Benchmark for pilat

Steven de Oliveira

Abstract. The following document is a compilation of some of the benchmarks we performed with the tool PILAT. Annotations are generated by our tool, where were none before the generation.

1 test/cohencu.c

Degree : 2

```
/* Generated by Frama-C */
int cohencu(void)
{
  /*@ ghost float k4; */
  /*@ ghost float k3; */
  /*@ ghost float k2; */
  /*@ ghost float k1; */
  int __retres;
  int N;
  int n;
  int x;
  int y;
  int z;
  n = 0;
  x = 0;
  y = 1;
  z = 6;
  /*@ ghost k4 = (-12 * (z * n) + 36 * (n * n)) + 1 * (z * z);
    */
  /*@ ghost k3 = -6 * n + 1 * z; */
  /*@ ghost k2 = ((3 * n + -1 * (z * n)) + 3 * (n * n)) + 1 *
    y; */
  /*@ ghost k1 = 1 * 1; */
  /*@ loop invariant Pilat_emitter: 1*1 == k1;
      loop invariant Pilat_emitter: ((3*n+-1*(z*n))+3*(n*n))
    +1*y == k2;
      loop invariant Pilat_emitter: -6*n+1*z == k3;
      loop invariant Pilat_emitter: (-12*(z*n)+36*(n*n))+1*(z*
    z) == k4;
  */
  while (n <= N) {
    n ++;
```

```
29      x += y;
        y += z;
31      z += 6;
    }
33  ;
    __retres = 0;
35  return __retres;
}
```

2 test/cohendiv.c

Degree : 2

```
/* Generated by Frama-C */
int cohendiv(int x, int y)
{
4   /*@ ghost float k11; */
    /*@ ghost float k10; */
6   /*@ ghost float k9; */
    /*@ ghost float k8; */
8   /*@ ghost float k7; */
    /*@ ghost float k6; */
10  /*@ ghost float k5; */
    /*@ ghost float k4; */
12  /*@ ghost float k3; */
    /*@ ghost float k2; */
14  /*@ ghost int k1; */
    int q;
16  int r;
    q = 0;
18  r = x;
    /*@ ghost k5 = 1 * 1; */
20  /*@ ghost k4 = 1 * y; */
    /*@ ghost k3 = 1 * (y * y); */
22  /*@ ghost k2 = 1 * (q * y) + 1 * r; */
    /*@ loop invariant Pilat_emitter: k1*(-1*(d*y)+1*dd) == 0;
24      loop invariant Pilat_emitter: 1*(q*y)+1*r == k2;
        loop invariant Pilat_emitter: 1*(y*y) == k3;
26      loop invariant Pilat_emitter: 1*y == k4;
        loop invariant Pilat_emitter: 1*1 == k5;
28  */
    while (r >= y) {
30    int d;
      int dd;
32    d = 1;
      dd = y;
34    /*@ ghost k11 = 1 * 1; */
```

```
    /*@ ghost k10 = 1 * dd; */
    /*@ ghost k9 = 1 * d; */
    /*@ ghost k8 = 1 * (dd * dd); */
    /*@ ghost k7 = 1 * (d * d); */
    /*@ ghost k6 = 1 * (dd * d); */
    /*@ loop invariant Pilat_emitter: 1*(dd*d) >= k6;
        loop invariant Pilat_emitter: 1*(d*d) >= k7;
        loop invariant Pilat_emitter: 1*(dd*dd) >= k8;
        loop invariant Pilat_emitter: 1*d >= k9;
        loop invariant Pilat_emitter: 1*dd >= k10;
        loop invariant Pilat_emitter: 1*1 == k11;
    */
    while (r >= 2 * dd) {
      d = 2 * d;
      dd = 2 * dd;
    }
    r -= dd;
    q += d;
  }
  return r;
}
```

3 test/couzot.c

Degree : 2

```
/* Generated by Frama-C */
int couzot(void)
{
  /*@ ghost float k1; */
  int __retres;
  int x;
  int y;
  x = 0;
  y = 0;
  /*@ ghost k1 = 1 * 1; */
  /*@ loop invariant Pilat_emitter: 1*1 == k1; */
  while (1) {
    if (x) {
      x += 2;
      y ++;
    }
    else {
      x += 4;
      y = y;
    }
  }
```

```
  __retres = 0;
  return __retres;
}
```

4 test/cplx_ex.c

Degree : 2

```
/* Generated by Frama-C */
int cplx_ex(void)
{
  /*@ ghost float k4; */
  /*@ ghost float k3; */
  /*@ ghost float k2; */
  /*@ ghost float k1; */
  int __retres;
  int x;
  int y;
  /*@ ghost k4 = -2 * (x * x) + 1 * (y * y); */
  /*@ ghost k3 = 1 * 1; */
  /*@ ghost k2 = 2 * (x * x) + 1 * (y * y); */
  /*@ ghost k1 = 1 * (y * x); */
  /*@ loop invariant Pilat_emitter: 1*(y*x) >= k1;
      loop invariant Pilat_emitter: 2*(x*x)+1*(y*y) >= k2;
      loop invariant Pilat_emitter: 1*1 == k3;
      loop invariant Pilat_emitter: -2*(x*x)+1*(y*y) >= k4;
  */
  while (1) {
    x = y + x;
    y = 2 * x - 2 * y;
    x -= y / 2;
  }
  __retres = 0;
  return __retres;
}
```

5 test/disj.c

Degree : 2

```
/* Generated by Frama-C */
void disj(void)
{
  /*@ ghost float k1; */
```

```
  int x;
  int y;
  x = 0;
  y = 50;
  /*@ ghost k1 = 1 * 1; */
  /*@ loop invariant Pilat_emitter: 1*1 == k1; */
  while (x < 100)
    if (x < 50) x ++;
    else {
      x ++;
      y ++;
    }
  return;
}
```

