Zeitschrift: L'Enseignement Mathématique

Band: 28 (1982)

Heft: 1-2: L'ENSEIGNEMENT MATHÉMATIQUE

Artikel: RFDUCIBILITY BY ALGEBRAIC PROJECTIONS

Kapitel: 2. Definitions

Autor: Valiant, L. G.

DOI: https://doi.org/10.5169/seals-52240

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

$$P = \{P_i \mid P_i \in F[x_1, \ldots, x_i], i \in X\},$$

where X is a set of positive integers. The arguments of P_i are exhibited sometimes as $P_i(x_1, \ldots, x_i)$ or $P_i(\mathbf{x})$ for short.

A *formula* f over F is an expression that is of one of the following forms: (i) "c" where $c \in F$, or (ii) "x_j" where x_j is an indeterminate, or (iii) "$(f_1 \circ f_2)$ where f_1 and f_2 are themselves formulae over F and $\circ$ is one of the two ring operators $\{+, \times\}$. The *size* of a formula is the number of operations of type (iii) needed in its construction, and is denoted by $|f|$. The *formula size* $|P_i|$ of polynomial P_i is the size of the minimal size formula that specifies it.

A *program* f over F is a sequence of instructions $v_i \leftarrow v_j \circ v_k$ $(i = 1, 2, \ldots, C)$ where (i) $j, k < i$, (ii) $\circ$ is one of the two ring operations $\{+, \times\}$, and (iii) if $j \leqslant 0$ then v_j is either an indeterminate x_m or a constant $c \in F$. The polynomial computed at v_i in the program is denoted by val (v_i) and its degree by deg (v_i). The size of a program is the number C of instructions. The *program size* $\|P_i\|$ of a polynomial P_i is the size of the minimal program that computes it.

Since formulae are just programs of a special form, in which each computed term can be used at most once, formula size is always at least as great as program size. A non-trivial converse relationship is due to Hyafil [5, 14].

A function from positive integers to positive integers we shall denote typically by t. Such a t is *p-bounded* if for some constant k, for all $n > 1$ $t(n) \leqslant n^k$. A family P has *p-bounded formula size* if for some p-bounded t for each i $|P_i| < t(i)$. P is *p-computable* iff for some p-bounded t for each i (a) $\|P_i\| < t(i)$ *and* (b) deg $(P_i) < t(i)$.

$Q_i \in F[y_1, \ldots, y_i]$ is a *projection* of $P_j \in F[x_1, \ldots, x_j]$ iff there is a mapping

$$\sigma\colon \{x_1, \ldots, x_j\} \to \{y_1, \ldots, y_i\} \cup F$$

such that $Q_i = P_j(\sigma(x_1), \ldots, \sigma(x_j))$.

Family Q is a *t-projection* of P if for each i for some $j < t(i)$ Q_i is the projection of P_j. It is the *p-projection* of P if it is the t-projection of P for some p-bounded t.

Among polynomial families that are generally regarded as intractable both mathematically and computationally, perhaps the simplest is the permanent [11] which is defined as follows.

$$\text{Perm}_{n\times n}(x_{ij}) = \sum_{\pi} \prod_{i=1}^{n} x_{i,\pi(i)}$$

where summation is over the n ! permutations on n elements. This contrasts with the similar looking determinant which is tractable in both senses.

Another one is Hamiltonian Circuits:

$$HC_{n \times n}(x_{ij}) = \sum_{\pi} \prod_{i=1}^{n} x_{i, \pi(i)}$$

where summation is now over all $(n-1)$! permutations consisting of a single cycle. Related to the latter are HG and $\# HG$ which are defined by

$$\sum_{\tau} N_\tau \cdot \prod_{x_{ij} \in \tau} x_{ij}$$

where summation is over those subsets τ of $\{ x_{ij} \mid 1 \leqslant i, j \leqslant n \}$ that *contain* a Hamiltonian circuit when interpreted as graphs. In HG $N_\tau = 1$. In $\# HG$ N_τ equals the number of Hamiltonian circuits in τ.

To treat Boolean computations we can use the same terminology as for polynomials except that $\{ +, \times \}$ are now interpreted as $\{$ or, and $\}$. For the above polynomials the graphical interpretation, where the value of x_{ij} denotes the presence or absence of edge (i, j), is natural. The permanent becomes the perfect matching function which is tractable [9]. HC, HG and $\# HG$ become identical and test for the presence of Hamiltonian circuits in a graph.

The Boolean versions of formulae, programs and projections differ only in the following ways: In formulae and programs an occurrence of an indeterminate x_i can now be either x_i or its negation $\bar{x}_i$, and constants need not occur at all. In a projection the mappings allowed are

$$\sigma : \{ x_1, \ldots, x_j \} \rightarrow \{ y_1, \bar{y}_1, y_2, \bar{y}_2, \ldots, y_i, \bar{y}_i \} \cup \{ 0, 1 \} .$$

The concept of degree is not defined and p-computability means just p-bounded program size. Lemma 18 in [4] ensures that this measure does correspond to the familiar notion of circuit size.

We shall be interested often in polynomials that have certain desired behaviour on $\{ 0, 1 \}$ inputs. In particular let $\mathrm{Sym}_n^r \in F[x_1, \ldots, x_n]$ be such that on any input from $\{ 0,1 \}^n$ it has value 1 or 0 according to whether exactly r of the inputs have value 1. A p-computable candidate for Sym_n^r is

$$(1 - T_n^n)\,(1 - T_n^{n-1})\ \ldots\ (1 - T_n^{r+1})\, T_n^r$$

where T_n^i is the sum of the $\binom{n}{i}$ multilinear monomials of degree i, each with coefficient 1, i.e. the i'th elementary symmetric function.