1a) Free occurrences are red:

→

∨ x = y

∀x ∧

P(x) ¬ ∃y

Q(x) ¬

R(x, y)

1b)

1. p ∨ q given
2. (p ∨ q) ∧ ⊤ A ≡ A ∧ ⊤
3. (p ∨ q) ∧ (¬q ∨ q) (¬A ∨ A) ≡ ⊤ EM
4. (p ∧ ¬q) ∨ q (A ∨ C) ∧ (B ∨ C) ≡ (A ∧ B) ∨ C distributivity
5. (¬¬p ∧ ¬q) ∨ q A ≡ ¬¬A
6. ¬(¬p ∨ q) ∨ q ¬(A ∨ B) ≡ ¬A ∧ ¬B De Morgan’s law
7. ¬(p → q) ∨ q ¬A ∨ B ≡ A → B
8. (p → q) → q ¬A ∨ B ≡ A → B

Note: can use Equivalence 20 to skip from step 4 to 7

1c)

i) To show Alt∨E is a derived rule of ⊢, we need to show A ∨ B, A → B ⊢ B:

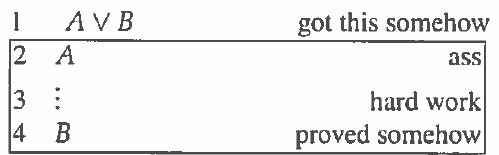
|  |  |
| --- | --- |
| 1. A ∨ B given 2. A → B given | |
| 3. A ass  4. B →E(2,3) | 5. B ass  6. B √(5) |
| 7. B ∨E(1,3,4,5,6) | |

NB: I used the implication A → B as a given, because the rule (Alt∨E) will always be applied to three lines (as in the example in the exam paper), two of which will form the beginning and the end of a box, thereby forming an implication.

(alternative maybe: ) I suggest we don’t need to use A->B.

|  |  |
| --- | --- |
| 1.A ∨ B given | |
| 2. A ass  3 … hard work  4. B proved somehow | 5. B ass  6. B √(5)lj |
| 7. B ∨E(1,2,4,5,6) | |

Another solution:



5 A -> B ->I(2,4)

|  |  |
| --- | --- |
| 6. A ass  7. B  -> E (5, 6) | 8. B ass  9. B √(8) |

10. B vE(6,7,8,9,1)

ii) To show A ∨ B, A → C, B → C ⊢a C:

|  |  |  |
| --- | --- | --- |
| 1. A ∨ B given 2. A → C given 3. B → C given  |  |  | | --- | --- | | 4. ¬C ass   |  | | --- | | 5. A ass  6. C →E(2,5)  7. ⊥ ¬E(4,6)  8. B ⊥E(7) |   9. B Alt∨E(1,5,8)  10. C →E(3,9)  11. ⊥ ¬E(4,10) |   12. C PC(4,11) |

iii) We have shown that A ∨ B, A → C, B → C ⊢a C in c(ii). In other words, if we can start with either disjunct in A ∨ B and obtain the same C, we know C is true, because at least one of A and B is true. Furthermore, if we can start with either disjunct in A ∨ B and using ND obtain the other, we know this one is true.

(Question is “Explain why?” ??)

iii) Alternative: Argument by definition of ∨E:

Definition:

|  |  |
| --- | --- |
| 1. A ∨ B | |
| 2. A ass  3. ...  4. C we did it | 5. B ass  6. ...  7. C we did it |
| 8. C ∨E(1,2,5,7) | |
|
|
|
|

This requires A -> C and B -> C, and we have shown in part (ii) that A∨B, A -> C, B -> C ⊢a C, so we can replace any usage of ∨E by these rules in ⊢a . Therefore, it is a derived rule.

