Construals for Computational Reasoning

Dr Errol Thompson
Computing Education and Economics Reform Researcher
Focus conceptual change
Senior Lecture, Computer Science, Aston University

Aston University
Engineering & Applied Science

Computational Reasoning

- Why not computational thinking?
  - Goal: To have students reason about problem utilising computational techniques
- Computational thinking
  - "Mathematical abstractions called models of computation are at the heart of computation and computational thinking. Computation is a process that is defined in terms of an underlying model of computation and computational thinking is the thought processes involved in formulating problems so their solutions can be represented as computational steps and algorithms" (Aho, 2011)

Goal of Computational Models

- "to facilitate computing work in other disciplines" (Guzdial, 2016, p 40)
- "Computational thinking involves solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science." (Wing 2006)
- but ...
  - "applying computing ideas in daily life is less likely" (Guzdial, 2016, p. 40)

"A programming language is like a natural, human language in that it favours certain metaphors, images, and ways of thinking."

S. Papert (1980) Mindstorms: Children, computers, and powerful ideas

So what are we teaching?

- Thought / reasoning processes
- Computational ideas that lead to computational models
- Solution models
- Programming paradigms
- Programming languages

Does it really matter?

Variation Theory

(Marton 2015)

Discern object of learning from the background of other objects

What was I trying to help the learner learn?

- Instantiation
-
-
-
-

Concept: Colour

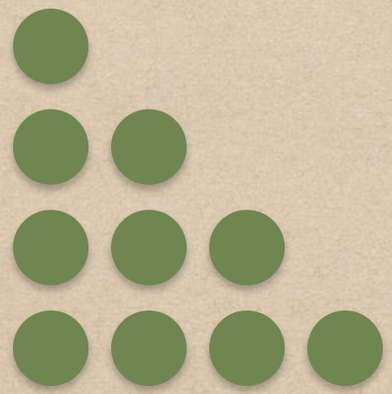
Aspect: Green

Aston U
Engineering & Applied Science

Seven Stone Nim

Find the invariant that defines a win

- Play the game ➔ Instantiation
- Prediction Pattern ➔ Contrast?



Win
Win if I take 2
Lose regardless
Win if take 1

etc...

- Change the number of start stones (solution) or change the win rule (solving strategy)➔ Generalisation?

Aston University
Engineering & Applied Science

Construal: http://jseden.dcs.warwick.ac.uk/construit/?load=236

Three Pile Nim

Can they apply the process to find the invariant?

- Play the game ➔ Instantiation
- Contrast set: [0, 1, 1], [0, 2, 2] (losing position)

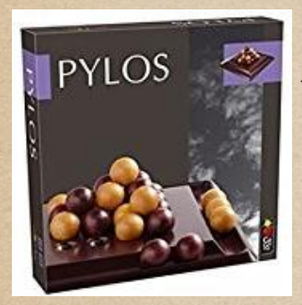
  vs. [0, 2, 7], [1, 2, 2] (win positions)
- Generalise set: [1, 2, 3], [4, 1, 5], [2, 6, 4] (losing positions)

  Binary: [01, 10, 11], [100, 001, 101], [010, 110, 100]

  vs. [1, 2, 7], [1, 4, 3], [8, 5, 3] (win positions)

  Binary: [001, 010, 111], [001, 100, 011], [1000, 0101, 0011]
- Fusion set 3: [5, 2, 7] (lose), [10, 13, 15] (win), [10, 13, 7] (lose), [8, 9, 2] (win), [8, 9, 1] (lose)
- Is the strategy for finding the solution similar to that of seven stone nim?

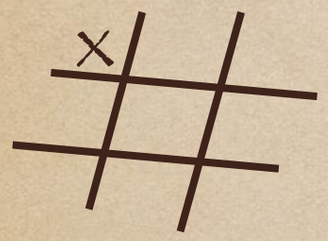
Other variants



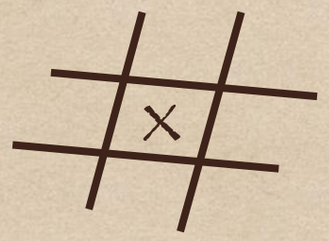
- Pylos - Can we use the same strategy to determine how to win this game?
  - Objective is to build a pyramid taking turns to place balls on the pyramid
  - Winner person to place ball at the top of the pyramid
  - Use a ball on the board to go up a level
  - Remove two balls if you get a square of four balls with no balls on top
  - Strategy is about saving balls (i.e. not placing new balls or taking balls off the board
- Do we need a different computational model?

Tic-tac-toe (OXO games)

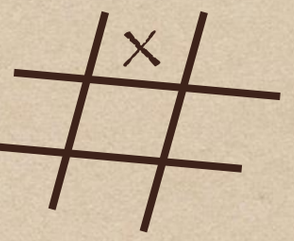
- Play the game
- Predicting the most likely out comes



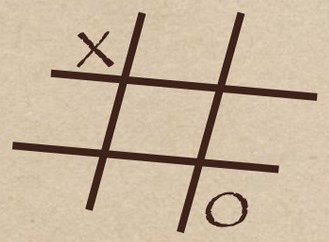
Draw but higher probability of win



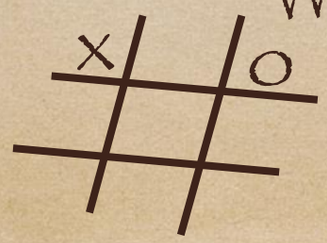
Almost certainly a draw

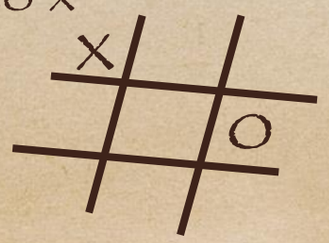


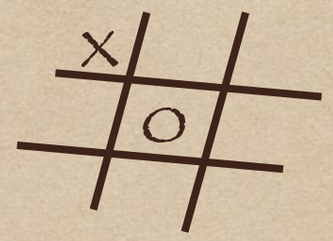
Draw but higher probability of loss



Win to x







Should be draw

Aston University
Engineering & Applied Science

Play Strategies

- Min-Max Tree Pruning
- State Transition
- Simple Rules (Win if possible, Block if possible, Play centre, Play corner, Toss)
- Logic rules

Other OXO Variants

- Cubic (3 x 3 x 3)
- Qubic (4 x 4 x 4)
- Quixo

Construal Development

- What is the knowledge we want the learner to construct?

- Constructing through programming is not the solution for all problems

- However, something that allows the learner to play with the idea and see the variation in consequences can help

References

Aho, A. V. (2011). Ubiquity symposium: Computation and Computational Thinking. *Ubiquity, 2011*(January). doi:10.1145/1922681.1922682

Guzdial, M. (2016). Learner-Centered Design of Computing Education: Research on Computing for Everyone. *Synthesis Lectures on Human-Centered Informatics, 8*(6), 1-165. doi:10.2200/S00684ED1V01Y201511HCI033

Marton, F. (2015). *Necessary conditions of learning*. New York and London: Routledge.

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*: Basic Books, Inc.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.

Wing, J. (2010). Research Notebook: Computational Thinking-What and Why? *The Link: The magazine of Carnegie Mellon University's School of Computer Science*. Retrieved from https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

Zingaro, D. (2008). *Invariants: A Generative approach to programming*. London: College Publications.