SMT-Based Weighted Model Integration

Roberto Sebastiani[1]

joint work with
Paolo Morettin[1], Andrea Passerini[1]

with contributions by
Samuel Kolb[2], Luc De Raedt[2], Francesco Sommavilla[1], Pedro Zuidberg[2]

[1] University of Trento, Italy

[2] KU Leuven; Belgium

– 17^{th} International Workshop on Satisfiability Modulo Theories, SMT 2019 –

– 6^{th} Vampire Workshop, Vampire 2019 –

Context

Goal

- Efficiently perform probabilistic inference in hybrid domains
  - both Boolean and continuous variables
  - arithmetical and logical constraints
- Using SMT-based Weighted Model Integration.

Brief History

- Weighted Model Counting (WMC) [10] [Chavira & Darwiche, AIJ 2008]
  - SAT-based probabilistic inference in Boolean domains
- Weighted Model Integration (WMI) [8] [Belle, Passerini & Van den Broeck, IJCAI 2015]
  - SMT-based probabilistic inference in hybrid domains (Boolean+arithmetic)
- Weighted Model Integration, revisited [19, 20] [Morettin, Passerini & Sebastiani, IJCAI 2017, AIJ 2019]
  - WMI reformulated from scratch, novel SMT-based algorithms

Context

Goal

- Efficiently perform probabilistic inference in hybrid domains
  - both Boolean and continuous variables
  - arithmetical and logical constraints
- Using SMT-based Weighted Model Integration.

Brief History

- Weighted Model Counting (WMC) [10] [Chavira & Darwiche, AIJ 2008]
  - SAT-based probabilistic inference in Boolean domains
- Weighted Model Integration (WMI) [8] [Belle, Passerini & Van den Broeck, IJCAI 2015]
  - SMT-based probabilistic inference in hybrid domains (Boolean+arithmetic)
- Weighted Model Integration, revisited [19, 20] [Morettin, Passerini & Sebastiani, IJCAI 2017, AIJ 2019]
  - WMI reformulated from scratch, novel SMT-based algorithms

Context

Goal

- Efficiently perform probabilistic inference in hybrid domains
  - both Boolean and continuous variables
  - arithmetical and logical constraints
- Using SMT-based Weighted Model Integration.

Brief History

- Weighted Model Counting (WMC) [10] [Chavira & Darwiche, AIJ 2008]
  - SAT-based probabilistic inference in Boolean domains
- Weighted Model Integration (WMI) [8] [Belle, Passerini & Van den Broeck, IJCAI 2015]
  - SMT-based probabilistic inference in hybrid domains (Boolean+arithmetic)
- Weighted Model Integration, revisited [19, 20] [Morettin, Passerini & Sebastiani, IJCAI 2017, AIJ 2019]
  - WMI reformulated from scratch, novel SMT-based algorithms

Context

Goal

- Efficiently perform probabilistic inference in hybrid domains
  - both Boolean and continuous variables
  - arithmetical and logical constraints
- Using SMT-based Weighted Model Integration.

Brief History

- Weighted Model Counting (WMC) [10] [Chavira & Darwiche, AIJ 2008]
  - SAT-based probabilistic inference in Boolean domains
- Weighted Model Integration (WMI) [8] [Belle, Passerini & Van den Broeck, IJCAI 2015]
  - SMT-based probabilistic inference in hybrid domains (Boolean+arithmetic)
- Weighted Model Integration, revisited [19, 20] [Morettin, Passerini & Sebastiani, IJCAI 2017, AIJ 2019]
  - WMI reformulated from scratch, novel SMT-based algorithms

Context

Goal

- Efficiently perform probabilistic inference in hybrid domains
  - both Boolean and continuous variables
  - arithmetical and logical constraints
- Using SMT-based Weighted Model Integration.

Brief History

- Weighted Model Counting (WMC) [10] [Chavira & Darwiche, AIJ 2008]
  - SAT-based probabilistic inference in Boolean domains
- Weighted Model Integration (WMI) [8] [Belle, Passerini & Van den Broeck, IJCAI 2015]
  - SMT-based probabilistic inference in hybrid domains (Boolean+arithmetic)
- Weighted Model Integration, revisited [19, 20] [Morettin, Passerini & Sebastiani, IJCAI 2017, AIJ 2019]
  - WMI reformulated from scratch, novel SMT-based algorithms

Outline

1. Background
2. Weighted Model Integration, Revisited
3. SMT-Based WMI Computation
4. A Case Study: The Road Network Problem
5. Experimental Evaluations
6. Ongoing and Future Work

Outline

Weighted Model Counting

Definition (Weighted Model Count)

Let φ be a propositional formula on $\mathbf{A} \stackrel{\text{def}}{=} \{A_1, ..., A_M\}$ and let w be a function associating a non-negative weight to each literal on $Atoms(\varphi)$. Then the Weighted Model Count of φ is:

$$\mathrm{WMC}(\varphi, w) = \sum_{\mu \in \mathcal{TTA}(\varphi)} \prod_{\ell \in \mu} w(\ell).$$

Proposition ([10, 8])

The probability of a query q given evidence e in a Boolean Markov Network N is computed as:

$$Pr_N(q|e) = \frac{WMC(q \wedge e \wedge \Delta, w)}{WMC(e \wedge \Delta, w)}, \quad \text{where } \Delta \text{ encodes } N \text{ and } w \text{ the potential.}$$

Many efficient computing techniques

- based on knowledge compilation [12, 21] or exhaustive DPLL search [23]
- improved by component caching techniques [22, 6]

Weighted Model Counting

Definition (Weighted Model Count)

Let φ be a propositional formula on $\mathbf{A} \stackrel{\text{def}}{=} \{A_1, ..., A_M\}$ and let w be a function associating a non-negative weight to each literal on $Atoms(\varphi)$. Then the Weighted Model Count of φ is:

$$\text{WMC}(\varphi, w) = \sum_{\mu \in \mathcal{TTA}(\varphi)} \prod_{\ell \in \mu} w(\ell).$$

Proposition ([10, 8])

The probability of a query q given evidence e in a Boolean Markov Network N is computed as:

$$Pr_N(q|e) = \frac{WMC(q \wedge e \wedge \Delta, w)}{WMC(e \wedge \Delta, w)}, \quad \textit{where } \Delta \textit{ encodes } N \textit{ and } w \textit{ the potential.}$$

Many efficient computing techniques

- based on knowledge compilation [12, 21] or exhaustive DPLL search [23]
- improved by component caching techniques [22, 6]

Weighted Model Counting

Definition (Weighted Model Count)

Let φ be a propositional formula on $\mathbf{A} \stackrel{\text{def}}{=} \{A_1, ..., A_M\}$ and let w be a function associating a non-negative weight to each literal on $Atoms(\varphi)$. Then the Weighted Model Count of φ is:

$$\text{WMC}(\varphi, w) = \sum_{\mu \in \mathcal{TTA}(\varphi)} \prod_{\ell \in \mu} w(\ell).$$

Proposition ([10, 8])

The probability of a query q given evidence e in a Boolean Markov Network N is computed as:

$$Pr_N(q|e) = \frac{WMC(q \wedge e \wedge \Delta, w)}{WMC(e \wedge \Delta, w)}, \quad \textit{where } \Delta \textit{ encodes } N \textit{ and } w \textit{ the potential.}$$

Many efficient computing techniques

- based on knowledge compilation [12, 21] or exhaustive DPLL search [23]
- improved by component caching techniques [22, 6]

Weighted Model Integration [8]

Definition (Weighted Model Integral [8])

Let φ be a $\mathcal{LRA}$ formula on $\mathbf{x} \stackrel{\text{def}}{=} \{x_1, ..., x_N\}$ and $\mathbf{A} \stackrel{\text{def}}{=} \{A_1, ..., A_M\}$. Let w be a function associating an expression (possibly constant) over $\mathbf{x}$ to each literal whose atom occurs in φ. Then the Weighted Model Integral of φ is defined as:

$$\text{WMI}_{\text{old}}(\varphi, w) = \sum_{\mu \in \mathcal{TTA}(\varphi)} \int_{\mu^{\mathcal{LRA}}} \prod_{\ell \in \mu} w(\ell) \, d\mathbf{x}, \quad s.t. \; \mu \stackrel{\text{def}}{=} \mu^{\mathbf{A}} \wedge \mu^{\mathcal{LRA}}$$

Note: $\langle \varphi, w \rangle$ implicitly defines an un-normalized probability distribution

If $P(\mathbf{x})$ is polynomial and $\mu^{\mathcal{LRA}}(\mathbf{x})$ is a conjunction of linear constraints, then $\int_{\mu^{\mathcal{LRA}}} P(\mathbf{x}) \, d\mathbf{x}$ can be exactly computed [7] (e.g., by LATTE INTEGRALE [18])

Proposition ([8])

The probability of a query q given evidence e in a Hybrid Markov Network N is computed as:

$$Pr_N(q|e) = \frac{\text{WMI}_{\text{old}}(q \wedge e \wedge \Delta, w)}{\text{WMI}_{\text{old}}(e \wedge \Delta, w)}, \quad \textit{where } \Delta \textit{ encodes } N \textit{ and } w \textit{ the potential.}$$

Weighted Model Integration [8]

Definition (Weighted Model Integral [8])

Let φ be a $\mathcal{LRA}$ formula on $\mathbf{x} \stackrel{\text{def}}{=} \{x_1, ..., x_N\}$ and $\mathbf{A} \stackrel{\text{def}}{=} \{A_1, ..., A_M\}$. Let w be a function associating an expression (possibly constant) over $\mathbf{x}$ to each literal whose atom occurs in φ. Then the Weighted Model Integral of φ is defined as:

$$\mathsf{WMI}_{\text{old}}(\varphi, w) = \sum_{\mu \in \mathcal{TTA}(\varphi)} \int_{\mu^{\mathcal{LRA}}} \prod_{\ell \in \mu} w(\ell)\, d\mathbf{x}, \quad s.t.\ \mu \stackrel{\text{def}}{=} \mu^{\mathbf{A}} \wedge \mu^{\mathcal{LRA}}$$

Note: $\langle \varphi, w \rangle$ implicitly defines an un-normalized probability distribution

If $P(\mathbf{x})$ is polynomial and $\mu^{\mathcal{LRA}}(\mathbf{x})$ is a conjunction of linear constraints, then $\int_{\mu^{\mathcal{LRA}}} P(\mathbf{x})\, d\mathbf{x}$ can be exactly computed [7] (e.g., by LATTE INTEGRALE [18])

Proposition ([8])

The probability of a query q given evidence e in a Hybrid Markov Network N is computed as:

$$Pr_N(q|e) = \frac{\mathsf{WMI}_{\text{old}}(q \wedge e \wedge \Delta, w)}{\mathsf{WMI}_{\text{old}}(e \wedge \Delta, w)}, \quad \textit{where } \Delta \textit{ encodes } N \textit{ and } w \textit{ the potential.}$$

Weighted Model Integration [8]

Definition (Weighted Model Integral [8])

Let φ be a $\mathcal{LRA}$ formula on $\mathbf{x} \stackrel{\text{def}}{=} \{x_1, ..., x_N\}$ and $\mathbf{A} \stackrel{\text{def}}{=} \{A_1, ..., A_M\}$. Let w be a function associating an expression (possibly constant) over $\mathbf{x}$ to each literal whose atom occurs in φ. Then the Weighted Model Integral of φ is defined as:

$$\mathsf{WMI}_{\mathsf{old}}(\varphi, w) = \sum_{\mu \in \mathcal{TTA}(\varphi)} \int_{\mu^{\mathcal{LRA}}} \prod_{\ell \in \mu} w(\ell)\, \mathrm{d}\mathbf{x}, \quad s.t.\ \mu \stackrel{\text{def}}{=} \mu^{\mathbf{A}} \wedge \mu^{\mathcal{LRA}}$$

Note: $\langle \varphi, w \rangle$ implicitly defines an un-normalized probability distribution

If $P(\mathbf{x})$ is polynomial and $\mu^{\mathcal{LRA}}(\mathbf{x})$ is a conjunction of linear constraints, then $\int_{\mu^{\mathcal{LRA}}} P(\mathbf{x})\, \mathrm{d}\mathbf{x}$ can be exactly computed [7] (e.g., by LattE Integrale [18])

Proposition ([8])

The probability of a query q given evidence e in a Hybrid Markov Network N is computed as:

$$Pr_N(q|e) = \frac{\mathsf{WMI}_{\mathsf{old}}(q \wedge e \wedge \Delta, w)}{\mathsf{WMI}_{\mathsf{old}}(e \wedge \Delta, w)}, \quad \textit{where } \Delta \textit{ encodes } N \textit{ and } w \textit{ the potential.}$$

Weighted Model Integration - Example

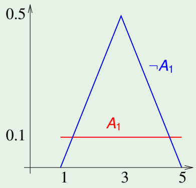
Example

$$\varphi \stackrel{\text{def}}{=} (A_2 \rightarrow ((1 \leq x) \wedge (x \leq 3))) \wedge (A_3 \rightarrow (\neg(x \leq 3) \wedge (x \leq 5))) \wedge (A_1 \leftrightarrow (\neg A_2 \wedge \neg A_3)) \wedge (1 \leq x) \wedge (x \leq 5).$$

Let $w(A_1) = 0.1$, $w(A_2) = (0.25 \cdot x - 0.25)$, $w(A_3) = (1.25 - 0.25 \cdot x)$, $w(l) = 1$ for the others.

