Answer Set Programming with Epistemic Defaults*

Shutao Zhang[1][0000−0003−0853−9281], Zhizheng Zhang[1][0000−0001−9851−6184], Bin Wang[1][0000−0002−6852−0761], and Jun Shen[1][0000−0001−7245−7075]

School of Computer Science and Engineering, Southeast University, Nanjing, China
{shutao_zhang, seu_zzz, kse.wang, junshen}@seu.edu.cn

Abstract. Epistemic logic programming(ELPs) languages are based on Answer Set Prolog(ASP) and allow for introspective reasoning with incomplete knowledge through using epistemic modal operators. The ELP defined by Gelfond and Kahl uses **K** and **M** as its epistemic operators. Intuitively, $\mathbf{K}l$ means the literal l is known to be true and $\mathbf{M}l$ means it is possible to be true. In this paper, we develop a logic programming language with epistemic defaults. Firstly, we extend the ELP with a new epistemic default operator **A**, which intuitively means that a literal is acceptable, i.e., will not lead to any conflict in some belief sets. Secondly, we define the semantics of epistemic defaults by reduct an ELP program into a new program of an ASP-based default logic programming language we proposed. Finally, we discuss the relationship between our ELP and some existing ELPs. It shows that by epistemic defaults, we can make a clear distinction between possibilities and defaults. This new ELP language provides a method to represent incomplete and default knowledge more naturally.

Keywords: epistemic specification · epistemic defaults · logic programming.

1 Introduction

Answer Set Prolog[4] is a successful KR language under the stable semantics[5] with plenty of extensions. Epistemic logic programming(ELPs) languages are extensions of ASP that allow for introspective reasoning with incomplete knowledge through using epistemic modal operators. Gelfond[3] and Kahl[8] presented an extension of answer set prolog by introducing the epistemic operators **K** and **M**, which is called ES, to support strong introspection. Shen and co-workers[13] proposed a language of ES for strong introspection through the use of epistemic negation operator instead of modal operators **K** or **M**.

Since Gelfond proposed the first ELP language in [7], the intended world view caused by circular justification of subjective literals has attracted a lot of

* The work was supported by the Pre-research Key Laboratory Fund for Equipment (Grant No. 6142101190304) and the National Key Research and Development Plan of China (Grant No.2017YFB1002801).

attention of researchers in KR area. Gelfond refined his semantics definition in [3] to solve the unsupported beliefs caused by the circular justification of **K**. Based on his work, Kahl focused on solving the unintended world view caused by **M** in [8] by redefining the modal reduct. Shen tried to find an unambiguous interpretation of the rational principle for ES in [13], thus he defined a partial order relation among candidate world views by the set of epistemic negations. By this partial order relation, a world view is a candidate world view only if it satisfies maximal epistemic negations. However, all these semantics will cause the self-supported by subjective literals with **M**. In [15], Yuan Zhang and Yuanlin Zhang emphasize that the semantics of ES should follow the principle of justification. They redefined the semantics of ES to eliminate the unintended worldview caused by circular justifications of modal operator **M**.

Example 1 (Circular Justification with **M***).* Consider the following program Π_1:

$$r_1 : flies(X) \leftarrow bird(X), \mathbf{M} flies(X).$$
$$r_2 : flies(X) \leftarrow horse(X), \mathbf{M} flies(X).$$

With a set of facts on *bird*, this program concludes that all birds can fly under the semantics of ES2014. However, with a set of facts on *horse*, the world views indicating all horse can fly do not make sense. Under the semantics Yuan Zhang and Yuanlin Zhang defined, the only world view of Π_1 is $\{\{\}\}$.

A further observation of the unjustified world views caused by operator **M** reminds us that the semantics of modal operator **M** may be ambiguous. For the instance of Π_1, the subjective literal $\mathbf{M}flies(X)$ in r_1 expresses a default knowledge that a bird can fly if it is consistent. Meanwhile, in r_2 it is a knowledge of possibility, which means a horse can fly if it is proved to be possible. The difference between these two knowledge is that for a ground subjective literal $\mathbf{M}flies(a)$ in r_2 where a is a substitution of X, r_2 requires $flies(a)$ be merely consistent in some belief sets rather than true.

Defaults are very useful in logic programs because they allow drawing conclusions based on common or typical knowledge with incomplete information. Researchers have paid a lot of attention on the relationships between modal logic and default logic, and have found many inherent connections between these two kind of knowledge. Konolige has analyzed the relation between Default and Autoepistemic logic in [9], and proved there exist a translation of default logic into autoepistemic logic and its reverse. Truszczynski has showed that the nonmonotonic logic S4F captures the default logic in [14]. On the other hand, some researchers focused on represent default by logic programming languages. In [6], Gelfond has showed that the nonmonotonic logic ASP without constraints or disjunctions captures the default logic. He also present a method to represent defaults in ASP by the addition of exceptions in [4].

To develop a method dividing the representation of these two kinds of knowledge, we propose a new epistemic logic programming language which extends Epistemic Specifications with epistemic defaults. We introduce a new epistemic modal, **A**, to represent the default knowledge. By the new semantics we define, subjective literal $\mathbf{K}l$ means l is known to be true, $\mathbf{M}l$ means l is proved to be possible, $\mathbf{A}l$ means l is possible to be acceptable. With the modal operators we defined, the program Π_1 in Example 1 can be rewritten as:

$$r_1 : flies(X) \leftarrow bird(X), \mathbf{A}flies(X).$$
$$r_2 : flies(X) \leftarrow horse(X), \mathbf{M}flies(X).$$

The aim of this paper is to develop a new language for Epistemic Specifications with epistemic defaults. Under the semantics of this new language, the circular justification of $\mathbf{M}$ does not lead to unsupported beliefs. Meanwhile, the new language provides a method to represent epistemic defaults with a new modal operator $\mathbf{A}$, which is defined with a new ASP-based default logic program language.

