

SoCV hw5 Report

Name: Yu-Ting Cheng 鄭宇廷

ID: B08202013

1. Implementation

buildInitState(): Similar to hw3, setting all state to 0 is equivalent to AND all \sim state, i.e. $(\sim x_0) \& (\sim x_1) \& (\sim x_2) \dots = 1$. Therefore, I set the initState CirGate I as `_cirMgr->const1` initially and AND it with all other states.

itpUbmC(): I implement this function based on the algorithm provided in lecture note 09 p.63. I use `getItp()` function to get the interpolation. To check the reachability has reach fixed point, I use `createXorGate()` function to check if `R_prime == R`.

Interpolation-based UBMC

```
let k = 0
repeat_1
  if  $\text{BMC}_k(S_0, F) = \text{SAT}$ , answer reachable
   $R = S_0$ 
  let i = 0
  repeat_2
     $S_{i+1} = \text{Img}'(S_i, C)$ 
    if  $(\text{BMC}_k(S_{i+1}, F) = \text{SAT})$  break repeat_2
     $R' = R \vee S_{i+1}$ 
    if  $R' = R$  answer unreachable
     $R = R'$ 
    increase i
  end repeat_2
  increase k
end repeat_1
```

2. Verification Results

(1) Basic

	gv	gv-ref
a.v	<pre>setup> cirread -v ./design/SoCV/basic/a.v Converted 0 1-valued FFs and 16 DC-valued FFs. setup> set sys vrf vrf> satv itp -o 0 Monitor "z1[0]" is safe. vrf> satv itp -o 1 Monitor "z2[0]" is safe. vrf> satv itp -o 2 Monitor "z3[0]" is safe. vrf> satv itp -o 3 Monitor "z4[0]" is safe. vrf> satv itp -o 4 Monitor "z5[0]" is safe. vrf> quit -f</pre>	<pre>setup> cirread -v ./design/SoCV/basic/a.v Converted 0 1-valued FFs and 16 DC-valued FFs. setup> set sys vrf vrf> satv itp -o 0 Monitor "z1[0]" is safe. vrf> satv itp -o 1 Monitor "z2[0]" is safe. vrf> satv itp -o 2 Monitor "z3[0]" is safe. vrf> satv itp -o 3 Monitor "z4[0]" is safe. vrf> satv itp -o 4 Monitor "z5[0]" is safe. vrf> quit -f</pre>

b.v

```
setup> cirread -v ./design/SoCV/basic/b.v  
Converted 0 1-valued FFs and 10 DC-valued FFs.
```

```
setup> set sys vrf
```

```
vrf> satv itp -o 0
```

```
Monitor "p1[0]" is safe.
```

```
vrf> satv itp -o 1
```

```
Monitor "p2[0]" is safe.
```

```
vrf> satv itp -o 2
```

```
Monitor "p3[0]" is safe.
```

```
vrf> satv itp -o 3
```

```
Monitor "p4[0]" is safe.
```

```
vrf> satv itp -o 4
```

```
Monitor "p5[0]" is violated.
```

```
0: x1  
1: x1  
2: x1  
3: x1  
4: x1  
5: x1  
6: x1  
7: x1  
8: x1  
9: x1  
10: x1  
11: x1  
12: x1  
13: x1  
14: x1  
15: x1  
16: x1  
17: x1  
18: x1  
19: x1  
20: x1  
21: x1  
22: x1  
23: x1  
24: x1  
25: x1  
26: x1  
27: x1  
28: x1  
29: x1  
30: x1  
31: x1  
32: x1  
33: x1  
34: x1  
35: x1  
36: x1  
37: x1  
38: x1  
39: x1  
40: x1  
41: x1  
42: xx
```

```
setup> cirread -v ./design/SoCV/basic/b.v  
Converted 0 1-valued FFs and 10 DC-valued FFs.
```

```
setup> set sys vrf
```

```
vrf> satv itp -o 0
```

```
Monitor "p1[0]" is safe.
```

```
vrf> satv itp -o 1
```

```
Monitor "p2[0]" is safe.
```

```
vrf> satv itp -o 2
```

```
Monitor "p3[0]" is safe.
```

```
vrf> satv itp -o 3
```

```
Monitor "p4[0]" is safe.
```

```
vrf> satv itp -o 4
```

```
Monitor "p5[0]" is violated.
```

```
0: x1  
1: x1  
2: x1  
3: x1  
4: x1  
5: x1  
6: x1  
7: x1  
8: x1  
9: x1  
10: x1  
11: x1  
12: x1  
13: x1  
14: x1  
15: x1  
16: x1  
17: x1  
18: x1  
19: x1  
20: x1  
21: x1  
22: x1  
23: x1  
24: x1  
25: x1  
26: x1  
27: x1  
28: x1  
29: x1  
30: x1  
31: x1  
32: x1  
33: x1  
34: x1  
35: x1  
36: x1  
37: x1  
38: x1  
39: x1  
40: x1  
41: x1  
42: xx
```

