Answer Set Programming with Epistemic Defaults

Shutao Zhang[1], Zhizheng Zhang[1] and Jun Shen[1]

[1]School of Computer Science and Engineering, Southeast University, China

Abstract

Answer Set Prolog (ASP) and its extensions are considered powerful tools for encoding defaults. However, some defaults that are usually originated from *permitted* modal expressions seem hard to deal with through ASP and its existing extensions. In this paper, we develop two extensions of ASP, which are called $\mathrm{ASP^D}$ and $\mathrm{ES^D}$, by introducing permitted operators. The former uses an operator **C** preceding a literal to express the literal is permitted to be true in the current belief set. The latter extends Epistemic Specification (ES) with an epistemic operator **A** preceding a literal to express the literal is permitted to be true in some belief sets. The syntax and semantics of $\mathrm{ASP^D}$ and $\mathrm{ES^D}$ are introduced sequentially. Then, the relationship between $\mathrm{ES^D}$ and the ES is carefully discussed to show the difference between epistemic operators **M** and **A**. Besides, several examples in the paper are used to illustrate the necessity of using permitted operators in ASP and its extensions.

Keywords

answer set prolog, epistemic specifications, epistemic defaults

1. Introduction

Answer Set Prolog (ASP) [1] is a successful KR language under stable semantics [2] with plenty of extensions. Epistemic Specifications (ES) is one of these extensions that allow for introspective reasoning with incomplete knowledge through epistemic operators. Gelfond [3] and Kahl [4] presented an extension of the answer set prolog by introducing the epistemic operators **K** and **M** to support strong introspection. Shen and Eiter [5] proposed a language for strong introspection through epistemic negation operator **NOT** instead of **K** and **M**. Although ASP and ES are considered as powerful tools for encoding defaults, some defaults that are usually originated from permitted expressions seem hard to deal with in ASP and ES.

Example 1 (p by default)**.** *An interpretation of "p by default" is "p is true if it is permitted." Under this interpretation, "p by default" naturally means a belief set* $\{p\}$*. Moreover, with additional information that "p is not permitted", the belief set is* $\{\}$*.*

There are usually two ASP programs Π_1 and Π_2 that are used to encode "p by default" respectively. Π_1 : $\{p \leftarrow \neg\neg p.\}$, where $\neg$ denotes negation as failure (NAF). Π_2 : $\{p \leftarrow \neg \sim p.\}$, where $\sim$ denote strong negation. However, neither Π_1 nor Π_2 follows our interpretation. Π_1 has two answer sets, $\{p\}$ and $\{\}$. Π_2 seems fine, but $\Pi_2 \cup \{\leftarrow p.\}$ is not satisfiable, which we expect to have an answer set $\{\}$.

ASPOCP 2021: Workshop on Answer Set Programming and Other Computing Paradigms, September 30, 2021, Porto, Portugal
shutao_zhang@seu.edu.cn (S. Zhang); seu_zzz@seu.edu.cn (Z. Zhang); junshen@seu.edu.cn (J. Shen)
0000-0003-0853-9281 (S. Zhang); 0000-0001-9851-6184 (Z. Zhang); 0000-0001-7245-7075 (J. Shen)

Furthermore, let us see the introspection situation in ES as shown in Example 2.

Example 2. *"p if q is possible" is often encoded as an ES program* Π_3*:*

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

However, Π_3 has an unique world view $\{\emptyset\}$, which is against our interpretation that "p is true because q is permitted", and we expect a world view $\{\{p\}\}$.

Here we have another example of introspection in Example 3, which is a variant of an example in [6].

Example 3. *A sentinel should raise the alarm if he found some evidence that it is possible to be dangerous. Meanwhile, he should keep alert if the danger has not been eliminated.*

There are two rules about the sentinel's introspection and decisions in this example. The first one is a rule with introspection about the possibility and can be encoded by a classical ES rule

$$alarm \leftarrow \mathbf{M}dangerous. \qquad (r_1)$$

However, we can not find an accurate ES rule to express the second one, which can be interpreted as the sentinel should keep alert if *dangerous* is permitted by some belief sets. For instance, we try to describe the sentinel's introspection with the following rules.

$$alert \leftarrow \mathbf{M}dangerous. \qquad (r_2')$$

$$alert \leftarrow \neg\mathbf{K} \sim dangerous. \qquad (r_2'')$$

Rule (r_2') means *alert* can be derived if there exists at least one answer set in which *dagerous* is true, but a belief set permitting *dangerous* may not contain it. Rule (r_2'') contains a new literal $\sim$*dangerous* in its body, which can not be derived if there is no rule with $\sim$*dangerous* in its head. Therefore, neither rule (r_2') nor rule (r_2'') has the semantics we want.

Hence, this paper aims to develop the extensions of ASP and ES respectively to address the issues of handling defaults originated from permitted expressions. Specifically, we develop two extensions of ASP, which are called ASP^{D} and ES^{D} respectively, by introducing permitted operators. The former uses an operator **C** preceding a literal to express the literal is permitted to be true in the current belief set. The latter extends Epistemic Specification (ES) with an epistemic operator **A** preceding a literal to express the literal is permitted to be true in some belief sets. Intuitively, "p by default" can be encoded as $p \leftarrow \mathbf{C}p$. in ASP^{D}, and "p if q is possible" can be encoded as $p \leftarrow \mathbf{A}q$. in ES^{D}. Furthermore, the second rule in Example 3 can be written as

$$alert \leftarrow \mathbf{A}dangerous. \qquad (r_2)$$

According to rule (r_2), the sentinel should keep alert if *dangerous* is not proved to be inconsistent in at least one of his belief sets.

Usually, a non-ground logic program, which is a logic program with variables, is considered as a shorthand for the corresponding ground program. Therefore, we only consider ground logic programs in this paper unless in some examples.

The rest of this paper is organized as follows. In Section 2, we review Default Logic and Epistemic Specifications. In Section 3, we propose the syntax and semantics of ASP^{D}. In Section 4, we propose the syntax and semantics of ES^{D}. In section 5, we discuss the relationship between the new language and ES^{GK} [4]. At last, we conclude the paper with some future work.

