Algebraic Model Counting

Angelika Kimmig Guy Van den Broeck
Luc De Raedt
Department of Computer Science, KU Leuven
Celestijnenlaan 200a – box 2402, 3001 Heverlee, Belgium
{firstname.lastname}@cs.kuleuven.be

Abstract

Weighted model counting (WMC) is a well-known inference task on knowledge bases, used for probabilistic inference in graphical models. We introduce algebraic model counting (AMC), a generalization of WMC to a semiring structure. We show that AMC generalizes many well-known tasks in a variety of domains such as probabilistic inference, soft constraints and network and database analysis. Furthermore, we investigate AMC from a knowledge compilation perspective and show that all AMC tasks can be evaluated using `sd-DNNF` circuits. We identify further characteristics of AMC instances that allow for the use of even more succinct circuits.

1 Introduction

Today, some of the most efficient techniques for probabilistic inference employ reductions to weighted model counting (WMC) both for propositional and for relational probabilistic models (Park, 2002; Sang et al., 2005; Darwiche, 2009; Fierens et al., 2011). On the other hand, it is well-known that probabilistic inference as well as many other tasks can be generalized to a sum of products computation over models with suitable operators from a semiring structure. This has led to common inference algorithms for a variety of different inference problems in a variety of different fields, see for instance Goodman (1999), Eisner et al. (2005), Meseguer et al. (2006), Green et al. (2007), Bacchus et al. (2009), Larrosa et al. (2010), Baras and Theodorakopoulos (2010) and Kimmig et al. (2011). The work presented here builds on these two lines of work.

As our first contribution, we introduce the task of *algebraic model counting (AMC)*. AMC generalizes weighted model counting to the semiring setting and supports various types of labels, including numerical ones as used in WMC, but also sets, polynomials, Boolean formulae, and many more. It thereby also generalizes many different tasks from a variety of different fields. As our second contribution, we investigate how to solve AMC problems using knowledge

compilation. As AMC is defined in terms of the set of models of a propositional logic theory, we can exploit the succinctness results of the knowledge compilation map of Darwiche and Marquis (2002). We show that AMC can in general be evaluated using `sd-DNNF` circuits, which are more succinct than a direct representation of the set of models. Furthermore, we identify a number of characteristics of AMC tasks that allow for using even more succinct types of circuits. Our results generalize well-known results for satisfiability and model counting in circuits to broad classes of AMC tasks and extend the task classification in algebraic Prolog (Kimmig et al., 2011) to more succinct types of circuits. As our third contribution, we provide conditions under which AMC generalizes semiring sums of products defined over derivations, that is, sequences of possibly repeated variables, instead of over models.

This paper is organized as follows. We introduce algebraic model counting in Section 2. Section 3 provides task characteristics that allow for sound evaluation on specific classes of circuits and shows how these generalize previous results. We discuss future work and conclude in Section 4.

2 Algebraic Model Counting

We first define the task of algebraic model counting, provide some example instances, and briefly discuss its relationships to existing frameworks. The underlying mathematical structure is that of a commutative semiring.

Definition 1 ((Commutative) Semiring)**.** A *semiring* is a structure $(\mathcal{A}, \oplus, \otimes, e^{\oplus}, e^{\otimes})$, where *addition* $\oplus$ and *multiplication* $\otimes$ are associative binary operations over the set $\mathcal{A}$, $\oplus$ is commutative, $\otimes$ distributes over $\oplus$, $e^{\oplus} \in \mathcal{A}$ is the neutral element of $\oplus$, $e^{\otimes} \in \mathcal{A}$ that of $\otimes$, and for all $a \in \mathcal{A}$, $e^{\oplus} \otimes a = a \otimes e^{\oplus} = e^{\oplus}$. In a *commutative semiring*, $\otimes$ is commutative as well.

Algebraic model counting is now defined as follows:

Definition 2 (AMC Problem)**.** Given

- a *propositional logic theory* T over a set of variables $\mathcal{V}$,
- a *commutative semiring* $(\mathcal{A}, \oplus, \otimes, e^{\oplus}, e^{\otimes})$, and
- a *labeling function* $\alpha : \mathcal{L} \to \mathcal{A}$, mapping literals $\mathcal{L}$ of the variables in $\mathcal{V}$ to elements of the semiring set $\mathcal{A}$,

compute

$$\mathbf{A}(T) = \bigoplus_{I \in \mathcal{M}(T)} \bigotimes_{l \in I} \alpha(l), \tag{1}$$

where $\mathcal{M}(T)$ denotes the set of models of T.

