Assignment 6: Decision Procedures, Temporal Properties
15-414/15-424 Bug Catching: Automated Program Verification

Due: **11:59pm**, Sunday 12/2/18
Total Points: 50

1. **Pigeonhole SAT (10 points)** The pigeonhole problem asks us to find a one-to-one mapping between n pigeons and m holes. Obviously, this isn't possible when $n > m$. Consider an encoding of this problem as SAT for n pigeons and $n-1$ holes, where we have the following CNF clauses and propositional variables p_{ij} which assert that pigeon i is placed in hole j.

   - *Pigeon clauses*: For each pigeon $1 \leqslant i \leqslant n$, assert that it is placed in some hole.

   $$p_{i,1} \vee \cdots \vee p_{i,n-1}$$

   - *Hole clauses*: For each hole $1 \leqslant j < n$ and each pair of pigeons $1 \leqslant i < k \leqslant n$, these two pigeons aren't placed in the same hole:

   $$\neg p_{i,j} \vee \neg p_{k,j}$$

   First, write down a CNF for the pigeonhole problem for $n = 3$. Then, apply the DPLL algorithm with clause learning to the formula. You should write down the steps of your evaluation in the following form:

   (1) Decide p
   (2) Unit propagate q from clause C_2
   (3) Decide $\neg r$
   (4) Unit propagate s from clause C_1
   (5) Conflicted clause C_1
   (6) Learn conflict clause $\neg p \vee r$
   (7) ...

   You are free to generate conflict clauses using any of the methods described in Lecture 13 [1], but you may want to look at the next problem before choosing one.

2. **Resolving conflict (10 points)** Use the resolution rule to derive a proof that one of your conflict clauses from question 1 is entailed by the original formula.

   $$\text{(res)} \quad \frac{\Gamma \vdash P, \Delta \quad \Gamma \vdash \neg P, \Delta}{\Gamma \vdash \Delta}$$

   As the formula is quite long, you may use the symbol F to denote the formula in your premises, so if your conflict clause is C, then you are to use the res rule to prove $F \vdash C$.

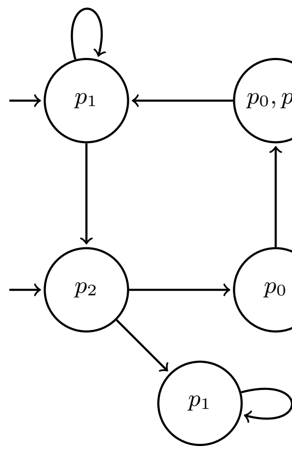
   How many applications of res were necessary in your proof? Do you think that it is possible to find a shorter one? Explain your answer.

[1] Available at `https://www.cs.cmu.edu/~15414/lectures/13-dpll.pdf`

3. **Computation tree semantics (5 points).** Consider the computation structure K below, and the CTL formula:

$$\mathbf{A}(p_0 \mathbf{U} p_1) \vee \mathbf{EX}(\mathbf{AG} p_1)$$

For each state in the computation structure, write down the subformulas of the above CTL satisfied by the state. Then, say whether the structure satisfies the formula, i.e. $K \models \mathbf{A}(p_0 \mathbf{U} p_1) \vee \mathbf{EX}(\mathbf{AG} p_1)$.



4. **Temporal distinctions (10 points).** Show that the following pair of CTL and LTL formulas are not equivalent:

$$\mathbf{AF}(a \wedge \mathbf{AX} a) \qquad \Diamond(a \wedge \circ a)$$

To do so, write down a computation structure that satisfies one but not the other. Show that this is the case by providing a counterexample path for the non-statisfied formula, and explaining why the other is modeled by your system.

5. **Distributing correctly (15 points).** Consider the following LTL equivalences that characterize distributive properties of temporal operators:

$$\Diamond(P \vee Q) \leftrightarrow \Diamond P \vee \Diamond Q$$
$$\Diamond(P \wedge Q) \leftrightarrow \Diamond P \wedge \Diamond Q$$
$$\Box(P \vee Q) \leftrightarrow \Box P \vee \Box Q$$
$$\Box(P \wedge Q) \leftrightarrow \Box P \wedge \Box Q$$

First, identify which of those equivalences are correct and which are not. Then use the semantics of LTL given in lecture 16 [2] to justify your answer with a proof. For the formulas that are not correct, describe an infinite trace that satisfies one side of the equivalence but not the other, i.e., provide a counterexample.

[2] Available at `https://www.cs.cmu.edu/~15414/lectures/16-temporal.pdf`