On Abstraction Refinement for Program Analyses in Datalog

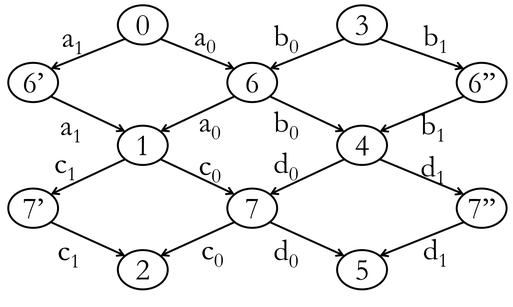
Xin Zhang, Ravi Mangal, Radu Grigore, Mayur Naik, Hongseok Yang

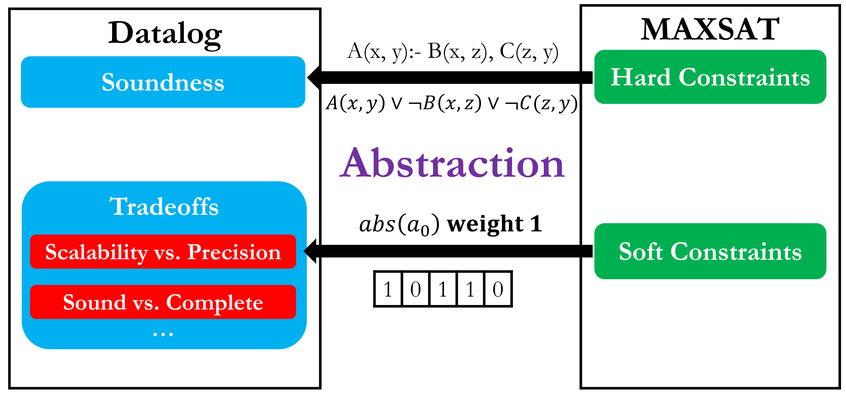
UNIVERSITY OF OXFORD

Pointer Analysis As Graph Reachability

```
f(){                          g(){
   v1 = new ...;                 v4 = new ...;
   v2 = id1(v1);                 v5 = id1(v4);
   v3 = id2(v2);                 v6 = id2(v5);
q2:assert(v3!= v1);           q1:assert(v6!= v1);
}                             }

id1(v){return v;}             id2(v){return v;}
```





Encode As MAXSAT

Hard constraints:

$$path(0,0) \wedge$$
$$(path(0,6) \vee \neg path(0,0) \vee \neg abs(a_0)) \wedge$$
$$(path(0,1) \vee \neg path(0,6) \vee \neg abs(a_0)) \wedge$$
$$(path(0,7) \vee \neg path(0,1) \vee \neg abs(c_0)) \wedge$$
$$(path(0,4) \vee \neg path(0,6) \vee \neg abs(b_0)) \wedge$$
$$\ldots$$

Soft constraints:

$$(abs(a_0)\ \textbf{weight 1}) \wedge (abs(b_0)\ \textbf{weight 1}) \wedge$$
$$(abs(c_0)\ \textbf{weight 1}) \wedge (abs(d_0)\ \textbf{weight 1}) \wedge$$
$$(\neg path(0,2)\ \textbf{weight 5}) \wedge (\neg path(0,5)\ \textbf{weight 5})$$

Solution:

$path(0,0) = \text{true}$, $path(0,6) = \text{false}$, $path(0,1) = \text{false}$, $path(0,4) = \text{false}$, $path(0,7) = \text{false}$, $path(0,2) = \text{false}$, $path(0,5) = \text{false}$, $\boldsymbol{abs(a_0) = \text{false}}$, $\boldsymbol{abs(b_0) = \text{true}}$, $\boldsymbol{abs(c_0) = \text{true}}$, $\boldsymbol{abs(d_0) = \text{true}}$.

Graph Reachability in Datalog

Input relations:

edge(i, j, n), abs(n)

Output relations:

path(i, j)

Rules:

path(i, i).

path(i, j) :- path(i, k), edge(k, j, n), abs(n).

Input tuples:

edge(0, 6, a_0), edge(0, 6', a_1), edge(3, 6, b_0),

…

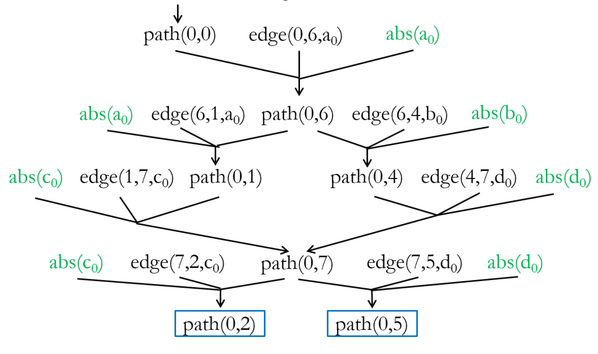
$abs(a_0) \oplus abs(a_1)$, $abs(b_0) \oplus abs(b_1)$,
$abs(c_0) \oplus abs(c_1)$, $abs(d_0) \oplus abs(d_1)$.

Query Tuple	Original Query
q_1: path(0, 5)	`assert(v6!= v1)`
q_2: path(0, 2)	`assert(v3!= v1)`

Derivation As Counterexample

$abs(a_0)$
$abs(b_0)$
$abs(c_0)$
$abs(d_0)$

path(0, 0).
path(0, 6) :- path(0, 0), edge(0, 6, a_0), $abs(a_0)$.
path(0, 1) :- path(0, 6), edge(6, 1, a_0), $abs(a_0)$.
path(0, 7) :- path(0, 1), edge(1, 7, c_0), $abs(c_0)$.
path(0, 2) :- path(0, 7), edge(7, 2, c_0), $abs(c_0)$.
path(0, 4) :- path(0, 6), edge(6, 4, b_0), $abs(b_0)$.
path(0, 7) :- path(0, 4), edge(4, 7, d_0), $abs(d_0)$.
path(0, 5) :- path(0, 7), edge(7, 5, d_0), $abs(d_0)$.
…



Experiment Result

	queries			abstraction size		iterations
	total	resolved				
		current	baseline	final	max	
toba-s	7	7	0	170	18K	10
javasrc-p	46	46	0	470	18K	13
weblech	5	5	2	140	31K	10
hedc	47	47	6	730	29K	18
antlr	143	143	5	970	29K	15
luindex	138	138	67	1K	40K	26
lusearch	322	322	29	1K	39K	17
schroeder-m	51	51	25	450	58K	15

k = 4, 3h28m

k = 3, 590s

k = 2, 214s

k = 1, 153s