Equilibria in Heterogeneous Nonmonotonic Multi-Context Systems

Gerhard Brewka
Dept. of Computer Science
University of Leipzig
Johannisgasse 26, D-04103 Leipzig
brewka@informatik.uni-leipzig.de

Thomas Eiter
Institute of Information Systems
Vienna University of Technology
Favoritenstrasse 11, A-1040 Vienna
eiter@kr.tuwien.ac.at

Abstract

We propose a general framework for multi-context reasoning which allows us to combine arbitrary monotonic and nonmonotonic logics. Nonmonotonic bridge rules are used to specify the information flow among contexts. We investigate several notions of equilibrium representing acceptable belief states for our multi-context systems. The approach generalizes the heterogeneous monotonic multi-context systems developed by F. Giunchiglia and colleagues as well as the homogeneous nonmonotonic multi-context systems of Brewka, Serafini and Roelofsen.

Background and Motivation

Interest in formalizations of contextual information and inter-contextual information flow has steadily increased over the last years. Based on seminal papers by McCarthy (1987) and Giunchiglia (1993), several approaches have been proposed, most notably McCarthy's propositional logic of context (1993) and the multi-context systems of the Trento school devised by Giunchiglia and Serafini (1994).

Intuitively, a multi-context system describes the information available in a number of contexts (i.e., to a number of people/agents/databases/modules, etc.) and specifies the information flow between those contexts. The contexts themselves may be heterogeneous in the sense that they can use different logical languages and different inference systems, and no notion of global consistency is required. The information flow is modeled via so-called *bridge rules* which can refer in their premises to information from other contexts.

Figure 1 provides a simple illustration of the main underlying intuitions. Two agents, Mr.1 and Mr.2, are looking at a box from different angles. As some sections of the box are out of sight, both agents have partial information about the box. To express this information, Mr.1 only uses proposition letters l (there is a ball on the left) and r (there is a ball on the right), while Mr.2 also uses a third proposition letter c (there is a ball in the center). The two agents' reasoning may be based on different inference systems, and bridge rules model the information flow among them.

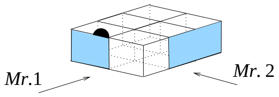

Figure 1: a magic box.

Almost all existing work in the field is based on classical, monotonic reasoning. The two exceptions we are aware of are (Roelofsen & Serafini 2005) and (Brewka *et al.* 2007).[1] To allow for reasoning based on the absence of information from a context, the authors of both papers add default negation to rule based multi-context systems and thus combine contextual and default reasoning. The former paper is based on a model theoretic approach where so-called information chains are manipulated. The latter is based on a multi-context variant of default logic (respectively its specialization to logic programs under answer set semantics).

Although these approaches are more general than Giunchiglia et al.'s multi-context systems because they allow for nonmonotonic rules, they are less general in another respect: they are homogeneous. Although different logical languages can be used in different contexts, the inference methods are the same in all contexts (default logic, respectively answer set logic programming).

In this paper, we go an important step further and present a framework for *heterogeneous nonmonotonic multi-context reasoning*. The proposed systems are capable of combining arbitrary monotonic and nonmonotonic logics. For instance, we can imagine a multi-context system where one context is a default logic reasoner, another one a description logic system, and a third one a logic program under answer set or well-founded semantics, or a circumscription engine.

The main contribution of this paper is thus a generalization of both heterogeneous monotonic and homogeneous nonmonotonic to heterogenenous nonmonotonic multi-context systems.

Formalization

In this section, we formalize heterogeneous nonmonotonic multi-context systems. We will simply call these systems multi-context systems (or MCSs) from now on. We first in-

Copyright © 2007, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1] The contextual default logic of (Besnard & Schaub 1995) addresses a different issue: unifying several variants of default logic in a single system. This is achieved through generalized defaults with different types of applicability conditions.

troduce MCSs, and then define several notions of equilibria for them. An equilibrium is a collection of belief sets for each context such that each belief set is based on the knowledge base of the respective context together with the information conveyed through applicable bridge rules.

As we will see, equilibria allow for certain forms of self-justification via the bridge rules. For this reason, we discuss minimal equilibria and finally grounded equilibria. The latter can only be defined for the subclass of reducible MCSs.

Multi-context systems

The idea behind heterogeneous MCSs is to allow different logics to be used in different contexts, and to model information flow among contexts via bridge rules. We first describe what we mean by a logic here.

We will characterize the syntax of a logic L by specifying the set $\mathbf{KB}_L$ of its well-formed knowledge bases. Without loss of generality, we assume the knowledge bases are sets. In cases where the standard definition of a knowledge base has more structure (like in default logic where default theories consist of a set of formulas and a set of defaults), we can always make sure that elements of different substructures can be distinguished syntactically.

