

# Lecture 1

## Introducing Symbolic AI

COMP24412: Symbolic AI

Giles Reger and Andre Freitas

February 2019

# This Lecture

1. A bit about what we mean by Symbolic AI
2. Course mechanics
3. A brief introduction to the first part of the course (Giles)
4. A brief introduction to the second part of the course (Andre)

# Some definitions of Artificial Intelligence

<b>Thinking Humanly</b> “The exciting new effort to make computers think... <i>machines with minds</i> , in the full and literal sense.”	<b>Thinking Rationally</b> “The study of mental faculties through the use of computational models.”
<b>Acting Humanly</b> “The study of how to make computers do things at which, at the moment, people are better.”	<b>Acting Rationally</b> “AI...is concerned with the intelligent behaviour of artefacts.”

*Artificial Intelligence: A Modern Approach. Third Edition.*

# Some definitions of Artificial Intelligence

<b>Thinking Humanly</b> “The exciting new effort to make computers think... <i>machines with minds</i> , in the full and literal sense.”	<b>Thinking Rationally</b> “The study of mental faculties through the use of computational models.”
<b>Acting Humanly</b> “The study of how to make computers do things at which, at the moment, people are better.”	<b>Acting Rationally</b> “AI...is concerned with the intelligent behaviour of artefacts.”

*Artificial Intelligence: A Modern Approach. Third Edition.*

# Acting Humanly

The main AI disciplines required to pass the so-called *total Turing Test*.

1. Natural Language Processing
2. Knowledge Representation
3. Automated Reasoning
4. Machine Learning
5. Computer Vision
6. Robotics

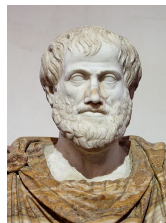
But is replicating humans the right approach? Do we aim to copy birds with artificial flight?

# Thinking Humanly



# Thinking Rationally

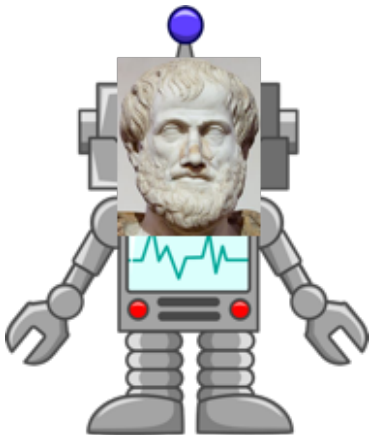
All men are mortal.  
Socrates is a man.  
Therefore, Socrates is mortal.



Everything related to logic is fun.  
Symbolic AI is related to logic.  
Therefore, Symbolic AI is fun.



