Succinct Representations for Abstract Interpretation

Combined Analysis Algorithms and Experimental Evaluation.

Julien Henry, David Monniaux, Matthieu Moy

September 13, 2012

Sources of Imprecision in Abstract Interpretation

- Abstract domain
- Widening operator
  - ensures fast convergence
  - BUT: may induce huge imprecisions
  - Narrowing tends to recover some precision…
- Consider paths that are unfeasible in reality: least upper bound operations

Sources of Imprecision in Abstract Interpretation

- Abstract domain
- Widening operator
  - ensures fast convergence
  - BUT: may induce huge imprecisions
  - Narrowing tends to recover some precision...
- Consider paths that are unfeasible in reality: least upper bound operations

Summary

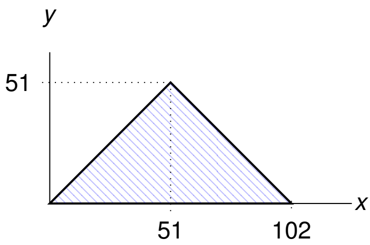
Summary

1. Introduction: Weakness of the standard approach & Guided Static Analysis
2. Using SMT-solving to focus new paths
3. Combining Both Techniques
4. Computing Disjunctive Invariants

Example of Standard Abstract Interpretation

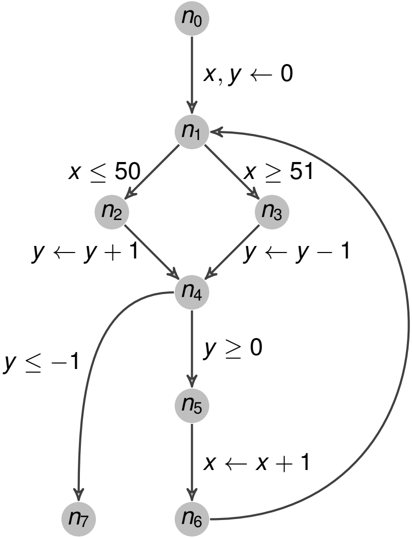
Example from Gopan & Reps, SAS'07

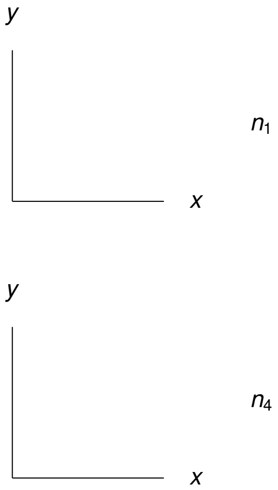
```
x = 0;
y = 0;
while (true) {
        if (x <= 50)
                y++;
        else
                y--;

        if (y < 0) break;
        x++;
}
```



- x and y incremented during 51 iterations
- x incremented and y decremented during 51 iterations

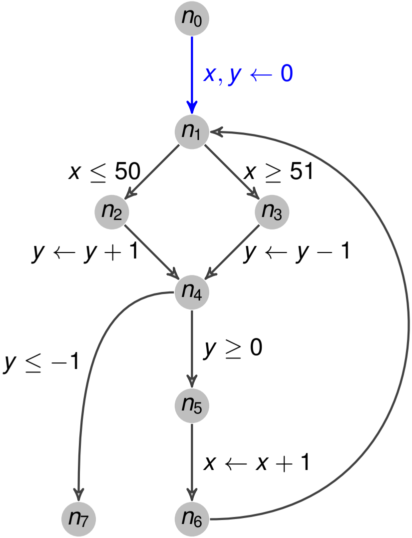
Example of Standard Abstract Interpretation

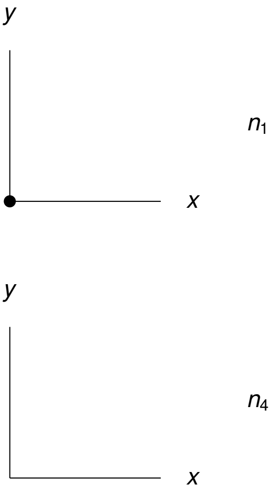




Ascending iterations

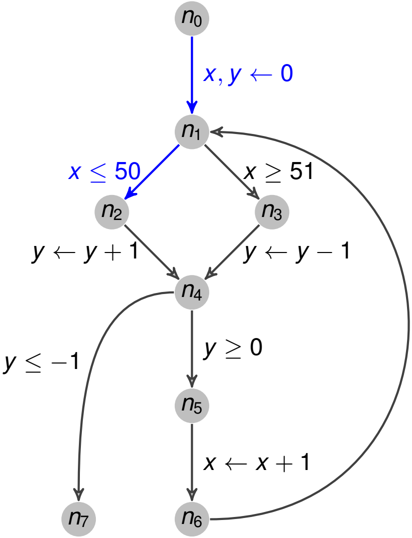
Example of Standard Abstract Interpretation

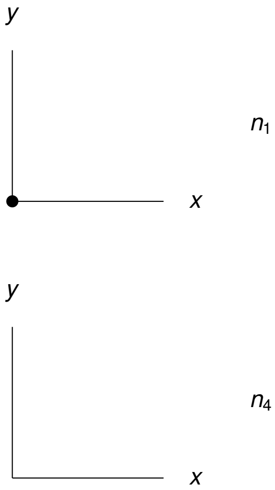




Ascending iterations

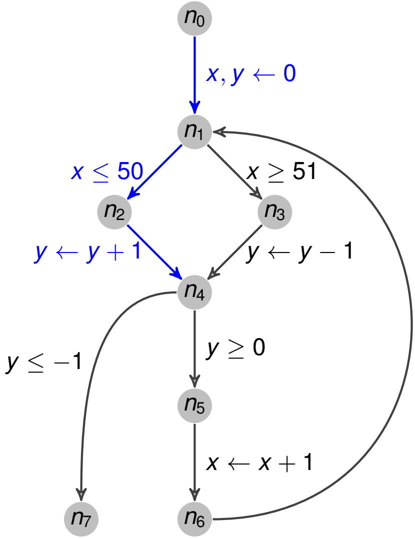
Example of Standard Abstract Interpretation

