Preferences, Contexts and Answer Sets

Gerhard Brewka

Computer Science Institute
University of Leipzig
brewka@informatik.uni-leipzig.de

with contributions from T. Eiter, I. Niemelä, M. Truszczyński

Outline

- Part I: Preferences
  - Why combining ASP with preferences?
  - Two (related) approaches
  - Applications
- Part II: Contexts
  - Why nonmonotonic extensions of multi-context systems?
  - Equilibria in nonmonotonic MCS
  - Groundedness
- Part III: Putting things together (outlook)

Background

- Concepts underlying answer set programming taken for granted:
  - Logic programs
  - Answer sets
  - ASP as a constraint-based problem solving paradigm

- Sometimes use Smodels cardinality constraints

$$L\{a_1, \ldots, a_k\}U$$

Read: at least L and at most U of the a_is must be true

Background

- Concepts underlying answer set programming taken for granted:
  - Logic programs
  - Answer sets
  - ASP as a constraint-based problem solving paradigm

- Sometimes use Smodels cardinality constraints

$$L\{a_1, \ldots, a_k\}U$$

Read: at least L and at most U of the a_is must be true

Part I: Preferences

Preferences

- Determine how agents decide and act
- Pop up everywhere:

coffee	>	tea
car	>	train
relax	>	work
FC Porto	>	Bayern München
marry	>	don't marry
sleep	>	listen to talk

- Also in many AI applications: diagnosis, planning, configuration, revision, ontologies etc.

Issues

- How to represent space of alternatives:
  often used: constraints; here: answer sets
- How to represent preferences:
  traditionally: numbers; here: qualitative
  numbers difficult to obtain; not always necessary
- How to interpret preferences:
  strict vs. defeasible; ceteris paribus
- How to represent (in)dependencies:
  preferences almost always context dependent

Adding Preferences to ASP

Options

	rule preference	formula preference
fixed	$(P, <)$ $<$ order on P B-Eiter Delgrande-Schaub ...	$(P, <)$ $<$ order on Lit Sakama-Inoue Foo-Zhang ...
conditional	$<$ predicate in P applied to rules B-Eiter Delgrande-Schaub ...	ordered disjunction ASO programs B-Niemelä-Syrjänen B-N-Truszczyński ...

Ordered Disjunction

LPOD: finite set of rules of the form:

$$C_1 \times \ldots \times C_n \leftarrow body$$

if body then some C_j must be true, preferably C_1, if impossible then C_2, if impossible C_3, etc.

- Answer sets defined through split programs:
  - Pick one option for each ordered disjunction
  - Each AS of a split program is AS of original LPOD
- Satisfy LPOD rules to different degrees, depending on best satisfied head literal
- Use degrees to define global preference relation on answer sets

Preferences Among Answer Sets

How to generate global preference ordering from satisfaction degrees?

Many options, for instance:

$P^i(S)$ = P-rules i-satisfied in S. $S_1 > S_2$ iff

- some rule has better satisfaction degree in S_1 and no rule better degree in S_2,
- at smallest degree i with $P^i(S_1) \neq P^i(S_2)$, S_1 satisfies superset of rules satisfied in S_2,
- at smallest degree i with $|P^i(S_1)| \neq |P^i(S_2)|$, S_1 satisfies more rules than S_2.

Prioritized Graph Coloring

$$col(X,r) \times col(X,b) \times col(X,g) \leftarrow node(X)$$
$$\leftarrow col(X,C), col(Y,C), edge(X,Y)$$

M preferred over M' if

- *par* at least 1 node has nicer color in M than in M', no node less preferred color.
- *incl* nodes red in M superset of nodes red in M', or same nodes red in M and M' and nodes blue in M superset of nodes blue in M'.
- *card* more nodes red in M than in M', or as many nodes red in M as in M' and more blue in M.

The ASO Approach

- Decoupled approach to answer set optimization
- Logic program G generates answer sets
- Preference program P used to compare them
- Preference program set of rules

$$C_1 > \ldots > C_k \leftarrow body$$

C_i boolean combination built using $\vee$, $\wedge$, $\neg$, **not**

- Rule satisfaction and combination as for LPODs

$$1\{col(X,Y) : color(Y)\}1 \leftarrow node(X)$$
$$\leftarrow col(X,C), col(Y,C), edge(X,Y)$$

$$col(X,r) > col(X,b) > col(X,g) \leftarrow node(X)$$

LPODs vs. ASO

- ASO: arbitrary generating programs, no implicit generation of options, general preferences:

  Combinations of properties preferred over others:

$$a > (b \wedge c) > d \leftarrow f$$

  Equally preferred options:

$$a > (b \vee c) > \textbf{not } d \leftarrow g$$

- LPODs: compact and readable representations

Applications: Configuration

- Configuration problems often represented as AND/OR trees
- Simple representation with Smodels cardinalities:

$$
\begin{array}{rcl}
4\{starter, main, dessert, drink\}4 & \leftarrow & dinner \\
1\{soup, salad\}1 & \leftarrow & starter \\
1\{fish, beef, lasagne\}1 & \leftarrow & main \\
1\{beer, wine\}1 & \leftarrow & drink \\
 & \ldots &
\end{array}
$$

- Add case description and preferences, e.g.

$$
\begin{array}{rcl}
fish \vee beef > lasagne & & \\
beer > wine & \leftarrow & beef \\
wine > beer & \leftarrow & \textbf{not}\ beef
\end{array}
$$

- Preferred answer sets: optimal configurations

Applications: Abductive Diagnosis

- Background knowledge:

$fever \leftarrow measles$	$nausea \leftarrow migraine$	$headache \leftarrow flu$
$red\text{-}spots \leftarrow measles$	$headache \leftarrow migraine$	$fever \leftarrow flu$

- Possible hypotheses: *measles*, *flu*, *migraine*
  Observations: *headache*, *fever*
- Diseases normally don't hold:

$$\neg measles \times measles; \quad \neg flu \times flu; \quad \neg migraine \times migraine$$

- Observations must hold:

$$\leftarrow \textbf{not}\ headache; \quad \leftarrow \textbf{not}\ fever$$

- Diagnoses = (parts of) preferred answer sets: $\{migraine, measles\}$, $\{flu\}$

Applications: Abductive Diagnosis

- Background knowledge:

$$\begin{array}{lll} \textit{fever} \leftarrow \textit{measles} & \textit{nausea} \leftarrow \textit{migraine} & \textit{headache} \leftarrow \textit{flu} \\ \textit{red-spots} \leftarrow \textit{measles} & \textit{headache} \leftarrow \textit{migraine} & \textit{fever} \leftarrow \textit{flu} \end{array}$$

- Possible hypotheses: *measles*, *flu*, *migraine*
  Observations: *headache*, *fever*
- Diseases normally don't hold:

$$\neg \textit{measles} \times \textit{measles}; \quad \neg \textit{flu} \times \textit{flu}; \quad \neg \textit{migraine} \times \textit{migraine}$$

- Observations must hold:

$$\leftarrow \textbf{not } \textit{headache}; \quad \leftarrow \textbf{not } \textit{fever}$$

- Diagnoses = (parts of) preferred answer sets: $\{\textit{migraine}, \textit{measles}\}$, $\{\textit{flu}\}$

Applications: Abductive Diagnosis

- Background knowledge:

$fever \leftarrow measles$	$nausea \leftarrow migraine$	$headache \leftarrow flu$
$red\text{-}spots \leftarrow measles$	$headache \leftarrow migraine$	$fever \leftarrow flu$

- Possible hypotheses: *measles*, *flu*, *migraine*
  Observations: *headache*, *fever*
- Diseases normally don't hold:

$$\neg measles \times measles; \quad \neg flu \times flu; \quad \neg migraine \times migraine$$

- Observations must hold:

$$\leftarrow \textbf{not}\ headache; \quad \leftarrow \textbf{not}\ fever$$

- Diagnoses = (parts of) preferred answer sets: $\{migraine, measles\}$, $\{flu\}$

Applications: Abductive Diagnosis

- Background knowledge:

$$\begin{array}{lll} fever \leftarrow measles & nausea \leftarrow migraine & headache \leftarrow flu \\ red\text{-}spots \leftarrow measles & headache \leftarrow migraine & fever \leftarrow flu \end{array}$$

- Possible hypotheses: *measles*, *flu*, *migraine*
  Observations: *headache*, *fever*
- Diseases normally don't hold:

$$\neg measles \times measles; \quad \neg flu \times flu; \quad \neg migraine \times migraine$$

- Observations must hold:

$$\leftarrow \textbf{not } headache; \quad \leftarrow \textbf{not } fever$$

- Diagnoses = (parts of) preferred answer sets: $\{migraine, measles\}$, $\{flu\}$

