General Default Logic

Yi Zhou[1], FangZhen Lin[2], and Yan Zhang[1]

[1] School of Computing and Mathematics
University of Western Sydney
Penrith South DC NSW 1797, Australia
[2] Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong

Abstract. In this paper, we present a logic $\mathcal{R}$ for rule bases by introducing a set of rule connectives. We define both the models and extensions of a rule base. The semantics of extensions intuitively captures all possible beliefs which can be derived from a rule base. We show that this logic is a generalization of Reiter's default logic [1] and Gelfond *et al.*'s disjunctive default logic [2] in propositional case. We also show that this logic is a generalization of Ferraris's general logic programming [3]. Finally, we demonstrate that this logic is flexible enough to capture several important situations in common sense reasoning.

1 Introduction

In this paper, we consider a language with connectives from both classical logic and logic programming, and provide a fixed-point semantics for the language so that our logic is both a generalization of Reiter's default logic [1] and Ferraris's general logic programs with stable model semantics [3].

The interplay between Reiter's default logic and logic programming with negation-as-failure has been going on since the beginning of nonmonotonic reasoning. Reiter's default logic was considered to be a formalization of default reasoning including the negation-as-failure mechanism used in Prolog. This is confirmed when Gelfond and Lifschitz [4] showed that their stable model semantics of normal logic programs [5] can be embedded in Reiter's default logic. In a normal logic program, the head of a rule must be an atom. If one allows disjunctions of atoms in the head of rule, then one obtains what has been called disjunctive logic programs, and Gelfond *et al.* [2] showed that the answer set semantics of these disjunctive logic programs can be embedded in their disjunctive default logic. Notice that Reiter's default logic also allows disjunction in the formulas occurring in default rules, but this disjunction is different from the disjunction used in disjunctive logic programming. One could say that the former is *classical* disjunction as in classical logic while the latter is *default* disjunction. The difference is that default disjunction means *minimality* while classical disjunction does not. For instance, the logic program $\{a|b \leftarrow\}$ has two answer sets

$\{a\}$ and $\{b\}$, but the default theory $(\{\}, \{: a \vee b / a \vee b\})$ has one extension $\{a \vee b\}$ which has three models[3] $\{a\}$, $\{b\}$, and $\{a, b\}$.

Another difference between disjunction in disjunctive logic programs and disjunction in formulas in classical logic is that the former is allowed only in the head of a rule. It cannot occur in the body of a rule nor can it be nested. This limitation in the use of disjunction in disjunctive logic programming has been addressed by Lifschitz *et al.* [6], Pearce [7] and recently by Ferraris [3]. He defined logic programs that look just like propositional formulas but are interpreted according to a generalized stable model semantics. In other words, in these formulas called general logic programs, negations are like negation-as-failure, implications are like rules, and disjunctions are like those in disjunctive logic programs. While Ferarris's general logic programs go well-beyond disjunctive logic programs by allowing disjunctions, negation-as-failure, and rules to occur anywhere, they do not allow any classical disjunctions and implications. Thus the natural question is whether a meaningful logic can be defined that allow both connectives from logic programming and classical logic. This paper answers this question positively. Before we proceed to define our logic formally, the following example illustrates the need for classical disjunction in default reasoning.

Example 1. Suppose that we have the following information about the students in a high school:

(*) A student good at math is normally good at physics. Conversely, a student good at physics is normally good at math.

In logic programming, this can be represented as follows:

1. $GoodAtPhysics(x) \leftarrow GoodAtMath(x)$, $\texttt{not}\, \neg GoodAtPhysics(x)$[4];
2. $GoodAtMath(x) \leftarrow GoodAtPhysics(x)$, $\texttt{not}\, \neg GoodAtMath(x)$.

Now suppose we are told that Mike is either good at math or good at physics. If we represent this disjunctive information as:

3. $GoodAtMath(Mike) \mid GoodAtPhisics(Mike)$

then we would conclude that Mike is both good at math and good at physics as the logic program $\{1, 2, 3\}$ has a unique answer set. One could argue whether this is a reasonable conclusion. If we do not wish the defaults to be applied here, then we could replace (3) by the following fact:

3'. $GoodAtMath(Mike) \vee GoodAtPhisics(Mike)$.

As we shall see, in our logic, the theory representing $\{1, 2, 3'\}$ will have a unique extension $\{GoodAtMath(Mike) \vee GoodAtPhisics(Mike)\}$.

[3] We identify a model with the set of atoms that are true in the model.
[4] We use a first order notation here to denote the set of all grounded propositional rules.

