J Bolton Set 6

Logic for Applications
(Ch 6) P 118: 1, 12a, 15
(Ch 7) P 125: 1, 6

1.

```
         F((Ex)(p(x) v q(x)) <-> (Ex)p(x) v (Ex)q(x))
                               /    \
         T((Ex)(p(x) v q(x))      F((Ex)(p(x) v q(x))
              |                        |
         F((Ex)p(x) v (Ex)q(x))   T((Ex)p(x) v (Ex)q(x))
              |                       /              \
         F((Ex)p(x))              T((Ex)p(x))        T((Ex)q(x))
              |                        |               |
         F((Ex)q(x))              T(p(t))            T(q(t))
              |                        |               |
         F(q(t))        F((Ex)(p(x) v q(x))    F((Ex)(p(x) v q(x))
              |                        |               |
         F((Ex)p(x))         F(p(t) v q(t))      F(p(t) v q(t))
              |                        |               |
         F(p(t))                    F(p(t))          F(p(t))
              |                        |               |
         T((Ex)(p(x) v q(x))        F(q(t))          F(q(t))
              |                        |               |
         T(p(c1) v q(c1))              X               X
         /         \
    T(p(c1))       T(q(c1))
        |            |
        X            X
```

12.
(a)

```
         F (AxEy~Az. P(x,y,z) <-> AxEyEz. ~p(x,y,z))
                       /         \
         T(AxEy~Az. P(x,y,z))     F(AxEyEz. ~p(x,y,z))
                      |            |
         F(AxEyEz. ~p(x,y,z))     T(AxEy~Az. P(x,y,z))
                      |            |
         F(EyEz. ~p(c1,y,z))      T(Ey~Az. P(t,y,z))
                       |           |
         F(Ez. ~p(c1,t,z))        T(~Az. P(t,c1,z))
                       |           |
              F(~p(c1,t,t'))      F(Az. P(t,c1,z))
                       |           |
              T(p(c1,t,t'))       F(P(t,c1,c2))
                       |           |
         T(AxEy~Az. P(x,y,z))     F(AxEyEz. ~p(x,y,z))
                       |           |
```

```
          T(Ey~Az. P(t,y,z))      F(EyEz. ~p(c3,y,z))
                         |          |
          T(~Az. P(t,c2,z))       F(Ez. ~p(c3,t,z))
                         |          |
          F(Az. P(t,c2,z))        F(~p(c3,t,t'))
                         |          |
               F(P(t,c2,c3))      T(p(c3,t,t'))
                         |          |
                         X          X
```

15.

Is prop 6.10 saying that if we have a non-contradictory entry, we extend it to n+1 levels, its entry becomes reduced? How?

```
     tk
       \
        TP(c0,c0)
               \
                T(Ax)R(x,x)
                    \
                     TR(t1,t1)
                         |
                     T(Ax)R(x,x)
                         |
                     TR(t2,t2)
                         |
                     T(Ax)R(x,x)
                         |
                     TR(t3,t3)
```

Ch. 7

1.

->

For all means that we can take any term n in our language and use them for the parameters in the functions and predicates in our scope (p 85) for all structures. If we extend L to include new constants c1, c2... cn, then the forall will encompass them as well.

<-

The constants c1, c2 .. cn were not in the structure L, so they extend L. Again, the forall quantifier refers to any term in *any* structure, so that c1, c2 .. cn are included within that quantifier.

6.

By 7.8, the soundness and completeness theorems, we know (i) and (iii) to be equivalent. We also know (ii) and (iv) to be equivalent. We must then prove that (i) and (ii) are equivalent, as well as (iii) and (iv). Semantically sigma |= phi states that on the truth of all the premises in sigma, then phi must be true. |= &sigma -> phi states that if we take the conjunction of all the formulas in sigma, then phi will then be true. If &sigma is false, then the implication is still trivially true, so it is a tautology. The statement sigma |- phi is the syntactic version and means that we have an atomic tableau proof of phi with every formula of sigma added as an atomic tableau to the tree. |- &sigma -> phi means that we again make a tableau proof and add each conjunction in sigma. According to 6.2b, each conjunctive formula that we add is added in the same way as we did the formulas in sigma |- phi.