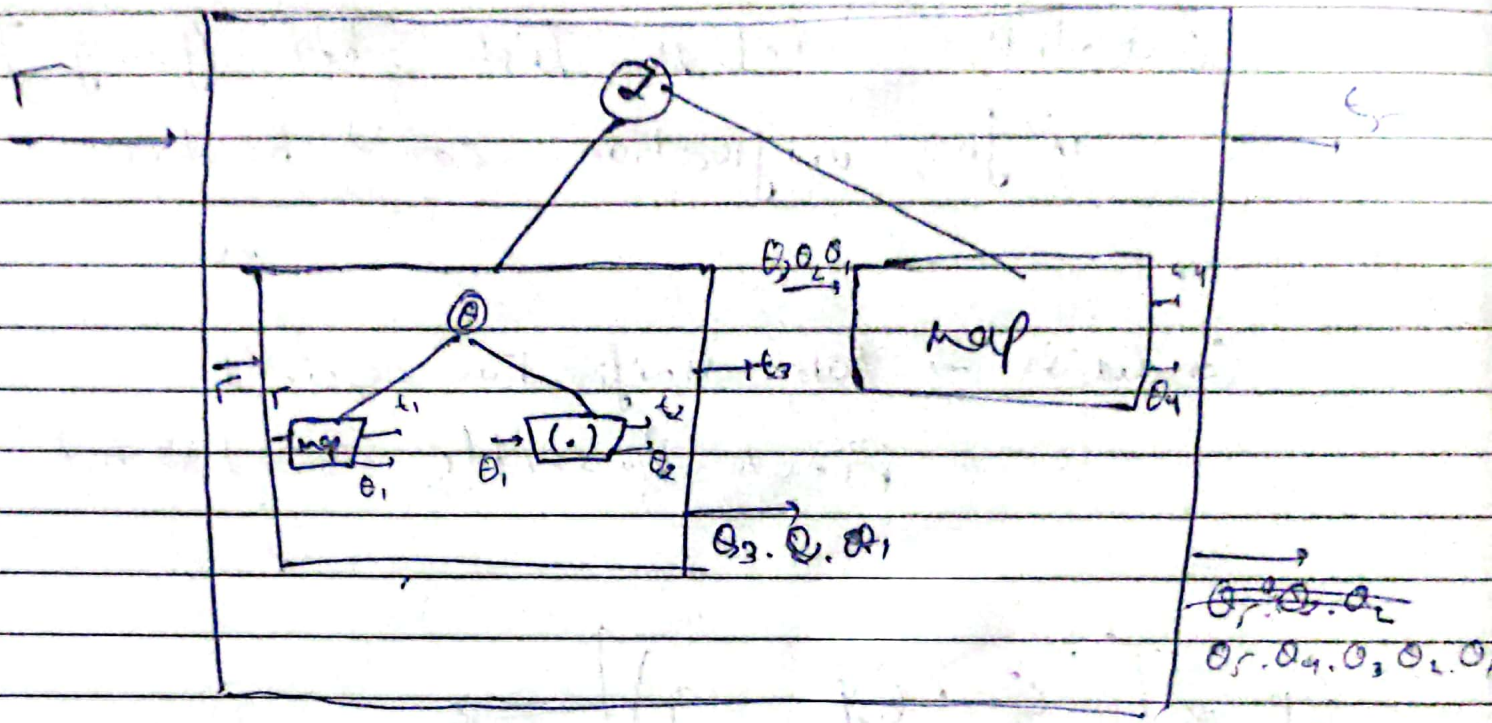
CS350, Nikhil Mehta, Endsem

Q2 map (.) map

(i)



We observe that, their are two pairs of terms which requires to be unified.

- Also observe that t_1, t_2, t_3 are monomorphic in nature.

