**Problem1**

For R1 , R2 and R3, we let R1 ⊆ A×B, R2 ⊆ B×C and R3 ⊆ C×D;

And we let a ∈ A, b ∈ B, c ∈ C, d ∈ D

Suppose (a, d) ∈ (R1 ; R2) ; R3 , we have :

there is a c with (a, c) ∈ R1 ; R2 and (c, d) ∈ R3

🡨🡪 ∃ c { ∃ b [ (a, b) ∈ R1 and (b, c) ∈ R2 ] and (c, d) ∈ R3 }

🡨🡪 ∃ c ∃ b { (a, b) ∈ R1 and (b, c) ∈ R2 and (c, d) ∈ R3 }

🡨🡪 ∃ b ∃ c { (a, b) ∈ R1 and (b, c) ∈ R2 and (c, d) ∈ R3 }

🡨🡪 ∃ b { [ (a, b) ∈ R1 ] and ∃ c [ (b, c) ∈ R2 and (c, d) ∈ R3 ] }

🡨🡪 ∃ b { (a, b) ∈ R1 and (b, d) ∈ R3 }

🡨🡪 (a, d) (a, d) ∈ R1 ; (R2 ; R3)



For R1 and I, we let R1 ⊆ A×B, I ⊆ S×S;

Suppose (a, b) ∈ I ; R1 , we have :

there is an a with (a, a) ∈ I and (a, b) ∈ R1

🡨🡪 ∃ a { [ (a, a) ∈ I ] and (a, b) ∈ R1 }

🡨🡪 ∃ a { (a, a) ∈ I and (a, b) ∈ R1 }

🡨🡪 ∃ a ∃ b { (a, a) ∈ I and (a, b) ∈ R1 and (b, b) ∈ I }

🡨🡪 ∃ b { (a, b) ∈ R1 and (b, b) ∈ I }

🡨🡪 (a, b) ∈ R1 ; I

🡨🡪 ∃ a ∃ b { (a, b) ∈ R1 }🡨🡪 (a, b) ∈ R1

Counterexample: let R1 = {(1, 3)}, R2 = {(3, 7)}

So R1 ; R2 = { (1, 7) } : there is a 3 with (1, 3) ∈ R1 and (3, 7) ∈ R2

And (R1 ; R2)← = { (7, 1) }

R1 ←; R2← = Φ

because there doesn’t exist an S with (3, S) ∈ R1← and (S, 3) ∈ R2←

so it is not True.



Suppose (a, c) ∈ (R1 ∪ R2) ; R3, we have :

there is a b with (a, b) ∈ R1 ∪ R2 and (b, c) ∈ R3

🡨🡪 ∃ b { [ (a, b) ∈ R1 ∨ (a, b) ∈ R2 ] ∧(b, c) ∈ R3 }

🡨🡪 ∃ b { [ (a, b) ∈ R1 ∧(b, c) ∈ R3 ] ∨ [ (a, b) ∈ R2 ∧ (b, c) ∈ R3 ] } (using distribution)

🡨🡪 ∃ b { [ (a, b) ∈ R1 ∧(b, c) ∈ R3 ] }∨ ∃ b [ (a, b) ∈ R2 ∧ (b, c) ∈ R3 ] }

🡨🡪 (a, c) ∈ (R1 ; R3)∨(a, c) ∈ (R1 ; R3)

🡨🡪 (a, c) ∈ (R1 ; R3)∪(R2 ; R3)

🡨🡪(R1 ∪ R2) ; R3 = (R1 ; R3)∪(R2 ; R3)



Suppose (a, c) ∈ R1 ; (R2 ∩ R3), we have :

there is a b with (a, b) ∈ R1 and (b, c) ∈ (R2 ∩ R3)

🡨🡪 ∃ b { (a, b) ∈ R1 ∧ [ (b, c) ∈ R2 ∧(b, c) ∈ R3 ] }

🡨🡪 ∃ b { [ (a, b) ∈ R1 ∧(b, c) ∈ R2 ] ∧ [ (a, b) ∈ R1 ∧ (b, c) ∈ R3 ] } ①

Here in ①, we have the same b with above condition, that means ① can be able to imply :

🡪 ∃ b { [ (a, b) ∈ R1 ∧(b, c) ∈ R2 ] and ∃ b { [ (a, b) ∈ R1 ∧ (b, c) ∈ R3 ] } ②

And here in ②, this two b can be different value, eg: like the first b have value 1 and the second b with the value of 3;

we can get ① 🡪 ②, but we cannot get ② 🡪 ① due to ∃ b difference;

②🡨🡪 (a, c) ∈ (R1 ; R2) ∧(a, c) ∈ (R1 ; R3)

🡨🡪 (a, c) ∈ (R1 ; R2)∩(R1 ; R3)

So, finally, we have R1 ; (R2 ∩ R3)🡪 (R1 ; R2)∩(R1 ; R3);

but (R1 ; R2)∩(R1 ; R3) -/-> R1 ; (R2 ∩ R3)

**Problem2**

if there is an i such that Ri = Ri+1, that is :

① **[Basic Case]** when j = i, Rj = Ri ;

