

# Efficient Weighted Model Integration via SMT-Based Predicate Abstraction

Paolo Morettin, Andrea Passerini, Roberto Sebastiani

DISI, University of Trento, Italy.

paolo.morettin@unitn.it, andrea.passerini@unitn.it, roberto.sebastiani@unitn.it

## Abstract

Weighted model integration (WMI) is a recent formalism generalizing weighted model counting (WMC) to run probabilistic inference over hybrid domains, characterized by both discrete and continuous variables and relationships between them. Albeit powerful, the original formulation of WMI suffers from some theoretical limitations, and it is computationally very demanding as it requires to explicitly enumerate all possible models to be integrated over. In this paper we present a novel general notion of WMI, which fixes the theoretical limitations and allows for exploiting the power of SMT-based predicate abstraction techniques. A novel algorithm combines a strong reduction in the number of models to be integrated over with their efficient enumeration. Experimental results on synthetic and real-world data show drastic computational improvements over the original WMI formulation as well as existing alternatives for hybrid inference.

## 1 Introduction

Weighted model counting (WMC) is the task of computing the weighted sum of all satisfying assignments of a propositional formula, where weights are associated to models and are typically factorized into the product of weights of individual variables. In recent years, WMC was shown to be an effective solution for addressing probabilistic inference in a wide spectrum of formalisms [Chavira and Darwiche, 2008; Choi *et al.*, 2013; Suciú *et al.*, 2011].

An inherent limitation of WMC is that it can only deal with discrete distributions. In order to overcome this restriction, weighted model integration (WMI) [Belle *et al.*, 2015] was recently introduced as a formulation generalizing WMC to deal with hybrid domains, characterized by both discrete and continuous variables and their relationships. The formalism relies on satisfiability modulo theory (SMT) [Barrett *et al.*, 2009] technology, which allows to reason about the satisfiability of formulas involving e.g. linear constraints over integers or reals. WMI works by replacing the weighted sum of models of a propositional formula with a sum of integrals

over weight functions defined over the models of an SMT formula. Weight functions here play the role of (un-normalized) densities, whereas logic formulas in the model define the integration domain.

In this paper, we elaborate on the notion of WMI and provide a refined formalization fixing some theoretical and practical limitations of the original definition. The novel formalization guarantees that equivalent formulas have the same WMI, and that probabilistic manipulations like marginalization are always consistent. A novel notion of conditional weight function allows for a more effective decomposition of the function into its separately integrable components. Building on the properties of this novel formulation, we devise an efficient algorithm combining a strong reduction in the number of models to be generated and integrated over, with efficiency in enumerating these models. The key ingredient is the use of SMT-based *predicate abstraction* techniques [Graf and Saïdi, 1997; Lahiri *et al.*, 2006; Cavada *et al.*, 2007] to efficiently and effectively generate the set of models needed to compute the exact integral. Our experimental evaluation confirms that the approach is drastically faster than existing alternatives over synthetic and real-world problems, and that both aspects contribute to the gain.

## 2 Background

### 2.1 SMT, AllSMT and Predicate Abstraction

We assume the reader is familiar with the basic syntax, semantics and results of propositional and first-order logics. We adopt some terminology and concepts from Satisfiability Modulo Theories (SMT), which we briefly summarize below (see [Sebastiani, 2007; Barrett *et al.*, 2009] for details).

Our context is that of SMT on quantifier-free formulas in the theory of *linear arithmetic over the reals*,  $\mathcal{LRA}$ .  $\mathbb{R}$  denotes the set of real values and  $\mathbb{B} \stackrel{\text{def}}{=} \{\top, \perp\}$  the set of Boolean values.  $\mathcal{LRA}$  formulas are combinations by means of the standard Boolean operators  $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$  of atomic propositions  $A_i \in \mathbb{B}$  (aka Boolean atoms/variables) and of  $\mathcal{LRA}$  atomic formulas (aka  $\mathcal{LRA}$  atoms) in the form  $(\sum_i c_i x_i \bowtie c)$ , s.t.  $x_i$  are variables in  $\mathbb{R}$ ,  $c_i$  are rational values and  $\bowtie \in \{=, \neq, \geq, \leq, >, <\}$ , with their usual semantics. “ $\models_{\mathcal{LRA}}$ ” denotes entailment in  $\mathcal{LRA}$  (e.g.  $(x \geq 2) \models_{\mathcal{LRA}} (x \geq 1)$ ), whereas “ $\models_{\mathbb{B}}$ ” denotes tautological entailment (e.g.  $A_1 \wedge (x \geq 2) \models_{\mathbb{B}} ((A_1 \vee (x \leq 1)) \wedge (\neg A_1 \vee (x \geq 2)))$ ). Notice

that  $\models_{\mathbb{B}}$  is strictly stronger than  $\models_{\mathcal{LRA}}$ .  $\varphi_1, \varphi_2$  are  $\mathcal{LRA}$ -equivalent, written  $\varphi_1 \Leftrightarrow_{\mathcal{LRA}} \varphi_2$ , iff  $\models_{\mathcal{LRA}} (\varphi_1 \leftrightarrow \varphi_2)$ .

We frequently use the following abbreviations. Let  $t, t_i$  be  $\mathcal{LRA}$  terms,  $\varphi, \varphi_i$  be  $\mathcal{LRA}$  formulas, and  $I = [l, u]$  be some interval; then we use “[ $t \in I$ ]” as a shortcut for the formula  $(t \geq l) \wedge (t \leq u)$ , possibly with “ $>$ ” or “ $<$ ” if some end of the interval is open; we use “(If  $\varphi$  Then  $t_1$  Else  $t_2$ )” to represent an if-then-else conditional expression, returning the value of  $t_1$  if  $\varphi$  holds, the value of  $t_2$  otherwise; we use “(Case  $\varphi_1 : t_1; \varphi_2 : t_2; \dots$ )” to generalize the if-then-else to the case of multiple mutually-exclusive conditions. We use “OneOf( $\{\varphi_1, \dots, \varphi_n\}$ )” as a shortcut for the formula  $(\bigvee_{i=1}^n \varphi_i) \wedge \bigwedge_{1 \leq i < j \leq n} \neg(\varphi_i \wedge \varphi_j)$ , i.e., exactly one  $\varphi_i$  holds.

Given a set of  $\mathcal{LRA}$  formulas  $\Psi \stackrel{\text{def}}{=} \{\psi_1, \dots, \psi_K\}$ , we call a total [resp. partial] truth assignment  $\mu$  for  $\Psi$  any total [resp. partial] map from  $\Psi$  to  $\mathbb{B}$ . We represent  $\mu$  alternatively and equivalently either as a set or a conjunction:

$$\begin{aligned} \mu &\stackrel{\text{def}}{=} \{\psi \mid \psi \in \Psi, \mu(\psi) = \top\} \cup \{\neg\psi \mid \psi \in \Psi, \mu(\psi) = \perp\}, \\ \mu &\stackrel{\text{def}}{=} \bigwedge_{\psi \in \Psi, \mu(\psi) = \top} \psi \wedge \bigwedge_{\psi \in \Psi, \mu(\psi) = \perp} \neg\psi. \end{aligned}$$

We denote by  $\mathbb{B}^K$  the set of all truth assignments over  $\Psi$ .

Let  $\mathbf{x} \stackrel{\text{def}}{=} \{x_1, \dots, x_N\} \in \mathbb{R}^N$  and  $\mathbf{A} \stackrel{\text{def}}{=} \{A_1, \dots, A_M\} \in \mathbb{B}^M$  for some  $N$  and  $M$ . Consider a generic  $\mathcal{LRA}$  formula  $\varphi$  on (subsets of)  $\mathbf{x}$  and  $\mathbf{A}$ , and let  $\Psi \stackrel{\text{def}}{=} \text{Atoms}(\varphi)$ , i.e. the set of propositional and  $\mathcal{LRA}$  atoms occurring in  $\varphi$ . Given a truth assignment  $\mu$  for  $\text{Atoms}(\varphi)$ , we denote by  $\mu^{\mathbf{A}}$  and  $\mu^{\mathcal{LRA}}$  its two components on the Boolean atoms in  $\mathbf{A}$  and on the  $\mathcal{LRA}$  atoms respectively, so that  $\mu = \mu^{\mathbf{A}} \wedge \mu^{\mathcal{LRA}}$ . (E.g., if  $\mu = A_1 \wedge \neg A_2 \wedge (x \geq 1) \wedge \neg(x \geq 3)$ , then  $\mu^{\mathbf{A}} = A_1 \wedge \neg A_2$  and  $\mu^{\mathcal{LRA}} = (x \geq 1) \wedge \neg(x \geq 3)$ ). Importantly, and unlike with pure propositional logic,  $\mu$  can be  $\mathcal{LRA}$ -unsatisfiable due to its  $\mu^{\mathcal{LRA}}$  component (e.g.  $\mu \stackrel{\text{def}}{=} \neg A_1 \wedge (x_1 + x_2 = 3) \wedge \neg(x_1 + x_2 \geq 2)$ ). A (possibly partial) truth assignment  $\mu$  propositionally satisfies  $\varphi$  iff  $\mu \models_{\mathbb{B}} \varphi$ . The SMT problem for  $\varphi$  in  $\mathcal{LRA}$  is the problem of checking the existence of a  $\mathcal{LRA}$ -satisfiable assignment  $\mu$  s.t.  $\mu \models_{\mathbb{B}} \varphi$ .

We denote by  $\mathcal{T}\mathcal{A}(\varphi) \stackrel{\text{def}}{=} \{\mu_1, \dots, \mu_j, \dots\}$  the set of all  $\mathcal{LRA}$ -satisfiable total truth assignments  $\mu_j$  on  $\text{Atoms}(\varphi)$  propositionally satisfying  $\varphi$ .  $\mathcal{T}\mathcal{A}(\varphi)$  is unique. We denote by  $\mathcal{A}(\varphi) \stackrel{\text{def}}{=} \{\mu_1, \dots, \mu_j, \dots\}$  any set of  $\mathcal{LRA}$ -satisfiable possibly-partial truth assignments  $\mu_j$  propositionally satisfying  $\varphi$ , s.t. every pair  $\mu_i, \mu_j$  assign opposite truth values to at least one element, i.e.,  $\mu_i \wedge \mu_j \models_{\mathbb{B}} \perp$  (hence  $\mu_i \wedge \mu_j \models_{\mathcal{LRA}} \perp$ ).  $\mathcal{A}(\varphi)$  is not unique, and  $\mathcal{T}\mathcal{A}(\varphi)$  is a particular case of  $\mathcal{A}(\varphi)$ . The disjunction of the  $\mu_j$ 's in  $\mathcal{A}(\varphi)$  is  $\mathcal{LRA}$ -equivalent to  $\varphi$  (see e.g. [Sebastiani, 2007]):

$$\varphi \Leftrightarrow_{\mathcal{LRA}} \bigvee_{\mu_j \in \mathcal{A}(\varphi)} \mu_j. \quad (1)$$

The AllSMT problem for  $\varphi$  in  $\mathcal{LRA}$  is the problem of enumerating one set  $\mathcal{A}(\varphi)$  matching the above definition.

