Extending answer set programming using generalized possibilistic logic

Steven Schockaert
(joint work with Didier Dubois and Henri Prade)

School of Computer Science & Informatics
Cardiff University, Cardiff, UK
schockaerts1@cardiff.ac.uk
http://users.cs.cf.ac.uk/S.Schockaert
1. Possibilistic logic
2. Generalized possibilistic logic
3. Answer set programming
4. Extending answer set programming
Possibilistic logic: syntax

Knowledge bases in possibilistic logic are sets of weighted formulas of the form:

\[(p \land (\neg q \rightarrow r), 0.7)\]

- **propositional formula**
- **certainty degree, taken from**

\[\Lambda = \{0, \frac{1}{k}, \frac{2}{k}, \ldots, 1\}\]
Possibilistic logic: syntax

Knowledge bases in possibilistic logic are sets of weighted formulas of the form:

\[(p \land (\neg q \rightarrow r), 0.7)\]

\[K = \{(\text{loc}(IJCAI2015, Argentina), 1),\]
\[(\text{loc}(IJCAI2015, Argentina) \rightarrow \text{loc}(IJCAI2015, SouthAmerica), 1),\]
\[(\text{loc}(IJCAI2016, NewYork) \rightarrow \text{loc}(IJCAI2016, NorthAmerica), 1),\]
\[(\text{loc}(IJCAI2015, SouthAmerica) \rightarrow \neg \text{loc}(IJCAI2016, SouthAmerica), 0.75),\]
\[(\text{loc}(IJCAI2015, SouthAmerica) \rightarrow \neg \text{loc}(IJCAI2016, NorthAmerica), 0.5)\}\]

reflects my degree of surprise if I found out that the formula were false
Possibilistic logic: syntax

Certainty degrees are interpreted qualitatively, i.e. possibilistic logic theories can be seen as stratified classical logic theories

\[
\begin{array}{l}
\text{loc}(\text{IJCAI2015, Argentina}) \\
\text{loc}(\text{IJCAI2015, Argentina}) \rightarrow \text{loc}(\text{IJCAI2015, SouthAmerica}) \\
\text{loc}(\text{IJCAI2016, NewYork}) \rightarrow \text{loc}(\text{IJCAI2016, NorthAmerica}) \\
\text{loc}(\text{IJCAI2015, SouthAmerica}) \rightarrow \neg \text{loc}(\text{IJCAI2016, SouthAmerica}) \\
\text{loc}(\text{IJCAI2015, SouthAmerica}) \rightarrow \neg \text{loc}(\text{IJCAI2016, NorthAmerica})
\end{array}
\]

Using numbers makes it easier to describe the semantics and to formulate inference rules
Possibilistic logic: semantics

At the semantic level, the role of models is taken up by **possibility distributions**, which assign to every propositional interpretation (or possible world) a degree of possibility, e.g.

\[
\begin{align*}
\pi(\omega_1) &= 0 \\
\pi(\omega_2) &= 0.4 \\
\pi(\omega_3) &= 0.7 \\
\pi(\omega_4) &= 1
\end{align*}
\]

Intuitively, a model corresponds to the **epistemic state** of an agent, encoded as the weighted set of worlds it considers possible.
Possibilistic logic: semantics

If $\pi$ represents the epistemic state of an agent, then that agent considers a formula $\alpha$ **possible** to the degree that some model of $\alpha$ is considered possible.

\[
\Pi(\alpha) = \Pi(\{\omega \mid \omega \models \alpha\}) = \max\{\pi(\omega) \mid \omega \models \alpha\}
\]

*possibility measure*
Possibilistic logic: semantics

If $\pi$ represents the epistemic state of an agent, then that agent considers a formula $\alpha$ **possible** to the degree that some model of $\alpha$ is considered possible.

$$\Pi(\alpha) = \Pi(\{\omega \mid \omega \models \alpha\}) = \max\{\pi(\omega) \mid \omega \models \alpha\}$$

The agent considers the formula **necessary** to the degree that all counter-models of $\alpha$ are impossible.

$$N(\alpha) = N(\{\omega \mid \omega \models \alpha\}) = \min\{1 - \pi(\omega) \mid \omega \not\models \alpha\}$$

$$= 1 - \Pi(\neg \alpha)$$

**necessity measure**
A possibility distribution $\pi$ satisfies a formula $(\alpha, \lambda)$ iff $N(\alpha) \geq \lambda$, with $N$ the necessity measure induced by $\pi$.

