Homework 2 solutions

1. Prove the following algebraic properties:

   (a) $(g \wedge (f \downarrow g)) = (f \wedge g)$

   Let us consider any point x in the domain of f and g. Let x_0 be the closest point to x such that $g(x_0)$. Now there are two possible cases:

   - $g(x) = 1$: In this case, $x_0 = x$ so $(f \downarrow g)(x_0) = f(x_0)$. Hence, $g(x_0) \wedge (f \downarrow g)(x_0) = f(x_0) \wedge g(x_0)$.
   - $g(x) = 0$. In this case, $g(x_0) \wedge (f \downarrow g)(x_0) = 0 = f(x_0) \wedge g(x_0)$.

   The identity holds in both cases.

   (b) $(f \wedge g) \downarrow h = (f \downarrow h) \wedge (g \downarrow h)$. (In fact, $\downarrow$ distributes over all Boolean operators.)

   Let x be any point in the input space. Let x_0 be the closest point to x in h. Thus LHS $((f \wedge g) \downarrow h)(x) = (f \wedge g)(x_0) = f(x_0) \wedge g(x_0)$. The RHS $((f \downarrow h) \wedge (g \downarrow h))(x) = (f \downarrow h)(x) \wedge (g \downarrow h)(x) = f(x_0) \wedge g(x_0)$ which matches LHS. Hence the identity holds.

   Essentially the same proof can be used to show that $\downarrow$ distributes over other Boolean operators.

2. Shannon decomposition works with generalized cofactor as well as cofactoring on a single variable. In other words, $f = \text{ite}(g, f \downarrow g, f \downarrow \neg g) = (g \wedge (f \downarrow g)) \vee (\neg g \wedge (f \downarrow \neg g))$. Prove it.

   From the previous problem, we know $(g \wedge (f \downarrow g)) = (f \wedge g)$. Similarly, $(\neg g \wedge (f \downarrow \neg g)) = (f \wedge \neg g)$. Thus RHS of the identity we are interested in proving is equivalent to $(f \wedge g) \vee (f \wedge \neg g)$ which by DeMorgan's Law equals f.

3. Find two functions f and g where the BDD for $f \downarrow g$ is bigger than the BDD for f.

   The idea is to choose a function f for which there is a lot of subgraph sharing and then let constrain destroy some of the sharing. Consider the family of functions $f_n = x_1 \oplus \ldots \oplus x_n$ and $c_n = x_1 \vee \ldots \vee x_n$. The variable order is irrelevant because of symmetry. For plain BDDs, f_n has $2n+1$ nodes (including the terminal nodes), whereas $f_n \downarrow c_n$ has $3n-2$ nodes. For BDDs with complement edges, f_n had $n+1$ nodes, whereas $f_n \downarrow c_n$ has $2n-1$ nodes. In either case the BDD size increases after the constrain operation.

4. SAT with BDDs, with and without constrain.

   Results for all students: BDDs without constrain were almost completely useless. The method using constrain could solve a few problems, especially when students added more optimizations. But, in all cases, it was much worse than a modern SAT solver. I did not expect BDDs to be competitive, but results were worse than I expected.