View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Elsevier - Publisher Connector

Available online at www.sciencedirect.com
ScienceDirect
Artificial Intelligence 172 (2008) 1360–1399
Artificial Intelligence
www.elsevier.com/locate/artint

Semiring induced valuation algebras: Exact and approximate local computation algorithms

J. Kohlas [a,1], N. Wilson [b,*,2]

[a] *Department of Informatics, University of Fribourg, Switzerland*
[b] *Cork Constraint Computation Centre, Department of Computer Science, Ireland*

Received 4 April 2006; received in revised form 7 March 2008; accepted 18 March 2008

Available online 27 March 2008

Abstract

Local computation in join trees or acyclic hypertrees has been shown to be linked to a particular algebraic structure, called valuation algebra. There are many models of this algebraic structure ranging from probability theory to numerical analysis, relational databases and various classical and non-classical logics. It turns out that many interesting models of valuation algebras may be derived from semiring valued mappings. In this paper we study how valuation algebras are induced by semirings and how the structure of the valuation algebra is related to the algebraic structure of the semiring. In particular, c-semirings with idempotent multiplication induce idempotent valuation algebras and therefore permit particularly efficient architectures for local computation. Also important are semirings whose multiplicative semigroup is embedded in a union of groups. They induce valuation algebras with a partially defined division. For these valuation algebras, the well-known architectures for Bayesian networks apply. We also extend the general computational framework to allow derivation of bounds and approximations, for when exact computation is not feasible.
© 2008 Elsevier B.V. All rights reserved.

Keywords: Semirings; Local computation; Join tree decompositions; Soft constraints; Uncertainty; Valuation networks; Valuation algebras

1. Introduction

Many different formalisms from artificial intelligence, including constraint systems, probabilistic networks, systems of possibility measures or belief functions, from database theory, from logic, statistics and from numerical analysis exhibit a common structure permitting local computation, i.e. computation on acyclic hypertrees, or join trees. This algebraic structure has first been isolated in an abstract setting and related to local computation in [59], see also [37,53]. It has been further extended and studied in detail in [34]. The algebraic structure has been called a *valuation algebra* in [34].

* Corresponding author.
E-mail addresses: juerg.kohlas@unifr.ch (J. Kohlas), n.wilson@4c.ucc.ie (N. Wilson).
URL: http://diuf.unifr.ch/tcs/juerg.kohlas (J. Kohlas).
[1] Research supported by grant No. 2100–042927.95 of the Swiss National Foundation for Research.
[2] This material is based partly upon works supported by the Science Foundation Ireland under grant No. 00/PI.1/C075 and grant No. 05/IN/I886.

0004-3702/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.artint.2008.03.003

In a valuation algebra, each piece of information ϕ, called a *valuation*, has an associated set s of variables; ϕ gives information about variables s. For example, in a constraint satisfaction problem or relational database ϕ may express a relation on s, saying which assignments of these variables are feasible. Alternatively, in a soft constraints system it may express preferences for different assignments, or in a system for reasoning with uncertainty such as possibility theory or Bayesian networks, it may express degrees of uncertainty of the different assignments to variables s.

It is assumed that we have a way of combining valuations, through an operation $\otimes$, which gives their combined effect; the combination operator is associative and commutative. If ψ is another valuation on variables t then the combination $\phi \otimes \psi$ is a valuation on variables $s \cup t$, since ϕ and ψ together say something about variables $s \cup t$.

Valuation algebras also assume another operation, called *projection* or *marginalization*, which focuses information onto a smaller set of variables. Suppose u is a subset of s. Then $\phi^{\downarrow u}$ represents what valuation ϕ tells us about u. If, for example, ϕ is a probability distribution or potential, then $\phi^{\downarrow u}$ is the marginal on u. Alternatively, if ϕ represents a binary constraint relating variables X_1 and X_2 then it tells us which assignments to $\{X_1, X_2\}$ are possible, and $\phi^{\downarrow \{X_1\}}$ tells us which assignments to X_1 are possible.

The inputs of many important computational reasoning problems can be expressed as a collection of valuations $\phi_1, \ldots, \phi_k$ (in an appropriate valuation algebra), where the associated sets of variables are all fairly small. The combination of these gives us the combined effect of all our information. For example, in a constraint satisfaction problem, the combination represents all the solutions, and in a relational database, the combination is the join of all the relations. In a Bayesian network, the combination represents the full probability distribution over all the variables. It will very often be infeasible to represent this whole combination directly, since involves all the variables, for which there are an exponential number of assignments. Typically we are interested in what the information tells us about certain small sets of variables. So, for particular sets u, we want to compute the projection of the whole combination to u. This is known as the *projection problem*. Direct computation is very often not feasible. For a single set u, an approach based on sequential variable elimination can be used to compute the associated marginal. For the computation for several sets u, faster methods have been developed based on use of an appropriate *join tree*, that is, a tree whose nodes are associated with sets of variables which satisfy the running intersection property: that if variable X is associated with two nodes then it is associated with every node in the path between the two nodes.

Such join tree algorithms for computing several marginals have two parts: an inward phase where information is passed iteratively from the leaves to a chosen root node; and an outward phase where information is distributed out again, iteratively from the root to all the nodes. As discussed below (and in more detail in [34]) there are a number of different variations on this local computational architecture.

It turns out that many important examples of valuation algebras can be induced by valuations taking values in a semiring. This has first been proposed in the domain of constraint systems, where classical crisp constraints are generalized to fuzzy constraints, weighted constraints and partially satisfied constraints [8,9]. But probability potentials as used in Bayesian networks [41] belong also to the same class of valuations, as do relational systems [4,42]. Possibility potentials and Spohn potentials [60] provide further examples of valuations based on semirings. Other instances and applications of semiring-induced valuation algebras are described in [1].

In the second section we introduce semirings and give several examples which are related to important valuation algebras. A semiring consists of a set A with two operations on it, conventionally labeled $+$ and $\times$, both of which we assume to be associative and commutative; it is also assumed that $\times$ distributes over $+$. An example is the nonnegative real numbers under addition and multiplication. A valuation on variables s in the induced valuation algebra is a function which assigns a nonnegative real number to each assignment to variables s. Combination is based on pointwise multiplication, and marginalization involves summation over the values of variables being eliminated. This semiring induced valuation algebra is therefore that of probability potentials, used for reasoning with Bayesian networks.

Any semiring induces a valuation algebra in just the same way, as shown in Section 3, which also discusses local computation based on these semirings.

In this paper we study valuation algebras induced by semirings in some detail. In particular, we want to know how the structure of a valuation algebra is conditioned by the structure of the inducing semiring. These are important questions for practical purposes: Valuation algebras provide the structure needed for local computation architectures.

There exist particularly efficient architectures which use some form of division in the valuation algebra [29,41]. It is therefore important to know, what properties of the inducing semiring guarantee the existence of a concept of division in the induced valuation algebra and thus the usability of the corresponding architecture for local computation [34,37]. Further, idempotent valuation algebras, so-called information algebras, have interesting properties and allow particularly simple local computation architectures. Therefore it is important to know which semirings lead to idempotent valuation algebras. Idempotent valuation algebras and their corresponding computational structure are analyzed in Section 4, and in Section 5 we study semirings which induce valuation algebras with division and discuss their local computation architectures. This study helps to extend computational schemes, well known in probability networks and relational algebra, to more general structures and to develop generic architectures for local computation [47].

It may well happen that, for a problem of interest, exact local computation is not feasible. For certain important systems of valuations, it has been demonstrated, e.g., in [24], how the local computations can be approximated using the 'mini-buckets' and 'mini-clustering' techniques. We show in Section 6 how this kind of technique can be applied very generally to valuation algebras, in particular, those induced by a semiring; we focus especially on computing upper and lower bounds. We also consider the use of constraint propagation for improving the efficiency of local computation.

Provisional versions of parts of this work appear in [35] and [62].

2. Semirings

This section defines different kinds of semirings which are relevant to valuation algebras of interest in areas of automated reasoning, such as uncertain reasoning and constraint-based reasoning.

Semirings are algebraic structures composed of two operations. So, let A be a set with two binary operations $+$ and $\times$ defined in it. We call a tuple $\mathcal{A} = \langle A, +, \times \rangle$ a *semiring*, if both operations $+$ and $\times$ are commutative and associative, and if $\times$ distributes over $+$. Elsewhere this is often called a *commutative semiring*. If there is an element $0 \in A$ such that $0 + a = a + 0 = a$ and $0 \times a = a \times 0 = 0$ for all $a \in A$, then A is called a semiring with *zero element*. In this case the zero element 0 is clearly unique. A zero element can be adjoined to any semiring. Let $\langle A, +, \times \rangle$ be a semiring. Add an extra element 0 to A and extend $+$ and $\times$ to $A \cup \{0\}$ by, for all $a \in A$, $a + 0 = 0 + a = a$ and $a \times 0 = 0 \times a = 0$. Then it is easy to verify that $\langle A \cup \{0\}, +, \times \rangle$ is a semiring.

Similarly, an element $1 \in A$ is said to be a *unit element*, if $1 \times a = a \times 1 = a$ for all $a \in A$. There can be at most one unit element 1 in a semiring. Note that if in these cases A is a group under the operation $+$, then A is a *ring*. If furthermore $A - \{0\}$ is a group under the operation $\times$, then A is a *field*.

The associativity of $+$ allows us to write expressions like $a_1 + a_2 + \cdots + a_n$ or $\sum_i a_i$, and in particular, if $I = I_1 \cup \cdots \cup I_n$, where the I_j are finite and disjoint, then commutativity and associativity entail that

$$\sum_{j=1}^{n} \sum_{i \in I_j} a_i = \sum_{i \in I} a_i.$$

If A is a semiring with zero element and if

$$a + b = 0 \quad \text{implies} \quad a = b = 0,$$

then A is called *positive*. In the same way, the associativity of $\times$ permits to write expressions like $a_1 \times a_2 \times \cdots \times a_n$ or $\prod_i a_i$.

If the operation $+$ is idempotent, i.e. $a + a = a$ for all $a \in A$, then the semiring $\mathcal{A} = \langle A, +, \times \rangle$ can be extended to include a unit element as follows: For each $a \in A$ define a new element a_1 such that $a \neq b$ implies $a_1 \neq b_1$. Let then $A' = A \cup A_1$, where $A_1 = \{a_1\colon a \in A\}$. Define $+'$ as follows, when a and b are arbitrary elements of A: $a +' b = a + b$, and $a +' b_1$, $a_1 +' b$ and $a_1 +' b_1$ are all defined to be $(a + b)_1$. Further define $\times'$ as follows: $a \times' b = a \times b$ and $a \times' b_1$ and $a_1 \times' b$ are both defined to be $(a \times b) + a$ and $a_1 \times' b_1$ is defined to be $(a_1 \times' b) +' a_1$. The system $\mathcal{A}' = \langle A', +', \times' \rangle$ is then a semiring with unit element 0_1; and $+'$ is also idempotent.

Let $\mathcal{A} = \langle A, +, \times \rangle$ be a semiring. We define a relation $\preceq_A$ on A (abbreviated to $\preceq$ in this section) by: $a \preceq b$, if and only if, either $a = b$ or there exists a $c \in A$ such that $a + c = b$. If $\mathcal{A}$ has a zero element, then the last condition covers the first one, since we may take $c = 0$.

Proposition 1. *For any semiring $\mathcal{A} = \langle A, +, \times \rangle$, the associated relation $\preceq$ satisfies the following properties*:

(1) *Relation $\preceq$ is a pre-order, i.e., it is reflexive and transitive*;

(2) *$a \preceq b$ and $a' \preceq b'$ imply $a + a' \preceq b + b'$ and $a \times a' \preceq b \times b'$;*
(3) *if, for all $i = 1, \ldots, n$, $a_i \preceq b_i$, then $\sum_{i=1}^{n} a_i \preceq \sum_{i=1}^{n} b_i$ and $\prod_{i=1}^{n} a_i \preceq \prod_{i=1}^{n} b_i$;*
(4) *for all $a, b \in A$, $a \preceq a + b$ and if a has a zero element 0 then $0 \preceq a$;*
(5) *if $a + a = a$ and $b + b = b$, then $a \preceq b$ if, and only if, $a + b = b$.*

Proof. (1) Clearly $\preceq$ is reflexive. Suppose that $a \preceq b$ and $b \preceq c$. Thus, there exist d and e such that $a + d = b$ and $b + e = c$, hence $a + (d + e) = c$. This proves that $a \preceq c$.

(2) Suppose $a \preceq b$ and $a' \preceq b'$, i.e. there exist c and c' with $a+c = b$ and $a'+c' = b'$. Then $a+a'+(c+c') = b+b'$, hence $a+a' \preceq b+b'$ as required. For the second part it is sufficient to show that $a \preceq b$ implies $a \times a' \preceq b \times a'$ since this can be applied twice using commutativity of $\times$ to get the result. Suppose $a \preceq b$, so that there exists c with $a + c = b$. Then $b \times a' = (a + c) \times a' = (a \times a') + (c \times a')$ which implies that $a \times a' \preceq b \times a'$.

(3) follows by repeated application of (2).

(4) follows, since $0 + a = a$.

(5) If $a+b = b$, then by definition $a \preceq b$. Conversely, if $a \preceq b$, and there exists a c with $a+c = b$. Then $a+b+c = b + b = b$. Hence $a + b = a + a + b + c = a + b + c = b$ as required. □

Often the operation $+$ is assumed to be *idempotent*, i.e. $\forall a \in A$ we have $a + a = a$. Note that idempotency of $+$ is implied by idempotency of the unit, since if $1 + 1 = 1$ then $a + a = a \times (1 + 1) = a \times 1 = a$. If A has a zero and a unit element and if furthermore for all $a \in A$,

$$a + 1 = 1,$$

(and hence $+$ is idempotent) then we call A a *c-semiring* [9] ("c" standing for constraint) [8]. This is a special kind of commutative *dioid* [3]. According to Proposition 1 (4) the pre-order $\preceq_A$ becomes a *partial order* $\leqslant_A$ (abbreviated to $\leqslant$) in A defined in the following way:

$$a \leqslant_A b \quad \text{if, and only if,} \quad a + b = b.$$

The intended meaning of this order in applications is often that b is preferred over a, or that b is "better" than a. We refer to the examples below. The following lemma summarizes a few elementary, but important properties of this order.

Lemma 1. *Let A be a c-semiring.*

(1) *$\forall a \in A$ we have $0 \leqslant a \leqslant 1$;*
(2) *$\forall a, b \in A$ we have $a \leqslant a + b$ and $a \times b \leqslant a$;*
(3) *$a \leqslant a'$ and $b \leqslant b'$ imply $a + b \leqslant a' + b'$ and $a \times b \leqslant a' \times b'$;*
(4) *$a \times b = a$ implies $a \leqslant b$;*
(5) *$a + b = \sup\{a, b\}$;*
(6) *A is positive.*

Proof. (1) This follows from $0 + a = a$ and from $a + 1 = 1$.

(2) First we have $a + (a + b) = (a + a) + b = a + b$ by idempotency. Further, by the distributive law, $a + (a \times b) = (a \times 1) + (a \times b) = a \times (1 + b) = a \times 1 = a$ since $b \leqslant 1$ by (1).

(3) By assumption we have $a + a' = a'$ and $b + b' = b'$. Hence we obtain that $(a + b) + (a' + b') = (a + a') + (b + b') = a' + b'$ and also, by distributivity $(a \times b) + (a' \times b) = (a + a') \times b = a' \times b$. So we see that $a \times b \leqslant a' \times b$. But then it follows also that $a' \times b \leqslant a' \times b'$ and hence, by transitivity $a \times b \leqslant a' \times b'$.

(4) We have $a + b = (a \times b) + b = (a + 1) \times b = 1 \times b = b$.

(5) By (2) $a, b \leqslant a + b$. Let c be another upper bound of a and b, $a \leqslant c$ and $b \leqslant c$. Then by (3) $a + b \leqslant c + c = c$. Thus $a + b$ is the least upper bound.

(6) Suppose $a + b = 0$. Then $0 \leqslant a \leqslant a + b = 0$ (see Proposition 1). By transitivity of the order we get thus $0 \leqslant a \leqslant 0$, hence from the antisymmetry of the partial order $\leqslant$ it follows that $a = 0$. Similarly $b = 0$ can be derived. □

Our definition of c-semiring is equivalent to that given in [9], and that given in later papers such as [5]. The definition of c-semiring in [8] is somewhat stronger since it assumes that summation is defined over infinite sets. Most of the properties proved in [8] hold also for the slightly weaker definition of c-semiring, in particular properties in Lemma 1.

One result from [8] which does not hold for the weaker definition of c-semiring is Theorem 9 of [8], stating that a c-semiring is a complete lattice; with the definition used in this paper, a c-semiring is not necessarily a lattice, as shown by the following example.

Consider the set A of all finite unions of closed discs (i.e., circles and their interiors) in $\mathbb{R}^2$, together with the empty set and the whole set $\mathbb{R}^2$, and including also discs of radius zero, i.e., points. For $a, b \subseteq \mathbb{R}^2$ define $a + b = a \cup b$, and define (cf. Example 4.17 of [3]) $a \times b = \{x \in \mathbb{R}^2\colon\ x = y + z,\ y \in a, z \in b\}$, so that $a \times \emptyset = \emptyset$ for all $a \in S$. It can be shown that $+$ and $\times$ are commutative and associative and for all $a, b, c \subseteq \mathbb{R}^2$, $(a + b) \times c = (a \times c) + (b \times c)$, i.e. $\times$ distributes over $+$. A is clearly closed under $+$. It is also closed under $\times$: this follows using the fact that if a and b are discs then $a \times b$ is also a disc; this can be seen, for example, by translating both discs to have centre at the origin; disc $a \times b$ has radius equal to the sum of the radii of a and b. The distributive property then implies that A is closed under $\times$. Hence $\langle A, +, \times\rangle$ is a semiring with zero element $\emptyset$ and unit element $\mathbb{R}^2$. For all $a \in A$, $a + \mathbb{R}^2 = \mathbb{R}^2$, so $\langle A, +, \times\rangle$ is a c-semiring (with the above definition). Consider any pair a and b of overlapping discs, where neither contains the other. It can be seen that their intersection $a \cap b$ is not in A (e.g., by considering the curvature at a point on the boundary of $a \cap b$). Element $c \in A$ is a lower bound for a and b if and only if c is a subset of $a \cap b$. But a and b have no greatest lower bound in A. In fact, for any lower bound c in A, one can construct a strictly greater lower bound in A by taking $c \cup \{x\}$, where point x is an element of $(a \cap b) - c$. Therefore A is not a lattice.

We shall also consider c-semirings where the operation $\times$ is idempotent too, i.e. $a \times a = a$ for all $a \in A$. Then $a \leqslant b$ if $a \times b = a$ defines also a partial order in A. According to Lemma 1 it is identical to the order $\leqslant_A$.

Theorem 10 of [8] shows that a c-semiring (in their sense) with idempotent $\times$ is a distributive lattice. The following simple result states that this holds also for the definition of c-semiring used here. (However, unlike c-semirings defined in [8], it need not be a complete distributive lattice. Consider for example the c-semiring of rational numbers in the interval [0, 1] with $+$ being max and $\times$ being min.)

Theorem 1. *(Cf. Theorem 10 of [8].) If A is a c-semiring, and $\times$ idempotent, then A is a distributive lattice and $a \times b = \inf\{a, b\}$.*

Proof. Since $\sup\{a, b\} = a + b$ exists, it remains only to prove that $a \times b = \inf\{a, b\}$, as distributivity is guaranteed in the semiring. In fact, by Lemma 1 (2) $a \times b \leqslant a, b$. Assume $c \leqslant a, b$. Then, by Lemma 1 (3), $c = c \times c \leqslant a \times b$ which shows that $a \times b$ is the largest lower bound. □

There are many instances of semirings. We look now at a variety of examples of semirings to get a sense of the different systems of practical and theoretical interest covered by these algebraic structures.

Example 1 *(Arithmetic semirings).* Take for A the set of nonnegative real numbers $\mathbb{R}^+ \cup \{0\}$ with $+$ and $\times$ designating the usual addition and multiplication. This is clearly a semiring with the number 0 as zero element and the number 1 as unit element. The semiring is positive too. The order $\preceq$ is in this case the usual total order between numbers. This semiring is needed for defining *probability potentials* as used in probabilistic networks, e.g. Bayesian networks, etc. (see Section 3). We could also consider the field of reals, integers or natural numbers. For example, ordinary addition and multiplication on the nonnegative integers $\mathbb{N} \cup \{0\}$ yield also a positive semiring.

Example 2 *(Boolean semiring).* Here we take $A = \{0, 1\}$ (with the intention that 0 designates "false" and 1 "true"). Define then operation $+$ as $a + b = \max\{a, b\}$ and $a \times b = \min\{a, b\}$. Operation $+$ represents then the logical "or" (disjunction) and $\times$ the logical "and" (conjunction). This is a semiring with zero element 0 and unit element 1. Further, both $+$ as well as $\times$ are *idempotent* operations. In addition, we have $0 + 1 = 1$. Therefore, A is a c-semiring. In fact, this semiring is used to describe (crisp) constraint systems and the relational algebra (see Section 3).

Example 3 *(Bottleneck algebra).* We may also take max for the $+$ operation and min for the $\times$ operation on the set of real numbers $\mathbb{R}$ augmented with $+\infty$ and $-\infty$. Then $-\infty$ is the 0-element and $+\infty$ the unity. This algebra is a c-semiring and in fact a distributive lattice. It is called *bottleneck algebra* [12].

Example 4 *(Distributive lattices).* We have seen that a c-semiring with $\times$ idempotent is a distributive lattice (Theorem 1). Conversely, every distributive lattice is clearly a semiring with joins for $+$ and meets for $\times$-operations (or inversely). Both operations are idempotent. If the lattice has a bottom element $\bot$ then this is the zero element of the semiring. If it has also a top element $\top$, then this is the unit element. In this case the semiring is a c-semiring. This example generalizes the Boolean semiring above. The bottleneck algebra, Example 3, is also an example of a distributive lattice. But distributive lattices can be more generally used to express qualitative degrees of membership of elements to fuzzy sets. Further, Boolean algebras are distributive lattices. Elements of Boolean algebras can also describe assumptions to be satisfied for membership to certain sets. This will be discussed in Example 12 in Section 3.

Example 5 *((*max / min, +*) semirings).* We consider here A to consist of all nonnegative integers $\mathbb{N} \cup \{0, +\infty\}$. We take min as the $+$ operation: $a + b = \min\{a, b\}$, whereas $\times$ is the usual addition with the convention that $a + \infty = \infty$. Both operations are commutative and associative. The distributive law holds too,

$$a + \min\{b, c\} = \min\{a + b, a + c\}.$$

The operation min is idempotent, ∞ is the zero element, the integer 0 is the unit element, and we have

$$\min\{a, 0\} = 0.$$

This shows that we have again a c-semiring. It is also called the *tropical semiring*. This structure has been used in [60] to define a dynamic theory of graded belief states based on ordinal numbers, see Section 3. It arises also in the context of applying *dynamic programming* to minimizing a sum of functions [34,55], and applies to weighted and partially satisfied constraints [9]. Instead of min for the $+$-operation we can also take max. Further we may take for A also the reals $\mathbb{R}$ or nonnegative reals $\mathbb{R}^+ \cup \{0\}$ with or without $+\infty$ or $-\infty$ adjoined. These (min, +) or (max, +) semirings have many applications in networks, graph optimization, queuing systems and discrete event systems [38]; they can also be used (by taking the logarithms of the probabilities) for computing the most probable complete assignment to a Bayesian network, and hence for finding the most probable explanation [46].

