

一种判定树同构的 BULT 算法^{*}

温雪莲, 梁华金

(中山大学 计算机科学系, 广东 广州 510275)

摘要: 将判定两棵树的同构问题转化成“图的同构”问题和“两棵树根结点之间的对应关系”问题的判定。基于图与树的关系, 提出一种自底向上分层遍历图结点(Bottom-Up Layer Traversing)的方法, 简称 BULT 方法, 解决以上两个问题, 从而得到一种线性的时间复杂度与空间复杂度的树同构判定算法, 并给出了算法正确性证明。该算法很容易扩展为图同构的判定算法。

关键词: 树; 树同构; 时间复杂度; 空间复杂度

中图分类号: TP301.6 **文献标识码:** A **文章编号:** 0529-6579(2005)06-0024-05

本文讨论的图都是有限、简单、无向和无环的连通图, 下面只简称图。

树是带有一个根结点的图^[1]。对树算法的研究是图论与算法中的热点问题, 其中包括对树同构问题^[2,3]、子树同构问题^[4,5]、树的比较问题^[6]和生成树算法^[7,8]的研究。

两棵树同构, 当且仅当这两棵树根结点的孩子之间存在一一对应关系, 并且以孩子结点为根的子树也同构^[3], 判定两棵树的同构问题可转化成二个相关问题的判定, 即: “图的同构”问题和“两棵树根结点之间的对应关系”问题。本文提出的自底向上层次遍历结点(Bottom-Up Layer Traversing)的树同构判定算法(下面简称 BULT 算法)就是通过解决这上述 2 个问题得到。

1 算法的主要思想

为了清晰地描述 BULT 算法的主要思想, 我们引入一些新概念: 结点的层次、图的层数、第 i 层结点集、层次划分、结点的生成树和第 i 层子树集。

定义 1 对图 G , $\forall v \in V(G)$, 结点 v 的层次 $L_G(v)$ 递归定义如下:

- (1) 若 $\deg_G(v) \in \{0, 1\}$, 则 $L_G(v) = 1$;
- (2) 若 $\deg_G(v) \in \{0, 1\}$, 则 $L_G(v) = k$, 其中 $G' = G - \{u \mid L_G(u) < k \text{ 且 } k \geq 1\}$, $v \in V(G')$ 。

定义 2 图 G 的层数 $L(G) = \max(\{L_G(v) \mid v \in V(G)\})$ 。

定义 3 图 G 的第 i ($1 \leq i \leq L(G)$) 层结点集

$$V_G(i) = \{v \mid v \in V(G) \text{ 且 } L_G(v) = i\}.$$

定义 4 对图 G , 称 $\{V_G(1), V_G(2), \dots, V_G(L(G))\}$ 为图 G 的层次划分。

由定义 1 和定义 4 可知: $V_G(1) \cup V_G(2) \cup \dots \cup V_G(L(G)) = V(G)$, 且 $V_G(i) \cap V_G(j) = \emptyset, i, j = 1, 2, \dots, L(G)$ 且 $i \neq j$ 。

定义 5 对图 G , $\forall v \in V(G)$, 结点 v 的生成树 $\text{Tree}_G(v)$ 是指以 v 为根、以 $\text{Tree}_G(v_1), \dots, \text{Tree}_G(v_k)$ 为子树的树。其中, $v_j \in \{v_i \mid v_i \in E(G) \text{ 并且 } L_G(v_i) < L_G(v)\}, j = 1, 2, \dots, k$ 。

由定义 5 可知: 当 $L_G(v) = 1$ 时, $\text{Tree}_G(v) = G[\{v\}]$; 当 $L_G(v) > 1$ 时, $\text{Tree}_G(v) = G[\{v\} \cup V(\text{Tree}_G(v_1)) \cup \dots \cup V(\text{Tree}_G(v_k))]$, 其中: $v_j \in \{v_i \mid v_i \in E(G) \text{ 且 } L_G(v_i) < L_G(v)\}, j = 1, 2, \dots, k$ 。

定义 6 图 G 的第 i 层子树集 $TS_G(i)$ 为: $TS_G(i) = \{\text{Tree}_G(v) \mid L_G(v) = i\}$ 。