Applications: Game Theory

Prisoners' dilemma

	Coop.	Defect
Coop.	3,3	0,5
Defect	5,0	1,1

Player 1:

$$D_1 \times C_1 \leftarrow C_2$$
$$D_1 \times C_1 \leftarrow D_2$$

Player 2:

$$D_2 \times C_2 \leftarrow C_1$$
$$D_2 \times C_2 \leftarrow D_1$$

Move clause: $1\{C_1, D_1\}1$

Preferred answer set = Nash equilibrium

Applications: Game Theory

Prisoners' dilemma

	Coop.	Defect
Coop.	3,3	0,5
Defect	5,0	1,1

Player 1:

$$D_1 \times C_1 \leftarrow C_2$$
$$D_1 \times C_1 \leftarrow D_2$$

Player 2:

$$D_2 \times C_2 \leftarrow C_1$$
$$D_2 \times C_2 \leftarrow D_1$$

Move clause: $1\{C_1, D_1\}1$

Preferred answer set = Nash equilibrium

Applications: Game Theory

Prisoners' dilemma

	Coop.	Defect
Coop.	3,3	0,5
Defect	5,0	1,1

Player 1:

$$D_1 \times C_1 \leftarrow C_2$$
$$D_1 \times C_1 \leftarrow D_2$$

Player 2:

$$D_2 \times C_2 \leftarrow C_1$$
$$D_2 \times C_2 \leftarrow D_1$$

Move clause: $1\{C_1, D_1\}1$

Preferred answer set = Nash equilibrium

Further Contributions

- Meta-preferences: one preference rule/ordered disjunction more important than another
- Preference description language: combines different preference strategies; integrates qualitative with quantitative methods
- Implementation: *generate and improve* method; iterative calls to answer set solver generate sequence of strictly improving answer sets
- Integration with CP-nets: general preference framework combining graph based methods with flexibility of ASO preferences

Part II: Contexts

Basic Motivation

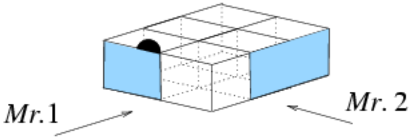
- Larger and larger bodies of knowledge being formalized
- Size of, say, medical ontologies requires methods for structuring and modularizing KBs
- Wealth of existing logical tools to model different forms of reasoning
- No single all-purpose formalism: necessary to integrate several formalisms into a single system
- Often done somewhat ad hoc for particular pair of formalisms

- **Can we do this in a more principled way?**
  Which role can multi-context systems play?
  And LP techniques?

Contexts

- In AI first investigated by John McCarthy (1987), without definition
- Intuitively, a context describes information based on a particular viewpoint, perspective, granularity, person/agent/database ...
- Here: (almost/somewhat) independent unit of reasoning
- Features of multi-context systems:
  - **Locality:** different languages, reasoning methods, logics
  - **Compatibility:** information flow between contexts
- Provide a particular form of information integration

Example: Magic Box



Existing Work I: The Trento School

Monotonic multi-context systems
(Giunchiglia & Serafini, AIJ 94)

- Heterogeneous: integrate different inference systems

$$MCS = (\{T_i\}, \Delta_{br})$$

- each $T_i = (L_i, \Omega_i, \Delta_i)$ is a formal system (language, axioms, inf. rules)
- Δ_{br} consists of *bridge rules* using labeled formulas $(c{:}p)$ where p is from the language L_c:

$$(c_1{:}p_1), \ldots, (c_k{:}p_k) \Rightarrow (c_j{:}q_j)$$

- Semantics: local models + compatibility
- Information flow across contexts via bridge rules
- Reasoning within/across contexts is monotonic

Existing Work II: Nonmonotonic MCS

Contextual Default Logic (CDL)

(Brewka, Roelofsen & Serafini, IJCAI 07)

follow-up of (Roelofsen & Serafini, IJCAI 05)

- CDL integrates nonmonotonic inference systems
- **But:** they all *must be of the same kind*:

  Theories in Reiter's Default Logic

- Defaults may refer to other contexts
- Defaults play the role of bridge rules

Our Goals

- Generalize existing approaches
- Define a *heterogeneous* multi-context framework accommodating both *monotonic and nonmonotonic* contexts
- Should be capable of integrating logics like description logics, modal logics, default logics, logic programs, etc.

“Logics”

Want to capture the “typical” KR logics, including nonmonotonic logics with multiple acceptable belief sets (e.g., Reiter’s Default Logic).

Logic

A logic L is a tuple

$$L = (\mathbf{KB}_L, \mathbf{BS}_L, \mathbf{ACC}_L)$$

- $\mathbf{KB}_L$ is a set of well-formed knowledge bases (each a set)
- $\mathbf{BS}_L$ is a set of possible belief sets (each a set)
- $\mathbf{ACC}_L : \mathbf{KB}_L \rightarrow 2^{\mathbf{BS}_L}$ assigns to each knowledge base a set of acceptable belief sets

L monotonic: $\mathbf{ACC}_L$ singleton set, growing monotonically with *kb*

Example Logics Over Signature Σ

Propositional logic

- **KB**: the sets of prop. Σ-formulas
- **BS**: the deductively closed sets of prop. Σ-formulas
- **ACC**(kb): $Th(kb)$

