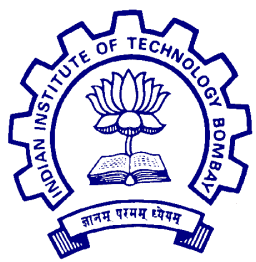
Flow and Context Sensitive Points-to Analysis using Higher Order Reachability

Pritam Gharat (113050036)

Department of Computer Science and Engineering,
Indian Institute of Technology, Bombay

April 2016

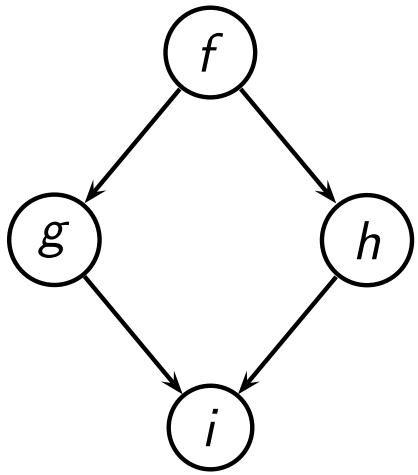


Motivation

- Pointer Analysis
  - which data is read?
  - which data is written?
  - which function is being called?
- Bottom-Up Interprocedural Approach
  - For top-down approach, a procedure is analyzed for every call
  - For bottom-up approach, a procedure is analyzed only once

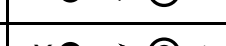
Motivation

- Pointer Analysis
  - which data is read?
  - which data is written?
  - which function is being called?
- Bottom-Up Interprocedural Approach
  - For top-down approach, a procedure is analyzed for every call
  - For bottom-up approach, a procedure is analyzed only once

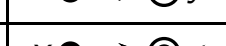
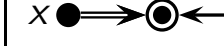


Call Graph

Higher Order Reachability

Pointer assignment	The effect of pointer assignment on memory		Edge traversals for reaching the shared node		Higher order path in M
	Graph	Constraint	From x	From y	
$x = \&y$		$M^1\{x\} = M^0\{y\}$	1 edge	0 edge	$x \overset{1,0}{\rightsquigarrow} y$
$x = y$		$M^1\{x\} = M^1\{y\}$	1 edge	1 edge	$x \overset{1,1}{\rightsquigarrow} y$
$x = *y$		$M^1\{x\} = M^2\{y\}$	1 edge	2 edges	$x \overset{1,2}{\rightsquigarrow} y$
$*x = y$		$M^2\{x\} = M^1\{y\}$	2 edges	1 edge	$x \overset{2,1}{\rightsquigarrow} y$

Higher Order Reachability

Pointer assignment	The effect of pointer assignment on memory		Edge traversals for reaching the shared node		Higher order path in M
	Graph	Constraint	From x	From y	
$x = \&y$		$M^1\{x\} = M^0\{y\}$	1 edge	0 edge	$x \overset{1,0}{\rightsquigarrow} y$
$x = y$		$M^1\{x\} = M^1\{y\}$	1 edge	1 edge	$x \overset{1,1}{\rightsquigarrow} y$
$x = *y$		$M^1\{x\} = M^2\{y\}$	1 edge	2 edges	$x \overset{1,2}{\rightsquigarrow} y$
$*x = y$		$M^2\{x\} = M^1\{y\}$	2 edges	1 edge	$x \overset{2,1}{\rightsquigarrow} y$

- $x \overset{i,j}{\rightsquigarrow} y \in M \Rightarrow M^i\{x\} = M^j\{y\}$
- Eliminates the need of explicit placeholders
- Information from the caller not required

Results: Effectiveness of Summary Flow Functions (HRGs)

- Reusability
- Compactness
- % of Context Independent Information

Program	# of call sites	# of procs.	Proc. count for different buckets of # of calls (reuse of HRGs)				# of procs. for different sizes of HRG in terms of the number of edges						# of procs. for different % of context ind. info. (for non-empty HRGs)			
			2-5	5-10	10-20	20+	0	1-2	3-4	5-8	9-50	50+	0-20	20-40	40-60	60+
lbm	30	19	5	0	0	0	13	4	2	0	0	0	3	0	0	3
mcf	29	23	11	0	0	0	10	5	2	3	2	1	5	1	1	6
libquantum	277	80	24	11	4	3	42	10	7	12	9	0	20	12	1	5
bzip2	288	89	35	7	2	1	62	13	4	5	5	0	26	0	0	1
milc	782	190	60	15	9	1	157	11	19	2	7	0	6	10	9	14
sjeng	726	133	46	20	5	6	99	20	6	3	5	0	3	4	10	17
hmmer	1328	275	93	33	22	11	167	56	20	15	15	2	54	20	11	23
h264ref	2393	566	171	60	22	16	419	76	23	15	30	3	54	13	27	53
gobmk	9379	2697	317	110	99	134	1374	93	8	1083	97	42	41	1192	39	51

