

1.

```

N1 = empty | e Sign Digits
N2 = empty | . Digits
Numeral = Digits N1 N2
Digits ::= Digit Digits | Digit
Digit  ::= 0 | 1 | 2 | 3

```

2.

```

a. A ::= a Q
   Q ::= b Q | empty

```

```

first(Q) = {b, empty}
first(A) = {a}
follow(A) = {$}
follow(Q) = {$}

```

```

b. S ::= ABC
   A ::= a | Cb | empty
   B ::= c | dA | empty
   C ::= e | f

```

```

first(C) = {e, f}
first(B) = {c, d, empty}
first(A) = {a, e, f, empty}
first(S) = {a, c, d, e, f}

```

```

follow(S) = {$}
follow(C) = {$, b}
follow(B) = {e, f}
follow(A) = {c, d, e, f}

```

```

c. Exp ::= - Exp | (Exp) | Var ExpTail
   ExpTail ::= -Exp | empty
   Var ::= id VarTail
   VarTail ::= (Exp) | empty

```

```

first(VarTail) = { (, empty }
first(Var) = {id}
first(ExpTail) = {-, empty}
first(Exp) = {-, (, id}

```

```

follow(Exp) = {$, ) }
follow(ExpTail) = follow(Exp)
follow(Var) = {-, $, ) }
follow(VarTail) = follow(Var)

```

3. Table

	-		()		id		\$	
Expr	- Expr		(Expr)				Var ExprTail			
ExprTail	- Expr				empty				empty	
Var							id VarTail			
VarTail	empty		(Expr)		empty				empty	

4.a. The grammar is not LR(1). There is a shift-reduce conflict on lookahead “+” in one of the states:

```
[P -> id := E., +]
[E -> E.+ P, , +]
```

4.b. The grammar is SLR(1).

4.c. The grammar is not SLR(1). There is a shift-reduce conflict on lookahead “+” in the following state:

```
[P -> id := E.]
[E -> E.+ P, ]
```

d. The grammar is SLR(1).