Contextual Default Reasoning

Gerhard Brewka
Universität Leipzig
Augustusplatz 10-11
04109 Leipzig, Germany
brewka@informatik.uni-leipzig.de

Floris Roelofsen
University of Amsterdam, ILLC
Nieuwe Doelenstraat 15
1012 Amsterdam, Netherlands
froelofs@science.uva.nl

Luciano Serafini
ITC-irst
Via Sommarive 18
38100 Trento, Italy
luciano.serafini@itc.it

Abstract

In this paper we introduce a multi-context variant of Reiter's default logic. The logic provides a syntactical counterpart of Roelofsen and Serafini's information chain approach (IJCAI-05), yet has several advantages: it is closer to standard ways of representing nonmonotonic inference and a number of results from that area come "for free"; it is closer to implementation, in particular the restriction to logic programming gives us a computationally attractive framework; and it allows us to handle a problem with the information chain approach related to skeptical reasoning.

1 Introduction

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 the propositional logic of context developed by McCarthy [1993] and McCarthy and Buvač [1998], and the multi-context systems devised by Giunchiglia and Serafini [1994], which later have been associated with the local model semantics introduced by Giunchiglia and Ghidini [2001]. Serafini and Bouquet [2004] have argued that multi-context systems constitute the most general among these formal frameworks.

Intuitively, a multi-context system describes the information available in a number of contexts (i.e., to a number of people/agents/databases, etc.) and specifies the information flow between those contexts. A simple illustration of the main intuitions underlying the multi-context system framework is provided by the situation depicted in Figure 1, one of the standard examples in the area. Two agents, Mr.1 and Mr.2, are looking at a box from different angles. The box is called magic, because neither Mr.1 nor Mr.2 can make out its depth. 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). To model situations of this kind, formulas are labeled with the contexts in which they hold, and so-called bridge rules are used to represent information flow.

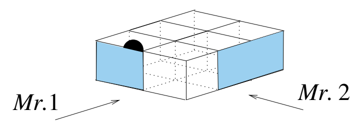

Figure 1: a magic box.

Most of the existing work in the field is based on classical, monotonic reasoning. The single exception we are aware of is [Roelofsen and Serafini, 2005]. To allow for reasoning based on the absence of information from a context, the authors add default negation to a rule based multi-context system and thus combine contextual and default reasoning.

This paper presents a related approach. We propose a contextual variant of Reiter's Default Logic DL [Reiter, 1980] called Contextual Default Logic (ConDL) which shares a lot of motivation with the Roelofsen/Serafini paper, in particular the basic idea of keeping information local for conceptual and computational reasons (as opposed to merging default theories [Baral *et al.*, 1994]). A major difference is that our description is syntactical rather than semantical. This has several advantages: from a computational perspective, it is more convenient to manipulate sets of formulas rather than sets of models; it allows us to link multi-context default reasoning more closely to earlier work in nonmonotonic reasoning; syntactic restrictions lead directly to contextual variants of logic programming under answer set and well-founded semantics and thus to a fully computational approach; and it paves the way to handle a serious weakness of the approach to skeptical reasoning developed in [Roelofsen and Serafini, 2005].

The outline of the paper is as follows: we first briefly review the approach of Roelofsen and Serafini and discuss the weakness of skeptical, well-founded reasoning in this approach. We then introduce ConDL and show that extensions of ConDL are in exact correspondence with stable information chains in [Roelofsen and Serafini, 2005]. We next show how well-founded reasoning can be defined for ConDL, escaping the difficulty of the information chain approach by appeal to paraconsistent reasoning. We finally discuss contextual logic programming and give various examples to illustrate that our formalism is indeed useful.

2 The information chain approach

We now give a brief review of the approach in [Roelofsen and Serafini, 2005]. The authors consider a set of contexts $\mathcal{C} = \{1, \ldots, n\}$ and a language L_i for each context $i \in \mathcal{C}$. $\mathcal{C}$ and L_i are assumed to be fixed, each L_i is built over a finite set of proposition letters, using standard propositional connectives.