For example, if  $\varphi \stackrel{\text{def}}{=} (x \leq 0) \vee (x \geq 1)$ , then  $\mathcal{T}\mathcal{A}(\varphi) \stackrel{\text{def}}{=} \{(x \leq 0) \wedge \neg(x \geq 1), \neg(x \leq 0) \wedge (x \geq 1)\}$ . ( $\mathcal{T}\mathcal{A}(\varphi)$  does not contain  $(x \leq 0) \wedge (x \geq 1)$  as it is not  $\mathcal{LRA}$ -satisfiable.) The other admissible  $\mathcal{A}(\varphi)$ 's are:  $\{(x \leq 0), \neg(x \leq 0) \wedge (x \geq 1)\}$  and  $\{(x \geq 1), (x \leq 0) \wedge \neg(x \geq 1)\}$ . The set of unary truth assignments  $\{(x \leq 0), (x \geq 1)\}$  is not an admissible  $\mathcal{A}(\varphi)$  be-

cause there is no atom to which they assign different truth values, i.e.,  $(x \leq 0) \wedge (x \geq 1) \not\models_{\mathbb{B}} \perp$ .

**Definition 1** Let  $\varphi(\mathbf{x}, \mathbf{A})$  be a  $\mathcal{LRA}$ -formula on  $\mathbf{x}$  and  $\mathbf{A}$ ; let  $\Psi \stackrel{\text{def}}{=} \{\psi_1, \dots, \psi_K\}$  be a set of  $\mathcal{LRA}$ -formulas over  $\mathbf{x}$  and  $\mathbf{A}$ , and  $\mathbf{B} \stackrel{\text{def}}{=} \{B_1, \dots, B_K\}$  be a set of fresh atomic propositions s.t.  $\mathbf{A} \cap \mathbf{B} = \emptyset$ . Then we call a **Predicate Abstraction** of  $\varphi$  wrt.  $\Psi$  on  $\mathbf{B}$ , namely  $\text{PredAbs}_{[\varphi, \Psi]}(\mathbf{B})$ , (any formula equivalent to) the propositional formula

$$\exists \mathbf{A} \exists \mathbf{x}. \left( \varphi(\mathbf{x}, \mathbf{A}) \wedge \bigwedge_{k=1}^K (B_k \leftrightarrow \psi_k(\mathbf{x}, \mathbf{A})) \right). \quad (2)$$

We define  $\text{PredAbs}_{[\varphi]}(\Psi) \stackrel{\text{def}}{=} \text{PredAbs}_{[\varphi, \Psi]}(\mathbf{B})[\mathbf{B} \leftarrow \Psi]$ , that is, the  $\mathcal{LRA}$ -formula obtained from  $\text{PredAbs}_{[\varphi, \Psi]}(\mathbf{B})$  by substituting each  $B_k$  with its corresponding  $\psi_k$ . Importantly, if  $\Psi$  is  $\mathbf{A}$ , then  $\text{PredAbs}_{[\varphi]}(\mathbf{A})$  reduces to  $\exists \mathbf{x}. \varphi(\mathbf{x}, \mathbf{A})$ .

Notice that in Definition 1 the formulas  $\psi_i$  are neither necessarily atomic, nor necessarily sub-formulas of  $\varphi$ .  $\text{PredAbs}_{[\varphi, \Psi]}(\mathbf{B})$  defines an equivalence class of Boolean formulas over  $\mathbf{B}$ , i.e., (2) may represent many syntactically-different albeit logically-equivalent Boolean formulas.

Very effective SMT-based techniques for computing  $\text{PredAbs}_{[\varphi, \Psi]}(\mathbf{B})$ —and hence  $\text{PredAbs}_{[\varphi]}(\Psi)$ —have been proposed in the literature (e.g. [Lahiri *et al.*, 2006; Cavada *et al.*, 2007]) and are implemented in modern SMT solvers like MATHSAT5 [Cimatti *et al.*, 2013]. Very importantly for our purposes, these techniques work by iteratively producing a set of propositional truth assignments on  $\mathbf{B}$ , which are then disjoined as in (1). In particular MATHSAT5, on demand, can either produce a set of *total* assignments  $\mathcal{T}\mathcal{A}(\text{PredAbs}_{[\varphi, \Psi]}(\mathbf{B}))$ , or a set of *partial* ones  $\mathcal{A}(\text{PredAbs}_{[\varphi, \Psi]}(\mathbf{B}))$  by means of assignment-minimization techniques.

**Example 1** Consider  $\mathbf{A} \stackrel{\text{def}}{=} \{A_1\}$ ,  $\mathbf{x} \stackrel{\text{def}}{=} \{x_1, x_2\}$ ,  $\varphi \stackrel{\text{def}}{=} A_1 \wedge (x_1 + x_2 > 12)$ ,  $\psi_1 \stackrel{\text{def}}{=} (x_1 + x_2 = 2)$ ,  $\psi_2 \stackrel{\text{def}}{=} (x_1 - x_2 < 10)$ .

$$\begin{aligned} \text{PredAbs}_{[\varphi, \Psi]}(\mathbf{B}) &\stackrel{\text{def}}{=} \exists A_1. \left[ \begin{array}{l} A_1 \wedge (x_1 + x_2 > 12) \wedge \\ (B_1 \leftrightarrow (x_1 + x_2 = 2)) \wedge \\ (B_2 \leftrightarrow (x_1 - x_2 < 10)) \end{array} \right] \\ &= (\neg B_1 \wedge \neg B_2) \vee (\neg B_1 \wedge B_2) \quad (3) \\ &= \neg B_1. \quad (4) \end{aligned}$$

$$\text{PredAbs}_{[\varphi]}(\Psi) = \neg(x_1 + x_2 = 2). \quad (5)$$

(3) is built as the disjunction of *total* assignments on  $\mathbf{B}$ , whereas (4) is built as the disjunction of *partial* ones—e.g., by the clause-minimization techniques in MATHSAT5—s.t.:

$$\begin{aligned} \mathcal{T}\mathcal{A}(\text{PredAbs}_{[\varphi, \Psi]}(\mathbf{B})) &= \{(\neg B_1 \wedge \neg B_2), (\neg B_1 \wedge B_2)\} \\ \mathcal{A}(\text{PredAbs}_{[\varphi, \Psi]}(\mathbf{B})) &= \{(\neg B_1)\} \end{aligned}$$

$$\mathcal{T}\mathcal{A}(\text{PredAbs}_{[\varphi]}(\Psi)) = \left\{ \begin{array}{l} (\neg(x_1 + x_2 = 2) \wedge \\ \neg(x_1 - x_2 < 10)), \\ (\neg(x_1 + x_2 = 2) \wedge \\ (x_1 - x_2 < 10)) \end{array} \right\}$$

$$\mathcal{A}(\text{PredAbs}_{[\varphi]}(\Psi)) = \{(\neg(x_1 + x_2 = 2))\}$$

Notice also that the other two total assignments,  $B_1 \wedge B_2$  and  $B_1 \wedge \neg B_2$ , are not part of (3) because they force the formula to be  $\mathcal{LRA}$ -unsatisfiable.  $\diamond$

## 2.2 Weighted Model Counting and Integration

WMC is the task of computing the weighted sum of all satisfying assignments of a propositional formula, with weights typically factorized as a product of weights over literals.

**Definition 2 (Weighted Model Count)** Let  $\varphi$  be a propositional formula and let  $w$  be a function associating a non-negative weight to each atom in  $\varphi$ , both asserted and negated. The **Weighted Model Count** of  $\varphi$  is defined as:

$$WMC(\varphi, w) = \sum_{\mu \in \mathcal{T}A(\varphi)} \prod_{\ell \in \mu} w(\ell). \quad (6)$$

WMI generalizes WMC to hybrid domains. Following is the original definition of WMI [Belle *et al.*, 2015], which serves as a starting point for our revised formulation. The definition assumes  $\mathcal{LRA}$  formulas, for which efficient solvers exist, albeit the concept could in principle accommodate other theories over continuous domains.

**Definition 3 (Weighted Model Integral)** Let  $\varphi$  be a  $\mathcal{LRA}$  formula on the set of  $\mathcal{LRA}$  variables  $\mathbf{x} \stackrel{\text{def}}{=} \{x_1, \dots, x_N\}$  and Boolean atoms  $\mathbf{A} \stackrel{\text{def}}{=} \{A_1, \dots, A_M\}$ . Let  $w$  be a function associating an expression (possibly constant) over  $\mathbf{x}$  to each atom in  $\varphi$ , both asserted and negated. The **Weighted Model Integral** of  $\varphi$  is defined as:

$$WMI(\varphi, w) = \sum_{\mu \in \mathcal{T}A(\varphi)} \int_{\mu \mathcal{LRA}} \prod_{\ell \in \mu} w(\ell) \, d\mathbf{x}. \quad (7)$$

## 3 Weighted Model Integration, Revisited

Definition 3 is a very direct and intuitive generalization of WMC to the hybrid case. Nevertheless, it suffers from some theoretical and practical limitations, which are fixed by the revised formalization we introduce in this section.

First, Definition 3 implicitly assumes WMI to be computed over the Boolean and  $\mathcal{LRA}$  variables in the formula. Explicitly defining the domain is however of primary importance in order to guarantee consistency e.g. in marginalization, as will be shown in the following (see Remarks 1 and 2).

Second, the fact that weights are associated to literals can produce situations in which two formulas are equivalent, but their WMI over the same weight function  $w$  differ.

