Our benchmark submission for the SAT 2018 Competition consist of two sets of word-level properties originally formulated as SMT problems in the quantifier-free theory of bit-vectors in BTOR [1] or SMTLIB [2] format. We then use our SMT solver Boolector [3] to synthesize AIGs [4], which in turn were translated to DIMACS format.

DIVISION

The first set specifies word-level (modulo $2^n$) division using multiplication for various bit-widths $n$ in BTOR format [1].

We consider both unsigned and signed dividers. For unsigned division we check validity over unsigned $n$-bit bit-vectors (“$/$” denotes unsigned division):

\[ y \neq 0 \Rightarrow (x - (x / u y) \cdot y) < u y \]

As common in bit-vector logics arithmetic operators take two $n$-bit bit-vectors as input and produce one $n$-bit bit-vector as output, with the effect, that there is no difference between signed and unsigned versions of multiplication nor subtraction.

For signed division it is more complicated and we have to take signs into account (now “$/$” denotes signed division):

\[ y \neq 0 \Rightarrow |x - (x / s y) \cdot y| < u |y| \]

where “$|$” is actually implemented with an if-then-else operator testing the argument to be smaller than zero (using signed “$<$” comparison) and if so negating it (two-complement). These signed benchmarks are as a consequence much harder.

INVERSION

The second set of benchmarks checks that bit-vector multiplication modulo $2^n$ has unique inverses for odd numbers, which translates to the following SMT benchmark for $n = 32$ in SMTLIB format [2]:

```verbatim
(set-logic QF_BV)
(declare-fun x () (_ BitVec 32))
(declare-fun y () (_ BitVec 32))
(declare-fun z () (_ BitVec 32))
(assert (= (bvmul x y) (bvmul x z)))
(assert ((_ extract 0 0) x))
(assert (distinct y z))
(check-sat)
(exit)
```

REFERENCES