2. Preliminaries

2.1. Default Logic

Defaults are very useful in logic programs because they allow drawing conclusions based on commonsense or typical knowledge with incomplete knowledge. Default Logic (DL) [7] has been extensively researched since it was proposed as a nonmonotonic paradigm to represent default.

The defeasible rules in default logic are called *default rules* of the form

$$\frac{\alpha\ :\ \beta_1, \ldots, \beta_n}{\omega} \qquad (1)$$

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* is a pair (D, W), where D is a set of default rules, and W is a set of formulas.

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 $i \geq 0$:*

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

where $Th(E_i)$ is the set of all classical propositional consequences of E_i, $Conseq(\delta)$ is the consequent of the default rule δ. Then E is an extension for (D, W) iff $E = \bigcup_{i=0}^{\infty} E_i$. Note that we use operator $\neg$ for negation as failure *(NAF) and $\sim$ for classical negation in this paper. An extension of default theory (D, W) represents a possible set of beliefs of this theory.*

Gelfond et al. [8] proposed the Disjunctive Default Logic (DDL) that extended classical DL with disjunction to extend the representation ability of default logic. The disjunctive defaults have the form of

$$\frac{\alpha\ :\ \beta_1, \ldots, \beta_m}{\omega_1 | \ldots | \omega_n} \qquad (2)$$

Definition 2. *(Extension of Disjunctive Default Theories) Let (D, W) be a disjunctive default theory, E be a set of formulas. E is an extension for (D, W) if E is one of the minimal deductively closed set of formulas E', where for any default rule from D, if $\alpha \in E'$ and $\sim \beta_1, \ldots, \sim \beta_m \notin E$, then $\exists \omega_i : \omega_i \in E'$.*

Researchers have paid much attention to the relationships between modal logic and default logic to capture the semantics of default logic. They have found many inherent connections between these two kinds of knowledge. Konolige [9] proved the existence a reversible translation from default logic into strongly grounded autoepistemic logic (AEL). Based on his work, Gottlob [10] constructed a nonmodular translation from default logic to standard AEL. Truszczynski [11] has shown that the nonmonotonic logic S4F captures the default logic. Cabalar et al. [12] proposed intuitionistic default logic, a variation of default logic inside S4F. Meanwhile, some researchers focused on represent default by logic programming languages. Gelfond [13] has shown that the nonmonotonic logic ASP without constraints or disjunctions captures the default logic. He also presented a method to represent defaults in ASP by adding literals and rules about abnormal information [1]. Lifschitz [14] introduced an idea to translate ASP programs into default theories, which Chen et al. [15] used to develop a default logic solver based on ASP solvers.

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

Table 1
Modal reduct of $\mathrm{ES}^{\mathrm{GK}}$

2.2. Epistemic Specifications and Justified Semantics

Epistemic Specifications (ES) extends traditional answer set programs with epistemic modal operators **K** and **M**. An ES program is a finite set of rules of the form:

$$e_1 \; or \ldots or \; e_k \leftarrow e_{k+1}, \ldots, e_m, s_1, \ldots, s_n. \tag{3}$$

where l is a literal, e_is are objective literals of the form l or $\neg l$, s_is are subjective literals with an epistemic operator $\mathbf{K}$, $\neg\mathbf{K}$, $\mathbf{M}$, or $\neg\mathbf{M}$.

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

- $\langle A, W\rangle \models l$ iff $l \in A$, where l is a literal.
- $\langle A, W\rangle \models \neg l$ iff $l \notin A$.
- $\langle A, W\rangle \models \mathbf{K}e$ iff $\forall A \in W : A \models e$.
- $\langle A, W\rangle \models \mathbf{M}e$ iff $\exists A \in W : A \models e$.
- $\langle A, W\rangle \models \neg\mathbf{K}e$ iff $\exists A \in W : A \not\models e$.
- $\langle A, W\rangle \models \neg\mathbf{M}e$ iff $\forall A \in W : A \not\models e$.

For an objective literal e, it can be denoted as $A \models e$. For a subjective literal s, it can be denoted as $W \models s$.

Many versions of semantics have been proposed for the language of Epistemic Specifications. Kahl [4] introduced a typical one, which we call $\mathrm{ES}^{\mathrm{GK}}$ in this paper, by the definition of modal reduct and world view.

Definition 3 (Modal Reduct of $\mathrm{ES}^{\mathrm{GK}}$)**.** *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.*

Definition 4 (World Views of $\mathrm{ES}^{\mathrm{GK}}$)**.** *$W$ 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 the **M**-cycle, a cycle in modal support graph of a program with an edge of **M**.

Definition 5 (Modal Supported Graph)**.** *Given an 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 e in the language of Π, there is a* literal node *labeled by e;*
- *for each objective literal e in the head of rule r, there is an unlabeled edge from the rule node r to literal node e;*
- *for each extended literal in the body of rule r, there is an edge labeled according to Table 2 going from literal node e to rule node r.*

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

Table 2
Labels of outcoming edges from literal nodes

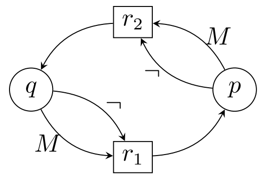

Figure 1: Modal support graph of Π_4 with M-cycle

Definition 6 (M-Cycle)**.** *A cycle in the MS graph of an epistemic logic program is called an* M-cycle *if M labels an edge within the cycle.*

Example 4 (M-cycle)**.** *Consider a program Π_4:*

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

Figure 1 shows the modal support graph of Π_4, which contains an M-cycle through q, rule (r_1), p and rule (r_2).

In Example 4, program Π_4 should has a unique world view $\{\{p\}, \{q\}\}$ under the semantics of $\mathrm{ES}^{\mathrm{GK}}$. However, many researchers claimed their disagreement against this example, including Yi-Dong shen [5] and Yuanling Zhang [16].

The aim of justified semantics of ES, which we called $\mathrm{ES}^{\mathrm{ZZ}}$ in this paper, is to develop an intuitive understanding of the **M** operator and get rid of circular justification in a stronger sense. It refines 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.

$\mathrm{ES}^{\mathrm{ZZ}}$ provides a classical method to define the views with a maximal guess of epistemic negations.