A possibility distribution $\pi$ is called a model of a set of formulas $K$ iff $\pi$ satisfies all formulas in $K$. 
Possibilistic logic: semantics

A possibility distribution $\pi$ satisfies a formula $(\alpha, \lambda)$ iff $N(\alpha) \geq \lambda$, with $N$ the necessity measure induced by $\pi$.

A possibility distribution $\pi$ is called a model of a set of formulas $K$ iff $\pi$ satisfies all formulas in $K$.

$$K = \{(a, 0.8), (a \rightarrow b, 1)\}$$

$$\pi(\{a, b\}) = 1$$
$$\pi(\{a\}) = 0$$
$$\pi(\{b\}) = 0.2$$
$$\pi(\{\}\} = 0.2$$
Possibilistic logic: semantics

A possibility distribution $\pi$ satisfies a formula $(a, \lambda)$ iff $N(a) \geq \lambda$, with $N$ the necessity measure induced by $\pi$.

A possibility distribution $\pi$ is called a model of a set of formulas $K$ iff $\pi$ satisfies all formulas in $K$.

$K = \{(a, 0.8), (a \rightarrow b, 1)\}$

$\pi(\{a, b\}) = 1$
$\pi(\{a\}) = 0$
$\pi(\{b\}) = 0.2$
$\pi(\{\}\} = 0.2$

$N(a) = 1 - \Pi(\neg a)$
$= 1 - \max(\pi(\{b\}), \pi(\{\})\})$
$= 0.8$

$N(a \rightarrow b) = 1 - \Pi(a \land \neg b)$
$= 1 - \max(\pi(\{a\}))$
$= 1$
Possibilistic logic: semantics

\[ \pi_1 \text{ is less specific than } \pi_2 \text{ if } \forall \omega . \pi_1(\omega) \geq \pi_2(\omega) \]

\[ \forall \omega . \pi(\omega) = 1 \]

Completely uninformative: every world remains possible

\[ \pi(\omega_0) = 1 \text{ and } \forall \omega \neq \omega_0 . \pi(\omega) = 0 \]

Maximally informative: exactly one world is considered possible
Possibilistic logic: semantics

\( \pi_1 \) is less specific than \( \pi_2 \) if

\[ \forall \omega . \pi_1(\omega) \geq \pi_2(\omega) \]

\( \forall \omega . \pi(\omega) = 1 \)

Completely uninformatve: every world remains possible

\( \pi(\omega_0) = 1 \) and \( \forall \omega \neq \omega_0 . \pi(\omega) = 0 \)

Maximally informative: exactly one world is considered possible

Every consistent set of formulas \( K \) in possibilistic logic has a unique least specific model \( \pi_K \)
Possibilistic logic: inference

**inference rules**

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>if ((\alpha, \lambda) \in K) then (K \vdash (\alpha, \lambda))</td>
<td></td>
</tr>
<tr>
<td>if (\alpha \equiv \beta) and (K \vdash (\alpha, \lambda)) then (K \vdash (\beta, \lambda))</td>
<td></td>
</tr>
<tr>
<td>if (\lambda_1 \geq \lambda_2) and (K \vdash (\alpha, \lambda_1)) then (K \vdash (\alpha, \lambda_2))</td>
<td></td>
</tr>
<tr>
<td>if (K \vdash (\alpha \lor \beta, \lambda_1)) and (K \vdash (\neg \alpha \lor \gamma, \lambda_2)) then (K \vdash (\beta \lor \gamma, \min(\lambda_1, \lambda_2)))</td>
<td></td>
</tr>
</tbody>
</table>

possibilistic logic is based on a weakest link idea
Possibilistic logic: inference

**Inference rules**

if \((\alpha, \lambda) \in K\) then \(K \vdash (\alpha, \lambda)\)

if \(\alpha \equiv \beta\) and \(K \vdash (\alpha, \lambda)\) then \(K \vdash (\beta, \lambda)\)

if \(\lambda_1 \geq \lambda_2\) and \(K \vdash (\alpha, \lambda_1)\) then \(K \vdash (\alpha, \lambda_2)\)

if \(K \vdash (\alpha \lor \beta, \lambda_1)\) and \(K \vdash (\neg \alpha \lor \gamma, \lambda_2)\) then \(K \vdash (\beta \lor \gamma, \min(\lambda_1, \lambda_2))\)

