

⑧

树的定义及遍历

48-52

殷新春

叶志敏

TP311.12

(数学与计算机科学系)

(人事处)

A

摘要 比较了国内外数据结构教材中有关一般树的定义及一般树(及森林)的遍历算法,提出了一种统一的、规则的、实用的定义及遍历算法。

关键词 树, 森林, 数据结构, 教材, 遍历

中图法分类号 G423.3, TP392

数据

计算机

树是一种重要的数据结构,无论从教学的角度还是从应用的角度看,有关树的概念、存储、操作等内容都是必不可少的。然而,在国内外一些普遍使用的数据结构教材中,对树的定义有不同的处理方法和概念描述。本文结合笔者^[1, 2]编写教材的体会,介绍我们使用的概念以及对遍历一般树、森林的定义,以求教于大方。

1 树的定义

1.1 递归定义

在数据结构教材中,如国内普遍使用的清华大学严蔚敏教授等编著的教材^[3],将树定义成结点的非空有限集,也就是说,一般树是非空的。

定义 1^[3] 树是 $n(n > 0)$ 个结点的有限集。在任意一棵树中: 1) 有且仅有一个特定的称为根的结点; 2) 当 $n > 1$ 时,其余结点可分为 $m(m > 0)$ 个互不相交的有限集, T_1, T_2, \dots, T_m , 其中每一个集合本身又是一棵树,并且称为根的子树。

绝大多数教材都持这种观点,如复旦大学施伯乐教授等编著的教材^[4],中国科学技术大学唐策善教授等编著的教材^[5],国内早期使用较普遍的两种教材——国防科学技术大学王广芳教授等编著的教材^[6],Horowitz 等编著的教材^[7]及其它教材^[8~10]。

定义 2^[4~10] 树是由一个结点或多个结点组成的有限集 T , 它满足下面两个条件:

1) 有一个特定的结点,称之为根结点。

2) 其余的结点分成 $m(m \geq 0)$ 个互不相交的有限集 T_1, T_2, \dots, T_m , 其中每个集合又都是一棵树,称 T_1, T_2, \dots, T_m 为根结点的子树。

本文于 1995 年 4 月 26 日收到。

本文曾在全国第六届数据结构与组织学术会议上交流。

定义3^[11] 1) 一个单一的结点是树, 该结点即该树的根.

2) 设 n 是一个结点, T_1, T_2, \dots, T_k 是根分别为 n_1, n_2, \dots, n_k 的树, 我们可以构造一棵新树, 使 n 为结点 n_1, n_2, \dots, n_k 的双亲. 在这棵新树中, n 为根, T_1, T_2, \dots, T_k 为 n 的子树, 结点 n_1, n_2, \dots, n_k 称为结点 n 的孩子.

1.2 用图定义树

另一种定义方法是先介绍图后介绍树, 将树定义成一种特殊的图. 如离散数学的教材^[12], 国外的一些数据结构教材^[13, 14].

定义4^[12, 13] 设 $T = \langle V, E \rangle$ 是一个连通无向图, 如果 T 中没有任何循环, 则称 T 是棵树.

定义5^[14] 树是满足下列条件的连通有向图: 1) 存在唯一顶点, 称为根, 其入度为 0; 2) 其它所有顶点的入度为 1.

众所周知, 图中顶点集是非空的、有限的, 因而一般树也是非空的.

1.3 不定义空树引起的问题

前面的各种定义, 都没有定义空树, 有些教材或试题则将能否为空作为一般树与二叉树的区别之一. 然而, 不定义空树会导致操作困难甚至矛盾.

以严蔚敏教授等编著的教材^[3]为例, 至少下列几处引用了空树的概念: 1) 森林可定义成空的, 而树是森林的特例. 2) 树的基本操作中, 运算 $\text{INITIATE}(T)$ 置 T 为空树, $\text{CLEAR}(T)$ 置 T 为空树.

因此, 一些教材^[5, 11] 采用折衷的方法, 在树的定义之后加一段注释: “为方便起见, 也将结点数为 0 的空集合定义为空树.”

空树是平凡的, 在离散数学中基本不作考虑, 但从程序设计和数据结构的角度看, 我们认为应引进空树的概念, 以保证数据结构内容体系的完整性、连贯性, 而且也方便算法的描述与实现.

