Knowledge Representation and Reasoning in NMR and Extended LP

Jürgen Dix

University of Koblenz-Landau
Dept. of Computer Science

and

Technical University of Vienna
Institute for Knowledge-Based Systems

III. Automating NML's

- Sources of Complexity
- Finitary Characterizations
- Algorithm for DL
- Algorithm for CIRC
- Algorithm for Rational Closure

1 Sources of Complexity

The fundamental decision problems for default logic and nonmonotonic modal logics are

- *sceptical reasoning*,
- *credulous reasoning*, and
- deciding whether a set of premises has at least one extension.

First-Order Case

For first-order default logic these problems are not even semi-decidable (Reiter 1980).
General first-order circumscription is highly uncomputable, too (Schlipf 1986).

Propositional Case

Default logic Existence problem and credulous reasoning are Σ_2^P-complete.

Sceptical default reasoning (the dual problem) is Π_2^P-complete (Gottlob 1992; Stillman 1992).

Circumscription The problem of deciding whether a propositional sentence is a circumscriptive consequence of a set of propositional sentences is also a Π_2^P-complete problem (Eiter and Gottlob 1993; Eiter and Gottlob 1995).

Implications of the Results

1. Theorem proving systems for first-order nonmonotonic logics cannot provide a similar level of completeness as classical theorem provers: a query can be a nonmonotonic consequence of the premises but no nonmonotonic theorem prover is able to establish this.
2. Propositional Case: most important decision problems (sceptical and credulous consequence) are complete problems with respect to the second level of the polynomial time hierarchy.

3. NMR is strictly harder than classical reasoning (unless the polynomial time hierarchy collapses).

4. Two *independent* sources of complexity:

   (a) classical reasoning on top of which nonmonotonic reasoning is built,

   (b) handling of the nonmonotonic constructs.

1. Even with an oracle for classical reasoning, NMR would be NP-complete.

Tractable Subclasses

Not only restrict the syntactic form of underlying formulae, but also limit the interaction of the nonmonotonic constructs (stratification).

2 Finitary Characterizations

Theorem 2.1 (Generating Defaults)
Let E be an extension of a default theory $(\mathcal{D}, W)$. Then there is a set $\mathrm{GD} \subseteq \mathcal{D}$ such that $E = \mathrm{Cn}(W \cup \{\mathrm{cons}(d) \mid d \in \mathrm{GD}, \mathrm{pre}(d) \in E$ and for all $\psi \in \mathrm{just}(d), \neg\psi \notin E\})$.

- A set of defaults $\mathcal{D}$ is grounded in a set formulae W if and only if for all $d \in \mathcal{D}$, $\mathrm{pre}(d) \in \mathrm{Cn}_{\mathrm{Mon}(\mathcal{D})}(W)$ where $\mathrm{Mon}(\mathcal{D}) = \{\frac{\mathrm{pre}(d)}{\mathrm{cons}(d)} \mid d \in \mathcal{D}\}$.
- Extensions are closed under default rules, i.e. all applicable defaults are applied.

Theorem 2.2 (Extensions in Terms of Generating Defaults)
A set of formulae E is an extension of a default theory $(\mathcal{D}, W)$ if and only if $E = \mathrm{Cn}(W \cup \{\mathrm{cons}(d) \mid d \in \mathcal{D}'\})$ for a set of defaults $\mathcal{D}' \subseteq \mathcal{D}$ such that

1. *$\mathcal{D}'$ is grounded in W and*
2. *for all $d \in \mathcal{D}$:*

$$
\begin{array}{lll}
d \in \mathcal{D}' & \text{if and only if} & \mathrm{pre}(d) \in \mathrm{Cn}(W \cup \{\mathrm{cons}(d) \mid d \in \mathcal{D}'\}) \textit{ and} \\
 & & \textit{for all } \psi \in \mathrm{just}(d): \\
 & & \quad \neg\psi \notin \mathrm{Cn}(W \cup \{\mathrm{cons}(d) \mid d \in \mathcal{D}'\}).
\end{array}
$$

For a set of formulae F, $\mathrm{Mon}(\mathcal{D},\mathrm{F}) = \{\frac{\mathrm{pre}(d)}{\mathrm{cons}(d)} \mid d \in \mathcal{D}, \mathrm{just}(d) \subseteq \mathrm{F}\}$.
A set of justifications corresponding to an extension is called *full* with respect to the default theory.

