ALGORITHMIC ASPECTS OF THE SUBSTITUTION DECOMPOSITION IN OPTIMIZATION OVER RELATIONS, SET SYSTEMS AND BOOLEAN FUNCTIONS

R.H. MÖHRING

*Institut für Ökonometrie und Operations Research, Universität Bonn,
Nassestrasse 2, D-5300 Bonn, FRG*

Abstract

In the last years, decomposition techniques have seen an increasing application to the solution of problems from operations research and combinatorial optimization, in particular in network theory and graph theory. This paper gives a broad treatment of a particular aspect of this approach, viz. the design of algorithms to compute the decomposition possibilities for a large class of discrete structures. The decomposition considered is the *substitution decomposition* (also known as modular decomposition, disjunctive decomposition, X-join or ordinal sum). Under rather general assumptions on the type of structure considered, these (possibly exponentially many) decomposition possibilities can be appropriately represented in a *composition tree* of polynomial size. The task of determining this tree is shown to be polynomially equivalent to the seemingly weaker task of determining the closed hull of a given set w.r.t. a closure operation associated with the substitution decomposition. Based on this reduction, we show that for arbitrary relations the composition tree can be constructed in polynomial time. For clutters and monotonic Boolean functions, this task of constructing the closed hull is shown to be Turing-reducible to the problem of determining the circuits of the independence system associated with the clutter or the prime implicants of the Boolean function. This leads to polynomial algorithms for special clutters or monotonic Boolean functions. However, these results seem not to be extendable to the general case, as we derive exponential lower bounds for oracle decomposition algorithms for arbitrary set systems and Boolean functions.

Keywords and phrases

Boolean function, clutter, combinatorial optimization, computational complexity, composition tree, decomposition algorithm, graph, independence system, matroid, modular decomposition, oracle algorithm, relation, set system, substitution decomposition.

1. Introduction

This paper deals with the algorithmic complexity of the substitution decomposition. This type of structural decomposition, which occurs in the literature under many different names such as *modular decomposition, ordinal sum* or *X-join*, has many applications in discrete mathematics, operation research and computer science, ranging from switching design (disjunctive decomposition of Boolean functions [1,14]) via reliability theory (modular decomposition of coherent systems [2,6]), and game theory (decomposition of simple n-person games [34,35]) to combinatorial (optimization) problems over graphs, networks and independence systems or clutters. A list of some of the major applications within this last field is given in table 1. In all these applications, the objective can be obtained in a two-step procedure by exploiting a (given) decomposition of the underlying structure in a natural way, cf. [29]. Furthermore, for the problems marked with an asterisk, it can be shown that the substitution decomposition is in fact the only possible two-step decomposition under certain, very weak assumptions on the decomposition approach, cf. [29] for more details.

Table 1

Some combinatorial (optimization) problems solvable by substitution decomposition

1. *Undirected graphs:*

   maximum weighted clique problem* [10], finding a transitive orientation [18], constructing perfect graphs [18], constructing the automorphism group [21]

2. *Partial orders:*

   minimal covering by chains/antichains [29], determining the dimension [22] and the Moebius function [29], counting partial orders [31]

3. *Project-networks* (cf. [29] for an overview):

   shortest overall duration*, time-cost tradeoff*, determining the distribution of the shortest overall duration in stochastic networks*

4. *(Acyclic) flow and reliability networks:*

   maximum/minimum flow* [33,23], reliability of a network [2,38]

5. *Scheduling problems:*

   scheduling with series-parallel precedence constraints and/or certain 'compatible' objectives [25,29,30]

6. *Clutters and independence systems:*

   solving general combinatorial optimization problems over clutters or independence systems* [5,24,29]

This large scope of application naturally makes it important to have efficient algorithms for finding all, or at least several, decomposition possibilities for a given structure. As yet, fast decomposition algorithms are known only for binary relations [7,12,19] and matroids [13]. For clutters and Boolean functions, however, the known methods involve either the solution of an NP-complete problem (as in [4]) or have exponential running time [15,16,36,37,40].

Given this background, it is the aim of this article to obtain more insight into the computational complexity of the substitution decomposition, both on a general level for 'abstract' discrete structures and for 'concrete' structures such as relations, set systems and Boolean functions.

The general approach is based on the fact that, in all applications, the decomposition possibilities of a structure S on a set A can be equivalently described by certain subsets of A. These so-called *S-autonomous sets* form a set system $\mathcal{A}(S)$ with certain set-theoretic properties (cf. sect. 2).

The properties of $\mathcal{A}(S)$ lead, in sect. 3, to the construction of the so-called *composition tree* $\mathcal{B}(S)$ of the structure S. This construction generalizes and unifies the tree constructions developed earlier for Boolean functions [14,15], clutters [35], graphs [11,19,32] and partial orders [7]. $\mathcal{B}(S)$ contains all the 'essential' information about the (possibly exponentially many) decomposition possibilities of the structure S in a polynomial (in $|A|$) number of nodes, which makes $\mathcal{B}(S)$ a suitable data structure for handling the decomposition possibilities.

In sect. 4, we aim at the determination of this composition tree. We show that this task is polynomially equivalent to two seemingly weaker tasks, viz. determining the smallest S-autonomous set containing a given set B (the *S-autonomous closure* of B), or deciding whether S is decomposable or not, and producing a nontrivial S-autonomous set if it is.

Based on the first polynomial reduction, we show in sect. 5 that the composition tree can be constructed in polynomial time for arbitrary k-ary relations.

For clutters, the determination of the tree is shown in sect. 6 to be Turing-reducible to the problem of determining the circuits of the independence system associated with the clutter. This leads to polynomial (in $|A|$) time decomposition algorithms for clutters $\mathcal{C}$, where $|\mathcal{C}|$ is polynomial in $|A|$ (e.g. bounded clutters) and the associated monotonic Boolean functions. The question whether arbitrary clutters admit polynomial time decomposition algorithms must, however, be answered negatively w.r.t. oracle algorithms. Even the apparently simple problem of recognizing decomposibility for clutters (and thus also for arbitrary set systems and Boolean functions) is shown to require an exponential number of steps for rather powerful oracle algorithms. This shows that efficient decomposition methods can almost certainly only be expected for special classes of set systems or Boolean functions.

2. Basic properties of the substitution decomposition

In this section we present the necessary definitions and properties of the substitution decomposition for relations, set systems and Boolean functions. Throughout this paper, we will consider only the *finite* case, i.e. structures with a finite ground set. For more information on the substitution decomposition, also in the infinite case, we refer to [29].

Let $\mathcal{S}$ be a class of structures (e.g. graphs, clutters, etc.). The substitution decomposition for the structures from $\mathcal{S}$ is the inverse operation of a *composition* or *substitution operation* which works as follows: Let S' be a structure from $\mathcal{S}$ on a set A' and let, for each $\beta \in A'$, S_β be a structure from $\mathcal{S}$ on a set A_β, where the sets A_β are non-empty and pairwise disjoint. The composition operation assigns to S' and S_β, $\beta \in A'$, a unique structure S on $A := \cup_{\beta \in A'} A_\beta$, which contains the structures $S_\beta, \beta \in A'$, as substructures (i.e. S_β equals the restriction $S|A_\beta$ of A to A_β), and in which the relationship between the different S_β is defined via S'. This structure S is called the *composition* of the structures S' and S_β, $\beta \in A'$, and is denoted by $S = S'[S_\beta, \beta \in A']$.

For k-ary ($k \geqslant 2$) *relations* R' on A', R_β on A_β, $\beta \in A'$, the composition $R = R'[R_\beta, \beta \in A']$ is defined by

$$(\alpha_1, \ldots, \alpha_k) \in R \;:\Longleftrightarrow\; \begin{cases} (\alpha_1, \ldots, \alpha_k) \in R_\beta & \text{for some } \beta \in A' \text{ or} \\ (\alpha_1, \ldots, \alpha_k) \in A_{\beta_1} \times \ldots \times A_{\beta_k} & \text{for some } (\beta_1, \ldots, \beta_k) \in R' \\ & \text{with } \quad |\{\beta_1, \ldots, \beta_k\}| > 1. \end{cases}$$

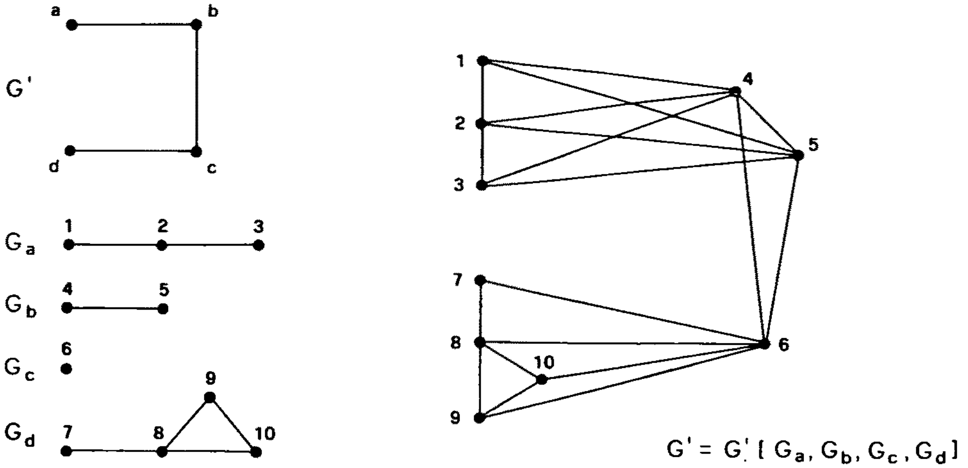


Fig. 1. A composition of graphs.

Special cases of this composition are the X-join for undirected graphs [19], the ordinal sum for partial orders [22], and the modular decomposition of networks [2]. An example for graphs is given in fig. 1.

For *set systems* $\mathcal{T}'$ on A', $\mathcal{T}_\beta$ on A_β, $\beta \in A'$, the composition $\mathcal{T} = \mathcal{T}'[\mathcal{T}_\beta, \beta \in A']$ is defined by

$$T \in \mathcal{T} \;:\Longleftrightarrow\; \begin{cases} \text{there exist } T' \in \mathcal{T}' \text{ and } T_\beta \in \mathcal{T}_\beta \text{ for each } \beta \in T' \\ \text{such that } T = \bigsqcup_{\beta \in T'} T_\beta \,. \end{cases}$$

Special cases here are the composition for clutters (i.e. set systems of pairwise incomparable sets) [4], and the modular decomposition of coherent systems [2,6].

For an illustration, consider the clutters $\mathcal{C}' = \{\{a,b\},\{b,c\},\{c,d\}\}$, $\mathcal{C}_a = \{\{1,2\},\{2,3\}\}$, $\mathcal{C}_b = \{\{4,5\}\}$, $\mathcal{C}_c = \{\{6\}\}$ and $\mathcal{C}_d = \{\{7,8\},\{8,9,10\}\}$ on the sets $\{a,b,c,d\}$, $\{1,2,3\}$, $\{4,5\}$, $\{6\}$, $\{7,8,9,10\}$, respectively. Then $\mathcal{C} = \mathcal{C}'[\mathcal{C}_a, \mathcal{C}_b, \mathcal{C}_c, \mathcal{C}_d] = \{\{1,2,4,5\},\{2,3,4,5\},\{4,5,6\},\{6,7,8\},\{6,8,9,10\}\}$.

For Boolean functions

$$F'(y_1,\ldots,y_m), F_1(x_1,\ldots,x_{k_1}),\ldots,F_m(x_{k_{m-1}+1},\ldots,x_n)$$

with disjoint sets of variables

$$\{x_1,\ldots,x_{k_1}\},\{x_{k_1+1},\ldots,x_{k_2}\},\ldots,\{x_{k_{m-1}+1},\ldots,x_n\},$$

the composition $F = F'[F_1,\ldots,F_m]$ is defined by

$$F(x_1,\ldots,x_n) = F'(F_1(x_1,\ldots,x_{k_1}),\ldots,F_m(x_{k_{m-1}+1},\ldots,x_n)),$$

i.e. it corresponds to the notion of *disjunctive decomposition* in switching theory [1,14].