**Soundness and completeness**

The following statements are equivalent:

1. \(K \vdash (\alpha, \lambda)\) can be derived from (1)–(4).
2. Every model \(\pi\) of \(K\) is a model of \((\alpha, \lambda)\).
3. The least specific model \(\pi_K\) of \(K\) is a model of \((\alpha, \lambda)\).
Possibilistic logic: applications

Possibilistic logic is closely related to AGM belief revision and the rational closure of default rules

\[ K = \{(\text{loc}(\text{IJCAI2015}, \text{Argentina}), 1), \\
(\text{loc}(\text{IJCAI2015}, \text{Argentina}) \rightarrow \text{loc}(\text{IJCAI2015}, \text{SouthAmerica}), 1), \\
(\text{loc}(\text{IJCAI2016}, \text{NewYork}) \rightarrow \text{loc}(\text{IJCAI2016}, \text{NorthAmerica}), 1), \\
(\text{loc}(\text{IJCAI2015}, \text{SouthAmerica}) \rightarrow \neg \text{loc}(\text{IJCAI2016}, \text{SouthAmerica}), 0.75), \\
(\text{loc}(\text{IJCAI2015}, \text{SouthAmerica}) \rightarrow \neg \text{loc}(\text{IJCAI2016}, \text{NorthAmerica}), 0.5)\} \]

+ 

\( (\text{loc}(\text{IJCAI2016}, \text{NewYork}), 1) \)
Possibilistic logic: applications

Possibilistic logic is closely related to AGM belief revision and the rational closure of default rules

\[
K = \{(\text{loc}(\text{IJCAI}2015, \text{Argentina}), 1), \\
(\text{loc}(\text{IJCAI}2016, \text{New York}), 1), \\
(\text{loc}(\text{IJCAI}2015, \text{Argentina}) \rightarrow \text{loc}(\text{IJCAI}2015, \text{South America}), 1), \\
(\text{loc}(\text{IJCAI}2016, \text{New York}) \rightarrow \text{loc}(\text{IJCAI}2016, \text{North America}), 1), \\
(\text{loc}(\text{IJCAI}2015, \text{South America}) \rightarrow \neg \text{loc}(\text{IJCAI}2016, \text{South America}), 0.75), \\
(\text{loc}(\text{IJCAI}2015, \text{South America}) \rightarrow \neg \text{loc}(\text{IJCAI}2016, \text{North America}), 0.5)\} 
\]
Possibilistic logic: applications

Possibilistic logic is closely related to AGM belief revision and the rational closure of default rules.

\[ K = \{(\text{loc(IJCAI2015, Argentina), 1}), \]
\[ (\text{loc(IJCAI2016, New York), 1}), \]
\[ (\text{loc(IJCAI2015, Argentina) \rightarrow loc(IJCAI2015, SouthAmerica), 1}), \]
\[ (\text{loc(IJCAI2016, New York) \rightarrow loc(IJCAI2016, NorthAmerica), 1}), \]
\[ (\text{loc(IJCAI2015, SouthAmerica) \rightarrow ¬loc(IJCAI2016, SouthAmerica), 0.75}), \]
\[ (\text{loc(IJCAI2015, SouthAmerica) \rightarrow ¬loc(IJCAI2016, NorthAmerica), 0.5})\]
1. Possibilistic logic

2. Generalized possibilistic logic

3. Answer set programming

4. Extending answer set programming
Generalized possibilistic logic

In possibilistic logic, a formula $(\alpha, \lambda)$ corresponds to the constraint $N(\alpha) \geq \lambda$ at the semantic level, and a knowledge base corresponds to a conjunction of such constraints

$$\Pi(\alpha) \geq \lambda$$
Generalized possibilistic logic

In **possibilistic logic**, a formula \((\alpha, \lambda)\) corresponds to the constraint \(N(\alpha) \geq \lambda\) at the semantic level, and a knowledge base corresponds to a conjunction of such constraints.

In **generalized possibilistic logic** (GPL), arbitrary propositional combinations of such formulas are allowed.

