COMS W4115: Programming Languages and Translators

Homework Assignment 3

Due Wednesday, April 14 at **5:40 PM EDT** via Courseworks

Place your answers in the boxes below and upload your solutions as a PDF file on Courseworks. You may modify this PDF; edit it in Inkscape; print, write on, and scan; or something else. Don't add additional pages.

Do this assignment alone. You may consult the instructor and the TAs, but not other students.

Name: [   ]   Uni: [   ]

1. For the grammar $\quad _0\ A' \to A \quad$ with terminals x, y, and z and non-terminals A', A, and B.
   $_1\ A \to B\, x\, A$
   $_2\ A \to y$
   $_3\ B \to$
   $_4\ B \to z$

   (a) (15 pts.) Complete its LR(0) automaton. Fill in the **items**, indicate the **accepting states**, and label the unmarked **transitions**. No additional transitions are necessary.

   S6 [   ]   S2 [   ]

   y

   S0 [   ]   S1 [   ]   S4 [   ]

   S3 [   ]   S5 [   ]
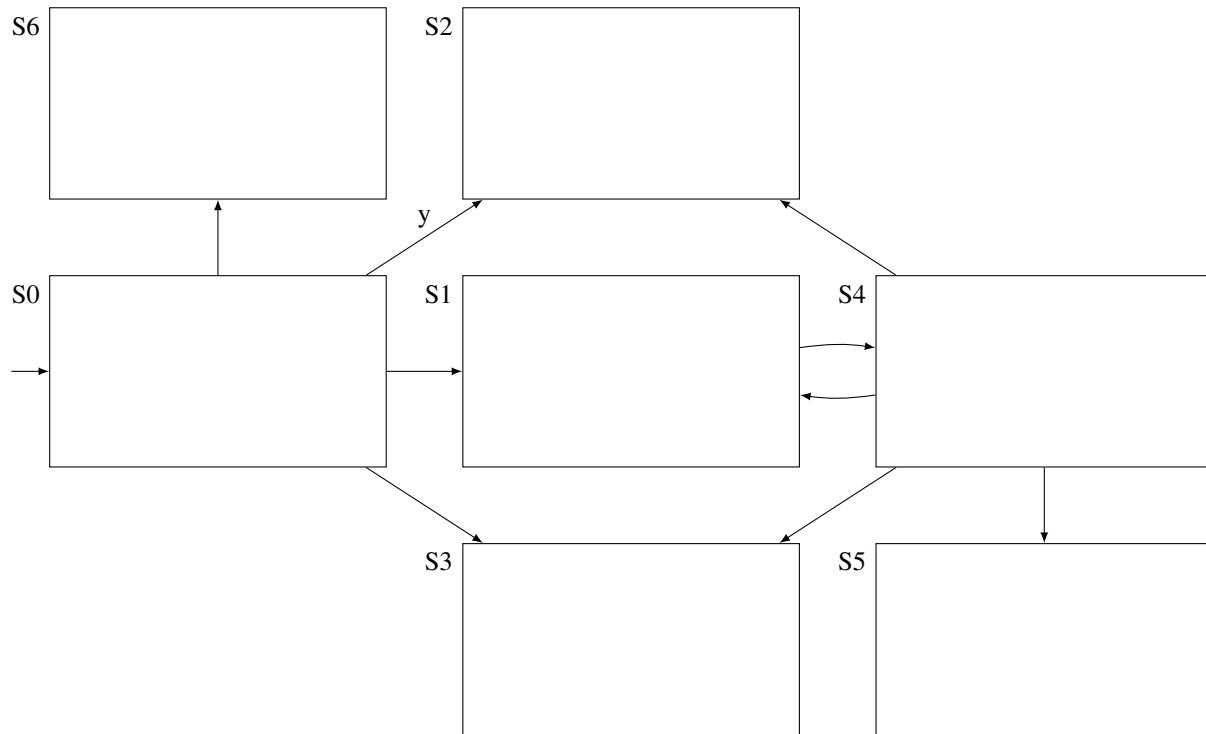
   (b) (10 pts.) **Compute** the FIRST and FOLLOW sets. **Explain** your reasoning.

   $\text{FIRST}(A') = \{ \qquad \}$

   $\text{FIRST}(A) = \{ \qquad \}$

   $\text{FIRST}(B) = \{ \qquad \}$

   $\text{FOLLOW}(A') = \{ \qquad \}$

   $\text{FOLLOW}(A) = \{ \qquad \}$

   $\text{FOLLOW}(B) = \{ \qquad \}$

(c) (15 pts.) **Fill in** the SLR parsing table:

State	Action				Goto	
	x	y	z	$	A	B
S0						
S1						
S2						
S3						
S4						
S5						
S6						

(d) (15 pts.) Use your table to **show the steps** a shift/reduce parser would take parsing "*x z x y*". When describing the contents of the stack, write alternating state numbers and terminals/nonterminals, e.g., "0 B 1 x 4". We've done the first two steps for you.

Stack	Input	Action
0	*x z x y* $	reduce 3
0 B 1	*x z x y* $	shift

2. (20 pts.) For the following C program, calling **foo(1, 2)** eventually causes **baz()** to be called. **Show the layout**, including values, of the stack **just before baz() is called.** Show function arguments being passed on the stack, local variables, previous frame pointers, return addresses, and the current stack and frame pointers. You do not have to include space for registers.

```c
void foo(int a, int b)
{
  int x = a + b;
  int y;
  bar(a, x);
}

void bar(int c, int d)
{
  int y = c + 3;
  int z = d * 2;
  baz(y + z);
}

void baz(int q)
{ ... }
```

3. For the lambda expression "$(\lambda x . \lambda y . + y\ 10)\ ((\lambda z . + z\ z)\ 3)\ 21$"

   (a) (10 pts.) Show how it reduces to normal form using **normal order evaluation**

   (b) (10 pts.) Show how it reduces to normal form using **applicative order evaluation**