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

1. Introduction
   - Why Prioritize Defaults?
   - Two possible approaches
   - Used Approach
2. The Language of Prioritized Defaults
   - Syntax
   - Axioms of P

Outline

1. Introduction
   - Why Prioritize Defaults?
   - Two possible approaches
   - Used Approach
2. The Language of Prioritized Defaults
   - Syntax
   - Axioms of P

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

1. Introduction
   - Why Prioritize Defaults?
   - Two possible approaches
   - Used Approach
2. The Language of Prioritized Defaults
   - Syntax
   - Axioms of P

Two Possible Approaches

- Develop language to express Default logic (special syntax)
- Use standard Logic programming syntax augmented by the preference relation

Outline

1. Introduction
   - Why Prioritize Defaults?
   - Two possible approaches
   - Used Approach

2. The Language of Prioritized Defaults
   - Syntax
   - Axioms of P

Used Approach

- Design simple language $\mathcal{L}$
- Understands narrow sence "a's are normally b's"
- Allows dynamic priorities (i.e. defaults with preference relation)
- Gives semantics without new general purpose nonmonotonic formalism
- Elaboration tolerant
- Some inference mechanism already available

The Language of Prioritized Defaults

- Logic program $\mathcal{P}$ composed of:
  - domain independent axioms (denoted $\mathcal{P}_\sigma$)
  - domain description (denoted $\mathcal{D}$)
  - notion of entailment between query and domain description

Outline

1. Introduction
   - Why Prioritize Defaults?
   - Two possible approaches
   - Used Approach
2. The Language of Prioritized Defaults
   - Syntax
   - Axioms of P

Domain description

- Introducing language $\mathcal{L}_0(\sigma)$
- parametrized by sorted signature σ
  - names from user's domain
  - terms to describe defaults and strict rules
- describing Domain knowledge using:
  - collection of literals of σ
  - statemens describing strict rules, defaults and preferences

Domain description

- Terms to Describe Defaults and Strict Rules:

$$rule(r_0, l_0, [l_1, \ldots l_m]) \tag{1}$$
$$default(d_0, l_0, [l_1, \ldots l_m]) \tag{2}$$
$$conflict(d_1, d_2) \tag{3}$$
$$prefer(d_1, d_2) \tag{4}$$

Domain description - intuition

- "logic counter-part" as intuitive explanaiton
- rule r_0: $l_0 \leftarrow l_1, \ldots, l_n$.
- default d_0: $l_0 \leftarrow l_1, \ldots, l_n, not\ \neg l_0$.

Example - programming students

dept, mary, isin etc..

```
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)]).
```

Outline

1 Introduction
- Why Prioritize Defaults?
- Two possible approaches
- Used Approach

2 The Language of Prioritized Defaults
- Syntax
- Axioms of P

Entailment in domain description

Definition 2.2 We say that a domain description $\mathcal{D}$ entails a query q ($\mathcal{D} \models q$) if q belongs to every answer set of the program
$\mathcal{P}_\sigma(\mathcal{D}) = \mathcal{P}_\sigma \cup \{holds(l) \mid l \in fact(\mathcal{D})\} \cup laws(\mathcal{D})$.
$laws(\mathcal{D})$ denotes set of statements of the form 1 and 2

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