A New Paradigm for Computational Agents: Turing machines

James Harland
james.harland@rmit.edu.au
Overview

- Role of talks
- Turing on Turing machines
- Developments of Turing machines
- Turing `Agents'
- Discussion ....
Role of Talks

- Polished talks on completed work have a role
  - “I am giving this talk at AAMAS/IJCAI/KR/.. next month, so … “
- So do informal talks on less developed work
  - x-baked ideas for 0 < x <= 0.5
  - Informal discussion of developing ideas
  - Opportunities for interactive work

This talk is very much an invitation to a discussion …
Alan Turing

- English mathematician 1912 - 1954
- *Inventor of Turing machines* (1936)
- Built a digital multiplier in 1937
- Devised Turing test
- Led British codebreaking effort in World War II
- Designed ACE machine (1940s)
People used to be computers!
Turing Machines

- Turing set out to mimic what these people did
- Computation was mainly about numbers
- Computers were almost always women (!!)
- Computation had to be able to be interrupted and resumed later
- Concentrate on what was 'essential' to the machine
Turing on Turing machines

```
The behaviour of the computer at any moment is determined by the symbol which he is observing and his `state of mind' ''. - Alan Turing, 1936.
```

This is not too far from ...

The behaviour of the agent at any moment is determined by the environment which he is observing and his beliefs and goals.
Turing’s description

“The most general single operation …

- A possible change of symbol together with a possible change of state of mind

- A possible change of observed squares, together with a possible change of state of mind”

\[ q_1 \quad t \quad s \quad q_2 \]
Turing on Turing machines

- "It is always possible for the computer to break off from his work, to go away and forget all about it, and later to come back and go on with it."
- "We will suppose that the computer ... never does more than one step at a sitting."
- Similar to `interruptable' nature of the BDI architecture
- "Observe, think, act"
Turing machines

- Turing machines are often thought of as capturing algorithmic computations
- Input is completely specified in advance
- Computation proceeds
- Output produced
- Only one processor, so no concurrency issues
- Factorisation, TSP, finding digits of pi, ...
Turing machines

- Turing machines read the tape and write to it at every step
- Algorithmic computation property arises from two assumptions:
  - There is only one Turing machine
  - All tape changes are due to the Turing machine
- Removing these assumptions won't `break' the Turing machine
Turing machines

- Turing machines usually have one semi-infinite tape
- Turing noted that there was nothing special about the dimensions of the tape
- Simplest version often best for proving properties of machines
- Can equally use two-dimensional `tape' without any special additions
## 2D Turing Machines

![2D Turing Machines Diagram](image-url)

**Turing Machine Agents**  
**25th May, 2009**
Turing machine developments

`Heat-sensitive' Turing machines

- Read from 'hotter' cells
- Write to 'cooler' cells
- Computation halts once temperature drops too much
- No problem with termination
- Modelling natural world, where persistence means constant recreation
Turing machine developments

- Looking for Turing-complete computational mechanisms occurring in nature
- Universal computational mechanisms
- Making Turing machines more agent-like?
Turing 'Agents'

- States generalised into beliefs, goals, ...
- Revert to Turing's idea of sensing more than one square at a time
- Actions to include sending messages to other agents
- Is there an analogy of the undecidability of the halting problem?
- Is there a 'universal agent'?
- ....