由上述定义可以得到以下结论。

引理 1 任意一个图只有唯一的层次划分。

引理 2 任意一个图的最高层结点集只能包含一个或 2 个结点。

证明 下面用反证来证明。

假设: 存在一个图 G , 其最高层结点集至少包含 3 个结点: v_1, v_2 和 v_3 。令 $G' = G - \{v_1, v_2, v_3\}$ 。又由 G 的连通性、定义 1 和定义 5 可知: G' 也是连通的。因此, 至少存在一个结点 v_i ($i \in \{1, 2, 3\}$), 使得: $d_G(v_i) = 2$ 。不妨假设, $d_G(v_3) = 2$, 即: $v_3 v_1 \in E(G'), v_3 v_2 \in E(G')$ 。由 G' 无环性

* 收稿日期: 2004-12-06

基金项目: 广东省科技厅工业攻关资助项目(A10103)

作者简介: 温雪莲(1979年生), 女, 博士生; E-mail: wenxuelian_sysu@yahoo.com.cn

可知: $d_G(v_1)=d_G(v_2)=1$ 。由定义 1 可知: v_1, v_2 与 v_3 不可能在同一层上, 假设相矛盾。引理 2 得证。

引理 3 任意 2 个同构图 G_1 和 G_2 , 若函数 $f: V(G_1) \rightarrow V(G_2)$ 是 G_1 和 G_2 的同构映射, 则有: $V_{G_2}(k)=\{f(v)|v \in V_{G_1}(k)\}, 1 \leq k \leq L(G_1)$ 。

证明 下面对 $k (1 \leq k \leq L(G_1))$ 用归纳法进行证明。由定义 2 和 G_1 与 G_2 同构可知: $L(G_1)=L(G_2)$ 。

- (1) 当 $k=1$ 时, 由 G_1 和 G_2 同构易得 $V_{G_2}(1)=\{f(v_i)|L_{G_1}(v_i)=1\}$ 。
- (2) 假设: 当 $k < n$ 时, 结论成立。
- (3) 下面用反证法证明: 当 $k=n$ 时, 命题也成立。假设: $\{f(v)|v \in V_{G_1}(n)\} = V_{G_2}(n)$ 不成立。不妨再假设 $\{f(v)|v \in V_{G_1}(n)\} \subseteq V_{G_2}(n)$ 不成立。任取一个结点 $v' \in \{f(v)|v \in V_{G_1}(n)\} - V_{G_2}(n)$ 。由 G_1 和 G_2 同构可知: 存在 $v \in V_{G_1}(n)$, 满足: $f(v)=v'$ 。

假设: $L_{G_2}(v')=m$ 。则 $m \neq n$ 。下面分 $m < n$ 和 $m > n$ 两种情况证明:

- (3.1) $m < n$
由归纳假设可知: $V_{G_2}(m)=\{f(v)|v \in V_{G_1}(m)\}$ 。所以, $L_{G_1}(v)=m$, 与“ $v \in V_{G_1}(n)$ ”矛盾。

- (3.2) $m > n$
由定义 1 可知: G_2 中存在一个结点 v'_1 , 满足: $L_{G_2}(v'_1)=L_{G_2}(v)-1$, 且 $v'_1 v' \in E(G_2)$ 。

则有: $L_{G_2}(v'_1)=m-1 \geq n$ 。
令 $v'_1=f(v_1)$, 则有: $v_1 \in V(G_1)$ 。由 G_1 与 G_2 同构和归纳假设可知: $v_1 v \in E(G_1)$ 且 $L_{G_1}(v_1) \geq n$ 。

根据 $m > n$ 和同构图具有相同的层数可知: 结点 v 不是 G_1 的最高层结点。所以, $L_{G_1}(v_1) > n$ 。

又由定义 1 可知: v_1 是 G_1 中唯一满足: $L_{G_1}(v_1) > n$ 且 $v_1 v \in E(G_1)$ 的结点。

进而, 由 G_1 与 G_2 同构和 $f(v)=v'$ 可知: v_1 是 G_2 中唯一满足: $L_{G_2}(v'_1) \geq n$, 且 $v'_1 v' \in E(G_2)$ 的结点, 而至少有一个层数大于 n 的结点 (v') 与 v'_1 相连。因此 $L_{G_2}(v'_1) \geq L_{G_2}(v')$, 与“ $L_{G_2}(v'_1)=L_{G_2}(v)-1$ ”矛盾。

