A study of set-sharing analysis via cliques

J. Navas,[2] F. Bueno,[1] and M. Hermenegildo[1,2]

CLIP Group

[1] *Department of Computer Science*
Technical University of Madrid (UPM)

[2] *Depts. of Comp. Science and El. and Comp. Eng.,*
U. of New Mexico (UNM)

The 15th Workshop on Logic-Based Methods in Programming Environments, WLPE'05
Sitges (Barcelona), Spain, October 5, 2005

Introduction

- Sharing analysis: the tracking of variables shared among terms
- Sharing domain: the most accurate abstract domain defined for tracking sharing
- Set Sharing: variables shared among **sets** of terms
- Pair Sharing: variables shared among **pairs** of terms
- Cliques: a representation for sets of sets of variables (powersets)
- Zaffanella,Bagnara and Hill studied Sharing+cliques for inferring pair-sharing in a bottom-up framework
- We study Sharing+cliques for inferring set-sharing in a top-down framework

The Sharing domain

- Let $V = \{x, y, z\}$ be a set of variables of interest
- Let a substitution $\{x/f(u_1, u_2, v_1, v_2, w), y/g(v_1, v_2, w), z/g(w, w)\}$
- Abstract substitution is $\{\{x\}, \{x, y\}, \{x, y, z\}\}$
  Notation: xyz for $\{x, y, z\}$.
  Say, abstract substitution is $\{x, xy, xyz\}$
  - x represents the occurrence of u_1 and u_2
  - xy represents v_1 and v_2
  - xyz represents w

Cliques

- How to say nothing is known about $V = \{x, y, z\}$
  $\{x, xy, xyz, xz, y, yz, z\}$
- If a sharing substitution includes the powerset of some set C of variables, use C to represent it
  $\{xyz\}$
- We use pairs (cl, sh) of two sharing sets.
  - sh is a sharing substitution
  - each C in cl represents the powerset of C

$$\{x, xy, xyz, xz, y, yz, z, w\} \Leftrightarrow (\{xyz\}, \{w\})$$

- Just a change in representation: no loss of precision implied
- However, the abstract operations for cliques are not precise
- Tradeoff between efficiency and accuracy

Use of Cliques

- Cliques used as alternative representation:
  - A normalization process moves powersets to cliques, producing efficiency gains, and also precision losses.
  - Problem: clique sets are produced in cases in which they are not necessary.
- Widening set-sharing via Cliques:
  - To limit the use of cliques only to the cases where it is necessary in order to avoid analysis running out of memory.
  - The clique representation will be used (only) to guarantee termination of the analysis.
  - It is not trivial: it is not easy to determine beforehand when analysis will need more memory than is available.

Use of Cliques: Widening set-sharing via Cliques

- The choice of a suitable value of the threshold is a key issue.
- This threshold is responsible for triggering widening only for the cases where it is needed.
- We studied two widenings:
  - Fetch: $\bigtriangledown^F(cl, sh) = (cl \cup sh, \emptyset)$
  - Zaffanella,Bagnara and Hill: $\bigtriangledown^n(cl, sh) = (\{C_1, \ldots, C_k\}, sh)$
    where $C_1, \ldots, C_k$ are all the maximal cliques of the graph induced from (cl, sh) and singletons are disregarded.
- $\bigtriangledown^F$ is very aggressive and therefore, not precise.
- $\bigtriangledown^n$ cannot be used in a top-down analysis framework.
- We developed a hybrid approach $\bigtriangledown^{nF}$
  - $\bigtriangledown^n$ is used in unifications
  - $\bigtriangledown^F$ is used in the $extend$ function

Results (cliques as alternative representation)

	Sh		SH^W			**Shfr**		SH^W**fr**		
	T	**P**	**T**	**P**	$\#C$	**T**	**P**	**T**	**P**	$\#C$
append	11	29 (60)	8	44 (60)	4	6	7 (30)	6	7 (30)	0
deriv	35	27 (546)	27	27 (546)	0	27	21 (546)	27	21 (546)	0
mmatrix	13	14 (694)	11	14 (694)	0	9	12 (694)	11	12 (694)	0
qsort	24	30 (1716)	25	30 (1716)	0	25	30 (1716)	27	30 (1716)	0
query	11	35 (501)	13	35 (501)	5	12	22 (501)	14	22 (501)	0
serialize	306	1734 (10531)	90	2443 (10531)	88	61	545 (5264)	55	736 (5264)	41
aiakl	35	145 (13238)	42	145 (13238)	0	37	145 (13238)	43	145 (13238)	0
boyer	369	1688 (4631)	267	1997 (4631)	158	373	1739 (5036)	278	2074 (5036)	163
browse	30	69 (776)	29	69 (776)	0	29	69 (776)	31	69 (776)	0
prolog_read	400	1080 (408755)	465	1080 (408755)	10	425	1050 (408634)	481	1050 (408634)	0
rdtok	325	1350 (11513)	344	1391 (11513)	182	335	1047 (11513)	357	1053 (11513)	2
warplan	3261	8207 (42089)	1430	8191 (26857)	420	1320	3068 (23501)	1264	5705 (25345)	209
zebra	25	280 ($67{\cdot}10^7$)	34	280 ($67{\cdot}10^7$)	0	41	280 ($67{\cdot}10^7$)	42	280 ($67{\cdot}10^7$)	0
ann	2382	10000 ($31{\cdot}10^4$)	802	19544 ($31{\cdot}10^4$)	700	1791	7811 ($40{\cdot}10^4$)	968	14108 ($39{\cdot}10^4$)	510
peephole	831	2210 (12148)	435	2920 (12118)	171	508	1475 (9941)	403	2825 (12410)	135
qplan	-	-	860	$42{\cdot}10^4$ ($38{\cdot}10^5$)	747	-	-	2181	$23{\cdot}10^4$ ($31{\cdot}10^5$)	529
witt	405	858 ($45{\cdot}10^5$)	437	858 ($45{\cdot}10^5$)	25	484	813 ($45{\cdot}10^5$)	451	813 ($45{\cdot}10^5$)	0