Example 6 *(t-norms).* Triangular norms (*t-norms*) were originally introduced in the context of probabilistic metric spaces [44,52]. They are simply binary operations on the unit interval $A = [0, 1]$ which are commutative and associative, have the number 1 as unit element and are, in addition nondecreasing in both arguments:

(1) $\forall a, b, c \in [0, 1]$ we have $T(a, b) = T(b, a)$ and $T(a, T(b, c)) = T(T(a, b), c)$.
(2) $a \leqslant a'$ and $b \leqslant b'$ imply $T(a, b) \leqslant T(a', b')$.
(3) $\forall a \in [0, 1]$ we have $T(a, 1) = T(1, a) = a$ and $T(a, 0) = T(0, a) = 0$.

We may define the operation $\times$ on the unit interval by a t-norm and $+$ as max. Both operations are commutative and associative. That the distributive law holds can be concluded from the following consideration:

$$T(a, b), T(a, c) \leqslant \max\bigl\{T(a, b), T(a, c)\bigr\},$$

hence

$$T\bigl(a, \max\{b, c\}\bigr) \leqslant \max\bigl\{T(a, b), T(a, c)\bigr\}.$$

But we have also, by the monotonicity of the t-norm,

$$T\bigl(a, \max\{b, c\}\bigr) \geqslant T(a, b), T(a, c),$$

hence

$$T\bigl(a, \max\{b, c\}\bigr) \geqslant \max\bigl\{T(a, b), T(a, c)\bigr\}.$$

This shows that

$$a \times (b + c) = T\big(a, \max\{b, c\}\big) = \max\big\{T(a, b), T(a, c)\big\} = (a \times b) + (a \times c).$$

The operation $+$ is idempotent and has the number 0 as zero element. Further, we have for all $a \in A$, $a + 1 = 1$, so A is a c-semiring.

The following are typical t-norms:

(1) *Minimum t-norm*: $T(a, b) = \min\{a, b\}$.
(2) *Product t-norm*: $T(a, b) = a \cdot b$.
(3) *Lukasziewicz t-norm*: $T(a, b) = \max\{a + b - 1, 0\}$.
(4) *Drastic product*: $T(a, 1) = T(1, a) = a$ whereas $T(a, b) = 0$ in all other cases.

In the first case the t-norm is idempotent. So the c-semiring induces a complete, distributive lattice. This is not the case for the other examples. We shall see later (Section 5) that different t-norms distinguish themselves also in other important aspects.

We note that distributivity depends only on the monotonicity of the t-norm, but not on 1 being the unit element. We may more generally require that any other element $e \in [0, 1]$ is the unit element. Then we obtain a *uninorm* [63] and we still have a semiring, albeit no more necessarily a c-semiring. Further, instead of max for the $+$ operation we may take any other commutative, associative and nondecreasing binary operation, i.e. any uninorm. If its unit element is the number 0, then the uninorm is called a *t-conorm*. Then the semiring has 1 as its zero element. We refer to [18,33] for more information on uninorms and t-norms. The concepts of t-norms and t-conorms are important in possibility theory and fuzzy set theory.

Example 7 *(Multidimensional semiring).* Let A be a semiring with operations $+$ and $\times$. We define in A^n operations $+$ and $\times$ as follows

$$(a_1, \dots, a_n) + (b_1, \dots, b_n) = (a_1 + b_1, \dots, a_n + b_n),$$
$$(a_1, \dots, a_n) \times (b_1, \dots, b_n) = (a_1 \times b_1, \dots, a_n \times b_n).$$

Clearly, these operations inherit associativity, commutativity and distributivity from the operations in A. So A^n becomes itself a semiring. Also if $+$ in A is idempotent, then so is $+$ in A^n. The same is true for the operation $\times$. If A has a zero element 0, then $(0, \dots, 0)$ is the zero element in A^n. If 1 is a unit element in A, then $(1, \dots, 1)$ becomes the unit element in A^n. Thus if A is a c-semiring, then so is A^n.

3. Valuation algebra induced by a semiring

The examples in the last section show the richness of semirings. In this section we describe how an algebraic structure, called valuation algebra, is induced by semiring-valued mappings. A wide variety of important reasoning problems can be expressed in terms of such a valuation algebra, and they can be solved by local computations based on a join tree decomposition. Frameworks similar to semiring-induced valuation algebras have been described in [1] (which also describes a number of important applications of the techniques), and in [13,14]; the framework of [32] can also be viewed in this way.

3.1. $\mathcal{A}$-valuations

Consider variables $X, Y, \dots$. For each variable X let Ω_X be the finite set of possible values of X called *frame* of X. We assume that at least one frame Ω_X contains at least two elements. Sets of variables are designated by lower-case letters like $x, y, \dots, r, s, t, \dots$. These sets are also always finite. For a set s of variables let Ω_s denote the Cartesian product of the frames Ω_X for the variables in s,

$$\Omega_s = \prod_{X \in s} \Omega_X.$$

The elements of Ω_s are called *tuples* or *configurations* with domain s.[3] We use lower-case, bold-face letters such as $\mathbf{x}, \mathbf{y}, \ldots$ to denote tuples. It is convenient to include the case where s is empty. We adopt the convention that the frame for the empty set of variables consists of a single tuple, denoted by $\diamond$, such that $\Omega_\emptyset = \{\diamond\}$. If $\mathbf{x}$ is a tuple with domain s and $t \subseteq s$, then $\mathbf{x}^{\downarrow t}$ denotes the projection of $\mathbf{x}$ to the subdomain t. In particular, we have $\mathbf{x}^{\downarrow \emptyset} = \diamond$. Sometimes, in order to emphasize the decomposition of a tuple $\mathbf{x}$ into components belonging to two disjoint subsets t and $s - t$ of s, we write $\mathbf{x} = (\mathbf{x}^{\downarrow t}, \mathbf{x}^{\downarrow s-t})$.

Consider a set A with operations $+$ and $\times$, where $+$ is assumed to be commutative and associative, and write $\mathcal{A}$ as the triple $\langle A, +, \times\rangle$. An $\mathcal{A}$-valuation ϕ with *domain s* associates a value in A with each configuration $\mathbf{x} \in \Omega_s$: ϕ is a function from Ω_s to A. We denote the set of all valuations with domain s by Φ_s. Consider a non-empty set of variables r and let then

$$\Phi = \bigcup_{s \subseteq r} \Phi_s$$

be the set of all $\mathcal{A}$-valuations. D denotes the lattice of subsets of the set of variables r, i.e. $D = \mathcal{P}(r)$ (the powerset of r). For any valuation $\phi \in \Phi$ we define the labeling function $d : \Phi \to D$ where $d(\phi)$ denotes the domain of the valuation ϕ (i.e. $d(\phi) = s$, if $\phi \in \Phi_s$).

We use now the operations $+$ and $\times$ in A to define two operations in the pair (Φ, D):

(1) *Combination:* $\otimes : \Phi \times \Phi \to \Phi$ defined for $\mathbf{x} \in \Omega_{d(\phi)\cup d(\psi)}$ by

$$\phi \otimes \psi(\mathbf{x}) = \phi(\mathbf{x}^{\downarrow d(\phi)}) \times \psi(\mathbf{x}^{\downarrow d(\psi)}).$$

(2) *Projection:* $\downarrow : \Phi \times D \to \Phi$ defined for all $\phi \in \Phi$ and $t \subseteq d(\phi)$ for $\mathbf{x} \in \Omega_t$ by

$$\phi^{\downarrow t}(\mathbf{x}) = \sum_{\mathbf{z} \in \Omega_{d(\phi)}:\ \mathbf{z}^{\downarrow t} = \mathbf{x}} \phi(\mathbf{z}).$$

The defining equation for projection can also be written in the following way, if we decompose the tuples $\mathbf{z}$ of domain $s = d(\phi)$ into subtuples $\mathbf{x}$ belonging to domain t and subtuples $\mathbf{y}$ belonging to domain $s - t$, $\mathbf{z} = (\mathbf{x}, \mathbf{y})$,

$$\phi^{\downarrow t}(\mathbf{x}) = \sum_{\mathbf{y} \in \Omega_{s-t}} \phi(\mathbf{x}, \mathbf{y}).$$

Note that

$$\phi^{\downarrow d(\phi)} = \phi.$$

We remark that projection is also sometimes called marginalization (motivated by applications to probability theory). Further projection could have been defined for arbitrary sets t simply by putting

$$\phi^{\downarrow t} := \phi^{\downarrow t \cap d(\phi)}.$$

Finally projection can also be used to define the operation of *variable elimination* for any variable X,

$$\phi^{-X} := \phi^{\downarrow d(\phi) - \{X\}}.$$

The following axioms have been shown to be sufficient to perform local computation based on a join tree decomposition of valuations [59]. The present form of the axioms for a system (Φ, D) has been introduced in [34,37]. In [51] it has been shown that these axioms are also sufficient for local computation based on a covering join tree of a factorization of valuations.

[3] Note that the terminology of valuation algebras differs from that used in the constraint satisfaction and constraint optimization literature, and also the relational database literature. The frame of a variable in valuation algebra terminology corresponds to the domain of a variable in the constraints literature or in the relational database literature; if, for example, a valuation is a constraint (as in Example 10), then the domain of the valuation is the scope of the constraint; if the valuation is a relation then the domain of the valuation is the relation type. A tuple or configuration is an assignment to a set of variables.

(1) *Semigroup.* Φ is associative and commutative under combination $\otimes$.
(2) *Labeling.* $\forall \phi, \psi \in \Phi$ we have that $d(\phi \otimes \psi) = d(\phi) \cup d(\psi)$.
(3) *Marginalization.* $\forall \phi \in \Phi$ and $s \subseteq d(\phi)$ we have $d(\phi^{\downarrow s}) = s$.
(4) *Transitivity.* $\forall \phi \in \Phi$ and $t \subseteq s \subseteq d(\phi)$ we have

$$(\phi^{\downarrow s})^{\downarrow t} = \phi^{\downarrow t}.$$

(5) *Combination.* $\forall \phi, \psi \in \Phi$ and $d(\phi) \subseteq s \subseteq d(\phi) \cup d(\psi)$ we have

$$(\phi \otimes \psi)^{\downarrow s} = \phi \otimes \psi^{\downarrow s \cap d(\psi)}.$$

A system (Φ, D) satisfying these axioms is called a *valuation algebra*.

The following theorem is a basic result connecting the properties of systems of $\mathcal{A}$-valuations with the properties of $\mathcal{A}$. It implies that a system of $\mathcal{A}$-valuations forms a valuation algebra if $\mathcal{A}$ is a semiring. Conversely if a system of $\mathcal{A}$-valuations forms a valuation algebra, and $\mathcal{A}$ has an additive identity element 0, then $\mathcal{A}$ must be a semiring. This theorem also implies the following converse: if $\mathcal{A}$ is such that any system of $\mathcal{A}$-valuations forms a valuation algebra then $\mathcal{A}$ is a semiring. Almost all of the standard examples of valuation algebras can be expressed as $\mathcal{A}$-valuations, an exception being Dempster–Shafer belief functions. In particular, each formalism covered by the framework in [32] is a system of $\mathcal{A}$-valuations for a semiring $\mathcal{A}$ (see their Definitions 3.1 and 3.3 and Theorem 4.4).

Part (2) and the first part of (3) generalize Theorems 18 and 19 (respectively) of [8], and corresponding standard results for probability potentials.

Theorem 2. *Consider a system of $\mathcal{A}$-valuations (Φ, D), with combination and projection, as defined above.*

(1) *Φ is a commutative semigroup if, and only if, $\mathcal{A}$ operation $\times$ is commutative and associative.*
(2) *Projection is transitive.*
(3) *If $\mathcal{A}$ operation $\times$ distributes over $+$ then the combination property* (5) *holds. Conversely, if the combination property holds, and there exists an additive identity* 0 *in $\mathcal{A}$ then $\times$ distributes over $+$. Also, if the combination property holds, and there exists a variable whose frame contains exactly two elements, then $\times$ distributes over $+$.*

Proof. (1) The commutativity of the combination follows directly from the commutativity of the $\times$ operation in the semiring A and the definition of combination. As for the associativity we have, assuming that ϕ, ψ and η are valuations with domains s, t and u

$$\begin{aligned}
&\big(\phi \otimes (\psi \otimes \eta)\big)(\mathbf{x}) \\
&\quad = \phi(\mathbf{x}^{\downarrow s}) \times (\psi \otimes \eta)(\mathbf{x}^{\downarrow t \cup u}) \\
&\quad = \phi(\mathbf{x}^{\downarrow s}) \times \big(\psi\big((\mathbf{x}^{\downarrow t \cup u})^{\downarrow t}\big) \times \eta\big((\mathbf{x}^{\downarrow t \cup u})^{\downarrow u}\big)\big) \\
&\quad = \phi(\mathbf{x}^{\downarrow s}) \times \psi(\mathbf{x}^{\downarrow t}) \times \eta(\mathbf{x}^{\downarrow u}).
\end{aligned}$$

The same result we obtain in exactly the same way for $((\phi \otimes \psi) \otimes \eta)(\mathbf{x})$. Thus associativity holds.

Conversely, assume that Φ is a commutative semigroup, and let $a, b, c \in A$. Consider valuations $\phi(\mathbf{x}) = a$, $\psi(\mathbf{x}) = b$ and $\eta(\mathbf{x}) = c$ for all configurations $\mathbf{x}$ of s. Then commutativity of $\otimes$ implies commutativity of $\times$: $a \times b = \phi(\mathbf{x}) \times \psi(\mathbf{x}) = \phi \otimes \psi(\mathbf{x}) = \psi \otimes \phi(\mathbf{x}) = \psi(\mathbf{x}) \times \phi(\mathbf{x}) = b \times a$. Associativity of $\times$ follows similarly from associativity of $\otimes$.

(2) Transitivity of projection means simply that we can sum out variables in two steps. That is, if $t \subseteq s \subseteq d(\phi) = u$, then, for all $\mathbf{x} \in \Omega_t$,

$$\begin{aligned}
(\phi^{\downarrow s})^{\downarrow t}(\mathbf{x}) &= \sum_{\mathbf{y} \in \Omega_{s-t}} \phi^{\downarrow s}(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{y} \in \Omega_{s-t}} \sum_{\mathbf{z} \in \Omega_{u-s}} \phi(\mathbf{x}, \mathbf{y}, \mathbf{z}) \\
&= \sum_{(\mathbf{y}, \mathbf{z}) \in \Omega_{u-t}} \phi(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \phi^{\downarrow t}(\mathbf{x}).
\end{aligned}$$

(3) The combination property also follows easily if $\times$ distributes over $+$. Suppose that ϕ has domain t and ψ domain u and $\mathbf{x} \in \Omega_s$, where $t \subseteq s \subseteq t \cup u$, so that $(t \cup s) - s = u - s$. Then we have for $\mathbf{x} \in \Omega_s$

$$\begin{aligned}(\phi\otimes\psi)^{\downarrow s}(\mathbf{x}) &= \sum_{\mathbf{y}\in\Omega_{t\cup u-s}}(\phi\otimes\psi)(\mathbf{x},\mathbf{y}) = \sum_{\mathbf{y}\in\Omega_{u-s}}\big(\phi(\mathbf{x}^{\downarrow t})\times\psi(\mathbf{x}^{\downarrow s\cap u},\mathbf{y})\big)\\ &= \phi(\mathbf{x}^{\downarrow t})\times\sum_{\mathbf{y}\in\Omega_{u-s}}\psi(\mathbf{x}^{\downarrow s\cap u},\mathbf{y}) = \phi(\mathbf{x}^{\downarrow t})\times\psi^{\downarrow s\cap u}(\mathbf{x}^{\downarrow s\cap u})\\ &= (\phi\otimes\psi^{\downarrow s\cap u})(\mathbf{x}).\end{aligned} \tag{3.1}$$

Conversely, assume that the combination property holds in (Φ, D) and consider any triple of values a, b, c in $\mathcal{A}$. Let Y be any variable whose frame contains at least two elements y_1 and y_2. Define valuation ψ with $d(\psi) = \{Y\}$ by $\psi(y_1) = b$, $\psi(y_2) = c$ and $\psi(y) = 0$ for all other $y \in \Omega_Y$. Define ϕ by $d(\phi) = \emptyset$ and $\phi(\diamond) = a$.

$$(\phi\otimes\psi)^{\downarrow s}(\diamond) = \sum_{y\in\Omega_Y}\big(\phi(\diamond)\times\psi(y)\big) = (a\times b)+(a\times c).$$

On the other hand, the left hand side of this equation equals, by the Combination property in (Φ, D),

$$(\phi\otimes\psi^{\downarrow s\cap u})(\diamond) = \phi(\diamond)\times\sum_{y\in\Omega_Y}\psi(y) = a\times(b+c).$$

This shows that $(a \times b) + (a \times c) = a \times (b + c)$ for any triple a, b, c; hence distributivity holds in $\mathcal{A}$. The same argument also works, even without a zero element in $\mathcal{A}$, if there exists some variable Y whose frame has precisely two elements. □

If A is a semiring, then property (1) of Theorem 2 means that the semigroup axiom is satisfied, property (2) assures the transitivity axiom and property (3) the combination axiom. The labeling axiom and the marginalization axiom are satisfied by definition of combination and marginalization of $\mathcal{A}$-valuations. This shows that the system of $\mathcal{A}$-valuations (Φ, D) is a valuation algebra, if A is a semiring.

This implies that local computation is possible with $\mathcal{A}$-valuations for solution of the following problem:

Definition 1. Projection Problem. Given a set of valuations $\phi_1, \dots, \phi_n$ and a set of domains $s_j \subseteq r$, compute

$$(\phi_1\otimes\cdots\otimes\phi_n)^{\downarrow s_j}, \tag{3.2}$$

for $j = 1, \dots, m$.

The graphical structure that underlies local computation is the *join tree*, i.e. a tree whose nodes i are labeled with a domain $\lambda(i)$ such that, if node k lies on the path from node i to node j, then

$$\lambda(k) \supseteq \lambda(i) \cap \lambda(j).$$

This condition is called *running intersection property*.

If the domains $d(\phi_i)$ of the projection problem form a join tree, and each s_j is a subset of some $d(\phi_i)$, then the projection problem can be solved by a sequence of combinations and projections which take place only on the domains of the join tree nodes, i.e. on the domains $d(\phi_i)$, and never on bigger domains [32,34,37,53,55,59]. (One wants, where possible, to generate a join tree with no large nodes and therefore a tree decomposition with small treewidth [10]; see for example, [2,25] and the survey paper [11] for approaches to this problem.)

A less strong condition for local computation has been worked out in [51] and states that the domains $d(\phi_i)$ must only be covered by some join tree node. If the valuation algebra (Φ, D) has neutral elements, these factors can easily be extended to the corresponding node domains. However, not all $\mathcal{A}$-valuation algebras have neutral elements (see examples below) and even if they exist, the proposed extension of the valuations to the join tree domains is not always efficient. In [51] it is shown that this extension is not necessary: local computation is possible even in valuation algebras without neutral elements. This is due to the fact, that one can adjoin a neutral element $e_\emptyset$, such that

$$\phi \otimes e_\emptyset = \phi$$

for all elements ϕ of the valuation algebra, if such an element does not already exist in the algebra. We describe below this modification of the Shenoy–Shafer propagation scheme.

Consider a projection problem (3.2) and assume that there is a join tree $\mathcal{J} = (V, E)$ whose nodes are labeled by $\lambda(k)$, $k = 1, \ldots, |V|$, such that for each $i = 1, \ldots, n$, there is a k such that $d(\phi_i) \subseteq \lambda(k)$ and also, for each $j = 1, \ldots, m$, there is a h such that $s_j \subseteq \lambda(h)$. Then $\mathcal{J}$ is called a *covering join tree* for the family of projection problems. We define an *assignment mapping* $a : \{1, \ldots, n\} \to \{1, \ldots, |V|\}$ such that $d(\phi_i) \subseteq \lambda(a(i))$ and define $\psi_k = \bigotimes_{i:a(i)=k} \phi_i$. If there is no i assigned to k, then define $\psi_k = e_\emptyset$. We have then

$$\phi = \bigotimes_{i=1}^{n} \phi_i = \bigotimes_{k=1}^{u} \psi_k \tag{3.3}$$

and $d(\psi_k) \subseteq \lambda(k)$, for $k = 1, \ldots, u$.

According to [59] messages $\mu_{k\to j}$ are then computed between neighboring nodes of the join tree. In [51] these messages are defined in a covering join tree as follows, if $ne(k)$ denotes the set of neighbor nodes of k in the join tree:

$$\mu_{k\to j} = \left(\psi_k \otimes \bigotimes_{i\in ne(k), i\neq j} \mu_{i\to k}\right)^{\downarrow \omega_{k\to j}\cap\lambda(j)},$$

where

$$\omega_{k\to j} = d(\psi_k) \cup \bigcup_{i\in ne(k), i\neq j} d(\mu_{i\to k}).$$

These messages can be computed sequentially, starting from the leaves of the join tree. The marginals of the factorization with respect to all nodes of the covering join are then obtained as

$$\phi^{\downarrow\lambda(k)} = \psi_k \otimes \bigotimes_{i\in ne(k)} \mu_{i\to k}. \tag{3.4}$$

Example 8. Fig. 1 illustrates a complete run of the Shenoy–Shafer architecture by presenting domains $\omega_{i\to j}$ and messages $\mu_{i\to j}$ for each step. The factors which are distributed over the nodes of the covering join tree have domains $d(\psi_1) = \{A, B\}$, $d(\psi_2) = \{C\}$, $d(\psi_3) = \{A, B, C\}$ and $d(\psi_4) = \{A, C, D\}$. Note that at the end for each node k combination (3.4) must be executed.

This procedure involves redundant computations, if the nodes in the join tree have more than three neighbors. Therefore Shenoy [58] proposes a variant of the method, where the join tree is transformed into a *binary join tree*, i.e. a join tree whose nodes have at most three neighbors (the join tree in Fig. 1 is a binary join tree). Since all the target domains s_j are subsets of some $\lambda(k)$, finally all marginals of the projection problem are computed by local computations only, i.e. computations involving combinations and marginalizations only on domains $\lambda(k)$ of the join tree.

To complete the discussion of general semiring-induced valuation algebras, we add some remarks on some additional properties, which are of some importance.

If the semiring A which induces the valuation algebra (Φ, D) has a unit element 1, then we have for every domain s a valuation $e_s(\mathbf{x}) = 1$ for all $\mathbf{x} \in \Omega_s$. This is the neutral valuation in the semigroup Φ_s defined by the combination, i.e. for all $\phi \in \Phi_s$ we have $e_s \otimes \phi = \phi \otimes e_s = \phi$. These neutral elements satisfy the following property:

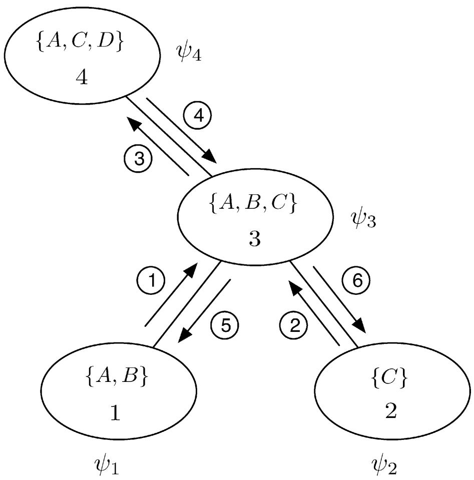
Theorem 3 *(Neutrality). If the semiring A inducing the valuation algebra (Φ, D) has a unit element, then, for all $s, t \subseteq r$, we have*

$$e_s \otimes e_t = e_{s\cup t}.$$

Proof. We have by definition for all $\mathbf{x} \in \Omega_{s\cup t}$

$$(e_s \otimes e_t)(\mathbf{x}) = e_s(\mathbf{x}^{\downarrow s}) \times e_t(\mathbf{x}^{\downarrow t}) = 1 \times 1 = 1. \qquad \square$$

In general it is not true that the projection of the neutral valuation e_s to some subdomain $t \subseteq s$ is still the neutral element e_t. A counter example is provided by probability potentials (see Example 9 below). In this example we have



	$\omega_{i\to j}$	Message content:
1	$\omega_{1\to 3} = d(\psi_1)$	$\mu_{1\to 3} = \psi_1^{\downarrow \omega_{1\to 3}\cap\lambda(3)}$
2	$\omega_{2\to 3} = d(\psi_2)$	$\mu_{2\to 3} = \psi_2^{\downarrow \omega_{2\to 3}\cap\lambda(3)}$
3	$\omega_{3\to 4} = d(\psi_3) \cup d(\mu_{1\to 3}) \cup d(\mu_{2\to 3})$	$\mu_{3\to 4} = (\psi_3 \otimes \mu_{1\to 3} \otimes \mu_{2\to 3})^{\downarrow \omega_{3\to 4}\cap\lambda(4)}$
4	$\omega_{4\to 3} = d(\psi_4)$	$\mu_{4\to 3} = \psi_4^{\downarrow \omega_{4\to 3}\cap\lambda(3)}$
5	$\omega_{3\to 1} = d(\psi_3) \cup d(\mu_{4\to 3}) \cup d(\mu_{2\to 3})$	$\mu_{3\to 1} = (\psi_3 \otimes \mu_{4\to 3} \otimes \mu_{2\to 3})^{\downarrow \omega_{3\to 1}\cap\lambda(1)}$
6	$\omega_{3\to 2} = d(\psi_3) \cup d(\mu_{4\to 3}) \cup d(\mu_{1\to 3})$	$\mu_{3\to 2} = (\psi_3 \otimes \mu_{4\to 3} \otimes \mu_{1\to 3})^{\downarrow \omega_{3\to 2}\cap\lambda(2)}$

Fig. 1. A complete run of the Shenoy–Shafer architecture.

$$e_s^{\downarrow t}(\mathbf{x}) = \sum_{\mathbf{y}\in\Omega_{s-t}} e_s(\mathbf{x},\mathbf{y}) = |\Omega_{s-t}|$$

since $e_s(\mathbf{x},\mathbf{y}) = 1$. In the case of c-semirings however neutral elements project to neutral elements. This important property is called *stability*.