task	$\mathcal{A}$	$e^{\oplus}$	$e^{\otimes}$	$\oplus$	$\otimes$	$\alpha(v)$	$\alpha(\neg v)$	ref
SAT	$\{true, false\}$	$false$	$true$	$\vee$	$\wedge$	$true$	$true$	B, BT, G, GK, K, L, M
#SAT	$\mathbb{N}$	0	1	$+$	$\cdot$	1	1	B, G, GK, K, L
WMC	$\mathbb{R}_{\geq 0}$	0	1	$+$	$\cdot$	$\in \mathbb{R}_{\geq 0}$	$\in \mathbb{R}_{\geq 0}$	
PROB	$\mathbb{R}_{\geq 0}$	0	1	$+$	$\cdot$	$\in [0, 1]$	$1 - \alpha(v)$	B, BT, E, G, K
SENS	$\mathbb{R}[\mathcal{V}]$	0	1	$+$	$\cdot$	v or $\in [0, 1]$	$1 - \alpha(v)$	K
GRAD	$\mathbb{R}_{\geq 0} \times \mathbb{R}$	$(0, 0)$	$(1, 0)$	Eq. (4)	Eq. (5)	Eq. (2)	Eq. (3)	E, K
MPE	$\mathbb{R}_{\geq 0}$	0	1	max	$\cdot$	$\in [0, 1]$	$1 - \alpha(v)$	B, BT, G, K, L, M
S-PATH	$\mathbb{N}^{\infty}$	∞	0	min	$+$	$\in \mathbb{N}$	0	BT, GK, K
W-PATH	$\mathbb{N}^{\infty}$	0	∞	max	min	$\in \mathbb{N}$	∞	BT
FUZZY	$[0, 1]$	0	1	max	min	$\in [0, 1]$	1	GK, M
kWEIGHT	$\{0, \ldots, k\}$	k	0	min	$+^k$	$\in \{0, \ldots, k\}$	$\in \{0, \ldots, k\}$	M
$\mathtt{OBDD}_<$	$\mathtt{OBDD}_<(\mathcal{V})$	$\mathtt{OBDD}_<(0)$	$\mathtt{OBDD}_<(1)$	$\vee$	$\wedge$	$\mathtt{OBDD}_<(v)$	$\neg\mathtt{OBDD}_<(v)$	K
WHY	$\mathcal{P}(\mathcal{V})$	$\emptyset$	$\emptyset$	$\cup$	$\cup$	$\{v\}$	n/a	GK
$\mathcal{RA}^+$	$\mathbb{N}[\mathcal{V}]$	0	1	$+$	$\cdot$	v	n/a	GK

Table 1: Examples of commutative semirings and labeling functions. The **WHY** and $\mathcal{RA}^+$ provenance semirings apply to positive literals only. Reference key: B (Bacchus et al., 2009), BT (Baras and Theodorakopoulos, 2010), E (Eisner, 2002), G (Goodman, 1999), GK (Green et al., 2007), K (Kimmig et al., 2011), L (Larrosa et al., 2010), M (Meseguer et al., 2006); more examples can be found in these references.

Theorem 1. *AMC generalizes satisfiability (**SAT**), model counting (**#SAT**), weighted model counting (**WMC**), probabilistic inference (**PROB**), sensitivity analysis (**SENS**), gradient (**GRAD**), probability of most likely states (**MPE**), shortest (**S-PATH**) and widest (**W-PATH**) paths, fuzzy (**FUZZY**) and k-weighted (k**WEIGHT**) constraints, and* $\texttt{OBDD}_<$ *construction.*

Proof. By verification of the definitions in Table 1. □

SAT, **#SAT**, **WMC** and **PROB** are well-known tasks that appear in many fields. **SENS** and **GRAD** allow one to investigate the effect of parameter changes and to learn parameters in a probabilistic setting, respectively. In **GRAD**, labels are tuples (p_i, g_i) with $p_i \in [0, 1]$ the probability of v_i and g_i the gradient with respect to the kth variable:

$$\alpha(v_i) = \begin{cases} (p_i, 1) & \text{if } i = k \\ (p_i, 0) & \text{if } i \neq k \end{cases} \tag{2}$$

$$\alpha(\neg v_i) = \begin{cases} (1 - p_i, -1) & \text{if } i = k \\ (1 - p_i, 0) & \text{if } i \neq k \end{cases} \tag{3}$$

$$(a_1, a_2) \oplus (b_1, b_2) = (a_1 + b_1, a_2 + b_2) \tag{4}$$

$$(a_1, a_2) \otimes (b_1, b_2) = (a_1 \cdot b_1, a_1 \cdot b_2 + a_2 \cdot b_1) \tag{5}$$

If the second element of the label denotes a cost, the **GRAD** semiring calculates expected costs. **MPE**, **S-PATH**, **W-PATH**, **FUZZY** and k**WEIGHT** (with bounded addition $+^k$) are examples of optimization tasks. Finally, AMC can also be used to construct a canonical representation of the set of models in the form of an $\texttt{OBDD}_<$ circuit, a popular data structure in many fields of computer science. The last two tasks in the table originate from probabilistic databases under the positive relational algebra $\mathcal{RA}^+$ and are not easily extended to negative literals. Why-provenance (**WHY**) collects the set of identifiers of all tuples an answer depends on, whereas $\mathcal{RA}^+$-provenance constructs polynomials that also take into account the number of times the tuples are used. While all tasks listed in Table 1 are representative examples from the literature, cf. the references given in the table, this is by no means an exhaustive list of semirings and labeling functions that can be used for AMC.