To state that the information expressed by a formula $\varphi \in L_i$ is established in context i, the *labeled formula* $(i : \varphi)$ is used. A *rule* r is an expression of the form:

$$F \leftarrow G_1 \wedge \ldots \wedge G_m \wedge \textbf{not } H_1 \wedge \ldots \wedge \textbf{not } H_n \qquad (1)$$

where F, all G's, and all H's are labeled formulas. F is called the consequence of r and denoted by $cons(r)$; all G's are called *positive premises* of r and together constitute the set $prem^+(r)$; all H's are called *negative premises* of r and together make up the set $prem^-(r)$. A rule without premises is called a *fact*. If a rule has positive premises only, it is called a *positive* rule. A *normal multi-context system* is a finite set of rules. Note that **not** is interpreted as default negation, the rules are thus nonmonotonic.

Example 1 (Integration) *Let d_1, d_2 be two meteorological databases collecting data from sensors located in different parts of the country. Each database sends its data to a third database d_3, which integrates the information obtained. Suppose that d_3 regards d_1 as more trustworthy than d_2: any piece of information that is established in d_1 is included in d_3, but information obtained in d_2 is only included in d_3 if it is not refuted by d_1. The following rules model this:*

$$\begin{array}{rcl} 3 : \varphi & \leftarrow & 1 : \varphi \\ 3 : \varphi & \leftarrow & 2 : \varphi \wedge \textbf{not } 1 : \neg\varphi \end{array}$$

A classical interpretation m of language L_i is called a *local model* of context i. A set of local models is called a *local information state*. Intuitively, every local model in a local information state represents a possible state of affairs. If a local information state contains exactly one local model, then it represents complete information. If it contains more than one local model, then it represents partial information: more than one state of affairs is considered possible.

A *distributed information state* is a collection of local information states, one for each context. Distributed information states are referred to as *chains*. For systems without **not**, the semantics is defined in terms of minimal solution chains: starting with the set of all models for all contexts, rule application is captured semantically by eliminating those models from a context in which the consequent of an applicable rule is not true. Iterating this model elimination process until a fixpoint is reached yields the unique minimal solution chain.

For the general case, Roelofsen and Serafini use a technique similar to the Gelfond/Lifschitz reduction for stable models or answer sets [Gelfond and Lifschitz, 1988; 1991]: a rule r is defeated by an information chain $c = (c_1, \ldots, c_n)$ whenever it has a negative premise **not** $(i : p)$ such that p is true in all models in c_i. By eliminating all c-defeated rules and all negative premises from the c-undefeated rules, we obtain a reduced multi-context system without negative premises. Now c is a stable solution chain iff c is the minimal solution chain of the c-reduced system.

Based on the observation that stable solution chains may not exist, Roelofsen and Serafini also define a skeptical semantics which draws its intuitions from well-founded semantics for logic programs [van Gelder *et al.*, 1991]. It is based on the construction of the so-called canonical chain c_S. We present this semantics in somewhat more detail because it has a serious problem which we will later solve.

The canonical chain for a multi-context system S is constructed iteratively by applying an operator $\mathbf{\Psi}_S$ to a pair of chains $\langle c, a\rangle$. Intuitively, the first chain c approximates c_S from above: at every stage of the iteration it contains the models that are *possibly* in c_S (initially, every model may possibly be in c_S, so in each context we start with the set of all models). The second chain a, which is referred to as the *anti-chain*, approximates c_S from below: at every stage it contains the models that are *necessarily* in c_S (initially, no model is necessarily in c_S, so in each context we start with the empty set of models).

Given a certain chain-anti-chain pair $\langle c, a\rangle$, the intended transformation $\mathbf{\Psi}_S$ first determines which rules in S will (not) be applicable w.r.t. c_S, and then refines $\langle c, a\rangle$ accordingly. The canonical chain c_S of S will be the first component of the $\leq$-least fixpoint of $\mathbf{\Psi}_S$, where $\langle c, a\rangle \leq \langle c', a'\rangle$ iff for every i, $c'_i \subseteq c_i$ and $a_i \subseteq a'_i$ (intuitively, iff $\langle c, a\rangle$ is "less evolved" than $\langle c', a'\rangle$).

