Abstract. In this paper a measure for the complexity of particular instances with respect to a given decision problem is introduced and investigated. Intuitively, an instance $x$ is considered to be hard for a problem $A$ if every algorithm that decides $A$ and runs "fast" on $x$ needs to look up (a description of) $x$ in a table. A main result states that all problems not in $P$ have infinitely many (polynomially) hard instances. Further, there exist problems in $EXPTIME$ with all their instances being hard. The behavior of hard instances under polynomial reductions and the connections with complexity cores and circuits are studied.
1. Introduction

There are (at least) two principal views of what causes the computational intractability of decision problems. The "distributional" view suggests that the yes- and no-instances of a difficult problem are distributed in some very irregular manner, but feasible algorithms can only determine "smooth" distributions. This is the common view in complexity theory, where the asymptotic behavior of algorithms is emphasized. Another view is suggested by the strong intuitive feeling that also individual problem instances can be inherently hard, i.e. hard independent of any particular algorithm used to decide the problem. Such ideas of "instance complexity" have been discussed, for instance, by Hartmanis [Har83b].

One approach to studying these issues has been the notion of a complexity core introduced by Lynch [Lyn75]. A (polynomial) complexity core for a problem A is an infinite collection C of instances such that every algorithm that decides A needs more than polynomial time almost everywhere on C. In a sense, a complexity core is a "uniformly hard" collection of problem instances. It is known that any problem not in the class P has such a complexity core [Lyn75], and that NP-complete sets have cores whose density majorizes every polynomial function [OS84]. Recently, complexity cores have been a subject of extensive study [ESY85, OS84, ORS85, Ko85, OS86].

A shortcoming of this approach is that the notion of a complexity core does not really say anything about the complexity of single instances: any finite alteration of a core still leads to a core. However, the "almost everywhere" cannot be removed from the definition because any finite set of instances can be decided trivially by a table look-up. This possibility of patching algorithms with tables is a basic difficulty in formulating what it means for a single instance to be hard.

This shows that a measure of instance complexity should also take into account the sizes of the decision algorithms for the problem under consideration. Here we take the following approach. For a given function t, define the t-bounded complexity of an instance x with