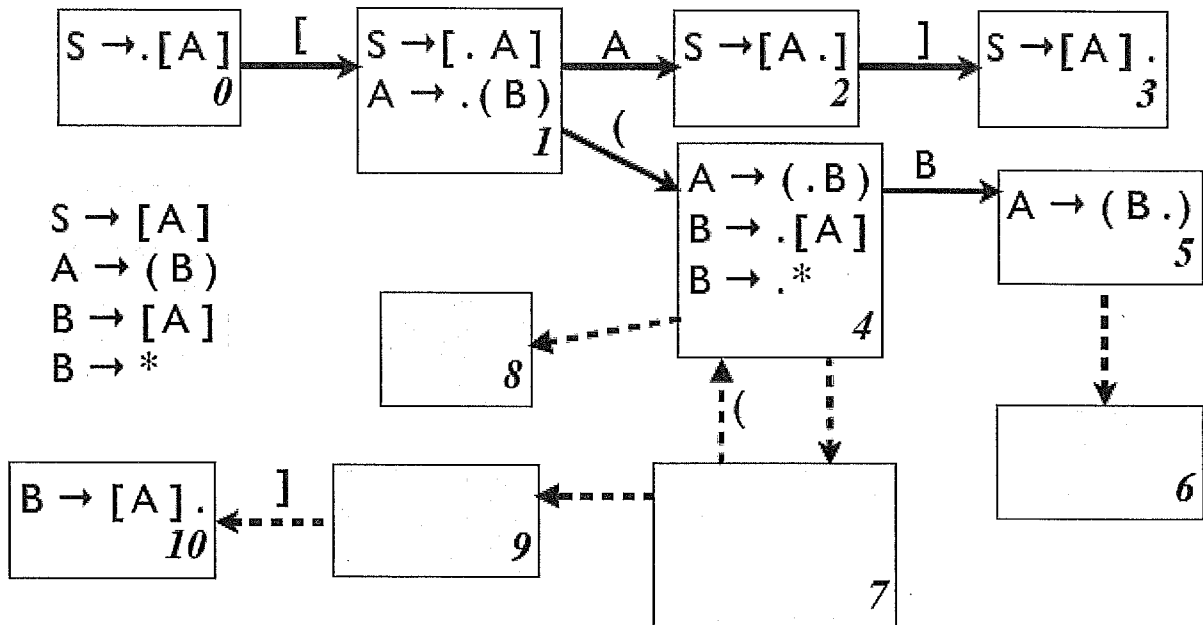


**CE-2**  
**August 2014 QE**

**Question 1 (20 points total):** Given the grammar G:

- $S \rightarrow [A]$
- $A \rightarrow (B)$
- $B \rightarrow [A]$
- $B \rightarrow *$

- (a) (4 points) Complete the LR(0) CFSM in your blue book. You only need to show symbol labels on the dashed edges and the contents of the shaded (gray) states (6, 7, 8 and 9), i.e., you do not need to redraw the entire CFSM.
- (b) (4 points) List the terminals and non-terminals of the grammar G.
- (c) (4 points) Given your table, the state stack 0 1 4 7 4 5 6, the symbol stack [ ( ( B ), and a next symbol of ], what action will be taken and what is the resulting state of the symbol and state stacks?
- (d) (4 points) Given your table, the state stack 0 1 4 7 4, the symbol stack [ ( ( and a next symbol of \* what action will be taken and what is the resulting state of the symbol and state stacks?
- (e) (4 points) Let S be a string containing k [ symbols that is accepted by the grammar. How



many ) symbols are in the string?

Write in Exam Book Only

Question 2 (25 points total).

The Intel 8085 processor has no cache and no registers. It does have 256 memory locations that can be accessed faster than all other memory locations. All other memory locations take the same amount of time to read or write, regardless of the order they are read or written. For the optimizations below say whether or not they will be useful on the 8085 and why. Your answers should be 25 words or less.

*Loop interchange*

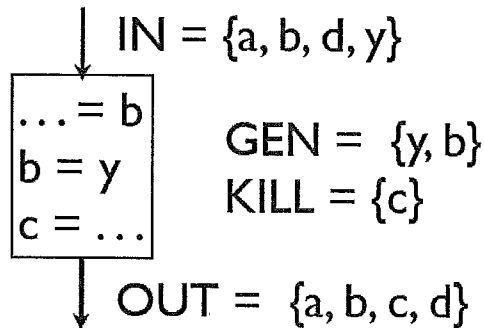
*Tiling*

*Dead code elimination*

*Constant propagation*

*Register allocation*

Question 3 (15 points total). A basic block from a control flow graph is shown below. The GEN and KILL sets and the IN and OUT sets for the basic block are shown immediately after a standard dataflow analysis visits the basic block.



(a) (10 points) Is the analysis a forward or backward analysis and give your reason why.

(b) (5 points) What common dataflow analysis could this be?

Write in Exam Book Only

Question 4 (20 points total). Consider the loop below.

```

for (i = 0; i < n; i++) {
  for (j = 0; j < n; j++) {
    a[i][j] = b[i][j] + c[i][j];
    d[i][j] = a[i][j-1] + a[i][j+1]
  }
}

```

- (a) (15 points) Two dependences exist in the loop. Fill in a table like the one below in your Blue Book with information about the dependences.
- (b) (5 points) Which of the *i* and *j* loop can be parallelized?

Source reference	Sink reference	Type of dependence	Dependence direction	Dependence distance

Question 5 (20 points). Given the loop

```

z = 0; // S1
// program point A
for (i = 0; i < n; i++) {
  a = x/y; // S2
  b = x/y; // S3
  x = x + z; // S4
}

```

- (a) (5 points) Without knowing the value of “z” in S4 can a compiler move statement S2 to program point A?
- (b) (5 points) If a compiler knows the value of “z” in S4 can it move statement S2 to program point A?
- (c) (5 points) Without knowing the value of “z” in S4 can a compiler change statement S3 from “b=x/y;” to “b=a;”?
- (d) (5 points) If a compiler knows the value of “z” in S4 can it change statement S3 from “b=x/y;” to “b=a;”?