6. Universality of Linear Programming

Objekttyp: **Chapter**

Zeitschrift: **L'Enseignement Mathématique**

Band (Jahr): **28 (1982)**

Heft 1-2: **L'ENSEIGNEMENT MATHÉMATIQUE**

PDF erstellt am: **21.09.2024**

We can attempt to generalise the definition of the above vacuous hierarchy by allowing the number of "alternations" to increase with the number of indeterminates.

Let t be any polynomial. Define $t\text{-}D^0$ to be the class of t-computable families. For $i > 0$ let $t\text{-}D^i$ be the class of families that are defined by some family in $t\text{-}D^{i-1}$ in the sense of Definition 3. Finally PD^* is the class of all families P such that for some t

$$P = \{P_i \mid P_i = Q_i \quad \text{for some} \quad Q \in t\text{-}D^{t\,(i)}\}\,.$$

THEOREM 6. $PD^* = PD^1$

Proof. Similar to previous theorem. □

The above two results should be contrasted with the Boolean case where they still hold formally, but are no longer natural. The above definition of the successive levels PD^i is only natural if each level is a robust closure class. In Boolean algebra, however, PD^i is not known to be closed under complementation for any $i \geqslant 1$. Analogues of PD^i and PD^* where complementation is allowed at each level of alternation are not known to collapse, and are merely finite versions of the Meyer-Stockmeyer hierarchy, and PSPACE respectively [10].

A simple application of Theorem 5 is in recognising such polynomials as $\# HG$ as being p-definable. An intriguing open question is whether HG itself is p-definable for each F. If it is not then $P \neq NP$ (see Proposition 4 in [13]). If it is then the Meyer-Stockmeyer hierarchy and PSPACE can be simulated within p-definable families of polynomials.

6. UNIVERSALITY OF LINEAR PROGRAMMING

Here we consider a Boolean function family LP that corresponds to a linear programming problem and show that every p-computable family is the p-projection of it. Thus for computing discrete functions in polynomial time a package for LP for each input size is sufficient and no further programming is required. If we fix certain of the arguments of LP_i according to the particular function and input size being computed, the package becomes a program for the required function. That LP is itself p-computable follows from the recent result of Khachian [8].

The reader should note that several tractable problems in combinatorial optimisation are already known to have linear programming formulations [9]. Our result shows that this is a universal phenomenon. It is related to the result in [3].

We define $LP_{2n(n+1)}$ to be the following Boolean function of arguments $\{ a_{ij}, b_{ij}, e_i, d_i \mid 1 \leqslant i, j \leqslant n \}$:

$$LP(a_{ij}, b_{ij}, e_i, d_i) = 1$$

if and only if the set of inequalities

$$\sum (\tilde{a}_{ij} x_j - \tilde{b}_{ij} x_j) \geqslant \tilde{e}_i - \tilde{d}_i$$

has a solution in real numbers, where each number $\tilde{a}_{ij}, \tilde{b}_{ij}, \tilde{e}_i, \tilde{d}_i$ is 1 or 0 according to whether the corresponding Boolean variable a_{ij}, b_{ij}, e_i, d_i is 1 or 0.

THEOREM 7. *Any p-computable family P of Boolean functions is the p-projection of LP.*

Proof. Consider some $P_m \in P$ with indeterminates $y_1, \ldots, y_m$, and a minimal program for it. The latter consists of a sequence of instructions of the form $v_i \leftarrow v_j \wedge v_k$ and $v_i \leftarrow v_j \vee v_k$, where $1 \leqslant i \leqslant C$ and each v_n with $n \leqslant 0$ equals some y_r or $\bar{y}_r$.