The rest of this paper is organized as follows. Firstly, we recall the preliminary knowledge about Default Logic and Epistemic Specifications. Secondly, we review some related works that aimed to solve unintended world views. Then, we introduce the new epistemic logic programming language with epistemic defaults. Finally, we discuss the relationship between the new language with ES2014 proposed in [8].

2 Preliminary

2.1 Default Logic

Default logic is a classical nonmonotonic logic defined by Reiter in [12]. It uses defeasible rules by *default rules* of the form

$$\frac{\alpha : \beta_1, \ldots, \beta_n}{\omega} \tag{1}$$

where α, β, ω are classical formulas. α is the *prerequisite* of the default, β_is are *justifications*, ω is the *consequent*. The default rule 1 intuitively means "If α is provable and all β_is are consistent with it, then assume ω as default." A default rule is *normal* if β is equivalent to ω; it is *semi-normal* if β implies ω. A *default theory* (D, W) is formed by a set of default rules D and a set of formulas W. The reasoning of a default theory is defined by its extension.

Definition 1 (Extension of Default Theories). *Let (D, W) be a default theory, E be a set of formulas. Define $E_0 = W$ and for $i \geq 0$:*

$$\begin{aligned} GD_i =&\{\frac{\alpha : \beta_1, \ldots, \beta_n}{\omega} \in D | \alpha \in E_i, \neg\beta_i \notin E, \ldots, \neg\beta_n \notin E\} \\ E_{i+1} =& Th(E_i) \cup \{Conseq(\delta) | \delta \in GD_i\}. \end{aligned}$$

, where $Th(E_i)$ is the set of all classical consequences of E_i, $Conseq(\delta)$ is the consequent of default rule δ. Then E is an extension for (D, W) iff $E = \bigcup_{i=0}^{\infty} E_i$.

An extension of default theory (D, W) represents a possible set of beliefs of this theory. It can be used to find if a proposition is true in an incomplete theory.

2.2 Epistemic Specifications and Justified View

Epistemic Specifications(ES) extends traditional answer set programs with modal operators $\mathbf{K}$ and $\mathbf{M}$. An ES program is a finite set of rules of the form:

$$l_1 \ or \ldots or \ l_k \leftarrow e_1, \ldots, e_m. \tag{2}$$

where ls are objective literals, es are extended literals, which are either objective literals or subjective literals with epistemic operators $\mathbf{K}$, $\neg\mathbf{K}$, $\mathbf{M}$ or $\neg\mathbf{M}$. Note that we use $\neg$ to denote *negation as failure* in this paper. If l is an objective literal without $\neg$, then we call l a positive objective literal.

A *belief set* of an ES program Π is a set of positive objective ground literals in the language of Π. A *view* is defined as a collection of belief sets. If an extended literal s is satisfied by a point structure $\langle A, W\rangle$, where $A \in W$, is defined as:

- $\langle A, W\rangle \models l$, also denoted by $A \models l$, iff $l \in A$;
- $\langle A, W\rangle \not\models l$, also denoted by $A \not\models l$, iff $l \notin A$;
- $\langle A, W\rangle \models \mathbf{K}l$, also denoted by $W \models \mathbf{K}l$, iff $\forall A \in W : A \models l$;
- $\langle A, W\rangle \models \mathbf{M}l$, also denoted by $W \models \mathbf{M}l$, iff $\exists A \in W : A \models l$;
- $\langle A, W\rangle \models \neg\mathbf{K}l$, also denoted by $W \models \neg\mathbf{K}l$, iff $W \not\models \mathbf{K}l$;
- $\langle A, W\rangle \models \neg\mathbf{M}l$, also denoted by $W \models \neg\mathbf{M}l$, iff $W \not\models \mathbf{M}l$.

Many versions of semantics have been proposed for the language of Epistemic Specifications. Kahl introduced a typical one in [8], which is called ES2014 in this paper, by the definition of modal reduct and world view.

Definition 2 (Modal Reduct of ES2014). *Let Π be a finite ES program, W be a collection of belief sets. The modal reduct of Π w.r.t W, denoted by Π^W, is obtained from Π by eliminating subjective literals as Table 1.*

Table 1. Modal Reduct of ES2014

s	$W \models s$	$W \not\models s$
$\mathbf{K}l$	replace s by l	delete the rule
$\neg\mathbf{K}l$	remove s	replace s with $\neg l$
$\mathbf{M}l$	remove s	replace s with $\neg\neg l$
$\neg\mathbf{M}l$	replace s by $\neg l$	delete the rule

Definition 3 (World Views of ES2014). *W is a world view of Π if and only if W is equal to a collection of all answer sets of Π^W.*

To have a clear view of circular modal justification, Kahl also introduced the conception of M-cycle, which is a cycle in modal support graph of a program with an edge of $\mathbf{M}$.

Definition 4 (Modal Supported Graph). *Given a ground epistemic logic program Π, a modal supported graph of Π, or MS graph for short, is a directed graph where:*

– *for each rule r_i in Π, there is a* rule node *labeled by r_i denoting the rule;*
– *for each distinct objective literal l in the language of Π, there is a* literal node *labeled by l;*
– *for each objective literal l in the head of rule r, there is an unlabeled edge from the rule node r to literal node l;*
– *for each extended literal in the body of rule r, there is an edge labeled according to Table 2 going from literal node l to rule node R.*

Table 2. labels of outcoming edges from literal nodes

Extended Literal	l	$\neg l$	$\mathbf{K}l$	$\neg\mathbf{K}l$	$\mathbf{M}l$	$\neg\mathbf{M}l$
Label		$\neg$	$\mathbf{K}$	$\neg\mathbf{K}$	$\mathbf{M}$	$\neg\mathbf{M}$

Definition 5 (M-Cycle). *A cycle in the MS graph of a ground epistemic logic program is called a* M-cycle *if an edge within the cycle is labeled by* $\mathbf{M}$.

