Bit-tuple Notation for Trees

Wolfgang R. Müller, Klaus Szymanski, and Jan V. Knop

Computer Centre, The Heinrich Heine University, D-40225 Düsseldorf, The Federal Republic of Germany

Nenad Trinajstić*

The Rugjer Bošković Institute, P. O. Box 1016, HR-41001 Zagreb, The Republic of Croatia

Received February 7, 1994®

A bit-tuple notation, based on the N-tuple concept, is introduced for trees.

Recently in this Journal Rücker and Rücker[1] stated in their article on counts of all walks as atomic and molecular descriptors, "As far as we know, no system for the unique but simple graph-theoretical description of atomic environments has been proposed, not even in tree graphs.... If such a system does exist, the code for a single atom would suffice for reconstruction of the whole tree, and it would be the natural basis for a substructure search system for atoms in tree molecules." However, we introduced such a system for trees in 1981 in this Journal,[2] namely, the N-tuple code for (rooted) trees.

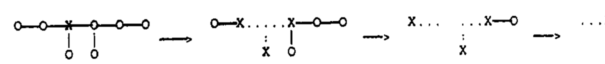
A rooted tree is a tree with one selected vertex, its root.[3] Therefore, a unique description of a rooted tree is at the same time the description of the atomic environment of its root. In our 1981 paper, we define for a rooted tree with N vertices (and hence with $N-1$ edges) its code as a sequence of N nonnegative integer numbers by recursion. To construct the code for a given rooted tree, we first eliminate the root vertex and all of its M incident edges, getting M subtrees with the neighbors of the former root as new roots. Then, we construct the codes for these rooted subtrees and append them in reverse lexicographic order to the 1-tuple (M) to get the code for the given rooted tree. As the subtrees become smaller for every level, the recursion will definitely terminate with cases where $M = 0$, i.e., at the minimum (rooted) tree with only one vertex, which by the same rule gets the code 0. The lexicographic ordering makes this code unique, and obviously the rooted trees are easily reconstructed.

When representing very large graph-theoretical (rooted) trees by N-tuples, there is a practical problem of storage size for the integeres composing the N-tuple. If the storage size is chosen too large, there may be plenty of wasted room, but if chosen too small, this may limit the size of the trees which are otherwise representable. Clearly, this is no problem for chemical trees since the valencies of their vertices are strictly limited. For a general graph-theoretical tree the following notation may help.

The integers which compose the N-tuple and whose sum is $N-1$ are replaced by a sequence of $2N-1$ bits of which $N-1$ are equal to unity and N are equal to zero. This is done in the following way: Each integer k in the N-tuple is replaced by $k+1$ bits of which the first k are 1 and the last is 0, and these bit sequences are concatenated. Obviously, the N-tuple is straightforwardly reconstructed by counting the 1-bits up to the next 0-bit, the sum $N-1$ of the integers implies the total of $N-1$ 1-bits, and the count N of members in the N-tuple leads to N 0-bits. We propose to name this code the "bit-tuple" notation for trees.

The number of bits is small enough to allow moderately sized rooted trees to be represented by a single number, if we

A given rooted tree with a root X is analyzed as shown below:



The N-tuple corresponding the rooted tree is synthesized (tuples of subtrees are ordered lexicographically) as follows:

```
(3 2100 10 0)  <--   (1 0)   (2 10 0)  <--   (0)    (1 0)  <--   (0)
                       (0)                           (0)
```

The N-tuple code:	3	2	1	0	0	1	0	0
and the corresponding bit sequence:	1110	110	10	0	0	10	0	0

The generated bit-tuple may be interpreted as the integer number:

```
 111011010001000

               0 *      1
+             0  *      2
+            0   *      4
+           1    *      8
+          0     *     16
+         0      *     32
+        0       *     64
+       1        *    128
+      0         *    256
+     1          *    512
+    1           *   1024
+   0            *   2048
+  1             *   4096
+ 1              *   8192
+1               *  16384        = 30344
```

or as the fraction:

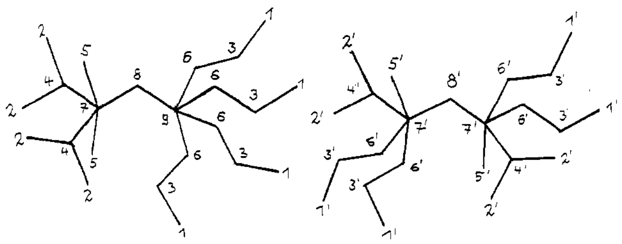
```
 1               * 0.5
+ 1              * 0.25
+  1             * 0.125
+   0            * 0.0625
+    1           * 0.03125
+     1          * 0.015625
+      0         * 0.0078125
+       1        * 0.00390625
+        0       * 0.001953125
+         0      * 0.0009765625
+          0     * 0.00048828125
+           1    * 0.000244140625
+            0   * 0.0001220703125
+             0  * 0.00006103515625
+              0 * 0.000030517578125    = 0.926025390625
```

Figure 1. Computation of the N-tuple for a given rooted tree, the construction of the related bit-tuple, and its conversion into the integer number and the fraction.

---

* Abstract published in *Advance ACS Abstracts*, June 1, 1994.

0095-2338/94/1634-0960$04.50/0



N-tuple codes (sorted lexicographically):

```
1 : 1114142002000011011O110
```

Bit-tuple notations:

Bit-tuples interpreted as integers or fractions:

Figure 2. Illustrative example from a paper by Ivanciuc and Balaban (equivalent vertices are labeled with the same number). Reprinted with permission from ref 4. Copyright 1992 Baltzer.

interpret the bit sequence as an integer binary number or as the binary digits behind the binary point. The latter way of storing contains all necessary information for a reconstruction because of the self-terminating character of the *N*-tuple code (the last position in the *N*-tuple is the first position where the sum of the numbers up to this position is less than the position itself, so that appended zeros do not cause any problem). This quality would also allow one to omit all trailing zero bits, if equal code length for equal tree size is not mandatory. As the last bit is always zero and for all but the trivial tree without edges the first bit is always 1 and the last two bits are always zero, there is a chance of squeezing the bit-tuple notation a little when short of available bits.

In Figure 1 we show how to compute the *N*-tuple of a rooted tree and the corresponding bit-tuple and how to convert the bit-tuple into the integer number and the fraction.

The efficiency of the bit-tuple notation is perhaps best documented by the fact that including the next higher number of vertices requires only two additional bits; i.e., the range of notation values must only be enlarged by a factor of 4, when the total number of rooted trees to be counted grows by a factor of more than 2.5 for vertex counts above 10 (with increasing tendency, e.g., more than 2.9 above 80).

Figure 2 displays an application of the bit-tuple notation to two difficult branched trees with 23 vertices taken from Ivanciuc and Balaban[4] (and discussed also by Rücker and Rücker[1]).

ACKNOWLEDGMENT

N.T. was supported in part by the Ministry of Science and Technology of the Republic of Croatia through Grant 1-07-159. We are thankful to Dr. Christoph Rücker (Freiburg, FRG) for constructive comments on the early version of this paper. We also thank the reviewers for helpful comments.

REFERENCES AND NOTES

(1) Rücker, G.; Rücker, Ch. Counts of All Walks as Atomic and Molecular Descriptors. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 683–695.

(2) Knop, J. V.; Müller, W. R.; Jeričević, Ž.; Trinajstić, N. Computer Enumeration and Generation of Trees and Rooted Trees. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 91–99. See also: Trinajstić, N.; Nikolić, S.; Knop, J. V.; Müller, W. R.; Szymanski, K. *Computational Chemical Graph Theory: Characterization, Enumeration and Generation of Chemical Structures by Computer Methods*; Simon & Schuster: New York, 1991; Chapter 3.

(3) Trinajstić, N. *Chemical Graph Theory*, 2nd revised ed.; CRC Press: Boca Raton, FL, 1992; p 13.

(4) Ivanciuc, O.; Balaban, A. T. Nonisomorphic Graphs with Identical Atomic Counts of Self-Returning Walks: Isocodal Graphs. *J. Math. Chem.* **1992**, *11*, 155–167.