IP=PSPACE

有同学在问我这个，我干脆就把我的讲解思路写出来了，下面是比较 **high-level** 的阐述；能阅读的参考资料倒是非常地多：

- https://cs.brown.edu/courses/gs019/papers/ip.pdf
- https://www.cs.umd.edu/~jkatz/complexity/f11/
- Dexter C.Kozen. Theory of Computation

1 $\texttt{IP} \subset \texttt{PSPACE}$

$L \in \texttt{IP}$ 指的是可以找到一组算法 $(\mathcal{P}, \mathcal{V})$ 满足：

- 【$\mathcal{P} \leftrightarrow \mathcal{V}$ 的 completeness】$x \in L$: $\mathcal{P} \leftrightarrow \mathcal{V}$ 输出 1 的概率 $\geq 2/3$
- 【$\mathcal{V}$ 的 soundness】$x \notin L$: $\forall\, \mathcal{P}^*, \mathcal{P}^* \leftrightarrow \mathcal{V}$ 输出 1 的概率 $\leq 1/3$

<u>相对于 $\mathcal{V}(x_0)$ 的最优策略 $\tilde{\mathcal{P}}_{x_0}$</u>：

- validness: 满足上述两个条件
- locally maximum: 所有满足 validness 的 $\mathcal{P}$ 当中，$\tilde{\mathcal{P}}_{x_0}(x_0) \leftrightarrow \mathcal{V}(x_0)$ 输出 1 的概率最大
- 关键观察：**我们甚至无需考虑 validness**，因为存在一个**合法**的 universal 的 $\tilde{\mathcal{P}}$ 可以在不同的输入 x 上呈现出 $\tilde{\mathcal{P}}_x$ 在 x 上呈现的策略选择
    - 【引理】若 $(\mathcal{P}, \mathcal{V})$ 为 L 的**合法**的交互证明系统，$\tilde{\mathcal{P}}$ 也为一个合法的交换证明系统；**即局部最优策略的组合则是合法的全局最优策略**
        - $x \in L$: $\tilde{\mathcal{P}}(x) \leftrightarrow \mathcal{V}(x)$ 输出 1 的概率合法 (locally maximum)
        - $x \notin L$: 由 $\mathcal{V}$ 的 soundness 性质保证

$\texttt{IP} \subset \texttt{PSPACE}$ **解决思路**：

- 用 $\texttt{PSPACE}$ 机器找到 $\tilde{\mathcal{P}}_{x_0}(x_0)$，并且该机器可以用 $\texttt{PSPACE}$ 机器模拟（一种常见的找**不变量**的方法）
- 计算 $\tilde{\mathcal{P}}_{x_0}(x_0) \leftrightarrow \mathcal{V}(x_0)$ 输出 1 的概率

$\texttt{PSPACE}$ 机器模拟 $\tilde{\mathcal{P}}_{x_0}(x_0)$，仅需在保持transcript的情况下去模拟每一步即可，即找到 next-message function $f_{opt}(tr)$：

- transcript tr 可以表示为 $x_0; m_0, \ell_0; \cdots; m_k, \square$
- $\ell_k = f_{opt}(x_0; m_0, \ell_0; \cdots; m_k)$
- $$\begin{aligned}\ell_k &= \mathrm{argmax}_{\tilde{\ell}_k} \Pr[\mathcal{V} \Rightarrow 1 \mid x_0; m_0, \ell_0; \cdots; m_k, \tilde{\ell}_k] \\ &= \mathrm{argmax}_{\tilde{\ell}_k} \sum_{m_{k+1}} \Pr[x_0; m_0, \ell_0; \cdots; m_k, \tilde{\ell}_k; m_{k+1}] \cdot \max_{\tilde{\ell}_{k+1}} \Pr[\mathcal{V} \Rightarrow 1 \mid x_0; m_0, \ell_0; \cdots; m_k, \tilde{\ell}_k; m_{k+1}, \tilde{\ell}_{k+1}]\end{aligned}$$
- Intuition类比于多项式谱系中“量词的交换一样”，这里类比到 $\sum m_k \max \ell_k$ 的交替；而之所以可以这样搞，也是这个优化问题具有“最优子结构”，即类似于动态规划问题.