We first specify how $\mathbf{\Psi}_S$ determines which rules will (not) be applicable w.r.t. c_S. Let $\langle c, a\rangle$ and a rule r in S be given. If r has a positive premise G, which is satisfied by c, then G will also be satisfied by c_S. On the other hand, if r has a negative premise H, which is *not* satisfied by a, then H will not be satisfied by c_S either. So if all positive premises of r are satisfied by c and all negative premises of r are not satisfied by a, then r will be applicable with respect to c_S:

$$S^+(c, a) \quad = \quad \left\{ r \in S \;\middle|\; \begin{array}{c} \forall G \in prem^+(r) : c \models G \\ \text{and} \\ \forall H \in prem^-(r) : a \not\models H \end{array} \right\}$$

If r has a positive premise G, which is not satisfied by a, then G will not be satisfied by c_S either. If r has a negative premise H, which is satisfied by c, then H will be satisfied by c_S as well. In both cases r will certainly not be applicable with respect to c_S:

$$S^-(c, a) \quad = \quad \left\{ r \in S \;\middle|\; \begin{array}{c} \exists G \in prem^+(r) : a \not\models G \\ \text{or} \\ \exists H \in prem^-(r) : c \models H \end{array} \right\}$$

For convenience, we write $S^\sim(c, a) = S \setminus S^-(c, a)$. Think of $S^\sim(c, a)$ as the set of rules that are *possibly* applicable with respect to c_S, and notice that $S^+(c, a) \subseteq S^\sim(c, a)$.

Next, we specify how $\mathbf{\Psi}_S$ refines $\langle c, a\rangle$, based on $S^+(c, a)$ and $S^\sim(c, a)$. Every local model $m \in c_i$ that does not satisfy the consequence of a rule in $S^+(c, a)$ should certainly not be in c_S and is therefore removed from c. On the other hand, every local model $m \in c_i$ that satisfies the consequences of every rule in $S^\sim(c, a)$ should certainly be in c_S (S provides no ground for removing it) and is therefore added to a.

$$\mathbf{\Psi}_S(\langle c, a\rangle) \quad = \quad \langle \mathbf{\Psi}^c_S(\langle c, a\rangle), \mathbf{\Psi}^a_S(\langle c, a\rangle)\rangle$$

where:

$$\mathbf{\Psi}_S^c(\langle c, a\rangle) \quad = \quad c \setminus \{m \mid \exists r \in S^+(c,a) : m \not\models cons(r)\}$$
$$\mathbf{\Psi}_S^a(\langle c, a\rangle) \quad = \quad a \cup \{m \mid \forall r \in S^\sim(c,a) : m \models cons(r)\}$$

Unfortunately, this approach has a serious problem. Consider the following example:

$$\begin{aligned} 1\!:\!p &\leftarrow \textbf{not}\ 1\!:\!\neg p \\ 1\!:\!\neg p &\leftarrow \textbf{not}\ 1\!:\!p \\ 2\!:\!t &\leftarrow \textbf{not}\ 1\!:\!q \end{aligned}$$

One would expect $(2\!:\!t)$ to be derivable. However, the canonical chain approach does not give any conclusion. The problem is that no model can satisfy both p and $\neg p$, so no model will ever be added to the anti-chain a and thus it is never established that $(1\!:\!q)$ cannot be derived. The essential problem is this: the canonical model approach assumes that the set of possible conclusions is deductively closed. This is exactly the problem addressed in [Brewka and Gottlob, 1997] in the context of default logic. We will later show how the solution presented there can be applied to the problem of well-founded multi-context reasoning as well.

3 Contextual default logic

As before let $\mathcal{C} = \{1, \ldots, n\}$ be the set of contexts/agents with associated propositional languages L_i. A default context system for $\mathcal{C}$ is a tuple

$$(\Delta_1, \ldots, \Delta_n)$$

where each $\Delta_i = (D_i, W_i)$ is a contextual default theory. A contextual default theory is like a regular Reiter default theory, with the exception that default rules may refer in their prerequisites and justifications (not in their consequent!) to other contexts.

More precisely, a contextual default rule is of the form

$$d = p_1, \ldots, p_m : q_1, \ldots, q_k / r$$

where $p_1, \ldots, p_m, q_1, \ldots, q_k$ are regular formulas or labeled formulas, and the consequent r (also denoted $cons(d)$) is a regular formula. A contextual default theory (D_i, W_i) then is just a pair consisting of a set of regular formulas W_i (the certain knowledge) and a set of contextual default rules D_i. W_i and the unlabeled formulas in defaults have to be expressed in L_i. Each context thus has its own language for expressing its particular view of the world.

