On the Parameterized Complexity of Default Logic and Autoepistemic Logic

Arne Meier   Johannes Schmidt   Michael Thomas   Heribert Vollmer

Universität Hannover

Université de la Méditerranée

TWT GmbH

2012-03-08

Lata 2012, A Coruña, Spain

What is Default Logic?

What is Default Logic?

- a non-monotone logic, introduced 1980 by Reiter
- models common-sense reasoning
- extends classical logic with *default rules*

What is Default Logic?

What is Default Logic?

- a non-monotone logic, introduced 1980 by Reiter
- models common-sense reasoning
- extends classical logic with *default rules*
- undecidable for first order logic (Reiter)
- here: propositional logic

Default Rules and Theories

Definition (Reiter 80)

A default rule is a triple $\frac{\alpha : \beta}{\gamma}$, where

α is called the prerequisite,

β is called the justification, and

γ is called the consequent,

for α, β, γ propositional formulae.

Default Rules and Theories

Definition (Reiter 80)

A default rule is a triple $\frac{\alpha : \beta}{\gamma}$, where

α is called the prerequisite,

β is called the justification, and

γ is called the consequent,

for α, β, γ propositional formulae.

Informally: infer a formula γ from a set of formulae W by a default rule $\frac{\alpha : \beta}{\gamma}$, if

- α is derivable from W and
- β is consistent with W.

Default Theories

Definition (Reiter 80)

A default theory is a tuple $\langle W, D\rangle$, where W is a set of formulae and D is a set of default rules.

Example: Playing Football with Default Rules

$$W := \{football, rain, cold \land rain \rightarrow snow\}$$

$$D := \left\{\frac{football : \neg snow}{takesPlace}\right\}$$

$\neg snow$ is consistent with W. Hence we can infer $takesPlace$.

Default Theories

Definition (Reiter 80)

A default theory is a tuple $\langle W, D \rangle$, where W is a set of formulae and D is a set of default rules.

Example: Playing Football with Default Rules

$$W := \{football, rain, cold \land rain \rightarrow snow, cold\}$$

$$D := \left\{ \frac{football : \neg snow}{takesPlace} \right\}$$

$\neg snow$ is inconsistent with W. Hence we cannot infer $takesPlace$.

Default logics are non-monotone!

Semantics: Stable Extensions

Definition (Reiter 80)

For default theory $\langle W, D \rangle$ and set of formulae E, we define $\Gamma(E)$ as the smallest set, s.t.

1. $W \subseteq \Gamma(E)$,
2. $\Gamma(E)$ is closed under deduction, and
3. for all defaults $\frac{\alpha : \beta}{\gamma}$ with $\alpha \in \Gamma(E)$ and $\neg\beta \notin E$, it holds that $\gamma \in \Gamma(E)$.

A stable extension of $\langle W, D \rangle$ is a set E s.t. $E = \Gamma(E)$.

Semantics: Stable Extensions

Definition (Reiter 80)

For default theory $\langle W, D \rangle$ and set of formulae E, we define $\Gamma(E)$ as the smallest set, s.t.

1. $W \subseteq \Gamma(E)$,
2. $\Gamma(E)$ is closed under deduction, and
3. for all defaults $\frac{\alpha : \beta}{\gamma}$ with $\alpha \in \Gamma(E)$ and $\neg\beta \notin E$, it holds that $\gamma \in \Gamma(E)$.

A stable extension of $\langle W, D \rangle$ is a set E s.t. $E = \Gamma(E)$.

Stable extensions correspond to possible views of an agent on the basis of $\langle W, D \rangle$.

Generating Defaults

Semantics: Generating Defaults (Reiter 80)

Given: a default theory $\langle W, D \rangle$ and set of formulae E:

Define the set of generating defaults as

$$G := \left\{ \frac{\alpha : \beta}{\gamma} \in D \;\middle|\; \alpha \in E \text{ and } \neg\beta \notin E \right\}.$$

Then: E is stable a extension of $\langle W, D \rangle$ iff

$$E = \mathrm{Th}\left(W \cup \left\{ \gamma \;\middle|\; \frac{\alpha : \beta}{\gamma} \in G \right\} \right).$$

The Decision Problem

Extension Existence Problem

Instance: a default theory $\langle W, D \rangle$

Question: Does $\langle W, D \rangle$ have a stable extension?