2 $\texttt{PSPACE} \subset \texttt{IP}$

SumCheck 协议是一个知名的协议证明：

$$H = \sum_{\mathbf{x}} f(\mathbf{x})$$

该协议可以将指数级别的运算转化为单次验证运算；

为了使用SumCheck协议，我们需要对问题进行算术化，即将问题用多项式的语言进行重新表述，下面以 $\#SAT$ 问题为例子：

- $\#SAT : \{(CNF, K) \mid \#(\mathbf{x} : CNF(\mathbf{x}) = 1) = K\}$ 即是判断 CNF 的所有可能赋值的数目总和是否为 K

算术化 $\#SAT$:

- SAT 问题的结构是 $(\cdot \vee \cdots \vee \cdot) \wedge (\cdot \vee \cdots \vee \cdot) \wedge (\cdot \vee \cdots \vee \cdot) \wedge \cdots \wedge (\cdot \vee \cdots \vee \cdot)$
- 则可以算术化为如下形式 $\phi(x_1, \cdots, x_n)$：

$$\prod (1 - \prod)$$

- 而求解所有可能赋值的总数即求解：

$$K = \sum_{x_1, \cdots, x_n} \phi(x_1, \cdots, x_n)$$

而这显然可以用SumCheck完成，从而 $\#SAT \in \texttt{IP}$

下面证明 $\texttt{PSPACE} \subset \texttt{IP}$，即只需证明 $QBF \in \texttt{IP}$

算术化 QBF:

- QBF 问题的结构是 $\forall x_1 \exists x_2 \forall x_3 \exists x_4 \cdots \forall : \phi(x_1, \cdots, x_n)$
- 容易算术化 $\phi(x_1, \cdots, x_n)$ 为 $\mathbb{F}_p$ 上的多项式 $\varphi(x_1, \cdots, x_n)$
- 则 QBF 可以算术化为：$(*)$

$$1 = \bigwedge_{x_1 \in \{0,1\}} \bigvee_{x_2 \in \{0,1\}} \bigwedge_{x_3 \in \{0,1\}} \bigvee_{x_4 \in \{0,1\}} \cdots \bigwedge_{x_n \in \{0,1\}} \varphi(x_1, \cdots, x_n)$$

  这里 $\bigwedge_{x \in \{0,1\}} f(x) := f(0) \cdot f(1)$，$\bigvee_{x \in \{0,1\}} f(x) := 1 - (1 - f(0))(1 - f(1))$

【关键障碍】次数过一次 $\bigvee, \bigwedge$ 会平方一下，则**次数过高**，无法保证 SumCheck 协议的 soundness

【关键问题】如何改写式子使得**次数增长很缓慢**

【关键技术】**线性化技术**：假定 $f(x)$ 的次数很高，通过 $\bigvee, \bigwedge$ 次数会平方放大，我们定义 线性化子 L_x：

$$L_x f(x) := (1 - x) \cdot f(0) + x \cdot f(1)$$

之所以可以线性化是因为注意到 $x_i \in \{0, 1\}$ 均成立，所以线性化在此成立：

$$\bigvee_{x_i \in \{0,1\}} f(x_i) = \bigvee_{x_i \in \{0,1\}} L_{x_i} f(x_i)$$

如果 $x_i \in B$，则需要进行插值。

总地来说我们改写 $(*)$ 如下：【由于 $x_i \in \{0,1\}$ 的实例化取值特点，我们可以验证这种改写是等价的】

$$1 = \bigwedge_{x_1\in\{0,1\}} L_{x_1} \bigvee_{x_2\in\{0,1\}} L_{x_2} \circ L_{x_1} \bigwedge_{x_3\in\{0,1\}} L_{x_3} \circ L_{x_2} \circ L_{x_1} \bigvee_{x_4\in\{0,1\}} \cdots \bigwedge_{x_n\in\{0,1\}} L_{x_n} \circ \cdots \circ L_{x_1}\ \varphi(x_1,\cdots,x_n)$$

由于我们始终加上上线性化子，在任何一层的 SumCheck协议中，$f(x)$ 的次数都很低，从而可以使用 SumCheck 协议.

