SOFTWARE-HARDWARE SYSTEMS

PROGRAMS WITH RESTORATION OF COMPUTATIONS

S. V. Surzhko and E. L. Yushchenko

In this article, the notion of a program with restoration of computations is introduced, i.e., such a program that makes it possible to renew interrupted computations from any intermediate value obtained before an interruption of its operation. As is shown, any operator specified by a regular scheme of algorithmic algebra can be computed by a program with this property.

Keywords: program synthesis, reliability of computations, restoration of computations, algorithmic algebras, regular schemes of operators, control over computations.

The introduction and investigation of the class of programs that are considered in this article is connected, first of all, with the following situation arising in the course of realization of practical computations. Let a program computing the value of some quantity on a computer be unexpectedly interrupted owing to, for example, a system failure or power cut. In this case, the restoration of computations, starting from an intermediate value that is already obtained in the course of computation, is, generally speaking, impossible even in the case where the value is stored in the external memory of the computer. To provide such a possibility, some special and rather complex technical measures are required, which are not directly related to methods of creation of the given program. In this connection, the following problem arises: whether it is possible, using the internal structure of the program, to restore computations that have already been executed by the program and then randomly interrupted, starting from any obtained and stored intermediate value. Or, in other words, the question is about the possibility of realizing an algorithm of computations that is specified in advance on the basis of the program and ensures the mentioned possibility. In what follows, we will show that in the cases where the initial algorithm can be described in terms of a regular scheme of an operator of algorithmic algebra, the question formulated can be answered in the affirmative. From our viewpoint, applications of this problem are not restricted only to the domain of program synthesis but are also directly related to the general problematics of computer system architecture since a processor designed in the context of the above-mentioned property of computation restorability can not only ensure a high reliability and "uninterrupted execution" of computations but also provide a very appreciable convenience for users.

Let $Sch$ be a scheme of a deterministic program, which uses a unique variable $x$, from some class, and $I$ be an interpretation of the basis of this class with a domain of interpretation $X$. In accordance with [1], the interpreted scheme $(Sch, I)$ will be called a program. Denote by $Val(Sch, I)$ the result of execution of the program $(Sch, I)$, assuming that $Val(Sch, I) = \lambda$, where $\lambda \in X$, if this value is not determined because of an endless loop or uncertainty of some of the base operators. Then, for any initial value $I(x) = x^0$ of the variable $x$, the value (specified by the formula $f(x^0) = Val(Sch, I)$) of some operator $f: Y \rightarrow Y$, where $Y = X \cup \{\lambda\}$ and $f(\lambda) = \lambda$, is uniquely defined. Therefore, the set $Y$ can be regarded as the domain of interpretation. The operator $f$ is specified by the totality of these interpretations (with the common domain of interpretation $Y$) that differ only in initial values of the variable $x$. In this connection, we will write $f = f[Sch, Y]$. We note that in the representation of the form $f(x) = \lambda$ given above, the symbol $\lambda$ should not be understood as a computed value of the operator $f$; this form is only a representation of the fact that the operator $f$ is not determined at the point $x$ owing to some reason or other.

---


1060-0396/00/3601-0141$25.00 © 2000 Kluwer Academic/Plenum Publishers
Let us make some comments, which will be important in the sequel.

First, the determinism of the program given by the scheme \( Sch \) and interpretation \( I \) means that, for a given value \( x^0 \in X \), the process of computation of \( f(x^0) \) uniquely defines a finite or infinite sequence \( s(f, x^0) = \{x^0, x^1, x^2, \ldots \} \) of values of the variable \( x \) computed by the program. Since the initial value \( x^0 \) is specified by the interpretation \( I \), we will also write \( s(f, I) \) instead of \( s(f, x^0) \). If this sequence is infinite, then the operator \( f(x) \) is not defined at the point \( x^0 \). In other cases, we can always assume that \( f(x^0) = x^n \), where \( x^n \) is the last value from \( s(f, I) \) that represents the result of execution of the program. We will assume that this case also includes the case where \( x^n = \lambda \).

Second, the case considered in this article (the dependence of the scheme on only one variable), as will be seen in what follows, is not an essential restriction since the vector of all variables of a scheme can be frequently regarded as one vector variable (or we can interpret a unique variable of a scheme as a vector). Thus, any term that is defined on a subset of variables of the scheme and whose value is assigned to one of these variables can be interpreted as an operator, which is defined on the set of values of the vector variable and changes its value.

Note also that the computed points of the sequence \( s(f, x^0) = s(f, I) \) that are different from \( x^0 \) cannot be considered as possible initial values for determining \( f(x^0) \). In fact, in general, for any value \( x^i \in s(f, I) \) that do not coincide with \( x^0 \), the sequence \( s(f, x^i) \) differs from \( s(f, I) \), i.e., it is defined by another interpretation, which is different from \( I \). This means that if the process of computation of \( f(x^0) \) is interrupted because of some reason or other on some step after computing a value \( x^i \in s(f, x^0) \), then it is impossible, generally speaking, to restore the process of computations from an obtained intermediate value. In this connection, the problem arises of existence and possibility of constructive description of a general mechanism for restoration of computations interrupted at an intermediate step, provided that the last computed value from the set \( s(f, x^0) \) is known. In a more specific statement, we assume that the three statements given below are simultaneously valid.

1. For any initial scheme \( Sch \) with an interpretation \( I \), a scheme \( Sch^0 \) with an interpretation \( J \) can be constructed such that the value of \( Val(Sch, I) \) is defined if and only if the value of \( Val(Sch^0, J) \) is defined.

2. On the basis of the value of \( Val(Sch^0, J) \) (including the uncertain one), the value of \( Val(Sch, I) \) can be uniquely determined.

3. If \( g \) is an operator specified by the scheme \( Sch^0 \) with the interpretation \( J \) and the sequence \( s(g, J) \) is finite, then the equality \( g(y) = g(z) \) is fulfilled for any values \( y, z \in s(g, J) \). If the sequence \( s(g, J) \) is infinite, then the operator \( g \) is not defined at any point \( y \in s(g, J) \).

In the case where a program \( (Sch^0, J) \) satisfies the conditions of the latter statement, we will call it a program with restoration of computations.

Thus, the validity of the above-mentioned statements for some class of programs or other would mean that, instead of using an ordinary “unrestorable” process of computing the value of any of interpreted program schemes of a given class, we can use a program with restoration of computations, which makes it possible to obtain the sought-for value starting from any stored partial result.

Programs with restoration of computations have, in particular, the property given below.

**Lemma.** If \( (Sch^0, J) \) is a program with restoration of computations and \( g \) is an operator specified by this program, then \( Val(Sch^0, J) \) is a fixed point of the operator \( g \).

**Proof.** Let \( x^0 \) be the initial value of the variable, the value of \( Val(Sch^0, J) \) be equal to \( t \), where \( t \) is different from \( \lambda \), and the sequence \( s(g, J) = s(g, x^0) \) be finite. Then \( t \in s(g, J) \) since \( t = g(x^0) \) coincides with the last computed value of the variable in this case. But \( g(y) = g(z) \) for any \( y, z \in s(g, J) \) and, hence, \( g(t) = t \), i.e., the point \( t \) is fixed. The case where \( Val(Sch^0, J) = \lambda \) is trivial.

Let us now consider a solution of the formulated problem for the class of regular schemes of operators of algorithmic algebras.

We will give auxiliary definitions of a general character.

Let \( f: X \to X \) be a partial operator, \( \lambda \notin X \), and \( Y = X \cup \{\lambda\} \). Taking into account the agreement concerning the