Optimizing Inference in Bayesian Networks and Semiring Valuation Algebras

Michael Wachter[1], Rolf Haenni[2], and Marc Pouly[3]

[1] University of Bern, Switzerland
{wachter,haenni}@iam.unibe.ch
[2] Bern University of Applied Sciences, Switzerland
rolf.haenni@bfh.ch
[3] University of Fribourg, Switzerland
marc.pouly@unifr.ch

Abstract. Previous work on context-specific independence in Bayesian networks is driven by a common goal, namely to represent the conditional probability tables in a most compact way. In this paper, we argue from the view point of the knowledge compilation map and conclude that the language of Ordered Binary Decision Diagrams (OBDD) is the most suitable one for representing probability tables, in addition to the language of Algebraic Decision Diagrams (ADD). We thus suggest the replacement of the current practice of using tree-based or rule-based representations. This holds not only for inference in Bayesian networks, but is more generally applicable in the generic framework of semiring valuation algebras, which can be applied to solve a variety of inference and optimization problems in different domains.

1 Introduction

Bayesian networks (BN) are a very flexible and powerful tool in many areas, particularly in AI related problems and applications [1]. Its power stems from the efficient encoding of independence relations among variables. Originally, BNs were mainly designed to exploit so-called *conditional* (or *structural*) *independences*, which allows the (global) joint probability function to be replaced by several (local) *conditional probability tables* (CPT). The locality of the CPTs in turn is responsible for the success of BNs as an efficient computational tool for probabilistic inference [2].

The exploitation of another type of independence relations, so-called *contextual* or *context-specific independences* (CSI), has been proposed in [3].[4] CSI deals with local independence relations *within* (rather than *between*) the given CPTs. A *context* within a CPT is a partial parent configuration.

[4] The notion of context-specific independence first appeared in the influence diagram literature [4]. Note that some authors prefer to use *contextual strong independence* as an alternative name with the same acronym [5]. Other similar notions are *asymmetric independence* [6] and *probabilistic causal irrelevance* [7].

A. Gelbukh and A.F. Kuri Morales (Eds.): MICAI 2007, LNAI 4827, pp. 236-247, 2007.

Most approaches to exploit CSI suggest a tree-structured CPT representation, but different names such as *CPT-trees* [3], *probability trees* [8], or *multi-resolution binary trees* [9] are in use for essentially the same concept. All these techniques share a common goal, namely to merge CPT entries with the same value for a specific context. Note that such a simplified CPT may still include the same value more than once.

More advanced CPT representations allow a complete partitioning of the parent configurations, in which each value occurs exactly once. A simple idea to achieve this is to represent the partitions by logical rules [10], but a more efficient approach is the use of *Algebraic Decision Diagrams* (ADD) as suggested in [11]. Note that ADDs are a generalization of *Ordered Binary Decision Diagrams* (OBDD) [12]. Technically speaking, this method exceeds CSI insofar as it considers the entire local structure to simplify a given CPT, thus possibly spanning over various contexts.

In this paper, we start by looking at the exploitation of local CPT structures from the perspective of the *knowledge compilation map* [13, 14]. This map supports the identification of the most appropriate representation language according to the *queries* and *transformations* it is supposed to offer in polynomial time. At the end, the main conclusion from this view will be the following:

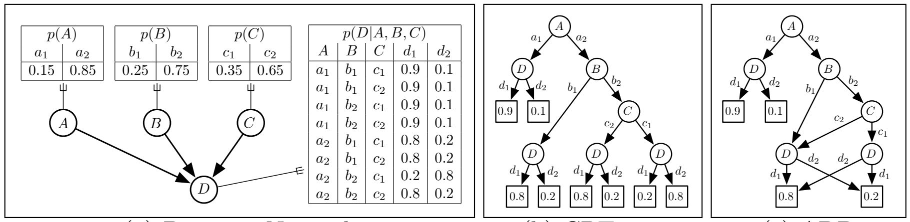
> *The languages of OBDDs (and possibliy DNFs) is the most appropriate representation language for local CPT structures in BNs, in addition to the language of ADDs.*

To obtain this result and to emphasize its generality, we will entirely shift our analysis from Bayesian networks into the generic framework of *semiring valuation algebras* [15, 16]. This is an abstract theory of inference in knowledge-based systems, which is based on two principal operations called *combination* and *variable elimination* (or *marginalization*). The generality of valuation algebras allows us to extend the applicability of the above results from BNs to a much broader range of formalisms and applications thereof. Moreover, it opens up new opportunities and possibilities for building generic approximations methods.

The structure of this paper is as follows. Section 2 provides a short summary of inference in BNs. Section 3 is devoted to semiring valuations and their connection to BNs. Section 4 discusses the optimization of the underlying representation. Section 5 concludes the paper.