2 General Default Logic

In this section, we define our logic $\mathcal{R}$ that allow both classical connectives and logic programming connectives. Then, we show that Reiter's default logic, Gelfond *et al.*'s disjunctive default and Ferraris's general logic programs are special cases of the extension semantics of general default logic.

2.1 Syntax and basic semantics

Let *Atom* be a set of atoms, also called propositional variables. By $\mathcal{L}$ we mean the classical propositional language defined recursively by *Atom* and classical connectives $\bot$, $\neg$, $\rightarrow$ as follows:

$$F ::= \bot \mid p \mid \neg F \mid F \rightarrow F,$$

where $p \in \mathit{Atom}$. $\top$, $F \wedge G$, $F \vee G$ and $F \leftrightarrow G$ are considered as shorthands of $\bot \rightarrow \bot$, $\neg(F \rightarrow \neg G)$, $\neg F \rightarrow G$ and $(F \rightarrow G) \wedge (G \rightarrow F)$ respectively, where F and G are formulas in $\mathcal{L}$. Formulas in $\mathcal{L}$ are called *facts*. The satisfaction relation between a set of facts and a fact is defined as usual. A *theory* T is a set of facts which is closed under the classical entailment. Let Γ be a set of facts, $Th(\Gamma)$ denotes the logic closure of Γ under classical entailment. We write Γ to denote the theory $Th(\{\Gamma\})$ if it clear from the context. For instance, we write $\emptyset$ to denote the theory of all tautologies; we write $\{p\}$ to denote the theory of all logic consequences of p. We say that a theory T is *inconsistent* if there is a fact F such that $T \models F$ and $T \models \neg F$, otherwise, we say that T is *consistent*.

We introduce a set of new connectives, called *rule connectives*. They are $-$ for *negation as failure* or *rule negation*, $\Rightarrow$ for *rule implication*, $\&$ for *rule and*, $\mid$ for *rule or* and $\Leftrightarrow$ for *rule equivalence* respectively. We define a propositional language $\mathcal{R}$ recursively by facts and rule connectives as follows:

$$R ::= F \mid R \Rightarrow R \mid R \,\&\, R \mid R \,|\, R,$$

where F is a fact. $-R$ and $R \Leftrightarrow S$ are considered as shorthands of $R \Rightarrow \bot$ and $(R \Rightarrow S) \,\&\, (S \Rightarrow R)$ respectively, where R and S are formulas in $\mathcal{R}$. Formulas in $\mathcal{R}$ are called *rules* or *rule formulas*. Particularly, facts are also rules. A *rule base* Δ is a set of rules.

The order of priority for these connectives are

$$\{\neg\} > \{\wedge, \vee\} > \{\rightarrow, \leftrightarrow\} > \{-\} > \{\,\&\,, \,|\,\} > \{\Rightarrow, \Leftrightarrow\}.$$

For example, $-p \vee \neg p \Rightarrow q$ is a well defined rule, which denotes the rule $(-(p \vee (\neg p))) \Rightarrow q$. Both $\neg p \vee p$ and $\neg p \,|\, p$ are well defined rules. The former is also a fact but the latter is not a fact. However, $\neg(p \,|\, p)$ is not a well defined rule.

We define the *subrule* relationship between two rules recursively as follows:

1. R is a subrule of R;
2. R and S are subrules of $R \Rightarrow S$;

3. R and S are subrules of $R \,\&\, S$;
4. R and S are subrules of $R \mid S$,

where R and S are rules. Thus, clearly, R is a subrule of $-R$. For example, p is a subrule of $\neg p \mid p$ but not a subrule of $\neg p \vee p$.

We now define the *satisfaction relation* $\models_R$ between a theory and a rule inductively:

– If R is a fact, then $T \models_R R$ iff $T \models R$.
– $T \models_R R \,\&\, S$ iff $T \models_R R$ and $T \models_R S$;
– $T \models_R R \mid S$ iff $T \models_R R$ or $T \models_R S$;
– $T \models_R R \Rightarrow S$ iff $T \not\models_R R$ or $T \models_R S$.

Thus, $T \models_R -R$ iff $T \models_R R \rightarrow \bot$ iff $T \not\models_R R$ or $T \models_R \bot$. If T is consistent, then $T \models_R -R$ iff $T \not\models_R R$. If T is inconsistent, then for every rule R, $T \models_R R$. $T \models_R R \Leftrightarrow S$ iff $T \models_R (R \Rightarrow S) \,\&\, (S \Rightarrow R)$ iff $T \models_R R \Rightarrow S$ and $T \models_R S \Rightarrow R$.

