Global Journal of Science Frontier Research
Mathematics and Decision Sciences
Volume 13 Issue 10 Version 1.0 Year 2013
Type : Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 2249-4626 & Print ISSN: 0975-5896

Regular Recursion Trees: Description and Theoretical Analysis

By Ulyanov M. V. & Goloveshkin V. A.
The National Research University

Abstract- Recursive algorithms analysis by generated trees node counting refers to detailed study of their structures. In this connection the paper presents the specific description of peculiar to a number of recursive algorithms regular trees. We have developed the method which is based on introduced regular description and provides an analytical solution for a number of generated nodes at such regular tree each level. The results obtained make possible a theoretical time complexity analysis of recursive algorithms generating regular recursion trees.

GJSFR-F Classification : MSC 2010: 03D65, 03D60

REGULAR RECURSION TREES DESCRIPTION AND THEORETICAL ANALYSIS

Strictly as per the compliance and regulations of :

Regular Recursion Trees: Description and Theoretical Analysis

Ulyanov M. V. [α] & Goloveshkin V. A. [σ]

Abstract - Recursive algorithms analysis by generated trees node counting refers to detailed study of their structures. In this connection the paper presents the specific description of peculiar to a number of recursive algorithms regular trees. We have developed the method which is based on introduced regular description and provides an analytical solution for a number of generated nodes at such regular tree each level. The results obtained make possible a theoretical time complexity analysis of recursive algorithms generating regular recursion trees.

I. Introduction

The rational algorithm selection problem solution in software development as a rule is based on one of the main procedure quality characteristics called the time complexity. The time complexity is understood to be a number of assumed computing model elementary operations specified at the concrete input by the algorithm [1]. The estimate of higher value is a time complexity function with the size of algorithm input as the function argument. Meaningfully the time complexity function is considered in the best, worst and average cases [2]. Existing methods of algorithms analysis produce algorithm computational complexity – the time complexity function asymptotical estimate in the worst case and in some cases – the time complexity true function to solve the rational choice problem in real input sizes ranges. The time complexity true function derivation is a highly topical problem both for iterative and recursive algorithm solutions in which terms of theoretical study several methods are suggested at the present time. The methods produce analytical expressions of time complexity functions [1, 2, 3]. The generated recursion tree analysis technique is one of advanced methods in this direction. The most complicated step of it is obtaining of analytic expressions for number of generated recursion tree nodes at each tree level [1].

The main problem of recursion tree theoretical study is that in common case a number of generated nodes is the function of both the tree level number and a number of the generating node at this level. Moreover, in some cases such a function is not given analytically and the tree structure is markedly caused by specifics of concrete algorithm input, so the problem becomes significantly complicated. As an example, we consider a recursive algorithm of the solution of the classical problem of optimal packing by the dynamic programming method for which a generated tree is defined by both packing size

Author α : Professor, D.Sc. in Engineering (Russian Scientific Degree is a Doctor of Technical Sciences (д.т.н.)) The National Research University "Higher School of Economics", Moscow State University of Printing Arts.

Author σ: Professor, D.Sc. in Engineering (Russian Scientific Degree is a Doctor of Technical Sciences (д.т.н.)) Moscow State University of Instrument Engineering and Informatics.

Ref

1. Goloveshkin V. A., Ulyanov M. V. Recursion theory for programmers. — M.: FIZMATLIT. 2006. 296.

and cargo types sizes [1]. Only solutions for complete m-ary trees are trivial to this problem.

Thus in the recursive algorithms study aspect the problem of generated recursion trees theoretical study in nontrivial cases is to be of evident interest.

II. Problem Setting Up

Notes

According to generated recursion tree nodes counting based time complexity analysis method [1, p. 181-185.] we will use the following notation:

— n is the depth of the recursion tree, i.e. the number of its levels, in addition the base of the tree is not numbered and has the specific index Root and the further numbering starts with zero;

— S is the formal description of generated recursion tree structure;

— $R_A(n,S)$ is the total node number for all n-deep levels of S structured recursion tree;

— $R_V(n,S)$ is the tree internal nodes total number;

— $R_L(n,S)$ is the terminal node number, i.e. the node number at the level n.