As these examples illustrate, the AMC task shares its basic structure with a number of other tasks. The class of sum-of-products problems generalizes factor graphs to the algebraic setting, but uses discrete valued variables or factors as basic building blocks (Bacchus et al., 2009). In this context, affine algebraic decision diagrams (Sanner and McAllester, 2005) and AND/OR multi-valued decision diagrams (Mateescu et al., 2008) have been used for inference with real-valued semirings. The restriction to two-valued variables allows us to directly compile AMC tasks to propositional circuits without adding constraints on legal variable assignments to the theory. In soft constraint programming, additional constraints are imposed on the semiring, which ensure that addition optimizes the degree of constraint satisfaction (Meseguer et al., 2006). Wilson (2005) provides an algorithm that compiles semiring-based systems into

semiring-labelled decision diagrams, which are closely related to unordered binary decision diagrams, to compute valuations. Semiring-induced propositional logic labels clauses with semiring elements with a weight associated to their falsification and is restricted to semirings whose induced pre-order is partial (Larrosa et al., 2010). In algebraic Prolog (aProbLog), a semiring-labeled logic program is reduced to AMC for inference (Kimmig et al., 2011).

While AMC sums over models, other tasks sum over sequences of possibly repeated variables. Examples include algebraic path problems (Baras and Theodorakopoulos, 2010), semiring parsing (Goodman, 1999), provenance semirings for positive relational algebra queries in databases (Green et al., 2007), and semiring-weighted dynamic programs (Eisner et al., 2005). We will discuss the difference between such derivation-based settings and AMC in more detail in Section 3.5.

3 AMC using Knowledge Compilation

In their knowledge compilation map, Darwiche and Marquis (2002) provide an overview of succinctness relationships between various types of propositional circuits. Furthermore, they show which reasoning tasks in propositional logic, such as (weighted) model counting (**#SAT/WMC**) or satisfiability checking (**SAT**), are evaluated on which circuits in time polynomial in the size of the circuit. Propositional circuits are often used as a representation language in weighted model counting and similar tasks, including for instance probability calculation and sensitivity analysis in probabilistic databases (Jha and Suciu, 2011; Kanagal et al., 2011) and inference in algebraic Prolog (Kimmig et al., 2011). In the following, we extend this approach to AMC. We first repeat the relevant knowledge compilation concepts, closely following Darwiche and Marquis (2002).

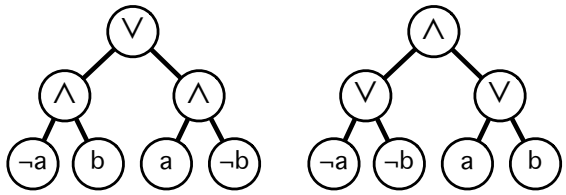
Definition 3 (`NNF`)**.** A sentence in *negation normal form* (`NNF`) over a set of propositional variables $\mathcal{V}$ is a rooted, directed acyclic graph where each leaf node is labeled with true ($\top$), false ($\bot$), or a literal of a variable in $\mathcal{V}$, and each internal node with disjunction ($\vee$) or conjunction ($\wedge$).

Definition 4 (Decomposability)**.** An `NNF` is *decomposable* if for each conjunction node $\bigwedge_{i=1}^{n} \phi_i$, no two children ϕ_i and ϕ_j share any variable.

Definition 5 (Determinism)**.** An `NNF` is *deterministic* if for each disjunction node $\bigvee_{i=1}^{n} \phi_i$, each pair of different children ϕ_i and ϕ_j is logically inconsistent.

Definition 6 (Smoothness)**.** An `NNF` is *smooth* if for each disjunction node $\bigvee_{i=1}^{n} \phi_i$, each child ϕ_i mentions the same set of variables.

`DNNF`, `d-NNF`, `s-NNF`, `sd-NNF`, `d-DNNF`, `s-DNNF`, and `sd-DNNF` are the subsets of `NNF` satisfying (combinations of) these properties, where `D` stands for decomposable, `d` for deterministic, and `s` for smooth. For instance, the circuit in Figure 1a is in `sd-DNNF`, while the one in Figure 1b has none of the three properties.



(a) `sd-DNNF` (b) `NNF`

Figure 1: Example of an `sd-DNNF` and `NNF` circuit.

	general $\otimes$		idempotent and consistency-pres. $(\otimes, \alpha)$	
	neutral $(\oplus, \alpha)$	non-neutral $(\oplus, \alpha)$	neutral $(\oplus, \alpha)$	non-neutral $(\oplus, \alpha)$
idempotent $\oplus$	`DNNF` (Th. 5)	`s-DNNF` (Th. 3)	`NNF` (Th. 7)	`s-NNF` (Th. 7)
non-idempotent $\oplus$	`d-DNNF` (Th. 4)	`sd-DNNF` (Th. 2)	`d-NNF` (Th. 7)	`sd-NNF` (Th. 6)

Table 2: Semiring characteristics and corresponding circuits that allow for sound AMC evaluation.

The algebraic model count $\mathbf{A}(T)$ is defined as a summation over the set of models $\mathcal{M}(T)$ of a propositional theory T, which corresponds to the `MODS` language in the knowledge compilation map. However, as this `MODS` language is exponentially less succinct than any other representation of T included in the map, converting to `MODS` in order to evaluate Equation (1) directly is undesirable. In the following, we therefore establish a connection between characteristics of AMC tasks and properties of the `NNF` circuits they can be evaluated on, resulting in the classification scheme summarized in Table 2.