Default logic

- **KB**: the default theories over Σ
- **BS**: the deductively closed sets of Σ-formulas
- **ACC**(kb): the extensions of *kb*

Normal LPs under answer set semantics

- **KB**: the logic programs over Σ
- **BS**: the sets of atoms of Σ
- **ACC**(kb): the answer sets of *kb*

Multi-Context Systems

- As in monotonic MCS, information integration via bridge rules
- As in CDL, bridge rules and logics can be nonmonotonic
- Unlike in CDL, arbitrary logics can be used

Bridge Rules

Let $L = L_1, \ldots, L_n$ be a collection of logics.

An L_k-bridge rule over L, $1 \leq k \leq n$, is of the form

$$\begin{aligned} s \leftarrow \quad & (r_1 : p_1), \ldots, (r_j : p_j), \\ & \textbf{not}\ (r_{j+1} : p_{j+1}), \ldots, \textbf{not}\ (r_m : p_m) \end{aligned}$$

where s is a possible element of an L_k *kb*, each p_k a possible element of an L_{r_k} belief set.

Multi-Context Systems, II

Multi-Context System

A Multi-Context System

$$M = (C_1, \ldots, C_n)$$

consists of contexts

$$C_i = (L_i, kb_i, br_i),\ i \in \{1, \ldots, n\},$$

where

- each L_i is a logic,
- each $kb_i \in \mathbf{KB}_i$ is a L_i-knowledge base, and
- each br_i is a set of L_i-bridge rules over M's logics.

M can be nonmonotonic because *one of its context logics* is AND/OR because a context has *nonmonotonic bridge rules*.

Example

Consider the multi-context system $M = (C_1, C_2)$, where the contexts are different views of a paper by the authors.

- C_1:
  - L_1 = Classical Logic
  - $kb_1 = \{ \mathit{unhappy} \supset \mathit{revision} \}$
  - $br_1 = \{ \mathit{unhappy} \leftarrow (2 : \mathit{work}) \}$
- C_2:
  - L_2 = Reiter's Default Logic
  - $kb_2 = \{ \mathit{good} : \mathit{accepted} / \mathit{accepted} \}$
  - $br_2 = \{ \mathit{work} \leftarrow (1 : \mathit{revision}),$
    $\mathit{good} \leftarrow \textbf{not}\ (1 : \mathit{unhappy}) \}$

Acceptable Belief States

- **Belief state:** sequence of belief sets, one for each context
- **Fundamental Question:** *Which belief states are acceptable?*
- Those based on the knowledge base of a context AND the information accepted/not accepted in other contexts (if there are appropriate bridge rules)
- **Intuition:** belief states must be in *equilibrium*:

> The selected belief set for each context C_i must be among the acceptable belief sets for C_i's knowledge base *together with the heads of C_i's applicable bridge rules.*

Acceptable Belief States, II

Applicable Bridge Rules

Let $M = (C_1, \dots, C_n)$.

$$s \leftarrow (r_1 : p_1), \dots, (r_j : p_j), \\ \textbf{not } (r_{j+1} : q_1), \dots, \textbf{not } (r_{j+m} : q_m)$$

is applicable in belief state $S = (S_1, \dots, S_n)$ iff each p is in the belief set chosen for its context, each q is not.

Equilibrium

A belief state $S = (S_1, \dots, S_n)$ of M is an equilibrium iff for $i \in \{1, \dots, n\}$

$$S_i \in \mathbf{ACC}_i(kb_i \cup \{head(r) \mid r \in br_i \text{ is applicable in } S\}).$$

Example (ctd)

Reconsider multi-context system $M = (C_1, C_2)$:

- $kb_1 = \{ unhappy \supset revision \}$ (Classical Logic)
- $br_1 = \{ unhappy \leftarrow (2 : work) \}$
- $kb_2 = \{ good : accepted / accepted \}$ (Default Logic)
- $br_2 = \{ work \leftarrow (1 : revision),$
  $good \leftarrow \textbf{not}\ (1 : unhappy) \}$

M has two equilibria:

- $E_1 = (Th(\{unhappy, revision\}), Th(\{work\}))$ and
- $E_2 = (Th(\{unhappy \supset revision\}), Th(\{good, accepted\}))$

Groundedness

- Problem: **self-justifying beliefs**
- Present e.g. in Autoepistemic Logic:

$$L\, rich \supset rich$$

- Other nonmonotonic formalisms are "grounded," e.g.
  - Reiter's Default Logic,
  - Logic programs under answer set semantics (Gelfond & Lifschitz, 91),
  - ...
- Equilibria of MCSs are possibly ungrounded (wanted or not).

