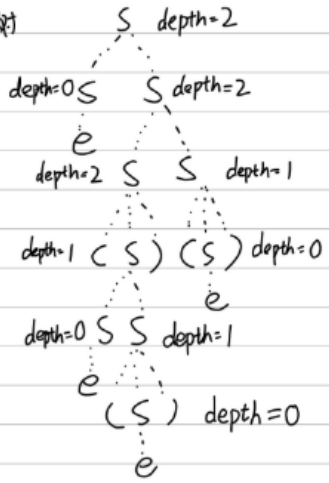
习题9.1(1)　S→e　　S.depth=0

S→(S)　　S[0].depth = S[1].depth+1

S→SS　　S[0].depth = max(S[0].depth, S[1].depth)

(2) e(e(e))(e)→S(S(S))(S)→S(SS)S→SSS→S

语法树



语法制导的属性求值过程

符号栈	d栈	剩余串	动作
#	#	e(e(e))(e)#	移进
#e	#-	(e(e))(e)#	return, S.depth=0
#S	#0	(e(e))(e)#	移进
#S(	#0-	e(e))(e)#	移进
#S(e	#0--	(e))(e)#	return, S.depth=0
#S(S	#0-0	(e))(e)#	移进
#S(S(	#0-0-	e))(e)#	移进
#S(S(e	#0-0--	))(e)#	return, S.depth=0
#S(S(S	#0-0-0	))(e)#	移进
#S(S(S)	#0-0-0-	)(e)#	return, S[0].depth=S[1].depth+1
#S(S	#0-01	)(e)#	return, S[0].depth=max(S[1].depth, S[0].depth)
#S(S	#0-1	)(e)#	移进
#S(S)	#0-1-	(e)#	return, S[0].depth=S[1].depth+1
#SS	#02	(e)#	return, S[0].depth=max(S[1].depth, S[2].depth)
#S	#2	(e)#	移进
#S(	#2-	e)#	移进
#S(e	#2--	)#	return, S.depth=0
#S(S	#2-0	)#	移进
#S(S)	#2-0-	#	return, S[0].depth=S[1].depth+1
#SS	#21	#	return, S[0].depth=max(S[1].depth, S[2].depth)
#S	#2	#	# acc

3) 句子的括号嵌套深度等于其语法树中最深的括号嵌套层数

句子的嵌套深度由最内层的括号决定

(1) @table (
outer: NULL
width: 84 argc: 0 arglist: NIL rtype: void level: 0 code: [...]
entry: (name: x type INT offset: 4)
entry: (name: b type ARRAY offset: 76 dims: 2 dim[0]: 3 dim[1]: 6 etype: INT)
entry: (name: foo type FUNC offset: 84 mytab: foo@table)
)

foo @table (
outer: @table
width: 8 argc: 1 arglist (x) rtype: INT level: 1 code: [...]
entry (name: x type: INT offset: 4)
entry (name: y type: INT offset: 8)
)

(2) @table (
outer: NULL
width: 8 argc: 0 arglist: NIL rtype: void level: 0 code: [...]
entry: (name: h type: FUNC offset: 8 mytab: h@table)
)

h@table (
outer: @table
width: 20 argc: 2 arglist (f y) rtype: INT level: 1 code: [...]
entry: (name: f type: FUNPTT offset: 8 rtype: INT)
entry: (name: y type: INT offset: 12)
entry: (name: g type: FUNC offset: 20 mytab: g@table)
)

g @table (
outer: h@table
width: 4 argc: 1 arglist (c) rtype: INT level: 2 code: (...)
entry (name: c type: ARRPTT offset: 4 etype: INT)
)