注意 $L_{x_n} \circ \cdots \circ L_{x_1}\ \varphi(x_1,\cdots,x_n)$ 也可以使用 sumcheck 协议：

$$\begin{gathered}
f_n(x_1,\cdots,x_n) := L_{x_n} \circ \cdots \circ L_{x_1}\ \varphi(x_1,\cdots,x_n) \\
f_{n-1}(x_1,\cdots,x_n) := L_{x_{n-1}} \circ \cdots \circ L_{x_1}\ \varphi(x_1,\cdots,x_n) \\
\cdots \\
f_1(x_1,\cdots,x_n) = L_{x_1}\varphi(x_1,\cdots,x_n)
\end{gathered}$$

即 MLE 的 SumCheck:

$$\sum_{\mathbf{y}} \widetilde{eq}(\mathbf{x}_0,\mathbf{y}) \cdot f(\mathbf{y}) = h_0$$

3 $O(f(n))$-SPACE $\subset O(n\cdot f(n))$ IP

假定 n 为输入大小，我们称一个语言 L 在 $O(n\cdot f(n))$ 是指存在一个无限计算能力的 P 和对应的复杂度为 $O(n\cdot f(n))$ 的验证时间的 V （假定使用多带图灵机的计算模型），上述命题是指任何可以被 $s = O(f(n))$ 空间的多带图灵机判定的语言，一定可以被命题对应的 IP 所判定，这是比 $\texttt{PSPACE} \subset \texttt{IP}$ 更强的结论，给出了一个谱系结果。

证明思路如下，我们还是依然将判定问题转为**路径可达性**问题，而路径可达性问题可以被矩阵的乘法所表征：

- s-SPACE 的图灵机的格局图 A 为 $2^{s+c} \times 2^{s+c}$ 的邻接矩阵 （记录状态、指针的位置以及带子上的信息）
- 不失去一般性，可以假设 i-index 代表的是初始状态，而 j-index 代表的接受状态
- 我们只需要证明 $A^{2^{s+c}}$ 的 (i,j) 坐标为 1 即可
- Prover 拥有无限的计算能力当然可以计算出来，问题在于如何向 Verifier 证明？
  - 想要递归地证明
    - 注意到在合适的算术化之后上述式子也能使用 sumcheck
    - $\widetilde{f_X}(\mathbf{i},\mathbf{j})$ 指对 X 矩阵进行算术化，成为 MLE
    - 有关系
    - $$\widetilde{f_{X^2}}(\mathbf{i},\mathbf{j}) = \sum_{\mathbf{t}} \widetilde{f_X}(\mathbf{i},\mathbf{t}) \cdot \widetilde{f_X}(\mathbf{t},\mathbf{j})$$
    - 从而 Prover 利用 sumcheck 可以把 X^2 上某个点的求值归约到 X 上两个点上的求值（这个又可以聚合成一个点）
  - 从而 Verifier 只需要可以计算 $\widetilde{f_A}(\mathbf{r}_x,\mathbf{r}_y)$ 就可以了，但是对于这么大的 A 应该如何计算？
    - 普通计算的开销为 2^{2s+2c}
    - 可以利用 A 的结构性：

- 计算的局部性：每次指针移动总只移动一位，改变带子上的某一位的内容，也就是 $A_{(i,j)}=1$ 的位置是可以按照计算的局部性进行**二次划分**的：第一次划分在于指针的位置、状态值、某一个被改变的带子上的值，然后二次划分是剩下带子上的内容，注意到二次划分上带子的内容**遍历性**特点，从而第二次划分对应的求和容易构造一个值，而一次划分对应的求和是容易做的；按照这个思路，我们就可以得到可以在 $O(s\cdot n)$ 的时间下求值 $\widetilde{f_A}(\mathbf{r}_x,\mathbf{r}_y)$