由 (3.1) 和 (3.2) 两种情形可知: $\{f(v)|v \in V_{G_1}(n)\} \subseteq V_{G_2}(n)$ 。同理可证: $V_{G_2}(n) \subseteq \{f(v)|v \in V_{G_1}(n)\}$ 。因此, $\{f(v)|v \in V_{G_1}(n)\} = V_{G_2}(n)$ 。

因此, $\{f(v)|v \in V_{G_1}(n)\} = V_{G_2}(n)$ 。

定理 1 任意两个图 G_1 和 G_2 同构的充要条件是: $L(G_1)=L(G_2)$, 并且当 $1 \leq k \leq L(G_1)$ 时, 都存在双射函数 $f_k: TS_{G_1}(k) \rightarrow TS_{G_2}(k)$ 满足: $\forall T \in TS_{G_1}(k)$ 有: T 与 $f_k(T)$ 同构。

证明

(1) 充分性

令 $w=L(G_1)$ 。

由前提和引理 2 可知: $|V_{G_1}(w)|=|V_{G_2}(w)|=1$ 或者 $|V_{G_1}(w)|=|V_{G_2}(w)|=2$ 。

下面根据 $|V_{G_1}(w)|$ 的值分 2 种情况讨论:

- (1.1) 若 $|V_{G_1}(w)|=|V_{G_2}(w)|=1$

不妨设 $V_{G_1}(w)=\{r_1\}$, $V_{G_2}(w)=\{r_2\}$, 则 $TS_{G_1}(w)=\{Tree_{G_1}(r_1)\}$, $TS_{G_2}(w)=\{Tree_{G_2}(r_2)\}$ 。

由前提条件可知: $Tree_{G_1}(r_1)$ 与 $Tree_{G_2}(r_2)$ 同构。由定义 5, G_1 和 G_2 同构。

- (1.2) 若 $|V_{G_1}(w)|=|V_{G_2}(w)|=2$

不妨设 $V_{G_1}(w)=\{v_{11}, v_{12}\}$, $V_{G_2}(w)=\{v_{21}, v_{22}\}$; $Tree_{G_1}(v_{ij})=T_{ij}, (i, j=1, 2)$ 。

不失一般性, 令 $f_w(T_{1j})=T_{2j}, j=1, 2$; 则 T_{1j} 与 T_{2j} 同构。

不妨设 f_{wj} 为 T_{1j} 与 T_{2j} 之间的同构映射, $j=1, 2$; 如下构造双射函数 $f': V_1 \rightarrow V_2$

$$f'(v)=\begin{cases} f_{w1}(v) & v \in V(T_{11}) \\ f_{w2}(v) & v \in V(T_{12}) \end{cases}$$

下面证明: f' 是 G_1 和 G_2 的同构映射。

由于 f_{wj} 是 T_{1j} 与 T_{2j} 之间的同构映射, $V(T_{11}) \cap V(T_{12}) = \emptyset, V(T_{21}) \cap V(T_{22}) = \emptyset, V(T_{11}) \cup V(T_{12}) = V(G_1)$, 并且 $V(T_{21}) \cup V(T_{22}) = V(G_2)$, 因此 f' 是 $V(G_1)$ 到 $V(G_2)$ 的双射函数。则还需证明: $\forall v_1 v_2 \in E(G_1)$ 有: $f'(v_1)f'(v_2) \in E(G_2)$ 。下面根据 G_1 中任意边 $v_1 v_2$ 所在的位置分 3 种情况讨论。

- (1.2.1) $v_1 v_2 \in E(T_{11})$

则 $f'(v_1)=f_{w1}(v_1) \in V(T_{21}), f'(v_2)=f_{w1}(v_2) \in V(T_{21})$ 。又由 f_{w1} 是 T_{11} 和 T_{21} 的同构映射和 $v_1 v_2 \in E(T_{11})$ 可知: $f'(v_1)f'(v_2)=f_{w1}(v_1)f_{w1}(v_2) \in E(T_{21})$ 。因此, $f'(v_1)f'(v_2) \in E(G_2)$ 。

