CITS3005 Knowledge Representation

Lecture 1: Knowledge and Information

Tim French

The University of Western Australia

2023

Welcome!

Acknowledgement to Country

The University of Western Australia acknowledges that it is situated on Noongar land, that the Noongar people remain the spiritual and cultural custodians of their land and continue to practise their values, languages, beliefs and knowledge. We pay respect to their Elders, past present and emerging.

Welcome to CITS3005 Knowledge Representation, which covers some old ideas in a new context.
It will explore:

- the nature of *knowledge*, both metaphysical and computational;
- the elements of reason;
- how knowledge is organised, accessed and acquired;
- the limits of knowledge in terms of both uncertainty and computability; and
- how we might pragmatically design machines capable of reasoning.

Timetable and Contact

Contact

Unit Coordinator	Tim French
Office	Computer Science Building, Rm 2.14
Email	tim.french@uwa.edu.au
Phone	6488 2794
Consultation hours	Thursday 12-1

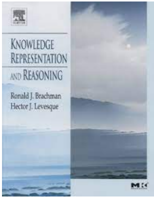
Timetable

Lecture	Wednesday 11-1	Webb LT
Workshop 1	Thursday 11-12	CSSE 2.07
Workshop 2	Thursday 1-2	Robert Street LT

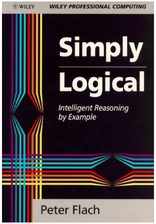
Lectures will be recorded and streamed via Microsoft Teams. Workshops begin in week 2.

Resources

Material will be hosted on the UWA Learning Management System (LMS), and the unit web page `https://teaching.csse.uwa.edu.au/units/CITS3005/`
The lectures and workshop exercises, will (normally) be made available at the start of the week in which they are taught.
There is no prescribed text for the unit, although the following books will be useful and are used to source some of the material in the unit.



Knowledge Representation and Reasoning, Ronald Brachman and Hector Levesque, Elsevier, 2004.


WILEY
WILEY PROFESSIONAL COMPUTING
Simply Logical
Intelligent Reasoning by Example
Peter Flach

Simply Logical, Peter Flach, Wiley 1994 `https://book.simply-logical.space/src/simply-logical.html`.

Additional resources will be made throughout semester.

Assessment

The assessment in this unit will involve regular homework assignments (40%) and an end of semester exam (60%).

Assessment	Topic	Due	Percent
HW1	Logic and Logic Programming	5pm August 18	10%
HW2	Knowledge Graphs	5pm September 22	10%
Project	Knowledge Systems	5pm October 20	40%
Exam	Everything	Exam Period	40%

The homework will consist of written discussions, programming exercises, and problem solving exercises.

In addition to these assessments there will be weekly workshop exercises, that you are strongly encouraged to complete, and will be discussed in the workshops, and have a similar format to the homework exercises.

Homework and projects should be submitted via cssubmit, and marks will be available via csmarks.

What is Knowledge Representation?

Knowledge Representation is the study of how information is processed contextualised, and synthesised into knowledge, to support decision making applications. Knowledge is more than just a collection of facts: it is a structure *built from symbols*, the *represents some condition* of the world, with *transformation rules* describing dependencies and entailment.

There are many different approaches to these tasks, but at the core of all of them is *logic*. A Logic given some language built from symbols, is the subset of "true" elements of the language.

Logic, reasoning and thought have been studied for millennia, and pose many difficult philosophical questions, which we will generally avoid in this course.

Our challenge is to consider what fragment of knowledge and reasoning can we capture and implement on a machine.

What will we study?

This course will focus on the practical aspects of implementing knowledge and reasoning systems in software.

We presume that all students have generally good programming skills, and are familiar with propositional and first order logic.

The topics we will aim to cover are:

1. **Knowledge, belief and information**: *Week 1*; what is knowledge, where does it come from, how do we use it.
2. **Logic**: *Weeks 2-4*; syntax and semantics of first order logic, proofs and entailment, expressing conditions.
3. **Logic Programming**: *Weeks 2-4*; syntax of `prolog`, resolution, unification, completeness
4. **Knowledge Graphs**: *Weeks 5-8*; description logic, `RDF`, OWL, SHACL, reasoning.
5. **Knowledge Systems**: *Weeks 9-12*; uncertainty, fuzzy logic, `problog`, learning, knowledge extraction, planning, search.

