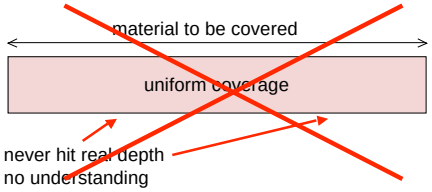
Artificial Intelligence

http://www.hcibook.com/alan/teaching/ai355/

Alan Dix (coordinator)
plus special topics:
Geoff Coulson, Paul Rayson, Gerd Kortoum, Manolis Sifalakis, Keith Cheverst, Hans Gellerson

course structure

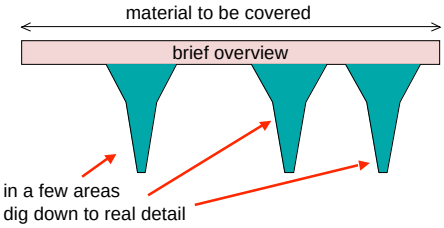
- traditional model …



… but I often do this :-(

course structure

- what we will (try to) do …



What is AI?

"The science of making machines do things that would require intelligence if done by people"

Marvin Minsky, MIT

Strong and Weak AI

'strong AI' position
- make computers *actually* intelligent
  - *"Intelligence is just a matter of physical symbol manipulation"*, Newell
  - *"We already have machines that can literally think"*, Simon
  - *"Even a machine as simple as a thermostat can be said to have beliefs"*, McCarthy

'weak AI' position
- make computers *mimic* human intelligence
- more pragmatic

Strong AI

How do you know if a machine is intelligent?

- Turing Test
  - Eliza …
- Searle's Chinese Room
  - is the room intelligent?
  - the person in the room?
  - nothing?

Weaker AI … Alien Intelligence

not as we know it …

Chess programs

- can be very good
- but NOT like a person
- Computer: very broad 'lookahead' scanning thousands of possible move paths
- Human: small number of 'sensible' moves

intuition
heuristics

the great divide

logic, search
expert systems
deduction

- symbolic (traditional) AI
  - based on high-level cognitive reasoning
  - small richer representations
  - well-defined formal representations, rules
- sub-symbolic AI
  - based on low-level neurological concepts or other 'natural computation'
  - large simple representations
  - simple attributes, weights

neural nets
genetic algorithms
emergent behaviour

Natural computation inspirations

- Neural networks / Connectionist
  - neuron firing in the brain
- Genetic Algorithms
  - natural selection / selective breeding
- Artificial life & emergent behaviour
  - colony behaviour, ants,
- Simulated annealing (not strictly 'AI')
  - crystal formation

broad areas of AI (traditional)

- knowledge representation
- reasoning
- search
- planning
- game playing
- machine learning
- language and speech
- vision

} now separate communities

(some) application areas

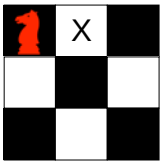
- expert systems (in many domains)
- theorem proving
- games
- robotics and control
- interfaces and ambient intelligence
- network routing
- text and data mining (inc. security)
- semantic web

recent directions

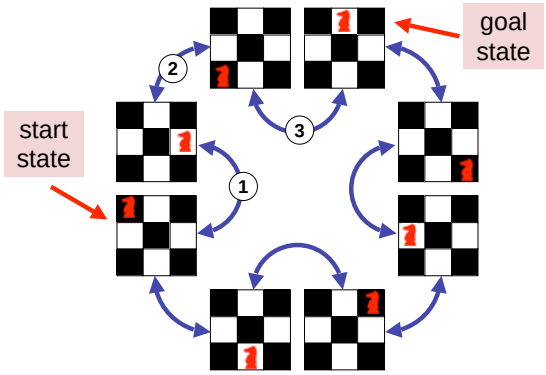
- Embodiment
  - Intelligence includes interactions with the world
- Emotion
  - intelligence includes feeling
- Emergence
  - intelligence arises in communities

Representation matters

How many moves for knight to get to square X?



