

CS 335 Semester 2023–2024-II: Assignment 3

22th March 2024

Due Your assignment is due by Apr 3 2024 11:59 PM IST.

General Policies

- You should do this assignment ALONE.
- Do not plagiarize or turn in solutions from other sources. You will be PENALIZED if caught.

Submission

- Submission will be through Canvas.
- Upload a PDF file named “`<roll>-assign3.pdf`” that contains the solutions for the pen-paper problems.
- We encourage you to use the L^AT_EX typesetting system for generating the PDF file. You can use tools like Tikz, Inkscape, or Draw.io for drawing the automata. You can include a scanned copy of a hand-drawn figure, but MAKE SURE the figure is legible.
- You will get up to TWO LATE days to submit your assignment, with a 25% penalty for each day.

Evaluation

- Write your solutions such that the EXACT output format (if any) is respected (e.g., SDT output).
- We WILL deduct marks if you disregard the listed rules.

Problem 1

[40 marks]

Consider the following SDD, where V, W, X, Y , and Z are non-terminals, where V is the starting non-terminal. The attribute *val* in the semantic rules represents a numeric value.

$V \rightarrow Y \# W$	$\{V.val = W.val \% Y.val\}$ where $\%$ is modulus operator
$W \rightarrow X @ Y$	$\{W.val = X.val + Y.val\}$
$X \rightarrow X_1 Z$	$\{X.val = X_1.val + 3 \times Z.val\}$
$X \rightarrow Z$	$\{X.val = Z.val\}$
$Y \rightarrow Z Y_1$	$\{Y.val = 2 \times (Z.val + Y_1.val)\}$
$Y \rightarrow Z$	$\{Y.val = 3 \times Z.val\}$
$Z \rightarrow 3$	$\{Z.val = 3\}$
$Z \rightarrow 4$	$\{Z.val = 4\}$

- Show the annotated parse tree for the input string $43\#43@443$.
- What is the value at V computed by the translation scheme for the above input string.
- Explain whether the grammar is S-attributed or L-attributed.

Question 2

[50 marks]

We have discussed generating 3AC for array accesses using semantic translations. Consider the following extended grammar with semantic translation.

$S \rightarrow \mathbf{id} = E$	$\{gen(symtop.get(\mathbf{id.lexeme}) " = " E.addr)\}$
$S \rightarrow L = E$	$\{gen(L.array.base "[" L.addr "]" " = " E.addr)\}$
$E \rightarrow E_1 - E_2$	$\{E.addr = \mathbf{new Temp}(); gen(E.addr " = " E_1.addr " - " E_2.addr)\}$
$E \rightarrow E_1 / E_2$	$\{E.addr = \mathbf{new Temp}(); gen(E.addr " = " E_1.addr " / " E_2.addr)\}$
$E \rightarrow \mathbf{id}$	$\{E.addr = symtop.get(\mathbf{id.lexeme})\}$
$E \rightarrow L$	$\{E.addr = \mathbf{new Temp}(); gen(E.addr " = " L.array.base "[" L.addr "]")\}$
$E \rightarrow *E_1$	$\{E.addr = \mathbf{new Temp}(); gen(E.addr " = " " * " E_1.addr)\}$
$L \rightarrow \mathbf{id}[E]$	$\{L.array = symtop.get(\mathbf{id.lexeme}); L.type = L.array.type.elem;$ $L.addr = \mathbf{new Temp}(); gen(L.addr " = " E.addr " * " L.type.width)\}$
$L \rightarrow L_1[E]$	$\{L.array = L_1.array; L.type = L_1.type.elem;$ $t = \mathbf{new Temp}(); L.addr = \mathbf{new Temp}();$ $gen(t " = " E.addr " * " L.type.width); gen(L.addr " = " L_1.addr " + " t); \}$

Assume the size of integers to be four bytes, and that the arrays are zero-indexed. Let A, B, and C be integer arrays of dimensions 11×8 , 12×6 , and $10 \times 10 \times 6$, respectively. Construct an annotated parse tree for the expression $C[i][j][k] - A[i][k]/B[i][j]$ and show the 3AC code sequence generated for the expression.

Question 3

[60 marks]

Construct an SDT translation scheme for array expressions using column-major organization of arrays. Use the following grammar.

$$\begin{aligned}S &\rightarrow \mathbf{id} = E \\S &\rightarrow L = E \\E &\rightarrow E_1 + E_2 \\E &\rightarrow L \\L &\rightarrow \mathbf{id} \\L &\rightarrow \mathbf{id}[Elist \\Elist &\rightarrow E] \\Elist &\rightarrow E, Elist_1\end{aligned}$$

- (i) Show the semantic actions for your proposed translation.
- (ii) Explain the attributes and auxiliary functions in your SDT.
- (iii) Show the annotated parse tree for the expression $x = c + A[i][j]$. Assume that A is a 10×20 array of integers, the size of an integer is 4 bytes, and the arrays are zero-indexed.
- (iv) Show the generated 3AC for the above expression.