We say that T *satisfies* R, also T is a *model* of R iff $T \models_R R$. We say that T satisfies a rule base Δ iff T satisfies every rule in Δ. We say that two rule bases are *weakly equivalent* if they have the same set of models.

For example, let T be $\emptyset$. T is a model of $\neg p \vee p$, but T is not a model of $\neg p \mid p$. This example also shows a difference between the two connectives $\vee$ and $\mid$. As another example, T is a model of $-p$ but not a model of $\neg p$. This example also shows a difference between the two connectives $\neg$ and $-$.

Theorem 1. *Let T be a theory and R, S two rule formulas in $\mathcal{R}$.*

1. $T \models_R - - R$ *iff* $T \models_R R$.
2. $T \models_R -(R \,\&\, S)$ *iff* $T \models_R -R \mid -S$.
3. $T \models_R -(R \mid S)$ *iff* $T \models_R -R \,\&\, -S$.
4. $T \models_R R \Rightarrow S$ *iff* $T \models_R -R \mid S$.

Proof. These assertions follow directly from the definitions. As an example, we write down the proof of assertion 2 here. According to the definition, $T \models_R -(R \,\&\, S)$ iff T is not a model of $R \,\&\, S$, which holds iff a) T is not a model of R or b) T is not a model of S. On the other hand, $T \models_R -R \mid -S$ iff a) T is a model of $-R$ or b) T is a model of $-S$, which holds iff a) T is not a model of R or b) T is not a model of S. Hence, assertion 2 holds.

2.2 Extension

Let T be a theory and R a rule in $\mathcal{R}$. The reduction of R on T, denoted by R^T, is the formula obtained from R by replacing every maximal subrules of R which is not satisfied by T with $\bot$. It can also be defined recursively as follows:

– If R is a fact, then $R^T = \begin{cases} R \text{ if } T \models_R R \\ \bot \text{ otherwise} \end{cases}$,
– If R is $R_1 \odot R_2$, then $R^T = \begin{cases} R_1^T \odot R_2^T & \text{if } T \models_R R_1 \odot R_2 \\ \bot & \text{otherwise} \end{cases}$, where $\odot$ is $\&$, $\Leftrightarrow$, $\mid$ or $\Rightarrow$.

Thus, if R is $-S$ and T is consistent, then $R^T = \begin{cases} \bot \text{ if } T \models_R S \\ \top \text{ otherwise} \end{cases}$.

Let T be a theory and Δ a rule base, the reduction of Δ on T, denoted by Δ^T, is the set of all the reductions of rules in Δ on T.

For example, let T be $\{p\}$. The reduction of $-p \Rightarrow q$ on T is $\bot \Rightarrow \bot$, which is weakly equivalent to $\top$. The reduction of $p \mid q$ on T is p. The reduction of $p \vee q$ on T is $p \vee q$. This example also shows that although two rules are weakly equivalent (e.g. $-p \Rightarrow q$ and $p \mid q$), their reductions on a same theory might not be weakly equivalent.

This notion of reduction is similar to Ferraris's notion of reduction for general logic programs [3].

Definition 1. *Let T be a theory and Δ a rule base. We say that T is an extension of F iff:*

1. $T \models_R \Delta^T$.
2. *There is no theory T_1 such that $T_1 \subset T$ and $T_1 \models_R \Delta^T$.*

Example 2. Consider the rule base $\Delta_1 = \{-p \Rightarrow q\}$.

- Let T_1 be $\emptyset$. The reduction of Δ_1 on T_1 is $\{\bot\}$. T_1 is not a model of $\Delta_1^{T_1}$. Hence, T_1 is not an extension of Δ_1.
- Let T_2 be $\{p\}$. The reduction of Δ_1 on T_2 is $\bot \Rightarrow \bot$. T_2 is a model of $\Delta_1^{T_2}$. But $\emptyset$ is also a model of $\Delta_1^{T_2}$ and $\emptyset \subset T_2$. Hence, T_2 is not an extension of Δ_1.
- Let T_3 be $\{q\}$. The reduction of Δ_1 on T_3 is $\top \Rightarrow q$, which is weakly equivalent to q. T_3 is a model of q and there is no proper subset of T_3 which is also a model of q. Hence, T_3 is an extension of Δ_1.
- Let T_4 be $\{\neg p \wedge q\}$. The reduction of Δ_1 on T_4 is $\top \Rightarrow q$, which is also weakly equivalent to q. T_4 is a model of $\Delta_1^{T_4}$. But $\{q\}$ is also a model of $\Delta_1^{T_4}$. Hence T_4 is not an extension of Δ_1.
- We can examine that the only extension of Δ_1 is T_3.

