Homework 2

Simon Zheng
260744353

February 17th, 2018

1 Searching under uncertainty

1.1

It is $2^n - 1$ where n is the total number of tiles (we exclude the G tile) and there is no belief state where we are in no tiles.

1.2

There are 7.

- Unique left-most tile (north, south, west)
- Unique right-most tile mirror to previous (north, east, south)
- Five top-most of any "tunnel" (north, east, west)
- Five or four excluding G bottom-most of any "tunnel" mirror to previous (east, south, west)
- Five center "hubs" with no walls (north, east, south, west)
- Four tiles between "hubs" (north, south)
- Five middle tiles in "long" tunnels (east, west)

1.3

Yes, there are two unique percepts: left-most and right-most tiles. They all give the same confidence otherwise, but there are less tiles with no walls (four) so the probability of being right is 1/4 ((3,3), (5,3), (7,3) (9,3)).

1.4

Yes.

$\Rightarrow N \Rightarrow E \Rightarrow N(\Rightarrow E)^2(\Rightarrow S)^2 \Rightarrow E(\Rightarrow S)^2$

The people starting at 1 and 2 will get to Grasshopper in 10 moves. We can also mirror this plan by flipping the west/east movements in which case it is the 3 and 4 starting point friends that will get there, since the map is symmetric.

2 Game playing

2.1 Game state tree

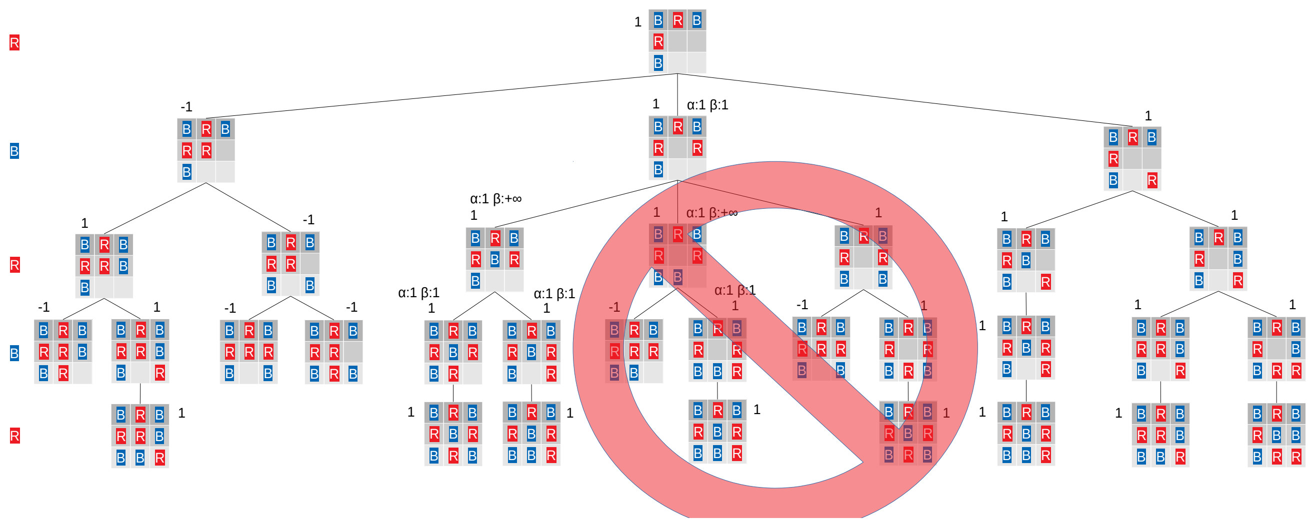

Figure 1: Game state tree with minimax

2.2 Minimax

Yes, Grasshopper can definitely be saved.

Going by the graph shown previously, the center and right child guarantee Grasshopper's safety the most. For example, the right child actually absolutely ensures a win no matter the what the following moves are.

So, if they pick the right child, then they are blocking every possible alignment. The initial top right blue is already blocked by the initial reds. These reds also block the possible row alignment for the top right blue, and the possible column alignment for the bottom left blue.

Thus, the only alignment left for blues all need the bottom right, which they blocked.

As for the red alignments, red has one turn left while blue has two. The initial reds only have one possible alignment each left. Since they are in the center they both need the center tile.

So the two possibilities are a blue box is put in the center which blocks both red alignments, or the two blue boxes left are placed in the center right and center bottom which each red alignment needs.

These apply for symmetries too, of course.

Therefore, picking the right child is the safest and guarantees a win.

2.3 $\alpha - \beta$ pruning

2.4 a)

We prune 2 of its 3 children.

2.5 b)

We saved 8 nodes. Two children are pruned in this branch.

2.6 c)

No. If we went down the one where we chose the third row, third column tile, then we could have pruned this whole node because they all lead to wins.

Since we only have a binary concept of "win" (1) or "lose" (-1), and so we only care about winning and not how much, we would return a "win" (1) so we could also prune the entire rest of the tree, saving us 25 or 27 nodes.

Explanation: we choose that branch, then either a blue box is put in the center, in which case this branch is linear and returns 1, so we would prune the other branch containing 5 nodes, or we choose that branch instead and we prune this one (3 nodes).

Now the whole branch return 1, in which case we can optimize for our problem because as mentioned previously we only care about winning at all, and since we get a guaranteed win here we just stop and prune the whole rest of the tree.

This means 25 or 27 nodes are pruned, depending on which sub-branch (of the optimal branch we picked) we consider first.

