Principia Mathematica's Propositional Logic in *Coq*

Landon D. C. Elkind[*]

April 20, 2021

Abstract

This file contains the *Coq* code for the *Principia* Rewrite project's encoding of the propositional logic given in ∗1 − ∗5. The Github repository with this *Coq* file is here: https://github.com/LogicalAtomist/principia. To receive updates about the project, visit the *Principia Rewrite* project page: https://www.principiarewrite.com/. You can also follow the *Principia* Rewrite project on Twitter: https://twitter.com/thePMrewrite.

```
Require Import Unicode.Utf8.
Require Import Classical_Prop.
Require Import ClassicalFacts.
Require Import PropExtensionality.

Module No1.

Import Unicode.Utf8.
Import ClassicalFacts.
Import Classical_Prop.
Import PropExtensionality.

  (*We first give the axioms of Principia
for the propositional calculus in *1.*)

Theorem Impl1_01 : ∀ P Q : Prop,
  (P → Q) = (¬P ∨ Q).
  Proof. intros P Q.
  apply propositional_extensionality.
  split.
  apply imply_to_or.
  apply or_to_imply.
```

[*]Department of Philosophy, University of Alberta, `elkind at ualberta dot ca`. This research was conducted as an Izaak Walton Killam Postdoctoral Fellow.

```
Qed.
  (*This is a notational definition in Principia:
    It is used to switch between "∨" and "→".*)

Theorem MP1_1 : ∀  P Q : Prop,
  (P → Q) → P → Q. (*Modus ponens*)
  Proof. intros P Q.
  intros iff_refl.
  apply iff_refl.
Qed.
  (*1.11 ommitted: it is MP for propositions
      containing variables. Likewise, ommitted
      the well-formedness rules 1.7, 1.71, 1.72*)

Theorem Taut1_2 : ∀ P : Prop,
  P ∨ P → P. (*Tautology*)
  Proof. intros P.
  apply imply_and_or.
  apply iff_refl.
Qed.

Theorem Add1_3 : ∀ P Q : Prop,
  Q → P ∨ Q. (*Addition*)
  Proof. intros P Q.
  apply or_intror.
Qed.

Theorem Perm1_4 : ∀ P Q : Prop,
  P ∨ Q → Q ∨ P. (*Permutation*)
Proof. intros P Q.
  apply or_comm.
Qed.

Theorem Assoc1_5 : ∀ P Q R : Prop,
  P ∨ (Q ∨ R) → Q ∨ (P ∨ R). (*Association*)
Proof. intros P Q R.
  specialize or_assoc with P Q R.
  intros or_assoc1.
  replace (P∨Q∨R) with ((P∨Q)∨R).
  specialize or_comm with P Q.
  intros or_comm1.
  replace (P∨Q) with (Q∨P).
```

```
  specialize or_assoc with Q P R.
  intros or_assoc2.
  replace ((Q∨P)∨R) with (Q∨P∨R).
  apply iff_refl.
  apply propositional_extensionality.
  apply iff_sym.
  apply or_assoc2.
  apply propositional_extensionality.
  apply or_comm.
  apply propositional_extensionality.
  apply or_assoc.
Qed.

Theorem Sum1_6 : ∀ P Q R : Prop,
  (Q → R) → (P ∨ Q → P ∨ R). (*Summation*)
Proof. intros P Q R.
  specialize imply_and_or2 with Q R P.
  intros imply_and_or2a.
  replace (P∨Q) with (Q∨P).
  replace (P∨R) with (R∨P).
  apply imply_and_or2a.
  apply propositional_extensionality.
  apply or_comm.
  apply propositional_extensionality.
  apply or_comm.
Qed.

(*These are all the propositional axioms of Principia.*)

Ltac MP H1 H2 :=
  match goal with
    | [ H1 : ?P → ?Q, H2 : ?P |- _ ] => specialize (H1 H2)
  end.
