

CMSC 22620
Spring 2009

Implementation
of
Computer Languages

Project 4
May 14, 2009

Value numbering
Due: May 22, 2009; 10:00pm

1 Introduction

You are to write another simple optimizer for LambdaIR to perform a simple form of *common subexpression elimination* (CSE) called *value numbering*.¹

2 Value numbering

The basic idea of value numbering is to map variables to a symbolic representation of their value, such that if the symbolic representation of two variables is *congruent*, then the variables are guaranteed to have the same value at runtime and one may be replaced by the other.

To implement value numbering, we define a mapping from variables to integers (their value number) and a mapping from right-hand-side terms to value numbers. The right-hand-side terms consist of an operator (*e.g.*, a primop, allocation, or select) and a list of the value numbers of the arguments. The sample code defines a property on variables for mapping variables to value numbers.

The algorithm assigns value numbers to bound variables in a top-down pass. When a variable can be assigned the number of some other variable, then we mark it as redundant. As we return from the value-numbering pass, we rewrite the program bottom up by replacing redundant right-hand-sides with variables. For example, consider the following LambdaIR fragment:

```
let j = IntAdd(i, 1)
let k = IntAdd(i, 1)
...
```

On the top-down pass, we identify that *k* has the same value number as *j*. As we rewrite the code bottom-up, we replace the binding of *k* with

```
let k = j
```

A subsequent contraction pass will then replace uses of *k* with *j*.

Note: when rewriting the program, you code should update census information. For example, the use count of *i* should be decremented by one when the right-hand-side “IntAdd(*i*, 1)” is replaced with *j* in the above code.

¹Value numbering is explained in Appel’s textbook in Chapter 17, Algorithm 17.7.

3 Extra credit

The conventional form of value numbering will treat function parameters as unknown values, but we can use the notion of congruence to get better estimations. For example, consider the following LambdaIR fragment:

```
fun f (x, h) =  
  let a = #0 (x)  
  let b = #1 (x)  
  ...  
  let y = [a, b]  
  ...
```

The bindings of `a` and `b` imply that `x` could be mapped to `[a, b]`, which is congruent with `y`. For extra credit, extend your value numbering algorithm to infer the structure of parameters.