Recent Epistemic Extensions of Answer Set Programming

Ezgi Iraz SU
Sinop University, Turkey

LIRICA Seminar 2021
LIS, Marseille, France

31 May 2021

Outline

ASP lacks expressivity [Gelfond 1991]

Example (Gelfond's eligibility program Π_G, ASP-version)

% university rules to decide eligibility for scholarship (X: arbitrary applicant)

$$\mathtt{eligible}(X) \leftarrow \mathtt{highGPA}(X).$$
$$\mathtt{eligible}(X) \leftarrow \mathtt{fairGPA}(X)\,,\ \mathtt{minority}(X).$$
$$\sim\mathtt{eligible}(X) \leftarrow \sim\mathtt{highGPA}(X)\,,\ \sim\mathtt{fairGPA}(X).$$

% disjunctive info: an applicant data for a specific student called Mike

$$\mathtt{highGPA}(\mathrm{mike})\ \mathrm{or}\ \mathtt{fairGPA}(\mathrm{mike}).$$

% if eligibility not determined, then interview required (ASP attempt)

$$\mathtt{interview}(X) \leftarrow \mathrm{not}\ \mathtt{eligible}(X)\,,\ \mathrm{not}\ \sim\mathtt{eligible}(X).$$

Quantification problem in ASP

Example (Mike's eligibility situation, ASP-version)

Π_G :

1. $\mathtt{eligible} \leftarrow \mathtt{highGPA}$.
2. $\mathtt{eligible} \leftarrow \mathtt{fairGPA}, \mathtt{minority}$.
3. $\sim\mathtt{eligible} \leftarrow \sim\mathtt{fairGPA}, \sim\mathtt{highGPA}$.
4. $\mathtt{highGPA} \text{ or } \mathtt{fairGPA} \leftarrow$.
5. $\mathtt{interview} \leftarrow \text{not } \mathtt{eligible}, \text{not } \sim\mathtt{eligible}$.

has the following answer sets

$$AS(\Pi_G) = \left\{ \{\mathtt{highGPA}, \mathtt{eligible}\}, \{\mathtt{fairGPA}, \mathtt{interview}\} \right\}.$$

$\Rightarrow$ $\mathtt{eligible}$? and $\sim\mathtt{eligible}$? *undetermined*

$\Rightarrow$ $\mathtt{interview}$? *undetermined* too...

So, epistemic modalities are required in ASP...

Example (Mike's eligibility situation, ASP-version)

Π_G :

1. $\mathtt{eligible} \leftarrow \mathtt{highGPA}.$
2. $\mathtt{eligible} \leftarrow \mathtt{fairGPA}, \mathtt{minority}.$
3. $\sim\mathtt{eligible} \leftarrow \sim\mathtt{fairGPA}, \sim\mathtt{highGPA}.$
4. $\mathtt{highGPA}\ \mathrm{or}\ \mathtt{fairGPA} \leftarrow .$
5. $\mathtt{interview} \leftarrow \mathrm{not}\ \mathtt{eligible}, \mathrm{not}\ \sim\mathtt{eligible}.$

Therefore:

$$\Pi_G \not\models \mathtt{eligible}$$

$$\Pi_G \not\models \sim\mathtt{eligible}$$

$$\Pi_G \not\models \mathtt{interview} \quad \text{(counter-intuitive!)}$$

$\Rightarrow$ **wanted:** quantification over possible answer sets. . .

Gelfond's solution [Gelfond 1991]

Example (Mike's scholarship eligibility revisited, EASP-version)

Π_{KG}:

1. eligible ← highGPA.
2. eligible ← minority, fairGPA.
3. ∼eligible ← ∼fairGPA, ∼highGPA.
4. highGPA or fairGPA ← .
5. interview ← not K eligible, not K ∼eligible.

will have slightly different answer sets

$$AS(\Pi_{KG}) = \big\{ \{\text{highGPA}, \text{eligible}, \text{interview}\}, \{\text{fairGPA}, \text{interview}\} \big\}.$$

⇒ eligible? and ∼eligible? *unknown*

⇒ interview? *YES* (intuitive!)