To emphasize the view of possibilistic logic as a **modal logic**, we use the following notations:

\[
\begin{align*}
N_{\lambda}(\alpha) & \quad \text{alternative notation for } (\alpha, \lambda) \\
\Pi_{\lambda}(\alpha) & \quad \text{abbreviation for } -N_{1-\lambda+\frac{1}{k}}(-\alpha) \\
& \quad \text{corresponds to the constraint } \Pi(\alpha) \geq \lambda
\end{align*}
\]
Generalized possibilistic logic

Syntax

\[ (\mathcal{N}_{0.4}(a \land \neg b) \lor \Pi_{0.3}(a \rightarrow (b \lor c))) \rightarrow \mathcal{N}_{0.7}(b) \]

Note: every propositional atom is encapsulated by a modality
Note: no nesting of modalities
Generalized possibilistic logic

**Syntax**

\[
(N_{0.4}(a \land \neg b) \lor \Pi_{0.3}(a \rightarrow (b \lor c))) \rightarrow N_{0.7}(b)
\]

Note: every propositional atom is encapsulated by a modality
Note: no nesting of modalities

**Semantics**

\[
(N(a \land \neg b) < 0.4 \land \Pi(a \rightarrow (b \lor c)) < 0.3) \lor N(b) \geq 0.7
\]

Note: models are possibility distributions, which are interpreted as epistemic states

**Axiomatization**

Weighted version of the axiomatization of the modal logic KD without introspection (KR 2012)
possibilistic logic

- **formulas:** express lower bounds on the necessity of a propositional formula

- **models:** weighted epistemic states

- **minimally specific models:** unique

  useful for reasoning about the consequences of one’s own beliefs

GPL

- **formulas:** express propositional combinations of lower bounds on the necessity of a propositional formula

- **models:** weighted epistemic states

- **minimally specific models:** 0, 1 or more

  useful for reasoning about the revealed beliefs of another agent
Inference relations

GPL naturally leads to one monotonic and two non-monotonic inference relations (all of which are equivalent in normal possibilistic logic):

**Standard reasoning**

\[ K \models_b \Phi \quad \text{iff } \Phi \text{ is true in all models of } K \]

**Brave reasoning**

\[ K \models_b \Phi \quad \text{iff } \Phi \text{ is true in at least one minimally specific model of } K \]

**Cautious reasoning**

\[ K \models_c \Phi \quad \text{iff } \Phi \text{ is true in all minimally specific models of } K \]
## Computational complexity

| Language   | $|=|$ | $|=b|$ | $|=c|$ |
|------------|-----|------|------|
| GPL        | coNP| $\sum_2^P$ | $\Pi_2^P$ |
| $\text{GPL}^\Delta$ | $\Theta_2^P$ | $\sum_2^P$ | $\Pi_2^P$ |
| $\text{GPL}^\Delta_R$ | $\Pi_3^P$ | $\sum_4^P$ | $\Pi_4^P$ |

Extension based on the language of GPL with a modality that corresponds to the guaranteed possibility measure.

Same complexity as inference in (disjunctive) answer set programming and many non-monotonic reasoning frameworks.

Extension based on a refinement of the guaranteed possibility measure.

---

Proposition 8. The problem of deciding whether a GPL formula is satisfiable is NP-complete, even when the only certainty levels are 0 and 1.

Proof. Hardness follows straightforwardly from the NP-completeness of satisfiability in propositional logic. In particular, note that the propositional formula $\varphi$ is satisfiable if the GPL formula $\Rightarrow (\varphi)$ is satisfiable.

We now propose an NP procedure for checking the satisfiability of an arbitrary GPL formula $\varphi$. Each GPL formula $\varphi$ is equivalent to a disjunction of meta-terms, and it is sufficient that one of these terms is satisfiable. In polynomial time, we can guess such a term:

$$N_1(\varphi_1)^\ldots^N_n(\varphi_n)^\Rightarrow \mu_1(\varphi_1)^\ldots^\mu_m(\varphi_m)$$

We know that $N_1(\varphi_1)^\ldots^N_n(\varphi_n)$ has a unique model $\models$ if $\varphi_1^\ldots^\varphi_n$ is satisfiable. All that we need to check is whether this is the case, and whether $\Rightarrow (\varphi)$ for each $i$, with $\Rightarrow$ the possibility measure induced by $\models$.