Example 2 (M-cycle). Consider a program Π_2:

$$r_1 : p \leftarrow \mathbf{M}q, \neg q.$$
$$r_2 : q \leftarrow \mathbf{M}p, \neg p.$$

Figure 1 shows the modal support graph of Π_2, which contains an M-cycle through q, r_1, p and r_2.

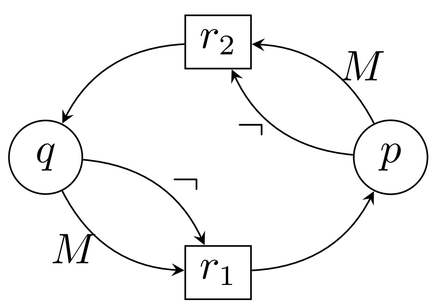

Fig. 1. Modal support graph of Π_2 with M-cycle

2.3 Equilibrium Logic

Equilibrium logic proposed by Pearce in [11] is a nonmonotonic logic that is used to characterize stable models. A formula of equilibrium logic is built from atoms in AT with operator $\wedge$, $\vee$, $\bot$, $\rightarrow$ in the usual way. An HT-interpretation is a pair $\langle H, T\rangle$, where $H \subseteq T \subset AT$, H and T are called "here" and "there" respectively. It is said to be total if $H = T$. If a formula or a theory is satisfied by an HT-interpretaion is defined as:

- $\langle H, T\rangle \not\models \bot$;
- $\langle H, T\rangle \models p$ iff $p \in H$ where p is an atom;
- $\langle H, T\rangle \models \varphi \wedge \psi$ iff $\langle H, T\rangle \models \varphi$ and $\langle H, T\rangle \models \psi$;
- $\langle H, T\rangle \models \varphi \vee \psi$ iff $\langle H, T\rangle \models \varphi$ or $\langle H, T\rangle \models \psi$;
- $\langle H, T\rangle \models \varphi \rightarrow \psi$ iff
  - $T \models \varphi \rightarrow \psi$ and
  - $\langle H, T\rangle \not\models \varphi$ or $\langle H, T\rangle \models \psi$;
- $\langle H, T\rangle \models \Pi$, where Π is a theory and $\langle H, T\rangle$ is said to be a model of Π iff $\langle H, T\rangle \models \varphi$ for all $\varphi \in \Pi$.

It is easy to prove that $\langle T, T\rangle \models \Pi$ iff $T \models \Pi$ classically. We use $CL(\Pi)$ to denote the set of all classical models of Π. Interpretation T is a stable model or an equilibrium model of Π iff $T \in CL(\Pi)$ and there is no $H \subset T$ such that $\langle H, T\rangle \models \Pi$.

3 Related Works

In this section, we will review some related works that aims to deal with the self-supported problem of Epistemic Specifications.

3.1 Epistemic Negation

Yi-dong Shen and Thomas Eiter argued that ES2014 does not eliminate the unintended worldviews caused by subjective operator **M**. In [13], they introduced a new subjective operator **not** as epistemic negation operator. The traditional subjective operators **K** and **M** are replaced by ¬**not** and **not**¬ respectively. From all candidate views of the program, they select those satisfying maximal epistemic negations as the world views. This strategy of knowledge minimization is also used in [2].

Epistemic negation provides a method to eliminate unintended world views. However, the subjective operator **M** is interpreted as a negation as failure of epistemic negation, which leads to a self-support problem of **M**.

3.2 Founded Epistemic Equilibrium Logic

Founded Epistemic Equilibrium Logic is a semantics of Epistemic Specification proposed by Pedro Cabalar et al. in [1] to capture the set of founded world views. They proposed a new semantics based on a combination of Moore's Autoepistemic Logic[10] and Pearce's Equilibrium Logic[11], and proved the foundness of world views under this semantics. Their work provide an insight of slef-supportedness in ES and coincide with ES2011 and ES2014 for the circular justification with subjective operator **K**. However, because the relation of subjective operators **K** and **M** was under debate, their work only focused on the study of operator **K**.

3.3 Justified View

Yuan Zhang and Yuanlin Zhang proposed a semantics of Epistemic Specifications in [15]. The aim of their work is to develop an intuitive understanding of **M** operator and get rid of circular justification in a stronger sense than Kahl, Shen and Eiter in [8, 13]. They refined the semantics of Epistemic Specifications by constructing a justified reduct and disjunction reduct. By the new semantics, since all literals in a world view need to be justified, the M-cycle does not cause a self-support problem.

Example 3 (one-line program with M-cycle). Consider an ES program Π_3

$$p \leftarrow \mathbf{M}p.$$

According to the semantics defined by Kahl and Shen, the only world view of Π_3 is $\{\{p\}\}$. However, according to the semantics defined by Yuan Zhang and Yuanlin Zhang, the only world view of Π_3 should be $\{\{\}\}$.

4 Epistemic Defaults

In this section, we firstly introduce Default Logic Programming language, a new ASP-based language with defaults. After that, we introduce the syntax of Epistemic Specifications with epistemic defaults, and use Default Logic Program to define the semantics of epistemic defaults.

4.1 Default Logic Program

Default logic program(DLP) is an extension of ASP to represent and reason knowledge of defaults. A DLP program Π is a finite collection of rules of the form

$$l_1 \ or \cdots or \ l_k \leftarrow l_{k+1}, \cdots, l_m, e_1, \cdots, e_n. \tag{3}$$

, where l_i are *literals* of the form p or $\neg p$, e_i are *extended literals* of the form $\mathbf{C}l$ or $\neg\mathbf{C}l$, l is a literal. Intuitively, $\mathbf{C}l$ means it is consistent to assume l is true. A rule containing operator $\mathbf{C}$ is a default rule. If the set of extended literals $\{\mathbf{C}l | \mathbf{C}l \in body(r)\}$ is equivalent to $head(r)$, we call the default rule r a normal default rule.

