Francesco M. Malvestuto
Marginalization in models generated by compositional expressions

Kybernetika, Vol. 51 (2015), No. 4, 541–570

Persistent URL: http://dml.cz/dmlcz/144467

KYBERNETIKA — VOLUME *51* (2015), NUMBER 4, PAGES 541–570

MARGINALIZATION IN MODELS GENERATED BY COMPOSITIONAL EXPRESSIONS

Francesco M. Malvestuto

In the framework of models generated by compositional expressions, we solve two topical marginalization problems (namely, the *single-marginal problem* and the *marginal-representation problem*) that were solved only for the special class of the so-called "canonical expressions". We also show that the two problems can be solved "from scratch" with preliminary symbolic computation.

Keywords: compositional expression, compositional model, marginalization, syntax tree

Classification: 05C65, 05C85, 65C50, 60E99, 68T37

1. INTRODUCTION

Compositional models were introduced to construct probability distributions from lower-order probability distributions [6, 8, 9, 11, 13] as an operational alternative to the graphical approach used to model Bayesian and causal networks. They were also applied to belief functions [10, 15, 16], possibility functions [15] and Shenoy valuations [14]. A topical problem common to graphical and compositional models is that of computing marginals [4, 7, 12].

In this paper, we consider a more general version of compositional models, namely, models generated by *compositional expressions* [27, 28] and formed by distributions whose values (such as probabilities) can be added, multiplied and divided according to the algebraic rules of a *semifield* [27], which is defined in the Appendix (see Section 10). Owing to their generality, such models find applications also to multidimensional databases (SUM-data, MAX-data, MIN-data, Boolean data), in which query answering requires to combine data stored in distinct tables [26].

In this framework, we solve two topical marginalization problems, namely, the *single-marginal* and *marginal-representation* problems (see Section 5 for their statements). We also show that they can be solved "from scratch" with preliminary symbolic computation. Only to give an illustrative example, consider the model generated by the compositional expression

$$\theta = (AB \rhd AC) \rhd ((BC \rhd AB) \rhd CD)\,,$$

DOI: 10.14736/kyb-2015-4-0541

and suppose we are interested in the marginal on AC of the value of θ under its "interpretation"

$$I = \langle f(AB), g(AC), h(BC), k(AB), l(CD) \rangle ,$$

where $f(AB), g(AC), h(BC), k(AB), l(CD)$ are real-valued distributions (for formal definitions and notation, see Section 2). Instead of computing numerically the value of θ under I and then marginalizing it on AC, we first construct the algebraic expression of the value of θ under I, which in our case reads

$$\frac{f(AB) \times g(AC) \times l(CD)}{\Big(\sum\limits_C g(AC)\Big) \times \Big(\sum\limits_D l(CD)\Big)} .$$

Next, using suitable reduction rules we simplify the sum

$$\sum_{BD} \frac{f(AB) \times g(AC) \times l(CD)}{\Big(\sum\limits_C g(AC)\Big) \times \Big(\sum\limits_D l(CD)\Big)}$$

and, thus, obtain the algebraic expression of the wanted marginal on AC

$$\frac{\Big(\sum\limits_B f(AB)\Big) \times g(AC)}{\sum\limits_C g(AC)} ,$$

which finally is evaluated using the numeric values of the distributions in I. The result of the evaluation will give the wanted marginal. It is worth observing that the distributions $h(BC)$ and $k(AC)$ are missing from the algebraic expression of the value of θ under I, which means that BC and the second occurrence of AB in θ are "redundant", and θ is "algebraically equivalent" (in the sense stated in Section 9) to the compositional expression $(AB \rhd AC) \rhd CD$.

Compositional expressions are essentially the same as "merge expressions" which are studied in multidimensional databases [26]. The single-marginal problem was discussed in [4, 7, 12] for "generating sequences" of probability distributions; moreover, both the single-marginal problem and the marginal-representation problem were solved in [26] for "perfect merge expressions", which correspond to the so-called "canonical expressions" [27].

The paper is organized as follows. Section 2 contains basic definitions of distributions over a semifield. Section 3 reviews some known results on the composition of distributions, with an explicit reference to the metric semifields reported in the Appendix. Section 4 introduces compositional expressions and their values under valid interpretations; moreover, it contains a validity test for the general case and a cheaper validity test for metric semifields. In Section 5 we state the two above-mentioned marginalization problems and solve them by including in the input also the data structures constructed during the validity test. In Sections 6 and 7 we solve the two marginalization problems from scratch. In Section 8 we discuss the case of Boolean distributions. Section 9 contains a note for future research, and the Appendix (in Section 10) contains the precise definition of a semifield and the list of the metric semifields recurring in this paper.

2. PRELIMINARIES

2.1. Discrete variables

Let X be a finite set of finite-valued variables. An X-*tuple* is an assignment of values to the variables in X. By $dom(X)$ we denote the set of all X-tuples; accordingly, $|dom(X)|$ denotes the number of all X-tuples. We use the initial capital-case letters of the alphabet (e. g., A, B, C) to denote single variables, and the final capital-case letters to denote sets of variables (e. g., X, Y, Z). Moreover, sets of variables are written as sequences of variables; thus, ABC stands for $\{A, B, C\}$. Finally, we denote an X-tuple by the corresponding lower-case bold-faced letter $\mathbf{x}$.

Let Y be a nonempty subset of X; given an X-tuple $\mathbf{x}$, by $\mathbf{x}_Y$ we denote the Y-tuple obtained from $\mathbf{x}$ by ignoring the values of the variables in $X \setminus Y$.

2.2. Distributions

Let $\Sigma = \langle \mathtt{S}, (\oplus, \mathtt{0}), (\otimes, \mathtt{1}) \rangle$ be a semifield (for its definition see the Appendix in Section 10) and let X be a finite set of finite-valued variables. A Σ-*distribution* with *scheme* X, written $f(X)$, is any $\mathtt{S}$-valued function defined on $dom(X)$. For example, if Σ is the max-sum semifield (see the Appendix), a Σ-*distribution* with scheme X is a non-negative real-valued function defined on $dom(X)$. Note that a probability distribution is such a Σ-distribution.

A Σ-distribution $f(X)$ is *null* if $f(\mathbf{x}) = \mathtt{0}$ everywhere (that is, for every X-tuple $\mathbf{x}$) and it will be denoted by $\mathtt{0}(X)$.

2.3. The support of a distribution

The *support* of a Σ-distribution $f(X)$, denoted by $\|f\|$, is the (possibly empty) set of X-tuples $\mathbf{x}$ with $f(\mathbf{x}) \neq \mathtt{0}$. Thus, $f(X)$ is the null distribution $\mathtt{0}(X)$ if and only if $\|f\| = \emptyset$.

The support of $f(X)$ can be viewed as a *relation* [3] with scheme X and, hence, we can apply the following two operators of *relational algebra* [3] to supports of distributions:

(*projection*) Let $f(X)$ be a distribution, and let Y be a nonempty subset of X. The projection of $\|f\|$ on Y is the relation

$$\pi_Y(\|f\|) = \{\mathbf{x}_Y : \mathbf{x} \in \|f\|\}.$$

Note that, if Z is a nonempty subset of Y, then $\pi_Z(\|f\|) = \pi_Z(\pi_Y(\|f\|))$.

(*join*) Let $f(X)$ and $g(Y)$ be distributions, and let $V = X \cup Y$. The (natural) join of $\|f\|$ and $\|g\|$ is the relation

$$\|f\| \bowtie \|g\| = \{\mathbf{v} \in dom(V) : \mathbf{v}_X \in \|f\| \ \& \ \mathbf{v}_Y \in \|g\|\}.$$

Note that the join operator is both associative and commutative [3].

Example 2.1. Let A, B and C be three binary variables. Consider the following three relations r_1, r_2 and r_3 with schemes AB, AC and BC respectively:

$$r_1 = \{(\mathbf{a}_1, \mathbf{b}_1), (\mathbf{a}_2, \mathbf{b}_2)\} \qquad r_2 = \{(\mathbf{a}_1, \mathbf{c}_1), (\mathbf{a}_2, \mathbf{c}_1)\} \qquad r_3 = \{(\mathbf{b}_1, \mathbf{c}_1), (\mathbf{b}_2, \mathbf{c}_1)\}.$$

It is easily seen that

$$r_1 \bowtie r_2 = r_1 \bowtie r_3 = r_1 \bowtie r_2 \bowtie r_3 = \{(\mathbf{a}_1, \mathbf{b}_1, \mathbf{c}_1), (\mathbf{a}_2, \mathbf{b}_2, \mathbf{c}_1)\},$$

$$r_2 \bowtie r_3 = \{(\mathbf{a}_1, \mathbf{b}_1, \mathbf{c}_1), (\mathbf{a}_1, \mathbf{b}_2, \mathbf{c}_1), (\mathbf{a}_2, \mathbf{b}_1, \mathbf{c}_1)(\mathbf{a}_2, \mathbf{b}_2, \mathbf{c}_1)\}.$$

2.4. Marginals and grand-total of a distribution

The *marginal* of $f(X)$ on a nonempty subset Y of X, written $f^{\downarrow Y}$, is defined as follows: for every Y-tuple $\mathbf{y}$

$$f^{\downarrow Y}(\mathbf{y}) = \bigoplus_{\mathbf{x} \in dom(X):\, \mathbf{x}_Y = \mathbf{y}} f(\mathbf{x}).$$

Lemma 2.2. Let $f(X)$ be a Σ-distribution, where Σ is any semifield, and let Y be a nonempty subset of X. Then $\|f^{\downarrow Y}\| \subseteq \pi_Y(\|f\|)$.

Proof. By the very definition of $f^{\downarrow Y}$, one has that

$$f^{\downarrow Y}(\mathbf{y}) = \begin{cases} \mathtt{0} & \text{if } \|f\| = \emptyset \\ \bigoplus_{\mathbf{x} \in \|f\|:\, \mathbf{x}_Y = \mathbf{y}} f(\mathbf{x}) & \text{otherwise} \end{cases}$$

so that

$$\|f^{\downarrow Y}\| = \{\mathbf{y} \in \pi_Y(\|f\|) : \bigoplus_{\mathbf{x} \in \|f\|: \mathbf{x}_Y = \mathbf{y}} f(\mathbf{x}) \neq \mathtt{0}\}$$

from which the statement follows. □

Note that, if Σ is the Galois field $GF(2)$ and the number of X-tuples $\mathbf{x} \in \|f\|$ with $\mathbf{x}_Y = \mathbf{y}$ is an even number greater than 0, then $f^{\downarrow Y}(\mathbf{y}) = \mathtt{0}$ so that $\mathbf{y} \in \pi_Y(\|f\|) \setminus \|f^{\downarrow Y}\|$.

The following property of metric semifields will be often applied in the sequel.

Lemma 2.3. Let $f(X)$ be a Σ-distribution, where Σ is a metric semifield, and let Y be a nonempty subset of X. Then $\|f^{\downarrow Y}\| = \pi_Y(\|f\|)$.