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

Table 3
General Modal Reduct of Maximal View

Definition 7 (Maximal Views). *Let Π be an ES program and $W = \{A_1, \dots, A_n\}$, where A_is are belief sets. A disjunctive program Π^W is called the* general modal reduct *$W = \{A_1, \dots, A_n\}$ if it is obtained by eliminating every subjective literals in Π with the transformation in Table 3. We call W a maximal view if W is the collection of all answer sets of Π^W.*

In addition to the maximal view, the justified view requires all belief sets in a world view to be justified, which means all subjective literals satisfied by a belief set are not only supported by themselves.

Definition 8 (Disjunction Reduct). *Let Π be a positive disjunctive program and A be a consistent set of literals in the language of Π. The* disjunction reduct *of Π w.r.t. A, denoted by $\Pi^{A,\vee}$, is a program obtained from Π removing all literals not in A from the head of all the rules in Π.*

The intuitive meaning of disjunction reduct is that if a literal l in the head of rule r is not contained in a belief set A, then l does not affect other literals in $head(r)$.

Definition 9 (Modal Operator Interpretation). *Let Π be a program with epistemic defaults, $W = \{A_1, \dots, A_n\}$ be a collection of belief sets. The mapping ρ from subjective literal s and belief set A_i is defined as $\rho(s, A_i)$ for all $A_i, A_j \in W$:*

- *if $W \models \mathbf{K}e$, $\rho(\mathbf{K}e, A_i) = \{e_i\}$;*
- *if $W \models \neg\mathbf{K}e$, $\rho(\neg\mathbf{K}e, A_i) = \{\neg e_j | A_j \not\models e\}$;*
- *if $W \models \mathbf{M}e$, $\rho(\mathbf{M}e, A_i) = \{e_j | A_j \models e\}$;*
- *if $W \models \neg\mathbf{M}e$, $\rho(\neg\mathbf{M}e, A_i) = \{\neg e_i\}$.*

Example 5. *Let $\Pi_5 = \{p \leftarrow \mathbf{M}q.\}$, $W = \{A_1 = \{q\}, A_2 = \{p, q\}\}$. The modal operator interpretation of q w.r.t A_1 is $\rho(\mathbf{M}q, A_1) = \{q_1, q_2\}$.*

Definition 10 (Modal Reduct for ES^{ZZ}). *Consider an ES program Π, a collection W of belief sets $\{A_1, \dots, A_n\}$. The* modal reduct *of Π w.r.t. W, A_i, and ρ, denoted as $\Pi^{W,A_i,\rho}$, is derived by following steps.*

1. *Renaming each objective literal e not occurring in any subjective literal in Π by e_i;*
2. *removing rules whose body contains a subjective literal s that $W \not\models s$;*
3. *replacing every occurrence of subjective literal s in rule r or its copies by a literal in $\rho(s, A_i)$.*

Example 6 (Continuing Example 5)**.** *The modal reduct of Π_5 w.r.t. W, A_1, ρ is*

$$p_1 \leftarrow q_1.$$
$$p_1 \leftarrow q_2.$$

Definition 11 (Justified Views). *Consider Π be a program with epistemic defaults, $W = \{A_1, \dots, A_n\}$ a collection of belief sets. Let $B = \{l_i | A_i \models l, 1 \le i \le n\}$, 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, i.e $\left(\left(\Pi^{W,A_i,\rho}\right)^B\right)^{B,\vee}$. W is a* justified view *of Π iff B is the unique stable model of $\bigcup_{i=1}^{n} \Pi_{full}^{\langle A_i, W\rangle}$.*

Definition 12 (World Views of ES^{ZZ})**.** *Let Π be an ES program and W a collection of belief sets. W is a* world view *of Π iff W is a justified view and maximal view of Π.*

3. Answer Set Programming with Defaults

This section introduce the syntax and semantics of ASP^{D}, which extends ASP with operator $\mathbf{C}$.

An ASP^{D} program Π is a finite collection of rules of the form

$$e_1 \text{ or} \cdots \text{or } e_k \leftarrow e_{k+1}, \cdots, e_m, d_1, \cdots, d_n. \tag{4}$$

where e_i are *extended literals* of the form l or $\neg l$, ls are *literals* in classical logic, d_i are *default literals* of the form $\mathbf{C}e$ or $\neg\mathbf{C}e$. Intuitively, $\mathbf{C}e$ means it is permitted (or not forbidden) to assume e is true, and $\neg\mathbf{C}e$ means e is proved to be not permitted. A rule containing operator $\mathbf{C}$ is a default rule.

Example 7. *Consider the following default logic programs Π_6*

$$bird(tweety). \tag{r_1}$$
$$flies(X) \leftarrow bird(X), \mathbf{C} flies(X). \tag{r_2}$$

and Π_6'

$$penguin(tweety). \tag{r_3}$$
$$\leftarrow penguin(X), flies(X). \tag{r_4}$$

Since $flies(tweety)$ does not conflict with rule (r_1) or rule (r_2), Π_6 concludes $flies(tweety)$. However, with the additional fact rule (r_3) and constraint rule (r_4) in Π_6', it is inconsistent to assume that $flies(tweety)$ is true. Thus $flies(tweety)$ is not in the consequent of $\Pi_6 \cup \Pi_6'$.

Example 7 illustrated the semantics of ASP^{D} we have defined intuitively. Now we will give a formal definition.

d	$\langle X,Y\rangle \models d$	$\langle X,Y\rangle \not\models d$
$\mathbf{C}l$	remove d	replace d with l
$\mathbf{C}\neg l$	remove d	replace d with $\neg l$
$\neg\mathbf{C}l$	replace d with $\neg l$	delete the rule
$\neg\mathbf{C}\neg l$	replace d with l	delete the rule

Table 4
Obtain $\Pi^{\langle X,Y\rangle}$ by eliminating defaults, where l is a positive literal without **C** or $\neg$.

