# CS226 Practice Problem Set 4 (Spring 2016)

**Date posted: Mar 6, 2016** **Expected Solving Time: 45 mins**

- *If you need to make any assumptions, state them clearly.*
- ***These are ungraded practice questions. You are strongly encouraged to solve these on your own to ensure you understand the material being taught in class.***
- ***Mutual discussion is allowed, but copying is not. Please read the guidelines on the course webpage if you don't understand the distinction between the two.***

1. Consider the and-inverter graph in Fig. 1 in which nodes have not necessarily been structurally hashed.

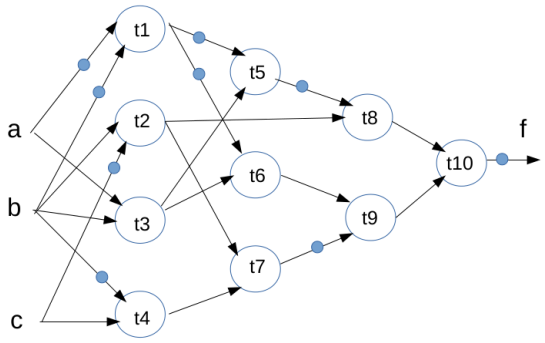


Figure 1: And-inverter graph (not strash-ed)

   (a) Identify all pairs of nodes in the above AIG that can be merged by structural hashing (strash-ing).
   (b) Identify all pairs of nodes in the above AIG that can be merged by fraig-ing.
   (c) Construct an AIG in which all pairs of nodes identified in the previous two sub-questions are merged.
   (d) Use the technique discussed in class (Tseitin encoding) to give a CNF (or product-of-sums) formula that is equisatisfiable to $f$ from the simplified AIG obtained above.
   (e) Give a satisfying assignment of the CNF formula obtained in the previous sub-question.

2. We have seen how the ROBDD for a function $F$ may be simplified using a set of don't-cares specified by a (possibly different) function $G$ (recall the mid-sem question, where you can think of $G'$ as specifying the set of don't cares). In this question, we wish to try something similar with AIGs.

   (a) Give a recursive formulation for simplifying $F$ using a set of don't cares specified by $G'$, where both $F$ and $G$ are given as FRAIGs. Clearly indicate the termination cases.
   (b) Let $F$ be the function represented by the FRAIG obtained in sub-question 1(c) above. Let $G$ be the function $a.b + b.c'$.
      i. Construct a FRAIG for $G$.

ii. Use the recursive formulation of sub-question 2(a) to simplify the FRAIG for $F$ using $G'$ (note: not $G$) as a specification of don't cares. Your representation for the simplified function need not be a FRAIG, but it should be structurally hashed.

(c) Suppose $F$ is represented by two structurally different AIGs (this is possible, since AIGs are not canonical). Similarly, suppose $G$ is represented by two structurally different AIGs. Is it possible that the result of simplifying $F$ using $G'$ as don't cares, and using your recursive formulation of sub-question 2(a), be different depending on which AIG for $F$ or $G$ you choose to start with?

Either provide a counter-example, or argue why the result is independent of your choice of AIGs for $F$ and $G$.