**Example 2** Consider  $w$  s.t.  $w((x \geq 0)) = w((x \leq 3)) = 1$ ,  $w((x \geq 1)) = w(\neg(x \geq 1)) = \frac{1}{2}$ ; consider the two  $\mathcal{LRA}$ -equivalent formulas  $\varphi_1 \stackrel{\text{def}}{=} ((x \geq 0) \wedge (x \leq 3))$  and  $\varphi_2 \stackrel{\text{def}}{=} ((x \geq 0) \wedge \neg(x \geq 1)) \vee ((x \geq 1) \wedge (x \leq 3))$ . Then

$$WMI(\varphi_1, w) = \int_{(x \geq 0) \wedge (x \leq 3)} 1 \cdot 1 \, d\mathbf{x} = \int_{[0,3]} 1 \, dx = 3.$$

$$\begin{aligned} WMI(\varphi_2, w) &= \int_{(x \geq 0) \wedge \neg(x \geq 1) \wedge (x \leq 3)} 1 \cdot \frac{1}{2} \cdot 1 \, d\mathbf{x} \\ &+ \int_{(x \geq 0) \wedge (x \geq 1) \wedge (x \leq 3)} 1 \cdot \frac{1}{2} \cdot 1 \, d\mathbf{x} \\ &= \int_{[0,1)} \frac{1}{2} \, dx + \int_{[1,3]} \frac{1}{2} \, dx = \frac{1}{2} + \frac{2}{2} = \frac{3}{2}. \end{aligned}$$

Our novel formalization guarantees instead that two  $\mathcal{LRA}$ -equivalent formulas have the same WMI (Property 2.b).  $\diamond$

Finally, the fact that the summation is over all *total* truth assignments may be a major source of inefficiency, because often many  $\mathcal{LRA}$  atoms do not really contribute to the computation of the WMI. Our novel formalization allows to model these cases, opening the way to a more efficient algorithm for WMI computation (see §5).

## 3.1 Basic case: WMI Without Atomic Propositions

We investigate first the simple case where no atomic proposition comes into play. Let  $\mathbf{x} \stackrel{\text{def}}{=} \{x_1, \dots, x_N\} \in \mathbb{R}^N$ . We consider thus a generic weight function  $w(\mathbf{x})$  s.t.  $w : \mathbb{R}^N \mapsto \mathbb{R}^+$ , and  $\mathcal{LRA}$  formulas  $\varphi(\mathbf{x})$  s.t.  $\varphi : \mathbb{R}^N \mapsto \mathbb{B}$ .

**Definition 4** Assume  $\varphi$  does not contain atomic propositions and  $w : \mathbb{R}^N \mapsto \mathbb{R}^+$ . Then the **Weighted Model Integral** of  $w$  over  $\varphi$  on  $\mathbf{x}$  is defined as:

$$WMI_{\text{nb}}(\varphi, w|\mathbf{x}) \stackrel{\text{def}}{=} \int_{\varphi(\mathbf{x})} w(\mathbf{x}) \, d\mathbf{x}, \quad (8)$$

“nb” meaning “no-Booleans”, that is, as the integral of  $w(\mathbf{x})$  over the set  $\{\mathbf{x} \mid \varphi(\mathbf{x}) \text{ is true}\}$ .

The following property of  $WMI_{\text{nb}}(\varphi, w|\mathbf{x})$  derives directly from Definition 4.

**Property 1** Given  $\mathbf{x}$ ,  $w$ ,  $\varphi$ , and  $\varphi'$  as above,

- if  $\varphi$  is  $\mathcal{LRA}$ -unsatisfiable, then  $WMI_{\text{nb}}(\varphi, w|\mathbf{x}) = 0$ .
- if  $\varphi \Leftrightarrow_{\mathcal{LRA}} \varphi'$ , then  $WMI_{\text{nb}}(\varphi, w|\mathbf{x}) = WMI_{\text{nb}}(\varphi', w|\mathbf{x})$
- for every  $\mathcal{LRA}$ -formula  $\psi(\mathbf{x})$ ,

$$\begin{aligned} WMI_{\text{nb}}(\varphi, w|\mathbf{x}) &= WMI_{\text{nb}}(\varphi \wedge \psi, w|\mathbf{x}) \\ &+ WMI_{\text{nb}}(\varphi \wedge \neg\psi, w|\mathbf{x}). \end{aligned}$$

**Remark 1** We stress the fact that in the definition of  $WMI_{\text{nb}}(\varphi, w|\mathbf{x})$  specifying the domain “ $|\mathbf{x}$ ” is of primary importance. In fact, even if some  $x_n$  does not occur in  $\varphi$ ,  $WMI_{\text{nb}}(\varphi, w|\mathbf{x}) = \int_{\mathbb{R}} WMI_{\text{nb}}(\varphi, w|\mathbf{x} \setminus \{x_n\}) \, dx_n \neq WMI_{\text{nb}}(\varphi, w|\mathbf{x} \setminus \{x_n\})$ . “ $|\mathbf{x}$ ” defines the dimensions of the space we are integrating on, which must be stated. (E.g., integrating on volumes differs from integrating on surfaces.)

## 3.2 General Case: WMI With Atomic Propositions

We investigate now the general case, where atomic propositions come into play and both  $w$  and  $\varphi$  depend also on them.

Let  $\mathbf{A} \stackrel{\text{def}}{=} \{A_1, \dots, A_M\} \in \mathbb{B}^M$ . We consider thus a generic weight function  $w(\mathbf{x}, \mathbf{A})$  s.t.  $w : \mathbb{R}^N \times \mathbb{B}^M \mapsto \mathbb{R}^+$ , and  $\mathcal{LRA}$  formulas  $\varphi(\mathbf{x}, \mathbf{A})$  s.t.  $\varphi : \mathbb{R}^N \times \mathbb{B}^M \mapsto \mathbb{B}$ .

In what follows,  $\mu^{\mathbf{A}}$  denotes a total truth assignment on  $\mathbf{A}$ ,  $\varphi_{[\mu^{\mathbf{A}}]}(\mathbf{x})$  denotes (any formula equivalent to) the formula obtained from  $\varphi$  by substituting every Boolean value  $A_i$  with its truth value in  $\mu^{\mathbf{A}}$ , and  $w_{[\mu^{\mathbf{A}}]}(\mathbf{x})$  is  $w$  computed on  $\mathbf{x}$  and on the truth values of  $\mu^{\mathbf{A}}$ . Thus,  $\varphi_{[\mu^{\mathbf{A}}]} : \mathbb{R}^N \mapsto \mathbb{B}$  and  $w_{[\mu^{\mathbf{A}}]} : \mathbb{R}^N \mapsto \mathbb{R}^+$ .

**Definition 5** Given  $\mathbf{x}$ ,  $\mathbf{A}$ , the **Weighted Model Integral** of  $w$  over  $\varphi$  is defined as follows:

$$WMI(\varphi, w|\mathbf{x}, \mathbf{A}) \stackrel{\text{def}}{=} \sum_{\mu^{\mathbf{A}} \in \mathbb{B}^M} WMI_{\text{nb}}(\varphi_{[\mu^{\mathbf{A}}]}, w_{[\mu^{\mathbf{A}}]}|\mathbf{x}), \quad (9)$$

where the  $\mu^{\mathbf{A}}$ 's are all total truth assignments on  $\mathbf{A}$ .

**Example 3** Let  $\varphi \stackrel{\text{def}}{=} (A \leftrightarrow (x \geq 0)) \wedge (x \geq -1) \wedge (x \leq 1)$ , and  $w(x, A) \stackrel{\text{def}}{=} (\text{If } A \text{ Then } x \text{ Else } -x)$ . If  $\mu^{\mathbf{A}} \stackrel{\text{def}}{=} \{(\neg A)\}$ , then  $\varphi_{[\mu^{\mathbf{A}}]} = \neg(x \geq 0) \wedge (x \geq -1) \wedge (x \leq 1)$  and  $w_{[\mu^{\mathbf{A}}]} = -x$ . Notice that  $\varphi_{[\mu^{\mathbf{A}}]}$  can be simplified into the equivalent

formula  $\neg(x \geq 0) \wedge (x \geq -1)$ . Similarly, if  $\mu^{\mathbf{A}} \stackrel{\text{def}}{=} \{(A)\}$ , then  $\varphi_{[\mu^{\mathbf{A}}]}$  can be simplified into  $(x \geq 0) \wedge (x \leq 1)$  and  $w_{[\mu^{\mathbf{A}}]} = x$ . Thus,

$$\begin{aligned} \text{WMI}(\varphi, w|\mathbf{x}, \mathbf{A}) &\stackrel{\text{def}}{=} \text{WMI}_{\text{nb}}(\varphi_{[\{\neg A\}]}, w_{[\{\neg A\}]}|x) + \\ &\quad \text{WMI}_{\text{nb}}(\varphi_{[\{A\}]}, w_{[\{A\}]}|x) \\ &= \int_{[-1,0)} -x \, dx + \int_{[0,1]} x \, dx \\ &= \frac{1}{2} + \frac{1}{2} = 1. \quad \diamond \end{aligned}$$

Notice that in Definition 5 the truth assignments  $\mu^{\mathbf{A}}$  of practical interest are only those for which  $\varphi_{[\mu^{\mathbf{A}}]}$  is  $\mathcal{LRA}$ -satisfiable, because for the others  $\text{WMI}_{\text{nb}}(\varphi_{[\mu^{\mathbf{A}}]}, w_{[\mu^{\mathbf{A}}]}|\mathbf{x}) = 0$  by Property 1.a. We address this issue in §5.

The following property of  $\text{WMI}(\varphi, w|\mathbf{x}, \mathbf{A})$  derives directly from Definition 5, by applying Property 1 to  $w_{[\mu^{\mathbf{A}}]}$ ,  $\varphi_{[\mu^{\mathbf{A}}]}$ ,  $\varphi'_{[\mu^{\mathbf{A}}]}$ ,  $(\varphi \wedge \psi)_{[\mu^{\mathbf{A}}]}$ , and  $(\varphi \wedge \neg\psi)_{[\mu^{\mathbf{A}}]}$ , for every  $\mu^{\mathbf{A}}$ .

**Property 2** Given  $\mathbf{x}$ ,  $\mathbf{A}$ ,  $w$ ,  $\varphi$ , and  $\varphi'$  as above,

- if  $\varphi$  is  $\mathcal{LRA}$ -unsatisfiable, then  $\text{WMI}(\varphi, w|\mathbf{x}, \mathbf{A}) = 0$ .
- if  $\varphi \Leftrightarrow_{\mathcal{LRA}} \varphi'$ , then  $\text{WMI}(\varphi, w|\mathbf{x}, \mathbf{A}) = \text{WMI}(\varphi', w|\mathbf{x}, \mathbf{A})$
- for every  $\mathcal{LRA}$ -formula  $\psi(\mathbf{x}, \mathbf{A})$ ,

$$\begin{aligned} \text{WMI}(\varphi, w|\mathbf{x}, \mathbf{A}) &= \text{WMI}(\varphi \wedge \psi, w|\mathbf{x}, \mathbf{A}) \\ &\quad + \text{WMI}(\varphi \wedge \neg\psi, w|\mathbf{x}, \mathbf{A}). \end{aligned}$$

**Remark 2** As with Remark 1, in  $\text{WMI}(\varphi, w|\mathbf{x}, \mathbf{A})$ , specifying not only “ $\mathbf{x}$ ”, but also “ $|\mathbf{x}, \mathbf{A}$ ” is of primary importance. In fact, even if some of the  $A_m$  does not occur in  $\varphi$ ,  $\text{WMI}(\varphi, w|\mathbf{x}, \mathbf{A}) = \text{WMI}(\varphi, w_{[\{A_m\}]}, \mathbf{x}, \mathbf{A} \setminus \{A_m\}) + \text{WMI}(\varphi, w_{[\{\neg A_m\}]}, \mathbf{x}, \mathbf{A} \setminus \{\neg A_m\}) \neq \text{WMI}(\varphi, w|\mathbf{x}, \mathbf{A} \setminus \{A_m\})$ .