→ First pair to be unified, in two terms ($\theta_2\ t_1, t_2 \rightarrow t$)

where $\theta_3 = unify(\theta_2\ t_1, t_2 \rightarrow t)$

$\theta_3 \alpha = t_3$

θ_1, θ_2 are θid functions

Nikhil Mehta 19057

- Now let us suppose that the terms (a, b, c ...) are not in Γ i.e. $\{a, b, c \ldots\} \notin FOV(F)$

$$map :: \forall \alpha \beta . (\alpha \to \beta) \to [\alpha] \to [\beta]$$

$$(.) :: \forall \beta \alpha . (\beta \to \gamma) \to (\alpha \to \beta) \to \alpha \to \gamma$$

$$t_1 = (a \to b) \to [a] \to [b]$$

$$t_2 = (d \to e) \to ((c \to d) \to c \to e$$

$$\therefore \theta_2 = \{\beta := d, \alpha := c, \gamma := e\}$$

Let us come to unification Now.

(1) First unification that we need:

$$\theta_3 = unify(\theta_2 t_1, t_2 \to \alpha)$$

$$\theta_3 \alpha = t_3$$

$$\theta_3 := \{a := (d \to e); b := (c \to d) \to c \to e;$$
$$\alpha = [d \to e] \to [(c \to d) \to c \to e]$$

$$t_3 = [d \to e] \to [(c \to d) \to c \to e]$$

Camlin

Nikhil Mehta 190749

We see that θ_3 is MGU.

Let us try to see second unification

$$t_4 \rightarrow (g \rightarrow h) \rightarrow [g] \rightarrow [h]$$

$$\theta_5 = Unify(\theta_4 t_3, t_4 \rightarrow \alpha)$$

MGU doesn't exist.

We observe that for the above $[d \rightarrow e] := (g \rightarrow h)$ is not possible because the former is a list & later one g, h are function.

First Unification possible whereas 2nd is not.

(ii) Fix second unification issue

To fix the issue convert the second occurrence of map to [map].

(iii) Consider the box 3.

$$t_3 = [(g \rightarrow h) \rightarrow [g] \rightarrow [h]]$$

$$\theta_1 = Unify([d \rightarrow e] \rightarrow [(c \rightarrow d) \rightarrow (c \rightarrow e)], [(g \rightarrow h) \rightarrow [g] \rightarrow [h]] \rightarrow \alpha)$$

$\theta_1 = \alpha \rightarrow \alpha = [(c \rightarrow d) \rightarrow (c \rightarrow c)]$

$d := f \rightarrow g, e := [f] \rightarrow [g]$

$$t_1 = [((C \to (g \to h)) \to (C \to [g] \to (h))]$$

$$t_1 = \forall \alpha \beta \gamma . [(\alpha \to \beta \to \gamma) \to (\alpha \to [\beta] \to [\gamma])]$$

190549

Q3

Let us assume two instead of file type, one being openFile & other being ClosedFile.

(a) DecL-INT : Captured by inference.

$$\frac{v \in FV(Decls) \quad INT_V}{v :: INT \quad INT_V :: Valid}$$

In the above v needs to be FV.

DecL-FILE

$$\frac{v :: FV(Decls) \quad FILE_V}{V :: ClosedFile \quad File_V :: Valid.}$$

- OPEN-FILE

$$\frac{v :: File \quad v.open()}{V :: openFile \quad v.open() :: Valid.}$$

We observe that any type of file can be opened.

CLOSE - FILE

V :: Opened File . v. close( )

V :: Closed File v. close( ) :: Valid.