 (*We give this Ltac "MP" to make proofs more human-
 readable and to more closely mirror Principia's style.*)

End No1.

Module No2.

Import No1.

```

```
(*We proceed to the deductions of of Principia.*)

Theorem Abs2_01 : ∀ P : Prop,
  (P → ¬P) → ¬P.
Proof. intros P.
  specialize Taut1_2 with (¬P).
  intros Taut1_2.
  replace (¬P ∨ ¬P) with (P → ¬P) in Taut1_2.
  apply Taut1_2.
  apply Impl1_01.
Qed.

Theorem Simp2_02 : ∀ P Q : Prop,
  Q → (P → Q).
Proof. intros P Q.
  specialize Add1_3 with (¬P) Q.
  intros Add1_3.
  replace (¬P ∨ Q) with (P → Q) in Add1_3.
  apply Add1_3.
  apply Impl1_01.
Qed.

Theorem Transp2_03 : ∀ P Q : Prop,
  (P → ¬Q) → (Q → ¬P).
Proof. intros P Q.
  specialize Perm1_4 with (¬P) (¬Q).
  intros Perm1_4.
  replace (¬P ∨ ¬Q) with (P → ¬Q) in Perm1_4.
  replace (¬Q ∨ ¬P) with (Q → ¬P) in Perm1_4.
  apply Perm1_4.
  apply Impl1_01.
  apply Impl1_01.
Qed.

Theorem Comm2_04 : ∀ P Q R : Prop,
  (P → (Q → R)) → (Q → (P → R)).
Proof. intros P Q R.
  specialize Assoc1_5 with (¬P) (¬Q) R.
  intros Assoc1_5.
  replace (¬Q ∨ R) with (Q → R) in Assoc1_5.
  replace (¬P ∨ (Q → R)) with (P → (Q → R)) in Assoc1_5.
  replace (¬P ∨ R) with (P → R) in Assoc1_5.
```

```
  replace (¬Q ∨ (P → R)) with (Q → (P → R)) in Assoc1_5.
  apply Assoc1_5.
  apply Impl1_01.
  apply Impl1_01.
  apply Impl1_01.
  apply Impl1_01.
Qed.

Theorem Syll2_05 : ∀ P Q R : Prop,
   (Q → R) → ((P  → Q) → (P → R)).
Proof. intros P Q R.
  specialize Sum1_6 with (¬P) Q R.
  intros Sum1_6.
  replace (¬P ∨ Q) with (P → Q) in Sum1_6.
  replace (¬P ∨ R) with (P → R) in Sum1_6.
  apply Sum1_6.
  apply Impl1_01.
  apply Impl1_01.
Qed.

Theorem Syll2_06 : ∀ P Q R : Prop,
   (P → Q) → ((Q → R) → (P → R)).
Proof. intros P Q R.
  specialize Comm2_04 with (Q → R) (P → Q) (P → R).
  intros Comm2_04.
  specialize Syll2_05 with P Q R.
  intros Syll2_05.
  MP Comm2_04 Syll2_05.
  apply Comm2_04.
Qed.

Theorem n2_07 : ∀ P : Prop,
  P → (P ∨ P).
Proof. intros P.
  specialize Add1_3 with P P.
  intros Add1_3.
  apply Add1_3.
Qed.

Theorem Id2_08 : ∀ P : Prop,
  P → P.
Proof. intros P.
```

```
  specialize Syll2_05 with P (P ∨ P) P.
  intros Syll2_05.
  specialize Taut1_2 with P.
  intros Taut1_2.
  MP Syll2_05 Taut1_2.
  specialize n2_07 with P.
  intros n2_07.
  MP Syll2_05 n2_07.
  apply Syll2_05.
Qed.

Theorem n2_1 : ∀ P : Prop,
  (¬P) ∨ P.
Proof. intros P.
  specialize Id2_08 with P.
  intros Id2_08.
  replace (¬P ∨ P) with (P → P).
  apply Id2_08.
  apply Impl1_01.
