Reasoning with Prioritized Defaults

Michail Gelfond, Tran Cao Son

Computer Science Department
University of Texas at El Paso

1998

Presented by:
Jakub Podlaha,
Vladislav Vavra

Outline

Outline

Why Prioritized Defaults?

- Defaults in natural language
- Defaults with contradictory conclusions
- Expressing relative strengths of defaults
- e.g. Legal Reasoning, Reasoning with Experts Knowledge

Outline

Two Possible Approaches

- Develop language to express prioritized defaults (special semantics)
  - ex. generate answer sets and reduce by preference relation
- Use standard Logic programming augmented by the preference relation

Outline

The Paper's Approach

- Understand narrow sence "a's are normally b's"
- Allow dynamic priorities (i.e. defaults with preference relation)
- Elaboration tolerant
- Give semantics without new general purpose nonmonotonic formalism
- Some inference mechanism already available

Outline

Example - Programming Students

```
student(mary).  dept(cs).   is_in(mary,cs).
student(mike).  dept(art).  is_in(mike,art).
student(sam).   dept(cis).  is_in(sam,cis).

default(d1(S), -can_progr(S), [student(S)]).
default(d2(S),  can_progr(S), [student(S),
                               is_in(S,cs)]).

prefer(d2(S), d1(S)).

rule(r1(S), can_read(S), [student(S)]).
```

Outline

Domain Description

- Facts
  - all the facts from our original domain
  - `prefer(D1, D2)`
  - `conflict(D1, D2)`
- Laws
  - `rule(R, H, Body)`
  - `default(D, H, Body)`

Domain Description - Refinement of Facts

Basic relations

- `holds(l)` denotes that literal `l` strictly holds
- `holds_by_default(l)` denotes that literal `l` holds defeasibly i.e. by some default

Translating facts of our domain into ASP

- $\mathcal{P}(\mathcal{D}) = \mathcal{P} \cup \{holds(l) \mid l \in fact(\mathcal{D})\} \cup laws(\mathcal{D})$.

Outline

Axioms

- handout + slides

Examples

- go through programming students example, show it running on computer
- example of flying penguin, ask them and let them quess, then show the correct answer on pc
- maybe use example 3.3 (not yet ready)

Extending the language

- $default(d, l_0, [l_1, l_2, ...l_m], [l_m + 1, ...l_n])$
- show new axioms
- example 4.1 : who can vote?

Weak Exceptions

- "do not apply default d to objects satisfiyng property p"
- How to do it?
- $exception(d, [l_1, ..., l_n][l_n + 1, ...ln + m])$
- Add new axiom: defeated(D) :- exception .....

Cautions reasoning

- Two contrary defaults - no answer set will be resolved
- Add new axiom

Summary