
MODULE *SetEuclid*

EXTENDS *Integers, GCD, FiniteSets*

```

*****
--fair algorithm SetEuclid{
variables S = {2, 4, 6};
{ while ( Cardinality(S) > 1 ) { with ( x ∈ S, y ∈ {s ∈ S : s > x} )
                                { S := (S \ {y}) ∪ {y - x} }
                                } ;
}
}
*****

```

```

BEGIN TRANSLATION
VARIABLES S, pc

vars ≜ ⟨S, pc⟩

Init ≜ Global variables
      ∧ S = {2, 4, 6}
      ∧ pc = "Lbl_1"

Lbl_1 ≜ ∧ pc = "Lbl_1"
      ∧ IF Cardinality(S) > 1
      THEN ∧ ∃ x ∈ S :
            ∃ y ∈ {s ∈ S : s > x} :
            S' = ((S \ {y}) ∪ {y - x})
            ∧ pc' = "Lbl_1"
      ELSE ∧ pc' = "Done"
            ∧ S' = S

Next ≜ Lbl_1
      ∨ Disjunct to prevent deadlock on termination
      (pc = "Done" ∧ UNCHANGED vars)

Spec ≜ ∧ Init ∧ □[Next]vars
      ∧ WFvars(Next)

Termination ≜ ◇(pc = "Done")

END TRANSLATION

```

\ * Modification History
 \ * Last modified Wed Jun 04 10:56:57 CST 2014 by yaojingguo
 \ * Created Wed Jun 04 10:23:24 CST 2014 by yaojingguo