Meta Level Reasoning and Default Reasoning*

Yi Zhou, Yan Zhang

Intelligent Systems Lab
University of Western Sydney
Locked Bag 1797, NSW, Australian

Abstract. In this paper, we propose a logic framework for meta level reasoning as well as default reasoning in a general sense, based on an arbitrary underlying logic. In this framework, meta level reasoning is the task of how to deduce new meta level rules by giving a set of rules, whilst default reasoning is the problem of what are the possible candidate beliefs by giving them. We define the semantics for both meta level reasoning and default reasoning and investigate their relationships. We show that this framework captures various nonmonotonic paradigms, including answer set programming, default logic, contextual default reasoning, by applying the underlying logic to different classes. Finally, we show that this framework can be reduced into answer set programming.

1 Introduction

Consider that an agent A is reasoning about a system S, where information can be captured by a logic consisting of a language $\mathcal{L}$ and an entailment relation $\models_{\mathcal{L}}$ among formulas in $\mathcal{L}$. In principle, if the agent A has perfect reasoning power, then its information about S should be a set of formulas in $\mathcal{L}$ closed under $\models_{\mathcal{L}}$, say a candidate belief. Suppose that Γ is a set of formulas, representing the information that A considers to be true about S. Thus, $Cn(\Gamma)$, the closure of Γ under $\models_{\mathcal{L}}$, should be included in every possible candidate beliefs.

However, $Cn(\Gamma)$ is not the only information that A can have about S. More can be obtained by meta level rules, which represent statements about possible candidate beliefs in a meta level language. For instance, a statement may claim that "if a candidate belief does not contain F_1, then it must contain F_2". In fact, the well-known closed world assumption is a special case of this statement when F_2 is $\neg F_1$, providing that the language $\mathcal{L}$ has the connective $\neg$ to represent negative information in the system.

Meta level rules cannot be represented in the language $\mathcal{L}$ itself since the objects they deal with are not formulas in $\mathcal{L}$ but statements about candidate beliefs. More precisely, meta level rules are composed by primitive statements and meta level connectives. The former are sentences stating whether a formula is contained in a possible candidate belief, while the latter are words connecting those primitive statements in a meta language. Consider the example mentioned above. There are two primitive statements, namely, "the candidate belief contains F_1" and "the candidate belief contains F_2". Furthermore, they are connected by two meta level connectives, namely "not" and "if then".

* This work is supported by Australian Research Council (ARC) Discovery Projects grant DP0666540.

Hence, the problem of how to represent meta level rules can be divided into two parts, how to represent primitive statements and how to represent meta level connectives. In this paper, we simply write a formula F in $\mathcal{L}$ to represent the primitive statement "the candidate belief contains F". On the other hand, we adopt a set of propositional meta level connectives, including rule and ($\&$), rule or ($\wr$), rule negation ($\sim$) and rule implication ($\Rightarrow$). For example, the meta level statement in the above example can be represented as $\sim F_1 \Rightarrow F_2$.

There are two fundamental reasoning tasks in relation to reasoning about meta level rules. The first one is called meta level reasoning. That is, which meta level rules can be deduced by giving a set of rules. Another reasoning task is default reasoning, which is the problem of what are the possible candidate beliefs by giving a set of rules.

In this paper, we propose a logic framework for both meta level reasoning and default reasoning in a general sense, based on an arbitrary underlying logic, which consists of a language $\mathcal{L}$ and an entailment relation $\models_{\mathcal{L}}$ under some restrictions. In this sense, there are numerous instances of the underlying logic, such as set inclusion, propositional logic, epistemic logic and so on.

The reasons why we consider arbitrary underlying logics are threefold. Firstly, due to diversity of applications, the logic for representing the system S may vary from the simplest one to more complicated ones. Secondly, considering the two reasoning tasks in a general sense may help us to reveal the nature of them. Finally, a general framework not only unifies a number of existing approaches but also initiates promising ones.

The rest of this paper is organized as follows. Next, we propose the syntax and basic semantics of the logic framework. In Section 3, we define both meta level reasoning and default reasoning of the framework semantically, and investigate their relationships. Then, we show that this framework is powerful enough to capture various existing approaches and possible new ones in Section 4. In Section 5, we show that it can be reduced into its simplest case, namely answer set programming. Finally, we draw our conclusions.

2 Syntax and Basic Semantics

To begin with, we need to specify what a logic is. We adopt Gentzen's idea [1] of standard logic system, which consists of two components. Firstly, it has a syntax, namely, a formal language to define what are the objects dealt with in this logic system. We denote it by a language $\mathcal{L}$. Basically, it can be represented as a set. Elements in $\mathcal{L}$ are called *formulas*. Secondly, the logic system should have reasoning ability, that is, to answer the question whether a formula can be derived by other formulas. This is formalized by an *entailment relation* $\models_{\mathcal{L}}$ between a set of formulas and a formula in $\mathcal{L}$. In other words, $\models_{\mathcal{L}}$ is a relation $\models_{\mathcal{L}} \subseteq 2^{\mathcal{L}} \times \mathcal{L}$, that satisfies the following two restrictions:

Reflexivity if $F \in \Gamma$, then $\Gamma \models_{\mathcal{L}} F$;
Transitivity (cut) if for all $F' \in \Gamma'$, $\Gamma \models_{\mathcal{L}} F'$, then $\Gamma' \models_{\mathcal{L}} F$ implies that $\Gamma \models_{\mathcal{L}} F$,

where $\Gamma, \Gamma' \subseteq \mathcal{L}$, $F, F' \in \mathcal{L}$.

According to reflexivity and transitivity, a logic system also satisfies the following properties.