Definition 2.3 (Full Sets)
For a default theory $(\mathcal{D},W)$, *a set of justifications* $\mathrm{F} \subseteq \mathrm{just}(\mathcal{D})$ *is called* $(\mathcal{D},W)$*-full if and only if the following condition holds:*

$$\textit{for all } \psi \in \mathrm{just}(\mathcal{D}),\ \psi \in \mathrm{F} \textit{ if and only if } \neg\psi \notin \mathrm{Cn}_{\mathrm{Mon}(\mathcal{D},\mathrm{F})}(W)$$

For each full set there is a corresponding extension and for each extension a full set that induces it.

Theorem 2.4 (Extensions in Terms of Full Sets)
Let $(\mathcal{D}, W)$ *a default theory.*

1. *If* $\mathrm{F} \subseteq \mathrm{just}(\mathcal{D})$ *is* $(\mathcal{D}, W)$*-full, then* $\mathrm{Cn}_{\mathrm{Mon}(\mathcal{D},\mathrm{F})}(W)$ *is an extension of* $(\mathcal{D}, W)$.
2. *If* E *is an extension of* $(\mathcal{D}, W)$*, then* $\mathrm{F} = \{B \in \mathrm{just}(\mathcal{D}) \mid \neg B \notin E\}$ *is* $(\mathcal{D}, W)$*-full and* $E = \mathrm{Cn}_{\mathrm{Mon}(\mathcal{D},\mathrm{F})}(W)$.

In parallel predicate circumscription the predicates are partitioned into three disjoint sets:

1. *minimized*, denoted by $\underline{p}$,
2. *varying*, denoted by $\underline{f}$, and
3. *fixed*, denoted usually by $\underline{z}$ (in this section we will simply omit them)

predicates.

The key notion in circumscription is that of a minimal model.

Definition 2.5 (Grounded Models)
Let T be a set of clauses and let $\underline{p}$ and $\underline{f}$ be sets of predicates. A Herbrand interpretation M is said to be grounded *in $\langle T, \underline{p}, \underline{f} \rangle$ if and only if for all ground atoms $p(\vec{t})$ such that $p \in \underline{p}$, $M \models p(\vec{t})$ implies $T \cup \mathrm{N}^{\langle \underline{p}, \underline{f} \rangle}(M) \models p(\vec{t})$ where*

$$\begin{aligned} \mathrm{N}^{\langle \underline{p}, \underline{f} \rangle}(M) \quad = \quad & \{\neg q(\vec{t}) \mid q(\vec{t}) \textit{ is a ground atom}, q \in \underline{p} \cup \underline{f}, M \not\models q(\vec{t})\} \cup \\ & \{q(\vec{t}) \mid q(\vec{t}) \textit{ is a ground atom}, q \in \underline{f}, M \models q(\vec{t})\} \end{aligned}$$

Theorem 2.6 (Minimal Models)
Let $\underline{p}$ and $\underline{f}$ be the sets of minimized and fixed predicates, respectively. A Herbrand interpretation M is a minimal model of T if and only if M is a model of T and grounded in $\langle T, \underline{p}, \underline{f} \rangle$.