② **[Inductive Step]** when j = i, we let j = k, Rk = Rj = Ri ;

And then j = k+1＞i: Rj = Rk+1 = Rk∪(R ; Rk) = Ri∪(R ; Ri) = Ri+1 = Ri ;

∴ if there is an i such that Ri = Ri+1, then Rj = Ri for all j≥i

① As I proved in (a) : if k≥i, then we have Rk = Ri 🡪 Rk ⊆ Ri ;

② now if 0≤k＜i, we have Rk+1 = Rk∪(R ; Rk) 🡪 Rk ⊆ Rk+1

Because k<i, so we have Rk ⊆ Ri;

That is : for k≥0, we have Rk ⊆ Ri

P(n) be the proposition that for all m∈ N: Rn; Rm = Rn+m ;

① **[Basic Case]** P(0) is that for all m∈ N: R0; Rm = Rm;

We have proved that I ; R1 = R1 ; I = R1 (in problem1), so R0; Rm = I ; Rm = Rm ;

So P(0) holds;

② **[Inductive Step]** Assume P(n) holds : and that is : for all m∈ N: Rn; Rm = Rn+m;

P(n+1) means for all m∈ N: Rn+1; Rm = Rn+1+m 🡪 (Rn∪(R ; Rn)); Rm =

We have Rn+1; Rm = (Rn∪(R ; Rn)); Rm = (Rn ; Rm)∪((R ; Rn) ; Rm)

= (Rn ; Rm)∪(R ; (Rn ; Rm)) = Rn+m∪(R ; Rm+n)= Rn+1+m ;

So P(n+1) holds;

Finally, that is P(n) holds for all n∈ N;

For Rand R0, R ⊆ S×S can be any binary relation on set S, R0 = {(x,x) : x ∈S};

To prove Rk = Rk+1, we can prove Rk ⊆ Rk+1 and Rk+1 ⊆ Rk instead;

By definition, Rk+1 = Rk∪(R ; Rk);

🡪 Rk ⊆ Rk+1 because of the definition;

Now we explain why Rk+1 ⊆ Rk :

If (a, b) ∈ Rk+1 then (a, b) ∈ Rk or (a, b) ∈ (R ; Rk) (by definition)

First case ① : if (a, b) ∈ Rk :

That is for any tuple (a, b) ∈ Rk+1, (a, b) ∈ Rk;

In this case, Rk+1 ⊆ Rk holds.

Second case ② : if (a, b) ∈ (R ; Rk ) :

Assuming that for any k >=0 : That means ∃ mk { (a, mk) ∈ R and (mk, b) ∈ Rk }

🡪∃ mk-1 { (mk, mk-1) ∈ R and (mk-1, b) ∈ Rk-1 }

🡪∃ mk-2 { (mk-1, mk-2) ∈ R and (mk-2, b) ∈ Rk-1 }

🡪 … …

🡪∃ m1 { (m2, m1) ∈ R and (m1, b) ∈ R1 }

🡪∃ m0 { (m1, m0) ∈ R and (m0, b) ∈ R0 }

∃ m0 { (m1, m0) ∈ R and (m0, b) ∈ R0 }**,**

* m0 = b , we have (b, b) ∈ R0 by definition of R0.

∴m0，m1，… …mk-1，mk ∈ S, from m0 to mk : there are at least (k+1) elements.

And we |S| = k, at least two of them are equal :

🡪there must exists mi and mj such that mi =mj  for 0 ≤ i < j ≤ k;

So (a, b) ∈ Ri with i < k + 1;

that is for any (a, b) ∈ Rk+1 , then (a, b) ∈ (R ; Rk)

from the above, Rk+1 ⊆ Rk and Rk ⊆ Rk+1, so we have Rk = Rk+1

Assume that for all a, b, c ∈ S,

If (a, b) ∈ Rk, (b, c) ∈ Rk

It is clear that Rk ; Rk = {(a, c) : there is a b with (a, b) ∈ Rk, (b, c) ∈ Rk } (by the definition of “ ; ”)

In the proof of (c) : we know Rn; Rm = Rn+m, so Rk ; Rk = R2k ;

∴ (a, c) ∈ Rk ; Rk = R2k ;

In the proof of (d) : If |S| = k, then we get Rk = Rk+1;

And in the proof of (a) : we know that there is a k such that Rk = Rk+1, then Rj = Rk for all j≥k

∵ 2k > k

∴ R2k = Rk 🡪 (a, c) ∈ Rk ; Rk = R2k = Rk 🡪 that is (a, c) ∈ Rk;

∵ (a, b) ∈ Rk, (b, c) ∈ Rk and (a, c) ∈Rk, so Rk transitive when |S| = k

Firstly, we have the common premise of if |S| = k: so we can use the proof of (d) and (e) as a part of our proof in (f).