The article has two objects in view:

1. To offer the formal description S of the recursion tree having the regular structure that is typical of generating recursive algorithms, including the algorithms that implement dynamic programming method.
2. To develop a solution procedure that produces the analytical solution of the tree nodes number at the level $R(j,S)$ from the present tree structure formal description S and the given level number j. Thus if the current algorithm input produces the tree with the depth n and the structure S then the main tree characteristics which are necessary for the recursive algorithm time complexity analysis are calculated as

$$R_A(n,S)=\sum_{j=0}^{n}R(j,S)+1;$$

$$R_V(n,S)=\sum_{j=0}^{n-1}R(j,S)+1; \qquad (1)$$

$$R_L(n,S)=R(n,S).$$

Note that the vertex of the recursion tree root is accounted for the complementary unity.

III. Formal Description of the Regular Recursion Tree Structure

Let us start generating of the description of the structure from consideration of complete m-ary tree. It is plain, that in this case each node at each level (besides leaf level) produces evenly m new tops and the tree is completely self-similar starting from any node. Because features of some recursive algorithms are that the tops of the same level produce different amount of the next level nodes that is however less then some prescribed value we propose the following regular tree formal description as ordered m-ary tuple (S line) with non-ascending sorted unit values

$$S = (k_1; k_2; \ldots; k_j; \ldots; k_m),\ j = \overline{1, m};$$
$$k_1 \geq k_2 \geq \ldots \geq k_j \geq \ldots \geq k_m. \qquad (2)$$

Notes

The S line structured tree is generated by the following algorithm using the specific node numeration within the level and the concept of node type. At the tree root level the **Root** node generates m tops of the level $n = 0$ using $1, 2, \ldots, m$ numeration. Let us assign the characteristic to each node – a number from the interval $[1, \ldots, m]$ – and call it the node type. We will say further that if a node has the type "j" then it generates k_j tops at the tree next level. At the level $n = 0$ the nodes generated by the root get the same of their number type, so the j-numbered node has the type "j" and thus generates k_j tops at the level 1. Next, at the next levels the generated nodes are numbered relative to the parent node starting from the number m from right to left in the order of numbers decreasing, and get the types that correspond to assigned numbers. Thus the nodes generated directly from the "j"-typed top (there are k_j of such nodes) are described by the following line

$$(0; 0; \ldots; k_l; k_{l+1}; \ldots; k_m),\ l = m - k_j + 1. \qquad (3)$$

The proposed formal description as the line S (2) specifies the regular tree, i.e. the tree conforming to single at all levels regularity of nodes generating defined by the node type according to node numeration and generating way given in the formula (3). We note that in case of $k_1 = m$ the root has the type "1" and the line S is given by $S = (m; k_2; \ldots; k_j, \ldots; k_m)$ and describes a regular incomplete (truncated) m-ary tree.

For example, the formal description $S = (3; 3; 3)$ specifies a complete ternary tree and the description $S = (3; 3; 2)$ specifies the regular incomplete ternary tree illustrated in fig. 1. The tree nodes are marked by their numbers coinciding to their types in accordance to described numeration principle. Thus all "3"-typed nodes generate two tops at the next level because the formal description line $S = (3; 3; 2)$ includes $k_3 = 2$. Also let us remark here that the tree root level has the **Root** index and is not numbered and the next tree levels have numbers 0, 1 etc. We note that the node type introduced above corresponds to the top number in the formal description line $S = (k_1; k_2; \ldots; k_j; \ldots; k_m)$ and the "j"-typed node generates k_j tops which description line is specified in the formula (3).

