Efficient Flow-Sensitive (FS) Points-to Analysis via Flow-Insensitive (FI) Insight

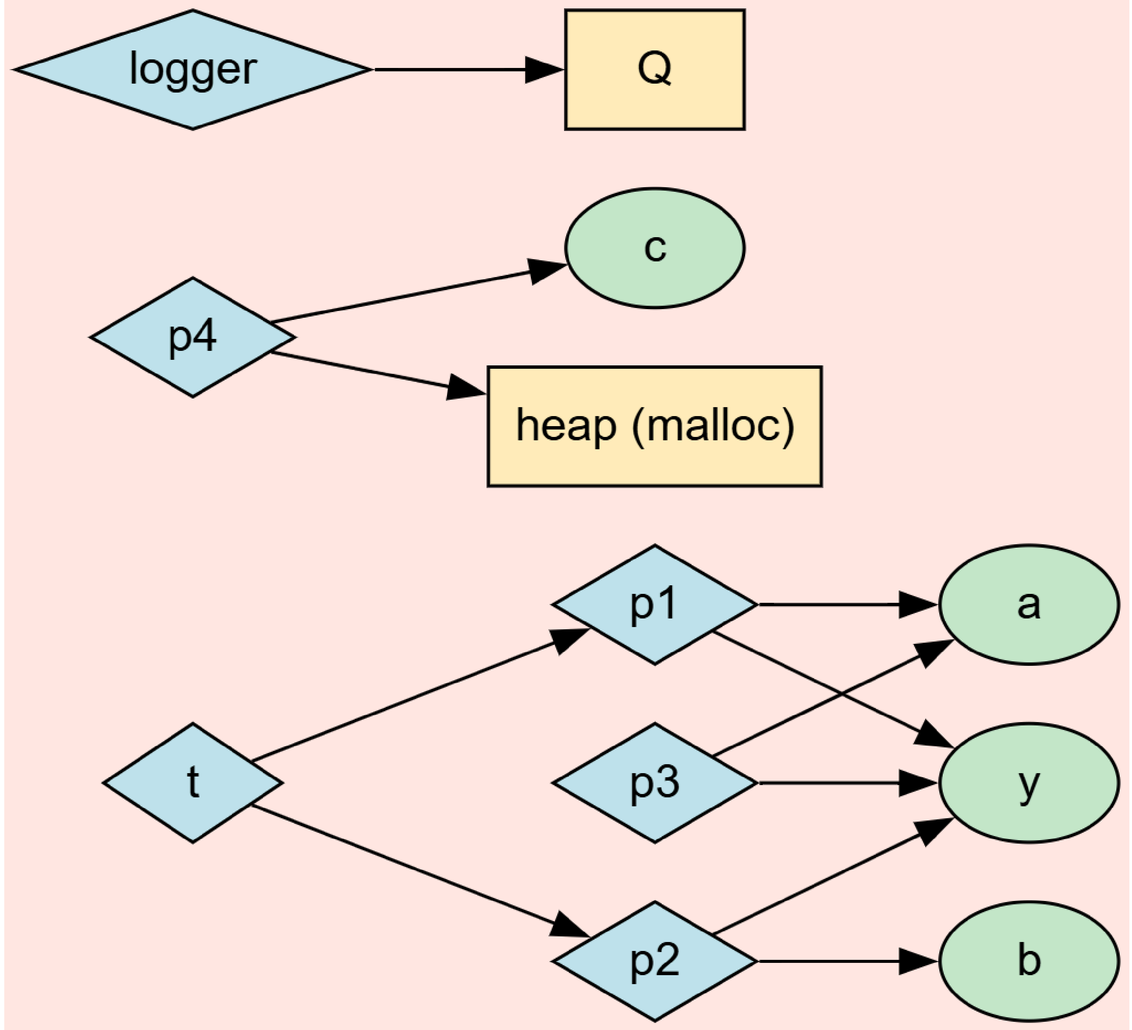
Chaitanya Garg
Guide: Prof. Uday Khedkar
Department of Computer Science and Engineering, IIT Bombay

Source Code

```
1  void main() {
2      int a, b, c;
3      int* p1 = &a;
4      int* p2 = &b;
5      int* p3 = p1;
6
7      P(&p2);
8
9      int* p4;
10     if (c>2) {
11       p4 = &c;
12       P(&p1);
13     }
14     else {
15       p4 = malloc(4);
16     }
17     void (*logger)(int);
18     logger = Q;
19     logger(*p3);
20 }
```

```
21 void P(int** t) {
22     int y;
23     *t = &y;
24 }
25
26 void Q(int x) {
27     printf("%d",x);
28 }
```

Andersen's Points-to-Graph



Issues with FS PTA

- Flow-sensitive analysis maintains a detailed mapping of points-to pairs, which rapidly increases as program size grows.
- Operations like union, difference, and equality on these unstructured pairs are computationally intensive, making the analysis slow and cumbersome.
- The number of points-to pairs grows exponentially with the number of variables and pointer types, severely impacting scalability.

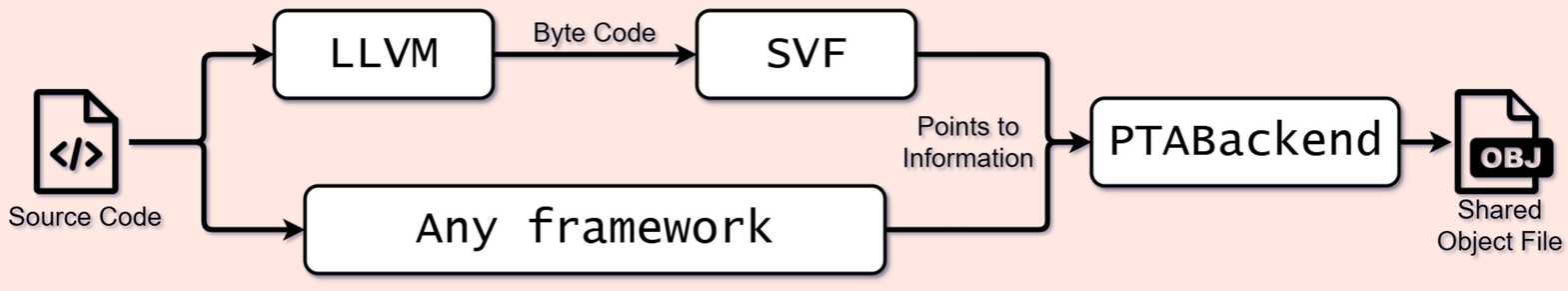
Need for FI PTA

- Flow-insensitive analysis, though less precise, quickly filters the total set of potential points-to pairs.
- By narrowing down the set of candidate pairs, it minimizes the expensive set operations performed later in a more precise analysis.
- This fast pre-filtering step is crucial when handling large programs, as it substantially reduces the number of pairs that need detailed flow-sensitive analysis.

Proposed Solution

- Develop a module that efficiently hashes, stores, and manages points-to pairs, enabling rapid set operations such as union, difference, and equality.
- Provide a set of fast APIs that can be directly utilized by flow-sensitive and other precise analyses, ensuring seamless integration and improved performance.
- By structuring the points-to pairs, the module minimizes the overhead and improves the scalability of analyses that would otherwise process unstructured data.

Pipeline



Future Work

- Investigate seamless integration with existing analysis frameworks to enhance real-world applicability.
- Adapt the module to support additional types of analyses and possibly integrate dynamic analysis features for richer program insights.