The key idea underlying `NNF` evaluation is to perform a bottom-up pass over the circuit, labeling each node with the value of the subcircuit rooted at that node. For disjunction nodes, the values of all their children are combined using $\oplus$, for conjunction nodes using $\otimes$.

Definition 7 (`NNF` Evaluation)**.** The function Eval specified in Algorithm 1 *evaluates* an `NNF` circuit for a commutative semiring $(\mathcal{A}, \oplus, \otimes, e^{\oplus}, e^{\otimes})$ and labeling function α.

Consider for example **#SAT** for the two circuits in Figure 1, which both represent an exclusive OR of two variables. Evaluation of the `sd-DNNF` in Figure 1a, which in fact is a `MODS` representation, assigns label 1 to each leaf, $1 \cdot 1 = 1$ to each conjunction node, and $1 + 1 = 2$ to the disjunction node at the root and thus the entire circuit, which is correct. On the other hand, evaluation on

Algorithm 1 Evaluating an NNF circuit N for a commutative semiring $(\mathcal{A}, \oplus, \otimes, e^{\oplus}, e^{\otimes})$ and labeling function α.

1: **function** EVAL($N, \oplus, \otimes, e^{\oplus}, e^{\otimes}, \alpha$)
2: **if** N is a true node $\top$ **then** return $e^{\otimes}$
3: **if** N is a false node $\bot$ **then** return $e^{\oplus}$
4: **if** N is a literal node l **then** return $\alpha(l)$
5: **if** N is a disjunction $\bigvee_{i=1}^{m} N_i$ **then**
6: return $\bigoplus_{i=1}^{m}$ EVAL($N_i, \oplus, \otimes, e^{\oplus}, e^{\otimes}, \alpha$)
7: **if** N is a conjunction $\bigwedge_{i=1}^{m} N_i$ **then**
8: return $\bigotimes_{i=1}^{m}$ EVAL($N_i, \oplus, \otimes, e^{\oplus}, e^{\otimes}, \alpha$)

the general NNF in Figure 1b assigns $1 + 1 = 2$ to each disjunction node and $2 \cdot 2 = 4$ to the conjunction node at the root. This overestimation is due to models shared by the children of the same disjunction node and variables shared by the children of the conjunction node, as we will see in more detail in Section 3.2 and 3.3.

Definition 8 (Soundness)**.** Evaluating an NNF representation N_T of a propositional theory T for a semiring $(\mathcal{A}, \oplus, \otimes, e^{\oplus}, e^{\otimes})$ and labeling function α is a *sound AMC computation* iff $\text{EVAL}(N_T, \oplus, \otimes, e^{\oplus}, e^{\otimes}, \alpha) = \mathbf{A}(T)$.

In the following, we establish a general soundness result for AMC evaluation on sd-DNNF circuits as well as properties of AMC tasks that guarantee soundness for various other subclasses of NNF. Given soundness, we inherit the polynomial complexity results of the knowledge compilation map (Darwiche and Marquis, 2002) for semiring operators with unit cost. Note however that there are semirings with more expensive operators. For instance, labels in $\text{OBDD}_{<}$ may grow exponentially in the circuit size.

3.1 sd-DNNF Evaluation

We show that AMC evaluation is sound on sd-DNNF circuits. As these are strictly more succinct than MODS representations, they allow for more efficient inference.

Theorem 2 (sd-DNNF Evaluation)**.** *Evaluating an* sd-DNNF *representation of the propositional theory T is a sound AMC computation.*

Proof. We show that $\text{EVAL}(N_T, \oplus, \otimes, e^{\oplus}, e^{\otimes}, \alpha)$ for an sd-DNNF representation N_T of the theory T computes $\mathbf{A}(T)$ with respect to all variables in N_T:

1. Line 2: $\mathbf{A}(\top) = \bigoplus_{I \in \{\emptyset\}} \bigotimes_{l \in I} \alpha(l) = e^{\otimes}$
2. Line 3: $\mathbf{A}(\bot) = \bigoplus_{I \in \{\}} \bigotimes_{l \in I} \alpha(l) = e^{\oplus}$
3. Line 4: $\mathbf{A}(l) = \bigoplus_{I \in \{\{l\}\}} \bigotimes_{k \in I} \alpha(k) = \alpha(l)$

Due to associativity and commutativity of the semiring operators, operands of each summation and each multiplication can be evaluated in arbitrary order. We therefore restrict ourselves to binary disjunction and conjunction nodes here. Given sound evaluation for subcircuits ϕ_i with variables $\mathcal{V}_i$ and models $\mathcal{M}_i$ with respect to these variables, we obtain:

4. Lines 5-6: Disjunction node $\phi_1 \vee \phi_2$: As $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2 = \mathcal{V}_i$ due to smoothness, we obtain $\mathcal{M}(\phi_1 \vee \phi_2) = \mathcal{M}_1 \cup \mathcal{M}_2$, which is a disjoint union due to determinism. Therefore, $\mathbf{A}(\phi_1) \oplus \mathbf{A}(\phi_2) = \bigoplus_{i=1,2} \bigoplus_{\mathcal{M}_i} \bigotimes_{l \in I} \alpha(l) = \bigoplus_{\mathcal{M}_1 \cup \mathcal{M}_2} \bigotimes_{l \in I} \alpha(l) = \mathbf{A}(\phi_1 \vee \phi_2)$.
5. Lines 7-8: Conjunction node $\phi_1 \wedge \phi_2$: As $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$ due to decomposability, the set $\mathcal{M}(\phi_1 \wedge \phi_2)$ of models of the conjunction is simply the set of all unions of models of its parts. Together with distributivity, we get $\mathbf{A}(\phi_1) \otimes \mathbf{A}(\phi_2) = \bigotimes_{i=1,2} \bigoplus_{\mathcal{M}_i} \bigotimes_{l \in I} \alpha(l) = \bigoplus_{\mathcal{M}(\phi_1 \wedge \phi_2)} \bigotimes_{l \in I} \alpha(l) = \mathbf{A}(\phi_1 \wedge \phi_2)$.

□

Clearly, the soundness of AMC evaluation on `sd-DNNF` depends on all three properties of this subclass of `NNF`. On the other hand, circuits without these properties may be exponentially smaller and thus allow for more efficient inference. In the following, we therefore analyze evaluation in the absence of these properties, which allows us to identify characteristics of the semiring and labeling function that ensure sound evaluation on the corresponding classes of circuits.

3.2 Evaluation on other Decomposable Circuits

If a circuit is not deterministic, children of a disjunction node may have common models, in which case evaluation sums over such shared models multiple times. For instance, consider the circuit in Figure 1b with **PROB**, $\alpha(a) = 0.6$ and $\alpha(b) = 0.3$. Evaluation on this circuit results in $0.6 + 0.3 = 0.9$ for the right disjunction node, while $\mathbf{A}(a \vee b) = 0.6 \cdot 0.3 + 0.4 \cdot 0.3 + 0.6 \cdot 0.7 = 0.72$.

Definition 9 (Idempotent Operator)**.** A binary operator $\odot$ over a set $\mathcal{A}$ is *idempotent* iff $\forall a \in \mathcal{A} : a \odot a = a$.

Theorem 3 (`s-DNNF` Evaluation)**.** *Evaluating an* `s-DNNF` *representation of the propositional theory T for a semiring with idempotent $\oplus$ is a sound AMC computation.*

Proof. Reconsider point (4) of the proof of Theorem 2. With smoothness, but without determinism, $\mathcal{M}_1(\phi_1) \cup \mathcal{M}_2(\phi_2)$ is no longer a union of disjoint sets, and $\mathbf{A}(\phi_1) \oplus \mathbf{A}(\phi_2) = \bigoplus_{i=1,2} \bigoplus_{\mathcal{M}_i} \bigotimes_{l \in I} \alpha(l)$ sums over the models in $\mathcal{M}_1(\phi_1) \cap \mathcal{M}_2(\phi_2)$ twice. Due to associativity and commutativity, this is sound for idempotent $\oplus$. □

If a circuit is not smooth, the children of a disjunction node may use different sets of variables. Each model of a child node corresponds to a set of models for the full set of variables, but evaluation on a non-smooth circuit ignores the labels of unmentioned variables. For instance, consider the circuit in Figure 1b with **MPE**, $\alpha(a) = 0.6$ and $\alpha(b) = 0.3$. Evaluating the right disjunction node of this circuit results in $\max(0.6, 0.3) = 0.6$, while $\mathbf{A}(a \vee b) = \max(0.6 \cdot 0.3, 0.4 \cdot 0.3, 0.6 \cdot 0.7) = 0.42$.

Definition 10 (Neutral $(\oplus, \alpha)$)**.** A semiring addition and labeling function pair $(\oplus, \alpha)$ is *neutral* iff $\forall v \in \mathcal{V} : \alpha(v) \oplus \alpha(\neg v) = e^{\otimes}$.

Theorem 4 (`d-DNNF` Evaluation)**.** *Evaluating a* `d-DNNF` *representation of the propositional theory T for a semiring and labeling function with neutral $(\oplus, \alpha)$ is a sound AMC computation.*

Proof. For lines 5-6 (point (4) of the proof of Theorem 2) to be sound, the sum of the AMCs computed by the children over their sets of variables V_i has to be equal to the AMC of the entire disjunction over the full set of variables $V_1 \cup V_2$. Given the child AMC $\mathbf{A}_{V_i}(\phi_i)$, adding a variable v to V_i replaces each model I of ϕ_i by two models $I^+ = I \cup \{v\}$ and $I^- = I \cup \{\neg v\}$. Due to distributivity, commutativity and the neutral sum property, the algebraic sum of these two models equals the AMC of the original model:

$$\begin{aligned}\mathbf{A}_{V_i \cup \{v\}}(I) &= \mathbf{A}_{V_i \cup \{v\}}(I^+) \oplus \mathbf{A}_{V_i \cup \{v\}}(I^-) \\ &= (\alpha(v) \oplus \alpha(\neg v)) \otimes \bigotimes_{l \in I} \alpha(l) \\ &= \bigotimes_{l \in I} \alpha(l) = \mathbf{A}_{V_i}(I)\end{aligned}$$

Evaluation therefore computes $\mathbf{A}_{V_1}(\phi_1) \oplus \mathbf{A}_{V_2}(\phi_2) = \mathbf{A}_{V_1 \cup V_2}(\phi_1) \oplus \mathbf{A}_{V_1 \cup V_2}(\phi_2)$, which due to determinism is equal to $\mathbf{A}_{V_1 \cup V_2}(\phi_1 \vee \phi_2)$. □

Note that from a practical point of view, non-neutral $(\oplus, \alpha)$ does not influence the tractability of inference, as any `NNF` can be smoothed in polytime preserving determinism and decomposability (Darwiche and Marquis, 2002).