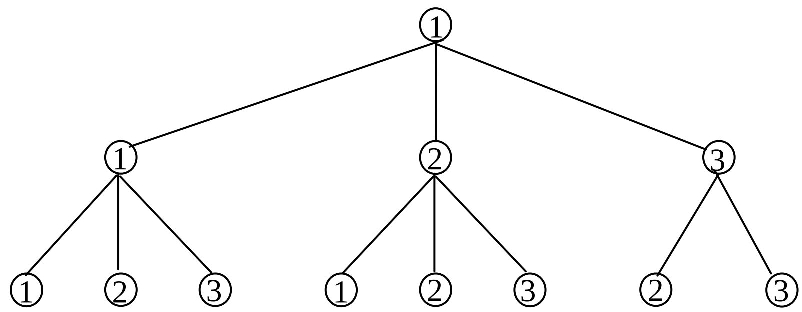

Fig. 1 : The fragment of the tree specified by the description $S = (3;\, 3;\, 2)$ with node types marked.

Notes

IV. Method of the Number of Regular Tree Nodes Problem Analytical Solving

Without losing generality we define the nodes number counting problem as the problem of the n-deep tree leaf number counting where the tree is generated by the description $S = \left(k_1;\, k_2;\, \ldots;\, k_j;\, \ldots;\, k_m\right)$, i.e. the counting of the regular tree nodes at the level n – the tree leaf level. With the preceding notation we solve the problem of determining $R(j, S)$ at $j = n$, i.e. the problem of obtaining the analytical solution for $R(n, S) = R_L(n, S)$.

Let us introduce the "j"-typed node characteristic $F(j)$. Meaningfully $F(j)$ is the type number of nodes generating the type "j" by the description S. $F(j)$ value is calculated from the formula

$$F(j) = \max_{1 \le i \le m} \left\{\, i : m - k_i + 1 \le j \,\right\}. \tag{4}$$

Let $w_{n,j}$ is the number of "j"-typed nodes at the tree level n. Let us introduce the vector $W_n = \left\{ w_{n,1};\, w_{n,2};\ldots;w_{n,m} \right\}$ with dimension m. Thus the vector components W_n contain the number of each type nodes at the level n of recursion tree. Then with consideration of the characteristic $F(j)$ introduced above we obtain the following system of linear recurrence relations to calculate the $w_{n,j}$ values

$$\begin{cases} w_{\text{Root},1} = 1, \quad w_{\text{Root},j} = 0,\ j = \overline{2, m}, \quad n = \text{Root}; \\ w_{0,j} = 1,\ j = \overline{1, m}, \quad n = 0; \\ w_{n+1,j} = \displaystyle\sum_{i=1}^{F(j)} w_{n,i}, \quad n \ge 1. \end{cases} \tag{5}$$

We note that in accordance with the (5) $W_0 = \{1; 1; \ldots; 1\}$ and if $k_1 = m$ then $W_{\text{Root}} = \{1;\, 0;\ldots;0\}$

From the known method of linear recurrence relations solving [1, chapter 1], we will try solutions of the relation (5) in the following form

$$\left\{ \lambda \cdot h_{0,j} = \sum_{i=1}^{F(j)} h_{0,i}, \quad \forall j = \overline{1,m}. \right. \tag{7}$$

Our problem required solution – the vector W_n – is given by the linear combination of the computed solutions (6) with consideration of initial conditions from (5).

Let us formulate the suggested method stages for the recurrence relation (5) solving in the case when the number of eigenvectors H_0 is equal to m.

1. Evaluation of the node type characteristics $F(j)$ by the formula (4) from the offered formal description of the regular recursion tree in the form of the line S (2).
2. Formation of the linear recurrence relations system (5).
3. The derivation of the solutions in the form $H_n = \lambda^n H_0$.
   - 3.1. Construction of the equations set (7).
   - 3.2. Defining λ values by the condition of equality of zero and the system determinant (7).
   - 3.3. Finding of eigenvectors H_0 values for every value of λ.
4. Construction of the initial system (5) solutions W_n as a linear combination of the obtained solutions for H_0 from defining unknown coefficients of the linear combination by the values of initial conditions defined by the vector W_0.
5. Counting the number of nodes at the level n of the recursion tree under study

$$R(n,S) = \sum_{j=1}^{m} w_{n,j}. \tag{8}$$

6. Since obtained analytical solution for W_n holds for every value of n the formula (8) gives the solution also for $R(j,S)\,\forall j = \overline{0,n}$ what enables us to define the total number of generated nodes at all n-deep tree levels and the number of the internal nodes according to the formula (1).