In other words, for each $j$ and each $\Theta_2^P + k$ such that $\mu_j(\varphi)$, the following formula needs to be consistent:

$$\{\varphi_i | i \models \Theta_2^P + k\}$$

To check satisfiability in NP, when we guess the term (10), we can also guess an interpretation for each of these SAT instances (including the SAT instance $\varphi_1^\ldots^\varphi_n$) and verify that the corresponding propositional formulas.

Corollary 1. The problem of deciding whether $\Rightarrow (\varphi)$, with GPL and/GPL formulas, is coNP-complete.

Proof. This follows immediately from the observation that $\Rightarrow (\varphi)$ holds if $\neg \varphi$ is not satisfiable.
1. Possibilistic logic
2. Generalized possibilistic logic
3. Answer set programming
4. Extending answer set programming
A (disjunctive) answer set program (ASP) consists of rules of the form:

\[ a_1 \lor \neg a_2 \lor a_3 \leftarrow b_1 \land b_2 \land \neg c_1 \land \neg c_2 \]

Intuition: if we already know that \( b_1 \) and \( b_2 \) are true, then we should accept that \( a_1 \) is true, that \( a_2 \) is false, or that \( a_3 \) is true, unless we know that \( c_1 \) or \( c_2 \) is true.
Answer set programming

A (disjunctive) **answer set program** (ASP) consists of rules of the form:

\[ a_1 \lor \neg a_2 \lor a_3 \leftarrow b_1 \land b_2 \land \neg c_1 \land \neg c_2 \]

Intuition: if we already know that \( b_1 \) and \( b_2 \) are true, then we should accept that \( a_1 \) is true, that \( a_2 \) is false, or that \( a_3 \) is true, unless we know that \( c_1 \) or \( c_2 \) is true.

Usual way to define the semantics is in terms of the **Gelfond-Lifschitz reduct**: guess what can be derived from the program, use this guess to evaluate literals preceded by not, and verify that the guess was correct.

The set of literals that can be derived from the program is then called an **answer set** or **stable model**.
Answer set programming

1. The program

\[ w \leftarrow \text{not } c \]
\[ c \leftarrow \text{not } w \]
Answer set programming

1. the program
   
   \[ w \leftarrow \text{not } c \]
   
   \[ c \leftarrow \text{not } w \]

2. guess
   
   is it \( \{w\} \)?
Answer set programming

1. The program
   \[ w \leftarrow \text{not } c \]
   \[ c \leftarrow \text{not } w \]

2. Guess
   is it \( \{w\} \)?

3. Reduct
   \[ w \leftarrow \text{not } c \]
   \[ c \leftarrow \text{not } w \]
Answer set programming

1. the program
   
   \[ w \leftarrow \text{not } c \\
   c \leftarrow \text{not } \text{not } w \]

2. guess
   
   is it \{w\}?  

3. reduct
   
   \[ w \leftarrow \text{not } c \\
   c \leftarrow \text{not } \text{not } w \]

Minimal model of the reduct is indeed \{w\}, which means that \{w\} is an answer set (or stable model)
Relationship with GPL

We associate a GPL theory $\Theta_P$ with an answer set program $P$ as follows

**ASP rule**

$$l_1 \lor \ldots \lor l_n \leftarrow p_1 \land \ldots \land p_m \land \neg c_1 \land \ldots \land \neg c_r$$

**GPL formula**

$$N_1(p_1) \land \ldots \land N_1(p_m) \land \Pi_1(\neg c_1) \land \ldots \land \Pi_1(\neg c_r) \rightarrow N_1(l_1) \lor \ldots \lor N_1(l_n)$$
A set of literals $M$ is an **answer set** of $P$ iff the possibility distribution $\pi$ defined as

$$ \pi(\omega) = \begin{cases} 1 & \text{if } \omega \models l \text{ for all } l \in M \\ 0 & \text{otherwise} \end{cases} $$

is a **minimally specific model** of $\Theta_P$

Note that $\pi$ is the least specific possibility distribution that satisfies $N_1(l)$ for each $l$ in $M$
Relationship with GPL: intuition

There are three ways of making the following GPL formula satisfied:

\[ N_1(p_1) \land \ldots \land N_1(p_m) \land \Pi_1(\neg c_1) \land \ldots \land \Pi_1(\neg c_r) \rightarrow N_1(l_1) \lor \ldots \lor N_1(l_n) \]