Note that if a default rule contains a regular formula, this formula is implicitly assumed to refer to the context of the default. We may thus assume without loss of generality that all prerequisites and justifications are labeled formulas. The reason we allow more than one prerequisite for a default – which is not necessary for Reiter's logic – is that we want to be able to refer to more than one context without using context labels inside logical formulas.

Now we can generalize the notion of an extension to default context systems. Given two tuples $(S_1, \ldots, S_n)$ and $(S'_1, \ldots, S'_n)$ we define component-wise inclusion $\subseteq_c$ as $(S_1, \ldots, S_n) \subseteq_c (S'_1, \ldots, S'_n)$ iff $S_i \subseteq S'_i$ for all i $(1 \leq i \leq n)$. When we speak of minimality of tuples in the rest of the paper we mean minimality with respect to $\subseteq_c$.

Definition 1 *Let $C = ((D_1, W_1), \ldots, (D_n, W_n))$ be a default context system. Let $(S_1, \ldots, S_n)$ be a tuple of sets of formulas. Define the operator Γ such that*

$$\Gamma(S_1, \ldots, S_n) = (S'_1, \ldots, S'_n)$$

where $(S'_1, \ldots, S'_n)$ is the minimal tuple of sets of formulas satisfying for all i $(1 \leq i \leq n)$:

1. *$W_i \subseteq S'_i$,*
2. *S'_i is deductively closed (over L_i), and*
3. *if $(c_1\!:\!p_1), \ldots, (c_t\!:\!p_t) : (c_{t+1}\!:\!q_1), \ldots, (c_{t+k}\!:\!q_k)/r \in D_i$, $p_i \in S'_{c_i}$ for all i $(1 \leq i \leq t)$, and $\neg q_j \notin S_{c_{t+j}}$ for all j $(1 \leq j \leq k)$, then $r \in S'_i$.*

The tuple $(S_1, \ldots, S_n)$ is a contextual extension of C if it is a fixpoint of Γ.

In the special case where default rules do not refer to other contexts, we obtain a tuple consisting of arbitrary extensions of the individual default theories. In the general case information flows, via the default rules, from one context to another. Defaults thus play the role of bridge rules.

It turns out that each extension corresponds exactly to a stable solution chain in the information chain approach. The translation between our default context systems and the systems used there (which we call RS-systems after their inventors from now on) is straightforward: each default

$$(c_1\!:\!p_1), \ldots, (c_t\!:\!p_t) : (c_{t+1}\!:\!q_1), \ldots, (c_{t+k}\!:\!q_k)/r$$

in D_i is translated to the rule

$$\begin{aligned} (i\!:\!r) \quad \leftarrow \quad & (c_1\!:\!p_1), \ldots, (c_t\!:\!p_t), \\ & \textbf{not}\ (c_{t+1}\!:\!\neg q_1), \ldots, \textbf{not}\ (c_{t+k}\!:\!\neg q_k) \end{aligned}$$

and each formula $p \in W_i$ to the rule $(i\!:\!p) \leftarrow$. We have the following proposition:

Proposition 1 *Let C be a default context system, R the corresponding RS-system. Let $S = (S_1, \ldots, S_n)$ be a sequence of deductively closed sets of formulas and $M = (M_1, \ldots, M_n)$ a sequence of sets of models such that for all i $(1 \leq i \leq n)$*

$$M_i = \{m \mid m \models S_i\}.$$

S is a contextual extension of C iff M is a stable solution chain of R.

We can thus view our approach based on contextual default logic as a syntactical characterization of the semantical approach in [Roelofsen and Serafini, 2005]. The advantage of our characterization is threefold: it is closer to standard approaches in nonmonotonic reasoning and allows us to transfer results which have been established for default logic quite easily to the multi-context case; it is more amenable to computation; it allows us to handle the difficulty of the semantical approach with respect to skeptical reasoning, as we will see in the next section.

As an example of the results we basically get "for free" we just mention the following:

Proposition 2 (Minimality)
Let E_1 and E_2 be extensions of a default context system C. If $E_1 \subseteq_c E_2$ then $E_1 = E_2$.