Example (ctd)

- Intuitively, $E_1 = (Th(\{unhappy, revision\}), Th(\{work\}))$ is ungrounded, since *unhappy* has a cyclic justification:

$$
\begin{aligned}
C_1 : kb_1 &= \{\, unhappy \supset revision \,\}; \\
br_1 &= \{\, unhappy \leftarrow (2 : work) \,\} \\
C_2 : kb_2 &= \{\, good : accepted / accepted \,\}; \\
br_2 &= \{ work \leftarrow (1 : revision), \;\; good \leftarrow \mathbf{not}\ (1 : unhappy) \}
\end{aligned}
$$

- Accept *unhappy* in C_1

Example (ctd)

- Intuitively, $E_1 = (Th(\{unhappy, revision\}), Th(\{work\}))$ is ungrounded, since *unhappy* has a cyclic justification:

$$
\begin{aligned}
C_1 : kb_1 &= \{\, unhappy \supset revision \,\}; \\
br_1 &= \{\, unhappy \leftarrow (2 : work) \,\} \\
C_2 : kb_2 &= \{\, good : accepted / accepted \,\}; \\
br_2 &= \{ work \leftarrow (1 : revision), \;\; good \leftarrow \textbf{not}\ (1 : unhappy) \}
\end{aligned}
$$

- Accept *unhappy* in C_1,
- since *work* is accepted in C_2

Example (ctd)

- Intuitively, $E_1 = (Th(\{unhappy, revision\}), Th(\{work\}))$ is ungrounded, since *unhappy* has a cyclic justification:

$$C_1 : kb_1 = \{\, unhappy \supset revision \,\};$$
$$br_1 = \{\, unhappy \leftarrow (2 : work) \,\}$$
$$C_2 : kb_2 = \{\, good : accepted / accepted \,\};$$
$$br_2 = \{ work \leftarrow (1 : revision), \;\; good \leftarrow \textbf{not}\ (1 : unhappy) \}$$

- Accept *unhappy* in C_1,
- since *work* is accepted in C_2,
- since *revision* is accepted in C_1

Example (ctd)

- Intuitively, $E_1 = (Th(\{unhappy, revision\}), Th(\{work\}))$ is ungrounded, since *unhappy* has a cyclic justification:

$$
\begin{aligned}
C_1 : kb_1 &= \{\, unhappy \supset revision \,\}; \\
br_1 &= \{\, unhappy \leftarrow (2 : work) \,\} \\
C_2 : kb_2 &= \{\, good : accepted / accepted \,\}; \\
br_2 &= \{ work \leftarrow (1 : revision), \;\; good \leftarrow \mathbf{not}\ (1 : unhappy) \}
\end{aligned}
$$

- Accept *unhappy* in C_1,
- since *work* is accepted in C_2,
- since *revision* is accepted in C_1,
- since *unhappy* is accepted in C_1.

Grounded Equilibria

- Only defined if all used logics L_i are *reducible*

Reducibility:

S_i is acceptable for kb_i iff it is the (single) acceptable belief set of a reduced (monotonic) KB $red_i(kb_i, S_i)$

- Assume that $red_i(kb_i, S_i) = kb_i$ if kb_i is from a monotonic target part of L_i, and that $red_i(kb_i, S_i)$ is anti-monotonic in S_i.
- The reducibility condition is satisfied by
  - all monotonic logics: trivial, *red* = identity,
  - Reiter's Default Logic: eliminate defeated defaults + consistency conditions from remaining defaults,
  - LPs under Answer Set Semantics: Gelfond-Lifschitz transformation
  - ...

Grounded Equilibria, II

- Given MCS $M = (C_1, \ldots, C_n)$ and belief state $S = (S_1, ..., S_n)$, use S to reduce
  - the KBs kb_i to $red_i(kb_i, S)$
  - the bridge rules br_i to br_i^S like in Answer Set Semantics, using a Gelfond-Lifschitz transformation
- The resulting monotonic MCS $M^S = (C_1^S, \ldots, C_n^S)$, has contexts

$$C_i^S = (L_i, red_i(kb_i, S), br_i^S), \quad i \in \{1, \ldots, n\}$$

Grounded Equilibrium.

A belief state S is a grounded equilibrium of a reducible MCS M iff S is the unique minimal equilibrium of M^S.

Here M is *reducible* if each L_i is reducible and the heads of bridge rules belong to the monotonic target language.

