Complexity

Intractable Problems: problems that cannot to be solvable in **polynomial** time

Complexity

在现实问题中，时间和空间的资源都是有限的，因此需要分析一个 algorithm 所需求的资源

在分析时一般进行 worst-case analysis，一个算法需要的资源是 input string 的长度的函数，而其值是对所有给定长度的 input，算法需要的**最大**资源数

Time complexity

一个 TM M 的 time complexity 是一个函数 $f:\mathbb{N}\to\mathbb{N}$，$f(n)$ 是 M 对于任意长度为 n 的 input string，所需要运行的最多的步数

在考虑 time complexity 时，通常不考虑精确的步数，且只考虑在输入规模很大的情况下的复杂度。具体而言，使用 asymptotic notation (big-Oh notation)，只关注 f 的最高项且忽略其常系数

> 对于函数 $f,g:\mathbb{N}\to\mathbb{R}^+$，如果存在正整数 c,n_0 满足对任意 $n\geqslant n_0$
>
> $$f(n)\leqslant cg(n)$$
>
> 则称 $f(n)=O(g(n))$

具体含义是规模较大的时候 f 的增长不会超过 g

根据 time complexity 即可定义 time complexity class

$$TIME(t(n))=\{L:\text{there exists a TM } M \text{ that decides } L \text{ in time } O(t(n))\}$$

这里的 TM 指的是 single-tape TM，而对于 multi-tape TM 来说，对于任意 $t(n)\geqslant n$，任何 $t(n)$ 的 multi-tape TM 都有一个等价的 $O(t(n)^2)$ 的 single-tape TM

$\mathcal{P}$ and $\mathcal{NP}$

$\mathcal{P}$ 是所有能在**确定性**单带 TM 上以**多项式时间**判定的语言的集合

$$\mathcal{P} = \bigcup_{k \geqslant 1} TIME(n^k)$$

$\mathcal{NP}$ 是所有能在**非确定性**单带 TM 上以**多项式时间**判定的语言的集合

如果定义

$$NTIME(t(n)) = \{L : \text{there exists a NTM } M \text{ that decides } L \text{ in time } O(t(n))\}$$

则可以定义

$$\mathcal{NP} = \bigcup_{k \geqslant 1} NTIME(n^k)$$

对于 $\mathcal{NP}$ 中的语言来说，有一个很重要的性质

$$L \in \mathcal{NP} \iff L = \{x : \exists y, |y| \leqslant |x|^k, < x, y > \in R\}$$

其中 $R \in \mathcal{P}$

> Proof.
>
> ($\Leftarrow$): 可以构造一个 NTM 判定 L，首先用 $|x|^k$ 步去猜测一个 y，然后用 R 判定 $< x, y >$
>
> ($\Rightarrow$): 假设 L 被一个 n^k 的 NTM M 判定，则定义 R 是所有满足“y 是 M 接受 x 的一个 accepting computation history” 的 $< x, y >$，则如果 M accept x，一定存在一个满足条件的 y，且 $|y| \leqslant |x|^k$ （因为 M 是一个 n^k 的 NTM），且 $< x, y > \in R$

y 被称为 certificate，即对于一个 $\mathcal{NP}$ 问题来说，验证其解是一个 $\mathcal{P}$ 问题。

> e. g. 最大团问题被定义为
>
> $$CLIQUE = \{< G, k >\}$$
>
> 满足图 G 中存在大小为 k 的团，这是一个 $\mathcal{NP}$ 问题，但是
>
> $$R = \{<< G, k >, S >\}$$
>
> 满足 S 是 G 中顶点的一个子集且 S 是大小为 k 的团，这是一个 $\mathcal{P}$ 问题，其中 S 就扮演了 certificate 的角色

$\mathcal{NP}$ 不是一个有实践意义的计算模型，因为现实生活中的计算不是非确定性的，但是 $\mathcal{NP}$ 中语言的性质说明了这类问题的一个特征，即可以找到问题的一个实例并且可以有效地测试其是否是一个满足要求的解

