CS188 Spring 1993 | Intro to AI Stuart Russell

Midterm 1 Solutions

1. (15 pts.) Definitions

(a) (3 pts) Dynamic environment: changes occur other than agent actions.

(b) (3 pts) Admissible heuristic: a function that provides estimates of the cost of the best completion of a solution through a node, such that the estimates never exceed the true cost.

(c) (3 pts) Minimax decision: a decision guaranteeing the best possible result in a two-player game, assuming best play by the opponent.

(d) (3 pts) Complete inference procedure: derives all sentences entailed by a set of sentences.

(e) (3 pts) Rational agent: acts to maximize goal achievement given its beliefs.

2. (17 pts.) Heuristic search

(a) (2 pts) The state description should include the number itself *and* the position of the digit changed in the previous move.

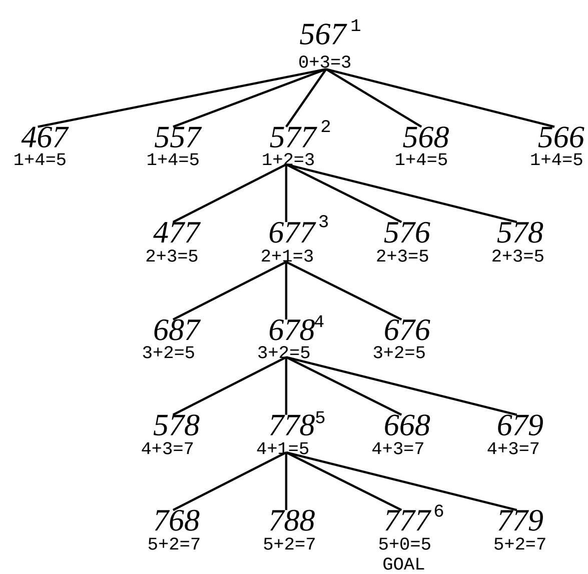
(b) (3 pts) If the digits of state n are $n_1 n_2 n_3$ and the digits of the goal are $G_1 G_2 G_3$, then use

$$h(n) = |G_1 - n_1| + |G_2 - n_2| + |G_3 - n_3|$$

This is the cost of the shortest path assuming that all moves are legal (i.e., $bad = \{\}$ and repeat moves allowed). Since the real problem disallows some of these moves, actual shortest paths can only be the same or longer.

(c) (2 pts)
$g(n)$ = cost of best path found so far from S to n
$h(n)$ = estimated cost of best path from n to G
$f(n) = g(n) + h(n)$ = estimated cost of best path constrained to go through n.

(d) (10 pts)



3. (12 pts.) Logic and knowledge representation

(a) (3 pts) There can't be two different robots at the same vertex at the same time.

$$\forall sxyv Vertex(v) \wedge Robot(x) \wedge At(x,v,s) \wedge Robot(y) \wedge At(y,v,s) \Rightarrow x = y$$

(b) (3 pts) At any given time, there's a robot at precisely one vertex.

$$\forall s \exists xv Robot(x) \wedge Vertex(v) \wedge At(x,v,s) \wedge \forall yw Robot(y) \wedge Vertex(w) \Rightarrow (x = y \wedge v = w)$$

(c) (3 pts) If a robot at one vertex tries to move to another visible vertex and there's no robot at the other vertex, then the robot ends up at the other vertex.

$$\begin{aligned}&\forall svw Robot(x) \wedge Vertex(v) \wedge Vertex(w) \wedge Visible(v,w)\\ &\quad \wedge(\neg \exists y Robot(y) \wedge At(y,w,s)) \Rightarrow At(x,w,Result(s,Move(x,v,w)))\end{aligned}$$

(d) (3 pts) If a robot at one vertex tries to move to another visible vertex and there's a robot at the other vertex, then the first robot stays put.

$$\begin{aligned}&\forall svw Robot(x) \wedge Vertex(v) \wedge Vertex(w) \wedge Visible(v,w)\\ &\quad \wedge(\exists y Robot(y) \wedge At(y,w,s)) \Rightarrow At(x,v,Result(s,Move(x,v,w)))\end{aligned}$$

$Robot(x)$: x is a robot
$Vertex(x)$: x is a vertex
$At(x,v,s)$: object x is at vertex v in situation s
$Result(s,a)$: the situation resulting from action a in situation s
$Move(x,v,w)$: action of x moving from v to w.

4. (16 pts.) Logical transformation and inference

- (A) $\forall x \exists y \ (x \geq y)$
- (B) $\exists y \forall x \ (x \geq y)$

(a) (2 pts)
(A) Every number is greater than or equal to some number
(B) There is a number that every number is greater than or equal to

(b) (1 pt) (A) is true (eg every number is greater than or equal to itself)

(c) (1 pt) (B) is true (0 is such a number)

(d) (1 pt) No

(e) (1 pt) Yes

(f) (5 pts)
(B) $\underline{(x \geq g)}$
($\neg$A) $\neg \forall x \exists y \ (x \geq y)$
($\neg$A) $\exists x \ \neg \exists y \ (x \geq y)$
($\neg$A) $\exists x \ \forall y \ \neg(x \geq y)$
($\neg$A) $\underline{\neg(h \geq y)}$
Now (B) resolves with ($\neg$A) with unifier $\{x/h, y/g\}$ to give the empty clause immediately.

(g) (5 pts)
(A) $\underline{(x \geq f(x))}$
($\neg$B) $\neg \exists y \forall x \ (x \geq y)$
($\neg$B) $\forall y \ \neg \forall x \ (x \geq y)$
($\neg$B) $\forall y \exists x \ \neg(x \geq y)$
($\neg$B) $\underline{\neg(g(y) \geq y)}$
There is no unifier for the two literals. If we try $\{x/g(y)\}$, then we are faced with unifying y with $f(g(y))$, which fails because of the occur check.