Table 1: Precision and Time-efficiency for sharing, clique-sharing, shfr and clique-shfr.

Comments on results

- If normalization causes no change in the representation:
  - analysis is the same as without cliques,
  - with a small extra overhead
- If normalization moves powersets to cliques:
  - efficiency gains and also precision losses,
  - but enables analysis of programs which otherwise ran out of memory.
- Same effects maintained with freeness,
  - efficiency gains are lower,
  - whereas precision gains are higher

Effectiveness

	Sharing+Freeness			Clique-Sharing+Freeness		
	Total	**NF (%)**	**Cov (%)**	**Total**	**NF (%)**	**Cov (%)**
append	1	1 (100)	1 (100)	1	1 (100)	1 (100)
deriv	1	1 (100)	1 (100)	1	1 (100)	1 (100)
qsort	3	3 (100)	3 (100)	3	3 (100)	3 (100)
serialize	5	0 (0)	2 (40)	5	0 (0)	2 (40)
rdtok	22	8 (36)	13 (59)	22	8 (36)	13 (59)
zebra	6	1 (16)	4 (66)	6	1 (16)	4 (66)

Table 2: Accuracy of the non-failure analysis

Results (widening set-sharing via cliques)

	Shfr		SH^W**fr**$+\bigtriangledown^F_{250}$			SH^W**fr**$+\bigtriangledown^{nF}_{250-40}$		
	T	**P**	**T**	**P**	$\#W$	**T**	**P**	$\#W$
append	6	7 (30)	11	7 (30)	0	10	7 (30)	0
deriv	27	21 (546)	48	21 (546)	0	35	21 (546)	0
mmatrix	9	12 (694)	16	12 (694)	0	16	12 (694)	0
qsort	25	30 (1716)	40	30 (1716)	0	43	30 (1716)	0
query	12	22 (501)	23	22 (501)	0	25	22 (501)	0
serialize	61	545 (5264)	74	722 (5264)	6	70	703 (5264)	10
aiakl	37	145 (13238)	63	145 (13238)	6	61	145 (13238)	33
boyer	373	1739 (5036)	561	1744 (5036)	2	536	1743 (5036)	4
browse	29	69 (776)	44	69 (776)	0	42	69 (776)	0
prolog_read	425	1050 (408634)	3419	24856 (1754310)	198	593	1050 (408634)	103
rdtok	335	1047 (11513)	472	1047 (11513)	0	466	1047 (11513)	0
warplan	1320	3068 (23501)	1878	5376 (21586)	42	1394	5121 (20894)	60
zebra	41	280 (671088746)	42	280 (671088746)	1	56	280 (671088746)	48
ann	1791	7811 (401220)	751	16122 (394800)	17	726	16122 (394800)	34
peephole	508	1475 (9941)	453	2827 (12410)	8	512	2815 (12410)	16
qplan	-	-	1722	238426 (3141556)	26	1897	233070 (3126973)	55
witt	484	813 (4545594)	2333	259366 (23378597)	110	736	813 (4545594)	140

Table 3: Precision and Time-efficiency for shfr and clique-shfr with $\bigtriangledown^F_{250}$ and $\bigtriangledown^{nF}_{250-40}$.

Comments on results

- Avoided the use of cliques in some cases in which cliques were (unnecessarily) used.
- Allows executing programs which the Sharing+freeness domain could not because memory capacity was exceeded
- $\bigtriangledown^{nF}$ at least as precise as $\bigtriangledown^{F}$.
- The difference in time efficiency between $\bigtriangledown^{nF}$ and $\bigtriangledown^{F}$ is quite acceptable

Implications of using Top-down framework

- Redefine all abstract operations, in particular the $extend$ function.
- The $extend$ function can make the sharing representation grow too much.
- Thus, the $extend$ function plays a crucial role:
  - The normalization process could not run at the limit.
  - $\bigtriangledown^n$ cannot be used.
  - The choice of the threshold for triggering widening is further complicated.
- Therefore, the $extend$ function is a very important bottleneck that affects strongly the efficiency and accuracy of this domain.

Conclusions

- Studied the problem of efficient, scalable set-sharing analysis.
- Provided the unexplored case of inferring set-sharing information in the context of top-down analysis, with Sharing and Sharing+Freeness.
- Proposed and evaluated several widenings: different levels of precision/efficiency tradeoff.
- Our experimental evaluation supports:
  - Cliques as alternative representation result in limited precision losses while useful efficiency gains are obtained.
  - The hybrid widening results quite useful in practice.
- Therefore, our results contribute to the practical application of top-down set sharing analysis.