We also need to describe L's possible belief sets $\mathbf{BS}_L$, that is sets of syntactical elements representing the beliefs an agent may adopt. In most cases this will be deductively closed sets of formulas, but we may have further restrictions, e.g. to sets of atoms, sets of literals, and the like.

We finally have to characterize acceptable belief sets an agent may adopt based on the knowledge represented in a knowledge base. For classical, monotonic logics this will normally just be the set of classical consequences of the knowledge base. However, in many of the typical rule-based nonmonotonic systems multiple acceptable belief sets may arise (called extensions, expansions, answer sets etc.). For this reason we will characterize acceptable belief sets using a function $\mathbf{ACC}_L$ assigning to each knowledge base kb a set of belief sets taken from $\mathbf{BS}_L$.

Definition 1 *A logic $L = (\mathbf{KB}_L, \mathbf{BS}_L, \mathbf{ACC}_L)$ is composed of the following components:*

1. *$\mathbf{KB}_L$ is the set of well-formed knowledge bases of L. We assume each element of $\mathbf{KB}_L$ is a set.*
2. *$\mathbf{BS}_L$ is the set of possible belief sets,*
3. *$\mathbf{ACC}_L : \mathbf{KB}_L \rightarrow 2^{\mathbf{BS}_L}$ is a function describing the "semantics" of the logic by assigning to each element of $\mathbf{KB}_L$ a set of acceptable sets of beliefs.*

Example 1 Let us first discuss some example logics, defined over a signature Σ:

- Default logic (DL) (Reiter 1980):
  - **KB**: the set of default theories based on Σ,
  - **BS**: the set of deductively closed sets of Σ-formulas,
  - $\mathbf{ACC}(kb)$: the set of kb's extensions.
- Normal logic programs under answer set semantics (NLP) (Gelfond & Lifschitz 1991):
  - **KB**: the set of normal logic programs over Σ,
  - **BS**: the set of sets of atoms over Σ,
  - $\mathbf{ACC}(kb)$: the set of kb's answer sets.
- Propositional logic under the closed world assumption:
  - **KB**: the set of sets of propositional formulas over Σ,
  - **BS**: the set of deductively closed sets of propositional Σ-formulas,
  - $\mathbf{ACC}(kb)$: the (singleton set containing the) set of kb's consequences under closed world assumption.

There are numerous other examples, nonmonotonic (e.g., circumscription, autoepistemic logic, defeasible logic) as well as monotonic (e.g., description logics, modal logics, temporal logics). We next define multi-context systems.

Definition 2 *Let $L = \{L_1, \ldots, L_n\}$ be a set of logics. An L_k-bridge rule over L, $1 \leq k \leq n$, is of the form*

$$\begin{aligned} s \leftarrow \quad & (r_1 : p_1), \ldots, (r_j : p_j), \\ & \textbf{\textit{not}}\ (r_{j+1} : p_{j+1}), \ldots, \textbf{\textit{not}}\ (r_m : p_m) \end{aligned} \tag{1}$$

where $1 \leq r_k \leq n$, p_k is an element of some belief set of L_{r_k}, and for each $kb \in \mathbf{KB}_k : kb \cup \{s\} \in \mathbf{KB}_k$.

Definition 3 *A multi-context system $M = (C_1, \ldots, C_n)$ consists of a collection of contexts $C_i = (L_i, kb_i, br_i)$, where $L_i = (\mathbf{KB}_i, \mathbf{BS}_i, \mathbf{ACC}_i)$ is a logic, kb_i a knowledge base (an element of $\mathbf{KB}_i$), and br_i is a set of L_i-bridge rules over $\{L_1, \ldots, L_n\}$.*

We use $head(r)$ to denote the head of a bridge rule r. Bridge rules refer in their bodies to other contexts and can thus add information to a context based on what is believed or disbelieved in other contexts. Note that in contrast with Giunchiglia's (monotonic) multi-context systems, in our systems there is no single, global set of bridge rules. To emphasize their locality, we add the bridge rules to those contexts to which they potentially add new information.

Equilibria

We now need to define the acceptable belief states a system may adopt. These belief states have to be based on the knowledge base of each context, but also on the information contained in other contexts in case there is an appropriate applicable bridge rule. Intuitively, we have to make sure that the chosen belief sets are in equilibrium: for each context C_i the selected belief set must be among the acceptable belief sets for C_i's knowledge base *together with the heads of C_i's applicable bridge rules*.

Definition 4 *Let $M = (C_1, \ldots, C_n)$ be an MCS. A belief state is a sequence $S = (S_1, \ldots, S_n)$ such that each S_i is an element of $\mathbf{BS}_i$.*

We say a bridge rule r of form (1) is applicable in a belief state $S = (S_1, \ldots, S_n)$ iff for $1 \leq i \leq j$: $p_i \in S_{r_i}$ and for $j + 1 \leq k \leq m$: $p_k \notin S_{r_k}$.