We particularly note that in case of the number of eigenvectors H_0 is less than m the known special solving methods [1, 3, 4] is necessarily to be used.

V. Examples of Regular Recursion Tree Analysis

Later we will give the examples of regular trees analysis for both cases: when the number of eigenvectors H_0 is equal to or less than m.

a) At first we consider the example of the tree specified by the description $S = (3; 3; 2)$ which form fig. 1 shows. We point out in advance that the number of eigenvectors H_0 is equal to m in this case. We illustrate the method in details pointing

Ref

3. R. Graham, D. Knuth, Patashnik O. Concrete Mathematics. Foundation of computer science. Translated from English. — M.: Mir. 1998. 703.

directly numbers and content of the stages of suggested solving method.

1. Evaluation of $F(j)$ by the description $S=(3;3;2)$:

$$F(1)=2,\ F(2)=3,\ F(3)=3.$$

2. Formation of the linear recurrence relations system (5):

$$\begin{cases} w_{0,j}=1,\ j=\overline{1,3}; \\ w_{n+1,1}=w_{n,1}+w_{n,2}; \\ w_{n+1,2}=w_{n,1}+w_{n,2}+w_{n,3}; \\ w_{n+1,3}=w_{n,1}+w_{n,2}+w_{n,3}. \end{cases}$$

3.1. Construction of the equations set (7):

$$\begin{cases} \lambda h_{0,1}=h_{0,1}+h_{0,2}; \\ \lambda h_{0,2}=h_{0,1}+h_{0,2}+h_{0,3}; \\ \lambda h_{0,3}=h_{0,1}+h_{0,2}+h_{0,3}. \end{cases} \tag{9}$$

3.2. Defining λ values by the condition of equality of zero and the system determinant:

$$\begin{vmatrix} 1-\lambda & 1 & 0 \\ 1 & 1-\lambda & 1 \\ 1 & 1 & 1-\lambda \end{vmatrix}=0.$$

Computing the determinant we obtain the equation $\lambda\cdot(\lambda^2-3\lambda+1)=0$ which roots are:

$$\lambda_1=0;\quad \lambda_2=\frac{3-\sqrt{5}}{2};\quad \lambda_3=\frac{3+\sqrt{5}}{2}.$$

3.3. Finding of eigenvectors H_0 values for every value of λ.

Plugging in (9) the roots obtained and solving obtained systems we get eigenvectors H_0 which corresponds to the secular equation roots by the numeration:

$$H_0^{(1)}=\begin{pmatrix}1\\-1\\0\end{pmatrix};\quad H_0^{(2)}=\begin{pmatrix}1\\ \frac{1-\sqrt{5}}{2}\\ \frac{1-\sqrt{5}}{2}\end{pmatrix};\quad H_0^{(3)}=\begin{pmatrix}1\\ \frac{1+\sqrt{5}}{2}\\ \frac{1+\sqrt{5}}{2}\end{pmatrix}.$$

4. Construction of the initial system (5) solutions W_n as a linear combination of the obtained solutions for H_0 with initial conditions W_0:

$$W_n=A\cdot 0^n\cdot H_0^{(1)}+B\cdot\left(\frac{3-\sqrt{5}}{2}\right)^n\cdot H_0^{(2)}+C\cdot\left(\frac{3+\sqrt{5}}{2}\right)^n\cdot H_0^{(3)},$$

Notes

Notes

where by definition $0^0 = 1$ and A, B, C are linear combination unknown coefficients defined by the initial conditions W_0:

$$\begin{cases} A + B + C = 1; \\ -A + \frac{1-\sqrt{5}}{2}B + \frac{1+\sqrt{5}}{2}C = 1; \\ \frac{1-\sqrt{5}}{2}B + \frac{1+\sqrt{5}}{2}C = 1. \end{cases}$$