Predicate abstraction in Def. 1 and other forms of frequently-used formula manipulations require the introduction of fresh propositions  $B$  “labelling” sub-formulas  $\psi$ . The next result shows that this does not affect the value of WMI.

**Theorem 1** Let  $\mathbf{x}$ ,  $\mathbf{A}$ ,  $w$ , and  $\varphi$  be as in Definition 5; let  $\psi(\mathbf{x}, \mathbf{A})$  be some  $\mathcal{LRA}$ -formula; let  $\varphi' \stackrel{\text{def}}{=} \varphi \wedge (B \leftrightarrow \psi)$ , where  $B \notin \mathbf{A}$ ; let  $w'$  extend  $w$  s.t.  $w'(\mathbf{x}, \mathbf{A} \cup \{B\}) = w(\mathbf{x}, \mathbf{A})$  for every  $\mathbf{x}$ ,  $\mathbf{A}$  and  $B$ . Then we have that

$$\text{WMI}(\varphi', w'|\mathbf{x}, \mathbf{A} \cup \{B\}) = \text{WMI}(\varphi, w|\mathbf{x}, \mathbf{A}). \quad (10)$$

**Proof** (sketch) We notice that  $\varphi'_{[\mu^{\mathbf{A} \cup \{B\}}]} \Leftrightarrow_{\mathcal{LRA}} (\varphi \wedge \psi)_{[\mu^{\mathbf{A}}]}$  and that  $\varphi'_{[\mu^{\mathbf{A} \cup \{B\}}]} \Leftrightarrow_{\mathcal{LRA}} (\varphi \wedge \neg\psi)_{[\mu^{\mathbf{A}}]}$ , so that (10) follows directly from Definition 5 and Property 1.c.  $\square$

### 3.3 Conditional Weight Functions

We call a (non-minimal) *support* of a weight function  $w(\mathbf{x}, \mathbf{A})$  any subset of  $\mathbb{R}^N \times \mathbb{B}^M$  out of which  $w(\mathbf{x}, \mathbf{A}) = 0$ . In many situations we may want to provide the representation of a support of  $w(\mathbf{x}, \mathbf{A})$  as a  $\mathcal{LRA}$ -formula  $\chi(\mathbf{x}, \mathbf{A})$ . When this is the case, the following property follows trivially.

**Property 3** Let  $\varphi$  and  $w$  be as above. If  $\chi(\mathbf{x}, \mathbf{A})$  is a  $\mathcal{LRA}$ -formula representing a support of  $w$ , then:

$$\text{WMI}(\varphi, w|\mathbf{x}, \mathbf{A}) = \text{WMI}(\varphi \wedge \chi, w|\mathbf{x}, \mathbf{A}). \quad (11)$$

We introduce a novel kind of weight function, which can be defined also in terms of  $\mathcal{LRA}$  conditions. (See §4 for an example application). We consider first the generic class of functions  $P(\mathbf{x})$ , which we call *feasibly integrable* on  $\mathcal{LRA}(FI^{\mathcal{LRA}})$ , which contain no combinatorial component, and for which there exists some procedure able to compute  $\text{WMI}_{\text{nb}}(\mu^{\mathcal{LRA}}, P|\mathbf{x})$  for every set of  $\mathcal{LRA}$  atoms on  $\mathbf{x}$ . (E.g., polynomials are  $FI^{\mathcal{LRA}}$  [Baldoni *et al.*, 2011]). Such background procedure, which we use as a blackbox, is the basic building block of our WMI calculations.

**Definition 6** We call a weight function  $w(\mathbf{x}, \mathbf{A})$ , **feasibly integrable under  $\mathcal{LRA}$  conditions** ( $FIUC^{\mathcal{LRA}}$ ) iff it can be described in terms of

- a support  $\mathcal{LRA}$ -formula  $\chi(\mathbf{x}, \mathbf{A})$  (if no support description is provided, then  $\chi \stackrel{\text{def}}{=} \top$ ),
- a set  $\Psi \stackrel{\text{def}}{=} \{\psi_1(\mathbf{x}, \mathbf{A}), \dots, \psi_K(\mathbf{x}, \mathbf{A})\}$  of  $\mathcal{LRA}$ -formulas (**conditions**),

in such a way that, for every total truth assignment  $\mu^{\mathbf{A}}$  to  $\mathbf{A}$  and for every total truth assignment  $\mu^{\Psi}$  to  $\Psi$ ,  $w_{[\mu^{\mathbf{A}} \mu^{\Psi}]}(\mathbf{x})$  is  $FI^{\mathcal{LRA}}$  in the domain given by the values of  $\langle \mathbf{x}, \mathbf{A} \rangle$  which satisfy  $(\chi \wedge \mu^{\Psi})_{[\mu^{\mathbf{A}}]}$ . We denote such  $FI^{\mathcal{LRA}}$  functions by  $f_{\mu^{\mathbf{A}} \mu^{\Psi}}(\mathbf{x})$ , s.t. for every  $\langle \mu^{\mathbf{A}}, \mu^{\Psi} \rangle$ ,

$$\text{if } \mu^{\mathbf{A}} \wedge \mu^{\Psi} \text{ holds, then } w(\mathbf{x}) = f_{\mu^{\mathbf{A}} \mu^{\Psi}}(\mathbf{x}). \quad (12)$$

(Notice that a plain  $FI^{\mathcal{LRA}}$  weight function is a subcase in which  $\chi \stackrel{\text{def}}{=} \top$  and  $\Psi \stackrel{\text{def}}{=} \emptyset$ .)

**Example 4** Let  $\mathbf{x} \stackrel{\text{def}}{=} \{x_1, x_2\}$ ,  $\mathbf{A} \stackrel{\text{def}}{=} \{A\}$ , and

$$\chi(\mathbf{x}, \mathbf{A}) \stackrel{\text{def}}{=} \llbracket x_1 \in [-1, 1] \rrbracket \wedge \llbracket x_2 \in [-1, 1] \rrbracket \quad (13)$$

$$\wedge (A \leftrightarrow (x_2 \geq 0)) \quad (14)$$

$$w(\mathbf{x}, \mathbf{A}) \stackrel{\text{def}}{=} (\text{If } x_1 \geq 0 \text{ Then } x_1^3 \text{ Else } -2x_1) \quad (15)$$

$$+ (\text{If } A \text{ Then } 3x_2 \text{ Else } -x_2^5). \quad (16)$$

$w$  is  $FIUC^{\mathcal{LRA}}$ . In fact, its value depends on the combination of the truth values of the conditions  $\Psi \stackrel{\text{def}}{=} \{(x_1 \geq 0)\}$  and  $\mathbf{A} \stackrel{\text{def}}{=} \{A\}$  -i.e., of  $\{(x_1 \geq 0), (x_2 \geq 0)\}$  by (14)- so that:  $f_{\{A, (x_1 \geq 0)\}} = x_1^3 + 3x_2$  s.t.  $x_1 \in [0, 1), x_2 \in [0, 1)$ ,  $f_{\{A, \neg(x_1 \geq 0)\}} = -2x_1 + 3x_2$ , s.t.  $x_1 \in [-1, 0), x_2 \in [0, 1)$ ,  $f_{\{\neg A, (x_1 \geq 0)\}} = x_1^3 - x_2^5$ , s.t.  $x_1 \in [0, 1), x_2 \in [-1, 0)$ ,  $f_{\{\neg A, \neg(x_1 \geq 0)\}} = -2x_1 - x_2^5$ , s.t.  $x_1 \in [-1, 0), x_2 \in [-1, 0)$ . All four  $f_{\mu^{\mathbf{A}} \mu^{\Psi}}$  are positive polynomials in their respective domain and as such they can be integrated.  $\diamond$

Intuitively, Definition 6 captures the class of all the weight functions which can be described by means of arbitrary combinations of nested if-then-elses on conditions in  $\mathbf{A}$  and  $\Psi$ , s.t. each branch  $\langle \mu^{\mathbf{A}}, \mu^{\Psi} \rangle$  results into a  $FI^{\mathcal{LRA}}$  weight function. Each pair  $\langle \mu^{\mathbf{A}}, \mu^{\Psi} \rangle$  describes a portion of the domain of  $w$ , inside which  $w$  is the  $FI^{\mathcal{LRA}}$  function  $f_{\mu^{\mathbf{A}} \mu^{\Psi}}$ .

**Theorem 2** Let  $w(\mathbf{x}, \mathbf{A})$ ,  $\Psi$  and  $\chi$  be as in Definition 6.

Let  $\mathbf{B} \stackrel{\text{def}}{=} \{B_1, \dots, B_K\}$  be fresh propositional atoms and let  $w^*(\mathbf{x}, \mathbf{A} \cup \mathbf{B})$  be the weight function obtained by substituting

in  $w(\mathbf{x}, \mathbf{A})$  each condition  $\psi_k$  with  $B_k$ , for every  $k \in [1..K]$ . Let  $\varphi^* \stackrel{\text{def}}{=} \varphi \wedge \chi \wedge \bigwedge_{k=1}^K (B_k \leftrightarrow \psi_k)$ . Then:

$$\text{WMI}(\varphi \wedge \chi, w|\mathbf{x}, \mathbf{A}) = \text{WMI}(\varphi^*, w^*|\mathbf{x}, \mathbf{A} \cup \mathbf{B}) \quad (17)$$

**Proof** To every truth assignment  $\mu^\Psi$  to  $\Psi$  we associate the corresponding truth assignment  $\mu^B$  to  $\mathbf{B}$  s.t.  $\mu^B(B_k) = \mu^\Psi(\psi_k)$ , for every  $k \in [1..K]$ . We notice that, for every  $\mu^A \in \mathbb{B}^M$  and  $\mu^B \in \mathbb{B}^K$  (with its corresponding  $\mu^\Psi$ ):

$$\varphi^*_{[\mu^A \wedge \mu^B]} \Leftrightarrow_{\mathcal{LRA}} (\varphi \wedge \chi \wedge \mu^\Psi)_{[\mu^A \wedge \mu^B]}, \quad (18)$$

because every  $\psi_k$  is forced by  $\mu^\Psi$  to assume the same truth value  $B_k$  assumes in  $\mu^B$ . Let  $w'$  extend  $w$  s.t.  $w'(\mathbf{x}, \mathbf{A} \cup \mathbf{B}) = w(\mathbf{x}, \mathbf{A})$  for every  $\mathbf{x}, \mathbf{A}$  and  $\mathbf{B}$ . Then, since  $\varphi^*$  forces every  $B_k$  to hold if and only if  $\psi_k$  holds, we have:

$$\begin{aligned} & \text{WMI}_{\text{nb}}(\varphi^*_{[\mu^A \wedge \mu^B]}, w'_{[\mu^A \wedge \mu^B]}|\mathbf{x}) \\ &= \text{WMI}_{\text{nb}}(\varphi^*_{[\mu^A \wedge \mu^B]}, w^*_{[\mu^A \wedge \mu^B]}|\mathbf{x}). \end{aligned} \quad (19)$$