ASP lacks expressivity ctd. [Gelfond 2011]

Example (Closed World Assumption (CWA), ASP-version)

% p is assumed to be false if there is no evidence to the contrary. (ASP attempt)

$$\sim p \leftarrow \texttt{not}\, p. \qquad (r_1)$$

Consider: $\Pi = \{r_1, r_2\}$ where $r_2 = p \,\texttt{or}\, q$.

has the following answer sets

$$AS(\Pi) = \Big\{\{p\}, \{\sim p, q\}\Big\}.$$

⇒ p? *unknown*

⇒ but also $\sim p$? *unknown* (counter-intuitive)

upshot: again quantification through answer sets is required....

Two different solutions [Gelfond 2011, Shen et al. 2016]

Example (CWA revisited , EASP-version)

% p is assumed to be false if there is no evidence to the contrary. (EASP attempt)

(r_1) $\sim p \leftarrow \texttt{not}\ \mathrm{M}\, p.$ Gelfond's approach [LPNMR, 2011]

(r_2) $\sim p \leftarrow \texttt{not}\ \mathrm{K}\, p.$ Shen and Eiter's approach [AIJ, 2016]

- Consider: $\mathrm{K}\,\Pi = \{r_2, r_3\}$ where $r_3 = p\ \texttt{or}\ q$.
- $\mathrm{K}\,\Pi$ has the unique answer set

$$AS(\mathrm{K}\,\Pi) = \Big\{\{\sim p, q\}\Big\}.$$

- Now, result is intuitive!

Outline

Language of ES18 [Kahl et al., ICLP 2018]

extended the language of ASP by epistemic modalities K and M

idea: quantify over all candidate answer sets and correctly represent *incomplete* information (*non-provability*)

K p – – – p is *known* to be true.

M p – – – p may be *believed* to be true.

- **atoms:** (extended) *objective* and *subjective* literals

l	L	g	G
p | $\sim p$	l | not l	K l | M l	g | not g

where p ranges over $\mathbb{P}$.

  - strong negation $\sim$
  - default negation (aka, negation as failure) not

notation:

(ex-)$\mathbb{O}Lit$ — the set of all (extended) objective literals

(ex-)$\mathbb{S}Lit$ — the set of all (extended) subjective literals

Syntax of ES18

rule: a logical statement of the form $\texttt{head} \leftarrow \texttt{body}$

- a *rule* $\texttt{r}$ of ES18 is of the following form:

$$\texttt{l}_1 \texttt{ or } \ldots \texttt{ or } \texttt{l}_m \leftarrow \texttt{e}_1 , \ldots , \texttt{e}_n$$

  - $\texttt{head}(\texttt{r})$: disjunction of objective literals
  - $\texttt{body}(\texttt{r})$: conjunction of arbitrary literals

When $m=0$, $\texttt{head}(\texttt{r}) = \bot$ and $\texttt{r}$: *constraint* (headless rule)

- if $\texttt{body}(\texttt{r})$ of a constraint consists solely of extended sub. literals, i.e., $\texttt{G}_1 , \ldots , \texttt{G}_n$, then $\texttt{r}$: *subjective constraint*.
- e.g., $\bot \leftarrow \mathrm{K}\, p$; $\bot \leftarrow \mathrm{M}\, p , \texttt{not} \mathrm{K}\, q$; etc.

When $n=0$, $\texttt{body}(\texttt{r}) = \top$ and $\texttt{r}$: *fact* (bodiless rule).

program: finite collection of rules

- finite set of EASP rules = *epistemic specifications*
  = *epistemic logic programs* (ELPs)

Truth conditions of ES18

For nonempty $\mathcal{A} \subseteq 2^{\mathbb{O}Lit}$, $\mathtt{l} \in \mathbb{O}Lit$, $\mathtt{L} \in$ ex-$\mathbb{O}Lit$, and $\mathtt{g} \in \mathbb{S}Lit$,

