http://pan.cin.ufpe.br

Randomized Solving of Floating-Point Constraints

Marcelo d'Amorim

Informatics Center, Federal University of Pernambuco (UFPE)

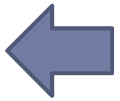
NASA Ames, Mountain View (CA), August 2010


• ~1.5 million people
• ~77F (25C) avg. temp.
• Intl. airport with flights to major cities
New Recife
Old Recife


7h AA flight to Miami
Map
MT
MN
SD
WI
MI
WY
NE
IA
NV
UT
CO
KS
MO
IL
IN
OH
PA
NY
VT
ME
NB
PE
NS
NH
MA
CT
RI
KY
WV
VA
DE
NJ
MD
NC
SC
GA
TN
AR
OK
AZ
NM
TX
MS
AL
LA
FL
North Atlantic Ocean
Gulf of Mexico
México
Cuba
PR
Guatemala
Nicaragua
Venezuela
Guyana
Suriname
Colombia
RR
AP
Ecuador
AM
PA
MA
PB
PE
AL
SE
PI
AC
Brasil
Brazil
RO
TO
Perú
MT
BA

PAN active projects

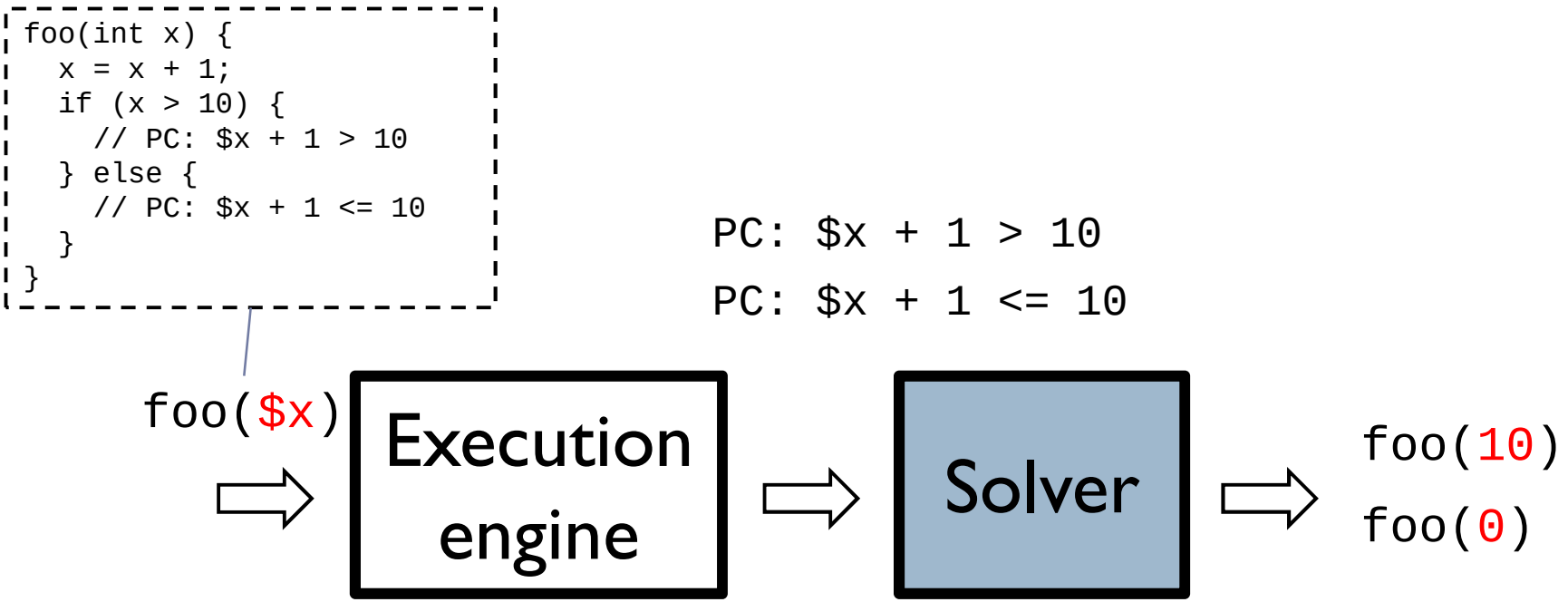
- Evolution aware lightweight debugging
  - Elton Alves
- Randomized Constraint Solving
  - Mateus Borges and Matheus Arrais
- A few others on the stack…



Focus of this talk

Context and Problem

- Context
  - Symbolic execution (SE)
- Problem
  - Inability of constraint solvers to deal with undecidable constraints