Proposition 1. *Let $\mathcal{L}$ be a language and $\models_{\mathcal{L}}$ the corresponding entailment relation satisfying reflexivity and transitivity. Then, it also satisfies the following properties:*

Monotonicity *if $\Gamma \subseteq \Gamma'$, then for all $F \in \mathcal{L}$ such that $\Gamma \models_{\mathcal{L}} F$, $\Gamma' \models_{\mathcal{L}} F$;*
Equivalency *if for all $F' \in \Gamma'$, $\Gamma \models_{\mathcal{L}} F'$ and for all $F \in \Gamma$, $\Gamma' \models_{\mathcal{L}} F$, then for all $G \in \mathcal{L}$, $\Gamma \models_{\mathcal{L}} G$ iff $\Gamma' \models_{\mathcal{L}} G$;*
Extendability *if for all $F' \in \Gamma'$, $\Gamma \models_{\mathcal{L}} F'$, then for all $F \in \mathcal{L}$, $\Gamma \cup \Gamma' \models_{\mathcal{L}} F$ iff $\Gamma \models_{\mathcal{L}} F$.*

Of course, classical propositional logic is a typical example of such a logic. There are numerous other examples, such as first order logic, modal logic, probabilistic logic, intuitionistic logic and so on. In particular, set inclusion can also be considered as a logic. Let $Atom$ be a set of atoms. The formulas in the language $\mathcal{L}$ of set inclusion are defined as elements in $Atom$, and the entailment relation between a set Γ of formulas (i.e. a subset of $Atom$) and a formula F (i.e. an element in $Atom$) is defined as set inclusion (i.e. $\Gamma \models_{\mathcal{L}} F$ iff $F \in \Gamma$). It is obvious that this entailment relation (i.e. set inclusion) satisfies reflexivity and transitivity.

However, Reiter's default logic is not a logic according to this definition if the entailment relation is defined as credulous reasoning or skeptical reasoning. One reason is that both credulous reasoning and skeptical reasoning do not satisfy transitivity. Another reason is that the consequence of both credulous reasoning and skeptical reasoning is not a default rule but a propositional formula[1].

A *closure* is a set C of formulas in $\mathcal{L}$ closed under the entailment relation $\models_{\mathcal{L}}$. That is, C is a closure iff for all $F \in \mathcal{L}$ such that $C \models_{\mathcal{L}} F$, $F \in C$. By reflexivity, it is easy to see that if $C \not\models_{\mathcal{L}} F$[2], then $F \notin C$. Hence, $C \models_{\mathcal{L}} F$ iff $F \in C$. It is easy to see that if C_1 and C_2 are two closures, then so is $C_1 \cap C_2$.

Proposition 2. *Let $\mathcal{L}$ be a language and $\models_{\mathcal{L}}$ the corresponding entailment relation satisfying reflexivity and transitivity. Let Γ be a set of formulas in $\mathcal{L}$. There exists a unique closure C such that for all $F \in \mathcal{L}$, $\Gamma \models_{\mathcal{L}} F$ iff $C \models_{\mathcal{L}} F$.*

We write $Cn(\Gamma)$ to denote this closure of Γ. For convenience, we simply use Γ to denote $Cn(\Gamma)$ if it is clear from the context. Clearly, if $\Gamma_1 \subseteq \Gamma_2$, then $Cn(\Gamma_1) \subseteq Cn(\Gamma_2)$.

Based on the underlying logic $\mathcal{L}$, we define a meta level language $\mathcal{ML}(\mathcal{L})$, following a similar construction of general default logic [3]. One major difference is that, instead of classical propositional logic, we use an arbitrary underlying logic as discussed above.

The meta level language $\mathcal{ML}(\mathcal{L})$ is defined upon $\mathcal{L}$ by introducing a set of *meta level rule connectives* (*rule connectives* for short), including *rule and* ($\&$), *rule or* ($\wr$), *rule implication* ($\Rightarrow$), and a special 0-ary connective *falsity* $\perp$ as follows:

$$R := F \mid \perp \mid R \,\&\, R \mid R \wr R \mid R \Rightarrow R,$$

[1] Hence, our definition of logic is not the same as Brewka and Eiter's [2]. According to their definition, both default logic and answer set programming are logics.
[2] We write $\Gamma \not\models_{\mathcal{L}} F$ if it is not the case that $\Gamma \models_{\mathcal{L}} F$, the same for other similar notations used later.

where $F \in \mathcal{L}$. We also introduce other rule connectives *truth* $\top$, *rule negation* $\sim$, and *rule equivalence* $\Leftrightarrow$. $\top$, $\sim R$ and $R_1 \Leftrightarrow R_2$ are considered as shorthand of $\bot \Rightarrow \bot$, $R \Rightarrow \bot$ and $(R_1 \Rightarrow R_2) \mathbin{\&} (R_2 \Rightarrow R_1)$ respectively. Formulas in $\mathcal{ML}(\mathcal{L})$ are called *meta level rules* (*rules* for short). In particular, formulas in $\mathcal{L}$ are also rules. For convenience, we call them *facts*. A *rule base* is a set of rules. The *subrule* relationship between two rules are defined recursively.

– R_1 is a subrule of R_1.
– Both R_1 and R_2 are subrules of $R_1 \mathbin{\&} R_2$, $R_1 \wr R_2$ and $R_1 \Rightarrow R_2$.

In particular, if F is a fact and also a subrule of R, we say that F is a *subfact* of R.

In the basic semantics, we define the *satisfaction relation* $\models_B$ between closures in the underlying language $\mathcal{L}$ and meta level rules recursively as follows:

– If R is a fact, then $C \models_B R$ iff $C \models_{\mathcal{L}} R$;
– $C \not\models_B \bot$;
– $C \models_B R \mathbin{\&} S$ iff $C \models_B R$ and $C \models_B S$;
– $C \models_B R \wr S$ iff $C \models_B R$ or $C \models_B S$;
– $C \models_B R \Rightarrow S$ iff $C \not\models_B R$ or $C \models_B S$.

