

Feb 4th Report

By Xiang Chen (Echo)

Feb 4, 2020

Contents

1	Script	2
2	Initialized Array Declaration	4

1 Script

I finish the script and discuss it with Paul, now the program has been installed to the prism lab. Below is the script:

```
1 #!/bin/bash
2 set -m
3
4 verifier_jar=/eecs/fac/pkg/verifier/verifier.jar
5 LD_LIBRARY_PATH=/eecs/fac/pkg/verifier/lib
6 export LD_LIBRARY_PATH
7
8
9 if [ -n "$1" ]
10 then
11     # if the first argument is "-h", print the help message
12     if [ $1 == "-h" ]
13     then
14         echo
15         echo "verifier [version 1.0]."
16         echo
17         echo "Usage: verifier Options [file]"
18         echo
19         echo "Options:"
20         echo "-p: For Pretty Printing."
21         echo "  Where you could achieve an output file for each formula that could be
           recognized by the z3 online tool."
22         echo "-v: For Verification."
23         echo "  Where you could test if the formula is a tautology. If it's not a
           tautology, you could also get a counter example."
24         echo "-r: For Regression testing."
25         echo "  Where you could perform the acceptance test on a series of test cases."
26         echo
27         echo "When no file is passed, the verifier is run in an interactive mode."
28     # if the first argument is "-p" - pretty printing
29     elif [ $1 == "-p" ]
30     then
31         # if there is an input file
32         if [ -n "$2" ]
33         then
34             echo "======"
35             echo "Now start executing the JAR file....."
36             echo "======"
37             echo "Below is the output for pretty printing of file: $2"
38             echo "======"
39             # run the JAR file
40             java -jar verifier.jar $1 $2
41         # if there is no input file
42         else
43             echo "Now start executing the JAR file....."
44             echo "======"
45             echo "You did not include an input file, please start typing your input
               below:"
46             echo "======"
47             # run the JAR file
48             java -jar verifier.jar $1
49         fi
50     # if the first argument is "-v" - verification
51     elif [ $1 == "-v" ]; then
52         # if there is an input file
53         if [ -n "$2" ]
```

```

54     then
55         echo "=====
56         echo "Now start executing the JAR file....."
57     echo "=====
58     echo "Below is the output for verification of file: $2"
59     echo "=====
60     # run the JAR file
61         java -jar verifier.jar $1 $2
62     # if there is no input file
63     else
64         echo "Now start executing the JAR file....."
65     echo "=====
66     echo "You did not include an input file, please start typing your input
        below:"
67     echo "=====
68         # run the JAR file
69         java -jar verifier.jar $1
70     fi
71 # if the first argument is "-r" - regression testing
72 elif [ $1 == "-r" ]; then
73     read -p "Do you want to perform regression test of pretty printing(p) or
        verification(v)? [p|v]: " test
74     # if user want to perform pretty printing
75     if [ $test == "p" ]
76     then
77         cd regression/Regression_Test_PrettyPrinter/
78         python3 JAVA_Test_PrerryPrinter.py
79         cd ../../
80     # if user want to perform verifier
81     elif [ $test == "v" ]
82     then
83         cd regression/Regression_Test_Verifier/
84         python3 JAVA_Test_z3Verifier.py
85         cd ../../
86     else
87         echo "Unrecognized argument. Please indicate if you want to perform
        regression test of pretty printing(p) or verification(v)"
88     fi
89 else
90     echo "Warning: Unrecognized argument"
91     echo
92     echo "Usage message:"
93     echo "  -p: Pretty Printing"
94     echo "  -v: Verification"
95     echo "  -r: Regression Testing"
96     echo "  -h: help"
97 fi
98 else
99     echo "Warning: Unrecognized argument"
100    echo
101    echo "Usage message:"
102    echo "  -p: Pretty Printing"
103    echo "  -v: Verification"
104    echo "  -r: Regression Testing"
105    echo "  -h: help"
106 fi

```

2 Initialized Array Declaration

I modify my program to support initialized Array declaration. The equivalent encoding in Z3 is as follows:

User input:

```
1 a : ARRAY[REAL] = << 2.1, 3, 4.5, 9.0 >>  
2 verify a[1] > 0
```

Equivalent encoding in Z3:

```
1 (declare-const a (Array Int Real))  
2 (assert (= (select a 1) 2.1))  
3 (assert (= (select a 2) 3))  
4 (assert (= (select a 3) 4.5))  
5 (assert (= (select a 4) 9.0))  
6 (assert (not (> (select a 1) 0)))  
7 (check-sat)
```