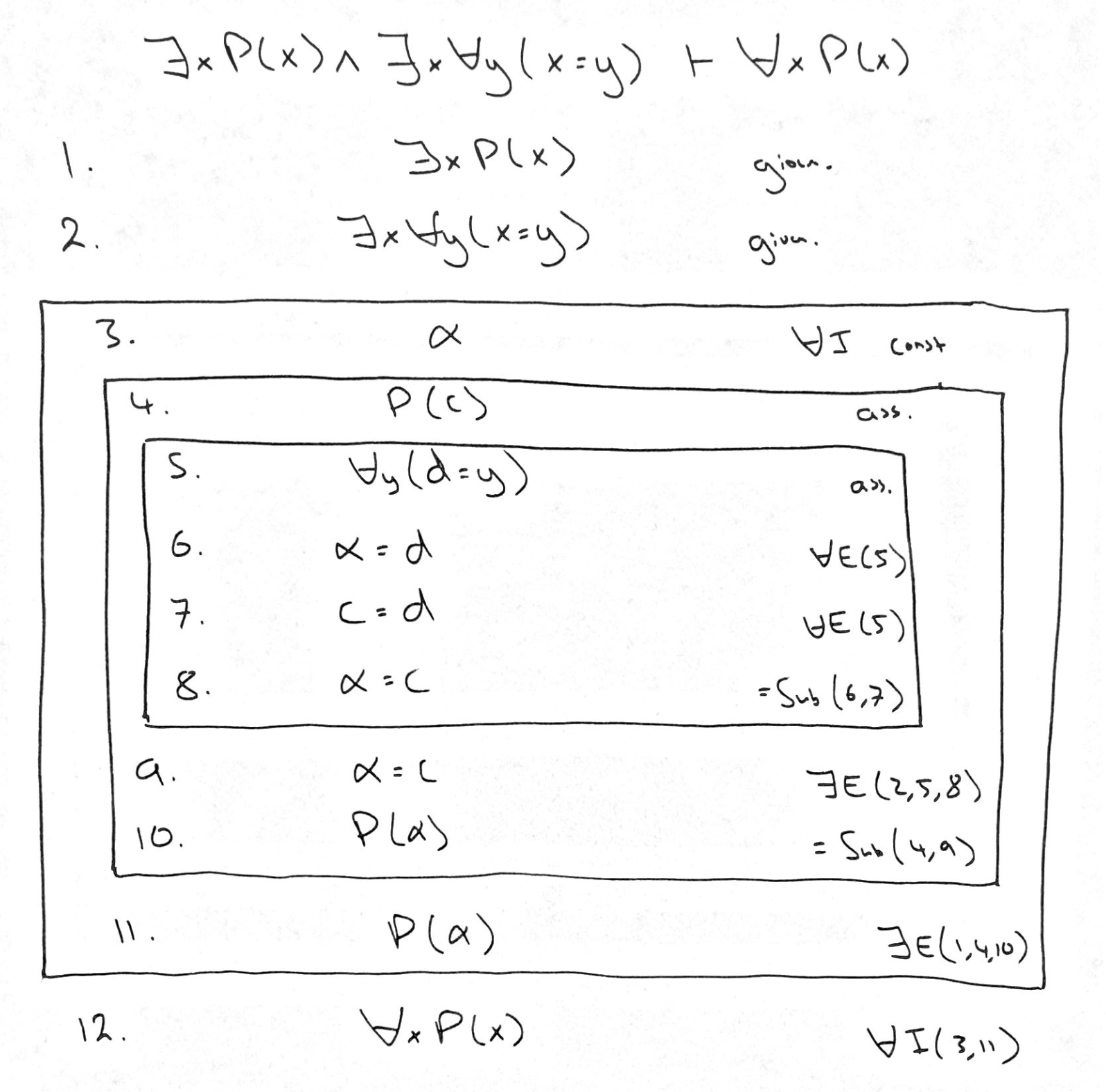
iv) Sound – any sentence proven by ⊢a is valid. Complete – any valid first-order formula can be proven in ⊢a.

v) Yes, because ⊢ is sound and complete, and it has been shown that ∨E can be derived in ⊢a. Because ∨E is the only rule present in ⊢ that is missing in ⊢a , this is sufficient.

