

CS 375 Final Problem Set

Due date: April 29th, 2010

1. Construct an NFA and a minimal state DFA for the following regular expressions.

- i) $(\epsilon + a)b^*$
- ii) $(a^* + b^*)^*$
- iii) $(a+b)^*a(a+b)$

2. Find nullable, FIRST and FOLLOW sets for the following grammar, and then compute the LL(1) parsing table. Is this grammar LL(1)? Modify the grammar as little as possible to produce an LL(1) grammar that generates the same language.

$S \rightarrow u B D z$
 $B \rightarrow B v$
 $B \rightarrow w$
 $D \rightarrow E F$
 $E \rightarrow y$
 $E \rightarrow \epsilon$
 $F \rightarrow x$
 $F \rightarrow \epsilon$

3. Consider the grammar

$S \rightarrow AS \mid b$
 $A \rightarrow SA \mid a$

- i) Construct the LR(0) automaton for this grammar.
- ii) Construct the SLR(1) parsing table for this grammar. Is the grammar SLR(1)?
- iii) Construct the LALR(1) parsing table for this grammar. Is the grammar LALR(1)?

4. Consider the following grammar

$E \rightarrow E + T \mid T$
 $T \rightarrow T F \mid F$
 $F \rightarrow F^* \mid a \mid b$

- i) Construct the SLR(1) parsing table for this grammar. Is the grammar SLR(1)?
- ii) Construct the LALR(1) parsing table for this grammar. Is the grammar LALR(1)?

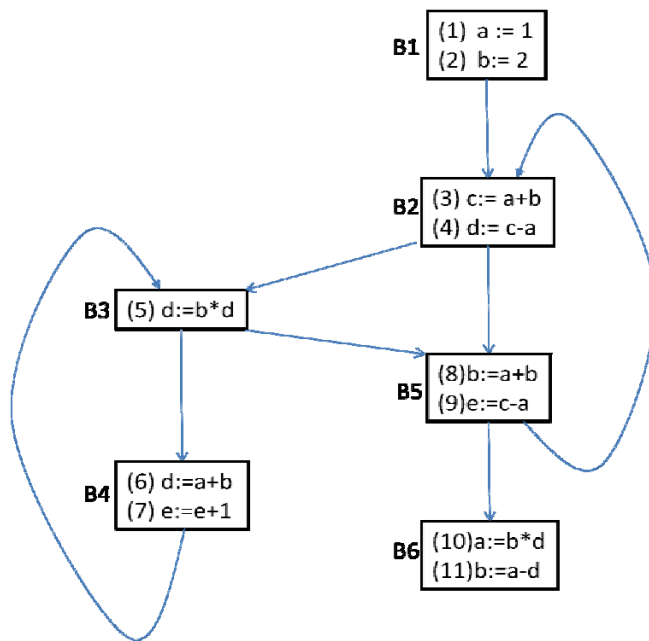
5. Consider a partially ordered set D and a function $f: D \rightarrow D$.

- (a) If f is extensive, is it necessarily monotonic? If yes, give a proof. If not, show a poset D and a function f that is extensive but not monotonic.
- (b) Prove that the composition of two monotonic functions is itself a monotonic function.

(c) Consider the set D of non-negative integers ordered by the standard ordering on integers. This is a partially ordered set with a least element. Consider the function $f:D \rightarrow D$ defined as $f(x) = x+1$. Show that this function is monotonic. Does it have a fixpoint? If so, compute it. If not, explain why it has no fixpoint even though it is a monotonic function.

6. For the control-flow graph shown below, compute

- (a) use-def and def-use chains
- (b) live variables at the end of each block
- (c) available expressions at the beginning of each block.



7. (a) A variable v is said to be possibly uninitialized if there is a path from START to a use of v on which there is no definition of v . Formulate the problem of finding possibly uninitialized variables in a program as a dataflow analysis problem by specifying (i) whether it is a forward-flow or backward flow problem, (ii) the transfer function for an assignment statement, and (iii) the confluence operator.

(b) A variable v is said to be definitely uninitialized if there is a use of v such that there is no definition of v on any path from START to that use. Formulate the problem of finding definitely uninitialized variables in a program as a dataflow analysis problem by specifying (i) whether it is a forward-flow or backward flow problem, (ii) the transfer function for an assignment statement, and (iii) the confluence operator.

8. Write a SaM program that reads a value n from the input and computes the value of $n!$ (n factorial).

9. Consider the following program. Perform live variable analysis, draw the interference graph for the variables and find the minimum number of colors required to color this graph.