Overview

This course considers the fundamentals of artificial intelligence as a *knowledge based system*: an intelligence must be able to *acquire*, *interpret* and *value* knowledge.

What does it mean to know something?

- Is knowledge just a collection of facts?
- How do we acquire new knowledge?
- Can our knowledge be wrong, and what do we do if it is?

Here we take a computational approach to knowledge. Knowledge must be:

- distilled from some data source;
- stored in some format;
- associated with meaning as a description of something external;
- synthesised, assessed and refined;
- and ultimately queried to provide insight and value.

Data, Information, Knowledge and Reason

Consider the grade book for a unit, and whether a student has a passing grade.

Data:

```
0000000 6469 202c 616e 656d 202c 5748 2c31 4820
0000010 3257 202c 7250 6a6f 6365 2c74 4520 6178
0000020 0a6d 3931 3136 3837 3031 202c 6954 2c6d
0000030 3520 202c 2c36 3520 2c38 3720 0a32 3931
0000040 3939 3737 3131 202c 6c41 7865 202c 2c39
0000050 3320 202c 3839 202c 3137 320a 3132 3031
0000060 3433 2c32 4a20 6e61 2c65 3220 202c 3031
0000070 202c 3038 202c 3038 000a 0000079
```

‘

Information:

```
id, name, HW1, HW2, Project, Exam 19617810, Tim, 5,
6, 58, 72 19997711, Alex, 9, 3, 98, 71 22110342,
Jane, 2, 10, 80, 80
```

‘

Knowledge:

```
# Class definitions :Student a owl:Class ;
rdfs:label "Student" ;
:Assessment a owl:Class ; rdfs:label "Assessment" ;
:Attempt a owl:Class ; rdfs:label "Attempt" ;
# Data property definition :hasMark a
owl:DatatypeProperty ; rdfs:label "has mark" ;
rdfs:domain :Attempt ; rdfs:range xsd:integer ;
# Object property definitions :attemptedBy a
owl:ObjectProperty ; rdfs:label "attempted by" ;
rdfs:domain :Attempt ; rdfs:range :Student ;
:belongsToAssessment a owl:ObjectProperty ;
rdfs:label "belongs to assessment" ; rdfs:domain
:Attempt ; rdfs:range :Assessment ;
```

‘

Reason:

```
w[1]*a[1][tim] + w[2]*a[2][tim] < 50 implies
grade[tim] = ’N’
```

‘

What is *Knowledge*? What is *Intelligence*?

- Is knowledge just a collection of facts, like an encyclopedia?
- Is knowledge experience, or memory?
- Is knowledge correlations and causes between events?
- Is intelligence the ability to predict unseen variables?
- Is intelligence the ability to abstract and generalise information?
- Is intelligence the ability to solve problems, and determine consequences?
- Is intelligence the ability to rationalise and defend a position?
- Must knowledge be about something real?
- Must intelligence be organic?
- Must intelligence be creative?

Is *Information* knowledge?

We will consider *Data* to simply be a set or sequence of values, whereas *Information* is the understanding of Data in context.

Information Theory (Shannon, Hartley and Nyquist) provides a mathematical theory of information, where *entropy* quantifies the amount of information in a signal:

$$H = -\sum_{i=1}^{n} p_i \lg p_i$$

where p_i is the prior probability of of the signal taking the value i.

Information theory provides a range of tools for measuring information, and determining how “surprising” an event is.

However, information is only useful if the agent knows what the values *mean*.

Are *Data Structures* knowledge?

Data structures and algorithms transform information, for example taking a map and producing a shortest path.

These are foundational in artificial intelligence and local search, linear programming and similar techniques are key technologies in artificial intelligence.

However, the *knowledge* is retained by the designer and is not explicit in the algorithms or data structures themselves.

Similarly, databases, JSON, key-stores, represent information, but require schema to interpret that information, and to relate it to some external context.