The previous two results can directly be combined for `DNNF` circuits that are neither smooth nor deterministic.

Theorem 5 (`DNNF` Evaluation)**.** *Evaluating a* `DNNF` *representation of the propositional theory T for a semiring and labeling function with idempotent and neutral $(\oplus, \alpha)$ is a sound AMC computation.*

Proof. Reconsider point (4) of the proof of Theorem 2. Due to neutral $(\oplus, \alpha)$, values for all children of a disjunction node are correct with respect to the full set of variables (Theorem 4). Due to idempotent $\oplus$, multiple occurrences of the same model do not influence the result (Theorem 3). □

This completes the left part of Table 2, where no conditions are imposed on semiring multiplication.

3.3 Evaluation on Non-Decomposable Circuits

If a circuit is not decomposable, the children of a conjunction node may share variables. In this case, simply combining all pairs of their models may produce sets of literals that are not models, because they either contain contradicting literals, or several copies of the same literal, which results in erroneous extra multiplications. For instance, in Figure 1b, $\{\neg a, b\}$ is a model of both disjunction nodes, and $\{a, b\}$ of the right one only. The conjunction node sums among others the products $\alpha(\neg a) \otimes \alpha(b) \otimes \alpha(a) \otimes \alpha(b)$, which does not correspond to a model, and $\alpha(\neg a) \otimes \alpha(b) \otimes \alpha(\neg a) \otimes \alpha(b)$, where labels of both literals are multiplied twice.

Definition 11 (Consistency-Preserving $(\otimes, \alpha)$)**.** A semiring multiplication and labeling function pair $(\otimes, \alpha)$ is *consistency-preserving* iff $\forall v \in \mathcal{V} : \alpha(v) \otimes \alpha(\neg v) = e^{\oplus}$.

Theorem 6 (`sd-NNF` Evaluation)**.** *Evaluating an* `sd-NNF` *representation of the propositional theory T for a semiring and labeling function with idempotent and consistency-preserving $(\otimes, \alpha)$ is a sound AMC computation.*

Proof. Reconsider point (5) of the proof of Theorem 2. The set of models of $\phi_1 \wedge \phi_2$ contains exactly all pairwise combinations of models of its parts that agree on all shared variables. The circuit evaluates the AMC of $\phi_1 \wedge \phi_2$ as $\mathbf{A}(\phi_1) \otimes \mathbf{A}(\phi_2)$, which due to distributivity is the sum over all pairwise combinations of models. Without decomposability, each such combination I contains two literals for each $v \in \mathcal{V}_1 \cap \mathcal{V}_2$. As $\otimes$ is associative, commutative and idempotent, repeated occurrences of a literal l in $\otimes_{i \in I} \alpha(i)$ do not affect the result. If $\{l, \neg l\} \subseteq I$, $\otimes_{i \in I} \alpha(i)$ includes a multiplication by $\alpha(l) \otimes \alpha(\neg l) = e^{\oplus}$, which in a semiring means $\otimes_{i \in I} \alpha(i) = e^{\oplus}$. Such inconsistent I thus do not contribute to the semiring sum. □

Theorem 6 affects only conjunction nodes, whereas Theorems 3, 4 and 5 only affect disjunction nodes. Their combination thus extends our results to non-decomposable circuits that do not satisfy (one of) the other two properties either:

Theorem 7 (`s-NNF`, `d-NNF`, and `NNF` Evaluations)**.** *For a semiring and labeling function with idempotent and consistency preserving $(\otimes, \alpha)$, evaluating the following representation of the propositional theory T is a sound AMC computation:*

- `s-NNF` *if $\oplus$ is idempotent*
- `d-NNF` *if $(\oplus, \alpha)$ is neutral*
- `NNF` *if $(\oplus, \alpha)$ is idempotent and neutral*

This completes the right part of Table 2, where using non-decomposable circuits is possible as a consequence of restrictions on semiring multiplication.

Given a new AMC instance, this table allows one to immediately choose the appropriate type of circuit for efficient evaluation. For the examples discussed here, cf. Table 1, these are: NNF for $\texttt{OBDD}_<$, DNNF for **SAT**, **S-PATH**, **W-PATH** and **FUZZY**, d-DNNF for **PROB**, **SENS** and **GRAD**, s-DNNF for **MPE** and k**WEIGHT**, and sd-DNNF for **#SAT** and **WMC**.

3.4 Discussion

The results summarized in Table 2 generalize the complexity results for **SAT** and **#SAT**, provide more succinct types of circuits for inference in algebraic Prolog, and show that all circuits that are practically relevant for AMC are well-studied in the knowledge compilation map. We now address these points in more detail.

