

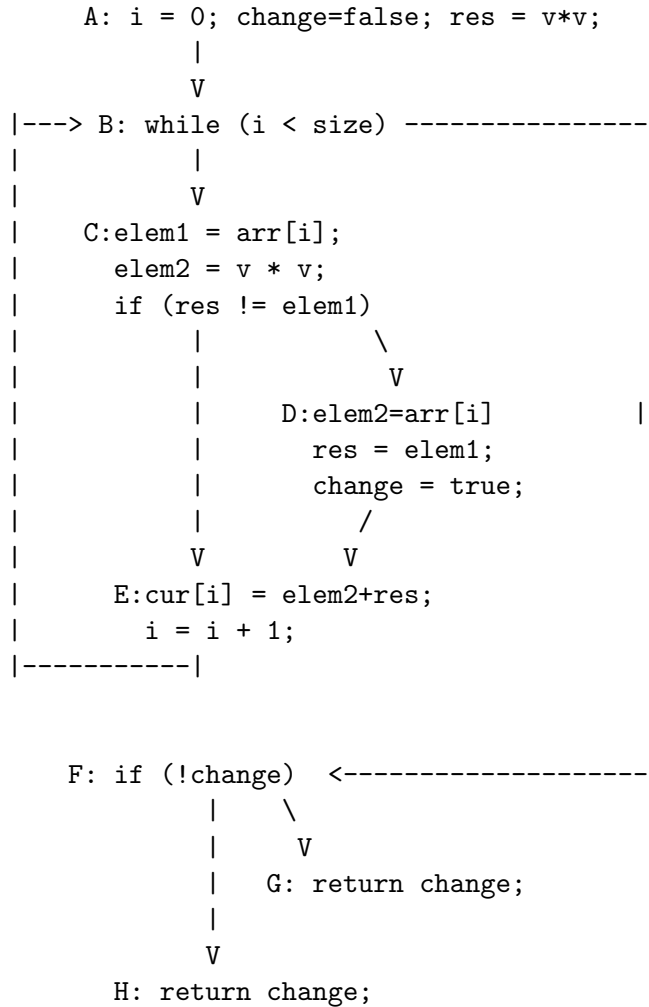
Homework 6: Machine Independent Optimization

Solution

1. Eliminate useless code in the given code. Build a control-flow graph for function *foo* after the optimization.

```
float cur[500];
bool foo(float *arr, int size, float v)
{
    float elem1, elem2, res;
    bool change;
    int i;
    i = 0; change=false; res = v * v;
    while (i < size) {
        elem1 = arr[i];
        elem2 = v * v;
        if (res != elem1)
            { elem2 = arr[i]; res=elem1; change = true; }
        cur[i] = elem2+res;
        i = i + 1;
    }
    if (!change) { return change;}
    return change;
}
```

Control flow graph:



2. Compute the set of available expressions at the entry and exit of each basic block. Eliminate redundant expressions in the given code.

Expressions to consider: (1) $v * v$, (2) $i < size$, (3) $arr[i]$, (4) $elem2 + res$, (5) $res != elem1$, (6) $i + 1$

	Kill	DEexp	Ain	Aout	Ain	Aout
A	2, 3, 4, 5, 6	1	\emptyset	1	\emptyset	1
B	\emptyset	2	\emptyset	2	1	1, 2
C	4, 5	1, 3, 5	2	1, 2, 3, 5	1, 2	1, 2, 3, 5
D	4, 5	3	1, 2, 3, 5	1, 2, 3	1, 2, 3, 5	1, 2, 3
E	2, 3, 6	4	1, 2, 3	1, 4	1, 2, 3	1, 4
F	\emptyset	\emptyset	2	2	1, 2	1, 2
G	\emptyset	\emptyset	2	2	1, 2	1, 2
H	\emptyset	\emptyset	2	2	1, 2	1, 2

Redundant expressions: $v*v$ in C and $arr[i]$ in D Optimized code:

|
v
H: return change;

3. Build a global interference graph for the purpose of applying graph coloring based register allocation for the optimized code after 2. How many physical registers do you need to fully allocate all the scalar variables to registers?

Variables: i, change, v, size, arr, i, elem1, elem2, res, tmp.

	Lout	Lin
A	change, v, arr, i, res, size, tmp	v, arr, size
B	change, v, arr, i, res, size, tmp	change, v, arr, i, res, size, tmp
C	change, v, arr, i, res, size, tmp, elem2, elem1	change, v, arr, i, res, size, tmp
D	change, v, arr, i, res, size, tmp, elem2	v, arr, i, size, tmp, elem1
E	change, v, arr, i, res, size, tmp	change, v, arr, i, res, size, tmp, elem2
F	change	change
G	\emptyset	change
H	\emptyset	change

Interference graph includes edges between the following pairs of nodes: (change, v), (change, arr) (change, i), (change, res), (change, size), (change, tmp), (v, arr), (v, i), (v, res), (v, size), (v, tmp), (arr, i), (arr, res), (arr, size), (arr, tmp), (i, res), (i, size), (i, tmp), (res, size), (res, tmp), (size, tmp), (elem1, v), (elem1, arr), (elem1, i), (elem1, size), (elem1, tmp) (elem2,change), (elem2, v), (elem2, arr), (elem2, i), (elem2, res), (elem2,size), (elem2, tmp)

8 physical registers are needed to hold all the values.