6 test/divbin.c

Degree : 2

```
/* Generated by Frama-C */
int divbin(void)
{
  /*@ ghost float k4; */
  /*@ ghost float k3; */
  /*@ ghost float k2; */
  /*@ ghost float k1; */
  int __retres;
  int A;
  int B;
  int q;
  int r;
  int b;
  q = 0;
  r = A;
  b = B;
  /*@ ghost k4 = -1 * (b * q) + -1 * r; */
  /*@ ghost k3 = -1 * 1; */
  /*@ ghost k2 = -1 * b; */
  /*@ ghost k1 = -1 * (b * b); */
  /*@ loop invariant Pilat_emitter: -1*(b*b) <= k1;
      loop invariant Pilat_emitter: -1*b <= k2;
      loop invariant Pilat_emitter: -1*1 == k3;
      loop invariant Pilat_emitter: -1*(b*q)+-1*r == k4;
  */
  while (1) {
    if (A) {
      q = 2 * q;
```

```
        b /= 2;
30      if (r >= b) {
          q ++;
32        r -= b;
        }
34    }
  }
36  __retres = 0;
  return __retres;
38}
```

7 test/djikstra.c

Degree : 2

```
/* Generated by Frama-C */
2int djikstra(void)
{
4  /*@ ghost float k5; */
  /*@ ghost float k4; */
6  /*@ ghost float k3; */
  /*@ ghost float k2; */
8  /*@ ghost int k1; */
  int n;
10  int p;
  int q;
12  int r;
  int h;
14  p = 0;
  q = 1;
16  r = n;
  /*@ ghost k5 = 1 * 1; */
18  /*@ ghost k4 = 1 * q; */
  /*@ ghost k3 = 1 * (p * p) + 1 * (r * q); */
20  /*@ ghost k2 = 1 * (q * q); */
  /*@ loop invariant
22        Pilat_emitter: k1*(((-4*(h*p)+4*(p*p))+-1*(q*q))+1*(h*
    h)) == 0;
      loop invariant Pilat_emitter: 1*(q*q) <= k2;
24    loop invariant Pilat_emitter: 1*(p*p)+1*(r*q) <= k3;
      loop invariant Pilat_emitter: 1*q <= k4;
26    loop invariant Pilat_emitter: 1*1 == k5;
  */
28  while (q != 1) {
    r = r;
30    q /= 4;
    h = p + q;
```

```
32      p /= 2;
        if (r >= h) {
34        p += q;
          r -= h;
36      }
    }
38  return p;
}
```

8 test/eucli_div.c

Degree : 2

```
1  /* Generated by Frama-C */
   int eucli_div(void)
3  {
     /*@ ghost float k4; */
5    /*@ ghost float k3; */
     /*@ ghost float k2; */
7    /*@ ghost float k1; */
     int __retres;
9    int x;
     int y;
11   int q;
     q = 0;
13   /*@ ghost k4 = 1 * (y * y); */
     /*@ ghost k3 = 1 * x + 1 * (q * y); */
15   /*@ ghost k2 = 1 * y; */
     /*@ ghost k1 = 1 * 1; */
17   /*@ loop invariant Pilat_emitter: 1*1 == k1;
         loop invariant Pilat_emitter: 1*y == k2;
19       loop invariant Pilat_emitter: 1*x+1*(q*y) == k3;
         loop invariant Pilat_emitter: 1*(y*y) == k4;
21   */
     while (1) {
23     x -= y;
       q ++;
25   }
     __retres = 0;
27   return __retres;
}
```

9 test/fermat2.c

Degree : 2

```
/* Generated by Frama-C */
int fermat2(void)
{
  /*@ ghost float k2; */
  /*@ ghost float k1; */
  int __retres;
  int N;
  int R;
  int u;
  int v;
  int r;
  u = 2 * R + 1;
  v = 1;
  r = R * R - N;
  /*@ ghost k2 = 1 * 1; */
  /*@ ghost k1 = (((4 * u + -2 * (u * u)) + -4 * v) + 2 * (v *
     v)) + 8 * r;

    */
  /*@ loop invariant
        Pilat_emitter: (((4*u+-2*(u*u))+-4*v)+2*(v*v))+8*r ==
    k1;
      loop invariant Pilat_emitter: 1*1 == k2;
  */
  while (r != 0)
    if (r > 0) {
      r -= v;
      v += 2;
      u = u;
    }
    else {
      r += u;
      u += 2;
      v = v;
    }
  __retres = (u - v) / 2;
  return __retres;
}
```

10 test/fibonacci.c

Degree : 2

```
/* Generated by Frama-C */
int fibonacci(void)
{
  /*@ ghost float k2; */
  /*@ ghost float k1; */
  int __retres;
  int xn1;
  int xn;
  /*@ ghost k2 = (1 * (xn * xn1) + -1 * (xn1 * xn1)) + 1 * (xn
    * xn); */
  /*@ ghost k1 = 1 * 1; */
  /*@ loop invariant Pilat_emitter: 1*1 == k1;
      loop invariant
        Pilat_emitter: (1*(xn*xn1)+-1*(xn1*xn1))+1*(xn*xn) ==
    k2;
  */
  while (1) {
    xn1 += xn;
    xn = xn1 - xn;
  }
  __retres = 0;
  return __retres;
}
```

11 test/forward.c

Degree : 2

```
/* Generated by Frama-C */
void forward(int n)
{
  /*@ ghost float k3; */
  /*@ ghost float k2; */
  /*@ ghost float k1; */
  int i;
  int a;
  int b;
  i = 0;
  a = 0;
  b = 0;
  /*@ ghost k3 = 1 * 1; */
  /*@ ghost k2 = (-3 * i + 1 * a) + 1 * b; */
  /*@ ghost k1 = ((((-6 * (a * i) + -6 * (b * i)) + 9 * (i * i
    )) + 2 * (

          b * a)) +
```

```
                    1 * (a * a)) + 1 * (b * b);

     */
  /*@ loop invariant
        Pilat_emitter:
          ((((-6*(a*i)+-6*(b*i))+9*(i*i))+2*(b*a))+1*(a*a))
    +1*(b*b) == k1;
      loop invariant Pilat_emitter: (-3*i+1*a)+1*b == k2;
      loop invariant Pilat_emitter: 1*1 == k3;
  */
  while (i < n) {
    if (a) {
      a ++;
      b += 2;
    }
    else {
      a += 2;
      b ++;
    }
    i ++;
  }
  return;
}
```