As an example, let

$$F'(y_1,\ldots,y_4) = y_1 y_2 + y_2 y_3 + y_3 y_4,$$

$$F_1(x_1,x_2,x_3) = x_1 x_2 + x_2 x_3,$$

$$F_2(x_4,x_5) = x_4 x_5, \quad F_3(x_6) = x_6, \text{ and}$$

$$F_4(x_7,\ldots,x_{10}) = x_7 x_8 + x_8 x_9 x_{10}.$$

Then $F = F'[F_1, \ldots, F_4]$ is given as

$$F(x_1, \ldots, x_{10})$$

$$= (x_1 x_2 + x_2 x_3)(x_4 x_5) + (x_4 x_5)x_6 + x_6(x_7 x_8 + x_8 x_9 x_{10})$$

$$= x_1 x_2 x_4 x_5 + x_2 x_3 x_4 x_5 + x_4 x_5 x_6 + x_6 x_7 x_8 + x_6 x_8 x_9 x_{10}.$$

In all cases, we also say that the composed structure $S = S'[S_\beta, \beta \in A']$ is obtained by *substitution* of the elements $\beta \in A'$ in S by the structures S_β, $\beta \in A'$.

The partition $\pi = \{A_\beta | \beta \in A'\}$ of A into the ground sets A_β of S_β is called the *congruence partition* of S associated with the composition $S = S'[S_\beta, \beta \in A']$, and the structure S' is called the *quotient* of S modulo π and is denoted by S/π. Note that this definition covers the trivial cases that S/π is isomorphic to S, i.e. $\pi = \{\{\alpha\} | \alpha \in A\}$, and $|A'| = 1$, i.e. $\pi = \{A\}$. These two partitions are called the *trivial* congruence partitions of S.

The composition is *proper* if $|A'| > 1$ and $|A_\beta| > 1$ for some $\beta \in A'$. A structure S on A is said to be *decomposable* if it has a representation as a proper composition $S = S'[S_\beta, \beta \in A']$. A structure which is not decomposable is called *indecomposable* or *prime*. So a structure is decomposable iff it has a non-trivial congruence partition, i.e. a congruence partition different from $\{\{\alpha\} | \alpha \in A\}$ and $\{A\}$.

Although the composition operations for the three classes of structures are defined in quite different ways, there are some very strong links between them.

In fact, it was these links that motivated the introduction and investigation of general decomposition models. For the substitution decomposition, such a model is given in [27,29]. A different model dealing with the so-called *split decomposition* is given in [13]. The split decomposition is closely related to the substitution decomposition, but more general. It gives more structural insights (in particular w.r.t. the characterization of certain highly decomposable structures in the sense of proposition 3.4), but seems to be less applicable to the solution of combinatorial optimization problems, cf. [29].

To go into more detail, the link between relations and set systems is given by *graphs* and *conformal clutters* (i.e. clutters of maximal cliques of a graph), cf. [9,28]. If we denote by $\mathcal{C}(G)$ the clutter of maximal cliques of graph G, and by $G(\mathcal{C})$ the graph defining the conformal clutter $\mathcal{C}$, then

$$\mathcal{C}(G'[G_\beta,\ \beta \in A']) = \mathcal{C}(G')\,[\mathcal{C}(G_\beta),\ \beta \in A'] \quad \text{and}$$

$$G(\mathcal{C}'[\mathcal{C}_\beta,\ \beta \in A']) = G(\mathcal{C}')\,[G(\mathcal{C}_\beta),\ \beta \in A'] \ ,$$

i.e. the composition operations are essentially equal.

The link between set systems and Boolean functions is given by *monotonic Boolean functions* and the corresponding *clutters* of their *prime implicants*, cf. [3,6]. If we denote by $\mathcal{C}(F)$ the clutter of prime implicants of F, and by $F(\mathcal{C})$ the monotonic Boolean function whose prime implicants are the sets of $\mathcal{C}$, then

$$\mathcal{C}(F'[F_1,\dots,F_m]) = \mathcal{C}(F')\,[\mathcal{C}(F_1),\dots,\mathcal{C}(F_m)] \quad \text{and}$$

$$F(\mathcal{C}\,[\mathcal{C}_1,\dots,\mathcal{C}_m]) = F(\mathcal{C})\ [F(\mathcal{C}_1),\dots,F(\mathcal{C}_m)].$$

Furthemore, if $F = F'[F_1, \dots, F_m]$, where F is monotonic but F' and the F_i need not be, then there exist monotonic Boolean functions $G', G_1, \dots, G_m$ with the same variables as $F', F_1, \dots, F_m$, respectively, such that $F = G'[G_1, \dots, G_m]$. In other words, the decomposition possibilities of a monotonic Boolean function are the same as those for its clutter of prime implicants.

These links are also demonstrated by the above examples, where $\mathcal{C} = \mathcal{C}(G)$ and $F = F(\mathcal{C})$.

In the rest of this section we will characterize the decomposition possibilities of a structure S by 'internal' properties of the classes of the corresponding congruence partition. Proofs of these results can be found for k-ary relations in [26,29,39], for clutters in [6,8,28], for arbitrary set systems in [29], and for Boolean functions in [1,14,15].

Let S be a structure on A. A subset B of A is called *S-autonomous*, if it is a class of some congruence partition of S.

PROPOSITION 2.1

(a) Let R be a k-ary relation on A. Then $B \subset A$ is R-autonomous iff in all $(\alpha_1, \dots, \alpha_k) \in R$ with $\{\alpha_1, \dots, \alpha_k\} \cap (A \setminus B) \neq \emptyset$ each $\alpha_j \in B$ can be replaced by any $\beta \in B$, i.e. $(\alpha_1, \dots, \alpha_{j-1}, \beta, \alpha_{j+1}, \dots, \alpha_k) \in R$.

(b) Let $\mathcal{T}$ be a set system on A. Then $B \subseteq A$ is $\mathcal{T}$-autonomous iff for all $T_1, T_2 \in \mathcal{T}$ which meet B (i.e. $T_i \cap B \neq \emptyset$) also the *exchange* $\mathrm{Ex}(T_1, B, T_2) = (T_1 \setminus B) \cup (T_2 \cap B)$ belongs to $\mathcal{T}$.

(c) Let F be a Boolean function with variables $x_1, \dots, x_n$. A set $B \subseteq \{x_1, \dots, x_n\}$ is F-autonomous iff it fulfills the Ashenhurst decomposition chart conditions (i.e. is a *bound set* in the sense of [1,14]).

In all cases, a partition of the base set of the structure S is a congruence partition of S iff all its classes are S-autonomous.

Let $\mathcal{A}(S)$ denote the system of S-autonomous sets. For all structures considered, $\mathcal{A}(S)$ has certain common properties. To formulate them let, for a structure S on A

and $B \in \mathcal{A}(S)$, $S \mid B$ denote the *substructure* of S induced by B, i.e. $R \mid B = R \cap B^k$ for a k-ary relation, $\mathcal{T} \mid B = \{T \cap B \mid T \in \mathcal{T},\ T \cap B \neq \emptyset\}$ for set systems, and $F \mid B(x_B) = F(x_B, c_{A \setminus B})$, where $x_B = (x_j \mid j \in B)$ and $c_{A\setminus B}$ is a suitable choice of constants for x_j, $j \notin B$, cf. [29].

Furthermore, for a congruence partition π of S, let η_π denote the *canonical mapping* from S onto S/π associated with π, i.e. $\eta_\pi(\alpha) = \eta_\pi(\beta)$ iff α and β are contained in the same class of π.

PROPOSITION 2.2

Let S be a relation, set system or Boolean function with ground set A. Then $\mathcal{A}(S)$ has the following properties:

(A1) $A \in \mathcal{A}(S), \{\alpha\} \in \mathcal{A}(S)$ for all $\alpha \in A$.

(A2) If B, $C \in \mathcal{A}(S)$ overlap, i.e. if $B \setminus C$, $B \cap C$ and $C \setminus B$ are non-empty, then $B \setminus C$, $B \cap C$, $C \setminus B$ and $B \cup C$ all belong to $\mathcal{A}(S)$.

(A3) Let $B \in \mathcal{A}(S)$ and $S \mid B$ be the substructure of S induced by B. Then $\mathcal{A}(S \mid B) = \{C \in \mathcal{A}(S) \mid C \subseteq B\}$.

(A4) Let π be a congruence partition of S and S/π be the corresponding quotient. Then

$$\eta_\pi(B) \in \mathcal{A}(S/\pi) \quad \text{for all } B \in \mathcal{A}(S)$$

$$\eta_\pi^{-1}(B') \in \mathcal{A}(S) \quad \text{for all } B' \in \mathcal{A}(S/\pi).$$

The sets in (A1) are called the *trivial* S-autonomous sets. If in (A2) the symmetric difference $B \mathbin{\Delta} C := (B \setminus C) \cup (C \setminus B)$ is also S-autonomous, then $\mathcal{A}(S)$ is said to be *symmetrically closed.* This is always the case for set systems and Boolean functions, but not, in general, for relation.

3. The composition tree

In this section we will develop a representation of the system $\mathcal{A}(S)$ in a tree, the so-called composition tree of S. A broad discussion of the principles leading to this tree representation, which is based on very general assumptions and also covers the infinite case, is given in the algebraic decomposition model in [27,29]. Since these construction principles (in particular theorems 3.5, 3.7 and proposition 3.4) are subsequently needed in sects. 4 and 6, we will present a simplified construction here, which is based on the properties (A1) – (A4) of $\mathcal{A}(S)$ and the fact that the congruence partitions of S are exactly the partitions of A into S-autonomous sets. In this way, we generalize and unify the tree constructions for clutters [4,35] and special relations [7,11,19], which make essential use of the underlying structure. A similar, but less far-reaching approach is implicitly also contained in [39]. Finally, note that the split decomposition leads to a similar (undirected rather than directed) tree which also applies for the substitution decomposition of binary relations and set systems [13].

The construction of the tree is based on two mutually exclusive decomposition principles, the first of which is the 'maximal disjoint decomposition':

Let S be a structure on A. A *maximal disjoint decomposition* of S is a partition $\sigma^\star$ of A into $\subseteq$-maximal S-autonomous sets $B \neq A$.

The following decomposition principle is then obvious.

PROPOSITION 3.1

If S admits a maximal disjoint decomposition $\sigma^\star$, then each S-autonomous set is either equal to A or $S \mid B$-autonomous for some $B \in \sigma^\star$, where $S \mid B$ denotes the substructure of S induced by B.

Furthermore, $\sigma^\star$ is the coarsest congruence partition π of S such that S/π is prime.

An important role for the existence of a maximal disjoint decomposition is played by certain highly decomposable structures:

A structure S on A is called *semi-linear* if there is a linear order $\leqslant$ on A such that $\mathcal{A}(S)$ contains $\mathcal{A}(\leqslant)$, i.e. all intervals $[\alpha,\beta] = \{\gamma \in A \mid \alpha \leqslant \beta \leqslant \gamma\}$ of $\leqslant$. By $\mathfrak{L}(S)$ we denote the system of all congruence partitions π of a structure S for which S/π is semi-linear.

PROPOSITION 3.2

A structure S on A has a maximal disjoint decomposition iff S has no *proper* semi-linear quotients, i.e. $|\pi| \leqslant 2$ for all $\pi \in \mathfrak{L}(S)$.