2 Inference in Bayesian Networks

A *Bayesian network* (BN) is an efficient representation of a *joint probability mass function* over a set $\mathbf{X}$ of variables [1]. We assume throughout this paper that all variables $X \in \mathbf{X}$ are *binary*, i.e. their associated sets of possible values are $\Omega_X = \{x_1, x_2\}$. The network itself consists of a directed acyclic graph (DAG), which represents the direct influences among the variables, each of them attached to one node, and a set of conditional probability tables (CPT), which quantify



(a) Bayesian Network (b) CPT-tree (c) ADD

Fig. 1. Example of a simple Bayesian network with four variables, and two other representations of the CPT $p(D|A, B, C)$.

the strengths of these influences. The whole BN represents a joint probability mass function $p : \Omega_{\mathbf{X}} \to [0, 1]$ over its variables in a compact manner by

$$p(\mathbf{X}) = \prod_{X \in \mathbf{X}} p(X|parents(X)), \tag{1}$$

where $parents(X)$ denotes the parents of node X in the DAG. Figure 1(a) depicts a simple BN. It consists of four variables A, B, C, and D, with corresponding CPTs for $p(A)$, $p(B)$, $p(C)$, and $p(D|A, B, C)$.

Inference in Bayesian networks means to compute the conditional probability $P(H{=}h \,|\, E_1{=}e_1, \ldots, E_r{=}e_r)$, or simply

$$P(h|\mathbf{e}) = \frac{P(h, \mathbf{e})}{P(\mathbf{e})}, \tag{2}$$

of a hypothesis $h \in \Omega_H$ for some observed evidence $\mathbf{e} = (e_1, \ldots, e_r) \in \Omega_{\mathbf{E}}$. We will call the elements of $\mathbf{E} = \{E_1, \ldots, E_r\} \subseteq \mathbf{X}$ *evidence variables*. To see how to solve the inference problem, let $\mathbf{Y} = \{Y_1, \ldots, Y_s\} \subseteq \mathbf{X}$ be an arbitrary subset of variables, $\mathbf{y} = (y_1, \ldots, y_s) \in \Omega_{\mathbf{Y}}$ a configuration of values $y_i \in Y_i$, and $\mathbf{Z} = \mathbf{X} \backslash \mathbf{Y}$. Then it is sufficient to compute

$$P(\mathbf{y}) = \sum_{\mathbf{z} \in \Omega_{\mathbf{Z}}} p(\mathbf{yz}) \tag{3}$$

twice, once with $\mathbf{Y} = \{H\} \cup \mathbf{E}$ and $\mathbf{y} = (h, \mathbf{e})$ to get the nominator and once with $\mathbf{Y} = \mathbf{E}$ and $\mathbf{y} = \mathbf{e}$ to get the denominator of the above formula. Note that the necessary sum-of-products involve exponentially many terms relative to $|\mathbf{Z}|$, but if the computations are performed *locally* in a join tree propagation or variable elimination process, it is almost always possible to replace it by a very compact factorization [2, 17, 18]. Local computation is a generic inference technique for all sorts of valuation algebras (see Section 3).

In the presence of context-specific independence, further efficiency improvements are possible. Consider the following simplified version of the original definition.

Definition 1 (Boutilier et al., 1996). *If* $\mathbf{X}$*,* $\mathbf{Y}$*, and* $\mathbf{Z}$ *are pairwise disjoint sets of variables, then* $\mathbf{X}$ *is* context-specific independent *of* $\mathbf{Y}$ *in the context* $\mathbf{z} \in \Omega_{\mathbf{Z}}$*, if* $p(\mathbf{X}|\mathbf{Y},\mathbf{z}) = p(\mathbf{X}|\mathbf{z})$ *whenever* $p(\mathbf{Y},\mathbf{z}) > 0$.

In the example of Figure 1(a), $\{D\}$ is context-specific independent of $\{B, C\}$ in the context a_1. Similarly, $\{D\}$ is context-specific independent of $\{C\}$ in the context (a_2, b_1), and so on.

The classical approach to exploit context-specific independence is to use tree-based CPT representations [3, 8, 9]. An example of such a *CPT-tree* (or *probability tree*) is depicted in Figure 1(b) for the CPT $p(D|A, B, C)$ in the BN of Figure 1(a). Each node in the tree represents a decision w.r.t. the possible values of the indicated variable, and the values attached to the terminal nodes are the conditional probabilities $p(d_i|\mathbf{z})$, where $\mathbf{z}$ denotes the context specified by the path up to the root.

A more sophisticated CPT representation has been proposed in [11] to speed up the logical compilation of BNs. The idea is to use (ordered) *algebraic decision diagrams* [19], an extension of OBDDs to multiple terminal nodes. In the particular application of representing CSI, one can simply think of an ADD as a CPT-tree in which all identical nodes are merged, as shown in Figure 1(c). The result is obviously a more compact CPT representations. In extreme cases, ADDs are even exponentially smaller than corresponding CPT-tres. Nevertheless, ADDs inherit all the nice computational properties from OBDDs.

