

Static Program Analysis

Program Slicing

Steven Vormwald

02/19/2008

Outline

- Definitions
- Program Representation
- Example – Wisconsin Program-Slicing Tool
- Future Work
- Questions

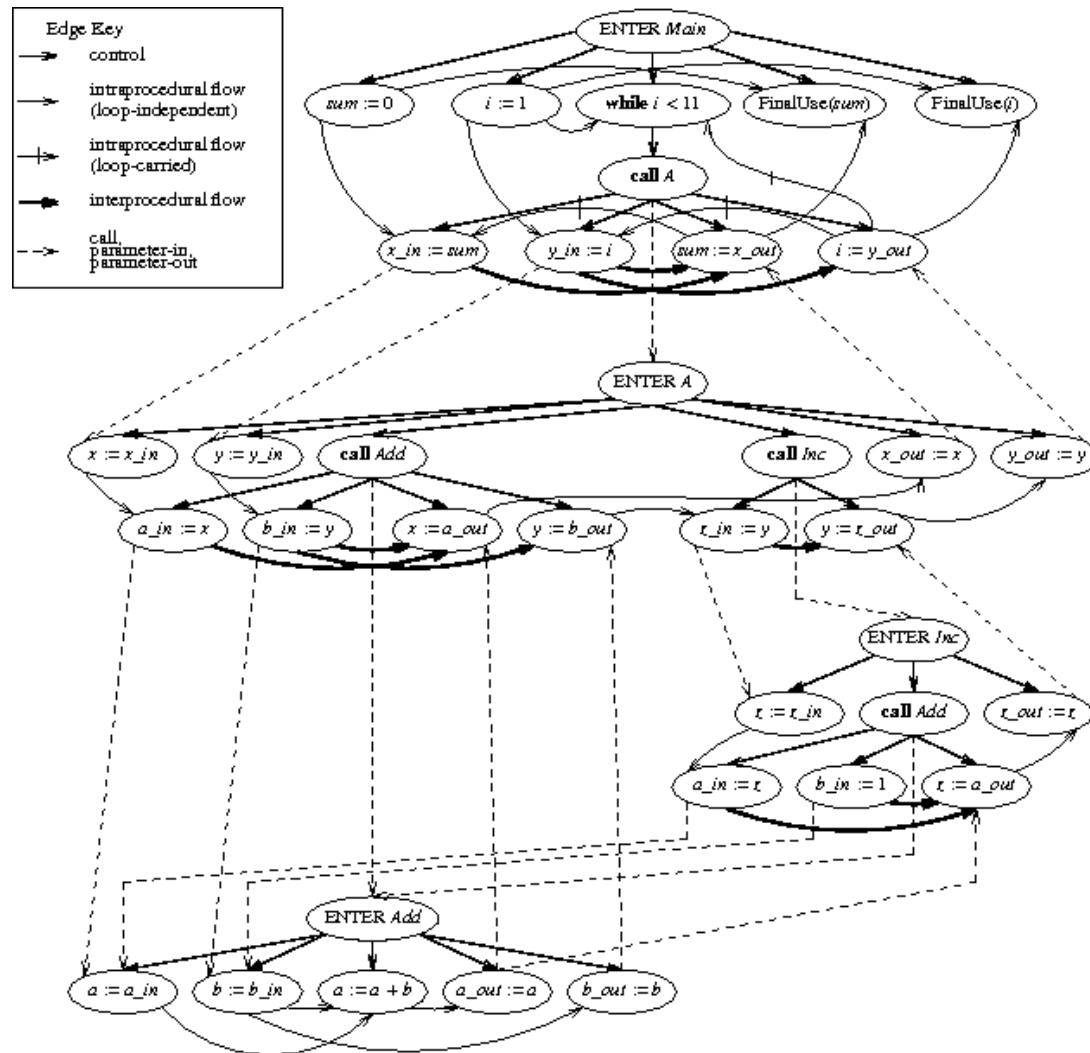
Definitions

- Program Slicing
 - Determine what statements affect/are affected by a given state of a program.
 - Forward Slicing
 - What later statements are affected by this state?
 - Backward Slicing
 - What earlier statements affected this state?

