On the Halting of Tree Replacement Systems

R. J. Lipton and L. Snyder

Research Report #99

Department of Computer Science
Yale University
New Haven, Connecticut 06520

This work was supported by the Office of Naval Research under grant N00014-75-C-0752.

In algebraic simplification of expressions, code generation and other areas one is often given a finite set of rewriting rules which are to be applied to an expression until no further rules apply. For example,

> Given well-formed arithmetic expressions composed of parentheses and variables with operators of addition (+) and multiplication (x), apply
>
> $$(\alpha + \beta)\times\gamma \to (\alpha \times \gamma) + (\beta \times \gamma) \qquad (1)$$
>
> until no further applications are possible.

We assume that the Greek letters can match any well formed subexpression or variable, and that "apply" has its usual meaning in symbolic manipulation: when ever the left side of the rule matches a subexpression, the subexpression is to be replaced by the right side of the rule with the Greek letters consistently instantiated.

The key phrase in the forgoing problem statement is "until no further applications are possible." This raises a difficult question:

> How do we prove that a particular set of rules halts (i.e. no further applications are possible) for all expressions and all possible sequences of rule applications?

For rule (1) the problem was first answered by Iturriaga [1] and later a simpler proof was presented by Manna and Ness [2]. In this latter paper the standard proof method was used which we call the <u>well ordering</u> method. In particular, given a set of rules, one seeks a measure M on the expressions such that

$$M(\ldots (\alpha + \beta) \times \gamma \ldots) > M(\ldots (\alpha \times \gamma) + (\beta \times \gamma)\ldots).$$

The task of finding a measure M is not always as simple as it might appear -- especially when there are many rules that interact in non-trivial ways. In the present case, if the "local measure" of γ has a large value, then the measure of the entire expression after the rule has been applied may increase rather than decrease, since there are now two copies of γ.

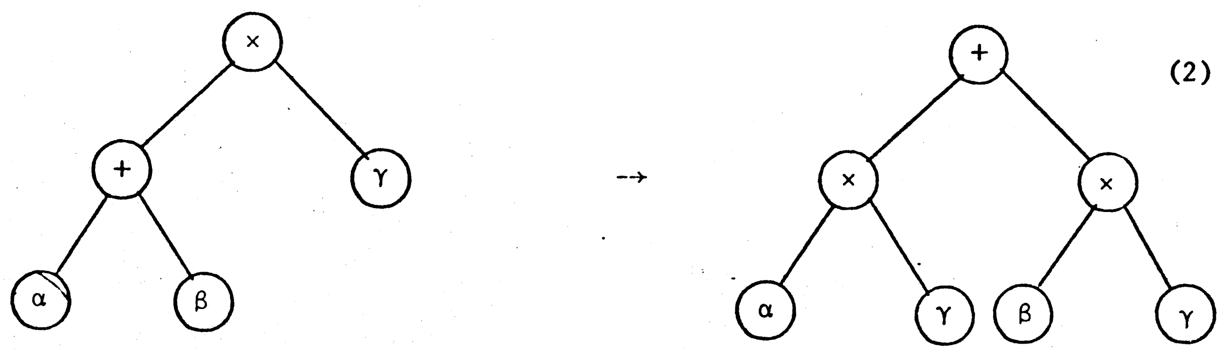
The main result of this paper is to present a new method, the value preserving method, for proving that a set of rules halts. The method is broadly applicable and as the name implies it takes advantage of the fact that rules for symbolic expression manipulation are often intended to be value preserving. Specifically, we prove a theorem stating that two properties of rule sets, value preserving and monotonicity are sufficient to imply that the rule set halts for all expressions and all sequences of rule applications. The exact statement of the general form of the theorem, as well as the proof itself require considerable technical development. For the purposes of this abstract we by-pass the tedious details and concentrate on examples, the generalization of which, we believe, will be clear.