Proof. By Lemma 2.2, it is sufficient to prove that $\pi_Y(\|f\|) \subseteq \|f^{\downarrow Y}\|$. Let $\mathbf{y}$ be any Y-tuple in $\pi_Y(\|f\|)$. By definition of $\pi_Y(\|f\|)$, there exists at least one X-tuple $\mathbf{x}^* \in \|f\|$ such that $\mathbf{x}^*_Y = \mathbf{y}$. Since $\mathbf{x}^* \in \|f\|$, one has that $f(\mathbf{x}^*) \neq \mathtt{0}$ and, since Σ is zero-sum free, one has that $f^{\downarrow Y}(\mathbf{y}) = \bigoplus_{\mathbf{x} \in \|f\|: \mathbf{x}_Y = \mathbf{y}} f(\mathbf{x}) \neq \mathtt{0}$. It follows that $\mathbf{y} \in \|f^{\downarrow Y}\|$ which proves that $\pi_Y(\|f\|) \subseteq \|f^{\downarrow Y}\|$. □

$f^{\downarrow Y}(\mathbf{y})$	$f^{\downarrow\emptyset}$	metric semifield
$\sum_{\mathbf{x}\in dom(X):\mathbf{x}_Y=\mathbf{y}} f(\mathbf{x})$	$\sum_{\mathbf{x}\in dom(X)} f(\mathbf{x})$	sum-product semifield
$\max_{\mathbf{x}\in dom(X):\mathbf{x}_Y=\mathbf{y}} f(\mathbf{x})$	$\max_{\mathbf{x}\in dom(X)} f(\mathbf{x})$	max-product semifield
$\max_{\mathbf{x}\in dom(X):\mathbf{x}_Y=\mathbf{y}} f(\mathbf{x})$	$\max_{\mathbf{x}\in dom(X)} f(\mathbf{x})$	max-sum semifield
$\min_{\mathbf{x}\in dom(X):\mathbf{x}_Y=\mathbf{y}} f(\mathbf{x})$	$\min_{\mathbf{x}\in dom(X)} f(\mathbf{x})$	min-product semifield
$\min_{\mathbf{x}\in dom(X):\mathbf{x}_Y=\mathbf{y}} f(\mathbf{x})$	$\min_{\mathbf{x}\in dom(X)} f(\mathbf{x})$	min-sum semifield
$\bigvee_{\mathbf{x}\in dom(X):\mathbf{x}_Y=\mathbf{y}} f(\mathbf{x})$	$\bigvee_{\mathbf{x}\in dom(X)} f(\mathbf{x})$	Boolean algebra

Tab. 1. Marginals and grand-totals for metric semifields.

The *grand-total* of $f(X)$, written $f^{\downarrow\emptyset}$, is defined as follows:

$$f^{\downarrow\emptyset} = \bigoplus_{\mathbf{x}\in dom(X)} f(\mathbf{x}) = \begin{cases} \mathtt{0} & \text{if } \|f\| = \emptyset \\ \bigoplus_{\mathbf{x}\in\|f\|} f(\mathbf{x}) & \text{otherwise.} \end{cases}$$

Note that, if Σ is the Galois field $GF(2)$ and $\|f\|$ contains an even number of X-tuples greater than 0, then $f^{\downarrow\emptyset} = \mathtt{0}$. However, for a metric semifield $f^{\downarrow\emptyset} = \mathtt{0}$ if and only if $f(X)$ is the null distribution $\mathtt{0}(X)$.

In this paper we pay a special attention to the metric semifields reported in Table 1, in which marginals and grand-totals are explicitly defined.

2.5. Extensions

Let $f(X)$ be a Σ-distribution, where Σ is a semifield. By an *extension* of $f(X)$ to a superset V of X we mean any Σ-distribution $e(V)$ whose marginal on X coincides with $f(X)$, that is, $e^{\downarrow X} = f(X)$.

3. COMPOSITION OF DISTRIBUTIONS

3.1. General properties

Let $f(X)$ and $g(Y)$ be Σ-distributions, where Σ is a semifield. We say that $f(X)$ is *composable* with $g(Y)$ if

(a) either $f(X) = \mathtt{0}(X)$, or

(b) if $X \cap Y = \emptyset$ then $g^{\downarrow\emptyset} \neq \mathtt{0}$; otherwise, for every X-tuple $\mathbf{x}$ with $f(\mathbf{x}) \neq \mathtt{0}$, one has $g^{\downarrow X\cap Y}(\mathbf{x}_{X\cap Y}) \neq \mathtt{0}$.

Theorem 3.1. A Σ-distribution $f(X)$ is composable with a Σ-distribution $g(Y)$ if and only if

(a) either $f(X) = \mathtt{0}(X)$, or

(b') if $X \cap Y = \emptyset$ then $g^{\downarrow\emptyset} \neq \mathtt{0}$; otherwise, $\pi_{X\cap Y}(\|f\|) \subseteq \|g^{\downarrow X\cap Y}\|$.

P r o o f. We need to prove that under $X \cap Y \neq \emptyset$ conditions (b) and (b') are equivalent. Let $Z = X \cap Y$.
$(b) \Rightarrow (b')$. Let $\mathbf{z}$ be any Z-tuple in $\pi_Z(\|f\|)$. Then there exists an X-tuple $\mathbf{x} \in \|f\|$ such that $\mathbf{x}_Z = \mathbf{z}$. Since $\mathbf{x} \in \|f\|$, one has $f(\mathbf{x}) \neq \mathtt{0}$. By (b), $g^{\downarrow Z}(\mathbf{z}) \neq \mathtt{0}$ so that $\mathbf{z} \in \|g^{\downarrow Z}\|$.
$(b') \Rightarrow (b)$. Let $\mathbf{x}$ be any X-tuple with $f(\mathbf{x}) \neq \mathtt{0}$. Then, $\mathbf{x} \in \|f\|$ and, hence, $\mathbf{x}_Z \in \pi_Z(\|f\|)$. By (b'), $\mathbf{x}_Z \in \|g^{\downarrow Z}\|$ so that $g^{\downarrow Z}(\mathbf{x}_Z) \neq \mathtt{0}$. □

Assume that $f(X)$ is composable with $g(Y)$. Let $V = X \cup Y$ and $Z = X \cap Y$. The *composition* (or the "merge" [26]) of $f(X)$ with $g(Y)$, denoted by $f \rhd g$, is the Σ-distribution with scheme V defined as follows [27]:

— if $f(X) = \mathtt{0}(X)$ then $f \rhd g = \mathtt{0}(V)$, otherwise

— for every V-tuple $\mathbf{v}$

$$(f \rhd g)(\mathbf{v}) = \begin{cases} f(\mathbf{v}_X) \otimes g(\mathbf{v}_Y) \otimes \overline{g^{\downarrow\emptyset}} & \text{if } Z = \emptyset \\ f(\mathbf{v}_X) \otimes g(\mathbf{v}_Y) \otimes \overline{g^{\downarrow Z}(\mathbf{v}_Z)} & \text{otherwise} \end{cases}$$

where $\overline{g^{\downarrow\emptyset}}$ and $\overline{g^{\downarrow Z}(\mathbf{v}_Z)}$ denote the multiplicative inverses of $g^{\downarrow\emptyset}$ and of $g^{\downarrow Z}(\mathbf{v}_Z)$, respectively.

Finally, if $f(X)$ is not composable with $g(Y)$, then we say that $f \rhd g$ is *undefined*.

Theorem 3.2. (Malvestuto [27]) Let $f(X)$ and $g(Y)$ be Σ-distributions, where Σ is a semifield. If $f(X)$ is composable with $g(Y)$, then

(i) $f \rhd g$ is an extension of $f(X)$ to $X \cup Y$,

(ii) $\|f \rhd g\| = \|f\| \bowtie \|g\|$,

(iii) $\|f\| = \pi_X(\|f \rhd g\|)$.

The following is a straightforward consequence of the definition of $f(X) \rhd g(Y)$.

Remark 3.3. Let $f(X)$ and $g(Y)$ be Σ-distributions. Assume that $Y \subseteq X$. If $f(X)$ is composable with $g(Y)$ then $f(X) \rhd g(Y) = f(X)$.

By Remark 3.3, the composition operator is idempotent; moreover, it is neither commutative nor associative [6, 9]. However, if $f(X)$ and $g(Y)$ are both marginals of a distribution with scheme $X \cup Y$ and if $f(X)$ is composable with $g(Y)$ and *vice versa*, then $f \rhd g = g \rhd f$.

3.2. Metric semifields

Consider now the case that Σ is a metric semifield. Let $f(X)$ and $g(Y)$ be Σ-distributions. Then, we know that

— $f(X) = \mathtt{0}(X)$ if and only if $\|f\| = \emptyset$,

— $g^{\downarrow\emptyset} = \mathtt{0}$ if and only if $\|g\| = \emptyset$,

— $\|g^{\downarrow X\cap Y}\| = \pi_{X\cap Y}(\|g\|)$ (by Lemma 2.3).

Therefore, for a metric semifield Σ, the conditions for composability (see Theorem 3.1 above) can be stated in terms of supports of distributions.

Theorem 3.4. Let Σ be a metric semifield. A Σ-distribution $f(X)$ is composable with a Σ-distribution $g(Y)$ if and only if

(a*) either $\|f\| = \emptyset$, or

(b*) if $X \cap Y = \emptyset$ then $\|g\| \neq \emptyset$; otherwise, $\pi_{X\cap Y}(\|f\|) \subseteq \pi_{X\cap Y}(\|g\|)$.

Table 2 reports the specific definition of $(f \rhd g)(\mathbf{v})$ for the metric semifields reported in Table 1.

Σ	$Z = \emptyset$	$Z \neq \emptyset$
sum-product	$\dfrac{f(\mathbf{v}_X) \times g(\mathbf{v}_Y)}{\sum_{\mathbf{y}\in dom(Y)} g(\mathbf{y})}$	$\dfrac{f(\mathbf{v}_X) \times g(\mathbf{v}_Y)}{\sum_{\mathbf{y}\in dom(Y):\mathbf{y}_Z=\mathbf{v}_Z} g(\mathbf{y})}$
min-product	$\dfrac{f(\mathbf{v}_X) \times g(\mathbf{v}_Y)}{\min_{\mathbf{y}\in dom(Y)} g(\mathbf{y})}$	$\dfrac{f(\mathbf{v}_X) \times g(\mathbf{v}_Y)}{\min_{\mathbf{y}\in dom(Y):\mathbf{y}_Z=\mathbf{v}_Z} g(\mathbf{y})}$
min-sum	$f(\mathbf{v}_X) + g(\mathbf{v}_Y) - \min_{\mathbf{y}\in dom(Y)} g(\mathbf{y})$	$f(\mathbf{v}_X) + g(\mathbf{v}_Y) - \min_{\mathbf{y}\in dom(Y):\mathbf{y}_Z=\mathbf{v}_Z} g(\mathbf{y})$
max-product	$\dfrac{f(\mathbf{v}_X) \times g(\mathbf{v}_Y)}{\max_{\mathbf{y}\in dom(Y)} g(\mathbf{y})}$	$\dfrac{f(\mathbf{v}_X) \times g(\mathbf{v}_Y)}{\max_{\mathbf{y}\in dom(Y):\mathbf{y}_Z=\mathbf{v}_Z} g(\mathbf{y})}$
max-sum	$f(\mathbf{v}_X) + g(\mathbf{v}_Y) - \max_{\mathbf{y}\in dom(Y)} g(\mathbf{y})$	$f(\mathbf{v}_X) + g(\mathbf{v}_Y) - \max_{\mathbf{y}\in dom(Y):\mathbf{y}_Z=\mathbf{v}_Z} g(\mathbf{y})$
Boolean	$f(\mathbf{v}_X) \wedge g(\mathbf{v}_Y)$	$f(\mathbf{v}_X) \wedge g(\mathbf{v}_Y)$

Tab. 2. The composition of $f(X)$ with $g(Y)$ in metric semifields depending on $Z = X \cap Y$.

4. COMPOSITIONAL EXPRESSIONS

A *compositional expression* is a parenthesized expression formed out by nonempty sets of (finite-valued) variables, and the symbol "$\rhd$" of the composition operator. Explicitly, the following provides a recursive definition of a compositional expression:

(i) if X is a set of variables, then X is a compositional expression;

(ii) if θ_1 and θ_2 are compositional expressions, then $(\theta_1) \rhd (\theta_2)$ is a compositional expression.

Let θ be a compositional expression. The *base sequence* [27] of θ is the sequence σ of the sets featured in θ arranged according to their order of appearance. We call the elements of σ the *terms* of σ; accordingly, a term of σ is specified by a set featured in θ and by its position in σ. The *frame* of θ is the union of the sets featured in θ, and the *key* of θ is the first term of σ. For example, the base sequence, the frame and the key of the compositional expression $(AB \rhd AC) \rhd ((BC \rhd AB) \rhd CD)$ are $\langle AB, AC, BC, AB, CD\rangle$, $ABCD$ and AB, respectively.

Note that, unlike in [27], we assume that each set featured in a compositional expression can have more than one occurrence. A compositional expression θ *contains no repetitions* if each set featured in θ has exactly one occurrence.