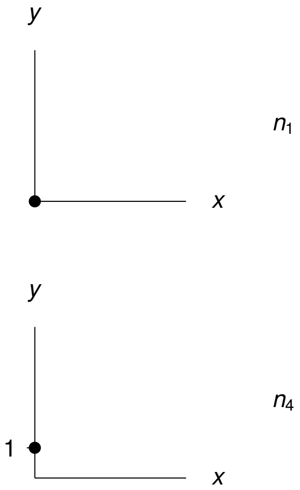




Ascending iterations

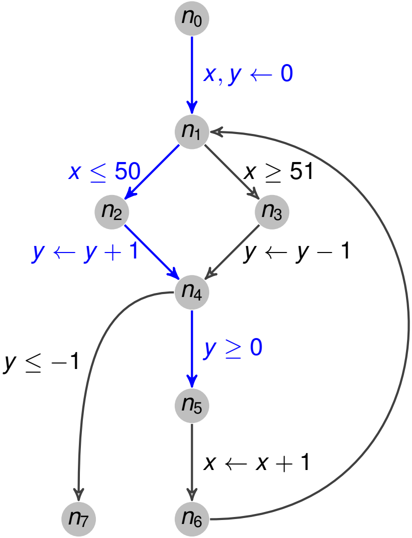
Example of Standard Abstract Interpretation

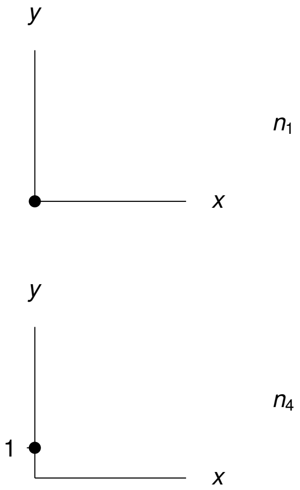




Ascending iterations

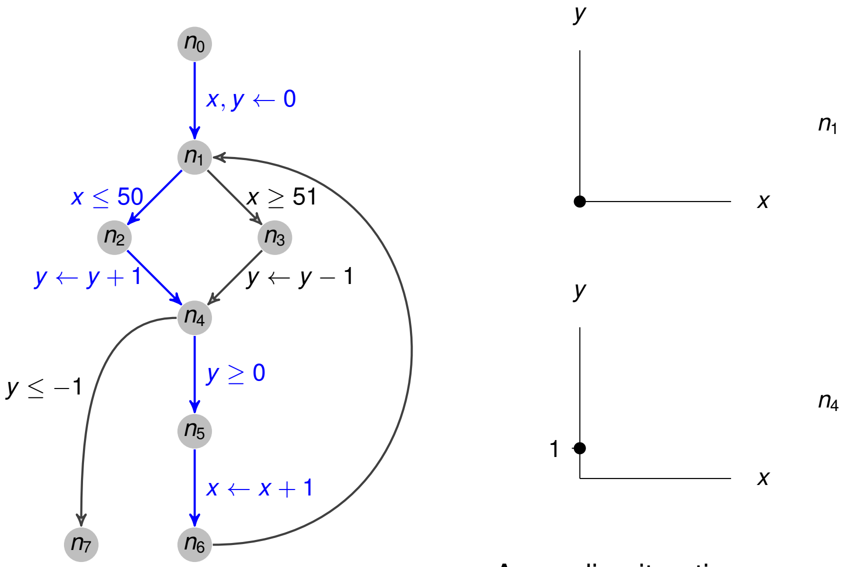
Example of Standard Abstract Interpretation





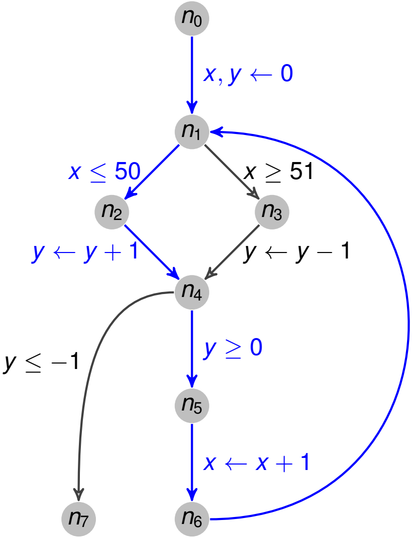
Ascending iterations

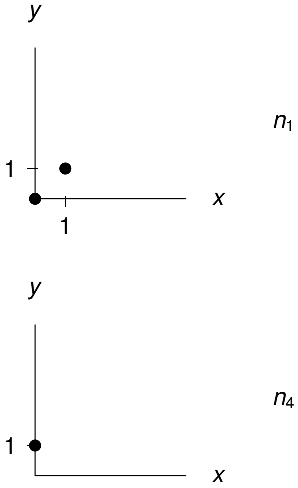
Example of Standard Abstract Interpretation



Ascending iterations

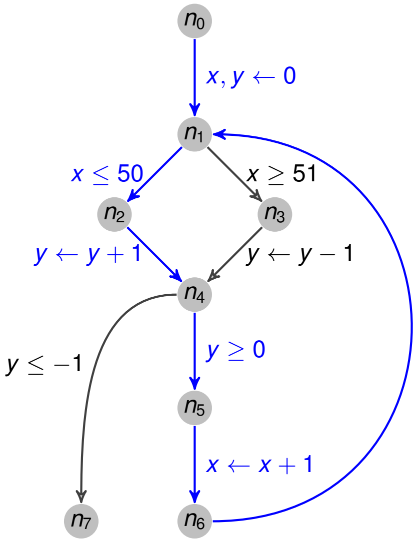
Example of Standard Abstract Interpretation

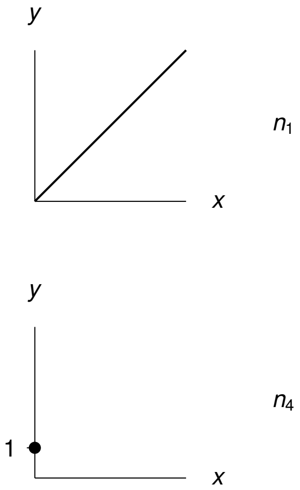




Ascending iterations

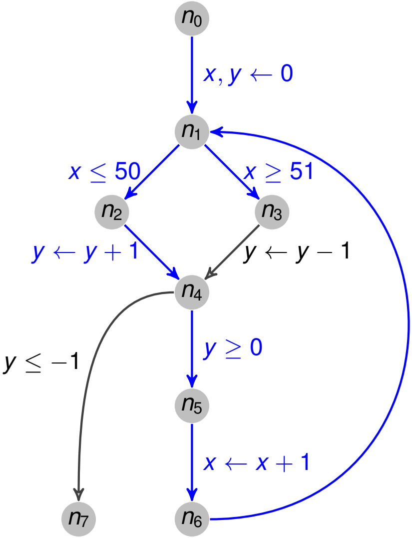
Example of Standard Abstract Interpretation

