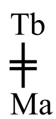
Sample Problem

P1. ∃x(Mx & ∀y(Ty → ~Ayx))
P2. ∀x(Mx → ∃y(Ty & ~Ayx))
P3. ∀x(Tx → ∃y(My & Axy))

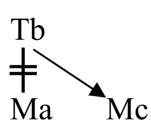
Aim is to show that:

Goal 1: ∃x∃y(Tx & Ty & x≠y)
Goal 2: ∃x∃y∃z(Mx & My & Mz & x≠y & x≠z & y≠z)

1 (1) ∃x(Mx & ∀y(Ty → ~Ayx)) A
2 (2) ∀x(Mx → ∃y(Ty & ~Ayx)) A
3 (3) ∀x(Tx → ∃y(My & Axy)) A
4 (4) Ma & ∀y(Ty → ~Aya) A According to line 1, there is an M that no T is related to. I will call it "a" by making an assumption for ∃E.
4 (5) Ma 4 &E
4 (6) ∀y(Ty → ~Aya) 4 &E
2 (7) Ma → ∃y(Ty & ~Aya) 2 ∀E
2,4 (8) ∃y(Ty & ~Aya) 5,7 →E
9 (9) Tb & ~Aba A P2 tells me something about every M – namely, that there is a T it isn't related to. I will call the T that isn't related to a "b". At this point, it becomes helpful to keep track of what we have shown by a diagram (or a partial interpretation) which we will continue to fill in as we go along.

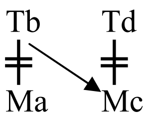


9 (10) Tb 9 &E
9 (11) ~Aba 9 &E
3 (12) Tb → ∃y(My & Aby) 3 ∀E
3,9 (13) ∃y(My & Aby) 10,12 →E
14 (14) Mc & Abc A P3 tells me that for each T, there is an M they are related to. I will call the M that b is related to "c".



14 (15) Mc 14 &E
14 (16) Abc 14 &E
9,14 (17) a≠c 11,16 NI b is related to c, but not to a; so a and c must be different. At this point I have proved that there are two different objects which are Ms – a and c.
2 (18) Mc → ∃y(Ty & ~Ayc) 2 ∀E
2,14 (19) ∃y(Ty & ~Ayc) 15,18 →E

20 (20) Td & ~Adc A By P2 there must be a T that isn't related to c. Call it "d".
20 (21) Td 20 &E
20 (22) ~Adc 20 &E
14,20 (23) b≠d 16,22 NI Since b is related to c but d isn't, they can't be the same. At this point I have proved there are two different Ts.



My first goal was that there are two Ts. I can now prove that.

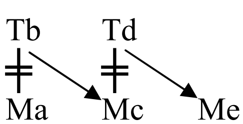
9,14,20 (24) Tb & Td & b≠d 10,21,23 &I
9,14,20 (25) ∃x∃y(Tx & Ty & x≠y) ∃I x2 24

If I was trying to minimize the number of lines to use, I could postpone using ∃Es now and do them all later. But if I wanted to make sure to get all of the partial credit along the way in case I messed up later, I should just go ahead and do the ∃Es now to show that this does follow just from 1-4.

2,9,14 (26) ∃x∃y(Tx & Ty & x≠y) 19,25 ∃E(20)
2,3,9 (27) ∃x∃y(Tx & Ty & x≠y) 13,26 ∃E(14)
2,3,4 (28) ∃x∃y(Tx & Ty & x≠y) 8,27 ∃E (9)
1,2,3 (29) ∃x∃y(Tx & Ty & x≠y) 1,28 ∃E (4)

I have now proved my first goal. To prove my second goal, I will go back to where I was (basically line 23) and continue from there rather than starting over so that I don't have to repeat my work again.

3 (30) Td → ∃y(My & Ady) 3 ∀E
3,20 (31) ∃y(My & Ady) 21,30 →E
32 (32) Me & Ade A By P3 this T has to be related to some M. I will call it "e". I have now produced the following diagram:

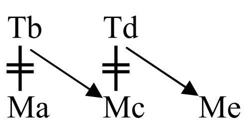


32 (33) Me 32 &E
32 (34) Ade 32 &E
20,32 (35) c≠e 22,34 NI Since d is related to e but isn't related to c, c and e can't be the same. I haven't yet shown that a isn't c. Maybe the M that T is related to is a. Nothing in my proof (or diagram) yet rules that out. However,

the first thing we did was get an M that nothing was related to. That means that d can't be related to it.

4 (36) Td → ~Ada 6 ∀E
4,20 (37) ~Ada 21,36 →E
4,20,32 (38) a≠e 34,38 NI I now have the three different Ms. I can finish my proof by putting them in the right place to use ∃I and ∃E.
4,9,14,20,32 (39) Ma & Mc & Me & a≠c & a≠e & c≠e 5,15,17,33,35,38 &I
4,9,14,20,32 (40) ∃x∃y∃z(Mx & My & Mz & x≠y & x≠z & y≠z) ∃I x3 39
3,4,9,14,20 (41) ∃x∃y∃z(Mx & My & Mz & x≠y & x≠z & y≠z) 31,40 ∃E(32)
2,3,4,9,14 (42) ∃x∃y∃z(Mx & My & Mz & x≠y & x≠z & y≠z) 19,41 ∃E(20)
2,3,4,9 (43) ∃x∃y∃z(Mx & My & Mz & x≠y & x≠z & y≠z) 13,42 ∃E(14)
2,3,4 (44) ∃x∃y∃z(Mx & My & Mz & x≠y & x≠z & y≠z) 8,43 ∃E(9)
1,2,3 (45) ∃x∃y∃z(Mx & My & Mz & x≠y & x≠z & y≠z) 1,44 ∃E(4)

I have now proved my goal. My final diagram looks like this:



(I also know that ~Ada – but I don't know a good symbol for that!)

So if asked to produce a model of these sentences, here is one:

U: {a,b,c,d,e}
T: {b,d}
M: {a,c,e}
A: {⟨b,c⟩,⟨d,e⟩}

There are other models that would work. More elements would be okay, I could add arrows going from the Ms to the Ts, etc. Practically the only things I can't add are Aba, Ada, and Adc.

Hopefully this should give you a great start on your take-home.