A *subexpression* of a compositional expression θ is defined as usual. Explicitly, a compositional expression θ' is a subexpression of θ if θ' is a substring of θ. Let $\sigma' = \langle X_i, \ldots, X_q\rangle$ be the base sequence of θ' for some i and q, $1 \le i \le q \le n$. We say that θ' is an *atomic subexpression* of θ if $i = q$; thus, a non-atomic subexpression of θ is always of the type $(\theta_1) \rhd (\theta_2)$. Note that, if a set X appears k times in θ, then there are exactly k atomic subexpressions $\theta_1, \ldots, \theta_k$ of θ and $\theta_h = X$ for all $h = 1, \ldots, k$.

Henceforth, a subexpression of θ of the type $(X) \rhd (\theta')$ or $(\theta') \rhd (X)$ or $(X) \rhd (Y)$ will be written simply as $X \rhd (\theta')$ or $(\theta') \rhd X$ or $X \rhd Y$, respectively.

The syntactic structure of a compositional expression θ can be represented by an ordered binary tree $\mathcal{T}$ [1], called the *syntax tree* for θ [27], whose leaves correspond one-to-one to the atomic subexpressions of θ, and whose interior nodes correspond one-to-one to the non-atomic subexpressions of θ. Thus, each interior node v of $\mathcal{T}$ has exactly two ordered children: its first (respectively, second) child is called the *left* (respectively, *right*) child. If an interior node v has left child u and right child w, the subexpression of θ corresponding to v is $(\theta_u) \rhd (\theta_w)$ where θ_u and θ_w are the subexpressions of θ corresponding to u and w respectively. The node corresponding to θ is called the *root* of $\mathcal{T}$ and, henceforth, we assume that arcs of $\mathcal{T}$ are oriented away from the root. For example, the syntax tree for the compositional expression

$$(ABC \rhd (BD \rhd BE)) \rhd (ADF \rhd FG)$$

is shown in Figure 1. Finally, by the *depth* of a node v of $\mathcal{T}$ we mean the length of the (unique) path from the root of $\mathcal{T}$ to v.

Remark 4.1. Let θ be a compositional expression whose base sequence has length n. Using the same arguments as in [27] for compositional expressions containing no repetitions, one can prove that the syntax tree for θ has exactly $n - 1$ interior nodes and, hence, $2n - 1$ nodes.

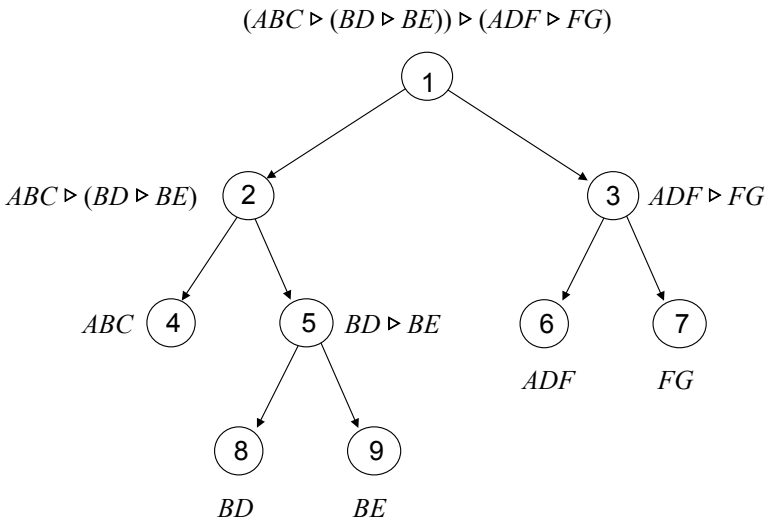

Fig. 1. The syntax tree for the compositional expression $(ABC \triangleright (BD \triangleright BE)) \triangleright (ADF \triangleright FG)$.

4.1. The value of a compositional expression

Let θ be a compositional expression with base sequence $\sigma = \langle X_1, \ldots, X_n \rangle$. An *interpretation* of θ over a given semifield Σ (a Σ-*interpretation* of θ, for short) is a sequence $I = \langle f_1(X_1), \ldots, f_n(X_n) \rangle$ of Σ-distributions. Note that even if $X_i = X_j$ for $i \neq j$, $f_i(X_i)$ and $f_j(X_j)$ may be distinct.

Let θ' be any subexpression of θ and let $\sigma' = \langle X_i, \ldots, X_q \rangle$, $1 \leq i \leq q \leq n$, be the base sequence of θ'. The evaluation of θ' under I consists in replacing each set X_j, $i \leq j \leq q$, with the corresponding distribution $f_j(X_j)$ in I, and then applying the composition operator if θ' is a non-atomic subexpression (that is, if $q > i$). The result will be referred to as the *value* of θ' under I. If the value of θ' under I is defined, then it is a Σ-distribution, denoted by $[\theta']_I$, whose scheme is precisely the frame of θ'. For $\theta' = \theta$, we obtain the value of θ under I; if it is defined, then we call I a *valid* Σ-*interpretation* of θ.

The *evaluation operator associated* with θ *over* Σ [27] is the function mapping valid Σ-interpretations I of θ to values of θ under I. Accordingly, we say that the *model generated* by θ *fits* a Σ-distribution $f(X)$, where X is the scheme of θ, if $f(X)$ belongs to the image of the evaluation operator associated with θ over Σ, that is, if there exists a valid Σ-interpretation I of θ such that $f(X)$ equals the value of θ under I.

The following is a straightforward consequence of part (i) of Theorem 3.2.

Lemma 4.2. Let θ be a compositional expression, and let $\theta' = (\theta_1) \triangleright (\theta_2)$ be a (non-atomic) subexpression of θ. If I is a valid Σ-interpretation of θ, then $[\theta_1]_I$ is the marginal of $[\theta']_I$ on the frame of θ_1.

We shall state a result more general than Lemma 4.2. Let $\mathcal{T}$ be the syntax tree for θ, and let v be a node of $\mathcal{T}$ for θ. Consider the subtree $\mathcal{T}_v$ of $\mathcal{T}$ rooted at v. We call the *leftmost branch* of $\mathcal{T}_v$ the set of nodes that is recursively defined as follows:

— v belongs to the leftmost branch of $\mathcal{T}_v$;

— if u belongs to the leftmost branch of $\mathcal{T}_v$ and u is not a leaf of $\mathcal{T}_v$, then the left child of u belongs to the leftmost branch of $\mathcal{T}_v$.

In what follows, the deepest node in the leftmost branch of $\mathcal{T}_v$ will be referred to as the *leftmost leaf* of $\mathcal{T}_v$. Note that the leftmost branch of $\mathcal{T}_v$ is formed by the nodes of $\mathcal{T}_v$ that are ancestors of the leftmost leaf of $\mathcal{T}_v$ (that is, by the leftmost leaf of $\mathcal{T}_v$ and by the nodes of $\mathcal{T}_v$ that are its proper ancestors). Thus, if v is the root of $\mathcal{T}$, then $\mathcal{T}_v = \mathcal{T}$ and the leftmost leaf of $\mathcal{T}$ corresponds to the first occurrence of the key of θ. For example, the leftmost node of the syntax tree $\mathcal{T}$ shown in Figure 1 is node 4, and the leftmost branch of $\mathcal{T}$ is $\{1, 2, 4\}$.

The following theorem generalizes Lemma 4.2.

Theorem 4.3. Let θ be a compositional expression, and let $\mathcal{T}$ be the syntax tree for θ. Let v be an interior node of $\mathcal{T}$, and let u be any node in the leftmost branch of $\mathcal{T}_v$. Let θ_v and θ_u be the subexpressions of θ corresponding to v and u, respectively. If I is a valid Σ-interpretation of θ, then $[\theta_u]_I$ is the marginal of $[\theta_v]_I$ on the frame of θ_u.

P r o o f. Along the path from v to u, we repeatedly apply Lemma 4.2 from each node to its left child. □

In the case that v is the root of $\mathcal{T}$, Theorem 4.3 states that, for every node u in the leftmost branch of $\mathcal{T}$, the value of the subexpression θ_u of θ corresponding to u is a marginal of the value of θ under I.

4.2. A general validity test

Let θ be a compositional expression with base sequence $\sigma = \langle X_1, \ldots, X_n \rangle$, and let $I = \langle f_1(X_1), \ldots, f_n(X_n) \rangle$ be a Σ-interpretation of θ, where Σ is any semifield. A procedure for deciding whether I is or is not a valid Σ-interpretation of θ was given in [26]. The procedure keeps up with the numeric computation of the values of the subexpressions of θ under I during a traversal of the syntax tree $\mathcal{T}$ for θ according to the *postorder scheme* [1], that is, we always visit an interior node after visiting its children (first its left child and afterwards its right one) — see Figure 2. Moreover, each node v of $\mathcal{T}$ has three "attributes":

- a set of variables, denoted by L_v and called the *label* of v, which stands for the frame of the subexpression θ_v of θ corresponding to v;
- a distribution with scheme L_v, denoted by $F_v(L_v)$, which stands for the value of θ_v under I;
- a relation with scheme L_v, denoted by r_v, which stands for the support of $F_v(L_v)$.

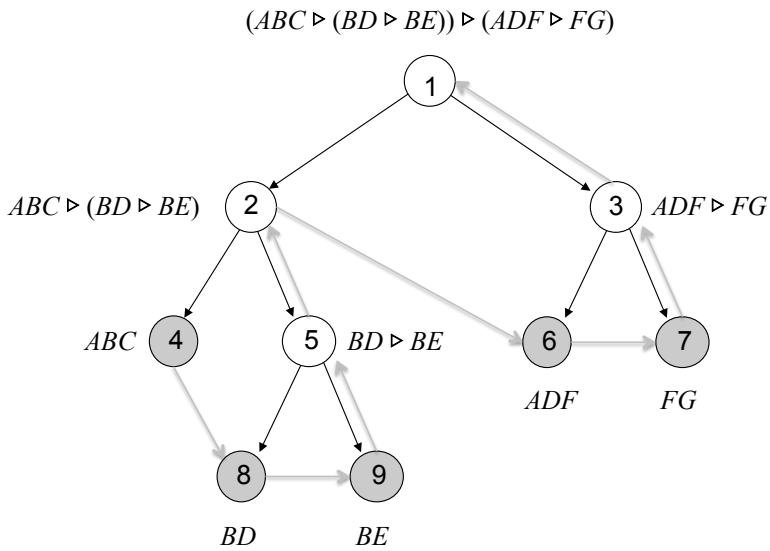

Fig. 2. The postorder traversal of the syntax tree of Fig. 1.

Initially, the three attributes are defined only for the leaves of $\mathcal{T}$; explicitly, if v is a leaf corresponding to the atomic subexpression X_i for some i, then

$$L_v = X_i \qquad F_v(L_v) = f_i(X_i) \qquad r_v = \|f_i\|\,.$$

During the postorder traversal of $\mathcal{T}$, when we visit an interior node v having left child u and right child w, from Theorem 3.1 we know the value of the subexpression θ_v corresponding to v is defined if and only if

(i) either r_u is an empty relation, or

(ii) if $L_u \cap L_w = \emptyset$ then $F_w^{\downarrow\emptyset} \neq 0$; otherwise, $\pi_{L_u \cap L_w}(r_u) \subseteq \|F_w{}^{\downarrow L_u \cap L_w}\|$.

Therefore, in order to check whether the value of θ_v is defined, we need to compute $F_w{}^{\downarrow L_u \cap L_w}$; moreover, if $L_u \cap L_w \neq \emptyset$, then we need to compute its support $\|F_w{}^{\downarrow L_u \cap L_w}\|$ and, finally, check the inclusion $\pi_{L_u \cap L_w}(r_u) \subseteq \|F_w{}^{\downarrow L_u \cap L_w}\|$. If the value of θ_v is not defined, then we stop the traversal of $\mathcal{T}$; otherwise, we set

- $L_v := L_u \cup L_w$;
- $F_v(L_v) := F_u(L_u) \rhd F_w(L_w)$;
- $r_v := r_u \bowtie r_w$ (by part (ii) of Theorem 3.2).

Finally, I is a valid Σ-interpretation of θ if and only if the value of the subexpression corresponding to the root of $\mathcal{T}$ (that is, the value of θ) is defined.

4.3. A metric validity test

Consider now the case that Σ is a *metric semifield* (for example, Σ is the sum-product semifield or a tropical semifield or the Boolean algebra). As was noted in [26] (page 12), one can test a Σ-interpretation I of θ for validity without computing the distributions associated with interior nodes of the syntax tree $\mathcal{T}$ since only their supports are needed. Thus, no numeric computation is needed and only set operations are executed. We shall see the importance of this result in Sections 6 and 7.