$$\mathcal{TTA}(\varphi) = \left\{ \begin{array}{l} A_1 \wedge \neg A_2 \wedge \neg A_3 \wedge (1 \leq x) \wedge (x \leq 5) \wedge (x \leq 3), \\ A_1 \wedge \neg A_2 \wedge \neg A_3 \wedge (1 \leq x) \wedge (x \leq 5) \wedge \neg(x \leq 3), \\ \neg A_1 \wedge A_2 \wedge \neg A_3 \wedge (1 \leq x) \wedge (x \leq 5) \wedge (x \leq 3), \\ \neg A_1 \wedge \neg A_2 \wedge A_3 \wedge (1 \leq x) \wedge (x \leq 5) \wedge \neg(x \leq 3) \end{array} \right\}.$$

$$\begin{aligned} \mathrm{WMI}_{\mathrm{old}}(\varphi, w) &= \int_{(1 \leq x) \wedge (x \leq 5) \wedge (x \leq 3)} w(A_1)\, \mathrm{d}x + \int_{(1 \leq x) \wedge (x \leq 5) \wedge \neg(x \leq 3)} w(A_1)\, \mathrm{d}x \\ &+ \int_{(1 \leq x) \wedge (x \leq 5) \wedge (x \leq 3)} w(A_2)\, \mathrm{d}x + \int_{(1 \leq x) \wedge (x \leq 5) \wedge \neg(x \leq 3)} w(A_3)\, \mathrm{d}x \\ &= \int_{[1,3]} 0.1\, \mathrm{d}x + \int_{(3,5]} 0.1\, \mathrm{d}x + \int_{[1,3]} 0.25 \cdot x - 0.25\, \mathrm{d}x + \int_{(3,5]} 1.25 - 0.25 \cdot x\, \mathrm{d}x = (\ldots) = 1.4 \end{aligned}$$



Models an unnormalized distribution over x in $[1, 5]$, which:

- is uniform with $w = 0.1$ if A_1 is true
- is modeled as a triangular distribution with mode $w = 0.5$ at $x = 3$ otherwise.

Weighted Model Integration - Example (cont.)

Example

Given the previous unnormalized distribution $\langle \varphi, w \rangle$ and the information that $A_1 = \bot$ (evidence), the probability that $x \leq 2$ (query) is:

$$P_{(\varphi,w)}(x \leq 2 | A_1 = \bot) = \frac{\mathrm{WMI_{old}}(\varphi \wedge \neg A_1 \wedge (x \leq 2), w)}{\mathrm{WMI_{old}}(\varphi \wedge \neg A_1, w)} = \frac{0.125}{1.0} = 0.125$$

$$\begin{aligned}
\mathrm{WMI_{old}}(\varphi \wedge \neg A_1, w) &= \int_{(1 \leq x) \wedge (x \leq 5) \wedge (x \leq 3)} w(A_2)\, \mathrm{d}x + \int_{(1 \leq x) \wedge (x \leq 5) \wedge \neg (x \leq 3)} w(A_3)\, \mathrm{d}x \\
&= \int_{[1,3]} 0.25 \cdot x - 0.25\, \mathrm{d}x + \int_{(3,5]} 1.25 - 0.25 \cdot x\, \mathrm{d}x = (...) = 1.0 \\
\mathrm{WMI_{old}}(\varphi \wedge \neg A_1 \wedge (x \leq 2), w) &= \int_{(1 \leq x) \wedge (x \leq 5) \wedge (x \leq 3) \wedge (x \leq 2)} w(A_2)\, \mathrm{d}x \\
&= \int_{[1,2]} 0.25 \cdot x - 0.25\, \mathrm{d}x = (...) = 0.125
\end{aligned}$$

Outline

Basic case: WMI Without Atomic Propositions

Definition

Assume φ does not contain atomic propositions and $w : \mathbb{R}^N \longmapsto \mathbb{R}^+$. Then we define the Weighted Model Integral of w over φ on $\mathbf{x}$ as:

$$\mathsf{WMI}_{\mathsf{nb}}(\varphi, w|\mathbf{x}) \stackrel{\text{def}}{=} \int_{\varphi(\mathbf{x})} w(\mathbf{x})\, \mathrm{d}\mathbf{x},$$

"$_{\mathsf{nb}}$" meaning "no-Booleans", that is, as the integral of $w(\mathbf{x})$ over the set $\{\mathbf{x} \mid \varphi(\mathbf{x})\ \textit{is true}\}$.

Proposition

$$\begin{aligned}\mathsf{WMI}_{\mathsf{nb}}(\varphi, w|\mathbf{x}) &= \sum_{\mu^{\mathcal{LRA}} \in \mathcal{TTA}(\varphi)} \mathsf{WMI}_{\mathsf{nb}}(\mu^{\mathcal{LRA}}, w|\mathbf{x}) \\ &= \sum_{\mu^{\mathcal{LRA}} \in \mathcal{TA}(\varphi)} \mathsf{WMI}_{\mathsf{nb}}(\mu^{\mathcal{LRA}}, w|\mathbf{x}).\end{aligned}$$

Note: $\mathsf{WMI}_{\mathsf{nb}}(\mu^{\mathcal{LRA}}, w|\mathbf{x}) \stackrel{\text{def}}{=} \int_{\mu^{\mathcal{LRA}}(\mathbf{x})} w(\mathbf{x})\, \mathrm{d}\mathbf{x}$ *can be computed exactly if* $w(\mathbf{x})$ *is polynomial [7].*

Basic case: WMI Without Atomic Propositions

Definition

Assume φ does not contain atomic propositions and $w : \mathbb{R}^N \longmapsto \mathbb{R}^+$. Then we define the Weighted Model Integral of w over φ on $\mathbf{x}$ as:

$$\mathsf{WMI}_{\mathsf{nb}}(\varphi, w|\mathbf{x}) \stackrel{\text{def}}{=} \int_{\varphi(\mathbf{x})} w(\mathbf{x})\, \mathrm{d}\mathbf{x},$$

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

Proposition

$$\begin{aligned}\mathsf{WMI}_{\mathsf{nb}}(\varphi, w|\mathbf{x}) &= \sum_{\mu^{\mathcal{LRA}} \in \mathcal{TTA}(\varphi)} \mathsf{WMI}_{\mathsf{nb}}(\mu^{\mathcal{LRA}}, w|\mathbf{x}) \\ &= \sum_{\mu^{\mathcal{LRA}} \in \mathcal{TA}(\varphi)} \mathsf{WMI}_{\mathsf{nb}}(\mu^{\mathcal{LRA}}, w|\mathbf{x}).\end{aligned}$$

Note: $\mathsf{WMI}_{\mathsf{nb}}(\mu^{\mathcal{LRA}}, w|\mathbf{x}) \stackrel{\text{def}}{=} \int_{\mu^{\mathcal{LRA}}(\mathbf{x})} w(\mathbf{x})\, \mathrm{d}\mathbf{x}$ *can be computed exactly if* $w(\mathbf{x})$ *is polynomial [7].*

General Case: WMI With Atomic Propositions

Definition

We consider a $\mathcal{LRA}$-formula $\varphi(\mathbf{x}, \mathbf{A})$ and $w(\mathbf{x}, \mathbf{A})$ s.t. $w : \mathbb{R}^N \times \mathbb{B}^M \longmapsto \mathbb{R}^+$.
The Weighted Model Integral of w over φ is defined as follows:

$$\mathsf{WMI}(\varphi, w|\mathbf{x}, \mathbf{A}) \stackrel{\text{def}}{=} \sum_{\mu^{\mathbf{A}} \in \mathbb{B}^M} \mathsf{WMI}_{\mathsf{nb}}(\varphi_{[\mu^{\mathbf{A}}]}, w_{[\mu^{\mathbf{A}}]}|\mathbf{x}) = \sum_{\mu^{\mathbf{A}} \in \mathbb{B}^M} \sum_{\mu^{\mathcal{LRA}} \in \mathcal{TA}(\varphi_{[\mu^{\mathbf{A}}]})} \int_{\mathcal{LRA}(\mathbf{x})} w_{[\mu^{\mathbf{A}}]}(\mathbf{x})\, \mathrm{d}\mathbf{x},$$

- the $\mu^{\mathbf{A}}$'s are all total truth assignments on $\mathbf{A}$,
- $\varphi_{[\mu^{\mathbf{A}}]}(\mathbf{x})$ denotes (any formula equivalent to) the formula obtained from φ by substituting every Boolean value A_i with its truth value in $\mu^{\mathbf{A}}$ (thus $\varphi_{[\mu^{\mathbf{A}}]} : \mathbb{R}^N \longmapsto \mathbb{B}$)
- $w_{[\mu^{\mathbf{A}}]}(\mathbf{x})$ is w computed on $\mathbf{x}$ and on the truth values of $\mu^{\mathbf{A}}$ ($w_{[\mu^{\mathbf{A}}]} : \mathbb{R}^N \longmapsto \mathbb{R}^+$ if $\mu^{\mathbf{A}}$ total)

Note

- $w(\mathbf{x}, \mathbf{A})$ generic, not restricted in the form of products of weighs on literals of φ.
- if $w_{[\mu^{\mathbf{A}}]}(\mathbf{x})$ polynomial for every $\mu^{\mathbf{A}}$, then $\int_{\mathcal{LRA}(\mathbf{x})} w_{[\mu^{\mathbf{A}}]}(\mathbf{x})\, \mathrm{d}\mathbf{x}$ can be computed exactly

WMI - Example

Example

Let

- $\varphi \stackrel{\text{def}}{=} (A \leftrightarrow (x \geq 0)) \wedge (x \geq -1) \wedge (x \leq 1)$,
- $w(x, A) \stackrel{\text{def}}{=} [\![\text{If } A \text{ Then } x \text{ Else } -x]\!]$.

Then:

- 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 $w_{[\mu^{\mathbf{A}}]} = -x$.
- If $\mu^{\mathbf{A}} \stackrel{\text{def}}{=} \{\ A\}$, then $\varphi_{[\mu^{\mathbf{A}}]} = \ (x \geq 0) \wedge (x \geq -1) \wedge (x \leq 1)$ and $w_{[\mu^{\mathbf{A}}]} = x$.

Thus,

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

Some Results on WMI

Proposition

Given $\mathbf{x}$, $\mathbf{A}$, $w(\mathbf{x},\mathbf{A})$, $\varphi(\mathbf{x},\mathbf{A})$ *and* $\mathcal{TTA}(\varphi)$ *as above, we have that:*

$$\mathrm{WMI}(\varphi, w|\mathbf{x},\mathbf{A}) = \sum_{\mu^{\mathbf{A}}\wedge\mu^{\mathcal{LRA}}\in\mathcal{TTA}(\varphi)} \mathrm{WMI}_{\mathrm{nb}}(\mu^{\mathcal{LRA}}, w_{[\mu^{\mathbf{A}}]}|\mathbf{x}) = \sum_{\mu^{\mathbf{A}}\wedge\mu^{\mathcal{LRA}}\in\mathcal{TTA}(\varphi)} \int_{\mu^{\mathcal{LRA}}(\mathbf{x})} w_{[\mu^{\mathbf{A}}]}(\mathbf{x})\, d\mathbf{x}$$

Proposition

Given $\mathbf{x}$, $\mathbf{A}$, $w(\mathbf{x},\mathbf{A})$, $\varphi(\mathbf{x},\mathbf{A})$ *and* $\mathcal{TTA}(\varphi)$ *as above, we have that:*

$$\mathrm{WMI}(\varphi, w|\mathbf{x},\mathbf{A}) = \sum_{\mu^{\mathbf{A}}\in\mathcal{TTA}(\exists\mathbf{x}.\varphi)} \mathrm{WMI}_{\mathrm{nb}}(\varphi_{[\mu^{\mathbf{A}}]}, w_{[\mu^{\mathbf{A}}]}|\mathbf{x}) = \sum_{\mu^{\mathbf{A}}\in\mathcal{TTA}(\exists\mathbf{x}.\varphi)} \sum_{\mu^{\mathcal{LRA}}\in\mathcal{TA}(\varphi_{[\mu^{\mathbf{A}}]})} \int_{\mu^{\mathcal{LRA}}(\mathbf{x})} w_{[\mu^{\mathbf{A}}]}(\mathbf{x})\, d\mathbf{x},$$

- enumerate only $\mathcal{LRA}$-consistent $\mu^{\mathbf{A}}\wedge\mu^{\mathcal{LRA}}$'s propositionally satisfying φ
- enumerate only $\mu^{\mathbf{A}}$'s for which exists a $\mathcal{LRA}$-consistent (partial) $\mu^{\mathcal{LRA}}$ s.t. $\mu^{\mathbf{A}}\wedge\mu^{\mathcal{LRA}} \models_{\mathbb{B}} \varphi$

Some Results on WMI

Proposition

Given $\mathbf{x}$, $\mathbf{A}$, $w(\mathbf{x},\mathbf{A})$, $\varphi(\mathbf{x},\mathbf{A})$ *and* $\mathcal{TTA}(\varphi)$ *as above, we have that:*

$$\mathsf{WMI}(\varphi, w|\mathbf{x},\mathbf{A}) = \sum_{\mu^{\mathbf{A}} \wedge \mu^{\mathcal{LRA}} \in \mathcal{TTA}(\varphi)} \mathsf{WMI}_{\mathsf{nb}}(\mu^{\mathcal{LRA}}, w_{[\mu^{\mathbf{A}}]}|\mathbf{x}) = \sum_{\mu^{\mathbf{A}} \wedge \mu^{\mathcal{LRA}} \in \mathcal{TTA}(\varphi)} \int_{\mu^{\mathcal{LRA}}(\mathbf{x})} w_{[\mu^{\mathbf{A}}]}(\mathbf{x})\, d\mathbf{x}$$