Qed.

Theorem n2_11 : ∀ P : Prop,
  P ∨ ¬P.
Proof. intros P.
  specialize Perm1_4 with (¬P) P.
  intros Perm1_4.
  specialize n2_1 with P.
  intros n2_1.
  MP Perm1_4 n2_1.
  apply Perm1_4.
Qed.

Theorem n2_12 : ∀ P : Prop,
  P → ¬¬P.
Proof. intros P.
  specialize n2_11 with (¬P).
  intros n2_11.
  replace (¬P ∨ ¬¬P) with (P → ¬¬P) in n2_11.
  apply n2_11.
  apply Impl1_01.
Qed.
```

```
Theorem n2_13 : ∀ P : Prop,
  P ∨ ¬¬¬P.
Proof. intros P.
  specialize Sum1_6 with P (¬P) (¬¬¬P).
  intros Sum1_6.
  specialize n2_12 with (¬P).
  intros n2_12.
  MP Sum1_6 n2_12.
  specialize n2_11 with P.
  intros n2_11.
  MP Sum1_6 n2_11.
  apply Sum1_6.
Qed.

Theorem n2_14 : ∀ P : Prop,
  ¬¬P → P.
Proof. intros P.
  specialize Perm1_4 with P (¬¬¬P).
  intros Perm1_4.
  specialize n2_13 with P.
  intros n2_13.
  MP Perm1_4 n2_13.
  replace (¬¬¬P ∨ P) with (¬¬P → P) in Perm1_4.
  apply Perm1_4.
  apply Impl1_01.
Qed.

Theorem Transp2_15 : ∀ P Q : Prop,
  (¬P → Q) → (¬Q → P).
Proof. intros P Q.
  specialize Syll2_05 with (¬P) Q (¬¬Q).
  intros Syll2_05a.
  specialize n2_12 with Q.
  intros n2_12.
  MP Syll2_05a n2_12.
  specialize Transp2_03 with (¬P) (¬Q).
  intros Transp2_03.
  specialize Syll2_05 with (¬Q) (¬¬P) P.
  intros Syll2_05b.
  specialize n2_14 with P.
  intros n2_14.
  MP Syll2_05b n2_14.
```

```
  specialize Syll2_05 with (¬P → Q) (¬P → ¬¬Q) (¬Q → ¬¬P).
  intros Syll2_05c.
  MP Syll2_05c Transp2_03.
  MP Syll2_05c Syll2_05a.
  specialize Syll2_05 with (¬P → Q) (¬Q → ¬¬P) (¬Q → P).
  intros Syll2_05d.
  MP Syll2_05d Syll2_05b.
  MP Syll2_05d Syll2_05c.
  apply Syll2_05d.
Qed.

Ltac Syll H1 H2 S :=
  let S := fresh S in match goal with
    | [ H1 : ?P → ?Q, H2 : ?Q → ?R |- _ ] =>
        assert (S : P → R) by (intros p; apply (H2 (H1 p)))
end.

Theorem Transp2_16 : ∀ P Q : Prop,
  (P → Q) → (¬Q → ¬P).
Proof. intros P Q.
  specialize n2_12 with Q.
  intros n2_12a.
  specialize Syll2_05 with P Q (¬¬Q).
  intros Syll2_05a.
  specialize Transp2_03 with P (¬Q).
  intros Transp2_03a.
  MP n2_12a Syll2_05a.
  Syll Syll2_05a Transp2_03a S.
  apply S.
Qed.

Theorem Transp2_17 : ∀ P Q : Prop,
  (¬Q → ¬P) → (P → Q).
Proof. intros P Q.
  specialize Transp2_03 with (¬Q) P.
  intros Transp2_03a.
  specialize n2_14 with Q.
  intros n2_14a.
  specialize Syll2_05 with P (¬¬Q) Q.
  intros Syll2_05a.
