Chapter 5: Dataflow analysis examples

Jonathan Lam

09/19/21

Contents

1 Integer sign analysis

This is carried out much the same as zero analysis.

This can be used to detect errors such as negative index into an array.

2 Constant propagation

This tracks constant values in the program, where possible.

This can be used for optimization, and for general program correctness (to track constant values where possible).

No two non-top elements values are compatible; either a value is known or it is not. (I.e., the join of any element with anything other than itself is top).

Before, we defined the partial order relation and the join operator on single variables; but we can lift these to the entire tuple of all of the variable values using the obvious combinations

3 Reaching definitions

"Reaching definitions analysis determines, for each use of a variable, which assignments to that variable might have set the value seen at that use." It can be used to detect if a variable is undeclared when used.

In this case, the abstract value is a set (which is a lattice).

For the initial state, we can define an initial artificial uninitialized value x_0.

GEN/KILL pattern: way to reason about changes to dataflow analysis by elements that are added (GEN) and removed (KILL)

4 Live variables

"Live variable analysis determines, for each program point, which variables might be used again before they are defined." This can be used for optimization: if a variable is not used after it is defined, it can be removed.

We need to be careful that there may be side effects of an assignment, such as automatic garbage collection of null values.

We perform **backwards analysis**, because we don't know if a variable is live unless it is used at a later point.