Then, by applying  $K$  times Theorem 1, and then (19):

$$\begin{aligned} & \text{WMI}(\varphi \wedge \chi, w|\mathbf{x}, \mathbf{A}) \\ &= \text{WMI}(\varphi \wedge \chi \wedge \bigwedge_{k=1}^K (B_k \leftrightarrow \psi_k), w'|\mathbf{x}, \mathbf{A} \cup \mathbf{B}) \\ &= \sum_{\mu^A \in \mathbb{B}^M, \mu^B \in \mathbb{B}^K} \text{WMI}_{\text{nb}}(\varphi^*_{[\mu^A \wedge \mu^B]}, w'_{[\mu^A \wedge \mu^B]}|\mathbf{x}) \\ &= \sum_{\mu^A \in \mathbb{B}^M, \mu^B \in \mathbb{B}^K} \text{WMI}_{\text{nb}}(\varphi^*_{[\mu^A \wedge \mu^B]}, w^*_{[\mu^A \wedge \mu^B]}|\mathbf{x}) \\ &= \text{WMI}(\varphi^*, w^*|\mathbf{x}, \mathbf{A} \cup \mathbf{B}). \end{aligned}$$

**Example 5** Let  $\mathbf{A} = \emptyset$ ,  $\chi \stackrel{\text{def}}{=} (x \geq -1) \wedge (x \leq 1)$ ,  $\varphi \stackrel{\text{def}}{=} \top$ ,  $\psi \stackrel{\text{def}}{=} (x \geq 0)$ , and the weight  $w(x) \stackrel{\text{def}}{=} (\text{If } (x \geq 0) \text{ Then } x \text{ Else } -x)$ . (I.e.,  $w(x) \stackrel{\text{def}}{=} |x|$ ). Then  $\text{WMI}(\varphi, w|x, \emptyset) = \text{WMI}_{\text{nb}}(\varphi, w|\mathbf{x}) = \int_{[-1,1]} |x| dx = 1$ . By Theorem 1,  $\varphi^* = (x \geq -1) \wedge (x \leq 1) \wedge (B \leftrightarrow (x \geq 0))$  and  $w^* = (\text{If } B \text{ Then } x \text{ Else } -x)$ , which are the same formula and weight function as in Example 3 (modulo some reordering and variable renaming), s.t.  $\text{WMI}(\varphi^*, w^*|\mathbf{x}, \mathbf{B}) = 1$ .  $\diamond$

Intuitively, during the computation of the WMIs, Theorem 2 allows for extracting out of the integrals the if-then-else component on  $\mathcal{LRA}$  conditions, which are labeled by Boolean atoms and can be thus handled externally.

Notice that the pairs of truth assignments  $(\mu^A, \mu^\Psi)$  of practical interest are only those for which  $(\chi \wedge \mu^\Psi)_{[\mu^A]}$  is  $\mathcal{LRA}$ -satisfiable. We will address this issue in §5.

## 4 A Case Study

Consider modelling journey time on a road network for e.g. a delivery agency. In order to safely organize priority deliveries, the agency could be interested in knowing well in advance the probability of completing the journey within a certain time, given the time of departure. An accurate estimate requires to consider how travel duration between locations can change according to the time of the day, and combine

these duration distributions over the entire route. A different encoding for the same problem was presented in the original WMI work [Belle *et al.*, 2015].

Suppose that the day is partitioned into  $\{I^1, \dots, I^M\}$  intervals such that, for each adjacent location  $l_i$  and  $l_j$  in the road network and for each  $I^m \stackrel{\text{def}}{=} [c_m, c_{m+1})$ , we know the distribution of the journey time from  $l_i$  to  $l_j$  given that we move at time  $t \in I^m$ . Let  $f_{l_i, l_j}^m : \mathbb{R} \mapsto \mathbb{R}^+$  denote such distribution and let the interval  $R_{l_i, l_j}^m \stackrel{\text{def}}{=} [a_{l_i, l_j}^m, b_{l_i, l_j}^m)$  be its support. (Notice that the  $I^m$ s are all disjoint whereas the  $R_{l_i, l_j}^m$ s are typically not.)

Given a path  $(l_0, \dots, l_N)$  and the departure time  $t_{\text{dep}}$ , we can encode the problem as follows. Let  $t_n$  be the time at step  $n$  and  $x_n$  the journey time between  $l_{n-1}$  and  $l_n$ . Let  $\mathbf{x} \stackrel{\text{def}}{=} \{x_1, \dots, x_N\}$ . (Here  $\mathbf{A} \stackrel{\text{def}}{=} \emptyset$ .)

$$\begin{aligned} \varphi(\mathbf{x}) &\stackrel{\text{def}}{=} \top \\ \chi(\mathbf{x}) &\stackrel{\text{def}}{=} \bigwedge_{n=1}^N \text{OneOf}(\{\llbracket t_{n-1} \in I^m \rrbracket\}_{m=1}^M) \\ &\quad \wedge \bigwedge_{n=1}^N \bigwedge_{m=1}^M (\llbracket t_{n-1} \in I^m \rrbracket \rightarrow \llbracket x_n \in R_{l_{n-1}, l_n}^m \rrbracket) \\ w(\mathbf{x}) &\stackrel{\text{def}}{=} \prod_{n=1}^N f_n(\mathbf{x}) \end{aligned}$$

where “ $\llbracket t_{n-1} \in I^m \rrbracket$ ” is a shortcut for the  $\mathcal{LRA}$  formula “ $(\sum_{i=1}^{n-1} x_i + t_{\text{dep}} \geq c_m) \wedge (\sum_{i=1}^{n-1} x_i + t_{\text{dep}} < c_{m+1})$ ”, “ $\llbracket x_n \in R_{l_{n-1}, l_n}^m \rrbracket$ ” is a shortcut for the  $\mathcal{LRA}$  formula “ $(x_n \geq a_{l_{n-1}, l_n}^m) \wedge (x_n < b_{l_{n-1}, l_n}^m)$ ”, and  $f_n(\mathbf{x})$  is: (Case  $\llbracket t_{n-1} \in I^1 \rrbracket$  :  $f_{l_{n-1}, l_n}^1(x_n); \dots$   $\llbracket t_{n-1} \in I^M \rrbracket$  :  $f_{l_{n-1}, l_n}^M(x_n)$ ). Under the assumption that each distribution  $f_{l_i, l_j}^m(x)$  is feasibly integrable if  $x \in R_{l_i, l_j}^m$ , then  $w(\mathbf{x})$  is  $FIUCLRA$  with  $N \cdot M$  conditions  $\psi_n^m \stackrel{\text{def}}{=} \llbracket t_n \in I^m \rrbracket$ . Thus we can introduce  $N \cdot M$  fresh Boolean atoms  $B_n^m$  and apply Theorem 2, s.t.:

$$\begin{aligned} \varphi^*(\mathbf{x}, \mathbf{B}) &\stackrel{\text{def}}{=} \varphi(\mathbf{x}) \wedge \chi(\mathbf{x}) \wedge \bigwedge_{n=1}^N \bigwedge_{m=1}^M (B_{n-1}^m \leftrightarrow \llbracket t_{n-1} \in I^m \rrbracket) \\ w^*(\mathbf{x}, \mathbf{B}) &\stackrel{\text{def}}{=} \prod_{n=1}^N f_n^*(x_n, \{B_{n-1}^1, \dots, B_{n-1}^M\}), \end{aligned}$$

where  $f_n^*(x_n, \{B_{n-1}^1, \dots, B_{n-1}^M\})$  is the case-split formula (Case  $B_{n-1}^1$  :  $f_{l_{n-1}, l_n}^1(x_n); \dots$   $B_{n-1}^M$  :  $f_{l_{n-1}, l_n}^M(x_n)$ ).

Each distribution  $f_{l_{n-1}, l_n}^m$  is thus associated to  $B_n^m$ . Notice that, for each step  $n$ , exactly one condition variable  $B_n^m$  is true, representing the fact that the  $n$ -th location is reached during the  $m$ -th interval. Intuitively, this allows to select at each step the distribution corresponding to the interval in which the location is reached.

**Example 6** Consider an instance of our case study where  $\mathbf{A} \stackrel{\text{def}}{=} \emptyset$ ,  $N = 2$ ,  $M = 3$ ,  $\varphi \stackrel{\text{def}}{=} \top$  and

$$\begin{aligned} \chi(\mathbf{x}) &\stackrel{\text{def}}{=} \text{OneOf}(\{\llbracket t_{\text{dep}} \in [7, 8] \rrbracket, \dots, \llbracket t_{\text{dep}} \in [9, 10] \rrbracket\}) \\ &\quad \wedge \text{OneOf}(\{\llbracket t_{\text{dep}} + x_1 \in [7, 8] \rrbracket, \dots, \llbracket t_{\text{dep}} + x_1 \in [9, 10] \rrbracket\}) \\ &\quad \wedge \llbracket t_{\text{dep}} \in [7, 8] \rrbracket \rightarrow \llbracket x_1 \in [0.5, 1] \rrbracket \\ &\quad \wedge \llbracket t_{\text{dep}} \in [8, 9] \rrbracket \rightarrow \llbracket x_1 \in [1, 1.5] \rrbracket \\ &\quad \wedge \llbracket t_{\text{dep}} \in [9, 10] \rrbracket \rightarrow \llbracket x_1 \in [1, 2] \rrbracket \end{aligned}$$

$$\begin{aligned}
 & \wedge \llbracket t_{\text{dep}} + x_1 \in [7, 8] \rrbracket \rightarrow \llbracket x_2 \in [1, 1.5] \rrbracket \\
 & \wedge \llbracket t_{\text{dep}} + x_1 \in [8, 9] \rrbracket \rightarrow \llbracket x_2 \in [1.5, 2] \rrbracket \\
 & \wedge \llbracket t_{\text{dep}} + x_1 \in [9, 10] \rrbracket \rightarrow \llbracket x_2 \in [1, 2] \rrbracket \\
 \\
 w(\mathbf{x}) \stackrel{\text{def}}{=} & \left( \begin{array}{l} \llbracket t_{\text{dep}} \in [7, 8] \rrbracket : f_{l_0 l_1}^1(x_1); \\ \text{Case } \llbracket t_{\text{dep}} \in [8, 9] \rrbracket : f_{l_0 l_1}^2(x_1); \\ \llbracket t_{\text{dep}} \in [9, 10] \rrbracket : f_{l_0 l_1}^3(x_1); \end{array} \right) \times \\
 & \left( \begin{array}{l} \llbracket t_{\text{dep}} + x_1 \in [7, 8] \rrbracket : f_{l_1 l_2}^1(x_2); \\ \text{Case } \llbracket t_{\text{dep}} + x_1 \in [8, 9] \rrbracket : f_{l_1 l_2}^2(x_2); \\ \llbracket t_{\text{dep}} + x_1 \in [9, 10] \rrbracket : f_{l_1 l_2}^3(x_2); \end{array} \right)
 \end{aligned}$$

where the  $f_{l_{n-1} l_n}^m(x_n)$  are function which are integrable and positive in their respective domain stated in  $\chi(\mathbf{x})$  (e.g.,  $f_{l_0 l_1}^1(x_1)$  is integrable and positive in  $\llbracket x_1 \in [0.5, 1] \rrbracket$ ).