Definition 13 (Satisfiability). *A* default interpretation *of program* Π *is a pair of consistent literal sets* $\langle X,Y\rangle$*, where* $X \subseteq Y \subseteq Literal(\Pi)$ *and* $Literal(\Pi)$ *is the set of all literals in the Herbrand universe of* Π*. Let* $\langle X,Y\rangle$ *be a default interpretation of* Π*,*

- $\langle X,Y\rangle \models l$ *iff* $l \in X$ *where* l *is a literal;*
- $\langle X,Y\rangle \models \neg e$ *iff* $X \not\models e$ *where* e *is an extended literal;*
- $\langle X,Y\rangle \models \mathbf{C}l$ *iff* $l \in Y$*;*
- $\langle X,Y\rangle \models \mathbf{C}\neg e$ *iff* $Y \not\models e$*;*
- $\langle X,Y\rangle \models \neg\mathbf{C}e$ *iff* $\langle X,Y\rangle \not\models \mathbf{C}e$*;*
- $\langle X,Y\rangle \models r$ *iff* $\exists e \in head(r) : \langle X,Y\rangle \models e$ *or* $\exists \phi \in body(r) : \langle X,Y\rangle \not\models \phi$*, where* r *is a rule in* Π*,* e *is an extended literal in the head of* r*,* ϕ *is an extended literal or default literal;*
- $\langle X,Y\rangle \models \Pi$ *iff* $\forall r \in \Pi : \langle X,Y\rangle \models r$*, and this default interpretation is called a* default model *of* Π*.*

In a default interpretation $\langle X,Y\rangle$ which satisifies an *ASP*$^{\mathrm{D}}$ program Π, if $l \in Y$, then $\mathbf{C}l$ is allowed by $\langle X,Y\rangle$. In that case, X should be an answer set of the default reduct $\Pi^{\langle X,Y\rangle}$, which is obtained by removing default literals from Π.

Definition 14 (Default Reduct). *Let* Π *be an ASP*^{D} *program,* $\langle X,Y\rangle$ *be a default model of* Π*. The* default reduct *of* Π *w.r.t.* $\langle X,Y\rangle$*, denoted by* $\Pi^{\langle X,Y\rangle}$*, is obtained by eliminating the occurrence of* $\mathbf{C}e$ *or* $\neg\mathbf{C}e$ *in rule* r *as Table 4.*

Now we use default reduct to define the default stable models of a program with default rules.

Definition 15 (Default Stable Models). *For an ASP*^{D} *program* Π*, a default model* $\langle X,Y\rangle$ *is a* default stable model *of* Π *iff*

1. X *is an answer set of* $\Pi^{\langle X,Y\rangle}$*,*
2. $Y \models \Pi^{\langle X,Y\rangle}$*,*
3. *let* $\Phi(Y,\Pi)$ *be a set of default literals of the form* $\mathbf{C}e$ *in* Π *that satisfied by* $\langle X,Y\rangle$*, there does not exist a consistent set of literals* Y' *that* $X \subseteq Y'$*,* $\Phi(Y,\Pi) \subset \Phi(Y',\Pi)$ *and* $Y' \models \Pi^{\langle X,Y'\rangle}$*.*

The set of all default stable models of a program Π *is denoted by* $DSM(\Pi)$*.*

Definition 16 (Stable Models of ASP$^{\mathrm{D}}$ programs). *Let* Π *be an ASP*^{D} *program. A literal set* A*, where* $A \subseteq Literal(\Pi)$*, is called a* stable model *of* Π *if* $\exists Y : \langle A,Y\rangle \in DSM(\Pi)$*. The set of all stable models of the program* Π *is denoted by* $SM(\Pi)$*.*

Let us take a close look at Definition 13 and 15. It shows that by the second condition of Definition 15, H is stable, i.e., X is minimal, while the third condition makes sure Y is maximal, thus the default stable model $\langle X,Y\rangle$ can satisfy as many extended literals in Π as possible.

Example 8 (Default Stable Models). *Consider the ASP*^{D} *program* Π_7

$$
\begin{aligned}
a &\leftarrow \mathbf{C}a.\\
b &\leftarrow \mathbf{C}b.\\
&\leftarrow a, b.\\
c &\leftarrow a.
\end{aligned}
$$

Consider default interpretations $M_1 = \langle\{a,c\},\{a,c\}\rangle$*, and* $M_2 = \langle\{b\},\{b\}\rangle$*,* $M_3 = \langle\{\},\{\}\rangle$*.* $\{a,c\}$ *is an answer set of* $\Pi_7^{M_1}$ *and* $\{b\}$ *is an answer set of* $\Pi_7^{M_2}$*. By Definition 16,* M_1 *and* M_2 *are default stable models of* Π_7*. For* M_3*, the* Y*-part of* M_3 *is* $Y_3 = \{\} \in AS(\Pi_7^{M_3})$*. However, there exists* $Y' = \{a,c\}$ *that* $Y' \models \Pi^{\langle\{\},\{a,c\}\rangle}$*, and* $\Phi(Y_3,\Pi) \subset \Phi(Y',\Pi_7)$*. Therefore,* M_3 *is not a default stable model of* Π_7*, and* $SM(\Pi_7) = \{\{a,c\},\{b\}\}$*.*

Example 8 is a typical instance of ASP$^{\mathrm{D}}$ that shows the feature of the operator **C**. Neither $\neg\neg e$ nor $\neg \sim e$ can capture the semantics of $\mathbf{C}e$. For example, program Π_7 has two stable models $\{a,c\}$ and $\{b\}$. Considering following ASP programs Π_7':

$$
\begin{aligned}
a &\leftarrow \neg\neg a.\\
b &\leftarrow \neg\neg b.\\
&\leftarrow a, b.\\
c &\leftarrow a.
\end{aligned}
$$

and Π_7'':

$$
\begin{aligned}
a &\leftarrow \neg \sim a.\\
b &\leftarrow \neg \sim b.\\
&\leftarrow a, b.\\
c &\leftarrow a.
\end{aligned}
$$

On the one hand, Π_7' has three answer sets, $\{a,c\}$, $\{b\}$, and $\{\}$. Apparently $\{\}$ is not a stable model of Π_7 that we want, thus $\neg\neg e$ can not represent defaults. On the other hand, Π_7'' is unsatisfiable, which means $\neg \sim e$ can not represent defaults neither.

ASP$^{\mathrm{D}}$ also provides a method to represent negative defaults, which is not provided by classical default logic.