- **truth conditions:**

$$
\begin{array}{lll}
\mathcal{A}, A \models \mathtt{l} & \text{if} & \mathtt{l} \in A; \\
\mathcal{A}, A \models \mathtt{not}\,\mathtt{l} & \text{if} & \mathtt{l} \notin A; \\
\mathcal{A}, A \models \mathsf{K}\,\mathtt{L} & \text{if} & \mathcal{A}, A' \models \mathtt{L} \text{ for every } A' \in \mathcal{A}; \\
\mathcal{A}, A \models \mathsf{M}\,\mathtt{L} & \text{if} & \mathcal{A}, A' \models \mathtt{L} \text{ for some } A' \in \mathcal{A}; \\
\mathcal{A}, A \models \mathtt{not}\,\mathtt{g} & \text{if} & \mathcal{A}, A \not\models \mathtt{g}.
\end{array}
$$

- **equivalences:**

$$
\begin{array}{lll}
\mathcal{A} \models \mathsf{M}\,\mathtt{l} & \text{iff} & \mathcal{A} \models \mathtt{not}\,\mathsf{K}\,\mathtt{not}\,\mathtt{l} \\
\mathcal{A} \models \mathtt{not}\,\mathsf{M}\,\mathtt{l} & \text{iff} & \mathcal{A} \models \mathsf{K}\,\mathtt{not}\,\mathtt{l}
\end{array}
$$

⇒ K and M are (1) dual and (2) interchangeable.

Kahl's reduct definition [Kahl, PhD thesis 2014]

Given $\mathcal{A} \subseteq 2^{\mathbb{O}Lit}$ and an epistemic logic program (ELP) Π:

- **K-reduct** $\mathtt{r}^{\mathcal{A}}$ of an ES rule $\mathtt{r}$ w.r.t. $\mathcal{A}$

extended subjective literal (G)	if *true* in $\mathcal{A}$	if *false* in $\mathcal{A}$
K l	replace by **l**	delete rule
not K l	remove literal	replace by not **l**
M l	remove literal	replace by not not **l**
not M l	replace by not **l**	delete rule

idea: eliminate K and M (whereas in ASP, we eliminate not !)

$$\Pi^{\mathcal{A}} = \{\mathtt{r}^{\mathcal{A}} \ : \ \mathtt{r} \in \Pi\}$$

remark:

K-reduct is rather complex and lacks an intuitive explanation.

Kahl et al.'s semantics approach [Kahl et al., ICLP 2018]

- First, define:

$$\mathtt{Ep}(\Pi) = \{\mathtt{not}\,\mathrm{K}\,\mathtt{l} \;:\; \mathrm{K}\,\mathtt{l} \text{ appears in } \Pi\} \cup \{\mathrm{M}\,l \;:\; \mathrm{M}\,l \text{ appears in } \Pi\}.$$

- Then, take its subset w.r.t. $\mathcal{A} \subseteq 2^{\mathbb{O}Lit}$

$$\mathtt{Ep}(\Pi)\big|_{\mathcal{A}} = \Phi_{\mathcal{A}} = \{\mathtt{G} \in \mathtt{Ep}(\Pi) \;:\; \mathcal{A} \models \mathtt{G}\}.$$

- So, for a prototypical program

$$\Pi' = \{t \leftarrow \mathrm{K}\,p, \mathrm{M}\,q, \mathtt{not}\mathrm{K}\,s, \mathtt{not}\mathrm{M}\,t\},$$

- we have:

$$\mathtt{Ep}(\Pi') = \{\mathtt{not}\mathrm{K}\,p, \mathrm{M}\,q, \mathtt{not}\mathrm{K}\,s, \mathrm{M}\,t\}.$$

- given $\mathcal{A}' = \{\{p, s\}, \{t, s\}\}$:

$$\mathtt{Ep}(\Pi')\big|_{\mathcal{A}'} = \Phi_{\mathcal{A}'} = \{\mathtt{not}\mathrm{K}\,p, \mathrm{M}\,t\}.$$

Kahl et al.'s world views (K-WV) [Kahl et al., ICLP 2018]

- Finally, $\mathcal{A}$ is a *K-world view* (K-WV) of a "constraint-free" Π if:

fixed point property

1. $\mathcal{A} = \texttt{AS}(\Pi^{\mathcal{A}}) = \{A : A \text{ is an answer set of } \Pi^{\mathcal{A}}\}$

knowledge-minimising property

2. there is no $\mathcal{A}'$ such that $\mathcal{A}' = \texttt{AS}(\Pi^{\mathcal{A}'})$ and $\Phi_{\mathcal{A}'} \supset \Phi_{\mathcal{A}}$.