12 test/gcd2.c

Degree : 2

```
/* Generated by Frama-C */
int gcd2(void)
{
  /*@ ghost float k4; */
  /*@ ghost float k3; */
  /*@ ghost float k2; */
  /*@ ghost float k1; */
  int x;
  int y;
  int a;
  int b;
  int p;
  int q;
  int r;
  int s;
  a = x;
  b = y;
  p = 1;
  q = 0;
```

```
20    r = 0;
      s = 1;
22    /*@ ghost k4 = 1 * 1; */
      /*@ ghost k3 = -1 * (q * a) + 1 * (p * b); */
24    /*@ ghost k2 = -1 * (s * a) + 1 * (r * b); */
      /*@ ghost k1 = -1 * (s * p) + 1 * (r * q); */
26    /*@ loop invariant Pilat_emitter: -1*(s*p)+1*(r*q) == k1;
          loop invariant Pilat_emitter: -1*(s*a)+1*(r*b) == k2;
28        loop invariant Pilat_emitter: -1*(q*a)+1*(p*b) == k3;
          loop invariant Pilat_emitter: 1*1 == k4;
30    */
      while (a != b)
32      if (a > b) {
          a -= b;
34        p -= q;
          r -= s;
36      }
        else {
38        b -= a;
          q -= p;
40        s -= r;
        }
42    return a;
}
```

13 test/gcd_nested.c

Degree : 2

```
1  /* Generated by Frama-C */
   int gcd2(void)
3  {
     /*@ ghost float k4; */
5    /*@ ghost float k3; */
     /*@ ghost float k2; */
7    /*@ ghost float k1; */
     int x;
9    int y;
     int a;
11   int b;
     int p;
13   int q;
     int r;
15   int s;
     a = x;
17   b = y;
     p = 1;
```

```
19    q = 0;
      r = 0;
21    s = 1;
      /*@ ghost k4 = 1 * 1; */
23    /*@ ghost k3 = -1 * (q * a) + 1 * (p * b); */
      /*@ ghost k2 = -1 * (s * a) + 1 * (r * b); */
25    /*@ ghost k1 = -1 * (s * p) + 1 * (r * q); */
      /*@ loop invariant Pilat_emitter: -1*(s*p)+1*(r*q) == k1;
27        loop invariant Pilat_emitter: -1*(s*a)+1*(r*b) == k2;
          loop invariant Pilat_emitter: -1*(q*a)+1*(p*b) == k3;
29        loop invariant Pilat_emitter: 1*1 == k4;
      */
31    while (a != b)
        if (a > b) {
33        a -= b;
          p -= q;
35        r -= s;
        }
37      else {
          b -= a;
39        q -= p;
          s -= r;
41      }
      return a;
43  }
```

14 test/hard.c

Degree : 2

```
1   /* Generated by Frama-C */
    int hard(void)
3   {
      /*@ ghost float k7; */
5     /*@ ghost float k6; */
      /*@ ghost float k5; */
7     /*@ ghost float k4; */
      /*@ ghost float k3; */
9     /*@ ghost float k2; */
      /*@ ghost float k1; */
11    int __retres;
      int N;
13    int D;
      int r;
15    int ds;
      int p;
17    int q;
```

```
  r = N;
  ds = D;
  p = 1;
  q = 0;
  /*@ ghost k7 = 1 * 1; */
  /*@ ghost k6 = 1 * ds; */
  /*@ ghost k5 = 1 * (p * r) + 1 * (q * ds); */
  /*@ ghost k4 = 1 * p; */
  /*@ ghost k3 = 1 * (p * ds); */
  /*@ ghost k2 = 1 * (ds * ds); */
  /*@ ghost k1 = 1 * (p * p); */
  /*@ loop invariant Pilat_emitter: 1*(p*p) <= k1;
      loop invariant Pilat_emitter: 1*(ds*ds) <= k2;
      loop invariant Pilat_emitter: 1*(p*ds) <= k3;
      loop invariant Pilat_emitter: 1*p <= k4;
      loop invariant Pilat_emitter: 1*(p*r)+1*(q*ds) <= k5;
      loop invariant Pilat_emitter: 1*ds <= k6;
      loop invariant Pilat_emitter: 1*1 == k7;
  */
  while (p != 1) {
    ds /= 2;
    p /= 2;
    if (r >= ds) {
      r -= ds;
      q += p;
    }
  }
  __retres = 0;
  return __retres;
}
```

15 test/hybrid_automata.c

Degree : 2

```
/* Generated by Frama-C */
int hybridautomata(void)
{
  /*@ ghost float k10; */
  /*@ ghost float k9; */
  /*@ ghost float k8; */
  /*@ ghost float k7; */
  /*@ ghost float k6; */
  /*@ ghost float k5; */
  /*@ ghost float k4; */
  /*@ ghost int k3; */
  /*@ ghost int k2; */
```

```
13    /*@ ghost int k1; */
      int __retres;
15    float x;
      float y;
17    /*@ ghost k4 = 1. * 1; */
      /*@ loop invariant
19          Pilat_emitter: (k1*(1.*(y*x))+k2*(0.*(y*x)+1.*y))+k3
       *(1.*(y*y)) == 0;
21        loop invariant Pilat_emitter: 1.*1 == k4;
      */
23    while (1) {
        /*@ ghost k7 = (-2. * (y * x) + 1. * (x * x)) + 1. * (y *
       y); */
25      /*@ ghost k6 = -1. * x + 1. * y; */
        /*@ ghost k5 = 1. * 1; */
27      /*@ loop invariant Pilat_emitter: 1.*1 == k5;
            loop invariant Pilat_emitter: -1.*x+1.*y == k6;
29          loop invariant Pilat_emitter: (-2.*(y*x)+1.*(x*x))
       +1.*(y*y) == k7;
        */
31      while (x < (float)5) {
          x = (float)((double)x + 0.25);
33        y = (float)((double)y + 0.25);
          if (x > (float)4) break;
35      }
        x = (float)0;
37      /*@ ghost k10 = (-2. * (y * x) + 1. * (x * x)) + 1. * (y *
       y); */
        /*@ ghost k9 = -1. * x + 1. * y; */
39      /*@ ghost k8 = 1. * 1; */
        /*@ loop invariant Pilat_emitter: 1.*1 == k8;
41          loop invariant Pilat_emitter: -1.*x+1.*y == k9;
            loop invariant Pilat_emitter: (-2.*(y*x)+1.*(x*x))
       +1.*(y*y) == k10;
        */
43      while (y < (float)10) {
          x = (float)((double)x + 0.25);
45        y = (float)((double)y + 0.25);
          if (y > (float)9) break;
47      }
        y = (float)0;
49    }
      return __retres;
51  }
```

