第九次作业-solution

Lecturer: 杨启哲 Last modified: 2025 年 12 月 1 日

1. 设计一个非确定算法来求解 SAT 可满足性问题。

解答. 该非确定算法如下:

SAT 可满足性问题 SAT(ϕ)

- **猜测阶段** 猜测长度为 n 一个序列，其中 n 为 ϕ 中变量的个数，序列中的每一位表示对应变量的赋值（0 或者 1）。
- **验证阶段** 验证该序列是否为一个可满足的赋值，即
  - 依次验证 ϕ 中的每一个子句是否被满足。
  - 若存在某个子句不被满足，则直接抛弃。

  如果满足上述条件，则接受，否则拒绝。

□

2. 考察图的团问题的判定版本和优化版本:

图的团问题的判定版本

- **输入：** 图 G 和正整数 k。
- **输出：** 是否存在一个大小为 k 的团。

图的团问题的优化版本

- **输入：** 图 G。
- **输出：** 图 G 的最大团的大小。

假设现在有一个多项式时间的算法可以解决图的团问题的判定版本，试证明可以在多项式时间内解决图的团问题的优化版本。

解答. 我们使用通用的策略，假设有一个多项式时间的算法 $\mathrm{DCliq}(G,k)$ 判定图的团问题的判定版本，注意到 G 中最大的团的大小不会超过 $|V|$，因此我们可以使用如下的策略:

求解图的团问题 $\mathrm{Cliq}(G)$

- 调用 $\mathrm{DCliq}(G, K)$，其中 $K = |V|$，如果返回 False，则拒绝。
- 否则令 $K' = \frac{K}{2}$，继续调用 $\mathrm{DCliq}(G, K')$。
- 如果返回 True，则继续调用 $\mathrm{DCliq}(G, \frac{K'}{2})$; 否则调用 $\mathrm{DCliq}(G, \frac{K'+K}{2})$，直到找到最小的 K 使得该算法接受。

$\square$

3. (三角剖分) 给定平面上包含 n 个顶点的凸多边形 P（给定各顶点的坐标），P 的一个三角剖分由 P 中除端点外不相交的 $n-3$ 条对角线构成，使得 P 被分割成 $n-2$ 个三角形。三角剖分的代价是所有对角线长度之和，请设计一个高效的算法来求解 P 中代价最小的三角剖分。

解答. 我们可以使用动态规划的方法来解决该问题, 将这 n 个顶点逆时针编号 $1, 2, \cdots, n$。我们定义 $dp[i][j]$ 为从第 i 个顶点逆时针到第 j 个顶点组成的凸多边形中的最小三角剖分代价, 则 $dp[i][j]$ 满足:

$$dp[i][j] = 0 \quad \text{if } j - i \leqslant 2$$

并且我们有如下的递推关系:

$$dp[i][j] = \min_{i+1<k<j} \{dp[i][k] + dp[k][j] + \mathrm{dist}(i, k)\}$$

其中 $\mathrm{dist}(i, j)$ 表示第 i 个顶点到第 j 个顶点的距离。我们可以使用如下的动态规划算法来求解该问题:

三角剖分的最小代价

输入: 顶点集合 V，$n = |V|$
输出: 三角剖分的最小代价

```
function MinCost(V, n)
    dp ← new array[n][n]
    for i = 1 to n do
        dp[i][i] ← 0
        dp[i][i + 1] ← 0
        dp[i][i + 2] ← 0
    end for
    for l = 3 to n do
        for i = 1 to n − l do
            j ← i + l
            dp[i][j] ← ∞
            for k = i + 2 to j − 1 do
                dp[i][j] ← min{dp[i][j], dp[i][k] + dp[k][j] + dist(i, k)}
            end for
```

```
        end for
    end for
    return dp[1][n]
end function
```

该算法的时间复杂度为 $O(n^3)$。 □

4. (2–SAT 问题)2–SAT 问题是在 SAT 问题基础上增加每个子句至多包含两个文字的限制，即每个子句形如 $(c \vee c')$，这里的 c 为 x_i 或者 $\neg x_i$。我们可以发现，增加了这样的限制后，该问题便是可以高效解决的了。请给出一个多项式时间的算法赖求解 2–SAT 问题。

Hint: 可以考虑将其转换成一张图

解答. 我们将其转换成一张图，给定一个具有 n 个变量的 2–SAT 公式 f，其有 m 个子句 C_i, 我们构造如下的图 $G = (V, E)$:

- $V = \{v_x, v_{\neg x} \mid x \text{ 是 } f \text{ 的一个变量}\}$.
- E 中包含这样的边:
  - 对于每个子句 C_i，如果 $C_i = (c \vee c')$，则 E 中包含边 $(v_{\neg c}, v_{c'})$ 和 $(v_{\neg c'}, v_c)$。