Example 9 (Programs with Defaults and Negation). *Consider the ASP^D program* Π_8

$$a \leftarrow \neg \mathbf{C} b.$$

and the ASP^D program Π_9

$$a \leftarrow \mathbf{C} \neg b.$$

and default interpretations $M_1 = \langle \{a\}, \{a\} \rangle$, *and* $M_2 = \langle \emptyset, \{b\} \rangle$.

For program Π_8 *and* M_1, *we have* $\Pi_8^{M_1} = \{a \leftarrow \neg b.\}$ *and* $\{\{a\}\} = AS(\Pi_8^{M_1})$. *However, there is another default interpretation* $M_1' = \langle \{a\}, \{a, b\} \rangle$ *that* $\Pi_8^{M_1'} = \{\}$ *and* $\{a, b\} \models \Pi_8^{M_1'}$, *and* $\Phi(\{a, b\}, \Pi_8) = \{\mathbf{C} b\}$ *while* $\Phi(\{a\}, \Pi_8) = \emptyset$. *By the third condition in Definition 15,* M_1 *is not a default stable model of* Π_8. *Because* $\{a\}$ *is not an answer set of* $\Pi_8^{M_1'}$, *by the first condition of Definition 16,* M_1' *is not a default stable model of* Π_8. *On the other hand,* $M_2 \in DSM(\Pi_8)$, *and* $\emptyset$ *is a stable model of* Π_8.

For program Π_9 *and default interpretation* M_2, *we have* $\emptyset = AS(\Pi_9^{M_2})$. *However, there exists a default interpretation* $M_2' = \langle \emptyset, \{a\} \rangle$, *and* $\Phi(\{a\}, \Pi_9) = \{\mathbf{C} \neg b\}$. *Meanwhile,* $\Phi(\{b\}, \Pi_9) = \emptyset$, $\Pi_9^{M_2'} = \{a.\}$, *thus* $\{a\} \models \Pi_9^{M_2'}$. *Therefore,* M_2 *is not a default stable model of* Π_9. *It is easy to check that* M_1 *is a default stable model of* Π_9.

Example 9 shows how does ASP$^{\text{D}}$ deal with the negation of defaults and how does the function Φ works.

Now, let us revisited Example 1. "p by default" can be described by an ASP$^{\text{D}}$ program containing only one rule:

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

It is easy to check that $\{p\}$ is the unique stable model of the program as we expect.

4. Epistemic Specifications with Defaults

This section introduces the syntax and semantics of ES$^{\text{D}}$, an extension of ES$^{\text{ZZ}}$.

A rule of ES$^{\text{D}}$ is of the form

$$e_1 \text{ or} \cdots \text{or } e_k \leftarrow e_{k+1}, \cdots, e_m, s_1, \cdots, s_n. \quad (5)$$

where l_i are literals, e_i are objective literals (or extended literals in ASP$^{\text{D}}$) of the form l or $\neg l$, s_is are subjective literals of the form $\mathbf{K}e$, $\neg \mathbf{K}e$, $\mathbf{M}e$, $\neg \mathbf{M}e$, $\mathbf{A}e$ or $\neg \mathbf{A}e$. Intuitively, $\mathbf{A}e$ means that e may be permitted.

Definition 17 (Satisfaction of $\mathbf{A}e$). *For a collection of default interpretations* W, $W \models \mathbf{A}e$ *iff* $\exists w \in W : w \models \mathbf{C}e$.

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

Table 5
Modal Reduct of ES with Epistemic Defaults.

Definition 18 (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 a default interpretation* $\langle X, Y \rangle$. *The* modal reduct *of* Π *w.r.t* W, *denoted by* Π^W, *is an ASP^D program obtained from* Π *as Table 5 by eliminating every subjective literal* s.

Definition 18 is a modification of the modal reduct of traditional Epistemic Specifications. The last two rows define the reduct of the new subjective operator $\mathbf{A}$. However, we still need a method to reduce the circular justifications: subjective interpretation and justified reduct.

Definition 19 (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* $\rho(s, A_i)$ *for all* $A_i, A_j \in W$:

- *if* $W \models \mathbf{K}e$, $\rho(\mathbf{K}e, A_i) = \{e_i\}$;
- *if* $W \models \neg\mathbf{K}e$, $\rho(\neg\mathbf{K}e, A_i) = \{\neg e_j | A_j \not\models e\}$;
- *if* $W \models \mathbf{M}e$, $\rho(\mathbf{M}e, A_i) = \{e_j | A_j \models e\}$;
- *if* $W \models \neg\mathbf{M}e$, $\rho(\neg\mathbf{M}e, A_i) = \{\neg e_i\}$;
- *if* $W \models \mathbf{A}e$, $\rho(\mathbf{A}e, A_i) = \{\mathbf{C}e_j | A_j \models \mathbf{C}e\}$;
- *if* $W \not\models \mathbf{A}e$, $\rho(\mathbf{A}e, A_i) = \{\mathbf{C}e_i\}$;
- *if* $W \models \neg\mathbf{A}e$, $\rho(\neg\mathbf{A}e, A_i) = \{\neg\mathbf{C}e_i\}$;
- *if* $W \not\models \neg\mathbf{A}e$, $\rho(\neg\mathbf{A}e, A_i) = \{\neg\mathbf{C}e_j | A_j \not\models \neg\mathbf{C}e\}$.

In Definition 19, 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. For a subjective literal s with classical epistemic operators, the justified reduct of s is ignored if $W \not\models s$, because circular justification is not permitted in this situation. However, for s with operator $\mathbf{A}$ or $\neg\mathbf{A}$, s need to be justified whenever $W \models s$ or not.

Example 10 (Subjective Interpretation). *Considering program* Π_{10} *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* $\rho(\mathbf{A}p, A_i) = \mathbf{C}p_1$ *and* $\rho(\mathbf{A}q, A_i) = \mathbf{C}q_2$ *for* $i \in \{1, 2\}$.

Definition 20 (Justified Reduct). *Consider an ES^D program Π, a collection W of belief sets $\{A_1, \dots, A_n\}$. 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 rule r if there exists a subjective literal $s \in body(r)$ with operator $\mathbf{K}$, $\neg\mathbf{K}$, $\mathbf{M}$ or $\neg\mathbf{M}$ if $W \nvDash s$;*
2. *replacing subjective literals s in rule r or its copies with literals in $\rho(s, A_i)$ if $W \vDash s$;*
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.

