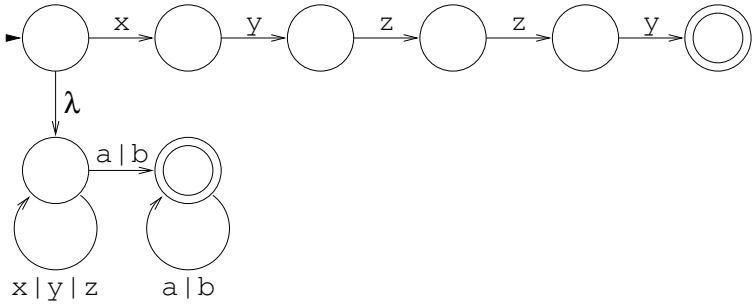
CS 352 – Compilers: Principles and Practice – Homework 1
Due: in class Wednesday, October 26, 2011

1. Consider the following regular expression:

$$((x \mid y \mid z)^* \ (a \mid b)^+) \mid xyzzy$$

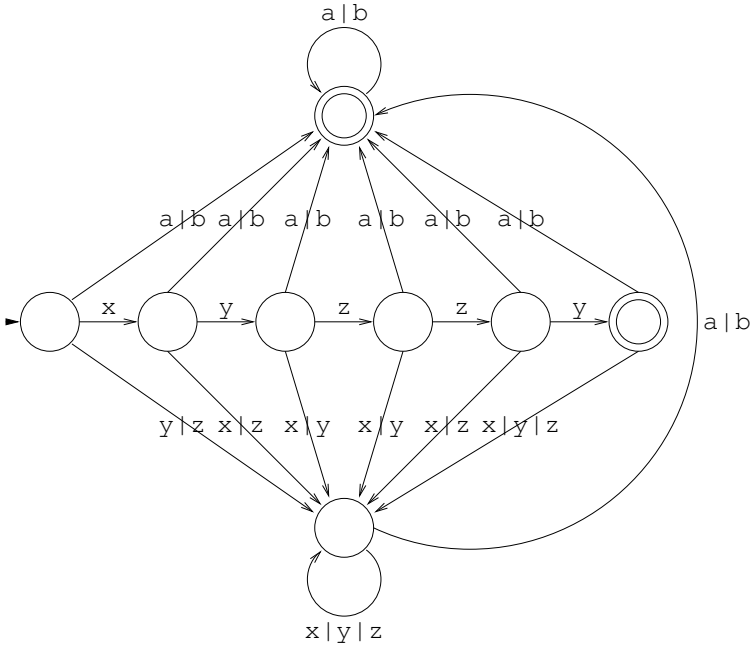
(a) Give an NFA corresponding to this regular expression.

Answer:



(b) Devise (without going through the full subset construction!) a DFA for this regular expression.

Answer:



2. Consider the following regular expression and the regular language it denotes:

$$a^*bc \mid ab^*(b \mid c)$$

(a) As described in class, construct an NFA that recognizes this regular language.

(b) Using the subset construction, convert the NFA to a DFA. Optimize the resulting DFA by merging equivalent states (if any).

3. Consider the following simple grammar:

$$\begin{array}{llllll} S \to L & L \to Ra & R \to aba & Q \to bbc \\ & L \to Qba & R \to caba & Q \to bc \\ & & R \to Rbc \end{array}$$

(a) This grammar is not suitable for a top-down predictive (i.e., LL) parser. What problems does it have?

Answer:

The grammar is left-recursive and some productions have common prefixes.

(b) Fix the problems by rewriting the grammar. **Answer:**

$$\begin{array}{llllll} S \to L & L \to Ra & R \to abaR' & Q \to bQ' \\ & L \to Qba & R \to cabaR' & Q' \to bc \\ & & R' \to bcR' & Q' \to c \\ & & R' \to \varepsilon \end{array}$$

(c) Construct the LL(1) parsing table for your new grammar.

Answer:

	FIRST	FOLLOW	a	b	c	\$
S	abc	\$	$S \to L$	$S \to L$	$S \to L$	
L	abc	\$	$L \to Ra$	$L \to Qba$	$L \to Ra$	
R	ac	a	$R \to abaR'$		$R \to cabaR'$	
R'	$b\varepsilon$	a	$R' \to \varepsilon$	$R' \to bcR'$		
Q	b	b		$Q \to bQ'$		
Q'	bc	b		$Q' \to bc$	$Q' \to c$	

(d) Consider the following input string: *cababca*. Show the steps of an LL parser as it uses your parsing table to predict the expansion of each non-terminal for this input, showing the input as it is consumed, and the parse stack at each step of the parse.

Answer:

Stack (top is left)	Input
$S\$$	*cababca*\$
$L\$$	*cababca*\$
$Ra\$$	*cababca*\$
$cabaR'a\$$	*cababca*\$
$R'a\$$	*bca*\$
$bcR'a\$$	*bca*\$
$R'a\$$	$a\$$
$a\$$	$a\$$
\$	\$

4. Consider the following simple (augmented) grammar:

$$
\begin{aligned}
S &\rightarrow L\$ \\
L &\rightarrow E;L \\
L &\rightarrow E \\
E &\rightarrow \texttt{nil}
\end{aligned}
$$

(a) Show the steps of a bottom-up parser as it parses the input `nil;nil;nil`, showing the input as it is consumed, the parse stack at each step of the parse, and the action applied at each step.

Answer:

Stack (top is right)	Input	Action
\$	`nil;nil;nil`\$	shift
\$`nil`	`;nil;nil`\$	reduce
\$$E$	`;nil;nil`\$	shift
\$$E$;	`nil;nil`\$	shift
\$$E$;`nil`	`;nil`\$	reduce
\$$E$;$E$	`;nil`\$	shift
\$$E$;$E$;	`nil`\$	shift
\$$E$;$E$;`nil`	\$	reduce
\$$E$;$E$;$E$	\$	reduce
\$$E$;$E$;$L$	\$	reduce
\$$E$;$L$	\$	reduce
\$$L$	\$	accept

(b) Construct the LR(0) item sets for this grammar:

Answer:

$$
\begin{aligned}
L_0: \quad & S \rightarrow \bullet L\$ \\
& L \rightarrow \bullet E;L \\
& L \rightarrow \bullet E \\
& E \rightarrow \bullet \texttt{nil} \\
L_1: \quad & S \rightarrow L \bullet \$ \\
L_2: \quad & S \rightarrow L\$ \bullet \\
L_3: \quad & L \rightarrow E \bullet ;L \\
& L \rightarrow E \bullet \\
L_4: \quad & E \rightarrow \texttt{nil} \bullet \\
L_5: \quad & L \rightarrow E; \bullet L \\
& L \rightarrow \bullet E;L \\
& L \rightarrow \bullet E \\
& E \rightarrow \bullet \texttt{nil} \\
L_6: \quad & L \rightarrow E;L \bullet
\end{aligned}
$$

(c) Is this grammar LR(0)? Why or why not?

Answer:

No, there is a shift-reduce conflict in item set L_3.

(d) Construct the SLR(1) parse table for this grammar.

Answer:

	nil	;	$	S	L	E
0	$s4$				1	3
1			a			
2						
3		$s5$	$r_{L\to E}$			
4		$r_{E\to\texttt{nil}}$	$r_{E\to\texttt{nil}}$			
5	$s4$				6	3
6		$r_{L\to E;L}$	$r_{L\to E;L}$			

(e) Is this grammr SLR(1)? Why or why not?

Answer:

Yes, there are no conflicts in the SLR(1) parse table.