The semantics of DLP is defined with here-and-there models(HT-models).

Definition 6 (Is Satisfied By HT-interpretations). *An HT-interpretation of program Π is a pair of consistent ground literal sets where $H \subseteq T \subseteq Literal(\Pi)$. Let $\langle H, T\rangle$ be an HT-structure, l be a literal in the language of Π, r is a rule in Π.*

- $\langle H, T\rangle \models l$ *iff* $H \models l$;
- $\langle H, T\rangle \models \mathbf{C}l$ *iff* $T \models l$;
- $\langle H, T\rangle \models \neg\mathbf{C}l$ *iff* $T \not\models l$;

– $\langle H,T\rangle \models r$ *iff* $\exists l \in head(r) : \langle H,T\rangle \models l$ *or* $\exists e \in body(r) : \langle H,T\rangle \not\models e$*, where* e *is a literal or extended literal;*
– $\langle H,T\rangle \models \Pi$ *iff* $\forall r \in \Pi : \langle H,T\rangle \models r$*, and this HT-structure is called an* HT-model *of* Π*.*

Definition 7 (Default Reduct). *Let* Π *be a DLP program,* $\langle H,T\rangle$ *be an HT-structure. The default reduct of* Π *w.r.t.* $\langle H,T\rangle$*, denoted by* $\Pi^{\langle H,T\rangle}$*, is obtained by eliminating the occurrence of* $\mathbf{C}l$ *or* $\neg\mathbf{C}l$ *in rule* r *as Table 3.*

Table 3. Obtain $\Pi^{\langle H,T\rangle}$ by eliminating defaults.

e	$\langle H,T\rangle \models e$	$\langle H,T\rangle \not\models e$
$\mathbf{C}p$	remove e	replace e with p
$\mathbf{C}\neg p$	remove e	replace e with $\neg p$
$\neg\mathbf{C}p$	replace e with $\neg p$	delete the rule
$\neg\mathbf{C}\neg p$	replace e with p	delete the rule

Now we can use default reduct to define the default HT-models of a program with default rules.

Definition 8 (Default Equilibrium Models). *For a DLP program* Π *with default rules, an HT-model* $\langle H,T\rangle$ *is a default equilibrium model of* Π *iff*

1. $\langle H,T\rangle$ *is an HT-model of* $\Pi^{\langle H,T\rangle}$*,*
2. $\langle H,H\rangle$ *is an HT-model of* $\Pi^{\langle H,T\rangle}$*, i.e.* H *is a stable model of* $\Pi^{\langle H,T\rangle}$*,*
3. $\nexists T' : T \subset T', T' \models \Pi^{\langle H,T'\rangle}$ *and* H *is a stable model of* $\Pi^{\langle H,T'\rangle}$*.*

The set of all default equilibrium models of a program Π *is denoted by* $DEM(\Pi)$*. We call* H *a* default stable model *of* Π *if there exist a* default equilibrium model $\langle H,T\rangle$ *of* Π*, which is denoted by* $H \models \Pi$*. For every literal* $l \in T$*, we have* $H \models_\Pi \mathbf{C}l$*. The set of all default stable models of* Π *is denoted by* $DSM(\Pi)$*.*

Definition 9 (Is Satisfied By a Stable Model). *Let* H *be a default stable model of a default logic program* Π*,* $\mathbf{C}p$ *in the language of* Π*. The satisfaction of* $\mathbf{C}l$ *and* $\neg\mathbf{C}l$ *can also be defined as follows:*

– $H \models_\Pi \mathbf{C}l$ *iff* $\exists T \supseteq H : H \in AS(\Pi^{\langle H,T\rangle})$*,* $T \models \Pi^{\langle H,T\rangle}$ *and* $l \in T$*;*
– $H \models_\Pi \neg\mathbf{C}l$ *iff* $H \not\models_\Pi \mathbf{C}l$*.*

Let us take a close look at Definition 6, 8 and 9. It shows that the third condition of Definition 8 is used to make sure T is maximal, thus the default equilibrium model $\langle H,T\rangle$ can satisfy as many extended literals in Π as possible.

Example 4 (Default Equilibrium Models). Consider the DLP program Π_4

$$a \leftarrow \mathbf{C}a.$$
$$b \leftarrow \mathbf{C}b.$$
$$\leftarrow a, b.$$
$$c \leftarrow a.$$

Let $M_1 = \langle\{a, c\}, \{a, c\}\rangle$, $M_2 = \langle\{b\}, \{b, c\}\rangle$, $M_3 = \langle\{b\}, \{b\}\rangle$. It is easy to find that all of these HT-interpretations are HT-models of Π. However, because the T-part of M_3 is a subset of the T-part of M_2, M_3 is not an HT-equilibrium model of Π. By Definition 8, $\{a, c\}$ is a stable model of $\Pi_4^{M_1}$ and $\{b\}$ is an answer set of $\Pi_4^{M_2}$, which means both M_1 and M_2 are HT-equilibrium models of Π_4.

4.2 Epistemic Specifications with Epistemic Defaults

With the definition of default logic program, we extend ELP with epistemic defaults by the addition of a new epistemic operator $\mathbf{A}$. A rule with epistemic defaults is of the form

$$l_1 \; or \cdots or \; l_k \leftarrow e_1, \cdots, e_m. \tag{4}$$

, where l_i are literals, e_i are extended literals of the form $\mathbf{K}l$, $\neg\mathbf{K}l$, $\mathbf{M}l$, $\neg\mathbf{M}l$, $\mathbf{A}l$ or $\neg\mathbf{A}l$. Intuitively, $\mathbf{A}l$ means that l is possibly acceptable.