Proposition

Given $\mathbf{x}$, $\mathbf{A}$, $w(\mathbf{x},\mathbf{A})$, $\varphi(\mathbf{x},\mathbf{A})$ *and* $\mathcal{TTA}(\varphi)$ *as above, we have that:*

$$\mathsf{WMI}(\varphi, w|\mathbf{x},\mathbf{A}) = \sum_{\mu^{\mathbf{A}} \in \mathcal{TTA}(\exists \mathbf{x}.\varphi)} \mathsf{WMI}_{\mathsf{nb}}(\varphi_{[\mu^{\mathbf{A}}]}, w_{[\mu^{\mathbf{A}}]}|\mathbf{x}) = \sum_{\mu^{\mathbf{A}} \in \mathcal{TTA}(\exists \mathbf{x}.\varphi)} \sum_{\mu^{\mathcal{LRA}} \in \mathcal{TA}(\varphi_{[\mu^{\mathbf{A}}]})} \int_{\mu^{\mathcal{LRA}}(\mathbf{x})} w_{[\mu^{\mathbf{A}}]}(\mathbf{x})\, d\mathbf{x},$$

- enumerate only $\mathcal{LRA}$-consistent $\mu^{\mathbf{A}} \wedge \mu^{\mathcal{LRA}}$'s propositionally satisfying φ
- enumerate only $\mu^{\mathbf{A}}$'s for which exists a $\mathcal{LRA}$-consistent (partial) $\mu^{\mathcal{LRA}}$ s.t. $\mu^{\mathbf{A}} \wedge \mu^{\mathcal{LRA}} \models_{\mathbb{B}} \varphi$

Feasibly computable WMIs: FIUC$^{\mathcal{LRA}}$ weight functions

FI$^{\mathcal{LRA}}$ weight function

A function $f(\mathbf{x})$ is feasibly integrable on a set of $\mathcal{LRA}$ constraints (FI$^{\mathcal{LRA}}$) if exists a procedure that can compute the integral $\int_{\mu^{\mathcal{LRA}}} f(\mathbf{x})\, d\mathbf{x}$, for all $\mu^{\mathcal{LRA}}$

- example: polynomials [7]

Definition (FIUC$^{\mathcal{LRA}}$ weight function)

Given $\langle \mathbf{x}, \mathbf{A} \rangle$ and

- a set of $\mathcal{LRA}$-conditions $\Psi \stackrel{\text{def}}{=} \{\psi_1(\mathbf{x}, \mathbf{A}), ..., \psi_K(\mathbf{x}, \mathbf{A})\}$;
- a support $\mathcal{LRA}$-formula $\chi(\mathbf{x}, \mathbf{A})$ s.t. $w(\mathbf{x}, \mathbf{A}) = 0$ where $\chi(\mathbf{x}, \mathbf{A})$ holds ($\top$ if not present);

we say that a weight function $w(\mathbf{x}, \mathbf{A})$ is feasibly integrable under $\mathcal{LRA}$ conditions (FIUC$^{\mathcal{LRA}}$) iff, for every total truth-value assignment $\mu^{\mathbf{A}}$ on $\mathbf{A}$ and μ^{Ψ} on Ψ, $w_{[\mu^{\mathbf{A}}, \mu^{\Psi}]}(\mathbf{x})$ is FI$^{\mathcal{LRA}}$.

Property

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

Feasibly computable WMIs: FIUC$^{\mathcal{LRA}}$ weight functions

FI$^{\mathcal{LRA}}$ weight function

A function $f(\mathbf{x})$ is feasibly integrable on a set of $\mathcal{LRA}$ constraints (FI$^{\mathcal{LRA}}$) if exists a procedure that can compute the integral $\int_{\mu^{\mathcal{LRA}}} f(\mathbf{x})\, d\mathbf{x}$, for all $\mu^{\mathcal{LRA}}$

- example: polynomials [7]

Definition (FIUC$^{\mathcal{LRA}}$ weight function)

Given $\langle \mathbf{x}, \mathbf{A} \rangle$ and

- a set of $\mathcal{LRA}$-conditions $\Psi \stackrel{\text{def}}{=} \{\psi_1(\mathbf{x}, \mathbf{A}), ..., \psi_K(\mathbf{x}, \mathbf{A})\}$;
- a support $\mathcal{LRA}$-formula $\chi(\mathbf{x}, \mathbf{A})$ s.t. $w(\mathbf{x}, \mathbf{A}) = 0$ where $\chi(\mathbf{x}, \mathbf{A})$ holds ($\top$ if not present);

we say that a weight function $w(\mathbf{x}, \mathbf{A})$ is feasibly integrable under $\mathcal{LRA}$ conditions (FIUC$^{\mathcal{LRA}}$) iff, for every total truth-value assignment $\mu^{\mathbf{A}}$ on $\mathbf{A}$ and μ^{Ψ} on Ψ, $w_{[\mu^{\mathbf{A}}, \mu^{\Psi}]}(\mathbf{x})$ is FI$^{\mathcal{LRA}}$.

Property

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

Feasibly computable WMIs: FIUC$^{\mathcal{LRA}}$ weight functions

FI$^{\mathcal{LRA}}$ weight function

A function $f(\mathbf{x})$ is feasibly integrable on a set of $\mathcal{LRA}$ constraints (FI$^{\mathcal{LRA}}$) if exists a procedure that can compute the integral $\int_{\mu^{\mathcal{LRA}}} f(\mathbf{x})\, d\mathbf{x}$, for all $\mu^{\mathcal{LRA}}$

- example: polynomials [7]

Definition (FIUC$^{\mathcal{LRA}}$ weight function)

Given $\langle \mathbf{x}, \mathbf{A} \rangle$ and

- a set of $\mathcal{LRA}$-conditions $\Psi \stackrel{\text{def}}{=} \{\psi_1(\mathbf{x}, \mathbf{A}), ..., \psi_K(\mathbf{x}, \mathbf{A})\}$;
- a support $\mathcal{LRA}$-formula $\chi(\mathbf{x}, \mathbf{A})$ s.t. $w(\mathbf{x}, \mathbf{A}) = 0$ where $\chi(\mathbf{x}, \mathbf{A})$ holds ($\top$ if not present);

we say that a weight function $w(\mathbf{x}, \mathbf{A})$ is feasibly integrable under $\mathcal{LRA}$ conditions (FIUC$^{\mathcal{LRA}}$) iff, for every total truth-value assignment $\mu^{\mathbf{A}}$ on $\mathbf{A}$ and μ^{Ψ} on Ψ, $w_{[\mu^{\mathbf{A}}, \mu^{\Psi}]}(\mathbf{x})$ is FI$^{\mathcal{LRA}}$.

Property

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

Support of FIUC$^{\mathcal{LRA}}$ weight functions: example

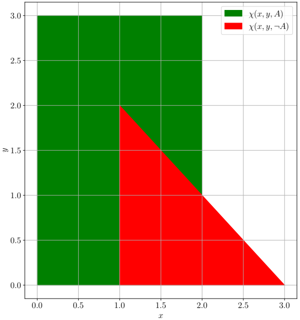
Example

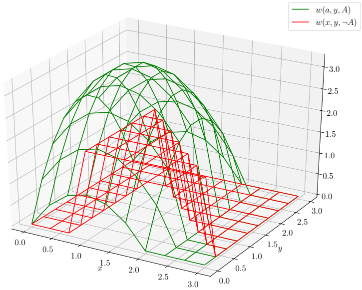
Let $\mathbf{x} \stackrel{\text{def}}{=} \{x, y\}$, $\mathbf{A} \stackrel{\text{def}}{=} \{A\}$,

$$\chi(\mathbf{x}, \mathbf{A}) \stackrel{\text{def}}{=} (A \rightarrow [\![x \in [0,2]]\!]) \wedge (\neg A \rightarrow ([\![x \in [1,3]]\!] \wedge (x + y \leq 3))) \wedge [\![y \in [1,3]]\!]$$

$$w(\mathbf{x}, \mathbf{A}) \stackrel{\text{def}}{=} [\![\text{If } A \text{ Then } (-x^2 - y^2 + 2x + 3y) \text{ Else } (-2x - 2y + 6)]\!].$$

(Note that outside the support the two polynomials may acquire negative values.)





A very relevant subcase of FIUC$^{\mathcal{LRA}}$ functions: P$^{\mathcal{LRA}}$ functions

Definition (P$^{\mathcal{LRA}}$ weight function)

Given $\langle \mathbf{x}, \mathbf{A} \rangle$, Ψ and χ as in FIUC$^{\mathcal{LRA}}$ definition, a weight function $w(\mathbf{x}, \mathbf{A})$ is called Polynomial under $\mathcal{LRA}$ conditions, P$^{\mathcal{LRA}}$ iff, for every total assignment $\mu^{\mathbf{A}}$ on $\mathbf{A}$ and μ^{Ψ} on Ψ, $w_{[\mu^{\mathbf{A}}\mu^{\Psi}]}(\mathbf{x})$ is a polynomial whose value is non-negative in the domain defined by μ^{Ψ}.

P$^{\mathcal{LRA}}$ functions are FIUC$^{\mathcal{LRA}}$ because polynomials can always be integrated exactly on sets of $\mathcal{LRA}$ literals [7].

We define a grammar to express P$^{\mathcal{LRA}}$ weight functions:

$$
\begin{aligned}
w \quad ::= \quad & c \mid x \mid -w \mid (w+w) \mid (w-w) \mid (w \cdot w) \mid \\
& [\![\text{If } \varphi \text{ Then } w \text{ Else } w]\!] \mid \\
& [\![\text{Case } \varphi : w;\ \varphi : w;\ \ldots]\!] \\
\chi \quad ::= \quad & \varphi
\end{aligned}
$$

where c is a real value, x is a real variable, w is a P$^{\mathcal{LRA}}$ weight function, φ is an $\mathcal{LRA}$ formula.

A very relevant subcase of FIUC$^{\mathcal{LRA}}$ functions: P$^{\mathcal{LRA}}$ functions

Definition (P$^{\mathcal{LRA}}$ weight function)

Given $\langle \mathbf{x}, \mathbf{A} \rangle$, Ψ and χ as in FIUC$^{\mathcal{LRA}}$ definition, a weight function $w(\mathbf{x}, \mathbf{A})$ is called Polynomial under $\mathcal{LRA}$ conditions, P$^{\mathcal{LRA}}$ iff, for every total assignment $\mu^{\mathbf{A}}$ on $\mathbf{A}$ and μ^{Ψ} on Ψ, $w_{[\mu^{\mathbf{A}}\mu^{\Psi}]}(\mathbf{x})$ is a polynomial whose value is non-negative in the domain defined by μ^{Ψ}.

P$^{\mathcal{LRA}}$ functions are FIUC$^{\mathcal{LRA}}$ because polynomials can always be integrated exactly on sets of $\mathcal{LRA}$ literals [7].

We define a grammar to express P$^{\mathcal{LRA}}$ weight functions:

$$\begin{array}{rcl} w & ::= & c \mid x \mid -w \mid (w + w) \mid (w - w) \mid (w \cdot w) \mid \\ & & [\![\text{If } \varphi \text{ Then } w \text{ Else } w]\!] \mid \\ & & [\![\text{Case } \varphi : w;\ \varphi : w;\ \ldots]\!] \\ \chi & ::= & \varphi \end{array}$$

where c is a real value, x is a real variable, w is a P$^{\mathcal{LRA}}$ weight function, φ is an $\mathcal{LRA}$ formula.

A very relevant subcase of FIUC$^{\mathcal{LRA}}$ functions: P$^{\mathcal{LRA}}$ functions

Definition (P$^{\mathcal{LRA}}$ weight function)

Given $\langle \mathbf{x}, \mathbf{A} \rangle$, Ψ and χ as in FIUC$^{\mathcal{LRA}}$ definition, a weight function $w(\mathbf{x}, \mathbf{A})$ is called Polynomial under $\mathcal{LRA}$ conditions, P$^{\mathcal{LRA}}$ iff, for every total assignment $\mu^{\mathbf{A}}$ on $\mathbf{A}$ and μ^{Ψ} on Ψ, $w_{[\mu^{\mathbf{A}}\mu^{\Psi}]}(\mathbf{x})$ is a polynomial whose value is non-negative in the domain defined by μ^{Ψ}.

P$^{\mathcal{LRA}}$ functions are FIUC$^{\mathcal{LRA}}$ because polynomials can always be integrated exactly on sets of $\mathcal{LRA}$ literals [7].

