ON THE TERMINATION OF MARKOV ALGORITHMS

by

Zohar Manna and Stephen Ness
Computer Science Department
Stanford University

Introduction

We consider the termination problem for Markov-like algorithms, each described by a set of productions and a set of expressions to which the productions may be applied. Common examples of such algorithms include algorithms for:

1. The simplification of symbolic arithmetic expressions, e.g. for rationalizing expressions formed from variables, constants and binary operators + , - , * and / :

$$p_1: v_1+(v_2/v_3) \to (v_1*v_3+v_2)/v_3 \qquad p_2: (v_1/v_2)+v_3 \to (v_1+v_3*v_2)/v_2$$
$$p_3: v_1-(v_2/v_3) \to (v_1*v_3-v_2)/v_3 \qquad p_4: (v_1/v_2)-v_3 \to (v_1-v_3*v_2)/v_2$$
$$p_5: v_1/(v_2/v_3) \to (v_1*v_3)/v_2 \qquad p_6: (v_1/v_2)/v_3 \to v_1/(v_2*v_3)$$
$$p_7: v_1*(v_2/v_3) \to (v_1*v_2)/v_3 \qquad p_8: (v_1/v_2)*v_3 \to (v_1*v_3)/v_2 \ .$$

2. The application of operators to symbolic arithmetic expressions, e.g. for differentiating (with respect to a variable x) expressions formed from variables, constants, binary operators + and * , and the unary operator D :

$$D(v_1+v_2) \to Dv_1+Dv_2 \qquad D(v_1*v_2) \to v_2*Dv_1+v_1*Dv_2$$
$$D\,\langle\text{variable not x}\rangle \to 0 \qquad Dx \to 1$$
$$D\,\langle\text{constant}\rangle \to 0 \ .$$

3. The simplification of symbolic logical expressions, e.g. for converting formulas of the propositional calculus into conjunctive normal form:

$$v_1 \supset v_2 \to \sim v_1 \vee v_2 \qquad v_1 \equiv v_2 \to (v_1 \supset v_2) \wedge (v_2 \supset v_1)$$
$$\sim(\sim v_1) \to v_1$$
$$\sim(v_1 \vee v_2) \to \sim v_1 \wedge \sim v_2 \ , \qquad \sim(v_1 \wedge v_2) \to \sim v_1 \vee \sim v_2$$
$$(v_1 \wedge v_2) \vee v_3 \to (v_1 \vee v_3) \wedge (v_2 \vee v_3) \qquad v_1 \vee (v_2 \wedge v_3) \to (v_1 \vee v_2) \wedge (v_1 \vee v_3) \ .$$

(*) The research reported here was supported in part by the Advanced Research Projects Agency of the Office of the Secretary of Defense (SD-183) and by the National Science Foundation.

Termination

We say that such an algorithm terminates if the process of applying productions repeatedly to any expression eventually must produce an expression to which no production applies, regardless of the choice of which production to use and which subexpression to apply it to at each stage.

It is often convenient to use the notion of transformation sequence of a given expression; e.g. the two transformation sequences of a+((b/c)/d) with respect to the rationalizing algorithm (see Example (1) above) are:

$$a+((b/c)/d) \overset{p_1}{\Rightarrow} ((a*d)+(b/c))/d \overset{p_1}{\Rightarrow} ((((a*d)*c)+b)/c)/d \overset{p_6}{\Rightarrow} (((a*d)*c)+b)/(c*d) ,$$

and

$$a+((b/c)/d) \overset{p_6}{\Rightarrow} a+(b/(c*d)) \overset{p_1}{\Rightarrow} ((a*(c*d))+b)/(c*d) .$$

Then we may say that a Markov-like algorithm terminates if every transformation sequence of every expression is finite.

Proofs of termination

The problem of proving the termination of algorithms like those given above is known to be undecidable in general.[1] It first arose in connection with the Markov-like algorithms of Formula Algol, and was studied at some length by Iturriaga.[2] He found an effective sufficient (but clearly not necessary) condition for determining termination. We have extended his result. In this short paper we present only the flavor of our results by means of examples.