Make \( N_1(p_i) \) false

Make \( \Pi_1(\neg c_i) \) false, i.e. make \( N_{1/k}(c_i) \) true

Make \( N_1(l_i) \) true
Relationship with GPL: intuition

There are three ways of making the following GPL formula satisfied:

\[ N_1(p_1) \land \ldots \land N_1(p_m) \land \Pi_1(\neg c_1) \land \ldots \land \Pi_1(\neg c_r) \rightarrow N_1(l_1) \lor \ldots \lor N_1(l_n) \]

- Make \( N_1(p_i) \) false

- Make \( \Pi_1(\neg c_i) \) false, i.e. make \( N_{1/k}(c_i) \) true

- Make \( N_1(l_i) \) true

This will always be preferred when we are looking for minimally specific models.
Relationship with GPL: intuition

There are three ways of making the following GPL formula satisfied:

\[ N_1(p_1) \land \ldots \land N_1(p_m) \land \Pi_1(\neg c_1) \land \ldots \land \Pi_1(\neg c_r) \rightarrow N_1(l_1) \lor \ldots \lor N_1(l_n) \]

1. Make \( N_1(p_i) \) false
2. Make \( \Pi_1(\neg c_i) \) false, i.e. make \( N_{1/k}(c_i) \) true
3. Make \( N_1(l_i) \) true

This corresponds to the **guess** that \( c_i \) will be derivable

Boolean models are those for which the guess can be **verified**, i.e. those for which \( N_1(c_i) \) can be derived
Relationship with GPL: intuition

There are three ways of making the following GPL formula satisfied:

\[ N_1(p_1) \land \ldots \land N_1(p_m) \land \Pi_1(\neg c_1) \land \ldots \land \Pi_1(\neg c_r) \rightarrow N_1(l_1) \lor \ldots \lor N_1(l_n) \]

Make \( N_1(p_i) \) false

Make \( \Pi_1(\neg c_i) \) false, i.e. make \( N_{1/k}(c_i) \) true

Make \( N_1(l_i) \) true

This intuitively corresponds to forward chaining

In minimal models, this option will only be chosen if all of the \( p_i \) literals can be derived and none of the \( c_i \) literals.
Relationship with GPL

Let us define:

\[ \Phi \equiv \bigwedge_{a \in At} N_1(a) \lor N_1(\neg a) \lor (\Pi_1(a) \land \Pi_1(\neg a)) \]

P has an answer set iff:

\[ \Theta_P \models^b \Phi \]

The literal l is contained in at least one answer set of P iff:

\[ \Theta_P \models^b \Phi \land N_1(l) \]

The literal l is contained in all answer sets of P iff:

\[ \Theta_P \models^c \Phi \rightarrow N_1(l) \]
Restricting the set of certainty degrees

The characterization of ASP requires that at least 3 certainty degrees are considered.

When only two certainty degrees, 0 and 1, are used, we have

\[ N_1(\alpha) = \neg \Pi_1(\neg \alpha) \]

This would mean that the following two ASP rules are incorrectly translated to the same GPL formula:

\[ a \leftarrow \text{not } b \]
\[ b \leftarrow \text{not } a \]
1. Possibilistic logic
2. Generalized possibilistic logic
3. Answer set programming
4. Extending answer set programming
Disjunction inside $\mathbf{N}$

\[
\text{strong disjunction}
\]

\[
\text{loc(IJCAI2016, NewYork)} \lor \text{loc(IJCAI2016, SanFrancisco)} \leftarrow
\]

\[
\text{weak disjunction}
\]

\[
\text{N}_1(\text{loc(IJCAI2016, NewYork)} \lor \text{loc(IJCAI2016, SanFrancisco)})
\]
Stable models of propositional theories

\[(\forall a \rightarrow b) \land (\forall c \rightarrow \neg d) \land (x \lor y)\]

More flexible than equilibrium logic (e.g. weak disjunction)

More intuitive semantics: all models respect the idea of minimal commitment
Possibilistic ASP

Certainty weights indicate our belief that the **rule is correct**
Semantics: possibility distribution over classical answer sets

Certainty weights indicate our belief in the **conclusion of the rule**, if the body is satisfied
Semantics: set of possibilistic answer sets

