Comparison
and
Efficient Computation
of
STATIC and the Disjunctive WFS

S. Brass, J. Dix, I. Niemelä, T. Przymusinski

Universities of Hildesheim, Koblenz, Helsinki and Riverside

Part I: STATIC and D-WFS

- Disjunctive Programs and Semantics
- Logic of Autoepistemic Beliefs
- STATIC
- D-WFS

Part II: Restricting STATIC and efficient Comp.

- STATIC in $\mathcal{L}_{DBL}$
- D-WFS as Iterated GCWA
- A Polynomial Space Algorithm

Part III: Outlook

1 Programs and Semantics

Definition I.1.1 (Logic Program P, $head(P)$)
A logic program P is a finite set of propositional rules

$$\boxed{A_1 \vee \cdots \vee A_k \leftarrow B_1 \wedge \cdots \wedge B_m \wedge \neg C_1 \wedge \cdots \wedge \neg C_n},$$

where the $A_i/B_i/C_i$ are $\mathcal{L}$-atoms different from $\bot$, $k \geq 1$, $m \geq 0$, $n \geq 0$.

We identify such a rule with a triple and write it as $\boxed{\mathcal{A} \leftarrow \mathcal{B}^+ \wedge \neg\mathcal{B}^-}$.

We write $head(P)$ for the set of all atoms occuring in rule heads in P: these are atoms that are possibly true. By $\boxed{\textit{pure}}$ *disjunctions we mean disjunctions consisting solely of* $\boxed{\textit{positive}}$ *or solely of* $\boxed{\textit{negative}}$ *literals.*

Definition I.1.2 (Operator $\mid\!\sim$, Semantics $\mathcal{S}_{\mid\sim}$)

By a *semantic operator* $\mid\!\sim$ *we mean a binary relation between logic programs and pure disjunctions satisfying*

1. *Right Weakening:* *If* $P \mid\!\sim \psi$ *and* $\psi \subseteq \psi'$[a], *then* $P \mid\!\sim \psi'$.
2. *Necessarily True:*
   If $\mathcal{A} \leftarrow true \in P$ *for a disjunction* $\mathcal{A}$, *then* $P \mid\!\sim \mathcal{A}$.
3. *Necessarily False:*
   If $A \notin head(P)$ *for some* $\mathcal{L}$*-ground atom* A, *then* $P \mid\!\sim \neg A$.

Given such $\mid\!\sim$ *and a program* P, *by the* *semantics* *of* P *determined by* $\mid\!\sim$ *we mean the* *set of all pure disjunctions* *derivable by* $\mid\!\sim$ *from* P, *i.e.*, $\{\psi \mid P \mid\!\sim \psi\}$.

[a] I. e. ψ is a subdisjunction of ψ'.

2 Logic of Autoepistemic Beliefs

Language: propositional modal language $\boxed{\mathcal{L}_{AELB}}$, Belief operator $\boxed{\mathcal{B}_{el}}$. $\mathcal{B}_{el}F$ are called *belief atoms*, where F is an arbitrary formula of $\mathcal{L}_{AELB}$.

Axioms:

(D) Consistency: $\neg\mathcal{B}_{el}\bot$

(K) Normality: $\mathcal{B}_{el}(F \to G) \ \to \ (\mathcal{B}_{el}F \to \mathcal{B}_{el}G)$

Inference-rule:

(N) Necessitation: $\frac{F}{\mathcal{B}_{el}F}$

Definition I.2.1 ($\boxed{Cn_{AELB}(T)}$ **)**
For any belief theory T, let $Cn_{AELB}(T)$ the smallest set of formulae in $\mathcal{L}_{AELB}$ which contains T, all the (substitution instances of) the axioms (K) and (D) and is closed under both standard prop. consequence and necessitation (N).

Definition I.2.2 (Minimal Model)
A model M of T with the property that there is no smaller model N of T which coincides with M on belief atoms $\mathcal{B}_{el}F$, is called a minimal model: Belief atoms are fixed.

3 STATIC

Definition I.3.1 (Static Expansion)
A belief theory $T^\diamond$ is called a static autoepistemic expansion *of a belief theory T if it satisfies the following fixed-point equation: F ranges over formulae of $\mathcal{L}_{AELB}$,*

$$\boxed{T^\diamond} = Cn_{AELB}(T \ \cup \{\mathcal{B}_{el}F : \ \boxed{T^\diamond} \models_{\min} F\}),$$

Definition I.3.2 (Belief-Programs)
A Belief-program is a set of clauses of the form:

$$A_1 \vee \ldots \vee A_k \leftarrow B_1 \wedge \ldots \wedge B_m \wedge \boxed{\mathcal{B}_{el}\neg C_1} \wedge \ldots \wedge \boxed{\mathcal{B}_{el}\neg C_n},$$

where A_i's, B_i's, and C_i's are $\mathcal{L}$-atoms and $k > 0$.