Proof

Assume that $\sigma^\star$ exists and that S has a proper semi-linear quotient S/π with base set $A' = \{\beta_1, \ldots, \beta_m\}$ and associated linear order $\beta_1 \leqslant \beta_2 \leqslant \ldots \leqslant \beta_m$. Then $B_1' = \{\beta_1, \ldots, \beta_{m-1}\}$ and $B_2' = \{\beta_2, \ldots, \beta_m\}$ are S/π-autonomous and thus because of (A4) the re-image sets $B_i = \eta_\pi^{-1}(B_i')$ $(i = 1, 2)$ are S-autonomous. But then the maximal autonomous sets $C_i \in \sigma^\star$ containing B_i $(i = 1, 2)$ overlap, a contradiction.

In the converse direction, let $\sigma^\star = \{C_1, \ldots, C_k\}$ be the collection of maximal S-autonomous sets $C_i \neq A$. If two C_i overlap, say C_1 and C_2, then $\pi = \{C_1 \setminus C_2, C_1 \cap C_2, C_2 \setminus C_1\}$ is a congruence partition of S because of (A2) and the maximality of the C_i. It follows from (A4) that S/π is semi-linear, a contradiction. □

It turns out that semi-linear structures can be completely characterized, both on the abstract level and for the concrete types of structures considered.

Let S be semi-linear with associated linear order $\leqslant$ on A. If $\mathcal{A}(S) = \mathcal{A}(\leqslant)$, we say that S is *linear*, if $\mathcal{A}(S) = 2^A$, the power set of A, we say that S is *degenerate*.

PROPOSITION 3.3

Let S be a semi-linear structure on A with $|A| \geqslant 3$. Then S is either linear or degenerate.

Proof

Let $\leqslant$ be a linear order on A such that $\mathcal{A}(\leqslant) \subseteq \mathcal{A}(S)$ and assume that $\mathcal{A}(\leqslant) \neq \mathcal{A}(S)$. We will show that S is degenerate. Because of (A2), it obviously suffices to show that there is $\alpha_1 \in A$ such that $\{\alpha_1, \alpha\} \in \mathcal{A}(S)$ for all $\alpha \in A$. This is done as follows.

Since $\mathcal{A}(\leqslant) \neq \mathcal{A}(S)$, there exist $C_0 \in \mathcal{A}(S)$, $\beta_1, \beta_2 \in C_0$ and $\gamma \notin C_0$ such that $\beta_1 < \gamma < \beta_2$. Let α_1 and α_2 be the greatest and least elements from C_0 such that $\alpha_1 < \gamma < \alpha_2$. Then $\{\alpha_1, \alpha_2\} = [\alpha_1, \alpha_2] \cap C_0 \in \mathcal{A}(S)$ because of $\mathcal{A}(\leqslant) \subseteq \mathcal{A}(S)$ and (A2).

We shall show that $\{\alpha_1, \alpha\} \in \mathcal{A}(S)$ for all $\alpha \in A$. If, for instance, $\alpha > \alpha_2$, then $B_1 := \{\alpha_1, \alpha_2\} \cup \,]\gamma, \alpha] \in \mathcal{A}(S)$ and $B_2 := [\gamma, \alpha[\, \in \mathcal{A}(S)$. (Observe that $]\gamma, x] = [\gamma, \alpha] \setminus \{\gamma\}$ and $[\gamma, \alpha[\, = [\gamma, \alpha] \setminus \{\alpha\}$ are intervals of $\leqslant$). Hence, also $\{\alpha_1\ \alpha\} = B_1 \setminus B_2 \in \mathcal{A}(S)$. The cases $\alpha_1 < \alpha < \alpha_2$ and $\alpha < \alpha_1$ follow similarly. □

The property that there are only two types of semi-linear structures also remains valid for the split decomposition, cf. [13]. For the concrete structures considered here, the following characterization of these two types can be given, cf. [13,29].

PROPOSITION 3.4

(a) Let R be a k-ary relation on A. R is degenerate iff R is (up to reflexive k-tuples $(\alpha, \ldots, \alpha)$) empty or complete, i.e. $R = \emptyset$ or $R = A^k$. R is linear iff R is (up to reflexive pairs (α, α)) a linear order.

(b) Let $\mathcal{C}$ be a set system on A. $\mathcal{C}$ is degenerate iff $\mathcal{C} = \{\{\alpha\} \mid \alpha \in A\}$, $\mathcal{C} = 2^A$, or $\mathcal{C} = \{B \in 2^A \mid B \supseteq B_0\}$ for some $\emptyset \neq B_0 \subseteq A$. Since $\mathcal{A}(\mathcal{C})$ is symmetrically closed, there are no linear set systems.

(c) Let F be a Boolean function with variables $x_1, \ldots, x_n$. F is degenerate iff $F(x_1, \ldots, x_n) = y_1 \star \ldots \star y_n$, where $y_i = x_i$ or $y_i = \bar{x}_i$ and $\star$ denotes Boolean sum $(+)$, Boolean product $(\cdot)$ or ring sum addition $(\oplus)$. Since $\mathcal{A}(F)$ is symmetrically closed, there are no linear Boolean functions.

It is easy to see by means of (A4) that each quotient of a linear or degenerate structure is again linear or degenerate, respectively. This suggests to represent all these quotients by a 'largest' linear or degenerate structure $S^\star = S/\pi^\star$, if such a structure exists. The following theorem shows that this is indeed the case. The corresponding finest partition $\pi^\star \in \mathcal{L}(S)$ then yields the second decomposition principle.

To this end, we introduce the following notation. Given $\pi = \{C_1, \dots, C_m\} \in \mathcal{L}(S)$, we will assume that the quotient S/π has the ground set $A' = \{\beta_1, \dots, \beta_m\}$ with $\eta_\pi(\alpha) = \beta_i$ iff $\alpha \in C_i$. If S/π is linear, the associated linear order will w.l.o.g. be $\beta_1 \leqslant \dots \leqslant \beta_m$. This induces the linear order $C_1 \leqslant \dots \leqslant C_m$ on π. Then we denote the system of all *interval unions* $C_i \cup C_{i+1} \cup \dots \cup C_{j-1} \cup C_j$ $(1 \leqslant i < j \leqslant m)$ of classes of π by $\mathcal{A}^{\text{lin}}(\pi)$, and the system of all unions $C_{i_1} \cup C_{i_2} \cup \dots \cup C_{i_k}$ $(1 \leqslant i_1 < i_2 < \dots < i_k \leqslant m)$ by $\mathcal{A}^{\text{deg}}(\pi)$.

THEOREM 3.5

Let S be a structure on A. Then $\mathcal{L}(S)$ contains a finest partition $\pi^\star = \{C_1, \dots, C_m\}$, $m \geqslant 1$, and there are two cases:

(a) $S/\pi^\star$ is linear. Then $\mathcal{A}(S)$ decomposes into

$$\mathcal{A}(S) = \mathcal{A}(S \mid C_1) \cup \dots \cup \mathcal{A}(S \mid C_m) \cup \mathcal{A}^{\text{lin}}(\pi^\star)$$

(b) $S/\pi^\star$ is degenerate. Then $\mathcal{A}(S)$ decomposes into

$$\mathcal{A}(S) = \mathcal{A}(S \mid C_1) \cup \dots \cup \mathcal{A}(S \mid C_m) \cup \mathcal{A}^{\text{deg}}(\pi^\star).$$

Proof

We show first that if $\pi_1, \pi_2 \in \mathcal{L}(S)$, then their intersection $\pi_1 \wedge \pi_2 = \{P \cap Q \mid P \in \pi_1, Q \in \pi_2, P \cap Q \neq \emptyset\}$ also belongs to $\mathcal{L}(S)$. Since $\mathcal{L}(S)$ is finite, this then yields the existence of $\pi^\star$.

Assume first that S/π_1 and S/π_2 are both linear. Let $P_1 \leqslant_1 P_2 \leqslant_1 \dots \leqslant_1 P_m$ and $Q_1 \leqslant_2 Q_2 \leqslant_2 \dots \leqslant_2 Q_n$ be the associated linear orders on $\pi_1 = \{P_1, \dots, P_n\}$ and $\pi_2 = \{Q_1, \dots, Q_m\}$. We may assume that $\pi_1 \neq \pi_1 \wedge \pi_2 \neq \pi_2$. Then there are classes $P_{i_1} \cap Q_{j_1}$ and $P_{i_2} \cap Q_{j_2}$ in $\pi_1 \wedge \pi_2$ such that $P_{i_1} \neq P_{i_2}$ and $Q_{j_1} \neq Q_{j_2}$. Assume w.l.o.g. that $P_{i_1} \leqslant_1 P_{i_2}$ and $Q_{j_1} \leqslant_2 Q_{j_2}$ (otherwise take the dual of $\leqslant_2$ and renumber the classes of π_2 appropriately). This fixation of $\leqslant_2$ with respect to $\leqslant_1$ induces the following compatibility between $\leqslant_1$ and $\leqslant_2$:

$$(\star) \begin{cases} \text{Let } P_i \neq P_j,\ Q_r \neq Q_s, \text{ and } P_i \cap Q_r \neq \emptyset \neq P_j \cap Q_s. \\ \text{Then } P_i <_1 P_j \text{ iff } Q_r <_2 Q_s. \end{cases}$$

The reason for this compatibility is given by the linearity of S/π_1 and S/π_2 which preserves the 'betweenness relation' for classes of π_1 and π_2, respectively. For a detailed proof of this property, cf. [27].

After this fixation of $\leqslant_2$ w.r.t. $\leqslant_1$, we order the classes of $\pi_1 \wedge \pi_2$ lexicographically according to

$$P_i \cap Q_j \leqslant P_r \cap Q_s \;:\Longleftrightarrow\; P_i <_1 P_r \;\text{ or }\; P_i = P_r \;\text{ and }\; Q_j <_2 Q_s\,.$$

We shall show that for this linear order $\leqslant$ on $\pi_1 \wedge \pi_2$ (more exactly: for the corresponding linear order on the quotient set) $\mathcal{A}(\leqslant) \subseteq \mathcal{A}(S/\pi_1 \wedge \pi_2)$, which proves that $\pi_1 \wedge \pi_2 \in \mathfrak{C}(S)$. But this is because of (A4), equivalent to showing that $\mathcal{A}^{\text{lin}}(\pi_1 \wedge \pi_2) \subseteq \mathcal{A}(S)$. So let $B := (P_{i_1} \cap Q_{j_1}) \cup \ldots \cup (P_{i_k} \cap Q_{j_k}) \in \mathcal{A}^{\text{lin}}(\pi_1 \wedge \pi_2)$. It follows from the definition of $\leqslant$, the compatibility relation ($\star$) and the linearity of S/π_1 and S/π_2, that $B_1 := P_{i_1} \cup \ldots \cup P_{i_k} \in \mathcal{A}(S)$ and $B_2 := Q_{j_1} \cup \ldots \cup Q_{j_k} \in \mathcal{A}(S)$. Then $B = B_1 \cap B_2 \in \mathcal{A}(S)$ because of (A2).

The cases in which one or both of S/π_1, S/π_2 are degenerate follow analogously.

To show (a), assume that there exist $C \in \mathcal{A}(S)$ and $L_1, L_2 \in \pi^\star$, with $L_1 \backslash C \neq \emptyset$, $L_1 \cap C \neq \emptyset$ and $L_2 \cap C \neq \emptyset$. Let $\leqslant$ denote the linear order on $\pi^\star$ and let w.l.o.g. $L_1 < L_2$. Then

$$B := \Bigl(C \cup \bigcup_{L_1 < L \leqslant L_2} L\Bigr) \cap \Bigl(\bigcup_{L_1 \leqslant L \leqslant L_2} L\Bigr)\,,\quad L_1\backslash B \;\text{ and }\; L_1 \cap B$$

are S-autonomous and $\sigma := (\pi^\star \backslash \{L_1\}) \cup \{L_1\backslash B,\, L_1 \cap B\}$ is a congruence partition of S which refines $\pi^\star$. It can then be verified that $\sigma \in \mathfrak{C}(S)$ (an associated linear order $\precsim$ is obtained from $\leqslant$ by replacing L_1 by $L_1\backslash B$ and $L_1 \cap B$ with the order $L_1\backslash B \precsim L_1 \cap B$). This contradicts the fact that $\pi^\star$ is the least partition in $\mathfrak{C}(S)$.

