

CSCD70 Compiler Optimization

Tutorial #3 Dataflow Analysis

Bojian Zheng

bojian@cs.toronto.edu

Department of Computer Science, University of Toronto

Acknowledgement: Thanks to Professor Gennady Pekhimenko, Professor Nandita Vijaykumar and students from previous offerings of CSCD70.

Abstract

In this tutorial, we will be discussing the following topics:

- ▶ Dataflow Analysis Case Study: Available Expressions
- ▶ Assignment 1 Q & A

Available Expressions

- ▶ Why do we want to study **Available Expressions**?

- ▶ Global Common Subexpressions, e.g.,

```
if (...) {  
    x = m + n;  
} else {  
    y = m + n;  
}  
z = m + n; // 'm + n' has already been computed,  
           therefore is redundant
```

- ▶ What happens if $m + n$ is NOT computed in the **else** branch?
 - ▶ Need a rigorous way for arguing about “redundancy”.
 - ☞ Available Expressions

- ▶ In **Available Expressions**, we care about **expressions**.
⇒ **Domain**: Sets of Expressions

Available Expressions

Terminologies

- ▶ An expression $x \oplus y$ is **available** at a point p if every path from the entry node to p evaluates $x \oplus y$.
- ▶ A block **generates** expression $x \oplus y$ if it definitely evaluates $x \oplus y$ and does not subsequently define x or y .
- ▶ A block **kills** expression $x \oplus y$ if it assigns (or may assign) x or y and does not subsequently recompute $x \oplus y$.
- ▶ E.g.,

```
x = y + 1; // generates 'y + 1'  
y = m + n; // generates 'm + n', also kills 'y + 1'
```

⇒ **Transfer Function:** $f_B := \text{gen}_B \cup (x - \text{kill}_B)$

Available Expressions

- ▶ What should be the **Direction** of analysis?
 - ▶ In Available Expressions, we eliminate an expression because it has been computed **in the past**.
 - ▶ In Live Variables, we eliminate a variable because it is not going to be used **in the future**.
- ⇒ Available Expressions is **Forward** while Live Variables is **Backward**.

⇒ **Direction:** Forward

OUT Equation: $\text{OUT}[B] = f_B(\text{IN}[B])$

IN Equation: $\text{IN}[B] = \bigwedge_{p \in \text{pred}(B)} \text{OUT}[p]$

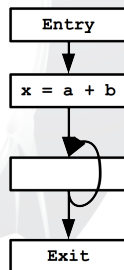
- ▶ What should be the **Meet Operator** between different paths?

Available Expressions

- ▶ What should be the **Initial Condition** and **Boundary Condition**?

IC: $\text{OUT}[B] = \mathbb{U}$, BC: $\text{OUT}[\text{entry}] = \emptyset$

- ▶ What does Initial and Boundary mean?
- ▶ How is this related to the Direction and Meet Operator we have discussed?
- ▶ What happens if we assume different conditions? Will they affect the results?



Available Expressions

Available Expressions Dataflow Analysis

Domain	Sets of Expressions
Direction	Forward
Transfer Function	$f_B := \text{gen}_B \cup (x - \text{kill}_B)$
Meet Operator	$\wedge := _$
OUT Equation	$\text{OUT}[B] = f_B(\text{IN}[B])$
IN Equation	$\text{IN}[B] = \wedge_{p \in \text{pred}(B)} \text{OUT}[p]$
Initial Condition	$\text{OUT}[B] = \mathbb{U}$
Boundary Condition	$\text{OUT}[\text{entry}] = \emptyset$

👉 **Homework Assignment:** Available Expressions