Knowledge Representation and Reasoning

Bernhard Nebel  Jussi Rintanen  Stefan Wölfl

Albert-Ludwigs-Universität Freiburg

Winter term 2004/2005

Lectures: Where, When, Webpage

Where
Lecture hall 52-02-017

When
Wednesday 14:15–15:50, Friday: 14:15–15:00 (+ exercises)

Christmas break
Last lecture before Christmas: Wednesday, December 22
First lecture after Christmas: Friday, January 7

Web page
http://www.informatik.uni-freiburg.de/~ki/teaching/ws0405/krr/

Lecturers

Prof Dr. Bernhard Nebel
Room 52-00-028
Consultation: Tuesday 14:00-15:00 and by appointment
Phone: 0761/203-8221
email: nebel@informatik.uni-freiburg.de

Dr. Jussi Rintanen
Room 52-00-045, Consultation: by appointment
Phone: 0761/203-8230
email: rintanen@informatik.uni-freiburg.de

Dr. Stefan Wölfl
Room 52-00-023, Consultation: by appointment
Phone: 0761/203-8247
email: woelfl@informatik.uni-freiburg.de
Exercises I

**Where**
Lecture hall 52-02-017

**When**
Friday 15:05-15:50

**exercise assistant: Malte Helmert**
Room 52-00-030, Phone: 0761/203-8225
email: helmert@informatik.uni-freiburg.de

Exercises II

- Exercises will be given at the web page on Wednesdays. (However, first exercise on Friday, October 22.)
- Solutions can be given in English and German.
- Students can work in pairs and hand in one solution.
- Larger groups and copied results will not be accepted.
- Previous week's exercises have to be handed in before the lecture on Wednesday.

Examination & Schein

- A written examination takes place in the semester pause.
- The examination is obligatory for *ACS Master* students.
- Grade:
  - max 100 points from the exam
  - max 10 bonus points from exercises
  - max 10 bonus points from projects (programming exercises)

Course Prerequisites & Goals

**Goals**

- Acquiring skills in representing knowledge
- Understanding the principles behind different knowledge representation techniques
- Being able to read and understand research literature in the area of KR&R
- Being able to complete a project in this research area

**Prerequisites**

- Basic knowledge in the area of AI
- Basic knowledge in formal logic
- Basic knowledge in theoretical computer science
Motivation

AI and Knowledge Representation

- **AI** can be described as: The study of **intelligent behavior** achieved through **computational means**

- **Knowledge representation and reasoning** could then be viewed as the study of how to **reason** (compute) with **knowledge** in order to decide what to do.
- Before we can start reasoning with knowledge, we have to **represent** it.

Knowledge

- We understand by “knowledge” all kinds of facts about the world.
- Knowledge is necessary for intelligent behavior (human beings, robots).
- What is knowledge? We shall not try to answer this question!
- Instead, in this course we consider “representations of knowledge”.

Representation

- If A represents B, then A stands for B and is usually more easily accessible than B.
- In our case we are interested in **groups of symbols** that stand for some **proposition**.

Knowledge Representation

The field of study concerned with **representations** of propositions (that are believed by some agent)

Reasoning

- **Reasoning** is the use of representations of propositions in order to derive new ones.
- While propositions are abstract objects, their representations are concrete objects and can be easily manipulated.
- Reasoning can be as easy as arithmetics ↔ mechanical symbol manipulation.
- For example:
  - raining is true
  - IF raining is true THEN wet street is true
  - wet street is true
Why is Knowledge Representation and Reasoning Useful?

- Describing/understanding the behavior of systems in terms of the knowledge it has.
- Generating the behavior of a system!
  - Declarative knowledge can be separated from its possible usages. (compare: procedural knowledge)
  - Understanding the behavior of an intelligent system in terms of the represented knowledge makes debugging and understanding much easier.
  - Modifications and extensions are also much easier to perform.

Advantages of Knowledge-Based Systems

- Why not use the first variant of the Prolog program?
- We can add new tasks and make them depend on previous knowledge.
- We can extend existing behavior by adding new facts.
- We can easily explain and justify the behavior.

Knowledge-Based Systems: An Example

```prolog
printC(snow) :- !, write("It’s white").
printC(grass) :- !, write("It’s green").
printC(sky) :- !, write("It’s yellow").
printC(X) :- !, write("Beats me").
```

```prolog
printC(X) :- color(X,Y), !, write("It’s "), write(Y).
printC(X) :- write("Beats me").
color(snow,white).
color(sky,yellow).
color(X,Y) :- madeof(X,Z), color(Z,Y).
madeof(grass,vegetation).
color(vegetation,green).
```

Why Reasoning?

- Note: there was no explicit statement about the color of grass in the program.
- In general: many facts will be there only implicitly.
- Use concept of entailment/logical implication.
- Can/shall we compute all implicit (all entailed) facts?
- It may be computationally too expensive.
The Role of Formal Logic

- Formal logic is the field of study of entailment relations, formal languages, truth conditions, semantics, and inference.
- All propositions are represented as formulae which have a semantics according to the logic in question.
- Formal logics give us a framework to discuss different kinds of reasoning.

Different Kinds of Reasoning

- Usually, we are interested in deriving implicit, entailed facts from a given collection of explicitly represented facts
  - in a logically sound way (the derived proposition must be true, given that the premises are true)
  - and complete way (all true consequences can be derived)
- Sometimes, however, we want logically unsound derivations (e.g. reasoning based on assumptions.)
- Sometimes, we want to give up completeness (e.g. for efficiency reasons.)

Model-Finding and Satisfiability

- In planning and configuration tasks, we often get a set of constraints and a goal specification. We then have to find a solution satisfying all the constraints.
  - Either round or square
  - Either red or blue
  - If red and round or if blue and square then wood
  - If blue then metallic
  - If square then not metallic
  - If red then square
  - square

  One solution: square, not metallic, red, wood
- Does not logically follow, but is one possible assignment (or model).

Abduction: Inference to the Best Explanation

- In diagnosis tasks, we often have to find a good explanation for a given observation or symptom.
- Given a background theory, a set of explanations and an observation, find the most likely set of explanations.
  - earthquake implies alarm
  - burglar implies alarm
  - { earthquake, burglar } is the set of abducibles
  - alarm is observed
  - One explanation is earthquake ...
- There can be many possible explanations.
- Not a sound inference
Default Reasoning: Jumping to Conclusions

- Often we do not have enough information, but nevertheless want to reach a conclusion (that is likely to be true).
- In the absence of evidence to the contrary, we jump to a conclusion.
  - Birds are usually able to fly.
  - Tweety is a bird.
  - So, you would expect that Tweety is able to fly.
- Unsound conclusion
- It might be necessary to withdraw conclusions when evidence to the contrary becomes available \( \rightarrow \) nonmonotonic reasoning.

The Role of Complexity Theory (1)

- Intelligent behavior is based on a vast amount of knowledge: Reddy's (1988) estimate is 70000 knowledge "units".
- Because of the huge amount of knowledge we have represented, reasoning should be easy in the complexity theory sense.
- Reasoning should scale well: we need efficient reasoning algorithms.

The Role of Complexity Theory (2)

Use complexity theory and recursion theory to
- determine the complexity of reasoning problems,
- compare and classify different approaches based on complexity results,
- identify easy (polynomial-time) special cases,
- use heuristics/approximations for provably hard problems, and
- choose among different approaches.

Course Outline

1. Introduction
2. Reminder: Classical Logic
3. A New Logic: Boxes and Diamonds
4. Nonmonotonic Logics
5. Qualitative Spatial and Temporal Reasoning
6. Description Logics
Literature I


Literature II