3 Inference in Valuation Algebras

To enlarge the applicability of the above ideas, the analysis is now shifted from BNs into the generic theory of *valuation algebras* [15]. The theory's basic elements are *valuations*, which can be regarded as pieces of information about the possible values of some variables. Thus, if $\mathbf{X}$ denotes the set of all variables relevant to a problem, then each valuation φ refers to a finite set of variables $d(\varphi) \subseteq \mathbf{X}$, called its *domain*. For an arbitrary set $\mathbf{Y} \subseteq \mathbf{X}$ of variables, $\Phi_{\mathbf{Y}}$ denotes the set of all valuations φ with $d(\varphi) = \mathbf{Y}$. With this notation, we can write

$$\Phi = \bigcup_{\mathbf{Y} \subseteq \mathbf{X}} \Phi_{\mathbf{Y}} \tag{4}$$

to denote the set of all possible valuations over $\mathbf{X}$. If $2^{\mathbf{X}}$ denotes the powerset[5] of $\mathbf{X}$, then

- *Labeling*: $\Phi \to 2^{\mathbf{X}}$, $\varphi \mapsto d(\varphi)$;
- *Combination*: $\Phi \times \Phi \to \Phi$, $(\varphi, \psi) \mapsto \varphi \otimes \psi$;
- *Variable elimination*: $\Phi \times \mathbf{X} \to \Phi$, $(\varphi, X) \mapsto \varphi^{-X}$;

are the three primitive operations of a valuation algebra.

[5] The more general definition given in [15] considers arbitrary distributive lattices. In this case, we must replace the operation of variable elimination by marginalization.

Definition 2 (Kohlas, 2003). *A tuple* $(\Phi, 2^{\mathbf{X}}, d, \otimes, -)$ *is a* valuation algebra*, if it satisfies the following set of axioms:*

1. Commutative Semigroup: *Φ is associative and commutative under $\otimes$.*
2. Labeling*: If φ, $\psi \in \Phi$, then $d(\varphi \otimes \psi) = d(\varphi) \cup d(\psi)$.*
3. Variable Elimination*: If $\varphi \in \Phi$ and $X \in d(\varphi)$, then $d(\varphi^{-X}) = d(\varphi) - \{X\}$.*
4. Commutativity of Elimination*: If $\varphi \in \Phi_{\mathbf{X}}$ and $X, Y \in d(\varphi)$, then $(\varphi^{-X})^{-Y} = (\varphi^{-Y})^{-X}$.*
5. Combination*: If φ, $\psi \in \Phi$ with $X \notin d(\varphi)$ and $X \in d(\psi)$, then $(\varphi \otimes \psi)^{-X} = \varphi \otimes \psi^{-X}$.*

Instances of valuation algebras are large in number and occur in very different contexts. One of the most prominent instances are the CPTs of BNs, where multiplication and summation over tables are the operations of combination and variable elimination, respectively. Valuations of this particular type are often called *probability potentials* [20]. For an extensive list of valuation algebra instances, we refer to [15].

3.1 Semiring Valuations

An important class of valuation algebras, which actually covers a majority of the known instances, results from the notion of *semiring valuations*. For this, let the elements $X \in \mathbf{X}$ be binary variables with frames $\Omega_X = \{x_1, x_2\}$.[6] If $\mathbf{Y} \subseteq \mathbf{X}$ is a subset of variables, then the Boolean vectors $\mathbf{y} \in \Omega_{\mathbf{Y}}$ are called configurations of $\mathbf{Y}$. By convention, we define the frame of the empty variable set as $\Omega_\emptyset = \{\diamond\}$. Furthermore, we write $\mathbf{y}^{\downarrow \mathbf{Z}}$ for the projection of some configuration $\mathbf{y} \in \Omega_{\mathbf{Y}}$ to a subset $\mathbf{Z} \subseteq \mathbf{Y}$. In particular, we have $\mathbf{y}^{\downarrow \emptyset} = \diamond$.

Consider now a (commutative) *semiring* $\mathcal{A} = \langle A, +, \times \rangle$, i.e. an algebraic structure over a set of values A, where the operations $+$ and $\times$ are both associative and commutative, and where $\times$ distributes over $+$.