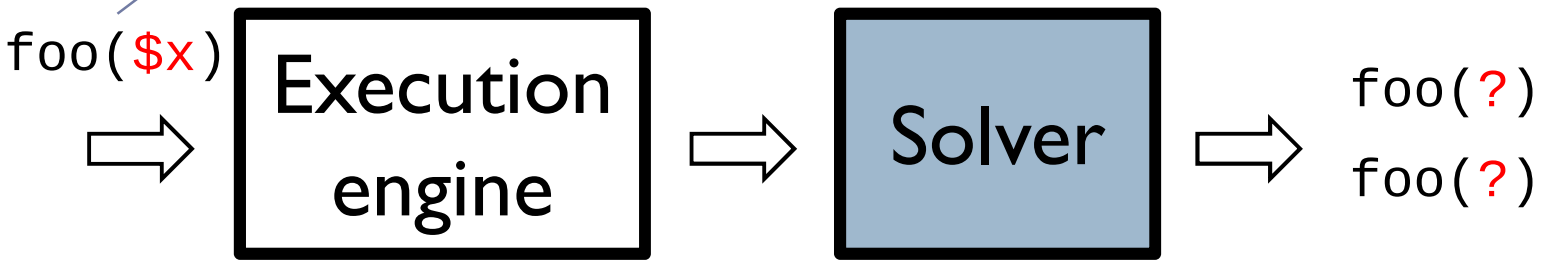
Context and Problem

- Context
  - Symbolic execution (SE)
- Problem
  - Inability of constraint solvers to deal with undecidable constraints

```
foo(double x) {
  x = x + 1;
  if (x > Math.pow(Math.sin(x),2)){
    // PC: $x + 1>($x + 1)^2
  } else {
    // PC: $x + 1<=($x + 1)^2
  }
}
```

PC: $x + 1 > sin($x + 1)^2

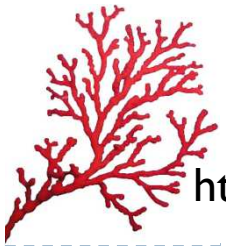
PC: $x + 1 <= sin($x + 1)^2



Target application

- Programs that manipulate floating-point variables
  - Floating-point arithmetic
  - Specific math functions (say, from java.lang.Math)
- Example
  - sqrt(exp(x+z))) < pow(z, x) , x>0, y>1, z>1, y<x+2, w=x+2
  - Solution {x=4.31, y=6.08, z=9.51, w=6.31}

CORAL solvers



CORAL
http://pan.cin.ufpe.br/coral/

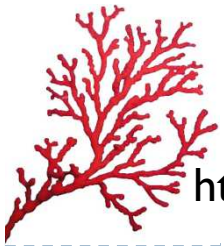
- Common to all solvers:
  - Rew. Rules + Equality decision procedure
  - Inference of value range
  - Evaluation with reverse polish calculator

pre-processing & evaluation

- Specific
  - Brute force: RANDOM
  - Heuristic search
    - Genetic Algorithms: GA
    - Particle Swarm Optimization: PSO

search strategy

CORAL solvers



CORAL
http://pan.cin.ufpe.br/coral/

- Common to all solvers:
  - Rew. Rules + Equality decision procedure
  - Inference of value range
  - Evaluation with reverse polish calculator

pre-processing & evaluation

- Specific
  - Brute force: RANDOM
  - Heuristic search
    - Genetic Algorithms: GA
    - Particle Swarm Optimization: PSO

Using opt4j library and SAW fitness function.

Illustration

sqrt(x+y)=sqrt(c), sin(z) > cos(y)

Illustration

sqrt(x+y)=sqrt(c), sin(z) > cos(y)

x+y=c, sin(z) > cos(y)

Rewrite

Illustration

sqrt(x+y)=sqrt(c), sin(z) > cos(y)

Rewrite

x+y=c, sin(z) > cos(y)

Rewrite

x=c-y, sin(z) > cos(y)

Illustration

sqrt(x+y)=sqrt(c), sin(z) > cos(y)

x+y=c, sin(z) > cos(y)

x=c-y, sin(z) > cos(y)

sin(z) > cos(y)

Rewrite

Rewrite

EQ_DP

x=c-y

Value of x depends on y.