Definition 10 (Is Al Satisfied). *For a collection of HT-models W, $W \models \mathbf{A}l$ iff $\exists w \in W : w \models \mathbf{C}l$.*

Definition 11 (Modal Reduct). *Let Π be a program with epistemic defaults and W be a non-empty collection of belief sets, where a belief set is an HT-structure $\langle H, T\rangle$, H and T are sets of literals in the language of Π and $H \subseteq T$. The modal reduct of Π w.r.t W, denoted by Π^W, is a DLP program obtained from Π as Table 4 by eliminating every subjective literal s.*

Table 4. Modal Reduct of ES with Epistemic Defaults.

s	$W \models s$	$W \not\models s$
$\mathbf{K}l$	replace s by l	delete the rule
$\neg\mathbf{K}l$	remove s	delete the rule
$\mathbf{M}l$	remove s	delete the rule
$\neg\mathbf{M}l$	replace s by $\neg l$	delete the rule
$\mathbf{A}l$	remove s	$\mathbf{C}l$
$\neg\mathbf{A}l$	replace s by $\neg\mathbf{C}l$	delete the rule

Definition 12 (Subjective Interpretation). *Let Π be a program with epistemic defaults, $W = \{A_1, \ldots, A_n\}$ be a collection of belief sets. The mapping ρ from subjective literal s and belief set A_i is defined as:*

- $\forall i : \rho(\mathbf{K}l, A_i) = l_i$, *if* $W \models \mathbf{K}l$;
- $\forall i : \rho(\mathbf{M}l, A_i) = l_j$, *if* $W \models \mathbf{M}l$ *and* $A_j \models l$;
- $\forall i : \rho(\mathbf{A}l, A_i) = \mathbf{C}l_j$, *if* $W \models \mathbf{A}l$ *and* $A_j \models \mathbf{C}l$;
- $\forall i : \rho(\mathbf{A}l, A_i) = \mathbf{C}l_i$, *if* $W \not\models \mathbf{A}l$;
- $\forall i : \rho(\neg\mathbf{K}l, A_i) = \neg l_j$, *if* $W \models \neg\mathbf{K}l$ *and* $A_i \not\models l$;

– $\forall i : \rho(\neg\mathbf{M}l, A_i) = \neg l_i$, *if* $W \models \neg\mathbf{M}l$*;*
– $\forall i : \rho(\neg\mathbf{A}l, A_i) = \neg l_i$, *if* $W \models \neg\mathbf{A}l$*.*

In Definition 12, ρ is a mapping from a subjective literal to the belief sets that justifies it. ρ provides a method to find the self-support cycles in a program.

Example 5 (Subjective Interpretetion). Considering program Π_5 with a self supported of $\mathbf{A}$.

$$p \leftarrow \mathbf{A}q, \neg q.$$
$$q \leftarrow \mathbf{A}p, \neg p.$$

Let $A_1 = \langle\{p\}, \{p\}\rangle$, $A_2 = \langle\{q\}, \{q\}\rangle$, W $= \{A_1, A_2\}$. The subjective interpretations w.r.t W is $\rho(\mathbf{A}p, A_i) = \mathbf{C}p_1,\ \rho(\mathbf{A}q, A_i) = \mathbf{C}q_2$ for $i \in \{1, 2\}$.

Definition 13 (Justified Reduct). *Consider a DLP program Π, a collection W of belief sets $\{A_1, \ldots, A_n\}$. The justified reduct of Π w.r.t $\langle A_i, W\rangle$, denoted by $\Pi^{\langle A_i, W\rangle}$, is obtained by the following steps:*

1. *removing subjective literals s if $W \not\models s$;*
2. *replacing subjective literals s with $\rho(s, A_i)$ if $W \models s$;*
3. *replacing objective literals l with l_i.*

The justified reduct of a program Π w.r.t W shows the justification of all literals in Π. However, a justified reduct is a default logic program with disjunctions, which is possible to have stable models not contained by W. In that case, it is necessary to eliminate disjunctions in a justified reduct of a program.

Definition 14 (Disjunction Reduct). *Let Π be a positive disjunctive program and A be a consistent set of ground literals in the language of Π. The* disjunction reduct *of Π w.r.t A is a program obtained from Π removing all literals not in A from the head of all the rules in Π.*

Definition 15 (Justified View). *Consider Π be a program with epistemic defaults, $W = \{A_1, \ldots, A_n\}$ a collection of HT-models. Let $B = \{l_i | A_i \models l, 1 \leq i \leq n\}$, $C = \{l_i | A_i \models \mathbf{C}l, 1 \leq i \leq n\}$, the* full reduct *of Π w.r.t $\langle A_i, W\rangle$, denoted by $\Pi_{full}^{\langle A_i, W\rangle}$, is obtained from Π by applying justified reduct w.r.t W, Gelfond-Lifschitz reduct and disjunction reduct w.r.t B. W is a* justified view *of Π iff $\langle B, B \cup C\rangle$ is the only default equilibrium model of $\bigcup_{i=1}^{n} \Pi_{full}^{\langle A_i, W\rangle}$.*

Example 6 (Justified Views of Π_5). Consider program Π_5 in Example 5. We have the justified reduct ${\Pi_5}^{\langle A_1, W\rangle} = \{p_1 \leftarrow \mathbf{C}q_2, \neg q_1 . q_1 \leftarrow \mathbf{C}p_1, \neg p_1 .\}$ and ${\Pi_5}^{\langle A_2, W\rangle} = \{p_2 \leftarrow \mathbf{C}q_2, \neg q_2 . q_2 \leftarrow \mathbf{C}p_1, \neg p_2 .\}$. By Definition 15, $B = \{p_1, q_2\}$, which is the answer set of ${\Pi_5}_{\text{full}}^{\langle A_1, W\rangle} \cup {\Pi_5}_{\text{full}}^{\langle A_2, W\rangle}$, thus W is a justified view of Π_5.