Definition 3 (Kohlas, 2004). *A* semiring valuation *φ with domain $d(\varphi) = \mathbf{Y}$ is a mapping $\varphi : \Omega_{\mathbf{Y}} \to A$ from the set of configurations $\Omega_{\mathbf{Y}}$ to the set of values A of a semiring $\mathcal{A} = \langle A, +, \times \rangle$.*

With respect to the set Φ of all semiring valuations over the variables $\mathbf{X}$, the operations of combination and variable elimination are defined in terms of the semiring operations $+$ and $\times$:

– *Combination*: for $\mathbf{Y}, \mathbf{Z} \subseteq \mathbf{X}$, $\varphi \in \Phi_{\mathbf{Y}}$, $\psi \in \Phi_{\mathbf{Z}}$, and $\mathbf{x} \in \Omega_{\mathbf{Y} \cup \mathbf{Z}}$, let

$$\varphi \otimes \psi\,(\mathbf{x}) := \varphi\left(\mathbf{x}^{\downarrow d(\varphi)}\right) \times \psi\left(\mathbf{x}^{\downarrow d(\psi)}\right).$$

[6] The theory of semiring valuation algebras can be developed with arbitrary finite variables. Here, we restrict ourselves to binary variables which will allow us later to identify configurations with models of Boolean functions. Note that this is no conceptual restriction [21].

	a_1	a_2
φ_A	0.15	0.85

	b_1	b_2
φ_B	0.25	0.75

	c_1	c_2
φ_C	0.35	0.65

$\varphi_{D\|A,B,C}$	a_1 b_1 c_1 d_1	a_1 b_1 c_1 d_2	a_1 b_1 c_2 d_1	a_1 b_1 c_2 d_2	a_1 b_2 c_1 d_1	a_1 b_2 c_1 d_2	a_1 b_2 c_2 d_1	a_1 b_2 c_2 d_2	a_2 b_1 c_1 d_1	a_2 b_1 c_1 d_2	a_2 b_1 c_2 d_1	a_2 b_1 c_2 d_2	a_2 b_2 c_1 d_1	a_2 b_2 c_1 d_2	a_2 b_2 c_2 d_1	a_2 b_2 c_2 d_2
	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.8	0.2	0.8	0.2	0.2	0.8	0.8	0.2

Fig. 2. The semiring valuations φ_A, φ_B, φ_C, and $\varphi_{D|A,B,C}$ representing the CPTs $p(A)$, $p(B)$, $p(C)$, and $p(D|A,B,C)$ of the BN in Figure 1(a).

– *Variable Elimination*: for $\mathbf{Y} \subseteq \mathbf{X}$, $\varphi \in \Phi_{\mathbf{Y}}$, $X \in \mathbf{Y}$, and $\mathbf{z} \in \Omega_{\mathbf{Y}\setminus\{X\}}$, let

$$\varphi^{-X}(\mathbf{z}) := \varphi(\mathbf{z}, x_1) + \varphi(\mathbf{z}, x_2).$$

The most important property of semiring valuations is described in the following theorem.

Theorem 1 (Kohlas, 2004; Kohlas & Wilson, 2006). *A set Φ of semiring valuations, with labeling, combination and variable elimination as defined above, satisfies the axioms of a valuation algebra.*

The insight that every semiring induces a valuation algebra foreshadows the richness of formalisms that are covered by this theory. If we take for example the *arithmetic semiring* $\langle \mathbb{R}_0^+, +, * \rangle$, we obtain the valuation algebra of probability potentials, which are commonly used as a CPT representation for BNs [20]. Figure 2 illustrates this for the CPTs of the BN in Figure 1(a).

3.2 Local Computation

The computational interest in valuation algebras arises from the following notion of an *inference problem* and the generality of the resulting solution. For a given set of valuations $\{\varphi_1, \ldots, \varphi_n\}$, called the *knowledge base*, the inference problem consists in eliminating from the joint valuation $\varphi = \varphi_1 \otimes \cdots \otimes \varphi_n$ with $d(\varphi) = \mathbf{X}$ all variables that do not belong to some set $\mathbf{Q} \subseteq \mathbf{X}$ of *query variables*. More formally, this means the computation of

$$\varphi^{-\mathbf{X}\setminus\mathbf{Q}} = (\varphi_1 \otimes \cdots \otimes \varphi_n)^{-\mathbf{X}\setminus\mathbf{Q}}. \tag{5}$$

Note that the transitivity of the variable elimination allows us to eliminate sets of variables without further specifying the ordering (see Axiom 4).

To solve the inference problem efficiently, it is clear that an explicit computation of the joint valuation is normally not feasible.[7] *Local computation* methods counteract this problem by organizing the computations in such a way that the

[7] In most cases, the complexity of valuation algebra operations tends to increase exponentially with the size of the involved domains.

maximal domain size remains reasonably bounded. In the following, we restrict our attention to one such algorithm called *fusion algorithm*[8] [23] and refer to [24] for a broad discussion of related local computation schemes.