Definition 5 *A belief state $S = (S_1, \ldots, S_n)$ of M is an equilibrium iff, for $1 \leq i \leq n$, the following condition holds:*

$$S_i \in \mathbf{ACC}_i(kb_i \cup \{head(r) \mid r \in br_i \textit{ applicable in } S\}).$$

An equilibrium thus is a belief state which contains for each context an acceptable belief set, given the belief sets of the other contexts.[2]

[2] One can view each context as a player in an n-person game where players choose belief sets. Assume for a given profile

Minimal equilibria

Equilibria are not necessarily minimal (under componentwise set inclusion). They allow for a certain form of self-justification of elements of belief sets via bridge rules.

Example 2 Consider $M = (C_1)$ with a single context based on classical reasoning in propositional logic. Assume kb_1 is empty and br_1 consists of the bridge rule $p \leftarrow (1:p)$. Now both $(Th(\emptyset))$ and $(Th(\{p\}))$ are equilibria. The latter is somewhat questionable as p is justified by itself.

In the example (and whenever all contexts are monotonic, see below) we can solve the problem by considering minimal equilibria. However, there may be situations where minimality is unwanted, or at least not for all contexts. For instance, LPs with cardinality constraints (Niemelä & Simons 2000) use expressions of the form $j\{a_1, \dots, a_k\}k$ to state that the number of atoms in $\{a_1, \dots, a_k\}$ contained in an answer set must be between j and k. Answer sets thus may be nonminimal.

Similarly, stable expansions in autoepistemic logic (AEL) (Moore 1985) are not necessarily minimal: the premise set $\{Lp \rightarrow p\}$ has two stable expansions, one in which only AEL tautologies are believed, and one where in addition p is believed. If a context in an MCS is based on a logic whose acceptable belief sets are not necessarily minimal, restricting the attention to minimal equilibria may not be a good idea.

The following definition allows us to explicitly select those contexts for which minimality is required, and to keep other contexts fixed in the minimization:

Definition 6 *Let $M = (C_1, \dots, C_n)$ be an MCS, $S = (S_1, \dots, S_n)$ an equilibrium of M, C^* a subset of the contexts in M. S is called C^*-minimal iff there is no equilibrium $S' = (S'_1, \dots, S'_n)$ such that*

1. *$S'_i \subseteq S_i$ for all contexts $C_i \in C^*$,*
2. *$S'_i \subset S_i$ for some context $C_i \in C^*$,*
3. *$S'_i = S_i$ for all contexts $C_i \notin C^*$.*

There are also cases where minimality is not strong enough. Several nonmonotonic systems like default logic or logic programs under answer set semantics have a stronger notion of groundedness which cannot be captured by minimality. In particular, the contextual default logic approach described in (Brewka, Roelofsen, & Serafini 2007) is not a proper special case of the notion described here. The reason is that contextual default logic has a stronger groundedness condition: some forms of "self-justification" are possible even in minimal equilibria which were not allowed in the quoted paper.

Example 3 Consider the MCS $M = (C_1, C_2)$ consisting of two default logic contexts. Assume kb_1 and kb_2 are empty, and br_1 consists of the single bridge rule $p \leftarrow (2:q)$ and br_2 of two rules $q \leftarrow (1:p)$ and $r \leftarrow \textbf{not}\,(1:p)$. Now both $(Th(\{p\}), Th(\{q\}))$ and $(Th(\emptyset), Th(\{r\}))$ are minimal equilibria, but only the latter is a contextual extension in the sense of (Brewka *et al.* 2007). The former contains beliefs which justify themselves, but this cannot be excluded by requiring minimality.

Grounded equilibria

We now define grounded equilibria for a restricted class of so called *reducible* MCSs. Intuitively, an MCS is reducible if each component logic is either monotonic (each kb has a single acceptable belief set which grows monotonically when information is added to kb), or it is one of the nonmonotonic systems like default logic or logic programs where a candidate belief set S is tested for groundedness by transforming the original knowledge base kb to a simpler one kb'. S is accepted if it coincides with the belief set of kb'. The standard example is the Gelfond/Lifschitz reduction for logic programs. For bridge rules we will introduce a similar reduction.

Definition 7 *Let $L = (\boldsymbol{KB}_L, \boldsymbol{BS}_L, \boldsymbol{ACC}_L)$ be a logic. L is called monotonic iff*

1. *$\boldsymbol{ACC}_L(kb)$ is a singleton set for each $kb \in \boldsymbol{KB}_L$, and*
2. *$kb \subseteq kb'$, $\boldsymbol{ACC}_L(kb) = \{S\}$, and $\boldsymbol{ACC}_L(kb') = \{S'\}$ implies $S \subseteq S'$.*

Remark: an MCS can be nonmonotonic even if all involved logics are monotonic: its bridge rules can be nonmonotonic.