Thus, $C \models_B \top$. $C \models_B \sim R$ iff $C \models_B R \Rightarrow \bot$ iff $C \not\models_B R$ or $C \models_B \bot$ iff $C \not\models_B R$. $C \models_B R \Leftrightarrow S$ iff $C \models_B (R \Rightarrow S) \mathbin{\&} (S \Rightarrow R)$ iff $C \models_B R \Rightarrow S$ and $C \models_B S \Rightarrow R$ iff (a) $C \models_B R$ and $C \models_B S$ or (b) $C \not\models_B R$ and $C \not\models_B S$. We say that C *satisfies* R, also C is a *model* of R iff $C \models_B R$. We say that two rules are *weakly equivalent* if they have the same set of models. We say that C satisfies a rule base Δ iff C satisfies all rules in Δ.

Example 1. Consider the rule $\sim F_1 \Rightarrow F_2$. If a closure contains neither F_1 nor F_2, then it is not a model of this rule. On the other hand, a closure containing F_1 satisfies this rule, so does a closure containing F_2.

Note that the underlying logic may have internal relationships among formulas. For instance, consider a rule $F_1 \mathbin{\&} \sim F_2$. If in an underlying logic $\mathcal{L}$, $\{F_1\} \models_{\mathcal{L}} F_2$, then there is no model of the rule $F_1 \mathbin{\&} \sim F_2$. However, if in another underlying logic $\mathcal{L}'$, $\{F_1\} \not\models_{\mathcal{L}'} F_2$, then a closure containing F_1 but not F_2 is a model of the rule $F_1 \mathbin{\&} \sim F_2$.

The basic semantics can be translated into classical propositional logic. Let $At(\mathcal{L})$ be a set of atoms in propositional logic and At a one-to-one mapping from $\mathcal{L}$ to $At(\mathcal{L})$. Given a meta level rule R, by $Tr_{CL}(R)$ we denote the propositional formula obtained from R by simultaneously replacing every subfact F in R with $At(F)$ and every rule connective with corresponding classical propositional connectives. Given a closure C, by $At(C)$, we denote the propositional assignment[3] over $At(\mathcal{L})$ such that $F \in C$ iff $At(F) \in At(C)$.

Theorem 1. *Let R be a rule and C a closure. $C \models_B R$ iff $At(C)$ is a model of $Tr_{CL}(R)$ in classical propositional logic.*

Corollary 1. *Let R_1 and R_2 be two rules. If $Tr_{CL}(R_1)$ is equivalent to $Tr_{CL}(R_2)$ in classical propositional logic, then for all closures C, $C \models_B R_1$ iff $C \models_B R_2$.*

[3] We identify a propositional assignment as the set of atoms assigned to be true in it.

3 Meta Level Reasoning and Default Reasoning

A natural question is so-called meta level reasoning, namely, how to derive new meta level rules by giving a set of rules. We use a bi-level semantics for this reasoning task. The semantics is originated from the logic of here-and-there, which was developed by Heyting and adopted by Pearce [4] for answer set programming.

A *bi-level interpretation* in $\mathcal{ML}(\mathcal{L})$ is a pair $\langle C_1, C_2 \rangle$, where C_1 and C_2 are both closures in $\mathcal{L}$. The satisfaction relation $\models_{BI}$ between bi-level interpretations and meta level rules is defined recursively as follows:

- if R is a fact, then $\langle C_1, C_2 \rangle \models_{BI} R$ iff $C_1 \models_B R$ and $C_2 \models_B R$;
- $\langle C_1, C_2 \rangle \not\models_{BI} \bot$;
- $\langle C_1, C_2 \rangle \models_{BI} R_1 \,\&\, R_2$ iff $\langle C_1, C_2 \rangle \models_{BI} R_1$ and $\langle C_1, C_2 \rangle \models_{BI} R_2$;
- $\langle C_1, C_2 \rangle \models_{BI} R_1 \wr R_2$ iff $\langle C_1, C_2 \rangle \models_{BI} R_1$ or $\langle C_1, C_2 \rangle \models_{BI} R_2$;
- $\langle C_1, C_2 \rangle \models_{BI} R_1 \Rightarrow R_2$ iff
  1. $\langle C_1, C_2 \rangle \not\models_{BI} R_1$ or $\langle C_1, C_2 \rangle \models_{BI} R_2$ and
  2. $C_2 \models_B R_1 \Rightarrow R_2$.

We say that $\langle C_1, C_2 \rangle$ is a *bi-level model* of R iff $\langle C_1, C_2 \rangle \models_{BI} R$. We say that a rule base Δ *implies* a rule R, denoted by $\Delta \models_{BI} R$, iff all bi-level models of Δ are bi-level models of R as well.

The reason why we call this semantics bi-level is that the two components of the pair represent two levels of information respectively. The second lies on the underlying level, which represents a possible guess of the agent about the system, while the first one lies on the meta level, which represents the actual set of information that the agent can have by fixing the underlying level information.

Example 2 (Example 1 continued). Consider the rule $\sim F_1 \Rightarrow F_2$. Suppose that C_0 is the closure $TH(\emptyset)$, while C_1 is a closure containing F_1. Then, $\langle C_1, C_1 \rangle$ is a bi-level model of $\sim F_1 \Rightarrow F_2$, so is $\langle C_0, C_1 \rangle$. However, $\langle C_1, C_0 \rangle$ and $\langle C_0, C_0 \rangle$ are not. Thus, $\{\sim F_1 \Rightarrow F_2\} \not\models_{BI} F_1 \wr F_2$ since $\langle C_0, C_1 \rangle$ is a bi-level model of $\sim F_1 \Rightarrow F_2$ but not a bi-level model of $F_1 \wr F_2$. However, one can check that $\{F_1 \wr F_2\} \models_{BI} \sim F_1 \Rightarrow F_2$ no matter what the underlying logic is.

