# COUNTING NODES IN BINARY TREES

Sami Khuri
School of Computer & Information Science
Syracuse University
Syracuse, New York 13244

This paper describes an original method for introducing linear recurrence relations. Boolean expressions are represented by binary trees and the counting of the internal nodes of these trees yield linear recurrence relations. The method allows the students to create their own family of boolean expressions, to draw the corresponding binary trees, to deduce the recurrence relation representing the number of nodes in the trees, and finally, to solve and check the solutions of these relations.

key words: linear recurrence relations, boolean expressions, binary trees.

## 1. Introduction

Binary trees have found many applications in computer science such as databases, pattern recognition, taxonomy, decision table programming, analysis of algorithms, switching theory and even in the theoretical design of circuits required for VLSI (Ta81). Any boolean expression of n variables can be represented by a full binary tree of height n.

The study of recurrence relations also, is becoming increasingly important. Hardly any discrete mathematics book is being published nowadays without devoting a large segment to recurrence relations. This topic is either covered in an analysis of algorithms course, or in a discrete mathematics course, whether taught on the Freshman/Sophomore level or the upper one.

This paper presents a method for introducing linear recurrence relations making use of binary trees. The latter are used to represent special kinds of boolean expressions and the counting of the nodes of the trees yields recurrence relations.



## 2. Boolean Expressions and Representations

Given a boolean expression, it can be represented by a mapping into the set {0,1}, by its minterms, by its truth table, by a Karnaugh map or by a full binary tree where each level corresponds to a different variable.

Example: The boolean expression , $A\bar{B} + \bar{A}B\bar{C}$ has the following representations:

1) Mapping:
$F : \{0,1\}^3 \longrightarrow \{0,1\}$ where

$\{0,1\}^3 = \{(a,b,c) \mid a,b,c \in \{0,1\}\}$
$F(1,0,0) = F(1,0,1) = F(0,1,0) = 1$ and
$F(0,0,0) = F(0,0,1) = F(0,1,1)$
$= F(1,1,0) = F(1,1,1) = 0.$

2) Minterms:

$A\bar{B}$, $\bar{A}B\bar{C}$ or (1,0,0), (1,0,1) and (0,1,0).

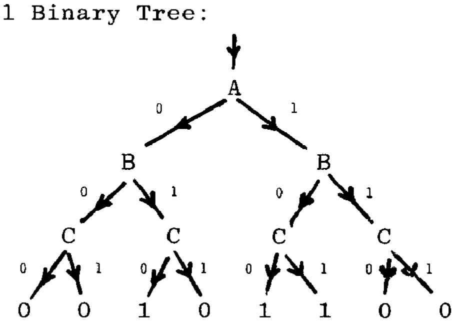
3) Truth Table:

| | |
|---|---|
| A | 0 0 0 0 1 1 1 1 |
| B | 0 0 1 1 0 0 1 1 |
| C | 0 1 0 1 0 1 0 1 |
| $A\bar{B}$ | 0 0 0 0 1 1 0 0 |
| $\bar{A}B\bar{C}$ | 0 0 1 0 0 0 0 0 |
| F | 0 0 1 0 1 1 0 0 |

4) Karnaugh Map:

| | $AB$ | $\bar{A}B$ | $\bar{A}\bar{B}$ | $A\bar{B}$ |
|---|---|---|---|---|
| $C$ | 0 | 0 | 0 | 1 |
| $\bar{C}$ | 0 | 1 | 0 | 1 |

5) Full Binary Tree:



In general, if the boolean expression has n variables, then the full binary tree has $2^n - 1$ internal nodes, $2^n$ external nodes, and is of height n. But, as one can readily see from the above tree, in general the number of nodes can be reduced by trimming the tree. In figure 1, the trimmed tree has 4 internal nodes instead of 7.

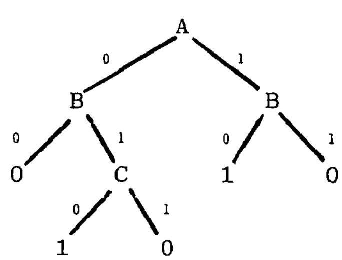


Figure 1.