Example (ctd)

$$M: \quad C_1: kb_1 = \{ unhappy \supset revision \};$$
$$br_1 = \{ unhappy \leftarrow (2 : work) \}$$
$$C_2: kb_2 = \{ good : accepted / accepted \};$$
$$br_2 = \{ work \leftarrow (1 : revision),$$
$$good \leftarrow \mathbf{not}\ (1 : unhappy) \}$$

- Both
  - $E_1 = (Th(\{unhappy, revision\}), Th(\{work\}))$ and
  - $E_2 = (Th(\{unhappy \supset revision\}), Th(\{good, accepted\}))$

  are minimal
- E_1 violates groundedness: M^{E_1} has the single minimal equilibrium $(Th(\{unhappy \supset revision\}), Th(\emptyset)) \neq E_1$
- E_2 is the single grounded equilibrium of M

Example (ctd)

$$M^{E_1}: \quad C_1^{E_1}: red(kb_1, E_1) = \{\, unhappy \supset revision \,\};$$
$$br_1^{E_1} = \{\, unhappy \leftarrow (2 : work) \,\}$$
$$C_2^{E_1}: red(kb_2, E_1) = \{\, good : /accepted \,\};$$
$$br_2^{E_1} = \{work \leftarrow (1 : revision)\}$$

- Both
  - $E_1 = (Th(\{unhappy, revision\}), Th(\{work\}))$ and
  - $E_2 = (Th(\{unhappy \supset revision\}), Th(\{good, accepted\}))$

  are minimal
- E_1 violates groundedness: M^{E_1} has the single minimal equilibrium $(Th(\{unhappy \supset revision\}), Th(\emptyset)) \neq E_1$
- E_2 is the single grounded equilibrium of M

Example (ctd)

$$M^{E_1}: \quad C_1^{E_1}: red(kb_1, E_1) = \{\, unhappy \supset revision \,\};$$
$$br_1^{E_1} = \{\, unhappy \leftarrow (2: work) \,\}$$
$$C_2^{E_1}: red(kb_2, E_1) = \{\, good : / accepted \,\};$$
$$br_2^{E_1} = \{ work \leftarrow (1: revision) \}$$

- Both
  - $E_1 = (Th(\{unhappy, revision\}), Th(\{work\}))$ and
  - $E_2 = (Th(\{unhappy \supset revision\}), Th(\{good, accepted\}))$

  are minimal
- E_1 violates groundedness: M^{E_1} has the single minimal equilibrium $(Th(\{unhappy \supset revision\}), Th(\emptyset)) \neq E_1$
- E_2 is the single grounded equilibrium of M

Results

- **Minimality**: Grounded equilibria are minimal equilibria
- **Proper generalization** of monotonic MCS (Giunchiglia et al., AIJ 94+) and of Contextual Default Logic (Brewka et al., IJCAI 07)
- **Computational Complexity:** Assuming logics with poly-size *kernels* and *kernel reasoning* in Δ^p_{k+1}:
  - Deciding existence of a (grounded) equilibrium is in Σ^p_{k+1}
  - Brave reasoning from (grounded) equilibria is in Σ^p_{k+1}
  - Cautious reasoning from (grounded) equilibria is in Π^p_{k+1}

  (For Default Logic, ASP this is not harder than the basic logic)
- **Well-founded semantics** approximating the $\bigcap$ of all equilibria
- **Encoding** of (grounded) equilibria in *HEX-programs* (Eiter et al., IJCAI 05) for logics with *kernels*

Results

- **Minimality**: Grounded equilibria are minimal equilibria
- **Proper generalization** of monotonic MCS (Giunchiglia et al., AIJ 94+) and of Contextual Default Logic (Brewka et al., IJCAI 07)
- **Computational Complexity:** Assuming logics with poly-size *kernels* and *kernel reasoning* in Δ^p_{k+1}:
  - Deciding existence of a (grounded) equilibrium is in Σ^p_{k+1}
  - Brave reasoning from (grounded) equilibria is in Σ^p_{k+1}
  - Cautious reasoning from (grounded) equilibria is in Π^p_{k+1}

  (For Default Logic, ASP this is not harder than the basic logic)
- **Well-founded semantics** approximating the $\bigcap$ of all equilibria
- **Encoding** of (grounded) equilibria in *HEX-programs* (Eiter et al., IJCAI 05) for logics with *kernels*

Results

- **Minimality**: Grounded equilibria are minimal equilibria
- **Proper generalization** of monotonic MCS (Giunchiglia et al., AIJ 94+) and of Contextual Default Logic (Brewka et al., IJCAI 07)
- **Computational Complexity:** Assuming logics with poly-size *kernels* and *kernel reasoning* in Δ^p_{k+1}:
  - Deciding existence of a (grounded) equilibrium is in Σ^p_{k+1}
  - Brave reasoning from (grounded) equilibria is in Σ^p_{k+1}
  - Cautious reasoning from (grounded) equilibria is in Π^p_{k+1}