Definition 8 *Let $L = (\boldsymbol{KB}_L, \boldsymbol{BS}_L, \boldsymbol{ACC}_L)$ be a logic. L is called reducible iff*

1. *there is $\boldsymbol{KB}^*_L \subseteq \boldsymbol{KB}_L$ such that the restriction of L to $\boldsymbol{KB}^*_L$ is monotonic,*
2. *there is a reduction function $red_L : \boldsymbol{KB}_L \times \boldsymbol{BS}_L \rightarrow \boldsymbol{KB}^*_L$ such that for each $k \in \boldsymbol{KB}_L$ and $S, S' \in \boldsymbol{BS}_L$:*
   - *$red_L(k, S) = k$ whenever $k \in \boldsymbol{KB}^*_L$,*
   - *red_L is antimonotone in the second argument, that is $red_L(k, S) \subseteq red_L(k, S')$ whenever $S' \subseteq S$,*
   - *$S \in \boldsymbol{ACC}_L(k)$ iff $\boldsymbol{ACC}_L(red_L(k, S)) = \{S\}$.*

The conditions for red_L say the following: (a) k should not be further reduced if it is already in the target class $\mathbf{KB}^*_L$, (b) justifying the elements in a larger belief set cannot be made easier than justifying a smaller set by using a larger reduced knowledge base, and (c) acceptability of a belief set can actually be checked by the reduction.

Definition 9 *A context $C = (L, kb, br)$ is reducible iff*

1. *its logic L is reducible,*
2. *for all $H \subseteq \{head(r) | r \in br\}$ and belief sets S: $red_L(kb \cup H, S) = red_L(kb, S) \cup H$.*

An MCS is reducible if all of its contexts are. Note that a context is reducible whenever its logic L is monotonic. In this case $\mathbf{KB}^*$ coincides with $\mathbf{KB}$ and red_L is identity with respect to the first argument.

We can now define grounded equilibria. We follow ideas (and terminology) from logic programming:

Definition 10 *Let $M = (C_1, \dots, C_n)$ be a reducible MCS. M is called definite iff*

1. *none of the bridge rules in any context contains **not***
2. *for all $i, S \in \boldsymbol{BS}_i$: $kb_i = red_i(kb_i, S)$.*

(choice of belief sets by the players) each player's utility is 1 if she picks an acceptable belief set given her knowledge base and the heads of bridge rules applicable under the profile, and 0 otherwise. The Nash equilibria of this game coincide with our equilibria.

In a definite MCS bridge rules are monotonic, and knowledge bases are already in reduced form. Inference is thus monotonic and a unique minimal equilibrium exists. We take this equilibrium to be the grounded equilibrium:

Definition 11 *Let $M = (C_1, \ldots, C_n)$ be a definite MCS. $S = (S_1, \ldots, S_n)$ is the grounded equilibrium of M iff S is the unique minimal equilibrium of M.*

A more operational characterization of the grounded equilibrium of a definite MCS $M = (C_1, \ldots, C_n)$ is provided by the following proposition. For $1 \leq i \leq n$, let $kb_i^0 = kb_i$ and define, for each successor ordinal $\alpha + 1$,

$$kb_i^{\alpha+1} = kb_i^{\alpha} \cup \{head(r) \mid r \in br_i \text{ is applicable in } E^{\alpha}\},$$

where $E^{\alpha} = (E_1^{\alpha}, \ldots, E_n^{\alpha})$ and $\mathbf{ACC}_i(kb_i^{\alpha}) = \{E_i^{\alpha}\}$. Furthermore, for each limit ordinal α, define $kb_i^{\alpha} = \bigcup_{\beta \leq \alpha} kb_i^{\beta}$, and let $kb_i^{\infty} = \bigcup_{\alpha > 0} kb_i^{\alpha}$. Then we have:

Proposition 1 *Let $M = (C_1, \ldots, C_n)$ be a definite MCS. A belief state $S = (S_1, \ldots, S_n)$ is the grounded equilibrium of M iff $\mathbf{ACC}_i(kb_i^{\infty}) = \{S_i\}$, for $1 \leq i \leq n$.*

Remark: For many logics (in particular, those satisfying compactness), $kb_i^{\infty} = kb_i^{\omega}$ holds; that is, any bridge rule is applied (if at all) after finitely many steps; this is also trivial if br_i is finite. However, there are logics conceivable which require a transfinite number of steps.