We define a grammar to express P$^{\mathcal{LRA}}$ weight functions:

$$
\begin{aligned}
w \quad ::= \quad & c \mid x \mid -w \mid (w + w) \mid (w - w) \mid (w \cdot w) \mid \\
& [\![\text{If } \varphi \text{ Then } w \text{ Else } w]\!] \mid \\
& [\![\text{Case } \varphi : w;\ \varphi : w;\ \ldots]\!] \\
\chi \quad ::= \quad & \varphi
\end{aligned}
$$

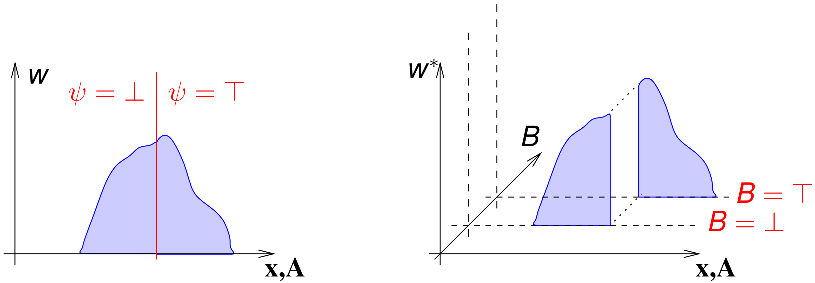
where c is a real value, x is a real variable, w is a P$^{\mathcal{LRA}}$ weight function, φ is an $\mathcal{LRA}$ formula.

Computing WMI with FIUC$^{\mathcal{LRA}}$ weight functions

Theorem

Let $w(\mathbf{x}, \mathbf{A})$, $\Psi \stackrel{\text{def}}{=} \{\psi_1, ..., \psi_K\}$ *and* χ *be as above. Let* $\mathbf{B} \stackrel{\text{def}}{=} \{B_1, ..., B_K\}$ *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* ψ_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:*

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



$\mathrm{WMI}(\varphi, w|\mathbf{x}, \mathbf{A})$ $\qquad$ $\mathrm{WMI}(\varphi \wedge (B \leftrightarrow \psi), w^*|\mathbf{x}, \mathbf{A} \cup \{B\})$

Computing WMI with FIUC$^{\mathcal{LRA}}$ weight functions - Example

Example

Let

- $\mathbf{A} = \emptyset$, $\mathbf{x} = \{x\}$
- $\chi \stackrel{\text{def}}{=} [\![x \in [-1,1]]\!]$,
- $\varphi \stackrel{\text{def}}{=} \top$,
- $\Psi \stackrel{\text{def}}{=} \{(x \geq 0)\}$,
- $w(x) \stackrel{\text{def}}{=} [\![\text{If } (x \geq 0) \text{ Then } x \text{ Else } -x]\!]$ (i.e., $w(x) \stackrel{\text{def}}{=} |x|$.)

Then $\mathsf{WMI}(\varphi, w|x, \emptyset) = \mathsf{WMI}_{\mathsf{nb}}(\varphi, w|\mathbf{x}) = \int_{[-1,1]} |x| \, \mathrm{d}x = 1$.

$\varphi^* = [\![x \in [-1,1]]\!] \wedge (B \leftrightarrow (x \geq 0))$ and $w^* = [\![\text{If } B \text{ Then } x \text{ Else } -x]\!]$.
Then $\mathsf{WMI}(\varphi^*, w^*|\mathbf{x}, \mathbf{B}) = 1$.
(See previous example, modulo reordering and variable renaming).

From WMI$_{old}$ to WMI and vice versa

From WMI$_{old}$ to WMI

We can easily express and compute WMI$_{old}$ as WMI by an equivalent FIUC$^{\mathcal{LRA}}$ weight function:

$$\mathsf{WMI}(\varphi, \prod_{\psi \in Atoms(\varphi)} [\![\mathsf{If}\ \psi\ \mathsf{Then}\ w(\psi)\ \mathsf{Else}\ w(\neg\psi)]\!]\ |\mathbf{x}, \mathbf{A}).$$

From WMI to WMI$_{old}$?

- AFAIK, there is no obvious general way to encode an arbitrary FIUC$^{\mathcal{LRA}}$ weight function into a WMI$_{old}$ one while always preventing an explosion in the size of its representation.
- Ex: $w(\mathbf{x}, \mathbf{A}) = \sum_{\mathbf{A}_j \in \mathbf{A}} [\![\mathsf{If}\ \mathbf{A}_j\ \mathsf{Then}\ w_{j1}(\mathbf{x})\ \mathsf{Else}\ w_{j2}(\mathbf{x})]\!]$.
- a trivial general solution:
  - *for every total truth assignment* $\mu \in \mathcal{TTA}(\varphi)$ add a fresh Boolean atom B_μ
  - $w(B_\mu) \stackrel{\text{def}}{=} w_\mu(\mathbf{x})$, $w(\neg B_\mu) \stackrel{\text{def}}{=} 1$, $w(l) \stackrel{\text{def}}{=} 1$ for every other literal l.

$\Longrightarrow$ blows up in size wrt. $||\mathbf{A}||$

From $\mathrm{WMI}_{\mathrm{old}}$ to WMI and vice versa

From $\mathrm{WMI}_{\mathrm{old}}$ to WMI

We can easily express and compute $\mathrm{WMI}_{\mathrm{old}}$ as WMI by an equivalent $\mathrm{FIUC}^{\mathcal{LRA}}$ weight function:

$$\mathrm{WMI}(\varphi, \prod_{\psi \in Atoms(\varphi)} [\![\text{If } \psi \text{ Then } w(\psi) \text{ Else } w(\neg\psi)]\!] \,|\mathbf{x}, \mathbf{A}).$$

From WMI to $\mathrm{WMI}_{\mathrm{old}}$?

- AFAIK, there is no obvious general way to encode an arbitrary $\mathrm{FIUC}^{\mathcal{LRA}}$ weight function into a $\mathrm{WMI}_{\mathrm{old}}$ one while always preventing an explosion in the size of its representation.
- Ex: $w(\mathbf{x}, \mathbf{A}) = \sum_{\mathbf{A}_j \in \mathbf{A}} [\![\text{If } \mathbf{A}_j \text{ Then } w_{j1}(\mathbf{x}) \text{ Else } w_{j2}(\mathbf{x})]\!]$.
- a trivial general solution:
  - *for every total truth assignment* $\mu \in \mathcal{TTA}(\varphi)$ add a fresh Boolean atom B_μ
  - $w(B_\mu) \stackrel{\text{def}}{=} w_\mu(\mathbf{x})$, $w(\neg B_\mu) \stackrel{\text{def}}{=} 1$, $w(l) \stackrel{\text{def}}{=} 1$ for every other literal l.

$\Longrightarrow$ blows up in size wrt. $||\mathbf{A}||$

Outline

Baseline

The Problem

Compute efficiently the WMI of a FIUC$^{\mathcal{LRA}}$ weight function $w(\mathbf{x}, \mathbf{A})$, with support formula χ and set of conditions $\Psi \stackrel{\text{def}}{=} \{\psi_1, ..., \psi_K\}$, over a formula $\varphi(\mathbf{x}, \mathbf{A})$.

Preprocessing

The problem is transformed into $\varphi^* \stackrel{\text{def}}{=} \varphi \wedge \chi \wedge \bigwedge_{k=1}^{K} (B_k \leftrightarrow \psi_k)$, $w^* \stackrel{\text{def}}{=} w[\mathbf{B} \leftarrow \Psi]$, and $\mathbf{A}^* \stackrel{\text{def}}{=} \mathbf{A} \cup \mathbf{B}$ by applying the Theorem: $\text{WMI}(\varphi \wedge \chi, w|\mathbf{x}, \mathbf{A}) = \text{WMI}(\varphi^*, w^*|\mathbf{x}, \mathbf{A} \cup \mathbf{B})$.

Baseline Procedure: WMI-AllSMT

- Based on the proposition: $\text{WMI}(\varphi^*, w^*|\mathbf{x}, \mathbf{A}^*) = \sum_{\mu^{\mathbf{A}^*} \wedge \mu^{\mathcal{LRA}} \in \mathcal{TTA}(\varphi^*)} \text{WMI}_{\text{nb}}(\mu^{\mathcal{LRA}}, w^*_{[\mu^{\mathbf{A}^*}]}|\mathbf{x})$.
- $\mathcal{TTA}(\varphi^*)$ is computed by AllSMT (e.g., in MathSAT5) without assignment-reduction
- $\text{WMI}_{\text{nb}}(\mu^{\mathcal{LRA}}, w^*_{[\mu^{\mathbf{A}^*}]}|\mathbf{x})$ is computed by invoking our background integration procedure for FI$^{\mathcal{LRA}}$ functions (e.g., by LattE Integrale [18])

Baseline

The Problem

Compute efficiently the WMI of a FIUC$^{\mathcal{LRA}}$ weight function $w(\mathbf{x}, \mathbf{A})$, with support formula χ and set of conditions $\Psi \stackrel{\text{def}}{=} \{\psi_1, ..., \psi_K\}$, over a formula $\varphi(\mathbf{x}, \mathbf{A})$.

Preprocessing

The problem is transformed into $\varphi^* \stackrel{\text{def}}{=} \varphi \wedge \chi \wedge \bigwedge_{k=1}^{K}(B_k \leftrightarrow \psi_k)$, $w^* \stackrel{\text{def}}{=} w[\mathbf{B} \leftarrow \Psi]$, and $\mathbf{A}^* \stackrel{\text{def}}{=} \mathbf{A} \cup \mathbf{B}$ by applying the Theorem: $\text{WMI}(\varphi \wedge \chi, w|\mathbf{x}, \mathbf{A}) = \text{WMI}(\varphi^*, w^*|\mathbf{x}, \mathbf{A} \cup \mathbf{B})$.

Baseline Procedure: WMI-AllSMT

- Based on the proposition: $\text{WMI}(\varphi^*, w^*|\mathbf{x}, \mathbf{A}^*) = \sum_{\mu^{\mathbf{A}^*} \wedge \mu^{\mathcal{LRA}} \in \mathcal{TTA}(\varphi^*)} \text{WMI}_{\text{nb}}(\mu^{\mathcal{LRA}}, w^*_{[\mu^{\mathbf{A}^*}]}|\mathbf{x})$.
- $\mathcal{TTA}(\varphi^*)$ is computed by AllSMT (e.g., in MathSAT5) without assignment-reduction
- $\text{WMI}_{\text{nb}}(\mu^{\mathcal{LRA}}, w^*_{[\mu^{\mathbf{A}^*}]}|\mathbf{x})$ is computed by invoking our background integration procedure for FI$^{\mathcal{LRA}}$ functions (e.g., by LattE Integrale [18])

Baseline

The Problem

Compute efficiently the WMI of a FIUC$^{\mathcal{LRA}}$ weight function $w(\mathbf{x}, \mathbf{A})$, with support formula χ and set of conditions $\Psi \stackrel{\text{def}}{=} \{\psi_1, ..., \psi_K\}$, over a formula $\varphi(\mathbf{x}, \mathbf{A})$.

Preprocessing

The problem is transformed into $\varphi^* \stackrel{\text{def}}{=} \varphi \wedge \chi \wedge \bigwedge_{k=1}^{K}(B_k \leftrightarrow \psi_k)$, $w^* \stackrel{\text{def}}{=} w[\mathbf{B} \leftarrow \Psi]$, and $\mathbf{A}^* \stackrel{\text{def}}{=} \mathbf{A} \cup \mathbf{B}$ by applying the Theorem: $\text{WMI}(\varphi \wedge \chi, w|\mathbf{x}, \mathbf{A}) = \text{WMI}(\varphi^*, w^*|\mathbf{x}, \mathbf{A} \cup \mathbf{B})$.

Baseline Procedure: WMI-AllSMT

- Based on the proposition: $\text{WMI}(\varphi^*, w^*|\mathbf{x}, \mathbf{A}^*) = \sum_{\mu^{\mathbf{A}^*} \wedge \mu^{\mathcal{LRA}} \in \mathcal{TTA}(\varphi^*)} \text{WMI}_{\text{nb}}(\mu^{\mathcal{LRA}}, w^*_{[\mu^{\mathbf{A}^*}]}|\mathbf{x})$.
- $\mathcal{TTA}(\varphi^*)$ is computed by AllSMT (e.g., in MATHSAT5) without assignment-reduction
- $\text{WMI}_{\text{nb}}(\mu^{\mathcal{LRA}}, w^*_{[\mu^{\mathbf{A}^*}]}|\mathbf{x})$ is computed by invoking our background integration procedure for FI$^{\mathcal{LRA}}$ functions (e.g., by LATTE INTEGRALE [18])

Efficient WMI procedure: WMI-PA

Efficient WMI procedure: WMI-PA

- Based on the propositions:

$$\begin{aligned} \mathsf{WMI}(\varphi^*, w^*|\mathbf{x}, \mathbf{A}^*) &= \sum_{\mu^{\mathbf{A}^*} \in \mathcal{TTA}(\exists \mathbf{x}.\varphi^*)} \mathsf{WMI}_{\mathsf{nb}}(\varphi^*_{[\mu^{\mathbf{A}^*}]}, w^*_{[\mu^{\mathbf{A}^*}]}|\mathbf{x}) \\ \mathsf{WMI}_{\mathsf{nb}}(\varphi^*_{[\mu^{\mathbf{A}^*}]}, w^*_{[\mu^{\mathbf{A}^*}]}|\mathbf{x}) &= \sum_{\mu^{\mathcal{LRA}} \in \mathcal{TA}(\varphi^*_{[\mu^{\mathbf{A}^*}]})} \mathsf{WMI}_{\mathsf{nb}}(\mu^{\mathcal{LRA}}, w^*_{[\mu^{\mathbf{A}^*}]}|\mathbf{x}). \end{aligned}$$

- $\mathcal{TTA}(\exists \mathbf{x}.\varphi^*)$ is computed by Predicate Abstraction $\mathcal{TTA}(\mathsf{PredAbs}_{[\varphi^*]}(\mathbf{A}^*))$ [17] (in MATHSAT5)
- $\mathsf{WMI}_{\mathsf{nb}}(\varphi^*_{[\mu^{\mathbf{A}^*}]}, w^*_{[\mu^{\mathbf{A}^*}]}|\mathbf{x})$ is computed by AllSMT with assignment-reduction [17] (in MATHSAT5)
- $\varphi^*_{[\mu^{\mathbf{A}^*}]}$ aggressively simplified before invoking $\mathcal{TA}()$ on it
  - reduces number of assignments in $\mathcal{TA}(\varphi^*_{[\mu^{\mathbf{A}^*}]})$
  - if $\varphi^*_{[\mu^{\mathbf{A}^*}]}$ reduced to a conjunction of $\mathcal{LRA}$-literals, then no need to invoke $\mathcal{TA}()$
- $\mathsf{WMI}_{\mathsf{nb}}(\mu^{\mathcal{LRA}}, w^*_{[\mu^{\mathbf{A}^*}]}|\mathbf{x})$ can exploit caching of integral values

Efficient WMI procedure: WMI-PA (cont.)