(b) follows analogously. □

Based on the two composition principles, we can assign the following *composition tree* $\mathfrak{B}(S)$ to a structure S on A.

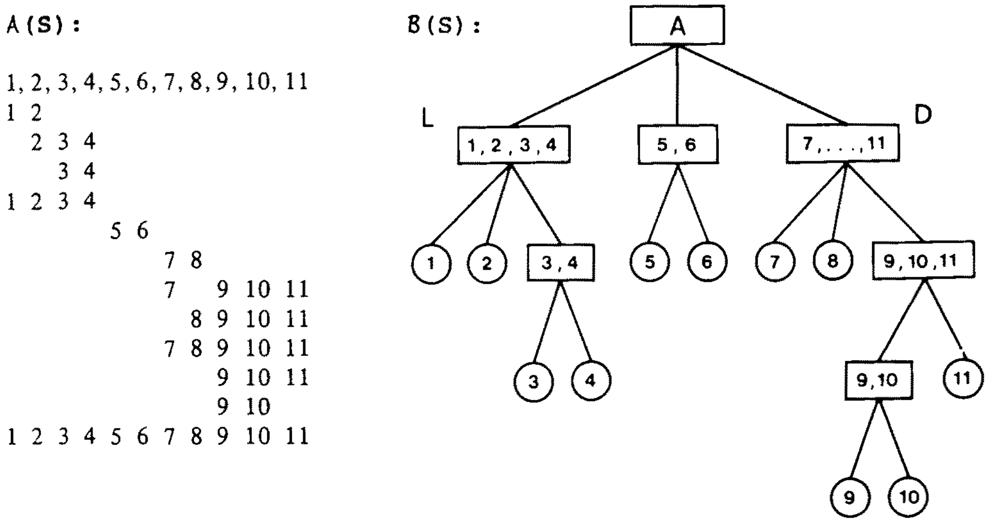
(1) The root of $\mathfrak{B}(S)$ is the set A. Each node of $\mathfrak{B}(S)$ is an S-autonomous set.

(2) If the first decomposition principle applies to a node B of $\mathfrak{B}(S)$, i.e. if $S \mid B$ has a maximal disjoint decomposition $\sigma^\star(B) = \{B_1, \ldots, B_m\}$, then $B_1, \ldots, B_m$ are the immediate successors of B.

(3) If the first decomposition principle does not apply to a node B of $\mathfrak{B}(S)$, then $\mathfrak{C}(S \mid B)$ has a finest partition $\pi^\star(B) = \{B_1, \ldots, B_m\}$, with $m \geqslant 3$. Then $B_1, \ldots, B_m$ are the immediate successors of B and B is labeled with D if $(S \mid B)/\pi^\star(B)$ is degenerate and with L if $(S \mid B)/\pi^\star(B)$ is linear.

(4) The leaves of $\mathfrak{B}(S)$ are the singletons $\{\alpha\} \in \mathcal{A}(S)$.

As an example, consider the structure S on $A = \{1, \ldots, 11\}$ in fig. 2.



```
A(S):

1,2,3,4,5,6,7,8,9,10,11
1 2
  2 3 4
    3 4
1 2 3 4
        5 6
            7 8
            7   9 10 11
              8 9 10 11
            7 8 9 10 11
                9 10 11
                9 10
1 2 3 4 5 6 7 8 9 10 11
```

Fig. 2. A structure S and its composition tree.

We shall now show that $\mathcal{B}(S)$ is sufficiently small when compared with the whole system $\mathcal{A}(S)$ (which may be exponential in $|A|$), but that $\mathcal{B}(S)$ nevertheless contains all the information on $\mathcal{A}(S)$ in a very straightforward way.

THEOREM 3.6

Let S be a structure on A, and let $|\mathcal{B}(S)|$ denote the number of nodes of $\mathcal{B}(S)$. Then $|\mathcal{B}(S)| \leqslant 2 \cdot |A| - 1$. This bound is tight for relations, set systems and Boolean functions.

Proof

Induction on $|\mathcal{B}(S)|$. □

THEOREM 3.7

Let S be a structure on A. A subset C of A is S-autonomous iff one of the following cases applies:

(1) C is a node of $\mathcal{B}(S)$.

(2) C is the interval union of immediate successors of a node labeled with L (i.e. $C \in \mathcal{A}^{\text{lin}}(\pi^\star(B))$ for some node B of $\mathcal{B}(G)$ labeled with L).

(3) C is the union of immediate successors of a node labeled with D (i.e. $C \in \mathcal{A}^{\deg}(\pi^\star(B))$ for some node B of $\mathfrak{B}(G)$ labeled with D).

Proof

(A3) and theorems 3.1 and 3.5. □

The last two theorems make $\mathfrak{B}(S)$ an appropriate data structure for algorithmic approaches to the substitution decomposition.

4. Tasks polynomially equivalent to the determination of $\mathfrak{B}(S)$

In order to obtain some insights into algorithms for determining the decomposition possibilities of a structure, we consider the following algorithmic tasks.

Task 1: Input: A structure S on A.
Output: 'Prime', if S is prime.
A non-trivial S-autonomous set, otherwise.

Task 2: Input: A structure S on A, a subset B of A.
Output: The *S-autonomous closure* $B^\star$ of B, which is defined as the least S-autonomous set containing B (which exists because of (A2)).

Task 3: Input: A structure S on A.
Output: The composition tree $\mathfrak{B}(S)$.

Apparently, these three tasks are of increasing ability. In particular, task 1 does what one would consider as a minimal requirement in order to carry out a decomposition, while task 3 determines all decomposition possibilities of S in the form of $\mathfrak{B}(S)$. It is therefore somewhat surprising that all three tasks turn out to be essentially equivalent in the sense that (up to certain structural operations) they are Turing reducible [17] to one other. To formulate the result, let $P_i(n)$ $(i = 1, 2, 3)$ denote the worst-case complexity of algorithm P_i for task i, when applied to structures on $A = \{1, \ldots, n\}$. Furthermore, let $Q_1(n)$ denote the complexity of testing a given set for S-autonomy, let $Q_2(n)$ denote the complexity of constructing a substructure $S \mid B$, and let $Q_3(n)$ denote the complexity of constructing a quotient S/π_B, where $\pi_B := \{B, \{\alpha\} \mid \alpha \in A \setminus B\}$.

THEOREM 4.1

(a) For each algorithm P_1 there is an algorithm P_3 with

$$P_3(n) = O(n^2 P_1(n) + n^2 Q_1(n) + n^2 Q_2(n) + n Q_3(n) + n^4).$$

(b) For each algorithm P_2 there is an algorithm P_3 with

$$P_3(n) = O(n^3 P_2(n)).$$

(c) For each algorithm P_3 there are algorithms P_1 and P_2 with

$$P_1(n) = O(1) + P_3(n), \quad P_2(n) = O(n^2) + P_3(n).$$

The proof of this theorem is based on the following two lemmas which follow easily from theorem 3.7 (cf. also the example in fig. 2).

LEMMA 4.2

Let B be a node of $\mathcal{B}(S)$ and let $C_1, \ldots, C_m$ be the $\subseteq$-maximal S-autonomous proper subsets of B.

(a) If $C_1 \cap C_2 = \emptyset$, then $S \mid B$ has the maximal disjoint decomposition $\sigma^\star(B) = \{C_1, \ldots, C_m\}$.

(b) If $C_1 \cap C_2 \neq \emptyset$ and $(C_1 \vartriangle C_2)^\star = (C_1 \vartriangle C_2)$ (i.e. $C_1 \vartriangle C_2 \in \mathcal{A}(S)$), then B is labeled with D. In this case, $\pi^\star(B) = \{B \setminus C_1, \ldots, B \setminus C_m\}$.

(c) If $C_1 \cap C_2 \neq \emptyset$ and $(C_1 \vartriangle C_2)^\star = B$ (i.e. $C_1 \vartriangle C_2 \notin \mathcal{A}(S)$), then B is labeled with L. In this case there exists a unique maximal strictly increasing sequence $D_1 = C_1 \setminus C_2 \subset D_2 \subset \ldots \subset D_k = B$ of S-autonomous sets. Then $\pi^\star(B) = \{D_1, D_2 \setminus D_1, \ldots, D_k \setminus D_{k-1}\}$ and $D_1 \leqslant D_2 \setminus D_1 \leqslant \ldots \leqslant D_k \setminus D_{k-1}$ is an associated linear order on $\pi^\star(B)$.

LEMMA 4.3

Let B be a node of $\mathcal{B}(S)$ and let $\emptyset \neq C_1 \subset \ldots \subset C_m = B$ be a strictly increasing sequence of S-autonomous sets.

(a) If $C_m \setminus C_{m-1} \notin \mathcal{A}(S)$, then $S \mid B$ has a maximal disjoint decomposition $\sigma^\star(B)$ and $C_{m-1} \in \sigma^\star(B)$.

(b) If $C_m \setminus C_{m-1} \in \mathcal{A}(S)$ but $C_m \setminus C_{m-2} \notin \mathcal{A}(S)$ or $|C_{m-1}| = 1$, then $S \mid B$ has the maximal disjoint decomposition $\sigma^\star(B) = \{C_{m-1}, B \setminus C_{m-1}\}$.

(c) If $C_m \setminus C_{m-1} \in \mathcal{A}(S)$ and $C_m \setminus C_{m-2} \in \mathcal{A}(S)$, then the second decomposition principle applies to B. The associated partition $\pi^\star(B) = \{B_1, \ldots, B_k\}$ is given by $B_i := C_{m-i+1} \setminus C_{m-i}$, $i = 1, \ldots, k$, where k is the first index below m such that $C_m \setminus C_{m-k-1} \notin \mathcal{A}(S)$. If, in this situation, $B_1 \cup B_k$ is S-autonomous, then B is labeled with D. Otherwise, B is labeled with L and $B_1 \leqslant B_2 \leqslant \ldots \leqslant B_k$ is an associated linear order.

Proof of theorem 4.1

(a) Let algorithm P_1 for task 1 be given.

Claim 1: If S is decomposable on A with $|A| = n$, one can construct an $\subseteq$-minimal non-trivial autonomous set in $O(n \cdot P_1(n) + n \cdot Q_2(n))$ time.

Since S is decomposable, P_1 finds a non-trivial S-autonomous set B_1. Then apply P_1 to $S \mid B_1$ and continue iteratively until P_1 arrives at an indecomposable substructure $(S \mid B_{k-1}) \mid B_k = S \mid B_k$. Because of (A3), B_k is a minimal non-trivial S-autonomous set.

Claim 2: If S is decomposable on A with $|A| = n$, one can construct in $O(n^2 \cdot P_1(n) + n^2 \cdot Q_2(n) + n \cdot Q_3(n))$ time a *composition series* of S, i.e. a sequence $S = S_1, S_2, \ldots, S_{m+1}$ of at most n structures S_i on A_i with the following properties:

(i) $S_{i+1} = S_i/\pi$, where $\pi_i = \{B_i, \{\alpha\} \mid \alpha \in A_i \backslash B_i\}$ and B_i is a minimal S-autonomous set with $|B_i| > 1$.

(ii) $|A_{m+1}| = 1$.

To obtain this sequence, find the set B_i of S_i by applying claim 1 and construct S_i/π_i. This takes $O(n^2 \cdot P_1(n) + n^2 \cdot Q_2(n)) + Q_3(n)$ time for each i. Since obviously $m \leqslant n$, claim 2 follows.