# Acting Rationally





# What is Symbolic AI?

## Symbolic Artificial Intelligence

# What is Symbolic AI?

Symbolic **Artificial** Intelligence

Not occurring naturally, produced by humans

# What is Symbolic AI?

## Symbolic Artificial Intelligence

Not occurring naturally, produced by humans

The *acquisition* and *application* of knowledge

# What is Symbolic AI?

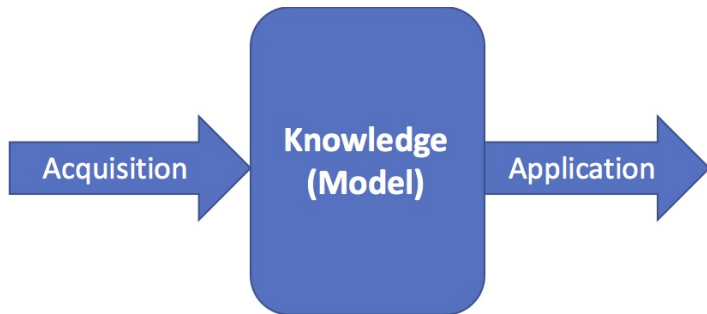
## Symbolic Artificial Intelligence

Not occurring naturally, produced by humans

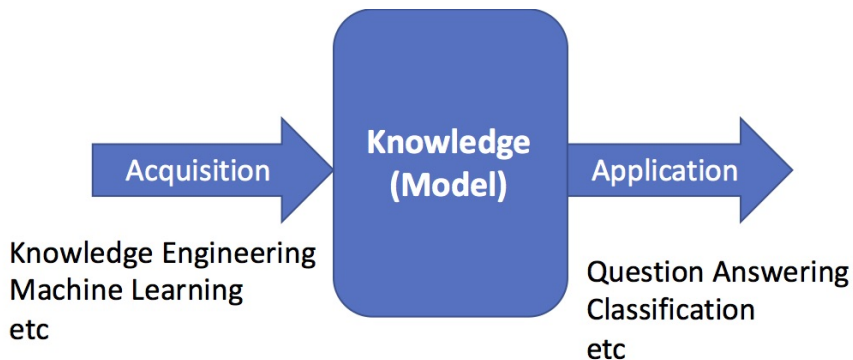
The *acquisition* and *application* of *knowledge*

Using symbols (rather than statistics) to represent the world

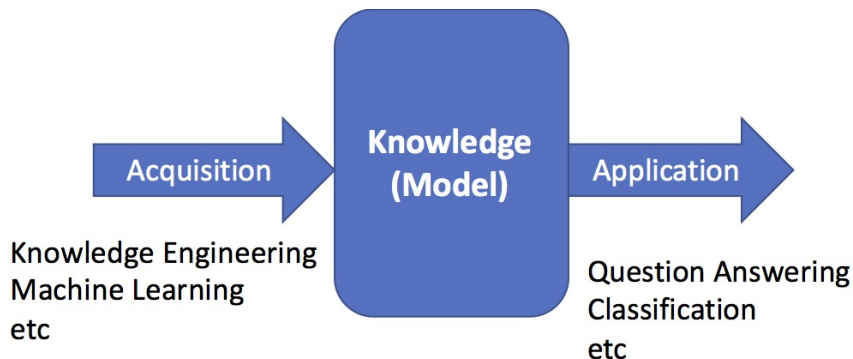
# It's All in the Model



# It's All in the Model

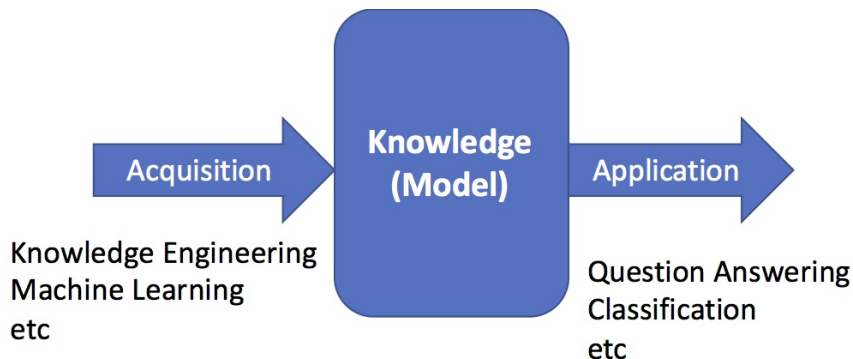


# It's All in the Model



Do we represent our knowledge/model the world using symbols or statistics... or a combination of the two?

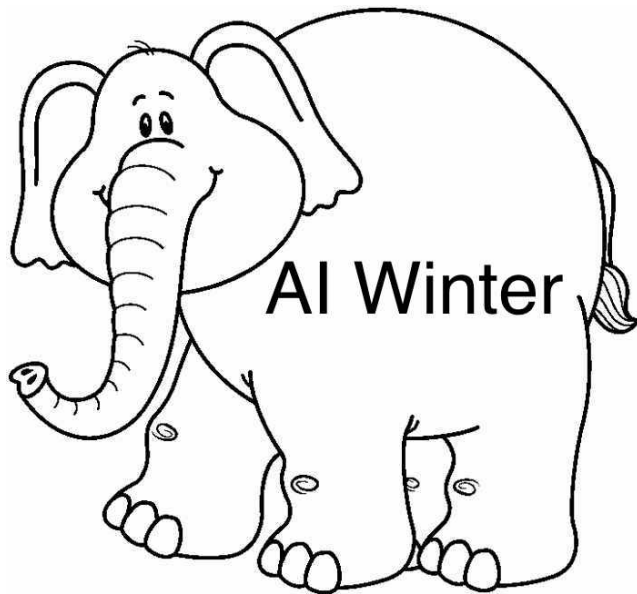
# It's All in the Model



Do we represent our knowledge/model the world using symbols or statistics... or a combination of the two? All about *relations* between stuff.



