Contexts and Preferences in Answer Set Programming

Gerhard Brewka

brewka@informatik.uni-leipzig.de

Universität Leipzig

Motivation 1: preferences

- preferences determine how agents decide and act
- pop up everywhere:

coffee	>	tea
car	>	train
relax	>	work
Chelsea London	>	Bayern München
Madonna	>	Britney Spears
marry	>	don’t marry
sleep	>	listen to talk

Preferences in AI

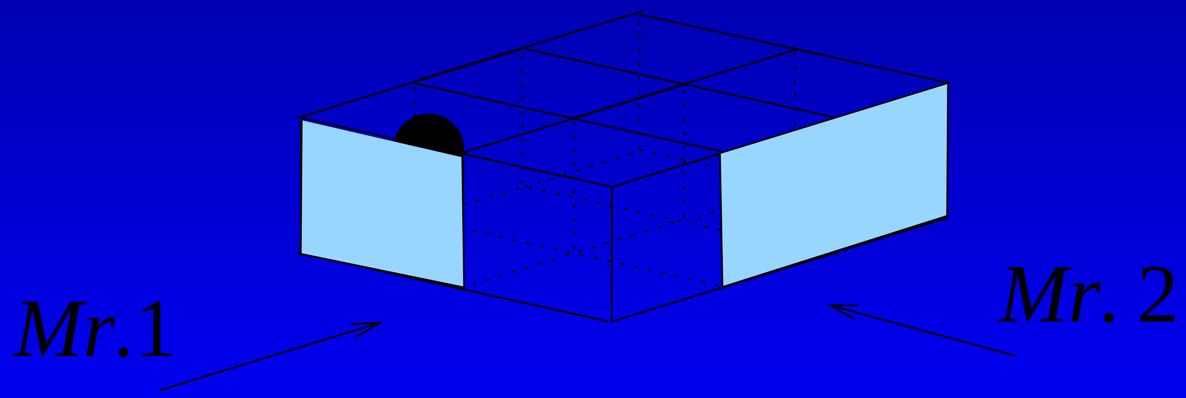
- diagnosis: prefer more plausible hypotheses
- planning/configuration: prefer cheaper plan; satisfy more important constraints
- revision: give up less preferred beliefs
- reasoning about action: prefer fewer unexplained changes
- ontologies: prefer more specific information
- legal/deontic reasoning: prefer more recent law
- linguistics: prefer more important constraints (optimality theory)

Issues

- how to represent space of outcomes
  here: answer sets
- how to represent preferences
  here: qualitative orderings
- how to interpret preferences
  here: defeasible multi-criteria
- how to represent dependencies
  here: rule preconditions specify context

Motivation 2: contexts

- McCarthy: *we offer no definition of context*
- viewpoints, perspectives, granularity, agents, ...
- (almost) independent units of reasoning
- locality: different languages, reasoning methods
- compatibility: information flow between contexts
- essential for handling complexity, consistency, ...



Outline

1. Motivation (done)
2. Answer set programming
3. Qualitative preferences
   - Formalisms
   - Applications
4. Contexts
   - Formalism
   - Examples
5. Conclusions

2. Answer set programming

Answer sets

- define semantics for logic programs with strict and default negation (extended LPs)
- rules of the form (a, b_i, c_j literals):

$$\boxed{a \leftarrow b_1, \ldots, b_n, \mathrm{not}\, c_1, \ldots, \mathrm{not}\, c_m}$$

- AS acceptable set of beliefs based on program
- requirements:
  closed: all rules used to generate AS
  if all $b_i \in$ AS, no $c_j \in$ AS, then $a \in$ AS
  grounded: a in AS implies derivation of a from rules whose not -preconds are not in AS

A simple test

To check whether S is AS of P

- remove S-defeated rules ($\mathrm{not}\, L$ in body, $L \in S$)
- remove not literals from remaining rules
- check whether S = closure of reduced program

A simple test

To check whether S is AS of P

- remove S-defeated rules ($\mathrm{not}\ L$ in body, $L \in S$)
- remove not literals from remaining rules
- check whether S = closure of reduced program

$$a \leftarrow \mathrm{not}\, b$$
$$b \leftarrow \mathrm{not}\, c$$

A simple test

To check whether S is AS of P

- remove S-defeated rules ($\mathrm{not}\ L$ in body, $L \in S$)
- remove not literals from remaining rules
- check whether S = closure of reduced program

$$a \leftarrow \cancel{\mathrm{not}\ b}$$
$$b \leftarrow \cancel{\mathrm{not}\ c}$$