Similarly, $p \mid q$ has two extensions: $\{p\}$ and $\{q\}$. $p \vee q$ has a unique extension $\{p \vee q\}$. This example shows that although two rules are weakly equivalent, their extensions might not be the same (e.g. $-p \Rightarrow q$ and $p \mid q$).

Example 3 (Example 1 continued). Consider the example mentioned in the introduction section again. Reformulated in general default logic with domain $D = \{Mike\}$, the rule base $1, 2, 3'$, denoted by Δ, is:

1. $GoodAtMath(Mike) \& -\neg GoodAtPhysics(Mike) \Rightarrow GoodAtPhysics(Mike)$;
2. $GoodAtPhysics(Mike)\ \&\ -\neg GoodAtMath(Mike) \Rightarrow GoodAtMath(Mike)$;
3'. $GoodAtMath(Mike) \vee GoodAtPhisics(Mike)$.

Let T_1 be the theory $Th(\{GoodAtMath(Mike), GoodAtPhysics(Mike)\})$. The reduction of Δ on T_1 is

1-1 $GoodAtMath(Mike) \Rightarrow GoodAtPhysics(Mike)$;

1-2 $GoodAtPhysics(Mike) \Rightarrow GoodAtPhysics(Mike)$;
1-3 $GoodAtMath(Mike) \vee GoodAtPhysics(Mike)$.

Although T_1 is a model of Δ^{T_1}, $GoodAtMath(Mike) \vee GoodAtPhysics(Mike)$ is also a model of Δ^{T_1} and it is a proper subset of T_1. Thus, T_1 is not a extension of Δ. And we can examine the only extension of Δ is $\{GoodAtMath(Mike) \vee GoodAtPhysics(Mike)\}$.

It is clear that if Δ is a set of facts, then Δ has exactly one extension, which is the deductive closure of itself.

Intuitively, the extensions of a rule base represent all possible beliefs which can be derived from the rule base. The following theorem shows that every extension of a rule base is also a model of it.

Theorem 2. *Let T be a consistent theory and Δ a rule base. If T is an extension of Δ, then T satisfies Δ.*

Proof. According to the definitions, it is easy to see that $T \models_R \Delta^T$ iff $T \models_R \Delta$. On the other hand, if T is an extension of Δ, then $T \models_R \Delta^T$. Hence, this assertion holds.

The converse of Theorem 2 does not hold in general. For instance, $\{p\}$ is a model of $\{- - p\}$, but not an extension of it.

2.3 Default logic

In this paper, we only consider Reiter's default logic in propositional case. A default rule has the form

$$p : q_1, \ldots, q_n / r,$$

where p, $q_i, 1 \leq i \leq n$ and r are propositional formulas. p is called the prerequisite, $q_i, 1 \leq i \leq n$ are called the justifications and r is called the consequent. A default theory is a pair $\Delta = (W, D)$, where W is a set of propositional formulas and D is a set of default rules. A theory T is called an extension of a default theory $\Delta = (W, D)$ if $T = \Gamma(T)$, where for any theory S, $\Gamma(S)$ is the minimal set (in the sense of subset relationship) satisfying the following three conditions:

1. $W \subseteq \Gamma(S)$.
2. $\Gamma(S)$ is a theory.
3. For any default rule $p : q_1, \ldots, q_n / r \in D$, if $p \in \Gamma(S)$ and $\neg q_i \notin S, 1 \leq i \leq n$, then $r \in \Gamma(S)$.

We now show that Reiter's default logic in propositional case can be embeded into the logic $\mathcal{R}$. Let R be a default rule with the form $p : q_1, \ldots, q_n / r$. By R^* we denote the following rule in $\mathcal{R}$

$$p \,\&\, -\neg q_1 \,\&\, \ldots \,\&\, -\neg q_n \Rightarrow r.$$

Let $\Delta = (W, D)$ be a default theory, by Δ^* we denote the rule base

$$W \cup \{R^* \mid R \in D\}.$$

Theorem 3. *Let T be a theory and $\Delta = (W, D)$ a default theory. T is an extension of Δ iff T is an extension of Δ^*.*

Proof. $\Rightarrow$: Suppose that T is an extension of Δ. Then $T \models_R W^T$ since W is a set of facts. Moreover, for all rule $R \in D$ with the form $p : q_1, \ldots, q_n/r$. There are three cases:

- $p \notin T$. In this case, R^{*T} is weakly equivalent to $\top$. Thus, $T \models_R R^{*T}$.
- There is a $q_i, 1 \leq i \leq n$ such that $\neg q_i \in T$. In this case, R^{*T} is also weakly equivalent to $\top$. Thus, $T \models_R R^{*T}$.
- $p \in T$ and there is no $q_i, 1 \leq i \leq n$ such that $\neg q_i \in T$. In this case, according to the definition of extensions in default logic, $r \in T$. Therefore, R^{*T} is weakly equivalent to $p \Rightarrow r$. Hence, $T \models_R R^{*T}$.