Definition 21 (Justified View). *Consider Π be a program with epistemic defaults, $W = \{A_1, \dots, A_n\}$ a collection of default interpretations. Let $B = \{l_i | A_i \vDash l, 1 \leq i \leq n\}$, $C = \{l_i | A_i \vDash \mathbf{C}l, 1 \leq i \leq n\}$, 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 stable model of $\bigcup_{i=1}^{n} \Pi_{full}^{\langle A_i, W\rangle}$.*

Example 11 (Justified Views of Π_{10}). *Consider program Π_{10} in Example 10. The justified reducts $\Pi_{10}{}^{\langle A_1, W\rangle} = \{p_1 \leftarrow \mathbf{C}q_2, \neg q_1. q_1 \leftarrow \mathbf{C}p_1, \neg p_1.\}$, $\Pi_{10}{}^{\langle A_2, W\rangle} = \{p_2 \leftarrow \mathbf{C}q_2, \neg q_2. q_2 \leftarrow \mathbf{C}p_1, \neg p_2.\}$. By Definition 21, $B = C = \{p_1, q_2\}$, which is the answer set of $\Pi_{10}{}_{full}^{\langle A_1, W\rangle} \cup \Pi_{10}{}_{full}^{\langle A_2, W\rangle}$. Thus W is a justified view of Π_{10}.*

Definition 22 (World View). *Let Π be a program with epistemic defaults, W be a collection of default interpretations. W is a* world view *of Π iff*

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

Example 12 (World View of Π_{10}). *Consider Program Π_{10} and $W_1 = \{A_1, A_2\}$ in Example 10. Since $W \vDash \mathbf{A}p \wedge \mathbf{A}q$, the modal reduct $\Pi_{10}^{W_1}$ is $\{p \leftarrow \neg q. q \leftarrow \neg p.\}$. A_1 and A_2 are the only default stable models of $\Pi_{10}^{W_1}$. Example 11 has shown that W_1 is a justified view of Π_{10}. Thus W_1 is a world view of Π_{10}.*

Consider $W_2 = \{\langle \emptyset, \emptyset\rangle\}$. The modal reduct $\Pi_{10}^{W_2} = \{p \leftarrow \mathbf{C}q, \neg q. q \leftarrow \mathbf{C}p, \neg p.\}$, and $DSM(\Pi_{10}^{W_2}) = W_2$. The justified reduct of Π_{10} w.r.t. W_2 is $\{p_1 \leftarrow \mathbf{C}q_1, \neg q_1. q_1 \leftarrow \mathbf{C}p_1, \neg p_1.\}$. Thus W_2 is also a world view of Π_{10}.

5. Relation with ES^{GK}

This section will compare the semantics of ES^D to the one of ES^{GK}, which we have introduced in Section 2.

highest conviction

$\mathbf{K}l$	↑	$\neg\mathbf{M}l$
l	│	$\neg l$
$\mathbf{M}l$	↓	$\neg\mathbf{K}l$

lowest conviction

Figure 2: Preference Relation of Subjective Literals

Example 13 (Compare $\mathbf{M}$-cycle). *Consider an ES program Π_{11}*

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

Let $W_1 = \{p\}$, $W_2 = \{\emptyset\}$.

Under the semantics of ES^{GK}, the modal reducts of Π_{11} are $\Pi_{11}^{W_1} = \{p.\}$ and $\Pi_{11}^{W_2} = \{p \leftarrow \neg\neg p.\}$. Because $W_1 = AS(\Pi_{11}^{W_1})$, $W_2 \neq AS(\Pi_{11}^{W_2})$, W_1 is a world view of Π while W_2 is not.

Kahl [4] assumes that a rational agent should prefer to believe the subjective literals $\mathbf{M}l$ than l than $\mathbf{K}l$. The epistemic negation defined by Shen [5] shows the same preference relation. This preference relation makes a rule with directly $\mathbf{M}$-cycle works like a justification part in a normal default rule. More generally, a program with $\mathbf{M}$-cycles works like a default theory.

Because the definition of the justified reduct and modal reduct of operator $\mathbf{K}$ and $\mathbf{M}$ is equal to Yan Zhang and Yuanlin Zhang [16], the circular justification of $\mathbf{M}$ and $\mathbf{K}$ 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.”, and other rules or belief sets should justify the possibility of l .

A further observation of the unjustified world views caused by operator $\mathbf{M}$ reminds us that the semantics of modal operator $\mathbf{M}$ may be ambiguous. In classical ES programs, the $\mathbf{M}$-cycle of one rule is often used to express default information. For example, the traditional description “a bird can fly by default” is usually expressed in ES programs by the following rule:

$$fly(X) \leftarrow bird(X), \mathbf{M}fly(X).$$

However, for a bird *tweety*, the only justification of subjective literal $\mathbf{M}fly(tweety)$ is exactly this rule and the unique belief set that contains $fly(tweety)$. It means $\mathbf{M}fly(tweety)$ is not justified by this belief set.

Example 14 (Continuing Example 3). *Consider the following extension Π_{12} of the program in Example 3.*

$$alarm \leftarrow \mathbf{M}dangerous. \qquad (r_1)$$
$$alert \leftarrow \mathbf{A}dangerous. \qquad (r_2)$$

$$dangerous \leftarrow alarm. \qquad (r_3)$$

Rule (r_3) *means the sentinel will belief it is dangerous if the alarm has been raised. This looks reasonable because the alarm can be raised by other sentinels. Rule* (r_1) *and rule* (r_3) *constitute an* **M***-cycle, and by the natural language description, alarm should not exist in any belief sets of the world views of* Π_{12}.

Example 3 shows the difference between the representation of the introspection about permitted and possibilities. With the idea in this example, "a bird can fly by default" should be represented in ES^{D} by the following rule:

$$fly(X) \leftarrow bird(X), \mathbf{A} fly(X).$$

This rule forms an **A**-cycle, which can be defined as the **M**-cycle in Section 2.

