Dataflow Analysis in First Order Languages

Emily Blackman

Alternative title: A General Algorithm for Dataflow Analysis of Procedureless Programs

```
@inproceedings{kildall,
        Author = {Gary A. Kildall},
        Title = {A unified approach to global program optimization},
        Booktitle = {Proceedings of the 1st annual ACM SIGPLAN Symposium on Principles of
        Programming Languages},
        Pages = {194-206},
        Year = {1973}
}
```

Summary: Kildall’s paper provides a unified framework for a variety of procedureless program optimization techniques. The main elements of the framework are (1) a transfer function for each control-graph node and (2) a function that describes how information from different nodes is merged together. By configuring these two elements appropriately, the same algorithm can be used for a variety of optimizations such as constant propagation, common subexpression elimination, and register optimization.

Evaluation: This paper is significant because there was not a unified approach before. Kildall explains and proves that his framework can be used for any optimization as long as it meets certain requirements. Thus, his techniques for intraprocedural analysis are still used today.

Alternative Title: Interprocedural Data Flow Analysis Done Right: Two New Methods

```
@book{sharir,
        author    = {Micha Sharir and Amir Pnueli},
        title     = {Two approaches to interprocedural data flow analysis},
        booktitle = {Program Flow Analysis: Theory and Applications},
        editor    = {Muchnick and Jones},
        publisher = {Prentice Hall International},
        year      = {1981}
}
```

Summary: This paper provides two different approaches for performing interprocedural data flow analysis: the functional approach and the call-strings approach. The functional approach involves assuming a conservative transfer function for each procedure in a given program and then using those to perform an iterative dataflow analysis of the program itself which converges to more precise transfer functions for each procedure. The authors prove that their method for finding this transfer function always has a fixed-point solution. They also provide detailed information on the call-strings approach, which works by storing information about pending calls. Both methods provide better guaranteed accuracy than previous methods were able to.

Evaluation: These techniques are the foundation of much of the later work done on interprocedural analysis. They are still used today and, thus, are incredibly influential to static analysis as a field.