Illustration

sqrt(x+y)=sqrt(c), sin(z) > cos(y)

Rewrite

x+y=c, sin(z) > cos(y)

Rewrite

x=c-y, sin(z) > cos(y)

EQ_DP

sin(z) > cos(y)

Search z, y

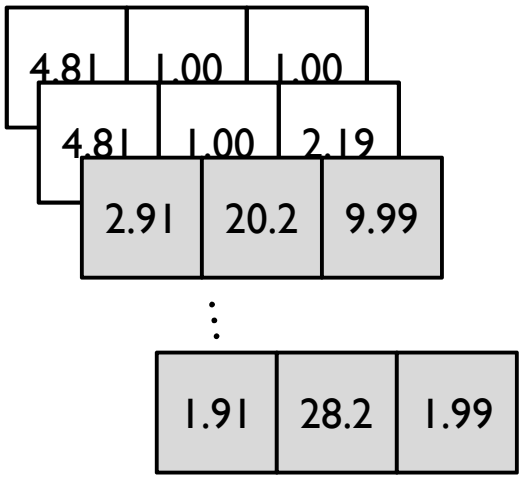
Infer value ranges of z and y from context before starting the search.

Quick outlook of heuristic search

Individuals

Fitness scores

sqrt(exp(x+z))) < pow(z, x)
, x>0, y>1, z>1, y<x+2,



2

3

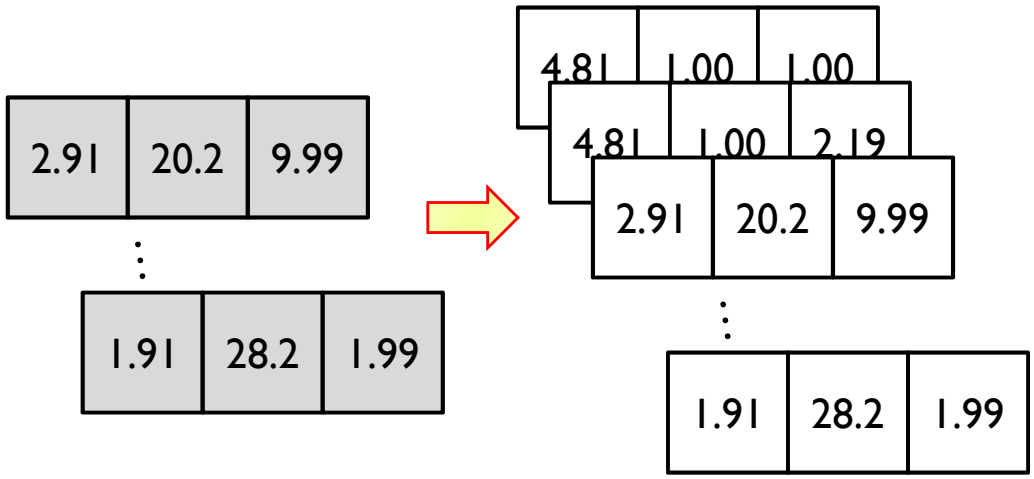
4

4

Identify best fit individuals with a fitness functions

Quick outlook of heuristic search

sqrt(exp(x+z))) < pow(z, x)
, x>0, y>1, z>1, y<x+2,



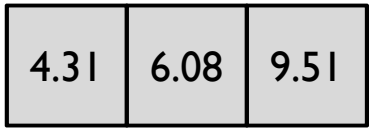
Best individuals guide next generation!

Quick outlook of heuristic search

Individuals

Fitness scores

sqrt(exp(x+z))) < pow(z, x)
, x>0, y>1, z>1, y<x+2,



5

x=4.31, y=6.08, z=9.51

Stop when find solution or timeouts

Related work

- DART [Godefroid *et al.*, PLDI'05]
- Our previous work [NFM'09, ISSE'10]
- FloPSy [Lakhotia *et al.*, ICTSS'10]
- From other domains:
  - Non-linear optimization
  - Root finding
  - …

Evaluation: Subjects

- 76 constraints manually written by 3 programmers
- Constraint = conjunction of boolean expressions
- On average:
  - 4 variables per constraint
  - 6 functions
  - 4 conjunctions
- Example

[x - y + tan(v)> (z+ t) / (w + t), sqrt(x-t) > z / y, log(x*y) > log(t+w+z) , z*2 + w*3 + x*7 < pow(y,t)*cos(v), z + w > x + y, w < x/y, x > (w+y-z), log10(t*x) < sqrt(w*y*z), x * cos(t + y) > log(w*z*3), cos(t) * sin(v) > cos(y), sin(x*y) + sin(z*w) + sin(t*v) < cos(x*y) + cos(z*w) + cos(t*v)]