This shows that for all $R \in D$, $T \models_R R^{*T}$. Hence, $T \models_R \Delta^{*T}$. On the other hand, there is no consistent theory $T_1 \subset T$ and $T_1 \models_R \Delta^{*T}$. Otherwise, suppose there is such a T_1. T_1 must satisfy W since $W \subseteq \Delta^{*T}$. For all rule $R \in D$, T_1 satisfies R^{*T}. Therefore, T_1 satisfies the third condition in the definition of default extensions. Therefore, $\Gamma(T) \subseteq T_1$. Hence, $\Gamma(T) \neq T$. This shows that T is not an extension of Δ, a contradiction.

$\Leftarrow$: Suppose that T is an extension of Δ^*. We now show that T is the smallest theory satisfying condition 1 to 3 in the definition of default extensions. First, $T \models_R W$ since $W \subseteq \Delta^*$ and W is a set of facts. Second, T is a theory. Finally, for all rule $R \in \Delta$ with the form $p : q_1, \ldots, q_n/r$, if $p \in T$ and there is no $q_i, 1 \leq i \leq n$ such that $\neg q_i \in T$, then R^{*T} is $p \Rightarrow r$. And $T \models_R R^{*T}$, therefore $r \in T$. This shows that T satisfies all those conditions. Now suppose otherwise there is a proper subset T_1 of T also satisfies Condition 1 to 3. Then, similarly $T_1 \models_R W$ and for all rule $R \in D$, $T_1 \models_R R^{*T}$. Thus, $T_1 \models_R \Delta^{*T}$. This shows that T is not an extension of Δ^*, a contradiction.

Observe that R and R^* are essential the same except the syntax to represent them. Thus, Reiter's default logic in propositional case is a special case of general default logic. In the rest of this section, we shall also show that Gelfond *et al.*'s disjunctive default logic is a special case of $\mathcal{R}$ by a similar translation.

A disjunctive default rule has the form

$$p : q_1, \ldots, q_n/r_1, \ldots, r_k,$$

where p, $q_i, 1 \leq i \leq n$ and $r_j, 1 \leq j \leq k$ are propositional formulas. A disjunctive default theory is a pair $\Delta = (W, D)$, where W is a set of propositional formulas and D is a set of disjunctive default rules. A theory T is called an extension of a disjunctive default theory $\Delta = (W, D)$ if $T = \Gamma(T)$, where for any theory S, $\Gamma(S)$ is the minimal set (in the sense of subset relationship) satisfying the following three conditions:

1. $W \subseteq \Gamma(S)$.
2. $\Gamma(S)$ is a theory.

3. For any default rule $p : q_1, \dots, q_n / r_1, \dots, r_k \in D$, if $p \in \Gamma(S)$ and $\neg q_i \notin S, 1 \leq i \leq n$, then for some $j, 1 \leq j \leq k$, $r_j \in \Gamma(S)$.

We now show that Gelfond *et al.*'s default logic in propositional case can be embeded into the logic $\mathcal{R}$ as well. Let R be a disjunctive default rule with the form $p : q_1, \dots, q_n / r_1, \dots, r_k$. By R^* we denote the following rule in $\mathcal{R}$

$$p \,\&\, -\neg q_1 \,\&\, \dots \,\&\, -\neg q_n \Rightarrow r_1 \mid \dots \mid r_k.$$

Let $\Delta = (W, D)$ be a disjunctive default theory, by Δ^* we denote the rule base

$$W \cup \{R^* \mid R \in D\}.$$

Theorem 4. *Let T be a theory and $\Delta = (W, D)$ a disjunctive default theory. T is an extension of Δ iff T is an extension of Δ^*.*

Proof. This proof is quite similar with the proof of Theorem 3.

2.4 General logic programming

Ferraris's general logic programs are defined over propositional formulas. Given a propositional formula F and a set of atoms X, the reduction of F on X, denoted by F^X, is the proposition formula obtained from F by replacing every subformula which is not satisfied by F into $\bot$. Given a set of propositional formulas Δ, Δ^X is the set of all reductions of formulas in Δ on X. A set of atoms X is said to be a stable model of Δ iff X is the minimal set (in the sense of subset relationship) satisfying Δ^X.