For the sake of completeness, we now give some details of the simplified version of the validity test, which will be referred to as the *metric validity test*. Suppose that, during the postorder traversal of $\mathcal{T}$, we visit an interior node v with left child u and right child w. By Theorem 3.4, the value of the subexpression θ_v of θ corresponding to v is defined if and only if

(i) either r_u is the empty relation, or

(ii) if $L_u \cap L_w = \emptyset$ then r_w is a non-empty relation; otherwise, $\pi_{L_u \cap L_w}(r_u) \subseteq \pi_{L_u \cap L_w}(r_w)$.

If this is the case then we set $L_v := L_u \cup L_w$ and $r_v := r_u \bowtie r_w$; otherwise, we stop the traversal of $\mathcal{T}$ and conclude that I is not a valid Σ-interpretation of θ.

From the foregoing it follows that the metric validity test works with the syntax tree $\mathcal{T}$ where each node v has only two "attributes":

- the label L_v (which stands for the frame of the subexpression θ_v of θ corresponding to v), and
- the relation r_v (which stands for the support of the value of θ_v under I).

The following is an illustrative example.

Example 4.1. Consider the compositional expression

$$\theta = (ABC \rhd (BD \rhd BE)) \rhd (ADF \rhd FG)$$

whose syntax tree $\mathcal{T}$ was shown in Figure 1. Let

$$I = \langle f(ABC), g(BD), h(BE), k(ADF), l(FG) \rangle$$

be a valid Σ-interpretation of θ, where Σ is any metric semifield. After the postorder traversal of $\mathcal{T}$ (see Figure 2), we obtain

$$\begin{array}{ll}
L_4 = ABC & r_4 = \|f\| \\
L_8 = BD & r_8 = \|g\| \\
L_9 = BE & r_9 = \|h\| \\
L_5 = BDE & r_5 = r_8 \bowtie r_9 \,(= \|g\| \bowtie \|h\|) \\
L_2 = ABCDE & r_2 = r_4 \bowtie r_5 \,(= \|f\| \bowtie \|g\| \bowtie \|h\|) \\
L_6 = ADF & r_6 = \|k\| \\
L_7 = FG & r_7 = \|l\| \\
L_3 = ADFG & r_3 = r_6 \bowtie r_7 \,(= \|k\| \bowtie \|l\|) \\
L_1 = ABCDEFG & r_1 = r_2 \bowtie r_3 \,(= \|f\| \bowtie \|g\| \bowtie \|h\| \bowtie \|k\| \bowtie \|l\|)\,.
\end{array}$$

5. MARGINALIZATION PROBLEMS

Consider the following two marginalization problems where Σ is any semifield.

Single-marginal problem: Given a compositional expression θ, a valid Σ-interpretation I of θ and a subset Y of the frame of θ, compute the marginal on Y of the value of θ under I.

Marginal-representation problem: Given a compositional expression θ and a valid Σ-interpretation I of θ, compute the marginals of the value of θ under I for all sets featured in θ.

Since I is required to be a valid Σ-interpretation of θ, we present two distinct methods depending on whether we want to solve the two marginalization problems above from scratch or from the output of the validity test mentioned in Subsection 4.2. In the rest of this section we discuss the latter case; the former case (an example was given in the Introduction) will be discussed in Sections 6 and 7. So, we start with the syntax tree $\mathcal{T}$ for θ where at each node v the following information is stored:

- the label L_v (which stands for the frame of the subexpression θ_v of θ corresponding to v),
- the distribution $F_v(L_v)$ (which stands for the value of θ_v under I),
- the relation r_v (which stands for the support of $F_v(L_v)$).

For convenience, we assume that the algebraic operations are the ordinary addition ($+$) and multiplication ($\times$).

5.1. The single-marginal problem

We can solve the single-marginal problem by exploiting Theorem 4.3, which implies that, for each node v in the leftmost branch of $\mathcal{T}$, $F_v(L_v)$ equals the marginal of $[\theta]_I$ on L_v. It follows that, if v is a node in the leftmost branch of $\mathcal{T}$ and $Y \subseteq L_v$, then the marginal of $[\theta]_I$ on Y, written $m(Y)$, can be obtained by marginalizing $F_v(L_v)$ on Y. In order to minimize the number of additions, we will choose the deepest node v in the leftmost branch of $\mathcal{T}$ such that $Y \subseteq L_v$. (Note such a node always exists because Y is a subset of the label of the root of $\mathcal{T}$.) This node, which we call the *node covering* Y, can be found by examinining the nodes in the leftmost branch of $\mathcal{T}$ either top-down or bottom-up. Let v be the node covering Y. By Theorem 4.3 we have that $m(Y) = F_v^{\downarrow Y}$ and, by Lemma 2.2, the support of $m(Y)$ is a subset of the projection on Y of the relation r_v ($= \|F_v\|$) stored at v, that is, $\|m\| \subseteq \pi_Y(r_v)$. Therefore, for every Y-tuple $\mathbf{y}$, since $m(\mathbf{y}) = 0$ for every Y-tuple $\mathbf{y} \notin \|m\|$, we can compute $m(\mathbf{y})$ as follows:

$$m(\mathbf{y}) := \begin{cases} 0 & \text{if } \mathbf{y} \notin \pi_Y(r_v) \\ \sum_{\mathbf{v} \in r_v : \mathbf{v}_Y = \mathbf{y}} F_v(\mathbf{v}) & \text{otherwise.} \end{cases}$$

So, solving the single-marginal problem requires a number of additions equal to the size $|r_v|$ of the relation r_v stored at v. We now give an illustrative example.

Example 5.1. Consider our compositional expression

$$\theta = (ABC \rhd (BD \rhd BE)) \rhd (ADF \rhd FG)$$

and let $I = \langle f(ABC), g(BD), h(BE), k(ADF), l(FG)\rangle$ be a valid Σ-interpretation of θ, where Σ is the real field. We want to compute the marginal $m(ACD)$ of the value of θ under I.

The syntax tree $\mathcal{T}$ for θ was shown in Figure 1. Recall that the leftmost branch of $\mathcal{T}$ is $\{1, 2, 4\}$ so that the node covering ACD is node 2 ($L_2 = ABCDE$). Then, we first compute the relation $\pi_{ACD}(r_2)$, where $r_2 = \|f\| \bowtie \|g\| \bowtie \|h\|$ (see Example 4.1). Next, for every ACD-tuple $(\mathbf{a}, \mathbf{c}, \mathbf{d})$, we set

$$m(\mathbf{a}, \mathbf{c}, \mathbf{d}) := \begin{cases} 0 & \text{if } (\mathbf{a}, \mathbf{c}, \mathbf{d}) \notin \pi_{ACD}(r_2) \\ \sum_{(\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d},\mathbf{e})\in r_2} F_2(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}) & \text{otherwise.} \end{cases}$$

So, computing $m(ACD)$ requires a number of additions equal to $|r_2|$, that is, to the size of the relation $\|f\| \bowtie \|g\| \bowtie \|h\|$.

5.2. The marginal-representation problem

We can solve the marginal-representation problem as follows. Let σ be the base sequence of θ. First of all, we reduce σ by keeping only the first occurrences of sets featured in θ. Let α be the resulting sequence. We also create a list β containing the nodes in the leftmost branch of $\mathcal{T}$ ordered by decreasing depth (that is, from the leftmost leaf to the root). Note that the key of θ is both the first term of α and the label of the first node in β. At this point, we run the following procedure, which will be referred to as the *α-β procedure*.

Until α is empty, repeat:

Step 1 Take the first set in α, denote it by X, and take the first node in β, denote it by v.

Step 2 If X is not a subset of L_v, then delete v from β; otherwise,

(2.1) compute the marginal of $F_v(L_v)$ on X;

(2.2) delete X from α.

In order to reduce the number of additions, we propose the following graphical implementation of the α-β procedure. For each set X in α, we "mark" the leaf of $\mathcal{T}$ that corresponds to the first occurrence of X in θ. Initially, we examine the first node in β, say v, and set the marginal on L_v equal to the corresponding distribution in I. For each other node v in β, we consider the subtree $\mathcal{T}_w$ of $\mathcal{T}$, where w is the right child of v, and repeatedly delete unmarked leaves of $\mathcal{T}_w$. Let $\mathcal{T}'_w$ be the residual of $\mathcal{T}_w$. If $\mathcal{T}'_w$ is not empty, then do (see Figure 3):

— Compute the marginal of $F_v(L_v)$ on L_w; denote it by $m_w(L_w)$.

— For each arc $x \to y$ of $\mathcal{T}'_w$, compute the marginal of $m_x(L_x)$ on L_y.

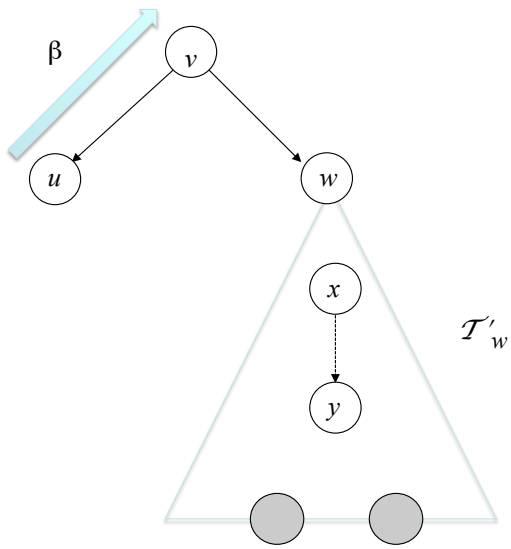

Fig. 3. The top-down traversal of the subtree $\mathcal{T}'_w$.

Thus, after a top-down traversal of $\mathcal{T}'_w$, we compute the marginals of $F_v(L_v)$ on the labels of marked leaves of $\mathcal{T}'_w$. By Theorem 4.3, these marginals are precisely marginals of the value of θ under I. Finally, we stop scanning β after visiting the node covering the frame of θ.

We shall most likely be better off using the graphical implementation of the α-β procedure if each subtree such as $\mathcal{T}'_w$ has relatively many marked leaves with respect to the total number of its nodes. The following is an illustrative example.

Example 5.2. Consider our compositional expression

$$\theta = (ABC \rhd (BD \rhd BE)) \rhd (ADF \rhd FG)$$

and let $I = \langle f(ABC), g(BD), h(BE), k(ADF), l(FG)\rangle$ be a valid Σ-interpretation of θ, where Σ is the real field. We want to compute the marginals of the value of θ under I on the sets ABC, BD, BE, ADF, FG. Recall that the leftmost branch of the syntax tree $\mathcal{T}$ is $\{1, 2, 4\}$ (see Figure 1). We first apply the α-β procedure, and then its graphical implementation.

(α-β *procedure*) We first create the two lists

$$\alpha = \langle ABC, BD, BE, ADF, FG\rangle \qquad \beta = \langle 4, 2, 1\rangle\,.$$

When we examine the first node 4 of β, we

- set the wanted marginal on ABC equal to $F_4(ABC) = f(ABC)$, and

- delete ABC from α which becomes $\langle BD, BE, ADF, FG\rangle$. Since BD is not a subset of $L_4 = ABC$, we delete the node 4 from β which becomes $\langle 2, 1\rangle$.

When we examine node 2, we

- compute the wanted marginals on both BD and BE by marginalizing $F_2(ABCDE)$, which requires $2\,|r_2|$ additions, and
- delete BD and BE from α which becomes $\langle ADF, FG\rangle$. Since ADF is not a subset of $L_2 = ABCDE$, we delete the node 2 from β which becomes $\langle 1\rangle$.

When we examine node 1, we

- compute the wanted marginals on both ADF and FG by marginalizing $F_1(ABCDEFG)$, which requires $2\,|r_1|$ additions, and
- delete ADF and FG from α which becomes empty.

So, the total number of additions is $2(|r_1| + |r_2|)$.

(*Graphical implementation of the α-β procedure*) We first mark all the five leaves of $\mathcal{T}$ (see Figure 4), and create the list $\beta = \langle 4, 2, 1\rangle$. Note that the node covering the frame of θ is the root (node 1) of $\mathcal{T}$.

