

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
 - The Paper's Approach
- 2 The Language of Prioritized Defaults
 - Domain Description
 - Domain Independent Axioms

Outline

- 1 Introduction
 - Why Prioritize Defaults?
 - Two possible approaches
 - The Paper's Approach
- 2 The Language of Prioritized Defaults
 - Domain Description
 - Domain Independent Axioms

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**
- The Paper's Approach

2

The Language of Prioritized Defaults

- Domain Description
- Domain Independent Axioms

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

- 1 Introduction
 - Why Prioritize Defaults?
 - Two possible approaches
 - The Paper's Approach
- 2 The Language of Prioritized Defaults
 - Domain Description
 - Domain Independent Axioms

The Paper's Approach

- Understand narrow sense "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

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(d1(S), d2(S)).
```

The Language of Prioritized Defaults

Logic program composed of:

- Domain description
 - Facts
 - Laws about defaults and preferences
 - `rule(R, H, Body)`
 - `default(D, H, Body)`
- Domain independent axioms

Outline

1 Introduction

- Why Prioritize Defaults?
- Two possible approaches
- The Paper's Approach

2 The Language of Prioritized Defaults

- Domain Description
- Domain Independent Axioms

Domain Description

NAHRADIT ZA HOLDY

- 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

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

$$\mathcal{P}_\sigma(\mathcal{D}) = \mathcal{P}_\sigma \cup \{holds(I) \mid I \in fact(\mathcal{D})\} \cup laws(\mathcal{D}).$$

$laws(\mathcal{D})$ denotes set of statements of the form 1 and 2

Domain Description

- Terms to Describe Defaults and Strict Rules:

$rule(r_0, l_0, [l_1, \dots l_m])$ (1)

$default(d_0, l_0, [l_1, \dots l_m])$ (2)

$conflict(d_1, d_2)$ (3)

$prefer(d_1, d_2)$ (4)

Domain Description - Intuition

- "logic counter-part" as intuitive explanation
- rule $r_0: l_0 \leftarrow l_1, \dots, l_n$.
- default $d_0: l_0 \leftarrow l_1, \dots, l_n, \text{not } \neg l_0$.

Outline

1 Introduction

- Why Prioritize Defaults?
- Two possible approaches
- The Paper's Approach

2 The Language of Prioritized Defaults

- Domain Description
- Domain Independent Axioms

Axioms

- handout + slides

Examples

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

Extending the language

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

Weak Exceptions

- "do not apply default d to objects satisfying property p "
- How to do it?
- $\text{exception}(d, [l_1, \dots, l_n][l_n + 1, \dots, l_n + m])$
- Add new axiom: $\text{defeated}(D) \text{ :- exception } \dots\dots$

Cautions reasoning

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

Summary