Let F be a propositional formula. By F^* we denote the formula in $\mathcal{R}$ obtained from F by replacing every classical connectives into corresponding rule connectives, that is, from $\rightarrow$ to $\Rightarrow$, from $\neg$ to $-$, from $\wedge$ to $\&$, from $\vee$ to $\mid$ and from $\leftrightarrow$ to $\Leftrightarrow$. Let Δ be a general logic program, by Δ^* we denote the rule base

$$\{F^* \mid F \in \Delta\}.$$

Lemma 1. *Let X be a set of atoms and F a propositional formula. $Th(X)$ is a model of F^* iff X is a model of F.*

Proof. We prove this assertion by induction on the structure of F.

1. If F is $\top$ or $\bot$, it is easy to see that this assertion holds.
2. If F is an atom p, then $Th(X) \models_R F^*$ iff $p \in X$ iff X is a model of F.
3. If F is $\neg G$, then $Th(X) \models_R F^*$ iff $Th(X) \models_R -G^*$ iff $Th(X)$ is not a model of G^* iff X is not a model of G iff X is a model of $\neg G$.
4. If F is $G \wedge H$, then $Th(X) \models_R F^*$ iff $Th(X) \models_R G^* \,\&\, H^*$ iff $Th(X)$ is a model of G^* and $Th(X)$ is a model of H^* iff X is a model of G and X is a model of H iff X is a model of $G \wedge H$.
5. If F is $G \vee H$, then $Th(X) \models_R F^*$ iff $Th(X) \models_R G^* \mid H^*$ iff $Th(X)$ is a model of G^* or $Th(X)$ is a model of H^* iff X is a model of G or X is a model of H iff X is a model of $G \vee H$.

6. If F is $G \rightarrow H$, then $Th(X) \models_R F^*$ iff $Th(X) \models_R G^* \Rightarrow H^*$ iff $Th(X)$ is not a model of G^* or $Th(X)$ is a model of H^* iff X is not a model of G or X is a model of H iff X is a model of $G \rightarrow H$.
7. If F is $G \leftrightarrow H$, then $Th(X) \models_R F^*$ iff $Th(X) \models_R G^* \Leftrightarrow H^*$ iff a) $Th(X)$ is both a model of G^* and a model of H^* or b) $Th(X)$ is neither a model of G^* nor a model of H^* iff a) X is both a model of G and a model of H or b) X is neither a model of G and nor a model of H iff X is a model of $G \leftrightarrow H$.

This completes the induction proof.

Theorem 5. *Let X be a set of atoms and Δ a general logic program. X is a stable model of Δ iff $Th(X)$ is an extension of Δ^*.*

Proof. By Lemma 1, it is easy to see that $(\Delta^X)^*$ is the same as $(\Delta^*)^{Th(X)}$.

$\Rightarrow$: Suppose X is a stable model of Δ. Then X is the minimal set satisfying Δ^X. By Lemma 1, $Th(X)$ is a model of $(\Delta^X)^*$. Thus $Th(X)$ is a model of $(\Delta^*)^{Th(X)}$. And there is no proper subset T_1 of $Th(X)$ such that $T_1 \models_R (\Delta^*)^{Th(X)}$. Otherwise, T_1 is a model of $(\Delta^X)^*$. Let X_1 be the set of atoms $\{p \mid T_1 \models p\}$. By induction on the structure, it is easy to see that for any set of propositional formulas Γ, T_1 is a model of Γ^* iff $Th(X_1)$ is a model of Γ^*. Hence, $Th(X_1)$ is a model of $(\Delta^X)^*$. Therefore by Lemma 1, X_1 is a model of Δ^X. Moreover $X_1 \subset X$ since $T_1 \subset Th(X)$. This shows that X is not a stable model of Δ, a contradiction.

$\Leftarrow$: Suppose $Th(X)$ is an extension of Δ^*. Then $Th(X)$ is the minimal set satisfying $(\Delta^*)^{Th(X)}$. Therefore, $Th(X)$ is the minimal set satisfying $(\Delta^X)^*$. By Lemma 1, it is easy to see that X is the minimal set satisfying Δ^X. Therefore, X is a stable model of Δ.

Actually, although Ferraris used the notations of classical connectives to denote the connectives in general logic programs, those connectives are still connectives in answer set programming. They are essentially rule connectives. In this paper, we use a set of rule connectives to denote them. Hence, the answer set semantics for general logic programs is also a special case of general default logic. Moreover, it is a special case of general default logic which only allows the facts are atoms, while general logic programming with strong negation (namely classical negation) is also a special case of general default logic which allows the facts are literals.

