n = 3 *Action condition (AC)*

1. open < n
2. close < open

[]

[ ( ]

e.g: found an upper having another valid

3. move to another AC of having another valid

1. goes down to get the **first comb** by applying the I. AC

[ () ] [ (( ]

[ ()( ] [ (() ] [ ((( ]

[ ()() ] [ ()(( ] [ (()) ] [ (())( ] [ ((() ]

[ ()()( ] [()(()] [ (())(] [ (())() ] [ ((()) ]

e.g: keep return back until found that upper

[ ()()() ] [ ()(()) ] **[ (())() ]** **[ (())()) ]** **[ ((())) ]**

**Last comb**

**fourth comb**

**Third comb** by **second comb** going up back to form.

4. **second comb** by applying that order of ACs

2. Each finished comb then pop the last char to see if the upper could form another valid comb