QUANTIFIER ELIMINATION IN A PROBLEM OF LOGICAL DESIGN

Calvin C. Elgot and Jesse B. Wright

Abstract. A fundamental problem in the theory of logical design is that of expressing behavior realizable by computer circuits. For time-independent circuits, formulas of the propositional calculus have been helpful in expressing behavior. For time-dependent circuits, attempts have been made to use, for this purpose, formulas in which time variables may be quantified.

By the method of quantifier elimination, we find the expressive power of a certain class of formulas which has been used in attempts to describe circuit behavior. We then prove the inadequacy of these formulas for expressing a kind of computer behavior.

1. INTRODUCTION

In the theory of logical design it is important to distinguish the behavior of a computer circuit from its structure. The behavior of a computer circuit is the relationship between every sequence of input signals and the resulting output signals, while the structure is the particular pattern of connections of the components which effect this relationship. In order to deal with some of the fundamental problems in logical design, such as synthesis and analysis, it is necessary to employ some systematic method of symbolizing behavior. Propositional calculus has long been used, often indiscriminately, for describing the structure or the behavior of circuits of the combinational or time-independent type. In the case of sequential circuits, the dependence of the output signals on the complete past history of input signals suggests the use of formulas with quantified time variables for expressing behavior. The discussion in this paper concerns certain sets of formulas in the first-order monadic predicate calculus.

In order to state and prove rigorously results about the use of formulas in expressing circuit behavior, it is first necessary to define precisely what it means for a formula to express the behavior of a circuit. Circuits are labeled by assigning to each input a distinct monadic predicate variable. An output p of a circuit *realizes* a formula $F(I_1, \cdots, I_n, t)$ if, for every assignment of propositional functions $\phi_1, \cdots, \phi_k$ to the predicate variables $I_1, \cdots, I_n$ and of a specific time t_0 to the variable t:

if (a) for all i and x, $\phi_i(x)$ if and only if input ϕ_i is active at time x,

then (b) $F(\phi_1, \cdots, \phi_n, t_0)$ if and only if p is active at time t_0.

Received January 20, 1958.

This research was supported by funds provided by the Office of Naval Research, Contract Nonr-1224(21), Project NR049-114, and by Project MICHIGAN, a Tri-Service contract (DA-36-039sc-57654) administered by the U. S. Army Signal Corps. Funds for the writing of this report were provided by the Office of Naval Research under Contract Nonr-1224(21).

The authors wish to thank J. Richard Büchi for many helpful suggestions.

The evaluation of the set of formulas from the point of view of its adequacy to express circuit behavior has two aspects: First, the variety of different circuit behaviors expressible by formulas in the set must be taken into account. Second, it should be possible to distinguish effectively between those formulas which do and those which do not express the behavior of some computer circuits. Our principal aim is to show that a familiar set of formulas, the singulary formulas defined below, are incapable of expressing the behavior of one of the simplest computer circuits, the modulo two time counter.

2. THE MAIN RESULT

The formulas to be considered are constructed from an alphabet which contains:

individual variables (ranging over nonnegative integers);
individual constant, 0 (zero);
unary operation constant, S (successor);
monadic predicate variables (ranging over properties of the nonnegative integers);
binary constant relation, $\leq$ (the binary relation "less than or equal");
propositional connectives, $\cdot$ (and), $\vee$ (or), ~ (not);
quantifier, $\exists$ (there exists);
auxiliary symbols, (,).

Definition of term and well-formed formula:

1. Any individual variable is a term; the symbol 0 is a term.
2. If T is a term, then S(T) is a term.
3. If I is a monadic predicate variable and T is a term, then I(T) is a wff (well-formed formula).
4. If T, U are any two terms, then $T \leq U$ is a wff.
5. If F, G are wff's, then $(F \cdot G)$, $(F \vee G)$, $(\sim F)$ are wff's.
6. If F is a wff and x an individual variable, then $(\exists x)$ F is a wff.
7. A formula is a term or a wff *only* as a consequence of the above rules.