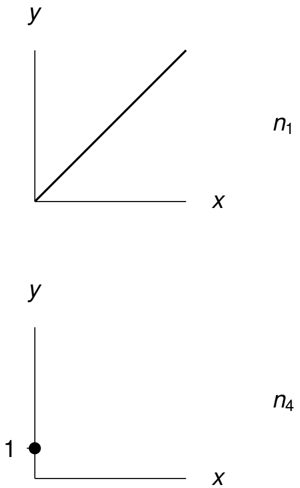




Ascending iterations

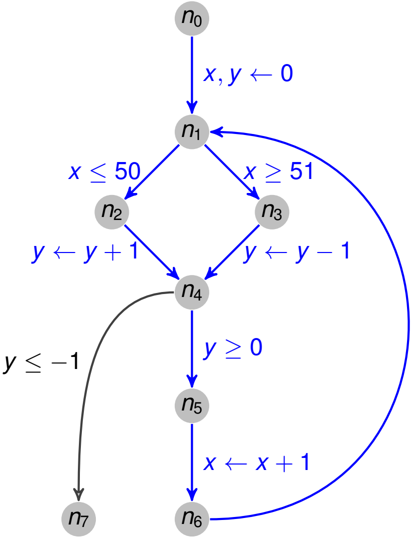
Example of Standard Abstract Interpretation

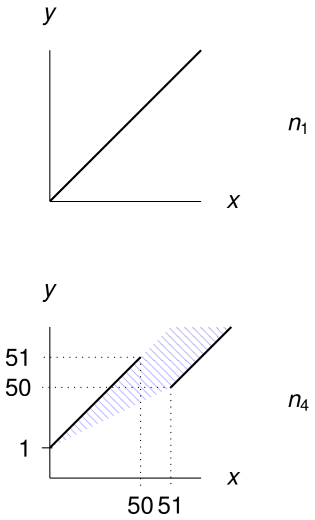




Ascending iterations

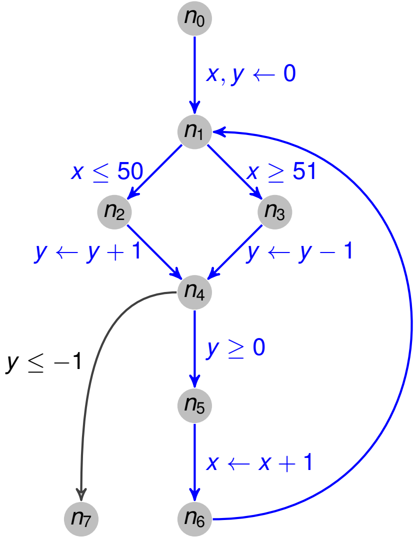
Example of Standard Abstract Interpretation

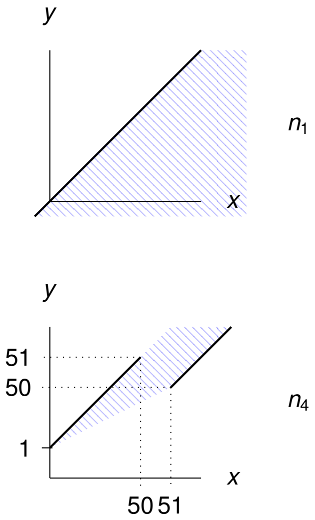




Ascending iterations

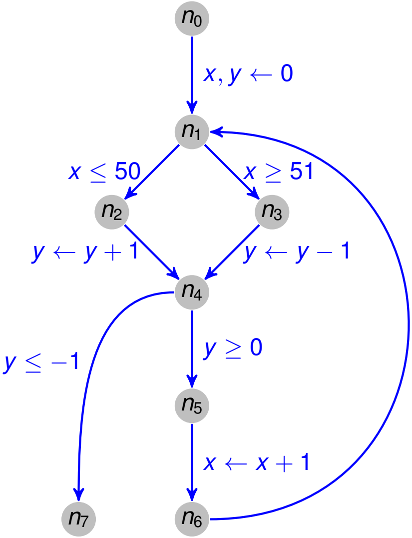
Example of Standard Abstract Interpretation

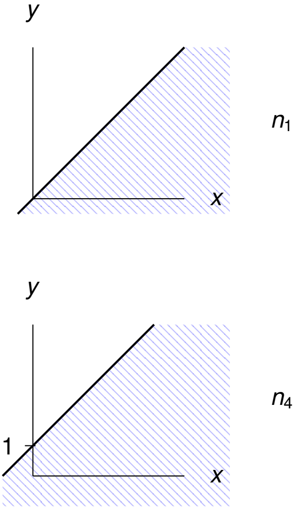




Ascending iterations

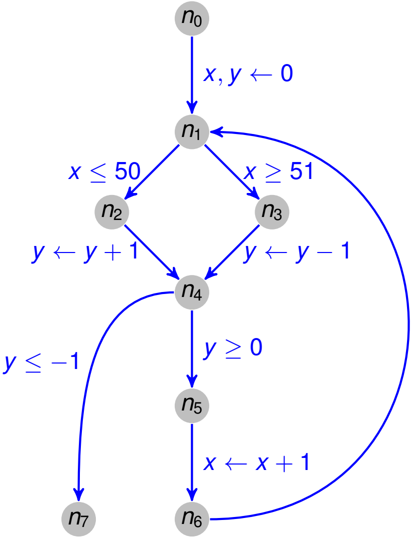
Example of Standard Abstract Interpretation