$S_1, \ldots, S_{m+1}$ may be regarded as a sequence of successive 'smallest' decompositions of the initial structure S. In the example of fig. 2, a possible sequence of sets B_i would be $B_1 = \{3, 4\} \rightarrow \overline{34}$, $B_2 = \{2, \overline{34}\} \rightarrow \overline{234}$, $B_3 = \{5, 6\} \rightarrow \overline{56}$, $B_4 = \{9, 10\} \rightarrow \overline{910}$, $B_5 = \{\overline{910}, 11\} \rightarrow \overline{91011}$, $B_6 = \{7, 8\} \rightarrow \overline{78}$, $B_7 = \{\overline{78}, \overline{91011}\} \rightarrow \overline{7891011}$, $B_9 = \{1, \overline{234}\} \rightarrow \overline{1234}$, $B_{10} = \{\overline{1234}, \overline{56}, \overline{7891011}\} \rightarrow \overline{1 \ldots 11}$, where we denote the element in A_{i+1} representing the set B_i in S_i by a bar.

Note that to each set B_i in this sequence, there corresponds an S-autonomous set C_i which represents the extent of the decomposition up to i. These C_i are the barred expressions $C_1 = \{3,4\}$, $C_2 = \{2, 3, 4\}$, etc. The whole sequence $C_1, \ldots, C_m$, with $C_m = A$, can of course be constructed simultaneously with the sequence $B_1, \ldots, B_m$. It will be used several times in the proof.

Now suppose that node B of $\mathfrak{B}(S)$ has already been constructed. Then we can construct a sequence in the sense of lemma 4.3 by taking from the above constructed sequence $C_1, \ldots, C_m$ the subsequence $C_{i_1}, \ldots, C_{i_k}$ with $C_{i_j} \subseteq B$ and by deleting from this subsequence all C_{i_j} with $C_{i_{j-1}} \not\subseteq C_{i_j}$, $j = 2, \ldots, k$. This takes $O(n^2)$ time.

To the resulting sequence $C'_1, \ldots, C'_\varrho$ we can apply lemma 4.3 and obtain the successors of B (and also the label and an associated linear order) in the cases (b) and (c) of the lemma in $O(n^2 + nQ_1(n))$ time. In case (a), we can find the other immediate successors of B among the sets $C'_1, \ldots, C'_\varrho$ by deleting from the inverse sequence $C'_\varrho, \ldots, C'_1$ all sets C'_i with $C'_i \subset C'_j$ for some $j > i$. This requires at most $O(n^3)$ time.

Putting all these steps together, and taking into account that $\mathfrak{B}(S)$ has at most $n - 1$ non-trivial nodes (theorem 3.6), we obtain an algorithm P_3 for task 3 with $P_3(n) = O(n^2 \cdot P_1(n) + n^2 \cdot Q_1(n) + n^2 \cdot Q_2(n) + n \cdot Q_3(n) + n^4)$.

(b) Let algorithm P_2 for task 2 be given, let S be a structure on A with $|A| = n$, and let B be an already constructed node of $\mathfrak{B}(S)$.

Claim 1: One can construct two different $\subseteq$-maximal S-autonomous sets $C_1, C_2 \neq B$ in $O(|B| \cdot P_2(n))$ time.

To obtain C_1, construct successively autonomous closures $\{\alpha_1\} = D_0 \subset D_1 \subset \ldots \subset D_k = C_1$ by putting $D_{i+1} := (D_i \cup \{\beta\})^\star$ for some $\beta \in B \setminus D_i$ with $(D_i \cup \{\beta\})^\star \neq B$. If there exists no such β, then $D_i = C_1$. Then take $\alpha_2 \in B \setminus C_1$ and proceed analogously for C_2.

Having constructed C_1 and C_2, we can apply lemma 4.2 to determine the type of node B. There are three cases:

(i) Case (a) of lemma 4.2 applies. Then $C_1, C_2 \in \sigma^\star(B)$ and if $B \neq C_1 \cup C_2$, the other members $C_3, \ldots, C_m$ of $\sigma^\star(B)$ are obtained by constructing for $\alpha \in B \setminus (C_1 \cup \ldots \cup C_i)$ the maximal S-autonomous set $D \subset B$ with $\alpha \in D$ according to claim 1. Then $D = C_{i+1}$. This requires at most $O(|B|^2 \cdot P_2(n))$ time.

(ii) Case (b) of lemma 4.2 applies. Then $\pi^\star(B) = \{B_1, \ldots, B_m\}$ can be constructed as follows: $B_1 := C_1 \setminus C_2$, $B_{i+1} := (B_i \cup \{\beta_i\})^\star \setminus B_i$ for some $\beta_i \in B \setminus (B_1 \cup \ldots \cup B_i)$ until $B \setminus (B_1 \cup \ldots \cup B_i) = \emptyset$. Thus the construction of $\pi^\star(B)$ requires $(3|B|+1) \cdot P_2(n)$ steps altogether.

(iii) Case (c) of lemma 4.3 applies. Then the corresponding unique strictly increasing sequence $D_1 \subset \ldots \subset D_k$ which determines $\pi^\star(B)$ is obtained as follows: $D_1 := C_1 \setminus C_2$, and D_{i+1} is the last non-empty set in the strictly decreasing sequence $E_1 \supset E_2 \supset \ldots$ defined by

$$E_1 := B$$

$$E_{j+1} := \begin{cases} (D_i \cup \{\beta\})^\star & \text{if there is } \beta \in E_j \setminus D_i \text{ such that } (D_i \cup \{\beta\})^\star \neq E_j \\ \emptyset & \text{otherwise.} \end{cases}$$

(Note that the last non-empty set in this sequence is the unique smallest S-autonomous set properly containing D_i.) Altogether, the construction of $\pi^\star(B)$ then requires $|B|^2 \cdot P_2(n)$ time.

Putting these steps together we obtain an algorithm P_3 for task 3 with $P_3(n) = O(n^3 \cdot P_2(n))$.

(c) Let algorithm P_3 for task 3 be given. Then P_3 solves task 1 trivially. To solve task 2, find in $\mathcal{B}(S)$ the lowest node C containing the given set B as a subset, i.e. covering all leaves $\{\alpha\}$ with $\alpha \in B$. This takes $d \cdot |B| \leqslant n^2$ time, where d is the depth of $\mathcal{B}(S)$. If C is unlabeled, then $C = B^\star$. If C is labeled with D, then $B^\star$ is the union of all immediate successors of C containing some $\alpha \in B$. If C is labeled with L, let $B_1 \leqslant \ldots \leqslant B_m$ be the ordered immediate successor of C and let (w.r.t. that order) B_i and B_{i+j} be the first and last sets containing some $\alpha \in B$. Then $B^\star = B_i \cup B_{i+1} \cup \ldots \cup B_{i+j}$.

Altogether, this requires $O(n^2) + P_3(n)$ time for the construction of $B^\star$. □

This shows that for an efficient determination of $\mathcal{B}(S)$, it suffices to find efficient methods for constructing the autonomous closure of a given set B. This is actually what will be done for relations and clutters in the following sections.

5. Decomposition of relations

In view of the characterization of degenerate and linear relations in proposition 3.4, we introduce in the composition tree of relations the distinction of label D into D_0 and D_1, depending on whether the associated quotient structure is (up to reflexive tuples) empty or complete. Furthermore, nodes with only two successors are also labeled by D_0, D_1, or L if the associated quotient has the corresponding property of proposition 3.4. An example is given in fig. 3, where the binary relation R is represented by a digraph.Note that for partial orders (where label D_1 can not occur), labels

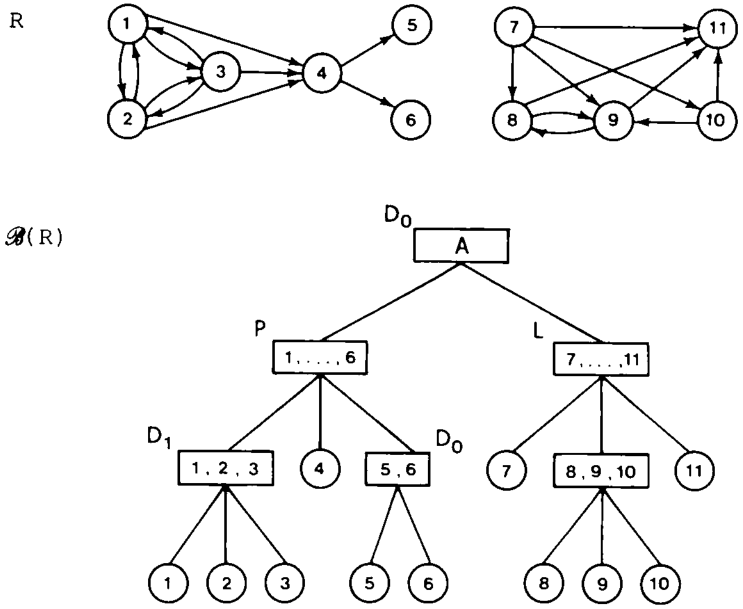


Fig. 3. A relation and its composition tree.

D_0 and L correspond to parallel and series composition of the associated successor nodes. The same holds for labels D_0 and D_1 in undirected graphs (where L can not occur). Many known results for series–parallel partial orders and graphs are covered here via this connection. For more details on this and on the connection between composition trees of partial orders and their comparability graphs, cf. [7,29].

There already exist fast decomposition algorithms for binary relations, cf. [19] for graphs ($O(n^3)$), [7] for graphs and partial orders ($O(n^3)$), and [12] for arbitrary binary relations (also $O(n^3)$ but with a different composition tree than the one described here). This alternative approach was originally developed for the split decomposition, for which this tree can be constructed in $O(n^4)$ time [12]. For still other approaches, cf. the summary in [29].

There is, however, nothing known for arbitrary k-ary relations. We shall show in the following that also in this very general case, the composition tree can still be constructed in polynomial time. One of the reasons for this is that, for fixed k, each k-ary relation has a representation which is polynomial in $|A|$, something which is no longer true, for example, for set systems (cf. sect. 6).

THEOREM 5.1

Let R be a k-ary relation on A and let $B \subseteq A$. Then $B^\star$ is obtained as follows:

(1) Put $C := B$.

(2) If there are $\alpha_1, \ldots, \alpha_k \in R$, $\alpha_i \in C$, $\alpha_j \notin C$ for some $i, j \in \{1, \ldots, k\}$, and $\beta \in C$ with $(\alpha_1, \ldots, \alpha_{i-1}, \beta, \alpha_{i+1}, \ldots, \alpha_k) \notin R$, then replace C by $C \cup \{\alpha_1, \ldots, \alpha_k\}$ and apply (2) again.

(3) Otherwise, $B^\star = C$.

Proof

Let $B = C_0 \subset C_1 \subset \ldots \subset C$ be a sequence of sets constructed in the algorithm. Assume that $B^\star \neq C$. Since C is obviously R-autonomous, we have $B \subseteq B^\star \subset C$. Let ℓ be the unique index such that $C_\ell \subseteq B^\star \subset C_{\ell+1}$, where $C_{\ell+1} = C$ is possible. In step (2) the algorithm finds a tuple $(\alpha_1, \ldots, \alpha_k) \in R$, with $\alpha_i \in C_\ell$, $\alpha_j \notin C_\ell$ for some $i, j \in \{1, \ldots, k\}$, and an element $\beta \in C_\ell$ such that $(\alpha_1, \ldots, \alpha_{i-1}, \beta, \alpha_{i+1}, \ldots, \alpha_k) \notin R$. Thus each set D with $C_\ell \subseteq D$ and $\alpha_j \notin D$ is not R-autonomous, which yields $C_\ell \cup \{\alpha_j\} \subset C_\ell^\star$ for each $j \in \{1, \ldots, k\}$ with $\alpha_j \notin C_\ell$. So $C_\ell \cup \{\alpha_1, \ldots, \alpha_k\} = C_{\ell+1} \subseteq C^\star$. Then $C_\ell \subseteq B^\star \subset C_{\ell+1}$ implies $C_\ell^\star \subseteq B^\star \subset C_{\ell+1}$, a contradiction. □

Obviously, this algorithm has complexity $O(n^2 m^2) \leqslant O(n^{2k+2})$, where $n = |A|$ and $m = |R|$. This, together with theorem 4.1, shows that the composition tree $\mathfrak{B}(R)$ of an arbitrary relation R on A can be constructed in $O(n^5 m^2)$ time. Of course, more efficient algorithms may be possible, as in the case for graphs or partial orders, cf. the bibliographical notes above.