The bi-level semantics and the basic semantics are closely related. By induction on the structure of R, we have the following result.

Proposition 3. *Let $\langle C_1, C_2 \rangle$ be a bi-level interpretation and R a rule.*

- $\langle C_1, C_2 \rangle \models_{BI} \sim R$ *iff* $C_2 \models_B \sim R$.
- $\langle C_1, C_1 \rangle \models_{BI} R$ *iff* $C_1 \models_B R$.
- *If* $\langle C_1, C_2 \rangle \models_{BI} R$, *then* $C_2 \models_B R$.

The result of Proposition 3 is not new; it holds for the logic of here-and-there as well [5]. In fact, the bi-level semantics shares the nature of the logic here-and-there. There are two major differences. Firstly, the bi-level semantics is generalized into an arbitrary case, whilst the logic of here-and-there is only concerned with set inclusion (i.e. atom sets). Secondly, in the bi-level semantics, we do not require the restriction that the first component of the pair has to be a subset of the second one.

Proposition 4. *Let Δ be a rule base and R_1 and R_2 two rules. $\Delta \cup \{R_1\} \models_{BI} R_2$ iff $\Delta \models_{BI} R_1 \Rightarrow R_2$.*

Proposition 5. *Let Δ be a rule base and R_1 and R_2 two rules. If $\Delta \models_{BI} R_2$, then $\Delta \cup \{R_1\} \models_{BI} R_2$.*

Proposition 6. *Let Δ be a rule base, R a rule and C a closure. If $C \models_B \Delta$ and $\Delta \models_{BI} R$, then $C \models_B R$.*

Proposition 7. *Let $\langle C_1, C_2 \rangle$ be a bi-level interpretation and R a rule. $\langle C_1, C_2 \rangle \models_{BI} R$ iff $\langle C_1 \cap C_2, C_2 \rangle \models_{BI} R$.*

According to Proposition 7, the bi-level semantics, when the underlying logic is set inclusion, is indeed identical to the logic of here-and-there. The reasons why we make this minor change (i.e. to remove the restriction) are twofold. On the one hand, the restriction seems unnecessary and not natural from a mathematical point of view. On the other hand, the intuitions behind the bi-level semantics without the restriction are clearer than that with it.

Perhaps, another reasoning task is more interesting, namely default reasoning, which is the problem of what are the possible candidate beliefs by giving a set of meta level rules. We introduce two semantics for default reasoning. One is a reduction style extension semantics, following the idea from Ferraris' work [6] on answer set semantics for so-called propositional theories, and extended to general default logic by Zhou et al. [3]. The other is equilibrium semantics, originated from Pearce's equilibrium logic [4].

The *reduct* of a rule R relative to a closure C, denoted by R^C, is the rule obtained from R by simultaneously replacing every maximal subrule not satisfied by C with $\bot$. A closure C is said to be a *candidate belief*[4] of a rule R if it is the minimal closure (in the sense of set inclusion) satisfying R^C. That is, $C \models_B R^C$ and there does not exist another closure $C_1 \subset C$ such that $C_1 \models_B R^C$. We say that two rules are *equivalent* if they have the same set of candidate beliefs. Clearly, this definition can be generalized to rule bases, similar for definitions presented later.

Example 3 (Example 2 continued). Consider the example $\sim F_1 \Rightarrow F_2$ again. Assume that F_1 and F_2 are not related in the underlying logic[5]. Let C_1 be the closure $Cn(\{F_1\})$ and C_2 be the closure $Cn(\{F_2\})$. Then, $(\sim F_1 \Rightarrow F_2)^{C_1}$ is $\bot \Rightarrow F_2$. Of course, C_1 is a model of $\bot \Rightarrow F_2$. However, $Cn(\emptyset)$ is also a model of $\bot \Rightarrow F_2$. Thus, C_1 is not a candidate belief of $\sim F_1 \Rightarrow F_2$. On the other hand, C_2 is the minimal closure satisfying $(\sim F_1 \Rightarrow F_2)^{C_2}$, which is $\sim\bot \Rightarrow F_2$. Thus, C_2 is a candidate belief of $\sim F_1 \Rightarrow F_2$.

We introduce a notion of *strong equivalence* between two rules. The notion of strong equivalence, introduced by Lifschitz [7] for answer set programming, and extended to default logic by Turner [8], plays a very important role in default reasoning. Two rules

[4] This is different from the notion of belief or knowledge in epistemic reasoning.

[5] There are four possible relationships between F_1 and F_2: (a) there is no closures containing both F_1 and F_2; (b) $\{F_1\} \models_{\mathcal{L}} F_2$; (c) $\{F_2\} \models_{\mathcal{L}} F_1$, or (d) none of the above. In the first three cases, F_1 and F_2 are related.

R_1 and R_2 are said to be *strongly equivalent* iff for all other rules R_3, R_1 & R_3 has the same set of candidate beliefs as R_2 & R_3.

We also define the equilibrium semantics for default reasoning. A bi-level interpretation $\langle C_1, C_2 \rangle$ is said to be an *equilibrium model* of a rule R iff (a) $\langle C_1, C_2 \rangle$ is a bi-level model of R; (b) $C_1 = C_2$; (c) there does not exist $C_1' \subset C_1$ such that $\langle C_1', C_2 \rangle$ is also a bi-level model of R.

Equilibrium semantics is essentially a fixed point semantics. In this sense, it shares the same basic idea of the extension semantics. Both of them can be viewed as three steps. First, guess a possible set of information. Second, derive a minimal set of information by fixing the guess. Finally, if these two sets coincide with each other, then it is a possible candidate belief.