3 Algorithm for Default Logic

S-answer:= Yes; C-answer:= No;
For every subset S of just$(\mathcal{D})$ **do**
  If S is $(\mathcal{D}, W)$-full **then**
    If $\varphi \in \mathrm{Cn}_{\mathrm{Mon}(\mathcal{D},S)}(W)$ **then**
      C-answer:= Yes
    else
      S-answer:= No
    end if
  end if

Figure 1: A Naive Decision Procedure for Credulous and Sceptical DL

function extensions$((\mathcal{D},W),\mathrm{FS},\varphi)$
$\mathrm{FS}' :=$ **expand**$((\mathcal{D},W),\mathrm{FS})$;
if conflict$((\mathcal{D},W),\mathrm{FS}')$ returns Yes **then** return No
else if FS' covers the negation of each justification in $\mathcal{D}$ **then**
    return **test**$((\mathcal{D},W),\mathrm{FS}',\varphi)$
else
    $\chi :=$ **choose**$((\mathcal{D},W),\mathrm{FS}')$;
    if extensions$((\mathcal{D},W),\mathrm{FS}'\cup\{\chi\},\varphi)$ returns Yes **then** return Yes
    else return **extensions**$((\mathcal{D},W),\mathrm{FS}'\cup\{\mathrm{compl}(\chi)\},\varphi)$
    end if
end if

Figure 2: An optimized decision procedure for default logic

1. Represent a partially constructed full set by using a set FS which is built from the *negations* of justifications and which explicates the current assumptions about their membership in the extension under consideration.

2. For representing the fact that a formula φ is not in the extension we employ what we call a not-formula: $\text{not}(\varphi)$.

   An expression $\text{not}(\neg\chi)$ in FS indicates that χ belongs to the full set to be constructed (and $\neg\chi$ does not belong to the corresponding extension).

   $\neg\chi$ in FS indicates that χ does not belong to the full set under consideration (and $\neg\chi$ belongs to the corresponding extension).

$$\mathrm{FS} = \{\mathrm{not}(\neg\neg p), \mathrm{not}(\neg q), \mathrm{not}(\neg\neg r), \neg s\},$$

corresponds to

$$\mathrm{F} = \{\neg p, q, \neg r\}$$

- A set of formulae S *agrees* with a set of formulae and not-formulae FS if for every ordinary formula $\varphi \in \mathrm{FS}$, $\varphi \in S$ and for all $\mathrm{not}(\varphi) \in \mathrm{FS}$, $\varphi \notin S$.
- A set of formulae and not-formulae FS *covers* a formula φ if either $\varphi \in \mathrm{FS}$ or $\mathrm{not}(\varphi) \in \mathrm{FS}$.
- $\mathrm{compl}(\chi)$: $\mathrm{compl}(\chi) = \mathrm{not}(\chi)$ and $\mathrm{compl}(\mathrm{not}(\chi)) = \chi$.

1. **extensions** performs *focused extension search* (determined by FS).
2. **extensions** starts by expanding cautiously the set FS to get a set FS$'$.
3. A conflict test is performed for FS$'$.

Soundness and completeness of the procedure can be guaranteed under quite weak assumptions about the three functions.

choose: **choose**$((\mathcal{D},W),\mathrm{FS})$ returns either $\neg\psi$ or not$(\neg\psi)$ for some justification ψ in $(\mathcal{D},W)$ for which $\neg\psi$ is not covered by FS.

expand: **expand**$((\mathcal{D},W),\mathrm{FS})$ returns a set FS' of formulae and not-formulae which extends FS and with which every extension of $(\mathcal{D},W)$ agreeing with FS agrees.

conflict: (i) If **conflict**$((\mathcal{D},W),\mathrm{FS})$ returns Yes, there is no extension agreeing with FS and (ii) if FS covers negations of every justification in $(\mathcal{D},W)$, and for some not$(\chi) \in \mathrm{FS}$, $\chi \in \mathrm{Cn}_{\mathrm{Mon}(\mathcal{D},\mathrm{FS})}(W)$ or for some $\chi \in \mathrm{FS}$, $\chi \notin \mathrm{Cn}_{\mathrm{Mon}(\mathcal{D},\mathrm{FS})}(W)$, then **conflict**$((\mathcal{D},W),\mathrm{FS})$ returns Yes.