1: \( \text{loc}(\text{IJCAI2015, Argentina}) \leftarrow \)
0.6: \( \text{loc}(\text{IJCAI2016, NewYork}) \leftarrow \text{not unreliable(website)} \)
1: \( \text{loc}(\text{IJCAI2015, SouthAmerica}) \leftarrow \text{loc}(\text{IJCAI2015, Argentina}) \)
1: \( \text{loc}(\text{IJCAI2016, NorthAmerica}) \leftarrow \text{loc}(\text{IJCAI2016, NewYork}) \)
0.75: \( \neg \text{loc}(\text{IJCAI2015, SouthAmerica}) \leftarrow \text{loc}(\text{IJCAI2015, SouthAmerica}) \)
0.5: \( \neg \text{loc}(\text{IJCAI2015, NorthAmerica}) \leftarrow \text{loc}(\text{IJCAI2015, SouthAmerica}) \)
Certainty weights indicate our belief that the **rule is correct**

Semantics: possibility distribution over classical answer sets

Certainty weights indicate our belief in the **conclusion of the rule**, if the body is satisfied

Semantics: set of possibilistic answer sets

Possibilistic ASP

1: \(\text{loc}(\text{IJCAI2015}, \text{Argentina}) \leftarrow\)

0.6: \(\text{loc}(\text{IJCAI2016}, \text{NewYork}) \leftarrow \text{not unreliable(website)}\)

1: \(\text{loc}(\text{IJCAI2015}, \text{SouthAmerica}) \leftarrow \text{loc}(\text{IJCAI2015}, \text{Argentina})\)

1: \(\text{loc}(\text{IJCAI2016}, \text{NorthAmerica}) \leftarrow \text{loc}(\text{IJCAI2016}, \text{NewYork})\)

0.75: \(\neg \text{loc}(\text{IJCAI2015}, \text{SouthAmerica}) \leftarrow \text{loc}(\text{IJCAI2015}, \text{SouthAmerica})\)

0.5: \(\neg \text{loc}(\text{IJCAI2015}, \text{NorthAmerica}) \leftarrow \text{loc}(\text{IJCAI2015}, \text{SouthAmerica})\)
Possibilistic ASP

\[
\frac{l}{k} : a_1 \lor \ldots \lor a_n \leftarrow b_1 \land \ldots \land b_m \land \text{not } c_1 \land \ldots \land \text{not } c_r
\]

**Intuition:** \(\text{certainty(head)} \geq \min(l/k, \text{certainty(body)})\)
Possibilistic ASP

\[
\frac{l}{k} : a_1 \lor \ldots \lor a_n \leftarrow b_1 \land \ldots \land b_m \land \text{not } c_1 \land \ldots \land \text{not } c_r
\]

assume that \(l\) is even

\[
\bigwedge_{i=1}^{l} \left( \text{N}_{\frac{i}{k}}(b_1) \land \ldots \land \text{N}_{\frac{i}{k}}(b_m) \land \prod_{\frac{i}{k}}(\neg c_1) \land \ldots \land \prod_{\frac{i}{k}}(\neg c_r) \rightarrow \text{N}_{\frac{i}{k}}(a_1) \lor \ldots \lor \text{N}_{\frac{i}{k}}(a_n) \right)
\]

**Intuition:** for each \(i \leq l\), if the body can be derived with certainty \(i/k\) then we should also believe the head with certainty \(i/k\)
Conclusions

Possibilistic logic offers a convenient way to encode the epistemic state of an agent

- natural characterization of (AGM) belief revision and default reasoning

Generalized possibilistic logic (GPL) offers a convenient way to encode our knowledge about the possible epistemic states of another agent

- natural characterization of NMR frameworks based on multiple extensions, such as answer set programming (ASP)

The GPL characterization of ASP suggests several natural generalizations

- Weak disjunction
- Stable models of arbitrary propositional theories
- Modelling uncertainty in answer set programming
References and acknowledgments

**Generalized possibilistic logic**


Didier Dubois, Henri Prade, Steven Schockaert. Reasoning about uncertainty and explicit ignorance in generalized possibilistic logic, Proc. ECAI, 2014

**Possibilistic answer set programming**


Kim Bauters, Steven Schockaert, Martine De Cock, Dirk Vermeir. Possibilistic answer set programming revisited, Proc. UAI, 2010