$$\{a\}$$

NO, not closed

A simple test

To check whether S is AS of P

- remove S-defeated rules ($\mathrm{not}\ L$ in body, $L \in S$)
- remove not literals from remaining rules
- check whether S = closure of reduced program

$$\cancel{a \leftarrow \mathrm{not}\ b}$$
$$b \leftarrow \cancel{\mathrm{not}\ c}$$

$$\{a, b\}$$

NO, not grounded

A simple test

To check whether S is AS of P

- remove S-defeated rules ($\mathrm{not}\ L$ in body, $L \in S$)
- remove not literals from remaining rules
- check whether S = closure of reduced program

$$\cancel{a \leftarrow \mathrm{not}\ b}$$
$$b \leftarrow \cancel{\mathrm{not}\ c}$$

$$\{b\}$$

YES, grounded and closed

Example: graph coloring

Description of graph:

$$node(v_1), ..., node(v_n), edge(v_i, v_j), \ldots$$

Generate: every node needs exactly 1 color

$$\begin{array}{l} col(X,r) \leftarrow node(X), \text{not}\, col(X,b), \text{not}\, col(X,g) \\ col(X,b) \leftarrow node(X), \text{not}\, col(X,r), \text{not}\, col(X,g) \\ col(X,g) \leftarrow node(X), \text{not}\, col(X,r), \text{not}\, col(X,b) \end{array}$$

Test: linked nodes cannot have same color

$$\leftarrow edge(X,Y), col(X,Z), col(Y,Z)$$

Each answer set describes a solution!

Example: graph coloring

Description of graph:

$$node(v_1), ..., node(v_n), edge(v_i, v_j), \ldots$$

Generate: every node needs exactly 1 color

$$\begin{aligned}
col(X,r) &\leftarrow node(X), \mathrm{not}\, col(X,b), \mathrm{not}\, col(X,g) \\
col(X,b) &\leftarrow node(X), \mathrm{not}\, col(X,r), \mathrm{not}\, col(X,g) \\
col(X,g) &\leftarrow node(X), \mathrm{not}\, col(X,r), \mathrm{not}\, col(X,b)
\end{aligned}$$

Test: linked nodes cannot have same color

$$f \leftarrow edge(X,Y), col(X,Z), col(Y,Z), \mathrm{not}\, f$$

Each answer set describes a solution!

A useful language extension

Bounds on number of satisfied literals:

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

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

Allows us to replace 3 color assignment rules with:

$$1\{col(X,r), col(X,b), col(X,g)\}1 \leftarrow node(X)$$

or, if extension of $color$ is $\{r, b, g\}$:

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

Why LPs under AS semantics?

- simple yet expressive language
- transitive closure, nonmonotonic rules, ...

$$flies(X) \leftarrow \mathrm{not}\, ab(X), bird(X)$$

- simple epistemic distinctions, particularly useful for preference reasoning

$$safe > \mathrm{not}\, \neg safe > \neg safe$$

- interesting implementations: dlv, Smodels, nomore, ASSAT ...
- interesting applications: configuration, diagnosis, planning, reasoning about action, shuttle control, model checking, information integration, ...

3. Qualitative Preferences

3.1 Formalisms

Adding preferences to ASP

	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:

$$\boxed{C_1 \times \ldots \times C_n \leftarrow body}$$

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
- satisfy rules to different degrees, depending on best satisfied head literal
- use degrees to define global preference relation on answer sets
- different options how to do this

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

$$\boxed{C_1 > \ldots > C_k \leftarrow body}$$

  C_i boolean combination built using $\vee$, $\wedge$, $\neg$, not
- rule satisfaction and combination as for LPODs

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:

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

- LPODs: compact and readable representations

3.2 Applications

Configuration

- often represented as AND/OR trees
- simple representation with Smodels cardinalities:

$$\begin{aligned} 4\{starter, main, dessert, drink\}4 &\leftarrow dinner \\ 1\{soup, salad\}1 &\leftarrow starter \\ 1\{fish, beef, lasagne\}1 &\leftarrow main \end{aligned}$$

- add case description and preferences, e.g.

$$\begin{aligned} fish \vee beef > lasagne & \\ beer > wine &\leftarrow beef \\ wine > beer &\leftarrow \text{not}\, beef \end{aligned}$$

- preferred answer sets: optimal configurations

Abductive diagnosis

$$
\begin{array}{ll}
H: & measles, flu, migraine \\
O: & headache, fever \\
K: & fever \leftarrow measles \\
 & red\text{-}spots \leftarrow measles \\
 & headache \leftarrow migraine \\
 & nausea \leftarrow migraine \\
 & fever \leftarrow flu \\
 & headache \leftarrow flu
\end{array}
$$

