Artificial Intelligence 1

Exercises 1: introduction, search, games and constraint satisfaction problems

Dr Sean B Holden, 2010

1 Introduction

These notes provide some extra exercises for the AI1 course. Solutions are available to Supervisors and will be made available to all after the course has finished.

2 Introduction and Agents

It is notoriously difficult to predict what will be possible in the future, so your answers might well be amusing to you when you find them in twenty years time.

1. If you haven't seen it already, watch the film *A.I. Artificial Intelligence* paying particular attention to the character *"Teddy"*.
2. A large number of subjects were covered in the initial lectures in terms of how they've influenced AI: for example philosophy, mathematics, economics and so on. How do these show up in Teddy's design?
3. What aspects of Teddy are within our current capabilities to design?
4. What aspects of Teddy would you expect to be able to implement within the next fifteen years. How about the next fifty years?
5. Are there aspects of Teddy that you would expect to elude us for one hundred years or more?
6. To what extent does the "natural basic structure" for an agent, as described in the notes, form a useful basis for implementing Teddy's internals? What is missing?

3 Search

1. In the graph search algorithm, assume a node is taken from the fringe and found *not* to be a goal and *not* to be in `closed`. We then add it to `closed` and add its descendants to `fringe`. Why do we *not* check the descendants first to see if they are in `closed`?
2. The $A^\star$ algorithm does not perform a goal test on any state *until it has selected it for expansion*. We might consider a slightly different approach: namely, each time a node is expanded check all of its descendants to see if they include a goal.

Give two reasons why this is a misguided idea, where possible illustrating your answer using a specific example of a search tree for which it would be problematic.

3. The f-cost is defined in the usual way as

$$f(n) = p(n) + h(n)$$

where n is any node, p denotes path cost and h denotes the heuristic. An admissible heuristic is one for which, for any n

$$h(n) \leq \text{actual distance from } n \text{ to the goal}$$

and a heuristic is monotonic if for consecutive nodes n and n' it is always the case that

$$f(n') \geq f(n).$$

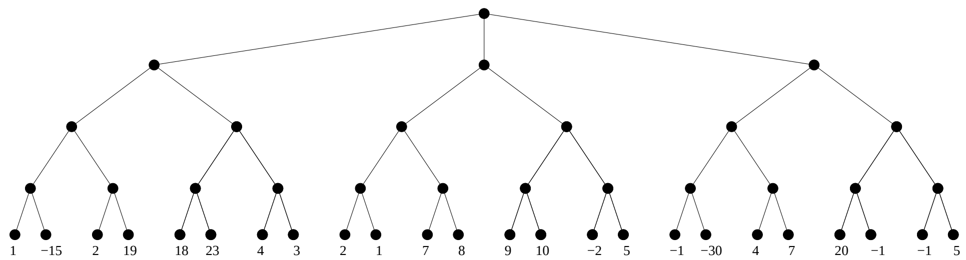
- Prove that h is monotonic if and only if it obeys the triangle inequality, which states that for any consecutive nodes n and n'

$$h(n) \leq c_{n \to n'} + h(n')$$

  where $c_{n \to n'}$ is the cost of moving from n to n'.
- Prove that if a heuristic is monotonic then it is also admissible.
- Is the converse true? (That is, are all admissible heuristics also monotonic?) Either prove that this is the case or provide a counterexample.

4 Games

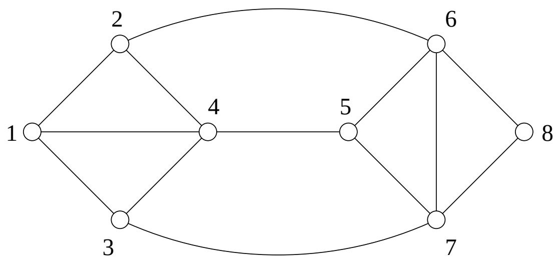
1. Consider the following game tree:



Large outcomes are beneficial for Max. How is this tree pruned by $\alpha - \beta$ minimax if Max moves first? (That is, Max is the root.) How is it pruned if Min is the root, and therefore moves first?

5 Constraint satisfaction problems

1. Consider the following constraint satisfaction problem:



We want to colour the nodes using the colours red (R), gold (G) and blue (B) in such a way that connected nodes have different colours.

- Assume we attempt the assignments $1 = R$, $4 = G$, $5 = R$, $8 = G$, $6 = B$. Explain how *forward checking* operates in this example, and how it detects the need to backtrack.
- Will the AC-3 algorithm detect a problem earlier in this case? Explain the operation of the algorithm in this example.