# Addressing the Elephant



The symbolic approach was overhyped at first

Lack of success led to negative view of approach and a slump in research

Now acknowledged that symbolic and statistical approaches are complementary

# Connecting to Other Courses

## COMP14112: Fundamentals of Artificial Intelligence

Probabilistic models, Robot Localisation, Speech Understanding

## COMP11120: Mathematical Techniques for Computer Science

Introduction to logic and reasoning

## COMP21111: Logic and Modelling

Propositional models and reasoning

## COMP24111: Machine Learning and Optimisation

Statistical Models of the world, Classification and Clustering

## COMP34412 Natural Language Systems

Statistical Models of language, How to build and use those models

The aim of this course is:

To provide the conceptual and practical (systems building)  
foundations for knowledge representation and reasoning in  
Artificial Intelligence

You should learn some fundamental ideas and how to apply those ideas to  
do some useful things.

# Learning Outcomes

By the end of this course you will be able to:

- 1 Describe the syntax and semantics of first-order logic and use it to model problems
- 2 Apply reasoning techniques (transformation to clausal form, resolution, saturation) to establish properties of first-order problems
- 3 Explain the theoretical limitations of automated theorem provers
- 4 Write Prolog programs to solve automated reasoning tasks and explain how they will execute
- 5 Describe, differentiate and apply different knowledge representation formalisms for modelling knowledge bases.
- 6 Explain how these formalisms affect the reasoning process.
- 7 Apply, demonstrate and program knowledge-based learning methods.
- 8 Apply, demonstrate and program formal models for natural language processing in the context of semantic parsing and natural logic inference.

# Course Structure

## Part 1 (weeks 1-6) Automated Reasoning (Giles and Martin)

- The basics of representation and reasoning in Datalog
- Prolog as a programming language
- Reasoning in first-order logic
- The theoretical limits of automated reasoning

## Part 2 (weeks 6-12) Applications of Knowledge (Andre)

- More on Knowledge Representation
- Knowledge in Learning
- Inductive Logic Programming
- Symbolic Natural Language Processing
- Symbolic and Non-Symbolic Models

## Labs (25%)

- 3 exercises (1 small, 2 big)
- 5 lab sessions
- Hands-on experience with tools

## Quizzes (5%)

- 5 Blackboard quizzes (every fortnight)
- Each worth 1%
- Purpose is to give useful feedback and preparation for exam

## Exam (70%)

- Hybrid (half on Blackboard, half on paper)
- split between two parts

Giles Reger

Room 2.62

`giles.reger@manchester.ac.uk`



Andre Freitas

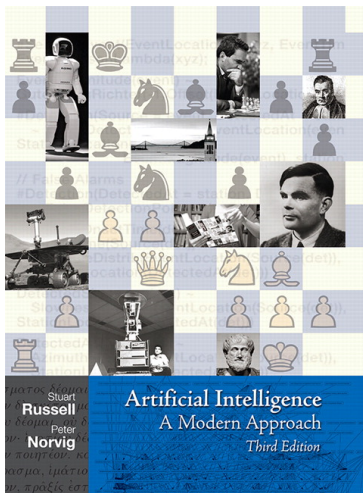
`andre.freitas@manchester.ac.uk`



Martin Riener



# Suggested Reading





# A brief Introduction to Part 1

Should you take COMP24412?...

Let us represent the knowledge we have about the course and then see what follows (logically).

# Representing Some Knowledge

## The Facts

COMP24412 teaches Logic

COMP24412 is about AI

Prolog is a programming language

COMP24412 teaches Prolog

AI is cool

Yachts cost lots of money

# Representing Some Knowledge

## The Facts

COMP24412 teaches Logic

COMP24412 is about AI

Prolog is a programming language

COMP24412 teaches Prolog

AI is cool

Yachts cost lots of money

## The Rules

If you take a course and it teaches X then you know X

If you take a course about X and X is cool then you are cool

If you know a programming language then you can program

If you can program and know logic you can get a good job

If you have a good job you get lots of money

If you have X and Y costs X then you can have Y

# Representing Some Knowledge

## The Facts

teaches(COMP24412, Logic)

