Prove: $\forall n \geq k(P(n))$

1. (Base) $P(k)$
2. (Inductive step) $\forall n \geq k[(\forall i, k \leq i \leq n(P(n))) \Rightarrow P(n+1)]$
    1. Fix $n \geq k$
    2. Assume $\forall i, k \leq i \leq n(P(i))$
    3. $P(n+1)$
    4. QED — Implication, UG
3. QED — Strong induction

Prove: $\forall n \geq k(P(n))$

1. (Base) $P(k)$
2. (Base) $P(k+1)$
3. (Inductive step) $\forall n \geq k+1[(\forall i, k \leq i \leq n(P(n))) \Rightarrow P(n+1)]$
    1. Fix $n \geq k+1$
    2. Assume $\forall i, k \leq i \leq n(P(i))$
    3. $P(n+1)$
    4. QED
4. QED

Prove: $\forall e \in F(P(e))$

1. (Base) $P(0)$
2. (Base) $P(1)$
3. (Inductive step) $\forall e, e' \in F(P(e) \wedge P(e') \Rightarrow P((e \wedge e'))$
4. (Inductive step) $\forall e, e' \in F(P(e) \wedge P(e') \Rightarrow P((e \vee e'))$
5. (Inductive step) $\forall e \in F(P(e) \Rightarrow P((\neg e))$
6. QED Struct. ind. on Boolean exps.

Prove: $\forall t \in T(P(t))$

1. (Base) $P(n)$, where n is the single-node tree.
2. (Inductive step) $\forall t \in T(P(t) \Rightarrow P(makeleft(t))$
3. (Inductive step) $\forall t \in T(P(t) \Rightarrow P(makeright(t))$
4. (Inductive step) $\forall t, t' \in T(P(t) \wedge P(t') \Rightarrow P(makeboth(t, t'))$
5. QED Struct. ind. on binary trees

Prove: $\forall s \in S(P(s))$

1. (Base) $P(\lambda)$
2. (Inductive step) $\forall s \in S(P(s) \Rightarrow P(s0))$
3. (Inductive step) $\forall s \in S(P(s) \Rightarrow P(s1))$
4. QED Struct. ind. on strings.

Prove: $\forall e \in F(P(e))$

1. (Base) $P(0)$ — 0 has no parens.
2. (Base) $P(1)$ — No parens
3. (Inductive step) $\forall e, e' \in F(P(e) \wedge P(e') \Rightarrow P((e \wedge e'))$
    1. Fix $e, e' \in F$.
    2. Assume $P(e) \wedge P(e')$, that is, each of e and e' has same no. of left and right parens.
    3. $P((e \wedge e'))$, that is, the exp. $e \wedge e'$ has same no. of left and right parens. — By ind. hyp. (3.2), and adding one left and one right paren.
    4. QED — Implication, UG
4. (Inductive step) $\forall e, e' \in F(P(e) \wedge P(e') \Rightarrow P((e \vee e'))$ — Similar to the previous case.
5. (Inductive step) $\forall e \in F(P(e) \Rightarrow P((\neg e))$ — Similar to the previous case.
6. QED — Struct. ind. on Boolean exps

Prove: $\forall t \in T(P(t))$

1. (Base) $P(n)$, where n is single-node tree. 0 edges, 1 node
2. (Inductive step) $\forall t \in T(P(t) \Rightarrow P(makeleft(t))$
   1. Fix t.
   2. Assume $P(t)$, that is, $|\mathit{edges}(t)| + 1 = |\mathit{nodes}(t)|$.
   3. $P(\mathit{makeleft}(t))$, that is, $|\mathit{edges}(\mathit{makeleft}(t))| + 1 = |\mathit{nodes}(\mathit{makeleft}(t))|$.
      1. $|\mathit{edges}(\mathit{makeleft}(t))| = |\mathit{edges}(t)| + 1$
      2. $|\mathit{nodes}(\mathit{makeleft}(t))| = |\mathit{nodes}(t)| + 1$.
      3. QED Algebra, 2.2, 2.3.1, 2.3.2
   4. QED Implication, UG
3. (Inductive step) $\forall t \in T(P(t) \Rightarrow P(makeright(t))$ Similar to prev. case.
4. (Inductive step) $\forall t, t' \in T(P(t) \wedge P(t') \Rightarrow P(makeboth(t, t'))$
   1. Fix t.
   2. Assume $P(t) \wedge P(t')$, that is, $|\mathit{edges}(t)| + 1 = |\mathit{nodes}(t)|$ and $|\mathit{edges}(t')| + 1 = |\mathit{nodes}(t')|$.
   3. $P(\mathit{makeboth}(t, t'))$, that is, $|\mathit{edges}(\mathit{makeboth}(t, t'))| + 1 = |\mathit{nodes}(\mathit{makeboth}(t))|$.
      1. $|\mathit{edges}(\mathit{makeboth}(t, t'))| = |\mathit{edges}(t)| + |\mathit{edges}(t')| + 2$
      2. $|\mathit{nodes}(\mathit{makeboth}(t, t'))| = |\mathit{nodes}(t)| + |\mathit{nodes}(t')| + 1$.
      3. QED Algebra, 4.2, 4.3.1, and 4.3.2
   4. QED Implication, UG
5. QED Struct. ind. on binary trees