Abductive diagnosis

$$
\begin{array}{ll}
H: & measles, flu, migraine \\
O: & headache, fever \\
K: & fever \leftarrow measles \\
 & red\text{-}spots \leftarrow measles \\
 & headache \leftarrow migraine \\
 & nausea \leftarrow migraine \\
 & fever \leftarrow flu \\
 & headache \leftarrow flu
\end{array}
$$

$$\neg measles \times measles \qquad \neg flu \times flu$$
$$\neg migraine \times migraine$$

Abductive diagnosis

$$
\begin{array}{ll}
H: & measles, flu, migraine \\
O: & headache, fever \\
K: & fever \leftarrow measles \\
 & red\text{-}spots \leftarrow measles \\
 & headache \leftarrow migraine \\
 & nausea \leftarrow migraine \\
 & fever \leftarrow flu \\
 & headache \leftarrow flu
\end{array}
$$

$$
\begin{array}{ll}
\neg measles \times measles & \neg flu \times flu \\
\neg migraine \times migraine &
\end{array}
$$

$$
\begin{array}{ll}
\leftarrow \text{not } headache & \leftarrow \text{not } fever
\end{array}
$$

Abductive diagnosis

$$
\begin{array}{rl}
H: & measles, flu, migraine \\
O: & headache, fever \\
K: & fever \leftarrow measles \\
 & red\text{-}spots \leftarrow measles \\
 & headache \leftarrow migraine \\
 & nausea \leftarrow migraine \\
 & fever \leftarrow flu \\
 & headache \leftarrow flu
\end{array}
$$

$$
\begin{array}{ll}
\neg measles \times measles & \neg flu \times flu \\
\neg migraine \times migraine &
\end{array}
$$

$$
\begin{array}{ll}
\leftarrow \text{not } headache & \leftarrow \text{not } fever
\end{array}
$$

inclusion preferred: $\{migraine, measles\}$, $\{flu\}$

Inconsistency handling

- program P, possibly inconsistent; consistency restoring rules R
- names N_P and N_R for rules in P and R
- generate weakening of $P \cup R$ by replacing

$$\boxed{head \leftarrow body} \quad \text{with} \quad \boxed{head \leftarrow body, r_i}$$

  where r_i rule's name
- add $\{r \times \neg r \mid r \in N_P\} \cup \{\neg r \times r \mid r \in N_R\}$
- minimal set of P-rules switched off, minimal set of R-rules switched on

Game theory

Prisoners' dilemma

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

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

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

Related issues

- 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: combines graph based methods with flexibility of ASO preferences

4. Contexts

4.1 Formalism

Contexts: the Trento school

Multi-context system

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

each $T_i = (L_i, \Omega_i, \Delta_i)$ formal system, $Delta_{br}$ bridge rules using labeled formulas $c : p$ with $p \in L_c$.

- semantics: local models + compatibility
- information flow across contexts via bridge rules
- reasoning within/across contexts monotonic
- exception (Roelofsen/Serafini 05) has problems

Contextual default logic

Contextual default theory $((D_1, W_1), \ldots, (D_n, W_n))$.

(D_i, W_i) default theory, default rules in D_i possibly refer to other contexts.

$\Gamma(S_1, \ldots, S_n)$ minimal tuple $(S'_1, \ldots, S'_n)$ with:

1. $W_i \subseteq S'_i$,
2. S'_i deductively closed (over L_i), and
3. $(c_1 : p_1), ..., (c_t : p_t) : (c_{t+1} : q_1), ..., (c_{t+k} : q_k)/r \in D_i$ and for all i, j: $p_i \in S'_{c_i}$ and $\neg q_j \notin S_{c_{t+j}}$, then $r \in S'_i$.

Extension fixpoint of Γ.

Contextual logic programming

- Contextual LP system: $C = (P_1, \ldots, P_n)$.
- P_i contextual LP: labeled literals $(c : l)$ allowed in rule bodies (c context, l literal in c's language).
- Minimal context model of definite C (no **not**): smallest $(S_1, \ldots, S_n)$ with:
  1. $a \in S_i$ if $a \leftarrow (c_1 : b_1), \ldots, (c_k : b_k) \in P_i$, $b_1 \in S_{c_1}$, . . ., $b_k \in S_{c_k}$,
  2. $S_i = Lit_i$ if S_i has complementary literals.

Answer sets

- $C = (P_1, \ldots, P_n)$ arbitrary system,
  $S = (S_1, \ldots, S_n)$, S_i literals in P_i's language.
- C^S obtained from C by
  1. deleting rules with literal **not** $(c : l)$ s.t. $l \in S_c$,
  2. deleting **not** literals from remaining rules.
- C^S definite, has minimal context model M_{C^S}.
- S answer set iff $S = M_{C^S}$.