Before proceeding, we note that we are presenting general sufficient conditions for a rule set to halt. The impossibility of finding necessary and sufficient conditions is implied by

Theorem: The problem of determining if a finite set of rules halts is undecidable, even if the set contains as few as 3 rules.

Thus, sufficient conditions are all that can be hoped for.

Turning now to a proof using the value preserving method for the distributive law (1), we assume that the expressions are given as trees (variables at the leaves, operators at non-leaf vertices). The distributive law is then written



(2)

where α, β and γ match any subtree.

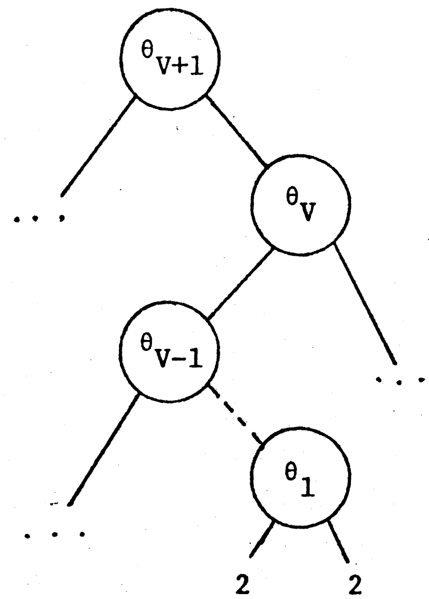
Let E be any expression in this tree form. Replace each leaf in E by the integer constant 2 and let the resulting tree be E'. Evaluate E' in the obvious way and let the resulting number be V. We note that the distributive law preserves this value, i.e. if E'_{i+1} is the result of applying (2) to E'_i then the values of each expression are equal to V.

Assume, for the purpose of contradiction, that (2) does not halt for E'. That is, there exists a sequence

$$E' = E'_0, E'_1, E'_2, \ldots$$

such that E'_{i+1} follows from E'_i by application of (2). If we denote by $\| E'_i \|$ the number of vertices of E'_i, then as $i \to \infty$, $\| E_i \| \to \infty$, i.e. the size of each expression increases without bound. This implies that for some E'_k there is a root-to-leaf path of length V, i.e.

E'_k =



where Θ_j is either + or ×. But by the monotonic properties of + and ×, the subtree rooted at Θ_{i+1} has value greater than the subtree rooted at Θ_i $(1 \le i < V)$ and thus the value of E'_k must be greater than V. Contradiction! The rules must halt. □

The rules rely on the facts

I. Both + and × are monotone in the sense that

$$a + b > \max(a,b)$$

$$a \times b > \max(a,b)$$

for all integers $a,b \ge 2$,

II. If E'_{i+1} follows from E'_i by application of the distributive law, then

$$\| E'_{i+1} \| > \| E'_i \|$$

and their <u>values</u> are equal.

Our theorem proves the substance of the forgoing argument for generalized statements of I and II. All that would be required to prove halting for the distributive law given the theorem would be to find an interpretation in which I and II were satisfied. Informally, the general monotone property can be stated:

> An operator is monotone if its value when applied to operends whose values are members of a subset (not necessarily proper) of its legal arguments yields a result which is (1) in the subset and (2) greater than the value of any of the operands.

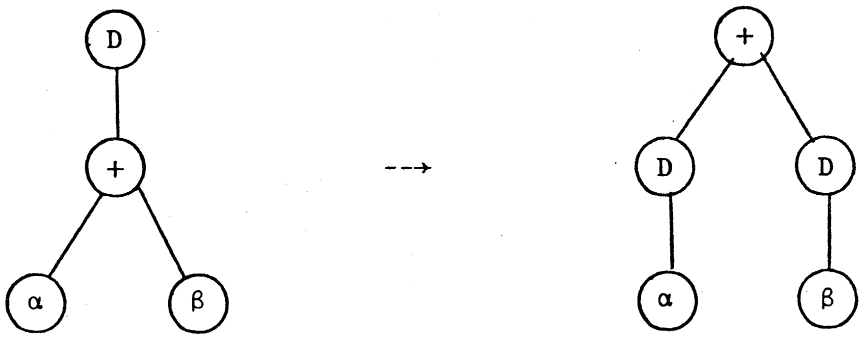
Thus, for the present case the subset is the subset of integers {2,3,4,...} Notice also that the ordering $\|E'_i\| < \|E'_{i+1}\|$ is opposite of the that required by the well ordering method of proof. Thus, the value preserving method in one sense complements the well ordering method in that if the size increases as rules are applied, the value preserving method can be used, but if the size decreases then a well ordering proof is immediate.