图 G 具有如下的性质：如果存在一条 v_x 到 v_y 的路径，则必然存在一条 $v_{\neg y}$ 到 $v_{\neg x}$ 的路径。我们断言，如果 G 中存在某个强连通分量 C，使得 C 中同时包含了 v_x 和 $v_{\neg x}$，则 f 不可满足；否则 f 可满足。

事实上，如果 C 中同时包含了 v_x 和 $v_{\neg x}$，则存在一条从 v_x 到 $v_{\neg x}$ 的环。反设 f 是可满足的，考虑任何一个使其可满足的赋值，其一定将 x 赋值为 True 或者 False，不妨设为 True，令 v_x 到 $v_{\neg x}$ 的环上的顶点依次如下:

$$v_x, v_{x_1}, \dots v_{x_k}, v_{\neg x}, v_{y_1}, \dots v_{y_l}, v_x$$

注意到 $(v_x, v_{x_1}) \in E$，由定义：$\neg x \vee x_1$ 是 f 的一个子句，由 f 是可满足的，可得 $\neg x \vee x_1$ 为真，从而 x_1 必须为真。同理可得 $x_2, \dots, x_k$ 都为真，从而 $v_{\neg x}$ 到 v_x 的环上的顶点都为真，从而 $\neg x$ 为真，与 x 为真矛盾，因此 f 不可满足。

另一方面，如果 G 中不存在一个强连通分量同时包含了 v_x 和 $v_{\neg x}$，则我们可以通过如下的方式构造一个相应的使 f 满足的赋值:

- 对于每个变元 x，如果存在 v_x 到 $v_{\neg x}$ 的路径，则令 x 为 False；如果存在 $v_{\neg x}$ 到 v_x 否则令 x 为 True；我们记这一部分为真的文字为 $Var = \{x_1, \dots, x_k\}$(也就是说如果 x 赋值为 0，我们就添加 $\neg x$ 进去)
- 对于剩余的变元 x，如果存在 Var 中对应的点到 v_x 的路径，则赋值为 True; 如果存在 Var 中对应的点到 $v_{\neg x}$ 的路径，则赋值为 False。
- 若还剩余变元，则对其随意赋值。

我们证明上述赋值是使 f 满足的。事实上我们只需要证明上述赋值时不会矛盾的即可，因为其满足了所有边对应的析取子句。这由下述事实保证：

- 如果在第一步中，x 和 y 被赋值，不妨设为 $False$ 和 $True$。如果此时存在一条 v_y 到 v_x 的路径，则必然存在一条 $v_{\neg x}$ 到 $v_{\neg y}$ 的路径，从而存在一个 $v_x \to *v_{\neg x} \to^* v_{\neg y} \to^* v_y \to^* v_x$ 的环，矛盾。
- 如果在第二步中，对于某个变元 z；其从第一步赋值的变元 x, y 出发会得到不同的赋值结果，即令 x 和 y 分别被赋为 $False$ 和 $True$，则存在如下的两条路径：

  $$v_x \to^* v_{\neg x} \to^* v_{\neg z},\ v_{\neg y} \to^* v_y \to^* v_z$$

  则由上述的性质，我们存在如下的圈：

  $$v_x \to^* v_{\neg x} \to^* v_{\neg z} \to^* \ v_{\neg y} \to^* v_y \to^* v_z \to^* v_x$$

  形成矛盾。

注意到构造 G 的过程是多项式的，而求解强连通分量的过程也是多项式的，因此我们可以在多项式时间内求解 $2-SAT$ 问题。 □

5. 证明如果有人能想办法为 SAT 问题设计出一个多想试时间算法，那么 $NP = P$。

解答. 注意到 $P \subseteq NP$，只要证明 $NP \subseteq P$。由题设存在一个多项式时间确定算法 Π 解决 SAT 问题。下证对于 NP 中的任一问题 L，存在一个多项式时间确定算法 Π_L 解决 L，注意到由于 SAT 问题是 NP 完全的，从而 $L \propto_{poly} SAT$，即存在一个多项式时间的归约，因此可设计如下 Π_L：

(1) 对于任意 L 的输入 x，构造出对应的 SAT 问题实例 ϕ_x，满足 $x \in L$ 当且仅当 ϕ_x 可满足。

(2) 使用算法 Π 求解 ϕ_x，如果 ϕ_x 可满足，则接受，否则拒绝。

由于归约和算法 Π 均为多项式时间，因此 Π_L 是多项式时间的，从而 $L \in P$，因此 $NP \subseteq P$。 □