Definition 16 (World View). *Let Π be a program with epistemic defaults, W be a collection of HT-models. W is a* world view *of Π iff*

1. *W is equal to the collection of all default equilibrium models of the modal reduct Π^W and*
2. *W is a justified view of Π.*

Example 7 (World View of Π_5). Consider Program Π_5 and $W_1 = \{A_1, A_2\}$ in Example 5. Since $W \models \mathbf{A}p \wedge \mathbf{A}q$, the modal reduct $\Pi_5^{W_1}$ is $\{p \leftarrow \neg q. q \leftarrow \neg p.\}$. Apparently A_1 and A_2 are the only default equilibrium models of $\Pi_5^{W_1}$. Example 6 has shown that W_1 is a justified view of Π_5, thus W_1 is a world view of Π_5.

Consider $W_2 = \{\emptyset, \emptyset\}$. It is easy to find that $W_2 = AS(\Pi_5^{W_2})$. The justified reduct of Π_5 w.r.t W_2 is $\emptyset$, thus W_2 is also a world view of Π_5.

5 Relationship with Other ES

In this section, we will compare our ES with Epistemic Defaults, or ESD for short, with ES2014 in [8] and ES2017 in [15].

5.1 Relation with ES2014

Example 8 (Compare M-cycle). Consider program Π_3 in Example 3. Let $W_1 = \{\langle\{p\}, \{p\}\rangle\}$, $W_2 = \{\langle\emptyset, \emptyset\rangle\}$.

Under the semantics of ES2014, the modal reducts of Π_3 are $\Pi_3^{W_1} = \{p.\}$ and $\Pi_3^{W_2} = \{p \leftarrow \neg\neg p.\}$. Because $W_1 = AS(\Pi_3^{W_1})$, $W_2 \neq AS(\Pi_3^{W_2})$, W_1 is a world view of Π while W_2 is not.

In [8], Kahl assumes that a rational agent should have a preference order to believe the subjective literals. He thinks it is easier to accept $\mathbf{M}l$ than l. The epistemic negation defined by Shen shows the same preference relation in [13]. This makes a rule with directly M-cycle works like a justification part in a normal default rule.

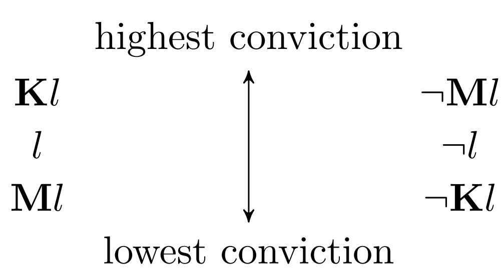

Fig. 2. Preference Relation of Subjective Literals

Because the definition of the justified reduct and modal reduct of operator $\mathbf{K}$ and $\mathbf{M}$ are equal to the work of Yuan Zhang and Yuanlin Zhang, the circular justification of **M** and **K** will be omitted if the literals in this loop do not have any external support. As a result, a subjective literal of the form $\mathbf{M}l$ can be interpreted as "it is safe to believe l is possible", while the possibility of l should be justified by other rules or belief sets. However, because of the feature of epistemic defaults, the cycle of A does lead to a different conclusion.

Example 9 (A-cycle). Consider program Π_6:

$$p \leftarrow \mathbf{A}p.$$

Let $W = \{A_1 = \langle \{p\}, \{p\} \rangle\}$. The justified reduct of Π_6 w.r.t $\langle A_1, W \rangle$ is $\Pi_6^{\langle A_1, W \rangle} = \{p_1 \leftarrow \mathbf{C}p_1.\}$, and $\langle \{p_1\}, \{p_1\} \rangle$ is the only default equilibrium model of it, which means W is a justified view of Π. On the other hand, the modal reduct of Π_6 w.r.t W is $\{p.\}$. As a result, it is obvious that W is a world view of Π_6.

With a close observation of Example 8 and Example 9, we can find that although the semantics of **M** is defined differently, operator **A** provides a method to represent defaults information, which is represented by M-cycles under the semantics of ES2014.

Definition 17 (Default View Image). *For a view W of an ES2014 program Π, the default view image $\widehat{W}$ of W is a collection of HT-model that*

$$\widehat{W} = \{\langle A, A \rangle | A \in W\} \tag{5}$$

Proposition 1 (Relationship between direct M-cycle and A-cycle). *Let Π be an ES program that every modal operator* **M** *in Π occurs in a rule of the form*

$$p \leftarrow \mathbf{M}p, B. \tag{6}$$

, where B is a collection of objective literals or extended subjective literals $\mathbf{K}l$ or $\neg\mathbf{K}l$, Π' be a program obtained from Π by replacing **M** *with* **A**. *A collection of belief sets W is a world view of Π under ES2014 semantics if and only if its default view image $\widehat{W}$ is a world view of Π' under the semantics of epistemic defaults.*

Proof. Let rule r be a rule in Π of the form (6), r' is obtained from r by replacing $\mathbf{M}p$ with $\mathbf{A}p$. If $W \not\models B$, both r and r' are satisfied, thus we only need to consider the situations that $W \models B$.

To prove the soundness of Proposition 1, consider the following situations:

- For $\neg\mathbf{K}l \in B$, if $W \not\models \neg\mathbf{K}l$, the modal reduct of ES2014 replaces $\neg\mathbf{K}l$ with $\neg l$. By the definition of satisfiability, $\forall A_i \in W : l \in A_i$, which means r is deleted in the Gelfond-Lifschitz reduct. By Definition 11, r is deleted in $\Pi'^{\widehat{W}}$. If $W \models \neg\mathbf{K}l$, then $\neg\mathbf{K}l$ is removed in both reducts.
- For $\mathbf{M}l \in B$ or $\mathbf{M}l \in \Pi/r$, if $W \models \mathbf{M}l$, then $\mathbf{M}l$ is removed in both reducts. If $W \not\models \mathbf{M}l$, it is replaced by $\neg\neg l$ in Π'^{W}, while the rule is removed in $\Pi'^{\widehat{W}}$. Because $\mathbf{M}l$ does not occurs in any M-cycles, $\neg\neg l$ does not support l.