Theorem 4 *(Stability). The valuation algebra (Φ, D), induced by the semiring A with unit is stable, i.e. for all s and $t \subseteq s \subseteq r$ it holds that*

$$e_s^{\downarrow t} = e_t,$$

if the addition operation $+$ in the semiring is idempotent.

Proof. Assume that $+$ is idempotent. Let $\mathbf{x} \in \Omega_t$. Then we obtain

$$e_s^{\downarrow t}(\mathbf{x}) = \sum_{\mathbf{y}\in\Omega_{s-t}} e_s(\mathbf{x},\mathbf{y}) = \sum_{\mathbf{y}\in\Omega_{s-t}} 1 = 1. \tag{3.5}$$

This shows that $e_s^{\downarrow t} = e_t$. □

Stability is important because it permits to extend valuations from a given domain to a superdomain and more generally to transport valuations from domains to other domains [34,53]. This means that in the case of c-semirings, valuations can be regarded as *generalized constraints* which can be extended to larger domains and even to the domain of all variables. More precisely, if ϕ is a valuation with domain s and $s \subseteq t \subseteq r$, then

$$\phi^{\uparrow t} \overset{\text{def}}{=} e_t \otimes \phi$$

is a valuation on domain t by the labeling property (Theorem 2). Using the combination property of Theorem 2 and stability we find that

$$(\phi^{\uparrow t})^{\downarrow s} = (e_t \otimes \phi)^{\downarrow s} = e_t^{\downarrow s} \otimes \phi = e_s \otimes \phi = \phi.$$

So we are entitled to call $\phi_s^{\uparrow t}$ a *vacuous extension* of ϕ, since we do not change its content. We may then more generally transport any valuation ϕ with domain s to any other domain t by

$$\phi^{\rightarrow t} = (\phi^{\uparrow s \cup t})^{\downarrow t}.$$

In this case we may say that two valuations ϕ and ψ with domains s and t represent the same constraint, if

$$\phi^{\rightarrow t} = \psi \quad \text{and} \quad \psi^{\rightarrow s} = \phi.$$

In this sense, ϕ and $\phi^{\uparrow r}$ represent the same constraint. Therefore, we may treat all constraints on the level of the set r of all variables. In particular, it can easily be proved that

$$(\phi_1 \otimes \cdots \otimes \phi_k)^{\uparrow r} = \phi_1^{\uparrow r} \otimes \cdots \otimes \phi_k^{\uparrow r}.$$

So, stability is important if we want to consider valuations as generalized constraints, e.g. soft constraints or fuzzy constraints. We refer to [34] for a discussion of stability and its consequences.

If the semiring A inducing the valuation algebra (Φ, D) has a null element, then this introduces also null (i.e. absorbing) elements with respect to combination in Φ_s. In fact, define for all $\mathbf{x} \in \Omega_s$

$$z_s(\mathbf{x}) = 0.$$

Then, if ϕ is a valuation with domain s, we have

$$(z_s \otimes \phi)(\mathbf{x}) = z_s(\mathbf{x}) \times \phi(\mathbf{x}) = 0.$$

Thus, we see that $z_s \otimes \phi = z_s$ for all $\phi \in \Phi_s$, i.e. z_s is the null (or absorbing) element in Φ_s. Intuitively, we might expect, that a valuation ϕ with domain s, which projects to the null valuation in a domain $t \subseteq s$, must itself be a null element. This is however not automatically the case. For example, if we consider the semiring of real numbers, then we may have

$$0 = \phi^{\downarrow t}(\mathbf{x}) = \sum_{\mathbf{y} \in \Omega_{s-t}} \phi(\mathbf{x}, \mathbf{y})$$

without necessarily $\phi(\mathbf{x}, \mathbf{y}) = 0$ for all tuples. However, this can not happen, if the semiring is *positive*.

Theorem 5 *(Nullity). If the positive semiring A with null element induces the valuation algebra (Φ, D) then, for all s and $t \subseteq s \subseteq r$*

$$\phi^{\downarrow t} = z_t$$

implies $\phi = z_s$.

Proof. Let $\mathbf{x} \in \Omega_t$. Then,

$$0 = \phi^{\downarrow t}(\mathbf{x}) = \sum_{\mathbf{y} \in \Omega_{s-t}} \phi(\mathbf{x}, \mathbf{y})$$

implies by the positivity of the semiring that $\phi(\mathbf{x}, \mathbf{y}) = 0$ for all tuples. □

A null valuation represents generally *contradiction*. If, for example in a c-semiring induced valuation algebra we have $\phi \otimes \psi = z_{s \cup t}$, then the two general constraints are contradictory, they have no "common" (non-zero) configurations. In particular, if the projection problem with respect to the empty set yields the null element

$$(\phi_1 \otimes \cdots \otimes \phi_n)^{\downarrow \emptyset} = z_\emptyset$$

then this means that the set of generalized constraints $\phi_1, \ldots, \phi_n$ is totally contradictory, i.e. not satisfiable. So, null elements play an important role.

3.2. Examples of semiring-induced valuation algebras

Example 9 *(Probability potentials).* If we take for A the semiring of nonnegative real numbers (see Example 1 in Section 2), then the corresponding valuations are called *probability potentials*. They are used in the inference from probability networks, especially Bayesian networks, [41,54]. In fact they represent, up to normalization, discrete probability densities and families of conditional probability densities. The combination is point-wise multiplication which models the computation of multidimensional densities from prior densities and conditional densities, as for example $p(\mathbf{x},\mathbf{y}) = p(\mathbf{x}|\mathbf{y}) \cdot p(\mathbf{y})$. Projection corresponds to the usual marginalization operation in probability theory. Essentially the projection problem consists in computing a marginal of some factorized probability distribution. This is the basic problem for example in Bayesian networks. We refer to [34] for a discussion of the use of these valuations for inference in probabilistic networks. If the usual addition is replaced by the max operator, then the resulting valuation algebra serves to compute *maximum likelihood* or *most probable* values and configurations (see Example 15 below).

Example 10 *(Relational or constraint systems (CSPs)).* If we use the Boolean semiring of Example 2 from Section 2, then a valuation ϕ over domain s defines a *relation* over domain s, i.e. a set of tuples, by

$$R_\phi = \{\mathbf{x} \in \Omega_s\colon\ \phi(\mathbf{x}) = 1\}.$$

This relation can also be considered as a (crisp) constraint on the variables in s. Combination of two valuations ϕ and ψ with domains s and t corresponds then to the natural join of the corresponding relations,

$$R_{\phi\otimes\psi} = R_\phi \bowtie R_\psi = \{\mathbf{x} \in \Omega_{s\cup t}\colon\ \mathbf{x}^{\downarrow s} \in R_\phi, \mathbf{x}^{\downarrow t} \in R_\psi\}.$$

Projection corresponds to the ordinary projection of relations,

$$R_{\phi^{\downarrow t}} = \pi_t(R_\phi) = \{\mathbf{x}^{\downarrow t}\colon\ \mathbf{x} \in R_\phi\}.$$

This gives us a subset of relational algebra, which is useful in query processing for relational databases and for constraint solving. The projection problem formulated in terms of a set of constraints consists in computing the set of tuples in s which can be extended to tuples satisfying all constraints. If the projection is to the empty set, then the problem is to find out whether the constraints have a solution (i.e. $\phi^{\downarrow\emptyset} = 1$) or not (i.e. $\phi^{\downarrow\emptyset} = 0$).

Example 11 *(Propositional logic).* This is a variant of the previous example, where we consider only binary or Boolean variables X. Tuples $\mathbf{x}$ are then Boolean vectors. Valuations $\phi(\mathbf{x}) \in \{0,1\}$ represent then constraints, which may have been defined by formulae of propositional logic. The projection problem

$$(\phi_1 \otimes \cdots \otimes \phi_n)^{\downarrow\emptyset}$$

is then the problem to decide whether the set of propositional formulae defining the valuations $\phi_1,\ldots,\phi_n$ is satisfiable or not. We refer to [36] for a discussion of local computation in propositional logic. Further, Mengin and Wilson [45] study local computation for logic in general.

Example 12 *(Set-based constraints).* We may generalize the example above and replace $\{0,1\}$ by a more general Boolean algebra A. This would then be an instance of a c-semiring which is a distributive lattice (see Example 4 in Section 2). In particular the subsets of a set generate a c-semiring. Let D be a finite set. Define $\mathcal{A} = \langle 2^D, \cap, \cup\rangle$ with null element $\emptyset$ and unit D so that A is the set of subsets of D. The ordering $\preceq$ equals $\subseteq$.

As an example, consider k binary variables a_i, $i = 1,\ldots,k$, each taking values in $\{0,1\}$. The vector $\mathbf{a}$ of all binary variables takes value in $\{0,1\}^k$. Let then $A = \mathcal{P}(\{0,1\}^k)$, the power set of $\{0,1\}^k$. A valuation $\phi(\mathbf{x})$ can then be considered as a statement that $\mathbf{x}$ might be an acceptable tuple, if $\mathbf{a} \in \phi(\mathbf{x})$. The variables a_i are considered as unknown assumptions, which may hold or not. If then, for example, $c \subseteq \Omega_s$ and we define $\phi(\mathbf{x}) = A_c \subseteq A$, if $\mathbf{x} \in c$, and $\phi(\mathbf{x}) = A$ otherwise, then this valuation defines an *assumption-based constraint*: If assumption A_c holds, then $\mathbf{x}$ must belong to constraint c, otherwise $\mathbf{x}$ is free. Combination of two (or more) of such assumption-based constraints ϕ_1 and ϕ_2, corresponding to constraints c_1 and c_2, gives a new assumption-based constraint, where for example $\phi(\mathbf{x}) = A_{c_1} \cap A_{c_2}$ means that $\mathbf{x}$ belongs to $c_1 \cap c_2$, whereas $\phi(\mathbf{x}) = A_{c_1}$ means that $\mathbf{x}$ belongs to c_1 (it may or may not belong to c_2). This is related to *assumption-based reasoning* [19] which may be enriched with a probabilistic structure on the set of assumptions, which then leads to *probabilistic argumentation systems* [27]. Of course, the simple Boolean semiring of the previous example is a special case of this more general example.

Example 13 *(Counting solutions of a CSP).* Here a semiring $\mathcal{A} = \langle \mathbb{N} \cup \{0\}, +, \times \rangle$ with null element 0 and unit 1 is used. As in Example 10 a configuration $\mathbf{x}$ is a solution of a CSP if $\phi(\mathbf{x}) = 1$. But now $\phi^{\downarrow\emptyset}(\diamond) = \sum_{\mathbf{x}\in\Omega_r} \phi(\mathbf{x})$ equals the number of solutions of the CSP. The value of $\phi^{\downarrow\{X\}}(x)$, for example, will give the number of solutions satisfying the assignment $X = x$.

Example 14 *(Possibilistic constraints or fuzzy sets).* If we take the semiring A with a t-norm for multiplication and max for addition (see Example 6 in Section 2), then a valuation $\phi(\mathbf{x})$ is also called a *possibilistic distribution*, or a *possibilistic constraint* or also a *fuzzy set*. The t-norm is used to compute intersections of fuzzy sets or possibilistic constraints. And the max-operator serves to compute projections of fuzzy sets or constraints. More generally, any t-conorm could also be used for projection. This is related to the soft constraints in [49].

Example 15 *(Optimization, weighted CSPs).* Consider the (max / min, +)-semiring of reals (see Example 5 in Section 2). Suppose that we have, say, n valuations over domains $s_1, \dots, s_n$ such that $s_1 \cup \cdots \cup s_n = r$. If we combine them and project the combination to the empty domain, then we have, for $\mathbf{x} \in \Omega$,

$$(\phi_1 \otimes \cdots \otimes \phi_n)^{\downarrow\emptyset}(\diamond) = \max_{\mathbf{x}}\bigl(\phi_1(\mathbf{x}^{\downarrow s_1}) + \cdots + \phi_n(\mathbf{x}^{\downarrow s_n})\bigr).$$

This projection problem is an *optimization problem*, which can be solved by local computation [57]. If multiplication is ordinary multiplication, instead of $+$, then the projection problem corresponds to the maximization of probability in a Bayesian network.

These examples illustrate a number of quite different systems, which all represent different problems. As explained above, the projection problem in all these instances can be solved by the same generic local computation procedure. However there are structural differences between these examples which can in some cases be exploited to design alternative, and often more efficient local computation architectures. This will be discussed in the following Sections 4 and 5.

4. Idempotent valuation algebras

This section considers the special case of idempotent valuation algebras. We derive sufficient conditions for a semiring to induce an idempotent valuation algebra which allows computation to be performed using an especially efficient computational architecture.

4.1. Idempotency

A valuation algebra (Φ, D) is called *idempotent*, if the following additional property holds: For all $\phi \in \Phi$ and $t \subseteq d(\phi)$ we have

$$\phi \otimes \phi^{\downarrow t} = \phi.$$

This is a property we would like to have, if a valuation ϕ is to be interpreted as a piece of information in a strict sense. An information combined with a piece of itself should give nothing new. The existence of neutral elements e_s for all domains s, representing vacuous information relative to a domain s, is usually also required. Further focusing, i.e. projection, of vacuous information should yield a vacuous information; hence *stability* should also hold. So, formally, an *information algebra* is a valuation algebra

(1) with neutral elements satisfying neutrality (Theorem 3),
(2) satisfying stability (Theorem 4),
(3) satisfying idempotency.

Important examples of information algebras are relational or ordinary constraint systems and systems related to logic (propositional, predicate logic and others).

The idempotency property must hold in particular for $t = d(\phi) = s$ so that we must have for all tuples $\mathbf{x} \in \Omega_s$

$$\phi(\mathbf{x}) = (\phi \otimes \phi)(\mathbf{x}) = \phi(\mathbf{x}) \times \phi(\mathbf{x}).$$

This implies that the $\times$ operation of the semiring must be *idempotent*. As we have seen for the stability, the idempotency of the $+$ operation, is a sufficient condition. Thus, a sufficient condition for semiring $\mathcal{A}$ to induce an idempotent, stable valuation algebra is that A is a c-semiring with idempotent multiplication $\times$. By Theorem 1 the semiring $\mathcal{A}$ is then a distributive lattice, $+$ is the supremum and $\times$ the infimum.

The question is, whether $\mathcal{A}$ being a distributive lattice is also sufficient for the induced valuation algebra to be idempotent. The answer is affirmative. In fact, we see that for all $\mathbf{x} \in \Omega_s$, since $\times$ corresponds to the infimum,

$$(\phi \otimes \phi^{\downarrow t})(\mathbf{x}) = \phi(\mathbf{x}) \times \phi^{\downarrow t}(\mathbf{x}^{\downarrow t}) \leqslant \phi(\mathbf{x}).$$

On the other hand, we find that for $\mathbf{x} \in \Omega_t$ and $\mathbf{y} \in \Omega_{s-t}$,

$$\begin{aligned}(\phi \otimes \phi^{\downarrow t})(\mathbf{x}, \mathbf{y}) &= \phi(\mathbf{x}, \mathbf{y}) \times \sum_{\mathbf{y}' \in \Omega_{s-t}} \phi(\mathbf{x}, \mathbf{y}') \\ &\geqslant \phi(\mathbf{x}, \mathbf{y}) \times \phi(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}, \mathbf{y})\end{aligned}$$

since $\phi(\mathbf{x}, \mathbf{y})$ is a term of the sum. We see that indeed for all $\mathbf{x} \in \Omega_s$

$$(\phi \otimes \phi^{\downarrow t})(\mathbf{x}) = \phi(\mathbf{x}).$$

Thus idempotency holds. If the lattice has a top element $\top$, then clearly neutral elements $e_s(\mathbf{x}) = \top$ exists for all sets of variables s and they satisfy neutrality (Theorem 3) and stability (Theorem 4).

We have proven the following theorem.

Theorem 6. *The valuation algebra induced by a semiring A is an information algebra* (*i.e. a stable and idempotent valuation algebra*) *if the semiring A is a distributive lattice with a top element.*

Information algebras (Φ, D) have many interesting properties, for a detailed account see [34]. Idempotency allows for example to define a partial order in Φ, similar as in A, by

$$\phi \geqslant \psi, \quad \text{if } \phi \otimes \psi = \psi.$$

This is a partial order.[4] Clearly $\phi \geqslant \psi$ implies $t = d(\phi) \subseteq d(\psi) = s$ and for all $\mathbf{x} \in \Omega_s$, we have

$$\phi(\mathbf{x}^{\downarrow t}) \times \psi(\mathbf{x}) = \psi(\mathbf{x}),$$

hence $\phi(\mathbf{x}^{\downarrow t}) \geqslant_A \psi(\mathbf{x})$. So the information order is induced by the order in the underlying lattice $\mathcal{A}$.

There are many examples of information algebras, induced by semirings. They include relational systems or CSPs, propositional logic, valuation algebras induced by distributive lattices, etc.

4.2. Local computation in information algebras

Most important however, from a computational point of view, is that idempotency allows to simplify considerably the architectures for the solution of the projection problem. The point is, that division becomes trivial in idempotent valuation algebras (see Section 5), such that architectures for local computation as proposed for Bayesian networks [29,41] can be considerably simplified [34]. Here the corresponding local computation architecture is only sketched; for proofs we refer to [34,51].

Consider a projection problem (3.2) and assume that there is join tree $\mathcal{J}$ whose nodes are labeled by $\lambda(k)$, $k = 1, \ldots, u$. Let then ψ_k be the valuations associated to the nodes k such that (see (3.3))

[4] In [34], the order is defined the other way round, $\phi \leqslant \psi$, meaning that ϕ is less informative than ψ if its combination with ψ does not change the latter. More interesting, however, is a variant of this order, which applies to domain-free valuations, see [34].

$$\phi = \bigotimes_{i=1}^{n} \phi_i = \bigotimes_{k=1}^{u} \psi_k$$

and $d(\psi_k) \subseteq \lambda(k)$, for $k = 1, \ldots, u$.

The marginals $\phi^{\downarrow\lambda(k)}$ may then be computed for all $k = 1, \ldots, u$ by a local computation scheme as follows: An arbitrary node of the join tree, say u, is selected as root node. All edges are directed towards the root and the nodes are numbered such that $j < i$ if node i is on the path from node j to the root node u. The neighbor of node k towards the root node will be denoted by $ch(k)$. For each node i of the join tree we store two associated items denoted by χ_i and ω_i. To initiate the algorithm we assign for $k = 1, \ldots, u$,

$$\chi_k =: \psi_k, \qquad \omega_k =: d(\psi_k).$$

Then, in a first phase, for $k = 1, \ldots, u$, repeat the following steps: Send message

$$\mu_{k\to ch(k)} = \chi_k^{\downarrow\omega_k \cap \lambda(ch(k))}$$

to its neighbor $ch(k)$. In the node $ch(k)$ combine the incoming message with its storage content

$$\chi_{ch(k)} =: \chi_{ch(k)} \otimes \mu_{k\to ch(k)},$$

and update also

$$\omega_{ch(k)} =: \omega_{ch(k)} \cup (\omega_k \cap \lambda(ch(k))).$$

This is also called *collect algorithm*.

In a second phase repeat, for $k = u - 1, \ldots, 1$, the following steps: Send message

$$\mu_{ch(k)\to k} = \chi_{ch(k)}^{\downarrow\lambda(ch(k))\cap\lambda(k)}$$

from the child $ch(k)$ of k back to k. In the receiving node k combine the incoming message with the stored valuation

$$\chi_k =: \chi_k \otimes \mu_{ch(k)\to k}.$$

The second phase is also called *distribute algorithm*. For the proof of the following theorem we refer to [34,51]:

Theorem 7. *In the second phase, the stored valuation χ_k at node k when sending messages to its outward neighbors is equal to the marginal $\phi^{\downarrow\lambda(k)}$.*

Example 16. Reconsider the join tree in Fig. 1 previously used in Example 8 and let $\phi = \psi_1 \otimes \psi_2 \otimes \psi_3 \otimes \psi_4$. For the idempotent architecture we determine:

	Message content:	χ
1	$\mu_{1\to3} = \chi_1^{\downarrow\omega_{1\to3}\cap\lambda(3)} = \psi_1^{\downarrow\omega_{1\to3}\cap\lambda(3)}$	$\chi_3 =: \chi_3 \otimes \mu_{1\to3} = \psi_3 \otimes \mu_{1\to3}$
2	$\mu_{2\to3} = \chi_2^{\downarrow\omega_{2\to3}\cap\lambda(3)} = \psi_2^{\downarrow\omega_{2\to3}\cap\lambda(3)}$	$\chi_3 =: \chi_3 \otimes \mu_{2\to3}$
3	$\mu_{3\to4} = \chi_3^{\downarrow\omega_{3\to4}\cap\lambda(4)}$	$\chi_4 =: \chi_4 \otimes \mu_{3\to4} = \psi_4 \otimes \mu_{3\to4} = \phi^{\downarrow\lambda(4)}$
4	$\mu_{4\to3} = \chi_4^{\downarrow\lambda(4)\cap\lambda(3)} = \phi^{\downarrow\lambda(4)\cap\lambda(3)}$	$\chi_3 =: \chi_3 \otimes \mu_{4\to3} = \phi^{\downarrow\lambda(3)}$
5	$\mu_{3\to1} = \chi_3^{\downarrow\lambda(3)\cap\lambda(1)} = \phi^{\downarrow\lambda(3)\cap\lambda(1)}$	$\chi_1 =: \chi_1 \otimes \mu_{3\to1} = \psi_1 \otimes \mu_{3\to1} = \phi^{\downarrow\lambda(1)}$
6	$\mu_{3\to2} = \chi_3^{\downarrow\lambda(3)\cap\lambda(2)} = \phi^{\downarrow\lambda(3)\cap\lambda(2)}$	$\chi_2 =: \chi_2 \otimes \mu_{3\to2} = \psi_2 \otimes \mu_{3\to2} = \phi^{\downarrow\lambda(2)}$

Note that the messages of the collect algorithm (steps 1–3) correspond exactly to the messages of the first three steps of the Shenoy–Shafer architecture in Example 8. For the computation of χ in steps 3–6 we apply Theorem 7. It is important to note that in difference to the Shenoy–Shafer architecture, a message cache is not needed, since the computation of the messages during the distribute algorithm does not refer to the messages of the collect algorithm.