To describe the fusion algorithm, we consider first the elimination of a single variable X from a set of valuations $\Psi \subseteq \Phi$, which is defined by

$$\mathrm{Fus}_X(\Psi) := \{\psi_X^{-X}\} \cup \{\varphi \in \Psi : X \notin d(\varphi)\} \tag{6}$$

with $\psi_X = \otimes\{\varphi \in \Psi : X \in d(\varphi)\}$. The fusion algorithm follows then from a repeated application of this basic operation to all variables in $\mathbf{X} \backslash \mathbf{Q} = \{X_1, \ldots, X_k\}$. This leads to the following general solution for the inference problem:

$$\begin{aligned}\varphi^{-\mathbf{X}\backslash\mathbf{Q}} &= (\varphi_1 \otimes \cdots \otimes \varphi_n)^{-\{X_1,\ldots,X_k\}} \\ &= \otimes\mathrm{Fus}_{X_k}(\cdots(\mathrm{Fus}_{X_1}(\{\varphi_1,\ldots,\varphi_n\})\cdots).\end{aligned} \tag{7}$$

We refer to [15] for a proof and further considerations regarding the complexity of this generic inference algorithm.

In the particular case of probabilistic inference in a BN, the fusion algorithm has to be performed twice, once for the query variables $\mathbf{H} = \{H\} \cup \mathbf{E}$ and once for $\mathbf{Q} = \mathbf{E}$ (see Section 2). The resulting valuation $\varphi^{-\mathbf{X}\backslash\mathbf{H}}$ contains the probabilities $P(h, \mathbf{e})$ of all configurations $(h, \mathbf{e}) \in \Omega_{\mathbf{H}}$. Similarly, $\varphi^{-\mathbf{X}\backslash\mathbf{Q}}$ contains the probabilities $P(\mathbf{e})$ of all configurations $\mathbf{e} \in \Omega_{\mathbf{Q}} = \Omega_{\mathbf{E}}$. This means that $P(h|\mathbf{e}) = P(h, \mathbf{e})/P(\mathbf{e})$ can be derived from the corresponding semiring values of $(h, \mathbf{e})$ in $\varphi^{-\mathbf{X}\backslash\mathbf{H}}$ and $\mathbf{e}$ in $\varphi^{-\mathbf{X}\backslash\mathbf{Q}}$, respectively.[9]

4 Compact Representations

An optimized CPT representation is important to further speed up inference in BNs, i.e. to go beyond the capacities offered by local computation. The CSI approach with its tree-structured representations (see Section 2) is a good starting point, but now we will show that we can do better than that. For this, we will no longer look at identical CPT entries as the result of CSI, but instead consider them from a purely technical point of view within the generic framework of semiring valuations. This will then allow us to use the knowledge compilation map to select the most appropriate representation language.

4.1 Partitioned Semiring Valuations

The basic idea consists in partitioning the configurations space $\Omega_{\mathbf{Y}}$ of a semiring valuation φ with $d(\varphi) = \mathbf{Y}$ according to its semiring values into a collection

[8] Other names for exactly the same type of algorithm are *bucket elimination* [18] or simply *variable elimination* [22].

[9] This way of using the fusion algorithm may not always be optimal, especially if $\mathbf{E}$ is large. A better idea is to create for each evidence variable $E \in \mathbf{E}$ an additional *evidence valuation* φ_E with $d(\varphi_E) = \{E\}$ and $\varphi_E(e) = 1$, and to apply the fusion algorithm for $\mathbf{H} = \{H\}$ and $\mathbf{Q} = \emptyset$ to the extended set of valuations.

$\varphi_{D\|A,B,C}$		
S_1	$\{(a_1,b_1,c_1,d_1),(a_1,b_1,c_2,d_1),(a_1,b_2,c_1,d_1),(a_1,b_2,c_2,d_1)\}$	0.9
S_2	$\{(a_1,b_1,c_1,d_2),(a_1,b_1,c_2,d_2),(a_1,b_2,c_1,d_2),(a_1,b_2,c_2,d_2)\}$	0.1
S_3	$\{(a_2,b_1,c_1,d_1),(a_2,b_1,c_2,d_1),(a_2,b_2,c_1,d_2),(a_2,b_2,c_2,d_1)\}$	0.8
S_4	$\{(a_2,b_1,c_1,d_2),(a_2,b_1,c_2,d_2),(a_2,b_2,c_1,d_1),(a_2,b_2,c_2,d_2)\}$	0.2

(a) Sets

$f_i : \Omega_{\{A,B,C,D\}} \rightarrow \{0,1\}$		
f_1	$a_1 \wedge d_1$	0.9
f_2	$a_1 \wedge d_2$	0.1
f_3	$a_2 \wedge ((b_1 \wedge d_1) \vee (b_2 \wedge ((c_1 \wedge d_2) \vee (c_2 \wedge d_1))))$	0.8
f_4	$a_2 \wedge ((b_1 \wedge d_2) \vee (b_2 \wedge ((c_1 \wedge d_1) \vee (c_2 \wedge d_2))))$	0.2

(b) Boolean Functions

Fig. 3. The partitioning of the configurations in the semiring valuation $\varphi_{D|A,B,C}$ into sets of configurations and their representation as Boolean functions.