1.4 树的新定义

定义6^[1, 2, 15] 树是 $n(n \geq 0)$ 个结点的有限集. $n = 0$ 时称为空树. 在任一非空树 ($n > 0$) 中, 1) 有且仅有一个称为根的结点; 2) 其余结点可分为 $m(m \geq 0)$ 个互不相交的子集 T_1, T_2, \dots, T_m , 其中每一个集合本身又是一棵树, 并且称为根的子树.

2 树及森林的遍历

树及森林的遍历是一个有趣且较复杂的问题, 大多数教科书不作介绍, 即使介绍也不统一. 一般教材都依照这样的次序讲授: 二叉树的遍历 \rightarrow 二叉树与树 (及森林) 的转换 \rightarrow 树 (及森林) 的遍历.

2.1 树的两种遍历

设树 T 中, R 为根, 根的子树从左到右依次是 T_1, T_2, \dots, T_k .

树的前序遍历方法定义为^[3, 5, 6]: 若 T 非空, 则 1) 访问根结点 R ; 2) 依次按树的前序遍历根 R 的各子树 T_1, T_2, \dots, T_k .

树的后序遍历方法定义为^[3, 5, 6]: 若 T 非空, 则 1) 依次按树的后序遍历根 R 的各子树 T_1, T_2, \dots, T_k ; 2) 访问根结点 R (注: 在 [6] 中, 以上定义称为树的中序遍历).

由于一般树 (记为 T) 唯一对应一棵二叉树 (记为 T'), 以上定义导致如表 1 的对应关系.

表 1 T 与 T' 的遍历

二叉树 T'	前序	中序	后序
一般树 T	前序	后序	(无)

2.2 森林的两种遍历

森林的前序遍历方法定义为^[3, 5, 6]: 若森林非空, 则可按下述规则遍历之: 1) 访问森林中第一棵树的根结点; 2) 前序遍历第一棵树中根结点的子树森林; 3) 前序遍历除去第一棵树之后剩余的树构成的森林.

森林的中序遍历方法定义为^[3, 5, 6]: 若森林非空, 则可按下述规则遍历之: 1) 中序遍历森林中第一棵树的根结点的子树森林; 2) 访问第一棵树的根结点; 3) 中序遍历除去第一棵树之后剩余的树构成的森林 (注: 在文献 [5] 中, 以上定义称为森林的后序遍历).

设森林为 F , 与之对应的二叉树为 T , 以上定义导致如表 2 的对应关系.

表 2 T 与 F 的遍历

二叉树 T	前序	中序	后序
一般树 F	前序	中序	(无)

2.3 树及森林的三种遍历^[1, 2]

一些教材指出: 与二叉树的后序遍历对应的树或森林的遍历不存在. 然而, 事实上, 设 $F = \{T_1, T_2, \dots, T_m\}$ 是森林, T_1 的根为 R , R 的子树构成的森林为 $\{T_{11}, T_{12}, \dots, T_{1m}\}$, 对森林的遍历可以定义如下:

2.3.1 前序遍历

- 1) 若 F 为空集, 则返回.
- 2) 访问 F 中第一棵树 (T_1) 的根 R ; 按森林的前序, 遍历第一棵树的子树森林 $\{T_{11}, T_{12}, \dots, T_{1m}\}$; 按森林的前序, 遍历 F 中其余各树组成的森林 $\{T_2, T_3, \dots, T_m\}$.

2.3.2 中序遍历

- 1) 若 F 为空集, 则返回.
- 2) 按森林的中序, 遍历 $\{T_{11}, T_{12}, \dots, T_{1m}\}$; 访问 T_1 的根 R ; 按森林的中序, 遍历 $\{T_2, T_3, \dots, T_m\}$.

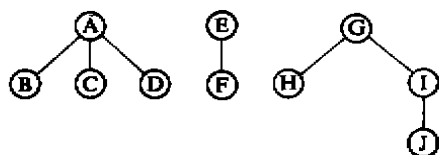
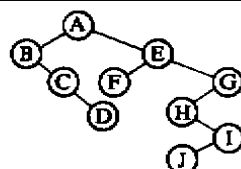
2.3.3 后序遍历

- 1) 若 F 为空集, 则返回.
- 2) 按森林的后序, 遍历 $\{T_{11}, T_{12}, \dots, T_{1m}\}$; 按森林的后序, 遍历 $\{T_2, T_3, \dots, T_m\}$; 访问 T_1 的根 R .