Is *Belief* knowledge?

Knowledge and belief are often treated as interchangeable, or sometimes knowledge is treated as “true belief”. Belief tends to have a wide range of possible meanings.

When we say “belief” we could mean

- a *weak form of knowledge* (“I believe I will pass, but I’m not sure”)
- *incontestable faith* (“I am certain God exists”)
- *falsehoods with commitment* (“I definitely won the election”)

In the first case we are considering *uncertainty* or *probability*, while the latter two would typically not be thought of as knowledge.

This raises the questions of whether *true knowledge* is ever really attainable, and whether there is any pragmatic difference between belief and knowledge.

$Knowledge = Belief + Evidence?$

Logic, Language and Information

This leads us to a characterisation of knowledge as *logic*, *language* and *information*.

Logic

The known or believed propositions and rules of a system. The notion of consequence and the process of inference. The algorithms and functions that may be applied to propositions.

Language

The symbols upon which the logic acts, and the way we query and transfer knowledge.

Information

The observations and experience of the external system. The ground truth as it is given to us.

Representations

The concept of a representation has three parts:

- The *actual domain* that is being represented, the consequences and causes and effects, and the information that is transferred from the domain as observations;

  This lecture room, the people, the seats, ...

- The *symbolic domain* as a language describing an abstraction of the actual domain, with rules and logic approximating the actual domain;

  seat(x), free(x), nearest(x,y), ...

- and a correspondence between the two; the *semantics*, "what does it all mean".

  The nearest free seat to the door.

Representations consider time, space, agents as real world domain entities, that must be abstracted to apply reasoning upon them.

Reasoning

Reasoning is the process of inferring one proposition from another.
It can be inductive (based on past experience) or deductive (based on commonsense rules and axioms).
Logical rules of inference can be applied to propositions:

‘

‘I *need* to pass one more unit to *complete* my degree”
Therefore
“If I pass CITS3005, I will finish my degree”

...but it is easy to get wrong:

‘

‘I *need* to pass one more unit to *complete* my degree”
Therefore
“I need to pass CITS3005 to finish my degree”

Knowledge based systems

Consider the following two prolog programs from Brachman and Levesque:

```
pCol(snow) :- !, write("white").
pCol(grass) :- !, write("green").
pCol(sky) :- !, write("blue").
pCol(X) :- !, write("idk").
```

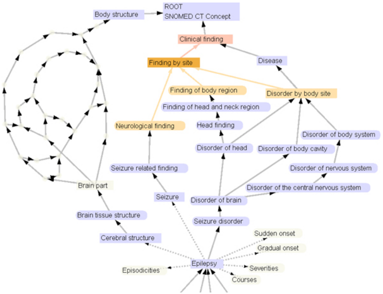
```
pCol(X) :- col(X,Y), !, write(Y).
pCol(X) :- !, write("idk").
col(snow, white).
col(sky, blue).
col(X,Y) :- made(X,Z),col(Z,y).
made(grass, vegetation).
col(vegetation,green).
```

Which is the better representation of knowledge?

KR in industry

Knowledge representation is a critical step in automation and robotics. Consider the processes required to run driverless trucks in a mine site, or an unmanned oil rig.

Knowledge representation is also very significant in medicine, where diagnosis is implicitly a knowledge based task.



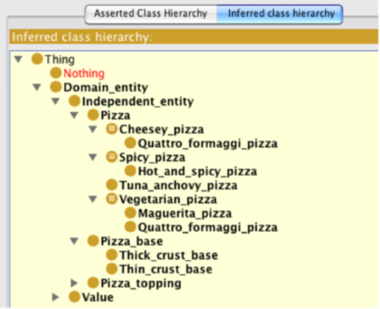
Planning and Robotics

Contextual knowledge is essential for autonomous machines. Driverless cars must know and understand the road rules, and also when they need to break them?

This knowledge can be coded in platforms like *STRIPS*, *PDDL*, or *Prolog*, and *search*, *answer set programming*, or *reinforcement learning* can be applied to find best actions.