$\{S_1,\ldots,S_s\}$ of exclusive and exhaustive subsets $S_i \subseteq \Omega_{\mathbf{Y}}$. In other words, instead of mapping single configurations into (possibly identical) semiring values, we will now map partitions of configurations S into (pairwise distinct) semiring values $\varphi(S) \in A$. A valuation represented in this way will be called *partitioned semiring valuation*. Note that in the extreme case, where $\varphi(\mathbf{y}) = c$ is the same constant value $c \in A$ for all $\mathbf{y} \in \Omega_{\mathbf{Y}}$, we will end up with a single partition $\Omega_{\mathbf{Y}}$ with $\varphi(\Omega_{\mathbf{Y}}) = c$. Figure 3(a) shows the partitioned semiring valuation $\varphi_{D|A,B,C}$ of Figure 2.

With this idea in mind, it is clear that the question of representing semiring valuations becomes a question of representing sets of configurations, i.e. subsets of a Cartesian product. In the case of binary variables, we can identify such a set $S \subseteq \Omega_{\mathbf{Y}}$ with a *Boolean function* (BF) $f : \Omega_{\mathbf{Y}} \rightarrow \{0,1\}$, which evaluates to 1 for all $\mathbf{y} \in S$ and to 0 for all $\mathbf{y} \notin S$. Then S becomes the so-called *satisfying set* of f. With this, our problem of representing semiring valuations turns into a problem of representing Boolean functions.[10]

The optimal representation of a BF is a lively research topic with contributions from many different areas. To get a good survey of the vast number of existing techniques and their relationships, the most convenient and comprehensive access is the *knowledge compilation map* in [13] and its extension in [14]. The goal of this map is to support the identification of the most appropriate representation language according to the *queries* and *transformations* it is supposed to offer in polynomial time. Therefore, as soon as we know which queries and

[10] The more general case of non-binary variables leads to general indicator functions, for which similar representation languages and an analogue knowledge compilation map exist [21].

transformations are required for dealing with partitioned semiring valuations, we can use the map to identify the most appropriate language.

In the following, we use $\mathbf{0}$ to denote the constant BF that always evaluates to 0. Similarly, $\mathbf{1}$ denotes the constant BF that always evaluates to 1. Furthermore, $x_i \in \Omega_X$ represents the BF which evaluates to 1 iff $X = x_i$. If f_1 and f_2 are Boolean functions, then $f_1 \wedge f_2$ is the BF that evaluates to 1, iff both f_1 and f_2 evaluate to 1. Similarly, $f_1 \vee f_2$ denotes the BF that evaluates to 1, iff either f_1 or f_2 evaluates to 1. Figure 3(b) shows the partitioned semiring valuation $\varphi_{D|A,B,C}$ in terms of their BFs.

4.2 Queries and Transformations

To determine the required queries and transformations, the two essential operations of combination and variable elimination have to be analyzed in the light of the suggested representation. The labeling operation is negligible, since it can be achieved easily. In the following, let (f_i, v_i), $i \in \{1, \ldots, s\}$, be the entries of a partitioned semiring valuation $\varphi \in \Phi_{\mathbf{Y}}$. With f_i we denote the BF of the partition S_i and with $v_i = \varphi(S_i)$ the corresponding semiring value. Similarly, let (g_j, w_j), $j \in \{1, \ldots, t\}$, be the entries of another partitioned semiring valuation $\psi \in \Phi_{\mathbf{Z}}$, where g_j denotes the BF of the partition T_i and $w_j = \psi(T_i)$ the corresponding semiring value.

Let us now discuss the combination and variable elimination according to the definitions given in Section 3.

Combination. It is quite obvious that $\varphi \otimes \psi$ essentially consists of all combined entries $(f_i \wedge g_j, v_i \times w_j)$, except of the ones with $f_i \wedge g_j = \mathbf{0}$. In the terminology of [14], $f_i \wedge g_j$ corresponds to the transformation "*binary conjunction*", denoted by $\mathtt{AND}_2$, and the test $f_i \wedge g_j = \mathbf{0}$ corresponds to the query "*consistency test*", denoted by `CO`. Hence, both $\mathtt{AND}_2$ and `CO` are required for the combination of two partitioned semiring valuations.

Variable Elimination. For the elimination of a variable $X \in d(\varphi)$ from φ, we will now see that φ^{-X} can be expressed in terms of an operation similar to the above combination. For this, let $\varphi|x$ denote the result of "*conditioning*" φ on a value $x \in \Omega_X$ using the method proposed in [13]. The idea is to delete from each partition S_i the configurations which do not contain x, and then project all remaining configurations to $\mathbf{Y} \setminus X$. In terms of the involved BFs, this can be achieved by conditioning each f_i on x. Note that $f_i|x$ may become equivalent to $\mathbf{0}$ for some $i \in \{1, \ldots, s\}$. In [14], the conditioning of a Boolean function is denoted by `TC` (for *term conditioning*).