Why “constraint-free” restriction?

- in ASP, constraints function regularly:
  - at most rule out answer-sets, violating them.
- in ES18, behaviour of constraints is not monotonic.

Example

Consider the following EASP rules:

$$a \;\texttt{or}\; b \leftarrow . \qquad (r_1)$$

$$c \leftarrow \mathrm{K}\, a. \qquad (r_2)$$

$$\leftarrow \texttt{not}\; c. \qquad (r_3)$$

- $\Pi = \{r_1, r_2\}$ has a unique K-WV: $\big\{\{a\}, \{b\}\big\}$.
- if we add r_3, then we expect to have no K-WVs, but:
- $\Pi = \{r_1, r_2, r_3\}$ has a unique K-WV: $\big\{\{a, c\}\big\}$.

Some new language constructs in ES18

- So, effect of a constraint $\mathtt{r}$ over world views
  - may be additive or subtractive

- Solution by Kahl and Leclerc: *world view constraints* (WVCs)
  - in the form of subjective constraints
  - replace $\leftarrow$ by $\overset{wv}{\leftarrow}$
    - $\overset{wv}{\leftarrow}\varphi$ is read: "it is not a world view if it satisfies φ"

Ex: $\overset{wv}{\leftarrow}\mathtt{not}\,\mathrm{K}\,p$: "it is not a world view if p is not known"
(any world view satisfying $\mathtt{not}\,\mathrm{K}\,p$ should be eliminated)

WVCs can solve the constraint problem?

...**to some extent!** because only works for subjective constraints

- what about for $\leftarrow \mathrm{K}\, p\ ,\ q$?

Definition (Kahl and Leclerc's restricted solution)

Let Π be an ELP containing WVCs such that $\Pi = \Pi_0 \cup \Pi_{wvc}$

- Π_0 is a constraint-free part of Π.
- Π_{wvc}: set of all WVCs occurring in Π

Then, $\mathcal{A}$ is a K-WV of Π if

1. $\mathcal{A}$ is a K-WV of Π_0 and
2. $\mathcal{A}$ satisfies every constraint in Π_{wvc}.

Outline

Language of ES16 [Shen and Eiter, AIJ 2016]

differs from the language of ES18 as follows:

- instead of K and M, we have epistemic negation NOT
- NOT*p* (in ES16) corresponds to notK *p* (in ES18).

intuitive reading:

NOT *p* – – – *p* is *not proved* to be true.

- use the equivalences notnotK ≡ K and notK not ≡ M
- obtain the following equivalent transformations between:

ES18	**K**	not**K**	**M**	not**M**
ES16	notNOT	NOT	NOTnot	notNOTnot

- Programs of ES16-18 share the same syntax.

Shen and Eiter (SE)'s reduct definition

SE use $\mathtt{not}\mathsf{K}$ (epistemic negation $\mathtt{NOT}$) to minimise knowledge

- First, remember:

$$\mathtt{Ep}(\Pi) = \{\mathtt{not}\,\mathsf{K}\,\mathtt{l} \ : \ \mathsf{K}\,\mathtt{l} \text{ appears in } \Pi\} \cup \{\mathsf{M}\,\mathtt{l} \ : \ \mathsf{M}\,\mathtt{l} \text{ appears in } \Pi\}$$

- Then, given $\mathcal{A} \subseteq 2^{\mathbb{O}Lit}$ (we call it a *guess*),
  - take its subset $\Phi_{\mathcal{A}} = \{\mathtt{G} \in \mathtt{Ep}(\Pi) \ : \ \mathcal{A} \models \mathtt{G}\}$
- **SE-reduct** $\mathtt{r}^{\Phi_{\mathcal{A}}}$ of an ES rule $\mathtt{r}$ w.r.t. $\Phi_{\mathcal{A}}$

idea: eliminate K and M (aligning with K-reduct)

epis. negation ($\mathtt{G}$)	if $\mathtt{G} \in \Phi_{\mathcal{A}}$	if $\mathtt{G} \in \mathtt{Ep}(\Pi) \setminus \Phi_{\mathcal{A}}$
$\mathtt{not}\,\mathsf{K}\,\mathtt{l}$	replace by $\top$	replace by $\mathtt{not}\ \mathtt{l}$
$\mathsf{M}\,\mathtt{l}$	replace by $\top$	replace by $\mathtt{not}\ \mathtt{not}\ \mathtt{l}$