As a second example of the value preserving method, consider the problem of showing that for

> arithmetic expressions formed from variables, the binary operators + and ×, and the monodic differential operator D, the rules
>
> $$D(\alpha + \beta) \rightarrow D\alpha + D\beta \qquad (3a)$$
>
> $$D(\alpha \times \beta) \rightarrow (\beta \times D\alpha) + (\alpha \times D\beta) \qquad (3b)$$

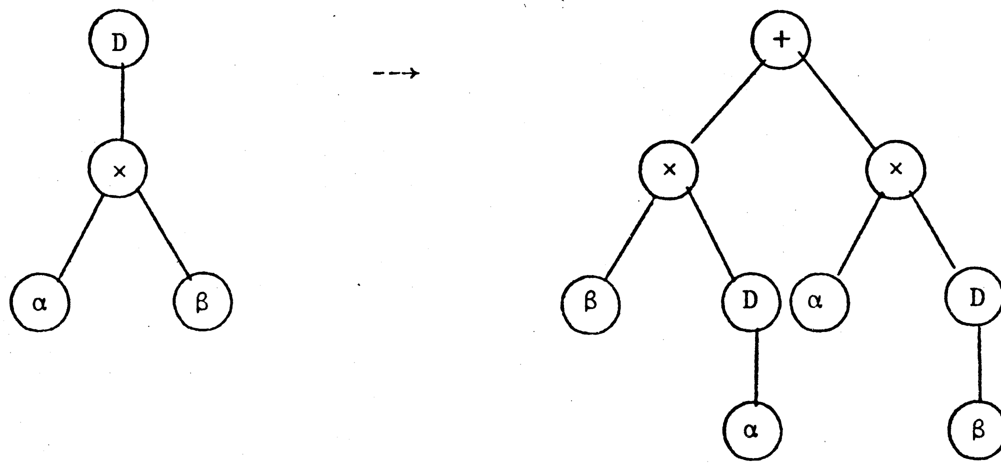
halt.

Following our earlier strategy we express these rules in tree form as



(4a)

and



(4b)

For the interpretation of expression E we select the usual interpretation for +, × and D and we replace the leaves by the functional quantity $2e^{2x}$. The value of the expression E' so interpreted is its value when $x=0$.

For monotonicity it is easy to see that

I. (a) $Df > f$

(b) $f + g > f$ and g

(c) $f \times g > f$ and g

for all f, g of the form $\sum_{i=1}^{n} a_i e^{b_i x}$ where $a_i \geq 2$ and $b_i \geq 2$.

Property II is as before. Indeed, since the interpretation just given holds for the distributive law, it follows that rules (1), (3a) and (3b) taken together must always halt!

In summary we present necessary conditions for a set of rewriting rules to halt for all expressions and all possible sequences of rule applications. The method depends on finding a interpretation in which the value of the expression is preserved under rule application, and in which the operators are monotonic (i.e. the value of an expression is greater than the value of its operands).

References

[1] R. Iturriaga, Contributions to Mechanical Mathematics, Ph.D. Carnegie-Mellon University, 1967.

[2] Z. Manna and S. Ness, On the termination of Marker Algorithms. Proc. of the third Hawaii International Conference on System Sciences, 1970.