Available online at www.sciencedirect.com
SCIENCE DIRECT®

Discrete Applied Mathematics 149 (2005) 1–13

DISCRETE APPLIED MATHEMATICS

www.elsevier.com/locate/dam

The complexity of modular decomposition of Boolean functions

Jan C. Bioch

Department of Econometrics, Erasmus University Rotterdam, FEW, PO Box 1738, 3000 DR Rotterdam, Netherlands

Received 30 March 2002; received in revised form 14 November 2003; accepted 11 December 2003
Available online 10 May 2005

Abstract

Modular decomposition is a thoroughly investigated topic in many areas such as switching theory, reliability theory, game theory and graph theory. We propose an $O(mn)$-algorithm for the recognition of a modular set of a monotone Boolean function f with m prime implicants and n variables. Using this result we show that the computation of the modular closure of a set can be done in time $O(mn^2)$. On the other hand, we prove that the recognition problem for general Boolean functions is coNP-complete. Moreover, we introduce the so-called generalized Shannon decomposition of a Boolean function as an efficient tool for proving theorems on Boolean function decompositions.

Keywords: Boolean functions; Committees; Computational complexity; Modular decomposition; Modular sets; Substitution decomposition; Game theory; Reliability theory; Switching theory

1. Introduction

Substitution decomposition has been thoroughly studied by researchers in many different contexts such as switching theory, game theory, reliability theory, network theory, graph theory and hypergraph theory. Möhring and Radermacher [21,22] give an excellent survey for the various applications of substitution decomposition and connections with combinatorial optimization. They also present a framework for the algebraic and algorithmic aspects of substitution decomposition for a number of discrete structures. Substitution

E-mail address: bioch@few.eur.nl.

0166-218X/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2003.12.010

decomposition (disjunctive and non-disjunctive decomposition) for general Boolean functions and partially defined Boolean functions in switching theory is mainly developed by Ashenhurst, Singer, Curtis and Hu [1,2,16–18]. Nowadays decomposition of Boolean functions is an important design-methodology in automatic synthesis for Field Programmable Gate Arrays (FPGAs), see e.g. [27,23]. Recently [11,10,19] the complexity of non-disjunctive decompositions of partially defined Boolean functions has been determined for various classes of Boolean functions. In this direction the recent paper of Zupan et al. [31] on concept hierarchies deserves further attention. In this paper a recursive decomposition of partially defined discrete function is used to obtain structural information of a data set. Decomposition for monotone Boolean functions has been studied in several contexts: game theory (decomposition of n-person games [29]), reliability theory (decomposition of coherent systems [8]) and set systems (clutters [3]). The concepts decomposition and *modular set* are very basic in many contexts and applications. Not surprisingly, the concept of a modular set is rediscovered several times under various names: bound sets, autonomous sets, closed sets, stable sets, clumps, committees, externally related sets, intervals, nonsimplifiable subnetworks, partitive sets and modules, see [12,22] and references therein. In all these contexts the collection of all modular sets is efficiently represented by the so-called decomposition tree introduced by Shaply in [29]. In graph theory efficient algorithms are known to compute this tree [12,20,14]. The notion of a module in a graph has been recently generalized to hypergraphs in [9]. A unified treatment of all algorithms (up to 1990) related to modular sets known in game theory, reliability theory and set systems (clutters) is given by Ramamurthy [25]. A systematic account using Boolean function theory based on the idea of ‘generalized Shannon decomposition’ is developed in our accompanying paper [4].

In this paper we are interested in the algorithmic complexity of the decomposition of Boolean functions given in DNF. After introducing some definitions and concepts in Section 2, we introduce in Section 3 the useful concept of ‘generalized Shannon decompostion’ and we argue that this concept can be used to simplify decomposition theory. In Section 4 we will show that the complexity of decomposition for general Boolean functions is coNP-complete. Decompositions of monotone Boolean functions, modular sets and the modular closure are discussed in Section 5. In Section 6 we discuss the computational aspects of decomposing positive functions and we prove that for a positive function f the recognition problem of the modularity of a set can be solved in time $O(mn)$, where n is the number of variables of f and m is the number of prime implicants of f. Moreover, we show that the modular closure of set can be computed in time $O(mn^2)$. The last section contains the conclusions and topics for further research.

2. Definitions and notations

A Boolean function $f : \{0,1\}^n \mapsto \{0,1\}$ is called monotone(positive) on $N=\{1,2,\ldots,n\}$, if $x \leqslant y \Rightarrow f(x) \leqslant f(y)$. A Boolean function f is *constant* if: $f \equiv 0$ (denoted by $f = \perp$) or $f \equiv 1$ (denoted by $\top$). A variable x_j of f is called *essential* if the *restrictions* respectively defined by: $f(x_j = 0) = f(x_1,\ldots,x_{j-1},0,x_{j+1},\ldots,x_n)$ and $f(x_j = 1) = f(x_1,\ldots,x_{j-1},1,x_{j+1},\ldots,x_n)$, are not identical. The *dual* of a function f is defined