WMI-PA(φ, w, $\mathbf{x}$, $\mathbf{A}$)

$\langle \varphi^*, w^*, \mathbf{A}^* \rangle \leftarrow$ LabelConditions($\varphi, w, \mathbf{x}, \mathbf{A}$) // Apply Theorem
$\mathcal{M}^{\mathbf{A}^*} \leftarrow \mathcal{TTA}(\text{PredAbs}_{[\varphi^*]}(\mathbf{A}^*))$ // $\mathcal{TTA}(\exists \mathbf{x}.\varphi^*)$
$vol \leftarrow 0$
for $\mu^{\mathbf{A}^*} \in \mathcal{M}^{\mathbf{A}^*}$ **do**
  Simplify($\varphi^*_{[\mu^{\mathbf{A}^*}]}$) // remove as many $\mathcal{LRA}$-atoms as possible from $\varphi^*_{[\mu^{\mathbf{A}^*}]}$
  if (IsLiteralConjunction($\varphi^*_{[\mu^{\mathbf{A}^*}]}$)) **then**
    $vol \leftarrow vol + \text{WMI}_{\text{nb}}(\varphi^*_{[\mu^{\mathbf{A}^*}]}, w^*_{[\mu^{\mathbf{A}^*}]} | \mathbf{x})$
  else
    $\mathcal{M}^{\mathcal{LRA}} \leftarrow \mathcal{TA}(\text{PredAbs}_{[\varphi^*_{[\mu^{\mathbf{A}^*}]}]}(Atoms(\varphi^*_{[\mu^{\mathbf{A}^*}]})))$ // AllSMT with assignment-reduction
    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 vs. WMI-AllSMT

WMI-PA decouples the enumeration of the $\mu^{\mathbf{A}^*}$s from that of the $\mu^{\mathcal{LRA}}$s

- $\mathcal{TTA}(\exists \mathbf{x}.\varphi^*)$ removes *a priori* all the assignments $\mu^{\mathbf{A}^*}$ which cannot be expanded by any $\mathcal{LRA}$-satisfiable assignment $\mu^{\mathcal{LRA}}$ s.t. $\mu^{\mathbf{A}^*} \wedge \mu^{\mathcal{LRA}}$ propositionally satisfies φ^*
- *Atoms*$(\varphi^*_{[\mu^{\mathbf{A}^*}]})$ can be much smaller than *Atoms*(φ^*) by Simplify
  - (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.)

  $\Longrightarrow$ the number of assignments $\mu^{\mathbf{A}^*} \wedge \mu^{\mathcal{LRA}}$ can be drastically reduced
- search for a set $\mathcal{TA}(...)$ of partial assignments $\mu^{\mathcal{LRA}}$, each substituting $2^{(\ldots)}$ total ones

WMI-PA vs. WMI-ALLSMT: Example

Example

$$
\begin{aligned}
w(\mathbf{x},\mathbf{A}) &= [\![\text{If } (y \le 1) \text{ Then } f(x,y) \text{ Else } g(x,y)]\!]\\
\chi(\mathbf{x},\mathbf{A}) &\wedge (0 \le y) \wedge (y \le 2)\\
&\wedge (\ \ (y \le 1) \rightarrow ((0 \le x) \wedge (x \le 2)))\\
&\wedge (\neg(y \le 1) \rightarrow ((1 \le x) \wedge (x \le 3)))\\
\varphi(\mathbf{x},\mathbf{A}) &= \top
\end{aligned}
$$

Note: $f(x,y)$ defined on $x \in [0,2], y \in [0,1]$, $g(x,y)$ defined on $x \in [1,3], y \in [1,2]$

After labelling

$$
\begin{aligned}
w^*(\mathbf{x},\mathbf{A}^*) &= [\![\text{If } B_1 \text{ Then } f(x,y) \text{ Else } g(x,y)]\!]\\
\varphi^*(\mathbf{x},\mathbf{A}^*) &= (B_1 \leftrightarrow (y \le 1))\\
&\wedge (0 \le y) \wedge (y \le 2)\\
&\wedge ((y \le 1) \rightarrow ((0 \le x) \wedge (x \le 2)))\\
&\wedge (\neg(y \le 1) \rightarrow ((1 \le x) \wedge (x \le 3)))
\end{aligned}
$$

WMI-PA vs. WMI-ALLSMT: Example

Example

$$
\begin{aligned}
w(\mathbf{x}, \mathbf{A}) &= [\![\text{If } (y \le 1) \text{ Then } f(x,y) \text{ Else } g(x,y)]\!] \\
\chi(\mathbf{x}, \mathbf{A}) &\wedge (0 \le y) \wedge (y \le 2) \\
&\wedge (\quad (y \le 1) \rightarrow ((0 \le x) \wedge (x \le 2))) \\
&\wedge (\neg(y \le 1) \rightarrow ((1 \le x) \wedge (x \le 3))) \\
\varphi(\mathbf{x}, \mathbf{A}) &= \top
\end{aligned}
$$

Note: $f(x,y)$ defined on $x \in [0,2], y \in [0,1]$, $g(x,y)$ defined on $x \in [1,3], y \in [1,2]$

After labelling

$$
\begin{aligned}
w^*(\mathbf{x}, \mathbf{A}^*) &= [\![\text{If } B_1 \text{ Then } f(x,y) \text{ Else } g(x,y)]\!] \\
\varphi^*(\mathbf{x}, \mathbf{A}^*) &= (B_1 \leftrightarrow (y \le 1)) \\
&\wedge (0 \le y) \wedge (y \le 2) \\
&\wedge ((y \le 1) \rightarrow ((0 \le x) \wedge (x \le 2))) \\
&\wedge (\neg(y \le 1) \rightarrow ((1 \le x) \wedge (x \le 3)))
\end{aligned}
$$

WMI-PA vs. WMI-AllSMT: Example (cont.)

After labelling

$$w^*(\mathbf{x}, \mathbf{A}^*) = [\![\text{If } B_1 \text{ Then } f(x,y) \text{ Else } g(x,y)]\!]$$

$$\begin{aligned}\varphi^*(\mathbf{x}, \mathbf{A}^*) = & (B_1 \leftrightarrow (y \le 1)) \wedge (0 \le y) \wedge (y \le 2) \\ & \wedge ((y \le 1) \rightarrow ((0 \le x) \wedge (x \le 2))) \\ & \wedge (\neg(y \le 1) \rightarrow ((1 \le x) \wedge (x \le 3)))\end{aligned}$$

WMI-AllSMT

- With WMI-AllSMT, the integration on 4 total truth assignments is needed:

$$\left\{\begin{array}{llllllll} \{ & B_1, & (0 \le y), & (y \le 2), & (y \le 1), & (0 \le x), & (x \le 2), & (1 \le x), & (x \le 3)\} \\ \{ & B_1, & (0 \le y), & (y \le 2), & (y \le 1), & (0 \le x), & (x \le 2), & \neg(1 \le x), & (x \le 3)\} \\ \{ & \neg B_1, & (0 \le y), & (y \le 2), & \neg(y \le 1), & (0 \le x), & (x \le 2), & (1 \le x), & (x \le 3)\} \\ \{ & \neg B_1, & (0 \le y), & (y \le 2), & \neg(y \le 1), & (0 \le x), & \neg(x \le 2), & (1 \le x), & (x \le 3)\} \end{array}\right\}$$

$$\int_0^1\int_0^1 f(x,y)\,dx\,dy + \int_0^1\int_1^2 f(x,y)\,dx\,dy + \int_1^2\int_1^2 g(x,y)\,dx\,dy + \int_1^2\int_2^3 g(x,y)\,dx\,dy$$

- two useless partitions: on $(1 \le x)$ and on $(x \le 2)$

WMI-PA vs. WMI-ALLSMT: Example (cont.)

After labelling

$$w^*(\mathbf{x}, \mathbf{A}^*) = [\![\text{If } B_1 \text{ Then } f(x,y) \text{ Else } g(x,y)]\!]$$

$$\begin{aligned}\varphi^*(\mathbf{x}, \mathbf{A}^*) = & (B_1 \leftrightarrow (y \leq 1)) \wedge (0 \leq y) \wedge (y \leq 2) \\ & \wedge ((y \leq 1) \rightarrow ((0 \leq x) \wedge (x \leq 2))) \\ & \wedge (\neg(y \leq 1) \rightarrow ((1 \leq x) \wedge (x \leq 3)))\end{aligned}$$

WMI-ALLSMT

- With WMI-ALLSMT, the integration on 4 total truth assignments is needed:

$$\left\{\begin{array}{l} \{\ B_1, (0 \leq y), (y \leq 2), (y \leq 1), (0 \leq x), (x \leq 2), (1 \leq x), (x \leq 3)\} \\ \{\ B_1, (0 \leq y), (y \leq 2), (y \leq 1), (0 \leq x), (x \leq 2), \neg(1 \leq x), (x \leq 3)\} \\ \{\neg B_1, (0 \leq y), (y \leq 2), \neg(y \leq 1), (0 \leq x), (x \leq 2), (1 \leq x), (x \leq 3)\} \\ \{\neg B_1, (0 \leq y), (y \leq 2), \neg(y \leq 1), (0 \leq x), \neg(x \leq 2), (1 \leq x), (x \leq 3)\} \end{array}\right\}$$

$$\int_0^1 \int_0^1 f(x,y)\,dx\,dy + \int_0^1 \int_1^2 f(x,y)\,dx\,dy + \int_1^2 \int_1^2 g(x,y)\,dx\,dy + \int_1^2 \int_2^3 g(x,y)\,dx\,dy$$

- two useless partitions: on $(1 \leq x)$ and on $(x \leq 2)$

WMI-PA vs. WMI-AllSMT: Example (cont.)

After labelling

$$w^*(\mathbf{x}, \mathbf{A}^*) = [\![\text{If } B_1 \text{ Then } f(x,y) \text{ Else } g(x,y)]\!]$$

$$\begin{aligned}\varphi^*(\mathbf{x}, \mathbf{A}^*) = {} & (B_1 \leftrightarrow (y \leq 1)) \land (0 \leq y) \land (y \leq 2) \\ & \land ((y \leq 1) \rightarrow ((0 \leq x) \land (x \leq 2))) \\ & \land (\neg(y \leq 1) \rightarrow ((1 \leq x) \land (x \leq 3)))\end{aligned}$$

WMI-AllSMT

- With WMI-AllSMT, the integration on 4 total truth assignments is needed:

$$\left\{\begin{array}{llllllll} \{ B_1, & (0 \leq y), & (y \leq 2), & (y \leq 1), & (0 \leq x), & (x \leq 2), & (1 \leq x), & (x \leq 3)\} \\ \{ B_1, & (0 \leq y), & (y \leq 2), & (y \leq 1), & (0 \leq x), & (x \leq 2), & \neg(1 \leq x), & (x \leq 3)\} \\ \{\neg B_1, & (0 \leq y), & (y \leq 2), & \neg(y \leq 1), & (0 \leq x), & (x \leq 2), & (1 \leq x), & (x \leq 3)\} \\ \{\neg B_1, & (0 \leq y), & (y \leq 2), & \neg(y \leq 1), & (0 \leq x), & \neg(x \leq 2), & (1 \leq x), & (x \leq 3)\} \end{array}\right\}$$

$$\int_0^1 \int_0^1 f(x,y)\, dx\, dy + \int_0^1 \int_1^2 f(x,y)\, dx\, dy + \int_1^2 \int_1^2 g(x,y)\, dx\, dy + \int_1^2 \int_2^3 g(x,y)\, dx\, dy$$

- two useless partitions: on $(1 \leq x)$ and on $(x \leq 2)$

WMI-PA vs. WMI-ALLSMT: Example (cont.)