  MP n2_14a Syll2_05a.
  Syll Transp2_03a Syll2_05a S.
```

```
  apply S.
Qed.

Theorem n2_18 : ∀ P : Prop,
  (¬P → P) → P.
Proof. intros P.
  specialize n2_12 with P.
  intro n2_12a.
  specialize Syll2_05 with (¬P) P (¬¬P).
  intro Syll2_05a.
  MP Syll2_05a n2_12.
  specialize Abs2_01 with (¬P).
  intros Abs2_01a.
  Syll Syll2_05a Abs2_01a Sa.
  specialize n2_14 with P.
  intros n2_14a.
  Syll H n2_14a Sb.
  apply Sb.
Qed.

Theorem n2_2 : ∀ P Q : Prop,
  P → (P ∨ Q).
Proof. intros P Q.
  specialize Add1_3 with Q P.
  intros Add1_3a.
  specialize Perm1_4 with Q P.
  intros Perm1_4a.
  Syll Add1_3a Perm1_4a S.
  apply S.
Qed.

Theorem n2_21 : ∀ P Q : Prop,
  ¬P → (P → Q).
Proof. intros P Q.
  specialize n2_2 with (¬P) Q.
  intros n2_2a.
  specialize Impl1_01 with P Q.
  intros Impl1_01a.
  replace (¬P∨Q) with (P→Q) in n2_2a.
  apply n2_2a.
Qed.
```

```
Theorem n2_24 : ∀ P Q : Prop,
  P → (¬P → Q).
Proof. intros P Q.
  specialize n2_21 with P Q.
  intros n2_21a.
  specialize Comm2_04 with (¬P) P Q.
  intros Comm2_04a.
  apply Comm2_04a.
  apply n2_21a.
Qed.

Theorem n2_25 : ∀ P Q : Prop,
  P ∨ ((P ∨ Q) → Q).
Proof. intros P Q.
  specialize n2_1 with (P ∨ Q).
  intros n2_1a.
  specialize Assoc1_5 with (¬(P∨Q)) P Q.
  intros Assoc1_5a.
  MP Assoc1_5a n2_1a.
  replace (¬(P∨Q)∨Q) with (P∨Q→Q) in Assoc1_5a.
  apply Assoc1_5a.
  apply Impl1_01.
Qed.

Theorem n2_26 : ∀ P Q : Prop,
  ¬P ∨ ((P → Q) → Q).
Proof. intros P Q.
  specialize n2_25 with (¬P) Q.
  intros n2_25a.
  replace (¬P∨Q) with (P→Q) in n2_25a.
  apply n2_25a.
  apply Impl1_01.
Qed.

Theorem n2_27 : ∀ P Q : Prop,
  P → ((P → Q) → Q).
Proof. intros P Q.
  specialize n2_26 with P Q.
  intros n2_26a.
  replace (¬P∨((P→Q)→Q)) with (P→(P→Q)→Q) in n2_26a.
  apply n2_26a.
  apply Impl1_01.
```

```
Qed.

Theorem n2_3 : ∀ P Q R : Prop,
  (P ∨ (Q ∨ R)) → (P ∨ (R ∨ Q)).
Proof. intros P Q R.
  specialize Perm1_4 with Q R.
  intros Perm1_4a.
  specialize Sum1_6 with P (Q∨R) (R∨Q).
  intros Sum1_6a.
  MP Sum1_6a Perm1_4a.
  apply Sum1_6a.
Qed.

Theorem n2_31 : ∀ P Q R : Prop,
  (P ∨ (Q ∨ R)) → ((P ∨ Q) ∨ R).
Proof. intros P Q R.
  specialize n2_3 with P Q R.
  intros n2_3a.
  specialize Assoc1_5 with P R Q.
  intros Assoc1_5a.
  specialize Perm1_4 with R (P∨Q).
  intros Perm1_4a.
  Syll Assoc1_5a Perm1_4a Sa.