Theorem (Gottlob 92)

The Extension Existence Problem is Σ_2^{P}-complete.

Boolean function, B-formula

n-ary Boolean function $f : \{0,1\}^n \to \{0,1\}$

Let B be a set of Boolean functions.

A *B-formula* is a formula whose connectives are functions from B
$\mathcal{L}(B)$ denotes the set of all B-formulae

Boolean function, B-formula

n-ary Boolean function $f : \{0,1\}^n \to \{0,1\}$

Let B be a set of Boolean functions.

A *B-formula* is a formula whose connectives are functions from B
$\mathcal{L}(B)$ denotes the set of all B-formulae

$B = \{\vee, \neg, f, g\}$

$$\left.\begin{array}{l} \neg x \\ x \vee \neg y \\ \neg(x \vee y) \vee z \\ \neg\neg\neg x \\ f(g(x,y), f(x, g(z,y), y), z) \end{array}\right\} \in \mathcal{L}(B)$$

Parameterized Complexity Theory

Definition (FPT)

- $Q \subseteq \Sigma^*$ a decision problem
- $\kappa : \Sigma^* \to \mathbb{N}$ associates a parameter to the instances

$\mathbb{A}$ is an fpt-algorithm for (Q, κ) if there is a computable function $f : \mathbb{N} \to \mathbb{N}$, a polynomial p s.t. $\mathbb{A}$ decides Q and for every input x, $\mathbb{A}$s runtime is at most $f(k) \cdot p(|x|)$, where $k = \kappa(x)$.

(Q, κ) is fixed-parameter tractable if there is an fpt-algorithm that decides (Q, κ). All such problems are in the class FPT.

tree width

- a parameter definable on any relational structure
- graph: intuitively it measures the distance of the graph from a tree
- many NP-hard problems become FPT with this parameter
- this parameter is not computable in polynomial time (but FPT)

tree width

- a parameter definable on any relational structure
- graph: intuitively it measures the distance of the graph from a tree
- many NP-hard problems become FPT with this parameter
- this parameter is not computable in polynomial time (but FPT)

Powerfull Theorem by Courcelle: all MSO-definable problems become FPT with the tree width parameter

A handy tool

Definition

Let Q be a decision problem, x be an instance.
Q is MSO-definable iff there exists an MSO-formula ϕ_Q and a function $x \mapsto \mathcal{A}_x$ s.t. it holds that $x \in Q$ if and only if $\mathcal{A}_x \models \phi_Q$.

A handy tool

Definition

Let Q be a decision problem, x be an instance.
Q is MSO-definable iff there exists an MSO-formula ϕ_Q and a function $x \mapsto \mathcal{A}_x$ s.t. it holds that $x \in Q$ if and only if $\mathcal{A}_x \models \phi_Q$.

Theorem (Courcelle 1990, Elberfeld Jakoby Tantau 2010)

Let Q be an MSO-definable decision problem, and let $\mathcal{A}_x$ be the structure associated with an instance x. Further let $k \in \mathbb{N}$ s.t. the treewidth of $\mathcal{A}_x$ is bounded by k. Then Q is solvable in time $O(f(k) \cdot |x|)$ and space $O(\log(f(k)) + \log |x|)$.

A handy tool

Definition

Let Q be a decision problem, x be an instance.
Q is MSO-definable iff there exists an MSO-formula ϕ_Q and a function $x \mapsto \mathcal{A}_x$ s.t. it holds that $x \in Q$ if and only if $\mathcal{A}_x \models \phi_Q$.

Theorem (Courcelle 1990, Elberfeld Jakoby Tantau 2010)

Let Q be an MSO-definable decision problem, and let $\mathcal{A}_x$ be the structure associated with an instance x. Further let $k \in \mathbb{N}$ s.t. the treewidth of $\mathcal{A}_x$ is bounded by k. Then Q is solvable in time $O(f(k) \cdot |x|)$ and space $O(\log(f(k)) + \log |x|)$.

Describe a problem with MSO-formula ϕ.
Apply Courcelle's Theorem and get an FPT-result.