The solution of obtained system is of the form:

$$A = 0; \quad B = \frac{\sqrt{5}-1}{2\sqrt{5}}; \quad C = \frac{\sqrt{5}+1}{2\sqrt{5}},$$

that gives the terminal solution for the number of nodes of the regular tree under review

$$W_n = \frac{\sqrt{5}-1}{2\sqrt{5}} \cdot \left(\frac{3-\sqrt{5}}{2}\right)^n \cdot H_0^{(2)} + \frac{\sqrt{5}+1}{2\sqrt{5}} \cdot \left(\frac{3+\sqrt{5}}{2}\right)^n \cdot H_0^{(3)}.$$

5. Counting the number of nodes at the level n of the recursion tree under study

According to (8) the total nodes number at the level n is the sum of each type nodes:

$$R(n, S) = \sum_{j=1}^{3} w_{n,j}.$$

Making the necessary transformations we obtain the desired solution for $S = (3; 3; 2)$:

$$R(n, S) = \frac{7+3\sqrt{5}}{2\sqrt{5}} \cdot \left(\frac{3+\sqrt{5}}{2}\right)^n + \frac{3\sqrt{5}-7}{2\sqrt{5}} \cdot \left(\frac{3-\sqrt{5}}{2}\right)^n.$$

It may be verified by direct substitution that $R(1, S) = 8$, $R(2, S) = 21$.

б) The second example where the regular tree is given by the description $S = (3; 3; 1)$ also gives three different eigenvectors. In this case we will give the main results only. The values of λ obtained by the condition of equality of zero and the system determinant are

$$\lambda_1 = 0; \quad \lambda_2 = 1; \quad \lambda_3 = 2.$$

that gives the following solution for corresponding typed nodes number

$$w_{n,1} = 2^n; \quad w_{n,2} = 2^n; \quad w_{n,3} = 2 \cdot 2^n - 1.$$

Summing over all types we obtain the solution for the tree level

$$R(n, S) = 2^{n+2} - 1.$$

Notes

в) The next example produces the case that needs the specific solving method. Let us consider the description $S=(m;\, m-1;\, m-2;\ \ldots;1)$. Fig. 2 shows the tree that corresponds to this description at $m=3$ – $S=(3;\,2;\,1)$.

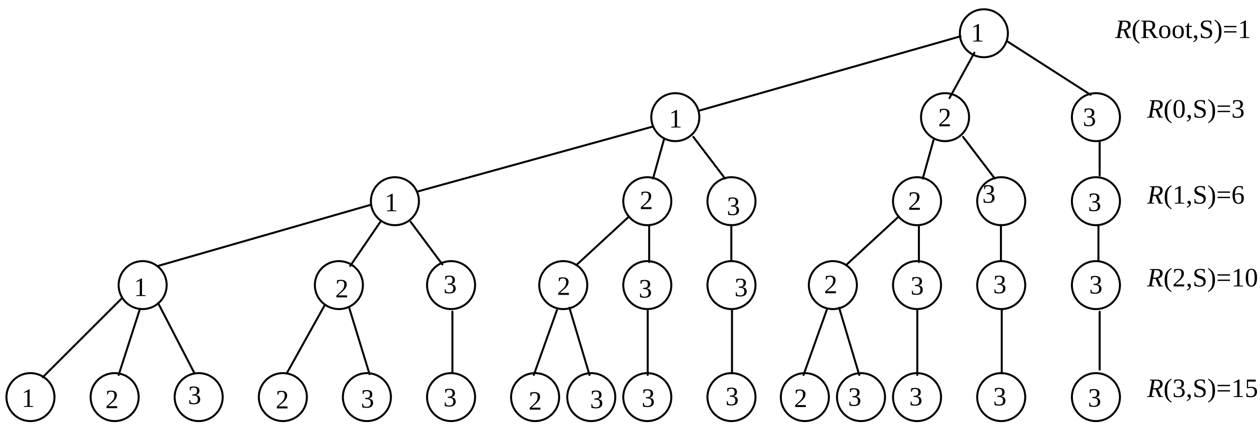

Fig. 2 : The fragment of the tree specified by the description $S=(3;\,2;\,1)$ with node types marked.