Then, by applying Theorem 2, we can introduce 6 Boolean variables  $B_n^m$  and reformulate the problem as follows:

$$\begin{aligned}
 \varphi^*(\mathbf{x}, \mathbf{B}) & \stackrel{\text{def}}{=} \varphi(\mathbf{x}) \wedge \chi(\mathbf{x}) \quad (20) \\
 & \wedge (B_0^1 \leftrightarrow \llbracket t_{\text{dep}} \in [7, 8] \rrbracket) \wedge \dots \wedge \\
 & \wedge (B_1^3 \leftrightarrow \llbracket t_{\text{dep}} + x_1 \in [9, 10] \rrbracket) \\
 \\
 w^*(\mathbf{x}, \mathbf{B}) & \stackrel{\text{def}}{=} \left( \begin{array}{l} B_0^1 : f_{l_0 l_1}^1(x_1); \\ \text{Case } B_0^2 : f_{l_0 l_1}^2(x_1); \\ B_0^3 : f_{l_0 l_1}^3(x_1); \end{array} \right) \times \quad (21) \\
 & \left( \begin{array}{l} B_1^1 : f_{l_1 l_2}^1(x_2); \\ \text{Case } B_1^2 : f_{l_1 l_2}^2(x_2); \\ B_1^3 : f_{l_1 l_2}^3(x_2); \end{array} \right)
 \end{aligned}$$

## 5 Efficient WMI Computation

We consider the general problem of computing a WMI with a  $FIUC^{\mathcal{LRA}}$  weight function. The first step (if needed) is a preprocessing in which the problem is transformed by labeling all conditions  $\Psi$  with fresh Boolean atoms  $\mathbf{B}$ , as in Theorem 2. Let  $\varphi^*, w^*, \mathbf{x}, \mathbf{A}^*$  be the result of such process, where  $\mathbf{A}^* \stackrel{\text{def}}{=} \mathbf{A} \cup \mathbf{B}$ . Consequently, for every  $\mu^{\mathbf{A}^*}$ ,  $w_{[\mu^{\mathbf{A}^*}]}^*$  is feasibly integrable on  $\varphi_{[\mu^{\mathbf{A}^*}]}^*$ .

**WMI-AllSMT.** Consider  $\mu = \mu^{\mathbf{A}^*} \wedge \mu^{\mathcal{LRA}} \in \mathcal{T}\mathcal{A}(\varphi^*)$ . Then  $\mu^{\mathcal{LRA}} \in \mathcal{T}\mathcal{A}(\varphi_{[\mu^{\mathbf{A}^*}]}^*)$ , so that we can compute  $\text{WMI}_{\text{nb}}(\mu^{\mathcal{LRA}}, w_{[\mu^{\mathbf{A}^*}]}^* | \mathbf{x})$ . This allows us to compute (9) as:

$$\begin{aligned}
 \text{WMI}(\varphi, w | \mathbf{x}, \mathbf{A}) &= \text{WMI}(\varphi^*, w^* | \mathbf{x}, \mathbf{A}^*) \\
 &= \sum_{\mu^{\mathbf{A}^*} \wedge \mu^{\mathcal{LRA}} \in \mathcal{T}\mathcal{A}(\varphi^*)} \text{WMI}_{\text{nb}}(\mu^{\mathcal{LRA}}, w_{[\mu^{\mathbf{A}^*}]}^* | \mathbf{x}) \quad (22)
 \end{aligned}$$

The set  $\mathcal{T}\mathcal{A}(\varphi^*)$  is computed by an AllSMT procedure implemented on top of an SMT solver like MATHSAT5—i.e., as  $\mathcal{T}\mathcal{A}(\text{PredAbs}_{[\varphi^*]}(\text{Atoms}(\varphi^*)))$ , without the assignment-minimization technique (see §2.1); each  $\text{WMI}_{\text{nb}}(\mu^{\mathcal{LRA}}, w_{[\mu^{\mathbf{A}^*}]}^* | \mathbf{x})$  is computed by invoking our background integration procedure for  $FI^{\mathcal{LRA}}$  functions of §3.3. We call this algorithm WMI-AllSMT.

---

### Algorithm 1 WMI-PA( $\varphi, w, \mathbf{x}, \mathbf{A}$ )

---

```

 $\langle \varphi^*, w^*, \mathbf{A}^* \rangle \leftarrow \text{LabelConditions}(\varphi, w, \mathbf{x}, \mathbf{A})$ 
 $\mathcal{M}^{\mathbf{A}^*} \leftarrow \mathcal{T}\mathcal{A}(\text{PredAbs}_{[\varphi^*]}(\mathbf{A}^*))$ 
 $vol \leftarrow 0$ 
for  $\mu^{\mathbf{A}^*} \in \mathcal{M}^{\mathbf{A}^*}$  do
  Simplify( $\varphi_{[\mu^{\mathbf{A}^*}]}^*$ )
  if LiteralConjunction( $\varphi_{[\mu^{\mathbf{A}^*}]}^*$ ) then
     $vol \leftarrow vol + \text{WMI}_{\text{nb}}(\varphi_{[\mu^{\mathbf{A}^*}]}^*, w_{[\mu^{\mathbf{A}^*}]}^* | \mathbf{x})$ 
  else
     $\mathcal{M}^{\mathcal{LRA}} \leftarrow \mathcal{T}\mathcal{A}(\text{PredAbs}_{[\varphi_{[\mu^{\mathbf{A}^*}]}^*]}(\text{Atoms}(\varphi_{[\mu^{\mathbf{A}^*}]}^*)))$ 
    for  $\mu^{\mathcal{LRA}} \in \mathcal{M}^{\mathcal{LRA}}$  do
       $vol \leftarrow vol + \text{WMI}_{\text{nb}}(\mu^{\mathcal{LRA}}, w_{[\mu^{\mathbf{A}^*}]}^* | \mathbf{x})$ 
    end for
  end if
end for
return  $vol$ 

```

---

**WMI-PA.** A much more efficient technique, which we call WMI-PA because it exploits the full power of SMT-based predicate abstraction, can be implemented by noticing that

$$\text{WMI}(\varphi, w | \mathbf{x}, \mathbf{A}) = \sum_{\mu^{\mathbf{A}^*} \in \mathcal{T}\mathcal{A}(\exists \mathbf{x}. \varphi^*)} \text{WMI}_{\text{nb}}(\varphi_{[\mu^{\mathbf{A}^*}]}^*, w_{[\mu^{\mathbf{A}^*}]}^* | \mathbf{x}) \quad (23)$$

and that each  $\text{WMI}_{\text{nb}}(\varphi_{[\mu^{\mathbf{A}^*}]}^*, w_{[\mu^{\mathbf{A}^*}]}^* | \mathbf{x})$  can be computed as:

$$\sum_{\mu^{\mathcal{LRA}} \in \mathcal{T}\mathcal{A}(\varphi_{[\mu^{\mathbf{A}^*}]}^*)} \text{WMI}_{\text{nb}}(\mu^{\mathcal{LRA}}, w_{[\mu^{\mathbf{A}^*}]}^* | \mathbf{x}). \quad (24)$$

The pseudocode of WMI-PA is reported in Algorithm 1. First, the problem is transformed (if needed) by labeling conditions  $\Psi$  with fresh Boolean variables  $\mathbf{B}$ . After this preprocessing stage, the set  $\mathcal{M}^{\mathbf{A}^*} \stackrel{\text{def}}{=} \mathcal{T}\mathcal{A}(\exists \mathbf{x}. \varphi^*)$  is computed by invoking  $\mathcal{T}\mathcal{A}(\text{PredAbs}_{[\varphi^*]}(\mathbf{A}^*))$  (see §2.1). Then, the algorithm iterates over each Boolean assignment  $\mu^{\mathbf{A}^*}$  in  $\mathcal{M}^{\mathbf{A}^*}$ .  $\varphi_{[\mu^{\mathbf{A}^*}]}^*$  is simplified by the Simplify procedure, by propagating truth values (e.g.,  $\varphi_1 \wedge (\top \vee \varphi_2) \wedge (\perp \vee \varphi_3) \wedge (\neg \varphi_3 \vee \varphi_4) \Rightarrow \varphi_1 \wedge \varphi_3 \wedge \varphi_4$ ) and by applying arithmetical simplifications like theory propagation [Barrett *et al.*, 2009]. This improves the chances of reducing  $\varphi_{[\mu^{\mathbf{A}^*}]}^*$  to a conjunction of literals, and allows for reducing the size of  $\text{Atoms}(\varphi_{[\mu^{\mathbf{A}^*}]}^*)$  to feed to PredAbs (see below). Then, if  $\varphi_{[\mu^{\mathbf{A}^*}]}^*$  is already a conjunction of literals, then the algorithm directly computes its contribution to the volume by calling  $\text{WMI}_{\text{nb}}(\varphi_{[\mu^{\mathbf{A}^*}]}^*, w_{[\mu^{\mathbf{A}^*}]}^* | \mathbf{x})$ . Otherwise,  $\mathcal{T}\mathcal{A}(\varphi_{[\mu^{\mathbf{A}^*}]}^*)$  is computed as  $\mathcal{T}\mathcal{A}(\text{PredAbs}_{[\varphi_{[\mu^{\mathbf{A}^*}]}^*]}(\text{Atoms}(\varphi_{[\mu^{\mathbf{A}^*}]}^*)))$ , using the minimization technique to produce partial assignments (see §2.1), and the algorithm iteratively computes contributions to the volume for each  $\mu^{\mathcal{LRA}}$ .

