1 Bug Detection in IC fabrication

Suppose you work for the Indian Space Research rgani ation (ISR) and need to have a inte- grated circuit (IC aka chip) developed that will go into the next satellite launch. Unfortunately, India does not have a state-of-the-art IC fabrication facility. So, while ISR can design the chip, simulate it and test it, the actual manufacturing will have to be done in China. ISR may be concerned that the design that they send to China may not be the one that is actually fabricated. In fact, some research has shown that changing just a few gates out of several million gates is enough to introduce serious security vulnerabilities into an IC.

Unfortunately, these modifications are nearly impossible to detect because the IC will work correctly for almost all inputs. It will only be a few secret and carefully chosen inputs for which the vulnerability will be triggered. In theory one could just test the output of the fabricated circuit for all possible inputs because there are only a finite number of inputs and compare this output with the expected output from the design for each input. If there are any discrepancies, we know for sure that there has been some hanky-panky in the fabrication facility. In practice, this does not work for circuits with more than 50 inputs.

Your job is to use SAT solvers to detect such modifications by carefully choosing test inputs that will expose them.

Problem

- Look at the file `Circuit.scala`.

  Your job is to complete the function `checkEquivalenceOfCircuits` which takes in the actual circuit that you wanted to manufacture - `actualCircuit` (also sometimes called the "golden" circuit) and the fabricated circuit you got from China `givenCircuitEval`.

- The latter, `givenCircuitEval`, is not given as an `Expr` but instead as a Scala function that returns the result of the fabricated circuit s output given an assignment to the input variables of the chip in the form of a map - `Map[Variable, Boolean]`.

  This models the fact that once you have a manufactured circuit, you can t really see the gates that comprise it. Instead you can provide it inputs and see what the outputs are.

- A na ve implementation of this function without the use of SAT solvers has been presented to you. It creates all possible inputs to the actual circuit that you wanted to manufacture and compares the output of this so-called "golden" circuit and the fabricated circuit s output. If at any point, you find that the golden circuit s output and the fabricated circuit s outputs don t match, you

know that the manufactured circuit is not the circuit you sent for fabrication.

- Now, you have a friend in the fabrication facility in China who gave you some essential information regarding the fabricated chip C. He said that it is possible that **either C is equivalent to the actual circuit you asked for** or ***exactly one*** **of the many AND gates in the actual circuit has been modifted to an OR gate**, but he doesn t know which one.

- It is not possible to check the circuits for all possible input variable assignments as there may be thousands inputs to the chip. Hence, the na ve implementation will not work. With the information given by your friend, you know all possible circuits that you can get from the fabrication facility in China, by modifying each AND gate into an R gate one at a time. Can you use *this* fact to reduce the search space ?

- The function `checkEquivalenceOfCircuits` should return `false` if the circuit you wanted and the circuit you received from China aren t equivalent and `true` if they are. A sample test has been provided in `CircuitTest.scala` to help you understand better. You can add more tests here.

  The actual testcases will be much larger than the one given.

  Submit only the file `Circuit.scala`.