Example 4 (Example 3 continued). Again, consider the example $\sim F_1 \Rightarrow F_2$. Let C_0, C_1 and C_2 be three closures $Cn(\emptyset)$, $Cn(\{F_1\})$ and $Cn(\{F_2\})$ respectively. We have that $\langle C_1, C_1 \rangle$ is a bi-level model of $\sim F_1 \Rightarrow F_2$, so is $\langle C_0, C_1 \rangle$. Thus, $\langle C_1, C_1 \rangle$ is not an equilibrium model of $\sim F_1 \Rightarrow F_2$. On the other hand, $\langle C_2, C_2 \rangle$ is a bi-level model of $\sim F_1 \Rightarrow F_2$, and there is no other closure C' such that $C' \subset C_2$ and $\langle C', C_2 \rangle$ is also a bi-level model of $\sim F_1 \Rightarrow F_2$. Thus, $\langle C_2, C_2 \rangle$ is an equilibrium model of $\sim F_1 \Rightarrow F_2$.

Certainly, the four semantics, basic semantics, bi-level semantics, extension semantics and equilibrium semantics are closely related.

Proposition 8. *Let Δ be a rule base and C a closure. If C is a candidate belief of Δ, then $C \models_B \Delta$.*

However, the converse of Proposition 8 does not hold in general. For instance, $Cn(\{F_1\})$ is a model of $\sim F_1 \Rightarrow F_2$ but not a candidate belief of it.

Proposition 9. *Let Δ be a rule base and C_1 and C_2 two closures. $\langle C_1, C_2 \rangle \models_{BI} \Delta$ iff $C_1 \models_B \Delta^{C_2}$.*

Theorem 2. *Let Δ be a rule base and C a closure. C is a candidate belief of Δ iff $\langle C, C \rangle$ is an equilibrium model of Δ.*

Proposition 10. *Let R_1 and R_2 be two rules. R_1 and R_2 are strongly equivalent iff $\models_{BI} R_1 \Leftrightarrow R_2$.*

Corollary 2. *Let Δ be a rule base and R a rule. $\Delta \models_{BI} R$ iff $\Delta \cup \{R\}$ is strongly equivalent to Δ.*

To sum up, we have defined three levels of semantics, the basic semantics lies on the underlying level and the bi-level semantics lies on the meta level, whilst on the middle level are two equivalent semantics, namely the extension semantics and the equilibrium semantics. Although basic semantics, equilibrium semantics and reduction-style semantics are used for answer set programming, the idea that they can be used in a much more general sense has not been proposed yet. In addition, the idea of using bi-level semantics for meta level reasoning is a novel approach. The equivalence relations of those three levels are captured by weak equivalence, strong equivalence and equivalence respectively.

Proposition 11. *Let R_1 and R_2 be two rules. If R_1 and R_2 are strongly equivalent, then R_1 and R_2 are equivalent and weakly equivalent as well.*

Example 5 (Example 4 continued). Consider four rules $\sim F_1 \Rightarrow F_2$, F_2, $\sim\sim F_1 \wr F_2$ and $F_1 \wr F_2$ and assume that F_1 and F_2 are not related in the underlying logic (See Footnote 5). We have that $\sim F_2 \Rightarrow F_1$ is strongly equivalent to $\sim\sim F_1 \wr F_2$. $\sim F_1 \Rightarrow F_2$ is equivalent to F_2, but neither strongly equivalent nor weakly equivalent to F_2. $\sim F_1 \Rightarrow F_2$ is weakly equivalent to $F_1 \wr F_2$, but neither strongly equivalent nor equivalent to $F_1 \wr F_2$.

The results proposed in this section are not surprising since they hold for answer set programming as well. However, their proofs in general do not follow directly from the simplest cases since the underlying logic is not simply a set. There are many features which can be exploited in set inclusion. For instance, in set inclusion, a set of atoms is a closure. However, it might be not the case for an arbitrary logic. In addition, the atoms in set are not related, but they may have very complex relationships in an arbitrary logic.

4 The Underlying Logic

In this section, we show that the logic framework presented above is powerful enough to capture various existing approaches by applying the underlying logic to different classes. Due to a space limit, we only briefly outline the basic ideas and results in this paper and leave backgrounds, detailed comparisons and discussions to a future full version.

4.1 Set inclusion (with classical negation)

As we have shown in Section 2, set inclusion can be considered as a logic. Similarly, set inclusion with classical negation can be treated as a logic as well. Let $Atom$ be a set of atoms and Lit the set of literals, i.e., atoms or their classical negations. The formulas in the language $\mathcal{S}^{\neg}$ of set inclusion with classical negation are defined as elements in Lit. Let Γ be a set of formulas (i.e. a set of literals) and F a formula (i.e. a literal). $\Gamma \models_{\mathcal{S}^{\neg}} F$ iff (a) $F \in \Gamma$ or (b) there exists an atom a such that $a, \neg a \in \Gamma$. Clearly, this entailment relation satisfies reflexivity and transitivity.

Answer set programming (with classical negation) corresponds to default reasoning in meta level language when the underlying logic is set inclusion (with classical negation). On the other hand, it also explains why the answer set semantics for logic programs with classical negation work very well.

Theorem 3. *Let P be a disjunctive logic program (with classical negation) [9] and X a set of atoms (literals). X is an answer set of P iff X is a candidate belief of $\widehat{P}$ in $\mathcal{ML}(\mathcal{S})$ ($\mathcal{ML}(\mathcal{S}^{\neg})$), where $\widehat{P}$ is the set of meta level rules obtained from P by replacing each rule $p_1 \mid \ldots \mid p_n \leftarrow q_1, \ldots, q_m, \texttt{not}\, r_1, \ldots, \texttt{not}\, r_l$ by a meta level rule $q_1 \,\&\, \ldots \,\&\, q_m \,\&\, \sim r_1 \,\&\, \ldots \,\&\, \sim r_l \Rightarrow p_1 \wr \ldots \wr p_n$.*

Clearly, Theorem 3 also holds for normal logic programming [10]. More generally, it holds for Ferraris' answer set semantics for propositional theories [6] as well.