Comparing (23) with (9)—even if  $\varphi^*, w^*, \mathbf{x}, \mathbf{A}^*$  were respectively  $\varphi, w, \mathbf{x}, \mathbf{A}$ —we notice that the restriction of the sum to  $\mathcal{T}\mathcal{A}(\exists \mathbf{x}. \varphi^*)$  in (23) removes *a priori* all the assignments  $\mu^{\mathbf{A}^*}$  which cannot be expanded by any assignment  $\mu^{\mathcal{LRA}}$  s.t.  $\mu^{\mathbf{A}^*} \wedge \mu^{\mathcal{LRA}}$  propositionally satisfy  $\varphi^*$  and  $\mu^{\mathcal{LRA}}$  is  $\mathcal{LRA}$ -consistent, whose integrals would be 0-valued.

We argue that WMI-PA produces much less calls to the background integration procedure  $\text{WMI}_{\text{nb}}(\mu^{\mathcal{LR}\mathcal{A}}, w_{[\mu^{\mathbf{A}^*}]}^* | \mathbf{x})$  than WMI-ALLSMT, for two main reasons.

First, the size of  $\text{Atoms}(\varphi_{[\mu^{\mathbf{A}^*}]}^*)$  which is fed to  $\text{PredAbs}$  in (24) can be made much smaller than the number of  $\mathcal{LR}\mathcal{A}$ -atoms in  $\text{Atoms}(\varphi^*)$  fed to  $\text{PredAbs}$  in (22), since many  $\mathcal{LR}\mathcal{A}$ -atoms are simplified out by  $\mu^{\mathbf{A}^*}$ . (E.g.,  $(x \leq 1) \wedge (A_2 \vee (x \geq 0))_{[A_2]}$  is simplified into  $(x \leq 1)$ , so that  $(x \geq 0)$  is eliminated.) Thus, for each  $\mu^{\mathbf{A}^*}$ , the number of assignments in the form  $\mu^{\mathbf{A}^*} \wedge \mu^{\mathcal{LR}\mathcal{A}}$  which are enumerated in (23)-(24) can be drastically reduced wrt. those enumerated in (22).

Second, with (24) it is possible to search for a set  $\mathcal{TA}(\dots)$  of *partial* assignments, each of which substitutes  $2^i$  total ones,  $i$  being the number of unassigned  $\mathcal{LR}\mathcal{A}$ -atoms. Notice that, unlike with Boolean atoms, we can safely produce partial assignments on  $\mathcal{LR}\mathcal{A}$ -atoms because  $w(\mathbf{x}, \mathbf{A})$  does not depend directly on them. (If  $\varphi_{[\mu^{\mathbf{A}^*}]}^* \stackrel{\text{def}}{=} (x \geq 0) \wedge ((x \leq 2) \vee (x \leq 1))$ , the partial assignment  $\mu^{\mathcal{LR}\mathcal{A}} \stackrel{\text{def}}{=} (x \geq 0) \wedge (x \leq 2)$  prevents enumerating the two total ones  $\mu^{\mathcal{LR}\mathcal{A}} \wedge (x \leq 1)$  and  $\mu^{\mathcal{LR}\mathcal{A}} \wedge \neg(x \leq 1)$ , computing one integral rather than two.

**Example 7** Consider the problem described by  $\varphi^*$  and  $w^*$  in Example 6. Suppose we want to leave  $l_0$  no earlier than 7 and no later than 10, and arrive to  $l_2$  strictly before 11. These constraints correspond to conjoining  $\llbracket t_{\text{dep}} \in [7, 10] \rrbracket \wedge (t_{\text{arr}} = t_{\text{dep}} + x_1 + x_2) \wedge (t_{\text{arr}} < 11)$  to  $\varphi^*$ . In such case,  $\text{PredAbs}_{[\varphi^*]}(\mathbf{B})$  is the following formula:

$$(B_0^1 \wedge \neg B_0^2 \wedge \neg B_0^3 \wedge B_1^1 \wedge \neg B_1^2 \wedge \neg B_1^3) \quad (25)$$

$$\vee (B_0^1 \wedge \neg B_0^2 \wedge \neg B_0^3 \wedge \neg B_1^1 \wedge B_1^2 \wedge \neg B_1^3) \quad (26)$$

$$\vee (\neg B_0^1 \wedge B_0^2 \wedge \neg B_0^3 \wedge \neg B_1^1 \wedge \neg B_1^2 \wedge B_1^3) \quad (27)$$

so that  $\mathcal{TTA}(\text{PredAbs}_{[\varphi^*]}(\mathbf{B}))$  is the set of the three disjuncts (25)-(27). Importantly, notice that the other 6 assignments, which would make  $\varphi^*$   $\mathcal{LR}\mathcal{A}$ -unsatisfiable causing  $\text{WMI}_{\text{nb}}$  to return 0, *are not generated* by  $\text{PredAbs}_{[\varphi^*]}(\mathbf{B})$ . (E.g., if  $B_0^1 = \top$  then  $l_1$  is necessarily reached strictly before 9, which forces  $B_1^3 = \perp$ , s.t. the assignment  $(B_0^1 \wedge \dots \wedge B_1^3)$  is not generated.)

Suppose we fix  $t_{\text{dep}}$  to some value  $val \in [7, 10]$  by conjoining  $(t_{\text{dep}} = val)$  to  $\varphi^*$  (see §7): if  $val \in [7, 7.5)$ , this forces  $B_0^1 = \top$ , so that  $\mathcal{TTA}(\text{PredAbs}_{[\varphi^*]}(\mathbf{B}))$  contains only (25) and (26); if  $val \in [7.5, 8)$ , this forces  $B_0^1 = \top$ , so that  $\mathcal{TTA}(\text{PredAbs}_{[\varphi^*]}(\mathbf{B}))$  contains only (26); if  $val \in [8, 9)$ , this forces  $B_0^2 = \top$ , so that  $\mathcal{TTA}(\text{PredAbs}_{[\varphi^*]}(\mathbf{B}))$  reduces only to (27); if  $val \in [9, 10)$ , this makes the whole formula  $\mathcal{LR}\mathcal{A}$ -inconsistent, so that  $\mathcal{TTA}(\text{PredAbs}_{[\varphi^*]}(\mathbf{B}))$  is empty.

Now suppose, e.g., we set  $(t_{\text{dep}} = 7.4)$ , so that  $\mathcal{TTA}(\text{PredAbs}_{[\varphi^*]}(\mathbf{B}))$  contains only (25) and (26). Let (25) be the first assignment selected in the for loop, that is,  $\mu^{\mathbf{B}} \stackrel{\text{def}}{=} (B_0^1 \wedge \neg B_0^2 \wedge \neg B_0^3 \wedge B_1^1 \wedge \neg B_1^2 \wedge \neg B_1^3)$ . Propagating its truth values inside  $\varphi^*$  and  $w^*$  in (20) and (21) and simplifying the

truth values, we obtain:<sup>1</sup>

$$\begin{aligned} \varphi_{[\mu^{\mathbf{B}}]}^* &= (t_{\text{dep}} = 7.4) \wedge \llbracket t_{\text{dep}} \in [7, 10] \rrbracket \\ &\quad \wedge (t_{\text{arr}} = t_{\text{dep}} + x_1 + x_2) \wedge (t_{\text{arr}} < 11) \\ &\quad \wedge \llbracket t_{\text{dep}} \in [7, 8] \rrbracket \wedge \llbracket x_1 \in [0.5, 1] \rrbracket \\ &\quad \wedge \llbracket t_{\text{dep}} + x_1 \in [7, 8] \rrbracket \wedge \llbracket x_2 \in [1, 1.5] \rrbracket \\ w_{[\mu^{\mathbf{B}}]}^* &= f_{l_0 l_1}^1(x_1) \cdot f_{l_1 l_2}^1(x_2) \end{aligned}$$

$\varphi_{[\mu^{\mathbf{B}}]}^*$  is a conjunction of  $\mathcal{LR}\mathcal{A}$ -literals, so that  $\text{WMI}_{\text{nb}}$  can be invoked on it directly without further invoking another predicate abstraction.  $\diamond$

## 6 Related Work

Most works on probabilistic inference in hybrid graphical models are either limited to joint Gaussian distributions [Lauritzen and Jensen, 2001], or perform approximate inference [Gogate and Dechter, 2005; Afshar *et al.*, 2016]. A recent line of research focused on developing exact inference algorithms for graphical models with mixtures of polynomials [Shenoy and West, 2011; Sanner and Abbasnejad, 2012; Wang *et al.*, 2014]. The WMI formalism extends these approaches allowing to represent constraints in terms of arbitrary combinations of Boolean connectives.

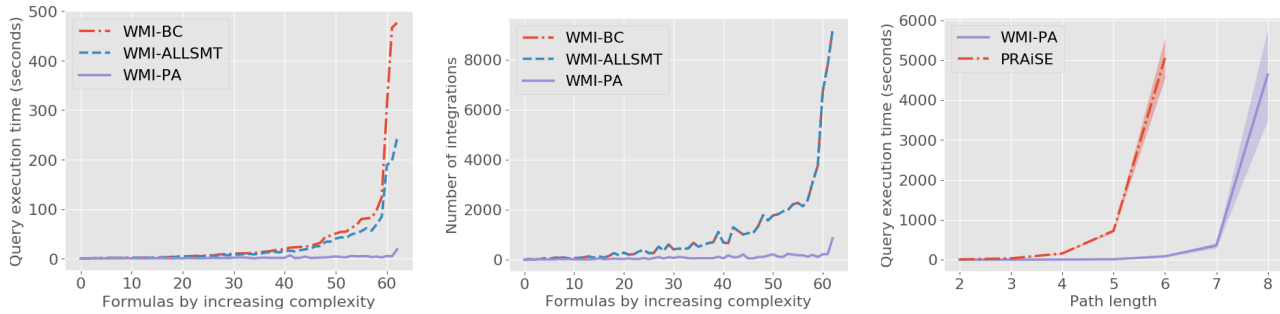
The first solver for exact WMI [Belle *et al.*, 2015] was a proof-of-concept relying on a simple block-clause strategy (WMI-BC in the following), which iteratively generates new models by adding the negation of the latest model to the formula for the following iteration. In the propositional (WMC) case, substantial efficiency gains can be obtained by leveraging component caching techniques [Sang *et al.*, 2004; Bacchus *et al.*, 2009]. Unfortunately, these strategies are difficult to apply in the WMI case, because of the additional coupling induced by algebraic constraints. A recent work [Belle *et al.*, 2016] did show substantial computational savings in adapting #DPLL with component caching from the WMC to the WMI case. The approach however works with purely piecewise polynomial densities, and cannot be applied whenever algebraic constraints exist between variables, e.g. their sum being within a certain range, a rather common situation in many practical cases (see §4). In the same paper, an approach equivalent to WMI-ALLSMT applied to the original WMI formulation was shown to improve over the WMI-BC baseline. Nevertheless, further improvements are not possible without revising the formulation as we do in this work.