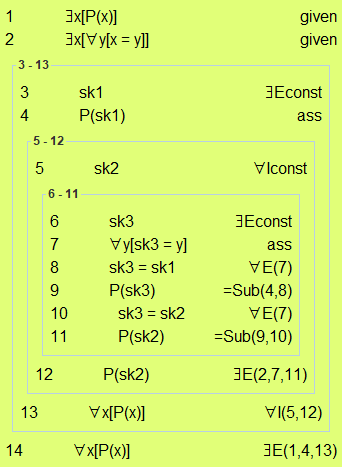
Also, it has been shown that AltvE can be derived from ⊢, which we know to be sound, and so ⊢a must be sound also.

9 2a) Natural deduction question

I want to die --same, also me, and me plus me, and me too.

2 a (Rohan Pritchard’s Version) - Basically the same but fixes the line 5 issue

2a) Pandora



2b)

i) ∃x∃y(in(x, xs) ∧ x = y \* 2)

ii) ∀x∀n(count(x, xs, n) ↔ n ≤ 1)

iii) ∀x∃n(count(x, xs, n) ∧ count(x, ys, n)) ∧ ∀m∀n((n < #(ys)) ∧ (m ≤ n) → ys !! m ≤ ys !! n)

iv) ∃ys∃zs(merge(ys, zs, xs) ∧ ∀y(in(y, ys) → ∃n(y = n \* 2)) ∧ ∀z(in(z, zs) → ∃n(z = (n \* 2) + 1)) ∧ #(zs) < #(ys))

2c)

i)

A1. xs = [1], ys = [1, 1] (membership does not ensure uniqueness in ys)

A2. xs = [], ys = [1, 1] (any number of occurrences in ys means not in xs, 2 is a valid count)

A3. xs = [1, 2], ys = [2, 1] (merge is order sensitive)cate

ii) ∀n(in(n, xs) ↔ in(n, ys)) ∧ ∀i(i < #(xs) → count(xs!!i, ys, 1))

or: ∀n(in(n, xs) ↔ in(n, ys)) ∧ ∀n(in(n, xs) → count(n, ys, 1))

What about this?: ∀n(in(n, xs) → count(n, ys, 1)) ^ ∀n(¬in(n, xs) → ¬in(n,ys))

Yep, its fine

Shouldn’t it be: ∀n(in(n, xs) → count(n, ys, 1)) ^ ∀n(¬in(n, xs) ↔ ¬in(n,ys))?

🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈

🎈 AND THAT’S AN EXAM DONE🎈

🎈ONLY TOOK LIKE 5 HOURS, EASY 🎈

🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈 🎈

.''.

.''. \*''\* :\_\/\_: .   
 :\_\/\_: . .:.\*\_\/\_\* : /\ : .'.:.'.  
 .''.: /\ : \_\(/\_ ':'\* /\ \* : '..'. -=:o:=-  
 :\_\/\_:'.:::. /)\\*''\* .|.\* '.\'/.'\_\(/\_'.':'.'  
 : /\ : ::::: '\*\_\/\_\* | | -= o =- /)\ ' \*  
 '..' ':::' \* /\ \* |'| .'/.\'. '.\_\_\_\_\_  
 \* \_\_\*..\* | | : |. |' .---"|  
 \_\* .-' '-. | | .--'| || | \_| |  
 .-'| \_.| | || '-\_\_ | | | || |  
 |' | |. | || | | | | || |  
 \_\_\_| '-' ' "" '-' '-.' '` |\_\_\_\_  
jgs~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

1. TheyM = fireworks ass
2. TheyM = fireworks √(1)

| 10 PRINT “THEY M ARE FIREWORKS” |

| 20 GOTO 10 |