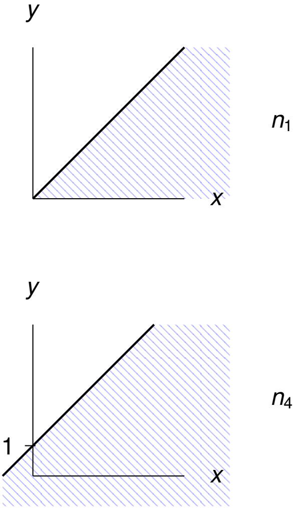




Ascending iterations

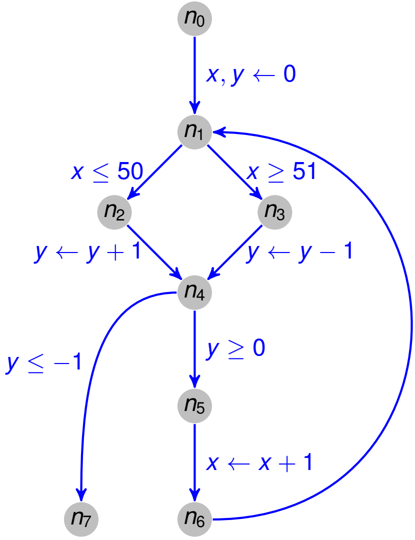
Example of Standard Abstract Interpretation

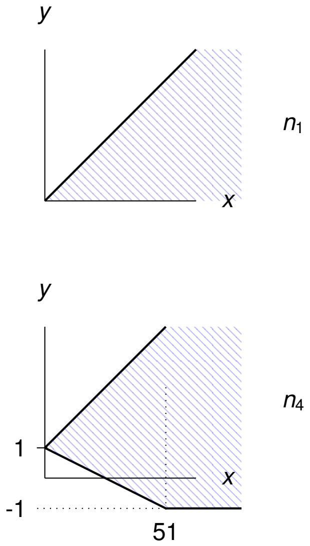




Descending iterations

Example of Standard Abstract Interpretation





Descending iterations

Guided Static Analysis

D. Gopan & T. Reps, SAS'07

- separate loops into distinct phases.
- obtaining a solution for each loop phase before proceeding to the next.
- widening & narrowing at each loop phase.
  - Better precision

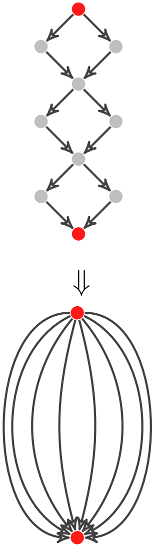
$\Rightarrow$ Ascending sequence of subsets of transitions

Summary

Principle of Path Focusing

D. Monniaux & L. Gonnord - SAS 2011

- Compute the fixpoint iterations on a multigraph
- Take a set P_R of nodes
- Distinguish all the paths between 2 nodes of P_R



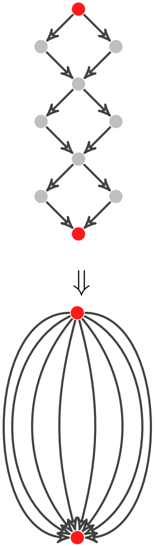
Principle of Path Focusing

D. Monniaux & L. Gonnord - SAS 2011

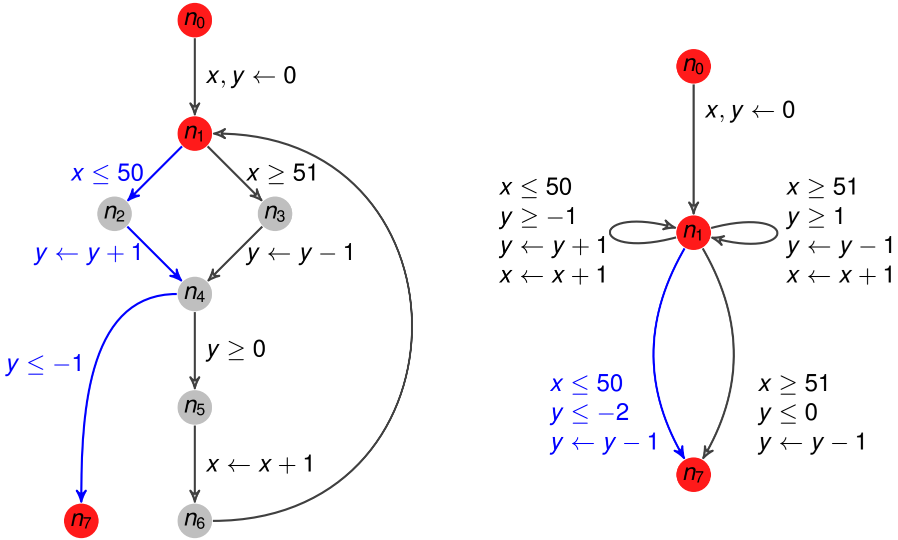
- Compute the fixpoint iterations on a multigraph
- Take a set P_R of nodes
- Distinguish all the paths between 2 nodes of P_R

Exponential number of paths $\Rightarrow$

- We don't construct this graph explicitly
- We use SMT-solving to find interesting paths



Reducing the Graph

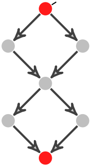


Using SMT-solving to Find New Paths

- SMT formula ρ expressing the semantics of the program
- ρ contains reachability predicates

"Does there exist a path starting in the invariant candidate, that arrives in a state outside the invariant ?"

$$f(p_i) = \rho \wedge b_i^s \wedge \bigwedge_{\substack{j \in P_R \\ j \neq i}} \neg b_j^s \wedge X_i \wedge \bigvee_{j \in Succ(i)} \left(b_j^d \wedge \neg X_j \right)$$



starting point

starting value

destination

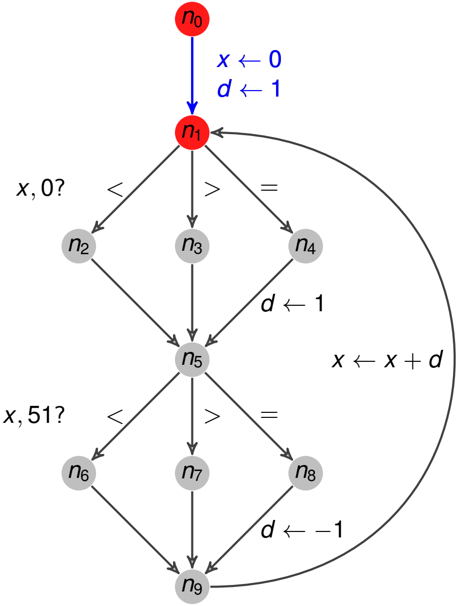
destination value

Example

```
int x = 0;
int d = 1;

while (true) {
        if (x == 0) d=1;
        if (x == 51) d=-1;
        x +=d;
}
```