  Syll n2_3a Sa Sb.
  apply Sb.
Qed.

Theorem n2_32 : ∀ P Q R : Prop,
  ((P ∨ Q) ∨ R) → (P ∨ (Q ∨ R)).
Proof. intros P Q R.
  specialize Perm1_4 with (P∨Q) R.
  intros Perm1_4a.
  specialize Assoc1_5 with R P Q.
  intros Assoc1_5a.
  specialize n2_3 with P R Q.
  intros n2_3a.
  specialize Syll2_06 with ((P∨Q)∨R) (R∨P∨Q) (P∨R∨Q).
  intros Syll2_06a.
  MP Syll2_06a Perm1_4a.
  MP Syll2_06a Assoc1_5a.
  specialize Syll2_06 with ((P∨Q)∨R) (P∨R∨Q) (P∨Q∨R).
  intros Syll2_06b.
```

```
  MP Syll2_06b Syll2_06a.
  MP Syll2_06b n2_3a.
  apply Syll2_06b.
Qed.

Theorem Abb2_33 : ∀ P Q R : Prop,
  (P ∨ Q ∨ R) = ((P ∨ Q) ∨ R).
Proof. intros P Q R.
  apply propositional_extensionality.
  split.
  specialize n2_31 with P Q R.
  intros n2_31.
  apply n2_31.
  specialize n2_32 with P Q R.
  intros n2_32.
  apply n2_32.
Qed.
  (*This definition makes the default left association.
      The default in Coq is right association.*)

Theorem n2_36 : ∀ P Q R : Prop,
  (Q → R) → ((P ∨ Q) → (R ∨ P)).
Proof. intros P Q R.
  specialize Perm1_4 with P R.
  intros Perm1_4a.
  specialize Syll2_05 with (P∨Q) (P∨R) (R∨P).
  intros Syll2_05a.
  MP Syll2_05a Perm1_4a.
  specialize Sum1_6 with P Q R.
  intros Sum1_6a.
  Syll Sum1_6a Syll2_05a S.
  apply S.
Qed.

Theorem n2_37 : ∀ P Q R : Prop,
  (Q → R) → ((Q ∨ P) → (P ∨ R)).
Proof. intros P Q R.
  specialize Perm1_4 with Q P.
  intros Perm1_4a.
  specialize Syll2_06 with (Q∨P) (P∨Q) (P∨R).
  intros Syll2_06a.
  MP Syll2_06a Perm1_4a.
```

```
  specialize Sum1_6 with P Q R.
  intros Sum1_6a.
  Syll Sum1_6a Syll2_06a S.
  apply S.
Qed.

Theorem n2_38 : ∀ P Q R : Prop,
  (Q → R) → ((Q ∨ P) → (R ∨ P)).
Proof. intros P Q R.
  specialize Perm1_4 with P R.
  intros Perm1_4a.
  specialize Syll2_05 with (Q∨P) (P∨R) (R∨P).
  intros Syll2_05a.
  MP Syll2_05a Perm1_4a.
  specialize Perm1_4 with Q P.
  intros Perm1_4b.
  specialize Syll2_06 with (Q∨P) (P∨Q) (P∨R).
  intros Syll2_06a.
  MP Syll2_06a Perm1_4b.
  Syll Syll2_06a Syll2_05a H.
  specialize Sum1_6 with P Q R.
  intros Sum1_6a.
  Syll Sum1_6a H S.
  apply S.
Qed.

Theorem n2_4 : ∀ P Q : Prop,
  (P ∨ (P ∨ Q)) → (P ∨ Q).
Proof. intros P Q.
  specialize n2_31 with P P Q.
  intros n2_31a.
  specialize Taut1_2 with P.
  intros Taut1_2a.
  specialize n2_38 with Q (P∨P) P.
  intros n2_38a.
  MP n2_38a Taut1_2a.
  Syll n2_31a n2_38a S.
  apply S.
Qed.