Theorem 3.1 *Let $(\mathcal{D}, W)$ be a default theory and φ a formula.*

1. *Let* $\mathbf{test}((\mathcal{D}, W), \mathrm{FS}, \varphi)$ *return Yes for all* $(\mathcal{D}, W)$, FS, *and* φ.
   Then $\mathbf{extensions}((\mathcal{D}, W), \emptyset, \varphi)$ *returns Yes if and only if* $(\mathcal{D}, W)$ *has an extension.*

2. *Let* $\mathbf{test}((\mathcal{D}, W), \mathrm{FS}, \varphi)$ *return Yes if and only if* $\varphi \in \mathrm{Cn}_{\mathrm{Mon}(\mathcal{D},\mathrm{FS})}(W)$.
   Then $\mathbf{extensions}((\mathcal{D}, W), \emptyset, \varphi)$ *returns Yes if and only if there exists an extension of* $(\mathcal{D}, W)$ *containing* φ.

3. *Let* $\mathbf{test}((\mathcal{D}, W), \mathrm{FS}, \varphi)$ *return Yes if and only if* $\varphi \notin \mathrm{Cn}_{\mathrm{Mon}(\mathcal{D},\mathrm{FS})}(W)$.
   Then $\mathbf{extensions}((\mathcal{D}, W), \emptyset, \varphi)$ *returns Yes if and only if there exists an extension of T not containing* φ *(i.e.,* $\mathbf{extensions}((\mathcal{D}, W), \emptyset, \varphi)$ *returns No if and only if* φ *is contained in every extension of T).*

4 Algorithm for Circumscription

- How to build a circumscriptive theorem prover on top of a classical one? (Only a polynomial overhead is not possible).
- Build it on top of a classical model generating procedure, like tableau.
- But only the minimal models have to be considered. How can they be identified?

Let $\underline{p}$ and $\underline{f}$ be sets of minimized and fixed atoms, respectively. By Theorem 2.6 a model M of a set of clauses T is a minimal model of T if and only if for each minimized atom p true in model M, $T \cup \mathrm{N}^{\langle \underline{p},\underline{f} \rangle}(M) \models p$ holds where

$$\begin{aligned} \mathrm{N}^{\langle \underline{p},\underline{f} \rangle}(M) \quad &= \quad \{\neg q(\vec{t}) \mid q \in \underline{p} \cup \underline{f}, M \not\models q(\vec{t})\} \cup \\ & \quad\quad \{q(\vec{t}) \mid q \in \underline{f}, M \models q(\vec{t})\} \end{aligned}$$

This means that for establishing minimality of a model a single classical consequence test is enough: M is a minimal model of T if and only if

$$T \cup \mathrm{N}^{\langle \underline{p},\underline{f} \rangle}(M) \models p_1 \wedge \cdots \wedge p_n$$

where $p_1, \ldots, p_n$ are the minimized atoms true in M.

5 Implementing Rational Closure

Definition 5.1 ($\widetilde{\mathcal{K}}$)

Let $\widetilde{\mathcal{K}}$ be the set obtained from $\mathcal{K}$ by replacing all occurences of $\mid\sim$ with the material implication $\rightarrow$. Thus $\widetilde{\mathcal{K}}$ is a set of ordinary propositional formulas.

The main notion for defining an equivalent formulation of $\mathcal{K}^{\mathrm{rat}}$ is to reformulate the notion of exceptionality of a formula (see Definition 3.3). The following theorem gives us a simple test:

Theorem 5.2 (Testing Exceptionality)

Let $\mathcal{K}$ be a knowledge base. A formula α is exceptional for $\mathcal{K}$ if and only if $\widetilde{\mathcal{K}} \models \neg\alpha$.

