M. Burgin

Algorithmic problems for super-recursive algorithms

Department of Mathematics, University of California, Los Angeles, CA 90095-1555, USA mburgin@math.ucla.edu

The traditional formalism for a substantive concept of algorithm presupposes two essential conditions: the finiteness and constructibility of the input, processed, and output information over a finite interval of time, as well as the finiteness and constructibility of the list of rules for realization of the processing. In contrast to an informal notion, other conditions of algorithm specification appear only in mathematical models. One of these conditions is the demand that to obtain a result of a computation, it is necessary to finish the computational process. However, it is superfluous in many situations because quantities of processes that are naturally considered as algorithmic do not satisfy this restriction. That is why, a more general model, called a super-recursive algorithm has been developed making possible to increase to a great extent performance abilities of systems of algorithms [**1**].

Inductive Turing machines (iTm) are the closest super-recursive systems to the ordinary algorithms having at the same time much more computing power [**2**].

Let K, H be two classes of algorithms. We consider the following algorithmic problem.

(K, H)-Problem. Is there some A from H that determines for any input value x whether an arbitrary algorithm B from K gives the result being applied to x?

Let RITM and TM be the classes of all iTm with a recursive memory and ordinary Turing machines, correspondingly.

Theorem:

a) (TM, RITM)-problem has the positive solution;

b) (RITM, RITM)-problem has the negative solution.

References

[1] Burgin M.S. *The notion of algorithm and the Turing-Churh's thesis*, ***VIII International Congress on logic, methodology and philosophy of science***, Moscow, v. 5, part 1, (1987), pp. 138–140.

[2] Burgin M.S. *Inductive Turing machines*, ***Soviet Math. Doklady***, (1983), 270, 1289–1293.