## 3. Recurrence Relations

Let BE(n,4) represent the set of boolean expressions of n variables and four terms having the following form:

$$X_1X_2\ldots X_{\frac{n}{2}}X_{\frac{n}{2}+1}\ldots X_n + \bar{X}_1\bar{X}_2\ldots\bar{X}_{\frac{n}{2}}X_{\frac{n}{2}+1}\ldots X_n +$$

$$X_1X_2\ldots X_{\frac{n}{2}}\bar{X}_{\frac{n}{2}+1}\ldots\bar{X}_n + \bar{X}_1\bar{X}_2\ldots\bar{X}_{\frac{n}{2}}\bar{X}_{\frac{n}{2}+1}\ldots\bar{X}_n$$

where $n \geq 4$ is even.

Examples

1) n=4 ; $X_1X_2X_3X_4 + \bar{X}_1\bar{X}_2X_3X_4 + X_1X_2\bar{X}_3\bar{X}_4 +$

$\bar{X}_1\bar{X}_2\bar{X}_3\bar{X}_4$

Note that in figure 2, $X_i$ is used at level i for i = 1,2,3,4 in the trimmed tree.

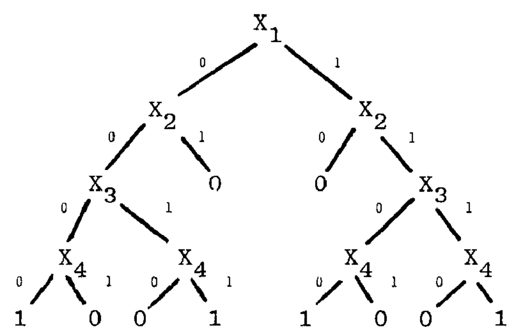


Figure 2.

2) n=6 ; $X_1X_2X_3X_4X_5X_6 + \bar{X}_1\bar{X}_2\bar{X}_3X_4X_5X_6 +$
$X_1X_2X_3\bar{X}_4\bar{X}_5\bar{X}_6 + \bar{X}_1\bar{X}_2\bar{X}_3\bar{X}_4\bar{X}_5\bar{X}_6$.

Figures 2 and 3 are the trimmed tree representations of the boolean expressions in BE(n,4) for n = 4 and n = 6 respectively.

Let T(n) represent the number of internal nodes of the trimmed tree representations of the expressions in BE(n,4).

The shape of these trimmed trees is the same for all the different values of n. There is one root node at level 1, 2 nodes at levels 2,3,4 ... $\frac{n}{2}+1$ and 4 nodes at levels $\frac{n}{2}+2$, $\frac{n}{2}+3$ ... n.

The difference in the number of internal nodes between two consecutive trees corresponding to two consecutive even values of n, is similar to the difference between the number of internal nodes of

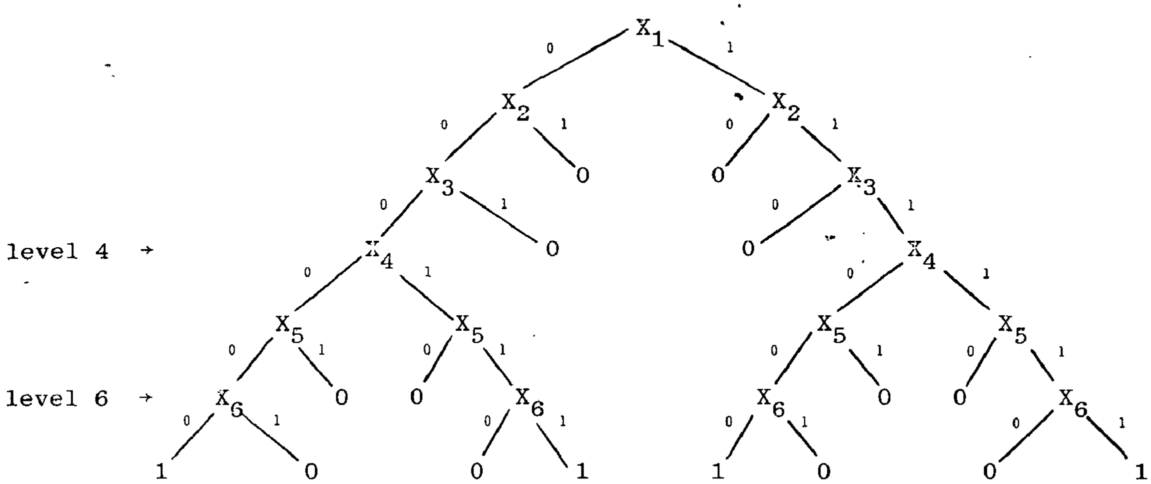



Figure 3.

the trees in figures 2 and 3. The tree in figure 3 has two more levels than the one in figure 2; one that has 2 nodes at level 4 and the other that has 4 nodes at level 6.