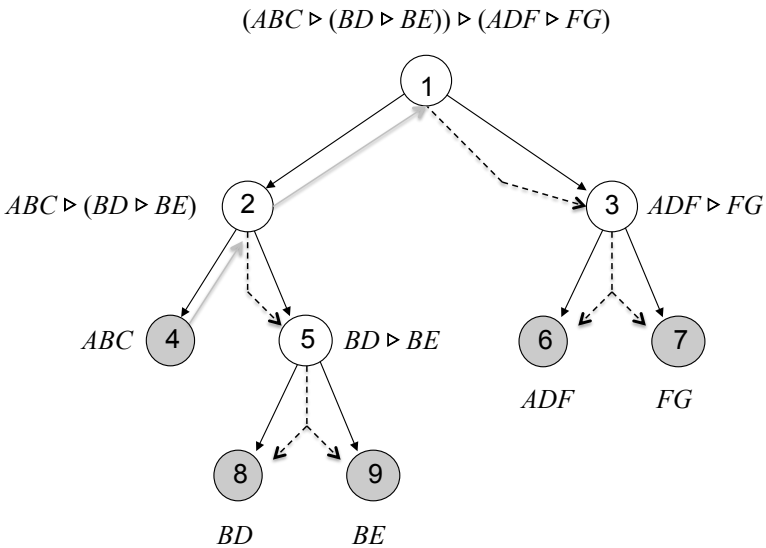

Fig. 4. The traversal of the syntax tree with the graphical implementation.

When we examine node 4, we set the wanted marginal $m_4(ABC)$ equal to $F_4(ABC)$ $(= f(ABC))$.

When we examine node 2, we compute

- the marginal $m_5(BDE)$ of $F_2(ABCDE)$, and
- the wanted marginals $m_8(BD)$ and $m_9(BE)$ by marginalizing $m_5(BDE)$

which requires $|r_2| + 2\,|\pi_{BDE}(r_2)|$ additions.

When we examine node 1, we compute

- the marginal $m_3(ADFG)$ of $F_1(ABCDEFG)$, and
- the wanted marginals $m_6(ADF)$ and $m_7(FG)$ by marginalizing $m_3(ADFG)$

which requires $|r_1| + 2\,|\pi_{ADFG}(r_1)|$ additions.

So, the total number of additions is $|r_1| + |r_2| + 2(|\pi_{ADFG}(r_1)| + |\pi_{BDE}(r_2)|)$.

Therefore, the graphical implementation of the α-β procedure is convenient if

$$2(|\pi_{ADFG}(r_1)| + |\pi_{BDE}(r_2)|) < |r_1| + |r_2| \ .$$

6. THE SINGLE-MARGINAL PROBLEM FROM SCRATCH

Consider again the single-marginal problem stated in Section 5:

Given a subset Y of the frame of θ, we want to compute the marginal $m(Y)$ of the value of θ under a valid Σ-interpretation I of θ.

We want to solve it "from scratch", that is, by taking as input the syntax tree $\mathcal{T}$ for θ where the three attributes (label, distribution, relation) are given only for the leaves of $\mathcal{T}$. This case is of special interest when Σ is a metric semifield, since we can use the metric validity test (see Subsection 4.3) which does not compute the values of the subexpressions of θ corresponding to interior nodes of $\mathcal{T}$.

As in Section 5 we assume that the algebraic operations are the ordinary addition ($+$) and multiplication ($\times$). (If Σ is a tropical semifield, we only need a change of notation for addition, multiplication and division). The case that Σ is the Boolean algebra will be discussed separately in Section 8.)

Since the values of the subexpressions of θ corresponding to interior nodes of $\mathcal{T}$ are unknown, we cannot get $m(Y)$ by marginalizing $[\theta_v]_I$ on Y, where v is the node covering Y, as we did in Subsection 5.1. Instead, we first construct an algebraic expression $\mathtt{M}$ for $m(Y)$ and, then, we evaluate $\mathtt{M}$ using the numeric values of the Σ-distributions in I. In order to construct the algebraic expression $\mathtt{M}$, we perform a postorder traversal of $\mathcal{T}$ and stop after visiting the node covering Y. During the traversal of $\mathcal{T}$, when a node v is visited, we construct an algebraic expression $\mathtt{E}_v$ for the (unknown) value of θ_v under I, where θ_v is the subexpression of θ corresponding to v. Moreover, if v is an interior node with left child u and right child w, we set $L_v := L_u \cup L_w$. Finally, after visiting the node covering Y, say v, we take $\mathtt{M}$ to be the "reduced form" (see below) of the sum

$$\sum_{A \in L_v \setminus Y} \mathtt{E}_v \ . \tag{1}$$

6.1. Symbolic computation

Let $I = \langle f_1(X_1), \ldots, f_n(X_n)\rangle$ be a (valid) Σ-interpretation of θ. We now show how algebraic expressions $\mathtt{E}_v$ and $\mathtt{M}$ over I are constructed. To this end, we need some more definitions.

Algebraic expressions over I view each f_i as a symbolic name; they are defined recursively as follows:

— each $f_i(X_i)$ is an algebraic expression and its scheme is X_i;

— if $\mathtt{E}$ and $\mathtt{E}'$ are algebraic expressions with schemes S and S' respectively, then $(\mathtt{E}) \times (\mathtt{E}')$ is an algebraic expression and its scheme is $S \cup S'$;

— if $\mathtt{E}$ is an algebraic expression with scheme S, and if R is a proper subset of S, then $\sum_{A\in S\setminus R}(\mathtt{E})$ is an algebraic expression and its scheme is R;

— if $\mathtt{E}$ is an algebraic expression with scheme S, then $\frac{1}{\mathtt{E}}$ is an algebraic expression and its scheme is S.

Henceforth, we make a parsimonious use of parentheses; moreover, we abridge an algebraic expression such as $\sum_{A\in X_i\setminus Z} f_i(X_i)$ to $f_i^{\downarrow Z}$.

An algebraic expression is *factorable* if, from a formal point of view, it can be written as a product of two or more algebraic expressions, and *non-factorable* otherwise. The *factors* of an algebraic expression $\mathtt{E}$ are non-factorable algebraic expressions $\mathtt{F}_1, \ldots, \mathtt{F}_q$, $q \geq 1$, such that $\mathtt{E}$ can be written as $\mathtt{E} = \mathtt{F}_1 \times \ldots \times \mathtt{F}_q$. For example, the factors of the algebraic expression

$$\frac{k(ADF) \times l(FG)}{k^{\downarrow AD} \times l^{\downarrow F}} \times \sum_B \frac{f^{\downarrow AB} \times g(BD)}{g^{\downarrow B}}$$

are

$$k(ADF) \qquad l(FG) \qquad \sum_B \frac{f^{\downarrow AB} \times g(BD)}{g^{\downarrow B}} \qquad \frac{1}{k^{\downarrow AD}} \qquad \frac{1}{l^{\downarrow F}}\,.$$

From a computational point of view, the factors of an algebraic expression can be identified during its syntactical analysis (parsing).

6.1.1. COMPUTING THE ALGEBRAIC EXPRESSION $\mathtt{E}_v$

When a node v is visited during the postorder traversal of $\mathcal{T}$, $\mathtt{E}_v$ is constructed as follows. Let us distinguish two cases depending on whether v is a leaf or an interior node.

Case 1: v is a leaf. If $L_v = X_i$ for some i, then we set $\mathtt{E}_v := f_i(X_i)$.

Case 2: v is an interior node of $\mathcal{T}$ with left child u and right child w. Recall that $L_v = L_u \cup L_w$. Then, we take $\mathtt{E}_v$ to be the "reduced form" of the product

$$\frac{\mathtt{E}_u \times \mathtt{E}_w}{\sum\limits_{A\in L_w\setminus L_u} \mathtt{E}_w}\,, \tag{2}$$

which is obtained as follows. If $L_v = L_u$ (that is, if $L_w \subseteq L_u$) then we soon set $\mathtt{E}_v := \mathtt{E}_u$; otherwise, using suitable reduction rules (see below), we first simplify the sum

$$\sum_{A \in L_w \setminus L_u} \mathtt{E}_w \tag{3}$$

and, then, simplify the product

$$\frac{\mathtt{E}_u \times \mathtt{E}_w}{\mathtt{e}}\,, \tag{4}$$

where $\mathtt{e}$ is the result of the reduction of the sum (3). The result of the reduction of the product (4) will give $\mathtt{E}_v$.

We now detail the procedure for reducing the sum (3) and, then, the procedure for reducing the product (4).

Reduction of the sum (3). By Theorem 4.3, reducing the sum $\sum_{A \in L_w \setminus L_u} \mathtt{E}_w$ is equivalent to reducing the sum $\sum_{A \in L_z \setminus L_u} \mathtt{E}_z$ where z is the deepest node belonging to the leftmost branch of the subtree $\mathcal{T}_w$ such that $L_u \cap L_w \subseteq L_z$. After finding z, we perform the following steps:

Step 1. We first find the factors of $\mathtt{E}_z$, say $\mathtt{F}_1, \ldots, \mathtt{F}_q$, $q \geq 1$; thus, $\mathtt{E}_z$ can be written as

$$\mathtt{F}_1 \times \ldots \times \mathtt{F}_q\,. \tag{5}$$

Let S_i be the scheme of $\mathtt{F}_i$, $1 \leq i \leq q$; thus, $L_z = \cup_{1 \leq i \leq m} S_i$. Let us construct an (undirected) graph $\mathcal{G}$ with node set $Q = \{1, \ldots, q\}$, where node i stands for the factor $\mathtt{E}_i$ and is labeled by S_i, and two nodes i and j, $i \neq j$, are joined by an edge if their labels have a nonempty intersection, that is, if $S_i \cap S_j \neq \emptyset$. For each edge (i, j) of $\mathcal{G}$, we label (i, j) by the (nonempty) set $S_i \cap S_j$. Let $\mathcal{G}_1, \ldots, \mathcal{G}_k$ be the connected components of the subgraph of $\mathcal{G}$ resulting from the deletion of the edges of $\mathcal{G}$ that are labeled by subsets of L_u (or, equivalently, by subsets of $L_u \cap L_z$). Let Q_h be the node set of $\mathcal{G}_h$, and let $Z_h = \cup_{i \in Q_h} S_i$, $1 \leq h \leq k$. Note that, by construction, for $h \neq l$ one has that $Z_h \cap Z_l \subseteq L_u$ and, hence, $(Z_h \setminus L_u) \cap (Z_l \setminus L_u) =$ *emptyset*; moreover, it may happen that $Z_h \subseteq L_u$ for some h and, if this is the case, then Q_h is a singleton.

By the associativity and commutative properties of $\times$, we re-write (5) as

$$\Big(\prod_{i \in Q_1} \mathtt{F}_i \Big) \times \ldots \times \Big(\prod_{i \in Q_k} \mathtt{F}_i \Big)$$

and re-write (3) as

$$\sum_{A \in L_w \setminus L_u} \left(\Big(\prod_{i \in Q_1} \mathtt{F}_i \Big) \times \ldots \times \Big(\prod_{i \in Q_k} \mathtt{F}_i \Big) \right). \tag{6}$$

By exploiting the fact that $(Z_h \setminus L_u) \cap (Z_l \setminus L_u) = \emptyset$ for $h \neq l$, we re-write (6) as

$$\left(\sum_{A \in Z_1 \setminus L_u} \prod_{i \in Q_1} \mathtt{F}_i \right) \times \ldots \times \left(\sum_{A \in Z_k \setminus L_u} \prod_{i \in Q_k} \mathtt{F}_i \right). \tag{7}$$

Note that, if $Z_h \subseteq L_u$ for some h, then Q_h is a singleton and, if $Q_h = \{j\}$, then $\prod_{i \in Q_h} \mathtt{F}_i$ is nothing but $\mathtt{F}_j$ and $\sum_{A \in Z_h \setminus L_u} \prod_{i \in Q_h} \mathtt{F}_i$ is simply $\mathtt{F}_j$.

Step 2. For each h, $1 \leq h \leq k$, we reduce

$$\sum_{A \in Z_h \setminus L_u} \prod_{i \in Q_h} \mathtt{F}_i \,. \tag{8}$$

The result of the reduction, which we denote by $\mathtt{e}_h$, is obtained as follows. Let us distinguish two cases depending on whether or not $Z_h \subseteq L_u$.