- next form

$$\Pi^{\Phi_{\mathcal{A}}} = \{\mathtt{r}^{\Phi_{\mathcal{A}}} \ : \ \mathtt{r} \in \Pi\}$$

New arrangement of SE-reduct

ext. sub. literal (`G`)	if *true* in $\mathcal{A}$	if *false* in $\mathcal{A}$
`K l`	replace by `not not l`	delete rule
`not K l`	remove literal	replace by `not l`
`M l`	remove literal	replace by `not not l`
`not M l`	replace by `not l`	delete rule

SE's semantics approach [SE, AIJ 2016]

$\mathcal{A}$ is a *SE-world view* (SE-WV) of an ELP Π if:

fixed point property

1. $\mathcal{A} = \texttt{AS}(\Pi^{\Phi_{\mathcal{A}}}) = \{A \ : \ A$ is an answer set of $\Pi^{\Phi_{\mathcal{A}}}\}$;

knowledge-minimising property

2. $\Phi_{\mathcal{A}}$ is maximal, i.e., for no other guess $\mathcal{A}'$, we have:
$\mathcal{A}' = \texttt{AS}(\Pi^{\Phi_{\mathcal{A}'}})$ and $\Phi_{\mathcal{A}'} \supset \Phi_{\mathcal{A}}$.

$\Rightarrow$ but SE-WVs cannot treat well with ELPs including constraints...

An example program with constraints

Example

Consider the following EASP rules:

$$a \text{ or } b \leftarrow . \qquad (r_1)$$

$$\leftarrow \texttt{not}\, \mathsf{K}\, a. \qquad (r_2)$$

- r_1 has a unique SE-WV: $\big\{\{a\},\{b\}\big\}$.
- if we consider it with r_2:
- $\Pi = \{r_1, r_2\}$ has a unique SE-WV: $\big\{\{a\}\big\}$.

Outline

Novelty offered by ES21 [Su, Jelia 2019]

- nondual epistemic operators K and $\hat{K}$
- more natural generalisation of ASP
  - our reduct definition is oriented to eliminate `not`
- knowledge minimisation technique from reflexive autoepistemic logic (nonmonotonic SW5, [Schwarz 1992])

Language of ES21

extended the language of ASP with epistemic modalities K and $\hat{K}$

- K and $\hat{K}$ are not dual: $\hat{K}$ is not equivalent to `notKnot`.
- literals (λ) :*objective* literals (l) and *subjective* literals (g)

l	**g**
p | $\sim p$	K l | $\hat{K}$ l

where p ranges over $\mathbb{P}$.

- ES21 rules are of the following form:

$$\lambda_1 \; \texttt{or} \ldots \texttt{or} \; \lambda_k \leftarrow \lambda_{k+1}, \ldots, \lambda_m, \texttt{not} \; \lambda_{m+1}, \ldots, \texttt{not} \; \lambda_n$$

  - *positive* rules — without negation as failure (NAF) `not`
- (pos.) EASP program: finite collection of (pos.) EASP rules

⇒ **ASP**: EASP in which literals are restricted to objective literals

Positive ES21 programs

semantics: via *stable S5 models*

Definition (weakening of a point in an S5 model $\mathcal{A} \subseteq 2^{\mathbb{O}Lit}$)

Given a (subset) map $s : \mathcal{A} \to 2^{\mathbb{O}Lit}$ such that $s(A) \subseteq A$ for every $A \in \mathcal{A}$, $s \neq id$ on $\mathcal{A}$ and $s|_{\mathcal{A}\setminus\{A\}} = id$,
$\langle s[\mathcal{A}], s(A)\rangle$: weakening of $\mathcal{A}$ at a point $A \in \mathcal{A}$.
notation: $\langle s[\mathcal{A}], s(A)\rangle \lhd \langle \mathcal{A}, A\rangle$.