Express a Set of Boolean Functions Γ in MSO

- Let B be a set of Boolean functions. Define the Vocabulary as

$$\begin{aligned}\tau_B :=&\{\mathrm{const}^1_f \mid f \in B, \mathrm{arity}(f) = 0\} \cup \\ &\{\mathrm{conn}^2_{f,i} \mid f \in B, 1 \leq i \leq \mathrm{arity}(f)\}, \\ \tau_{B,prop} :=&\tau_B \cup \{\mathrm{variable}^1, \mathrm{formula}^1\}.\end{aligned}$$

Express a Set of Boolean Functions Γ in MSO

- Let B be a set of Boolean functions. Define the Vocabulary as

for constants f

$$\begin{aligned}\tau_B :=& \{\mathrm{const}^1_f \mid f \in B, \mathrm{arity}(f) = 0\} \cup \\ & \{\mathrm{conn}^2_{f,i} \mid f \in B, 1 \leq i \leq \mathrm{arity}(f)\}, \\ \tau_{B,prop} :=& \tau_B \cup \{\mathrm{variable}^1, \mathrm{formula}^1\}.\end{aligned}$$

Express a Set of Boolean Functions Γ in MSO

- Let B be a set of Boolean functions. Define the Vocabulary as

$$\tau_B := \{\mathrm{const}^1_f \mid f \in B, \mathrm{arity}(f) = 0\} \cup \{\mathrm{conn}^2_{f,i} \mid f \in B, 1 \le i \le \mathrm{arity}(f)\},$$

$$\tau_{B,prop} := \tau_B \cup \{\mathrm{variable}^1, \mathrm{formula}^1\}.$$

for constants f

ith argument of f

Express a Set of Boolean Functions Γ in MSO

- Let B be a set of Boolean functions. Define the Vocabulary as

$$\tau_B := \{\mathrm{const}^1_f \mid f \in B, \mathrm{arity}(f) = 0\} \cup \{\mathrm{conn}^2_{f,i} \mid f \in B, 1 \leq i \leq \mathrm{arity}(f)\},$$

$$\tau_{B,prop} := \tau_B \cup \{\mathrm{variable}^1, \mathrm{formula}^1\}.$$

for constants f

ith argument of f

for variables

for formulae

Express a Set of Boolean Functions Γ in MSO

- Let B be a set of Boolean functions. Define the Vocabulary as

$$\begin{aligned}\tau_B :=& \{\mathrm{const}^1_f \mid f \in B, \mathrm{arity}(f) = 0\} \cup \\ & \{\mathrm{conn}^2_{f,i} \mid f \in B, 1 \leq i \leq \mathrm{arity}(f)\}, \\ \tau_{B,prop} :=& \tau_B \cup \{\mathrm{variable}^1, \mathrm{formula}^1\}.\end{aligned}$$

for constants f

ith argument of f

for variables

for formulae

With Γ a set of B-formulae we associate the $\tau_{B,prop}$-structure $\mathcal{A}_\Gamma$.

Universe: Formulae and their subformulae.

Expressing Satisfiability

Lemma

Let B be a finite set of Boolean functions. Then there exists an MSO-formula θ_{sat} over $\tau_{B,prop}$ such that for $\Gamma \subseteq \mathcal{L}(B)$

$$\Gamma \textit{ is satisfiable iff } \mathcal{A}_\Gamma \models \theta_{sat}.$$

Expressing Satisfiability

Lemma

Let B be a finite set of Boolean functions. Then there exists an MSO-formula θ_{sat} over $\tau_{B,prop}$ such that for $\Gamma \subseteq \mathcal{L}(B)$

$$\Gamma \textit{ is satisfiable iff } \mathcal{A}_\Gamma \models \theta_{sat}.$$

Proof (sketch)

Consider the MSO-formula

$$\theta_{sat} := \theta_{struc} \wedge \exists M\big(\theta_{assign}(M) \wedge \forall x\Big(\mathrm{formula}(x) \rightarrow M(x)\Big)$$

Expressing Satisfiability

Lemma

Let B be a finite set of Boolean functions. Then there exists an MSO-formula θ_{sat} over $\tau_{B,prop}$ such that for $\Gamma \subseteq \mathcal{L}(B)$

$$\Gamma \text{ is satisfiable iff } \mathcal{A}_\Gamma \models \theta_{sat}.$$

Proof (sketch)

Consider the MSO-formula

$$\theta_{sat} := \theta_{struc} \wedge \exists M \big(\theta_{assign}(M) \wedge \forall x \Big(\mathrm{formula}(x) \rightarrow M(x)\Big)$$

Let $f(1) = 0$, $g(0,1) = 1$. $\theta_{assign}(M)$ demands that, if $x \in M$, then
$g(f(x), x) \in M$ and
$f(x) \notin M$.