After labelling

$$w^*(\mathbf{x}, \mathbf{A}^*) = [\![\text{If } B_1 \text{ Then } f(x,y) \text{ Else } g(x,y)]\!]$$

$$\begin{aligned}\varphi^*(\mathbf{x}, \mathbf{A}^*) = & (B_1 \leftrightarrow (y \leq 1)) \wedge (0 \leq y) \wedge (y \leq 2) \\ & \wedge ((y \leq 1) \rightarrow ((0 \leq x) \wedge (x \leq 2))) \\ & \wedge (\neg(y \leq 1) \rightarrow ((1 \leq x) \wedge (x \leq 3)))\end{aligned}$$

WMI-ALLSMT

- With WMI-ALLSMT, the integration on 4 total truth assignments is needed:

$$\left\{\begin{array}{l}\{\ B_1, (0 \leq y), (y \leq 2), (y \leq 1), (0 \leq x), (x \leq 2), (1 \leq x), (x \leq 3)\} \\ \{\ B_1, (0 \leq y), (y \leq 2), (y \leq 1), (0 \leq x), (x \leq 2), \neg(1 \leq x), (x \leq 3)\} \\ \{\neg B_1, (0 \leq y), (y \leq 2), \neg(y \leq 1), (0 \leq x), (x \leq 2), (1 \leq x), (x \leq 3)\} \\ \{\neg B_1, (0 \leq y), (y \leq 2), \neg(y \leq 1), (0 \leq x), \neg(x \leq 2), (1 \leq x), (x \leq 3)\}\end{array}\right\}$$

$$\int_0^1 \int_0^1 f(x,y)\, dx\, dy + \int_0^1 \int_1^2 f(x,y)\, dx\, dy + \int_1^2 \int_1^2 g(x,y)\, dx\, dy + \int_1^2 \int_2^3 g(x,y)\, dx\, dy$$

- two useless partitions: on $(1 \leq x)$ and on $(x \leq 2)$

WMI-PA vs. WMI-ALLSMT: Example (cont.)

After labelling

$$w^*(\mathbf{x}, \mathbf{A}^*) = [\![\text{If } B_1 \text{ Then } f(x,y) \text{ Else } g(x,y)]\!]$$

$$\begin{aligned}\varphi^*(\mathbf{x}, \mathbf{A}^*) = & (B_1 \leftrightarrow (y \leq 1)) \wedge (0 \leq y) \wedge (y \leq 2) \\ & \wedge ((y \leq 1) \rightarrow ((0 \leq x) \wedge (x \leq 2))) \\ & \wedge (\neg(y \leq 1) \rightarrow ((1 \leq x) \wedge (x \leq 3)))\end{aligned}$$

WMI-ALLSMT

- With WMI-ALLSMT, the integration on 4 total truth assignments is needed:

$$\left\{\begin{array}{l} \{\ B_1,\ (0 \leq y),\ (y \leq 2),\ (y \leq 1),\ (0 \leq x),\ (x \leq 2),\ (1 \leq x),\ (x \leq 3)\} \\ \{\ B_1,\ (0 \leq y),\ (y \leq 2),\ (y \leq 1),\ (0 \leq x),\ (x \leq 2), \neg(1 \leq x),\ (x \leq 3)\} \\ \{\neg B_1,\ (0 \leq y),\ (y \leq 2), \neg(y \leq 1),\ (0 \leq x),\ (x \leq 2),\ (1 \leq x),\ (x \leq 3)\} \\ \{\neg B_1,\ (0 \leq y),\ (y \leq 2), \neg(y \leq 1),\ (0 \leq x), \neg(x \leq 2),\ (1 \leq x),\ (x \leq 3)\} \end{array}\right\}$$

$$\int_0^1 \int_0^1 f(x,y)\, dx\, dy + \int_0^1 \int_1^2 f(x,y)\, dx\, dy + \int_1^2 \int_1^2 g(x,y)\, dx\, dy + \int_1^2 \int_2^3 g(x,y)\, dx\, dy$$

- two useless partitions: on $(1 \leq x)$ and on $(x \leq 2)$

WMI-PA vs. WMI-ALLSMT: Example (cont.)

After labelling

$$w^*(\mathbf{x}, \mathbf{A}^*) = [\![\text{If } B_1 \text{ Then } f(x,y) \text{ Else } g(x,y)]\!]$$

$$\begin{aligned}\varphi^*(\mathbf{x}, \mathbf{A}^*) = & (B_1 \leftrightarrow (y \leq 1)) \wedge (0 \leq y) \wedge (y \leq 2) \\ & \wedge ((y \leq 1) \rightarrow ((0 \leq x) \wedge (x \leq 2))) \\ & \wedge (\neg(y \leq 1) \rightarrow ((1 \leq x) \wedge (x \leq 3)))\end{aligned}$$

WMI-ALLSMT

- With WMI-ALLSMT, the integration on 4 total truth assignments is needed:

$$\left\{\begin{array}{l} \{\ B_1, (0 \leq y), (y \leq 2), (y \leq 1), (0 \leq x), (x \leq 2), (1 \leq x), (x \leq 3)\} \\ \{\ B_1, (0 \leq y), (y \leq 2), (y \leq 1), (0 \leq x), (x \leq 2), \neg(1 \leq x), (x \leq 3)\} \\ \{\neg B_1, (0 \leq y), (y \leq 2), \neg(y \leq 1), (0 \leq x), (x \leq 2), (1 \leq x), (x \leq 3)\} \\ \{\neg B_1, (0 \leq y), (y \leq 2), \neg(y \leq 1), (0 \leq x), \neg(x \leq 2), (1 \leq x), (x \leq 3)\} \end{array}\right\}$$

$$\int_0^1 \int_0^1 f(x,y)\,\mathrm{d}x\,\mathrm{d}y + \int_0^1 \int_1^2 f(x,y)\,\mathrm{d}x\,\mathrm{d}y + \int_1^2 \int_1^2 g(x,y)\,\mathrm{d}x\,\mathrm{d}y + \int_1^2 \int_2^3 g(x,y)\,\mathrm{d}x\,\mathrm{d}y$$

- two useless partitions: on $(1 \leq x)$ and on $(x \leq 2)$

WMI-PA vs. WMI-ALLSMT: Example (cont.)

After labelling

$$w^*(\mathbf{x}, \mathbf{A}^*) = [\![\text{If } B_1 \text{ Then } f(x,y) \text{ Else } g(x,y)]\!]$$

$$\begin{aligned}\varphi^*(\mathbf{x}, \mathbf{A}^*) = & (B_1 \leftrightarrow (y \le 1)) \wedge (0 \le y) \wedge (y \le 2) \\ & \wedge ((y \le 1) \rightarrow ((0 \le x) \wedge (x \le 2))) \\ & \wedge (\neg(y \le 1) \rightarrow ((1 \le x) \wedge (x \le 3)))\end{aligned}$$

WMI-PA Computation

$$\mathcal{M}^{\mathbf{A}^*} = \{\overbrace{\{B_1\}}^{\mu_1}, \{\neg B_1\}\}$$

$$w^*_{[\mu_1]}(\mathbf{x}, \mathbf{A}^*) = f(x,y)$$

$$\begin{aligned}\varphi^*_{[\mu_1]}(\mathbf{x}, \mathbf{A}^*) = & (\top \leftrightarrow (y \le 1)) \wedge (0 \le y) \wedge (y \le 2) \\ & \wedge ((y \le 1) \rightarrow ((0 \le x) \wedge (x \le 2))) \\ & \wedge (\neg(y \le 1) \rightarrow ((1 \le x) \wedge (x \le 3)))\end{aligned}$$

$$\text{Simplify}(\varphi^*_{[\mu_1]}) = (y \le 1) \wedge (0 \le y) \wedge (y \le 2) \wedge ((0 \le x) \wedge (x \le 2))$$

$$\int_{\varphi^*_{[\mu_1]}} w^*_{[\mu_1]} \, d\mathbf{x} = \int_0^1 \int_0^2 f(x,y) \, dx \, dy$$

WMI-PA vs. WMI-ALLSMT: Example (cont.)

After labelling

$$w^*(\mathbf{x}, \mathbf{A}^*) = [\![\text{If } B_1 \text{ Then } f(x,y) \text{ Else } g(x,y)]\!]$$

$$\begin{aligned}\varphi^*(\mathbf{x}, \mathbf{A}^*) = & (B_1 \leftrightarrow (y \leq 1)) \wedge (0 \leq y) \wedge (y \leq 2) \\ & \wedge ((y \leq 1) \rightarrow ((0 \leq x) \wedge (x \leq 2))) \\ & \wedge (\neg(y \leq 1) \rightarrow ((1 \leq x) \wedge (x \leq 3)))\end{aligned}$$

WMI-PA Computation

$$\mathcal{M}^{\mathbf{A}^*} = \{\{B_1\}, \overbrace{\{\neg B_1\}}^{\mu_2}\}$$

$$w^*_{[\mu_2]}(\mathbf{x}, \mathbf{A}^*) = g(x,y)$$

$$\begin{aligned}\varphi^*_{[\mu_2]}(\mathbf{x}, \mathbf{A}^*) = & (\bot \leftrightarrow (y \leq 1)) \wedge (0 \leq y) \wedge (y \leq 2) \\ & \wedge ((y \leq 1) \rightarrow ((0 \leq x) \wedge (x \leq 2))) \\ & \wedge (\neg(y \leq 1) \rightarrow ((1 \leq x) \wedge (x \leq 3)))\end{aligned}$$

$$\text{Simplify}(\varphi^*_{[\mu_2]}) = \neg(y \leq 1) \wedge (0 \leq y) \wedge (y \leq 2) \wedge ((1 \leq x) \wedge (x \leq 3))$$

$$\int_{\varphi^*_{[\mu_2]}} w^*_{[\mu_2]} \, d\mathbf{x} = \int_1^2 \int_1^3 g(x,y) \, dx \, dy$$

Outline

Case Study 1: The Road Network Problem, Fixed Path

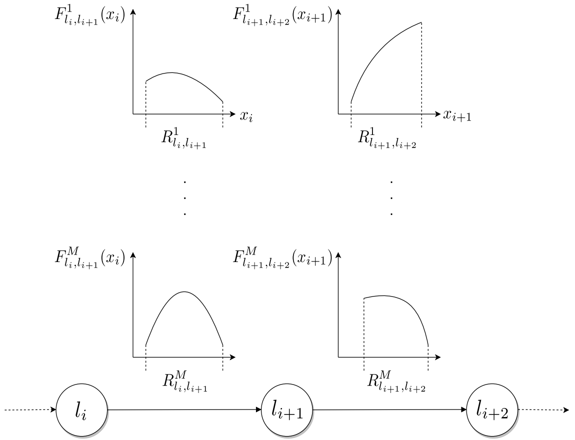
Given :

- a path of $N+1$ consecutive adjacent locations $\{l_0, ..., l_N\}$ in a road network (implicit)
- (the part of interest of) the day partitioned into disjoint consecutive intervals $\{I^1, ..., I^M\}$
- for each pair $\langle l_i, l_j\rangle$ of adjacent locations, for each time interval $I^m \stackrel{\text{def}}{=} [c_m, c_{m+1})$, the distribution of the journey time from l_i to l_j at any time $t \in I^m$:
  - $f^m_{l_i,l_j} : \mathbb{R} \mapsto \mathbb{R}^+$ is such distribution
  - $R^m_{l_i,l_j} \stackrel{\text{def}}{=} [a^m_{l_i,l_j}, b^m_{l_i,l_j})$ is its support.

  Note: the time slots I^ms are disjoint, the supports $R^m_{l_i,l_j}$s are not disjoint.
- two values: departure time t_{dep} and maximum arrival time t_{arr}
- variables: for $n \in [0...N]$,
  - x_n the journey time between l_{n-1} and l_n,
  - t_n is the time at step n, i.e., $t_n \stackrel{\text{def}}{=} t_0 + \sum_{i=1}^{n-1} x_i$

Query: $P(t_N \leq t_{\text{arr}} \mid t_0 = t_{\text{dep}}, \{l_i\}_{i=0}^N)$. (The locations $\{l_i\}_{i=0}^N$ are left implicit.)

Case Study 1: The Road Network Problem, Fixed Path (cont.)



Journey time densities for a pair of consecutive time steps, from location l_i to l_{i+2}.
Each edge shows the corresponding journey time distribution for each of the intervals.