Ex: $\left\{\underline{\emptyset}, \{q,r\}\right\} \lhd \left\{\underline{\{p\}}, \{q,r\}\right\}$.

Definition (nonmono. satisfaction reln $\models^*$ minimising truth)

Given a pointed S5 model $\langle \mathcal{A}, A\rangle$ and an EASP program Π,
$\mathcal{A}, A \models^* \Pi$ iff

1. $\mathcal{A}, A \models \Pi$ and
2. $s[\mathcal{A}], s(A) \not\models \Pi$ for every map s viz. $\langle s[\mathcal{A}], s(A)\rangle \lhd \langle \mathcal{A}, A\rangle$.

Ex: $\left\{\{p\}, \underline{\{r\}}\right\} \models^* q \,\texttt{or}\, r$.

Definition (generalisation of answer set defn to EASP)

$\mathcal{A}$ is a *minimal model* of Π if $\mathcal{A}, A \models^* \Pi$ for every $A \in \mathcal{A}$.

Example

Consider the following positive program Σ:

$$p \;\texttt{or}\; q \leftarrow .$$
$$s \leftarrow q.$$
$$r \leftarrow \mathrm{K}\, p.$$

Claim: $\big\{\{p\},\{q,s\}\big\}$ is a minimal model of Σ: indeed,

- $\big\{\underline{\{p\}},\{q,s\}\big\} \models \Sigma$ while its only weakening $\big\{\underline{\emptyset},\{q,s\}\big\} \not\models \Sigma$.
- $\big\{\{p\},\underline{\{q,s\}}\big\} \models \Sigma$ while all its weakenings, i.e, $\big\{\{p\},\underline{\{q\}}\big\}$, $\big\{\{p\},\underline{\{s\}}\big\}$ and $\big\{\{p\},\underline{\emptyset}\big\}$ do not satisfy it.

$\{\{p,r\}\}$ and $\{\{q,s\}\}$ are the other (unintended) minimal models of Σ.

$\Rightarrow$ minimality of truth does not guarantee intuitive results.

A quick introduction to SW5 models

An **SW5 model** $\mathcal{M} = \langle W, R, V \rangle$ is a Kripke model in which

- W: non-empty set of possible worlds;
  - $W = C \cup \{a\}$: $C \neq \emptyset$.
- for every $w \in W$: $V(w) \subseteq \mathbb{P}$ — a valuation, i.e.,
  - a set of propositional variables
- $R \subseteq W \times W$ a binary relation on W.
  - xRy iff $y \in C$ or $x = y$.
  - $R = (W \times C) \cup (a, a)$.

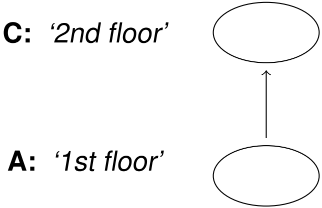
Cluster-decomposable Kripke models

Definition

C is a $\mathcal{T}$-cluster if $w\mathcal{T}u$ for every $w, u \in C$.

We can transform an SW5 model $\mathcal{M} = \langle W, \mathcal{T}, V \rangle$ into $\langle A, C, V \rangle$:

- C is a nonempty cluster
- $A = \emptyset$ or $A = \{a\}$
  - $\mathcal{T}(a) = W$ ('a can see any point in W including itself')
  - C can be accessed from every world in W — C is final!
  - but a point in C cannot access $a \in A$.



Nonmonotonic SW5

Minimal model semantics over SW5 $\Rightarrow$ 'nonmonotonicity'

Definition

$\mathcal{M} = (W, \mathcal{T}, V)$ is preferred over a valuated cluster (C, V) in SW5

- $W = C \cup \{a\}$: $a \notin C$;
- $\mathcal{T} = (W \times C) \cup \{(a, a)\}$;
- The valuations V agree on C;
- There exists $\varphi \in Prop$ s.t. $C \models \varphi$ and $\mathcal{M} \not\models \varphi$. (i.e., $a \not\models \varphi$.)

$\Rightarrow$ we write: $\mathcal{M} \succ (C, V)$.

Definition

(C, V) is a minimal model of a theory Γ in SW5 if

- $(C, V), x \models \Gamma$ for every $x \in C$ (i.e., $(C, V) \models \Gamma$);
- $\mathcal{M} \not\models \Gamma$ for every $\mathcal{M}$ s.t. $\mathcal{M} \succ (C, V)$.