Expressing Implication

Let B be a set of Boolean functions and F, G be sets of B-formulae.
Extend our vocabulary to express $F \models G$:

$$\tau_{B,imp} := \tau_{B,prop} \cup \{\mathrm{formula}^1_{prem}, \mathrm{formula}^1_{conc}\},$$

where $\mathrm{formula}_{prem}(x)$ is true iff x represents a formula from F,
and $\mathrm{formula}_{conc}(x)$ is true iff x represents a formula from G.

Lemma

Let B be a finite set of Boolean functions. Then there exists an MSO-formula θ_{imp} over $\tau_{B,imp}$ such that for any $\Gamma \subseteq \mathcal{L}(B)$ and any $\Delta_1, \Delta_2 \subseteq \Gamma$ it holds that there exists a structure $\mathcal{A}_\Gamma$ such that

$$\Delta_1 \models \Delta_2 \textit{ iff } \textit{ and } \mathcal{A}_\Gamma \models \theta_{imp}.$$

Proof.

Define the MSO-formulae $\theta_{premise}(M)$, $\theta_{conclusion}(M)$, and $\theta_{implies}$ as follows:

□

Lemma

Let B be a finite set of Boolean functions. Then there exists an MSO-formula θ_{imp} over $\tau_{B,imp}$ such that for any $\Gamma \subseteq \mathcal{L}(B)$ and any $\Delta_1, \Delta_2 \subseteq \Gamma$ it holds that there exists a structure $\mathcal{A}_\Gamma$ such that

$$\Delta_1 \models \Delta_2 \text{ iff } \text{ and } \mathcal{A}_\Gamma \models \theta_{imp}.$$

Proof.

Define the MSO-formulae $\theta_{premise}(M)$, $\theta_{conclusion}(M)$, and $\theta_{implies}$ as follows:

$$\theta_{premise}(M) := \forall x(\mathrm{formula}_{prem}(x) \to M(x))$$

□

Lemma

Let B be a finite set of Boolean functions. Then there exists an MSO-formula θ_{imp} *over* $\tau_{B,imp}$ *such that for any* $\Gamma \subseteq \mathcal{L}(B)$ *and any* $\Delta_1, \Delta_2 \subseteq \Gamma$ *it holds that there exists a structure* $\mathcal{A}_\Gamma$ *such that*

$$\Delta_1 \models \Delta_2 \textit{ iff } \textit{ and } \mathcal{A}_\Gamma \models \theta_{imp}.$$

Proof.

Define the MSO-formulae $\theta_{premise}(M)$, $\theta_{conclusion}(M)$, and $\theta_{implies}$ as follows:

$$\theta_{premise}(M) := \forall x(\text{formula}_{prem}(x) \rightarrow M(x))$$
$$\theta_{conclusion}(M) := \forall x(\text{formula}_{conc}(x) \rightarrow M(x)))$$

□

Lemma

Let B be a finite set of Boolean functions. Then there exists an MSO-formula θ_{imp} over $\tau_{B,imp}$ such that for any $\Gamma \subseteq \mathcal{L}(B)$ and any $\Delta_1, \Delta_2 \subseteq \Gamma$ it holds that there exists a structure $\mathcal{A}_\Gamma$ such that

$$\Delta_1 \models \Delta_2 \text{ iff and } \mathcal{A}_\Gamma \models \theta_{imp}.$$

Proof.