- (1.2.2) $v_1 v_2 \in E(T_{12})$

与 (1.2.1) 同理可得: $f'(v_1)f'(v_2) \in E(G_2)$ 。

- (1.2.3) $v_1 v_2 = v_{11} v_{12}$

不失一般性, 设 $v_1=v_{11}, v_2=v_{12}$, 则有 $v_{11} \in$

$V(T_{11}), v_{12} \in V(T_{12})$ 。由 f' 的定义可得: $f'(v_1) = f_{w1}(v_1) = f_{w1}(v_{11}) = v_{21}, f'(v_2) = f_{w2}(v_2) = f_{w2}(v_{12}) = v_{22}$ 。又由定义 1 和 G_2 的连通性可知: $v_{21} v_{22} \in E(G_2)$ 。因此, $f'(v_1)f'(v_2) \in E(G_2)$ 。

综上所述 f' 是 G_1 和 G_2 之间的同构映射。

因此, G_1 和 G_2 同构。

(2) 必要性

由 G_1 和 G_2 同构可知: 存在 G_1 和 G_2 的一个同构映射 $f: V(G_1) \rightarrow V(G_2)$ 。

不妨任取 G_1 的第 k 层, 由引理 3 可得: $|TS_{G_1}(k)| = |TS_{G_2}(k)|$ 。

假设 $TS_{G_1}(k) = \{T_{11}, T_{12}, \dots, T_{1m}\}, TS_{G_2}(k) = \{T_{21}, T_{22}, \dots, T_{2m}\}$, 不失一般性, 令 T_{ij} 的根结点为 v_{ij} , ($i=1, 2; j=1, \dots, m$)。由引理 3 可知 $f: \{v_{1i} | i=1, \dots, m\} \rightarrow \{v_{2i} | i=1, \dots, m\}$ 为双射函数, 不失一般性, 令 $f(v_{1i}) = v_{2i}, (i=1, \dots, m)$ 。定义双射函数 $f_k: TS_{G_1}(k) \rightarrow TS_{G_2}(k)$ 满足: $f_k(T_{1j}) = T_{2j}, (j=1, \dots, m)$ 。

下面证明: 对任意 $j \in \{1, \dots, m\}$, T_{1j} 与 $f_k(T_{1j})$ (即 T_{2j}) 同构。

由定义 5 可知: $T_{1j} = G_1[V(T_{1j})], T_{2j} = G_2[V(T_{2j})]$ 。

假设 $V(T_{1j}) = \{w_1, w_2, \dots, w_t\}$, 并令 $S = \{f(w_1), f(w_2), \dots, f(w_t)\}$ 。则 $G_2[S]$ 与 T_{1j} 同构。因此, 欲证 T_{1j} 与 T_{2j} 同构只需要证明 $V(T_{2j}) = S$ 。

由 $f(v_{1j}) = v_{2j}$ 和引理 3 可得: $S \subseteq V(T_{2j})$, 下面用反证法证明 $V(T_{2j}) \subseteq S$ 。

假设 $V(T_{2j}) \not\subseteq S$ 不成立, 则由 T_{2j} 连通和 $S \subseteq V(T_{2j})$ 可知: $\exists v_x' \in V(T_{2j}) - S, v_y' \in S$ 满足: $v_x' v_y' \in E(G_2)$ 。

由 G_1 和 G_2 同构可知: $\exists v_x, v_y \in V(G_1)$ 满足: $f(v_x) = v_x', f(v_y) = v_y'$ 且 $v_x v_y \in E(G_1)$ 。由 S 的定义易得: $v_x \notin V(T_{1j}), v_y \in V(T_{1j})$ 。

由定义 5、 $v_x' \in V(T_{2j})$ 和 $L_{G_2}(v_{2j}) = k$ 可知: $L_{G_2}(v_x') < k$, 进而, $L_{G_1}(v_x) < k$ 。又因为 $v_x v_y \in E_1$, 所以, $v_x \in V(T_{1j})$, 这与 “ $v_x \notin V(T_{1j})$ ” 相矛盾。因此, $V(T_{2j}) = S$, 即: T_{1j} 与 T_{2j} 同构。 f_k 为满足定理要求的双射函数。

2 算法的实现步骤

BULT 算法的形式描述如下:

(1) 初始化计数器 $C=0$, 队列 Q_1 和 Q_2 为空;

(2) 计算 T_1 和 T_2 中所有结点的度数;

(3) 将 T_1 和 T_2 中度为 0 或 1 的结点分别加入到 Q_1 和 Q_2 中;