Since the target domains s_j, for which that marginal $\phi^{\downarrow s_j}$ are desired, are contained in the domains $\lambda(k)$, it follows that these marginals can all be obtained by local computation in the covering join tree $\mathcal{J}$.

5. Division in valuation algebras

There are efficient architectures for local computation which make use of some concept of division. One advantage of these architectures with respect to the Shenoy-Shafer architecture is that they, like the idempotent architecture, do not become less efficient if the join tree is not binary. These architectures have been developed for probabilistic networks [29,41]. But they can be applied to other systems provided that they share some properties with probabilistic systems. The essential point is that valuations must have some kind of inverses [40]. However, this alone is not sufficient, the inverses must also satisfy some consistency conditions relative to marginalization [34]. In this section we want to examine exactly what conditions the semiring A has to satisfy in order to induce a valuation algebra which allows local computation with one of these architectures. In fact, in [34] *regular* and *separative valuation algebras* were shown to allow architectures with division as proposed in [29,41]. The question is, what kind of semirings induce valuation algebras with an appropriate concept of division.

The problem can be solved by studying how to introduce division[5] into semirings, or more generally, into the commutative semigroup of multiplication of a semiring. This is a well studied problem in semigroup theory. The simplest case is the one of a regular semigroup, which decomposes into a union of disjoint groups [17]. We show in Section 5.1 that corresponding regular semirings induce regular valuation algebras, i.e. valuation algebras which decompose also into a disjoint union of groups. This is identified as a first case where the local computation architectures with division generalized from probability theory work. Two further cases are considered: cancellative algebras in Section 5.2, and separative semirings leading to separative valuation algebras in Section 5.3.

5.1. Regular algebras

A semigroup A with an operation $\times$ is called *regular*, if for all $a \in A$ there is an element $b \in A$ such that

$$a \times b \times a = a.$$

The theory of regular semigroups as semigroups with inverses has been developed in [17] and we summarize the results as far as we need them here. Two elements a and b of A are called *inverses*, if

$$a \times b \times a = a, \quad \text{and} \quad b \times a \times b = b.$$

In a regular semigroup, any element a has a unique inverse, which we denote by a^{-1}. Further, for any element $a \in A$ the element $a \times a^{-1}$ is *idempotent*. These idempotent elements in A play an important role. First, if f_1 and $f_2 \in A$ are idempotent elements, then $f_1 \times f_2$ is idempotent. There is a partial order between idempotent elements of A defined by $f_1 \leqslant_I f_2$ if, and only if $f_1 \times f_2 = f_1$. And $f_1 \times f_2$ is the largest lower bound of f_1 and f_2. Let $a \times A$ denote the set of all elements $a \times b$, $b \in A$. Then, in a regular semigroup A there exists for all elements a a unique idempotent element f such that $a \times A = f \times A$.

The relation $a \equiv b$ if, and only if $a \times A = b \times A$ is an equivalence relation on A, a so-called Green relation. And furthermore, if $a_1 \equiv b_1$ and $a_2 \equiv b_2$, then $a_1 \times a_2 \equiv b_1 \times b_2$, i.e. it is a congruence in the semigroup A. Let $[a]$ denote the congruence class containing a. Then $[a]$ is a *commutative group* with $\times$ as the group operation, a^{-1} the inverse of a and the unique idempotent $f_{[a]}$ in the congruence class $[a]$ as the unit element. Thus, we have that

$$A = \bigcup_{a \in A} [a]$$

is the union of a disjoint family of groups. We remark that we may partially order these groups by defining $[a] \leqslant_I [b]$ if, and only if $f_{[a]} \leqslant_I f_{[b]}$. The group $[a]$ is also called the *support* of a.

If A is a semiring, we call it *regular*, if it is regular as a semigroup under the operation $\times$. We have introduced the notion of a positive semiring in Section 2. For regular semirings we strengthen this notion and call the semiring A *positive* if, and only if, for all $a, b \in A$ we have that $[a] \leqslant_I [a + b]$. Note that in any case, if a regular semiring A has a zero element, then 0 is idempotent and $[0] = \{0\}$. Hence, if A is positive in the new sense, then $a + b = 0$ implies $a = 0$. Therefore, A is positive in the former sense too.

[5] This problem is also considered briefly in [13,14], where they suggest adding the condition that multiplication is invertible; as we show here, much weaker conditions are sufficient.

Here follow a few examples.

Example 17 *(Arithmetic-semirings).* Consider the set of reals $\mathbb{R}$ with ordinary addition and multiplication for $+$ and $\times$ (i.e. the field of reals). This is a regular semiring. In this case $A - \{0\}$ is a group, with 1 as the unit element. $\{0\}$ is itself a one-element group. So we have the decomposition of A into these two groups:

$$A = \{0\} \cup \big(A - \{0\}\big).$$

It holds that $[0] \leqslant_I [a]$ for $a \neq 0$. This holds still, if we restrict A to the nonnegative real numbers $\mathbb{R}^+ \cup \{0\}$. However, the semiring of all reals is not positive. The semiring of nonnegative reals $\mathbb{R} \cup \{0\}$ on the other hand is positive. This is the case of probability potentials. The arithmetic semiring on the integers $\mathbb{N}$ or the nonnegative integers $\mathbb{N} \cup \{0\}$ is not regular.

Example 18 *(t-norms).* Most of the t-norms are not regular. The Lukasziewicz t-norm for example is clearly not regular. The product t-norm however is regular, as the previous example shows. The min-t norm is also regular, because it is idempotent (see Example 20 below).

Example 19 *(Multidimensional semiring of real numbers).* We refer to Example 7 in Section 2. If the semiring A is regular, then the multidimensional semiring A^n is clearly also regular. Let A for example be the semiring of reals with the usual operations of addition and multiplication. The idempotents in the semiring A^n are then the vectors consisting only of components 0 and 1. If we define the support of such a vector f to be the set of variables $supp(f)$, for which the components equal 1, then we have $f_1 \leqslant_I f_2$ if, and only if, $supp(f_1) \subseteq supp(f_2)$. We may identify the support $[a]$ with $supp(f_a)$. The inverse of an element $(a_1, \ldots, a_n)$ is the element $(a_1^{-1}, \ldots, a_n^{-1})$, where, as before, the inverse of 0 is 0. The regular semiring A^n is positive if, and only if, the regular semiring A is positive. We have here with A^n an example of a semiring which decomposes into more than two groups, in fact into 2^n groups.

Example 20 *(Idempotent semirings).* If A is a semiring with an idempotent operation $\times$, then A is trivially regular: Each element is an idempotent, hence an inverse of itself. Thus, each element forms for itself a trivial group. If A is also a c-semiring, then the order $\leqslant_I$ is identical to the order $\leqslant_A$. Then A is also positive and the induced information algebra is regular.

We show now, how a *regular, positive semiring A* induces a regular valuation algebra [34]. A valuation algebra (Φ, D) is called regular [34], if for all $\phi \in \Phi$ and $t \subseteq d(\phi)$ there exists a valuation χ with domain t such that

$$\phi \otimes \phi^{\downarrow t} \otimes \chi = \phi. \tag{5.1}$$

Note that this implies that Φ is regular as the semigroup of combination (provided that $\phi^{\downarrow d(\phi)} = \phi$). However, the definition (5.1) of regularity also involves the projection operation, which is essential, if we want to use architectures of local computation with division [34].

Clearly, a necessary condition for a semiring-induced valuation algebra to be regular, is that the underlying semiring is regular. This is however not sufficient. We claim that a regular, positive semiring induces a regular valuation algebra.

Theorem 8. *Let (Φ, D) be the valuation algebra, induced by a regular, positive semiring A. Then (Φ, D) is regular.*

Proof. Suppose $d(\phi) = s$. Take any $\mathbf{x} \in \Omega_t$. Define

$$\chi(\mathbf{x}) = \big(\phi^{\downarrow t}(\mathbf{x})\big)^{-1}.$$

Then we have for any $\mathbf{x} \in \Omega_s$

$$\begin{aligned}(\phi \otimes \phi^{\downarrow t} \otimes \chi)(\mathbf{x}) &= \phi(\mathbf{x}) \times \phi^{\downarrow t}(\mathbf{x}^{\downarrow t}) \times \chi(\mathbf{x}^{\downarrow t}) \\ &= \phi(\mathbf{x}) \times \phi^{\downarrow t}(\mathbf{x}^{\downarrow t}) \times \big(\phi^{\downarrow t}(\mathbf{x}^{\downarrow t})\big)^{-1} \\ &= \phi(\mathbf{x}) \times f_{\phi^{\downarrow t}}.\end{aligned}$$

We use the abbreviations f_ϕ and $f_{\phi^{\downarrow t}}$ for $f_{[\phi(\mathbf{x})]}$ and $f_{[\phi^{\downarrow t}(\mathbf{x}^{\downarrow t})]}$. Thanks to the positivity of A we have $[\phi(\mathbf{x})] \leqslant_I [\phi^{\downarrow t}(\mathbf{x}^{\downarrow t})]$, hence for all $\mathbf{x} \in \Omega_s$ we have $f_\phi \leqslant_I f_{\phi^{\downarrow t}}$ and therefore

$$\begin{aligned}\phi(\mathbf{x}) \times f_{\phi^{\downarrow t}} &= (\phi(\mathbf{x}) \times f_\phi) \times f_{\phi^{\downarrow t}} \\ &= \phi(\mathbf{x}) \times (f_\phi \times f_{\phi^{\downarrow t}}) \\ &= \phi(\mathbf{x}) \times f_\phi \\ &= \phi(\mathbf{x}).\end{aligned}$$

This shows that (5.1) holds. □

The examples of positive regular semirings presented above induce thus regular valuation algebras. They include probability potentials and possibility potentials with multiplication as the t-norm. For regular valuation algebras we can use the *Lauritzen–Spiegelhalter architecture* (LS-architecture) [41] as well as the *HUGIN-architecture* [29] (see Section 5.4). However regularity is not necessary for the applicability of these architectures, there are less restrictive properties which allow for these architectures.

5.2. Cancellative algebras

There are important examples, where A is not regular. For example, consider the $(\max, +)$ semiring on the non-negative integers (see Example 5 in Section 2); this is not regular, since there is no nonnegative integer b such that $a + b + a = a$. However, the negative integer $b = -a$ would serve as a solution for this equation. The arithmetic semirings on integers are not regular either, but again, inverses for all integers exist, as rational numbers. In these examples, the commutative multiplicative semigroup of the semiring must be embedded into larger groups.

The first example can be generalized as follows: A semigroup A is called *cancellative*, if

$$a \times b = a \times c$$

always implies $b = c$ [15]. Such a semigroup can be embedded into a group G in the following way: We consider pairs (a, b) with $a, b \in A$ and define

$$(a, b) = (c, d) \quad \text{if } a \times d = b \times c.$$

Multiplication between such pairs is then defined by

$$(a, b) \times (c, d) = (a \times c, b \times d).$$

This is well defined and multiplication is clearly commutative and associative. The unit e of multiplication is given by pairs (a, a). Then we have

$$(a, b) \times (b, a) = (a \times b, a \times b) = e.$$

So (a, b) and (b, a) are inverses and the set G of pairs (a, b) is a group. The semigroup A is embedded into G by the mapping $a \mapsto (a \times a, a)$. If A itself has a unit element 1, then $1 \mapsto (1, 1) = e$. In the following we consider A as a subset of the group G.

We call a semiring *cancellative*, if the semigroup of A under the operation $\times$ is cancellative.

Example 21 *(Tropical semirings).* If multiplication is defined by addition as in the tropical $(\max / \min, +)$ semirings on nonnegative integers $\mathbf{N}^+ \cup \{0\}$, then the semiring is cancellative, since $a + b = a + c$ always implies $b = c$. This holds also for nonnegative reals $\mathbb{R}^+ \cup \{0\}$. To a pair of numbers a, b we assign the difference $a - b$, which is no more necessarily in the semiring. Clearly, the additive semigroup is embedded into the group G of all integers. The $(\max, +)$-semiring on all integers or reals is already itself a group under addition.

Example 22 *(Positive arithmetic semirings).* The semiring of (strictly) positive integers or reals with the ordinary addition and multiplication is cancellative. In the case of reals the multiplicative semigroup is already itself a group and we have $A = G$. This is because A is not only cancellative, but also regular. Note however that the semirings on the nonnegative integers and real numbers are no more cancellative.

A valuation algebra (Φ, D) is called *cancellative* if, for all $s \in D$, the semigroup Φ_s is cancellative. When the valuation algebra is induced by a cancellative semiring, then, if for all $\mathbf{x} \in \Omega_s$, $\phi(\mathbf{x}) \times \psi(\mathbf{x}) = \phi(\mathbf{x}) \times \eta(\mathbf{x})$ it follows for all $\mathbf{x} \in \Omega_s$, $\psi(\mathbf{x}) = \eta(\mathbf{x})$, so that the valuation algebra (Φ, D) is cancellative. The converse can be shown by considering valuations with empty domain. Hence, if (Φ, D) is induced by a semiring, then it is cancellative if, and only if, the semiring is cancellative.

In this case Φ_s is embedded into a group G_s and this is in fact the group of valuations $\phi : \Omega_s \to G$. The inverse of ϕ is defined by

$$\phi^{-1}(\mathbf{x}) = \big(\phi(\mathbf{x})\big)^{-1}$$

for all $\mathbf{x} \in \Omega_s$. The unit element of group G_s is defined by

$$e_s(\mathbf{x}) = e \in G$$

for all $\mathbf{x} \in \Omega_s$. If e belongs to A, then $e = 1$ is the unit of the semiring A. So we see that Φ, as a semigroup, is embedded into the disjoint union of groups

$$\bigcup_{s \in D} G_s.$$

If A has a unit element, then e_s belongs to Φ, otherwise it is outside Φ. Also the inverses ϕ^{-1} in general do not belong to Φ. We note that

$$e_s \otimes e_t(\mathbf{x}) = e_s(\mathbf{x}^{\downarrow s}) \times e_t(\mathbf{x}^{\downarrow t}) = e \times e = e = e_{s \cup t}(\mathbf{x}).$$

This implies that for any ϕ with $d(\phi) = s$ and $t \subseteq s$ we have

$$\phi \otimes e_t = \phi \otimes e_s \otimes e_t = \phi \otimes e_s = \phi.$$

This condition, together with the existence of inverse valuations outside Φ is sufficient to permit the use of the LS- and the HUGIN architectures for local computation in a valuation algebra induced by a cancellative semiring, see Section 5.4 [34,51].

5.3. *Separative semirings*

Above we noted that the arithmetic semiring on nonnegative integers $\mathbb{N} \cup \{0\}$ is neither cancellative, nor regular. Yet it is possible to embed it into a union of disjoint groups, i.e. the group $\{0\}$ and the multiplicative group of the positive rational numbers. This indicates that there are more general cases of commutative semigroups which can be embedded into a union of disjoint groups [15,61]. The corresponding semirings may under some additional conditions generate valuation algebras which still allow the use of the architectures with division.

In fact, it is known from semigroup theory [15,28,61] that a commutative semigroup can be embedded into a semigroup which is a union of disjoint groups if, and only if, it is *separative*. This means (expressed by the $\times$-operation of a semiring $\mathcal{A}$), that for all $a, b \in A$,

$$a \times b = a \times a = b \times b$$

implies $a = b$. Now, if (Φ, D) is the valuation algebra, induced by the semiring $\mathcal{A}$, then its semigroup is separative if, and only if, the semiring is separative. So this semigroup can then also be embedded into a semigroup which is the union of disjoint groups. But this is not sufficient for the application of local computation architectures with division [34]. We need an additional condition, which links separativity to marginalization (or to the $+$-operation in the underlying semiring). The reason is that in local computation with division, inverses are used to divide marginals $\phi^{\downarrow t}$ of a valuation ϕ out of it at some time and later the marginal is again multiplied into it. So, essentially, the combination of a marginal with its inverse gives a neutral element f of some group, which must also be neutral with respect to ϕ, though ϕ is not in the same group in general,

$$\phi \otimes (\phi^{\downarrow t})^{-1} \otimes \phi^{\downarrow t} = \phi \otimes f = \phi.$$

This is what the additional condition must guarantee.

In this section we develop the corresponding theory, which covers the two preceding structures (Sections 5.1 and 5.2) as special cases. So let $\{G_\alpha\colon \alpha \in Y\}$ be a family of *disjoint groups*, whose union

$$G = \bigcup_{\alpha \in Y} G_\alpha$$

is a semigroup and assume that the multiplicative semigroup of the semiring $\mathcal{A} = \langle A, +, \times\rangle$ is embedded into it. This means, that there is an injective mapping $h : A \to G$ such that $h(a \times b) = h(a) \times h(b)$, where on the left $\times$ is the multiplication in A and on the right the semigroup operation in G. This is the situation we may assume if the multiplicative semigroup of A is separative. For clarity, we identify each element a of A with its image $h(a)$ in G, i.e. we assume without loss of generality that $A \subseteq G$.

There is a unique *unit element* f_α in each group G_α. This is an idempotent element, $f_\alpha \times f_\alpha = f_\alpha$. Let f be an idempotent element in G. Then f belongs to some group G_α and $f \times f = f \times f_\alpha$ which implies $f = f_\alpha$. So the unit elements of the groups G_α are the only idempotent elements in G. Thus, if the semiring possesses a unit element, then it will be the unit element of some group. Now, $f_\alpha \times f_\beta$ is also an idempotent element, hence $f_\alpha \times f_\beta = f_\gamma$ for some $\gamma \in Y$. We define $\alpha \leqslant \beta$ if

$$f_\alpha \times f_\beta = f_\alpha.$$

This relation is clearly reflexive, antisymmetric and transitive, i.e. it is a *partial order* between the elements of Y. Now, if $f_\alpha \times f_\beta = f_\gamma$, then it follows that $\gamma \leqslant \alpha, \beta$. Let $\delta \in Y$ be any other lower bound of α and β, i.e., such that $f_\alpha \times f_\delta = f_\delta$ and $f_\beta \times f_\delta = f_\delta$. Then, $f_\gamma \times f_\delta = f_\alpha \times f_\beta \times f_\delta = f_\alpha \times f_\delta = f_\delta$. So $\delta \leqslant \gamma$, hence γ is the greatest lower bound of α and β, so we write $\gamma = \alpha \wedge \beta$. We have thus

$$f_\alpha \times f_\beta = f_{\alpha\wedge\beta}.$$

The family Y of groups forms therefore a *semilattice*, i.e. a partially ordered set where the infimum exists between any pair of elements.

We denote the inverse element of an element a in some group G_α by a^{-1}. Then $a \times a^{-1} = f_\alpha$. Suppose b in some group G_β. Then $(a \times b) \times (a^{-1} \times b^{-1}) = f_\alpha \times f_\beta = f_{\alpha\wedge\beta}$. Therefore $(a \times b)^{-1} = a^{-1} \times b^{-1}$. Suppose now that $a \times b \in G_\gamma$. Then $(a \times b)^{-1} \in G_\gamma$ and $(a \times b) \times (a \times b)^{-1} = f_\gamma$. But as we have seen $f_\gamma = f_{\alpha\wedge\beta}$, hence $\gamma = \alpha \wedge \beta$ and $a \times b \in G_{\alpha\wedge\beta}$.

We define $a \equiv b$ in A if a and b belong to the same group G_α. This is an equivalence relation in A. Assume that $a \equiv a'$ and $b \equiv b'$. Then $a \times b \equiv a' \times b'$ and the relation is a $\times$-congruence in A. This implies that the equivalence classes $[a]$ of this equivalence relation in A are semigroups. Thus A decomposes into a family of disjoint semigroups,

$$A = \bigcup_{a \in A} [a].$$

The partial order of Y carries over to equivalence classes $[a]$. In fact, we have $[a] \leqslant [b]$ if and only if $[a \times b] = [a]$ and, for all $a, b \in A$ also $[a \times b] = [a] \wedge [b]$. Thus, the semigroups $[a]$ form a semilattice, isomorph to Y. We call the equivalence class $[a]$ of a the *support* of a.

Reflexivity $a \equiv a$ implies that $a \times a \equiv a$. So, a and $a \times a$ have the same support, i.e. $[a] = [a \times a]$. We introduce now an *additional* requirement, which generalizes this relation, and which links the decomposition of the multiplicative semigroup of a semiring to the $+$-operation of the semiring. We call a semiring $\mathcal{A} = \langle A, +, \times\rangle$ *separative*, if its multiplicative semigroup is separative and, in addition, there is an embedding into a union of groups, such that for all $a, b \in A$,

$$[a] \leqslant [a + b]. \tag{5.2}$$

This is a kind of strengthening of positivity. In fact, if $\mathcal{A}$ has a zero element, then (5.2) implies that $[0] \leqslant [a]$ for all elements a of A. Also, if $a \preceq b$ (see Section 2) then from condition (5.2) we conclude that $[a] \leqslant [b]$.

Let's illustrate these results by some examples. In particular, it must be stressed that the embedding of a semigroup into an union of disjoint groups is not necessarily unique, as the second example shows.

Example 23 *(Regular and cancellative semirings).* A cancellative semiring (Section 5.2) is clearly separative, since cancellativity implies that from $a \times a = a \times b$ it follows that $a = b$. Condition (5.2) is trivially satisfied, since there is only one support. A regular positive semiring (Section 5.1) is also separative, since regularity implies that any element $a \in A$ has an inverse in A and hence from $a \times a = a \times b = b \times b$ it follows that a and b are in the same group, $[a] = [b]$, of the decomposition of the semiring, and then, multiplying with the inverse of a (or b) it follows that $a = f_{[a]} \times b = b$. Condition (5.2) is required for regular semirings too (see Section 5.1). So regular and cancellative semirings are particular cases of separative semirings. In the first case the multiplicative semiring decomposes not only into a semilattice of semigroups, but into a semilattice of groups, in the second case the semigroup is embedded into a group.

Example 24 *(Arithmetic semirings).* Some of the arithmetic semirings (see Example 1) are regular or even cancellative. But consider the arithmetic semiring on nonnegative integers $\mathbb{N} \cup \{0\}$. It is neither cancellative nor regular. But it is separative. It decomposes into the semiring $\{0\}$ and the arithmetic semiring of natural numbers $\mathbb{N}$. The first is already a (trivial) group, the second is embedded into the multiplicative group of positive rational numbers. And their union, the nonnegative rational numbers form a multiplicative semiring too. The partial order between the two groups is $\{0\} \leqslant \mathbb{N}$. Condition (5.2) holds too. So this arithmetic semiring is separative.

There is an alternative embedding of the multiplicative semigroup. Consider finite sets of prime numbers. For any such set of prime numbers, the natural numbers which factor exactly into those prime numbers form a semigroup which can be embedded into a group. The partial order between these semigroups is defined by set inclusion. However, with this decomposition condition (5.2) is not satisfied. For example $2 + 3 = 5$, but $[2] \nleqslant [5]$ since $\{2\}$ is not a subset of $\{5\}$.

Example 25 *(Nonnegative semirings).* In many cases a semiring $\mathcal{A} = \langle A, +, \times \rangle$ with zero element decomposes into two multiplicative semirings $\{0\}$ and $A - \{0\}$. If the latter is cancellative, then the semiring is separative. It is then embedded into the semiring which is the union of the group $\{0\}$ and the group G into which $A - \{0\}$ is embedded. In fact $\{0\} \cup G$ is a semigroup, since we may define $0 \times g = 0$ for all $g \in G$. The partial order between groups is $\{0\} \leqslant A - \{0\}$. Further, since $0 + b = b$ for all $b \in A$ condition (5.2) is clearly satisfied. The previous example belongs to this class of separative semirings. But we may in an arithmetic semiring for example replace addition by the max operator and then it remains a separative semiring.