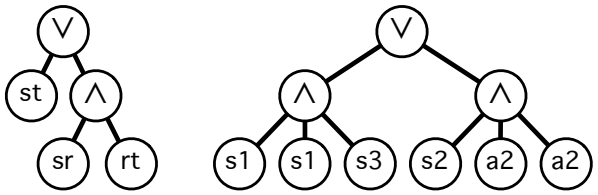
First, Darwiche and Marquis (2002) show that **SAT** can be evaluated in polynomial time on DNNF, while **#SAT** can be evaluated in polynomial time on d-DNNF, as smoothing is possible in polynomial time. Our Theorem 5 generalizes soundness of DNNF evaluation to all semirings and labeling functions with idempotent and neutral $(\oplus, \alpha)$, which includes **SAT**, while our Theorem 2 generalizes soundness of sd-DNNF evaluation to all semirings and labeling functions, including those with non-idempotent, non-neutral $(\oplus, \alpha)$, such as **#SAT**. As discussed above, the complexity of evaluation is polynomial if each semiring operation has unit cost.

Second, Kimmig et al. (2011) reduce inference in algebraic Prolog (aProbLog) to AMC evaluation on disjunctive normal form (DNF). For non-idempotent addition, the DNF is compiled into an ordered binary decision diagram (OBDD) if its conjunctions are not mutually exclusive. For non-neutral $(\oplus, \alpha)$, circuits are smoothed before evaluation. This results in the following settings:

	neutral $(\oplus, \alpha)$	non-neutral $(\oplus, \alpha)$
idempotent $\oplus$	DNF	s-DNF
non-idempotent $\oplus$	d-DNF OBDD	sd-DNF s-OBDD

Table 2 uses the same characteristics of semiring operators and labeling function, but does not assume a DNF as starting point and thus also does not rely on properties of such a DNF. Its left half generalizes the aProbLog scheme to more succinct superclasses, namely (s-)DNNF instead of (s-)DNF, and (s)d-DNNF instead of (s)d-DNF or (s-)OBDD.

Third, we observe that while there are interesting inference tasks that allow for sound evaluation on the more succinct class of DNNF instead of the general sd-DNNF evaluation, the conditions for sound evaluation on non-decomposable circuits are too strict in most practical cases. This is in line with the knowledge compilation map, which excludes non-decomposable circuits (with the exception of the most general class NNF) as they do not support any of the studied tasks in polytime (Darwiche and Marquis, 2002).



(a) Path problem (b) Semiring grammar

Figure 2: NNF circuits for an algebraic path problem and a semiring grammar. See Section 3.5 for details.

3.5 AMC and Algebraic Derivation Counting

While AMC is a sum over models, many other semiring-based tasks require a sum over derivations, that is, sequences of possibly repeated variables. Examples include algebraic path problems (Baras and Theodorakopoulos, 2010), semiring parsing (Goodman, 1999), provenance semirings for positive relational algebra queries in databases (Green et al., 2007), and semiring-weighted dynamic programs (Eisner et al., 2005). We refer to this type of task as *algebraic derivation counting* (ADC). While AMC and ADC appear very similar, they cannot easily be exchanged, as we illustrate next. We restrict the discussion to finite ADC.

Where AMC is based on a MODS representation, that is, a smooth, deterministic DNF, ADC is based on a type of NNF that is not included in the knowledge compilation map: a disjunction of conjunctions of possibly repeated variables. Because of the repeated variables, this is not a DNF in general. Figure 2 shows two examples of such circuits. The NNF in Figure 2a encodes the two paths between nodes s and t in a graph with three edges (s,t), (s,r) and (r,t). It is decomposable, as the conjunction does not repeat variables, but neither smooth nor deterministic and thus a DNNF. The NNF in Figure 2b encodes the two derivations of aa in a context-free grammar with rules $S \to aS \mid AA \mid \epsilon$ and $A \to AA \mid a$. Variable x_i denotes an application of the ith production rule for a non-terminal X, that is, the right conjunction in the circuit encodes the derivation $S \to AA \to aA \to aa$. The circuit is neither decomposable nor smooth, and also not deterministic, even though the derivations in the underlying grammar setting are mutually exclusive (cf. below).

In order to represent an ADC task as an AMC task, the labeling function α needs to be extended to negative literals. Together with the semiring, such an extension determines one of the settings of Table 2. If the ADC's NNF belongs to the class of circuits corresponding to this setting, the ADC and AMC tasks coincide. Clearly, this is always the case for commutative semirings and labeling functions with idempotent, consistency-preserving $(\otimes, \alpha)$ and idempotent, neutral $(\oplus, \alpha)$ such as for instance $\texttt{OBDD}_<$. However, as noted above, such tasks are rare.

For an ADC instance with idempotent addition defined on a DNNF, that is, without repeated variables in conjunctions, an equivalent AMC instance can be constructed if α can be extended such that $(\oplus, \alpha)$ is neutral. For instance, consider the circuit in Figure 2a in a shortest path setting. By setting $\alpha(\neg v) = 0$ for all variables v, the **S-PATH** semiring in Table 1 provides an equivalent AMC task.