Theorem 4. *Let Γ be a set of propositional formulas and X a set of atoms. X is an answer set of Γ iff X is a candidate belief of $\widehat{\Gamma}$ in $\mathcal{ML}(\mathcal{S})$, where $\widehat{\Gamma}$ is the set of meta level rules obtained from Γ by replacing every classical propositional connectives with corresponding rule connectives.*

4.2 Propositional logic

Certainly, classical propositional logic $\mathcal{CL}$ is a typical example of the underlying logic. The following theorem shows that Zhou et al.'s general default logic is a special case of the logic framework by applying the underlying logic to propositional logic.

Theorem 5. *Let Δ be a rule base in general default logic [3]. A theory T is an extension of Δ iff T is a candidate belief of Δ in $\mathcal{ML}(\mathcal{CL})$.*

As shown in [3], Reiter's default logic in the propositional case [11] and Gelfond et al.'s disjunctive default logic [12] are special cases of general default logic. Therefore these two approaches are also special cases of default reasoning of $\mathcal{ML}(\mathcal{CL})$.

4.3 First order logic: closed case

Let $\mathcal{FOL}$ be a first order language. By $\mathcal{FOL}_{\mathcal{S}}$, we denote the subclass of $\mathcal{FOL}$ by restricting the formulas to sentences, i.e., first order formulas without free variables.

Theorem 6. *A set of sentences is an extension of a closed default theory $\langle D, W\rangle$ iff it is a candidate belief of $W \cup \widehat{D}$ in $\mathcal{ML}(\mathcal{FOL}_{\mathcal{S}})$*[6].

4.4 Multi-context logic

It is well argued that the notion of context plays a very important role in AI [13–16]. A context of an agent about the environment represents its own (local) subjective view of the environment. There is an increasing interest on formalizing a multi-context language, which defines not only the information of a number of contexts themselves but also the information of interrelationships among them.

Given a set $\mathcal{L}_1, \ldots, \mathcal{L}_n$ of n languages and their corresponding entailment relations satisfying both reflexivity and transitivity, we define a multi-context language $\mathcal{L}_1 \times \ldots \times \mathcal{L}_n$. The formulas in $\mathcal{L}_1 \times \ldots \times \mathcal{L}_n$ are labeled formulas, which have the form $\langle k, F\rangle$, where k is a label to denote which context it comes from and F is a formula in $\mathcal{L}_k$. A formula $\langle k, F\rangle$ is entailed by a set Γ of formulas in $\mathcal{L}_1 \times \ldots \times \mathcal{L}_n$ iff F is entailed in the logic $\mathcal{L}_k$ by the set of formulas in Γ labeled by k. Formally,

$$\Gamma \models_{\mathcal{L}_1 \times \ldots \times \mathcal{L}_n} \langle k, F\rangle \text{ iff } \{G \mid \langle k, G\rangle \in \Gamma\} \models_{\mathcal{L}_k} F.$$

Clearly, $\models_{\mathcal{L}_1 \times \ldots \times \mathcal{L}_n}$ satisfies reflexivity and transitivity as well.

[6] $\widehat{D}$ is the set of rules in $\mathcal{ML}(\mathcal{FOL}_{\mathcal{S}})$ by rewriting each rule $F_1, MF_2, \ldots, MF_n/F_{n+1}$ in D to F_1 & $\sim\neg F_2$ & $\ldots$ & $\sim\neg F_n \Rightarrow F_{n+1}$.

A set Γ of formulas in $\mathcal{L}_1 \times \ldots \times \mathcal{L}_n$ can be equivalently written as an n-tuple $(\Gamma_1, \ldots, \Gamma_n)$, where $\Gamma_k = \{G \mid \langle k, G\rangle \in \Gamma\}$. Under this reformulation, we show that Brewka et al.'s contextual default reasoning, which can be considered as a "syntactical" counterpart of Roelofsen and Serafini's information chain approach [15], is a special case of the framework when the underlying logic is a multi-context propositional logic.

Theorem 7. *Let $\mathcal{L}_1, \ldots, \mathcal{L}_n$ be n propositional languages (built over different sets of atoms). $(\Gamma_1, \ldots, \Gamma_n)$ is a contextual extension of a normal multi-context system C [16] iff $(\Gamma_1, \ldots, \Gamma_n)$ is a candidate belief of $\widehat{C}$ in $\mathcal{ML}(\mathcal{L}_1 \times \ldots \times \mathcal{L}_n)$*[7].

Furthermore, Brewka et al.'s approach is a homogeneous one (i.e. all the contexts are propositional languages), whilst our approach allows heterogenous contexts. Our approach is also an generalization of Giunchiglia's heterogenous multi-context logic [14], which does not consider nonmonotonic rules.

A related work is due to Brewka and Eiter [2]. They also intend to integrate heterogenous contexts by nonmonotonic rules. However, in their approach, the underlying logic is defined as a tuple (KB, BS, ACC), where KB is the set of all possible knowledge bases, BS is the set of all possible belief sets, and ACC is a function from KB to 2^{BS}, for describing the "semantics" of this logic. A logic in our sense can be decried as a special kind of this tuple as follows: KB are the sets of formulas in $\mathcal{L}$; BS are the sets of closures; ACC is the closure operator. When restricting Brewka and Eiter's sense of logic to ours, their approach of equilibria coincides with default reasoning of the meta level language by applying the underlying logic to multi-context logic.

5 Reducing into Answer Set Programming

In this section, we show that the logic framework proposed in this paper can be reduced into its simplest case, namely answer set programming, by identifying the internal relationships among formulas in the underlying logic. Here, we consider the general case of answer set programming in the propositional case [6].