In the binary case, i.e. for $\Omega_X = \{x_1, x_2\}$, let $I, J \subseteq \{1, \ldots, s\}$ contain the indices of each $f_i|x_1 \neq \mathbf{0}$ resp. $f_i|x_2 \neq \mathbf{0}$. With this, we obtain φ^{-X} by computing all combined entries $(f_i|x_1 \wedge f_j|x_2, v_i + v_j)$ for $i \in I$ and $j \in J$. Note that we may again get and delete some functions equivalent to $\mathbf{0}$. Eliminating a variable from a partitioned semiring valuations requires thus `TC`, $\mathtt{AND}_2$, and `CO`.

To complete our analysis of the above operations, consider the case where some of the resulting partitions obtain the same semiring value $a \in A$. This may occur frequently[11] (especially if A is small) and for both the combination (as a result of the semiring operation $\times$) and the variable elimination (as a result of the semiring operation $+$). To merge two such entries (f_1, a) and (f_2, a) into $(f_1 \vee f_2, a)$, an additional transformation $\texttt{OR}_2$ (called *binary disjunction*) is required. In general, we may need to merge several entries $(f_1, a), \ldots, (f_k, a)$ into $(f_1 \vee \cdots \vee f_k, a)$, which requires the transformation OR (called *general disjunction*).

4.3 Selecting the Optimal Representation Language

Now that we know that working with partitioned semiring valuations requires TC, CO, $\texttt{AND}_2$, and $\texttt{OR}_2$ (or preferably OR), we may use the knowledge compilation map in [13, 14] to select the most appropriate language. It turns out that two languages are valuable candidates:

- OBDDs with a fixed variable ordering support TC, CO, $\texttt{AND}_2$, and $\texttt{OR}_2$ (but not OR) in polynomial time;
- DNFs support TC, CO, $\texttt{AND}_2$, $\texttt{OR}_2$, and OR in polynomial time.

No other language offers CO and $\texttt{AND}_2$ together. Note that in terms of their succinctness, the OBDD and DNF languages are incomparable [13]. Nevertheless, OBDDs are often much smaller than corresponding DNFs, which is why we recommend OBDDs to be used as representation language for partitioned semiring valuations.[12] Note that the proposed rule-based representation proposed in [10] is very close to using DNFs.

Figure 4 shows the OBDDs representing the BFs f_1, f_2, f_3, and f_4 of the partitioned semiring valuation $\varphi_{D|A,B,C}$ together with the associated semiring values. Note that there is some substantial overlap between the OBDDs of f_3 and f_4. Such overlapping OBDD structures are called *shared OBDDs* in [25].

5 Conclusion

This paper has two main messages. The first one is that the results in the context of optimizing the representation of a BN can be applied to semiring valuations. We also conclude that OBDDs (and possibly DNFs) form the most appropriate language. Our argumentation is entirely based on the knowledge compilation map and covers the special case of representing CPTs (and CSI) in Bayesian networks. Future work will focus on elaborating a generic approximation method, its implementation, and experimental evaluation thereof. In addition, we will analyze the relationship between shared OBDD and ADD representations, which appear to be closely connected.

[11] Or never, such as in the compilation method described in [11], where A itself is a set of Boolean functions.

[12] In the knowledge compilation map, the DNF language is situated in the family of *flat* languages [13], which are generally not very attractive.

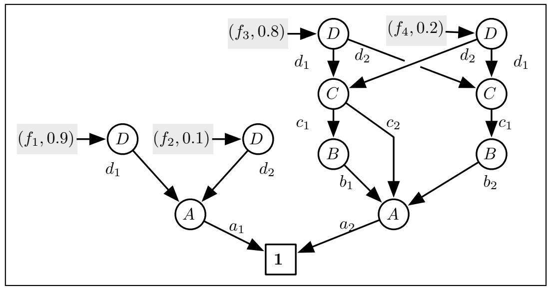


Fig. 4. The OBDDs representing the BFs of the semiring valuation $\varphi_{D|A,B,C}$. Edges leading towards 0 and the node 0 itself are omitted.

Acknowledgements

This research is supported by the *Swiss National Science Foundation*, Project No. PP002-102652/1 & Grant No. 200020-109510, and *The Leverhulme Trust*.