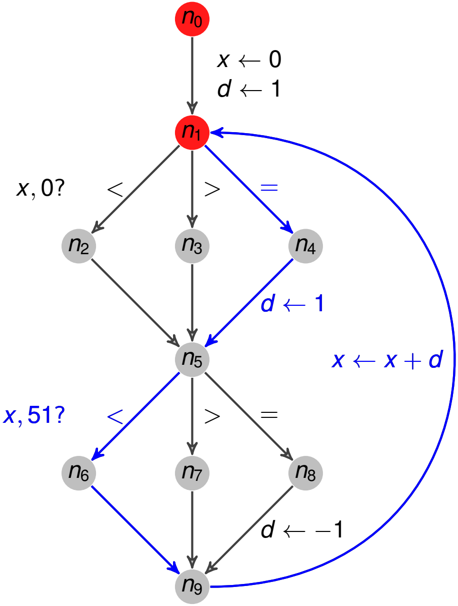
- *x* incremented until it is equal to 51,
- *x* decremented until it is equal to 0,
- restart...

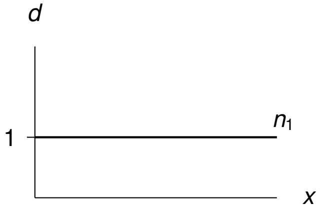
Example


n_0
$x \leftarrow 0$
$d \leftarrow 1$
n_1
$x, 0?$
$<$
$>$
$=$
n_2
n_3
n_4
$d \leftarrow 1$
n_5
$x \leftarrow x + d$
$x, 51?$
$<$
$>$
$=$
n_6
n_7
n_8
$d \leftarrow -1$
n_9

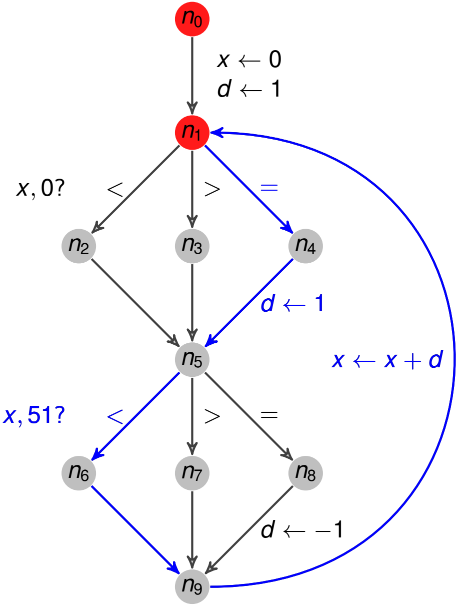

d
1
n_1
x

Example

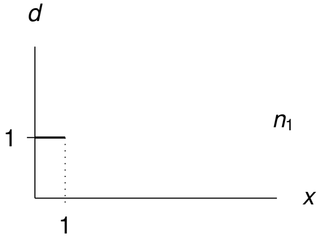

n_0
$x \leftarrow 0$
$d \leftarrow 1$
n_1
$x, 0?$
$<$
$>$
$=$
n_2
n_3
n_4
$d \leftarrow 1$
n_5
$x \leftarrow x + d$
$x, 51?$
$<$
$>$
$=$
n_6
n_7
n_8
$d \leftarrow -1$
n_9


d
1
n_1
x

Example

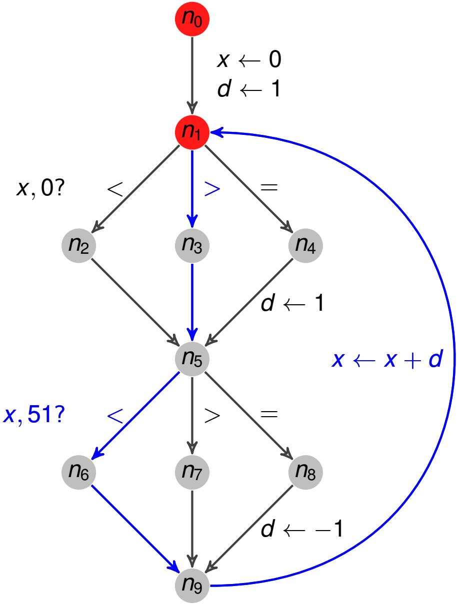


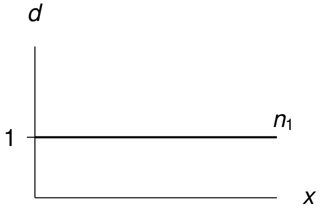
n_0
$x \leftarrow 0$
$d \leftarrow 1$
n_1
$x, 0?$
$<$
$>$
$=$
n_2
n_3
n_4
$d \leftarrow 1$
n_5
$x \leftarrow x + d$
$x, 51?$
$<$
$>$
$=$
n_6
n_7
n_8
$d \leftarrow -1$
n_9



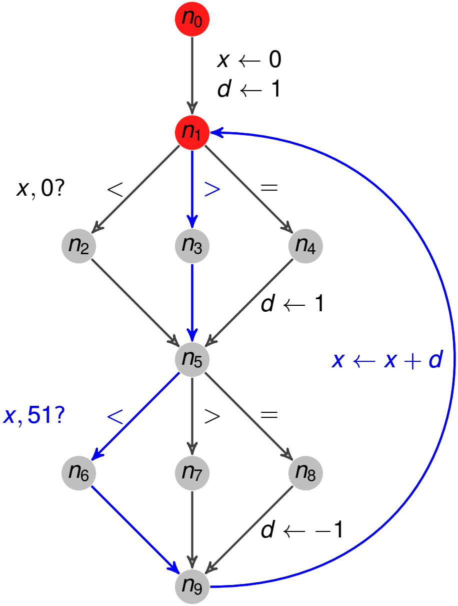
d
1
n_1
x
1

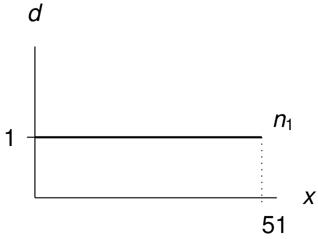
Example


n_0
$x \leftarrow 0$
$d \leftarrow 1$
n_1
$x, 0?$
$<$
$>$
$=$
n_2
n_3
n_4
$d \leftarrow 1$
n_5
$x \leftarrow x + d$
$x, 51?$
$<$
$>$
$=$
n_6
n_7
n_8
$d \leftarrow -1$
n_9

