

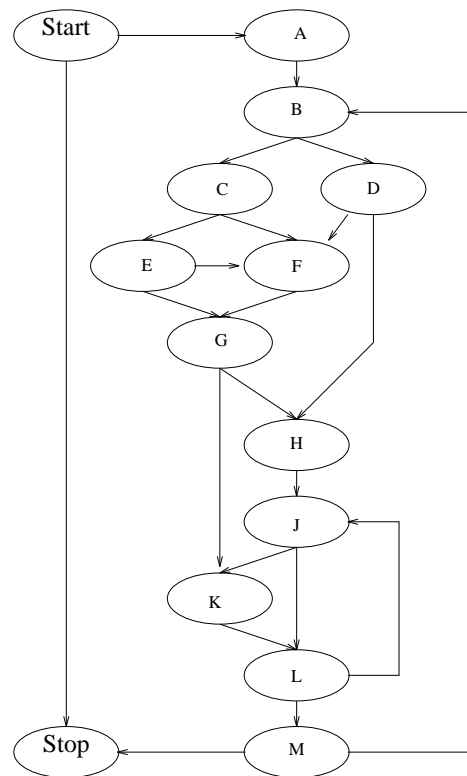
Final Exam

Given: 8 December 2011

Due: 20 December 2011

This exam is a take-home, open-notes exam. You may not discuss any problems from this exam with anybody except the instructor. The exam is due in the CS office (509 Bryan Hall) by closing at 5 PM.

Questions should be referred to the instructor, preferably by e-mail (cytron@cs.wustl.edu).



- (15 points) In the above flow graph, suppose nodes A and E have nonidentity transference. Show the corresponding sparse evaluation graph, assuming no nodes have constant transference.
- (10 points) In the above flow graph, suppose nodes Start, E, H, and K contain definitions of the variable x , while all other nodes contain a use of x . Show the resulting Static Single Assignment form program.

3. (15 points)

(a) Construct an interference graph for the program shown below.

Procedure *DFS*(*X*)

num \leftarrow *num* + 1

dfn(*X*) \leftarrow *num*

vertex(*num*) \leftarrow *X*

foreach (*Y* \in *Succ*(*X*)) **do**

if (*dfn*(*Y*) = 0) **then**

parent(*Y*) \leftarrow *X*

sibling(*Y*) \leftarrow *child*(*X*) ; *child*(*X*) \leftarrow *Y*

call *DFS*(*Y*)

fi

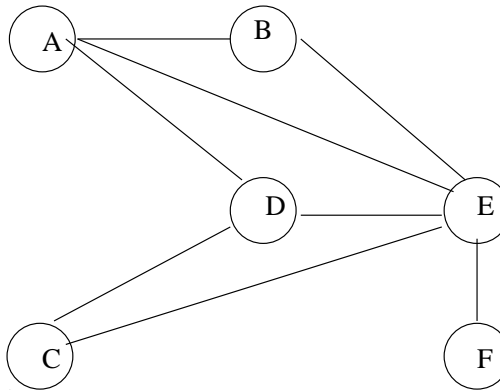
od

progeny(*X*) \leftarrow *num* - *dfn*(*X*)

end

\Leftarrow I

(b) Show how the Chaitin/Chandra coloring heuristic attempts to color the following interference graph using 3 colors:



4. (20 points) Consider the following program:

```

do i=1 to 100
  a(i) =
    = a(100-i+1)
enddo
  