16 test/illinois.c

Degree : 1

```
/* Generated by Frama-C */
int illinois(void)
{
  /*@ ghost float k2; */
  /*@ ghost float k1; */
  int __retres;
  int dirty;
  int shared;
  int exclusive;
  int invalid;
  exclusive = 0;
  dirty = 0;
  shared = 0;
  /*@ ghost k2 = 1 * 1; */
  /*@ ghost k1 = ((1 * dirty + 1 * shared) + 1 * exclusive) +
    1 * invalid; */
  /*@ loop invariant
        Pilat_emitter: ((1*dirty+1*shared)+1*exclusive)+1*
    invalid == k1;
      loop invariant Pilat_emitter: 1*1 == k2;
  */
  while (1) {
    exclusive = exclusive;
    dirty = dirty;
    shared = shared;
    invalid = invalid;
    if (shared) {
      invalid --;
      exclusive ++;
    }
    else
      if (shared) {
        invalid --;
        dirty --;
        shared += 2;
      }
      else
        if (shared) {
          invalid --;
          shared = (shared + exclusive) + 1;
          exclusive = 0;
        }
        else
          if (shared) {
            invalid --;
            shared = (shared + exclusive) + 1;
```

```
45              exclusive = 0;
            }
47          else
              if (shared) {
49              exclusive --;
                dirty ++;
51            }
              else
53              if (shared) {
                  invalid = (invalid + shared) - 1;
55                dirty ++;
                  shared = 0;
57              }
                else
59                if (shared) {
                    invalid = (((invalid + exclusive) + dirty) +
    shared) - 1;
61                  exclusive = 0;
                    shared = 0;
63                  dirty = 1;
                  }
65                else
                    if (shared) {
67                    dirty --;
                      invalid ++;
69                  }
                    else
71                    if (shared) {
                        shared --;
73                      invalid ++;
                      }
75                    else {
                        exclusive --;
77                      invalid ++;
                      }
79  }
  return __retres;
81 }
```

