Homework 4 solutions

5.20

a. No pruning. In a max tree, the value of the root is the value of the best leaf. Any unseen leaf might be the best, so we have to see them all.

b. No pruning. An unseen leaf might have a value arbitrarily higher or lower than any other leaf, which (assuming non-zero outcome probabilities) means that there is no bound on the value of any incompletely expanded chance or max node.

c. No pruning. Same argument as in (a).

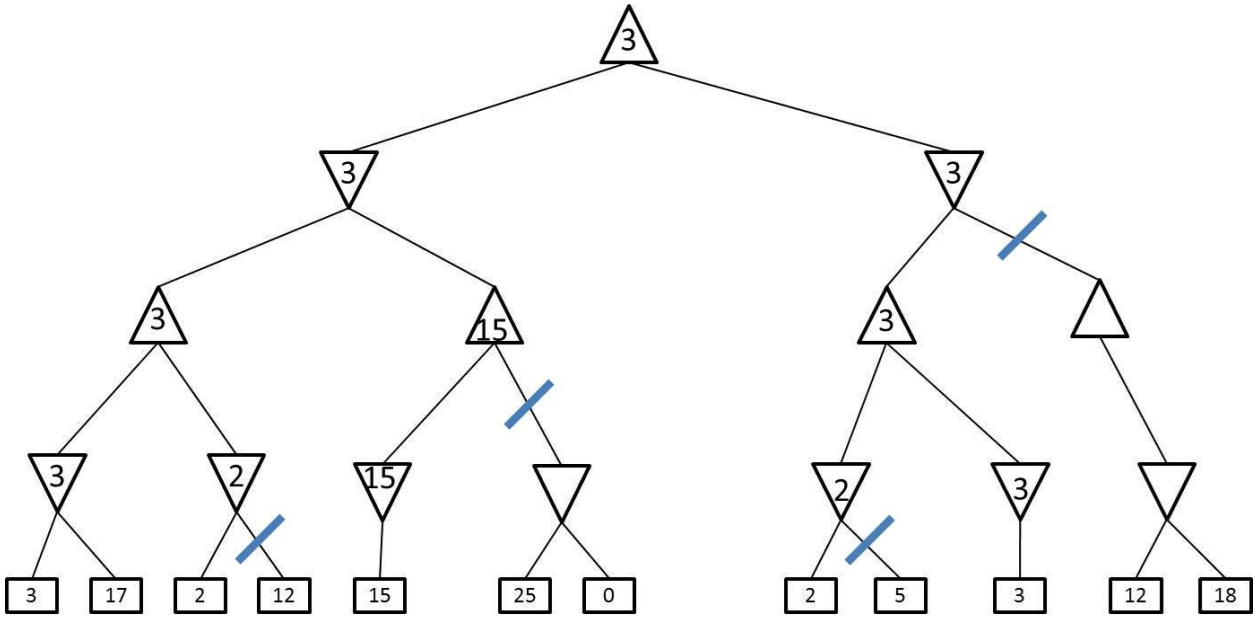
d. No pruning. Nonnegative values allow *lower* bounds on the values of chance nodes, but a lower bound does not allow any pruning.

e. Yes. If the first successor has value 1, the root has value 1 and all remaining successors can be pruned.

f. Yes. Suppose the first action at the root has value 0.6, and the first outcome of the second action has probability 0.5 and value 0; then all other outcomes of the second action can be pruned.

g. (ii) Highest probability first. This gives the strongest bound on the value of the node, all other things being equal.

Problem 2



Problem 3

(a) A=False and B=True is a world that satisfies the given sentence

(b) A=False and B=False is a world that satisfies the given sentence

Problem 4 (Try to prove this; is simple)

(a) The sentence is valid.
(b) The sentence is not valid. The world (A=False and B=True) does not satisfy the sentence.
(c) The sentence is valid.