For any fixed assignment of truth values to $y_1, \ldots, y_m$ we can define a set E_0 of linear inequalities:

$$E_0 = \{x_r \leqq 0 \mid r < 0 \text{ and } v_r \text{ has value } 0\} \cup \{x_r \geqslant 1 \mid r < 0 \text{ and } v_r \text{ has value } 1\}$$

For each sequence $v_1, v_2, \ldots, v_i$ we define E_i by induction from E_0:

$$E_i = \begin{cases} E_{i-1} \cup \{x_j - x_i \geqslant 0, x_k - x_i \geqslant 0, x_i + 1 - x_j - x_k \geqslant 0\} \\ \qquad \text{if} \quad v_i \leftarrow v_j \wedge v_k\ , \\ E_{i-1} \cup \{x_j + x_k - x_i \geqslant 0, x_i - x_j \geqslant 0, x_i - x_k \geqslant 0\} \\ \qquad \text{, if} \quad v_i \leftarrow v_j \vee v_k \end{cases}$$

Claim 1. For any i, j $(j < i)$ every solution of E_i has $x_j \leqslant 0$, or every solution of E_i has $x_j \geqslant 1$.

Proof. The claim is true for E_0 by definition. Assume inductively that it is true for E_{i-1}. (a) If $v_i \leftarrow v_j \wedge v_k$ then $x_j \leqslant 0$ implies that $x_i \leqslant 0$ since $x_j - x_i \geqslant 0$. Similarly if $x_k \leqslant 0$. In the remaining case $x_j, x_k \geqslant 1$ inequality $x_i + 1 - x_j - x_k \geqslant 0$ ensures that $x_i \geqslant 1$. (b) If $v_i \leftarrow v_j \vee v_k$ then $x_j \geqslant 1$

implies that $x_i \geqslant 1$ since $x_i - x_j \geqslant 0$. Similarly if $x_k \geqslant 1$. If $x_j, x_k \leqslant 0$ then $x_j + x_k - x_i \geqslant 0$ ensures that $x_i \leqslant 0$. □

Claim 2. If val $(v_i) = 0$ then $E_i \cup \{ x_i \leqslant 0 \}$ has a solution. If val $(v_i) = 1$ then $E_i \cup \{ x_i \geqslant 1 \}$ has a solution.

Proof. By induction on i it is easy to see that the point

$$x_j = \begin{cases} 1 & \text{if val } (v_j) = 1 \\ 0 & \text{if val } (v_j) = 0 \end{cases}$$

for $1 \leqslant j \leqslant i$ is a solution of E_i. □

Claim 3. If for some i, j $(j \leqslant i)$ $E_i \cup \{ x_j \geqslant 1 \}$ has a solution in reals then val $(v_j) = 1$.

Proof. By Claim 1, if $E_i \cup \{ x_j \geqslant 1 \}$ has a solution then $E_i \cup \{ x_j \leqslant 0 \}$ has no solution. Hence by Claim 2 val $(v_j) = 1$. □

Finally we observe that the given program of size C for P_m translates to $3C + 2m$ inequalities in E_C, of which the $2m$ of E_o depend on the values of $y_1, \ldots, y_m$, while the remaining $3C$ are fixed. It remains to note that P_m is the projection under σ of $LP_{2n(n+1)}$ for $n = 3C + 2m$, where σ maps $3C$ of the inequalities to those of $E_C - E_o$, and the remaining $2m$ values of i as follows. If v_i equals y_j or $\bar{y}_j$ then: $\sigma(a_{ik}) = \sigma(b_{ik}) = 0$ if $j \neq k$, $\sigma(d_i) = 0$, $\sigma(a_{ij}) = \sigma(e_i) = v_i$, $\sigma(b_{ij}) = \bar{v}_i$. □

Acknowledgements. It is a pleasure to thank Volker Strassen and Mark Jerrum for suggesting corrections and simplifications on a first draft of this paper.

Appendix 1

We show here that in the concept of p-definability it is immaterial whether the defining polynomials allowed are the p-computable ones or merely those of p-bounded formula size. We shall suppose that the family P is p-definable in the sense of Definition 3, i.e.

$$P_n(x_1, \ldots, x_n) = \sum_{b \in \{0,1\}^{m-n}} Q_m(x_1, \ldots, x_n, b_{n+1}, \ldots, b_m)$$

It will suffice to prove that any p-computable family, such as Q, is p-definable in the sense of Definition 4. By Theorem 5 it then follows that P itself is also p-definable in the sense of Definition 4.