Let R be a rule. We write $Fact(R)$ to denote the set of subfacts of R. Suppose that $Fact(R) = \{F_1, \ldots, F_n\}$. We introduce n new atoms $P = \{p_1, \ldots, p_n\}$ associated with each fact in $Fact(R)$. By $P(R)$ we denote the answer set program obtained from R by simultaneously replacing each occurrence of $F_i, (1 \le i \le n)$ in R with p_i. By $I(R)$ we denote the programs of all rules of the following form:

$$p_{i1} \,\&\, p_{i2} \,\&\, \ldots \,\&\, p_{ik} \Rightarrow p_j,$$

where $\{i_1, i_2, \ldots, i_k, j\} \subseteq \{1, \ldots, n\}$ such that $\{F_{i1}, F_{i2}, \ldots, F_{ik}\} \models_{\mathcal{L}} F_j$. By $Tr(R)$ we denote the program $\{P(R), I(R)\}$.

Let Γ and Γ' be two sets of formulas in $\mathcal{L}$ such that $\Gamma \subseteq \Gamma'$. We say that Γ is *maximal* to Γ' iff for all formulas $F \in \Gamma' \backslash \Gamma$, $\Gamma \not\models_{\mathcal{L}} F$.

[7] Here, $\widehat{C}$ is the set of rules in $\mathcal{ML}(\mathcal{L}_1 \times \ldots \times \mathcal{L}_n)$ obtained from C by rewriting each fact F in W_i to $\langle i, F_i\rangle$, and each rule $\langle c_1, G_1\rangle, \ldots, \langle c_m, G_m\rangle : \langle c_{m+1}, H_1\rangle, \ldots, \langle c_{m+n}, H_n\rangle / F$ in D_i to $\langle c_1, G_1\rangle \,\&\, \ldots \,\&\, \langle c_m, G_m\rangle \,\&\, {\sim}\langle c_{m+1}, \neg H_1\rangle \,\&\, \ldots \,\&\, {\sim}\langle c_{m+n}, \neg H_n\rangle \Rightarrow \langle i : F\rangle$, where F, G_i, H_j are propositional formulas in corresponding contexts.

Proposition 12. *Let R be a rule and C a closure. If C is a candidate belief of R, then there exists a subset Γ of $Fact(R)$ such that $C = Cn(\Gamma)$.*

Theorem 8. *Let R be a rule and $Fact(R) = \{F_1, \ldots, F_n\}$. Let $P = \{p_1, \ldots, p_n\}$ be n new atoms associated with each fact in $Fact(R)$.*

1. *If $\{p_{i1}, \ldots, p_{ik}\}$ is an answer set of $Tr(R)$, then $Cn\{F_{i1}, \ldots, F_{ik}\}$ is a candidate belief of R and $\{F_{i1}, \ldots, F_{ik}\}$ is maximal to $Fact(R)$.*
2. *If C is a candidate belief of R and $\{F_{i1}, \ldots, F_{ik}\}$ is the set of formulas obtained in Proposition 12 such that $C = Cn(\{F_{i1}, \ldots, F_{ik}\})$, then $\{p_{i1}, \ldots, p_{ik}\}$ is an answer set of $Tr(R)$.*

Theorem 9. *Let R be a rule. $\models_{BI} R$ iff all HT-models of $I(R)$ are also HT-models of $P(R)$.*

Intuitively, the rule R is constructed from $Fact(R)$ by rule connectives. $P(R)$ represents the structure of R (i.e. the way of constructing R), while $I(R)$ identifies all the internal relationships among $Fact(R)$. Theorem 8 and 9 show that, both meta level reasoning and default reasoning in any meta level language can be captured in answer set programming by separating the structure of rules and the interrelationships among underlying facts.

The translation introduces n new atoms, where n is the number of facts in $Fact(R)$. Clearly, n is polynomial, in fact linear, in the length of R. Interestingly, the original atoms in R no longer occur in $Tr(R)$. However, in some cases, n could be exponential in the number of the original atoms since atoms can compose exponential number of formulas in some underlying logics (e.g. propositional logic).

One of the most important problems is whether this translation is polynomial or not. Unfortunately, although $P(R)$ is linear in the size of R, $I(R)$ may contain exponential number of rules. The reason is that there may be exponential number of such internal relationships among $Fact(R)$. Observing that not all rules in $I(R)$ are necessary, we can pick up those ”minimal” ones, namely, the set of premises is the minimal set satisfying the consequence in $Fact(R)$. However, even only taken these internal relationships into account, the number is still exponential. For example, in classical propositional logic, let $Atom = \{a_1, \ldots, a_m\}$ be m atoms. Let $F_{ij}, (1 \leq i \leq m), (1 \leq j \leq m)$ be the formula $a_i \rightarrow a_j$. Then, we have m^2 formulas. However, we have exponential number of such internal relationships. For instance, for any set of atoms $a_{i1}, \ldots, a_{ik}$ different from two atoms a_i, a_j, $\{a_i \rightarrow a_{i1}, a_{i1} \rightarrow a_{i2}, \ldots, a_{ik-1} \rightarrow a_{ik}, a_{ik} \rightarrow a_j\}$ is a minimal set satisfying $a_i \rightarrow a_j$.

Despite this negative result, the translation is not only of theoretical interests but also of practical uses. It enables us to easily analyze the meta level language in small-scale case studies. Also, it is a useful tool to investigate properties in meta level language. For instance, the following propositions follow from Theorem 8 and 9 straightforwardly.

