Decidability

Diagonalization method

Uncountable sets

Diagonalization method 是康托尔提出的一种数学证明方法， 用于证明某个集合是不可数的（即无法建立起该集合和自然数集之间的双射）

> countable and uncountable sets
>
> 自然数集 $\mathbb{N} = \{1, 2, ...\}$ 是 countable 的
>
> 一个集合 S 是 countable ，则其是有限集，或其是无限集且存在一个 bijection
>
> $$f : \mathbb{N} \to S$$

e. g. 有理数集是 countable 的，但是实数集是 uncountable 的

> Proof. 如果实数集是 countable，则存在一个双射 f ，使得每一个实数都有一个唯一的整数 i 对应，则将所有实数按照自然数的顺序排列，则令
>
> $$x = 0.a_1 a_2 \dots$$
>
> 其中 a_i **不等于** $f(i)$ 小数点后的第 i 位数字。显然 x 是一个实数，且如果存在满足要求的双射，则 x 也有一个对应的自然数 j ，则产生矛盾，如果 x 是第 j 个实数，则其小数点后第 j 位应当等于 $f(j)$ 小数点后第 j 位，而这与 x 的定义矛盾，故实数集是 uncountable

Non-RE languages

根据之前的结论，语言之间的包含关系为

regular language $\subset$ CFL $\subset$ recursive language $\subset$ RE $\subset$ all language

前两个真包含关系已经可以根据形如 $a^n b^n, a^n b^n c^n$ 的语言得出，现在首先证明最后一个真包含关系，即

存在语言不是 RE language

> Proof. 证明分为三步
>
> - 证明所有 TM 的集合是 countable 的

> - 证明所有 language 的集合是 uncountable 的
> - 根据上述结论，不可能存在从 TM 集合到 language 集合的满射
>
> 所有 TM 的集合是 countable 的
>
> > Proof. 所有 string 的集合 Σ^* 是 countable 的，因为可以将所有 string 按照其长度和字典序排成一列，为其标号
> >
> > 而每个图灵机 M 可以表示为一个有限长度的 string，设其为 $< M >$ ，则从所有 string 的集合中删去所有不是 TM 的 string，则得到 TM 的集合，显然这个集合是 countable 的
>
> 所有 language 的集合是 uncountable 的
>
> > Proof. 所有 string 的集合是 countable 的，则可以将每个 string 编号，得到 $s_1, s_2, \ldots$ ，则一个 language L 可以由一个特征向量 x^L 描述，如果 $s_i \in L$ 则 $x_i^L = 1$ ，否则 $x_i^L = 0$ ，则同样使用对角法，假设所有 language 的集合是 countable，则对每个特征向量存在一个唯一的自然数与其对应。则考虑语言 L
> >
> > $$x_i^L = \begin{cases} 1 & f(i)_i = 0 \\ 0 & f(i)_i = 1 \end{cases}$$
> >
> > 即如果第 i 个语言包含 s_i 则 L 不含 s_i ，反之亦然
> >
> > 则如果所有 language 的集合是 countable 则产生矛盾，若 L 对应的自然数为 j ，则按照映射如果 $s_j \in L$ 则 $f(j)_j = 1$ ，按照定义则 $x_j^L = 0$ ，即 $s_j \notin L$ ，矛盾
>
> 由于 TM 的集合是 countable 而 language 的集合是 uncountable，故存在语言不会被任何 TM 接受

The halting problem is undecidable

Halting problem

停机问题的定义是语言

$$\text{HALT} = \{< M, x >: \text{TM } M \text{ halts on input } x\}$$

其中 $< M, x >$ 是 TM M 和 x 的字符串形式

显然这个语言是 RE，模拟 M 对于 x 为输入的运行即可，现在需要证明 HALT 是否是 decidable，即是否存在一个 algorithm 接受 HALT