(4) while (Q_1 和 Q_2 非空) do {

(4.1) if (Q_1 和 Q_2 中结点数目不相等) then { write (“ T_1 与 T_2 不同构”); return; }

(4.2) for (Q_1 中每个结点 v) { if (与 v 相连的结点都没有编号) 结点 v 对应的元组为 (0);

else v 的元组由与 v 相连的已编号结点的编号按非递减顺序排列组成; }

(4.3) 用与 (4.2) 相同的方法对 Q_2 中所有结点赋相应的元组;

(4.4) 将 Q_1 和 Q_2 中所有结点对应的元组集合 S_1 和 S_2 分别按非递减顺序排列;

(4.5) if ($S_1 \neq S_2$) then { write (“ T_1 与 T_2 不同构”); return; }

(4.6) while (S_1 和 S_2 非空) do { 将 S_1 和 S_2 中最小元组对应的所有结点编号赋值为 C ;

将 S_1 和 S_2 中所有最小元组删除;

$C = C + 1$; }

(4.7) for (T_1 中每一个未编号的结点 v) $\deg(v) = \deg(v) - Q_1$ 中与 v 相连的结点数目;

(4.8) 用与 (4.7) 相同的方法对 Q_2 中的所有结点赋相应的度数

(4.9) 清空 Q_1 和 Q_2 , 将 T_1 和 T_2 中所有度为 0 或 1 且未编号的结点添加到 Q_1 和 Q_2 中;

(5) }

(6) if (T_1 和 T_2 根结点编号相同) write (“ T_1 与 T_2 同构”);

else write (“ T_1 与 T_2 不同构”);

根据 BULT 算法对图 1 中的结点分层, 所得到

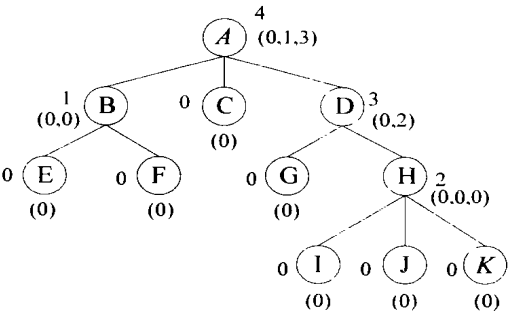


图 1 图 G 结点的编号

Fig 1 The numbering of each vertex in graph G

的分层结果如下：

- 第一层结点：C, E, F, G, I, J, K
- 第二层结点：B, H
- 第三层结点：D
- 第四层结点：A

3 算法正确性的证明

BULT 算法在步骤(4.2)和(4.6)对 T 中结点 v 赋相应的元组和编号，用 $Vctr(v)$ 和 $Num_T(v)$ 分别表示结点 v 的元组和编号。

定理 2 假设树 T_1 和 T_2 对应的图分别为 G_1 和 G_2 ，任取 $v_1 \in V(G_1)$, $v_2 \in V(G_2)$ ，则 $Tree_{G_1}(v_1)$ 和 $Tree_{G_2}(v_2)$ 同构，当且仅当 $Num_{T_1}(v_1) = Num_{T_2}(v_2)$ 。

证明

令 $t_1 = Tree_{G_1}(v_1)$, $t_2 = Tree_{G_2}(v_2)$ 。

(1) 充分性

下面对 $Num_{T_1}(v_1)$ 和 $Num_{T_2}(v_2)$ 的值用归纳法证明： t_1 与 t_2 同构。

(1.1) 当 $Num_{T_1}(v_1) = Num_{T_2}(v_2) = 0$ 时显然： t_1 与 t_2 同构。

(1.2) 假设当 $Num_{T_1}(v_1) = Num_{T_2}(v_2) < k$ 时， t_1 和 t_2 同构。

(1.3) 当 $Num_{T_1}(v_1) = Num_{T_2}(v_2) = k$ 时
由算法的步骤(4.6)和 $Num_{T_1}(v_1) = Num_{T_2}(v_2) = k$ 可知： v_1 的孩子结点和 v_2 的孩子结点数目相等。不妨设 $\{v_{11}, v_{12}, \dots, v_{1m}\}$ 和 $\{v_{21}, v_{22}, \dots, v_{2m}\}$ 分别是 v_1 和 v_2 的孩子结点集。假设对 v_1 和 v_2 对应的元组分别是 $(d_{11}, d_{12}, \dots, d_{1m})$ 和 $(d_{21}, d_{22}, \dots, d_{2m})$ 。