The Road Network Problem, Fixed Path: Encoding

Let $\mathbf{x} \stackrel{\text{def}}{=} \{x_1, ..., x_N\}$, $\mathbf{A} \stackrel{\text{def}}{=} \emptyset$, and "$t_n$" be a shortcut for the term "$\sum_{i=1}^{n} x_i + t_0$". Then:

$$w(\mathbf{x}) \stackrel{\text{def}}{=} \prod_{n=1}^{N} \left[\!\!\left[\text{Case } [\![t_{n-1} \in I^1]\!] : f^1_{l_{n-1},l_n}(x_n); \; \ldots \; [\![t_{n-1} \in I^M]\!] : f^M_{l_{n-1},l_n}(x_n)\right]\!\!\right]$$

$$\chi(\mathbf{x}) \stackrel{\text{def}}{=} \bigwedge_{n=0}^{N} [\![t_n \in \bigcup_{m=1}^{M} I^m]\!] \wedge \bigwedge_{n=1}^{N} \bigwedge_{m=1}^{M} ([\![t_{n-1} \in I^m]\!] \rightarrow [\![x_n \in R^m_{l_{n-1},l_n}]\!])$$

$$\varphi(\mathbf{x}) \stackrel{\text{def}}{=} \top$$

$$P(t_N \leq t_{\text{arr}} \mid t_0 = t_{\text{dep}}, \{l_i\}_{i=0}^{N}) = \frac{\mathsf{WMI}_{\text{nb}}(\chi(\mathbf{x}) \wedge (t_N \leq t_{\text{arr}}) \wedge (t_0 = t_{\text{dep}}), w(\mathbf{x}) | \mathbf{x})}{\mathsf{WMI}_{\text{nb}}(\chi(\mathbf{x}) \wedge (t_0 = t_{\text{dep}}), w(\mathbf{x}) | \mathbf{x})}$$

If each $f^m_{l_i,l_j}(x)$ is polynomial in $x \in R^m_{l_i,l_j}$, then $w(\mathbf{x})$ is $\mathsf{P}^{\mathcal{LRA}}$ and hence $\mathsf{FIUC}^{\mathcal{LRA}}$. Thus we can apply the theorem:

$$\varphi^*(\mathbf{x}, \mathbf{B}) \stackrel{\text{def}}{=} \varphi(\mathbf{x}) \wedge \chi(\mathbf{x}) \wedge \bigwedge_{n=1}^{N} \bigwedge_{m=1}^{M} (B^m_{n-1} \leftrightarrow [\![t_{n-1} \in I^m]\!])$$

$$w^*(\mathbf{x}, \mathbf{B}) \stackrel{\text{def}}{=} \prod_{n=1}^{N} \left[\!\!\left[\text{Case } B^1_{n-1} : f^1_{l_{n-1},l_n}(x_n); \ldots B^M_{n-1} : f^M_{l_{n-1},l_n}(x_n)\right]\!\!\right].$$

The Road Network Problem, Fixed Path: Encoding

Let $\mathbf{x} \stackrel{\text{def}}{=} \{x_1, ..., x_N\}$, $\mathbf{A} \stackrel{\text{def}}{=} \emptyset$, and "$t_n$" be a shortcut for the term "$\sum_{i=1}^{n} x_i + t_0$". Then:

$$
\begin{aligned}
w(\mathbf{x}) &\stackrel{\text{def}}{=} \prod_{n=1}^{N} \left[\!\left[\text{Case } [\![t_{n-1} \in I^1]\!] : f^1_{l_{n-1},l_n}(x_n);\ \ldots\ [\![t_{n-1} \in I^M]\!] : f^M_{l_{n-1},l_n}(x_n) \right]\!\right] \\
\chi(\mathbf{x}) &\stackrel{\text{def}}{=} \bigwedge_{n=0}^{N} [\![t_n \in \bigcup_{m=1}^{M} I^m]\!] \\
&\wedge \bigwedge_{n=1}^{N} \bigwedge_{m=1}^{M} ([\![t_{n-1} \in I^m]\!] \rightarrow [\![x_n \in R^m_{l_{n-1},l_n}]\!]) \\
\varphi(\mathbf{x}) &\stackrel{\text{def}}{=} \top
\end{aligned}
$$

$$
P(t_N \leq t_{arr} \mid t_0 = t_{dep}, \{l_i\}_{i=0}^{N}) = \frac{\mathrm{WMI}_{\mathrm{nb}}(\chi(\mathbf{x}) \wedge (t_N \leq t_{arr}) \wedge (t_0 = t_{dep}), w(\mathbf{x})|\mathbf{x})}{\mathrm{WMI}_{\mathrm{nb}}(\chi(\mathbf{x}) \wedge (t_0 = t_{dep}), w(\mathbf{x})|\mathbf{x})}
$$

If each $f^m_{l_i,l_j}(x)$ is polynomial in $x \in R^m_{l_i,l_j}$, then $w(\mathbf{x})$ is $\mathrm{P}^{\mathcal{LRA}}$ and hence $\mathrm{FIUC}^{\mathcal{LRA}}$. Thus we can apply the theorem:

$$
\varphi^*(\mathbf{x}, \mathbf{B}) \stackrel{\text{def}}{=} \varphi(\mathbf{x}) \wedge \chi(\mathbf{x}) \wedge \bigwedge_{n=1}^{N} \bigwedge_{m=1}^{M} (B^m_{n-1} \leftrightarrow [\![t_{n-1} \in I^m]\!])
$$

$$
w^*(\mathbf{x}, \mathbf{B}) \stackrel{\text{def}}{=} \prod_{n=1}^{N} \left[\!\left[\text{Case } B^1_{n-1} : f^1_{l_{n-1},l_n}(x_n); \ldots\ B^M_{n-1} : f^M_{l_{n-1},l_n}(x_n) \right]\!\right].
$$

The Road Network Problem, Fixed Path: Example

Example

$$\begin{aligned}\chi(\mathbf{x}) \stackrel{\text{def}}{=} & [\![t_0 \in [7,10)]\!] \\ & \wedge [\![t_0 + x_1 \in [7,10)]\!] \\ & \wedge [\![t_0 \in [7,8)]\!] \rightarrow [\![x_1 \in [0.5,1)]\!] \\ & \wedge [\![t_0 \in [8,9)]\!] \rightarrow [\![x_1 \in [1,1.5)]\!] \\ & \wedge [\![t_0 \in [9,10)]\!] \rightarrow [\![x_1 \in [1,2)]\!] \\ & \wedge [\![t_0 + x_1 \in [7,8)]\!] \rightarrow [\![x_2 \in [1,1.5)]\!] \\ & \wedge [\![t_0 + x_1 \in [8,9)]\!] \rightarrow [\![x_2 \in [1.5,2)]\!] \\ & \wedge [\![t_0 + x_1 \in [9,10)]\!] \rightarrow [\![x_2 \in [1,2)]\!]\end{aligned}$$

$$w(\mathbf{x}) \stackrel{\text{def}}{=} \left[\!\!\left[\begin{array}{ll} \text{Case} & \\ \quad [\![t_0 \in [7,8)]\!] & : w^1_{[l_0 l_1]}(x_1); \\ \quad [\![t_0 \in [8,9)]\!] & : w^2_{[l_0 l_1]}(x_1); \\ \quad [\![t_0 \in [9,10)]\!] & : w^3_{[l_0 l_1]}(x_1); \end{array} \right]\!\!\right] \cdot \left[\!\!\left[\begin{array}{ll} \text{Case} & \\ \quad [\![t_0 + x_1 \in [7,8)]\!] & : w^1_{[l_1 l_2]}(x_2); \\ \quad [\![t_0 + x_1 \in [8,9)]\!] & : w^2_{[l_1 l_2]}(x_2); \\ \quad [\![t_0 + x_1 \in [9,10)]\!] & : w^3_{[l_1 l_2]}(x_2); \end{array} \right]\!\!\right]$$

$$\varphi(\mathbf{x}) \stackrel{\text{def}}{=} \top$$

where the $w^m_{[l_{n-1} l_n]}(x_n)$ are functions which are integrable and positive in their respective domain stated in $\chi(\mathbf{x})$ (e.g., $w^1_{[l_0 l_1]}(x_1)$ is integrable and positive in $[\![x_1 \in [0.5,1)]\!]$).

The Road Network Problem, Fixed Path: Example (cont.)

Example

Then, by applying the theorem:

$$\begin{aligned}
\varphi^*(\mathbf{x}, \mathbf{B}) &\stackrel{\text{def}}{=} \varphi(\mathbf{x}) \wedge \chi(\mathbf{x}) \\
&\wedge \quad (B_0^1 \leftrightarrow [\![t_0 \in [7, 8)]\!]) \\
&\wedge \quad \ldots \\
&\wedge \quad (B_1^3 \leftrightarrow [\![t_0 + x_1 \in [9, 10)]\!])
\end{aligned}$$

$$w^*(\mathbf{x}, \mathbf{B}) \stackrel{\text{def}}{=} \left[\!\!\left[\begin{array}{ll} \text{Case} & \\ B_0^1 & : w^1_{[l_0 l_1]}(x_1); \\ B_0^2 & : w^2_{[l_0 l_1]}(x_1); \\ B_0^3 & : w^3_{[l_0 l_1]}(x_1); \end{array} \right]\!\!\right] \cdot \left[\!\!\left[\begin{array}{ll} \text{Case} & \\ B_1^1 & : w^1_{[l_1 l_2]}(x_2); \\ B_1^2 & : w^2_{[l_1 l_2]}(x_2); \\ B_1^3 & : w^3_{[l_1 l_2]}(x_2); \end{array} \right]\!\!\right]$$

Case Study 2: The Road Network Problem under Conditional Plan

Given:

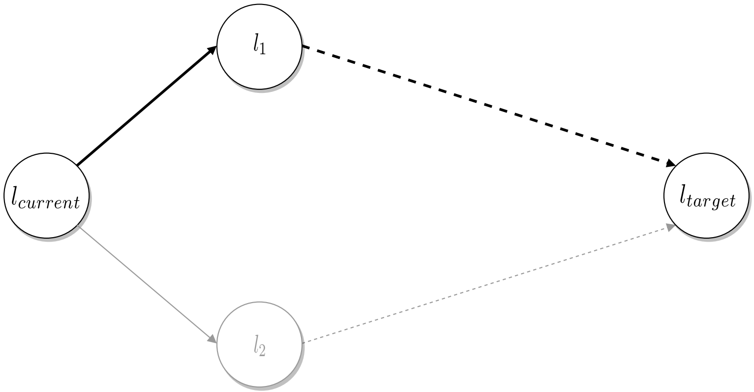
- Time intervals I^ms, variables x_ns and t_ns, values t_{dep} and t_{arr}, distributions of journey times $f^m_{l_i,l_j}$s and supports $R^m_{l_i,l_j}$ (for all $\langle l_i, l_j \rangle$ s in the network): as with the fixed-path case.
- The path in the road network is not given in advance. Instead it is given:
  - a maximum path length N
  - an initial location l_{dep} and final target location l_{target}
  - a conditional plan, s.t., for any current location l and time interval index m, $\text{next}(l, m, l_{target})$ is the next location in the path (mimics empirical knowledge of the driver)

(for l_{target}, $\text{next}(l_{target}, m, l_{target}) \stackrel{\text{def}}{=} l_{target}$ and $R^m_{l_{target}, l_{target}} \stackrel{\text{def}}{=} [0, 0]$)

Query: $P(t_N \leq t_{arr} \mid t_0 = t_{dep}, l_{dep}, l_{target}, \text{next})$.

Case Study 2: The Road Network Problem under Cond. Plan (cont.)

k	$next(l_{current}, k, l_{target})$
1	l_2
m	l_1
M	l_2



Two alternative (sub)paths from l_{curr} to l_{target}.
The successor of l_{curr} is selected according to the time interval at which the node is reached.

The Road Network Problem with Conditional Plan: Encoding

Let $\mathbf{x} \stackrel{\text{def}}{=} \{x_1, ..., x_N\}$, $\mathbf{A} \stackrel{\text{def}}{=} \{A_{01}, ..., A_{NL}\}$, and "$t_n$" be a shortcut for the term "$\sum_{i=1}^{n} x_i + t_0$".

$$\chi(\mathbf{x}, \mathbf{A}) \stackrel{\text{def}}{=} \bigwedge_{n=0}^{N} [\![t_n \in \bigcup_{m=1}^{M} I^m]\!] \wedge \bigwedge_{n=0}^{N} [\![\text{OneOf}\{A_{n,l} \mid l \in [1, L]\}]\!]$$
$$\wedge \bigwedge_{n=1}^{N} \left(\bigwedge_{l=1}^{L} \left(A_{n-1,l} \rightarrow \bigwedge_{m=1}^{M} ([\![t_{n-1} \in I^m]\!] \rightarrow [\![x_n \in R^m_{l,\text{next}(l,m,l_{\text{target}})}]\!]) \right) \right),$$

$$\varphi(\mathbf{x}, \mathbf{A}) \stackrel{\text{def}}{=} A_{0,l_0} \wedge \bigwedge_{n=1}^{N} \left(\bigwedge_{l=1}^{L} \left(A_{n-1,l} \rightarrow \bigwedge_{m=1}^{M} ([\![t_{n-1} \in I^m]\!] \rightarrow A_{n,\text{next}(l,m,l_{\text{target}})}) \right) \right)$$

$$w(\mathbf{x}, \mathbf{A}) \stackrel{\text{def}}{=} \prod_{n=1}^{N} \left[\!\!\left[\begin{array}{l} \text{Case} \\ \quad (A_{n-1,l_1} \wedge A_{n,l_2}) : \\ \quad\quad [\![\text{Case} \;\; [\![t_{n-1} \in I^1]\!] : f^1_{l_1,l_2}(x_n); \; \ldots \; ; \; [\![t_{n-1} \in I^M]\!] : f^M_{l_1,l_2}(x_n) \;]\!]; \\ \quad (A_{n-1,l_1} \wedge A_{n,l_3}) : \\ \quad\quad [\![\text{Case} \;\; [\![t_{n-1} \in I^1]\!] : f^1_{l_1,l_3}(x_n); \; \ldots \; ; \; [\![t_{n-1} \in I^M]\!] : f^M_{l_1,l_3}(x_n) \;]\!]; \\ \quad \ldots \\ \quad (A_{n-1,l_L} \wedge A_{n,l_{L-1}}) : \\ \quad\quad [\![\text{Case} \;\; [\![t_{n-1} \in I^1]\!] : f^1_{l_L,l_{L-1}}(x_n); \; \ldots \; ; \; [\![t_{n-1} \in I^M]\!] : f^M_{l_L,l_{L-1}}(x_n) \;]\!]; \end{array} \right]\!\!\right]$$

Note: in $w(\mathbf{x}, \mathbf{A})$ the case "$A_{n-1,l_i} \wedge A_{n,l_j}$" is considered only if $\langle l_i, l_j \rangle$ adjacent and $l_j = \text{next}(l_i, m, l_{\text{target}})$ for some m.