Case 1: $Z_h \subseteq L_u$. In this case, as we saw above, if $Q_h = \{j\}$ then, since $\sum_{A \in Z_h \setminus L_u} \prod_{i \in Q_h} \mathtt{F}_i$ is simply $\mathtt{F}_j$, we set $\mathtt{e}_h := \mathtt{F}_j$.

Case 2: $Z_h \setminus L_u \neq \emptyset$. In this case, we repeatedly apply the following three operations until they cannot be longer applied:

(*delete*) If a variable $B \in Z_h \setminus L_u$ belongs to the scheme S_j of exactly one $\mathtt{F}_j$, $j \in Q_h$, then delete B from Z_h, and replace $\mathtt{F}_j$ with $\sum_B \mathtt{F}_j$. In the special case that $\mathtt{F}_j = f_i(X_i)$ for some i, $1 \leq i \leq n$, replace $\mathtt{F}_j$ with $f_i^{\downarrow X_i \setminus \{B\}}$.

(*cancel*) If there exist $i, j \in Q_h$, $i \neq j$, such that $\mathtt{F}_j$ is the multiplicative inverse of $\mathtt{F}_i$ (that is, $\mathtt{F}_j = \frac{1}{\mathtt{F}_i}$), then cancel both $\mathtt{F}_i$ and $\mathtt{F}_j$, that is, delete both i and j from Q_h.

(*factor out*) If there exists $j \in Q_h$ such that $S_j \subseteq L_u$, then move $\mathtt{F}_j$ to the left side of $\sum_{A \in Z_h \setminus L_u}$, and delete j from q_h.

Step 3. Set $\mathtt{e} := \mathtt{e}_1 \times \ldots \times \mathtt{e}_k$.

Reduction of the product (4). We first find the factors of the product $\frac{\mathtt{E}_u \times \mathtt{E}_w}{\mathtt{e}}$. Then, we cancel every pair of factors one being the multiplicative inverse of the other. The result of the reduction will give $\mathtt{E}_v$.

6.1.2. COMPUTING THE ALGEBRAIC EXPRESSION M

Let v be the node covering Y. In order to get $\mathtt{M}$ we need to reduce the sum (1), which can be done, *mutatis mutandis*, by performing Steps 1-3 (see above). The result of the reduction will give $\mathtt{M}$.

6.2. Numeric computation

Suppose that we have performed the postorder traversal of the subtree of $\mathcal{T}$ rooted at the node v covering Y and have reduced the sum $\sum_{A \in L_v \setminus Y} \mathtt{E}_v$ to obtain the algebraic expression $\mathtt{M}$ of $m(Y)$. At this point, what remains to do is the numeric evaluation of $\mathtt{M}$. We could reduce the amount of numeric computation if we knew the support $\|m\|$ of $m(Y)$, since $m(\mathbf{y}) = 0$ for every Y-tuple $\mathbf{y} \notin \|m\|$. But, in the general case we don't know $\|m\|$; however, we can find a superset of $\|m\|$ as follows. Let θ_v be the subexpression of θ corresponding to v. By Theorem 4.3, the value of θ_v under I, that is, $[\theta_v]_I$, is the marginal on L_v of the value of θ under I, and, hence, $m(Y)$ is the marginal of $[\theta_v]_I$ on Y. By Lemma 2.2, $\|m\| \subseteq \pi_Y(\|[\theta_v]_I\|)$ where the equality holds if Σ is a metric semifield (by Lemma 2.3). On the other hand, by part (ii) of Theorem 3.2, the support $\|[\theta_v]_I\|$ of $[\theta_v]_I$ is given by the join of the relations stored at the leaves of the subtree $\mathcal{T}_v$. So, after computing $\|[\theta_v]_I\|$ and, then, its projection $\pi_Y(\|[\theta_v]_I\|)$ on Y, for every Y-tuple $\mathbf{y}$ we can compute $m(\mathbf{y})$ as follows:

> if $\mathbf{y} \notin \pi_Y(\|[\theta_v]_I\|)$, then set $m(\mathbf{y}) := 0$; otherwise, compute $m(\mathbf{y})$ by evaluating the algebraic expression $\mathtt{M}$ using the numeric values of the Σ-distributions in I.

6.3. The marginalization procedure

From the foregoing it follows that the single-marginal problem can be solved using the following procedure, where we make use of a Boolean variable $lb(v)$ which will be `true` if and only if v belongs to the leftmost branch of $\mathcal{T}$. Initially, for each node of $\mathcal{T}$, we set $lb(v)$ to `true` if v is the leftmost leaf of $\mathcal{T}$, and to `false` otherwise. Then, we order the nodes of $\mathcal{T}$ according to the postorder scheme and, for each node v, we perform the following two steps:

Step 1. Let us distinguish the following two cases:

> *Case* 1: v is a leaf. Set $\mathtt{E}_v := f_i(X_i)$ for that i, for which $L_v = X_i$.
>
> *Case* 2: v is an interior node of $\mathcal{T}$. Let u and w be the left child and the right child of v, respectively.
>
> If $lb(u) = \mathtt{true}$, then set $lb(v) := \mathtt{true}$.
>
> Set $L_v := L_u \cup L_w$. If $L_v = L_u$ (that is, if $L_w \subseteq L_u$) then set $\mathtt{E}_v := \mathtt{E}_u$; otherwise, construct the reduction $\mathtt{e}$ of the sum $\sum_{A \in L_w \setminus L_u} \mathtt{E}_w$ and, then, take $\mathtt{E}_v$ to be the reduction of the product $\frac{\mathtt{E}_u \times \mathtt{E}_w}{\mathtt{e}}$.

Step 2. If $lb(v) = \mathtt{true}$ and $Y \subseteq L_v$ (v is the node covering Y), then do:

> (2.1) Set $\mathtt{M}$ to the reduction of the sum $\sum_{A \in L_v \setminus Y} \mathtt{E}_v$.
>
> (2.2) Compute the join of the relations stored at the leaves of the subtree $\mathcal{T}_v$. Let r be the resulting relation.
>
> (2.3) Compute the marginal $m(Y)$ as follows. For every Y-tuple $\mathbf{y}$
>
> if $\mathbf{y} \notin \pi_Y(r)$, then set $m(\mathbf{y}) := 0$; otherwise, compute $m(\mathbf{y})$ by evaluating the algebraic expression $\mathtt{M}$ using the numeric values of the Σ-distributions in I.
>
> (2.4) Exit.

Example 6.1. Consider our compositional expression

$$\theta = (ABC \triangleright (BD \triangleright BE)) \triangleright (ADF \triangleright FG)$$

and let $I = \langle f(ABC), g(BD), h(BE), k(ADF), l(FG)\rangle$ be a valid Σ-interpretation of θ, where Σ is the sum-product semifield. Suppose we want to compute the marginal $m(ACD)$ of the value of θ under I. We now apply the marginalization procedure above. We shall see that the node covering ACD is node 2 of $\mathcal{T}$ (see Figure 5); therefore, during the postorder traversal of $\mathcal{T}$, we shall visit (in order) the nodes 4, 8, 9, 5 and 2 only. Thus, we obtain:

$$L_4 = ABC \qquad \mathtt{E}_4 = f(ABC) \qquad lb(4) = \mathtt{true}$$

$$L_8 = BD \qquad \mathtt{E}_8 = g(BD) \qquad lb(8) = \mathtt{false}$$

$$L_9 = BE \qquad \mathtt{E}_9 = h(BE) \qquad lb(9) = \mathtt{false}$$

$$L_5 = BDE \qquad \mathtt{E}_5 = \frac{g(BD) \times h(BE)}{h^{\downarrow B}} \qquad lb(5) = \mathtt{false}$$

$$L_2 = ABCDE \qquad \mathtt{E}_2 = \frac{f(ABC) \times g(BD) \times h(BE)}{g^{\downarrow B} \times h^{\downarrow B}} \qquad lb(2) = \mathtt{true}$$

Since $lb(2) = \mathtt{true}$ and $ACD \subseteq L_2 = ABCDE$ we stop the traversal of $\mathcal{T}$ (see Figure 5).

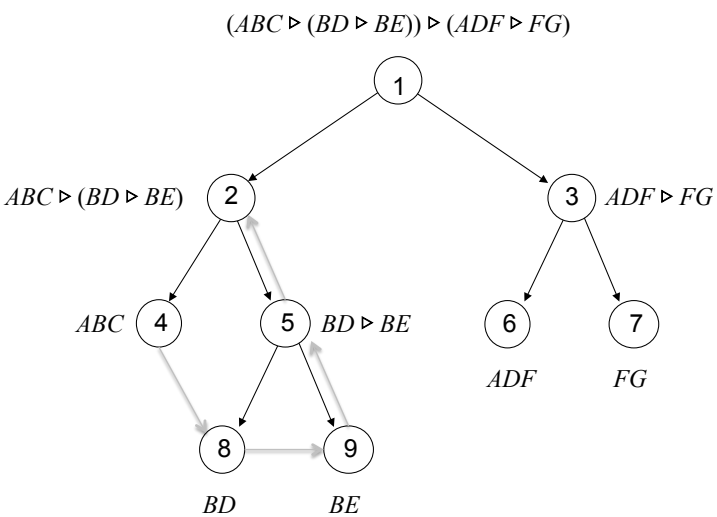

Fig. 5. The (partial) postorder traversal of the syntax tree of Fig. 3.

Note that, when we computed $\mathtt{E}_2$, we reduced the sum $\sum_{DE} \mathtt{E}_5$ simply by reducing the sum $\sum_D \mathtt{E}_8$ $(= g^{\downarrow B})$ since $L_4 \cap L_5$ $(= B) \subseteq L_8$ $(= BD)$.

Given $\mathtt{E}_2$, we reduce the sum

$$\sum_{BE} \mathtt{E}_2 = \sum_{BE} \frac{f(ABC) \times g(BD) \times h(BE)}{g^{\downarrow B} \times h^{\downarrow B}}$$

and set $\mathtt{M}$ to the result of the reduction; thus, we obtain

$$\mathtt{M} = \sum_{B} \frac{f(ABC) \times g(BD)}{g^{\downarrow B}} \,.$$

Next, we compute the join $\|f\| \bowtie \|g\| \bowtie \|h\|$ of the supports of the distributions associated with the leaves 4, 8 and 9 of the subtree $\mathcal{T}_2$.
Finally, for every ACD-tuple $(\mathbf{a}, \mathbf{c}, \mathbf{d})$, we compute $m(\mathbf{a}, \mathbf{c}, \mathbf{d})$ as follows:

$$m(\mathbf{a}, \mathbf{c}, \mathbf{d}) := \begin{cases} 0 & \text{if } (\mathbf{a}, \mathbf{c}, \mathbf{d}) \notin \pi_{ACD}(\|f\| \bowtie \|g\| \bowtie \|h\|) \\ \displaystyle\sum_{\mathbf{b}} \frac{f(\mathbf{a}, \mathbf{b}, \mathbf{c}) \times g(\mathbf{b}, \mathbf{d})}{g^{\downarrow B}(\mathbf{b})} & \text{otherwise.} \end{cases}$$

Since computing $g^{\downarrow B}$ requires a number of additions equal to the size of the relation $\|g\|$, computing $m(ACD)$ requires a number of algebraic operations equal to the sum of the sizes of the relations $\|g\|$ and $\|f\| \bowtie \|g\| \bowtie \|h\|$, which was the number of additions needed to compute $m(ACD)$ in Example 5.1.

7. THE MARGINAL-REPRESENTATION PROBLEM FROM SCRATCH

Consider again the marginal-representation problem stated in Section 5:

Given a valid Σ-interpretation I of compositional expressions θ, we want to compute the marginals of the value of θ for all sets featured in θ.

As in Section 6, we want to solve it "from scratch", that is, by taking as input the syntax tree $\mathcal{T}$ for θ where the three attributes (label, distribution, relation) are given only for the leaves of $\mathcal{T}$; moreover, we assume that the algebraic operations are the ordinary addition ($+$) and multiplication ($\times$).