6. Decomposition of set systems

For set systems, we shall first concentrate on *normal clutters* (i.e. systems of pairwise incomparable sets which cover the ground set), since these play the most important role in the applications of the substitution decomposition to combinatorial optimization, cf. [29].

Similarly to relations, we introduce the distinction of label D into D_0 and D_1, depending on whether the associated quotient structure is of the form $\{\{\alpha\} \mid \alpha \in A\}$ or $\{A\}$, respectively. (Note that these are the only degenerate normal clutters on A in view of proposition 3.4.) Also, nodes with only two successors are labeled in this way. An example is given in fig. 4, where the clutter $\mathcal{C}$ is represented by its incidence matrix.

	1	2	3	4	5	6	7	8	9	10
T_1	1	1	0	1	0	0	0	0	0	0
T_2	0	0	1	1	0	0	0	0	0	0
T_3	0	0	0	1	1	0	0	0	0	0
T_4	0	0	0	0	1	1	0	1	1	0
T_5	0	0	0	0	1	0	1	1	1	0
T_6	0	0	0	0	1	1	0	1	0	1
T_7	0	0	0	0	1	0	1	1	0	1
T_8	0	0	0	0	1	0	0	0	1	1

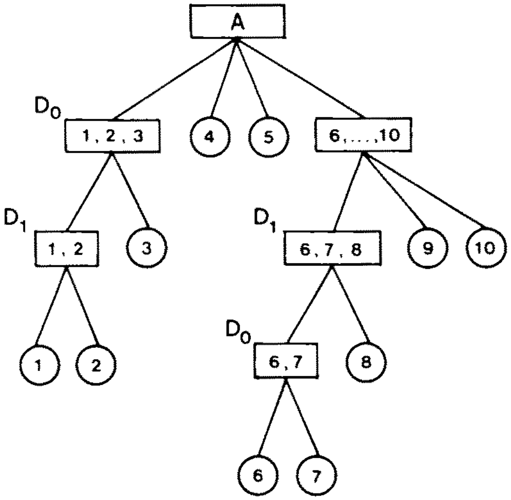


Fig. 4. A clutter and its composition tree.

Note that labels D_0 and D_1 correspond to the operations 'sum' and 'product' on the associated successor nodes in the sense of [4]:

The *sum* $\mathcal{C} = \mathcal{C}_1 + \ldots + \mathcal{C}_m$ of the clutters $\mathcal{C}_1, \ldots, \mathcal{C}_m$ on the (pairwise disjoint) sets $A_1, \ldots, A_m$ has the ground set

$$A = \bigsqcup_{i=1}^{m} A_i$$

and contains the sets

$$T \in \bigsqcup_{i=1}^{m} \mathcal{C}_i\,,$$

whereas the product $\mathcal{C} = \mathcal{C}_1 \times \ldots \times \mathcal{C}_m$ of $\mathcal{C}_1, \ldots, \mathcal{C}_m$ also has the ground set

$$A = \bigsqcup_{i=1}^{m} A_i$$

but contains all sets of the form $T = T_1 \cup \ldots \cup T_m$ for each choice of sets $T_i \in \mathcal{C}_i$, $i = 1, \ldots, m$.

These operations are dual to each other under the blocking operation, i.e.

$$b[\mathcal{C}_1 + \ldots + \mathcal{C}_m] = b[\mathcal{C}_1] \times \ldots \times b[\mathcal{C}_m] \qquad \text{and}$$

$$b[\mathcal{C}_1 \times \ldots \times \mathcal{C}_m] = b[\mathcal{C}_1] + \ldots + b[\mathcal{C}_m],$$

cf. [4].

With regard to the computational complexity of decomposing clutters, the situation is different from that for relations, since the characterization of autonomy in proposition 2.1 does not (as it does for relations) provide evident criteria for how to construct the autonomous closure. A further difference is that the representation of a clutter $\mathcal{C}$ on A as a set system may be *exponential* in $|A|$, i.e. $\mathcal{C}$ may consist of exponentially (in $|A|$) many sets.

There are of course a few exceptions to which methods developed for relations can still be applied. For instance, if $\mathcal{C}$ is conformal, then $\mathcal{B}(\mathcal{C}) = \mathcal{B}(G(\mathcal{C}))$ because of the links presented in sect. 2. Also the observation $\mathcal{A}(\mathcal{C}) \subseteq \mathcal{A}(G(\mathcal{C}))$ may be helpful (e.g. $\mathcal{C}$ is indecomposable if $G(\mathcal{C})$ is), as may the invariance principle between $\mathcal{C}$ and its blocker $b[\mathcal{C}]$, which states that $\mathcal{B}(\mathcal{C}) = \mathcal{B}(b[\mathcal{C}])$ with labels D_0 and D_1 interchanged, cf. [4,29] (e.g. if $\mathcal{C} = b(\mathcal{C}(G))$ for some graph G).

For arbitrary clutter, however, different methods must be applied. One such method has been developed in [4]. It is based on the construction of the blocker of certain subclutters of the given clutter. Thus it is polynomially bounded in $|A|$ iff the blocker of a given clutter $\mathcal{C}$ can be constructed in polynomial time. But since finding a minimal blocking set is an NP-complete problem even in the case $|T| \leqslant 2$ for all $T \in \mathcal{C}$ [17], this algorithm will probably be exponential in general.

In the following we will follow a different approach and show that task 2, i.e. finding the $\mathcal{C}$-autonomous closure of a given set, can be essentially reduced to the construction of separators (in the matroidal sense) in certain derived clutters. This will lead to polynomial decomposition algorithms for certain classes of clutters.

To this end, observe first that if $\mathcal{C}$ is a *sum* of clutters, the associated finest partition π^* of $\mathcal{L}(\mathcal{C})$ is just the partition of the graph $G(\mathcal{C})$ into its connected components, which can be found in polynomial time by standard graph methods. So we will restrict ourselves to *non-sum* clutters in the following.

In accordance with properties of separators for the system of bases of a matroid [41,42], we define the *separator* of a clutter $\mathcal{C}$ on A to be a set $B \subseteq A$ such that

$$\mathrm{Ex}(T_1, B, T_2) := (T_1 \backslash B) \cup (T_2 \cap B) \in \mathcal{C}$$

for all $T_1, T_2 \in \mathcal{C}$.

There is a strong connection between autonomous sets and separators:

PROPOSITION 6.1

Let $\mathcal{C}$ be a normal clutter on A. A set $B \subseteq A$ is $\mathcal{C}$-autonomous iff B is a separator of the normal clutter $\mathcal{C}(B) := \{T \in \mathcal{C} \mid T \cap B \neq \emptyset\}$ with ground set

$$A(B) := \bigcup_{T \in \mathcal{C}(B)} T.$$

Proof

Proposition 2.1 (b) and the defintion of a separator. □

To exploit this connection, we first note the following properties of separators (cf. [41,42] for separators in matroids):

PROPOSITION 6.2

Let $\mathcal{C}$ be a normal clutter on A.

(a) A and $\emptyset$ are always separators (the *trivial* separators) of $\mathcal{C}$.

(b) B is a separator of $\mathcal{C}$ iff B is $\mathcal{C}$-autonomous and each $T \in \mathcal{C}$ meets both B and $A \backslash B$.

(c) $\mathcal{C}$ has non-trivial separators iff $\mathcal{C}$ is a product of clutters. Then the non-trivial separators of $\mathcal{C}$ are exactly the classes of the finest partition $\pi^\star \in \mathfrak{L}(\mathcal{C})$ in the sense of theorem 3.5 and all unions of these classes.

(d) Union, intersection and complements of separators are again separators. For each $\alpha \in A$ there exists a minimal α containing separator $B(\alpha)$. Then $\{B(\alpha) \mid \alpha \in A\}$ is the partition $\pi^\star$ in (c).

Proof

(a) is trivial. To show the non-obvious direction in (b), let B be a non-trivial separator of $\mathcal{C}$. B is $\mathcal{C}$-autonomous by proposition 2.1. Furthermore, $A \backslash B$ is a separator of $\mathcal{C}$ for symmetry reasons. Assume that there is $T_1 \in \mathcal{C}$ such that w.l.o.g. $T_1 \cap B = \emptyset$. Choose $T_2 \in \mathcal{C}$ with $T_2 \cap B \neq \emptyset$. Then $T_0 := \mathrm{Ex}(T_1, B, T_2) \in \mathcal{C}$ and $T_1 \subset T_0$, a contradiction.

If B is a non-trivial separator of $\mathcal{C}$, then $\mathcal{C} = (\mathcal{C} \mid B) \times (\mathcal{C} \mid A \setminus B)$ because of (b) and $\pi = \{B, A \setminus B\} \in \mathcal{L}(\mathcal{C})$. Let $\pi^\star$ be the finest partition in $\mathcal{L}(\mathcal{C})$. Since $\pi^\star$ is a refinement of π, say $\pi = \{B_1, \ldots, B_m\}$ with $B = B_1 \cup \ldots \cup B_k$, we obtain from the properties of $\mathcal{B}(\mathcal{C})$ that $\mathcal{C} \mid B$ (and similarly $\mathcal{C} \mid A \setminus B$) is either the product or the sum of $\mathcal{C} \mid B_1, \ldots, \mathcal{C} \mid B_k$. But since all unions of classes of $\pi^\star$ are $\mathcal{C}$-autonomous, it must be the product, because otherwise $B_i \cup B_j \notin \mathcal{A}(\mathcal{C})$ for $1 \leqslant i \leqslant k < j \leqslant m$. Thus each $T \in \mathcal{C}$ meets all classes $B_i \in \pi^\star$ and $\pi^\star$ is by construction the finest congruence partition with this property. This yields (c) and (d). □

Let $\mathcal{C}$ be a clutter on A. Because of proposition 6.1 (d) there exists for each set $B \subseteq A$ a smallest B containing separator C of $\mathcal{C}$. This set C is called the *separator hull* of B and is denoted by $B^\vartriangle$.

THEOREM 6.3

Let $\mathcal{C}$ be a normal non-sum clutter on A and $B \subseteq A$. Then the $\mathcal{C}$-autonomous closure $B^\star$ of B is obtained as follows:

(1) Put $C := B$.

(2) Determine the separator hull $C^\vartriangle$ of C in $\mathcal{C}(C)$. If $C^\vartriangle \neq C$, replace C by $C^\vartriangle$ and apply (2) again.

(3) If $C^\vartriangle = C$, then $B^\star = C$.

Proof

Let $B = C_0 \subset C_1 \subset C_2 \subset \ldots \subset C$ the sequence of sets constructed by the algorithm. Because of proposition 6.1, C is $\mathcal{C}$-autonomous and hence $B^\star \subseteq C$. Assume that $B^\star \neq C$ and let k be the (unique) index such that $C_k \subseteq B^\star \subset C_{k+1}$ (where $C_{k+1} = C$ is possible).

