



COMS30106 Artificial Intelligence with Logic Programming

- Unit director: Peter Flach (MVB 3.31)
 - Lectures Monday 4-6pm, Chemistry LT2
- Teaching assistants: Kacper Sokol, Torty Sivill
 - Lab group 1 Tuesday 9-11am, MVB 2.11
 - Lab group 2 Tuesday 4-6pm, MVB 2.11
- Two assignments during term (0%, 50%)
- Exam in May-June (50%)



Artificial Intelligence

- Artificial Intelligence (AI) deals with implementing intelligent behaviour on a computer
 - Robotics, vision, speech, language understanding, problem solving, ...
- In this unit we concentrate on intelligent *reasoning* behaviour
 - Reasoning with incomplete information, handling exceptions, inferring explanations, learning by generalisation...



Logic Programming

- Logic programming is a form of declarative programming particularly suited to intelligent reasoning
 - Functional programming: program is a function
 - Logic programming: program is a relation
- No strict distinction between knowledge representation and programming
 - Computation is search
 - Query may return 0, 1, 2, ... answers



Prolog

- Prolog is the most commonly used logic programming language
 - Simple syntax, no typing
- For this unit we use SWI-Prolog
 - Open-source, available for a variety of platforms
 - SWISH: SWI for Sharing
<http://swish.swi-prolog.org/>



Myth: Prolog is hard...

- Prolog is different
 - variables behave like in mathematics, not like 'changeable constants'
 - the main control structure is recursion
- Prolog is powerful
 - pattern matching
 - non-determinism
- Prolog requires a different way of thinking
 - forget procedural programming!
 - instead, concentrate on the logic of the problem



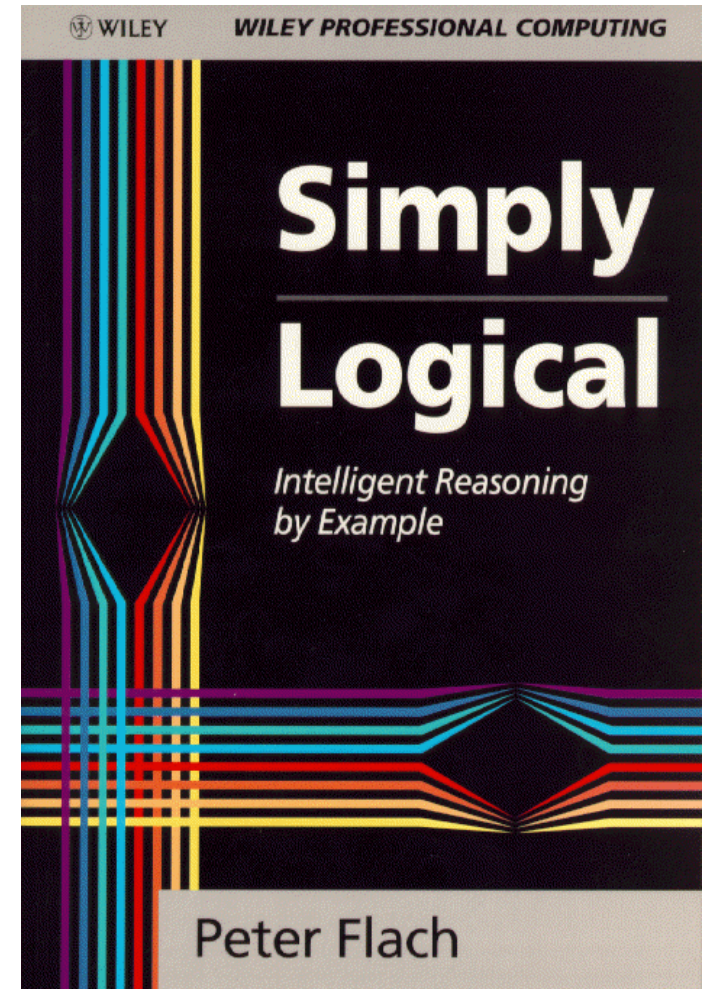
What *AIwLP* is **not** about...

- Non-symbolic AI
 - Neural networks, evolutionary computing
- most “AI” in computer games
 - The problem here is to appear human-like, ignoring the fact that most of the game state is known to the computer
 - We do cover A* search (path finding)



Simply Logical

- Most lectures will follow this fairly closely
 - Freely available at www.cs.bris.ac.uk/~flach/SimplyLogical.html
 - hardcopies available in library and on Amazon
- Interactive version under development
 - book.simply-logical.space





Overview of the lectures

- Background (6 lectures)
 - Chapter 1: Introduction
 - Chapter 3: Prolog programming
 - Chapter 5: Blind search
 - Chapter 6: Informed search
- Intelligent reasoning (10 lectures)
 - Chapter 2: Logic
 - Chapter 7: Reasoning with natural language
 - Chapter 9: Inductive reasoning
 - Chapter 8: Reasoning with incomplete information
- Advanced topics (4 lectures)
 - Answer set programming
 - Spatio-temporal reasoning



How to make the most of this unit

- Spend 7-9 hours per week as follows:
 - 1/2-1 hour preparation prior to the lecture
 - 2 hours attending the lecture
 - handouts are provided, use them to make additional notes!
 - 1/2-1 hour revision after lecture and preparation for lab
 - make the most of assistance during help desk hour!
 - 2 hours working in the lab
 - 2-3 hours working on assignment outside lab



IMDB example

(very small snapshot of 64 movies taken around 2006)

```
% movie(M,Y) <- movie M came out in year Y  
movie(the_big_lebowski, 1998).
```

```
% director(M,D) <- director D directed movie M  
director(the_big_lebowski, joel_coen).
```

```
% actor(M,A,R) <- actor A played role R in movie M  
actor(the_big_lebowski, jeff_bridges, jeffrey_lebowski__the_dude).
```

```
% actress(M,A,R) <- actress A played role R in movie M  
actress(the_big_lebowski, julianne_moore, maude_lebowski).
```



Prolog as query language

?- movie(M,2000) .

M = down_from_the_mountain ;

M = o_brother_where_art_thou ;

M = ghost_world ;

No

?- movie(M,Y),Y>2006 .

M = no_country_for_old_men

Y = 2007 ;

No

?- director(M,D),actor(M,D,_) .

M = blade_runner

D = joseph_d_kucan ;

M = ghost_busters

D = ivan_reitman ;

M = groundhog_day

D = harold_ramis ;

M = torrance_rises

D = spike_jonze ;

M = fall

D = eric_schaeffer ;

M = if_lucy_fell

D = eric_schaeffer ;

...



Understanding queries & answers

- What's the difference between these queries?

?- actor(M1,D,_),actor(M2,D,_).

?- actor(M1,D,_),actor(M2,D,_),M1\=M2.

?- actor(M1,D,_),actor(M2,D,_),M1@<M2.

- Why do these queries return answers multiple times?

?- director(_,D),actor(_,D,_). ?- director(_,D),actress(_,D,_).



Prolog as programming language

```
% worked(P,M) <- person P has worked on movie M
```

```
worked(P,M) :- actor(M,P,_).
```

```
worked(P,M) :- actress(M,P,_).
```

```
worked(P,M) :- director(M,P).
```

```
% met(P1,P2) <- P1 and P2 worked together on some film (ordered version)
```

```
met(P1,P2) :- worked(P1,M),worked(P2,M),P1@<P2.
```

```
% connect(P1,P2,L) <- P1 is connected to P2 through the list of people L
```

```
connect(P1,P2,[]) :- met(P1,P2).
```

```
connect(P1,P2,[P|Ps]) :- met(P1,P),connect(P,P2,Ps).
```