

Termination of the Engel4 Program

Program Verification & Analysis

July 6, 2026

Abstract

This file provides some additional information about the termination proof of the Engel4 program

1 The Engel4 Program

The Engel4 program shown below is based on problem 4 in [1, pp. 9]. Since the program listed in [1, pp. 9] terminates only if the sum of the variables **a** and **b** is a power of 2, the program below computes for a given number **a** the least power of two greater than **a** such that **b** can be determined accordingly:

```
module Engel4(nat ?aIn, event !rdy) {
  nat a,b,lg1,lg2,pow2;
  a = aIn;
  // -----
  // determine the least power of 2 greater than a and its logarithm
  // -----
  lg1 = 0;
  pow2 = 1;
  while(a>=pow2) {
    next(lg1) = lg1+1;
    next(pow2) = pow2*pow2;
    p1:pause;
  }
  b = pow2-a;
  lg2 = 0;
  // -----
  // the following is the core of the program which only terminates
  // if a+b is a power of 2, and a,b>0 holds as ensured by the loop above.
  // We have next(a) == (2*a<pow2 ? 2*a : 2*a-pow2) during the loop.
  // -----
  while(a>0) {
    if(a<b) {
      next(a) = a+a;
      next(b) = b-a;
    } else {
      next(a) = a-b;
      next(b) = b+b;
    }
    next(lg2) = lg2 + 1;
    p2:pause;
  }
  emit(rdy);
}
```

Hence, the above program consists of two parts: The first part stores the given input **aIn** in the local variable **a** and computes then **pow2** as the least power of two which is greater than **a**. In each iteration of the second loop, there are two possible computation paths, and both maintain

the invariant $(a+b) == \text{pow2}$ until termination of the program. To determine a ranking function, we add further variable lg2 to count the number of iterations of the second loop.

Since SMT solvers such as Z3 cannot deal with nonlinear functions, we assume the following axioms about the exponential function:

- (A1) $G(\text{exp}(2,0)==1)$
- (A2a) $G(\text{exp}(2,\text{lg2}+1)==2*\text{exp}(2,\text{lg2}))$
- (A2b) $G(\text{exp}(2,\text{lg1}+1)==2*\text{exp}(2,\text{lg1}))$
- (A3) $G(\text{lg1}<\text{lg2} \rightarrow \text{exp}(2,\text{lg1})<\text{exp}(2,\text{lg2}))$
- (A4) $G(\text{lg2}<\text{lg1} \rightarrow ((2*a) \% \text{exp}(2,\text{lg1})) \% \text{exp}(2,\text{lg2}+1) == (2*a) \% \text{exp}(2,\text{lg2}+1))$

Axiom A4 is thereby a key property which could be reduced further since $\text{lg2}<\text{lg1}$ implies that $\text{exp}(2,\text{lg2}+1) * \text{exp}(2,\text{lg1}-(\text{lg2}+1)) == \text{exp}(2,\text{lg1})$ holds. Since in general $(a\%(b*c))\%c == a\%c$ holds, this implies axiom A4. However, it was not possible to prove this using Z3, so that axiom A4 is stated in the above form. Using these axioms, we can prove the following invariants in the given order:

- (I1) $G(1 \leq \text{pow2})$
- (I2) $G(p1 \rightarrow a > 0)$
- (I3) $G(\text{pow2} == \text{exp}(2,\text{lg1}))$
- (I4) $G(\text{start} \ \& \ a==0 \rightarrow b==1 \ \& \ \text{pow2}==1 \ \& \ (a+b) == \text{pow2} \ \& \ \text{rdy})$
- (I5) $G(p1 \ \& \ a < \text{pow2} \rightarrow 0 < b \ \& \ b < \text{pow2} \ \& \ 0 < a \ \& \ a < \text{pow2} \ \& \ \text{pow2}==a+b \ \& \ \text{lg2}==0 \ \& \ 0 < \text{lg1})$
- (I6) $G(p2 \rightarrow 0 < b \ \& \ b \leq \text{pow2} \ \& \ a < \text{pow2} \ \& \ (a+b) == \text{pow2} \ \& \ (a+b) == \text{pow2})$
- (I7) $G(p1 \ \& \ 0 < a \ \& \ a < \text{pow2} \mid p2 \ \& \ 0 < a \rightarrow \text{next}(a) == (2*a) \% \text{pow2})$
- (I8) $G(0 < a \ \& \ a < \text{pow2} \ \& \ (a \% \text{exp}(2,\text{lg2})) == 0 \rightarrow \text{lg2} < \text{lg1})$
- (I9) $G(p1 \ \& \ 0 < a \ \& \ a < \text{pow2} \rightarrow (a \% \text{exp}(2,\text{lg2})) == 0)$
- (I10) $G(p2 \ \& \ 0 < a \rightarrow (a \% \text{exp}(2,\text{lg2})) == 0)$