Our proofs of termination use the notion of well-ordered sets. A set W is well-ordered by $\succ$ if $\succ$ is an irreflexive, asymmetric, and transitive relation on W such that any decreasing sequence $x_1 \succ x_2 \succ x_3 \ldots$ of elements of W has only finitely many elements.[3] The most familiar and useful example of a well-ordered set is the set I_n of all n-tuples of natural numbers (for fixed n) with the usual lexicographic ordering, i.e. $(x_1,\ldots,x_n) \succ (x_1',\ldots,x_n')$ if there exists an i, $1 \leq i \leq n$, such that $x_1 = x_1', \ldots, x_{i-1} = x_{i-1}'$, and $x_i > x_i'$. As a special case we have the set of natural numbers I_1 when $n = 1$.

1/ A. A. Markov, The Theory of Algorithms. Akademiya Nauk SSSR, 1954 (Russian). Israel Program for Scientific Translation, 1962 (English).

2/ R. Iturriaga, Contributions to Mechanical Mathematics. Ph.D. Thesis, Carnegie-Mellon University, 1967.

3/ See, for example, P. R. Halmos, Naive Set Theory. Van Nostrand, 1960.

We prove the termination of a given Markov-like algorithm by specifying a well-ordered set W and a mapping φ from the set of all expressions into W, such that $\varphi(e_1) > \varphi(e_2)$ whenever e_1 can be transformed to e_2 by some production. Since all decreasing sequences in a well-ordered set are finite, the above condition implies that all transformation sequences must be finite, i.e. that the algorithm terminates.

To prove the termination, we choose a well-ordered set W, and then assign a monotone increasing (in each variable)[4] function $F_\theta: W^n \to W$ to each n-ary operator θ and a constant $c \in W$ to the atoms of the expression. If for any assignment of value a_i of W to v_i, the function

$$\bar{\varphi}(t) = \begin{cases} c & t \text{ is an atom} \\ a_i & t \text{ is } v_i \\ F_\theta(\bar{\varphi}(t_1),\dots,\bar{\varphi}(t_n)) & t \text{ is } \theta(t_1,\dots,t_n) \end{cases}$$

satisfies $\bar{\varphi}(\tau) > \bar{\varphi}(\tau')$ whenever $\tau \to \tau'$ is a production of our algorithm, then it follows by a simple proof that the algorithm terminates. In practice, one can assign simple unspecified functions to the operators -- e.g. taking W as I_1 and $c \geq 1$, assign a linear function $F_\theta(x) = \alpha x+\beta$ $(\alpha \geq 1,\ \beta \geq 0)$ if θ is a unary operator or $F_\theta(x,y) = \alpha x+\beta y+\gamma\ (\alpha,\beta \geq 1,\ \gamma \geq 0)$ if θ is a binary operator -- and find the required coefficients by solving an appropriate set of inequalities, as illustrated below.

Examples

1. $(v_1+v_2)+v_3 \to v_1+(v_2+v_3)$. (the associative rule)

Choose $W = I_1$. Letting $F_+(x,y) = \alpha x+\beta y+\gamma$, we require that

$$\alpha(\alpha a_1+\beta a_2+\gamma)+\beta a_3+\gamma > \alpha a_1+\beta(\alpha a_2+\beta a_3+\gamma)+\gamma$$

i.e. for every $a_1, a_3 \geq c$,

$$\alpha^2 a_1+\beta a_3+\alpha\gamma > \alpha a_1+\beta^2 a_3+\beta\gamma \quad ,$$

which is satisfied if $\alpha^2 \geq \alpha$, $\beta \geq \beta^2$ and $\alpha\gamma \geq \beta\gamma$, with strict inequality in at least one of the three cases and with constraints $\alpha,\beta \geq 1$, $\gamma \geq 0$. One solution is therefore $\alpha = 2$, $\beta = 1$, $\gamma = 0$, i.e. $F_+(x,y) = 2x+y$, with $c = 1$. We note that Iturriaga's condition is not sufficient to show the termination of this production.