We shall show that $B^\star$ is a separator of $\mathcal{C}(C_k)$. By construction, $B^\star \subset C_{k+1} \subseteq A(C_k)$, the ground set of $\mathcal{C}(C_k)$. Let $T_1, T_2 \in \mathcal{C}(C_k)$. Then $T_i \cap C_k \neq \emptyset$ and thus also $T_i \cap B^\star \neq \emptyset$, $i = 1, 2$. Since $B^\star \in \mathcal{A}(\mathcal{C})$ and $\mathcal{C}(C_k) \subseteq \mathcal{C}$, $T_0 := \operatorname{Ex}(T_1, B^\star, T_2) \in \mathcal{C}$. Obviously, $T_0 \cap B^\star = T_2 \cap B^\star \supseteq T_2 \cap C_k \neq \emptyset$. Hence $T_0 \in \mathcal{C}(C_k)$ and so $B^\star$ is $\mathcal{C}(C_k)$-autonomous. Since $B^\star \neq A(C_k)$, it remains to be shown, in view of proposition 6.2 (b), that each $T \in \mathcal{C}(C_k)$ meets both $B^\star$ and $A(C_k) \setminus B^\star$. This is obvious for $B^\star$ since $C_k \subseteq B^\star$. Now if $T \subseteq B^\star$ for some $T \in \mathcal{C}(C_k)$, then choose $T_1 \in \mathcal{C}(C_k)$ which meets $A(C_k) \setminus B^\star$. By definition of $\mathcal{C}(C_k)$, T_1 also meets $C_k \subseteq B$, and thus $T_0 := \operatorname{Ex}(T_1, B^\star, T) \in \mathcal{C}(C_k)$ because of the $\mathcal{C}(C_k)$-autonomy of $B^\star$. But then $T \subset T_0$, a contradiction.

So $B^\star$ is a separator of $\mathcal{C}(C_k)$ with $C_k \subseteq B \subset C_{k+1}$. This contradicts C_{k+1} being the separator hull of C_k in $\mathcal{C}(C_k)$. □

Since the clutters $\mathcal{C}(C)$ can be simply constructed, the determination of $B^\star$ is reduced to the determination of the separator hull.

In order to construct separator hulls, we make use of the relationship between separators and circuits in matroids, cf. [41,42].

Let $\mathcal{C}$ be a clutter on A and let $\mathfrak{J}$ denote the corresponding *independence system*, i.e. the system $\mathfrak{J} = \{I \in 2^A \mid I \subseteq T \text{ for some } T \in \mathcal{C}\}$. Analogously to matroids, we call the sets of $\mathfrak{J}$ the *independent* sets of $\mathcal{C}$, the sets in $2^A \setminus \mathfrak{J}$ the *dependent* sets of $\mathcal{C}$, and the $\subseteq$-minimal dependent sets the *circuits* of $\mathcal{C}$.

One then obtains the following generalization of the common matroid properties:

PROPOSITION 6.4

Let $\mathcal{C}$ be a clutter on A.

(a) $B \subseteq A$ is a separator of $\mathcal{C}$ iff $K \subseteq B$ or $K \subseteq A \setminus B$ for each circuit K of $\mathcal{C}$.

(b) Write $\alpha \sim \beta$ if there is a sequence $\alpha = \alpha_1, \ldots, \alpha_k = \beta$ such that for each $i = 1, \ldots, k-1$, α_i and α_{i+1} are contained in a common circuit of $\mathcal{C}$. Then $\sim$ is an equivalence relation on A and the equivalence classes of $\sim$ are the minimal non-empty separators of $\mathcal{C}$.

(c) For the separator hull $B^\vartriangle$ of B, we have $B^\vartriangle = \{\alpha \in A \mid \alpha \sim \beta \text{ for some } \beta \in B\}$.

Proof

(a) Let B be a separator of $\mathcal{C}$ and assume that there exists a circuit K of $\mathcal{C}$ such that K meets both B and $A \setminus B$. Then $K \setminus B$ and $K \cap (A \setminus B)$ are independent and thus contained in two clutter members, say T_1 and T_2. Then also $T := \mathrm{Ex}(T_1, B, T_2) \in \mathcal{C}$ and $K \subseteq T$, a contradiction.

In the opposite direction, let $K \subseteq B$ or $K \subseteq A \setminus B$ for all circuits K of $\mathcal{C}$. Then for any independent sets $I_1 \subseteq B$ and $I_2 \subseteq A \setminus B$, $I_1 \cup I_2$ is also independent. Now let $T_1, T_2 \in \mathcal{C}$. Then $T_0 := \mathrm{Ex}(T_1, B, T_2)$ is independent because $T_1 \setminus B = T_1 \cap (A \setminus B)$ and $T_2 \cap B$ are. If $T_0 \notin \mathcal{C}$, there exists $T \in \mathcal{C}$ with $T_0 \subset T$. Then $T_1 \setminus B \subset T \setminus B$ or $T_2 \cap B \subset T \cap B$. In the first case, $(T \setminus B) \cup (T_1 \cap B)$ is independent and properly contains T_1, in the second case, one obtains the same contradiction with $(T \cap B) \cup (T_2 \setminus B)$ and T_2. Therefore, $T_0 \in \mathcal{C}$. This shows that B is a separator.

(b) and (c) are immediate consequences of (a). □

Note that for matroids, $\alpha \sim \beta$ iff $\{\alpha, \beta\} \subseteq K$ for some circuit K of $\mathcal{C}$. This is no longer true for arbitrary clutters $\mathcal{C}$.

Theorem 6.3 and proposition 6.4 reduce the determination of the autonomous closure to the (iterated) determination of circuits or the corresponding equivalence relation $\sim$. For arbitrary clutters, this leads to the following complexity result.

THEOREM 6.5

Let $\mathcal{C}$ be a normal clutter on A. Let $n := |A|$ and $m := |\mathcal{C}|$. Then the autonomous closure B^* of a set $B \subseteq A$ can be constructed in $O(n^4 m^3)$ time and $O(n \cdot m)$ space.

Proof

We shall make use of the following characterization of $\sim$, which is obtained straightforwardly.

$$(\star) \qquad \alpha_1 \sim \alpha_2 \iff \begin{cases} \exists \;\; T_1, T_2 \in \mathcal{C} \text{ such that } \alpha_1 \in T_1 \setminus T_2, \;\; \alpha_2 \in T_2 \setminus T_1 \text{ and} \\ (T_1 \cap T_2) \cup \{\alpha_1, \alpha_2\} \quad \text{is independent.} \end{cases}$$

Now any set $B \subseteq A$ can be tested for dependence in $O(n \cdot m)$ time by checking whether $B \subseteq T$ for some $T \in \mathcal{C}$. So using $(\star)$, the validity of $\alpha_1 \sim \alpha_2$ can be tested in $O(n \cdot m^3)$ time. Thus the complete equivalence relation $\sim$ can be determined in $O(n^3 m^3)$ time. Since at most n separator closures must be computed, the whole procedure requires $O(n^4 m^3)$ time and $O(n \cdot m)$ space. □

Of special interest are of course classes of clutters for which the procedure described above leads to decomposition algorithms which are polynomial in $|A|$. This is of course the case for clutters with a given *polynomial representation* of either $\mathcal{C}$ itself (i.e. $|\mathcal{C}|$ is polynomial in $|A|$) or the system $\mathcal{K}$ of all circuits of $\mathcal{C}$ (i.e. $\mathcal{C}$ is given by $\mathcal{K}$ and $|\mathcal{K}|$ is polynomial in $|A|$).

Special examples for the first case are, for each $k \in \mathbb{N}$, the classes of *k-bounded clutters*, i.e. clutters $\mathcal{C}$ with $|T| \leqslant k$ for all $T \in \mathcal{C}$. They correspond to k-ary relations in the sense that for k-ary relations, the size of the tuples $(\alpha_1, \ldots, \alpha_k) \in R$ is also bounded.

An example for the second case is given by the class of conformal clutters because of the following characterization.

THEOREM 6.6

A normal clutter $\mathcal{C}$ is conformal iff $|K| = 2$ for all circuits K of $\mathcal{C}$.

In that case, the circuits of $\mathcal{C}$ are given by the unconnected pairs of vertices in the graph $G(\mathcal{C})$.

Proof

If $\mathcal{C}$ is conformal, then the independent sets are exactly the cliques of $G(\mathcal{C})$. Thus, each independent set contains two unconnected vertices.

In the opposite direction, let G be the graph with vertex set A in which two vertices α, β are connected if $\{\alpha,\beta\}$ is not a circuit of $\mathcal{C}$. Then the independent sets of $\mathcal{C}$ are obviously exactly the cliques of G. Thus, $\mathcal{C} = \mathcal{C}(G)$, i.e. $\mathcal{C}$ is conformal. □

In fact, in view of theorem 6.6, the above methods reduce to the algorithm in theorem 5.1 applied to the graph $G(\mathcal{C})$. In this sense, the algorithmic methods developed in this section form a 'natural' although not evident generalization of the method developed for relations in sect. 5.

For arbitrary clutters, however, it is very unlikely that a polynomial (in $|A|$) decomposition algorithm exists. A strong argument for this is the following result, which shows that even for rather strong orable algorithms in the sense of [17,20], the decision whether a clutter is decomposable or not requires an exponential number of calls on the oracle.

We will consider the following, very informative oracle for a clutter $\mathcal{C}$ on A:

Input: A subset B of A.

Output:
$<$ if $B \subset T$ for some $T \in \mathcal{C}$
$=$ if $B \in \mathcal{C}$
$>$ if $T \subset B$ for some $T \in \mathcal{C}$
c if B is a,circuit of $\mathcal{C}$
d if B is dependent, but $>$ and c do not apply.

THEOREM 6.7

Let P be an oracle algorithm with the above oracle, which decides for arbitrary clutters $\mathcal{C}$ whether $\mathcal{C}$ is decomposable or not. Then P requires at least $2^{n/2}$ calls on the oracle for clutters $\mathcal{C}$ on A with $|A| = n$.

Proof

Let $n \geqslant 8$ be a multiple of 4, A be a set of size n, and A_1, A_2 be disjoint subsets of A of size $n/2$. Let $\mathcal{C}_1$ be the clutter $\mathcal{C}_1 := \{T_1 \cup T_2 \,|\, T_i \subseteq A_i, \ |T_i| = n/4, \ i = 1, 2\}$ on A. Obviously, A_1 and A_2 are the only non-trivial $\mathcal{C}_1$-autonomous sets (they are even separators of $\mathcal{C}_1$). Therefore, $\mathcal{C}_1$ is decomposable. If algorithm P obtains this result with less than $m := |\mathcal{C}_1| = \binom{n/2}{n/4}^2 \geqslant 2^{n/2}$ calls on the oracle, then there is a set $T_0 \in \mathcal{C}_1$ for which the oracle is not called.

Then $\mathcal{C}_2 := \mathcal{C}_1 \backslash \{T_0\}$ is again a normal clutter on A which has the following properties:

(i) $\mathcal{C}_2$ is indecomposable.

(ii) For each set $B \subseteq A$ with $B \neq T_0$, the answers $<$, $=$, $>$, c, d, of the oracle are the same for $\mathcal{C}_1$ and $\mathcal{C}_2$.

To see (i), observe that since all $T \in \mathcal{C}_2$ have the same size, all intersections of sets $T \in \mathcal{C}_2$ with a fixed $\mathcal{C}_2$-autonomous set must have the same size. By construction, this can only hold for A_1 and A_2. But since $T_0 \notin \mathcal{C}_2$, A_1 and A_2 are not $\mathcal{C}_2$-autonomous.

To see (ii), observe that the circuits of $\mathcal{C}_2$ are exactly the circuits of $\mathcal{C}_1$ and the set T_0.

As a consequence of (ii), the oracle algorithm P arrives after at most $m - 1$ calls on the oracle for $\mathcal{C}_2$ at the same decision as for $\mathcal{C}_1$, which is in contradiction to (i). □

This argument shows that even a slight change in the structure can turn a decomposable into an indecomposable set system and vice versa. Thus, decomposition algorithms must in general exploit the total information on the set system, also if it is exponential in $|A|$.

It could, of course, still be the case that a given exponential set system has another, polynomial representation which lends itself to faster decomposition methods (e.g. conformal clutters and graphs). However, if one does not know whether such a representation exists, finding it may again require an exponential effort. For conformal clutters, this is demonstrated by the following theorem.

THEOREM 6.8

Let P be an oracle algorithm with the above oracle, which decides for arbitrary clutters $\mathcal{C}$ whether $\mathcal{C}$ is conformal or not. Then P requires at least $n^{\sqrt{n}/2}$ calls on the oracle for clutters $\mathcal{C}$ on A with $|A| = n$.

