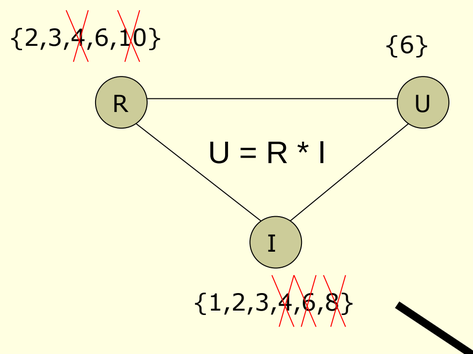
CAVERN:
Constraints and Abstractions for program VERificatioN

ANR-07-SESUR-003
http://www-anr-ci.cea.fr

Sécurité et Sûreté Informatique

Constraint Programming

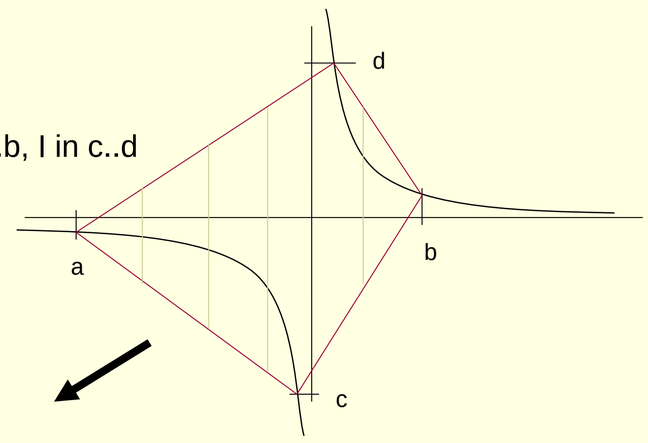
Exploit relations (constraints) to infer new informations on objects that represent unknowns (variables)



Abstractions

Over-approximate the computation of relations to benefit from powerful solving techniques (e.g. Linear Programming)

1 = R * I, R in a..b, I in c..d



Program Verification

P1a	/*@ behavior P1a : assumes
	flag = 0 && n = 8 && forall i in 1..8, ar[i] = cos(2*M_PI*i/n);
	ensures \result == 0;

P1b	/*@ behavior P1b : assumes
	flag = 1 && n = 8 && forall i in 1..8, ar[i] = cos(2*M_PI*i/n);
	ensures \result == 0;

…	

Are properties P1a,P1b, … verified by this implementation ?

```
int fft1(int n, int flag)  {
   int i, j, k, it, xp, xp2, j1, j2, iter;
   double sign, w, wr, wi, dr1, dr2, di1, di2, tr, ti, arg;

   if(n < 2) return(999);    iter = log((double)n)/log(2.0);   j = 1;
   for(i = 0; i < iter; i++)    j *= 2;    if(fabs(n-j) > 1.0e-6)   return(1);
   sign = ((flag == 1) ? 1.0 : -1.0);  xp2 = n;
   for(it = 0; it < iter; it++) {    xp = xp2;  xp2 /= 2;  w = PI / xp2;
      for(k = 0; k < xp2; k++) {   arg = k * w;   wr = cos(arg);  wi = sign * sin(arg); i = k - xp;
         for(j = xp; j <= n; j += xp)  {
                  j1 = j + i;   j2 = j1 + xp2;  dr1 = ar[j1];  dr2 = ar[j2];  di1 = ai[j1]; di2 = ai[j2];
                  tr = dr1 - dr2; ti = di1 - di2;
                  ar[j1] = dr1 + dr2; ai[j1] = di1 + di2; ar[j2] = tr * wr - ti * wi; ai[j2] = ti * wr + tr * wi;
                  } } }
          j1 = n / 2; j2 = n - 1;  j = 1;
          for(i = 1; i <= j2; i++)  {
             if(i < j)  {  tr = ar[j-1];  ti = ai[j-1]; ar[j-1] = ar[i-1]; ai[j-1] = ai[i-1]; ar[i-1] = tr;  ai[i-1] = ti; }
                   k = j1;  while(k < j)  {  j -= k;  k /= 2; }   j += k; }
          if(flag == 0) return(0);  w = n;
          for(i = 0; i < n; i++) {  ar[i] /= w;  ai[i] /= w;}
          return(0);   }
```

WP2: Memory models

Fine-grained memory model of C and Bytecode Java

PhD of Florence Charreteur (Mar. 2010)
PhD of Mickael Delahaye (Oct. 2011)