Evaluation: Results

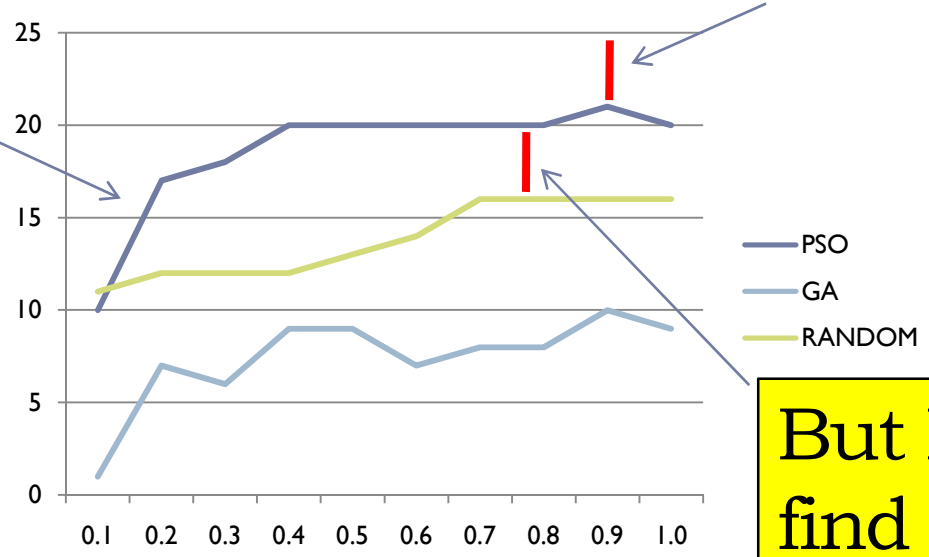
- All solvers miss 9 constraints
- All solvers find solution for 43 constraints (out of 76)
  - Value inference is critical for 3
  - Rew + EQ_DP is critical for 7
- Results vary for 24 constraints
  - GA is subsumed by RANDOM and PSO
  - RANDOM and PSO complementary. (Seed=1 ant timeout=1s):
    - In 3 cases PSO solves but RANDOM does not
    - In 2 cases RANDOM solves but PSO does not

Evaluation: Results

- Seed=1, varying timeout 0.1 to 1s, with a 0.1 step

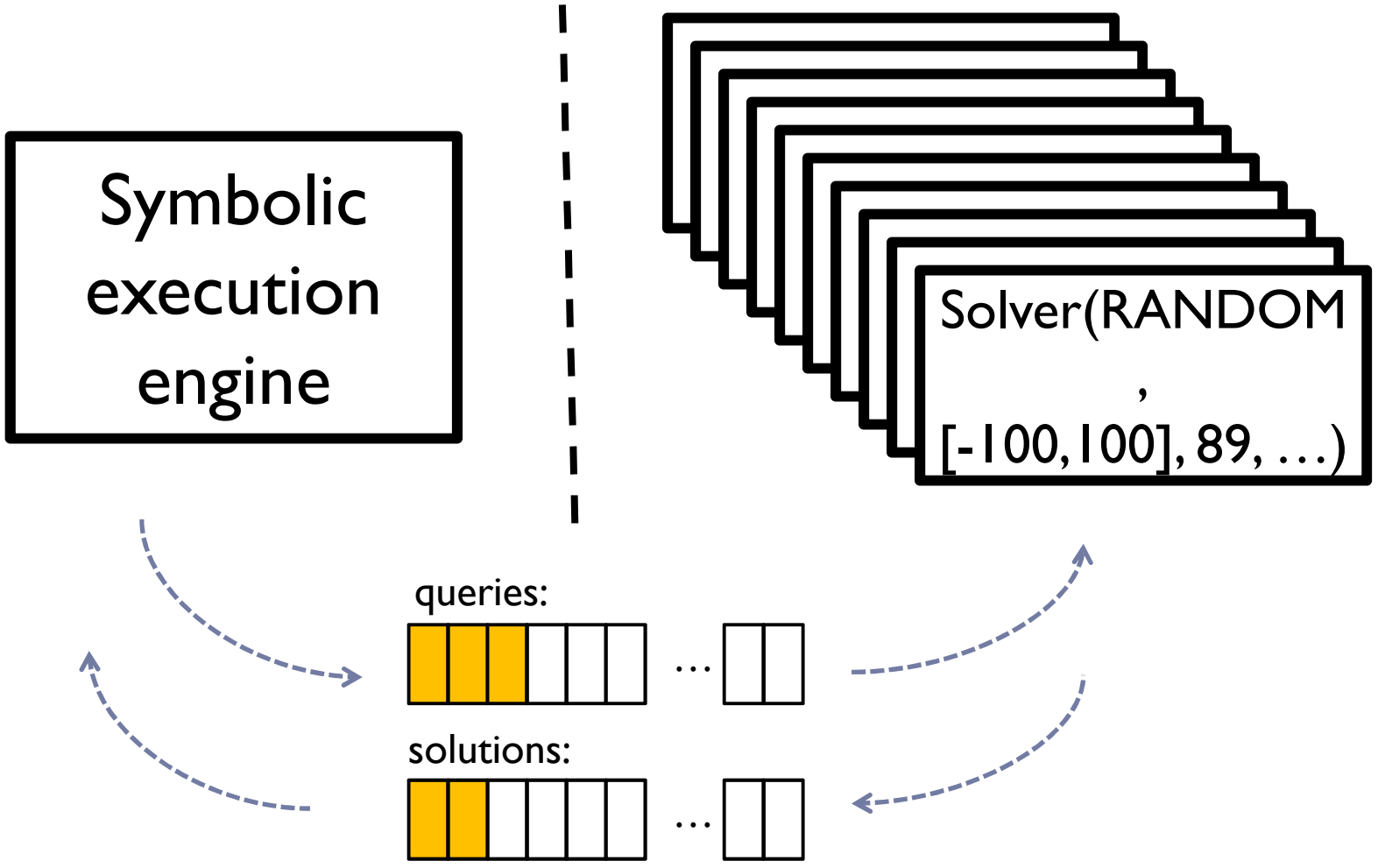
Misses 4 constraints with this seed and timeout