Example 26 *(Multidimensional semirings).* Consider a multidimensional semiring (Example 7) $\langle A^n, +, \times \rangle$ whose component semiring $\mathcal{A} = \langle A, +, \times \rangle$ is separative. Clearly the multidimension multiplicative semigroup is separative too. If A is embedded into a union of disjoint groups G_α, then the multiplicative semigroup of A^n is embedded into the union of the disjoint Cartesian product of groups G_α

$$G_{\alpha_1,\dots,\alpha_n} = G_{\alpha_1} \times \cdots \times G_{\alpha_n}.$$

The idempotent elements of these groups are $f_{\alpha_1,\dots,\alpha_n} = (f_{\alpha_1}, \dots, f_{\alpha_n})$. It follows that $(\alpha_1, \dots, \alpha_n) \leqslant (\beta_1, \dots, \beta_n)$ if, and only if, $\alpha_i \leqslant \beta_i$ for all $i = 1, \dots, n$. In the same way $[(a_1, \dots, a_n)] \leqslant [(b_1, \dots, b_n)]$ if $[a_i] \leqslant [b_i]$ for all i. It follows immediately that condition (5.2) is satisfied in the multidimensional semiring, if it is in $\mathcal{A}$.

If a semiring A is cancellative then it has no zero element (unless $A = \{0\}$). (If the semiring has a zero element 0 then let $a = b = 0$, and let c be an arbitrary element of the semiring. The above cancellative property for semigroups implies that $c = 0$.) In particular, c-semirings are not cancellative. It is therefore natural to consider a weaker cancellation property, see [7]: let us say that A is weakly cancellative if for any $a, b, c \in A$, if $a \neq 0$ and $a \times b = a \times c$ then $b = c$. This property implies that if $a \times b = 0$ then either $a = 0$ or $b = 0$. If a semiring is weakly cancellative then it is separative. For suppose $a \times a = a \times b = b \times b$; if $a = 0$ then $b \times b = 0$ and so $b = 0 = a$. Otherwise, if $a \neq 0$ then $a \times a = a \times b$ which implies $a = b$.

A separative semiring satisfies: if $a \neq 0$ then $a \times a \neq 0$. Consider $A = \{0, 1, 2, \dots, k\}$, (for some $k \geqslant 2$) with the semiring addition operation being minimum, and the semiring multiplication being integer addition, truncated to keep the result at most k. The value k is the zero element of the semiring and the value 0 is the unit element. This is an important semiring for reasoning with weighted constraints [39], where the k arises from the weight of the best solution found so far. It is not separative: consider, for example, $a = k$ and $b = k - 1$.

More generally, consider a totally ordered c-semiring (which corresponds to a valuation structure, used for valued CSPs [9]), where there exists an idempotent element a and a non-idempotent element b with $b > a$ and such that there is no element between a and b. Then $b > b \times b \geqslant a \times a = a$ and so $a \times a = a \times b = b \times b$, implying that such a semiring is not separative. Because of this, many fair valuation structures [16], when viewed as totally ordered semirings, are not separative.

The use of kinds of division in semirings for soft constraints has been studied in [7,16].

Let now $\mathcal{A} = \langle A, +, \times \rangle$ be a separative semiring and (Φ, D) a valuation algebra induced by this semiring. Then the combination semigroup of Φ is also separative, i.e.

$$\phi \otimes \psi = \phi \otimes \phi = \psi \otimes \psi$$

implies $\phi = \psi$. So, this semigroup can also be embedded into a semigroup which is a union of disjoint groups. In fact, the decomposition which is for our purposes of interest is the particular one induced by the decomposition of the underlying semiring $\mathcal{A}$.

The decomposition of A induces a congruence in the combination semigroup of Φ as follows:

$$\phi \equiv \psi$$

if

(1) $d(\phi) = d(\psi)$,
(2) for all $\mathbf{x} \in \Omega_{d(\phi)}$, $\phi(\mathbf{x}) \equiv \psi(\mathbf{x})$.

This is clearly an equivalence relation on Φ. Assume that $\phi \equiv \psi$ and $\phi \equiv \eta$ and $d(\phi) = d(\psi) = d(\eta) = s$. Then it follows that $d(\phi \otimes \psi) = d(\phi \otimes \eta) = s$ and for all $\mathbf{x} \in \Omega_s$ also that $\phi(\mathbf{x}) \times \psi(\mathbf{x}) \equiv \phi(\mathbf{x}) \times \eta(\mathbf{x})$. So this equivalence is also a combination congruence in Φ. It follows then that the equivalence classes $[\phi]$ are subsemigroups of the combination semigroup of Φ.

For any valuation ϕ with $d(\phi) = s$ define the mapping $sp_{[\phi]} : \Omega_s \to Y$, where Y is the semilattice of the group decomposition of the separative semiring $\mathcal{A}$, by

$$sp_{[\phi]}(\mathbf{x}) = \alpha, \quad \text{if } \phi(\mathbf{x}) \in G_\alpha.$$

Note that this mapping is well defined, since $sp_{[\phi]} = sp_{[\psi]}$, if $[\phi] = [\psi]$. We define for a valuation ϕ with $d(\phi) = s$

$$G_{[\phi]} = \bigl\{ g : \Omega_s \to G\colon \forall \mathbf{x} \in \Omega_s\, g(\mathbf{x}) \in G_{sp_{[\phi]}(\mathbf{x})} \bigr\}.$$

It follows that $G_{[\phi]}$ is a group, and the semigroup $[\phi]$ is embedded in it. The unit element $f_{[\phi]}$ of group $G_{[\phi]}$ is given by $f_{[\phi]}(\mathbf{x}) = f_{sp_{[\phi]}(\mathbf{x})}$. The inverse of ϕ is defined by $\phi^{-1}(\mathbf{x}) = (\phi(\mathbf{x}))^{-1}$. This induces also the partial order $[\phi] \leqslant [\psi]$ if $f_{[\phi]}(\mathbf{x}) \leqslant f_{[\psi]}(\mathbf{x})$ for all $\mathbf{x} \in \Omega_s$ or $[\phi \otimes \psi] = [\phi]$. In fact, this is a semilattice, i.e. $f_{[\phi \otimes \psi]} = f_{[\phi]} \wedge f_{[\psi]}$. The union of these groups

$$G^* = \bigcup_{\phi \in \Phi} G_{[\phi]}$$

is a semigroup. In fact, if $g_1 \in G_{[\phi]}$ and $g_2 \in G_{[\psi]}$, then $g_1 \otimes g_2$ is defined for $\mathbf{x} \in \Omega_{s \cup t}$, if $d(\phi) = s$ and $d(\psi) = t$ by

$$g_1 \otimes g_2(\mathbf{x}) = g_1(\mathbf{x}^{\downarrow s}) \times g_2(\mathbf{x}^{\downarrow t})$$

and belongs to $G_{[\phi \otimes \psi]}$.

We have the equivalence $\phi \otimes \phi \equiv \phi$ because $[\phi]$ is a semigroup. But, due to the separativity of the underlying semiring $\mathcal{A}$ it follows for any $t \subseteq d(\phi)$ and also for all $\mathbf{x} \in \Omega_s$,

$$\bigl[\phi(\mathbf{x})\bigr] \leqslant \bigl[\phi^{\downarrow t}(\mathbf{x}^{\downarrow t})\bigr].$$

This means that $[\phi] \leqslant [\phi^{\downarrow t}]$ or also

$$\phi^{\downarrow t} \otimes \phi \equiv \phi. \tag{5.3}$$

This condition guarantees that

$$\phi \otimes (\phi^{\downarrow t})^{-1} \otimes \phi^{\downarrow t} = \phi \otimes f_{[\phi^{\downarrow t}]} = \phi \tag{5.4}$$

because, for any neutral element $f_{[\psi]}$ such that $[\phi] \leqslant [\psi]$ we have that $f_{[\psi]} \otimes \phi = \phi$ (see [34]).

This in turn is, what is needed for the local computation architectures with division to be applicable [34]. A valuation algebra (Φ, D) which has a combination congruence which satisfies this condition and such that the equivalence classes $[\phi]$ are cancellative semigroups are called *separative valuation algebras* in [34].

Let's illustrate these results by the important example of nonnegative semirings.

Example 27 *(Valuation algebra induced by a nonnegative semiring).* According to Example 25 a nonnegative semiring $\mathcal{A} = \langle A, +, \times\rangle$ is embedded into a union of groups $\{0\} \cup G$, where group G contains the positive part $A - \{0\}$ of the semiring $\mathcal{A}$. If (Φ, D) is the valuation algebra induced by such a semiring, then we define the support $supp(\phi)$ of a valuation ϕ with domain $d(\phi) = s$ as

$$supp(\phi) = \{\mathbf{x} \in \Omega_s\colon \phi(\mathbf{x}) \neq 0\}.$$

Then, in this particular case, the congruence $\phi \equiv \psi$ holds exactly if the two valuations have the same support, i.e. $supp(\phi) = supp(\psi)$ and the equivalence class $[\phi]$ contains all valuations with the same domain as ϕ and the same support. In the case of arithmetic nonnegative semirings it becomes clear, that in such an equivalence class we can define the inverse of a valuation ϕ as

$$\phi^{-1}(\mathbf{x}) = \frac{1}{\phi^{-1}(\mathbf{x})}, \quad \text{if } \mathbf{x} \in supp(\phi),$$

and $\phi^{-1}(\mathbf{x}) = 0$ otherwise. This defines the group $G_{[\phi]}$. The partial order between classes $[\phi]$ or groups $G_{[\phi]}$ is defined by inclusion of supports: $[\phi] \leqslant [\psi]$ if $supp(\phi) \subseteq supp(\psi)$.

5.4. Local computation with division

As claimed above, for all the semiring-induced valuation algebras with division as defined by the division in the underlying separative semiring, the local computation architectures proposed for probability potentials such as the LS- and the HUGIN-architectures can be applied.

In the LS-architecture, first the collect algorithm is executed as in the architecture for idempotent valuation algebras, except that in node i, the node content χ_i is divided by the outgoing message to $ch(i)$. So, in node i we store

$$\chi_i =: \chi_i \otimes \left(\chi_i^{\downarrow s_i \cap s_{ch(i)}}\right)^{-1}.$$

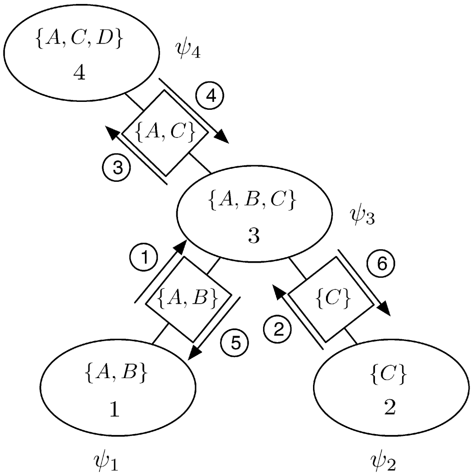
After the collect algorithm, a distribute algorithm follows exactly as in the idempotent architecture. For a proof of the correctness of this architecture in valuation algebras with division as described in the previous sections we refer to [34]. The idempotent architecture for local computation (Section 4.2) is a special case of the LS-architecture, since in information algebras each element is its own inverse.

The HUGIN architecture is a variant of the LS-algorithm in which between all nodes i and $ch(i)$ of the join tree an additional node, the so-called *separator* is introduced. The collect algorithm is as originally, except that the message $\mu_{i\to ch(i)}$ is stored in the separator. After the collect algorithm a distribute phase follows, where each node i, starting with the root node m sends messages out as in the idempotent architecture. However the message is sent to the separator nodes, where it is divided by the inverse of the content of the separator,

$$\mu_{ch(i)\to i} \otimes (\mu_{i\to ch(i)})^{-1}.$$

This message arrives then at node i, where it is combined with the node content χ_i. The difference with LS-architecture is that division occurs on the smaller domain $s_i \cap s_{ch(i)}$ of the separator, instead of on the domain s_i. This is an advantage. In the distribute phase, if a node k is ready to send a message, it stores $\chi_k = \phi^{\downarrow\lambda(k)}$. We refer again to [34,51] for a proof of correctness of this architecture for regular and separative valuation algebras.

Example 28. Reconsider the join tree in Fig. 1 previously used in Examples 8 and 16. Fig. 2 illustrates a complete run of the HUGIN architecture with $\phi = \psi_1 \otimes \psi_2 \otimes \psi_3 \otimes \psi_4$. The collect algorithm corresponds again to the first three steps of the Shenoy–Shafer architecture, but every message is stored in a separator node represented as diamond in the figure. Steps 3–6 use the correctness of the algorithm in the computation of χ.



For the collect algorithm the messages are:

	Message to separator:	Message from separator:	χ
1	$\chi_1^{\downarrow\omega_{1\to3}\cap\lambda(3)} = \psi_1^{\downarrow\omega_{1\to3}\cap\lambda(3)}$	$\mu_{1\to3} = \psi_1^{\downarrow\omega_{1\to3}\cap\lambda(3)}$	$\chi_3 =: \chi_3 \otimes \mu_{1\to3} = \psi_3 \otimes \mu_{1\to3}$
2	$\chi_2^{\downarrow\omega_{2\to3}\cap\lambda(3)} = \psi_2^{\downarrow\omega_{2\to3}\cap\lambda(3)}$	$\mu_{2\to3} = \psi_2^{\downarrow\omega_{2\to3}\cap\lambda(3)}$	$\chi_3 =: \chi_3 \otimes \mu_{2\to3}$
3	$\chi_3^{\downarrow\omega_{3\to4}\cap\lambda(4)}$	$\mu_{3\to4} = \chi_3^{\downarrow\omega_{3\to4}\cap\lambda(4)}$	$\chi_4 =: \chi_4 \otimes \mu_{3\to4} = \psi_4 \otimes \mu_{3\to4} = \phi^{\downarrow\lambda(4)}$

In the distribute phase, we have then:

	Message to separator:	Message from separator:	χ
4	$\chi_4^{\downarrow\lambda(4)\cap\lambda(3)} = \phi^{\downarrow\lambda(4)\cap\lambda(3)}$	$\mu_{4\to3} = \phi^{\downarrow\lambda(4)\cap\lambda(3)} \otimes (\mu_{3\to4})^{-1}$	$\chi_3 =: \chi_3 \otimes \mu_{4\to3} = \phi^{\downarrow\lambda(3)}$
5	$\chi_3^{\downarrow\lambda(3)\cap\lambda(1)} = \phi^{\downarrow\lambda(3)\cap\lambda(1)}$	$\mu_{3\to1} = \phi^{\downarrow\lambda(3)\cap\lambda(1)} \otimes (\mu_{1\to3})^{-1}$	$\chi_1 =: \chi_1 \otimes \mu_{3\to1} = \phi^{\downarrow\lambda(1)}$
6	$\chi_3^{\downarrow\lambda(3)\cap\lambda(2)} = \phi^{\downarrow\lambda(3)\cap\lambda(2)}$	$\mu_{3\to2} = \phi^{\downarrow\lambda(3)\cap\lambda(2)} \otimes (\mu_{2\to3})^{-1}$	$\chi_2 =: \chi_2 \otimes \mu_{3\to2} = \phi^{\downarrow\lambda(2)}$

Fig. 2. A complete run of the HUGIN architecture.

It should be stressed that in both architectures the inverses used in the computation may be elements outside the valuation algebra, whereas the final results in all nodes of the join tree belong to the algebra. Moreover, the division allows to define concepts like "conditional valuations" generalizing conditional probabilities. Then even in the original factorization of an element of the valuation algebra, the factors need not necessarily be elements of the algebra and yet the local computation architectures return the correct marginals. This permits a generalization of Bayesian networks to more general structures than probability potentials. For details we refer to [34].

6. Propagating upper and lower bounds

A problem with the join tree based computational schemes (including fusion and bucket elimination) is that the propagation will tend not to be feasible unless all the sets of variables associated with the nodes in the join tree are small. However, for a given problem, we may well not be able to find such a join tree; in particular, by definition, there exists no such join tree unless the induced width (treewidth) [10,23,32] is small.

In this section we consider how to compute upper and lower bounds of valuations. In particular, we consider how to adapt the general join tree propagation algorithm (see Section 3.1) to efficiently compute bounds for the projection problem; the key to the efficiency is avoiding having to perform the hardest combinations, such as those involving large number of variables. The mini-buckets and mini-clustering techniques of Dechter et al. [20,22,24,30,31,43], have been developed for approximations and bounds of this kind for a number of important problems: belief updating, most probable explanation and combinatorial optimization for weighted constraints. We show how this kind of algorithmic approach can be made much more general, so that it can be applied for general semiring-induced valuation algebras[6] and other valuation algebras.

Upper bounds are important for an optimization problem, when using a branch-and-bound algorithm. For example, consider the problem of finding a maximum assignment to a collection M of $\mathcal{A}$-valuations where $\mathcal{A}$ is a $(\max, +)$ semiring (Example 5) i.e., with the semiring addition as max and the semiring multiplication being $+$. (The application of the mini-clustering approach for (almost) this semiring, and within a branch-and-bound framework, has been described in [22]). We generate a search tree where each node of the tree is associated with (i) an assignment $\mathbf{z} \in \Omega_t$ to some set t of variables, and (ii) a multiset M' of valuations, which is M with variables t instantiated to $\mathbf{z}$. $(\bigotimes M)^{\downarrow\emptyset}$ is the value of the maximum assignment, and $(\bigotimes M')^{\downarrow\emptyset}$ is the value of the maximum assignment which extends $\mathbf{z}$. Let b be the value of the best solution found so far; suppose we have an efficient algorithm which generates an upper bound a for $(\bigotimes M')^{\downarrow\emptyset}$. If $b \not\preceq a$ then we know that $\mathbf{z}$ cannot be extended to an assignment with value better than b, and so we can backtrack at this node (given that we are looking for a single maximal assignment). This generalizes to other semirings with idempotent addition, including only partially ordered semirings.

Furthermore, upper and/or lower bounds may be sufficient to answer a particular query. In a problem where the variables are decision variables, an upper bound may be sufficiently low to imply that no decision is adequate. In another situation a lower bound may be sufficiently high to imply that it is possible to make a good choice, so it may be worth investing in more computation time to find such a choice.

A somewhat different approach to approximation is given in [26], which has the advantage of enabling one to keep a careful control on the computation time. This is based on the usual join tree message passing algorithms, but where a combination on a node is approximated to keep the computation of the combination within a set time limit. For probability potentials ([26], Section 5.5.1) this approximate combination can be performed by processing tuples of the product set sequentially, and implicitly assigning zeros to tuples which are not reached before the time limit; (an analogous approach is also suggested for Dempster–Shafer belief potentials). A significant disadvantage of this is that, if the product frame (associated with a node in the join tree) is very large then there are a very large number of tuples to process; there will typically then be time to process only a tiny fraction of these, which will tend to lead to a very poor approximation of the individual combination, and also of the overall result. However, it would be interesting to explore the potential for combining the resource-bounding ideas in [26] with the generalized mini-clustering approach developed here.

In Section 6.1 we construct the upper and lower bounds framework for the general case of valuation algebras, and give the associated propagation algorithm in Section 6.2. The join tree propagation algorithms involve repeated application of combination followed by projection i.e., computations of the form $(\bigotimes M)^{\downarrow u}$. Mini-buckets/clustering algorithms and our extended algorithms approximate such marginalized combinations; they produce a multiset M' of valuations whose combination is an approximation of the message $(\bigotimes M)^{\downarrow u}$. (An important feature of these algorithms is that we do not need then to combine together the valuations M': instead these will form the inputs for approximations of further messages.) In Section 6.3 we consider the case of semiring-induced valuations. Section 6.4 discusses the same kind of algorithm for other types of approximation. Another important consideration for the efficiency of the propagation algorithm is the number of non-zero elements of the input valuations, since having few non-zero elements makes a combination much faster. In Section 6.5, it is shown how one can use a pre-processing step of constraint propagation to potentially decrease the number of non-zero tuples in the input valuations. This idea is taken a step further in Section 6.6, where it is shown how, for certain types of query, one can increase the number of zeros in the input valuations without changing the answer.

6.1. Bounding the projection of a combination

In this section we extend valuation algebras by adding an associated ordering, and we consider the problem of constructing upper and lower bounds of the projection of a combination of valuations. (Our definition of ordered valuation algebra is slightly different to the one given in [26].)

---

[6] Independently, Chang and Mackworth have suggested a special form of this kind of approximation for semiring-induced valuation algebras; see [13] and Section 6.1 of [14]. They also consider other approximation methods, as does Aji and McEliece [1].

Ordered valuation algebras

$(\Phi, D, \preceq)$ is said to be an *ordered valuation algebra (with neutral elements)* if

(i) (Φ, D) is a valuation algebra;
(ii) for each set of variables s, there exists an identity element e_s (so that for each $\phi \in \Phi$, $\phi \otimes e_{d(\phi)} = \phi$) and for any sets of variables s and t, $e_s \otimes e_t = e_{s \cup t}$;
(iii) the relation $\preceq$ is a pre-order on Φ (i.e., a reflexive and transitive relation) which only orders valuations with the same domain, i.e., $\phi \preceq \psi$ implies $d(\phi) = d(\psi)$; furthermore, projection and combination both respect $\preceq$, that is, for arbitrary $\phi, \psi, \chi \in \Phi$, if $\phi \preceq \psi$ then
  (a) $\phi^{\downarrow u} \preceq \psi^{\downarrow u}$ for any $u \subseteq d(\phi) = d(\psi)$; and
  (b) $\phi \otimes \chi \preceq \psi \otimes \chi$.

This last property implies that if $\phi_i \preceq \psi_i$ for all $i = 1, \ldots, k$, then $\phi_1 \otimes \cdots \otimes \phi_k \preceq \psi_1 \otimes \cdots \otimes \psi_k$.

If $\phi \preceq \psi$, we say that ϕ is a lower bound for ψ [with respect to $\preceq$], and that ψ is an upper bound for ϕ.

The propagation algorithms involve sequences of combinations and projections. Because projection and combination both respect $\preceq$, if at any point we replace any valuation by an upper bound of it, the result will be an upper bound of the correct result. Similarly with lower bounds.

We can extend the notion of upper and lower bounds to valuations with smaller domains. Suppose $u = d(\phi) \subseteq d(\psi)$. We say *$\phi$ is a u-lower bound for ψ* if $\phi \otimes e_{d(\psi)-d(\phi)} \preceq \psi$. Similarly, we say that *ϕ is a u-upper bound for ψ* if $\psi \preceq \phi \otimes e_{d(\psi)-d(\phi)}$.

Least upper bounds and greatest lower bounds can be defined in the obvious way: for valuations ϕ and ψ with $d(\phi) = u \subseteq d(\psi)$, we say that ϕ is a least u-upper bound of ψ if (i) ϕ is a u-upper bound of ψ, and (ii) $\phi \preceq \theta$ for any u-upper bound θ of ψ. If $\preceq$ is a partial order then there can be at most one least u-upper bound. We define greatest u-lower bounds analogously.

A valuation ϕ is a lower bound for a valuation ψ if and only if it is a $d(\psi)$-lower bound for ψ (and similarly for upper bounds) since $\phi \otimes e_\emptyset = \phi \otimes e_{d(\phi)} \otimes e_\emptyset = \phi \otimes e_{d(\phi)} = \phi$, so $\phi \preceq \psi \iff \phi \otimes e_\emptyset \preceq \psi$. The properties for neutral elements also imply that for any valuation ϕ and set of variables q, $\phi \otimes e_q = \phi \otimes e_{q-d(\phi)}$. This is because $\phi \otimes e_{d(\phi)\cap q} = \phi \otimes e_{d(\phi)} \otimes e_{d(\phi)\cap q} = \phi \otimes e_{d(\phi)} = \phi$; so $\phi \otimes e_{q-d(\phi)} = \phi \otimes e_{d(\phi)\cap q} \otimes e_{q-d(\phi)} = \phi \otimes e_q$. In particular, this implies that if $u = d(\phi) \subseteq d(\psi)$ then: ϕ is a u-lower bound for ψ if and only if $\phi \otimes e_{d(\psi)} \preceq \psi$; similarly for u-upper bounds.