3 Propositional logic

3.1 Satisfy

3.1.1 a)

If any one of A or B is true. So: A only is true, B only is true, both are true. This is 3 models.

3.1.2 b)

If at least one of variables is true. So $2^5 - 1 = 31$ (every variable is either true or false and we want every distinct combination where at least one variable is true so we exclude the case where they are all false).

3.1.3 c)

A	B	C	$(A \land B) \lor C$
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	F
F	T	T	T
F	T	F	F
F	F	T	T
F	F	F	F

We can count 5 models.

3.1.4 d)

We can apply some reductions:

$$\begin{aligned}
&A \land (A \Rightarrow B) \land \neg B \\
&\equiv A \land (\neg A \lor B) \land \neg B \\
&\equiv ((A \land \neg A) \lor (A \land B)) \land \neg B \\
&\equiv S \land (B \land \neg B) \\
&\equiv S \land False \\
&\equiv False
\end{aligned}$$

We get a contradiction so there are 0 models that satisfy it.

3.1.5 e)

If at least one of the 6 variables is false then it is satisfied, or if both B and C are true.

But when we think about it, all of the right-hand side cases are included in the left-hand side cases.

There is one special case where all variables are true and the left-hand side is false, but then the right-hand side would be true, thus this model still satisfies it.

Thus, it is valid. $2^6 = 64$ models.

3.2 Valid, Unsatisfiable, Satisfiable

3.2.1 a)

This is valid as the only possible values for A are true or false and since if either A is true or false satisfies the sentence then it will be true no matter the value of A.

A	$\neg A$	A$\lor\neg A$
T	F	T
F	T	T

3.2.2 b)

$(A \land \neg A) \lor B \equiv False \lor B \equiv B$

So if B is true then it is satisfied. Thus it is satisfiable.

3.2.3 c)

$$\begin{aligned}
&\equiv (\neg((\neg A \lor B) \land A) \lor B) \iff True \\
&\equiv \neg((\neg A \land A) \lor (B \land A)) \lor B \\
&\equiv \neg(B \land A) \lor B \\
&\equiv (\neg B \lor \neg A) \lor B \\
&\equiv (\neg B \lor B) \lor \neg A \\
&\equiv True
\end{aligned}$$

It is valid.

3.2.4 d)

Because the left-hand side of the implication is $False$, the sentence is (vacuously) valid.

3.2.5 e)

For the right-hand side:

$$\begin{aligned}
&\equiv ((A \wedge \neg A) \vee (B \wedge \neg A)) \wedge \neg B \\
&\equiv B \wedge \neg A \wedge \neg B \\
&\equiv False
\end{aligned}$$

So we get $True \models False$ which is false. True implying a contradiction is a contradiction in itself. Every model that is true implying false is a contradiction.

Thus, the sentence is unsatisfiable.

4 First Order Logic

4.1

- D (constant): Dustey
- E (constant): Elody
- M (constant): Michael
- W (constant): William
- Eggo (constant): Eggo
- Pudd (constant): Pudding
- Musk (constant): 3-musketeers
- Bought(x,y): x bought y
- Mad(x,y): x is mad at y

$$\begin{aligned}
\forall x.Bought(x, Pudd) &\implies \neg Bought(x, Eggo) \\
\forall x.Bought(x, Musk) &\implies Bought(x, Pudd) \\
\forall x.Bought(E, x) &\iff \neg Bought(M, x) \\
Bought(M, Musk) &\wedge Bought(D, Musk)
\end{aligned}$$

4.2

1 $(\neg Bought(x, Pudd) \vee \neg Bought(x, Eggo))$

2 $(\neg Bought(x, Musk) \vee Bought(x, Pudd))$

3 $(Bought(M, x) \vee \neg Bought(E, x)) \wedge (Bought(E, x) \vee \neg Bought(M, x))$

4 $Bought(M, Musk) \wedge Bought(D, Musk)$

4.3

Yes. Elody only bought eggos. Using proof by contradiction, we negate the query:

$$\forall x. \neg Bought(x, Eggo) \vee Bought(x, Pudd) \vee Bought(x, Musk)$$

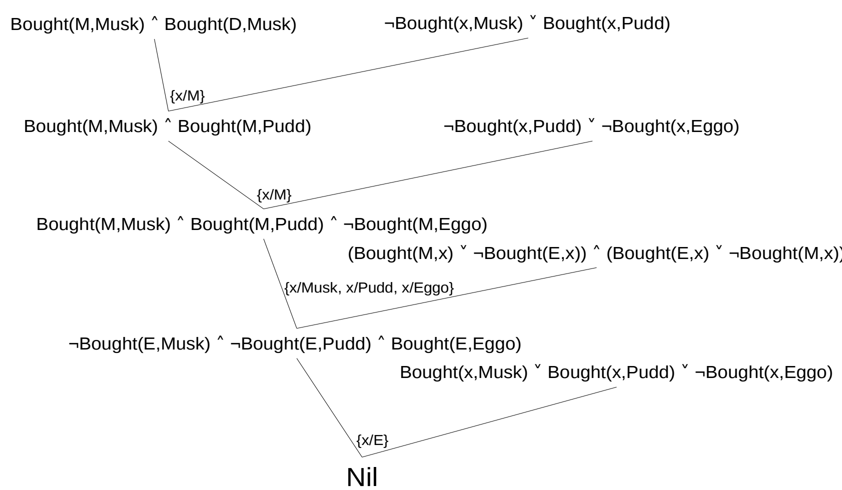

Figure 2: Resolution diagram

We get an empty set, as the query is never true (with our knowledge base).