Theorem n2_41 : ∀ P Q : Prop,
  (Q ∨ (P ∨ Q)) → (P ∨ Q).
```

```
Proof. intros P Q.
  specialize Assoc1_5 with Q P Q.
  intros Assoc1_5a.
  specialize Taut1_2 with Q.
  intros Taut1_2a.
  specialize Sum1_6 with P (Q∨Q) Q.
  intros Sum1_6a.
  MP Sum1_6a Taut1_2a.
  Syll Assoc1_5a Sum1_6a S.
  apply S.
Qed.

Theorem n2_42 : ∀ P Q : Prop,
  (¬P ∨ (P → Q)) → (P → Q).
Proof. intros P Q.
  specialize n2_4 with (¬P) Q.
  intros n2_4a.
  replace (¬P∨Q) with (P→Q) in n2_4a.
  apply n2_4a. apply Impl1_01.
Qed.

Theorem n2_43 : ∀ P Q : Prop,
  (P → (P → Q)) → (P → Q).
Proof. intros P Q.
  specialize n2_42 with P Q.
  intros n2_42a.
  replace (¬P ∨ (P→Q)) with (P→(P→Q)) in n2_42a.
  apply n2_42a.
  apply Impl1_01.
Qed.

Theorem n2_45 : ∀ P Q : Prop,
  ¬(P ∨ Q) → ¬P.
Proof. intros P Q.
  specialize n2_2 with P Q.
  intros n2_2a.
  specialize Transp2_16 with P (P∨Q).
  intros Transp2_16a.
  MP n2_2 Transp2_16a.
  apply Transp2_16a.
Qed.
```

```
Theorem n2_46 : ∀ P Q : Prop,
  ¬(P ∨ Q) → ¬Q.
Proof. intros P Q.
  specialize Add1_3 with P Q.
  intros Add1_3a.
  specialize Transp2_16 with Q (P∨Q).
  intros Transp2_16a.
  MP Add1_3a Transp2_16a.
  apply Transp2_16a.
Qed.

Theorem n2_47 : ∀ P Q : Prop,
  ¬(P ∨ Q) → (¬P ∨ Q).
Proof. intros P Q.
  specialize n2_45 with P Q.
  intros n2_45a.
  specialize n2_2 with (¬P) Q.
  intros n2_2a.
  Syll n2_45a n2_2a S.
  apply S.
Qed.

Theorem n2_48 : ∀ P Q : Prop,
  ¬(P ∨ Q) → (P ∨ ¬Q).
Proof. intros P Q.
  specialize n2_46 with P Q.
  intros n2_46a.
  specialize Add1_3 with P (¬Q).
  intros Add1_3a.
  Syll n2_46a Add1_3a S.
  apply S.
Qed.

Theorem n2_49 : ∀ P Q : Prop,
  ¬(P ∨ Q) → (¬P ∨ ¬Q).
Proof. intros P Q.
  specialize n2_45 with P Q.
  intros n2_45a.
  specialize n2_2 with (¬P) (¬Q).
  intros n2_2a.
  Syll n2_45a n2_2a S.
  apply S.
```

```
Qed.

Theorem n2_5 : ∀ P Q : Prop,
  ¬(P → Q) → (¬P → Q).
Proof. intros P Q.
  specialize n2_47 with (¬P) Q.
  intros n2_47a.
  replace (¬P∨Q) with (P→Q) in n2_47a.
  replace (¬¬P∨Q) with (¬P→Q) in n2_47a.
  apply n2_47a.
  apply Impl1_01.
  apply Impl1_01.
Qed.

Theorem n2_51 : ∀ P Q : Prop,
  ¬(P → Q) → (P → ¬Q).
Proof. intros P Q.
  specialize n2_48 with (¬P) Q.
  intros n2_48a.
  replace (¬P∨Q) with (P→Q) in n2_48a.
  replace (¬P∨¬Q) with (P→¬Q) in n2_48a.
  apply n2_48a.
  apply Impl1_01.
  apply Impl1_01.