The grounded equilibrium of a definite MCS M will be denoted $\mathbf{GE}(M)$. Grounded equilibria for general MCSs are defined based on a reduct which generalizes the Gelfond/Lifschitz reduct to the multi-context case:

Definition 12 *Let $M = (C_1, \ldots, C_n)$ be a reducible MCS, $S = (S_1, \ldots, S_n)$ a belief state of M. The S-reduct of M is*

$$M^S = (C_1^S, \ldots, C_n^S)$$

where for each $C_i = (L_i, kb_i, br_i)$ we define $C_i^S = (L_i, red_i(kb_i, S_i), br_i^S)$. Here br_i^S results from br_i by deleting

1. *every rule with some **not** $(k{:}p)$ in the body such that $p \in S_k$, and*
2. *all **not** literals from the bodies of remaining rules.*

For each MCS M and each belief set S, we have that M^S is definite. We can thus check whether S is a grounded equilibrium in the usual manner:

Definition 13 *Let $M = (C_1, \ldots, C_n)$ be a reducible MCS, $S = (S_1, \ldots, S_n)$ a belief state of M. S is a grounded equilibrium of M iff S is the grounded equilibrium of M^S, that is $S = \mathbf{GE}(M^S)$.*

We have the following result:

Proposition 2 *Every grounded equilibrium of a reducible MCS M is a minimal equilibrium of M.*

Examples of reducible nonmonotonic systems are default logic, NLPs, and LPs with cardinality constraints. For Reiter's default logic the reduction function $red(\Delta, S)$ takes a default theory Δ and eliminates all defaults with justification p such that $\neg p \in S$, and deletes all justifications from the remaining defaults. The resulting justification-free default theories are monotonic and satisfy the required conditions.

Example 4 Consider again MCS $M = (C_1, C_2)$ of Example 3. We saw earlier that among the two minimal equilibria $E_1 = (Th(\{p\}), Th(\{q\}))$ and $E_2 = (Th(\emptyset), Th(\{r\}))$ only the second one has no self-justified beliefs. It is easy to verify that $\mathbf{GE}(M^{E_1}) = (Th(\emptyset), Th(\emptyset)) \neq E_1$. Thus E_1 is not a grounded equilibrium, as intended. On the other hand, $\mathbf{GE}(M^{E_2}) = (Th(\emptyset), Th(\{r\})) = E_2$, and thus E_2 is the single grounded extension of M.

Both the multi-context systems by Giunchiglia & Serafini (1994) and contextual default logic by Brewka *et al.* (2007) are special cases of MCSs: the former have monotonic contexts, the latter use default logic in all contexts.

Well-founded Semantics

We can define a well-founded semantics for a reducible MCS M based on the operator $\gamma_M(S) = \mathbf{GE}(M^S)$, provided $\mathbf{BS}_i$ for each logic L_i in any of M's contexts has a least element S^*. We call MCSs satisfying this condition *normal*. The following result is a consequence of the antimonotony of red_i:

Proposition 3 *Let $M = (C_1, \ldots, C_n)$ be a reducible MCS. Then γ_M is antimonotone.*

We thus obtain a monotone operator by applying γ_M twice. Similar to what is done in logic programming we define the well-founded semantics of M as the least fixpoint of $(\gamma_M)^2$. This fixpoint exists according to the Knaster-Tarski theorem.

Definition 14 *Let $M = (C_1, \ldots, C_n)$ be a normal reducible MCS. The well-founded semantics of M, denoted **WFS**(M), is the least fixpoint of $(\gamma_M)^2$.*

The fixpoint can be computed by iterating $(\gamma_M)^2$ starting with the least belief state $S^\star = (S_1^\star, \ldots, S_n^\star)$. The grounded extensions of M and $\mathbf{WFS}(M)$ are related as usual:

Proposition 4 *Let $M = (C_1, \ldots, C_n)$ be a normal reducible multi-context system such that **WFS**$(M) = (W_1, \ldots, W_n)$. Let $S = (S_1, \ldots, S_n)$ be a grounded equilibrium of M. Then $W_i \subseteq S_i$ for $1 \leq i \leq n$.*

The well-founded semantics can thus be viewed as an approximation of the belief state representing what is accepted in all grounded equilibria. Note that $\mathbf{WFS}(M)$ itself is not necessarily an equilibrium.

WFS as defined here still suffers from a weakness discussed by (Brewka & Gottlob 1997) and (Brewka, Roelofsen, & Serafini 2007). Assume $\gamma_M(S^\star) = (T_1, \ldots, T_n)$ and one of the belief sets T_i is inconsistent and deductively closed. Then none of M's bridge rules referring to context C_i through a body literal **not** $(i : p)$ (for arbitrary p) will be used in the computation of $(\gamma_M)^2(S^\star)$. This may lead to overly cautious reasoning. As discussed in the quoted papers, this problem can be dealt with by considering modified operators, producing sets of formulas which are not deductively closed. Similar techniques can be used for MCSs.