由步骤(4.2)~(4.6)可知： $d_{11} = d_{21} \leq d_{12} = d_{22} \leq \dots \leq d_{1m} = d_{2m} < k$ 。

不失一般性，假设 $Num_{T_1}(v_{1i}) = d_{1i}$, $Num_{T_2}(v_{2i}) = d_{2i}$ 。则有 $Num_{T_1}(v_{1i}) = Num_{T_2}(v_{2i})$, ($i = 1, \dots, m$)。则由递归假设可知： $Tree_{G_1}(v_{1i})$ 与 $Tree_{G_2}(v_{2i})$ 同构。

因此，由同构定义可得： t_1 和 t_2 也同构。

(2) 必要性

由同构图层数相同和条件 t_1 与 t_2 同构可知： $L_{G_1}(v_1) = L_{G_2}(v_2)$ ，不妨设 $L_{G_1}(v_1) = w$ 。

下面对 w 的值用归纳法证明： $Num_{T_1}(v_1) =$

$Num_{T_2}(v_2)$ ：

(2.1) 当 $w = 1$ 时由算法步骤(4.2)可得：

$Num_{T_1}(v_1) = Num_{T_2}(v_2) = 0$ 。

(2.2) 假设当 $w < k$ 时， $Num_{T_1}(v_1) = Num_{T_2}(v_2)$ 。

(2.3) 当 $w = k$ 时， v_1 的孩子结点数与 v_2 的孩子结点数相等。不妨设 v_1 和 v_2 的孩子结点集分别为 $S_1 = (v_{11}, v_{12}, \dots, v_{1m})$ 和 $S_2 = (v_{21}, v_{22}, \dots, v_{2m})$ 。则存在双射函数 $f: S_1 \rightarrow S_2$ ，使得对任意 $v \in S_1$ ， $Tree_{G_1}(v)$ 与 $Tree_{G_2}(f(v))$ 同构。不失一般性，令 $f(v_{1i}) = v_{2i}$ ，则由递归假设可知： $Num_{T_1}(v_{1i}) = Num_{T_2}(v_{2i})$, ($i = 1, 2, \dots, m$)。因此，经过算法步骤(4.2)~(4.4)后 v_1 和 v_2 对应的元组相同。由算法步骤(4.6)，可得 $Num_{T_1}(v_1) = Num_{T_2}(v_2)$ 。

推论 1 假设树 T_1 和 T_2 对应的图分别为 G_1 和 G_2 ，任取 $v_1 \in V(G_1)$, $v_2 \in V(G_2)$ ，则 $Tree_{G_1}(v_1)$ 和 $Tree_{G_2}(v_2)$ 同构，当且仅当 $Vctr_{T_1}(v_1) = Vctr_{T_2}(v_2)$ 。

证明 由定理 2 和算法第(4.6)步易证。

定理 3 BULT 算法能正确判定两棵树是否同构。

证明 由树与图的关系可知：判定两棵树的同构问题可转化成(1)“判定两个图的同构性”问题和(2)“确定两棵树根结点之间的对应关系”问题。

由定理 1 可知：问题(1)可以转化成同层结点生成树的同构问题。由定理 2 和推论 1 可知：同层结点生成树的同构问题可以等价于经过算法步骤(4.2)或(4.6)后，结点所对应的元组或编号相等的问题，因此，问题(1)可以通过执行算法步骤(1)~(4)解决。

由定理 2 可知：问题(2)可以转化为根结点编号是否相等的问题，由算法步骤(6)解决。因此，BULT 算法可以正确判定两棵树是否同构。

4 算法的复杂度分析

假设：树 T_1 和 T_2 对应的图为 $G_1 = (V_1, E_1)$ 和 $G_2 = (V_2, E_2)$ ， $|V_1| + |V_2| = n$ 。

用 BULT 算法判断 T_1 和 T_2 是否同构的时间与空间复杂度分析如下

4.1 时间复杂度

算法所需的时间主要由以下 2 部分组成：

(1) 计算两棵树中结点度数和将两棵树第一层

结点放入队列中的时间为 $O(|V_1| + |V_2|)$ (见步骤 (2—3))

(2) 判断 G_1 和 G_2 相同层结点的生成树同构所花的时间 (见步骤 (4))

步骤 (4) 对 G_1 中第 k ($1 \leq k \leq L(G_1)$) 层结点所做的操作包括下面 3 个部分:

① 赋元组 (步骤 (4.2))

任选 $v \in V_{G_1}(k)$ 对结点 v 赋元组所需的时间为: $O(d_{G_1}(v))$, 所以, 对 G_1 第 k 层中所有结点赋元组所需要的时间为: $O(\sum_{v \in V_{G_1}(k)} d_{G_1}(v))$.