Any disjunctive logic program P can be translated into a special belief theory in $\mathcal{L}_{AELB}$. The meaning of $\neg$ is given by $\boxed{\mathit{not}\, F \equiv \ F \text{ is false in all minimal models}}$.

Definition I.3.3 (STATIC)
Given a logic program P, we translate it into a belief program, compute the least static expansion and consider all formulae of the form

- $\boxed{A_1 \vee \ldots \vee A_n}$ *or*
- $\boxed{\mathcal{B}_{el}\neg A_1 \vee \ldots \vee \mathcal{B}_{el}\neg A_m}$

true in it. Their re-translations $A_1 \vee \ldots \vee A_n$ resp. $\neg A_1 \vee \ldots \vee \neg A_m$ constitute the set of derivable pure disjunctions: we call this set STATIC(P).

4 D-WFS

Theorem I.4.1 (D-WFS, (Brass/Dix '95/'96))
There exists the weakest semantics $\mathcal{S}_{\mid\!\sim}$ which is invariant under the following transformations: $\boxed{\textit{GPPE}}$, $\boxed{\textit{Tautology}}$, $\boxed{\textit{Non-minimal Rules}}$, $\boxed{\textit{Positive and negative Reduction}}$.
Since these transformations constitute a $\boxed{\textit{confluent}}$ *and* $\boxed{\textit{terminating}}$ *calculus, any program is associated a unique normalform* $\hat{P}$.

We also have:

$$P \mid\!\sim_{\mathrm{DWFS}} \psi \iff \begin{array}{l}\text{there is } \mathcal{A} \subseteq \psi \text{ with } \mathcal{A} \leftarrow \mathit{true} \in \hat{P} \text{ or} \\ \text{there is } \neg A \in \psi \text{ and } A \notin \mathit{head}(\hat{P}).\end{array}$$

Tautology: $A \vee B \vee C \leftarrow D, A, \neg E$

$$\begin{array}{lcl} A \vee B \vee C & \leftarrow & D, \neg E, \neg F \\ A \vee B & \leftarrow & D, \neg F \end{array} \quad \mapsto \quad \begin{array}{lcl} A \vee B & \leftarrow & D, \neg F \end{array}$$

pos.

$$\begin{array}{lcl} A \vee B & \leftarrow & \neg F \\ E & \leftarrow & A, \neg F \end{array} \quad \mapsto \quad \begin{array}{lcl} A \vee B & \leftarrow & \\ E & \leftarrow & A \end{array}$$

neg.

$$\begin{array}{lcl} A \vee B & \leftarrow & \\ F \vee E & \leftarrow & C, \neg A, \neg B, \neg F \end{array} \quad \mapsto \quad \begin{array}{lcl} A \vee B & \leftarrow & \end{array}$$

1 STATIC in $\mathcal{L}_{DBL}$

Language: Start with propositional language $\mathcal{L}$. Add a Belief operator $\boxed{\mathcal{B}_{el}}$ which can be only applied to a $\boxed{\text{positive}}$ or $\boxed{\text{negative}}$ formula $\boxed{\text{from } \mathcal{L}}$. Language: $\boxed{\mathcal{L}_{DBL}}$.

Axioms:

(D') Restricted Consistency: $\mathcal{B}_{el}F \rightarrow \neg\mathcal{B}_{el}\neg F$.

(C) Distributivity: $\mathcal{B}_{el}(F \wedge G) \equiv \mathcal{B}_{el}F \wedge \mathcal{B}_{el}G$

Inference-rule: (N) Necessitation: $\frac{F}{\mathcal{B}_{el}F}$

As before a static autoepistemic expansion is a theory satisfying

$$\boxed{T^{\diamond}} = Cn_{\boxed{\text{DBL}}}(T \cup \{\boxed{\mathcal{B}_{el}F} : \boxed{T^{\diamond}} \models_{\min} F\}).$$

Theorem II.1.1 (Least Static Expansion in $\mathcal{L}_{DBL}$) *Every belief program P in DBL has the least static expansion, which is consistent. The set of pure disjunctions contained in the least static expansion in $\mathcal{L}_{DBL}$ is already obtained as the least fixpoint of the following simplified closure operator $\Phi'_P(S)$*

$$
\begin{array}{llll}
\boxed{Cn}(P & \cup\{\boxed{\neg\mathcal{B}_{el}\neg q_1 \vee \ldots \vee \neg\mathcal{B}_{el}\neg q_k} : & S \models & q_1 \vee \ldots \vee q_k \ \} \\
 & \cup\{\boxed{\mathcal{B}_{el}(\neg p_1)} : & S \models_{min} & \neg p_1 \ \}),
\end{array}
$$

2 D-WFS as Iterated GCWA

The following characterization of D-WFS can be seen as a reduction of D-WFS to the minimal model-entailment.

