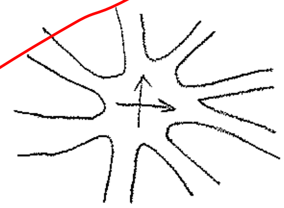
CSC 671 Take Home Final Shuowen Wei 35.5

Very Good!!

1. Based on the description of the question, we can assume that there are at most 4 corridors at each intersection in the maze, thus there're no intersections like this:



also, we assume the number of points in the maze is n;

a. State: each state includes the robot's current location (the coordinates of the point where the robot is standing on) in the maze and the direction which the robot is face to;

Initial State: the center of the maze, facing north.

Actions: turn to face north/east/west/south, move forward a certain distance, stop before hitting the wall;

Transition Model: each turn or move forward will change the robot's state to a new one;

Goad Test: Be out of the maze;

Path cost: the times of turns, the distance the robot moved.

State space: 4n (4 directions at each points, n points in total)

b. State: the robot's current location and direction

Initial State: center of the maze, facing north

Actions: turn to face the directions of the other corridors when at the intersection of two or more corridors, move forward, stop at wall, turn 180° at the dead end of a corridor. (I emailed you about assuming that the robot can't move backward, thus he can turn 180° at dead end)

Transition Model: each action above will change the robot's current state to a new one.

Goal Test: Be out of the maze

Path Cost: the times of turns, the distance the robot moved.

State space now: since each point has at least 2 directions:

Assume, the number of intersections with 3 corridors is: n_3

4 : n_4

State space is now: $2n + n_3 + 2n_4$

c. State: the robot's current location and direction

Initial State: center of the maze and facing north.

Actions: move forward or stop before hitting the wall, turn to face north/east/west/south only when reach a turning point.

Transition Model: each action above will change the robot's state to a new one.

Goal Test: Be out of the maze.

9.5

Path Cost: the times of turns, the distance the robot moved.

Since the robot only needs to turn at each turning point, and the robot can only move forward (assumed by the description of the problem, and I emailed you about this), thus we only need to which direction the robot chooses to face at each turning point, because he will keep facing direction all the way until he meets the next turning point.

d. ① we assumed there're at most 4 corridors at each intersection.

② we assumed each corridor is straight such that the robot only need to move forward

③ we assumed each corridor in the maze is either parallel to north-south direction (vertical) or east-west direction (Horizontal)

You also assumed discrete state space. This can be a bad assumption

2. ① When it's not a zero-sum game, then at every node, the two-element vector can't be reduced to a single value. Thus the evaluation function returns a vector of utilities, one for each player, and the backup step selects whichever vector has the highest value for the player whose turn it is to move, just like the multiplayer game.

② No, the alpha-beta pruning is not possible in general non-zero-sum games, since an unexamined leaf node might be optimal for both players.

③ If the two values just differ by at most a constant k, then it's easier to compute the other value if given one value. So both of the algoriths will not change too much because every node's utilities' vector can be reduced to one value, each of the player still choose the highest value for himself when it's his turn.

(All above are cited from my HW2-5.12 and your solution to HW2)

3. ① (Smoke → Smoke) ⟺ (¬Smoke ∨ Smoke)

Smoke	¬Smoke	¬Smoke ∨ Smoke
T	F	T
F	T	T

Valid !

② (Smoke → Fire) ⟺ (¬Smoke ∨ Fire)

Smoke	Fire	¬Smoke ∨ Fire
F	F	T
F	T	T
T	F	F
T	T	T

satisfiable,
neither valid nor unsatisfiable

③

Smoke	Fire	(Smoke → Fire)	(¬Smoke → ¬Fire)	WFF
F	F	T	T	T
F	T	T	F	F
T	F	F	T	T
T	T	T	T	T

satisfiable,
neither valid nor unsatisfiable

④ (Smoke ∨ Fire ∨ ¬Fire) ⟺ (Smoke)

9.5

Smoke	Fire	WFF
F	F	F
F	T	F
T	F	T
T	T	T

Satisfiable

This is VALID

neither valid nor unsatisfiable.

⑤. right = (Smoke → Fire) ∨ (Heat → Fire)

⟺ (¬Smoke ∨ Fire) ∨ (¬Heat ∨ Fire)

⟺ (¬Smoke ∨ ¬Heat) ∨ Fire

⟺ ¬(Smoke ∧ Heat) ∨ Fire

⟺ (Smoke ∧ Heat) → Fire = Left

thus, it's valid !

⑥.

Smoke	Fire	Heat	(Smoke → Fire)	((Smoke ∧ Heat) → Fire)	WFF
F	F	F	T	T	T
F	F	T	T	T	T
F	T	F	T	T	T
F	T	T	T	T	T
T	F	F	F	T	T
T	F	T	F	F	T
T	T	F	T	T	T
T	T	T	T	T	T

valid!

⑦.

Big	Dumb	(Big → Dumb)	WFF
F	F	T	T
F	T	T	T
T	F	F	T
T	T	T	T

valid !

4. ① Notation: $B \triangleq$ the taxi is blue actually

$G \triangleq$ the taxi is green actually

$b \triangleq$ the taxi appears blue to the witness

$g \triangleq$ the taxi appears green to the witness.

$$P(B) = \frac{\text{number of blue taxis}}{\text{number of all taxis}}, \quad P(G) = \frac{\text{number of green taxis}}{\text{number of all taxis}}$$

Since all taxis in Athens are blue or green, $P(B) + P(G) = 1$

Now, we know discrimination between blue and green is 0.75 reliable,

thus $P(b|B) = 0.75$ $P(g|G) = 0.75$ $(P(b|G) = 1 - P(g|G))$

Our purpose is to compute $P(B|b)$ and $P(G|b)$:

$$P(B|b) = \frac{P(B,b)}{P(b)} = \frac{P(b|B)P(B)}{P(b|B)P(B) + P(b|G)P(G)} = \frac{0.75P(B)}{0.75P(B) + (1-0.75)P(G)}$$

$$P(B|b) = \frac{0.75\,P(B)}{0.75P(B) + 0.25(1-P(B))} = \frac{3P(B)}{1+2P(B)}$$

Hence, $P(G|b) = 1 - P(B|b) = \frac{1-P(B)}{1+2P(B)}$

YOU CAN PRECISELY
CALCULATE THIS
NUMBER!!!

As long as we know the ratio of blue taxis or green taxis, we can calculate the most likely color of the taxi.

② $P(B) = 0.1$, thus $P(B|b) = \frac{0.3}{1+0.2} = \frac{1}{4}$, $P(G|b) = \frac{3}{4}$

thus, the most likely color of the taxi is green, the witness may change his belief of what he saw.

7.5

— Thank you!