Definition (**stable** S5 **model**)

Let $\mathcal{A}$ be an S5 model of a positive EASP program Π.
Then, $\mathcal{A}$ is a *stable S5 model* of Π if

truth-minimising condition

1. $\mathcal{A}$ is a minimal model of Π;

knowledge-minimising condition

2. any preferred SW5-extension of $\mathcal{A}$ is not a minimal model of Π.

(i.e., for every $A' \in 2^{\mathbb{P}} \setminus \mathcal{A}$, $\mathcal{A}, A' \not\models \Pi$ or $\mathcal{A}, s(A') \models \Pi$ for some subset map s satisfying $s(A') \subset A'$ and $s|_{\mathcal{A}} = id$.)

- 1st cnd: truth-minimality — intuition from ASP
- 2nd cnd: knowledge-minimality — intuition from reflexive autoepistemic logic (nonmonotonic SW5)

⇒ our special S5 models are now stable w.r.t. truth and knowledge.

Positive ES21 programs ctd.

Example

Consider the following epistemic logic program Σ once again:

$$p \,\texttt{or}\, q \leftarrow$$
$$s \leftarrow q$$
$$r \leftarrow \mathsf{K}\, p$$

Σ has 3 min. models: $\mathcal{A}_1 = \{\{p\}, \{q, s\}\}$, $\mathcal{A}_2 = \{\{p, r\}\}$ & $\mathcal{A}_3 = \{\{q, s\}\}$.

- $\mathcal{A}_2$ is not stable: it has a preferred model $\mathcal{A}_2' = \{\{p, r\}, \underline{\{q, s\}}\}_{SW5}$ ($\mathcal{A}_2' \succ \mathcal{A}_2$) and $\mathcal{A}_2'$ is also minimal.
- $\mathcal{A}_3$ is not stable: it has a preferred model $\mathcal{A}_3' = \{\{q, s\}, \underline{\{p\}}\}_{SW5}$ ($\mathcal{A}_3' \succ \mathcal{A}_3$) and $\mathcal{A}_3'$ is also minimal.
- any preferred model of $\mathcal{A}_1$ is not a minimal model of Σ.

∴ $\mathcal{A}_1$ is the unique stable S5 model of Σ.

What if Π is not positive?

then we first take the reduct!

- our reduct defn is oriented to eliminate NAF only as in ASP!

Definition (generalisation of the reduct definition of ASP)

Let Π be an epistemic logic program.
Let $\mathcal{A} \subseteq 2^{\mathbb{O}Lit}$ be nonempty and $A \in \mathcal{A}$.Then,

- the *reduct* $\Pi^{\langle \mathcal{A}, A \rangle}$ of Π w.r.t. $\langle \mathcal{A}, A \rangle$ is given by replacing every occurrence of $\mathtt{not}\,\lambda$ with
  - $\bot$ if $\mathcal{A}, A \models \lambda$ (for $\lambda = \mathtt{l}$ if $A \models \mathtt{l}$; for $\lambda = \mathtt{K\,l}$ if $\mathcal{A} \models \mathtt{K\,l}$);
  - $\top$ if $\mathcal{A}, A \not\models \lambda$ (for $\lambda = \mathtt{l}$ if $A \not\models \mathtt{l}$; for $\lambda = \mathtt{K\,l}$ if $\mathcal{A} \not\models \mathtt{K\,l}$).
- Thus, $\mathcal{A}$ is a *minimal model* of Π if

$$\mathcal{A}, A \models^* \Pi^{\langle \mathcal{A}, A \rangle} \text{ for every } A \in \mathcal{A}.$$

- The rest (knowledge-minimality) is the same.

Let's see an example!