Theorem II.2.1 (D-WFS)
Let P be a disjunctive logic program over $\mathcal{L}$. We define a set $Dis_{\min}(P)$ of pure disjunctions as follows

$$
\begin{aligned}
Dis_{\min}(P) := \ & \{A_1 \vee \cdots \vee A_k : \quad P \models A_1 \vee \cdots \vee A_k\} \cup \\
& \{\neg A : \quad P \boxed{\models_{\min(\mathrm{N})}} \neg A\}
\end{aligned}
$$

Then, the series D_i defined by $D_{n+1} := Dis_{\min}(\boxed{P/D_n})$ grows monotonically and eventually gets constant. It coincides with D-WFS(P).

The relation $\boxed{\models_{\mathrm{min(N)}}}$ captures a particular notion of minimal entailment: a model I (a set of atoms and formulae $\neg(\phi)$) of P is called **NAF-consistent** for P, if it satisfies the following condition:

if $P \models A_1 \vee \cdots \vee A_n$, then for some $i \in \{1, \ldots, n\}, \neg(A_i) \notin I$.

For example, $\{A, C, \neg A, \neg B\}$ is not a NAF-consistent model of $P = \{A \vee B \leftarrow, C \leftarrow \neg A \wedge \neg B\}$ but $\{A, C, \neg A\}$ is (however, it is not minimal).

3 A Polynomial Space Algorithm

- Implement D-WFS as an iterative reduction.
- Start with the original program P and compute P^* s.t. every query can be answered from P^* with one call to a decision procedure of $\models_{\min(\mathrm{N})}$.
- More precise: The reduced program is obtained as a fixed point of $\mathrm{R}_{\mathrm{D}}(\cdot)$ mirroring the reduction P/D.

$$
\begin{aligned}
\mathrm{R}_{\mathrm{D}}(P) \quad = \quad & \{\mathcal{A} \leftarrow \mathcal{B}^{+} \wedge \neg \mathcal{B}^{-\prime} \mid \mathcal{A} \leftarrow \mathcal{B}^{+} \wedge \neg \mathcal{B}^{-} \in P, \\
& P/At_{\mathcal{L}} \not\models C_1 \vee \cdots \vee C_n \text{ where} \\
& \mathcal{B}^{-} = \{C_1, \ldots, C_n\}, \\
& \mathcal{B}^{-\prime} = \{C \in \mathcal{B}^{-} \mid P \not\models_{\min(\mathrm{N})} \neg C\}\}
\end{aligned}
$$

Hence, the reduced program is obtained by

- removing each rule such that $C_1, \ldots, C_n$ are all the negative body literals and $C_1 \vee \cdots \vee C_n$ is a classical consequence of $P/At_{\mathcal{L}}$ and
- removing all $\neg(C)$ from the bodies of the remaining rules for which $\neg C$ is minimally entailed by the remaining program

Minimal Models: Tableau Method of I. Niemelä:

ordinary tableau + groundedness test.

For D-WFS: NAF-consistency has to be considered.

Test for NAF-consistency: a model is NAF-consistent for P iff

$$P \not\models A_1 \vee \cdots \vee A_m$$

where $\{\neg A_1, \ldots, \neg A_m\}$ are all the $\neg A$ atoms in the model.

This is one theorem prover call.

Theorem II.3.1 (D-WFS in Polynomial Space)

D-WFS can be implemented within a polynomial space bound.

Outlook

- How to incorporate Goal-directedness?
- How to get full STATIC and STABLE?
- Extension to first-order programs:
  Constraint Disjunctive Programs: DLP + CLP
  (Dix/Stolzenburg, forthcoming in a special issue of JLP on Constraints)

General Idea

The underlying idea behind semantics SEM* is to reduce the current program P into two programs without negative literals.

- use $P/At_{\mathcal{L}}$ for deriving positive information
- use $P/\neg At_{\mathcal{L}}$ for deriving negative information.

References

[Prz95] Teodor Przymusinski. Static Semantics For Normal and Disjunctive Logic Programs. *Annals of Mathematics and Artificial Intelligence*, 14:323–357, 1995.