It is now possible to state more precisely the principal objective of this investigation. Singulary formulas are not capable of expressing the behavior of all computer circuits; that is, there exists a circuit with an output which realizes no singulary formula. Because an essentially simple circuit, one with no inputs, will suffice for the argument, the important part of the discussion can be limited to the consideration of predicate-free singulary formulas. We first ascertain the expressive power of predicate-free singulary formulas (Theorem 1). We next show that the behavior of the modulo two time counter cannot be expressed by any singulary formula (Theorem 2).

A wff which contains exactly one free individual variable is called a *singulary formula.* We shall call singulary formulas with no monadic predicate variables *predicate-free singulary formulas.*

THEOREM 1. *For every predicate-free singulary formula* F(t), *if* M *is the set of natural numbers* t *for which* F(t) *is true, then* M *is either a finite set or the complement of a finite set.* (See the note at the end of this paper.)

Conversely, if M *is a set of natural numbers which is either a finite set or the complement of a finite set, there is a predicate-free singulary formula* F(t) *such that* t *belongs to* M *if and only if* F(t).

Proof. There is a formula $F'(t)$ equivalent to $F(t)$ (in the sense that it is true when and only when the original wff is true) in prenex normal form. We modify this form by replacing universal quantifiers by negations of existential quantifiers. The matrix of F' may be assumed to be in disjunctive normal form. The rightmost symbol of the prefix is an existential quantifier, say $(\exists x)$. The existential quantifier may be distributed over the disjuncts to yield an equivalent formula F''. For convenience, we shall write $u + 1$ instead of $S(u)$, $u + 2$ instead of $S(S(u))$, $u + 3$ instead of $S(S(S(u)))$, and so forth. Now each disjunct D in F'' is of the form

$$(\exists x)[x+a_1 \leq v_1 \cdot x+a_2 \leq v_2 \cdot \ldots \cdot x+a_m \leq v_m$$

$$\cdot\, u_1 \leq x+a_{m+1} \cdot u_2 \leq x+a_{m+2} \cdot \ldots \cdot u_n \leq x+a_{m+n} \cdot p_1 \leq q_1 \cdot \ldots \cdot p_r \leq q_r],$$

where the a_i and b_j are numerals (that is, 0, 1, 2, $\cdots$) and the u_i, v_i, p_i, q_i are terms in which x does not occur, provided each atomic formula of the form $\sim(u \leq v)$ is replaced by its equivalent, $v + 1 \leq u$. Then D is equivalent to a formula D' obtained as follows: add $(\max_i(a_i)) - a_i$ $(i = 1, \cdots, m + n)$ to both sides of the ith inequality in D. Then, with the notation $a = \max_i(a_i)$, formula D' is equivalent to

$$(1) \qquad (\exists x) \prod_{\substack{1 \leq j \leq m \\ 1 \leq i \leq n}} (u_i' \leq x+a \cdot x+a \leq v_j') \cdot \prod_{1 \leq k \leq r} p_k \leq q_k,$$

where $u_i' = u_i + (a - a_{m+i})$ and $v_j' = v_j + (a - a_j)$.

The truth of (1) implies the truth of the formula

$$(2) \qquad \prod_{\substack{1 \leq j \leq m \\ 1 \leq i \leq n}} (u_j' \leq v_j' \cdot a \leq v_j') \cdot \prod_{1 \leq k \leq r} p_k \leq q_k$$

(since x varies over *nonnegative* integers). Conversely, the truth of (2) implies the truth of (1), for, if x is chosen such that $x + a = \min_j v_j'$, then for all j, $x + a \leq v_j'$; and since for all i, $u_i' \leq \min_j v_j'$, it is also true that for all i, $u_i' \leq x + a$. Thus the truth of (2) implies the truth of (1).

