Meeting: 1007, Santa Barbara, California, SS 10A, Special Session on Complexity of Computation and Algorithms

1007-03-78 **Mark Burgin***, Department of Mathematics, UCLA, Los Angeles, CA 90095, and **Gregory J. Chaitin**, IBM Research, Yorktown Heights, New York, NY 10598. *Algorithmic complexity and inductive decidability.* Preliminary report.

There is a dependency between computability of algorithmic complexity and decidability of different algorithmic problems. In [G.J. Chaitin, Program-size complexity computes the halting problem, Bull. European Assoc. Theor. Computer Sci., v. 57, 1995], it is proved that computability of the algorithmic complexity $C(x)$ is equivalent to decidability of the halting problem for Turing machines. Here we extend this result to the realm of superrecursive algorithms, considering algorithmic complexity for inductive Turing machines [M.Burgin, Superrecursive Algorithms, Springer, 2005].

Theorem. The following algorithmic problems are equivalent: (a) Decidability by inductive Turing machines of the first order of the halting problem for inductive Turing machines of the first order; (b) Computability of the inductive algorithmic complexity $IC(x)$ by inductive Turing machines of the first order. (c) Computability of the relative to the Turing jump 0' algorithmic complexity $C(x/0')$ by inductive Turing machines of the first order.

Corollary 1. The inductive algorithmic complexity $IC(x)$ is inductively noncomputable.

Corollary 2. The problem of being an elegant inductive program is inductively undecidable. (Received February 03, 2005)