Journal of Automata, Languages and Combinatorics 2 (1997) 3, 161–175

NEW RESULTS ON THE STACK RAMIFICATION OF BINARY TREES

MARKUS E. NEBEL
Fachbereich Informatik, Johann Wolfgang Goethe-Universität, D-60054 Frankfurt am Main, Germany
e-mail: nebel@sads.informatik.uni-frankfurt.de

ABSTRACT

The stack-size of a tree T is the number of cells of a stack needed to traverse T in postorder. In this paper we show that the average number of proper subtrees having the same stack-size as the whole tree is asymptotically 1 with a variance of $2 + o(1)$. The total number of subtrees with a stack-size one less than that of the whole tree is identical to 2. Counting only maximal subtrees changes this number to $1 + o(1)$ with a variance of $o(1)$.

Keywords: analysis of algorithms, combinatorial problems.

1. Introduction and fundamental definitions

Let $T = (I, L, r)$ be an (extended) binary tree [9, p. 399] with the set of internal nodes I, the set of leaves L, and the root r. We call $|T| := |I|$ the *size* of T. Choosing a node $v \in I \cup L$ we denote by T_v the subtree of T with the root v. For $v \in I$ we denote by v_l (resp. v_r) the left (resp. right) son of v. Let $r, v_1, v_2, \ldots, v_i$ be the path from the root to node v_i and let p be some predicate defined on $I \cup L$. If $p(v_i) = \mathsf{true} \wedge \forall j \in [1 : i[\colon p(v_j) = \mathsf{false}$ we call T_{v_i} *maximal* w.r.t. the predicate p.

Each expression consisting of brackets, binary operators and operands may be represented by a binary tree (*syntax tree*) where the operands are the labels of the leaves and the internal nodes represent the operators. For example, the expressions $E_1 := x_1 \;/\; ((x_2 - x_3) \uparrow ((x_4 + x_5) * x_6))$ and $E_2 := x_1 \;/\; ((x_2 - x_3) \uparrow x_4 + x_5 * x_6)$ correspond with the trees T_1 and T_2 of Figure 1, respectively.

A well known strategy to evaluate the corresponding expression from its syntax tree is the postorder traversal of the tree using a stack [6, pp. 130 ff.]. The *stack-function* $S \colon I \cup L \to \mathbb{N}$ of a binary tree $T = (I, L, r)$ is defined by

$$S(v) := \begin{cases} 1 & \text{if } v \in L, \\ \max(S(v_l), S(v_r) + 1) & \text{if } v \in I. \end{cases}$$

$S(v)$ is the maximum number of nodes stored in the stack during the postorder traversal of the subtree T_v [2, 6]. For a detailed description of the relation between the