Computational Complexity

In this section, we consider complexity aspects of MCSs. We focus on the problem of deciding whether an MCS $M = (C_1, \ldots, C_n)$ has an equilibrium, and on brave (resp., cautious) reasoning from its equilibria, i.e., given an element p, and some C_i, is $p \in S_i$ for some (resp., each) equilibrium $S = (S_1, \ldots, S_n)$ of M. We assume familiarity with the basics of complexity (see (Papadimitriou 1994)), in particular the polynomial hierarchy ($\Sigma^p_0 = \Pi^p_0 = \Delta^p_0 = P$, and, for all $k \geq 0$, $\Sigma^p_{k+1} = \mathrm{NP}^{\Sigma^p_k}$, $\Pi^p_{k+1} = \text{co-}\Sigma^p_{k+1}$, and $\Delta^p_{k+1} = \mathrm{P}^{\Sigma^p_k}$; here (N)P^C is (non)deterministic polynomial time with an oracle for C, and co-C is the complement of class C).

Let us say that a logic L has *poly-size kernels*, if there is a mapping κ which assigns to every $kb \in$ **KB** and $S \in$ **ACC**(kb) a set $\kappa(kb, S) \subseteq S$ of size (written as a string) polynomial in the size of kb, called the *kernel of* S, such that there is a one-to-one correspondence f between the belief sets in **ACC**(kb) and their kernels, i.e., $S \rightleftharpoons f(\kappa(kb, S))$. Standard propositional non-monotonic logics like DL, AEL, NLP, etc. all have poly-size kernels.

If furthermore, given any knowledge base kb, an element b, and a set of elements K, deciding whether (i) $K = \kappa(kb, S)$ for some $S \in$ **ACC**(kb) and (ii) $b \in S$ is in Δ^p_k, then we say that L has *kernel reasoning in* Δ^p_k.

Note that the standard propositional NMR formalisms DL and AEL have kernel reasoning in Δ^p_2, and under suitable restrictions even in Δ^p_1 = P, i.e., in polynomial time.

For convenience, we assume that any belief set S in any logic L contains a distinguished element **true**; so for $b =$ **true**, (i) and (ii) together are equivalent to (i), i.e., whether K is a kernel for some acceptable belief set of kb.

We concentrate on *finite MCS*, where all knowledge bases kb_i and sets of bridge rules br_i are finite; the L_i are from an arbitrary but fixed set. The following result gives an upper bound on deciding the existence of an equilibrium.

Theorem 1 *Given a finite MCS $M = (C_1, \ldots, C_n)$ where all logics L_i have poly-size kernels and kernel reasoning in Δ^p_k, deciding whether M has an equilibrium is in Σ^p_{k+1}.*

In particular, for an MCS in which the components are knowledge bases in an NMR formalism like propositional DL, AEL, NLP, etc. deciding the existence of an equilibrium is in the worst case not more complex than deciding the existence of an acceptable belief set in the component logics. This property, however, is not automatically inherited by all fragments of a logic, because the bridge rules might add complexity (e.g., if all logics are propositional Horn logic programs, but the bridge rules are unstratified).

Informally, Theorem 1 holds since we can guess kernels $\kappa_1, \ldots, \kappa_n$ of the belief sets $S_1, \ldots, S_n$ in an equilibrium S for M, together with the sets of heads $H_1, \ldots, H_n$ of bridge rules that are applicable in S, and check that each κ_i is the kernel of some $S_i \in$ **ACC**$(kb_i \cup H_i)$, and that H_i is correct; by hypothesis, checking is in Δ^p_k. On the other hand, we often get Σ^p_{k+1}-completeness via Σ^p_{k+1}-completeness of deciding belief set existence for the component logics; e.g., NP-completeness for NLPs, and Σ^p_2-completeness for DL and AEL. The following result is easy to see.

Theorem 2 *In the setting of Theorem 1, brave reasoning from the equilibria of a finite MCS M is in Σ^p_{k+1} and cautious reasoning is in Π^p_{k+1} = co-Σ^p_{k+1}.*

Again, completeness when using DL, AEL etc. in the components is inherited from respective reasoning results.

Brave reasoning from the minimal equilibria can be more complex than from all equilibria. Assuming that, in the setting of Theorem 1, given kernels κ and κ' of acceptable belief sets S and S' of kb, deciding whether $S \subseteq S'$ is in Δ^p_k brave reasoning is in Σ^p_{k+2} while cautious reasoning stays in Π^p_{k+1}(as usual). Again, completeness for Σ^p_{k+2} may be inherited from the complexity of minimal belief sets.