Example

Consider the following EASP program Γ:

$$\begin{aligned} p &\leftarrow \texttt{not}\, {\sim}q \\ {\sim}q &\leftarrow \texttt{not}\, p \\ r &\leftarrow \texttt{not}\, \mathsf{K}\, p \end{aligned}$$

Claim: $\mathcal{A} = \big\{\{p, r\}, \{{\sim}q, r\}\big\}$ is a minimal model of Γ: indeed,

$$\Gamma^{\{\underline{\{p,r\}},\{\sim q,r\}\}} : \begin{aligned} p &\leftarrow \top \\ {\sim}q &\leftarrow \bot \\ r &\leftarrow \top \end{aligned} \qquad \Gamma^{\{\{p,r\},\underline{\{\sim q,r\}}\}} : \begin{aligned} p &\leftarrow \bot \\ {\sim}q &\leftarrow \top \\ r &\leftarrow \top \end{aligned}$$

- $\big\{\underline{\{p, r\}}, \{{\sim}q, r\}\big\} \models \Gamma^{\{\underline{\{p,r\}},\{\sim q,r\}\}}$, but all its weakenings do not.
- $\big\{\{p, r\}, \underline{\{{\sim}q, r\}}\big\} \models \Gamma^{\{\{p,r\},\underline{\{\sim q,r\}}\}}$, but all its weakenings do not.

How we deal with constraints?

epistemic logic program Π	K-WVs	SE-WVs	S-WVs
Π_1 : p or q	$\{\{p\},\{q\}\}$	$\{\{p\},\{q\}\}$	$\{\{p\},\{q\}\}$
p or q $\leftarrow$ not K p	none	$\{\{p\}\}$	$\{\{p\}\}$
Π_2 : $p \leftarrow$ not q $q \leftarrow$ not p $r \leftarrow$ K p	$\{\{p\},\{q\}\}$	$\{\{p\},\{q\}\}$	$\{\{p\},\{q\}\}$ $\{\{p,r\}\}$
$p \leftarrow$ not q $q \leftarrow$ not p $r \leftarrow$ K p $\leftarrow$ not r	$\{\{p,r\}\}$	$\{\{p,r\}\}$	$\{\{p,r\}\}$

Examples continued...

epis. spec. Π	K-WVs	SE-WVs	S-WVs
$\Pi_1 : p \leftarrow \mathtt{not}\, q$ $q \leftarrow \mathtt{not}\, p$ $r \mathtt{or} s \leftarrow \mathtt{not}\, \mathrm{K}\, p$	$\big\{\{p, r\}, \{q, r\}, \{p, s\}, \{q, s\}\big\}$	*same*	*same*
$\Pi_2 : p \leftarrow \mathtt{not}\, q$ $q \leftarrow \mathtt{not}\, p$ $r \vee s \leftarrow \mathtt{not}\, \mathrm{K}\, p$ $\leftarrow r$ $\leftarrow s$	$\big\{\{p\}\big\}$	$\big\{\{p\}\big\}$	$\big\{\{p\}\big\}$

- What would we expect? no world views/AEEMs
- Intuitive? no!

Discussion: inclusion of belief operator

let's call our belief operator B :

1. can consider B as dual of K (same as M in ES), i.e.,
   - B is equivalent to `not` K `not`
   - can treat it neither positive nor negative construct (similar to `notnot` in ASP)
   - ? shoud we take its reduct? probably YES!
   - complicated because then we have to define how to take the reduct of K `not`
2. can consider B (similar to $\hat{K}$ in ES21) as non-dual of K
   - reasonable because EASP is a 3-valued formalism
   - treat it as a positive subjective literal like K p
   - and we do not take its reduct
   - ? but then $p \leftarrow \mathrm{B}\,p$ has a unique ESM $\{\emptyset\}$. Intuitive?
   - remember that $p \leftarrow \mathrm{M}\,p$ has a unique SE-WV and K-WV: $\{\{p\}\}$.

Outline

To sum it up

- many different semantics approaches for ELPs
  - most of them are obsolete today:
    [Gelfond 1991,1994,2011; Kahl et al. 2014,2016, Wang&Zhang 2005,...]
  - successful candidates (to some extent):
    [Kahl 2018, SE 2016, FHS 2015, CFF 2019]
    - cannot cope with programs including constraints except
      [Cabalar et al., 2019]
- Our approach:
  - propose a more standard generalisation of ASP
  - but cannot offer a solution to the constraint problem
  - still, functionality of constraints can be discussed in ES
    (see [Shen and Eiter, 2019])

Thank you!