```

The dependence equation, with x and y representing values of i at the store and fetch, respectively, is:

$$x + y = 101$$

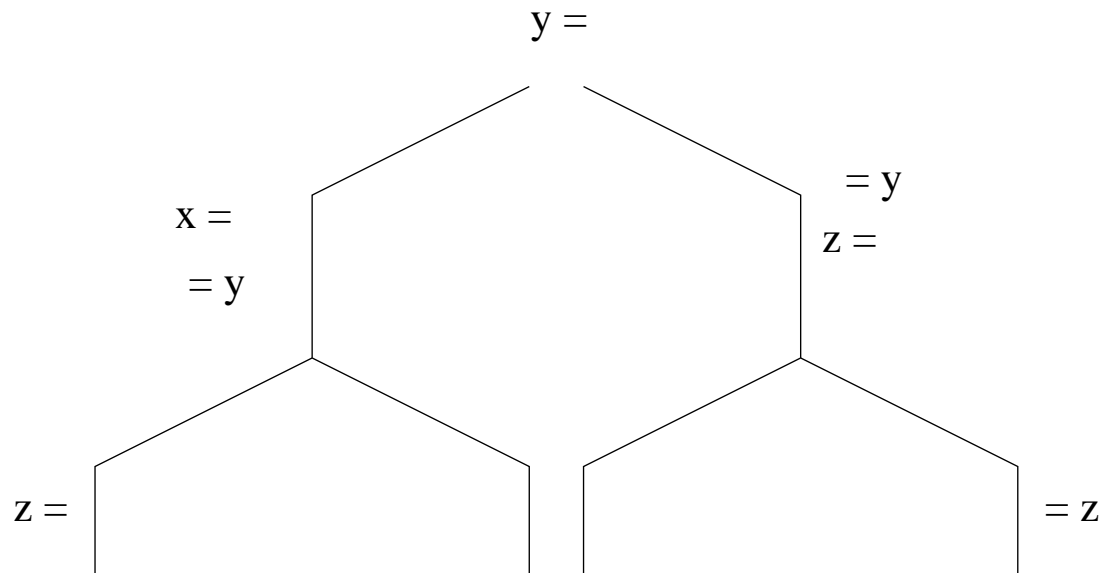
(a) Show the application of Banerjee-Wolfe inexact test to the equation under the (\star) direction vector:

MIN (x+y)	constant term	MAX (x+y)
	101	

- (b) Failing to show independence, now refine the Banerjee-Wolfe test for each of ($<$), ($=$), and ($>$):

	MIN ($x+y$)	constant term	MAX ($x+y$)
$(x < y)$		101	
$(x = y)$		101	
$(x > y)$		101	

- (c) Although Banerjee-Wolfe shows dependence for ($x=y$), another decision algorithm can show independence for ($x=y$). Which one? Show its application just for the ($=$) direction vector.
- (d) Can the statements corresponding to the store and fetch of a be interchanged in this example? Why or why not?
5. (15 points) Consider the following parallel program (branches indicate parallelism, not conditional choice):



- (a) Show, using the Shasha-Snir algorithm, those references within each process whose order must be maintained in any program transformation.
- (b) Show how this program will execute if access anomalies are detected.

6. (5 points) Consider the following statement and discuss whether it is true or false. Support your position with an example if it is false, or a proof sketch if it is true:

Consider a program P and its Sasha-Snir analysis that reveals which program-order edges suffice to provide correct execution. Now suppose the statements of P are reordered while respecting the appropriate program-order edges. Call the resulting program Q .

Claim: The program P can always be recreated from any such Q by computing Sasha-Snir analysis on Q and then reordering statements in Q while respecting the program-order analysis on Q .

7. (20 points) Consider the program shown below:

```
for (int i=1; i < 50; ++i) {
  for (int j=1; j < 50; ++j) {
    a[i-1,j+1] = b[i,j]
    b[i-1,j-1] = a[i,j]
    t           = a[i,j]
    c[i,j]      = t
  }
}
print t
```

- Show the dependence graph for this loop, labeling the edges with direction vectors.
- What can be done to eliminate the output and anti-dependences that involve t ? Show the resulting program and consider that program for the rest of this question.
- (Reminder, you are now using the version of the program you obtained by removing output and anti-dependences on t .) Is loop interchange possible? Why or why not?
- If the inner loop is reversed, is loop interchange possible? Why or why not?
- Is there a sequence of loop reversals and interchanges that will work? For each such possibility show the resulting loop structure and explain why dependences are honored.