

Solutions to 1996 CS Comprehensive Exam: Compilers

Question 1 (9 points)

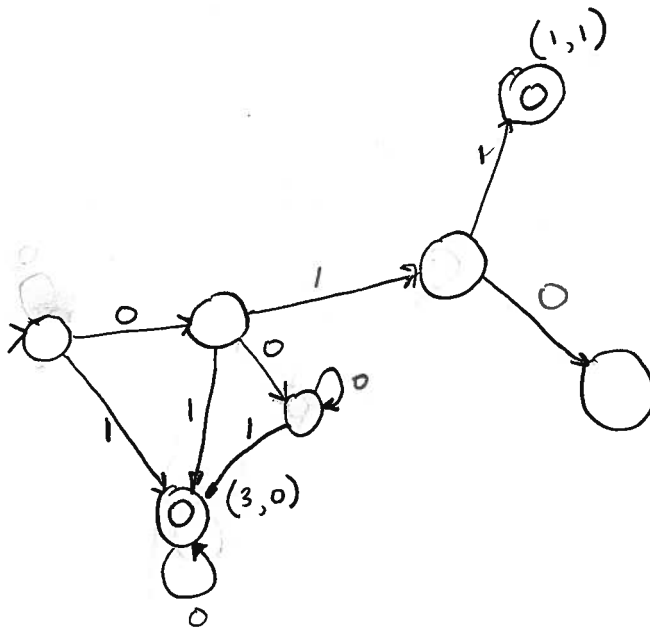
Consider the following block of 3-address code:

- 1) $a := b + c$
- 2) $d := b$
- 3) $e := c + d$
- 4) $f := a - e$
- 5) $d := a$

Assume that the value of each of a through f is needed following the block. Give six transformations that may be performed on this block to improve the running time on a typical machine. Indicate the change made at each of your six steps (it is sufficient to give the rewritten line only, rather than the whole block) and *give the name of the transformation or brief justification*. You may apply more than one transformation to a single statement, but use separate steps to do so.

Solution

1. Replace line (3) by $e := c+b$ [copy propagation, using line (2)].
2. Replace line (3) by $e := b+c$ [use of commutative law].
3. Replace line (3) by $e := a$ [common subexpression elimination, using line (1)].
4. Replace line (4) by $f := a-a$ [copy propagation, using line (3)].
5. Replace line (4) by $f := 0$ [algebraic simplification].
6. Eliminate line (2) [dead code elimination].



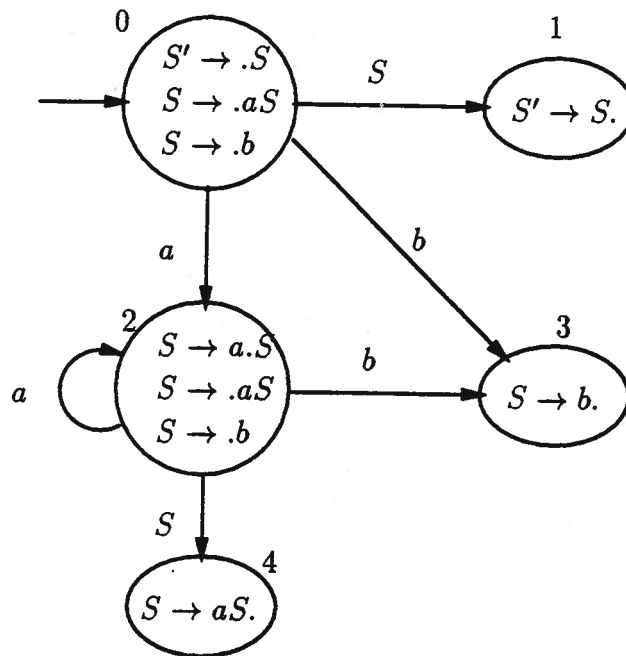
Question 2 (11 points)

For the grammar

- 1) $S \rightarrow aS$
- 2) $S \rightarrow b$

- a) [8 points] Construct the LR(0) sets of items (sometimes called SLR items; they involve no lookahead) and show their transition diagram.
- b) [3 points] Give the LR(0) parsing table and indicate any conflicts that arise.

Solution (a)



Solution (b)

State	Action			Goto
	a	b	\$	
0	shift 2	shift 3		1
1			accept	
2	shift 2	shift 3		4
3			reduce $S \rightarrow b$	
4			reduce $S \rightarrow aS$	

Question 3 (10 points)

Suppose we wish to build a lexical analyzer that recognizes the following three tokens:

- 1) **01**
- 2) **010**
- 3) **0*10***

Assuming the LEX priority rules (longest token recognized, and in case of ties, pick the token listed first), draw a deterministic finite automaton that recognizes the above three tokens. Indicate, for each state, which, if any, token is recognized. Also, since a token may not be recognized until further symbols are read by the automaton, indicate how far back the recognized token ends. For example, attaching (2,3) to an accepting state means that the second token, **010**, was recognized, and the last 3 symbols read by the automaton are not part of the token.

Solution