In [4], Gelfond and Lifschitz showed that the answer set semantics for normal logic programs is a special case of Reiter's default logic; in [2], Gelfond *et al.* showed that the answer set semantics for disjunctive logic programs is a special case of their disjunctive default logic. Together with this work, one can observe that the series of semantics for answer set programs (with classical negation) are essentially special cases of corresponding semantics for default logics which restrict the facts into atoms (literals).

3 Applications

In this section, we show that this logic is flexible enough to represent several important situations in common sense reasoning, including rule constraints, general closed world assumptions and conditional defaults.

3.1 Representing rule constraints

Similar to constraints in answer set programming, constraints in general default logic eliminate the extensions which do not satisfy the constraints. Let R be a rule, the constraint of R can be simply represented as

$$-R.$$

Theorem 6. *T is an extension of $\Delta \cup \{-R\}$ iff T is an extension of Δ and $T \not\models_R R$.*

Proof. $\Rightarrow$: Suppose that T is an extension of $\Delta \cup \{-R\}$. Then by the definition, $T \models_R (-R)^T$. Thus, $T \models_R -R$. Therefore $T \not\models_R R$. On the other hand, $T \models_R \Delta^T$. We only need to prove that T is a minimal theory satisfying Δ^T. Suppose otherwise, there is a theory T_1 such that $T_1 \subset T$ and $T_1 \models_R \Delta^T$. Notice that $(\Delta \cup \{-R\})^T$ is $\Delta^T \cup \{(-R)^T\}$, which is $\Delta^T \cup \{\top\}$. Thus, $T_1 \models_R (\Delta \cup \{-R\})^T$. This shows that T is not an extension of $\Delta \cup \{-R\}$, a contradiction.

$\Leftarrow$: Suppose that T is an extension of Δ and $T \not\models_R R$. Then T is the minimal theory satisfying Δ^T. Thus, T is also the minimal theory satisfying $(\Delta \cup \{-R\})^T$ since $(-R)^T$ is $\top$. Therefore, T is an extension of $\Delta \cup \{-R\}$.

3.2 Representing general closed world assumptions

In answer set programming, given an atom p, closed world assumption for p is represented as follows:

$$\neg p \leftarrow \texttt{not}\, p.$$

Reformulated in general default logic, it is

$$-p \Rightarrow \neg p.$$

However, this encoding of closed world assumption may lead to counterintuitive effects when representing incomplete information. Consider the following example [8, 9].

Example 4. Suppose we are give the following information:

(*) If a suspect is violent and is a psychopath then the suspect is extremely dangerous. This is not the case if the suspect is not violent or not a psychopath.

This statement can be represented (in general default logic) as three rules:

1. $\mathit{violent} \,\&\, \mathit{psychopath} \Rightarrow \mathit{dangerous}.$

2. $\neg violent \Rightarrow \neg dangerous.$
3. $\neg psychopath \Rightarrow \neg dangerous.$

Let us also assume that the DB has complete positive information. This can be captured by closed world assumption. In the classical approach, it can be represented as follows:

4. $-violent \Rightarrow \neg violent.$
5. $-psychopath \Rightarrow \neg psychopath.$

Now suppose that we have a disjunctive information that a person is either violent or a psychopath. This can be represented as:

6. $violent \,|\, psychopath.$

It is easy to see that the rule base $1-6$ [5] has two extensions:

$$Th(\{\neg violent, psychopath, \neg dangerous\});$$

$$Th(\{violent, \neg psychopath, \neg dangerous\}).$$

Thus, we can get a result $\neg dangerous$. Intuitively, this conclusion is too optimistic.

In our point of view, the reason is that the closed world assumption (4 and 5) are too strong. It should be replaced by

7. $-(violent \vee psychopath) \Rightarrow \neg(violent \vee psychopath).$

We can see that $1-3, 6, 7$ has two extensions

$$Th(\{violent\}) \text{ and } Th(\{psychopath\}).$$

Here, the answer of query *dangerous* is unknown.

Generally, given a fact F, the general closed world assumption of F can be represented as (in general default logic)

$$-F \Rightarrow \neg F.$$

Given a rule base Δ such that $-F \Rightarrow \neg F \in \Delta$, it is clear that if a theory T is an extension of Δ and $T \not\models F$, then $T \models \neg F$.

[5] There are fours kinds of formalization for this example in general default logic. It depends on the way of representing the conjunctive connective in 1 and the way of representing the disjunctive connective in 6. This kind of formalization is a translation from the representation in disjunctive logic program. However, all these four kinds of formalization are fail to capture the sense of this example if the classical approach of closed world assumption is adopted.

3.3 Representing conditional defaults

A conditional rule has the following form:

$$R \Rightarrow S,$$

where R and S are rules. R is said to be the condition and S is said to be the body. Of course, this yields a representation of conditional defaults in Reiter's default logic.