A normal default context system is one where each default in each context is of the form:

$$(c_1 : p_1), \ldots, (c_t : p_t) : r/r.$$

Proposition 3 (Existence)
Each normal default context system possesses at least one extension.

Proposition 4 (Consistency)
Let $C = ((D_1, W_1), \ldots, (D_n, W_n))$ be a default context system, $E = (E_1, \ldots, E_n)$ an extension of C. If all W_i are consistent and each default possesses at least one justification, then each E_j is consistent.

A lot more results for which we do not have space here carry over. For instance, we can give a quasi-inductive definition of extensions as in [Reiter, 1980]. We can define the notion of a stratified default context system for which a unique extension exists. Also complexity results carry over which establish that the main reasoning tasks for contextual default logic are on the second level of the polynomial hierarchy.

4 Skeptical contextual default reasoning

The essential problem of the canonical model approach is as follows: it assumes that the set of potential conclusions is deductively closed. Thus, whenever two conflicting formulas p and $\neg p$ are considered as potential conclusions, then this is also the case for an arbitrary formula q, even if q is entirely unrelated.

This is exactly the problem addressed in [Brewka and Gottlob, 1997] in the context of default logic. The solution is to apply paraconsistent reasoning in determining potential conclusions: both p and $\neg p$ are considered as possible conclusions, but not their deductive closure, i.e. not the set of all formulae. In the example discussed above, one should detect that $(1 : q)$ is not a possible conclusion because the only way to derive this labeled formula is based on an inconsistent set of potential conclusions. The semantics thus should derive $(2 : t)$.

In [Brewka and Gottlob, 1997] a sequence of different semantics was introduced which allows to trade-off the effort spent for consistency checking with the strength of skeptical inference. Rather than presenting the different semantics here, we focus on a single one (called WFS_2 in the cited paper) and directly describe its generalization to contextual default theories.

Definition 2 *Let $C = ((D_1, W_1), \ldots, (D_n, W_n))$ be a default context system. Let $D' = (D'_1, \ldots, D'_n)$ be a tuple of subsets of the defaults in C. Let p be a formula. A C-default proof for p from D' in context i is a finite sequence*

$$P = ((c_1 : d_1), \ldots, (c_m : d_m))$$

of context/default pairs such that the following conditions are satisfied:

1. *$d_j \in D'_{c_j}$, for all j $(1 \leq j \leq m)$,*
2. *$c_m = i$,*
3. *for each l and each prerequisite $(c : q)$ of d_l, q is a logical consequence of*
$$W_c \cup \{cons(d_k) \mid k < l, (c : d_k) \in P\},$$
4. *$W_i \cup \{cons(d_k) \mid (i : d_k) \in P\} \vdash p$.*

Let $S = (S_1, \ldots, S_n)$ be a sequence of sets of formulas, $D = (D_1, \ldots, D_n)$ a sequence of sets of contextual defaults. Define

$$D^S = (D'_1, \ldots, D'_n)$$

where D'_i is the set of defaults from D_i not defeated by S (d is defeated by S iff it has a justification $(i : q)$ such that $\neg q \in S_i$). With the notion of a default proof, we can express the Γ operator introduced above as follows: $\Gamma(S_1, \ldots, S_n) = (S'_1, \ldots, S'_n)$ iff each S'_i is the set of formulas possessing a default proof from D^S.

We will now define a similar operator Γ^*, but with an important restriction to consistent proofs. This will be sufficient to handle the problem described above.

Definition 3 *Let $P = ((c_1 : d_1), \ldots, (c_m : d_m))$ be a default proof, $S = (S_1, \ldots, S_n)$ a sequence of sets of formulas. We say P is S-consistent iff $S_i \cup \{cons(d_j) \mid (i : d_j) \in P\}$ is consistent, for all i $(1 \leq i \leq n)$.*

Now let $\Gamma^*(S_1, \ldots, S_n) = (S'_1, \ldots, S'_n)$ iff each S'_i is the set of formulas possessing a consistent default proof from D^S. Note that both Γ and Γ^* are antimonotone operators. Applying the two in sequence thus yields a monotone operation which has a least fixpoint. The least fixpoint can be reached by iterative applications of the two operators to the sequence consisting of empty sets only.

