```
COSC 4200 HW#5
1. check
   (read from bottom -> top)
2.
    we are left with any voltal terminals,
    :. loc:=0; while@loc,=0dolocz:=@loc,+lod:comm is well-typed.
                    lociliated olint / @locilidapilintexp
        V 0: int @loci: intexp 0: intexp loci: inthe @loci +1: intexp
     loci: infloc o: intexp @loci = O: boolexp loci = @loci + 1: comm
         104:=0:40mm while@loc, = 030 loc2:=@loc, +1 001:00mm
          loci:=0; while @loci=0 do loci:=@loci+1 od:comm
3. (read from bottom -> top)
     we are left with only valid terminals,
      :. loc; = 0; while 7(@loc; = 0) do loc; =@locz + 1 od : comm
      is well-typed.
                loci: infloc o inf
                                                    locz:infloc 1:tat
               @loci:intexp o:: Aexp
                                              / @bei:intap 1:intexp
              Blaci: Zexp o: Zexp if Zezint, bool} loci: infloc @locz+1: infexp
               (@ loc, = 0): boolexp
                  7(@104 = 0): boolexp 106: =@106241: comm
lociinfloco: ntexp
  la;=0:comm while 7(@loc,=0) to loc, =@loc, +1 od: comm
   loc:=0; while 7(@10c,=0) do loc:=@10c2+1 od:comm
```

(continued on next page) ->

(read from bottom -> top) (continued from previous page) @loc,:intexp, not infloc! Because @loci:=0: comm reamves × 0::n4 @loc, :intloc, yet @loc, : ntexp, @loci: 19toc o: 19texp then the command cannot be reduced @loci:=0: comm into valid terminals. :. @loc, := 0: comm is NOT well-typed. we are left with only valid terminals, :. loc, := @loc, = 0: comm :5 well- typed. @loci:intexp o:intexp I @loci: texp O: Texp where ZE & int, bool} loci:infloc @loci = 0: boolexp 106:=@10c,=0:comm