Of course, the marginal-representation problem can be solved by repeating the marginalization procedure of Section 6 for each set featured in θ, but we can do better with only one traversal of $\mathcal{T}$ using the graphical implementation of the α-β procedure developed in Subsection 5.2. Explicitly, we perform the postorder traversal of $\mathcal{T}$ and stop after visiting the node covering the frame of θ. When we visit a node v, we construct $\mathtt{E}_v$ as we did in Subsection 6.1.1; moreover, if v belongs the leftmost branch of $\mathcal{T}$ (that is, if $lb(v) = \mathtt{true}$), then

— We set $\mathtt{M}_v := \mathtt{E}_v$.

— If v is a leaf, then we set r_v to the relation stored at v.

— If v is an interior node with left child u and right child w, we set r_w to the join of the relations stored at the leaves of the subtree $\mathcal{T}_w$ and set $r_v := r_u \bowtie r_w$. Moreover, if $\mathcal{T}_w$ contains marked leaves (that is, if $\mathcal{T}_w$ contains at least one leaf

that corresponds to the first occurence of some set in θ), then we repeatedly delete unmarked leaves of $\mathcal{T}_w$. Let $\mathcal{T}'_w$ be the residual of $\mathcal{T}_w$. At this point, in order to compute the algebraic expression $\mathtt{M}_u$ of the marginal $m_u(L_u)$ for each marked leaf u of $\mathcal{T}'_w$, we *backtrack* (as in Figure 3) by performing the following two steps:

Step 1. Reduce the sum $\sum_{A\in L_v\setminus L_w}\mathtt{M}_v$ and let $\mathtt{M}_w$ be the result of the reduction.

Step 2. For each arc $x\to y$ of $\mathcal{T}'_w$, reduce the sum $\sum_{A\in L_x\setminus L_y}\mathtt{M}_x$ and let $\mathtt{M}_y$ be the result of the reduction.

When a (marked) leaf u of $\mathcal{T}'_w$ is reached, we compute the relation $\pi_{L_u}(r_v)$ and, finally, the wanted marginal $m_u(L_u)$ as follows: for every L_u-tuple $\mathbf{t}$, if $\mathbf{t}\notin\pi_{L_u}(r_v)$, then we set $m_u(\mathbf{t})=0$; otherwise, we compute $m_u(\mathbf{t})$ by evaluating $\mathtt{M}_u$.

We now give an illustrative example.

Example 7.1. Consider our compositional expression

$$\theta=(ABC\rhd(BD\rhd BE))\rhd(ADF\rhd FG)$$

and let $I=\langle f(ABC),g(BD),h(BE),k(ADF),l(FG)\rangle$ be a valid Σ-interpretation of θ, where Σ is the sum-product semifield. We now apply the marginal-representation procedure above to compute the marginals of the value of θ under I on ABC, BD, BE, ADF and FG.

Since θ contains no repetitions, all the five leaves (nodes 4, 6, 7, 8, 9) of the syntax tree $\mathcal{T}$ are marked (see Figure 6) and, accordingly, the wanted marginals will be denoted by $m_4(ABC)$, $m_6(ADF)$, $m_7(FG)$, $m_8(BD)$ and $m_9(BE)$.

- When we visit node 4, we have $\mathtt{E}_4=f(ABC)$. Since $lb(4)=\mathtt{true}$, we set

$$\mathtt{M}_4:=\mathtt{E}_4=f(ABC)\qquad r_4=\|f\|\,.$$

  Since node 4 is a marked leaf, we also evaluate $\mathtt{M}_4$ and the result of the evaluation will be the marginal $m_4(ABC)$. Explicitly, $m_4(\mathbf{a},\mathbf{b},\mathbf{c})=0$ if $(\mathbf{a},\mathbf{b},\mathbf{c})\notin r_4\,(=\|f\|)$, and $m_4(\mathbf{a},\mathbf{b},\mathbf{c})=f(\mathbf{a},\mathbf{b},\mathbf{c})$ otherwise.

- When we visit node 8, we have $\mathtt{E}_8=g(BD)$ and $lb(8)=\mathtt{false}$.

- When we visit node 9, we have $\mathtt{E}_9=h(BE)$ and $lb(9)=\mathtt{false}$.

- When we visit node 5, we have $\mathtt{E}_5=\dfrac{g(BD)\times h(BE)}{h^{\downarrow B}}$ and $lb(5)=\mathtt{false}$.

- When we visit node 2, we have $\mathtt{E}_2=\dfrac{f(ABC)\times g(BD)\times h(BE)}{g^{\downarrow B}\times h^{\downarrow B}}$ and $lb(2)=\mathtt{true}$ (since $lb(4)=\mathtt{true}$). Since $lb(2)=\mathtt{true}$ and the subtree $\mathcal{T}_5$ has two marked leaves, we first compute the join r_5 of the relations stored at the leaves 8 and 9 of $\mathcal{T}_5$

$$r_5:=\|g\|\bowtie\|h\|$$

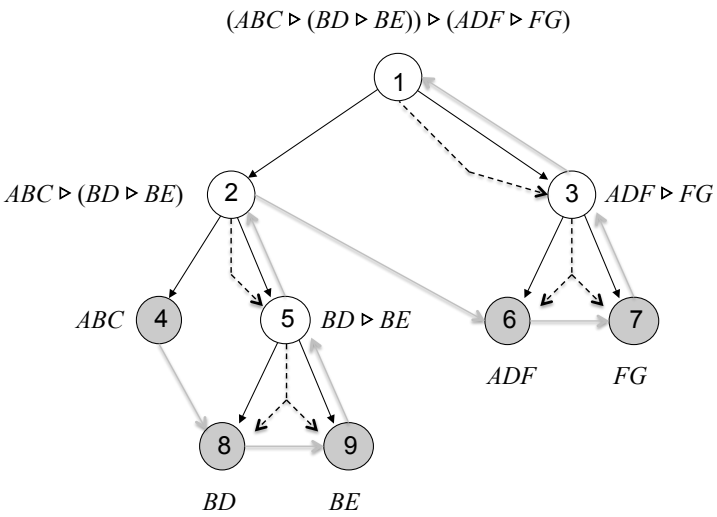

Fig. 6. The postorder traversal of a syntax tree with two backtrackings (dashed lines).

and then

$$r_2 := r_4 \bowtie r_5 \,.$$

At this point, we backtrack. The subtree $\mathcal{T}_5$ cannot be reduced ($\mathcal{T}_5' = \mathcal{T}_5$). Then, we reduce the sum

$$\sum_{AC} \mathtt{E}_2 = \sum_{AC} \frac{f(ABC) \times g(BD) \times h(BE)}{g^{\downarrow B} \times h^{\downarrow B}}$$

and set $\mathtt{M}_5$ to the result of the reduction; thus, we obtain

$$\mathtt{M}_5 = \frac{f^{\downarrow B} \times g(BD) \times h(BE)}{g^{\downarrow B} \times h^{\downarrow B}} \,.$$

At this point, we start a top-down traversal of the subtree $\mathcal{T}_5$. When the arcs $5 \to 8$ and $5 \to 9$ are traversed, we reach the marked leaves 8 and 9. Then, we set $\mathtt{M}_8$ and $\mathtt{M}_9$ to the reductions of the two sums $\sum_E \mathtt{M}_5$ and $\sum_D \mathtt{M}_5$ respectively; thus, we obtain

$$\mathtt{M}_8 = \frac{f^{\downarrow B} \times g(BD)}{g^{\downarrow B}} \qquad \mathtt{M}_9 = \frac{f^{\downarrow B} \times h(BE)}{h^{\downarrow B}} \,.$$

Next, we compute the two relations $\pi_{BD}(r_2)$ and $\pi_{BE}(r_2)$ and, then $m_8(BD)$ and $m_9(BE)$ by evaluating $\mathtt{M}_8$ and $\mathtt{M}_9$.

- When we visit nodes 6, 7 and 3, we obtain

$$\begin{aligned} \mathtt{E}_6 &= k(ADF) & lb(6) &= \mathtt{false} \\ \mathtt{E}_7 &= l(FG) & lb(7) &= \mathtt{false} \\ \mathtt{E}_3 &= \frac{k(ADF)\times l(FG)}{l^{\downarrow F}} & lb(3) &= \mathtt{false}\,. \end{aligned}$$

- When we visit node 1, we obtain

$$\mathtt{E}_1 = \frac{f(ABC)\times g(BD)\times h(BE)\times k(ADF)\times l(FG)}{g^{\downarrow B}\times h^{\downarrow B}\times k^{\downarrow AD}\times l^{\downarrow F}}$$

and $lb(1) = \mathtt{true}$ (since $lb(2) = \mathtt{true}$). Since $lb(1) = \mathtt{true}$ and the subtree $\mathcal{T}_3$ has two marked leaves, we first compute the join of the relations stored at the leaves 6 and 7 of $\mathcal{T}_3$

$$r_3 := \|k\| \bowtie \|l\|$$

and then

$$r_1 := r_2 \bowtie r_3\,.$$

At this point, we backtrack. The subtree $\mathcal{T}_3$ has two marked leaves and cannot be reduced. Then, we reduce the sum

$$\sum_{BCE} \mathtt{E}_1 = \sum_{BCE} \frac{f(ABC)\times g(BD)\times h(BE)\times k(ADF)\times l(FG)}{g^{\downarrow B}\times h^{\downarrow B}\times k^{\downarrow AD}\times l^{\downarrow F}}$$

and set $\mathtt{M}_3$ to the result of the reduction; thus, we obtain

$$\mathtt{M}_3 = \frac{k(ADF)\times l(FG)}{k^{\downarrow AD}\times l^{\downarrow F}} \times \sum_{B} \frac{f^{\downarrow AB}\times g(BD)}{g^{\downarrow B}}\,.$$

At this point, we start a top-down traversal of the subtree $\mathcal{T}_3$. When the arcs $3 \to 6$ and $3 \to 7$ are traversed, we reach the marked leaves 6 and 7. Then, we set $\mathtt{M}_6$ and $\mathtt{M}_7$ to the reductions of the two sums $\sum_G \mathtt{M}_3$ and $\sum_{AD} \mathtt{M}_3$ respectively; thus, we obtain

$$\mathtt{M}_6 = \frac{k(ADF)}{k^{\downarrow AD}} \times \sum_{B} \frac{f^{\downarrow AB}\times g(BD)}{g^{\downarrow B}}$$

$$\mathtt{M}_7 = \frac{l(FG)}{l^{\downarrow F}} \times \sum_{AD} \left(\frac{k(ADF)}{k^{\downarrow AD}} \times \sum_{B} \frac{f^{\downarrow AB}\times g(BD)}{g^{\downarrow B}} \right).$$

Next, we compute the relations $\pi_{ADF}(r_1)$ and $\pi_{FG}(r_1)$ and, finally, compute the wanted marginals $m_6(ADF)$ and $m_7(FG)$.

After a pedantic analysis of the number of additions, multiplications and divisions executed, we find that the computational complexity (measured in terms of algebraic operations) is of the same order as in Example 5.2.

8. MARGINALIZATION WITH BOOLEAN DATA

For a Boolean distribution $f(X)$ one has that $f(\mathbf{x}) = \texttt{true}$ if and only if $\mathbf{x} \in \|f\|$ so that $f(X)$ is uniquely determined by its support $\|f\|$. Therefore, we can solve the two marginalization problems as follows:

Boolean Marginalization Procedure

(*Step* 1) We perform a postorder traversal of $\mathcal{T}$ and stop after examining the node v covering Y.

(*Step* 2) Compute the join r_v of the relations stored at the leaves of the subtree $\mathcal{T}_v$.

(*Step* 3) For every Y-tuple $\mathbf{y}$, set $m(\mathbf{y})$ to `true` if and only if $\mathbf{y} \in \pi_Y(r_v)$.

Boolean Marginal-Representation Procedure

We perform a postorder traversal of $\mathcal{T}$ and stop after examining the node covering the frame of θ.

When a node v is visited, if $lb(v) = \texttt{true}$ then do:

Case 1: v is a (marked) leaf. Set r_v to the relation stored at v. For every L_v-tuple $\mathbf{t}$, set $m_v(\mathbf{t})$ to `true` if and only if $\mathbf{t} \in r_v$.