Definition 23 (**A**-cycle)**.** *For a subjective literal s of the form* $\mathbf{A}e$ *or* $\neg\mathbf{A}e$ *in a rule r, label the edge from e to the objective literals* $e' \in head(r)$ *in MS graphs by the leading modal operators* **A** *or* $\neg\mathbf{A}$*. A cycle in the MS graph of an epistemic logic program is called an* **A**-cycle *if* **A** *labels an edge within the cycle.*

Example 15 (A-cycle)**.** *Consider program* Π_{13}*:*

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

Let $W = \{A_1 = \langle\{p\},\{p\}\rangle\}$*. The modal reduct of* Π_{13} *w.r.t W is* $\{p.\}$*, which means* $W = DSM(\Pi_{13}^W)$*. The justified reduct of* Π_{13} *w.r.t.* $\langle A_1, W\rangle$ *is* $\Pi_{13}^{\langle A_1,W\rangle} = \{p_1 \leftarrow \mathbf{C}p_1.\}$*, and* $\langle\{p_1\},\{p_1\}\rangle$ *is the unique default stable model of it, which means W is a justified view of* Π*. As a result, W is a world view of* Π_{13}*.*

With close observation of Example 13 and Example 15, we can find that although **M**-cycles's semantics are defined differently, operator **A** provides a method to represent defaults information, which **M**-cycles represent under the semantics of $\mathrm{ES}^{\mathrm{GK}}$.

Definition 24 (Default View Image)**.** *For a view W of an* ES^{GK} *program* Π*, the default view image* $\widehat{W}$ *of W is a collection of default interpretations that*

$$\widehat{W} = \{\langle A, A\rangle | A \in W\} \qquad (6)$$

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*

$$p \leftarrow \mathbf{M}p, B. \qquad (7)$$

where B is a collection of objective literals or extended subjective literals, 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* ES^{GK} *semantics if and only if its default view image* $\widehat{W}$ *is a world view of* Π' *under the semantics of* ES^D*.*

Proof. Let rule r be a rule in Π of the form (7), 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 $\mathrm{ES}^{\mathrm{GK}}$ 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 18, 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 occur in any **M**-cycles, $\neg\neg l$ does not support l. According to the definition of satisfiability in $\mathrm{ES}^{\mathrm{GK}}$, 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 **M**, the modal reduct of s by Definition 3 is equal to the one by Definition 18.
- 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 22.
- 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 stable models of $\Pi'^{\widehat{W}}$. It shows that W is a world view of Π' under the semantics in Definition 22.

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

- As shown 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 $\widehat{W} \not\models \mathbf{A}p$, $\mathbf{A}p$ is replaced by $\mathbf{C}p$ 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 $\Pi^W = \Pi'^{\widehat{W}}/\{p \leftarrow \mathbf{C}p, B.\} \cup \{p \leftarrow \neg\neg p, B.\}$, 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 $\mathbf{M}$-cycle.

Theorem 1 (Relationship between $\mathbf{M}$-cycle and $\mathbf{A}$-cycle)**.** *Let Π be an arbitrary ES^{GK} program, Π' be an ES^D 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 $\mathbf{M}$-cycle. A view W of Π under the semantics of ES^{GK} 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 shown in the proof of Proposition 1, it needs to prove that the modal reducts of rules in $\mathbf{M}$-cycle and translated $\mathbf{A}$-cycle under two semantics respectively are equal. The proof needs to consider the 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 ES^{GK} program containing following rules*

$$q_1 \leftarrow \mathbf{M}p, B_1. \qquad (r_1)$$
$$q_2 \leftarrow q_1, B_2. \qquad (r_2)$$
$$\dots$$
$$q_i \leftarrow q_{i-1}, B_i. \qquad (r_i)$$
$$p \leftarrow q_i, B_{i+1}. \qquad (r_{i+1})$$

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

$$q_1 \leftarrow \mathbf{A}p, B_1. \qquad (r'_1)$$
$$q_2 \leftarrow q_1, B_2. \qquad (r'_2)$$
$$\dots$$
$$q_i \leftarrow q_{i-1}, B_i. \qquad (r'_i)$$
$$p \leftarrow q_i, B_{i+1}. \qquad (r'_{i+1})$$

Proof. According to the proof of Proposition 1, rules (r_2) $,\dots,$ (r_{i+1}) are equivalent to rules (r'_2) $,\dots,$ (r'_{i+1}). Thus we only need to prove the equivalence of rule (r_1) and (r'_1) when $W \models B_1$. For the soundness part, considering following situations:

- If $W \models \mathbf{M}p$, the modal reduct of r_1 is $q_1 \leftarrow B_1.$. Meanwhile, for every $\widehat{A_j} \in \widehat{W}$ the justified reduct of r'_1 is $q_{1j} \leftarrow \mathbf{C}p_i, B_{1j}.$. If $\mathbf{M}p$ is concluded by $\Pi/\{r_1, \dots, r_{i+1}\}$, then according to the definition of justified view, p_i is justified by $(\Pi'/\{r'_1, \dots, r'_{i+1}\})^{\langle \widehat{A_j}, \widehat{W} \rangle}_{full}$. Otherwise, $\mathbf{M}p$ is satisfied only when $B_1, \dots, B_{i+1}$ are satisfied by W, which means p_j is justified by the justified reduct of $r'_1, \dots, r'_{i+1}$. As a result, p_j is justified for every $\widehat{A_j} \in \widehat{W}$, $\widehat{W}$ is a world view of Π'.
- If $W \not\models \mathbf{M}p$, the modal reduct of r_1 is $q_1 \leftarrow \neg\neg B_1.$. Because $\forall A_i \in W : A_i \not\models p$, p must not be consistent with Π/r, thus $\forall \widehat{A_i} \in \widehat{W} : A_i \not\models \mathbf{C}p$ and $W \not\models \mathbf{A}p$, r' is deleted from Π' in the modal reduct. As a result, $\widehat{W}$ is a justified view of Π' and equals to the collection of default equilibrium models of $\Pi'^{\widehat{W}}$, which means $\widehat{W}$ is a world view of Π'.

