CS616
Knowledge Representation and Reasoning

Ulrike Sattler

Renate Schmidt

Department of Computer Science
University of Manchester

http://www.cs.man.ac.uk/~schmidt/CS616/

Overview

- Do check the website regularly for announcements
- Teaching week: 30 Jan. – 3 Feb. 2006
- Option for FM and AI specialisations
- Lectures presented in 4 parts:
  - Part I (Sattler, 3) Early KR formalisms, first-order logic
  - Part II (Schmidt, 8) Modal logic
  - Part III (Schmidt, 4) Description logic
  - Part IV (Sattler, 10) Extensions and applications

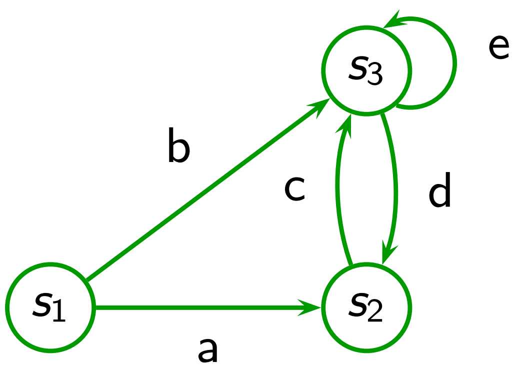
Part I: Early KR formalisms and first-order logic

- Early AI/KR research was very enthusiastic with very high goals
  - early KR formalisms were quite attractive
  - but also came with several problems which are well understood today
- Why first order logic? What is missing in propositional logic?
  - FOL allows to describe different objects and their relationship
  - FOL is “the unifying formalism” of many KR formalisms

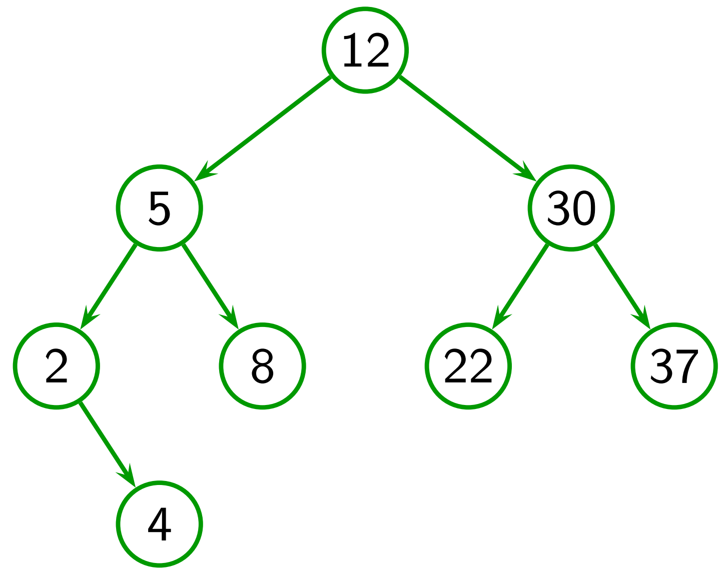
Why modal logic? Why description logic? (1)

First-order logic is a very expressive language, can capture wide range of knowledge $\rightsquigarrow$ Why ML? Why DL?

- ML and DL are expressively weaker than FOL.
- ML and DL are simpler, more natural languages, yet powerful enough to describe useful structures:



transition systems used to model program executions



trees

Why modal logic? Why description logic? (2)

- ML and DL are very popular in CS and AI, have been "reinvented" many times.
- There are many applications.
- Modal and description logics have nice computational properties.
  - Reasoning in first-order logic is undecidable
    many MLs and DLs are decidable
    some MLs and DLs are undecidable

    We will mostly study decidable logics.
  - The MLs and DLs we study have nice computational complexity.

Part II: Modal logic

- The language of modal logic & structures which interpret ML
- Symbolic model checking $\rightsquigarrow$ querying of graphs
- Styles of reasoning
  - Hilbert-style deduction systems
  - Reduction to first-order logic, using SPASS
- Application:
  - agents-based systems;
    formalising the agents' beliefs and knowledge

Purpose of modal logics

- Modal logics are a *formal way* of handling notions of knowledge, belief, time, actions, necessity, possibility, etc ('modalities')
- Modal logics allows us to model different modes of truths:
  - Tony Blair is the prime minister of Britain
    is true now, but will not be true forever.
  - 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 who don't.

Sample specifications

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

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

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

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

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

Part III: Description logics

- Language & meaning
- Inferential services: consistency, subsumption, instance checking, classification, querying mechanisms
- Algorithms to solve these types of problems
- Application
  - DLs as ontology languages and the semantic web

Purpose of description logics

- Description logics are about
  - modelling world knowledge, i.e. ‘objective knowledge’ of a particular domain of application
  - and reasoning about it
- DL systems have similar applications as databases but are more flexible and more expressive
- DLs systems are used for modelling ontologies; important for semantic web

Motivating example

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

Query: Is there a grandfather?

Answer: No

has_child		male
Phillip	Charles	Phillip
Charles	William	Charles
...	...	William
...	...	...

Why not?

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

Suitable concept definitions in description logic would be:

$$\text{grandfather} \doteq \text{male} \sqcap \exists \text{has\_child}.\exists \text{has\_child}.\text{human}$$

$$\text{male} \ \dot{\sqsubseteq} \ \text{human}$$

Services of description logic systems

- DLs allow the description of both concrete (database) and abstract information (concept definitions).
- Sample inferential services:
  - consistency: KB consistent? grandfather consistent?
  - subsumption: grandfather subsumed by human?
  - instance checking: Charles an instance of ∃has_child.human?
  - querying KB
- In contrast to databases, DL systems can handle incomplete information.

Part IV: Extensions and applications

- **ICOM:** a tool for intelligent conceptual modeling
  built to design and reason about ER/UML schemas, based on DLs
- **Non-standard reasoning services:**
  - applying DLs requires more than classical logical reasoning (validity, satisfiability, etc)
  - to support domain experts which are not DL experts,
  - e.g. to add new concepts into a knowledge base
  - Example NSRS: approximating concepts
    computing the least common subsumer of some concepts
    computing the most specific concept for an individual, etc.

Part IV: Extensions and applications

- **Temporal DLs:**
  - so far, DLs were static
  - to express knowledge about changes, actions, processes, etc., requires a notion of time, e.g.,
  - `CS_Student` **implies** **eventually** (`Rich` **or** `Famous`)
- **Defaults:**
  - so far, we only have strict axioms `Bird` **implies** `CanFly`
  - some applications want default axioms `Bird` **implies_by_default** `CanFly` (because of Penguins, broken wings, oil desasters, etc.)
  - how to extend FOL or DLs with such "defaults"

Pre-course work

- **Elementary set theory**

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

- **Propositional logic** (Boolean logic)

  Very simple representation language; expressively weak; modal logics and description logics are natural extensions.

- **First-order logic**

  First order logic formulae, their meaning, validity and satisfiability, translating between natural language and first-order logic.

Exercise sheet will be made available.

Reading material

- Course unit does not follow a specific book: copies of the slides are made available.
- Recommended reading material: listed on the course description webpage. List is not final and may change until the start of the module.
- The books on the webpage are available in the Resources Centre. No need to buy a book for this course unit.
- 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, ICOM for the DL application 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