References

Berman, K. A., J. S. Schlipf, and J. V. Franco (1995, June). Computing the Well-Founded Semantics Faster. In A. Nerode, W. Marek, and M. Truszczyński (Eds.), *Logic Programming and Non-Monotonic Reasoning, Proceedings of the Third International Conference*, LNCS 928, Berlin, pp. 113–126. Springer.

Brass, S. and J. Dix (1995, September). D-WFS: A Confluent calculus and an Equivalent Characterization. Technical Report TR 12/95, University of Koblenz, Department of Computer Science, Rheinau 1.

Brass, S. and J. Dix (1996). Characterizing D-WFS: Confluence and Iterated GCWA. In L. P. J.J. Alferes and E. Orlowska (Eds.), *Logics in Artificial Intelligence (JELIA '96)*, LNCS 1126, pp. 268–283. Springer. (Extended version will appear in the *Journal of Automated Reasoning* in 1998.).

Dix, J. and P. Schmitt (1993). Nichtmonotones Schließen: Wieviel Nichtmonotonie ist nötig? *Kognitionswissenschaft 3*, 53–69.

Dix, J. and F. Stolzenburg (1998). A Framework to incorporate Nonmonotonic Reasoning into Constraint Logic Programming. *Journal of Logic Program-*

ming 36(1,2,3), 5—37. Special Issue on *Constraint Logic Programming*, Guest Editors: Kim Marriott and Peter Stuckey.

Eiter, T. and G. Gottlob (1993). Propositional Circumscription and Extended Closed World Reasoning are Π_2^P-complete. *Theoretical Computer Science 144*(2), 231–245, Addendum: vol. 118, p. 315, 1993.

Eiter, T. and G. Gottlob (1995). On the Computational Cost of Disjunctive Logic Programming: Propositional Case. *Annals of Mathematics and Artificial Intelligence 15*(3/4), 289–323.

Gottlob, G. (1992). Complexity results for nonmonotonic logics. *Journal of Logic and Computation 2*(3), 397–425.

Lehmann, D. and M. Magidor (1992). What does a conditional knowledge base entail? *Artificial Intelligence 55*, 1–60.

Lifschitz, V. (1989, January). Benchmark Problems for Formal NMR, Version 2.00. In Reinfrank, de Kleer, Ginsberg, and Sandewall (Eds.), *Non-Monotonic Reasoning*, LNAI 346, Berlin, pp. 202–219. Springer.

Lloyd, J. W. (1987). *Foundations of Logic Programming*. Berlin: Springer. 2nd edition.

McCarthy, J. (1986). Applications of Circumscription to formalizing Common-Sense Reasoning. *Artificial Intelligence 28*, 89–116.

Niemelä, I. and P. Simons (1996, September). Efficient Implementation of the Well-founded and Stable Model Semantics. In M. Maher (Ed.), *Proceedings of the Joint International Conference and Symposium on Logic Programming*, Bonn, Germany, pp. 289–303. The MIT Press.

Reiter, R. (1980). A Logic for Default-Reasoning. *Artificial Intelligence 13*, 81–132.

Schlipf, J. (1986, June). How uncomputable is general circumscription? In *Proceedings of the Symposium on Logic in Computer Science*, Cambridge, USA, pp. 92–95. IEEE Computer Society Press.

Stillman, J. (1992, July). The complexity of propositional default logics. In *Proceedings of the 10th National Conference on Artificial Intelligence*, San Jose, California, USA, pp. 794–800. The MIT Press.

Tarjan, R. (1972). Depth first search and linear graph algorithms. *SIAM Journal on Computing 1*, 146–160.

Van Gelder, A. (1989, March). The Alternating Fixpoint of Logic Programs with Negation. In *Proc. of the Eight ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Philadelphia, Pennsylvania*, pp. 1–10. ACM Press.

Van Gelder, A., K. A. Ross, and J. S. Schlipf (1988). Unfounded Sets and well-founded Semantics for general logic Programs. In *Proceedings 7th Symposion on Principles of Database Systems*, pp. 221–230.