Knowledge Representation and Reasoning CS616

Time: 17–21 March 2003.

www: Do check the website regularly.

Lecture Course comes in 4 parts:

Part I (Dix,3) Predicate Logic

Part II (Schmidt,15) Modal Logic

Part III (Dix,17) Answer Set Programming

Part IV (Schmidt,5) Description Logic

Exam: NN

Overview

1. Predicate Logic
2. Modal Logic
3. Answer Set Programming
4. Description Logics

1 Predicate Logic

1.1 From prop. to predicate logic

1.2 Some examples

1.3 (Un-) decidability

1.4 Restrictions

Why predicate logic? What is missing in propositional logic?

How to formalise problems in logic?

Logical puzzles (who let the zebra out?).

Five different people:

1. Having different nationalities.
2. Owning houses of different colours.
3. Having different pets.
4. Drinking different drinks.
5. Having a favourite sport.

Here are the known facts:

1. The Englishman lives in the red house.
2. The person living in the yellow house loves cricket.
3. The Ukrainian drinks tea.
4. The Spaniard owns the dog.
5. etc

How many solutions are there (if any)?

N-Queens, graph colouring, constraint satisfaction problems

How expressive is logic? Which problems can be solved by theorem provers?

Classical logic is monotone (not boring!).

2 Modal Logic

2.1 The language of modal logic

2.2 Structures which interpret ML

2.3 Symbolic model checking

2.4 Properties of ML structures

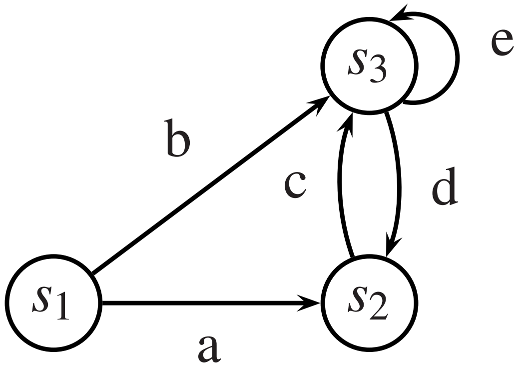
2.5 Hilbert-style deduction systems

2.6 Reduction to classical predicate logic

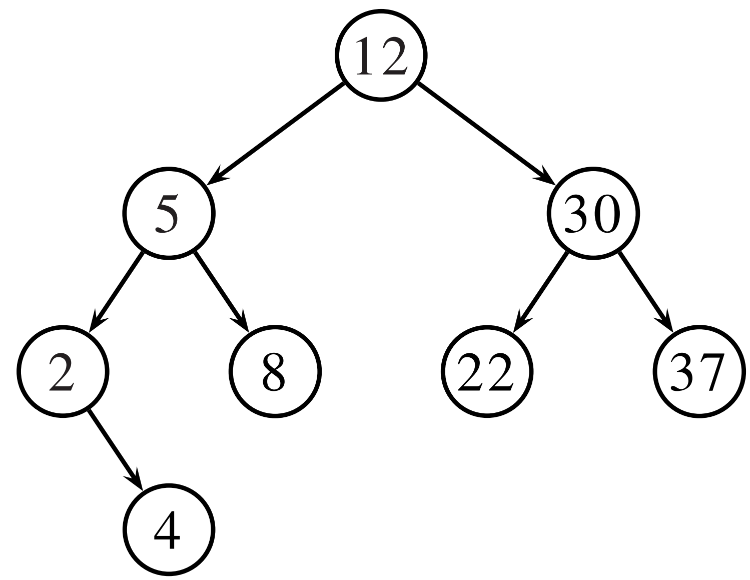
2.7 Modal logic and agents

Modal logic

- has a simple language; its expressiveness is restricted, but nevertheless
- is powerful enough to describe common graphical structures in computer science.



transition systems used to model program executions



trees

- provides a **formal way** of handling notions of necessity, possibility, knowledge, change, time, etc ('modalities')
- is very popular (has been reinvented in some form or other many times)
- are fundamental to disciplines like philosophy, linguistics, computer science

Modal logic is useful for expressing different **modes of truth**

- **Tony Blair is the prime minister of Britain**
  is true now, but will not be always true.
- **The square root of 625 is 25**
  is true (by definition), but it is not known by everyone.
- **This is the best of all possible worlds**
  may or may not be true, but there are people who believe it and others that don't.
- **`z:=y; y:=x; x:=z`**
  after the execution of this sequence the value of x will be the value of y before the execution of this sequence, and vice versa.

Sample ML formulae

$$\mathbf{K}_{\mathrm{Adam}}(\mathrm{prime\_minister}(\mathrm{Tony}, \mathrm{GB}) \wedge$$
$$\bigcirc (\mathrm{prime\_minister}(\mathrm{Tony}, \mathrm{GB}) \vee \neg \mathrm{prime\_minister}(\mathrm{Tony}, \mathrm{GB}))$$

Adam knows Tony is currently the p.m. and after the next election Tony will either be p.m. or not.

$$\mathbf{K}_{\mathrm{Eve}}(\mathrm{prime\_minister}(\mathrm{Tony}, \mathrm{GB}) \wedge$$
$$\mathbf{B}_{\mathrm{Eve}} \bigcirc \neg \mathrm{prime\_minister}(\mathrm{Tony}, \mathrm{GB})$$

Eve knows Tony is currently the p.m. and believes after the next election Tony will not be p.m.

Op.	Name	Meaning
$\mathbf{K}_i$	knowledge operator	agent i knows
$\mathbf{B}_i$	belief operator	agent i believes
$\bigcirc$	next operator	after next election

3 Answer Set Programming

Why not just Prolog? What went wrong with Prolog?