7.5 Remember, $\alpha \models \beta$ iff in every model in which α is true, β is also true. Therefore,

a. α is valid if and only if $True \models \alpha$.
Forward: If $alpha$ is *valid* it is true in all models, hence it is true in all models of $True$.
Backward: if $True \models \alpha$ then α must be true in all models of $True$, i.e., in all models, hence α must be valid.

b. For any α, $False \models \alpha$.
$False$ doesn't hold in any model, so α trivially holds in every model of $False$.

c. $\alpha \models \beta$ if and only if the sentence $(\alpha \Rightarrow \beta)$ is valid.
Both sides are equivalent to the assertion that there is no model in which α is true and β is false, i.e., no model in which $\alpha \Rightarrow \beta$ is false.

d. $\alpha \equiv \beta$ if and only if the sentence $(\alpha \Leftrightarrow \beta)$ is valid.
Both sides are equivalent to the assertion that α and β have the same truth value in every model.

e. $\alpha \models \beta$ if and only if the sentence $(\alpha \wedge \neg\beta)$ is unsatisfiable.
As in **c**, both sides are equivalent to the assertion that there is no model in which α is true and β is false.

7.17

a. The negated goal is $\neg G$. Resolve witht he last two clauses to produce $\neg C$ and $\neg D$. Resolve with the second and third clauses to produce $\neg A$ and $\neg B$. Resolve these successively against the first clause to produce the empty clause.

b. This can be answered with or without $True$ and $False$ symbols; we'll omit them for simplicity. First, each 2-CNF clause has two places to put literals. There are $2n$ distinct literals, so there are $(2n)^2$ syntactically distinct clauses. Now, many of these clauses are semantically identical. Let us handle them in groups. There are $C(2n, 2) = (2n)(2n - 1)/2 = 2n^2 - n$ clauses with two different literals, if we ignore ordering. All these

clauses are semantically distinct except those that are equivalent to $True$ (e.g., $(A \vee \neg A)$), of which there are n, so that makes $2n^2 - 2n + 1$ clauses with distinct literals. There are $2n$ clauses with repeated literals, all distinct. So there are $2n^2 + 1$ distinct clauses in all.

c. Resolving two 2-CNF clauses cannot increase the clause size; therefore, resolution can generate only $O(n^2)$ distinct clauses before it must terminate.

d. First, note that the number of 3-CNF clauses is $O(n^3)$, so we cannot argue for nonpolynomial complexity on the basis of the number of different clauses! The key observation is that resolving two 3-CNF clauses can *increase* the clause size to 4, and so on, so clause size can grow to $O(n)$, giving $O(2^n)$ possible clauses.

Problem 7

The DPLL trace would look very similar to the forward chaining. To perform DPLL, we have to first convert (KB and ¬Q) to CNF. The CNF is given below:

$$(\neg P \vee Q) \wedge (\neg L \vee \neg M \vee P) \wedge (\neg B \vee \neg L \vee M) \wedge (\neg A \vee \neg P \vee L) \wedge (\neg A \vee \neg B \vee L) \wedge A \wedge B \wedge \neg Q$$

The CNF has no pure-literals. After unit propagation, we will get seven unit clauses (A,B, ¬Q,L,M,P,Q). Since there is a contradiction here, the CNF is unsatisfiable and therefore KB entails Q.

Problem 8

(a) $\neg P \vee \neg Q \vee R$

(b) $(P \vee R) \wedge (P \vee \neg S) \wedge (\neg Q \vee R) \wedge (\neg Q \vee \neg S)$

7.19

a. Each possible world can be expressed as the conjunction of all the literals that hold in the model. The sentence is then equivalent to the disjunction of all these conjunctions, i.e., a DNF expression.

b. A trivial conversion algorithm would enumerate all possible models and include terms corresponding to those in which the sentence is true; but this is necessarily exponential-time. We can convert to DNF using the same algorithm as for CNF except that we distribute $\wedge$ over $\vee$ at the end instead of the other way round.

c. A DNF expression is satisfiable if it contains at least one term that has no contradictory literals. This can be checked in linear time, or even during the conversion process. Any completion of that term, filling in missing literals, is a model.

d. The first steps give

$$(\neg A \vee B) \wedge (\neg B \vee C) \wedge (\neg C \vee \neg A) \ .$$

Converting to DNF means taking one literal from each clause, in all possible ways, to generate the terms (8 in all). Choosing each literal corresponds to choosing the truth value of each variable, so the process is very like enumerating all possible models. Here, the first term is $(\neg A \wedge \neg B \wedge \neg C)$, which is clearly satisfiable.

e. The problem is that the final step typically results in DNF expressions of exponential size, so we require both exponential time AND exponential space.