Case 2: v is an interior node with left child u and right child w, we set r_w to the join of the relations stored at the leaves of the subtree $\mathcal{T}_w$ and set $r_v := r_u \bowtie r_w$. If $\mathcal{T}_w$ contains marked leaves, then repeatedly delete unmarked leaves of $\mathcal{T}_w$. Let $\mathcal{T}'_w$ be the residual of $\mathcal{T}_w$. For each leaf u of $\mathcal{T}'_w$ compute the relation $\pi_{L_u}(r_v)$ and, for every L_u-tuple $\mathbf{t}$, set $m_u(\mathbf{t})$ to `true` if and only if $\mathbf{t} \in \pi_{L_u}(r_v)$.

9. A CLOSING NOTE

Given a compositional expression θ, let v be the root of syntax tree for θ. The algebraic expression $\mathtt{E}_v$ over a Σ-interpretation I of θ can be viewed as an algebraic expression of the evaluation operator associated with θ over Σ (see Subsection 4.1). Then, it is natural to introduce the following notion of equivalence between compositional expressions: two compositional expressions θ and η are *algebraically equivalent over* Σ if the evaluation operators associated with θ and η over Σ have the same algebraic expressions.

Example 9.1. Consider the following three compositional expressions:

$$\theta = AB \rhd AC \qquad \eta = (AB \rhd AC) \rhd BC \qquad \zeta = (AB \rhd AC) \rhd (AB \rhd BC)\,.$$

Let Σ be the real field or the sum-product semifield. The evaluation operators associated with θ, η and ζ have the same algebraic expression

$$\frac{f(AB) \times g(AC)}{g^{\downarrow A}}$$

and, hence, θ, η and ζ are pairwise algebraically equivalent over Σ.

We leave to future research the comparison of algebraic equivalence with the notion of equivalence introduced in [27] and with "Markov equivalence" [13, 22].

10. APPENDIX

A *semifield* [27] is a tuple $\Sigma = \langle \mathtt{S}, (\oplus, \mathtt{0}), (\otimes, \mathtt{1}) \rangle$, where $\mathtt{S}$ is a set and

(P1) $(\mathtt{S}, \oplus, \mathtt{0})$ is a *commutative monoid*:

- the operation $\oplus$ is associative and commutative,
- $\mathtt{0}$ is the additive identity (that, is $\mathtt{a} \oplus \mathtt{0} = \mathtt{a}$ for all $\mathtt{a} \in \mathtt{S}$);

(P2) $(\mathtt{S}, \otimes, \mathtt{1})$ is a *zero-divisor free commutative group*:

- the operation $\otimes$ is associative and commutative,
- $\mathtt{1}$ is the multiplicative identity (that is, $\mathtt{a} \otimes \mathtt{1} = \mathtt{a}$ for all $\mathtt{a} \in \mathtt{S}$),
- for all $\mathtt{a} \in \mathtt{S} \setminus \{\mathtt{0}\}$ there is an element of $\mathtt{S}$, denoted by $\overline{\mathtt{a}}$, such that $\mathtt{a} \otimes \overline{\mathtt{a}} = \mathtt{1}$,
- $\mathtt{a} \otimes \mathtt{b} = \mathtt{0}$ if and only if $\mathtt{a} = \mathtt{0}$ or $\mathtt{b} = \mathtt{0}$;

(P3) the distributive law holds, that is, $\mathtt{a} \otimes (\mathtt{b} \oplus \mathtt{c}) = (\mathtt{a} \otimes \mathtt{b}) \oplus (\mathtt{a} \otimes \mathtt{c})$ for all $(\mathtt{a}, \mathtt{b}, \mathtt{c}) \in \mathtt{S}^3$.

$\mathtt{S}$	$(\oplus, \mathtt{0})$	$(\otimes, \mathtt{1})$	short name
$(-\infty, +\infty)$	$(+, 0)$	$(\times, 1)$	real field
$[0, \infty)$	$(+, 0)$	$(\times, 1)$	sum-product semifield
$(0, \infty]$	$(\min, \infty)$	$(\times, 1)$	min-product semifield
$(-\infty, +\infty]$	$(\min, +\infty)$	$(+, 0)$	min-sum semifield
$[0, \infty)$	$(\max, 0)$	$(\times, 1)$	max-product semifield
$[-\infty, +\infty)$	$(\max, -\infty)$	$(+, 0)$	max-sum semifield
$\{\mathtt{false}, \mathtt{true}\}$	$(\vee, \mathtt{false})$	$(\wedge, \mathtt{true})$	Boolean algebra
$\{0, 1\}$	$(+ \ mod \ 2, 0)$	$(\times, 1)$	Galois field $GF(2)$

Tab. 3. A short list of semifields.

Table 3 contains a short list of semifields that have found applications in information theory [2, 23] as well as in probability theory, statistical physics, language theory (see [5, 29]), in information systems [17, 18, 19, 20, 21] in relational databases [3] and in multidimensional databases [26].

The min-product, min-sum, max-product and max-sum semifields are called *tropical algebras* [5, 25, 29]. A semifield is *metric* [27] if it is zero-sum free:

$$\text{if } \mathtt{a} \oplus \mathtt{b} = \mathtt{0} \text{ then } \mathtt{a} = \mathtt{b} = \mathtt{0}.$$

Examples of metric semifields are the sum-product and the tropical semifields. In these semifields, the multiplicative inverse $\overline{\mathtt{a}}$ of $\mathtt{a} \in \mathtt{S} \setminus \{\mathtt{0}\}$ will be written as follows:

— if Σ is the sum-product, min-product or max-product semifield, $\overline{\mathtt{a}}$ is written as $\frac{1}{\mathtt{a}}$;

— if Σ is the min-sum or max-sum semifield, $\overline{\mathtt{a}}$ is written as $-\mathtt{a}$.

The Boolean algebra provides another example of a metric semifield. In this case, the multiplicative inverse of `true` is `true`.

(Received February 6, 2015)

REFERENCES

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman: Data Structures and Algorithms. Addison-Wesley Pub. Co, Reading 1987.

[2] S. M. Aji and R.-J. McEliece: The generalized distributive law. IEEE Trans. Inform. Theory *46* (2000), 325–343. DOI:10.1109/18.825794

[3] C. Beeri, R. Fagin, D. Maier, and M. Yannakakis: On the desirability of acyclic database schemes. J. ACM *30* (1983), 479–513. DOI:10.1145/2402.322389

[4] V. Bína and R. Jiroušek: Marginalization in multidimensional compositional models. Kybernetika *42* (2006), 405–422.

[5] S. Gaubert and Max Plus: Methods and applications of (max, +) linear algebra. In: Proc. XIV Symp. on Theoretical Aspects of Computer Science Hansestatdt Luebeck 1997. DOI:10.1007/bfb0023465

[6] R. Jiroušek: Composition of probability measures on finite spaces. In: Proc. XIII International Conf. on Uncertainty in Artificial Intelligence (D. Geiger and P. P. Shenoy, eds.), Morgan Kaufmann, San Francisco 1997, pp. 274–281.

[7] R. Jiroušek: Marginalization in composed probabilistic models. In: Proc. XVI International Conf. on Uncertainty in Artificial Intelligence, (C. Boutilier and M. Goldszmidt, eds.), Morgan-Kauffmann Pub., San Francisco 2000, vol. C, pp. 301-308. DOI:10.1016/b978-1-4832-1451-1.50041-x

[8] R. Jiroušek: Decomposition of multidimensional distributions represented by perfect sequences. Ann. Math. Artif. Intelligence *5* (2002), 215–226. DOI:10.1023/a:1014591402750

[9] R. Jiroušek: Foundations of compositional model theory. Int. J. General Systems *40* (2011), 623–678. DOI:10.1080/03081079.2011.562627

[10] R. Jiroušek: Local computations in Dempster-Shafer theory of evidence. Int. J. Approx. Reasoning *53* (2012), 1155–1167. DOI:10.1016/j.ijar.2012.06.012

[11] R. Jiroušek: On causal compositional models: simple examples. In: Proc. XIV International Conference on Information Processing and Management of Uncertainty in Knowledge-Bases Systems (IPMU 2014) (A. Laurent et al., eds.), Part I, CCIS 442, pp. 517–526. DOI:10.1007/978-3-319-08795-5_53

[12] R. Jiroušek and V. Kratochvíl: Marginalization algorithm for compositional models. In: Proc. XI International Conference on Information Processing and Management of Uncertainty in Knowledge-Bases Systems (IPMU 2006) (B. Bouchon-Meunier and R.R. Yager, eds.), pp. 2300–2307.

[13] R. Jiroušek and V. Kratochvíl: Foundations of compositional models: structural properties. Int. J. General Systems *44* (2015), 2–25. DOI:10.1080/03081079.2014.934370

[14] R. Jiroušek and P. P. Shenoy: Compositional models in valuation-based systems. Int. J. Approx. Reasoning *55* (2014), 277–293. DOI:10.1016/j.ijar.2013.02.002

[15] R. Jiroušek and J. Vejnarová: General framework for multidimensional models. Int. J. General Systems *18* (2003), 107–127. DOI:10.1002/int.10077

[16] R. Jiroušek, J. Vejnarová, and M. Daniels: Composition models of belief functions. In: Proc. V Symp. on Imprecise Probabilities and Their Applications (G. De Cooman, J. Vejnarová and M. Zaffalon, eds.), Action M Agency, Prague 2007, pp. 243–252.

[17] J. Kohlas: Information algebras: generic structures for inference. Springer-Verlag, 2003. DOI:10.1007/978-1-4471-0009-6

[18] J. Kohlas, M. Pouly, and C. Schneuwly: Generic local computation. J. Comput. System Sciences *78* (2012), 348–369. DOI:10.1016/j.jcss.2011.05.012

[19] J. Kohlas and J. Schmid: An algebraic theory of information: an introduction and survey. Information *5* (2014), 219–254. DOI:10.3390/info5020219

[20] J. Kohlas and P. P. Shenoy: Computation in valuation algebras. In: Handbook of Defeasible Reasoning and Uncertainty Management Systems, Volume 5: Algorithms for Uncertainty and Defeasible Reasoning (J. Kohlas and S. Moral, eds.), Kluwer, Dordrecht 2000, pp. 5–39. DOI:10.1007/978-94-017-1737-3_2

[21] J. Kohlas and N. Wilson: Semiring induced valuation algebra: exact and approximate local computation algorithms. Artificial Intelligence *172* (2008), 1360–1399. DOI:10.1016/j.artint.2008.03.003

[22] V. Kratochvíl: Probabilistic compositional models: solution of an equivalence problem. Int. J. Approx. Reasoning *54* (2013), 590–601. DOI:10.1016/j.ijar.2013.01.002

[23] F. R. Kschinschang, B. J. Frey, and H.-A. Loeliger: Factor graphs and the sum-product algorithm. IEEE Trans. Inform. Theory *47* (2001), 498–519. DOI:10.1109/18.910572

[24] S. L. Lauritzen: Graphical Models. Oxford University Press, Oxford 1996. DOI:10.1002/(sici)1097-0258(19991115)18:21¡2983::aid-sim198¿3.0.co;2-a

[25] G. L. Litvinov and S. N. Sergeev (eds.): Proc. of the International Workshop TROPICAL-07 on Tropical and Idempotent Mathematics. Contemporary Mathematics *495* (2007), American Mathematical Society. DOI:10.1090/conm/616

[26] F. M. Malvestuto: A join-like operator to combine data cubes, and answer queries from multiple data cubes. ACM Trans. Database Syst. *39* (2014), 3, 1–31. DOI:10.1145/2638545

[27] F. M. Malvestuto: Equivalence of compositional expressions and independence relations in compositional models. Kybernetika *50* (2014), 322–362. DOI:10.14736/kyb-2014-3-0322

[28] F. M. Malvestuto: Erratum: Equivalence of compositional expressions and independence relations in compositional models. Kybernetika *51* (2015), 387-388. DOI:10.14736/kyb-2015-2-0387

[29] D. Speyer and B. Sturmfels: Tropical mathematics. Mathematics Magazine *82* (2009), 163–173. DOI:10.4169/193009809x468760

Francesco M. Malvestuto, Department of Informatics, Sapienza University of Rome, Via Salaria 113, 00198 Rome. Italy.

e-mail: malvestuto@di.uniroma1.it