As the NNF circuits of ADC do not contain negation, they are not deterministic. However, the underlying tasks often impose additional constraints on their variables. For instance, in the grammar example in Figure 2b, the leftmost children $s1$ and $s2$ of the two conjunction nodes are in fact mutually exclusive, as only one of the right hand sides for S can be chosen as first step in a derivation. Such an n-ary variable can be encoded using n binary variables by adding corresponding constraints to the theory that restrict legal value assignments. For some semirings, for instance **PROB**, it is sufficient to adapt the labels of these variables without adding constraints. It is an open question under which general conditions such label transformations are possible.

Conversely, every AMC task can be trivially represented as an ADC task by introducing a derivation for each model. However, this is clearly not desirable from a complexity point of view. Baras and Theodorakopoulos (2010) provide an ADC encoding of network reliability under probabilistic edge failure, that is, the **PROB** AMC task for a positive propositional formula. They essentially modify multiplication to filter repeated literals, and addition to subtract shared models of its operands, which drastically increases complexity of these operations. Again, it is an open question under which general conditions such transformations are possible.

4 Conclusions and Future Work

We have introduced the task of algebraic model counting, which generalizes weighted model counting to a semiring setting and thus to various types of labels, including numerical ones as used in WMC, but also sets, polynomials, or Boolean formulae. We have shown that evaluating AMC is sound on sd-DNNF circuits, which are known to be more succinct than the MODS language used in its definition. Furthermore, we have provided characteristics of AMC tasks that guarantee sound evaluation on more succinct classes of circuits. This classification not only provides a means of directly choosing a circuit type that allows for efficient inference given a new AMC task, but also generalizes a number of known results and provides a framework to map restricted types of algebraic derivation counts onto AMC tasks.

Given the results presented here, it is worth investigating which other algebraic representations can be reduced to algebraic model counting. Another line of future work concerns the introduction of additional operators that would make it possible to express additional tasks, for instance, partial MAP, which requires a maximization operator in addition to summation and multiplication.

Acknowledgements A. Kimmig and G. Van den Broeck are supported by the Research Foundation Flanders (FWO Vlaanderen).

References

Bacchus, F., Dalmao, S., and Pitassi, T. (2009). Solving #SAT and Bayesian inference with backtracking search. *Journal of Artificial Intelligence Research*, 34(1):391–442.

Baras, J. S. and Theodorakopoulos, G. (2010). *Path Problems in Networks*, volume 3 of *Synthesis Lectures on Communication Networks*. Morgan & Claypool Publishers.

Darwiche, A. (2009). *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press.

Darwiche, A. and Marquis, P. (2002). A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264.

Eisner, J. (2002). Parameter estimation for probabilistic finite-state transducers. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1–8, Philadelphia.

Eisner, J., Goldlust, E., and Smith, N. (2005). Compiling Comp Ling: Weighted dynamic programming and the Dyna language. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 281–290.

Fierens, D., Van den Broeck, G., Thon, I., Gutmann, B., and De Raedt, L. (2011). Inference in probabilistic logic programs using weighted CNF's. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 211–220.

Goodman, J. (1999). Semiring parsing. *Computational Linguistics*, 25(4):573–605.

Green, T. J., Karvounarakis, G., and Tannen, V. (2007). Provenance semirings. In *Proceedings of the 26th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, pages 31–40.

Jha, A. K. and Suciu, D. (2011). Knowledge compilation meets database theory: compiling queries to decision diagrams. In *Proceedings of the 14th International Conference on Database Theory (ICDT)*, pages 162–173.

Kanagal, B., Li, J., and Deshpande, A. (2011). Sensitivity analysis and explanations for robust query evaluation in probabilistic databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 841–852.

Kimmig, A., Van den Broeck, G., and De Raedt, L. (2011). An algebraic Prolog for reasoning about possible worlds. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI)*, pages 209–214.

Larrosa, J., Oliveras, A., and Rodríguez-Carbonell, E. (2010). Semiring-induced propositional logic: Definition and basic algorithms. In Clarke, E. M. and Voronkov, A., editors, *Revised Selected Papers of the 16th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, volume 6355 of *Lecture Notes in Computer Science*, pages 332–347. Springer.

Mateescu, R., Dechter, R., and Marinescu, R. (2008). And/or multi-valued decision diagrams (AOMDDs) for graphical models. *Journal of Artificial Intelligence Research*, 33:465–519.

Meseguer, P., Rossi, F., and Schiex, T. (2006). Soft constraints. In Rossi, F., Van Beek, P., and Walsh, T., editors, *Handbook of constraint programming*, pages 281–328. Elsevier Science.

Park, J. D. (2002). Using weighted MAX-SAT engines to solve MPE. In *Proceedings of the 18th AAAI Conference on Artificial Intelligence (AAAI)*, pages 682–687.

Sang, T., Beame, P., and Kautz, H. (2005). Solving Bayesian networks by weighted model counting. In *Proceedings of the 20th AAAI Conference on Artificial Intelligence (AAAI)*, pages 475–482.

Sanner, S. and McAllester, D. A. (2005). Affine algebraic decision diagrams (AADDs) and their application to structured probabilistic inference. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1384–1390.

Wilson, N. (2005). Decision diagrams for the computation of semiring valuations. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 331–336.