Closest to WMI is probabilistic inference modulo theories [de Salvo Braz *et al.*, 2016], an elegant framework where a combination of DPLL, SMT and symbolic variable elimination allows to perform probabilistic inference on a variety of theories. While initially focused on integer arithmetic, the system developed by the authors (called PRAISE<sup>2</sup>) was recently provided with support for arithmetic over the reals and polynomial densities.

<sup>1</sup>Notice that  $\llbracket t_{\text{dep}} \in [7, 10] \rrbracket$  and  $\llbracket t_{\text{dep}} \in [7, 8] \rrbracket$  are made redundant by  $(t_{\text{dep}} = 7.4)$ ; however, they do not affect the result.

<sup>2</sup><http://aic-sri-international.github.io/aic-praise/>

Figure 1: Query execution times and number of integrals computed in the synthetic setting by the three WMI approaches (resp. left and center). Query execution times of WMI-PA and PRAISE on the Strategic Road Network setting (right).



## 7 Experiments

Our implementation uses MATHSAT5<sup>3</sup> [Cimatti *et al.*, 2013] for SMT reasoning and LATTE INTEGRALE<sup>4</sup> [Loera *et al.*, 2012] to compute integrals of polynomials. All experiments were run on a Virtual Machine with 7 cores running at a frequency of 2.2 GHz and 94 GB of RAM. Query timeout was set at 10000 seconds. When terminating, on each query all tools returned the same value modulo roundings.

**Synthetic setting.** To evaluate the ability of WMI-PA to deal with weight functions with arbitrarily complex supports, we randomly generated  $\mathcal{LR}\mathcal{A}$  formulas like:

$$\varphi(\mathbf{x}, \mathbf{A}) = \bigwedge_{i=1}^N \left( \left( \bigwedge_{x \in R_i^{\mathcal{LR}\mathcal{A}}(\mathbf{x})} x \right) \vee \bigvee_{A \in R_i^{\mathbb{B}}(\mathbf{A})} A \right)$$

where  $N$  is a parameter controlling the size of  $\varphi(\mathbf{x}, \mathbf{A})$ ,  $R_i^{\mathcal{LR}\mathcal{A}}(\mathbf{x})$  are randomly generated sets of  $\mathcal{LR}\mathcal{A}$ -atoms over  $\mathbf{x}$  and  $R_i^{\mathbb{B}}(\mathbf{A})$  are random subsets of  $\mathbf{A}$ , for  $i \in [1, N]$ . For each formula  $\varphi(\mathbf{x}, \mathbf{A})$ , randomly generated (polynomial) weight functions are associated to the Boolean variables  $\mathbf{A}$ . We then compared the performance of WMI-PA in computing randomly generated queries against the WMI-BC baseline (see §6) and WMI-ALLSMT.

Figure 1(left) shows the execution time in seconds of the three methods over all randomly generated formulas, where formulas are ordered by increasing running time according to the baseline. Whereas WMI-ALLSMT performs better than the baseline for the most difficult cases, WMI-PA achieves drastic speedups wrt both alternatives. Figure 1(center) reports the number of integrals computed by the three methods, where formulas are ordered as before. The curves for WMI-BC and WMI-ALLSMT are indistinguishable, an expected result as the two formulations generate the very same set of total truth assignments, with WMI-ALLSMT doing it more efficiently. Conversely, the predicate abstraction steps of WMI-PA allow it to drastically reduce the number of assignments, and thus integrals to be computed.

**Real-world setting.** In order to show the applicability of our method to real world tasks, we implemented the case

study described in §4. The data was taken from the Strategic Road Network Dataset<sup>5</sup>, which provides average journey times on all the motorways managed by the English Highways Agency. From this dataset we extrapolated polynomial distributions of journey times between all junctions, with a 15 minutes granularity. In this setting, the task is to perform queries of the form  $P((t_N \leq t_{arr}) | t_{dep} \wedge \{l_i\}_{i=0}^N)$ . We encoded an equivalent formulation for PRAISE and compared the two approaches<sup>6</sup> over random queries for increasing path lengths. The results in Figure 1(right) show the mean and standard deviation of the execution times for both methods, computed over 5 randomly generated queries for each path length. Our approach is much faster, being able to compute queries up to two steps longer without reaching the timeout.<sup>7</sup>

## 8 Conclusion and Future Work

In this paper we proposed a revised definition of WMI which addresses some theoretical and practical limitations of the original formulation. Building on the properties of the novel formulation, we developed an efficient WMI algorithm combining a substantial reduction in the number of integrations with their efficient enumeration. Experimental comparisons over synthetic and real-world data confirm the drastic efficiency improvement over existing alternatives.

Future research directions include the study of decomposition strategies for further scalability, and the generalization of WMI to combinations of theories, like linear arithmetic over reals and integers.

## Acknowledgments

This work has been partially supported by the EC project CogNet, 671625 (H2020-ICT-2014-2, Research and Innovation action). We thank Alberto Griggio for useful suggestions about the usage of MATHSAT5.

<sup>5</sup><https://data.gov.uk/dataset/dft-eng-srn-routes-journey-times>

<sup>6</sup>WMI-BC and WMI-ALLSMT are not considered here as their execution times are prohibitive for all but the smallest path lengths.

<sup>7</sup>Note that the complexity of the query is due to the combination of the path length and the number of time intervals in which the time horizon is divided ( $M=12$  in these experiments). For paths of length 8, the total number of potential cases is  $M^8 = 429981696$ . Clearly, many of these cases are unfeasible and are ruled out by the SMT solver before the integration.

<sup>3</sup><http://mathsat.fbk.eu/>

<sup>4</sup><https://www.math.ucdavis.edu/~latte/>



## References

- [Afshar *et al.*, 2016] Hadi Mohasel Afshar, Scott Sanner, and Christfried Webers. Closed-form gibbs sampling for graphical models with algebraic constraints. In *AAAI*, 2016.
- [Bacchus *et al.*, 2009] Fahiem Bacchus, Shannon Dalmao, and Toniann Pitassi. Solving #SAT and Bayesian inference with backtracking search. *Journal of Artificial Intelligence Research*, 34(1):391–442, 2009.
- [Baldoni *et al.*, 2011] Velleda Baldoni, Nicole Berline, Jesus De Loera, Matthias Köppe, and Michéle Vergne. How to integrate a polynomial over a simplex. *Mathematics of Computation*, 80(273):297–325, 2011.
- [Barrett *et al.*, 2009] Clark W. Barrett, Roberto Sebastiani, Sanjit A. Seshia, and Cesare Tinelli. Satisfiability Modulo Theories. In *Handbook of Satisfiability*, chapter 26, pages 825–885. IOS Press, 2009.
- [Belle *et al.*, 2015] Vaishak Belle, Andrea Passerini, and Guy Van den Broeck. Probabilistic inference in hybrid domains by weighted model integration. In *IJCAI*, 2015.
- [Belle *et al.*, 2016] Vaishak Belle, Guy Van den Broeck, and Andrea Passerini. Component caching in hybrid domains with piecewise polynomial densities. In *AAAI*, 2016.
- [Cavada *et al.*, 2007] Roberto Cavada, Alessandro Cimatti, Anders Franzén, Krishnamani Kalyanasundaram, Marco Roveri, and RK Shyamasundar. Computing Predicate Abstractions by Integrating BDDs and SMT Solvers. In *FM-CAD*, 2007.
- [Chavira and Darwiche, 2008] Mark Chavira and Adnan Darwiche. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6-7):772–799, 2008.
- [Choi *et al.*, 2013] Arthur Choi, Doga Kisa, and Adnan Darwiche. Compiling probabilistic graphical models using sentential decision diagrams. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 121–132. Springer, 2013.
- [Cimatti *et al.*, 2013] Alessandro Cimatti, Alberto Griggio, Bastiaan Joost Schaafsma, and Roberto Sebastiani. The MathSAT 5 SMT Solver. In *TACAS*, 2013.
- [de Salvo Braz *et al.*, 2016] Rodrigo de Salvo Braz, Ciaran O’Reilly, Vibhav Gogate, and Rina Dechter. Probabilistic Inference Modulo Theories. In *IJCAI*, 2016.
- [Gogate and Dechter, 2005] Vibhav Gogate and Rina Dechter. Approximate inference algorithms for hybrid bayesian networks with discrete constraints. In *UAI*, 2005.
- [Graf and Saïdi, 1997] Susanne Graf and Hassen Saïdi. Construction of abstract state graphs with pvs. In *CAV*, 1997.
- [Lahiri *et al.*, 2006] Shuvendu K. Lahiri, Robert Nieuwenhuis, and Albert Oliveras. SMT techniques for fast predicate abstraction. In *CAV*, 2006.
- [Lauritzen and Jensen, 2001] Steffen L. Lauritzen and Frank Jensen. Stable local computation with conditional gaussian distributions. *Statistics and Computing*, 11(2):191–203, 2001.
- [Loera *et al.*, 2012] Jesus De Loera, Brandon Dutra, Matthias Koeppel, Stanislav Moreinis, Gregory Pinto, and Jianqiu Wu. Software for exact integration of polynomials over polyhedra. *ACM Communications in Computer Algebra*, 45(3/4):169–172, 2012.
- [Sang *et al.*, 2004] Tian Sang, Fahiem Bacchus, Paul Beame, Henry A. Kautz, and Toniann Pitassi. Combining component caching and clause learning for effective model counting. In *SAT*, 2004.
- [Sanner and Abbasnejad, 2012] Scott Sanner and Ehsan Abbasnejad. Symbolic variable elimination for discrete and continuous graphical models. In *AAAI*, 2012.
- [Sebastiani, 2007] Roberto Sebastiani. Lazy Satisfiability Modulo Theories. *Journal on Satisfiability, Boolean Modeling and Computation*, *JSAT*, 3(3-4):141–224, 2007.
- [Shenoy and West, 2011] Prakash P. Shenoy and James C. West. Inference in hybrid bayesian networks using mixtures of polynomials. *International Journal of Approximate Reasoning*, 52(5):641–657, 2011.
- [Suciu *et al.*, 2011] Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. Probabilistic databases. *Synthesis Lectures on Data Management*, 3(2):1–180, 2011.
- [Wang *et al.*, 2014] Shenlong Wang, Alex Schwing, and Raquel Urtasun. Efficient inference of continuous markov random fields with polynomial potentials. In *NIPS*, 2014.