对于一般树, 我们将其看成是森林的特例, 即 $F = \{T_1\}$, 而 T_2, \dots, T_m 均为空树. 因此, 可以将对森林遍历的定义用于对一般树的遍历.

2.4 实例

例 1 设有森林如图 1 所示, 其二叉树如图 2 所示.

图1 森林 F 图2 与 F 对应的二叉树 T

按 2.2 中的定义, F 和 T 的遍历如表 3 所示.

表 3 F 和 T 的遍历

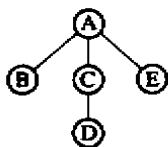
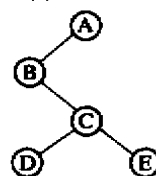
	前序	中序	后序
F	ABCDEFGHJIJ	BCDAFEHJIG	无
T	ABCDEFGHJIJ	BCDAFEHJIG	DCBFJIHGEA

按 2.3 中的定义, F 的三种遍历如表 4 所示.

表 4 F 的三种遍历

前序	中序	后序
ABCDEFGHJIJ	BCDAFEHJIG	DCBFJIHGEA

例 2 设有一般树 T 如图 3 所示, 其对应的二叉树 T' 如图 4 所示.

图3 一般树 T 图4 与 T 对应的二叉树 T'

按 2.1 中的定义, T 和 T' 的遍历如表 5 所示.

表 5 T 和 T' 的遍历

	前序	中序	后序
T	ABCDE	(无)	BDCEA
T'	ABCDE	BDCEA	DECBA

按 2.3 中的定义, T 的三种遍历如表 6 所示.

表 6 T 的三种遍历

前序	中序	后序
ABCDE	BDCEA	DECBA

从以上两例可以看出, 按照我们的定义, 树(或森林)与其对应二叉树的三种遍历存在自然而统一的遍历次序, 这样可大大方便学生的理解, 简化教学工作.

参 考 文 献

- 1 殷新春. 数据结构. 上海: 同济大学出版社, 1990.97,106
- 2 朱静华, 殷新春. 数据结构题例分析. 武汉: 华中理工大学出版社, 1995.106,121
- 3 严蔚敏, 吴伟民. 数据结构. 第2版. 北京: 清华大学出版社, 1992.118, 137~138
- 4 施伯乐等. 数据结构. 上海: 复旦大学出版社, 1988.169, 178~179
- 5 唐策善, 黄刘生. 数据结构. 合肥: 中国科学技术大学出版社, 1992.107~108, 138~139
- 6 王广芳等. 数据结构. 长沙: 湖南科学技术出版社, 1983.105, 124~125
- 7 Horowitz E, Sahni S. Fundamentals of data structures. New York: Computer Science Press Inc, 1976. 220
- 8 李平. 数据结构. 北京: 电子工业出版社, 1986.63
- 9 霍义兴等. 实用数据结构. 上海: 上海科学技术出版社, 1987.137~138
- 10 Aaron M T, Moshe J A. Data structures using Pascal. Englewood Cliffs, New Jersey: Prentice-Hall Inc, 1986. 293
- 11 Alfred V Aho et al. Data structures and algorithms. California: Addison-Wesley Publishing Company, 1983.75, 78
- 12 王遇科. 离散数学基础. 北京: 国防工业出版社, 1982.315
- 13 Thomas A S. Data structures techniques. California: Addison-Wesley Publishing Company, 1980.126
- 14 Michael B F. Data structures with Modula-2. Englewood Cliffs, New Jersey: Prentice-Hall Inc, 1990. 183
- 15 潘道才. 数据结构. 成都: 成都电讯工程学院出版社, 1988.118

DEFINITION AND TRAVERSING OF TREE

Yin Xinchun

(Department of Mathematics and Computer Science)

Ye Zhimin

(Personnel Division)

Abstract This paper compares definitions and traversing algorithms of trees (as well as forests) in all kinds of textbooks on data structure. A more general, regular and practical definition and traversing algorithm of tree is presented.

Key words Tree, Forest, Data structure, Textbook, Traversing