Semantic Web

The web is the ultimate democratic knowledge source, and there is a long standing goal to inject context and meaning into web documents to create a *semantic web*



Inferred class hierarchy:

- Thing
  - Nothing
  - Domain_entity
    - Independent_entity
      - Pizza
        - Cheesey_pizza
          - Quattro_formaggi_pizza
        - Spicy_pizza
          - Hot_and_spicy_pizza
        - Tuna_anchovy_pizza
        - Vegetarian_pizza
          - Maguerita_pizza
          - Quattro_formaggi_pizza
      - Pizza_base
        - Thick_crust_base
        - Thin_crust_base
      - Pizza_topping
    - Value

This context and meaning can be captured at various levels using *Description Logics*, *OWL*, *SHACL*, and *RDF*, using tools such as *OWLReady*, *Neo4J*, or *Protegè*.

Intelligent Assistants

Intelligent Assistants like Chat-GPT or Bard have recently emerged as powerful tools, with surprisingly intelligent behaviour.

It is still unclear whether these tools have an emergent internal knowledge representation, or are simply minimising entropy over a massive corpus, (or indeed if there is any difference).

Uncertainty, *hallucination*, *causation* have always been challenges for artificial intelligence. The increased responsibilities given to AI assistants makes these issues increasingly urgent.

Chat GPT and Knowledge Representation

Convert the following to first order predicate logic using the predicates SunnyDay(X,Y), Walking(X,Y) and constants tuesday, and me: "Every day that is sunny, I go walking. I did not go walking on Tuesday, so it must not have been sunny"

ChatGPT:

Let’s break down the given statements and convert them into first-order predicate logic using the provided predicates and constants:

"Every day that is sunny, I go walking."

Forall x (SunnyDay(x) Implies Walking(me, x))

This statement is a universal quantification that states that for every day x, if it is sunny (SunnyDay(x)), then I go walking (Walking(me, x)).

"I did not go walking on Tuesday."

Not Walking(me, tuesday)

This statement is a negation that states that I did not go walking (Walking(me, tuesday)) on Tuesday (constant tuesday).

"So, it must not have been sunny."

Not SunnyDay(tuesday)

This statement is a negation that states that it must not have been sunny (SunnyDay(tuesday)) on Tuesday (constant tuesday).

To summarise, the converted statements in first-order predicate logic are as follows:

Forall x (SunnyDay(x) Implies Walking(me, x))

Not Walking(me, tuesday)

Not SunnyDay(tuesday)

‘

Chat GPT and Reasoning

excellent work. Can you prove the deduction is valid?

ChatGPT:

Certainly! To prove the validity of the deduction, we can use natural deduction rules. I'll use a proof by contradiction to demonstrate that the conclusion is a logical consequence of the premises. Here's the proof:

Assume Not SunnyDay(tuesday) [Assumption]
Assume Walking(me, tuesday) [Assumption]
From premises 1 and 2: SunnyDay(tuesday) Implies Walking(me, tuesday) [Universal instantiation of premise 1]
From premise 2 and the assumption Walking(me, tuesday): Walking(me, tuesday) [Repetition]
From premises 4 and 3: SunnyDay(tuesday) [Modus ponens]
From premises 5 and 1: False (Contradiction)
From the contradiction in step 6: Not Walking(me, tuesday) [Negation introduction]
From premise 2 and step 7: False (Contradiction)
From the contradiction in step 8: Not Not SunnyDay(tuesday) [Negation introduction]
From step 9: SunnyDay(tuesday) [Double negation elimination]
From premises 10 and 1: Walking(me, tuesday) [Modus ponens]
From premise 2 and step 11: False (Contradiction)
From the contradiction in step 12: Not Walking(me, tuesday) [Negation introduction]
From steps 2 and 13: Walking(me, tuesday) And Not Walking(me, tuesday) [Conjunction introduction]

Next lecture ....

Logic!

For the workshop next week, we will be discussing the capabilities and limitations of AI assistants.

Choose a task (Maths, puzzles, creative writing, empathy, debugging ...) and an assistant (GPT, Bard, Wolfram Alpha) and investigate the capabilities and limitations of the assistant in doing the task.

Be prepared to share your findings!