The initial stages of this case analytical solution are the same as for the method described above:

1. Evaluation of $F(j)$ by the description $S=(m;\, m-1;\, m-2;\ \ldots;1)$:

$$F(1)=1,\ F(2)=2,\ F(3)=3,\ldots,F(m)=m\,.$$

2. Formation of the linear recurrence relations system (5):

$$\begin{cases} w_{0,j}=1,\ j=\overline{1,3}; \\ w_{n+1,1}=w_{n,1}; \\ w_{n+1,2}=w_{n,1}+w_{n,2}; \\ \ldots \\ w_{n+1,j}=w_{n,1}+w_{n,2}+\ldots+w_{n,j}; \\ \ldots \\ w_{n+1,m}=w_{n,1}+w_{n,2}+w_{n,3}+\ldots+w_{n,m}; \end{cases} \tag{10}$$

3.1. Construction of the equations set (7):

$$\begin{cases} \lambda h_{0,1}=h_{0,1}; \\ \lambda h_{0,2}=h_{0,1}+h_{0,2}; \\ \lambda h_{0,3}=h_{0,1}+h_{0,2}+h_{0,3}; \\ \ldots \\ \lambda h_{0,m}=h_{0,1}+h_{0,2}+h_{0,3}+\ldots+h_{0,m}. \end{cases} \tag{11}$$

3.2. Defining λ values by the condition of equality of zero and the system determinant:

Calculating the determinant of the system (11) we get the equation $(1-\lambda)^m = 0$ and thus the multiplicity of the root $\lambda = 1$ is m. In this problem the number of eigenvectors is less then m and we use the specific method that provides for the direct solving of the system (10).

Notes

4. The direct solving of the system (10).

4.1. Let us solve the recurrence relation for $j = 1$:

$$\begin{cases} w_{0,1} = 1; \\ w_{n+1,1} = w_{n,1}. \end{cases} \Rightarrow \forall n \; w_{n,1} = 1.$$

4.2. From the solution obtained we get the recurrence relation for $j = 2$:

$$\begin{cases} w_{0,2} = 1; \\ w_{n+1,2} = w_{n,2} + 1. \end{cases} \Rightarrow \forall n \; w_{n,2} = n + 1.$$

4.3. With consideration of this solution the recurrence relation for $j = 3$ is of the form

$$\begin{cases} w_{0,3} = 1; \\ w_{n+1,3} = w_{n,3} + (n+1) + 1. \end{cases} \quad (12)$$

We seek the specific solution in the form $w^*_{n,3} = An^2 + Bn + C$ and plugging it in (12) we get

$$2An + A + B = n + 2, \quad \Rightarrow \quad A = \frac{1}{2}, B = \frac{3}{2},$$

the initial condition gives the solution $C = 1$ and finally

$$w_{n,3} = \frac{1}{2}n^2 + \frac{3}{2}n + 1 = \frac{(n+1)(n+2)}{2}.$$

4.4. We note that $w_{n,1} = 1 = C_n^0$, $w_{n,2} = n + 1 = C_{n+1}^1$, $w_{n,3} = C_{n+2}^2$. Let us prove by induction that for an arbitrary value m

$$w_{n,m} = C_{n+m-1}^{m-1}. \quad (13)$$

The relation (13) is true for $m = 1, m = 2, m = 3$. By the induction hypothesis, the relation $w_{n,k} = C_{n+k-1}^{k-1}$ is true for all $k \le m$. Then by proceeding the linear recurrence relations system (10) for $m+1$ we obtain the recurrence relation

$$\begin{aligned} w_{n+1,m+1} &= w_{n,1} + w_{n,2} + w_{n,3} + \ldots + w_{n,m} + w_{n,m+1} = \\ &= C_n^0 + C_{n+1}^1 + C_{n+2}^2 + \ldots + C_{n+m-1}^{m-1} + w_{n,m+1}. \end{aligned} \quad (14)$$