17 test/knuth.c

Degree : 3

```
1 /* Generated by Frama-C */
int knuth(void)
3 {
  /*@ ghost int k36; */
```

```
  /*@ ghost int k35; */
  /*@ ghost int k34; */
  /*@ ghost int k33; */
  /*@ ghost int k32; */
  /*@ ghost int k31; */
  /*@ ghost int k30; */
  /*@ ghost int k29; */
  /*@ ghost int k28; */
  /*@ ghost int k27; */
  /*@ ghost int k26; */
  /*@ ghost int k25; */
  /*@ ghost int k24; */
  /*@ ghost int k23; */
  /*@ ghost int k22; */
  /*@ ghost int k21; */
  /*@ ghost int k20; */
  /*@ ghost int k19; */
  /*@ ghost int k18; */
  /*@ ghost int k17; */
  /*@ ghost int k16; */
  /*@ ghost float k15; */
  /*@ ghost int k14; */
  /*@ ghost float k13; */
  /*@ ghost float k12; */
  /*@ ghost float k11; */
  /*@ ghost int k10; */
  /*@ ghost float k9; */
  /*@ ghost int k8; */
  /*@ ghost float k7; */
  /*@ ghost float k6; */
  /*@ ghost float k5; */
  /*@ ghost float k4; */
  /*@ ghost float k3; */
  /*@ ghost float k2; */
  /*@ ghost float k1; */
  int n;
  int r;
  int rp;
  int q;
  int d;
  int s;
  int t;
  n = 11;
  r = 1;
  rp = 1;
  q = 1;
  d = 3;
  s = 1;
  t = 1;
  d *= d;
```

```
55  r = n % d;
    rp = n % (d - 2);
57  q = 4 * (n / (d - 2) - n / d);
    /*@ ghost k15 = 1. * 1; */
59  /*@ ghost k13 = 1. * 1; */
    /*@ ghost k12 = 1. * 1; */
61  /*@ ghost k11 = 1. * 1; */
    /*@ ghost k9 = 1. * 1; */
63  /*@ ghost k7 = 1. * 1; */
    /*@ ghost k6 = 1. * 1; */
65  /*@ ghost k5 = (((8. * r + -4. * (d * r)) + 4. * (d * rp)) +
       -2. * (d * q)) +
                  1. * (d * (d * q));
67
      */
69  /*@ ghost k4 = 1. * 1; */
    /*@ ghost k3 = (((8. * r + -4. * (d * r)) + 4. * (d * rp)) +
       -2. * (d * q)) +
71                1. * (d * (d * q));

73    */
    /*@ ghost k2 = 1. * 1; */
75  /*@ ghost k1 = (((8. * r + -4. * (d * r)) + 4. * (d * rp)) +
       -2. * (d * q)) +
                  1. * (d * (d * q));
77
      */
79  /*@ loop invariant
          Pilat_emitter:
81          (((8.*r+-4.*(d*r))+4.*(d*rp))+-2.*(d*q))+1.*(d*(d*q)
    ) >= k1;
        loop invariant Pilat_emitter: 1.*1 >= k2;
83      loop invariant
          Pilat_emitter:
85          (((8.*r+-4.*(d*r))+4.*(d*rp))+-2.*(d*q))+1.*(d*(d*q)
    ) <= k3;
        loop invariant Pilat_emitter: 1.*1 <= k4;
87      loop invariant
          Pilat_emitter:
89          (((8.*r+-4.*(d*r))+4.*(d*rp))+-2.*(d*q))+1.*(d*(d*q)
    ) == k5;
        loop invariant Pilat_emitter: 1.*1 == k6;
91      loop invariant Pilat_emitter: 1.*1 >= k7;
        loop invariant Pilat_emitter: k8*(1.*1) == 0;
93      loop invariant Pilat_emitter: 1.*1 >= k9;
        loop invariant Pilat_emitter: k10*(1.*1) == 0;
95      loop invariant Pilat_emitter: 1.*1 <= k11;
        loop invariant Pilat_emitter: 1.*1 <= k12;
97      loop invariant Pilat_emitter: 1.*1 >= k13;
        loop invariant Pilat_emitter: k14*(1.*1) == 0;
```

```
        loop invariant Pilat_emitter: 1.*1 >= k15;
        loop invariant
          Pilat_emitter:
            ((((((((((((((((((k16*(-1.*(t*(t*rp))+1.*(t*(t*t)))
    +k17*(-1.*

            (t*rp)+1.*

            (t*t)))+k18*
                                    (-1.*rp+1.*t))+k19*(-1.*(d*(d*rp)
    )+1.*(t*(d*d))))+k20*
                                 (-1.*(t*(d*rp))+1.*(t*(t*d))))+k21
    *(-1.*(d*rp)+1.*

        (t*d)))+k22*
                              (-1.*(q*(q*rp))+1.*(t*(q*q))))+k23
    *(-1.*(t*(q*rp))+1.*

    (t*(t*q))))+k24*
                          (-1.*(q*rp)+1.*(t*q)))+k25*(-1.*(d*(q*
    rp))+1.*
                                                       (t*(d*q)))
    )+k26*(-1.*

            (t*

            (t*rp))+1.*

            (rp*

            (rp*rp))))+k27*
                       (-1.*(t*(t*rp))+1.*(t*(rp*rp))))+k28
    *(-1.*(t*(d*rp))+1.*
                                                              (d
    *(rp*rp))))+k29*
                    (-1.*(t*(q*rp))+1.*(q*(rp*rp))))+k30*(-1.*(t
    *rp)+1.*
                                                            (rp*
    rp)))+k31*
                  (-1.*(rp*(r*r))+1.*(t*(r*r))))+k32*(-1.*(t*(rp
    *r))+1.*
                                                        (t*(t*r))
    ))+k33*
                (-1.*(rp*r)+1.*(t*r)))+k34*(-1.*(d*(rp*r))+1.*(t
    *(d*r))))+k35*
              (-1.*(q*(rp*r))+1.*(t*(q*r))))+k36*(-1.*(t*(rp*r))
    +1.*(rp*(rp*r)))
            == 0;
   */
   while (1) {
```

```
127     if (s >= d) {
          if (! (r != 0)) break;
129     }
        else break;
131     if ((2 * r - rp) + q < 0) {
          t = r;
133       r = (((2 * r - rp) + q) + d) + 2;
          rp = t;
135       q += 4;
          d += 2;
137     }
        else
139       if ((2 * r - rp) + q >= 0) {
            if ((2 * r - rp) + q < d + 2) {
141           t = r;
              r = (2 * r - rp) + q;
143           rp = t;
              d += 2;
145         }
            else goto _LAND_1;
147       }
          else {
149         _LAND_1: ;
            if ((2 * r - rp) + q >= 0) {
151           if ((2 * r - rp) + q >= d + 2) {
                if ((2 * r - rp) + q < 2 * d + 4) {
153               t = r;
                  r = (((2 * r - rp) + q) - d) - 2;
155               rp = t;
                  q -= 4;
157               d += 2;
                }
159             else goto _LAND_0;
              }
161           else goto _LAND_0;
            }
163         else {
              _LAND_0:
165             t = r;
                r = (((2 * r - rp) + q) - 2 * d) - 4;
167             rp = t;
                q -= 8;
169             d += 2;
            }
171       }
    }
173 return d;
}
```

18 test/lcm1.c

Degree : 2

```
/* Generated by Frama-C */
int lcm1(int a, int b)
{
  /*@ ghost float k16; */
  /*@ ghost float k15; */
  /*@ ghost float k14; */
  /*@ ghost float k13; */
  /*@ ghost float k12; */
  /*@ ghost float k11; */
  /*@ ghost float k10; */
  /*@ ghost float k9; */
  /*@ ghost float k8; */
  /*@ ghost float k7; */
  /*@ ghost float k6; */
  /*@ ghost float k5; */
  /*@ ghost float k4; */
  /*@ ghost float k3; */
  /*@ ghost float k2; */
  /*@ ghost float k1; */
  int __retres;
  int x;
  int y;
  int u;
  int v;
  x = a;
  y = b;
  u = b;
  v = 0;
  /*@ ghost k2 = 1 * 1; */
  /*@ ghost k1 = 1 * (u * x) + 1 * (v * y); */
  /*@ loop invariant Pilat_emitter: 1*(u*x)+1*(v*y) == k1;
      loop invariant Pilat_emitter: 1*1 == k2;
  */
  while (x != y) {
    /*@ ghost k9 = 1 * (u * u); */
    /*@ ghost k8 = 1 * u; */
    /*@ ghost k7 = 1 * (y * y); */
    /*@ ghost k6 = 1 * (u * x) + 1 * (v * y); */
    /*@ ghost k5 = 1 * (u * y); */
    /*@ ghost k4 = 1 * y; */
    /*@ ghost k3 = 1 * 1; */
    /*@ loop invariant Pilat_emitter: 1*1 == k3;
        loop invariant Pilat_emitter: 1*y == k4;
        loop invariant Pilat_emitter: 1*(u*y) == k5;
        loop invariant Pilat_emitter: 1*(u*x)+1*(v*y) == k6;
        loop invariant Pilat_emitter: 1*(y*y) == k7;
```

```
        loop invariant Pilat_emitter: 1*u == k8;
        loop invariant Pilat_emitter: 1*(u*u) == k9;
    */
    while (x > y) {
      x -= y;
      v += u;
    }
    /*@ ghost k16 = 1 * (v * v); */
    /*@ ghost k15 = 1 * v; */
    /*@ ghost k14 = 1 * (u * x) + 1 * (v * y); */
    /*@ ghost k13 = 1 * (x * x); */
    /*@ ghost k12 = 1 * (v * x); */
    /*@ ghost k11 = 1 * x; */
    /*@ ghost k10 = 1 * 1; */
    /*@ loop invariant Pilat_emitter: 1*1 == k10;
        loop invariant Pilat_emitter: 1*x == k11;
        loop invariant Pilat_emitter: 1*(v*x) == k12;
        loop invariant Pilat_emitter: 1*(x*x) == k13;
        loop invariant Pilat_emitter: 1*(u*x)+1*(v*y) == k14;
        loop invariant Pilat_emitter: 1*v == k15;
        loop invariant Pilat_emitter: 1*(v*v) == k16;
    */
    while (x < y) {
      y -= x;
      u += v;
    }
  }
  __retres = u + v;
  return __retres;
}
```