  (For Default Logic, ASP this is not harder than the basic logic)
- **Well-founded semantics** approximating the $\bigcap$ of all equilibria
- **Encoding** of (grounded) equilibria in *HEX-programs* (Eiter et al., IJCAI 05) for logics with *kernels*

Results

- **Minimality**: Grounded equilibria are minimal equilibria
- **Proper generalization** of monotonic MCS (Giunchiglia et al., AIJ 94+) and of Contextual Default Logic (Brewka et al., IJCAI 07)
- **Computational Complexity:** Assuming logics with poly-size *kernels* and *kernel reasoning* in Δ^p_{k+1}:
  - Deciding existence of a (grounded) equilibrium is in Σ^p_{k+1}
  - Brave reasoning from (grounded) equilibria is in Σ^p_{k+1}
  - Cautious reasoning from (grounded) equilibria is in Π^p_{k+1}

  (For Default Logic, ASP this is not harder than the basic logic)
- **Well-founded semantics** approximating the $\bigcap$ of all equilibria
- **Encoding** of (grounded) equilibria in *HEX-programs* (Eiter et al., IJCAI 05) for logics with *kernels*

Results

- **Minimality**: Grounded equilibria are minimal equilibria
- **Proper generalization** of monotonic MCS (Giunchiglia et al., AIJ 94+) and of Contextual Default Logic (Brewka et al., IJCAI 07)
- **Computational Complexity:** Assuming logics with poly-size *kernels* and *kernel reasoning* in Δ^p_{k+1}:
    - Deciding existence of a (grounded) equilibrium is in Σ^p_{k+1}
    - Brave reasoning from (grounded) equilibria is in Σ^p_{k+1}
    - Cautious reasoning from (grounded) equilibria is in Π^p_{k+1}

  (For Default Logic, ASP this is not harder than the basic logic)
- **Well-founded semantics** approximating the $\bigcap$ of all equilibria
- **Encoding** of (grounded) equilibria in *HEX-programs* (Eiter et al., IJCAI 05) for logics with *kernels*

Part III: Combination

MCS With Preferences

- General framework already admits:
  - Prioritized formalisms for contexts
  - Preference statements added to such contexts through bridge rule
- We also want:
  - Preferences among contexts: in case of conflict among bridge rules prefer information based on C_1 over information based on C_2
  - Preferences among bridge rules: in case of conflict among bridge rules prefer information based on r_1 over information based on r_2

MCS With Preferences (ctd)

- Issues:
  - What is a conflict?
    Application of bridge rules leads to inconsistent belief set; or non-existence of belief set; or non-existence of equilibrium?
  - Conflicts among bridge rules of different contexts?
  - Bridge rules with ordered disjunction?
  - ASO-style preference program for MCS? What kind of program?
- Also would like to:
  - Quantify over contexts
  - Represent information about contexts (trusted, reliable, ...) and reason about contexts
  - Use more general bridge rules, e.g. involving cardinality constraints
  - Express that a proposition is accepted if it holds, say, in more than half of the contexts (or use any other social choice rule)

Many open questions!!

Example: Information Fusion

Believe p if someone does and nobody believes $\neg p$:

$$\begin{array}{rcl} p & \leftarrow & (C{:}p), \textbf{not } rej(p) \\ rej(p) & \leftarrow & (C{:}\neg p) \end{array}$$

Believe p if someone you trust does and nobody you trust believes $\neg p$:

$$\begin{array}{rcl} p & \leftarrow & (C{:}p), trusted(C), \textbf{not } rej(p) \\ rej(p) & \leftarrow & (C{:}\neg p), trusted(C) \end{array}$$

Believe p if majority does:

$$p \leftarrow N\{(C{:}p) : context(C)\}N, N > n/2$$

Example: Information Fusion

Believe p if someone does and nobody believes $\neg p$:

$$\begin{array}{rcl} p & \leftarrow & (C{:}p), \textbf{not } rej(p) \\ rej(p) & \leftarrow & (C{:}\neg p) \end{array}$$

Believe p if someone you trust does and nobody you trust believes $\neg p$:

$$\begin{array}{rcl} p & \leftarrow & (C{:}p), trusted(C), \textbf{not } rej(p) \\ rej(p) & \leftarrow & (C{:}\neg p), trusted(C) \end{array}$$

Believe p if majority does:

$$p \leftarrow N\{(C{:}p) : context(C)\}N, N > n/2$$

Example: Information Fusion

Believe p if someone does and nobody believes $\neg p$:

$$\begin{array}{rcl} p & \leftarrow & (C{:}p), \textbf{not } rej(p) \\ rej(p) & \leftarrow & (C{:}\neg p) \end{array}$$