Results: Effectiveness of Summary Flow Functions (HRGs)

- Reusability
- Compactness
- % of Context Independent Information

Program	# of call sites	# of procs.	Proc. count for different buckets of # of calls (reuse of HRGs)				# of procs. for different sizes of HRG in terms of the number of edges						# of procs. for different % of context ind. info. (for non-empty HRGs)			
			2-5	5-10	10-20	20+	0	1-2	3-4	5-8	9-50	50+	0-20	20-40	40-60	60+
lbm	30	19	5	0	0	0	13	4	2	0	0	0	3	0	0	3
mcf	29	23	11	0	0	0	10	5	2	3	2	1	5	1	1	6
libquantum	277	80	24	11	4	3	42	10	7	12	9	0	20	12	1	5
bzip2	288	89	35	7	2	1	62	13	4	5	5	0	26	0	0	1
milc	782	190	60	15	9	1	157	11	19	2	7	0	6	10	9	14
sjeng	726	133	46	20	5	6	99	20	6	3	5	0	3	4	10	17
hmmer	1328	275	93	33	22	11	167	56	20	15	15	2	54	20	11	23
h264ref	2393	566	171	60	22	16	419	76	23	15	30	3	54	13	27	53
gobmk	9379	2697	317	110	99	134	1374	93	8	1083	97	42	41	1192	39	51

Results: Effectiveness of Summary Flow Functions (HRGs)

- Reusability
- Compactness
- % of Context Independent Information

Program	# of call sites	# of procs.	Proc. count for different buckets of # of calls (reuse of HRGs)				# of procs. for different sizes of HRG in terms of the number of edges						# of procs. for different % of context ind. info. (for non-empty HRGs)			
			2-5	5-10	10-20	20+	0	1-2	3-4	5-8	9-50	50+	0-20	20-40	40-60	60+
lbm	30	19	5	0	0	0	13	4	2	0	0	0	3	0	0	3
mcf	29	23	11	0	0	0	10	5	2	3	2	1	5	1	1	6
libquantum	277	80	24	11	4	3	42	10	7	12	9	0	20	12	1	5
bzip2	288	89	35	7	2	1	62	13	4	5	5	0	26	0	0	1
milc	782	190	60	15	9	1	157	11	19	2	7	0	6	10	9	14
sjeng	726	133	46	20	5	6	99	20	6	3	5	0	3	4	10	17
hmmer	1328	275	93	33	22	11	167	56	20	15	15	2	54	20	11	23
h264ref	2393	566	171	60	22	16	419	76	23	15	30	3	54	13	27	53
gobmk	9379	2697	317	110	99	134	1374	93	8	1083	97	42	41	1192	39	51

Results: Effectiveness of Summary Flow Functions (HRGs)

- Reusability
- Compactness
- % of Context Independent Information

Program	# of call sites	# of procs.	Proc. count for different buckets of # of calls (reuse of HRGs)				# of procs. for different sizes of HRG in terms of the number of edges						# of procs. for different % of context ind. info. (for non-empty HRGs)			
			2-5	5-10	10-20	20+	0	1-2	3-4	5-8	9-50	50+	0-20	20-40	40-60	60+
lbm	30	19	5	0	0	0	13	4	2	0	0	0	3	0	0	3
mcf	29	23	11	0	0	0	10	5	2	3	2	1	5	1	1	6
libquantum	277	80	24	11	4	3	42	10	7	12	9	0	20	12	1	5
bzip2	288	89	35	7	2	1	62	13	4	5	5	0	26	0	0	1
milc	782	190	60	15	9	1	157	11	19	2	7	0	6	10	9	14
sjeng	726	133	46	20	5	6	99	20	6	3	5	0	3	4	10	17
hmmer	1328	275	93	33	22	11	167	56	20	15	15	2	54	20	11	23
h264ref	2393	566	171	60	22	16	419	76	23	15	30	3	54	13	27	53
gobmk	9379	2697	317	110	99	134	1374	93	8	1083	97	42	41	1192	39	51