Definition 4 *Let $C = ((D_1, W_1), \ldots, (D_n, W_n))$ be a default context system. $S = (S_1, \ldots, S_n)$ is the well-founded conclusion set of C iff S is the least fixpoint of the operator $\Gamma\Gamma^*$.*

To see how this handles the problem consider the ConDL variant of the example discussed above. We have the contextual default theory $((D_1, W_1), (D_2, W_2))$ with $W_1 = W_2 = \emptyset$ and

$$D_1 = \{: p/p, : \neg p/\neg p\}$$
$$D_2 = \{: (1 : \neg q)/t\}.$$

Indeed, application of Γ^* to the sequence $S = (\emptyset, \emptyset)$ yields

$$S' = (Th(\{p\}) \cup Th(\{\neg p\}), Th(\{t\})).$$

Note that context 1 does not contain q. For this reason, applying Γ to S' gives us $(Th(\emptyset), Th(\{t\}))$. This is also a fixpoint and we establish t in context 2, as intended.

Based on a modification of a corresponding proof in [Brewka and Gottlob, 1997] we can show that well-founded semantics for contextual default theories is correct with respect to contextual extensions.

Proposition 5 (Correctness)
Let $C = ((D_1, W_1), \ldots, (D_n, W_n))$ be a default context system, $E = (E_1, \ldots, E_n)$ an extension of C and $S = (S_1, \ldots, S_n)$ the well-founded conclusion set of C. We have $S_i \subseteq E_i$ for all $i, 1 \leq i \leq n$.

5 Contextual ASP

A syntax restriction leads to contextual answer set programming (contextual ASP), respectively contextual logic programming under well-founded semantics. As before let $C =$

$\{1, \dots, n\}$ be a set of contexts/agents. A logic programming context system (LPCS) is a tuple $(P_1, \dots, P_n)$ where each P_i is a contextual logic program. A contextual logic program is a set of rules of the form

$$a \leftarrow b_1, \dots b_k, \textbf{not}\ b_{k+1}, \dots, \textbf{not}\ b_m$$

where a is a literal, each b_i is either a literal or a labeled literal of the form $(c : l)$ where c is a context and l a literal.

For LPCSs where **not** does not appear in the bodies of any rule (let's call them definite LPCSs), we can define the notion of a minimal context model:

Definition 5 *Let $C = (P_1, \dots, P_n)$ be a definite LPCS. An n-tuple of sets of literals $S = (S_1, \dots, S_n)$ is called the minimal context model of C iff S is the smallest n-tuple satisfying the following conditions:*

1. *$a \in S_i$ whenever $a \leftarrow (c_1 : b_1), \dots, (c_k : b_k) \in P_i$, $b_1 \in S_{c_1}$, ..., $b_k \in S_{c_k}$,*
2. *S_i is the set Lit_i of all literals in L_i whenever S_i contains a pair of complementary literals $l, \neg l$.*

The definition of stable model is now straightforward:

Definition 6 *Let $C = (P_1, \dots, P_n)$ be an (arbitrary) LPCS, and $S = (S_1, \dots, S_n)$ a tuple of sets of literals. The S-reduct of C, denoted C^S, is obtained from C by*

1. *deleting in each P_i all rules with body literal **not** $(c : l)$ such that $l \in S_c$,*
2. *deleting from all remaining rules in all programs P_i all default negated literals.*

Definition 7 *Let $C = (P_1, \dots, P_n)$ be an (arbitrary) LPCS, and $S = (S_1, \dots, S_n)$ a tuple of sets of literals. S is a stable context model of C iff it is the minimal context model of C^S.*

Well-founded semantics for LPCSs can be defined in the same spirit as for ConDL. However, consistency checking becomes much easier. For $C = (P_1, \dots, P_n)$ and a tuple of sets of literals $S = (S_1, \dots, S_n)$ let $\gamma(S)$ be the minimal context model of C^S. Define the minimal context set of a definite LPCS like the minimal context model, but without requirement 2 (inconsistent sets of literals do not have to be closed). Let operator $\gamma^*(S)$ produce the minimal context set of C^S. The operators γ and γ^* both are anti-monotone, the combined operator $\gamma\gamma^*$ is thus monotone and possesses a least fixpoint. We call this fixpoint the well-founded context model of C.

