IAIP Exercises Week 5

1. **(*)** Russell and Norvig p.237: 7.4

2. Russell and Norvig p.237: 7.5

3. Russell and Norvig p.237: 7.6

4. **(*)** Finish the proof of the ground resolution theorem Russell and Norvig p.217. That is, show that $P_1,\ldots,P_k$ exits and is a model of S

5. **(* -partly)** (part of Computational Complexity, Papadimitriou, 1994, p.84, Problem 4.4.4) Give the shortest expression you can find equivalent to each of these expressions (**true** and **false** are also allowed, if the expression is either a tautology or unsatisfiable, respectively). Prove the equivalence.
   In giving a proof you are allowed to use the equivalences given on the next page, the use of truth tables is not allowed as proof technique. In each step of the proof you should indicate which equivalence is being used (it is okay to skip steps involving A9, A10, A11 and A12 and its okay to use the same equivalence twice in one step).
   a. $x \vee \neg x$
   b. $x \Rightarrow (x \wedge y)$
   c. $(y \Rightarrow x) \vee x$
   d. **(*)** $((x \Rightarrow y) \Rightarrow (y \Rightarrow z)) \Rightarrow (x \Rightarrow z)$

6. **(*)** DPLL implementation (you can form groups of up to 3 students)
   For details of each of the implementation tasks given below, please be sure to consult the file readme.pdf which is included with the code.

   a. Implement the pure symbol and the unit clause heuristic in the provided code for DPLL.
   b. Demonstrate the phase transition on the provided test problems. Use the included helper program to plot the ratio of unsatisfiable instances, the runtime of WalkSat and the runtime and number of splits of DPLL as a function of the clause/symbol ratio *m/n*. At what ratio *m/n* do you observe a phase-transition?
   c. Extra (non-mandatory). Define a heuristic for choosing the symbol to split on that you believe will improve the performance of the algorithm. What is the intuition behind your heuristic? Implement the heuristic and repeat the DPLL runtime and split plot done in b. Have your heuristic reduced the number of splits? Has this reduction caused a decrease in the runtime of the algorithm?

Logic Equivalences

A1: $\alpha \wedge \neg\alpha \equiv$ false
A2: $\alpha \vee \neg\alpha \equiv$ true
A3: $\alpha \wedge$ true $\equiv \alpha$
A4: $\alpha \vee$ false $\equiv \alpha$
A5: $\alpha \wedge$ false $\equiv$ false
A6: $\alpha \vee$ true $\equiv$ true
A7: $\alpha \wedge \alpha \equiv \alpha$
A8: $\alpha \vee \alpha \equiv \alpha$

A9 $(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$ commutativity of $\wedge$
A10 $(\alpha \vee \beta) \equiv (\beta \vee \alpha)$ commutativity of $\vee$
A11 $((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$ associativity of $\wedge$
A12 $((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$ associativity of $\vee$
A13 $\neg(\neg\alpha) \equiv \alpha$ double-negation elimination
A14 $(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)$ contraposition
A15 $(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta)$ implication elimination
A16 $(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$ biconditional elimination
A17 $\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$ de Morgan
A18 $\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$ de Morgan
A19 $(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$ distributivity of $\wedge$ over $\vee$
A20 $(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$ distributivity of $\vee$ over $\wedge$