According to the definition of satisfiability in ES2014, l is not satisfied by any belief set in W, thus the rule $\mathbf{M}l$ occurs in is deleted in the Gelfond-Lifschitz reduct.
- For other subjective literals $s \in B$ without operator $\mathbf{M}$, the modal reduct of s by Definition 2 is equal to the one by Definition 11.
- If $W \models \mathbf{M}p$ and $W \models B$, the modal reduct of r w.r.t W is $p \leftarrow B.$, which means $\forall A_i \in W : p \in A_i$ and $\widehat{W} \models \mathbf{A}p$. By the definition of justified reduct, r is translated into $\forall \widehat{A_i} \in \widehat{W} : p_i \leftarrow \mathbf{C}p_i.$, p_i is justified. By the definition of modal reduct, r is translated into $p \leftarrow B..$ For the other rules in Π, the modal reducts under both semantics are equal, thus $(\Pi/r)^W = (\Pi'/r')^{\widehat{W}}$. It shows that W is a world view of Π' under the semantics in Definition 16.
- If $W \not\models \mathbf{M}p$ and $W \models B$, the modal reduct of r w.r.t W is $\{p \leftarrow \neg\neg p, B.\}$, which is equivalent to $\{p \textit{ or } \neg p \leftarrow B.\}$. Because $\forall A_i \in W : A_i \not\models p$, p must not be consistent with the other rules in Π/r, thus $\forall A_i \in \widehat{W} : A_i \not\models \mathbf{C}p$ and $W \not\models \mathbf{A}p$, rule r' is deleted from Π' in the modal reduct of r'. As a result, $\widehat{W}$ is a justified view of Π' and equals to the collection of default equilibrium models of $\Pi'^{\widehat{W}}$. It shows that W is a world view of Π' under the semantics in Definition 16.

To prove the completeness of Proposition 1, consider the following situations:

- As showed in the proof of soundness, subjective literals in B and Π/r are equivalent under the two semantics.
- If $\widehat{W} \models \mathbf{A}p$, the modal reduct of r' w.r.t $\widehat{W}$ is $p.$, thus $\forall \widehat{A_i} \in \widehat{W} : \widehat{A_i} \models p.$, $W \models \mathbf{M}p$. The modal reducts of rest rules in Π and Π' are equal, thus W is the collection of all answer sets of Π^W, which means W is a world view of Π.
- If $\widehat{W} \not\models \mathbf{A}p$, r' is deleted in the modal reduct $\Pi'^{\widehat{W}}$. By the definition of satisfiability, $\forall A_i \in W : A_i \not\models_{\Pi'^{\widehat{W}}} \mathbf{C}p$, thus $\forall A_i \in W : A_i \not\models p$, $W \not\models \mathbf{M}p$. The modal reduct of r is $p \leftarrow \neg\neg p, B.$, thus the modal reduct of $\Pi^W = \Pi'^{\widehat{W}} \cup \{p \leftarrow \neg\neg p, B.\}$, and $\Pi'^{\widehat{W}}$ is not consistent with p. It means $AS(\Pi^W) = AS(\Pi'^W) = W$, W is a world view of Π.

According to the proof of soundness and completeness above, Proposition 1 holds.

More trivially, we can expand Proposition 1 to all kinds of M-cycle.

Theorem 1 (Relationship between M-cycle and A-cycle). *Let Π be an ES2014 program, Π' be an ESD program obtained by replacing every subjective literal of the form $\mathbf{M}l$ in rule r with subjective literal $\mathbf{A}l$ if there is an edge labeled with $\mathbf{M}$ from rule node r to l in an M-cycle. A view W of Π under the semantics of ES2014 if and only if its default view image $\widehat{W}$ is a world view of Π'.*

Here is the sketch of the proof of Theorem 1. As showed in the proof of Proposition 1, it needs to prove that the modal reduct of rules in M-cycle and translated A-cycle under two semantics respectively are equal. The proof needs to consider following situations:

1. multiple rules in the cycle;
2. rules with disjunctive heads in the cycle;
3. NAF operator $\neg$ in the cycle;
4. modal operators $\mathbf{K}$ and $\neg\mathbf{K}$ in the cycle;

Here we use multiple rules as an example.

Proposition 2. *For an ES2014 program containing following rules*

$$\begin{aligned} r_1 &: q_1 \leftarrow \mathbf{M}p, B_1. \\ r_2 &: q_2 \leftarrow q_1, B_2. \\ &\dots \\ r_i &: q_i \leftarrow q_{i-1}, B_i. \\ r_{i+1} &: p \leftarrow q_i, B_{i+1}. \end{aligned}$$

, W is a world view of Π if and only if the default view image $\widehat{W}$ is a world view of the ESD program containing

$$\begin{aligned} r'_1 &: q_1 \leftarrow \mathbf{A}p, B_1. \\ r'_1 &: q_2 \leftarrow q_1, B_2. \\ &\dots \\ r'_i &: q_i \leftarrow q_{i-1}, B_i. \\ r'_{i+1} &: p \leftarrow q_i, B_{i+1}. \end{aligned}$$

Proof. According to the proof of Proposition 1, $r_2, \dots, r_{i+1}$ are equivalent to $r'_2, \dots, r'_{i+1}$. Thus we only need to proof the equivalence of r_1 and r'_1 when $W \models B_1$.