The use of this operator can be illustrated using our earlier example. We have the LPCS $C = (P_1, P_2)$ with

$$\begin{array}{lrl} P_1 : & p & \leftarrow \textbf{not}\ \neg p \\ & \neg p & \leftarrow \textbf{not}\ p \end{array}$$

and

$$P_2 : \quad t \quad \leftarrow \textbf{not}\ (1 : q)$$

Indeed, $\gamma^*(\emptyset, \emptyset) = (\{p, \neg p\}, \{t\})$. As in the case of contextual default logic, context 1 does not contain q. For this reason, applying γ to S' gives us $(\emptyset, \{t\})$. This is already a fixpoint and we establish t in context 2, as intended.

Contrary to well-founded semantics for contextual default logic, the computation time for well-founded semantics of LPCSs is polynomial: the number of iterations is bounded by the total number of literals in all contexts, and so is the time needed for each iteration.

6 Applications

In this section we illustrate the use of contextual logic programming with further examples. Our setting was propositional so far. In ASP it is common to use variables in rules as shorthand for the set of all ground instances of the rules. Users represent their knowledge in terms of programs with variables, a grounder (like *lparse*) then generates the purely propositional ground instantiation of the rules which is then passed on to an answer set solver like dlv [Leone *et al.*, 2002] or *smodels* [Simons *et al.*, 2002].

We will adopt and extend this use of variables for contextual logic programming. We assume three types of variables: term variables which are common in ASP and will be denoted by X, Y, possibly indexed; context variables denoted by C, possibly indexed; and proposition variables denoted by P, possibly indexed. Term variables are to be instantiated by ground terms, context variables by contexts (more precisely, integers denoting contexts), and proposition variables by ground literals. For convenience, we will also allow literals to appear as terms (strictly speaking we would have to distinguish between a proposition p and a term t_p representing this proposition; we assume the grounder is able to take care of this). As common in ASP we will also use rules with empty head of the form $\leftarrow body$ as abbreviation for $f \leftarrow \textbf{not}\ f, body$ where f is a symbol not appearing elsewhere in the program. The effect of the rule is that no answer set exists in which $body$ holds. With these conventions, it is easy to model several interesting multi-context scenarios.

Information fusion: Assume agent i decides to believe an arbitrary literal p whenever some other agent believes p and none of the agents believes $-p$ ($-p$ is the complement of p, that is $\neg p$ if p is an atom, and r if $p = \neg r$). This can be modeled by including in P_i the rules

$$\begin{array}{rcl} P & \leftarrow & (C : P), \textbf{not}\ rej(P) \\ rej(P) & \leftarrow & (C : -P) \end{array}$$

Again we assume the grounder handles the complement "$-$" adequately. Note that this representation implicitly guarantees that only information consistent with i's information is added since in case of conflict a proposition will be rejected.

One can also think of scenarios where agent i believes p whenever the majority of agents does so. Let $m = n + 1/2$ if n is odd, $m = n + 2/2$ otherwise. A corresponding rule is:

$$\begin{array}{rcl} P & \leftarrow & (C_1 : P), \dots, (C_m : P), \\ & & C_1 \neq C_2, C_1 \neq C_3, \dots, C_{m-1} \neq C_m. \end{array}$$

Game theory: We show how we can compute Nash equilibria for games in normal form using LPCSs. In general, we need to represent the choices available to each player, the best action given a particular choice of the other players, and a rule that says only the best action should be chosen.

Consider the famous prisoner's dilemma, a game involving 2 agents which can either cooperate (c) or defect (d). The gains obtained by the agents for each combination of choices are described in the following table:

	c	d
c	3,3	0,5
d	5,0	1,1

The single Nash equilibrium is obtained when both players defect. The game can be modeled as the 2-context system (P_1, P_2) where P_1 is

$$\begin{array}{rcl} choose(d) & \leftarrow & \textbf{not}\ choose(c) \\ choose(c) & \leftarrow & \textbf{not}\ choose(d) \\ best(d) & \leftarrow & (2 : choose(c)) \\ best(d) & \leftarrow & (2 : choose(d)) \\ & \leftarrow & choose(X), \textbf{not}\ best(X) \end{array}$$

and P_2 is as P_1 with context 2 replaced by 1. The single contextual answer set is

$$(\{choose(d), best(d)\}, \{choose(d), best(d)\})$$

and corresponds to the Nash equilibrium. In this fashion we can represent arbitrary games in normal form.