Also possible to define skeptical inference à la WFS

4.2 Examples

Abstraction

Car domain, P_1 abstraction of P_2, P_1 may contain

$$\begin{array}{rcl} expensive(X) & \leftarrow & 2 : price(X,Y), Y \geq 30000 \\ sportive(X) & \leftarrow & 2 : speed(X,Y), 180 < Y, \\ & & \textbf{not } bad\text{-}accel(X) \\ bad\text{-}accel(X) & \leftarrow & 2 : 0\text{-}to\text{-}100(X,Y), Y > 10 \end{array}$$

Assume now intelligent grounder instantiating context, proposition and domain variables correctly.

Information fusion

believe p if someone does and nobody believes $-p$:

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

believe p if someone you trust does and nobody you trust believes $-p$:

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

believe p if majority does:

$$P \;\leftarrow\; N\{(C:P) : con(C)\}N, N > n/2$$

Information fusion, ctd.

total preference order via numbering: 1 < 2 < 3 ...

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

partial preference order via predicate $\prec$, sceptical

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

Prisoners' dilemma II

2-context system (P_1, P_2) with P_1:

$$
\begin{array}{rcl}
choose(d) & \leftarrow & \textbf{not}\ choose(c) \\
choose(c) & \leftarrow & \textbf{not}\ choose(d) \\
best(d) & \leftarrow & (2 : choose(c)) \\
best(d) & \leftarrow & (2 : choose(d)) \\
 & \leftarrow & choose(X), \textbf{not}\ best(X)
\end{array}
$$

P_2 = P_1 with 2 replaced by 1.

Answer set = Nash equilibrium:

$$(\{choose(d), best(d)\}, \{choose(d), best(d)\})$$

Voting

Simple majority vote:

$$
\begin{aligned}
votes(X, N) &\leftarrow N\{(C : best(X)) : con(C)\}N, cnd(X) \\
wins(X) &\leftarrow \textbf{not}\ \neg wins(X) \\
\neg wins(X) &\leftarrow votes(X, N), votes(Y, M), M > N
\end{aligned}
$$

Condorcet rule:

$$
\begin{aligned}
beats(X, Y, N) &\leftarrow N\{(C : beats(X, Y)) : con(C)\}N, \\
&\qquad cnd(X), cnd(Y) \\
wins(X) &\leftarrow \textbf{not}\ \neg wins(X) \\
\neg wins(X) &\leftarrow beats(X, Y, N)), beats(Y, X, M), \\
&\qquad M \geq N
\end{aligned}
$$

5. Conclusions

What has been achieved?

- ASP a promising declarative paradigm
- simple yet expressive, interesting applications, interesting solvers
- adding preferences has great potential
- presented two approaches and possible applications
- showed how to add simple notion of context
- missing: current context, switching context, identification of most adequate context, ...

Future work

Better integration of contexts and preferences

- use prioritized logic programs within context
- use priorities among bridge rules
- modify priorities in a context using bridge rules

Thanks to:

I. Niemelä, M. Truszczynski, F. Roelofsen, L. Serafini;
you for listening

References

1. G. Brewka, Logic programs with ordered disjunction, Proc. AAAI-02, Edmonton, 2002
2. G. Brewka, I. Niemelä, M. Truszczynski: Answer set optimization, Proc. IJCAI-03, Acapulco, 2003
3. G. Brewka, S. Benferhat, D. Le Berre: Qualitative choice logic, Artificial Intelligence 157(1-2), Special Issue on Nonmonotonic Reasoning, 2004, 203-237
4. G. Brewka, I. Niemelä, T. Syrjänen: Logic programs with ordered disjunction, Computational Intelligence 20 (2), 2004, 335-357
5. G. Brewka: A rank based description language for qualitative preferences, Proc. ECAI-04, Valencia, 2004, 303-307 (also in: Proc. NMR 2004, 88-93)
6. G. Brewka: Complex preferences for answer set optimization, Proc. KR-04, Whistler, Canada, AAAI Press, 2004, 213-223
7. G. Brewka, I. Niemelä, M. Truszczynski: Prioritized component systems, Proc. AAAI-05, Pittsburgh 2005
8. G. Brewka, J. Dix: Knowledge Representation with Logic Programs, in: D. Gabbay and F. Guenthner (eds.), Handbook of Philosophical Logic, Vol. 12, Springer, 2005, 1-85
9. G. Brewka: Answer Sets and Qualitative Decision Making, Synthese 146 (2005), 171-181
10. G. Brewka, F. Roelofsen, L. Serafini: Contextual Default Reasoning, IJCAI-07