Grounded equilibria In case of a definite MCS C, we can take H_j as the kernel of the single acceptable belief set of $kb_i \cup H_i$. We then get the following result.

Theorem 3 *Let $M = (C_1, \ldots, C_n)$ be a (finite) definite MCS where all logics L_i have kernel reasoning (using kernels $\kappa_i(kb, S) = kb$) in Δ^p_k, $k \geq 1$. Then, the kernels κ_i of the belief sets S_i in **GE**(M) wrt. $kb_i \cup H_i$, $1 \leq i \leq n$, are computable in polynomial time with a Σ^p_{k-1} oracle, and brave/cautious reasoning is in Δ^p_k.*

In particular, for $k = 1$ no oracle is needed (as Σ^p_0 = P).

Indeed, we can compute the knowledge bases kb^0_i,kb^1_i,..., which are the kernels of the belief sets E^0_i, E^1_i,... with kernel reasoning; each step requires at most polynomially many inference tests for applicability, and the iteration stops after polynomially many steps (since M is finite).

A general reducible MCS M may have multiple grounded equilibria. Here, we can guess kernels of belief sets as above, and exploit Theorem 3.

Theorem 4 *Let $M = (C_1, \ldots, C_n)$ be a (finite) reducible MCS where each logic L_i has kernel reasoning in Δ^p_k. and where $red(kb_i, S_i)$ is computable, given a kernel of $S_i \in$ **ACC**$(kb_i \cup H_i)$, in polynomial time with a Σ^p_{k-1} oracle. Then (i) deciding whether M has a grounded equilibrium and brave reasoning from M are in Σ^p_{k+1}, and cautious reasoning is in co-Σ^p_k (=Π^p_k).*

Well-founded semantics In the setting of Theorem 4, computing $\gamma_M(S)$ = **GE**(M^S) is also feasible in polynomial time with a Σ^p_{k-1} oracle, and so is computing $(\gamma_M(S))^2 = \gamma_M(\gamma_M(S))$. Furthermore, the sequence $S^\star$, $(\gamma_M(S^\star))^2$, $(\gamma_M(S^\star))^4, \ldots$, can increase only polynomially often, since it reaches a fixpoint if no further bridge rule is applicable and M is finite. We thus get the following result:

Theorem 5 *Let $M = (C_1, \ldots, C_n)$ be a (finite) normal reducible MCS where all logics L_i have kernel reasoning in Δ^p_k, and where $red(kb_i, S_i)$ is computable, given a kernel for S_i (w.r.t. an arbitrary kb) resp. $S^\star_i$, in polynomial time with a Σ^p_{k-1} oracle. Then, deciding whether p is in a belief set in **WFS**(M) is in Δ^p_k as well.*

In particular, if all component logics in M admit polynomial kernel reasoning (like, e.g., NLPs), then reasoning under WFS is polynomial.

We finally remark that if kernel reasoning and computing $red(kb_i, S_i)$ is feasible in polynomial space, then all results hold with PSPACE in place of Δ^p_k resp. Σ^p_k.

Related Work

Our framework is related to *HEX programs* (Eiter *et al.* 2005), which generalize non-monotonic LPs by higher-order predicates and, more importantly here, by *external atoms* of the form $\#g[\vec{Y}](\vec{X})$ in rule bodies, where $\vec{Y}$ and $\vec{X}$ are respective lists of input and output terms. Intuitively, such an atom provides a way for deciding the truth value of an output tuple $\vec{X}$ depending on the extension of named input predicates $\vec{Y}$ using an external function g. It subsumes the generalized quantifier atoms proposed in (Eiter *et al.* 2000).

For example, $\#reach[edge, start](X)$ may single out in the graph specified by a binary predicate $edge$ all nodes X reachable from a node a given by $start(a)$; then, $\#reach[edge, start](b)$ is true iff b is reachable from a.

The semantics of a propositional HEX program P is defined in terms of answer sets, which are the interpretations I (viewing $\#g[\vec{y}](\vec{x})$ as propositional atom, where $\vec{y}$, $\vec{x}$ are lists of propositional atoms) that are minimal models of the *reduct* fP^I, which contains all rules from P whose body is true in I; here, $\#g[\vec{y}](\vec{x})$ is true in I iff an associated Boolean function $f_{\#g}$, which depends on I and $\vec{y}, \vec{x}$, returns 1; see (Eiter *et al.* 2005) for details.