19 test/lcm2.c

Degree : 2

```
/* Generated by Frama-C */
int lcm2(void)
{
  /*@ ghost float k2; */
  /*@ ghost float k1; */
  int __retres;
  int a;
  int b;
  int x;
  int y;
  int u;
  int v;
```

```
  x = a;
  y = b;
  u = b;
  v = a;
  /*@ ghost k2 = 1 * 1; */
  /*@ ghost k1 = 1 * (u * x) + 1 * (v * y); */
  /*@ loop invariant Pilat_emitter: 1*(u*x)+1*(v*y) == k1;
      loop invariant Pilat_emitter: 1*1 == k2;
  */
  while (x != y)
    if (x > y) {
      x -= y;
      v += u;
    }
    else {
      y -= x;
      u += v;
    }
  __retres = (u + v) / 2;
  return __retres;
}
```

20 test/mannadiv.c

Degree : 1

```
/* Generated by Frama-C */
int mannadiv(void)
{
  /*@ ghost float k1; */
  int __retres;
  int x1;
  int x2;
  int y1;
  int y2;
  int y3;
  y1 = 0;
  y2 = 0;
  y3 = x1;
  /*@ ghost k1 = -1 * 1; */
  /*@ loop invariant Pilat_emitter: -1*1 == k1; */
  while (1) {
    if (y2 + 1 == x2) {
      y1 ++;
      y2 = 0;
      y3 --;
    }
```

```
    else {
      y2 ++;
      y3 --;
    }
  }
  __retres = 0;
  return __retres;
}
```

21 test/mesi.c

Degree : 1

```
/* Generated by Frama-C */
int mesi(void)
{
  /*@ ghost int k2; */
  /*@ ghost float k1; */
  int __retres;
  int s;
  int e;
  int m;
  int i;
  int N;
  s = 0;
  e = 0;
  m = 0;
  i = 0;
  N = 30;
  /*@ ghost k1 = 1 * 1; */
  /*@ loop invariant Pilat_emitter: 1*1 == k1;
      loop invariant Pilat_emitter: k2*(((-1*1+1*s)+1*e)+1*m)
    == 0;
  */
  while (N > 0) {
    N --;
    if (i != 0) {
      s = (s + e) + m;
      i --;
      e = 0;
      m = 0;
    }
    else
      if (e != 0) {
        s = s;
        i = i;
        e --;
```

```
        m ++;
      }
      else {
        i = (((i + m) + e) + s) - 1;
        s = 0;
        e = 1;
        m = 0;
      }
  }
  __retres = 0;
  return __retres;
}
```

22 test/moesi.c

Degree : 1

```
/* Generated by Frama-C */
int moesi(void)
{
  /*@ ghost float k2; */
  /*@ ghost float k1; */
  int __retres;
  int modified;
  int shared;
  int exclusive;
  int invalid;
  int owned;
  exclusive = 0;
  modified = 0;
  shared = 0;
  /*@ ghost k2 = (((1 * modified + 1 * shared) + 1 * exclusive
    ) + 1 * invalid) +
                 1 * owned;

    */
  /*@ ghost k1 = 1 * 1; */
  /*@ loop invariant Pilat_emitter: 1*1 == k1;
      loop invariant
        Pilat_emitter:
          (((1*modified+1*shared)+1*exclusive)+1*invalid)+1*
    owned == k2;
  */
  while (1) {
    modified = modified;
    shared = shared;
    exclusive = exclusive;
```

```
29      invalid = invalid;
        owned = owned;
31      if (invalid) {
          shared = (shared + exclusive) + 1;
33        owned += modified;
          invalid --;
35        exclusive = 0;
          modified = 0;
37      }
        else
39        if (exclusive) {
            exclusive --;
41          modified ++;
          }
43        else
            if (shared) {
45            invalid = ((((invalid + modified) + exclusive) +
      shared) + owned) - 1;
              shared = 0;
47            exclusive = 1;
              modified = 0;
49            owned = 0;
            }
51          else
              if (owned) {
53              invalid = ((((invalid + modified) + exclusive) +
      shared) + owned) - 1;
                shared = 0;
55              exclusive = 1;
                modified = 0;
57              owned = 0;
              }
59            else {
                invalid = ((((invalid + modified) + shared) +
      exclusive) + owned) - 1;
61              shared = 0;
                exclusive = 1;
63              modified = 0;
                owned = 0;
65            }
    }
67  return __retres;
}
```