As we showed in the proof of Proposition 1, it is easy to proof that $\widehat{W}$ is a world view of Π' and W is a world view of Π.

Example 10 (Program with NAF and M-cycle). Consider a program Π_7:

$$\begin{aligned} p &\leftarrow \neg q. \\ p \textit{ or } q &\leftarrow \mathbf{M}q. \end{aligned}$$

By the definition of world view of ES2014, the only world view of Π_7 is $\{\{p\}, \{q\}\}$.

The corresponding ESD program is Π'_7

$$\begin{aligned} p &\leftarrow \neg q. \\ p \textit{ or } q &\leftarrow \mathbf{A}q. \end{aligned}$$

Assume $W_1 = \{\{p\}, \{q\}\}$, $W_2 = \{\{p\}\}$, $W_3 = \{\{q\}\}$, $W_4 = \{\{p, q\}\}$. It is obvious that W_1 is a world view of Π'_7. For W_2 consider following situations:

– assume $\widehat{W_2} = \{\langle\{p\}, \{p, q\}\rangle\}$, $\widehat{W_2} \models \mathbf{A}q$, then $W_2 \neq DEM(\Pi'^{\widehat{W_2}}_7)$;
– assume $\widehat{W_2} = \{\langle\{p\}, \{p\}\rangle\}$, $\widehat{W_2} \not\models \mathbf{A}q$, then $W_2 \neq DEM(\Pi'^{\widehat{W_2}}_7)$,

thus $\widehat{W_2}$ is not a world view of Π'_7. It can be showed W_3, W_4 are not world views of Π'_7, which means $\widehat{W_1}$ is the only world view of Π'_7.

5.2 Relation with ES2017

Since the definition of justified view and world view comes from the definition of ES2017, an ESD program without subjective operator $\mathbf{A}$ can be seen as an ES2017 program.

Theorem 2. *For an ES2017 program Π, W is a world view of Π if and only if $\widehat{W}$, the default view image W, is a world view of Π under the semantics of ESD.*

Example 11 (Compare M-cycle). Consider Program Π_3 in Example 3. Under the semantics of ES2017, the modal reducts of Π_3 are $\Pi_3^{W_1} = \{p.\}$ and $\Pi_3^{W_2} = \{\}$, the justified reducts are $\Pi_3^{\langle A_1, W_1\rangle} = \{p_1 \leftarrow p_1.\}$ and $\emptyset$. It shows that p in A_1 is circular justified by itself, thus W_1 is not a world view of Π and W_2 is. Meanwhile, if Π is an ESD program, $\widehat{W_2}$ is the only world view of it.

6 Conclusion

We present a logic programming paradigm, which is an extension of Epistemic Specifications with epistemic defaults, to provide a way capable of modeling default and incomplete knowledge. We also compared the ability of expression and semantics of this language with ES2014, and proposed a translation from programs with M-cycles of ES2014 to programs with A-cycles of our language with epistemic defaults. It shows that the new language can also provide to separate the representation of defaults and possibilities, which can be used to eliminate the ambiguity of $\mathbf{M}$ in ES2014 and other similar languages for Epistemic Specifications.

Reasoning of programs with Epistemic Specifications is always expensive in computation. Although we have defined the syntax and semantics of this new language, we still do not have an algorithm for program solving so far. In the future, we are planning to analysis the computational complexity of solving and develop an algorithm with acceptable efficiency for our further study on the application of epistemic defaults.

References

1. Cabalar, P., Fandinno, J., Luis, F.d.C.: Founded World Views with Autoepistemic Equilibrium Logic. In: Logic Programming and Nonmonotonic Reasoning - 15th International Conference, LPNMR 2019. pp. 134–147. Springer, Philadelphia, PA, USA (2019)

2. Del Cerro, L.F., Herzig, A., Su, E.I.: Epistemic equilibrium logic. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence. vol. 2015-Janua, pp. 2964–2970 (2015)
3. Gelfond, M.: New Semantics for Epistemic Specifications. In: Proceedings of the 11th International Conference on Logic Programming and Nonmonotonic Reasoning. vol. 6645, pp. 260–265. Springer, Vancouver, Canada (2011)
4. Gelfond, M., Kahl, Y.: Knowledge Representation, Reasoning, and the Design of Intelligent Agents. Cambridge University Press (2013)
5. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Proceedings of the Fifth International Conference and Symposium on Logic Programming. vol. 2, pp. 1070–1080. MIT Press, Cambridge, Massachusetts, USA (1988)
6. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Gener. Comput. **9**(3/4), 365–386 (1991)
7. Gelfond, M., Przymusinska, H.: Definitions in epistemic specifications. In: Logic Programming and Non-monotonic Reasoning, Proceedings of the First International Workshop, Washington, D.C., USA, July 1991. pp. 245–259 (1991)
8. Kahl, P.T.: Refining The Semantics For Epistemic Logic Programs. Phd, Texas Tech University (2014)
9. Konolige, K.: On the relation between default and autoepistemic logic. Artif. Intell. **35**(3), 343–382 (1988)
10. Moore, R.C.: Semantical considerations on nonmonotonic logic. Artificial Intelligence **25**(1), 75–94 (jan 1985)
11. Pearce, D.: Equilibrium logic. Annals of Mathematics and Artificial Intelligence **47**(1-2), 3–41 (2006)
12. Reiter, R.: A logic for default reasoning. Artificial Intelligence **13**(1-2), 81–132 (1980)
13. Shen, Y.D., Eiter, T.: Evaluating epistemic negation in answer set programming. Artificial Intelligence **237**, 115–135 (aug 2016)
14. Truszczynski, M.: Modal interpretations of default logic. In: Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, August 24-30, 1991. pp. 393–398 (1991)
15. Zhang, Y., Zhang, Y.: Epistemic specifications and conformant planning. AAAI Workshop - Technical Report **WS-17**, 781–787 (2017)