Zeitschrift: L'Enseignement Mathématique
Herausgeber: Commission Internationale de l'Enseignement Mathématique
Band: 28 (1982)
Heft: 1-2: L'ENSEIGNEMENT MATHÉMATIQUE

Artikel: RFDUCIBILITY BY ALGEBRAIC PROJECTIONS
Autor: Valiant, L. G.
Kapitel: 1. Introduction
DOI: https://doi.org/10.5169/seals-52240

REDUCIBILITY BY ALGEBRAIC PROJECTIONS *

by L. G. VALIANT

ABSTRACT

Substitution as a notion of reduction between two polynomials or two Boolean functions is considered. It is shown that in a strong sense linear programming is a universal technique for computing discrete functions in polynomial time. The robustness of the notion of p-definability for polynomials is demonstrated by showing that alternative formulations, whether based on formula or program size, are equivalent. Also it is closed under most natural operations including substitution, taking coefficients and differentiation. These results facilitate the recognition of particular polynomials as p-definable. The polynomial analogue of the Meyer-Stockmeyer hierarchy collapses.

1. INTRODUCTION

The programming concept of a *subroutine* is well represented in theoretical computer science in the notion of *reducibility*. A function $A(\mathbf{x})$ is many-one reducible to function $B(\mathbf{y})$ if there is an easily computed transformation f such that $A(\mathbf{x}) = B(f(\mathbf{x}))$. A can be computed by computing f and then calling a subroutine for B. Traditionally this is the strictest notion considered. It is relaxed sometimes to allow several subroutine calls, or further computation after the call. In this paper we proceed in the opposite direction by considering reductions stricter still.

We say that $A(x_1, ..., x_n)$ is a *projection* of $B(y_1, ..., y_m)$ if after substituting for each y_i either an x_j or a constant, B equals $A(x_1, ..., x_n)$. Mathematically this notion has the obvious advantages of simplicity and of independence from any computational models. In programming terms it corresponds naturally to the concept of a *package* rather than subroutine,

* This article has already been published in *Logic and Algorithmic*, an international Symposium in honour of Ernst Specker, Zürich, February 1980. Monographie de L'Enseignement Mathématique N° 30, Genève 1982.

since the value of A can be obtained by calling B with the same inputs suitably reinterpreted. If a subroutine for B is available, A can be computed without further programming or precomputation on the input being required. The distinction between subroutines and packages can be of considerable practical importance as far as the effort required of a human user.

The results in this paper extend and complement those in [13], but can be read independently. There it was shown that the determinant is a universal function for all polynomials that can be computed fast sequentially or in parallel, and transitive closure is universal for Boolean functions computable fast in parallel. Here we complete this rough picture by showing that linear programming has the same universal role for Boolean functions that can be computed fast sequentially.

The concept of p-definability introduced in [13] serves to explain the difficulty of many intractable problems by providing an extensive class in which they are provably of maximal difficulty. In the polynomial case this suggests new techniques for identifying hard problems e.g. [6]. A shortcoming of the original treatment in [13] was that recognizing particular polynomials to be p-definable was sometimes possible only by indirect contrived means. The current paper remedies this by providing some useful equivalent definitions and various closure properties.

In the Boolean case p-definability provides an alternative approach to formulating such notions as NP, the Meyer-Stockmeyer hierarchy and polynomial space. It can be checked, for example, that the twenty-one NP-complete problems of Karp [7] are all p-projections of each other, and complete in our class. An important difference between our approach and the established one is that ours does not contain any assumptions about "Turing uniformity" (i.e. computational uniformity over infinite domains.) Thus, while this latter ingredient is a *sine qua non* in recursion theory and high-level complexity, it may be no more than an optional extra at the lower levels.

2. Definitions

Our notation is taken from [13] but is repeated here for completeness. We start with the case of polynomials.

Let F be a field and $F[x_1, ..., x_n]$ the ring of polynomials over indeterminates $x_1, ..., x_n$ with coefficients from F. P and Q will denote families of polynomials where typically