

Systems and Solving Techniques for Knowledge Representation (Introduction)

Marco Maratea
University of Genoa, Italy

066 011 Double degree programme Computational Logic
(Erasmus-Mundus)

066 931 Computational Intelligence

066 937 Software Engineering & Internet Computing
Institute of Information Systems

Knowledge Representation and Reasoning

The need for knowledge representation

- e.g., intelligent agents, problem solving ...

Declarative vs Imperative Knowledge

- expressed in declarative sentences or propositions
- the knowledge of how to perform some task (implicit in some sequence of steps)

How to express/represent declarative knowledge?

- well-known problems of natural language
- formal language → Logic

Processing knowledge to obtain new knowledge

- reasoning via Systems and Solving techniques

Knowledge Representation and Reasoning

The need for knowledge representation

- e.g., intelligent agents, problem solving ...

Declarative vs Imperative Knowledge

- expressed in declarative sentences or propositions
- the knowledge of how to perform some task (implicit in some sequence of steps)

How to express/represent declarative knowledge?

- well-known problems of natural language
- formal language → Logic

Processing knowledge to obtain new knowledge

- reasoning via Systems and Solving techniques

Knowledge Representation and Reasoning

The need for knowledge representation

- e.g., intelligent agents, problem solving ...

Declarative vs Imperative Knowledge

- expressed in declarative sentences or propositions
- the knowledge of how to perform some task
(implicit in some sequence of steps)

How to express/represent declarative knowledge?

- well-known problems of natural language
- formal language → Logic

Processing knowledge to obtain new knowledge

- reasoning via Systems and Solving techniques

Knowledge Representation and Reasoning

The need for knowledge representation

- e.g., intelligent agents, problem solving ...

Declarative vs Imperative Knowledge

- expressed in declarative sentences or propositions
- the knowledge of how to perform some task
(implicit in some sequence of steps)

How to express/represent declarative knowledge?

- well-known problems of natural language
- formal language → Logic

Processing knowledge to obtain new knowledge

- reasoning via Systems and Solving techniques

Answer Set Programming (ASP) (1)

- **Answer Set Programming (ASP)**

- Declarative paradigm
- Non-monotonic reasoning and logic programming
- Stable model semantics

- **Expressive KR Language**

- Roots in Datalog
- Default negation, Disjunction, Constraints, etc
- Aggregates, Weak constraints, etc.
- Can model problems up to Σ_2^P/Π_2^P
 - i.e., problems not (polynomially) translatable to SAT or CSP

Answer Set Programming (ASP) (2)

Idea:

- 1 Represent a computational problem by a logic program
- 2 Answer sets correspond to problem solutions
- 3 Use an ASP solver to find these solutions

Well-known Example

Example (3-col)

Problem: Given a graph, assign one color out of 3 colors to each node s.t. adjacent nodes have different colors.

Input: a graph represented by *node/1* and *edge/2*.

% Each node X should be colored red or yellow or green.
 $col(X, red) \mid col(X, yellow) \mid col(X, green) \text{ :- } node(X).$

% Adjacent nodes cannot have the same color.
 $\text{:- } edge(X, Y), col(X, C), col(Y, C).$

Answer Set Programming (ASP) (3)

- **Applications in several fields**

- Artificial Intelligence, KR& R.,
- Information Integration, Data cleaning, Bioinformatics, etc
- employed for developing industrial applications

- **Robust and efficient implementations**

- Smodels, DLV, Clasp, GnT, Wasp and others
- *continuous improvement (see the ASP competitions)*

Course Goals (1)

Declarative knowledge

- expressed by means of declarative sentences in a symbolic language

Processing declarative knowledge

- obtained by running a procedure/system that works on these sentences

Goals:

- Study foundations and practicalities of a logic-based declarative formalism for KR& R
- Focus on Answer Set Programming
- Focus on Systems and Solving techniques

Course Goals (2)

What will you bring at home with you?

- an understanding of the foundations of ASP
- a methodology for developing ASP programs
- a detailed overview of the evaluation algorithms and ASP systems

Course Overview

- Datalog
- ASP: Syntax and Semantics for basic ASP programs
- ASP: Extensions
- Solving techniques for normal ASP programs
- Solving techniques for disjunctive ASP programs
- Portfolio ASP solving (focus on Multi-Engine approach)

Course organization

3 ECTS = 75 Hours

- Introduction to course: 1h
- Lectures: 23h (theory + exercises)
- Additional reading and preparation for exam 51h

Exam / Evaluation

- Written test
- Class participation (doing and presenting exercises) taken into account

Course dates

See next slide for details

Office hour

- by email (marco@dibris.unige.it; mmaratea@dbai),
- by appointment (at HA 03 14)

Course dates and Material

All lectures in this (Seminarraum Gödel) room

- 1 13:00-15:00 Fri 16.10.2015
- 2 16:00-18:00 Mon 19.10.2015
- 3 10:00-13:00 Fri 23.10.2015
- 4 11:00-13:00 Wed 28.10.2015
- 5 11:00-14:00 Fri 30.10.2015
- 6 10:00-13:00 Wed 04.11.2015
- 7 10:00-13:00 Fri 06.11.2015
- 8 16:00-18:00 Mon 09.11.2015
- 9 10:00-13:00 Fri 13.11.2015 (Final test)

Slides and exercises will be made available at

<http://www.star.dist.unige.it/~marco/SSTKR/>

**Thanks to Francesco Ricca for a preliminary
version of these slides**