C.V	<pre> setup> do ./hw/hw5/tests/c.dofile setup> cirread -v ./design/SoCV/basic/c.v Converted 0 1-valued FFs and 6 DC-valued FFs. setup> set sys vrf vrf> satv itp -o 0 Monitor "z0[0]" is violated. 0: x0 vrf> satv itp -o 1 Monitor "z1[0]" is violated. 0: x0 1: x1 2: x0 3: x1 4: x0 5: xx vrf> satv itp -o 2 Monitor "z2[0]" is violated. 0: x1 1: x1 2: x0 3: x1 4: x0 5: x1 6: xx vrf> satv itp -o 3 Monitor "z3[0]" is safe. vrf> quit -f </pre>	<pre> setup> do ./hw/hw5/tests/c.dofile setup> cirread -v ./design/SoCV/basic/c.v Converted 0 1-valued FFs and 6 DC-valued FFs. setup> set sys vrf vrf> satv itp -o 0 Monitor "z0[0]" is violated. 0: x0 vrf> satv itp -o 1 Monitor "z1[0]" is violated. 0: x0 1: x1 2: x0 3: x1 4: x0 5: xx vrf> satv itp -o 2 Monitor "z2[0]" is violated. 0: x1 1: x1 2: x0 3: x1 4: x0 5: x1 6: xx vrf> satv itp -o 3 Monitor "z3[0]" is safe. vrf> quit -f </pre>
-----	---	---

(2) HWMCC testcases

UNSAT cases

Testcase	Result	Time(s)	Memory usage (MB)
pdtpmfpmult.aig	Monitor "1348" is safe	7.75	35.3
6s6.aig	Monitor "5369" is safe	24.65	48.48
pj2018.aig	Monitor "26898" is safe	67.25	118
6s136.aig	Monitor "25378" is safe	231.7	200.4
6s206rb025.aig	Monitor "141223" is safe	2.44	141
6s221rb18.aig	Monitor "201417" is safe	4.62	181.1
6s326rb02.aig	Monitor "25376" is safe	137.9	174
6s327rb10.aig	Monitor "25050" is safe	0.75	27.39
6s380b129.aig	Monitor "43668" is safe	2.17	43.06

SAT cases

Testcases	Result	Time(s)	Memory usage (MB)
6s307rb06.aig	Monitor "37108" is violated.	3.42	74.95
abp4pold.aig	Monitor "955" is violated.	26.23	93.32
bob9234spec7neg.aig	Monitor "813" is violated.	227.5	48.7
bobpci215.aig	Monitor "4469" is violated.	15.91	40.57
6s374b029.aig	Monitor "326273" is violated.	60.91	279.8
6s388b07.aig	Monitor "34359" is violated.	0.05	19.35

(3) Vending Machine (vending_abs.v)