COMP24412 is about AI

Prolog is a programming language

COMP24412 teaches Prolog  
AI is cool

Yachts cost lots of money

## The Rules

If you take a course and it teaches X then you know X  
If you take a course about X and X is cool then you are cool  
If you know a programming language then you can program  
If you can program and know logic you can get a good job  
If you have a good job you get lots of money  
If you have X and Y costs X then you can have Y

# Representing Some Knowledge

## The Facts

teaches(COMP24412, Logic)	teaches(COMP24412, Prolog)
about(COMP24412, AI)	cool(AI)
language(Prolog)	costs(Yacht, LotsOfMoney)

## The Rules

If you take a course and it teaches X then you know X  
If you take a course about X and X is cool then you are cool  
If you know a programming language then you can program  
If you can program and know logic you can get a good job  
If you have a good job you get lots of money  
If you have X and Y costs X then you can have Y

# Representing Some Knowledge

## The Facts

teaches(COMP24412, Logic)	teaches(COMP24412, Prolog)
about(COMP24412, AI)	cool(AI)
language(Prolog)	costs(Yacht, LotsOfMoney)

## The Rules

$\text{take}(U, C) \wedge \text{teaches}(C, X) \Rightarrow \text{know}(U, X)$

If you take a course about X and X is cool then you are cool

If you know a programming language then you can program

If you can program and know logic you can get a good job

If you have a good job you get lots of money

If you have X and Y costs X then you can have Y

# Representing Some Knowledge

## The Facts

teaches(COMP24412, Logic)	teaches(COMP24412, Prolog)
about(COMP24412, AI)	cool(AI)
language(Prolog)	costs(Yacht, LotsOfMoney)

## The Rules

$\text{take}(U, C) \wedge \text{teaches}(C, X) \Rightarrow \text{know}(U, X)$   
 $\text{take}(U, C) \wedge \text{about}(C, X) \wedge \text{cool}(X) \Rightarrow \text{cool}(U)$   
 $\text{know}(U, X) \wedge \text{language}(X) \Rightarrow \text{canProgram}(U)$   
 $\text{canProgram}(U) \wedge \text{know}(U, \text{Logic}) \Rightarrow \text{hasGoodJob}(U)$   
 $\text{hasGoodJob}(U) \Rightarrow \text{has}(U, \text{LotsOfMoney})$   
 $\text{has}(U, X) \wedge \text{costs}(Y, X) \Rightarrow \text{has}(U, Y)$

# Inferring Some More Knowledge

teaches(COMP24412, Logic)	teaches(COMP24412, Prolog)
about(COMP24412, AI)	cool(AI)
language(Prolog)	costs(Yacht, LotsOfMoney)

$\text{take}(U, C) \wedge \text{teaches}(C, X) \Rightarrow \text{know}(U, X)$   
 $\text{take}(U, C) \wedge \text{about}(C, X) \wedge \text{cool}(X) \Rightarrow \text{cool}(U)$   
 $\text{know}(U, X) \wedge \text{language}(X) \Rightarrow \text{canProgram}(U)$   
 $\text{canProgram}(U) \wedge \text{know}(U, \text{Logic}) \Rightarrow \text{hasGoodJob}(U)$   
 $\text{hasGoodJob}(U) \Rightarrow \text{has}(U, \text{LotsOfMoney})$   
 $\text{has}(U, X) \wedge \text{costs}(Y, X) \Rightarrow \text{has}(U, Y)$



# Inferring Some More Knowledge

teaches(COMP24412, Logic)   teaches(COMP24412, Prolog)  
about(COMP24412, AI)   cool(AI)  
language(Prolog)   costs(Yacht, LotsOfMoney)

$\text{take}(U, C) \wedge \text{teaches}(C, X) \Rightarrow \text{know}(U, X)$   
 $\text{take}(U, C) \wedge \text{about}(C, X) \wedge \text{cool}(X) \Rightarrow \text{cool}(U)$   
 $\text{know}(U, X) \wedge \text{language}(X) \Rightarrow \text{canProgram}(U)$   
 $\text{canProgram}(U) \wedge \text{know}(U, \text{Logic}) \Rightarrow \text{hasGoodJob}(U)$   
 $\text{hasGoodJob}(U) \Rightarrow \text{has}(U, \text{LotsOfMoney})$   
 $\text{has}(U, X) \wedge \text{costs}(Y, X) \Rightarrow \text{has}(U, Y)$

take(you, COMP24412)

