COMP 408: BDD Size and Verification

Due in class on Wednesday, February 5. You do not need to turn in any SMV files for this assignment. Just turn in the results and commentary that I've asked for in the problems below.

Changing Variable Orderings in SMV: If you run `smv -o filename foo.smv`, SMV will write the variable ordering it used on foo.smv to file *filename* (SMV will not test the SPECs in the file under -o). Running `smv -i filename foo.smv` will test foo.smv using the variable ordering given in file *filename*. Use these options to do the following exercises.

Exercise 1 (**Shifters Revisited**) For this exercise, I want you to examine the effect of variable ordering in verifying a shifter like the one you used for the first assignment. Files *shifter1.smv* and *shifter2.smv* in `comp408/Assignments/3` extend our previous shifter example to eight bits instead of four bits. The files define the same shifter, but differ in how the previous value in the shifter is stored for purposes of verification. In *shifter1.smv*, we use a numeric variable to record the previous number represented by the bits in the shifter. In *shifter2.smv*, we use a separate variables to store the previous values of each bit individually. Copy these two files and do the following:

1. For each file, run SMV using several different variable orderings (including the default order used by SMV). Take three orderings (the ones with the smallest and largest number of BDD nodes, plus one of your choice in between) and write down the ordering and the number of BDD nodes produced for it. What in particular about your best/worst orderings do you think lead them to yield the results that they did?
2. Compare the BDD-size results between these two files. Which file yielded the smallest BDD? Do you see anything in the definition of that file that might explain this?

Note: The point of this exercise is not to have you find the smallest possible BDD for each file — the point is to get you thinking about how variable ordering can affect BDD size. In experimenting with variable orderings, try to come up with ones that you think will produce either really large or really small BDDs and use this as an exercise to test your intuitions.

Exercise 2 (**Rotate**) Given two bit vectors of the same length, the first is a rotation of the second iff we can shift the first circularly by some number of bits to obtain the second. For example, "10110" is a rotation of "11010" (by rotating two bits to the right), but "11001" is not a rotation of "10110". Rotation is similar to shifting, only the bit shifted out of one end is shifted back into the other end.

File `comp408/Assignments/3/rotate.smv` defines a variable *rotated* that is true exactly when array N is a rotation of array M. N and M are defined to be boolean arrays of size ten. As in the previous exercise, experiment with variable orderings and write down the three orderings that produced the smallest, largest, and one other BDD size. Compare the ratio between the smallest and largest figures here and those you got in the previous exercise. Can you find any characteristics of the rotate function that prohibit you from finding a variable ordering that produces a small BDD?