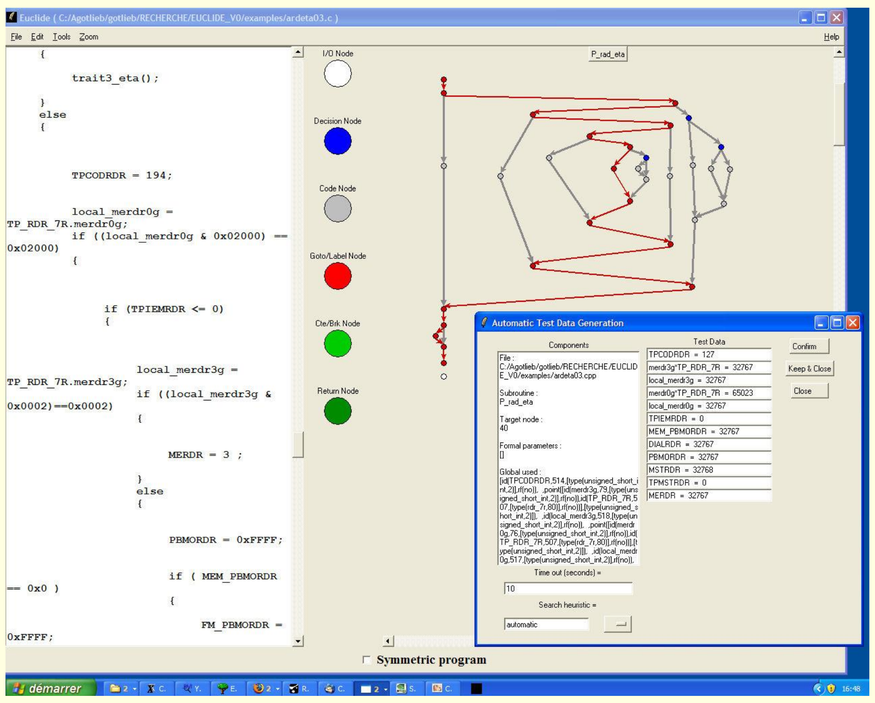
F. Charreteur, B. Botella, and A. Gotlieb.
Modelling dynamic memory management in constraint-based testing.
The Journal of Systems and Software, 82(11):1755–1766, 2009

F. Charreteur, A. Gotlieb
Constraint-based test inputs generation for J ava Bytecode.
ISSRE 2010

M. Delahaye, B. Botella, A.Gotlieb
Explanation-based generalization of infeasible path.
ICST'10, Paris.

…

WP3: Integers and iterative computations



PhD of Bruno Berstel

Post-doc Olivier Ponsini

HDR Arnaud Gotlieb (Dec. 2011)

A. Gotlieb, B. Marre, and M. Leconte.
Constraint solving on modular integers ModRef'10

Olivier Ponsini, Hélène Collavizza, Carine Fédèle, Claude Michel, Michel Rueher,
Automatic Verification of Loop Invariants ICSM 2010

…

WP4: Floating-point computations

Properties over floating-point data can be exploited to prune the search Space, e.g.

Soit $z \in \boldsymbol{F}_{p,+\infty}$, *avec* $+\infty > z > 0$ *et*

$$z = 1.b_2 \ldots b_i 0 \ldots 0 * 2^{e_z} \text{ avec } b_i = 1$$

Soient

$$y = 1.1 \ldots 1 * 2^{e_z + nb_z} \text{ avec } nb_z = p - i$$
$$x = y \oplus z$$

Alors

- $x \ominus y = x - y = z$,
- *et il n'y a pas d'*$x' \in \boldsymbol{F}_{p,+\infty}$, $x' > x$ *ou d'*$y' \in \boldsymbol{F}_{p,+\infty}$, $y' > y$ *tel que* $x' \ominus y' = z$

PhD of M. Said Belaid
Post-doc Matthieu Carlier

Bruno Marre, Claude Michel,
Improving the floating point addition and subtraction constraints CP'10

Mohammed Said Belaid, Claude Michel, Michel Rueher,
Approximating floating-point operations to verify numerical programs SCAN 2010,

Matthieu Carlier, A Gotlieb
Filtering by Maximum ULP ICTAI'2011

Partners **INRIA Rennes (Celtique)**
CEA List, Saclay
Université de Nice-Sophia Antipolis (CeP)
IBM ILOG Gentilly
Andy King -- University of York

Dates: **Février 2008 - 2011**

Contact: **Arnaud.Gotlieb@inria.fr** **(INRIA Rennes, coordinator)**

http://cavern.inria.fr