The fusion algorithm (bucket elimination) and join tree propagation algorithms [21,32,34,56,59] involve repeated application of: combination of a multiset M of valuations followed by projection to a set of variables u, i.e., $(\bigotimes M)^{\downarrow u}$. If M involves too many variables this may be infeasible. The key to mini-buckets and mini-clustering bounding techniques is to generate upper and lower bounds for $(\bigotimes M)^{\downarrow u}$ which involve only feasible combinations. The fundamental result is the following, showing that $(\bigotimes M)^{\downarrow u}$ can be bounded above (and, similarly, below) by the combination of a multiset of valuations derived from M but only involving variables in u.

Proposition 2. *For $i = 0, \ldots, k$, let ϕ_i be a valuation in an ordered valuation algebra $(\Phi, D, \preceq)$, let $s = d(\phi_0) \cup \cdots \cup d(\phi_k)$, the set of variables involved in these valuations, let u be a subset of s, and let $t = s - (u \cup d(\phi_0))$. For each $i = 1, \ldots, k$, let τ_i be a $u \cap d(\phi_i)$-lower bound for ϕ_i, and let θ_i be a $u \cap d(\phi_i)$-upper bound for ϕ_i. Then $\phi_0^{\downarrow u \cap d(\phi_0)} \otimes \tau_1 \otimes \cdots \otimes \tau_k \otimes e_t^{\downarrow \emptyset}$ is a lower bound for $(\phi_0 \otimes \cdots \otimes \phi_k)^{\downarrow u}$ and $\phi_0^{\downarrow u \cap d(\phi_0)} \otimes \theta_1 \otimes \cdots \otimes \theta_k \otimes e_t^{\downarrow \emptyset}$ is an upper bound.*

In Section 6.3, we give general ways of constructing the lower bound functions τ_i and the upper bound functions θ_i for semiring-induced valuation algebras. For example, under appropriate conditions, θ_i can be obtained by projecting ϕ_i (see Lemma 2), generalizing the approximations used for the MPE problem in [24, p. 116], and for discrete optimization; furthermore, defining θ_i using pointwise max (see Section 6.3.2) generalizes the mini-bucket approximation for belief updating in [24, p. 120].

Proof. For each $i = 1, \ldots, k$, by definition, $\tau_i \otimes e_{d(\phi_i)-u} \preceq \phi_i$, so, since projection respects $\preceq$, and by commutativity and associativity of combination,

$$\phi_0 \otimes e_{d(\phi_1)-u} \otimes \cdots \otimes e_{d(\phi_k)-u} \otimes \tau_1 \otimes \cdots \otimes \tau_k$$

is a lower bound for $\phi_0 \otimes \cdots \otimes \phi_k$. By the assumed property of neutral elements, $e_{d(\phi_1)-u} \otimes \cdots \otimes e_{d(\phi_k)-u}$ equals $e_{(d(\phi_1)\cup\cdots\cup d(\phi_k))-u}$, and so, by the property of neutral elements shown above,

$$\phi_0 \otimes e_{d(\phi_1)-u} \otimes \cdots \otimes e_{d(\phi_k)-u} = \phi_0 \otimes e_t,$$

since $t = (d(\phi_1) \cup \cdots \cup d(\phi_k)) - u - d(\phi_0)$. Hence $\phi_0 \otimes e_t \otimes \tau_1 \otimes \cdots \otimes \tau_k$ is a lower bound for $\phi_0 \otimes \cdots \otimes \phi_k$. Since $\preceq$ respects projection, this implies that

$$(\phi_0 \otimes e_t \otimes \tau_1 \otimes \cdots \otimes \tau_k)^{\downarrow u} \preceq (\phi_0 \otimes \cdots \otimes \phi_k)^{\downarrow u}.$$

Write $u_0 = u \cap d(\phi_0)$. Since $t \cap u = \emptyset$, we have $(d(\phi_0) \cup t) \cap u = u_0$. The combination axiom implies that

$$(\phi_0 \otimes e_t \otimes \tau_1 \otimes \cdots \otimes \tau_k)^{\downarrow u} = (\phi_0 \otimes e_t)^{\downarrow u_0} \otimes \tau_1 \otimes \cdots \otimes \tau_k,$$

since $\tau_1, \ldots, \tau_k$ only involve variables in u. Because ϕ_0 and e_t do not involve any common variables, the combination axiom and transitivity axiom imply that $(\phi_0 \otimes e_t)^{\downarrow u_0} = \phi_0^{\downarrow u_0} \otimes e_t^{\downarrow u_0 \cap t}$; in more detail:

$$(\phi_0 \otimes e_t)^{\downarrow u_0} = \left((\phi_0 \otimes e_t)^{\downarrow u_0 \cup t}\right)^{\downarrow u_0} = (\phi_0^{\downarrow u_0} \otimes e_t)^{\downarrow u_0} = \phi_0^{\downarrow u_0} \otimes e_t^{\downarrow u_0 \cap t},$$

which equals $\phi_0^{\downarrow u_0} \otimes e_t^{\downarrow \emptyset}$. Therefore $\phi_0^{\downarrow u \cap d(\phi_0)} \otimes \tau_1 \otimes \cdots \otimes \tau_k \otimes e_t^{\downarrow \emptyset}$ is a lower bound for $(\phi_0 \otimes \cdots \otimes \phi_k)^{\downarrow u}$.

The upper bound result is proved in exactly the same way. □

Consider the situation where each ϕ_i can be written as $\tau_i \otimes e_{d(\phi_i)-u}$, where $d(\tau_i) = d(\phi_i) \cap u$ (in this case ϕ_i really only depends on variables in u). Then a similar argument as that used in the above proof can be used to prove that $\phi_0^{\downarrow u \cap d(\phi_0)} \otimes \tau_1 \otimes \cdots \otimes \tau_k \otimes e_t^{\downarrow \emptyset}$ is actually equal to $(\phi_0 \otimes \cdots \otimes \phi_k)^{\downarrow u}$. This shows that the $e_t^{\downarrow \emptyset}$ terms are in general necessary.

However, in applying Proposition 2, we will often be able to ensure that $d(\phi_0)$ contains all variables being eliminated, i.e., $d(\phi_0) \supseteq s - u$ and hence $t = \emptyset$ and $e_t^{\downarrow \emptyset} = e_\emptyset$. For any ψ, we have $\psi \otimes e_\emptyset = \psi$ so then $\phi_0^{\downarrow u \cap d(\phi_0)} \otimes \tau_1 \otimes \cdots \otimes \tau_k$ is a lower bound for $(\phi_0 \otimes \cdots \otimes \phi_k)^{\downarrow u}$ and $\phi_0^{\downarrow u \cap d(\phi_0)} \otimes \theta_1 \otimes \cdots \otimes \theta_k$ is an upper bound. In particular when applying this to approximate the result of the fusion algorithm (bucket elimination), the set $s - u$ of eliminated variables is always just a singleton $\{X\}$; it can be assumed that there exists some valuation which involves variable X, and so the $e_t^{\downarrow \emptyset}$ terms disappear. Similarly, if the valuation algebra is *stable* then the terms $e_t^{\downarrow \emptyset}$ disappear also, since then $e_t^{\downarrow \emptyset} = e_\emptyset$.

Approximating $(\bigotimes M)^{\downarrow u}$ without performing expensive combinations

Let M be a multiset of valuations, and let u a subset of the variables involved. We will give a procedure that produces a multiset M' of valuations whose combination is a lower bound for $(\bigotimes M)^{\downarrow u}$; similarly, a procedure for generating an upper bound; furthermore we can restrict the combinations used in the procedures to ensure that only feasible combinations of valuations are involved. These procedures form the basis of the propagation algorithm in Section 6.2. An important point is that, except in the final step in the propagation algorithm, the approximating multisets M' will not need to be combined; instead the combination of M' and other multisets will be again approximated.

Proposition 2 already gives a way of approximating a marginalized combination $(\bigotimes M)^{\downarrow u}$ (and it does this without performing any combinations). However, we can typically improve the approximations by combining some of the valuations first, but still only performing feasible combinations.

We assume functions `UB` and `LB`, where for valuation ϕ and set of variables $u \subseteq d(\phi)$, the valuation $\mathtt{UB}(\phi, u)$ is a u-upper bound for ϕ and $\mathtt{LB}(\phi, u)$ is a u-lower bound for ϕ. (Methods of generating `UB` and `LB` for different formalisms are derived in Section 6.3.2.) We also assume function $\mathtt{Partition(Input}: M, B; \mathtt{Output}: M_0, M_1, \ldots, M_k)$ which takes multiset M and nonnegative number B as inputs, and produces multisets $M_0, M_1, \ldots, M_k$ which partition M, and are such that the `size` (see below) of each M_i is at most B. It is assumed that the implementation of the functions `UB`, `LB` and `Partition` do not involve any combinations of valuations.

The `size` of a non-empty multiset M of valuations is intended to be a quickly-evaluated measure of how hard it is to combine together the valuations M (see also the notion of *weight functions* in [48]). The `size` of M is assumed to be a nonnegative real number (though one could generalize it to a partially ordered scale if one wished; this would allow easy generalization of the pair of parameters (i, m) used in the mini-buckets approximations [24]). The only further property we assume of `size` is that if M is a singleton then the `size` of M is 0; this is because then combining

M requires no work. `size` can be defined in various ways. Let M be a multiset of valuations which contains at least two valuations. One definition is to say that `size` of M is $|d(M)| = |\bigcup_{\phi \in M} d(\phi)|$, i.e., the total number of variables involved in M. In this case the parameter B used below in e.g., UpperBound(M, u, B) corresponds to the parameter i used in the mini-buckets approximations such as $mbe - bel - max(i, m)$ [24], page 121. We allow other definitions of `size` because the number of variables is not the only factor in the complexity of a combination. Another natural definition of `size` of M is $|\Omega_{d(M)}|$, the cardinality of the frame associated with the combination of the valuations in M, as this gives an upper bound on the complexity of the combination. Other definitions are possible for semiring valuations, for example, that take into account the number of non-zero values in the valuations (which is also very relevant to the computational efficiency for such valuations).

Let M be a multiset of valuations, let u be a subset of the variables involved in M, and let B be a nonnegative real number. LowerBound(M, u, B), which we define by the algorithm below, is a function that returns a multiset of valuations; we will show that the combination of the returned multiset is a lower bound for $(\bigotimes M)^{\downarrow u}$. Multiset M is partitioned into multisets which are of sufficiently small `size` (no more than B), and each multiset is combined. Lower bounds for the results of these combinations are chosen which involve only variables in u.

Function LowerBound(M, u, B)
begin
  Partition(Input : M, B; Output : $M_0, M_1, \ldots, M_k$).
  For each $i = 0, \ldots, k$, let $\phi_i = \bigotimes_{\phi \in M_i} \phi$.
  Let $t = d(\phi_1) \cup \cdots \cup d(\phi_k) - (u \cup d(\phi_0))$.
  Return multiset $\{e_t^{\downarrow \emptyset}, \phi_0^{\downarrow u \cap d(\phi_0)}, \mathtt{LB}(\phi_1, u \cap d(\phi_1)), \ldots, \mathtt{LB}(\phi_k, u \cap d(\phi_k))\}$.
end

The algorithm for LowerBound(M, u, B) involves performing combinations, but each combination is of a multiset of `size` at most B.

Function UpperBound(M, u, B) is defined in an exactly analogous manner (and in practice the functions LowerBound and UpperBound might be combined):

Function UpperBound(M, u, B)
begin
  Partition(Input : M, B; Output : $M_0, M_1, \ldots, M_k$).
  For each $i = 0, \ldots, k$, let $\phi_i = \bigotimes_{\phi \in M_i} \phi$.
  Let $t = d(\phi_1) \cup \cdots \cup d(\phi_k) - (u \cup d(\phi_0))$.
  Return multiset $\{e_t^{\downarrow \emptyset}, \phi_0^{\downarrow u \cap d(\phi_0)}, \mathtt{UB}(\phi_1, u \cap d(\phi_1)), \ldots, \mathtt{UB}(\phi_k, u \cap d(\phi_k))\}$.
end

These procedures produce correct bounds on the projection of the combination, irrespective of the choice of functions Partition, LB and UB. Furthermore, the computations only require combinations of multisets of `size` at most B.

Proposition 3. *Whatever choices are made for functions* Partition, LB *and* UB, *the valuation* $\bigotimes$ LowerBound(M, u, B) *(the combination of all the elements in the multiset LowerBound(M, u, B)) is a lower bound for $(\bigotimes M)^{\downarrow u}$ and the valuation $\bigotimes$* UpperBound(M, u, B) *is an upper bound for $(\bigotimes M)^{\downarrow u}$. Furthermore, the computations of* LowerBound(M, u, B) *and* UpperBound(M, u, B) *do not involve the combination of any multiset of valuations of* `size` *more than B.*

Proof. By Proposition 2, $\bigotimes$ LowerBound(M, u, B) is a lower bound for $(\phi_0 \otimes \cdots \otimes \phi_k)^{\downarrow u}$, which is equal to $(\bigotimes M)^{\downarrow u}$ since $M_0, \ldots, M_k$ is a partition of M. Similarly, $\bigotimes$ UpperBound(M, u, B) is an upper bound for $(\bigotimes M)^{\downarrow u}$. The last part follows by the definition of Partition. □

Choosing the function Partition. It is always possible to choose a valid function Partition; even in the extreme case of $B = 0$ we can choose each M_i to be a singleton. However, the choice of partition will affect the closeness of the approximations. Ideally we would like, where possible, to choose each M_i so that its combination ϕ_i does not depend very much on variables not in u.

If the upper bound B on `size` is chosen sufficiently large then we can choose `Partition` to always return a single multiset, so $k = 0$ and $M_0 = M$. In this case, the propagation algorithm in Section 6.2 (for both lower and upper bounds) reduces essentially to the exact computation.

Choosing the functions `LB` *and* `UB`. Section 6.3 shows how we can generate these functions for semiring-induced valuation algebras. If possible, we would like to choose $\mathtt{LB}(\phi, u)$ to be a least u-upper bound of ϕ, and $\mathtt{UB}(\phi, u)$ to be a greatest u-lower bound of ϕ. As we will see, these exist in many situations.

6.2. Propagation algorithm for upper and lower bounds

Section 3.1 described a propagation algorithm for the projection problem (Eq. (3.2)) based on the Shenoy–Shafer architecture. In this section we show how this can be modified to generate upper and lower bounds. We are given valuations $\phi_1, \ldots, \phi_n$ and a set of target domains s_l, $l = 1, \ldots, m$, and we wish to compute upper and lower bounds for $(\phi_1 \otimes \cdots \otimes \phi_n)^{\downarrow s_l}$ for $l = 1, \ldots, m$. A value B is chosen globally; the computation ensures that we do not have to compute a combination of a multiset of valuations of `size` more than B. The propagation is based on repeated use of functions `LowerBound` and `UpperBound` described in Section 6.1.

We focus on the general projection problem; but if we just wish to compute bounds for a single set s_1 then we can choose a root of the join tree whose associated variables contain s_1, and we only need send messages towards the root (as in the *collect* algorithm [34]). The mini-buckets algorithm, approximating fusion or bucket elimination, can be considered as a special case of this, where the join tree is generated from a variable elimination sequence.

Let s_l be any of the target domains. We require that s_l is small enough in order for us to be able to perform arbitrary combinations of valuations with domain s_l. The reason for this is that at the very last stage of the computation, we combine valuations with domain s_l (or smaller). Formally, we assume that if $d(\psi) \subseteq s_l$ for all $\psi \in M$, then the `size` of M is at most B. For example, if we use the first suggested definition of `size`, then it is assumed that B is at least as large as the cardinality of any target set of variables s_l.

Let $L(k)$ be the multiset of input valuations associated with node k, so that, with the notation of Section 3.1, $L(k) = \{\phi_i\colon a(i) = k\}$.

As in [22,43], each message for the approximate propagations will be a multiset of valuations rather than a single valuation. We will inductively define message $\mu_{k\to j}$, message $\mu^{\text{lower}}_{k\to j}$ and message $\mu^{\text{upper}}_{k\to j}$ for each pair k and j of neighboring nodes. Formally, we could consider that the induction is on: *the length of a longest path, from k to a leaf node, which doesn't pass through j* (where a path is not allowed to double back on itself).

Assume, by induction, that we have defined messages $\mu_{i\to k}$ and $\mu^{\text{upper}}_{i\to k}$ and $\mu^{\text{lower}}_{i\to k}$ for all neighbors $i \neq j$ of k. (This includes the induction base case as well, i.e., when k is a leaf node, since leaf nodes only have one neighbor.)

Let $M_{k\to j} = L(k) \cup \{\mu_{i\to k}\colon i \in ne(k), i \neq j\}$, consisting of all input valuations associated with node k, and all messages coming into k from directions other than j. Let $u_{k\to j} = \omega_{k\to j} \cap \lambda(j)$ (where $\omega_{k\to j}$ is the set of variables involved in valuations in $M_{k\to j}$, and $\lambda(j)$ is the set of variables associated with node j). As in Section 3.1 we define the message $\mu_{k\to j}$ to be $(\bigotimes M_{k\to j})^{\downarrow u_{k\to j}}$.

Analogously, let

$$M^{\text{lower}}_{k\to j} = L(k) \cup \bigcup_{i\in ne(k), i\neq j} \mu^{\text{lower}}_{i\to k},$$

and let

$$M^{\text{upper}}_{k\to j} = L(k) \cup \bigcup_{i\in ne(k), i\neq j} \mu^{\text{upper}}_{i\to k}.$$

We define $\mu^{\text{lower}}_{k\to j}$ to be $\mathtt{LowerBound}(M^{\text{lower}}_{k\to j}, u_{k\to j}, B)$ and define

$$\mu^{\text{upper}}_{k\to j} = \mathtt{UpperBound}(M^{\text{upper}}_{k\to j}, u_{k\to j}, B).$$

Next we define the multisets of valuations associated finally with each node k. Let $M^k = L(k) \cup \{\mu_{i\to k}\colon i \in ne(k)\}$. Analogously, let $M^k_{\text{lower}} = L(k) \cup \bigcup_{i\in ne(k)} \mu^{\text{lower}}_{i\to k}$ and $M^k_{\text{upper}} = L(k) \cup \bigcup_{i\in ne(k)} \mu^{\text{upper}}_{i\to k}$.

Let s_l be any target set of variables; we choose (using some fixed deterministic method) k to be a node with $\lambda(k) \supseteq s_l$. The valuation ϕ^l_{lower} is defined to be

$$\bigotimes\big(\texttt{LowerBound}(M^k_{\text{lower}}, s_l, B)\big)$$

and ϕ^l_{upper} is defined to be

$$\bigotimes\big(\texttt{UpperBound}(M^k_{\text{upper}}, s_l, B)\big).$$

The results of Schneuwly et al. [51] (see Section 3.1), imply that $(\phi_1 \otimes \cdots \otimes \phi_n)^{\downarrow\lambda(k)} = \bigotimes M^k$, and hence $(\phi_1 \otimes \cdots \otimes \phi_n)^{\downarrow s_l} = (\bigotimes M^k)^{\downarrow s_l}$ by the transitivity [of projection] axiom.

The lower bound computation involves modifying the exact computation by successively replacing valuations by lower bounds (or rather by multisets of valuations whose combination is a lower bound). The exact computation involves sequences of combinations and projections, and combination and projection respect $\preceq$, which leads to the final results being correct bounds:

Theorem 9. *With the above definitions, for all $l = 1, \dots, m$, valuation ϕ^l_{lower} is a lower bound for $(\phi_1 \otimes \cdots \otimes \phi_n)^{\downarrow s_l}$ and ϕ^l_{upper} is an upper bound. Furthermore, the computation of these bounds does not involve combining multisets of valuations of* `size` *more than B.*

Proof. We first prove by induction that, for any neighboring nodes j and k, $\bigotimes \mu^{\text{lower}}_{k\to j}$ is a lower bound for message $\mu_{k\to j}$ and $\bigotimes \mu^{\text{upper}}_{k\to j}$ is an upper bound for $\mu_{k\to j}$.

Assume by induction that for all neighbors $i \neq j$ of k, valuation $\bigotimes \mu^{\text{lower}}_{i\to k}$ is a lower bound for message $\mu_{i\to k}$ and $\bigotimes \mu^{\text{upper}}_{i\to k}$ is an upper bound (this includes the base case of the induction, since leaf nodes only have one neighbor). Then, since combination respects $\preceq$, valuation $\bigotimes M^{\text{lower}}_{k\to j}$ is a lower bound for $\bigotimes M_{k\to j}$, and, because projection respects $\preceq$, valuation $(\bigotimes M^{\text{lower}}_{k\to j})^{\downarrow u_{k\to j}}$ is a lower bound for $\mu_{k\to j} = (\bigotimes M_{k\to j})^{\downarrow u_{k\to j}}$. By Proposition 3, we have $\bigotimes \texttt{LowerBound}(M^{\text{lower}}_{k\to j}, u_{k\to j}, B) \preceq (\bigotimes M^{\text{lower}}_{k\to j})^{\downarrow u_{k\to j}}$, and so, by transitivity of $\preceq$, we have $\bigotimes \mu^{\text{lower}}_{k\to j}$ is a lower bound for $\mu_{k\to j}$. In just the same way, we have that $\bigotimes \mu^{\text{upper}}_{k\to j}$ is an upper bound for $\mu_{k\to j}$. This completes the induction.

Let s_l be any target set of variables, and let k be the chosen node with $\lambda(k) \supseteq s_l$. Because combination and projection respect $\preceq$, the first part implies that $(\bigotimes M^k_{\text{lower}})^{\downarrow s_l} \preceq (\bigotimes M^k)^{\downarrow s_l} \preceq (\bigotimes M^k_{\text{upper}})^{\downarrow s_l}$. Using Proposition 3 we then have that

$$\phi^l_{\text{lower}} = \bigotimes\big(\texttt{LowerBound}(M^k_{\text{lower}}, s_l, B)\big) \preceq (\phi_1 \otimes \cdots \otimes \phi_n)^{\downarrow s_l}$$

and

$$(\phi_1 \otimes \cdots \otimes \phi_n)^{\downarrow s_l} \preceq \bigotimes\big(\texttt{UpperBound}(M^k_{\text{upper}}, s_l, B)\big) = \phi^l_{\text{upper}}.$$

The computations involve repeated use of functions `LowerBound` and `UpperBound`, each of which does not involve the combination of any multiset of valuations of `size` more than B, followed by, for each target set s_l, a final combination of valuations involving variables s_l. By the assumption on s_l and B, the `size` of such combinations is also no more than B. □

6.3. Application for semiring-induced valuation algebras

In this section we discuss how to generate the lower and upper bound functions `LB` and `UB` when the ordered valuation algebra is generated by a semiring with an ordering on it. This enables us to use the bounds propagation algorithms from the previous section for semiring-induced valuation algebras.

6.3.1. Orderings on semirings and on semiring-induced valuation algebras

Often we will wish to use some relation $\preceq$ to order the elements of a semiring, where $a \preceq b$ might indicate, for example, that b is a greater degree of preference than a. Such a relation $\preceq$ on A will always be assumed to be a pre-order, i.e., a reflexive and transitive relation.

We say that relation $\preceq$ satisfies:

$+$ ***is monotone over*** $\preceq$ if for all $a, b, c \in A$, $a \preceq b$ implies $a + c \preceq b + c$.
$\times$ ***is monotone over*** $\preceq$ if for all $a, b, c \in A$, $a \preceq b$ implies $a \times c \preceq b \times c$.

Let $\equiv$ be the equivalence relation corresponding to $\preceq$, so that $a \equiv b$ if and only if $a \preceq b$ and $b \preceq a$. Given that $+$ and $\times$ are monotone over $\preceq$, if it is helpful computationally, we can replace semiring $\mathcal{A}$ by the quotient semiring $\mathcal{A}/\equiv$ consisting of the set of equivalence classes of $\mathcal{A}$, and replace each $\mathcal{A}$-valuation by the corresponding $\mathcal{A}/\equiv$-valuation. This will lead to equivalent upper and lower bounds, since if $a \equiv b$ then $a \preceq c \iff b \preceq c$, and also $c \preceq a \iff c \preceq b$.

