0 右侧的那个符号，此处是 1）
  - （对应于图灵机 x 的指令 $q_iS_jS_ka_lq_m$）对任意的 t，如果在 t 步的格局是 $\ldots q_iS_j\ldots$，那么在 $t+1$ 步的格局就是 $\ldots q_mS_k\ldots$[2]
- 结论：存在这样的 t，在 t 步的格局是 $\ldots q_iS_j\ldots$，并且图灵机 x 的指令没有形如 $q_iS_jS_ka_lq_m$ 的指令。

[2]从第 t 步到第 $t+1$ 步，需要按照动作 a_l 分情况写出前后两种格局的推出关系，这里只明确思想，细节略去。

问题的归约

图灵机 x 在输入为 y 时停机，当且仅当下面的推理是有效的：

- 前提：
  - （图灵机的）初始格局是 $\_q_0\underbrace{11\ldots1}_{y+1\text{ 次}}\_$ （初始状态是 q_0，且读写头指向 q_0 右侧的那个符号，此处是 1）
  - （对应于图灵机 x 的指令 $q_iS_jS_ka_lq_m$）对任意的 t，如果在 t 步的格局是 $\ldots q_iS_j\ldots$，那么在 $t+1$ 步的格局就是 $\ldots q_mS_k\ldots$[2]
- 结论：存在这样的 t，在 t 步的格局是 $\ldots q_iS_j\ldots$，并且图灵机 x 的指令没有形如 $q_iS_jS_ka_lq_m$ 的指令。

---

[2]从第 t 步到第 $t+1$ 步，需要按照动作 a_l 分情况写出前后两种格局的推出关系，这里只明确思想，细节略去。

前面提到的推理很容易用一阶公式进行表达，于是：图灵机 x 在输入为 y 时停机，当且仅当某个用一阶语言表达的推理形式（与 x 和 y 相关）是有效的。

假设存在一个图灵机用于判定一阶语言中公式（或推理形式）是否有效。

注意，蓝字部分中的关系是充分必要的，因此，上述用于判定公式是否有效的图灵机同样可用于判定停机问题，即对任意 x、y，判定图灵机 x 在输入为 y 时是否停机。但已经知道，不存在图灵机用于判定停机问题，所以，假设不成立！

前面提到的推理很容易用一阶公式进行表达，于是：**图灵机 x 在输入为 y 时停机，当且仅当某个用一阶语言表达的推理形式（与 x 和 y 相关）是有效的。**

假设存在一个图灵机用于判定一阶语言中公式（或推理形式）是否有效。

注意，蓝字部分中的关系是充分必要的，因此，上述用于判定公式是否有效的图灵机同样可用于判定停机问题，即对任意 x、y，判定图灵机 x 在输入为 y 时是否停机。但已经知道，不存在图灵机用于判定停机问题，所以，假设不成立！

前面提到的推理很容易用一阶公式进行表达，于是：**图灵机 x 在输入为 y 时停机，当且仅当某个用一阶语言表达的推理形式（与 x 和 y 相关）是有效的。**

假设存在一个图灵机用于判定一阶语言中公式（或推理形式）是否有效。

注意，蓝字部分中的关系是充分必要的，因此，上述用于判定公式是否有效的图灵机同样可用于判定停机问题，即对任意 x、y，判定图灵机 x 在输入为 y 时是否停机。但已经知道，不存在图灵机用于判定停机问题，所以，假设不成立！

前面提到的推理很容易用一阶公式进行表达，于是：**图灵机 x 在输入为 y 时停机，当且仅当某个用一阶语言表达的推理形式（与 x 和 y 相关）是有效的。**

假设存在一个图灵机用于判定一阶语言中公式（或推理形式）是否有效。

注意，蓝字部分中的关系是充分必要的，因此，上述用于判定公式是否有效的图灵机同样可用于判定停机问题，即对任意 x、y，判定图灵机 x 在输入为 y 时是否停机。但已经知道，不存在图灵机用于判定停机问题，所以，假设不成立！

结论

- 不存在图灵机（算法）用来判定一阶公式的有效性问题！
- 这个结论一般称为：**一阶逻辑的不可判定性**。
- 注意，所谓“不可判定”不是指有哪个公式的有效性不能被判定，而是指不存在算法用于判定**每个**一阶公式的有效性。

结论

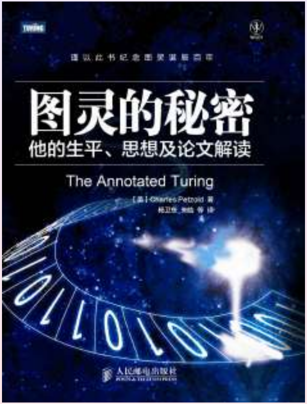
- 不存在图灵机（算法）用来判定一阶公式的有效性问题！
- 这个结论一般称为：**一阶逻辑的不可判定性**。
- 注意，所谓“不可判定”不是指有哪个公式的有效性不能被判定，而是指不存在算法用于判定**每个**一阶公式的有效性。

结论

- 不存在图灵机（算法）用来判定一阶公式的有效性问题！
- 这个结论一般称为：**一阶逻辑的不可判定性**。
- 注意，所谓“不可判定”不是指有哪个公式的有效性不能被判定，而是指不存在算法用于判定**每个**一阶公式的有效性。

推荐阅读

Charles Petzold，《图灵的秘密：他的生平、思想和论文解读》，杨东卫等（译），人民邮电出版社，2012。



Thanks for your attention!

Q & A