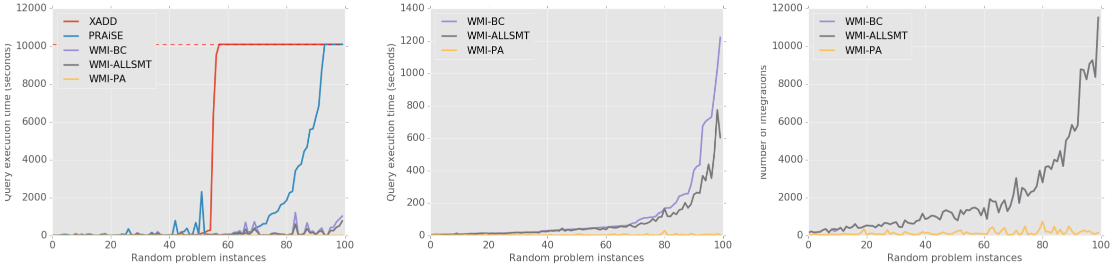
$$P(t_N \leq t_{\text{arr}} \mid t_0 = t_{\text{dep}}, l_{\text{dep}}, l_{\text{target}}, \text{next}) = \frac{\text{WMI}(\varphi(\mathbf{x}, \mathbf{A}) \wedge \chi(\mathbf{x}, \mathbf{A}) \wedge (t_N \leq t_{\text{arr}}) \wedge (t_0 = t_{\text{dep}}), w(\mathbf{x}, \mathbf{A}) | \mathbf{x}, \mathbf{A})}{\text{WMI}(\varphi(\mathbf{x}, \mathbf{A}) \wedge \chi(\mathbf{x}, \mathbf{A}) \wedge (t_0 = t_{\text{dep}}), w(\mathbf{x}, \mathbf{A}) | \mathbf{x}, \mathbf{A})}$$

Outline

Experimental Evaluation: Description

- We compared the following tools:
  - WMI-BC is our re-implementation of the WMI_{old} procedure in [8];
  - WMI-ALLSMT and WMI-PA;
  - SVE-XADD is the tool in [24] we adapted to parse our input format;
  - PRAISE is the tool of Probabilistic Inference Modulo Theories [13].
- In WMI-BC, WMI-ALLSMT and WMI-PA we use
  - MATHSAT5 [11, 1] for SMT reasoning
  - LATTE INTEGRALE [18, 2] to compute integrals of polynomials
  - SYMPY [3], a Python library for symbolic mathematics, for weight manipulations
- Experiments:
  - synthetic settings [19, 20]
  - real-world Strategic Road Network Dataset [4] by the English Highways Agency (both fixed-path and conditional-plan)
- run on 7-core Virtual Machine, 2.2 GHz and 94 GB of RAM
- timeout at 10,000 seconds for each ⟨query, tool⟩ job pair
- if terminating, all tools returned the same values on the same queries (modulo roundings)
- tools, data, and scripts used for experiments are publicly available [5]

Results on Synthetic Settings [19]

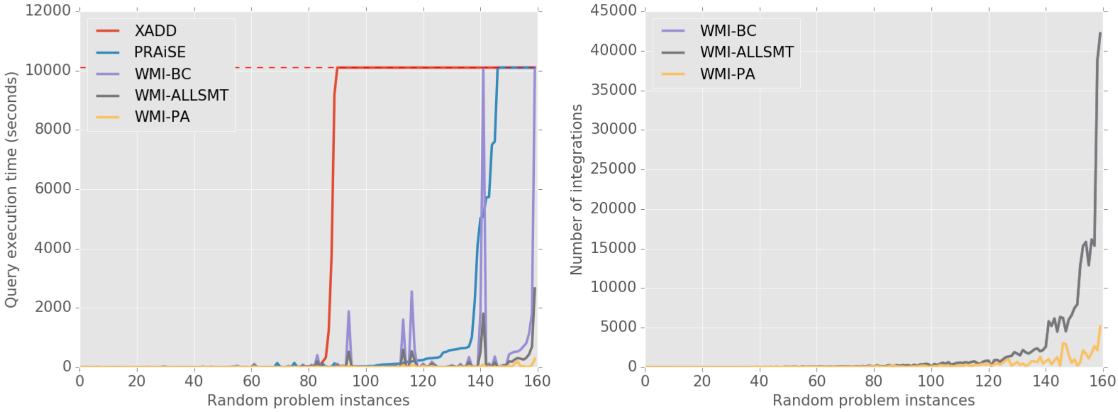


(left): Query execution times for all methods

(center): Query execution times for the three most performing algorithms

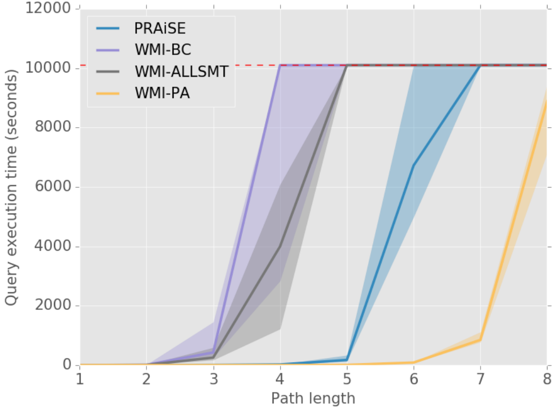
(right): Number of integrals for the three most performing algorithms

Results on Synthetic Settings [20]



(left): Query execution times (in seconds) for all methods on the synthetic experiment;
(right): Number of integrals for WMI-BC, WMI-ALLSMT and WMI-PA on the same instances.

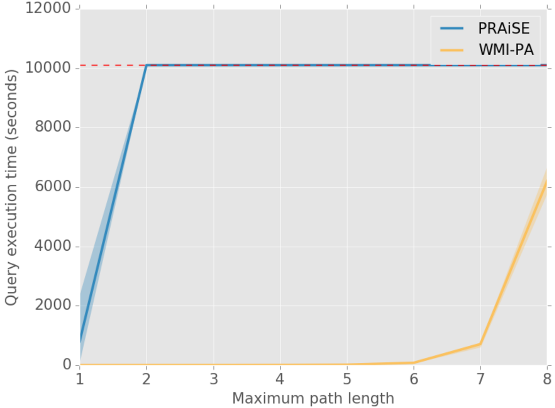
Strategic Road Network with Fixed Path



#	PRAiSE	WMI		
		BC	AllSMT	PA
1	2	1	0	0
2	3	10	8	0
3	7	425	253	0
4	22	> 10000	3994	2
5	174	> 10000	> 10000	8
6	6722	> 10000	> 10000	86
7	> 10000	> 10000	> 10000	850
8	> 10000	> 10000	> 10000	8884

(left): Query execution times in seconds (1^{st} quartile, median and 3^{rd} quartile).
(right): Table showing the medians for each length (right).

Strategic Road Network with Conditional Plan



#	PRAiSE	WMI-PA
1	799	1
2	> 10000	2
3	> 10000	4
4	> 10000	6
5	> 10000	14
6	> 10000	77
7	> 10000	708
8	> 10000	6203

(left): Query execution times in seconds (1^{st} quartile, median and 3^{rd} quartile).
(right): Table showing the medians for each length (right).

Outline

Conclusion

- Novel WMI formulation
  - easily captures the previous definition (not vice versa);
  - works with weight functions $w(\mathbf{x}, \mathbf{A})$ rather than $w(lit(\mathbf{x}, \mathbf{A}))$
  - w not restricted to products of weights *over literals*
    $\Longrightarrow$ allows for much more general forms, $\text{FIUC}^{\mathcal{LRA}}$
- Novel (WMI-ALLSMT and) WMI-PA procedure
  - 2 step: predicate abstraction + partial-assignment AllSMT, interleaved with formula simplification

  $\Longrightarrow$ reduces drastically the number of integrals to compute
- Empirical evaluation on both synthetic and real-word problems
  - WMI-PA outperforms WMI-ALLSMT and previous approaches
- A WMI-ALLSMT & WMI-PA tool available: pywmi [16] (`https://pypi.org/project/pywmi/`)

Note:

CPU times for WMI-PA largely dominated by $\text{WMI}_{\text{nb}}(\mu^{\mathcal{LRA}}, w|\mathbf{x}) \stackrel{\text{def}}{=} \int_{\mu^{\mathcal{LRA}}(\mathbf{x})} w(\mathbf{x})\, d\mathbf{x}$ calls
$\Longrightarrow$ computation of integrals current bottleneck

Ongoing & Future Work

Efficiency

- look for more efficient basic integrator for $\mathsf{WMI}_{\mathsf{nb}}(\varphi, w|\mathbf{x}) \stackrel{\text{def}}{=} \int_{\mu^{\mathcal{LRA}}(\mathbf{x})} w(\mathbf{x})\, d\mathbf{x}$
- $\mathcal{TA}(\varphi)$: more effective partial-assignment reduction techniques
- exploiting $w(\mathbf{x}, \mu^{\mathbf{A}})$ with partial $\mu^{\mathbf{A}}$s
- investigate forms of approximated enumeration [14]
- investigate forms of component caching [22, 6]

Expressiveness

- Extend WMI to integers and mixed real/integers
- Extend WMI integration domains to (subcases of) non-linear arithmetic constraints?

Others

- find other applications, other than probabilistic reasoning?


"That's all Folks!"

References I

[1] http://mathsat.fbk.eu/.

[2] https://www.math.ucdavis.edu/$\sim$latte/.

[3] http://www.sympy.org/.

[4] https://data.gov.uk/dataset/dft-eng-srn-routes-journey-times.

[5] https://github.com/unitn-sml/wmi-pa.

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

[7] V. Baldoni, N. Berline, J. D. Loera, M. Köppe, and M. Vergne.
How to integrate a polynomial over a simplex.
Mathematics of Computation, 80(273):297–325, 2011.

[8] V. Belle, A. Passerini, and G. V. den Broeck.
Probabilistic inference in hybrid domains by weighted model integration.
In *IJCAI*, 2015.

[9] R. Cavada, A. Cimatti, A. Franzén, K. Kalyanasundaram, M. Roveri, and R. Shyamasundar.
Computing Predicate Abstractions by Integrating BDDs and SMT Solvers.
In *FMCAD*, 2007.

[10] M. Chavira and A. Darwiche.
On probabilistic inference by weighted model counting.
Artificial Intelligence, 172(6-7):772–799, 2008.

[11] A. Cimatti, A. Griggio, B. J. Schaafsma, and R. Sebastiani.
The MathSAT 5 SMT Solver.
In *TACAS*, 2013.

[12] A. Darwiche.
New advances in compiling CNF to decomposable negation normal form.
In *Proceedings of ECAI*, pages 328–332, 2004.

References II

[13] R. de Salvo Braz, C. O'Reilly, V. Gogate, and R. Dechter.
Probabilistic Inference Modulo Theories.
In *IJCAI*, 2016.

[14] S. Ermon, C. P. Gomes, A. Sabharwal, and B. Selman.
Embed and project: Discrete sampling with universal hashing.
In *NIPS*, pages 2085–2093, 2013.

[15] S. Graf and H. Saïdi.
Construction of abstract state graphs with pvs.
In *CAV*, 1997.

[16] S. Kolb, P. Morettin, P. Z. D. Martires, F. Sommavilla, A. Passerini, and R. S. L. D. Raedt.
The pywmi framework and toolbox for probabilistic inference using weighted model integration.
In *Proc IJCAI*, 2019.
To appear.

[17] S. K. Lahiri, R. Nieuwenhuis, and A. Oliveras.
SMT techniques for fast predicate abstraction.
In *CAV*, 2006.

[18] J. D. Loera, B. Dutra, M. Koeppe, S. Moreinis, G. Pinto, and J. Wu.
Software for exact integration of polynomials over polyhedra.
ACM Communications in Computer Algebra, 45(3/4):169–172, 2012.

[19] P. Morettin, A. Passerini, and R. Sebastiani.
Efficient weighted model integration via SMT-based predicate abstraction.
In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 720–728, 2017.

[20] P. Morettin, A. Passerini, and R. Sebastiani.
Advanced smt techniques for weighted model integration.
Artificial Intelligence, 2019.
`https://doi.org/10.1016/j.artint.2019.04.003`.

[21] C. Muise, S. A. McIlraith, J. C. Beck, and E. I. Hsu.
Dsharp: fast d-dnnf compilation with sharpsat.
In *Advances in Artificial Intelligence*, pages 356–361. Springer, 2012.

References III

[22] T. Sang, F. Bacchus, P. Beame, H. A. Kautz, and T. Pitassi.
Combining component caching and clause learning for effective model counting.
In *SAT*, 2004.

[23] T. Sang, P. Beame, and H. A. Kautz.
Performing bayesian inference by weighted model counting.
In *AAAI*, volume 5, pages 475–481, 2005.

[24] S. Sanner, K. V. Delgado, and L. N. de Barros.
Symbolic dynamic programming for discrete and continuous state mdps.
In *UAI 2011, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, Barcelona, Spain, July 14-17, 2011*, pages 643–652, 2011.