② 赋编号 (步骤 (4.4) ~ (4.6))

共需 $O(\sum_{v \in V_{G_1}(k)} d_{G_1}(v))$ 的时间。

③ 更新结点度数 (步骤 (4.7))

共需 $O(|V_{G_1}(k)|)$ 时间。

因此, 判断 G_1 和 G_2 所有层结点生成树同构 (算法中步骤 (4)) 需花费的总时间为:

$$O(\sum_{k \in L(G_1)} (\sum_{v \in V_{G_1}(k)} d_{G_1}(v) + \sum_{v \in V_{G_2}(k)} d_{G_2}(v) + (|V_{G_1}(k)| + |V_{G_2}(k)|))) = O(|V_1| + |V_2|)$$

由 (1) 和 (2) 部分可知: 本算法的时间复杂度为 $O(|V_1| + |V_2|)$, 即 $O(n)$ 。

4.2 空间复杂度

BULT 算法主要需要的空间有: 树结构链表和当前层结点数组, 总共为 $O(n)$ 。

5 结 语

BULT 算法具有很好的扩展性。该算法先判定图的同构性, 再确定根结点的对应关系。因此, 若要判定两个图的同构性, 则只需在 BULT 算法中去掉对根结点对应关系的判定部分。

采用树同构的方法对图同构性的判断需要先确定图的中心, 然后再判断以中心为根的树的同构性, 该方法的时间复杂度也为 $O(n)$ 。但当图有两个中心时, 可能需要经过 4 次树同构的判断, 而本文所提出的自底向上的算法不需要确定图的中心, 且只需进行一次树同构的判定。

参考文献:

- [1] AHO A V, HOPCROFT J E, ULLMAN J D. The Design and Analysis of Computer Algorithms [M]. New York: Addison-Wesley 1974.
- [2] BUSS S R. A logtime algorithms for tree isomorphism, comparison and canonization [C] // Proceedings of the 5th Kurt Gödel Colloquium on Computational Logic and Proof Theory, 1997: 18—33.
- [3] JENNE B, MCKENZIE P, TORÁN J. A note on the hardness of tree isomorphism [C] // Thirteenth Annual IEEE Conference on Computational Complexity (Buffalo, NY, 1998), 101—105.
- [4] GROSSI R. A note on the subtree isomorphism for ordered tree and related problems [J]. Information and processing letters 1991, 49(2): 81—84.
- [5] GIBBONS P B, SOROKER D, KARP R M, et al. Subtree isomorphism is in random NC [C] // VLSI Algorithm and Architecture (Corfu, 1988): 43—52, LNCS.
- [6] LINDELL S. A logspace algorithm for tree canonization [C] // Proceedings of the 24th Annual ACM symposium on theory of computing, 1992: 400—404.
- [7] BERNARD C, RONITT R, LUCA T. Approximating the minimum spanning tree weight in sublinear time [J]. SIAM J Comput, 2005, 34(6): 1370—1379.
- [8] HASSIN R, LEVIN A. Approximation algorithms for quickest spanning tree problems [J]. Algorithmica, 2005, 41(1): 43—52.

A BULT Algorithm for Tree Isomorphism

WEN Xue-lian LIANG Hua-jin

(Department of Computer Science, Sun Yat-sen University, Guangzhou 510275, China)

Abstract: Solving the tree isomorphism problem is equivalent to solve two problems: whether there exists a bijection between two graphs and whether the bijection maps on distinguished node in one graph to one distinguished node in the other graph. Based on the relation between graphs and trees, a bottom up layer traversing algorithm, BULT algorithm for short, with linear time complexity and space complexity to solve these two problems is given. And the correctness of the algorithm is also proven in the paper. The result can be easily extended to graphs.

Key words: tree; tree isomorphism; time complexity; space complexity