The proofs can be simplified by first proving the following equations:

```
G(
  (next(start) <-> false) &
  (next(p1) <-> (start|p1)&pow2<=a ) &
  (next(p2) <-> p1&0<a&a<pow2 | p2&0<a ) &
  (next(a) == (p1&0<a&a<pow2 | p2&0<a ? (a<b ? a+a : a-b) : a) ) &
  (next(a) == (p1&0<a&a<pow2 | p2&0<a ? (2*a<pow2 ? 2*a : 2*a-pow2) : a) ) &
  (start&a==0|p1&0<a&a<pow2 -> b==pow2-a & lg2 == 0) &
  (p1&0<a&a<pow2 | p2&0<a -> next(b) == (a<b ? b-a : 2*b) & next(lg2)==lg2+1) &
  (next(pow2) == ((start|p1)&pow2<=a ? pow2+pow2 : pow2) ) &
  (next(lg1) == ((start|p1)&pow2<=a ? lg1+1 : lg1) ) &
  (rdy <-> (start|p2)&a==0 )
)
```

Invariants (I1-I7) can be easily proved. Due to (I6), the sum $(a+b) == \text{pow2}$ remains invariant, so that variable a is doubled in each iteration modulo pow2 . Hence, if we consider the binary representation of number a , then the bits are shifted to the left losing the bits with weight pow2 . Due to (I1), the second loop can therefore iterate at most lg1 many times (if all bits of the binary representation of a would be 1).

Invariant (I8) is seen as follows: If $\text{lg1} \leq \text{lg2}$ would hold, then $\text{exp}(2,\text{lg1}) \leq \text{exp}(2,\text{lg2}) == \text{pow1}$. Note that we may consider the cases $\text{lg1} > \text{lg2}$ and $\text{lg1} == \text{lg2}$ to prove this using

axiom A3. Moreover, since $a < \text{pow2}$ holds, it would follow $a \% \text{exp}(2, \text{lg2}) == a == 0$ which contradicts the assumption $0 < a$.

Invariant (I9) is proved as follows: if $p1 \ \& \ 0 < a \ \& \ a < \text{pow2}$ holds, then we have $\text{lg2} == 0$, so that by assumption (A1), we have $\text{exp}(2, \text{lg2}) == 1$, and therefore, we have $(a \% \text{exp}(2, \text{lg2})) == (a \% 1) == 0$.

Invariant (I10) is the main invariant and is harder to prove. We prove it by induction where the induction base is trivial, since $p2$ is initially false. For the induction step, we have the following proof goal:

```
0: next(p2) & 0 < next(a),
1: p2 & 0 < a -> (a \% exp(2,lg2)) == 0
|-t (next(a) \% exp(2,next(lg2))) == 0
```

We now prove the a couple of lemmata which is easily done with the available invariants so that we obtain the following proof goal:

```
0 : (2*a) \% exp(2,lg2+1) == 0
1 : (2*a) \% (2*exp(2,lg2)) == 0
2 : lg2 < lg1
3 : a < pow2
4 : next(lg2) == lg2+1
5 : next(a) == (2*a) \% pow2
6 : (a \% exp(2,lg2)) == 0
7 : p1 & 0 < a & a < pow2 | p2 & 0 < a
8 : next(p2) & 0 < next(a)
9 : p2 & 0 < a -> (a \% exp(2,lg2)) == 0
|-t (next(a) \% exp(2,next(lg2))) == 0
```

At this point, we use MpTac (pp " $((2*a) \% \text{exp}(2, \text{lg1})) \% \text{exp}(2, (\text{lg2}+1)) == 0$ ") to reduce the conclusion essentially to axiom A4.

The overall termination of the program follows now by the proven invariants where we use $\text{lg1}-\text{lg2}$ as ranking function for the second loop, and $2*a-\text{pow2}$ as ranking function for the first loop.

References

- [1] ENGEL, A. *Problem-Solving Strategies*. Problem Books in Mathematics. Springer, 1991.