EBVI, An Enhanced Bit-Vector Interpolator

Mitra Purandare[1]

Computer Systems Institute, ETH Zurich

1 What is EBVI?

EBVI is an interpolating decision procedure for bit-vector logic. It takes as input a pair (ϕ_1, ϕ_2) of bit-vector formulas and has two possible outcomes: EBVI returns 1. SATISFIABLE if the conjunction $\phi_1 \wedge \phi_2$ is satisfiable 2. with a formula ψ that is a Craig interpolant for (ϕ_1, ϕ_2) and a word-level proof of unsatisfiability.

2 How to use EBVI?

This section describes how to use EBVI.

2.1 Input

EBVI accpets standard Verilog expressions as formulae. A simple example is given in Fig. 1.

```
TYPE: reg [0:5] x,y,z;
x == y && y == z
z != x
```

Fig. 1. Simple Example

The first line declares registers x,y, and z to be bit-vectors of range 0 to 5. Every register declaration should be preceded by "TYPE:".

The last 2 lines are the expressions to be checked for satisfiability. The formula ϕ_1 is the first line following the declaration statements. The formula ϕ_2 is the second line following the declaration statements. For the above example,
ϕ_1 : x == y && y == z
ϕ_2 : z != x

2.2 Command Line Options

For a file expr.txt containing the expressions, the command to run EBVI is as follows:

ebvi -f expr.txt

2.3 Output

EBVI returns with either SAT or UNSAT. For the above example, the output of EBVI is given in Fig. 2.

EBVI also generates two files of with names final_proof and word_facts in the current working directory. These two files together represent the word-level proof. The file with name final_proof contains the propositional proof. The file with name word_facts contains the word-level fact corresponding to the nodes in the proof.

```
Given Expressions: UNSAT
Interpolating expressions::
PART1:: x == y && y == z
PART2:: z != x
interpolant is x == z
```

Fig. 2. Output of EBVI