In many situations the natural ordering relation is $\preceq_{\mathcal{A}}$ given by $a \preceq_{\mathcal{A}} b$ if and only if there exists $c \in A$ with $a + c = b$. As shown by Proposition 1 in Section 2, operations $+$ and $\times$ are monotone over $\preceq_{\mathcal{A}}$, and $0 \preceq a$ for all $a \in A$.

We can extend any pre-order $\preceq$ on A to a relation on semiring-induced valuations. We define relation $\preceq$ on $\mathcal{A}$-valuations, by $\phi \preceq \psi$ if ϕ and ψ involve the same set of variables s (i.e., $d(\phi) = d(\psi) = s$) and for all $\mathbf{x} \in \Omega_s$, $\phi(\mathbf{x}) \preceq \psi(\mathbf{x})$.

Proposition 4. *Let $\mathcal{A} = \langle A, +, \times \rangle$ be a semiring. Let $\preceq$ be pre-order on A.*

(i) *If $\preceq$ is a partial order then so is the associated relation $\preceq$ on $\mathcal{A}$-valuations.*
(ii) *If $\times$ is monotone over $\preceq$ then combination of $\mathcal{A}$-valuations respects $\preceq$.*
(iii) *If $+$ is monotone over $\preceq$ then projection $\mathcal{A}$-valuations respects $\preceq$.*

Proof. Let ϕ, ψ and χ be $\mathcal{A}$-valuations with $\phi \preceq \psi$. Let $s = d(\phi)$ and so $d(\psi) = s$, and let $t = d(\chi)$. For all $\mathbf{x} \in \Omega_s$, $\phi(\mathbf{x}) \preceq \psi(\mathbf{x})$.
(i) Suppose $\preceq$ on A is a partial order, and that $\psi \preceq \phi$. To prove that relation $\preceq$ on $\mathcal{A}$-valuations is a partial order we just need to show that $\phi = \psi$. For any $\mathbf{x} \in \Omega_s$, we have $\phi(\mathbf{x}) \preceq \psi(\mathbf{x}) \preceq \phi(\mathbf{x})$ and so $\phi(\mathbf{x}) = \psi(\mathbf{x})$, since $\preceq$ on A is a partial order. This implies that $\phi = \psi$.
(ii) For any $\mathbf{x} \in \Omega_{s\cup t}$, $(\phi \otimes \chi)(\mathbf{x}) = \phi(\mathbf{x}^{\downarrow s}) \times \chi(\mathbf{x}^{\downarrow t}) \preceq \psi(\mathbf{x}^{\downarrow s}) \times \chi(\mathbf{x}^{\downarrow t})$ (since $\times$ is monotone over $\preceq$) which equals $(\psi \otimes \chi)(\mathbf{x})$, showing that $\phi \otimes \chi \preceq \psi \otimes \chi$ as required.
(iii) Suppose $u \subseteq s$ and let $\mathbf{y}$ be an element of Ω_u. Since $+$ is monotone over $\preceq$,

$$\phi^{\downarrow u}(\mathbf{y}) = \sum\bigl\{\phi(\mathbf{x}):\ \mathbf{x} \in \Omega_s,\ \mathbf{x}^{\downarrow u} = \mathbf{y}\bigr\} \preceq \sum\bigl\{\psi(\mathbf{x}):\ \mathbf{x} \in \Omega_s,\ \mathbf{x}^{\downarrow u} = \mathbf{y}\bigr\},$$

which equals $\psi^{\downarrow u}(\mathbf{y})$, showing that $\phi^{\downarrow u} \preceq \psi^{\downarrow u}$. Hence projection respects $\preceq$. □

Consider a semiring $\mathcal{A} = \langle A, +, \times \rangle$ with a unit element 1, and where $+$ and $\times$ are both monotone over pre-order $\preceq$. For set of variables u, the neutral element e_u is the valuation which is uniformly equal to 1, i.e., for all $\mathbf{x} \in \Omega_u$, $e_u(\mathbf{x}) = 1$. By Theorem 2 and Proposition 4, $\mathcal{A}$-valuations based on ordering $\preceq$ form an ordered valuation algebra. Therefore the results and algorithms of Sections 6.1 and 6.2 apply. We show below how the upper and lower bound functions `LB` and `UB` can be generated.

6.3.2. Generating upper and lower bound functions `LB` *and* `UB`

For any relation $\preceq$ on A it's easy to generate upper and lower bounds of a valuation ϕ which involve less variables. Suppose u is a proper subset of $s = d(\phi)$. We can define u-lower and u-upper bounds τ and θ as follows: for each assignment $\mathbf{x} \in \Omega_u$ we define $\tau(\mathbf{x})$ to be some lower bound of [each element of] the set $\{\phi(\mathbf{xy}):\ \mathbf{y} \in \Omega_{s-u}\}$ and $\theta(\mathbf{x})$ to be some upper bound.

Proposition 5. *Let ϕ, τ and θ be $\mathcal{A}$-valuations, and suppose $u = d(\tau) = d(\theta) \subseteq d(\phi) = s$. Then τ is a u-lower bound for ϕ if and only if for all $\mathbf{x} \in \Omega_u$, $\tau(\mathbf{x})$ is a lower bound for $\{\phi(\mathbf{xy}):\ \mathbf{y} \in \Omega_{s-u}\}$ (i.e., for all $\mathbf{y} \in \Omega_{s-u}$, $\tau(\mathbf{x}) \preceq \phi(\mathbf{xy})$). Furthermore, τ is a greatest u-lower bound for ϕ if and only if for all $\mathbf{x} \in \Omega_u$, $\tau(\mathbf{x})$ is a greatest lower bound for $\{\phi(\mathbf{xy}):\ \mathbf{y} \in \Omega_{s-u}\}$.*

Similarly, θ is a u-upper bound for ϕ if and only if for all $\mathbf{x} \in \Omega_u$, $\theta(\mathbf{x})$ is a upper bound for $\{\phi(\mathbf{xy})\colon \mathbf{y} \in \Omega_{s-u}\}$. θ is a least u-upper bound for ϕ if and only if for all $\mathbf{x} \in \Omega_u$, $\theta(\mathbf{x})$ is a least upper bound for $\{\phi(\mathbf{xy})\colon \mathbf{y} \in \Omega_{s-u}\}$.

Proof. By definition, τ is a u-lower bound for ϕ if and only if $\tau \otimes e_{s-u} \preceq \phi$, which is if and only if for all $\mathbf{z} \in \Omega_s$, $(\tau \otimes e_{s-u})(\mathbf{z}) \preceq \phi(\mathbf{z})$, i.e., $\tau(\mathbf{z}^{\downarrow u}) \preceq \phi(\mathbf{z})$. This is if and only if for all $\mathbf{x} \in \Omega_u$ and $\mathbf{y} \in \Omega_{s-u}$, $\tau(\mathbf{x}) \preceq \phi(\mathbf{xy})$. Hence τ is a u-lower bound for ϕ if and only if for all $\mathbf{x} \in \Omega_u$, $\tau(\mathbf{x})$ is a lower bound for $\{\phi(\mathbf{xy})\colon \mathbf{y} \in \Omega_{s-u}\}$.

Suppose for all $\mathbf{x} \in \Omega_u$, $\tau(\mathbf{x})$ is a greatest lower bound for $\{\phi(\mathbf{xy})\colon \mathbf{y} \in \Omega_{s-u}\}$, and let χ be any u-lower bound for ϕ. Then for all $\mathbf{x} \in \Omega_u$, $\chi(\mathbf{x})$ is a lower bound for $\{\phi(\mathbf{xy})\colon \mathbf{y} \in \Omega_{s-u}\}$, so $\chi(\mathbf{x}) \preceq \tau(\mathbf{x})$; hence $\chi \preceq \tau$, showing that τ is a greatest u-lower bound for ϕ.

To prove the converse, suppose τ is a u-lower bound for ϕ and that there exists $\mathbf{x}_0 \in \Omega_u$ such that $\tau(\mathbf{x}_0)$ is not a greatest lower bound for $\{\phi(\mathbf{x}_0\mathbf{y})\colon \mathbf{y} \in \Omega_{s-u}\}$. It is sufficient to show that then τ is not a greatest u-lower bound for ϕ. There exists a lower bound a for $\{\phi(\mathbf{x}_0\mathbf{y})\colon \mathbf{y} \in \Omega_{s-u}\}$ such that $a \not\preceq \tau(\mathbf{x}_0)$. Define τ' by $\tau'(\mathbf{x}_0) = a$, and for $\mathbf{x} \neq \mathbf{x}_0$, let $\tau'(\mathbf{x}) = \tau(\mathbf{x})$. Then, by the first part, τ' is a u-lower bound for ϕ; however, $\tau' \not\preceq \tau$ which implies that τ is not a greatest u-lower bound for ϕ.

The u-upper bound results follow similarly. □

When $\preceq$ is a lattice one can define least u-upper bounds and greatest u-lower bounds in a simple canonical way.

Least upper bounds and greatest lower bounds when $\preceq$ defines a lattice. Suppose that A is a lattice under the ordering $\preceq$. Then any finite subset B of A has a least upper bound, which we write as $\sup B$, and a greatest lower bound, $\inf B$. In particular when $\preceq$ is a total order, sup is max with respect to $\preceq$ and inf is min. For any $\mathcal{A}$-valuation $\phi : \Omega_s \to A$, and any subset u of s we can define valuations $\phi^{\Downarrow u}$ and $\phi^{\downharpoonleft u}$ both on set of variables u, as follows: for $\mathbf{x} \in \Omega_u$, let $\phi^{\Downarrow u}(\mathbf{x}) = \sup\{\phi(\mathbf{z})\colon \mathbf{z} \in \Omega_s,\ \mathbf{z}^{\downarrow u} = \mathbf{x}\}$, and $\phi^{\downharpoonleft u}(\mathbf{x}) = \inf\{\phi(\mathbf{z})\colon \mathbf{z} \in \Omega_s,\ \mathbf{z}^{\downarrow u} = \mathbf{x}\}$. Then, by Proposition 5, $\phi^{\Downarrow u}$ is the least u-upper bound of ϕ and $\phi^{\downharpoonleft u}$ is the greatest u-lower bound of ϕ. Hence we can define the lower bound function LB used in the propagation algorithms by $\mathtt{LB}(\phi, u) = \phi^{\downharpoonleft u}$, and define the upper bound function UB by $\mathtt{UB}(\phi, u) = \phi^{\Downarrow u}$.

This covers most of the examples in the paper. The ordering used in each example is $\preceq_{\mathcal{A}}$; probability potentials (Examples 1/9), used for inference in Bayesian networks, is based on a semiring with a total order, and so we can define LB and UB in this simple way (as in the definition used for mini-bucket approximation for belief updating [24]); similarly, for any c-semiring which is totally ordered e.g., Examples 2, 3, 5, and 6; furthermore, any c-semiring with idempotent multiplication generates a distributive lattice (Example 4).

The following result implies that we could define in very general circumstances $\mathtt{UB}(\phi, u)$ to be $\phi^{\downarrow u}$ (though it may not be a close upper bound). However, if addition is idempotent, as in c-semirings, for example, it will be a least u-upper bound.

Lemma 2. *Let ϕ be a valuation and let u be a subset of $d(\phi)$. Then $\phi^{\downarrow u}$ is a u-upper bound for ϕ with respect to $\preceq_{\mathcal{A}}$. If addition is idempotent then $\phi^{\downarrow u}$ is a least u-upper bound for ϕ with respect to $\preceq_{\mathcal{A}}$.*

Let $\preceq$ be a pre-order such that addition is monotone with respect to $\preceq$, and suppose that 0 is a lower bound for $\phi(\mathbf{z})$ for each $\mathbf{z} \in \Omega_{d(\phi)}$. Then $\phi^{\downarrow u}$ is a u-upper bound for ϕ with respect to $\preceq$. If also addition is idempotent then $\phi^{\downarrow u}$ is a least u-upper bound for ϕ with respect to $\preceq$.

This upper bound generalizes that used in the mini-bucket approximation for MPE (most probable explanation) in [24], page 116. See also [13] and [14] (Section 6.1) which uses an approximation of a similar form.

Proof. We prove the second half of the lemma; the first half follows from the second half since addition is monotone with respect to $\preceq_{\mathcal{A}}$, and for all $a \in A$, $0 \preceq_{\mathcal{A}} a$.

Suppose 0 is a lower bound of finite sub-multiset G of A. For any $a \in G$, 0 is a lower bound for $\sum\{G - \{a\}\}$, so a is a lower bound for $\sum G$, since addition is monotone with respect to $\preceq$. This implies that $\sum G$ is an upper bound for G. If also addition is idempotent then consider any upper bound b for G. Since addition is monotone with respect to $\preceq$, we have $\sum_{a \in G} a \preceq \sum_{a \in G} b = b$. Hence if addition is idempotent then $\sum G$ is a least upper bound for G.

Let $s = d(\phi)$ and let $\mathbf{x}$ be any element of the frame Ω_u. Then $\phi^{\downarrow u}(\mathbf{x})$ equals $\sum\{\phi(\mathbf{xy})\colon \mathbf{y} \in \Omega_{s-u}\}$, so, by the above argument, $\phi^{\downarrow u}(\mathbf{x})$ is an upper bound for $\{\phi(\mathbf{xy})\colon \mathbf{y} \in \Omega_{s-u}\}$, and is a least upper bound if addition is idempotent. Therefore by Proposition 5, $\phi^{\downarrow u}$ is a u-upper bound for ϕ, which is a least u-upper bound if addition is idempotent. □

6.3.3. Computational efficiency of the propagation

We analyze the computational efficiency of computing the bounds, as described in Section 6.2, for semiring-induced valuations. We focus only on the efficiency of computing the lower approximations; almost identical analysis can be used for the upper approximations (generating the same upper bounds on the number of operations required). So, as in Theorem 9, we wish to compute, for all $l = 1, \dots, m$, the lower bound ϕ^l_{lower} for $(\phi_1 \otimes \cdots \otimes \phi_n)^{\downarrow s_l}$. Let v be the number of variables involved, i.e., $v = |d(\phi_1) \cup \cdots \cup d(\phi_n)|$. The algorithm involves a join tree embedding. We can always construct, by variable elimination (as in fusion/bucket elimination), a join tree with at most v nodes. So let us assume that the join tree has at most v nodes.

To simplify, we assume that the order $\preceq$ on the semiring is a lattice ordering, which covers most of the examples in the paper; this kind of analysis can be extended to more general cases.

We analyze computational efficiency in terms of the number of semiring *basic operations*. This is defined to be one of the following binary operations on the semiring: multiplication, addition, or computing the greatest lower bound (or the least upper bound) of a pair of elements in the semiring. The computational efficiency clearly depends on the choice of the function `size`. We first consider the case where `size` of a multiset M of valuations is defined to be the cardinality of the associated frame, i.e., $|\Omega_{d(M)}|$, where $d(M)$ is the set of variables involved in some valuation in M.

Consider a valuation ψ. Suppose that the cardinality of its associated frame is at most B i.e., $|\Omega_{d(\psi)}| \leqslant B$. Let u be any subset of $d(\psi)$. Computing (all the values of) $\psi^{\downarrow u}$ requires a total of less than B basic operations (additions). Similarly, computing (all the values of) the function $\mathtt{LB}(\psi, u)$ requires less than B basic operations (binary greatest lower bounds), where `LB` is defined as in the lattices paragraph in Section 6.3.2.

Consider a multiset M of valuations, where M contains p valuations and $\mathtt{size}(M) \leqslant B$, i.e., the cardinality of the associated product set is at most B. The computation of the combination of M involves less than $(p-1)B$ basic operations (multiplications).

The term $e_t^{\downarrow\emptyset}$ involved in the `LowerBound` function needs no computation if addition is idempotent, since it is then equal to 1. Otherwise, $e_t^{\downarrow\emptyset}$ can be written as $\prod_{X \in t} e_{\{X\}}^{\downarrow\emptyset}$, where $e_{\{X\}}^{\downarrow\emptyset}$ is equal to $\sum_{\omega \in \Omega_X} 1$. We pre-compute $e_{\{X\}}^{\downarrow\emptyset}$ for each variable X of interest (the cost of this pre-computation is at worst the number of variables times the mean domain size). Computing $e_t^{\downarrow\emptyset}$ then involves less than $|t|$ additional basic operations (multiplications).

Computations for `LowerBound` *function* We will assume that the `Partition` function does not involve any basic operations. By the previous remarks, computing ϕ_i for all $i = 0, \dots k$ involves less than $\sum_{i=0}^k (|M_i| - 1)B = B(|M| - (k+1))$ basic operations. Computing $e_t^{\downarrow\emptyset}$ involves less than $|d(M)|$ basic operations. Computing the valuations $\phi_0^{\downarrow u \cap d(\phi_0)}, \mathtt{LB}(\phi_1, u \cap d(\phi_1)), \dots, \mathtt{LB}(\phi_k, u \cap d(\phi_k))$ involves less than $(k+1)B$ additional basic operations. Hence computing the `LowerBound` function involves less than $B|M| + |d(M)|$ basic operations. Usually the second term will be much smaller than the first.

Let n^* be the maximum value of $|M|$ over all the applications of `LowerBound` in the algorithm, and let v^* be the maximum value of $|d(M)|$, the number of variables involved.

In computing the lower bound ϕ^l_{lower} for all $l = 1, \dots, m$, the procedure `LowerBound` is applied in both directions for each edge in the join tree; it is then applied once for each target set. So `LowerBound` is applied less than $2|V| + m$ times where $|V|$ is the number of nodes in the join tree, and m is the number of target sets, and hence, by our assumption, less than $2v + m$ times. Therefore the number of basic operations in all the applications of the `LowerBound` function is less than $(m + 2v)(Bn^* + v^*)$; the pre-processing adds another small term $\bar{D}v$ which is linear in the number v of variables, where $\bar{D}$ is the mean domain (frame) size of the v variables.

The final stage of the algorithm involves, for each $l = 1, \dots, m$, a combination of a multiset of valuations whose domains are included in s_l; this last part involves less than a total of $Bm(n^* - 1)$ basic operations. Hence overall the number of operations required is less than $(m + 2v)(Bn^* + v^*) + Bmn^* + \bar{D}v$. A crude upper bound for n^* is n, the number of input valuations (n^* will tend to be much smaller than n unless B is very small), and, similarly $v^* \leqslant v$, so an overall upper bound on the number of operations required is $(m + 2v)(Bn + v) + Bmn + \bar{D}v$, which is low order polynomial. In particular, for classes of problems where m and n grow linearly with respect to the number of variables v, and $\bar{D}$ is bounded, then this is $O(v^2)$, since B is a constant.

Suppose instead we define `size` of multiset M to be the number of variables involved in M; let D be an upper bound on the size of the frame of any variable involved. The number of operations is then less than $(m + 2v)(D^B n + v) + D^B mn + Dv$, which is again low order polynomial, since B is a constant.

6.4. Approximations rather than bounds

The lower and upper approximations defined above will in certain situations be sufficient to answer queries. However, they will not necessarily be close approximations. One can often get closer to the exact answers by using other approximations, based on replacing the bound function `LB` (or `UB`) by an approximating function `AP`.

Let ϕ be a valuation with $d(\phi) \supseteq u$. We let $\mathtt{AP}(\phi, u)$ be some valuation θ with $d(\theta) = u$ such that $\theta \otimes e_{d(\phi)}$ in some sense approximates ϕ. The propagation algorithm can then be used with function `AP` replacing `LB` in the lower bound propagation, to give approximations.

If the ordered valuation algebra was generated by a semiring $\mathcal{A}$ and ordering relation $\preceq$, then the approximation can be made pointwise. In particular if $\preceq$ is a total order then for each $\mathbf{x} \in \Omega_u$ we can set $\mathtt{AP}(\phi, u)(\mathbf{x})$ to be some intermediate value (or some "average" value) of $\{\phi(\mathbf{xy})\colon \mathbf{y} \in \Omega_{d(\phi)-u}\}$, as opposed to `LB` which takes the minimum value, and `UB` which uses the maximum value. For example, for the case of probability potentials we could use the mean value as suggested in [24], page 120, i.e., we define $\mathtt{AP}(\phi, u)(\mathbf{x})$ to be the mean value of $\{\phi(\mathbf{xy})\colon \mathbf{y} \in \Omega_{d(\phi)-u}\}$.

6.5. Using propagation of constraints

From a collection of semiring-induced valuations one can generate constraints based on the zeros. These can be used to deduce new constraints which may increase the number of zeros of the input valuations; in particular, this may allow us to eliminate elements of the frame of a variable. The number of non-zero elements in the valuations is relevant to the computational efficiency of the propagation algorithms in Sections 3.1 and 6.2; for example, if valuation ϕ has p non-zero tuples, and ψ has q, then the number of multiplications required in computing $\phi \otimes \psi$ is at most pq. This pre-processing step can sometimes greatly improve the efficiency, especially if many elements of frames are eliminated. This idea is related to the notion of *shrinking* in [6].

6.5.1. Generating implied constraints

A constraint R on set of variables u is a subset of Ω_u; we say that $d(R) = u$. Given $\mathcal{A}$-valuation ϕ, define R^ϕ to be the constraint on variables $d(\phi)$ given by $\mathbf{x} \in R^\phi$ if and only if $\phi(\mathbf{x}) \neq 0$. Constraint R^ϕ gives the non-zero tuples of ϕ. Implied constraints give partial information about the zeros. Let R be a constraint on variables $u \subseteq d(\phi)$. Constraint R is said to be an *implied constraint of* ϕ if the following condition holds, for any $\mathbf{z} \in \Omega_{d(\phi)}$: $\phi(\mathbf{z}) \neq 0 \Rightarrow \mathbf{z}^{\downarrow u} \in R$ (or equivalently: $\mathbf{z} \in R^\phi \Rightarrow \mathbf{z}^{\downarrow u} \in R$). So the complement of R gives zeros of ϕ: if $\mathbf{z}^{\downarrow u} \notin R$ then $\phi(\mathbf{z}) = 0$. R is said to be an implied constraint of multiset $\{\phi_1, \ldots, \phi_n\}$ if it is an implied constraint of $\phi_1 \otimes \cdots \otimes \phi_n$. We also say that $\{\phi_1, \ldots, \phi_n\}$ implies R.

Let $\mathcal{C}$ be a set of constraints involving variables s, and let R be a constraint on variables $u \subseteq s$. We say that $\mathcal{C}$ implies R if $\mathbf{z}^{\downarrow u} \in R$ holds for all $\mathbf{z} \in \Omega_s$ which are solutions of the Constraint Satisfaction Problem $\mathcal{C}$, i.e., such that $\mathbf{z}^{\downarrow d(R')} \in R'$ for all $R' \in \mathcal{C}$.

Proposition 6. *Let M be a multiset of $\mathcal{A}$-valuations.*

(i) *M implies R^ϕ for each $\phi \in M$.*
(ii) *Suppose that $\mathcal{C}$ is a set of implied constraints of M, and that $\mathcal{C}$ implies R. Then M implies R.*
(iii) *If $\{R^\phi\colon \phi \in M\}$ implies constraint R then M implies R.*
(iv) *Suppose now that $\mathcal{A}$ has no zero divisors, i.e., $a \times b$ is non-zero for all non-zero $a, b \in A$. Then $\{R^\phi\colon \phi \in M\}$ implies R if and only if M implies R.*

Proof. Let $\psi = \bigotimes M$ and let $s = d(\psi)$.

(i) Let $\mathbf{z} \in \Omega_s$. For any $\phi \in M$, if $\mathbf{z}^{\downarrow u} \notin R^\phi$ then $\phi(\mathbf{z}^{\downarrow u}) = 0$ so $\psi(\mathbf{z}) = 0$, showing that R^ϕ is an implied constraint of ψ and hence of M.