HEX programs don't simply subsume our MCSs. It might seem that each $(i:p)$ can be emulated by an external atom $\#member_i[\,](a_p)$, where the atom a_p encodes p, which tells whether p belongs to an acceptable belief set of $kb_i \cup H_i$, where H_i are the heads of the applicable bridge rules in br_i. However, there are subtle yet salient differences:

1. A naive usage of such "brave reasoning" atoms is flawed, since occurring atoms $\#member_i[\,](a_p)$ and $\#member_i[\,](a_q)$ may use different belief sets S_i of $kb_j \cup H_j$ to witness $p \in S_i$ resp. $q \in S_i$. However, in an equilibrium, all such atoms have to use *the same* S_i.
2. Answer sets of HEX Programs are minimal (w.r.t. set inclusion), while equilibria are not necessarily minimal.

However, if each $S \in \mathbf{ACC}(kb_j \cup H_j)$ has a kernel $S \cap K_j$ for some (finite) set K_j (which applies to many standard propositional NMR formalisms), then we can overcome the first problem by "guessing" and verifying the kernel of the right belief set of $kb_i \cup H_i$ in the HEX-program. Item 2 can be handled by blocking minimization.

In the extended paper, we describe how to encode such an MCS M into a HEX program P_M such that its answer sets correspond to the equilibria of M. We further show how the grounded equilibria of a definite such MCS and a reducible such MCS can be encoded elegantly into HEX programs P^d_M respectively P^r_M. In particular, the encodings can be applied for many standard NMR formalisms. In this way, an implementation of an MCS with equilibria semantics can be designed, by providing suitable external functions implemented on top of existing reasoners for NMR formalisms.

Conclusion

Motivated by semantic web applications, there is increasing interest in combining ontologies based on description logics with nonmonotonic formalisms, cf. (Motik & Rosati 2007; Bonatti, Lutz, & Wolter 2006; Eiter *et al.* 2004). Sometimes this is achieved by embedding the combined formalisms into a single, more general formalism, e.g. MKNF in the case of (Motik & Rosati 2007).

Rather than focusing on two specific formalisms, our approach aims at providing a general framework for arbitrary logics. We leave entirely open which logics to use (and how many, for that matter), and we leave the logics "untouched": there is no unifying formalism to which we translate.

The multi-context systems developed in this paper substantially generalize earlier systems. They do not suffer from the limitations of both monotonic and homogeneous systems. We believe they can provide a useful, general framework for integrating reasoning formalisms of various kinds, both monotonic and nonmonotonic.

References

Besnard, P., and Schaub, T. 1995. An approach to context-based default reasoning. *Fund. Inf.* 23(2-4):175–223.

Bonatti, P. A.; Lutz, C.; and Wolter, F. 2006. Description logics with circumscription. In *Proc. KR-06*, 400–410.

Brewka, G., and Gottlob, G. 1997. Well-founded semantics for default logic. *Fund. Inf.* 31(3/4):221–236.

Brewka, G.; Roelofsen, F.; and Serafini, L. 2007. Contextual default reasoning. In *Proc. IJCAI-07*, 268–273.

Eiter, T.; Lukasiewicz, T.; Schindlauer, R.; and Tompits, H. 2004. Combining answer set programming with description logics for the semantic web. In *Proc. KR-04*, 141–151.

Eiter, T.; Ianni, G.; Schindlauer, R.; and Tompits, H. 2005. A uniform integration of higher-order reasoning and external evaluations in answer-set programming. In *Proc. IJCAI-05*, 90–96.

Eiter, T.; Gottlob, G.; and Veith, H. 2000. Generalized Quantifiers in Logic Programs. In Väänänen, J., ed., *Generalized Quantifiers and Computation Workshop at ESSLLI 97*, number 1754 in LNCS, 72–98. Springer.

Gelfond, M., and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Computing* 9(3/4):365–386.

Giunchiglia, F., and Serafini, L. 1994. Multilanguage hierarchical logics, or: how we can do without modal logics. *Artificial Intelligence* 65(1):29–70.

Giunchiglia, F. 1993. Contextual reasoning. *Epistemologia* XVI:345–364.

McCarthy, J. 1987. Generality in artificial intelligence. *Communications of ACM* 30(12):1030–1035.

McCarthy, J. 1993. Notes on formalizing context. In *Proc. IJCAI-93*, 555–560.

Moore, R. C. 1985. Semantical considerations on nonmonotonic logic. *Artificial Intelligence* 25(1):75–94.

Motik, B., and Rosati, R. 2007. A faithful integration of description logics with logic programming. In *Proc. IJCAI-07*, 477–482.

Niemelä, I., and Simons, P. 2000. Extending the Smodels system with cardinality and weight constraints. In Minker, J., ed., *Logic-Based Artificial Intelligence*. Kluwer Academic Pub. 491–521.

Papadimitriou, C. H. 1994. *Computational Complexity*. Addison-Wesley.

Reiter, R. 1980. A logic for default reasoning. *Artificial Intelligence* 13:81–132.

Roelofsen, F., and Serafini, L. 2005. Minimal and absent information in contexts. In *Proc. IJCAI-05*.