3.1 What is Data Flow Analysis

Data-flow analysis is a type of static analysis. It studies how the data (variables) flow across basic blocks, the global pattern of data usage. The example of data-flow problems:

- Reaching definitions: for each program point, what are the definitions can reach
- Available expressions: for each program point, what are the expressions available
- Live variables: for each program point, what are the variables available

Applications of dataflow problems:

- compiler optimizations (global and local optimization)
- find bugs
- debugging
- ...

Input: program source code, semi-lattice, transfer functions (data flow problem)
Output: $S[N]$ (The global properties of data at a program point, also called dataflow facts)

3.2 Reaching Definition – see slides

3.3 Available Expression – see slides

3.4 Live Variables – see slides

3.5 Parametrize the dataflow framework for an instance of the dataflow problem

3.5.1 Lattice model

Reference: Lattice Tutorial by Nenad Jovanovich This tutorial gives an introduction to a number of concepts of lattice theory, especially into those needed for understanding data flow analysis. It starts with the basics, requires only little previous knowledge, and tries to illustrate the concepts with concrete examples.
It covers the following topics:

- Binary Relation
- Partial Order Relation
- Partially Ordered Set, Totally Ordered Set
- Least Upper Bound, Greatest Lower Bound
- Lattice, Complete Lattice, Semi-Lattice

Lattice: define domain of program properties computed by data, flow analysis (a set of potential solutions plus the order of these solutions)

at each program point, the dataflow fact is an element of a lattice, the instruction will model the dataflow fact, $F : L \rightarrow L$, the compositions of such functions over a basic block

if $a < b$, a is less precise, a is a conservative approximation of b

meet operator in lattice defines the most precise element that’s a conservative approximation of the both input elements

it is also a merge function

flow functions (transfer functions: how each node affects the flow value – out(B)) and merge functions over this domain (in(B)) using standard lattice operations

Define a dataflow problem:

- domain of values, meet operators, lattice operations (in(B)), also called dataflow equations
- a set of transfer functions (out(B)) – dataflow equations

### 3.5.2 dataflow algorithms: solving dataflow equations

Solving dataflow equations for all program points: performing dataflow analysis for the entire program, propagating dataflow until fixpoints are reached (stabilize via loops)

What can be shared among dataflow problems – propagation algorithms

- how to merge at the branches
- fix point problems for the loops
- interprocedural analysis
- demand-driven or exhaustive analysis

Determine the direction of dataflow analysis (See the summary for the three examples)

- backward
- forward: for forward problem, considering all possible paths from the entry to the given point, compute ow value for all the paths, and then meet the ow value at the end of the points
Determine how we merge the data flow facts

- may
- must

Efficient data structure for data flow algorithms:

- Bit vectors used to represent sets because we are computing binary information. 1) Does a definition reach a point? T or F 2) Is an expression available/very busy? T or F 3) Is a variable live? T or F
- For each expression, variable, definition we have one bit; intersection and union operations can be implemented using bitwise and & or operations.

Three example algorithms:

properties of dataflow analysis: conservative analysis, sound, safe

Inteprocedural dataflow analysis, IFDS, by Tom Reps et al. [POPL95]