① **Reflexive :**

(R∪R←)0 = I = {(x,x) : x ∈S}, so (R∪R←)0 is reflexive;

We let L represents relation R∪R←, and L 0 is sysmetric because L 0 = I;

By definition, we have L1 = L0 ∪ ( L; L0) ,

And in the proof of question (c), we have : Rn; Rm = Rn+m ;

So (R∪R←)0 ⊆ (R∪R←)1 ⊆ (R∪R←)2 … … (R∪R←)k-1 ⊆ (R∪R←)k;

That is : for all (x, x), (x, x) ∈ (R∪R←)k is reflexive;

② Sysmetric :

**[Basic Case]** we know (R∪R←)0 = I = {(x,x) : x ∈S};

We let L represents relation R∪R←, which we know that relation R∪R← is sysmetric by itself, and L 0 is sysmetric because L 0 = I;

By definition, we have L1 = L0 ∪ ( L; L0)

∵ L0 is sysmetric, and ( L; L0) is sysmetric due to ( L; L0) = L;

∴ L1 is sysmetric；

**[Inductive Step]** we assume Lk is sysmetric, and then based on the proof in question (c) : Rn;Rm = Rn+m; we can get L1; Lk = L1+k, we need to prove if P(k) : Lk is sysmetric holds, then P(k+1) : Lk+1 is sysmetric holds.

Lk+1 = L1+k = L1; Lk ;

Assume (a, c) ∈ L1+k = L1; Lk, then there is a c such that (a, b) ∈ L1 and (b, c) ∈ Lk,

∵ both L1 and Lk are sysmetric, so there are (b, a) and (c, b) such that (b, a) ∈ L1 and (c, b) ∈ Lk;

∴ (c, a) ∈ Lk+1 = L1+k = L1; Lk 🡪 P(k+1) holds

So Lk = (R∪R←)k is sysmetric.

So for all k >= 0 :

(R∪R←)k is sysmetric.

③ Transitive :

As we have proved in (e), If |S| = k, show that Rk is transitive, and R can be any binary relation on set S.

So we substitude R with (R∪R←)k, so (R∪R←)k  is transitive.

After we have proved ①②③ : (R∪R←)k is reflexive, sysmetric and transitive,so (R∪R←)k is an equivalence relation.

**Problem3**

recursive definition of the binary tree data structure is :

A binary tree is either :

* (Basic definition) an empty tree( with no successors), or
* (Recursive definition) a point pointing to two binary trees, one is left successor and the other is right successor.

1. Counting the number of nodes in a binary tree T:

count(T):

if(T.isEmpty()): **(B)**

return 0

else: **(R)**

return count(T.left\_child) + count(T.right\_child) + 1

1. Counting the number of leaves in a binary tree T:

leaves(T):

if(T.isEmpty()):  **(B)**

return 0

elif(T.left\_child.isEmpty() && T.right\_child.isEmpty()): **(B)**

return 1

else: **(R)**

return leaves(T.left) + leaves(T.right)

1. Counting the number of fully-internal nodes in a binary tree T:

internal(T):

if(T.isEmpty()):  **(B)**

return 0

elif(!T.left\_child.isEmpty() && !T.right\_child.isEmpty()): **(R)**

return internal(T.left) + internal(T.right) + 1

else: **(R)**

return internal(T.left) + internal(T.right)

we assign num\_T(viewed as total number of nodes), num\_T0(viewed as leaves), num\_T1(viewed as a tree which has one child) and num\_T2(viewed as full-internal nodes)to represent three different kinds of Tree structures respectively:

and we create an equation according to the relation between the number of these three kinds of trees and the number of lines (using num\_line to represent) connecting each node:

the fact is that a line comes from the head part of every node apart from the root node, so we get equations as below:

* num\_T = num\_T0 + num\_T1 + num\_T2;
* num\_line = num\_T – 1
* num\_line = num\_T1 + 2 \* num\_T2;

🡪 num\_T0 + num\_T1 + num\_T2 – 1 = num\_T1 + 2 \* num\_T2;

🡪 num\_T2 = num\_T0 – 1

That is : leaves(T) = 1+internal(T)

∴ P(T) holds.­­­­­­­­

**Problem4**



Defining proposition “Alpha uses channel hi” as AH, “Alpha uses channel lo” as AL;

So does BH, BL, CH, CL, DH, DL;

1. ϕ1 = ((((AH∨AL) ∧ (BH∨BL)) ∧ (CH∨CL)) ∧ (DH∨DL))
2. ϕ2 = (((¬(AH∧AL) ∧ ¬(BH∧BL)) ∧ ¬(CH∧CL)) ∧ ¬(DH∧DL))
3. ϕ3 = ((¬((AH∧BH)∨(AL∧BL)) ∧¬((BH∧CH)∨(BL∧CL)))∧¬((CH∧DH)∨(CL∧DL)))

**i.** in the situation of below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| AH | AL | BH | BL | CH | CL | DH | DL |
| T | F | F | T | T | F | F | T |

ϕ1 = T, ϕ2 = T, ϕ3 = T; 🡪 so ϕ1∧ϕ2∧ϕ3 is satisfiable;

**ii.** Based on answer to the previous question, Alpha uses channel hi, Bravo uses channel lo, Charlie uses channel hi and Delta uses channel lo, they can avoid interfere with each other under this assignment, or

In another case that Alpha uses channel lo, Bravo uses channel hi, Charlie uses channel lo and Delta uses channel hi, they can also avoid interfere with each other.