Corollary 3. *If $\mathcal{L}$ is a decidable language, then both meta level reasoning and default reasoning of $\mathcal{ML}(\mathcal{L})$ are decidable.*

Corollary 4. *Contextual ASP [16] has the same computational complexity as normal logic programming.*

6 Conclusion

In this paper, we proposed a logic framework for meta level reasoning as well as default reasoning about meta level rules. In this framework, meta level reasoning is the reasoning task of how to deduce new meta level rules by giving a set of rules, while default reasoning is the problem of what are the possible candidate beliefs by giving them. Default reasoning has attracted a lot of attentions in the past three decades [3, 6, 8–12, 16, 17]. On the other hand, meta level reasoning, although also important, was relatively less studied. A closely related topic is normative reasoning [18], which can be considered as a fragment of meta level reasoning of default logic. Another topic is so-called SE-consequence [19, 20] in answer set programming, which actually coincides with meta level reasoning task of answer set programming.

Some technical results, although not trivial, might not be surprising since they hold for corresponding cases of answer set programming and default logic as well [3, 5]. However, surprisingly, all these can be summarized in a general framework with simple semantics. Hence, we argue that, this framework, indeed, captures the nature of both meta level reasoning and default reasoning in a general sense.

It is worth mentioning that default reasoning is nonmonotonic (in the sense of skeptical reasoning or credulous reasoning), whilst meta level reasoning is actually monotonic (See Proposition 5). Furthermore, meta level reasoning also satisfies reflexivity and transitivity. This means that meta level reasoning itself can also be treated as a logic in our sense. However, default reasoning is not. In other words, in this framework, ”default” is not a logic but a meta level reasoning task.

We demonstrated this framework's expressiveness to capture several existing approaches of default reasoning by applying the underlying logic to different classes. More precisely, answer set programming, default logic in propositional case, default logic in closed first order case and contextual default logic coincide with default reasoning of meta level language of set inclusion, propositional logic, first order logic and multi-context logic respectively (See Theorem 3-7).

Also, the framework will initiate some new promising formalisms. One of them is to consider description logic, for instance $\mathcal{SHIQ}$, as the underlying logic. This provides a natural combination of description logic and rule-based formalism, which is a crucial step to fulfil the blueprint of Semantic Web Initiative [21, 22]. However, rule connectives considered in this paper are basically propositional. In other words, meta level rules with free variables cannot be represented in this approach. One possible way to overcome this barrier is to use the technique of grounding [9, 10]. That is, to define a first order meta level language powerful enough to represent rules with variables, and then to transfer them to propositional meta level rules by grounding for all instances. However, this topic is beyond the scope of this paper. We leave it to our future investigations. Certainly, there are many other interesting and important candidates of the underlying logic, such as epistemic logic, logic of multi-agents, temporal logics, logics of uncertainty, and so on. Many of them are worth pursuing. We leave them to our future work as well.

We showed that both meta level reasoning and default reasoning in a general sense can be reduced to its simplest case (Theorem 8 and 9), namely answer set programming, by identifying the internal relationships (represented by $I(R)$ in the translation) among formulas in the underlying logic. This provides a powerful tool to study the general framework without going through the details of the underlying logic.

To sum up, the main contributions of this framework are as follows. Firstly, it unifies a bunch of existing approaches for default reasoning, and it can be easily seen that this approach can also initiate other promising paradigms of default reasoning. In addition, it suggests a new reasoning task, namely meta level reasoning, for deriving new meta level rules given a rule base. Finally, it can be interestingly reduced to the simplest case, namely answer set programming.

References

1. Szabo, M.: The collected papers of Gerhard Gentzen (1969)
2. Brewka, G., Eiter, T.: Equilibria in heterogeneous nonmonotonic multi-context systems. In: AAAI. (2007) 385–390
3. Zhou, Y., Lin, F., Zhang, Y.: General default logic. In: LPNMR. (2007) 241–253
4. Pearce, D.: A new logical characterisation of stable models and answer sets. In: NMELP. (1997) 57–70
5. Ferraris, P., Lifschitz, V.: Mathematical foundations of answer set programming. In: We Will Show Them! (1). (2005) 615–664
6. Ferraris, P.: Answer sets for propositional theories. In: LPNMR. (2005) 119–131
7. Lifschitz, V., Pearce, D., Valverde, A.: Strongly equivalent logic programs. ACM Transactions on Computational Logic **2** (2001) 526–541
8. Turner, H.: Strong equivalence for logic programs and default theories (made easy). In: LPNMR. (2001) 81–92
9. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Computing **9** (1991) 365–385
10. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: ICLP. (1988) 1070–1080
11. Reiter, R.: A logic for default reasoning. Artificial Intelligence **13** (1980) 81–132
12. Gelfond, M., Lifschitz, V., Przymusinska, H., Truszczyński, M.: Disjunctive defaults. In: KR. (1991) 230–237
13. McCarthy, J.: Notes on formalizing context. In: IJCAI. (1993) 555–560
14. Giunchiglia, F.: Contextual reasoning. Epistemologia (1993) 345–364
15. Roelofsen, F., Serafini, L.: Minimal and absent information in contexts. In: IJCAI. (2005) 558–563
16. Brewka, G., Roelofsen, F., Serafini, L.: Contextual default reasoning. In: IJCAI. (2007) 268–273
17. Lin, F., Shoham, Y.: A logic of knowledge and justified assumptions. Artificial Intelligence **57** (1992) 271–289
18. Geffner, H., Pearl, J.: Conditional entailment: bridging two approaches to default reasoning. Artificial Intelligence **53**(2-3) (1992) 209–244
19. Eiter, T., Fink, M., Tompits, H., Woltran, S.: Simplifying logic programs under uniform and strong equivalence. In: LPNMR. (2004) 87–99
20. Wong, K.: Sound and complete inference rules for se-consequence. JAIR **31** (2008) 205–216
21. Eiter, T., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the semantic web. In: KR. (2004) 141–151
22. Motik, B., Rosati, R.: A faithful integration of description logics with logic programming. In: IJCAI. (2007) 477–482