References

1. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Mateo, USA (1988)
2. Shenoy, P.P., Shafer, G.: Axioms for probability and belief-function propagation. In Shachter, R.D., Levitt, T.S., Lemmer, J.F., Kanal, L.N., eds.: UAI'88, 4th Conference on Uncertainty in Artificial Intelligence, Minneapolis, USA (1988) 169–198
3. Boutilier, C., Friedman, N., Goldszmidt, M., Koller, D.: Context-specific independence in Bayesian networks. In Horvitz, E., Jensen, F., eds.: UAI'96, 12th Conference on Uncertainty in Artificial Inteligence, Portland, USA (1996) 115–123
4. Smith, J.E., Holtzman, S., Matheson, J.E.: Structuring conditional relationships in influence diagrams. Operations Research **41**(2) (1993) 280–297
5. Wong, S.K., Butz, C.: Contextual weak independence in Bayesian networks. In Laskey, K.B., Prade, H., eds.: UAI'99, 15th Conference on Uncertainty in Artificial Intelligence, Stockholm, Sweden (1999) 670–679
6. Geiger, D., Heckerman, D.: Advances in probabilistic reasoning. In: UAI'91, 6th Annual Conference on Uncertainty in Artificial Intelligence, Los Angeles, USA (1991) 118–126
7. Galles, D., Pearl, J.: Axioms of causal relevance. Artificial Intelligence **97**(1–2) (1997) 9–43
8. Cano, A., Moral, S., Salmeron, A.: Penniless propagation in join trees. International Journal of Intelligent Systems **15**(11) (2000) 1027–1059
9. Bellot, D., Bessière, P.: Approximate discrete probability distribution representation using a multi-resolution binary tree. In: ICTAI'03, 15th IEEE International Conference on Tools with Artificial Intelligence, Sacramento, USA (2003) 498–503
10. Poole, D., Zhang, N.L.: Exploiting contextual independence in probabilistic inference. Journal of Artificial Intelligence Research **18** (2003) 263–313
11. Chavira, M., Darwiche, A.: Compiling Bayesian networks using variable elimination. In: IJCAI'07, 20th International Joint Conference on Artificial Intelligence, Hyderabad, India (2007)

12. Bryant, R.E.: Graph-based algorithms for Boolean function manipulation. IEEE Transactions on Computers **35**(8) (1986) 677–691
13. Darwiche, A., Marquis, P.: A knowledge compilation map. Journal of Artificial Intelligence Research **17** (2002) 229–264
14. Wachter, M., Haenni, R.: Propositional DAGs: a new graph-based language for representing Boolean functions. In Doherty, P., Mylopoulos, J., Welty, C., eds.: KR'06, 10th International Conference on Principles of Knowledge Representation and Reasoning, Lake District, U.K., AAAI Press (2006) 277–285
15. Kohlas, J.: Information Algebras: Generic Stuctures for Inference. Springer, London (2003)
16. Kohlas, J., Wilson, N.: Exact and approximate local computation in semiring induced valuation algebras. Technical Report 06–06, Department of Informatics, University of Fribourg, Switzerland (2006)
17. Kohlas, J., Shenoy, P.P.: Computation in valuation algebras. In Gabbay, D.M., Smets, P., eds.: Handbook of Defeasible Reasoning and Uncertainty Management Systems. Volume 5: Algorithms for Uncertainty and Defeasible Reasoning. Kluwer Academic Publishers, Dordrecht, Netherlands (2000) 5–39
18. Dechter, R.: Bucket elimination: a unifying framework for reasoning. Artificial Intelligence **113**(1–2) (1999) 41–85
19. Bahar, R.I., Frohm, E.A., Gaona, C.M., Hachtel, G.D., Macii, E., Pardo, A., Somenzi, F.: Algebraic decision diagrams and their applications. In Lightner, M.R., Jess, J.A.G., eds.: 4th IEEE/ACM International Conference on Computer-Aided Design, Santa Clara, California (1993) 188–191
20. Shenoy, P.P., Shafer, G.: Axioms for probability and belief function propagation. In Shafer, G., Pearl, J., eds.: Readings in Uncertain Reasoning. Morgan Kaufmann, San Mateo, USA (1990) 575–610
21. Wachter, M., Haenni, R.: Multi-state directed acyclic graphs. In Kobti, Z., Wu, D., eds.: CanAI'07, 20th Canadian Conference on Artificial Intelligence. LNAI 4509, Montréal, Canada (2007) 464–475
22. Zhang, N.L., Poole, D.: Exploiting causal independence in Bayesian network inference. Journal of Artificial Intelligence Research **5** (1996) 301–328
23. Shenoy, P.P.: Valuation-based systems: A framework for managing uncertainty in expert systems. In Zadeh, L.A., Kacprzyk, J., eds.: Fuzzy Logic for the Management of Uncertainty. John Wiley and Sons, New York, USA (1992) 83–104
24. Schneuwly, C., Pouly, M., Kohlas, J.: Local computation in covering join trees. Technical Report 04–16, Department of Informatics, University of Fribourg, Switzerland (2004)
25. Wegener, I.: Branching Programs and Binary Decision Diagrams – Theory and Applications. Number 56 in Monographs on Discrete Mathematics and Applications. SIAM (2000)