HALT 不是 decidable 的

> Proof. 以下给出大致的证明流程

> 假设存在 TM H decides HALT，即如果 M 在 x 上 halt，H 接受，否则 H 拒绝
>
> 则定义一个新的 TM H' ，对于输入 $< M >$ ，如果 H 接受 $< M, < M >>$ ，则无限循环，否则 halt
>
> 现在考虑将 $< H' >$ 输入 H' ，则如果 H' halt，按照定义说明 H 拒绝 $< H', < H' >>$ ，而按照 H 的定义，H' 不应当 halt，反之亦然，矛盾
>
> 根据反证法，H, H' 均不存在

使用对角法也可以证明该结论

> Proof. 假设存在 TM H decides HALT，即如果 M 在 x 上 halt，H 接受，否则 H 拒绝
>
> 所有 TM 的集合是 countable 的，则可为每个 TM 编号，这样可构造一个矩阵 A，其中 a_{ij} 为 ACCEPT 表示 H 对输入 $< M_i, < M_j >>$ 接受，则可构造一个 TM H' ，对于任意自然数 i ，如果 H 接受 $< M_i, < M_i >>$ ，则 H' 在 $< M_i, < M_j >>$ 上 halt ，反之则接受，而其余 $i \neq j$ 的结果与 H 相同。
>
> 对于 H' ，由于其也是 TM，则存在一个唯一对应的自然数，设为 j ，则考虑 H' 以 $< M_j, < M_j >>$ 为输入，如果 H' 接受，则按照定义 H 拒绝，应当 halt，反之亦然，推出矛盾

RE and co-RE

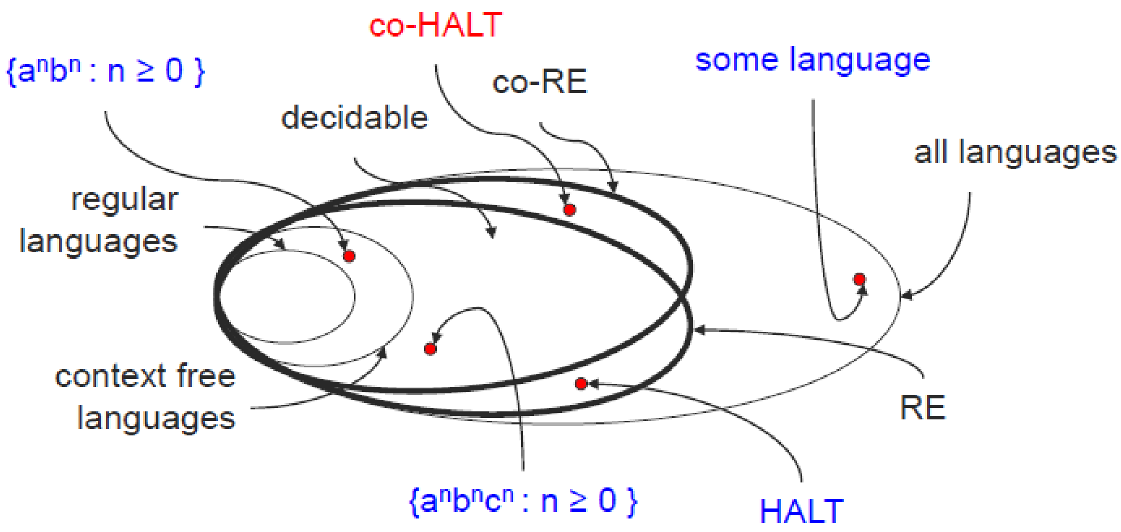
一个 RE 的补是叫做 co-RE，而所有 co-RE 的集合组成了 co-RE 语言（注意 co-RE 不是非 RE）

则有结论：一个语言 L 是 decidable $\iff$ L 是 RE 且 L 是 co-RE

> Proof.
>
> ($\Rightarrow$): 显然 L 如果是 decidable 则 L 一定是 RE，则 L 的补只需要反转 TM 的接受/拒绝，则 L 的补同样是 RE，得出 L 是 co-RE
>
> ($\Leftarrow$): 易得存在 TM M 接受 L，以及存在 TM M' 接受 L 的补，则只需要构造 TM 平行模拟 M, M' ，当 M 接受则接受，当 M' 接受则拒绝，即得到一个 algorithm 接受 L， L 为 decidable