如果定义

$$\mathcal{EXP} = \bigcup_{k \geqslant 1} TIME(2^{n^k})$$

则各个 time complexity class 之间的关系为

$$\mathcal{P} \subseteq \mathcal{NP} \subseteq \mathcal{EXP}$$

而这些语言都是 decidable languages 的子集

Polynomial-Time Reductions

reduction 是一个 computable function f，而如果 f 是一个 poly-time computable function，则称这个 reduction 是 poly-time reduction，记为

$$A \leqslant_p B$$

> poly-time computable: 对于函数 f，对于 $g(n) = n^{O(1)}$，存在一个 $g(n)$ TM M_f 满足对任意输入 w，M_f 都会 halt 且在 tape 上留下 $f(w)$

同样的，这个 reduction 隐含的意义为 B 至少和 A 一样难

则有，如果 $A \leqslant_p B, B \in \mathcal{P}$，则 $A \in \mathcal{P}$

> Proof. 可以在 poly-time 将 A 的实例 w 转换为 B 的实例 $f(w)$，且可以在 poly-time 使用 B 的 decider 判定 $f(w)$（由于 poly-time reduction，$|f(w)| \leqslant |w|^k$），由于多项式运算的封闭性，总体的时间仍是 poly-time 的

$\mathcal{NP}$-complete

称一个语言 L 是 $\mathcal{NP}$-complete，如果其满足

- $L \in \mathcal{NP}$
- $\forall L' \in \mathcal{NP}, L' \leqslant_p L$

如果仅满足第二条约束则称 L 是 $\mathcal{NP}$-hard

证明一个语言 L 属于 $\mathcal{NPC}$ 可以分为两步

1. 证明 $L \in \mathcal{NP}$
2. 对于已知的 $M \in \mathcal{NPC}$，有 $M \leqslant_p L$（根据 poly-time reduction 的传递性可得其满足第二条约束）

对于 $\mathcal{NPC}$ 有如下结论

如果 $B \in \mathcal{NPC}$ 且 $B \in \mathcal{P}$，则 $\mathcal{P} = \mathcal{NP}$

> Proof. 任取 $L \in \mathcal{NP}$，有 $L \leqslant_p B$，而由于之前的结论，$B \in \mathcal{P}, L \in \mathcal{P}$，则 $\mathcal{P} = \mathcal{NP}$

SAT problem

SAT 指的是一个布尔表达式是否为 satisfiable 的问题，这个问题是一个 $\mathcal{NPC}$ 问题

> 一个布尔表达式是 satisfiable 的，说明存在一个对其中 variable 的 assignment 使得整个表达式为真

其证明的思路分为两部分

- SAT 是 $\mathcal{NP}$ 问题：只需要构造一个 NTM 猜测并验证 assignment
- $\forall A \in \mathcal{NP}, A \leqslant_p SAT$：令 M 为判定 A 的 n^k NTM，则对于输入 w，可以构造一个布尔表达式 $\varphi_{M,w}$，满足 $\varphi_{M,w}$ $\text{is satisfiable} \iff w \in L(M)$，具体构造过程见课本 10.2.3 节

3SAT 是 SAT 问题的一个特例，即一个 3-cnf 的布尔表达式是否为 satisfiable

3SAT 也是 $\mathcal{NPC}$ 问题

More $\mathcal{NPC}$ problems

图中的最大独立集（independent set, IS）：独立集 I 是一个点集，满足其中任意两点都没有边相连

图中的最大团（clique）：团 S 是一个点集满足其中任意两点都有边相连

图中的最小顶点覆盖（vertex cover, VC）：覆盖集 C 是一个点集，满足对任意边 e，其至少有一个顶点在 C 中

图中的 Hamilton 回路：存在一个环经过每个顶点且仅经过一次，类似的问题还有 TSP 问题

子集和问题（subset sum）：对于一个集合 N 存在一个子集 S 满足 S 中的数的和等于给定值 k