The process of eliminating the quantifier $(\exists x)$ is applied to each of the disjuncts of F''. Call the resultant formula F'''. The formula F''' is equivalent to F'', but has one less quantifier. If a negation symbol occurs as the rightmost symbol in the prefix of F''', then the negation of a disjunction of conjunctions may be replaced by an equivalent conjunction of disjunctions, and each disjunct of the form $\sim(u \leq v)$ may be replaced by $v + 1 \leq u$. By repeated applications of the distributive law, the conjunction of disjunctions may be replaced by an equivalent disjunction of conjunctions. Call the resulting formula G. Formula G is in prenex normal form. Now the entire process beginning with F' is repeated for G, and this results in the elimination of another quantifier. By iterating this entire process a number of times (equal to the number of quantifiers in G) we obtain a formula equivalent to F but with no quantifiers.

By assumption, F had exactly one free variable t. The free variable in F was not eliminated by the transformations above, nor were new free variables introduced. Thus the final formula, say H, is of the form of a disjunction of conjunctions, each conjunct being of the form $t + a \leq b$ or $a \leq t + b$, where a and b are numerals.

Either the set of natural numbers for which such a conjunct is true is finite, or its complement is finite. The set of natural numbers for which a conjunction of such conjuncts is true is the intersection of a finite number of sets with this property, and therefore it also has the property of being a finite set or the complement of a finite set. Thus H is true for a finite union of sets with this property, and therefore H also has this property.

We now establish the converse. If M is a finite (the complement of a finite) set, then a disjunction (negation of a disjunction) of formulas of the form $t \leq a \cdot a \leq t$, where a is a numeral denoting an element of M (of the complement of M), is true when and only when $t \in M$ (complement of M). Q.E.D.

Note that if the symbol "=" is defined in the language, then every predicate-free singulary formula is equivalent to a disjunction, or negation of disjunctions, of formulas of the form $t = a$.

THEOREM 2. *Singulary formulas are not capable of expressing the behavior of all computer circuits; that is, there exists a circuit whose output does not realize any singulary formula.*

Proof. The modulo two time counter has an output which is active at time t when and only when t is even. If there were a singulary formula realized by such an output, then some predicate-free singulary formula could also be realized by that output.

With every F(t) is associated a set M which is finite or is the complement of a finite set. In the former case, there are even numbers which are not in M, so that "F(t) if and only if t is even" does not hold. In the latter case, there are odd numbers in M, so that "F(t) if and only if t is even" does not hold.

3. REMARKS

It is not the case that every singulary formula expresses the behavior of some circuit. For example, a circuit which realized $A(t + 1)$, where A is a monadic predicate variable, would have to respond to input signals before they occur. Singulary formulas with bounded quantifiers have been used in attempts to describe computer circuit behavior. Since, by Theorem 2, the singulary formulas are inadequate for this purpose, any subclass is *a fortiori* also inadequate.

The method of quantifier elimination, which played an important role in the proof of Theorem 1 above, has been used frequently in the solution of decision problems of formal logic (see [2, 3, 4]). For predicate-free singulary formulas, the method gives a decision procedure for validity over the natural numbers; but we have not found it necessary to make explicit use of this result.

It is anticipated that some of the procedures current in logic, as for example elimination of quantifiers, will be useful in various problems in the theory of logical design. Consider, for example, the arithmetic of the natural numbers in terms of +, = and first-order quantification. It can be shown, by means of [2, p. 369] and [1, p. 1363, Theorem 13], that the singulary formulas of this arithmetic express the behavior of input-free computer circuits, and of no other behavior.

Note. It has been called to our attention by J. R. Büchi that essentially the first part of Theorem 1 appears on p. 354 of [2].

REFERENCES

1. A. W. Burks and J. B. Wright, *Theory of logical nets*, Proc. I. R. E. 41 (1953), 1357-1365.

2. D. Hilbert and P. Bernays, *Grundlagen der Mathematik*, vol. 1, Berlin (Springer), 1934.

3. M. Presburger, *Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt*, Comptes-Rendus du I Congrès des Mathematiciens des Pays Slaves, Warsaw 1930, pp. 92-101.

4. A. Tarski, *A decision method for elementary algebra and geometry*, The Rand Corporation, Santa Monica, California, University of California Press, 1948.

University of Michigan