Prove: $\forall s \in S(P(s))$

1. (Base) $P(\lambda)$ — No patterns of either kind.
2. (Inductive step) $\forall s \in S(P(s) \Rightarrow P(s0))$
   1. Fix s.
   2. Assume $P(s)$.
   3. $P(s0)$
      1. *num*$(01, s0) \leq$ *num*$(10, s0) + 1$. — ???
      2. If $s0$ ends in 0 then *num*$(01, s0) \leq$ *num*$(10, s0)$. — ???
      3. QED — Conjunction
   4. QED — Implication, UG
3. (Inductive step) $\forall s \in S(P(s) \Rightarrow P(s1))$
   1. Fix s.
   2. Assume $P(s)$.
   3. $P(s1)$
      1. *num*$(01, s1) \leq$ *num*$(10, s1) + 1$. — ???
      2. If $s1$ ends in 0 then *num*$(01, s1) \leq$ *num*$(10, s1)$. — ???
      3. QED — Conjunction
   4. QED — Implication, UG
4. QED — Struct. ind. on strings.

3. $P(s1)$

1. *num*$(01, s1) \leq$ *num*$(10, s1) + 1$.
   1. If s ends in 0 then *num*$(01, s1) \leq$ *num*$(10, s1) + 1$.
      1. Assume s ends in 0.
      2. *num*$(01, s) \leq$ *num*$(10, s)$ — ind. hyp. (3.2), part 2.
      3. *num*$(01, s1) =$ *num*$(01, s) + 1$ — Adding one more 01.
      4. *num*$(10, s1) =$ *num*$(10, s)$
      5. *num*$(01, s1) \leq$ *num*$(10, s1) + 1$. — Algebra (3.3.1.1.2, ..3, and ..4)

      QED — Implication
   2. If s ends in 1 then *num*$(01, s1) \leq$ *num*$(10, s1) + 1$.
      ind. hyp. part 1; no new 01s.
   3. If $s = \lambda$ then *num*$(01, s1) \leq$ *num*$(10, s1) + 1$.
      $s1$ is just 1, which has no 01s.
   4. QED — Cases
2. If $s1$ ends in 0 then *num*$(01, s1) \leq$ *num*$(10, s1)$.
   Vacuously true,
   because $s1$ doesn't end in 0.
3. QED — Conjunction

3. $P(s0)$

1. *num*$(01, s0) \leq$ *num*$(10, s0) + 1$. ind. hyp. part 1; no new 01s.
2. If $s0$ ends in 0 then *num*$(01, s0) \leq$ *num*$(10, s0)$.
    1. *num*$(01, s0) \leq$ *num*$(10, s0)$.
        1. If s ends in 0 then *num*$(01, s0) \leq$ *num*$(10, s0)$. ind. hyp. part 2; no new 01s
        2. If s ends in 1 then *num*$(01, s0) \leq$ *num*$(10, s0)$.
            1. Assume s ends in 1.
            2. *num*$(01, s) \leq$ *num*$(10, s) + 1$. ind. hyp. part 1
            3. *num*$(01, s0) =$ *num*$(01, s)$
            4. *num*$(10, s0) =$ *num*$(10, s) + 1$ Exactly one new 10.
            5. *num*$(01, s0) \leq$ *num*$(10, s0)$ Algebra
            6. QED Implication
        3. If $s = \lambda$ then *num*$(01, s0) \leq$ *num*$(10, s0)$. $s0$ is just 1, which has no 01s or 10s.
        4. QED Cases
    2. QED Propositional reasoning (truth table)
3. QED Conjunction

1. (Base) $P(t_0)$, where t_0 is the single-node tree. By def., *numnodes*$(t_0) = 1$ and *numedges*$(t_0) = 0$
2. (Inductive step) $\forall t \in T(P(t) \Rightarrow P($*makeleft*$(t))$
   1. Fix t.
   2. Assume $P(t)$, that is, *numedges*$(t) + 1 =$ *numnodes*(t).
   3. $P($*makeleft*$(t))$, that is, *numedges*(*makeleft*$(t)) + 1$ $=$ *numnodes*(*makeleft*(t)).
      1. *numedges*(*makeleft*$(t)) =$ *numedges*$(t) + 1$ — Definition of *numedges*
      2. *numnodes*(*makeleft*$(t)) =$ *numnodes*$(t) + 1$ — Definition of *numnodes*
      3. QED — Algebra
   4. QED
3. (Inductive step) $\forall t \in T(P(t) \Rightarrow P($*makeright*$(t))$ — Similar to the previous case.
4. (Inductive step) $\forall t, t' \in T(P(t) \wedge P(t') \Rightarrow P($*makeboth*$(t, t'))$
   1. Fix t.
   2. Assume $P(t) \wedge P(t')$, that is, *numedges*$(t) + 1 =$ *numnodes*(t) and *numedges*$(t') + 1 =$ *numnodes*(t').
   3. $P($*makeboth*$(t, t'))$, that is, *numedges*(*makeboth*$(t, t')) + 1$ $=$ *numnodes*(*makeboth*(t)).
      1. *numedges*(*makeboth*$(t, t')) =$ *numedges*$(t) +$ *numedges*$(t') + 2$ — Definition of *numedges*
      2. *numnodes*(*makeboth*$(t, t')) =$ *numnodes*$(t) +$ *numnodes*$(t') + 1$ — Definition of *numnodes*
      3. QED — Algebra
   4. QED
5. QED — Struct. ind. on binary trees