We note that $C_n^0 = C_{n+1}^0$, $C_n^k + C_n^{k-1} = C_{n+1}^k$ [3], then

$$C_n^0 + C_{n+1}^1 + C_{n+2}^2 + \ldots + C_{n+m-1}^{m-1} = C_{n+2}^1 + C_{n+2}^2 + \ldots + C_{n+m-1}^{m-1} = C_{n+m}^{m-1}.$$

We show that

$$w_{n,m+1} = C_{n+m}^m$$

is the solution of the recurrence to n relation (14). Plugging this putative solution in (14) we get

$$w_{n+1,m+1} = C_{n+m}^{m-1} + C_{n+m}^m = C_{n+m+1}^m,$$

that corresponds to (13) with substituting for corresponding indexes. Thus the solution of the system (10) $\forall m \geq 1, \forall n \geq 1$ is of the form

$$w_{n,m} = C_{n+m-1}^{m-1},$$

that proves the inductive hypothesis (13).

From the solutions obtained we define the total nodes number at the tree level numbered n by summing the number of all type nodes at this level. We introduce the third argument m into the function of nodes number since we are concerned with generic description $S = (m; m-1; m-2; \ \ldots; 1)$ and using the known relations for the binomial coefficients [3] we get:

$$R(n,m,S) = \sum_{j=1}^{m} w_{n,j} = \sum_{j=1}^{m} C_{n+j-1}^{j-1} = C_{n+m}^{m-1}. \tag{15}$$

From (15) let us define the total nodes number of the tree having description $S = (m; m-1; m-2; \ \ldots; 1)$ by summing the number of nodes at all levels including the tree root

$$R_A(n,m,S) = 1 + \sum_{i=0}^{n} R(i,m,S) = \sum_{i=0}^{n} C_{i+m}^{m-1} = C_{n+m+1}^m. \tag{16}$$

Referring to the tree fragment shown in fig. 2 and having the description $S = (3; 2; 1)$, i.e. $m = 3$ we obtain the total nodes number at the level $n = 3$ by using the formula (15) – $R(3, 3, S) = C_{3+3}^{3-1} = C_6^2 = 15$ – and define the nodes sum at all levels including the level $n = 3$ from the formula (16) – $R_A(3, 3, S) = C_{3+3+1}^3 = C_7^3 = 35$. We note that the solution obtained by us is the same as the results of the generated recursion tree study of the algorithm of the solution of the optimal one-dimensional packing problem by dynamic programming method at the problem regular parametrization [1]. Notwithstanding the result contained in [1] is particular to this paper results because it gives the solution only for the trees having the description $S = (m; m-1; m-2; \ \ldots; 1)$.

VI. Conclusion

For the sake of recursive algorithm analysis and study this paper suggests the specific description of regular recursion trees typical of some recursive algorithms. Such

Notes

Global Journal of Science Frontier Research (F) Volume XIII Issue X Version I Year 2013

trees are generated for example by the algorithms that realize recursively the dynamic programming method with the special parametrization which regularizes the trees generated by recursive algorithms. Under this description the solution procedure is suggested that gives the number of generated nodes at each level of a such regular tree. The method devised by us produces the analytical solution for the number of generated nodes over every level of the recursion tree and thus the total number of nodes as the function of the tree depth.

Notes

The results obtained can be used for detailed theoretical analysis of recursive algorithms generating regular recursion trees and for getting true functions of time complexity and capacitory efficiency.

Bibliography

1. Goloveshkin V. A., Ulyanov M. V. Recursion theory for programmers. — M.: FIZMATLIT. 2006. 296.
2. T. Corman, C. Leiserson, R. Rivest, K. Stein. Algorithms: construction and analysis, Second Edition: Translated from English. — Moscow: Publishing house "Williams", 2005. — 1296.
3. R. Graham, D. Knuth, Patashnik O. Concrete Mathematics. Foundation of computer science. Translated from English. — M.: Mir. 1998. 703.
4. Sachkov V. N. Introduction to Combinatorial methods of discrete mathematics. — Second Edition, Rev. and add. — Moscow: MCCME, 2004. — 424.

Notes

This page is intentionally left blank