根据上述结论，可以得出 HALT 的补不是 RE

> Proof. 如果 HALT 的补是 RE，则 HALT 是 co-RE，根据上文结论，HALT 为 decidable，而这与之前的结论相矛盾



上述图片阐明了各个语言之间的关系

Undecidable Problems about TM

Reductions

归约（reduction）是一种转换。如果存在一个转换将问题 P_1 的实例转换为问题 P_2 的实例，且输出相同（如果该实例在 P_1 中是 YES，则转换后的实例在 P_2 中也是 YES），则称 P_1 reduces to P_2，一般记为

$$P_1 \leqslant P_2$$

归约可以说明 P_2 至少和 P_1 **一样难**，因为如果存在一个可以解决 P_2 的 oracle，可以通过归约解决 P_1，但反之不成立

归约不要求一定是满射和单射，事实上一般 P_1 映射过去后的集合只是 P_2 的一个子集。

更为正式的定义如下：

Many-one reduction: 考虑 A, B 是 alphabet Σ, Γ 上的形式语言，则从 A 到 B 的一个 **Many-one reduction** 是一个可计算的函数 $f : \Sigma^* \to \Gamma^*$ 满足

$$\forall w, w \in A \iff f(w) \in B$$

> Computable: f is computable if there exists a TM M_f s. t. on every $w \in \Sigma^*$, M_f halts on w with $f(w)$ written on its tape

若存在满足的 f 则称 A m-reduces to B，记为

$$A \leqslant_m B$$

如果 f 是 injective 则称 A 1-reduces to B，记为

$$A \leqslant_1 B$$

如果存在一个 A 到 B 的 reduction，则有

- 如果 A undecidable，则 B undecidable
- 如果 A 是 non-RE，则 B 是 non-RE

> Proof.
>
> 考虑陈述 1，如果 A 是 undecidable，但 B 是 decidable，则可以通过 reduction 将 A 的实例 w 转换为 $f(w)$ ，然后使用 B 的 decider 判定 $f(w)$ ，从而 A 也可判定，矛盾
>
> 陈述 2 同理，可以通过 B 的 recognizer 和 reduction 构造 A 的 recognizer，导出矛盾

因此 reduction 可以用来证明语言的 undecidable/non-RE

假设已知 A 是已知的 undecidable 的语言，欲证 B 是 undecidable，步骤为

- 假设 B 是 decidable，则存在 decider M_B
- 使用 M_B 判定 A （即构造从 A 到 B 的 reduction f，对于任意 w ，使用 M_B 判定 $f(w)$ ）
- 这样得出 A 是 decidable，导出矛盾，根据归谬法，B 是 undecidable

A_{TM} is undecidable

A_{TM} 的定义为

$$A_{TM} = \{< M, w >: w \in L(M)\}$$

而根据上文结论，已知 HALT 是 undecidable 的，那么通过 reduction 可以证明 A_{TM} 是 undecidable

> Proof.
>
> 将 HALT 归约到 A_{TM}
>
> 如果 A_{TM} 可判定，可以通过 A_{TM} 判定 HALT，从而导出矛盾
>
> 对于 HALT 的实例 $< M, w >$ ，可以构造一个 TM H ，对于输入 $< M, w >$ 判断 $< M, w >$ 是否属于 A_{TM}
>
> - 如果属于，则说明 M halt 并接收 w，H accept
> - 如果不属于，则说明 M 或是 halt 并 reject w ，或是在 w 上无限循环
>   - 构造 M' ，模拟 M 的运行，且当 M halt 时，若 M 结果为 accept 则 M' reject，反之若 M 结果为 reject 则 M' accept，若 M 无限循环则 M' 同样会无限循环

