Q2) 170722

Gren - generate an infinite list of 1's

Double - multiply every element of an infinite list by 2.

Xs = generate an infinite list of 1's

Ys = multiply each element of Xs by 2.

Case I = Xs → Ys

Xs will have ∞ 1's

Ys will multiply them with 2

It will not terminate

Case II Ys → Xs

↓

if no element present in list

ask for element from Xs.

if element present go ahead and execute

It will not terminate

Since both the execution orders will not terminate, we can say that the program is declaratively concurrent.

Q1>

170722

1) Semantic Stack

SAS

ϕ

local f .. end , $\langle \phi \rangle$

2) Semantic Stack

SAS

F= proc ... end, $F \rightarrow \langle f \rangle$

$\langle f \rangle$

local X .. end , $F \rightarrow \langle f \rangle$

3.) Semantic Stack

local X .. end , $F \rightarrow \langle f \rangle$

SAS

$\langle f \rangle$, proc .. end, F ← free variable

4.) Semantic Stack

$X = \{F\ 3\}$, $f \rightarrow \langle f \rangle$, $X \rightarrow \langle x \rangle$

SAS: $\langle f \rangle$, proc .. end , F

$\langle x \rangle$

5) The statement on top of stack ie.

X = {F 3} will be executed

SAS will remain same

ST

SAS

if . . . . end ,

f → <f> x → <x>

if will evaluate to false ,

6) ST

SAS

3 * {F 2}, f → <f>

x → <x>

~~if . . . . . end~~ .

7) again if condition will be executed and it will evaluate to false.

ST

SAS

3 * 2 * {F 1}, f → <f>

x → <x>

8) again if condition will evaluate to false

ST

3 * 2 * 1 * {F 0}, f → <f>

x → <a>

9) Finally the condition will execute to true -

and we get -

ST

f ⟶ <f>

x ⟶ <x>

SAS

{ <f>, ⟶ . . . }

{ <x>, 6 }