So $T(6) = T(4) + 2 + 4$ ; i.e. $T(6) = T(4) + 6$. Indeed, $T(4) = 9$, while $T(6) = 15$.

In general, the tree corresponding to n will have two more nodes at level $\frac{n}{2} + 1$, and four more nodes at level n, than the tree associated with $n - 2$.

So $T(n) = T(n-2) + 6$ for even $n \geq 4$ and $T(2) = 3$.

The above is called a recurrence relation since $T(n)$ is a function of $T(n-2)$. It is also linear since it is of the form $T(n) = aT(n-2) + b$. $T(2) = 3$ is the initial condition.

To solve the relation, consider the following $\frac{n}{2} - 1$ equations,

$$\begin{aligned}
T(n) &= T(n-2) + 6\\
T(n-2) &= T(n-4) + 6\\
&\;\vdots\\
T(4) &= T(2) + 6.
\end{aligned}$$

By summing both sides, one gets $T(n) = T(2) + 6(\frac{n}{2} - 1)$ and finally the closed form $T(n) = 3n - 3$.

Other methods for solving linear recurrence relations are found in (Li77) and (Pu85) for example.

One can get the closed form of $T(n)$ by counting the number of internal nodes of the trimmed tree corresponding to n. The count is done level by level.

| | | | | |
|---|---|---|---|---|
| So | level 1 | has | 1 | node |
| | level 2 | has | 2 | nodes |
| | level 3 | has | 2 | nodes |
| | . | . | | . |
| | level $\frac{n}{2} + 1$ | has | 2 | nodes |
| | level $\frac{n}{2} + 2$ | has | 4 | nodes |
| | level $\frac{n}{2} + 3$ | has | 4 | nodes |
| | . | . | | . |
| | level n | has | 4 | nodes. |

The total number of internal nodes is:

$$T(n) = 1 + 2(\tfrac{n}{2} + 1 - 1) + 4[n - (\tfrac{n}{2} + 2) + 1]$$

And finally $T(n) = 3n - 3$ as expected.

Another interesting example is to consider BE(n,3), obtained from BE(n,4) by dropping the last term. So the boolean expressions of BE(n,3) have the following form:

$$X_1X_2\ldots X_{\frac{n}{2}}X_{\frac{n}{2}+1}\ \ldots\ X_n\ +$$

$$\bar{X}_1\bar{X}_2\ldots \bar{X}_{\frac{n}{2}}X_{\frac{n}{2}+1}\ \ldots\ X_n\ +$$

$$X_1X_2\ldots X_{\frac{n}{2}}\bar{X}_{\frac{n}{2}+1}\ \ldots\ \bar{X}_n.$$

Then $T(n) = T(n - 2) + 5$; $T(2) = 3$

and the closed form is $T(n) = \frac{5}{2}n - 2$, which can be obtained by counting the number of internal nodes in the general trimmed tree, or by obtaining a second order linear recurrence relation by comparing the two trimmed trees representing two successive even values of n and then solving the relation by using methods found in textbooks (which includes the summing method used earlier).

## 4. Conclusion

This paper presented a method for introducing recurrence relations. The students have to use their own creativity to find families of boolean expressions similar to BE(n,4) and BE(n,3), to derive the corresponding recurrence relations and solve and check the solutions by using the two methods discussed in the paper.

It is worthwhile pointing out that the conversion of boolean expressions into trees is by itself an interesting and challenging problem. By removing the restriction of having $X_i$ at level i as was done with the trimmed trees in this paper, one can generally build more efficient trees whether the criteria of efficiency is the height of the tree, the average number of variables per path from the root node to the leaves, or just the number of nodes.

## 5. References


(Li77) C. L. Liu, Elements of Discrete Mathematics. New York: McGraw-Hill Computer Science Series, 1977.

(Pu85) P. W. Purdon, and C. A. Brown, The Analysis of Algorithms. Holt, Rinehart and Winston, 1985.

(Ta81) A. S. Tanenbaum, Computer Networks. New Jersey: Prentice-Hall, Inc., 1981.