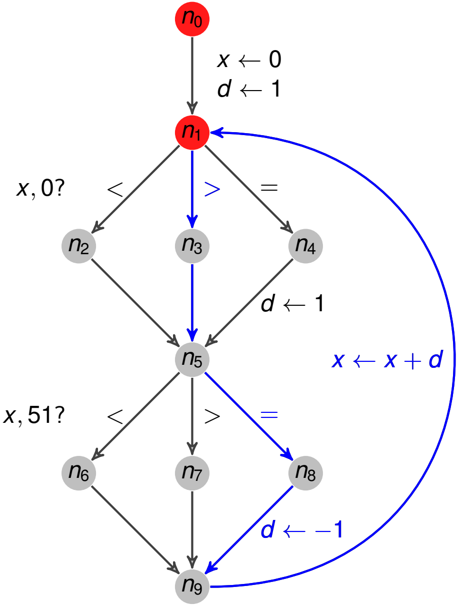

d
1
n_1
x

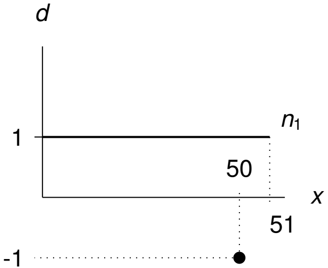
Example


n_0
$x \leftarrow 0$
$d \leftarrow 1$
n_1
$x, 0?$
$<$
$>$
$=$
n_2
n_3
n_4
$d \leftarrow 1$
n_5
$x \leftarrow x + d$
$x, 51?$
$<$
$>$
$=$
n_6
n_7
n_8
$d \leftarrow -1$
n_9

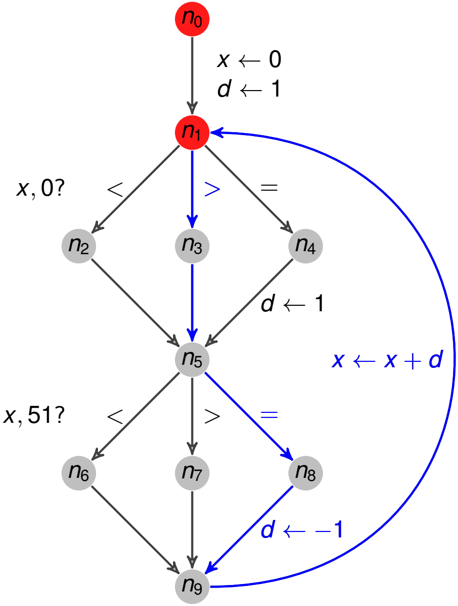

d
1
n_1
x
51

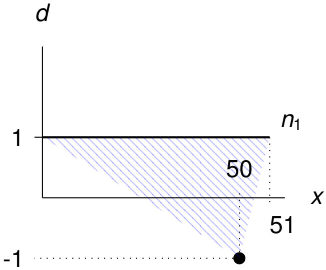
Example


n_0
$x \leftarrow 0$
$d \leftarrow 1$
n_1
$x, 0?$
$<$
$>$
$=$
n_2
n_3
n_4
$d \leftarrow 1$
n_5
$x \leftarrow x + d$
$x, 51?$
$<$
$>$
$=$
n_6
n_7
n_8
$d \leftarrow -1$
n_9

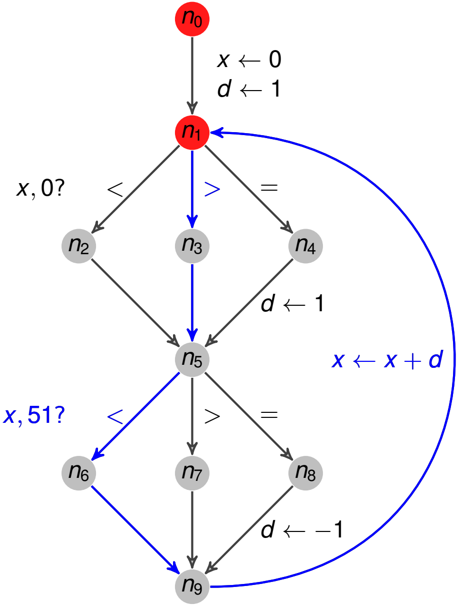

d
1
n_1
50
x
51
-1

Example

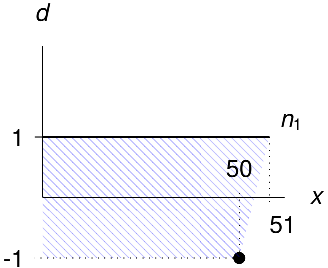

n_0
$x \leftarrow 0$
$d \leftarrow 1$
n_1
$x, 0?$
$<$
$>$
$=$
n_2
n_3
n_4
$d \leftarrow 1$
n_5
$x \leftarrow x + d$
$x, 51?$
$<$
$>$
$=$
n_6
n_7
n_8
$d \leftarrow -1$
n_9


d
1
n_1
50
x
51
-1

Example



n_0
$x \leftarrow 0$
$d \leftarrow 1$
n_1
$x, 0?$
$<$
$>$
$=$
n_2
n_3
n_4
$d \leftarrow 1$
n_5
$x \leftarrow x + d$
$x, 51?$
$<$
$>$
$=$
n_6
n_7
n_8
$d \leftarrow -1$
n_9



d
n_1
1
50
x
51
-1

Summary

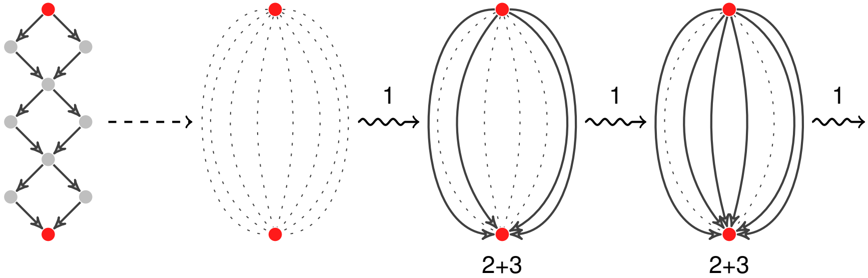
Our Contribution

We apply Guided Static Analysis over the reduced multigraph

Algorithm

3 phases:

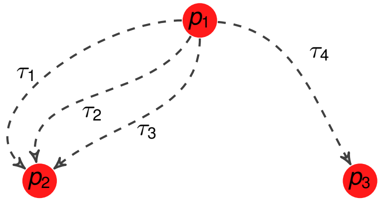
1. Compute new feasible paths
2. Path Focusing on the subset of the multigraph
3. Narrowing iterations



Computing New Paths

We store the set P of paths in a BDD.