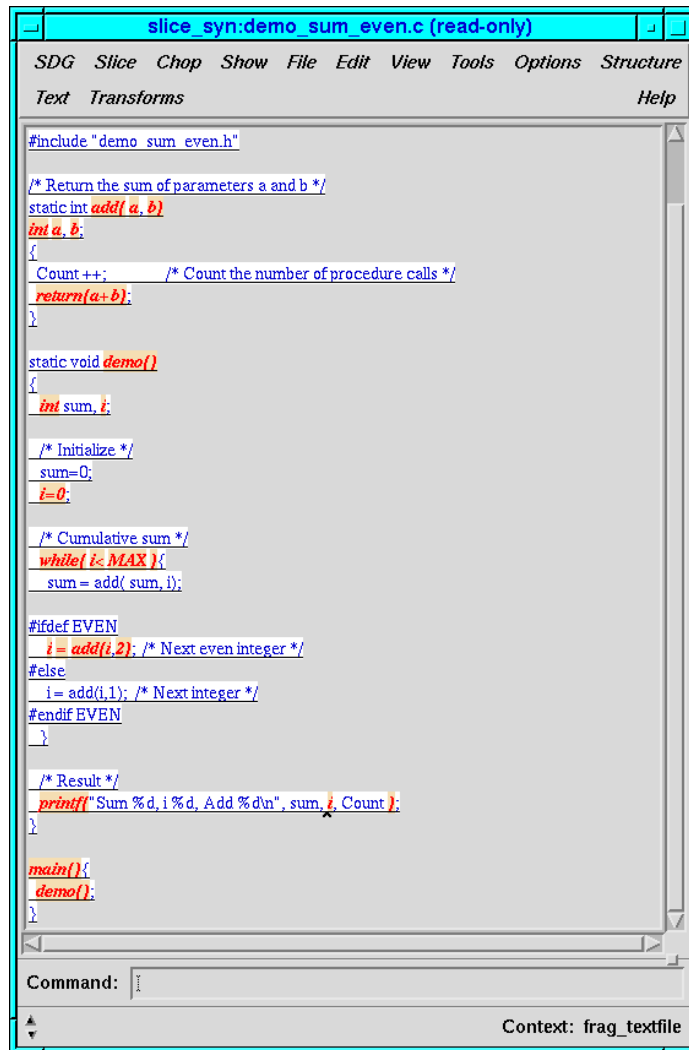
Definitions

- **Control-Flow Graph (CFG)**
 - A graph showing the possible execution paths through a procedure.
- **Program Dependence Graph (PDG)**
 - A graph showing data and control dependences between statements in a procedure.
- **System Dependence Graph (SDG)**
 - An extension to PDG that includes dependences across function calls.

Program Representation (SDG)



Example



The screenshot shows a window titled "slice_syn:demo_sum_even.c (read-only)". The menu bar includes "SDG", "Slice", "Chop", "Show", "File", "Edit", "View", "Tools", "Options", "Structure", "Text", "Transforms", and "Help". The main text area contains the following C code:

```
#include "demo_sum_even.h"

/* Return the sum of parameters a and b */
static int add( a, b)
int a, b;
{
    Count++; /* Count the number of procedure calls */
    return(a+b);
}

static void demo()
{
    int sum, i;

    /* Initialize */
    sum=0;
    i=0;

    /* Cumulative sum */
    while( i < MAX ){
        sum = add( sum, i);
    }

    #ifdef EVEN
        i = add(i, 2); /* Next even integer */
    #else
        i = add(i, 1); /* Next integer */
    #endif EVEN
}

/* Result */
printf("Sum %d, i %d, Add %d\n", sum, i, Count );
}

main(){
    demo();
}
```

At the bottom, there is a "Command:" field and a "Context: frag_textfile" label.

- Wisconsin Program-Slicing Tool
 - Supports forward and backward slicing of C programs.
 - Has been used on C sources containing over 51000 lines.

Future Work

(By Spring Break)

- Memory Aliasing
 - These slicing techniques all specify that the algorithms assume there are no aliasing problems, e.g. a call-by-reference system where the call $f(x,x)$ occurs. Aliasing is too common to ignore in languages such as C.
- Concurrent Programs
 - These slicing techniques work well for systems with a single thread of execution. What changes are necessary to analyze multi-threaded systems?

Questions?