Believe p if someone you trust does and nobody you trust believes $\neg p$:

$$\begin{array}{rcl} p & \leftarrow & (C{:}p), trusted(C), \textbf{not } rej(p) \\ rej(p) & \leftarrow & (C{:}\neg p), trusted(C) \end{array}$$

Believe p if majority does:

$$p \leftarrow N\{(C{:}p) : context(C)\}N, N > n/2$$

Prioritized Information Fusion

Total preference order via context numbering: 1 < 2 < 3 ...

$$
\begin{aligned}
p &\leftarrow (C{:}p), \textbf{not } rej(C,p) \\
rej(C,p) &\leftarrow (C{:}p), (C'{:}\neg p), C < C'
\end{aligned}
$$

Partial preference order via predicate $\prec$, sceptical:

$$
\begin{aligned}
p &\leftarrow acc(p), \textbf{not } acc(\neg p) \\
acc(p) &\leftarrow (C{:}p), \textbf{not } rej(C,p) \\
rej(C,p) &\leftarrow (C{:}p), (C'{:}\neg p), C \prec C'
\end{aligned}
$$

Additionally quantifying over propositions allows for declarative representation of fusion strategies

Prioritized Information Fusion

Total preference order via context numbering: 1 < 2 < 3 ...

$$
\begin{array}{rcl}
p & \leftarrow & (C{:}p), \textbf{not}\ rej(C,p) \\
rej(C,p) & \leftarrow & (C{:}p), (C'{:}\neg p), C < C'
\end{array}
$$

Partial preference order via predicate $\prec$, sceptical:

$$
\begin{array}{rcl}
p & \leftarrow & acc(p), \textbf{not}\ acc(\neg p) \\
acc(p) & \leftarrow & (C{:}p), \textbf{not}\ rej(C,p) \\
rej(C,p) & \leftarrow & (C{:}p), (C'{:}\neg p), C \prec C'
\end{array}
$$

Additionally quantifying over propositions allows for declarative representation of fusion strategies

Prioritized Information Fusion

Total preference order via context numbering: 1 < 2 < 3 ...

$$
\begin{aligned}
p &\leftarrow (C{:}p), \textbf{not } rej(C,p)\\
rej(C,p) &\leftarrow (C{:}p), (C'{:}\neg p), C < C'
\end{aligned}
$$

Partial preference order via predicate $\prec$, sceptical:

$$
\begin{aligned}
p &\leftarrow acc(p), \textbf{not } acc(\neg p)\\
acc(p) &\leftarrow (C{:}p), \textbf{not } rej(C,p)\\
rej(C,p) &\leftarrow (C{:}p), (C'{:}\neg p), C \prec C'
\end{aligned}
$$

Additionally quantifying over propositions allows for declarative representation of fusion strategies

Conclusions

- Overview of approaches combining ASP with preferences
  - Focus on conditional formula preference
  - Based on satisfaction degree of rules
  - Potential for numerous applications
- Presented current work on MCS
  - Accommodating *heterogeneous, nonmonotonic* contexts, generalizing existing approaches
  - Capable of integrating logics like description logics, modal logics, default logics, logic programs, etc.
  - Discussed groundedness
- Gave outlook on integrating the two

Conclusions

- Overview of approaches combining ASP with preferences
  - Focus on conditional formula preference
  - Based on satisfaction degree of rules
  - Potential for numerous applications
- Presented current work on MCS
  - Accommodating *heterogeneous, nonmonotonic* contexts, generalizing existing approaches
  - Capable of integrating logics like description logics, modal logics, default logics, logic programs, etc.
  - Discussed groundedness
- Gave outlook on integrating the two

Conclusions

- Overview of approaches combining ASP with preferences
  - Focus on conditional formula preference
  - Based on satisfaction degree of rules
  - Potential for numerous applications
- Presented current work on MCS
  - Accommodating *heterogeneous, nonmonotonic* contexts, generalizing existing approaches
  - Capable of integrating logics like description logics, modal logics, default logics, logic programs, etc.
  - Discussed groundedness
- Gave outlook on integrating the two

Future Work

- Preferences
  - Better implementations
- Multi-context systems
  - Implementation based on HEX programs and DLVHEX
  - Weakening reducibility requirements
  - Application to problems in Data Integration and Semantic Web
- Integration of contexts and preferences
  - See Part III of this talk

THANK YOU!

Future Work

- Preferences
  - Better implementations
- Multi-context systems
  - Implementation based on HEX programs and DLVHEX
  - Weakening reducibility requirements
  - Application to problems in Data Integration and Semantic Web
- Integration of contexts and preferences
  - See Part III of this talk

THANK YOU!