New feasible paths starting at p_1:



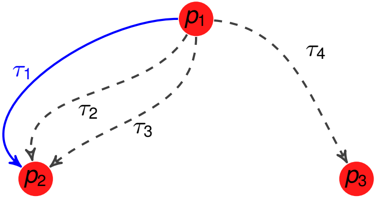
$p_2 : X_2$

$p_3 : X_3$

Computing New Paths

We store the set P of paths in a BDD.

New feasible paths starting at p_1:



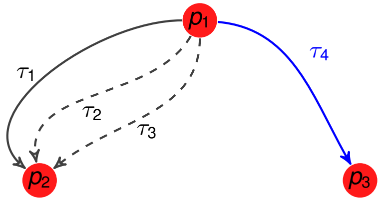
$p_2 : X_2 \sqcup \tau_1(X_1)$

$p_3 : X_3$

Computing New Paths

We store the set P of paths in a BDD.

New feasible paths starting at p_1:



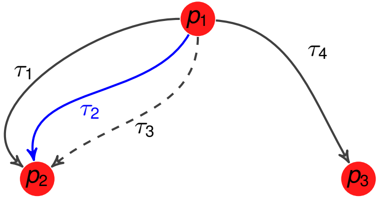
$p_2 : X_2 \sqcup \tau_1(X_1)$

$p_3 : X_3 \sqcup \tau_4(X_1)$

Computing New Paths

We store the set P of paths in a BDD.

New feasible paths starting at p_1:



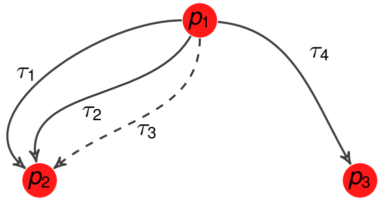
$p_2 : X_2 \sqcup \tau_1(X_1) \sqcup \tau_2(X_1)$

$p_3 : X_3 \sqcup \tau_4(X_1)$

Computing New Paths

We store the set P of paths in a BDD.

New feasible paths starting at p_1:



$p_2 : X_2 \sqcup \tau_1(X_1) \sqcup \tau_2(X_1)$

$p_3 : X_3 \sqcup \tau_4(X_1)$

τ_3 feasible, but:

$$\tau_3(X_1) \subset X_2 \sqcup \tau_1(X_1) \sqcup \tau_2(X_1)$$

Ascending Iterations

Path Focusing algorithm on the multigraph

But the formula is conjoined with P (subgraph):

$$f(p_i) \wedge P$$

Ascending Iterations

Path Focusing algorithm on the multigraph

But the formula is conjoined with P (subgraph):

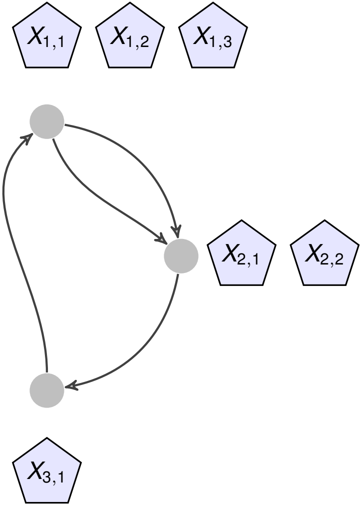
$$f(p_i) \wedge P$$

We also conjoin the formula with P for narrowing iterations...

Summary

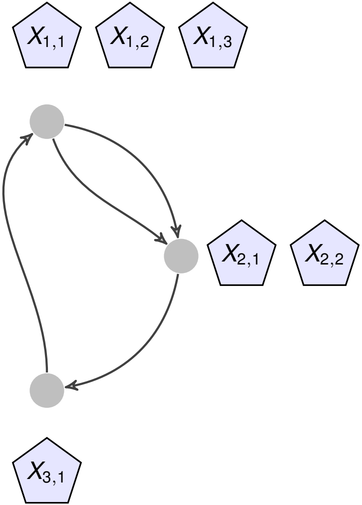
1. Introduction: Weakness of the standard approach & Guided Static Analysis
2. Using SMT-solving to focus new paths
3. Combining Both Techniques
4. Computing Disjunctive Invariants

Disjunctive Invariants



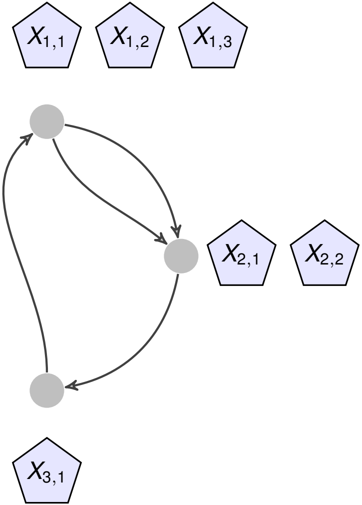
- How to choose the disjunct and the path to focus on?

Disjunctive Invariants



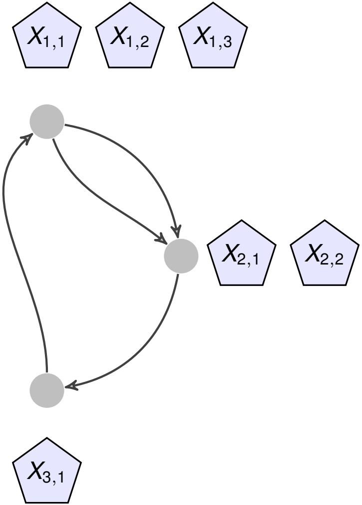
- How to choose the disjunct and the path to focus on?
  - **Use SMT**

Disjunctive Invariants



- How to choose the disjunct and the path to focus on?
  - **Use SMT**
- How to choose which disjunct to join with?