Social choice: So far we have assumed the logic programs representing contexts are so-called extended programs with two types of negation. Of course, we can also use other types of programs. A convenient language extension handled by the *smodels* system are cardinality constraints [Simons *et al.*, 2002] of the form $L\{a_1, \ldots, a_k\}U$. Here L and U are integers representing lower and upper bounds on the numbers of atoms a_j which are true in a model. Cardinality constraints can appear in the head or body of a rule and are highly convenient for many applications.

Without presenting the formal details, we want to mention that it is not difficult to base contextual answer set programming on such extended programs. Here is an example illustrating a possible use in social choice theory. Assume we have $n-1$ voters, each voter has a program describing candidates, and in particular which among the candidates she likes best. This information may be derived from preference criteria represented in the respective programs. We assume agent n is not a voter. Her role is to determine the winner based on the other agents' votes and a particular rule for selecting the winner. For example, in a simple majority vote we can use the program P_n (*con* stands for context, *cand* for candidate):

$$\begin{array}{rcl} votes(X, N) & \leftarrow & N\{(C : best(X)) : con(C)\}N, cand(X) \\ wins(X) & \leftarrow & \textbf{not}\ \neg wins(X) \\ \neg wins(X) & \leftarrow & votes(X, N), votes(Y, M), M > N \end{array}$$

The first rule says that candidate X has N votes if $best(X)$ holds in exactly N contexts. Other voting rules (like the Condorcet rule) can be represented in a similar way.

7 Conclusions

In an attempt to combine the fields of multi-context systems and nonmonotonic reasoning we introduced a multi-context variant of Reiter's default logic. Contextual default logic has several advantages over the information chain approach: it is closer to standard ways of representing nonmonotonic inference, which allows us to transfer a number of results from that area; it is closer to implementation, in particular the restriction to logic programming gives us a computationally attractive framework for nonmonotonic multi-context reasoning; and it allows us to handle a problem with the information chain approach related to skeptical reasoning. The examples we discussed suggest a number of interesting applications.

References

[Baral *et al.*, 1994] Chitta Baral, Sarit Kraus, Jack Minker, and V. S. Subrahmanian. Combining default logic databases. *Int. J. Cooperative Inf. Syst.*, 3(3):319–348, 1994.

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

[Gelfond and Lifschitz, 1988] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *International Conference on Logic Programming (ICLP 88)*, pages 1070–1080, 1988.

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

[Ghidini and Giunchiglia, 2001] C. Ghidini and F. Giunchiglia. Local models semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence*, 127(2):221–259, 2001.

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

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

[Leone *et al.*, 2002] N. Leone, G. Pfeifer, W. Faber, T. Eiter, G. Gottlob, S. Perri, and F. Scarcello. The DLV system for knowledge representation and reasoning. Technical report cs.AI/0211004, arXiv.org, November 2002.

[McCarthy and Buvač, 1998] J. McCarthy and S. Buvač. Formalizing context (expanded notes). In *Computing Natural Language*, volume 81 of *CSLI Lecture Notes*, pages 13–50. 1998.

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

[McCarthy, 1993] J. McCarthy. Notes on formalizing context. In *International Joint Conference on Artificial Intelligence (IJCAI 93)*, pages 555–560, 1993.

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

[Roelofsen and Serafini, 2005] F. Roelofsen and L. Serafini. Minimal and absent information in contexts. In *Proc. 19th International Joint Conference on Artificial Intelligence, IJCAI-05*, 2005.

[Serafini and Bouquet, 2004] L. Serafini and P. Bouquet. Comparing formal theories of context in AI. *Artificial Intelligence*, 155:41–67, 2004.

[Simons *et al.*, 2002] P. Simons, I. Niemelä, and T. Soininen. Extending and implementing the stable model semantics. *Artificial Intelligence*, 138(1-2):181–234, 2002.

[van Gelder *et al.*, 1991] A. van Gelder, K. Ross, and J. S. Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):620–650, 1991.