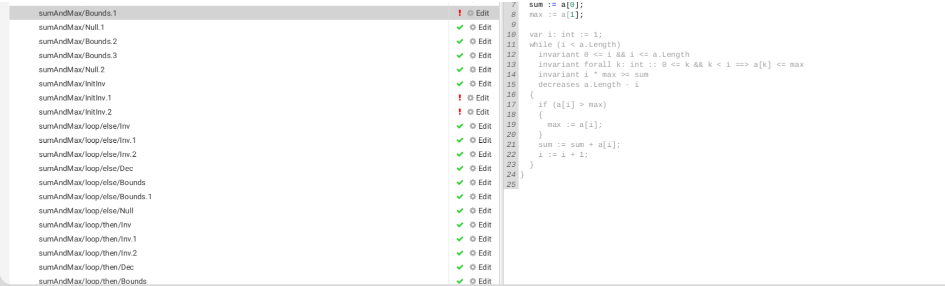
Karlsruher Institut für Technologie

Demo of AlgoVer

Institut für Theoretische Informatik



sumAndMax/Bounds.1	!	⚙ Edit
sumAndMax/Null.1	✓	⚙ Edit
sumAndMax/Bounds.2	✓	⚙ Edit
sumAndMax/Bounds.3	✓	⚙ Edit
sumAndMax/Null.2	✓	⚙ Edit
sumAndMax/InitInv	✓	⚙ Edit
sumAndMax/InitInv.1	!	⚙ Edit
sumAndMax/InitInv.2	!	⚙ Edit
sumAndMax/loop/else/Inv	✓	⚙ Edit
sumAndMax/loop/else/Inv.1	✓	⚙ Edit
sumAndMax/loop/else/Inv.2	✓	⚙ Edit
sumAndMax/loop/else/Dec	✓	⚙ Edit
sumAndMax/loop/else/Bounds	✓	⚙ Edit
sumAndMax/loop/else/Bounds.1	✓	⚙ Edit
sumAndMax/loop/else/Null	✓	⚙ Edit
sumAndMax/loop/then/Inv	✓	⚙ Edit
sumAndMax/loop/then/Inv.1	✓	⚙ Edit
sumAndMax/loop/then/Inv.2	✓	⚙ Edit
sumAndMax/loop/then/Dec	✓	⚙ Edit
sumAndMax/loop/then/Bounds	✓	⚙ Edit

```
  sum := a[0];
  max := a[1];

  var i: int := 1;
  while (i < a.Length)
    invariant 0 <= i && i <= a.Length
    invariant forall k: int :: 0 <= k && k < i ==> a[k] <= max
    invariant i * max >= sum
    decreases a.Length - i
  {
    if (a[i] > max)
    {
      max := a[i];
    }
    sum := sum + a[i];
    i := i + 1;
  }
}

```

Interactive Program Verification

AlgoVer: Seamless Program Verification

- Interactive program verification for Dafny
- Symbolic execution (à la KeY)
- Innovative interaction concepts
- Calling z3
- Bridging the gap between code and proof

Demo

DEMO

Features And Future Work

Features:

- Inspection of different parts of the proof in individual views
- Seamless transition between different views
- Proof construction using
    - direct manipulation
    - script based
    - annotation based

Future Work:

- More rules and strategies
- Improved SMT-support
- Feedback/Interaction back to annotations in source code
- Evaluation on larger examples
- ...