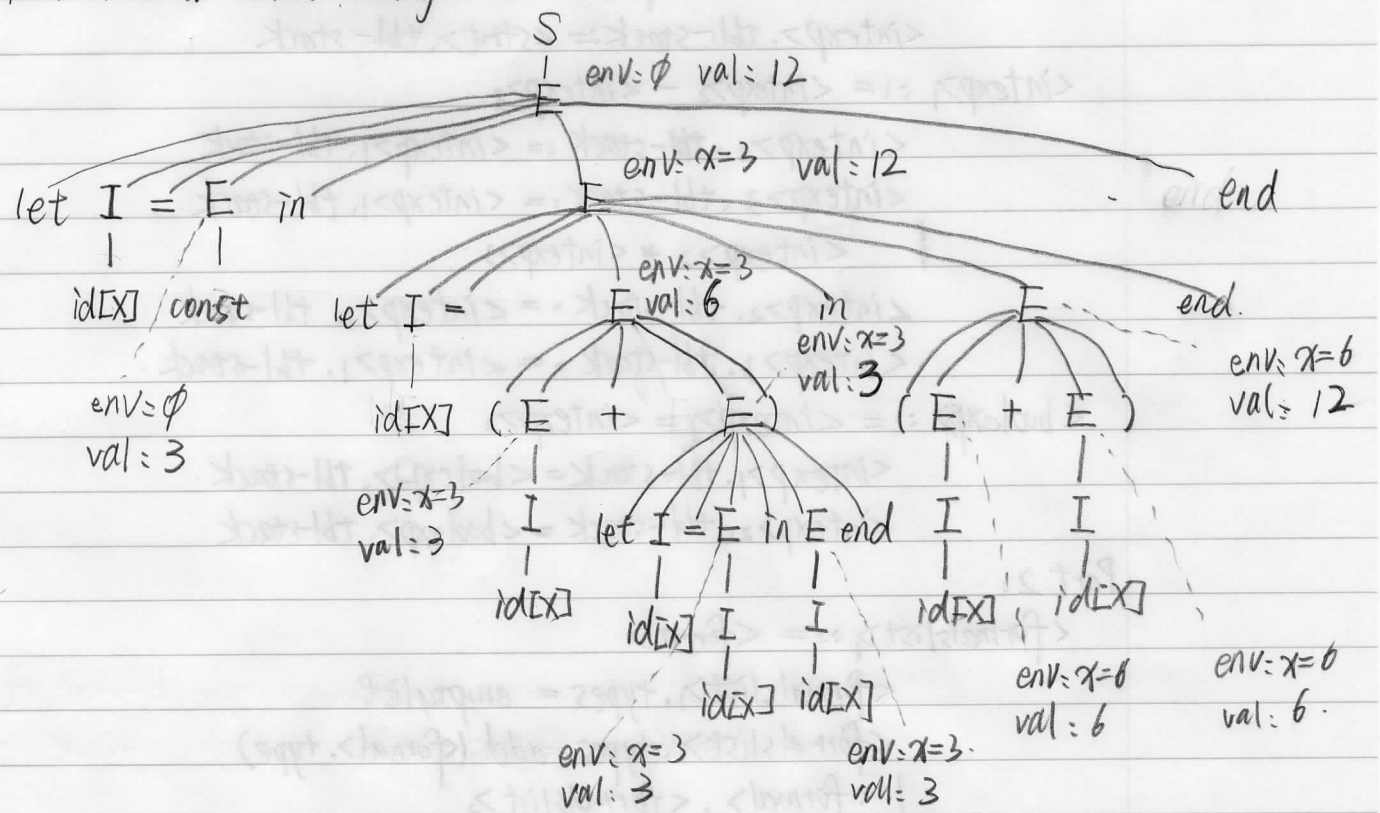
22/25

Programming Language HW1

Zewen Hua

1. Yes, it is a valid string

5/5



2. Attributes:

10/10

Label: associated with <C>

Type of possible values for Label: string set

Evaluation rules:

<C>1 ::= <C>2 ; <C>3

<C>2.Label = <C>1.Label

<C>3.Label = <C>1.Label

| while <be> do <C>2 end

<C>2.Label = <C>1.Label

| <label>: while <be> do <C>2 end

Contain(a,b):
if a∈b, return true
if a∉b, return false

Cond: Contain(label.name, <C>1.Label) = False

<C>2.Label = Add(label.name, <C>1.Label) → Add(a,b): return b with a added.

| break<label>

Cond: Contain(label.name, <C>1.Label) = True

<program> ::= <C>
<C>.Label = ∅

3. Part 1:

2/2

<stmt> ::= return <intexp>
<intexp>.tbl-stack := <stmt>.tbl-stack

<intexp>1 ::= <intexp>2 − <intexp>3
<intexp>2.tbl-stack := <intexp>1.tbl-stack
<intexp>3.tbl-stack := <intexp>1.tbl-stack
| <intexp>2 * <intexp>3.
<intexp>2.tbl-stack := <intexp>1.tbl-stack
<intexp>3.tbl-stack := <intexp>1.tbl-stack

<boolexp> ::= <intexp>1 = <intexp>2
<intexp>1.tbl-stack = <boolexp>.tbl-stack
<intexp>2.tbl-stack = <boolexp>.tbl-stack

Part 2:

2/3

<formalslist>1 ::= <formal>
<formalslist>1.types = emptylist
<formalslist>1.types.add(<formal>.type)
| <formal>, <formalslist>2
<formalslist>1.types = <formalslist>2.types
<formalslist>1.types.add(<formal>.type)

Order not correct. <formal>.type should be in front of <formalslist2>.types

<formal> ::= int id
<formal>.type = INT
| bool id
<formal>.type = BOOL

Part 3:

3/5

<actual> ::= <intexp>
<actual>.type = INT
| <boolexp>
<actual>.type = BOOL

<intexp> ::= id (<actualslist>)
Cond: typeof(id.lexval, <intexp>.tbl-stack) = FUN(...)
<actualslist>.expectedTypes := paramTypes(typeof(id.lexval, <intexp>.tbl-stack))

<actualslist>1 ::= <actual>
Cond: <actualslist>1.expectedTypes = {<actual>.type}
| <actual>, <actualslist>2
Cond: <actualslist>1.expectedTypetypes = {actual.type} ∪ <actualslist>2.expectedtypes

Here missing evaluation rule for <actualslist2>

In the condition, is the union between two sets? If so, there will be no order. And, as stated in the question, expectedTypes is inherited, that is to say there should be some evaluation rule for <actualslist2>.expectedTypes, otherwise it will be empty and your condition will not be valid. Thus in total -2.