… draw state space …



Knowledge Representation facts (examples)

- Predicate logic
  is_person(Jane) ∧ meeting(Jane,10am,tax_office)
- Frames (a bit like objects)
  Meeting { who:Jane, when:10am, where: tax_office)
- Semantic Web - triples/RDF
  id#15 class Person, id#15 name 'Jane',
  id#37 class Meeting, id#37 time '10am', id#37 who id#15
- may have probabilities, weights …
  meeting(Jane,*time*,tax_office), *time*=10am 75%, *time*=11am 25%

named 'slots'

in RDF URIs

Representing rules and actions

- Logical inference
  - smaller(X,Y) ∧ smaller(Y,Z) ⇒ smaller(X,Z)
- Production rules (like IF, but always 'active')
  - WHENEVER see(target) AND not moving
    DO point_towards(target), start_moving
- Scripts
  - Shopping: get trolley, fill trolley, go to checkout

Reasoning

- Forward vs. backward chaining
  - forward:
    - from start state towards goal
    - known facts infer knew ones
  - backward:
    - from goal towards start
    - from query towards facts

Forward vs. backward chaining

forward:

- from start state towards goal
- from known facts infer new ones
  eg. if we know *Dolly is a sheep* and *all sheep have wool*
  infer new fact *Dolly has wool*

backward:

- from goal towards start
- from query towards known facts

backward reasoning
example: Horn clauses used in Prolog

Known facts:

father(Henry VII,Henry VIII). father(Henry VIII,Elizabeth).

r1. ancestor(X,Y) :- father(X,Y). *{read father => ancestor}*

r2. ancestor(X,Y) :- ancestor(X,Z), father(Z,Y).

Query:

ancestor(Henry VII, Elizabeth).

Try r1: father(Henry VII, Elizabeth). – FAIL

Try r2: ancestor(Henry VII,Z), father(Z,Elizabeth).

Try r1. father(Henry VII,Z), father(Z,Elizabeth).

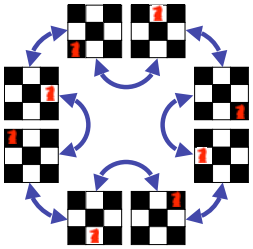
succeeds with Z = Henry VIII

search

- traditional AI algorithms with Geoff Coulson
- lots of things can be seen as search:
  - **reasoning** – find the pattern of rules that lead from premise to conclusion
  - **learning** – find the rules that explain the facts
  - **game playing** – find the move that is best no matter what my opponent does
  - **route finding** – directions and movements to destination
  - **puzzle solving** … examples …

knight's moves

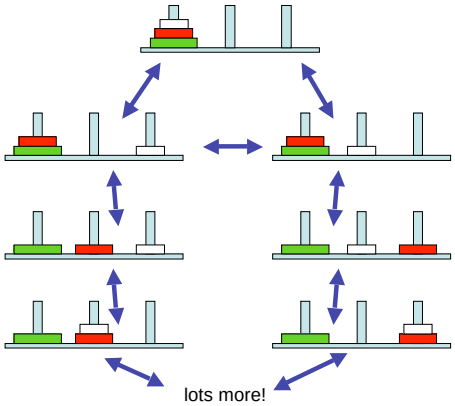
- find the shortest path from start state to end state
- state space
- goal
- start
- evaluation function (number of moves)



towers of Hanoi

- get rings from first tower to the second tower small on top of large

- state space – which ring on which tower N.B. size **constraint**
- goal – all the rings on second tower
- start – all the rings
- evaluation function – boolean succeed/fail or shortest path



lots more!

plan …

week	lecturer	topic
11	Alan Dix	Intro and my bits …
12	Geoff Coulson	Scheme Programming and Search Algorithms
13	Geoff Coulson	
14	Paul Rayson	Natural Language Processing
15	Gerd Kortuem	Reasoning, including Distributed Reasoning (plus maybe temporal reasoning)
16	Manolis Sifalakis	Emergent AI, Ant models, natural comp., …
17	Manolis Sifalakis	Applications to Networking
	Keith Cheverst	Decision Trees for Ambient Intelligence
18	Hans Gellerson	Machine Learning and N. Nets for AmbientI
19	Hans Gellerson	Computer Vision and Ubicomp
20	Alan Dix (& GC)	Group presentations
	Alan Dix	Wrap up (maybe bit of semantic web)