Proof

Let $n = k^2$, with $k \geqslant 3$, A be a set of size n, and $A_1, \ldots, A_k$ be a partition of A in k sets of size k. Let G be the graph with vertex set A and maximal cliques $A_1, \ldots, A_k$. Let $\mathcal{C}_1 = \mathcal{C}(\bar{G})$, where $\bar{G}$ is the complementary graph of G. $\mathcal{C}_1$ is conformal and P decides so. If P obtains this decision with less than $m := |\mathcal{C}_1| = k^k = n^{\sqrt{n}/2}$ calls on the oracle, then there exists a set $T_0 \in \mathcal{C}_1$ for which the oracle is not called. Then $\mathcal{C}_2 := \mathcal{C}_1 \setminus \{T_0\}$ is a (normal) clutter on A with:

(i) $\mathcal{C}_2$ is not conformal.

(ii) For each set $B \subseteq A$ with $B \neq T_0$, the answers $<$, $=$, $>$, c, d, of the oracle are the same for $\mathcal{C}_1$ and $\mathcal{C}_2$.

This gives a contradiction similar to the proof of theorem 6.6. □

Note that if is known that a clutter $\mathcal{C}$ is conformal, then an oracle algorithm can obviously construct $G(\mathcal{C})$ in $n(n-1)/2$ calls on the oracle and afterwards apply a graph decomposition algorithm. However, to obtain this knowledge, an exponential number of calls on the oracle may be necessary.

7. Decomposition of Boolean functions

As for relations and clutters, we introduce the distinction of label D into D_0, D_1 and D_2, depending on whether the associated quotient structure is a Boolean sum, product, or a ring sum, respectively (cf. proposition 3.4). Also nodes with only two successors are labeled accordingly.

As an example, consider the Boolean function $F(x_1, \dots, x_n)$ given by its normal disjunctive form:

$$\begin{aligned} F(x_1,\dots,x_{10}) = {} & x_1\bar{x}_2\bar{x}_3x_4 + \bar{x}_1x_2\bar{x}_3x_4 + \bar{x}_1\bar{x}_2x_3x_4 + x_1x_2x_3x_4 + x_5 \\ & + x_6x_7\bar{x}_8x_9 + \bar{x}_6x_8x_9 + \bar{x}_7x_8x_9 \\ & + x_6x_7\bar{x}_8x_{10} + \bar{x}_6x_8x_{10} + \bar{x}_7x_8x_{10} + x_9x_{10}\,. \end{aligned}$$

The composition tree of F is given in fig. 5. Based on this tree, one obtains the following, equivalent representation:

$$F(x_1,\dots,x_{10}) = (x_1 \oplus x_2 \oplus x_3)x_4 + x_5 + (x_6x_7 \oplus x_8)(x_9 + x_{10}) + x_9x_{10}$$

(cf. proposition 3.4), where the node $B = \{6, \dots, 10\}$ corresponds to the prime Boolean threshold function $G(y_1,y_2,y_3) = y_1y_2 + y_1y_3 + y_2y_3$.

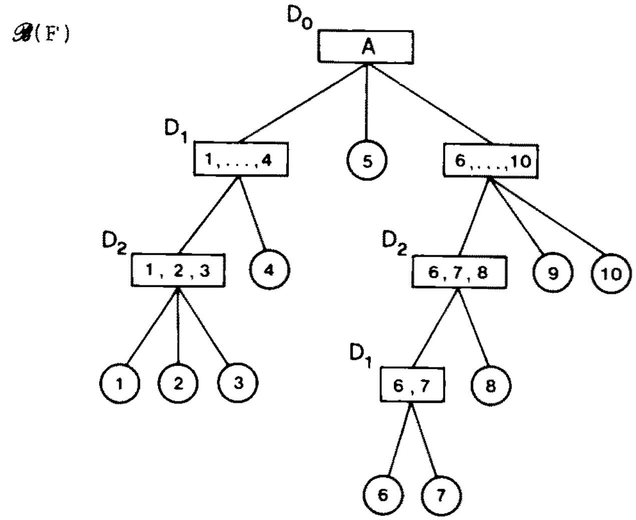


Fig. 5. The composition tree of the Boolean function F.

For the decomposition of Boolean functions, too, many algorithms have been developed owing to the significance of decomposition for switching design, cf. for instance [15,16,36,37,40]. These algorithms are either based on the evaluation of Ashenhurst's decomposition charts or use a differential calculus for determining F-autonomous sets. In all cases they have an exponential (in the number of variables) worst-case complexity, although in some cases a good average performance was observed in empirical studies.

For monotonic Boolean functions, the results on clutter decomposition in sect. 5 can be applied because of the link between clutters and monotonic Boolean functions described in sect. 2. This leads to more efficient methods than those cited above, since the number $m = |\mathcal{C}|$ of prime implicants in theorem 6.5 will usually be much smaller than the size 2^n of the complete table of a Boolean function in n variables which, for instance, is used in the evaluation of the Ashenhurst decomposition charts in each step [1].

On the other hand, the negative result on the complexity of oracle decomposition algorithms for clutters carries over to monotonic Boolean functions, if the oracle answers are interpreted accordingly. Since the decomposition possibilities of monotonic Boolean functions within the class of *monotonic* Boolean functions are the same as within the class of *arbitrary* Boolean functions (cf. sect. 2), it follows that for arbitrary Boolean functions, too, universally polynomial decomposition algorithms do not exist.

References

[1] R.L. Ashenhurst, The decomposition of switching functions, in: *Proc. Int. Symposium on the Theory of Switching,* Part I (Harvard University Press, Cambridge, 1959).

[2] R.E. Barlow and F. Proschan, *Statistical Theory of Reliability and Life Testing* (Holt, Rinehart and Winston, New York, 1975).

[3] L.J. Billera, Clutter decomposition and monotonic Boolean functions, Ann. of the New York Academy of Sciences 175(1970)41.

[4] L.J. Billera, On the composition and decomposition of clutters, J. Comb. Th. B 11(1971) 234.

[5] L.J. Billera and R.E. Bixby, Decomposition theory for a class of combinatorial optimization problems, in: *Optimization Methods for Resource Allocation,* Proc. Nato Conf. Elsinore (1971) (English University Press, London, 1974) p. 427.

[6] Z.W. Birnbaum and J.D. Esary, Modules of coherent binary systems, SIAM J. Applied Math. 13(1965)444.

[7] H. Buer and R.H. Möhring, A fast algorithm for the decomposition of graphs and posets, Math. Oper. Res. (1984) 170.

[8] R.W. Butterworth, A set theoretic treatment of coherent systems, SIAM J. Applied Math. 22(1972)590.

[9] M. Chein, M. Habib and M.C. Maurer, Partitive hypergraphs, Discrete Math. 37(1981)35.

[10] V. Chvatal, On certain polytopes associated with graphs, J. Comb. Th. (B) 18(1975)138.

[11] D.D. Cowan, L.O. James and R.G. Stanton, Graph decomposition for undirected graphs, in: *3rd South-Eastern Conf. Combinatorics, Graph Theory, and Computing,* ed. F. Hoffman and R.B. Levow (Utilitas Math., Winnipeg, 1972) p. 281.

[12] W.H. Cunningham, Decomposition of directed graphs, SIAM J. Algebraic and Discrete Methods 3(1982)214.

[13] W.H. Cunningham and J. Edmonds, A combinatorial decomposition theory, Can. J. Math. 32(1980)734.

[14] H.A. Curtis, *A New Approach to the Design of Switching Circuits* (Van Nostrand, Princeton, 1962).

[15] M. Davio, J.P. Deschamps and A. Thayse, *Discrete and Switching Functions* (McGraw – Hill, New York, 1978).

[16] J.P. Deschamps, Binary simple decomposition of discrete functions, Digital Processes 1 (1975)123.

[17] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness* (Freeman, San Francisco, 1979).

[18] M. Golumbic, *Algorithmic Graph Theory and Perfect Graphs* (Academic Press, New York, 1980).

[19] M. Habib and M.C. Maurer, On the X-join decomposition for undirected graphs, J. Appl. Discr. Math. 3(1979)198.

[20] D. Hausmann and B. Korte, Lower bounds on the worst-case complexity of some oracle algorithms, Discrete Math. 24(1978)261.

[21] R.L. Hemminger, The group of an X-join of graphs, J. Comb. Th. 5(1968)408.

[22] T. Hiragushi, On the dimension of partially ordered sets, Sci. Rep., Kanazawa University 1(1951)77.

[23] R. Kaerkes and B. Leipholz, Generalized network functions in flow networks, Methods of Oper. Res. 27(1977)225.

[24] R. Kaerkes and F.J. Radermacher, Profiles, network functions and factorization, Methods of Oper. Res. 27(1977)66.

[25] E.L. Lawler, Sequencing jobs to minimize total weighted completion time subject to precedence constraints, Ann. Discrete Math. 2(1978)75.

[26] R.H. Möhring, Untersuchungen zur Homomorphietheorie von Relationalsystemen, Thesis, Tech. Univ. of Aachen (1975).

[27] R.H. Möhring, Dekomposition diskreter Strukturen mit Anwendungen in der kombinatorischen Optimierung, Schriften zur Informatik und Angewandten Mathematik No. 95, Tech. Univ. of Aachen (1984).

[28] R.H. Möhring and F.J. Radermacher, Profiles and homomorphisms, Methods of Oper. Res. 27(1977)88.

[29] R.H. Möhring and F.J. Radermacher, Substitution decomposition of discrete structures and connections to combinatorial optimization, Ann. Discrete Math. 19(1984)257.

[30] C.L. Monma and J.B. Sidney, Sequencing with series-parallel precedence constraints, Math. of Oper Res. 4(1979)215.

[31] J. Neggers, Counting finite posets, Acta Math. Acad. Scient. Hung., Tom. 31(1978)233.

[32] J.L. Pfaltz, Graph structures, J. ACM 19(1972)411.

[33] F.J. Radermacher and H.G. Spelde, Reduktion von Flussnetzplänen, Proc. in Oper. Res. 3 (1974)177.

[34] L.S. Shapley, Solutions of compound simple games, in: *Advances in Game Theory,* Ann. of Math. Study No. 52 (Princeton University Press, Princeton, 1964) p. 267.

[35] L.S. Shapley, On Committees, in: *New Methods of Thought and Procedure,* ed. F. Zwicky and A.G. Wilson (Springer-Verlag, Berlin – New York, 1967) p. 246.

[36] V.Y. Shen and A.C. McKellar, An algorithm for the disjunctive decomposition of switching functions, IEEE Trans. Computers C – 19(1970)239.

[37] V.Y. Shen, A.C. McKellar and P. Weiner, A fast algorithm for the disjunctive decomposition of switching functions, IEEE Trans. Computers C – 20(1971)304.

[38] A.W. Shogan, Modular decomposition and reliability computation in stochastic transportation networks having cutnodes, Networks 12(1982)255.
[39] K. Strassner, Zur Strukturtheorie endlicher nichtdeterministischer Automaten I. Zum Verband der 1-Kongruenzen von endlichen Relationalsystemen, Elektronische Informationsverarbeitung und Kybernetik 17(1981)113.
[40] A. Thayse, A fast algorithm for the proper decomposition of Boolean functions, Philips Res. Rep. 27(1972)140.
[41] W. Tutte, Lectures on matroids, J. Res. Nat. Bur. Standard 69B(1965)1.
[42] D.J.A. Welsh, *Matroid Theory* (Academic Press, London, 1976).

Note added in proof

In the meantime, the fastest decomposition algorithms for graphs require only $O(n^2)$ time, cf.

[43] J.H. Muller and J. Spinrad, On-line modular decomposition, Tech. Rep. GIT-ICS-84/11, Georgia Institute of Technology (1984).
[44] J. Spinrad, Two-dimensional partial orders, Thesis, Princeton University (1982).