Disjunctive Invariants



- How to choose the disjunct and the path to focus on?
  - **Use SMT**
- How to choose which disjunct to join with?
  - **Gulwani & Zuleger - PLDI 2010**

Using SMT to Focus Path and Disjunct

"Does there exist a path starting in one disjunct, that arrives in a state outside all disjuncts ?"

$$
\begin{aligned}
g(p_i) \quad = \quad & \rho \wedge b_i^s \wedge \bigwedge_{\substack{j \in P_R \\ j \neq i}} \neg b_j^s \\
& \wedge \bigvee_{1 \leq k \leq m_i} \left(d_k \wedge X_{i,k} \wedge \bigwedge_{l \neq k} \neg d_l \right) \\
& \wedge \bigvee_{j \in Succ(i)} \left(b_j^d \wedge \bigwedge_{1 \leq k \leq m_i} (\neg X_{j,k}) \right)
\end{aligned}
$$

One starting disjunct

Not in any destination disjunct

In the model, $d_k = true \implies$ we use $X_{i,k}$ as the starting disjunct.

Gulwani & Zuleger's Technique

Gulwani & Zuleger: The Reachability Bound Problem - PLDI'10

Disjunctive invariant for p_i: $\bigvee_{1 \leq j \leq m_i} X_{i,j}$

- $\delta_i \in [1, m_i]$
- mapping function $\sigma_i : [1, m_i] \times [1, n_i] \mapsto [1, m_i]$

$X_{i,\delta_i} \leftarrow$ initial states

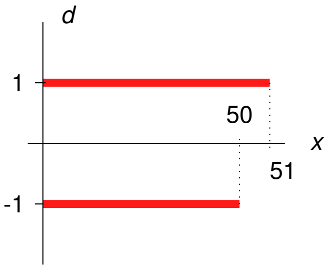
The image of the j-th disjunct $X_{i,j}$ by the k-th path $\tau_{i,k}$ is joined with $X_{i,\sigma_i(j,k)}$.

σ is computed dynamically (See Gulwani & Zuleger's paper)

Example

```
int x = 0;
int d = 1;

while (true) {
    if (x == 0) d=1;
    if (x == 51) d=-1;
    x +=d;
}
```

$$(d = 1 \wedge 0 \leq x \leq 51)$$
$$\vee(d = -1 \wedge 0 \leq x \leq 50)$$



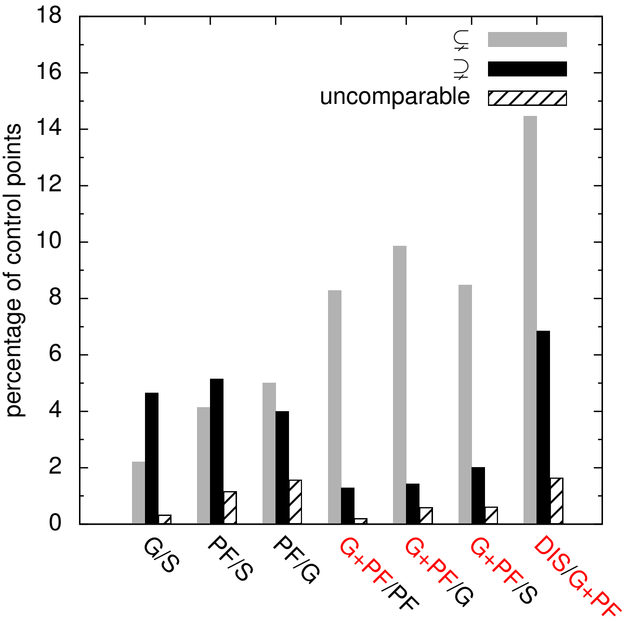
Experiments

These techniques are implemented in PAGAI: a prototype of static analyzer.

- LLVM IR as input
- Apron Library for the abstract domains
- SMT-lib 2 interface, Microsoft Z3

In TAPAS'12:
PAGAI: a Path Sensitive Static Analyser; Henry, Monniaux, Moy

Experiments on GNU programs and WCET benchmarks


percentage of control points
0
2
4
6
8
10
12
14
16
18
$\subsetneq$
$\supsetneq$
uncomparable
G/S
PF/S
PF/G
G+PF/PF
G+PF/G
G+PF/S
DIS/G+PF

Time

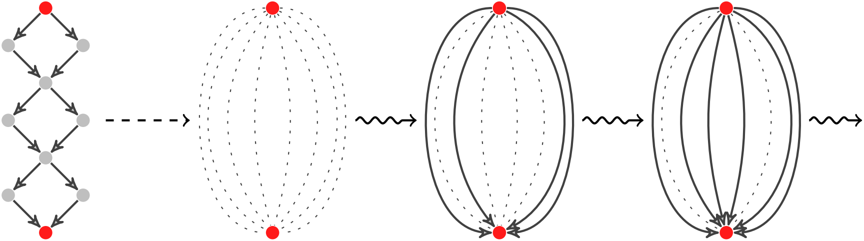
	Size		Execution time (seconds)				
Name	kLOC	$\|P_R\|$	**S**	**G**	**PF**	**G+PF**	**DIS**
a2ps-4.14	55	2012	23	74	34	115	162
gawk-4.0.0	59	902	15	46	12	40	50
gnuchess-6.0.0	38	1222	50	220	81	312	351
gnugo-3.8	83	2801	77	159	92	766	1493
grep-2.9	35	820	41	85	22	65	122
gzip-1.4	27	494	22	268	91	303	230
lapack-3.3.1	954	16422	294	3740	3773	8159	10351
make-3.82	34	993	67	108	53	109	257
tar-1.26	73	1712	37	218	115	253	396

Table: Execution times

Conclusion

- Path distinction avoids loss of precision due to join operators.
- Explicit exhaustive enumeration of paths can be avoided using SMT.
- This idea can be applied / combined with many existing techniques.

Questions ?



Dynamic Construction of σ

M: maximum number of disjunct
m_i: current number of disjunct

When $\sigma_i(j,k)$ is undefined:

1. if $\exists j', \tau_{i,k}(X_{i,j}) \sqcup X_{i',j'} = \tau_{i,k}(X_{i,j}) \cup X_{i',j'}$, we assign $\sigma_i(j,k)$ to j'
2. else:
   - if $m_i < M$, we increment m_i and define $\sigma_i(j,k) = m_i$
   - if $m_i = M$, we define $\sigma_i(j,k) = M$