23 test/prod.c

Degree : 2

```
/* Generated by Frama-C */
int prod(void)
{
  /*@ ghost float k4; */
  /*@ ghost float k3; */
  /*@ ghost float k2; */
  /*@ ghost float k1; */
  int __retres;
  int x;
  int y;
  int z;
  /*@ ghost k4 = 1 * (x * x); */
  /*@ ghost k3 = 1 * x; */
  /*@ ghost k2 = 1 * 1; */
  /*@ ghost k1 = 1 * (y * x) + 1 * z; */
  /*@ loop invariant Pilat_emitter: 1*(y*x)+1*z == k1;
      loop invariant Pilat_emitter: 1*1 == k2;
      loop invariant Pilat_emitter: 1*x >= k3;
      loop invariant Pilat_emitter: 1*(x*x) >= k4;
  */
  while (1) {
    if (x) {
      z += x;
      x *= 2;
      y = (y - 1) / 2;
    }
    else {
      x *= 2;
      y /= 2;
      z = z;
    }
  }
  __retres = 0;
  return __retres;
}
```

24 test/prod4br.c

Degree : 2

```
/* Generated by Frama-C */
int prod4br(void)
{
```

```
/*@ ghost float k3; */
/*@ ghost float k2; */
/*@ ghost float k1; */
int a;
int b;
int p;
int q;
/*@ ghost k3 = -1 * (p * p); */
/*@ ghost k2 = -1 * p; */
/*@ ghost k1 = -1 * 1; */
/*@ loop invariant Pilat_emitter: -1*1 == k1;
    loop invariant Pilat_emitter: -1*p >= k2;
    loop invariant Pilat_emitter: -1*(p*p) >= k3;
*/
while (1) {
  if (a != 0) {
    if (! (b != 0)) break;
  }
  else break;
  if (a % 2 == 0) {
    if (b % 2 == 0) {
      a /= 2;
      b /= 2;
      p = 4 * p;
    }
    else goto _LAND_1;
  }
  else {
    _LAND_1: ;
    if (a % 2 == 1) {
      if (b % 2 == 0) {
        a --;
        q += b * p;
      }
      else goto _LAND_0;
    }
    else {
      _LAND_0: ;
      if (a % 2 == 0) {
        if (b % 2 == 1) {
          b --;
          q += a * p;
        }
        else goto _LAND;
      }
      else {
        _LAND: a --;
               b --;
               q += ((a + b) - 1) * p;
      }
```

```
      }
    }
  }
  return p;
}
```

25 test/prodbin.c

Degree : 2

```
/* Generated by Frama-C */
int prodbin(int a, int b)
{
  /*@ ghost float k4; */
  /*@ ghost float k3; */
  /*@ ghost float k2; */
  /*@ ghost float k1; */
  int x;
  int y;
  int z;
  x = a;
  y = b;
  z = 0;
  /*@ ghost k4 = 1 * (x * x); */
  /*@ ghost k3 = 1 * x; */
  /*@ ghost k2 = 1 * 1; */
  /*@ ghost k1 = 1 * (y * x) + 1 * z; */
  /*@ loop invariant Pilat_emitter: 1*(y*x)+1*z == k1;
      loop invariant Pilat_emitter: 1*1 == k2;
      loop invariant Pilat_emitter: 1*x >= k3;
      loop invariant Pilat_emitter: 1*(x*x) >= k4;
  */
  while (y != 0) {
    if (y % 2 == 1) {
      z += x;
      y --;
    }
    x = 2 * x;
    y /= 2;
  }
  ;
  return z;
}
```

26 test/read_writ.c

Degree : 1

```
/* Generated by Frama-C */
int read_writ(void)
{
  /*@ ghost float k4; */
  /*@ ghost float k3; */
  /*@ ghost float k2; */
  /*@ ghost float k1; */
  int __retres;
  int c1;
  int c2;
  int k0;
  int r;
  int w;
  int k;
  r = 0;
  w = 0;
  k = k0;
  /*@ ghost k4 = 1 * 1; */
  /*@ ghost k3 = 1 * c1; */
  /*@ ghost k2 = 1 * c2; */
  /*@ ghost k1 = 1 * k0; */
  /*@ loop invariant Pilat_emitter: 1*k0 == k1;
      loop invariant Pilat_emitter: 1*c2 == k2;
      loop invariant Pilat_emitter: 1*c1 == k3;
      loop invariant Pilat_emitter: 1*1 == k4;
  */
  while (1) {
    r = r;
    w = w;
    k = k;
    k0 = k0;
    c1 = c1;
    c2 = c2;
    if (w == 0) {
      r ++;
      k -= c1;
    }
    else
      if (r == 0) {
        w ++;
        k -= c2;
      }
      else
        if (w == 0) {
          r --;
          k += c1;
```

```
47          }
            else {
49            w --;
              k += c2;
51          }
    }
53  __retres = 0;
    return __retres;
55  }
```

27 test/sqrt.c

Degree : 2

```
1   /* Generated by Frama-C */
    int sqrt(int n)
3   {
      /*@ ghost float k4; */
5     /*@ ghost float k3; */
      /*@ ghost float k2; */
7     /*@ ghost float k1; */
      int a;
9     int su;
      int t;
11    a = 0;
      su = 1;
13    t = 1;
      /*@ ghost k4 = (-4 * (t * a) + 4 * (a * a)) + 1 * (t * t);
        */
15    /*@ ghost k3 = -2 * a + 1 * t; */
      /*@ ghost k2 = ((-1 * a + -1 * (t * a)) + 1 * (a * a)) + 1 *
         su; */
17    /*@ ghost k1 = 1 * 1; */
      /*@ loop invariant Pilat_emitter: 1*1 == k1;
19        loop invariant Pilat_emitter: ((-1*a+-1*(t*a))+1*(a*a))
        +1*su == k2;
          loop invariant Pilat_emitter: -2*a+1*t == k3;
21        loop invariant Pilat_emitter: (-4*(t*a)+4*(a*a))+1*(t*t)
         == k4;
      */
23    while (su <= n) {
        a ++;
25      t += 2;
        su += t;
27    }
      return a;
29  }
```

```
31 int main(void)
{
33   int tmp;
   tmp = sqrt(16);
35   return tmp;
}
```