# Inferring Some More Knowledge

teaches(COMP24412, Logic)   teaches(COMP24412, Prolog)  
about(COMP24412, AI)   cool(AI)  
language(Prolog)   costs(Yacht, LotsOfMoney)

$\text{take}(U, C) \wedge \text{teaches}(C, X) \Rightarrow \text{know}(U, X)$   
 $\text{take}(U, C) \wedge \text{about}(C, X) \wedge \text{cool}(X) \Rightarrow \text{cool}(U)$   
 $\text{know}(U, X) \wedge \text{language}(X) \Rightarrow \text{canProgram}(U)$   
 $\text{canProgram}(U) \wedge \text{know}(U, \text{Logic}) \Rightarrow \text{hasGoodJob}(U)$   
 $\text{hasGoodJob}(U) \Rightarrow \text{has}(U, \text{LotsOfMoney})$   
 $\text{has}(U, X) \wedge \text{costs}(Y, X) \Rightarrow \text{has}(U, Y)$

take(you, COMP24412)  
know(you, Logic)   know(you, Prolog)

# Inferring Some More Knowledge

teaches(COMP24412, Logic)   teaches(COMP24412, Prolog)  
  about(COMP24412, AI)            cool(AI)  
  language(Prolog)            costs(Yacht, LotsOfMoney)

$\text{take}(U, C) \wedge \text{teaches}(C, X) \Rightarrow \text{know}(U, X)$   
 $\text{take}(U, C) \wedge \text{about}(C, X) \wedge \text{cool}(X) \Rightarrow \text{cool}(U)$   
 $\text{know}(U, X) \wedge \text{language}(X) \Rightarrow \text{canProgram}(U)$   
 $\text{canProgram}(U) \wedge \text{know}(U, \text{Logic}) \Rightarrow \text{hasGoodJob}(U)$   
 $\text{hasGoodJob}(U) \Rightarrow \text{has}(U, \text{LotsOfMoney})$   
 $\text{has}(U, X) \wedge \text{costs}(Y, X) \Rightarrow \text{has}(U, Y)$

                                 take(you, COMP24412)  
know(you, Logic)   know(you, Prolog)   cool(you)

# Inferring Some More Knowledge

teaches(COMP24412, Logic)   teaches(COMP24412, Prolog)  
about(COMP24412, AI)   cool(AI)  
language(Prolog)   costs(Yacht, LotsOfMoney)

$\text{take}(U, C) \wedge \text{teaches}(C, X) \Rightarrow \text{know}(U, X)$   
 $\text{take}(U, C) \wedge \text{about}(C, X) \wedge \text{cool}(X) \Rightarrow \text{cool}(U)$   
 $\text{know}(U, X) \wedge \text{language}(X) \Rightarrow \text{canProgram}(U)$   
 $\text{canProgram}(U) \wedge \text{know}(U, \text{Logic}) \Rightarrow \text{hasGoodJob}(U)$   
 $\text{hasGoodJob}(U) \Rightarrow \text{has}(U, \text{LotsOfMoney})$   
 $\text{has}(U, X) \wedge \text{costs}(Y, X) \Rightarrow \text{has}(U, Y)$

take(you, COMP24412)  
know(you, Logic)   know(you, Prolog)   cool(you)   canProgram(you)

# Inferring Some More Knowledge

teaches(COMP24412, Logic)   teaches(COMP24412, Prolog)  
about(COMP24412, AI)   cool(AI)  
language(Prolog)   costs(Yacht, LotsOfMoney)

$\text{take}(U, C) \wedge \text{teaches}(C, X) \Rightarrow \text{know}(U, X)$   
 $\text{take}(U, C) \wedge \text{about}(C, X) \wedge \text{cool}(X) \Rightarrow \text{cool}(U)$   
 $\text{know}(U, X) \wedge \text{language}(X) \Rightarrow \text{canProgram}(U)$   
 $\text{canProgram}(U) \wedge \text{know}(U, \text{Logic}) \Rightarrow \text{hasGoodJob}(U)$   
 $\text{hasGoodJob}(U) \Rightarrow \text{has}(U, \text{LotsOfMoney})$   
 $\text{has}(U, X) \wedge \text{costs}(Y, X) \Rightarrow \text{has}(U, Y)$

take(you, COMP24412)  
know(you, Logic)   know(you, Prolog)   cool(you)   canProgram(you)  
hasGoodJob(you)