ASP as a new programming paradigm.

ASP is a **nonmonotonic** logic.

I.e.: adding new information makes previously drawn conclusions useless (they have to be given up). It is therefore stronger than classical logic.

Given a (mathematical) problem. How to encode it in the ASP paradigm?

Example: k-colouring, N-Queens, CSP's

$$col\_of(V,C) \leftarrow vertex(V), col(C), \text{not } another\_col(V,C)$$

$$another\_col(V,C) \leftarrow vertex(V), col(C), col(D), D \neq C, col\_of(V,D)$$

$$\leftarrow vertex(U), vertex(V), edge(U,V), col(C), col\_of(V,C), col\_of(U,C)$$

The answer sets of this program correspond to colourings of the underlying graph.

Difference to other programming languages. Algorithm is built in!

Are there efficient ASP solvers out there?

DLV, smodels. How to use them.

You are given problems and have to represent them in the ASP framework. And then solve them using DLV or smodels.

4 Description Logics

4.1 Language, knowledge bases

4.2 Meaning/semantics

4.3 Inferential services

consistency, subsumption, instance checking
classification, querying mechanisms

4.4 Algorithms

4.5 DLs, ontologies and the semantic web

Description logics

- are formalisms for representing and reasoning about knowledge in a given domain of application.
- have similar applications as databases
- are increasingly being used for modelling **ontologies** which play an important role in the intelligent use of the web.

Motivation

A **classical database** stores information in a series of tables which represent relations.

has_son		male
Philip	Charles	Philip
Charles	William	Charles
...	...	William
...	...	...

Query: Is there a grandfather?

Answer: No

Why not?

What is missing is a definition of the concept *grandfather* (a view).

Description logic concept definitions:

$$\mathsf{grandfather} \doteq \mathsf{male} \sqcap \exists \mathsf{has\_son}.\exists \mathsf{has\_son}.\mathsf{human}$$

$$\mathsf{male} \sqsubseteq \mathsf{human}$$

A DL system provides **inferential services** which can fully automatically compute information not explicitly given, e.g. that Philip is a grandfather.

Application areas

- **Formal specification of terminologies** for application domains such as medical expert systems, telecom systems configurations, component knowledge bases in the chemical industry, ...

  New concepts are defined using already defined concepts; relationships are defined from concept definitions

  $\rightsquigarrow$ grandfather; men are human

- **Consistency checks of formal specifications.**
- **Automatic classification of concepts** $\rightsquigarrow$ concept hierarchies

  E.g. botanical classification hierarchies
- More expressive querying of databases
- **Ontologies, semantic web**
- ...

DLs as ontology languages and reasoning tools for the semantic web

- Internet search engines are based on keyword search.

  Disadvantage: the number of answers is huge; many answers are completely irrelevant, while some more interesting answers are not found

- Solution: The **semantic web** aims to facilitate *intelligent* use of the web.

  What is required are machine understandable web resources whose content can be comprehended and processed by automated tools and also human users.

- **Realisation:** Description logics provide content-based representation of information.

  The language of DLs is natural (contrast with classical predicate logic)

  It has a well-defined semantics, since otherwise there cannot be a common understanding of the represented information.

  Inconsistencies can be automatically detected.

  DL systems provide algorithms for automatically extracting implicit information from facts explicitly represented on webpages

Pre-course work

1.1 Elementary set theory What is a set, a relation, a function, set operations (intersection, union, etc), properties of binary relations (reflexivity, symmetry, transitivity, etc).

Reference: Any book on the mathematical foundations of CS or discrete mathematics.

Chapter 5 in (Cormen, Leiserson, and Rivest 1989) (details below).

1.2 Propositional logic very simple representation language; expressively weak; modal logics and description logics are natural extensions.

Reference: (Kelly).

1.3 Constraint satisfaction problems What is a CSP and what is its structure? N-Queens and Schur as instances. Problem representation via logic

Reference: Appropriate chapters in (Russell and Norvig 1995).

1.4 Graph theoretical problems What is k-colourability, the hamiltonian circuit problem, k-cliques and vertex cover?

Reference: any introductory textbook on algorithms, e.g. (Cormen, Leiserson, and Rivest 1989).

References

Baral, C. (2003). *Knowledge representation, reasoning and declarative problem solving with Answer sets*. Cambridge University Press.

Cormen, T. H., C. E. Leiserson, and R. L. Rivest (1989). *Introduction to Algorithms*. McGraw-Hill.

Kelly, J. *The essence of logic*. PHI.

Russell, S. J. and P. Norvig (1995). *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, NJ: Prentice-Hall.

Exercise sheet will be made available.

Reading material

Module does not follow a specific book: copies of the slides are made available. Part III follows partly some chapters of (Baral 2003)

Recommended reading material: listed on the course description webpage. List is not final and may change until the start of the module.

With the exception of books not as yet published, the books on the webpage are available in the Resources Centre or the Departmental Research Library. **No need to buy a book for this module.**

Copies of any additional papers will be made available on the web.

Coursework

Exercises and assignments are of varying difficulty – those in the teaching week are aimed to consolidate the material of the lectures and are thus easier.

Some exercises and assignments are to be done with pencil and paper, some will require the use of tools (SPASS for the ML & DL part, DLV for the ASP part).

For the **post-course work** you will be given a selection of topics from which you choose one.
This work may involve writing a program, formalising problems, using reasoning tools for solving such problems, a case study on some research in one of the areas, or a mixture of these.

Assessment

5% assignments in the pre course week, 25% assignments in the teaching week, 30% post-course work, 40% exam