Let us consider the following example about New Zealand birds from [10]. We shall show how we can represent it using conditional defaults in a natural way.

Example 5. Suppose that we have the following information:

(*) Birds normally fly. However, in New Zealand, birds normally do not fly.

One can represent this information in Reiter's default logic as follows:

d_1 : $bird : fly \,/\, fly$;
d_2 : $bird \wedge newzealand : \neg fly \,/\, \neg fly$.

Given the fact

1. $bird$,

the default theory $(1, \{d_1, d_2\})$ has exactly one extension $Th(\{bird, fly\})$. However, given the fact

2. $newzealand, bird$,

the default theory $(2, \{d_1, d_2\})$ has two extensions $Th(\{bird, newzealand, fly\})$ and $Th(\{bird, newzealand, \neg fly\})$.

In [10], Delgrande and Shaub formalized this example by using dynamic priority on defaults. We now show that the information $(*)$ can be represented by using conditional defaults in a natural way as follows:

3. $newzealand \Rightarrow (bird \,\&\, -fly \Rightarrow \neg fly)$.
4. $-newzealand \Rightarrow (bird \,\&\, -\neg fly \Rightarrow fly)$.

We can see that the rule base $1, 3, 4$ still has exactly one extension $Th(\{bird, fly\})$, and the rule base $2, 3, 4$ has a unique extension $Th(\{newzealand, bird, \neg fly\})$.

4 Conclusion

We have proposed a general default logic, called $\mathcal{R}$. It extends Reiter's default logic by adding rule connectives, and Ferraris's general logic program by allowing arbitrary propositional formulas to be the base in forming logic programs. We also show that this logic is flexible enough to capture several important situations in common sense reasoning.

Just as Lin and Shoham [11] showed that propositional default logic can be embedded in the logic of GK, a non-standard modal logic with two modal operators K for knowledge and A for assumption, and Lin and Zhou [12] showed that Ferraris's general logic programs can be embedded in the logic of GK, it is possible to show that our new logic $\mathcal{R}$ can also be embedded in the logic of GK. One potential benefit of doing so would be to obtain a way to check strong equivalence in $\mathcal{R}$ in classical logic. Another important task is to compare the expressive power of $\mathcal{R}$ with other non-monotonic formalisms. It is also possible to extend our logic $\mathcal{R}$ to allow facts to be first order sentences as in Reiter's default logic. We leave these to future work.

Acknowledgement

The research of the first and third authors is supported in part by Australian Research Council (ARC) Discovery Projects grant DP0666540. The second author is supported in part by a Croucher Senior Research Fellowship and by HK RGC under grant HKUST CERG 616806. The authors would like to thank the anonymous reviewers for their useful comments.

References

1. Reiter, R.: A logic for default reasoning. Artificial Intelligence **13** (1980) 81–132
2. Gelfond, M., Lifschitz, V., Przymusinska, H., Truszczyński, M.: Disjunctive Defaults. In Allen, J., Fikes, R., Sandewall, B., eds.: Proceedings of the second Conference on Principles of Knowledge Representation and Reasoning, Cambridge, Massachusetts, Morgan Kaufmann (1991)
3. Ferraris, P.: Answer sets for propositional theories. In: LPNMR. (2005) 119–131
4. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Computing **9** (1991) 365–385
5. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Proc. Fifth International Conference and Symposium on Logic Programming. (1988) 1070–1080
6. Lifschitz, V., Tang, L.R., Turner, H.: Nested expressions in logic programs. Annals of Mathematics and Artificial Intelligence **25**(3-4) (1999) 369–389
7. Pearce, D.: Equilibrium logic: an extension of answer set programming for non-monotonic reasoning. In: Proccedings of WLP2000. (2000) 17
8. Gelfond, M.: Logic programming and reasoning with incomplete information. Ann. Math. Artif. Intell. **12**(1-2) (1994) 89–116
9. Chan, E.P.F.: A possible world semantics for disjunctive databases. Knowledge and Data Engineering **5**(2) (1993) 282–292
10. Delgrande, J., Schaub, T.: Compiling reasoning with and about preferences into default logic. In: Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI–97), IJCAI Inc. Distributed by Morgan Kaufmann, San Mateo, CA. (1997) 168–174
11. Lin, F., Shoham, Y.: A logic of knowledge and justified assumptions. Artificial Intelligence **57** (1992) 271–289
12. Lin, F., Zhou, Y.: From answer set logic programming to circumscription via logic of gk. In: Proceedings of the IJCAI'07. (2007)