Sharp increase



But RANDOM can find solutions that PSO misses

Proposal: Parallel-Random Solving



Future work

- Integrate with Symbolic JPF
- Evaluate with a real case study
- Optimizations
  - Parallelize solving
  - Cache results from previous runs
  - Collaborate with decision procedures

Other projects on PAN

- Evolution aware lightweight debugging
  - Elton Alves
- Randomized Constraint Solving
  - Mateus Borges and Matheus Arrais
- A few others on the stack…
  - Interactive incremental test generation
  - …

Evolution Aware Lightweight Debugging

- Problem: Coverage-based debuggers are efficient but not very precise
- Approach: Leverage on evolution information
  - **Aging of statements**
  - **Bug incidence**
  - Refactorings
  - Augmented oracles

	test cases					suspiciousness	
stmt	5	4 3 2	6 7 8	**3 3 5**	7 **8 8**	S	δS
1	•	•	•	•	•		
2	•	•	•	•	•	0.63	0.63
3	•	•	•	•	•	0.63	0.63
4	•	•	•	•	•	0.63	
5	•	•	•	•	•	0.63	
6		•	•	•	•	0.70	
7		•	•	•	•	0.70	0.70
8		•	•	•	•	0.70	
9		•	•	•	•	0.70	
10		•				0.00	0.00
11		•				0.00	0.00
12		•				0.00	
13		•				0.00	
14		•				0.00	
15		•				0.00	
⋆16			•	•	•	0.81	0.81
17			•	•	•	0.81	0.81
18			•	•	•	0.81	0.81
19			•	•	•	0.81	
20			•	•	•	0.81	
21			•	•	•	0.81	
22			•	•	•	0.81	
23							
24							
25							
26		•	•	•	•	0.70	
27		•	•	•	•	0.70	
28		•	•	•	•	0.70	

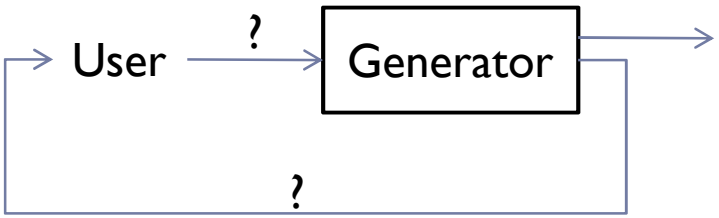
- People:
  - Elton Alves (UFPE)
  - Milos Gligoric, and Vilas Jagananth (University of Illinois)

Interactive and Incremental Test Gen.

- Problem: program test generation is challenging

Interactive and Incremental Test Gen.

- Problem: program test generation is challenging
- Approach: Exploit interaction of user and tool
- Question: How to communicate information from user to generator (and conversely)?



PAN **http://pan.cin.ufpe.br**

Informatics Center (CIn) on
Federal University of Pernambuco (UFPE)