gv	gv-ref
<pre> tim2811@venv2:~/socv/socv-1122\$./gv setup> cirread -v ./design/SoCV/vending/vending-abs.v Converted 0 1-valued FFs and 20 DC-valued FFs. setup> set sys vrf vrf> satv itp -o 0 Monitor "z0[0]" is safe. vrf> usage Period time used : 0.02 seconds Total time used : 0.02 seconds Peak memory used : 22.06 M Bytes Total memory used : 10.26 M Bytes Current memory used: 48.77 M Bytes vrf> satv itp -o 1 Monitor "z1[0]" is safe. vrf> usage Period time used : 0.01 seconds Total time used : 0.03 seconds Peak memory used : 22.31 M Bytes Total memory used : 10.53 M Bytes Current memory used: 49.04 M Bytes vrf> satv itp -o 2 Monitor "z2[0]" is safe. vrf> usage Period time used : 0.01 seconds Total time used : 0.04 seconds Peak memory used : 22.69 M Bytes Total memory used : 10.92 M Bytes Current memory used: 49.43 M Bytes vrf> quit -f </pre>	<pre> tim2811@venv2:~/socv/socv-1122\$./hw/hw5/gv-ref setup> cirread -v ./design/SoCV/vending/vending-abs.v Converted 0 1-valued FFs and 20 DC-valued FFs. setup> set sys vrf vrf> satv itp -o 0 Monitor "z0[0]" is safe. vrf> usage Period time used : 0.03 seconds Total time used : 0.03 seconds Peak memory used : 21.8 M Bytes Total memory used : 10.26 M Bytes Current memory used: 48.88 M Bytes vrf> satv itp -o 1 Monitor "z1[0]" is safe. vrf> usage Period time used : 0 seconds Total time used : 0.03 seconds Peak memory used : 22.05 M Bytes Total memory used : 10.53 M Bytes Current memory used: 49.16 M Bytes vrf> satv itp -o 2 Monitor "z2[0]" is safe. vrf> usage Period time used : 0.01 seconds Total time used : 0.04 seconds Peak memory used : 22.3 M Bytes Total memory used : 10.8 M Bytes Current memory used: 49.43 M Bytes vrf> quit -f </pre>

3. Comparison with the ref program and other model checker

The results (xxx monitor is safe/violated) are all identical between my implementation gv and gv-ref. Therefore, I only list the time latency and memory usage comparison between gv and gv-ref in the following table:

UNSAT

	gv		gv-ref	
Testcase	Time(s)	Memory usage (MB)	Time(s)	Memory usage (MB)
pdtmpsfpmult.aig	7.75	35.3	2.29	24.73
6s6.aig	24.65	48.48	27.12	48.54
pj2018.aig	67.25	118	4.21	48.96
6s136.aig	231.7	200.4	129.3	171.3
6s206rb025.aig	2.44	141	1.96	118.9
6s221rb18.aig	4.62	181.1	4.41	165.4
6s326rb02.aig	137.9	174	781	320.9
6s327rb10.aig	0.75	27.39	0.62	25.66
6s380b129.aig	2.17	43.06	1.77	44.27

SAT

	gv		gv-ref	
Testcase	Time(s)	Memory usage (MB)	Time(s)	Memory usage (MB)
6s307rb06.aig	3.42	74.95	3.68	71.39
abp4pold.aig	26.23	93.32	19.92	93.57
bob9234spec7neg.aig	227.5	48.7	20.46	18.89
bobpci215.aig	15.91	40.57	14.27	38.95
6s374b029.aig	60.91	279.8	63.95	282.9
6s388b07.aig	0.05	19.35	0.04	17.92

We can see that most of the testcases perform **nearly well** compared to gv-ref in terms of time latency and memory usage, except it takes around 10 times latency in two testcases (pj2018 and bob9234spec7neg). In testcases 6s326rb02, our implementation gv is **5.6 times faster** in time latency and **1.8 times smaller** in memory usage than gv-ref.

I've also try to run itp-ubmc on "vending-fixed.v" from hw1. However, both my gv and gv-ref failed to run correctly with error message "Aborted (core dumped)".

Comparing interpolation-based UBMC with BDD-based method in hw3, we can see that interpolation-based UBMC performs **significantly better** than BDD-based method in terms of total time latency and total memory usage. I use the same “vending_abs.v” file to test the performance of itp-ubmc method and bdd-based method. The following table shows the results:

	gv	gv-ref
ITP-UBMC (hw5)	<pre>vrf> usage Period time used : 0.02 seconds Total time used : 0.02 seconds Peak memory used : 22.69 M Bytes Total memory used : 10.8 M Bytes Current memory used: 49.31 M Bytes</pre>	<pre>vrf> usage Period time used : 0.04 seconds Total time used : 0.04 seconds Peak memory used : 22.3 M Bytes Total memory used : 10.81 M Bytes Current memory used: 49.43 M Bytes</pre>
BDD (hw3)	<pre>vrf> usage Period time used : 0.01 seconds Total time used : 70.33 seconds Peak memory used : 526.8 M Bytes Total memory used : 543.7 M Bytes Current memory used: 582.2 M Bytes</pre>	<pre>vrf> usage Period time used : 0 seconds Total time used : 73.25 seconds Peak memory used : 526.2 M Bytes Total memory used : 543.7 M Bytes Current memory used: 582.1 M Bytes</pre>