28 test/petter_1.c

Degree : 2

```
/* Generated by Frama-C */
2 int petter1(int N)
{
4   /*@ ghost float k2; */
   /*@ ghost float k1; */
6   int x;
   int y;
8   x = 0;
   y = 0;
10  /*@ ghost k2 = (-2 * x + -1 * y) + 1 * (y * y); */
   /*@ ghost k1 = 1 * 1; */
12  /*@ loop invariant Pilat_emitter: 1*1 == k1;
       loop invariant Pilat_emitter: (-2*x+-1*y)+1*(y*y) == k2;
14  */
   while (y < N) {
16    x += y;
     y ++;
18  }
   y --;
20  return x;
}
```

29 test/petter_2.c

Degree : 3

```
1 /* Generated by Frama-C */
int petter2(int N)
3 {
   /*@ ghost float k2; */
5  /*@ ghost float k1; */
   int x;
```

```
  int y;
  x = 0;
  y = 0;
  /*@ ghost k2 = ((-6 * x + 1 * y) + -3 * (y * y)) + 2 * (y *
    (y * y)); */
  /*@ ghost k1 = 1 * 1; */
  /*@ loop invariant Pilat_emitter: 1*1 == k1;
      loop invariant Pilat_emitter: ((-6*x+1*y)+-3*(y*y))+2*(y
    *(y*y)) == k2;
  */
  while (y < N) {
    x += y * y;
    y ++;
  }
  y --;
  return x;
}
```

30 test/petter_3.c

Degree : 4

```
/* Generated by Frama-C */
int petter3(int N)
{
  /*@ ghost float k2; */
  /*@ ghost float k1; */
  int x;
  int y;
  x = 0;
  y = 0;
  /*@ ghost k2 = ((-4 * x + 1 * (y * y)) + -2 * (y * (y * y)))
    + 1 * (

      y * (

      y * (

      y * y)));

    */
  /*@ ghost k1 = 1 * 1; */
  /*@ loop invariant Pilat_emitter: 1*1 == k1;
      loop invariant
        Pilat_emitter: ((-4*x+1*(y*y))+-2*(y*(y*y)))+1*(y*(y*(
    y*y))) == k2;
  */
```

```
21    while (y < N) {
        x += (y * y) * y;
23      y ++;
      }
25    y --;
      return x;
27  }
```

31 test/petter_20.c

Degree : 21

```
1   /* Generated by Frama-C */
    int petter20(int N)
3   {
      /*@ ghost float k2; */
5     /*@ ghost float k1; */
      int x;
7     int y;
      x = 0;
9     y = 0;
      /*@ ghost k2 = ((((((((((-6930 * x + -3666831 * y) +
        24126850 * (y * (
11
              y * y))) +
                                -47625039 * (y * (y * (y * (y * y)))
        )) + 44767800 * (
13
             y * (
             y * (
15           y * (
             y * (
17           y * (
             y * y)))))))) +
19                           -24551230 * (y * (y * (y * (y * (y * (
        y * (y * (
             y * y)))))))))) +
21                          8817900 * (y * (y * (y * (y * (y * (y * (y *
         (y * (
          y * (
```

```
      y * (
      y * y))))))))))) +
                    -2238390 * (y * (y * (y * (y * (y * (y *
    (y * (
      y * (
      y * (
      y * (
      y * (
      y * y))))))))))))) +
                    426360 * (y * (y * (y * (y * (y * (y * (y
    * (y * (
        y * (
        y * (
        y * (
        y * (
        y * (
        y * y))))))))))))))) +
                    -65835 * (y * (y * (y * (y * (y * (y * (y
    * (y * (
      y * (
      y * (
      y * (
      y * (
      y * (
      y * (
      y * (
      y * y))))))))))))))))) +
                    11550 * (y * (y * (y * (y * (y * (y * (y *
    (y * (y * (
```

```
            y * (
            y * (
            y * (
            y * (
            y * (
            y * (
            y * (
            y * (
            y * y))))))))))))))))))) +
                    -3465 * (y * (y * (y * (y * (y * (y * (y * (
    y * (y * (
            y * (
            y * (
            y * (
            y * (
            y * (
            y * (
            y * (
            y * (
            y * y))))))))))))))))))) +
                    330 * (y * (y * (y * (y * (y * (y * (y * (y *
      (y * (
            y * (
            y * (
            y * (
            y * (
```

```
73
      y * (
      y * (
75
      y * (
      y * (
77
      y * (
      y * (
79
      y * y)))))))))))))))))))));
81
    */
  /*@ ghost k1 = 1 * 1; */
83
  /*@ loop invariant Pilat_emitter: 1*1 == k1;
      loop invariant
85
        Pilat_emitter:
          ((((((((((((-6930*x+-3666831*y)+24126850*(y*(y*y)))
  +-47625039*
87                    (y*(y*(y*(y*y))))+44767800*(y*(y*(y*(y*(y
  *(y*y)))))))+-24551230*
                  (y*(y*(y*(y*(y*(y*(y*(y*y)))))))+8817900*(
  y*(y*(y*
89
          (y*
          (y*
91
          (y*
          (y*
93
          (y*
          (y*
95
          (y*y)))))))))))+-2238390*
                (y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*y)))))))))
  ))))+426360*
97              (y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*y))))))
  ))))))))))+-65835*
              (y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*
  y))))))))))))))))+11550*
99            (y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(
  y*(y*y)))))))))))))))))+-3465*
            (y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y
  *(y*(y*y)))))))))))))))))+330*
```

```
101             (y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(y*(
      y*(y*(y*y))))))))))))))))))
                == k2;
103   */
    while (y < N) {
105     x += ((((((((((((((((((((y * y) * y) * y) * y) * y) * y) * y
      ) * y) * y) * y) * y) * y) * y) * y) * y) * y) * y) * y) *
       y;
      y ++;
107   }
    y --;
109   return x;
}
```