# Inferring Some More Knowledge

teaches(COMP24412, Logic)   teaches(COMP24412, Prolog)  
about(COMP24412, AI)   cool(AI)  
language(Prolog)   costs(Yacht, LotsOfMoney)

$\text{take}(U, C) \wedge \text{teaches}(C, X) \Rightarrow \text{know}(U, X)$   
 $\text{take}(U, C) \wedge \text{about}(C, X) \wedge \text{cool}(X) \Rightarrow \text{cool}(U)$   
 $\text{know}(U, X) \wedge \text{language}(X) \Rightarrow \text{canProgram}(U)$   
 $\text{canProgram}(U) \wedge \text{know}(U, \text{Logic}) \Rightarrow \text{hasGoodJob}(U)$   
 $\text{hasGoodJob}(U) \Rightarrow \text{has}(U, \text{LotsOfMoney})$   
 $\text{has}(U, X) \wedge \text{costs}(Y, X) \Rightarrow \text{has}(U, Y)$

take(you, COMP24412)  
know(you, Logic)   know(you, Prolog)   cool(you)   canProgram(you)  
hasGoodJob(you)   has(you, LotsOfMoney)

# Inferring Some More Knowledge

teaches(COMP24412, Logic)   teaches(COMP24412, Prolog)  
about(COMP24412, AI)   cool(AI)  
language(Prolog)   costs(Yacht, LotsOfMoney)

$\text{take}(U, C) \wedge \text{teaches}(C, X) \Rightarrow \text{know}(U, X)$   
 $\text{take}(U, C) \wedge \text{about}(C, X) \wedge \text{cool}(X) \Rightarrow \text{cool}(U)$   
 $\text{know}(U, X) \wedge \text{language}(X) \Rightarrow \text{canProgram}(U)$   
 $\text{canProgram}(U) \wedge \text{know}(U, \text{Logic}) \Rightarrow \text{hasGoodJob}(U)$   
 $\text{hasGoodJob}(U) \Rightarrow \text{has}(U, \text{LotsOfMoney})$   
 $\text{has}(U, X) \wedge \text{costs}(Y, X) \Rightarrow \text{has}(U, Y)$

take(you, COMP24412)  
know(you, Logic)   know(you, Prolog)   cool(you)   canProgram(you)  
hasGoodJob(you)   has(you, LotsOfMoney)   has(you, Yacht)

# Inferring Some More Knowledge

teaches(COMP24412, Logic)   teaches(COMP24412, Prolog)  
about(COMP24412, AI)   cool(AI)  
language(Prolog)   costs(Yacht, LotsOfMoney)

$\text{take}(U, C) \wedge \text{teaches}(C, X) \Rightarrow \text{know}(U, X)$   
 $\text{take}(U, C) \wedge \text{about}(C, X) \wedge \text{cool}(X) \Rightarrow \text{cool}(U)$   
 $\text{know}(U, X) \wedge \text{language}(X) \Rightarrow \text{canProgram}(U)$   
 $\text{canProgram}(U) \wedge \text{know}(U, \text{Logic}) \Rightarrow \text{hasGoodJob}(U)$   
 $\text{hasGoodJob}(U) \Rightarrow \text{has}(U, \text{LotsOfMoney})$   
 $\text{has}(U, X) \wedge \text{costs}(Y, X) \Rightarrow \text{has}(U, Y)$

take(you, COMP24412)  
know(you, Logic)   know(you, Prolog)   cool(you)   canProgram(you)  
hasGoodJob(you)   has(you, LotsOfMoney)   has(you, Yacht)



# Automated Reasoning Beyond Puzzles (More Later)

## Datalog

This is basically relational algebra + recursion. Lots of use and interest from the database community.

## Prolog

A proper language used for real things, but not that widely. More likely to be used in specific AI-focussed domains. Used in IBM's Watson.

## Knowledge-Based Systems

Also Expert Systems, Ontologies, Semantic Web. Used in many domains with most success in clinical/health (SNOMED CT used by NHS Digital).

## Wider applications of reasoning

Formalising mathematics, hardware/software verification, security analysis

Pass over to Andre