Hierarchies of Complete Problems*

Zvi Galil

Received June 4, 1975

Summary. An attempt is made to present a framework for the diverse complete problems that have been found. A new concept—a Hierarchy of Complete Problems is defined. Several hierarchies in various domains such as graph theory, automata theory, theorem proving and games are established.

1. Introduction

In [1] S. Cook introduced the notions of polynomial time reducibility and complete languages in NP-TIME. Moreover, he showed that some natural combinatorial problems are complete in NP-TIME. Since then, using several efficient reducibilities, researchers have found many other complete problems in DLOG, NLOG, P-TIME, NP-TIME and P-SPACE ([13, 11, 12, 9, 15], etc.). Although DLOG \( \subseteq \) NLOG \( \subseteq \) P-TIME \( \subseteq \) NP-TIME \( \subseteq \) P-SPACE we cannot find in the literature any quintuple of complete problems, one in each of these five classes of languages, such that each is a special case of the next. The goal of this paper is to present such hierarchies of complete problems.

First we recall some well-known definitions:

Definitions. Let \( C \) be a class of functions, \( A \) and \( B \) be sets and \( S \) a class of sets.

(i) \( A <_c B \) if there is a function \( f \in C \) such that for every \( w, w \in A \) iff \( f(w) \in B \).
(ii) \( S <_c A \) if for every \( B \in S \) \( B <_c A \).
(iii) \( A \) is \( <_c \) complete in \( S \) if
   
   (a) \( S <_c A \) and
   
   (b) \( A \in S \).

In this paper we restrict ourselves to the set \( C \) of functions which can be computed by a Turing machine (Tm) which uses logarithmic tape. Thus we write \( <_c \).
rather than $<_{c}$ (this is also denoted $<_{\log}$ in the literature) and complete instead of $<_{-complete}$. We justify our restriction by the fact that most (if not all) of the reductions in the literature which are used to describe specific complete problems are $<_{\log}$. In one case, however, this restriction does not make much sense, since every $L \in \text{DLOG}$ is $<_{\log}$ complete in DLOG. To overcome this difficulty Jones has defined ([14]) the logu rudimentary functions. All the reductions in the sequel are $<_{\log}$. Since they are fairly straightforward in all cases, we omit the tedious details of the proofs that the reductions can be done using logarithmic space. In the case of DLOG our reductions can be shown also to be rudimentary.

The importance of complete problems is two-fold:

1) Often, a question about a class of languages can be translated into a seemingly simpler question about one language in the class, namely the complete language. In particular the question “Are $A$ and $B$ equal?” where $A \leq B$ can be translated into the question “Is $L$ in $A$?” for $L$ complete in $B$; and the question “Is $A$ closed under complementation?” can be translated into “Is $L$ in $A$?”.

Thus, most of the important open problems of computational complexity can be translated into questions about the corresponding complete problems.

2) In some sense the complete problem is the hardest in the class since every problem in the class can be efficiently encoded into it. Thus, studying a complete problem sheds light on the whole class that it is complete in.

We now give the definition which is of central importance to our paper.

**Definition.** A hierarchy of complete problems (HCP) is a quintuple of problems $(A_1, A_2, A_3, A_4, A_5)$ that satisfies:

1) $A_1$ is complete in DLOG, $A_2$ is complete in NLOG, $A_3$ is complete in P-TIME, $A_4$ is complete in NP-TIME, and $A_5$ is complete in P-SPACE; and

2) For $1 \leq i \leq 4$ $A_i$ is a special case of $A_{i+1}$, i.e. $A_i$ is obtained from $A_{i+1}$ by adding some restrictions to $A_{i+1}$.

In this paper we prove the existence of several hierarchies of complete problems in varied areas such as graph theory, automata theory, theorem proving and games. Although we introduce quite a few new complete problems, this is not the purpose of this paper. Our goal is to present a framework for the diverse complete problems that have been found. So far the numerous known complete problems do not fit a unified framework. We also investigate the question of what changes make a problem easier. Since we believe that DLOG $\subseteq$ NLOG $\subseteq$ P-TIME $\subseteq$ NP-TIME $\subseteq$ P-SPACE, a restriction which is added to a problem $A_{i+1}$ in an HCP to get $A_i$ really makes the problem easier. Recently ([5]) a closely related question was investigated—i.e.: Which restrictions on a problem do not affect its complexity? It was shown that some strictly restricted versions of complete problems in NP-TIME are still complete in NP-TIME.

As we said before, our intention was not to introduce new complete problems, but our paper includes for the first time simple combinatorial problems which are complete in P-SPACE. Until now, the only natural complete problems in P-SPACE were either questions about automata (like equivalence of regular expressions) or

---

4 This was first observed in [15] and [11].