> - 判断 $< M', w >$ 是否属于 A_{TM} ，如果属于则说明 M 在 w 上 halt 并 reject，则 H accept，否则说明无论 M, M' 在 w 上都循环，H reject
>
> 由于 A_{TM} 可判定，故 H 最终总会 halt，且有 $HALT = L(H)$ ，而 HALT 为 undecidable，矛盾。
>
> A_{TM} 不可判定
>
> 除了归约的证明，也可以直接用归谬法证明
>
> 如果 A_{TM} decidable，假设其 decider 为 H ，则构造 TM D 接受 $< M >$ 输入
>
> - 如果 $< M, < M >> \in A_{TM}$ 则 D reject
> - 如果 $< M, < M >> \notin A_{TM}$ 则 D accept
>
> 则考虑 D 以 $< D >$ 为输入，如果 accept，说明 $< D, < D >> \notin A_{TM}$ ，即 $< D > \notin D$ ，导出矛盾。反之亦然，故根据归谬法，不存在 D 和 H，亦即 A_{TM} 为 undecidable

E_{TM} is undecidable

E_{TM} 的定义为

$$E_{TM} = \{< M >: L(M) = \varnothing\}$$

E_{TM} 是 undecidable 的

> Proof.
>
> 将 A_{TM} 归约到 E_{TM}
>
> 对于 A_{TM} 的实例 $< M, w >$ ，可以构造一个 TM M' ，M' 模拟 M 在 w 为输入时的运行
>
> - 如果 M accept w ，则 M' accept 其输入 x
> - 否则 M' reject 其输入 x
>
> 如此构造后 $< M, w > \in A_{TM} \iff L(M') \neq \varnothing \iff < M' > \notin E_{TM}$
>
> 如果 E_{TM} decidable 则 A_{TM} 也 decidable，导出矛盾，故 E_{TM} undecidable

事实上 E_{TM} 是 non-RE 的，因为其补是 RE（构造 NTM 猜测一个输入 x 判断其是否属于 $L(M)$ ，只要 $L(M)$ 非空则一定能猜到一个 x 并接受 $< M >$），而如果 E_{TM} 也是 RE 则 E_{TM} 为 decidable

Rice's Theorem

Rice's Theorem: all *non-trivial*, *semantic properties* of programs are undecidable

semantic property: Rice's theorem 适用范围是关于语言的属性而非关于机器/程序的属性。e. g. 一个程序是否会在所有输入上 halt 就是一个 semantic property，而程序是否会运行超过 1000 步或 TM 是否有超过 5 个状态就不是 semantic property

如果将 property 看作一个接受 TM 为输入的 language P，则如果 $L(M_1) = L(M_2)$，有

$$< M_1 >\in P \iff < M_2 >\in P$$

non-trivial: 一个 trivial 的 property 满足：或是所有语言都有该 property，或是没有语言有该 property。否则其为 non-trivial

> P is non-trivial if
>
> $$\exists M_1, < M_1 >\in P$$
> $$\exists M_2, < M_2 >\notin P$$

要注意 Rice's Theorem 的适用范围仅限 **non-trivial semantic properties**

P is non-trivial $\iff$ P is undecidable

> Proof. 将 A_{TM} 归约到 P
>
> 思路为如果 P non-trivial 则一定存在 $< M_L >\in P$，对于 A_{TM} 的实例 $< M, w >$，构造 TM M'，模拟 M 在 w 上的运行，如果 accept，则模拟 M_L 在输入 x 上的运行，如果 accept 则 accept，这样 $< M, w >\in A_{TM} \iff L(M') = L(M_L) \iff < M' >\in P$
>
> P 如果 trivial 则证明过程可以推出矛盾，e. g. P 包含所有 TM，则
>
> $$< M, w >\notin A_{TM} \iff L(M') = \varnothing \iff M' \in P$$

以下 properties 都是 undecidable 的

- 一个 language 是否为空
- 一个 language 是否有限
- 一个 language 是否是 regular language
- 一个 language 是否是 CFL