REDUCIBILITY AMONG COMBINATORIAL PROBLEMS[†]

Richard M. Karp

University of California at Berkeley

Abstract: A large class of computational problems involve the
determination of properties of graphs, digraphs, integers, arrays
of integers, finite families of finite sets, boolean formulas and
elements of other countable domains. Through simple encodings
from such domains into the set of words over a finite alphabet
these problems can be converted into language recognition problems,
and we can inquire into their computational complexity. It is
reasonable to consider such a problem satisfactorily solved when
an algorithm for its solution is found which terminates within a
number of steps bounded by a polynomial in the length of the input.
We show that a large number of classic unsolved problems of cover-
ing, matching, packing, routing, assignment and sequencing are
equivalent, in the sense that either each of them possesses a
polynomial-bounded algorithm or none of them does.

1. INTRODUCTION

All the general methods presently known for computing the
chromatic number of a graph, deciding whether a graph has a
Hamilton circuit, or solving a system of linear inequalities in
which the variables are constrained to be 0 or 1, require a
combinatorial search for which the worst case time requirement
grows exponentially with the length of the input. In this paper
we give theorems which strongly suggest, but do not imply, that
these problems, as well as many others, will remain intractable
perpetually.

We are specifically interested in the existence of algorithms that are guaranteed to terminate in a number of steps bounded by a polynomial in the length of the input. We exhibit a class of well-known combinatorial problems, including those mentioned above, which are equivalent, in the sense that a polynomial-bounded algo-rithm for any one of them would effectively yield a polynomial-bounded algorithm for all. We also show that, if these problems do possess polynomial-bounded algorithms then all the problems in an unexpectedly wide class (roughly speaking, the class of problems solvable by polynomial-depth backtrack search) possess polynomial-bounded algorithms.

The following is a brief summary of the contents of the paper. For the sake of definiteness our technical development is carried out in terms of the recognition of languages by one-tape Turing machines, but any of a wide variety of other abstract models of computation would yield the same theory. Let Σ^* be the set of all finite strings of 0's and 1's. A subset of Σ^* is called a <u>language</u>. Let P be the class of languages recognizable in polynomial time by one-tape deterministic Turing machines, and let NP be the class of languages recognizable in polynomial time by one-tape nondeterministic Turing machines. Let Π be the class of functions from Σ^* into Σ^* computable in polynomial time by one-tape Turing machines. Let L and M be languages. We say that $L \propto M$ (L <u>is reducible to</u> M) if there is a function $f \in \Pi$ such that $f(x) \in M \Leftrightarrow x \in L$. If $M \in P$ and $L \propto M$ then $L \in P$. We call L and M equivalent if $L \propto M$ and $M \propto L$. Call L (polynomial) <u>complete</u> if $L \in NP$ and every language in NP is reducible to L. Either all complete languages are in P, or none of them are. The former alternative holds if and only if $P = NP$.

The main contribution of this paper is the demonstration that a large number of classic difficult computational problems, arising in fields such as mathematical programming, graph theory, combina-torics, computational logic and switching theory, are complete (and hence equivalent) when expressed in a natural way as language recognition problems.

This paper was stimulated by the work of Stephen Cook (1971), and rests on an important theorem which appears in his paper. The author also wishes to acknowledge the substantial contributions of Eugene Lawler and Robert Tarjan.

2. THE CLASS P

There is a large class of important computational problems which involve the determination of properties of graphs, digraphs, integers, finite families of finite sets, boolean formulas and