2. The algorithm given above for differentiating.

Choose $W = I_1$. Letting $F_+(x,y) = \alpha x+\beta y+\gamma$, $F_*(x,y) = \alpha' x+\beta' y+\gamma'$ and $F_D(x) = \alpha'' x+\gamma''$, we require (from the second production) that

---

[4] i.e., for any j $(1 \leq j \leq n)$: $x_j > x'_j$ implies $F_\theta(x_1,\dots,x_j,\dots,x_n) > F_\theta(x_1,\dots,x'_j,\dots,x_n)$.

$$\alpha''(\alpha' a_1+\beta' a_2+\gamma')+\gamma'' > \alpha(\alpha' a_2+\beta'(\alpha'' a_1+\gamma'')+\gamma')+\beta(\alpha' a_1+\beta'(\alpha'' a_2+\gamma'')+\gamma')+\gamma$$

i.e.

$$\alpha'\alpha'' a_1+\beta'\alpha'' a_2+(\gamma'\alpha''+\gamma'') > (\alpha\beta'\alpha''+\beta\alpha')a_1+(\alpha\alpha'+\beta\beta'\alpha'')a_2+(\alpha\beta'\gamma''+\alpha\gamma'+\beta\beta'\gamma''+\beta\gamma'+\gamma)$$

for every $a_1, a_2 \geq c$. But the constraint $\alpha,\beta,\alpha',\beta',\alpha'' \geq 1$ implies that $\beta'\alpha'' < \alpha\alpha'+\beta\beta'\alpha''$, so the inequality cannot be satisfied. However, we can prove termination by letting $F_D(x)$ be quadratic rather than linear; a simple check shows that $F_+(x,y) = F_*(x,y) = x+y$, $F_D(x) = x^2$ with $c = 2$ is a solution.

3. $v_1*(v_2+v_3) \rightarrow v_1*v_2+v_1*v_3$
$(v_1+v_2)*v_3 \rightarrow v_1*v_3+v_2*v_3$. (the distributive rules)

Choose $W = I_1$. Letting $F_+(x,y) = \alpha x+\beta y+\gamma$ and $F_*(x,y) = \alpha' x+\beta' y+\gamma'$, we require (from the first production) that

$$\alpha' a_1+\beta'(\alpha a_2+\beta a_3+\gamma)+\gamma' > \alpha(\alpha' a_1+\beta' a_2+\gamma')+\beta(\alpha' a_1+\beta' a_3+\gamma')+\gamma ,$$

i.e.

$$\alpha' a_1+\alpha\beta' a_2+\beta\beta' a_3+\beta'\gamma+\gamma' > (\alpha\alpha'+\beta\alpha')a_1+\alpha\beta' a_2+\beta\beta' a_3+(\alpha+\beta)\gamma'+\gamma ,$$

for every $a_1, a_2, a_3 \geq c$. But the constraint $\alpha,\beta,\alpha',\beta' \geq 1$ implies that $\alpha' < \alpha\alpha'+\beta\alpha'$, so the inequality cannot be satisfied. However, we can prove termination by letting $F_*(x,y)$ be quadratic rather than linear; a simple check shows that $F_+(x,y) = x+y+1$, $F_*(x,y) = x\cdot y$ with $c = 2$ is a solution.

4. The productions given above for rationalizing.

Choose $W = I_1$. Letting $F_+(x,y) = F_-(x,y) = \alpha x+\beta y+\gamma$, $F_*(x,y) = \alpha' x+\beta'\gamma+\gamma'$ and $F_/(x,y) = \alpha'' x+\beta'' y+\gamma''$, we find that the solution $F_+(x,y) = F_-(c,y) = 2x+2y$, $F_*(x,y) = x+y$ and $F_/(x,y) = x+2y+1$ with $c = 1$ gives the desired inequalities for p_1 through p_6 , but gives equality for p_7 and p_8 . Therefore we let $F'_+(x,y) = F'_-(x,y) = F'_*(x,y) = \alpha x+\beta y+\gamma$ and $F'_/(x,y) = \alpha' x+\beta' y+\gamma'$, and try to find constants which assure inequality <u>only</u> for p_7 and p_8 . A solution is $F'_+(x,y) = F'_-(x,y) = F'_*(x,y) = 2x+2y$ and $F'_/(x,y) = x+y$, with $c = 1$. Therefore let W be I_2 rather than I_1 , choose $c = (1,1)$, and use F_θ to compute the first component of the pair and F'_θ to compute the second component.

Conclusion

Our effective sufficient conditions for termination determine subclasses of algorithms for which we can show the existence of c and F_θ's which satisfy the appropriate inequalities.