Define the MSO-formulae $\theta_{premise}(M)$, $\theta_{conclusion}(M)$, and $\theta_{implies}$ as follows:

$$\begin{aligned}
\theta_{premise}(M) &:= \forall x(\mathrm{formula}_{prem}(x) \rightarrow M(x)) \\
\theta_{conclusion}(M) &:= \forall x(\mathrm{formula}_{conc}(x) \rightarrow M(x))) \\
\theta_{implies} &:= \forall M\Big(\big(\theta_{assign}(M) \wedge \theta_{premise}(M)\big) \rightarrow \theta_{conclusion}(M)\Big)
\end{aligned}$$

□

Lemma

Let B be a finite set of Boolean functions. Then there exists an MSO-formula θ_{imp} over $\tau_{B,imp}$ such that for any $\Gamma \subseteq \mathcal{L}(B)$ and any $\Delta_1, \Delta_2 \subseteq \Gamma$ it holds that there exists a structure $\mathcal{A}_\Gamma$ such that

$$\Delta_1 \models \Delta_2 \textit{ iff } \textit{ and } \mathcal{A}_\Gamma \models \theta_{imp}.$$

Proof.

Define the MSO-formulae $\theta_{premise}(M)$, $\theta_{conclusion}(M)$, and $\theta_{implies}$ as follows:

$$\begin{aligned}
\theta_{premise}(M) &:= \forall x(\mathrm{formula}_{prem}(x) \to M(x)) \\
\theta_{conclusion}(M) &:= \forall x(\mathrm{formula}_{conc}(x) \to M(x))) \\
\theta_{implies} &:= \forall M\Big(\big(\theta_{assign}(M) \wedge \theta_{premise}(M)\big) \to \theta_{conclusion}(M)\Big)
\end{aligned}$$

Then, we can define the formulae θ_{imp} as $\theta_{imp} := \theta_{struc} \wedge \theta_{implies}$. □

On the Way to Default Logic: Extend the Vocabulary

Default Theory (W, D), where $D := \{\frac{\alpha_i : \beta_i}{\gamma_i} \mid 1 \leq i \leq n\}$.

$$\tau_{B,dl} := \tau_{B,imp} \cup \{kb^1, def^1, prem^2, concl^2, just^2\}$$

On the Way to Default Logic: Extend the Vocabulary

Default Theory (W, D), where $D := \{\frac{\alpha_i : \beta_i}{\gamma_i} \mid 1 \leq i \leq n\}$.

$$\tau_{B,dl} := \tau_{B,imp} \cup \{kb^1, def^1, prem^2, concl^2, just^2\}$$

On the Way to Default Logic: Extend the Vocabulary

Default Theory (W, D), where $D := \{\frac{\alpha_i : \beta_i}{\gamma_i} \mid 1 \leq i \leq n\}$.

$$\tau_{B,dl} := \tau_{B,imp} \cup \{kb^1, def^1, prem^2, concl^2, just^2\}$$

On the Way to Default Logic: Extend the Vocabulary

Default Theory (W, D), where $D := \{\frac{\alpha_i : \beta_i}{\gamma_i} \mid 1 \leq i \leq n\}$.

$$\tau_{B,dl} := \tau_{B,imp} \cup \{kb^1, def^1, prem^2, concl^2, just^2\}$$

On the Way to Default Logic: Extend the Vocabulary

Default Theory (W, D), where $D := \{\frac{\alpha_i : \beta_i}{\gamma_i} \mid 1 \leq i \leq n\}$.

$$\tau_{B,dl} := \tau_{B,imp} \cup \{kb^1, def^1, prem^2, concl^2, just^2\}$$

On the Way to Default Logic: Extend the Vocabulary

Default Theory (W, D), where $D := \{\frac{\alpha_i : \beta_i}{\gamma_i} \mid 1 \leq i \leq n\}$.

$$\tau_{B,dl} := \tau_{B,imp} \cup \{kb^1, def^1, prem^2, concl^2, just^2\}$$

Extension Existence of Default Logic

Lemma

Let B be a finite set of Boolean functions, (W, D) a B-default theory and $\mathcal{A}_{(W,D)}$ the associated structrue. Then there exists an MSO-formula $\theta_{extension}$ such that (W, D) possesses a stable extension iff $\mathcal{A}_{(W,D)} \models \theta_{extension}$.

Extension Existence of Default Logic

Lemma

Let B be a finite set of Boolean functions, (W, D) a B-default theory and $\mathcal{A}_{(W,D)}$ the associated structrue. Then there exists an MSO-formula $\theta_{extension}$ such that (W, D) possesses a stable extension iff $\mathcal{A}_{(W,D)} \models \theta_{extension}$.