FILE - READ

~~V2~~ V2 :: Opened File V1 = V2. read( )

V1 :: INT V1 = V2. read( ) :: Valid

ASSN - INT

V2 :: INT V1 = V2

V1 :: INT V1 = V2 :: Valid

FILE - Write

V1 :: Opened File V2 :: INT ~~V1. write(V2)~~ V1. write(V2)

~~V1. write(V2)~~ V1. write(V2) :: Valid

Note:

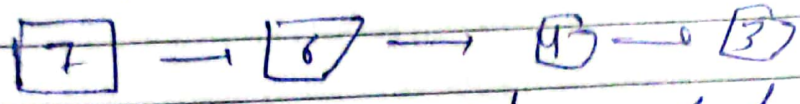
Declared files are closed

Nikhil Mehta 1490349

Que 4

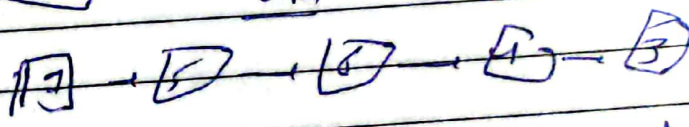
(i) False : Lock on pred is required

Consider example



Let us try to add b/w 6 & 4 a 5 node, if lock is not present, the following can be generated

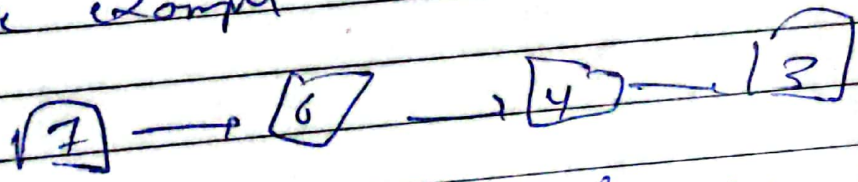
OR



which is incorrect

(ii) True : lock on curr not required.

Above example



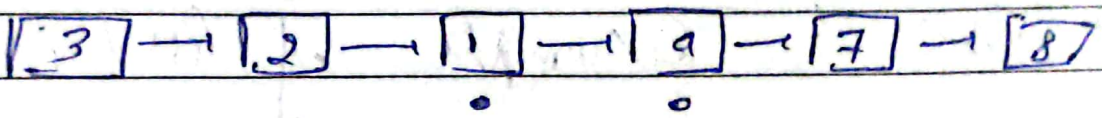
without adding a lock on 6 we can add 5 b/w 6 & 4.

Q-4

(iii) Lock on Pred is necesary for Correctness of remove. (False)

We will try to see using a counter example by first assuming lock on Pred is not necessary.

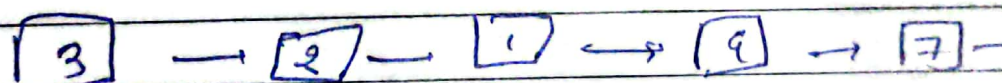
Example of a basic linked list



Let us see what happen if we want to remove node 1 & 9 concurrently.

149

We just lock nodes ~~9 & 7~~ 1 & 9 as assumption lock on Pred is not necessary. Let us assume 1 is removed first & see our list

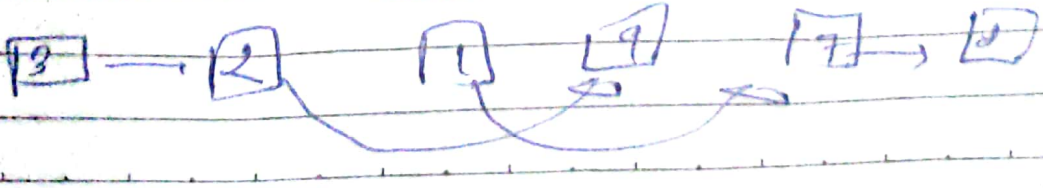


Locked by first T_1

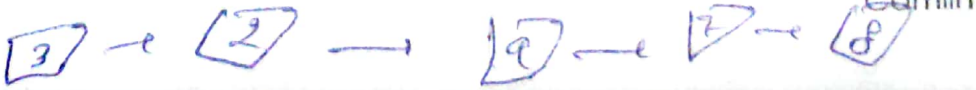
Locked by second T_2

Removing 1

removing 9



Final



Nikhil Mehta

(4)

(iii) ~~False~~:

This is not correct. final desired result because assumption was incorrect.

(v) False: Example of (iii) is sufficient for this part too.