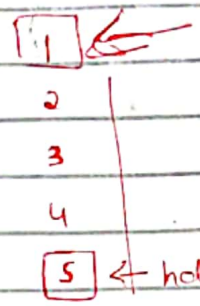
Implementation of Stack on VDM-SL

Stack
stack : Element [*]
push (Element)
pop() : Element
IsEmpty() : Boolean
IsFull() : Boolean

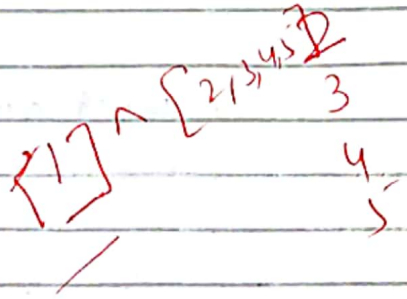


⇒ types

Element = TOKEN

⇒ values

LIMIT : N = 20



① state Stack of

Stack : Element*

inv mk-Stack(s) ≜ len s ≤ LIMIT

init mk-Stack(s) ≜ s = []

end

② push (ItemIn : Element)

ext wr Stack : Element*

pre len stack < LIMIT

post Stack = [ItemIn] ^ $\overline{Stack}$

③ pop() itemRemoved : Element

ext wr Stack : Element*

pre stack ≠ []

post Stack = tl $\overline{Stack}$ ^ ItemRemoved = hd $\overline{Stack}$

④ IsEmpty() query : B

ext rd Stack : Element*

pre TRUE

post query ⇔ stack = []

⑤ IsFull() query : B

ext rd Stack : Element*

pre TRUE

post query ⇔ len stack = LIMIT