Homework 6: Machine Independent Optimization

Due Nov 23, 2011

Suppose we have the following C code.

```
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)  { res = v; return change;}
       res = 0;
       for (i = 0; i < size; ++i)
          res = res + cur[i];
       return change;
}
```

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

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

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?