(ii) Let $u = d(R)$, the set of variables of constraint R. By definition, $u \subseteq s$. Let $\mathbf{z}$ be any element of Ω_s such that $\psi(\mathbf{z}) \neq 0$. Then for all $R' \in \mathcal{C}$, $\mathbf{z}^{\downarrow d(R')} \in R'$ since ψ implies R', and so $\mathbf{z}^{\downarrow u} \in R$, showing that M implies R.

(iii) follows immediately from (i) and (ii).

(iv) We need to show the converse of (iii). Suppose M implies R and that $\mathbf{z} \in \Omega_s$ is such that for all $\phi \in M$, $\mathbf{z}^{\downarrow d(\phi)} \in R^\phi$. Then for all $\phi \in M$, $\phi(\mathbf{z}^{\downarrow d(\phi)}) \neq 0$, so $\psi(\mathbf{z}) = \prod_{\phi \in M} \phi(\mathbf{z}^{\downarrow d(\phi)})$ is non-zero. Since R is an implied constraint of ψ, $\mathbf{z}^{\downarrow d(R)} \in R$, showing that $\{R^\phi \colon \phi \in M\}$ implies R. □

Part (i) of this proposition shows that, given input set of valuations M, we can initialize the set of implied constraints to $\{R^\phi \colon \phi \in M\}$. (Part (iv) of Proposition 6 shows that if $\mathcal{A}$ has no zero divisors, any implied constraint of M is an implied constraint of $\{R^\phi \colon \phi \in M\}$.) We can apply a propagation algorithm to generate more implied constraints from $\{R^\phi \colon \phi \in M\}$; for example, we can use the upper bound approach of Sections 6.1, 6.2 and 6.3 applied to the initial set of constraints (or, similarly, a mini-clustering approach); we could also use arc consistency or e.g., path consistency to generate new constraints. By part (iii) of Proposition 6, the new constraints will be implied constraints of M.

6.5.2. *Using implied constraints*

The following result shows how we can amend an input set of semiring-induced valuations by a set of implied constraints, increasing the number of zero values, but without changing the combination of the valuations.

Proposition 7. *Let $\mathcal{A}$ be a semiring and let $\mathcal{C}$ be a set of implied constraints of multiset of $\mathcal{A}$-valuations $\{\phi_1, \ldots, \phi_n\}$. For $i = 1, \ldots, n$, let ϕ'_i be given by defining $\phi'_i(\mathbf{x})$, for each $\mathbf{x} \in \Omega_{d(\phi_i)}$, as follows: if there exists an implied constraint $R \in \mathcal{C}$ with $d(R) \subseteq d(\phi_i)$ and $\mathbf{x}^{\downarrow d(R)} \notin R$ then let $\phi'_i(\mathbf{x}) = 0$; otherwise define $\phi'_i(\mathbf{x}) = \phi_i(\mathbf{x})$. Then*

$$\phi_1 \otimes \cdots \otimes \phi_n = \phi'_1 \otimes \cdots \otimes \phi'_n.$$

Proof. For $i = 1, \ldots, n$, let $s_i = d(\phi_i)$ and let $s = d(\phi_1) \cup \cdots \cup d(\phi_n)$. Let $\mathbf{z}$ be an element of the frame Ω_s. We need to show that $(\phi_1 \otimes \cdots \otimes \phi_n)(\mathbf{z}) = (\phi'_1 \otimes \cdots \otimes \phi'_n)(\mathbf{z})$. Clearly this holds if for all $i = 1, \ldots, n$, $\phi_i(\mathbf{z}^{\downarrow s_i}) = \phi'_i(\mathbf{z}^{\downarrow s_i})$. So suppose there exists $i \in \{1, \ldots, n\}$ with $\phi_i(\mathbf{z}^{\downarrow s_i}) \neq \phi'_i(\mathbf{z}^{\downarrow s_i})$. Then by definition $\phi'_i(\mathbf{z}^{\downarrow s_i}) = 0$ so $(\phi'_1 \otimes \cdots \otimes \phi'_n)(\mathbf{z}) = 0$. Also, there exists an implied constraint $R \in \mathcal{C}$ on variables $u \subseteq s_i$ with $\mathbf{z}^{\downarrow u} \notin R$. Since R is an implied constraint, $(\phi_1 \otimes \cdots \otimes \phi_n)(\mathbf{z}) = 0$ and so equals $(\phi'_1 \otimes \cdots \otimes \phi'_n)(\mathbf{z})$ as required. □

The implied constraints tell us that some tuples can be set to have zero value; ϕ'_i is obtained from ϕ_i by setting such tuples to zero. The point of replacing the elements ϕ_i by ϕ'_i is to make them easier to combine; the complexity of a combination is related to the number of non-zero tuples; decreasing the number of non-zero tuples can thus make the computations faster, potentially very substantially so if strong constraints can be deduced. An additional advantage is that the upper bounds generated in a join tree propagation upper approximation (Sections 6.2 and 6.3) will sometimes be made tighter by this pre-processing.

6.6. *Setting some semiring values to* 0

Let $\mathcal{A} = \langle A, +, \times \rangle$ be a semiring with zero element 0 and unit element 1. Let $\preceq$ be a pre-order on A such that $+$ and $\times$ are monotone over $\preceq$, and such that $0 \preceq a$ for every element $a \in A$.

The element 0 is a lower bound for every element $a \in A$ in the semiring. So a particular case of a lower bound of an $\mathcal{A}$-valuation is when we replace certain semiring values used in the input valuations by 0. This has computational advantages, as the efficiency of the computation is somewhat related to the number of non-zero values in the input valuations, and constraints propagation approaches can be used, as discussed in Section 6.5. We will consider the effect of choosing a subset P ($\not\ni 0$) of the semiring A, and replacing semiring values in the input valuations which are not in P by 0. With appropriate semiring and choice of P, it can be shown that this does not affect the answer to certain kinds of queries. This is related to the notion of *sinking* in [6].

Consider $\phi = \phi_1 \otimes \cdots \otimes \phi_n$. Let $s = d(\phi)$ and for $i = 1, \ldots, n$ let $s_i = d(\phi_i)$. Define $Q = \{\phi_i(\mathbf{x}_i) \colon i = 1, \ldots, n,\ \mathbf{x}_i \in \Omega_{s_i}\}$ to be the set of all semiring values taken by any of the input valuations. Define $Q^\times$ to be closure of Q under the $\times$ operation.

We consider input $\mathcal{A}$-valuations and subset P of A satisfying the following condition:

(∗) If $a, b \in Q^\times$ and $a \times b \in P$ then $a, b \in P$.

Condition (∗) implies that if elements a_i of $Q^\times$, for $i = 1, \ldots, n$ are such that their product $\prod_{i=1}^n a_i$ is in P then a_i is in P for all $i = 1, \ldots, n$. Hence we have for all $\mathbf{x} \in \Omega_s$, if $\phi(\mathbf{x})$ is in P then $\phi_i(\mathbf{x})$ is in P for all $i = 1, \ldots, n$.

Condition (∗) is satisfied if Q and P satisfy the pair of conditions (i) if $a \in Q$ then $a \preceq 1$, and (ii) if $a \in P$ and $a \preceq b \preceq 1$ then $b \in P$.

An important special case is when, for some $a \in A$, P equals $P_{\succeq a}$ which is defined to be $\{b \in A\colon b \succeq a\}$, or, similarly, when P equals $P_{\not\preceq a} = \{b \in A\colon b \not\preceq a\}$. In either case, condition (∗) is satisfied as long as (i) is satisfied, i.e., the input semiring values are all bounded above by 1.

Consider P satisfying (∗). Define ϕ_i^P by $\phi_i^P(\mathbf{x}_i) = \phi_i(\mathbf{x}_i)$ if $\phi_i(\mathbf{x}_i) \in P$; otherwise $\phi_i^P(\mathbf{x}_i) = 0$. Let $\phi' = \phi_1^P \otimes \cdots \otimes \phi_n^P$. By the above remarks, if $\phi(\mathbf{x}) \in P$ then $\phi'(\mathbf{x}) = \phi(\mathbf{x})$. Also, if $\phi'(\mathbf{x}) \in P$ then for all i, $\phi_i^P(\mathbf{x}^{\downarrow s_i}) \neq 0$ so $\phi_i^P(\mathbf{x}^{\downarrow s_i}) = \phi_i(\mathbf{x}^{\downarrow s_i})$ and so $\phi'(\mathbf{x}) = \phi(\mathbf{x})$. We also have $\phi(\mathbf{x}) \in P$ if and only if $\phi'(\mathbf{x}) \in P$. So if $\phi(\mathbf{x}) \in P$ (or $\phi'(\mathbf{x}) \in P$) then $\phi'(\mathbf{x}) = \phi(\mathbf{x})$.

Suppose we are interested in finding complete assignments $\mathbf{x}$ whose combined semiring value is in P; for example, if we are only interested in $\mathbf{x}$ whose semiring value has a lower bound of a, we could use $P = P_{\succeq a}$. The above argument shows that we can use ϕ' instead of ϕ, without changing the result. This can sometimes greatly improve efficiency, as the components of ϕ' can have many fewer non-zero values than those of ϕ. As in Section 6.5 we can propagate the constraints associated with each ϕ_i^P.

Consider now the case where $\mathcal{A}$ is a c-semiring, with $\preceq\, = \,\preceq_{\mathcal{A}}$ defined by $a \preceq b$ if and only if $a + b = b$. We consider P of the form $P_{\not\preceq a} = \{b \in A\colon b \not\preceq a\}$, for some $a \in A$; this is for a situation where semiring value a (or worse) is not considered significant. P always satisfies (∗). If $\phi(\mathbf{x}) \notin P$ then $0 \preceq \phi(\mathbf{x}) \preceq a$. The results above imply that for any $\mathbf{x} \in \Omega_s$, $\phi'(\mathbf{x}) \preceq \phi(\mathbf{x}) \preceq \phi'(\mathbf{x}) + a$, which leads to: for any $\mathbf{y} \in \Omega_u$, $(\phi')^{\downarrow u}(\mathbf{y}) \preceq \phi^{\downarrow u}(\mathbf{y}) \preceq (\phi')^{\downarrow u}(\mathbf{y}) + a$, giving bounds on $\phi^{\downarrow u}$. If $(\phi')^{\downarrow u}(\mathbf{y}) \succeq a$ then we have equality: $(\phi')^{\downarrow u}(\mathbf{y}) = \phi^{\downarrow u}(\mathbf{y})$.

If additionally, $\preceq$ is a total order and $\phi^{\downarrow u}(\mathbf{y}) \in P$ then $\phi^{\downarrow u}(\mathbf{y}) = (\phi')^{\downarrow u}(\mathbf{y})$. Also, $\phi^{\downarrow u}(\mathbf{y}) \in P$ if and only if $(\phi')^{\downarrow u}(\mathbf{y}) \in P$. Hence if we want to compute projections of combinations of $\mathcal{A}$-valuations then we can use the reduced representation ϕ' (only keeping input semiring values $\succ a$) if we are only interested in partial tuples with (output) semiring values in P (i.e., values more than a). This case of $\mathcal{A}$ based on a totally ordered c-semiring covers several interesting systems see [9,50] and Examples 3, 5 and 6.

7. Conclusion

Semirings are important algebraic structures which induce valuation algebras and permit thus the application of different architectures for local computation. Such semirings can be used to define soft constraints or to generate different uncertainty calculi. In any of these cases inference consists of the solution of the projection problem. A straightforward solution of this problem is in general not feasible, because the domains of the valuations to be treated become much too large and both computing time as well as space requirement grow exponentially. However, the fusion algorithm allows one to limit the domains in which the operations of combination and projection have to be executed to often much smaller dimensions. This may make an otherwise unfeasible computation very fast, in fact, linear in the size of the largest frame to be treated—which is exponential only in the largest node domain size in the join tree, in contrast with a naive algorithm which is exponential in the number of variables.

The fusion algorithm is the base for different derived architectures for local computation. Idempotent semirings lead to idempotent valuation algebras, so-called information algebras. For these algebras the particularly simple and efficient idempotent architecture can be used. In many other cases a notion of division exists in the semiring and can be exported to the induced valuation algebra. Then efficient architectures originally designed for probability networks that use a concept of division can be used.

For situations where the domains of the valuations are such that exact computation is still not feasible, upper and lower bounds can be efficiently derived for a broad class of formalisms, using a modification of the propagation algorithms. These can be used, for example, within a branch and bound algorithm for optimization.

The knowledge of these generic architectures which apply to a multitude of inference problems in very different contexts and formalisms should be part of the tool box of any designer of inference or reasoning systems. It can be useful to solve complex problems, which in the worst case demand an infeasible amount of computation and space.

Acknowledgements

We are very grateful to Marc Pouly, for his comments and proof reading, and to Cesar Schneuwly for his help, especially with the running propagation example.

References

[1] S.M. Aji, R.J. McEliece, The generalized distributive law, IEEE Trans. Inform. Theory 46 (2) (2000) 325–343.
[2] E. Amir, Efficient approximation for triangulation of minimum treewidth, in: Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence, 2001, pp. 7–15.
[3] F. Baccelli, G. Cohen, G.J. Olsder, J.-P. Quadrat, Synchronization and Linearity: An Algebra for Discrete Event Systems, Wiley, 1992.
[4] C. Beeri, R. Fagin, D. Maier, A. Mendelzon, J. Ullman, M. Yannakakis, Properties of acyclic database schemes, in: ACM Symposium on Theory of Computing, ACM Press, New York, 1981, pp. 355–362.
[5] S. Bistarelli, T. Fruewirth, M. Marthe, F. Rossi, Soft constraint propagation and solving in constraint handling rules, Comput. Intell. (2004), Special Issues on Preferences in AI and CP.
[6] S. Bistarelli, S.K.L. Fung, J.H.M. Lee, H. Leung, A local search framework for semiring-based constraint satisfaction problems, in: Proc. CP2003 Workshop on Soft Constraints (Soft-2003), 2003.
[7] S. Bistarelli, F. Gadducci, Enhancing constraints manipulation in semiring-based formalisms, in: Proc. 17th European Conference on Artificial Intelligence (ECAI 2006), 2006, pp. 63–67.
[8] S. Bistarelli, U. Montanari, F. Rossi, Semiring-based constraint satisfaction and optimization optimisation, J. ACM 44 (1997) 201–236.
[9] S. Bistarelli, U. Montanari, F. Rossi, T. Schiex, G. Verfaillie, H. Fargier, Semiring-based CSPs and valued CSPs: Frameworks, properties and comparison, CONSTRAINTS: An International Journal 4 (3) (1999).
[10] H.L. Bodlaender, A tourist guide through treewidth, Acta Cybern. 11 (1–2) (1993) 1–22.
[11] H.L. Bodlaender, Treewidth: Characterizations, applications, and computations, in: F.V. Fomin (Ed.), WG, in: Lecture Notes in Computer Science, vol. 4271, Springer, 2006, pp. 1–14.
[12] K. Cechlárová, J. Plávka, Linear independence in bottleneck algebras, Fuzzy Sets Syst. 77 (3) (1996) 337–348.
[13] L. Chang, Semiring-based unifying framework for constraint-based inference, Master's thesis, University of British Columbia, 2005.
[14] L. Chang, A.K. Mackworth, Generalized constraint-based inference, Tech. Rep. TR-2005-10, Dept. of Computer Science, Univ. of British Columbia, 2005.
[15] A.H. Clifford, G.B. Preston, Algebraic Theory of Semigroups. American Mathematical Society, Providence, Rhode Island, 1967.
[16] M. Cooper, T. Schiex, Arc consistency for soft constraints, Artif. Intell. 154 (1–2) (2004) 199–227.
[17] R. Croisot, Demi-groupes inversifs et demi-groupes réunions de demi-groupes simples, Ann. Sci. Ecole Norm. Sup. 79 (3) (1953) 361–379.
[18] B. De Baets, Idempotent uninorms, European J. Op. Res. 118 (631-642) (1996) 00.
[19] J. De Kleer, J. Brown, Theories of causal ordering, Artif. Intell. 29 (1986) 33–61.
[20] R. Dechter, Mini-buckets: A general scheme for generating approximations in automated reasoning, in: Proc. Fifteenth International Joint Conference of Artificial Intelligence (IJCAI97), 1997, pp. 1297–1303.
[21] R. Dechter, Bucket elimination: A unifying framework for reasoning, Artif. Intell. 113 (1–2) (1999) 41–85.
[22] R. Dechter, K. Kask, J. Larrosa, A general scheme for multiple lower bound computation in constraint optimization, in: Proc. CP2001, 2001, pp. 346–360.
[23] R. Dechter, J. Pearl, Network-based heuristics for constraint satisfaction problems, Artif. Intell. 34 (1) (1987) 1–38.
[24] R. Dechter, I. Rish, Mini-buckets: A general scheme for bounded inference, J. ACM 50 (2) (2003) 107–153.
[25] V. Gogate, R. Dechter, A complete anytime algorithm for treewidth, in: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence UAI-04, 2004, pp. 201–208.
[26] R. Haenni, Ordered valuation algebras: A generic framework for approximating inference, Internat. J. Approx. Reason. 37 (1) (2004) 1–41.
[27] R. Haenni, J. Kohlas, N. Lehmann, Probabilistic argumentation systems, in: J. Kohlas, S. Moral (Eds.), Handbook of Defeasible Reasoning and Uncertainty Management Systems, vol. 5: Algorithms for Uncertainty and Defeasible Reasoning, Kluwer, Dordrecht, 2000, pp. 221–287; http://diuf.unifr.ch/tcs/publications/ps/hkl2000.pdf.
[28] E. Hewitt, H. Zuckermann, The l1-algebra of a commutative semigroup, Trans. Amer. Math. Soc. 83 (1956) 70–97.
[29] F. Jensen, S. Lauritzen, K. Olesen, Bayesian updating in causal probabilistic networks by local computations, Comp. Stat. Q. 4 (1990) 269–282.
[30] K. Kask, R. Dechter, Branch and bound with mini-bucket heuristics, in: Proc. International Joint Conference on Artificial Intelligence (IJCAI99), 1999, pp. 426–433.
[31] K. Kask, R. Dechter, Mini-bucket heuristics for improved search, in: Proc. UAI99, 1999, pp. 314–323.
[32] K. Kask, R. Dechter, J. Larrosa, A. Dechter, Unifying cluster-tree decompositions for reasoning in graphical models, Artif. Intell. 166 (1–2) (2005) 165–193.
[33] E. Klement, R. Mesiar, E. Pap, Triangular Norms, Trends in Logic, Kluwer Academic Publ. Dordrecht, 2000.

[34] J. Kohlas, Information Algebras: Generic Structures for Inference, Springer-Verlag, 2003.
[35] J. Kohlas, Valuation algebras induced by semirings. Tech. Rep. 04-03, Department of Informatics, University of Fribourg, 2004; http://diuf.unifr.ch/tcs/publications/ps/kohlas2004a.pdf.
[36] J. Kohlas, R. Haenni, S. Moral, Propositional information systems, J. Logic Comput. 9 (5) (1999) 651–681; http://diuf.unifr.ch/tcs/publications/ps/kmh99.pdf.
[37] J. Kohlas, P. Shenoy, Computation in valuation algebras, in: J. Kohlas, S. Moral (Eds.), Handbook of Defeasible Reasoning and Uncertainty Management Systems, vol. 5: Algorithms for Uncertainty and Defeasible Reasoning, Kluwer, Dordrecht, 2000, pp. 5–40.
[38] V. Kolokoltsov, V. Maslov, Idempotent Analysis and its Applications, Kluwer Academic Publ. Dordrecht, 1997.
[39] J. Larrosa, T. Schiex, Solving weighted CSP by maintaining arc consistency, Artif. Intell. 159 (2004) 1–26.
[40] S. Lauritzen, F. Jensen, Local computation with valuations from a commutative semigroup, Ann. Math. Artif. Intell. 21 (1) (1997) 51–70.
[41] S. Lauritzen, D. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems, J. Royal Stat. Soc. 50 (2) (1988) 157–224.
[42] D. Maier, The Theory of Relational Databases, Pitman, London, 1983.
[43] R. Mateescu, R. Dechter, K. Kask, Tree approximation for belief updating, in: Proc. AAAI-2002, 2002, pp. 553–559.
[44] K. Menger, Statistical metrics, Proc. Nat. Acad. Sci. 28 (1942) 535–537.
[45] J. Mengin, N. Wilson, Logical deduction using the local computation framework, in: A. Hunter, S. Parsons (Eds.), European Conf. ECSQARU'99, London. Lecture Notes in Artif. Intell. Springer, 1999, pp. 386–396.
[46] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., 1988.
[47] M. Pouly, Nenok 1.1 user guide, Tech. Rep. 06-02, Department of Informatics, University of Fribourg, 2006.
[48] M. Pouly, J. Kohlas, Minimizing communication costs of distributed local computation, Tech. rep., Department of Informatics, University of Fribourg, 2005.
[49] T. Schiex, Possibilistic constraint satisfaction problems or "how to handle soft constraints?" in: D. Dubois, M.P. Wellman, B. D'Ambrosio, P. Smets (Eds.), Uncertainty in Artificial Intelligence: Proc. of the Eighth Conference, Kaufmann, San Mateo, CA, 1992, pp. 268–275.
[50] Schiex, T., Fargier, H., Verfaillie, G., Valued constraint satisfaction problems: Hard and easy problems, in: Proc. IJCAI-95, 1995, pp. 631–637.
[51] C. Schneuwly, M. Pouly, J. Kohlas, Local computation in covering join trees, Tech. Rep. 04-16, Department of Informatics, University of Fribourg, 2004; http://diuf.unifr.ch/tcs/publications/ps/schneuwlypoulykohlas04.pdf.
[52] B. Schweizer, A. Sklar, Statistical metric spaces, Pacific J. Math. 10 (1960) 313–334.
[53] G. Shafer, An axiomatic study of computation in hypertrees, Working Paper 232, School of Business, University of Kansas, 1991.
[54] G. Shafer, Probabilistic Expert Systems, CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 67, SIAM, Philadelphia, PA, 1996.
[55] G. Shafer, P. Shenoy, Local computation in hypertrees, Tech. Rep. 201, School of Business, University of Kansas, Lawrence, 1988.
[56] P. Shenoy, Valuation-based systems: A framework for managing uncertainty in expert systems, in: L. Zadeh, J. Kacprzyk (Eds.), Fuzzy Logic for the Management of Uncertainty, John Wiley & Sons, 1992, pp. 83–104.
[57] P. Shenoy, Axioms for dynamic programming, in: A. Gammerman (Ed.), Computational Learning and Probabilistic Reasoning, Wiley, Chichester, UK, 1996, pp. 259–275.
[58] P.P. Shenoy, Binary join trees for computing marginals in the Shenoy--Shafer architecture, Internat. J. Approx. Reason. 17 (1997) 239–263; http://citeseer.ist.psu.edu/article/shenoy97binary.html.
[59] P.P. Shenoy, G. Shafer, Axioms for probability and belief-function propagation, in: R.D. Shachter, T.S. Levitt, L.N. Kanal, J.F. Lemmer (Eds.), Uncertainty in Artificial Intelligence 4, in: Machine Intelligence and Pattern Recognition, vol. 9, Elsevier, Amsterdam, 1990, pp. 169–198.
[60] W. Spohn, Ordinal conditional functions: A dynamic theory of epistemic states, in: W. Harper, B. Skyrms (Eds.), Causation in Decision, Belief Change, and Statistics, vol. 2, Dordrecht, Netherlands, 1988, pp. 105–134.
[61] T. Tamura, N. Kimura, On decompositions of a commutative semigroup, Kodai Math. Sem. Rep. (1954) 109–112.
[62] N. Wilson, Bounds and pre-processing for local computation of semiring valuations, in: J. Kohlas, J. Mengin, N. Wilson (Eds.), ECAI'2004, Workshop 22: Local Computation for Logics and Uncertainty, 2004, pp. 53–56.
[63] R. Yager, A. Rybalov, Uninorm aggregation operators, Fuzzy Sets Syst. 80 (1996) 111–120.