Proof.

$\theta_{extension} := \theta_{struc} \wedge \exists G(\theta_{stable}(G))$, where

$\theta_{stable}(G) := \forall d\big(def(d) \rightarrow (G(d) \leftrightarrow \theta_{app}(d, G))\big)$ □

Apply Courcelle's Theorem

Theorem

Let B be a finite set of Boolean functions, let $k \in \mathbb{N}$ be fixed, and let (W, D) be a B-default theory such that the treewidth of structures for $\theta_{extension}$ is bounded by k.

Then the extension existence problem for B-default logic is solvable in time $O(f(k) \cdot |(W, D)|)$ and space $O(\log(f(k)) + \log |(W, D)|)$.

Apply Courcelle's Theorem

Theorem

Let B be a finite set of Boolean functions, let $k \in \mathbb{N}$ be fixed, and let (W, D) be a B-default theory such that the treewidth of structures for $\theta_{extension}$ is bounded by k.

Then the extension existence problem for B-default logic is solvable in time $O(f(k) \cdot |(W, D)|)$ and space $O(\log(f(k)) + \log |(W, D)|)$.

Lower bounds

Theorem

There exists a family of (very simple) default theories $(\emptyset, D)_k$ such that the tree width of $\mathcal{A}_{(\emptyset,D)_k}$ is not constant.

Transfer to Autoepistemic Logic

- Introduced by Moore 1985.
- New modal operator L to model the beliefs of a perfect rational agent.
- Satisfiability and reasoning Σ_2^{P}-complete.

Transfer to Autoepistemic Logic

- Introduced by Moore 1985.
- New modal operator L to model the beliefs of a perfect rational agent.
- Satisfiability and reasoning Σ_2^{P}-complete.

Theorem

Let B be a finite set of Boolean functions, let $k \in \mathbb{N}$ be fixed, and let Σ be a set of autoepistemic B-formulae such that $\mathcal{A}_\Sigma$ has tree width bounded by k. Then the expansion problem is solvable in time $O(f(k) \cdot |\Sigma|)$ and space $O(\log(f(k)) + \log|\Sigma|)$.

What Did We Learn and What is Next?

- We got some intuition about nonmonotonicity.

What Did We Learn and What is Next?

- We got some intuition about nonmonotonicity.
- We learned how to apply Courcelle's Theorem.

What Did We Learn and What is Next?

- We got some intuition about nonmonotonicity.
- We learned how to apply Courcelle's Theorem.
- We enjoyed Monadic Second Order Logic. A lot.

What Did We Learn and What is Next?

- We got some intuition about nonmonotonicity.
- We learned how to apply Courcelle's Theorem.
- We enjoyed Monadic Second Order Logic. A lot.

Further (not in this talk) we exhibit lower bounds for very restricted

- default theories,
- sets of autoepistemic formulae, and

What Did We Learn and What is Next?

- We got some intuition about nonmonotonicity.
- We learned how to apply Courcelle's Theorem.
- We enjoyed Monadic Second Order Logic. A lot.

Further (not in this talk) we exhibit lower bounds for very restricted

- default theories,
- sets of autoepistemic formulae, and
- sets of propositional formulae (w.r.t. the implication problem).

What Did We Learn and What is Next?

- We got some intuition about nonmonotonicity.
- We learned how to apply Courcelle's Theorem.
- We enjoyed Monadic Second Order Logic. A lot.

Further (not in this talk) we exhibit lower bounds for very restricted

- default theories,
- sets of autoepistemic formulae, and
- sets of propositional formulae (w.r.t. the implication problem).

The next two steps involve:

- Thinking about useful kinds of parameterizations of propositional Default Logic beyond treewidth.
- And the same for autoepistemic logic.

What Did We Learn and What is Next?

- We got some intuition about nonmonotonicity.
- We learned how to apply Courcelle's Theorem.
- We enjoyed Monadic Second Order Logic. A lot.

Further (not in this talk) we exhibit lower bounds for very restricted

- default theories,
- sets of autoepistemic formulae, and
- sets of propositional formulae (w.r.t. the implication problem).

The next two steps involve:

- Thinking about useful kinds of parameterizations of propositional Default Logic beyond treewidth.
- And the same for autoepistemic logic.

Thank you!