For the completeness part, consider following situations:

- if $\widehat{W} \models \mathbf{A}p$, thus the modal reduct of r'_1 w.r.t. $\widehat{W}$ is $q_1 \leftarrow B$. By the definition of default view image, $\widehat{W}$ also satisfies p, thus $\forall A_i \in W : A_i \models p, W \models \mathbf{M}p$. The modal reducts of Π and Π' are equal, thus W is a world view of Π.
- if $\widehat{W} \not\models \mathbf{A}p$, r' is deleted in the modal reduct $\Pi'^{\widehat{W}}$ and $\forall A_i \not\models_{\Pi'^{\widehat{W}}} \mathbf{C}p$, thus $\forall A_i \in W : A_i \not\models p, W \not\models \mathbf{M}p$, the modal reduct of r_1 is $q_1 \leftarrow \neg\neg p, B_1.$, thus the modal reduct $\Pi^W = \Pi'^{\widehat{W}} \cup \{q_1 \neg\neg p, B_1.\}$ 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, Proposition 2 holds.

□

Example 16 (Program with NAF and $\mathbf{M}$-cycle)**.** *Consider a program Π_{14}:*

$$p \leftarrow \neg q.$$
$$p \text{ or } q \leftarrow \mathbf{M}q.$$

By the definition of world view of ES^{GK}, the only world view of Π_{14} is $\{\{p\}, \{q\}\}$.

The corresponding ES^D program is Π'_{14}

$$p \leftarrow \neg q.$$
$$p \text{ or } q \leftarrow \mathbf{A}q.$$

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'_{14}. *For* W_2 *consider following situations:*

- *assume* $\widehat{W_2} = \{\langle\{p\},\{p,q\}\rangle\}$, $\widehat{W_2} \vDash \mathbf{A}q$, *then* $W_2 \neq DSM(\Pi'^{\widehat{W_2}}_{14})$;
- *assume* $\widehat{W_2} = \{\langle\{p\},\{p\}\rangle\}$, $\widehat{W_2} \nvDash \mathbf{A}q$, *then* $W_2 \neq DSM(\Pi'^{\widehat{W_2}}_{14})$,

thus $\widehat{W_2}$ *is not a world view of* Π'_{14}. *It can be showed* W_3, W_4 *are not world views of* Π'_{14}, *which means* $\widehat{W_1}$ *is the only world view of* Π'_{14}.

6. Conclusion

In this paper, with some examples, we illustrate that the defaults originated from permitted cannot be represented convincingly via the existing ASP and ES languages, and hence present logic programming languages to express defaults originated from permitted. Especially, we also compared the ability of expression and semantics of this language with $\mathrm{ES}^{\mathrm{GK}}$ and proposed a translation from programs of $\mathrm{ES}^{\mathrm{GK}}$ to programs of our language with epistemic defaults. It shows that the new language can also provide to separate the representation of permitted and possibilities, which can eliminate the ambiguity of $\mathbf{M}$ in $\mathrm{ES}^{\mathrm{GK}}$ and other similar languages for Epistemic Specifications.

In the future, we are intend to find a simplified definition of justified views for a more intuitive semantics. We are then planning to analyze the computational complexity of solving and develop an algorithm with acceptable efficiency for our further study on the application of our language. After that, we will do some further research on the semantics of ASP^{D} and ES^{D}, and see if their semantics can be captured by classical ASP and ES.

Acknowledgments

The work was supported by the Pre-research Key Laboratory Fund for Equipment (Grant No. 6142101190304).

References

[1] M. Gelfond, Y. Kahl, Knowledge Representation, Reasoning, and the Design of Intelligent Agents, Cambridge University Press, 2013.

[2] M. Gelfond, V. Lifschitz, The stable model semantics for logic programming, in: Proceedings of the Fifth International Conference and Symposium on Logic Programming, volume 2, MIT Press, Cambridge, Massachusetts, USA, 1988, pp. 1070–1080.

[3] M. Gelfond, New Semantics for Epistemic Specifications, in: Proceedings of the 11th International Conference on Logic Programming and Nonmonotonic Reasoning, volume 6645, Springer, Vancouver, Canada, 2011, pp. 260–265.

[4] P. T. Kahl, Refining The Semantics For Epistemic Logic Programs, Phd, Texas Tech University, 2014.

[5] Y.-D. Shen, T. Eiter, Evaluating epistemic negation in answer set programming, Artificial Intelligence 237 (2016) 115–135.

[6] E. I. Su, L. F. del Cerro, A. Herzig, Autoepistemic equilibrium logic and epistemic specifications, Artif. Intell. 282 (2020) 103249.

[7] R. Reiter, A logic for default reasoning, Artificial Intelligence 13 (1980) 81–132.

[8] M. Gelfond, H. Przymusinska, V. Lifschitz, M. Truszczynski, Disjective defaults, in: Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann, 1991, pp. 230–237.

[9] K. Konolige, On the relation between default and autoepistemic logic, Artificial Intelligence 35 (1988) 343–382.

[10] G. Gottlob, Translating default logic into standard autoepistemic logic, J. ACM 42 (1995) 711–740.

[11] M. Truszczynski, Modal interpretations of default logic, in: Proceedings of the 12th International Joint Conference on Artificial Intelligence, 1991, pp. 393–398.

[12] P. Cabalar, D. Lorenzo, New insights on the intuitionistic interpretation of default logic, in: Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004, IOS Press, 2004, pp. 798–802.

[13] M. Gelfond, V. Lifschitz, Classical negation in logic programs and disjunctive databases, New Gener. Comput. 9 (1991) 365–386.

[14] V. Lifschitz, Thirteen definitions of a stable model, in: Fields of Logic and Computation, Essays Dedicated to Yuri Gurevich on the Occasion of His 70th Birthday, volume 6300 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 488–503.

[15] Y. Chen, H. Wan, Y. Zhang, Y. Zhou, dl2asp: Implementing default logic via answer set programming, in: T. Janhunen, I. Niemelä (Eds.), Logics in Artificial Intelligence - 12th European Conference, JELIA 2010, Helsinki, Finland, September 13-15, 2010. Proceedings, volume 6341 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 104–116.

[16] Y. Zhang, Y. Zhang, Epistemic specifications and conformant planning, AAAI Workshop - Technical Report WS-17 (2017) 781–787.