Qed.

Theorem n2_52 : ∀ P Q : Prop,
  ¬(P → Q) → (¬P → ¬Q).
Proof. intros P Q.
  specialize n2_49 with (¬P) Q.
  intros n2_49a.
  replace (¬P∨Q) with (P→Q) in n2_49a.
  replace (¬¬P∨¬Q) with (¬P→¬Q) in n2_49a.
  apply n2_49a.
  apply Impl1_01.
  apply Impl1_01.
Qed.

Theorem n2_521 : ∀ P Q : Prop,
  ¬(P→Q)→(Q→P).
Proof. intros P Q.
  specialize n2_52 with P Q.
```

```
  intros n2_52a.
  specialize Transp2_17 with Q P.
  intros Transp2_17a.
  Syll n2_52a Transp2_17a S.
  apply S.
Qed.

Theorem n2_53 : ∀ P Q : Prop,
  (P ∨ Q) → (¬P → Q).
Proof. intros P Q.
  specialize n2_12 with P.
  intros n2_12a.
  specialize n2_38 with Q P (¬¬P).
  intros n2_38a.
  MP n2_38a n2_12a.
  replace (¬¬P∨Q) with (¬P→Q) in n2_38a.
  apply n2_38a.
  apply Impl1_01.
Qed.

Theorem n2_54 : ∀ P Q : Prop,
  (¬P → Q) → (P ∨ Q).
Proof. intros P Q.
  specialize n2_14 with P.
  intros n2_14a.
  specialize n2_38 with Q (¬¬P) P.
  intros n2_38a.
  MP n2_38a n2_12a.
  replace (¬¬P∨Q) with (¬P→Q) in n2_38a.
  apply n2_38a.
  apply Impl1_01.
Qed.

Theorem n2_55 : ∀ P Q : Prop,
  ¬P → ((P ∨ Q) → Q).
Proof. intros P Q.
  specialize n2_53 with P Q.
  intros n2_53a.
  specialize Comm2_04 with (P∨Q) (¬P) Q.
  intros Comm2_04a.
  MP n2_53a Comm2_04a.
  apply Comm2_04a.
```

```
Qed.

Theorem n2_56 : ∀ P Q : Prop,
  ¬Q → ((P ∨ Q) → P).
Proof. intros P Q.
  specialize n2_55 with Q P.
  intros n2_55a.
  specialize Perm1_4 with P Q.
  intros Perm1_4a.
  specialize Syll2_06 with (P∨Q) (Q∨P) P.
  intros Syll2_06a.
  MP Syll2_06a Perm1_4a.
  Syll n2_55a Syll2_06a Sa.
  apply Sa.
Qed.

Theorem n2_6 : ∀ P Q : Prop,
  (¬P→Q) → ((P → Q) → Q).
Proof. intros P Q.
  specialize n2_38 with Q (¬P) Q.
  intros n2_38a.
  specialize Taut1_2 with Q.
  intros Taut1_2a.
  specialize Syll2_05 with (¬P∨Q) (Q∨Q) Q.
  intros Syll2_05a.
  MP Syll2_05a Taut1_2a.
  Syll n2_38a Syll2_05a S.
  replace (¬P∨Q) with (P→Q) in S.
  apply S.
  apply Impl1_01.
Qed.

Theorem n2_61 : ∀ P Q : Prop,
  (P → Q) → ((¬P → Q) → Q).
Proof. intros P Q.
  specialize n2_6 with P Q.
  intros n2_6a.
  specialize Comm2_04 with (¬P→Q) (P→Q) Q.
  intros Comm2_04a.
  MP Comm2_04a n2_6a.
  apply Comm2_04a.
Qed.
```

```
Theorem n2_62 : ∀ P Q : Prop,
  (P ∨ Q) → ((P → Q) → Q).
Proof. intros P Q.
  specialize n2_53 with P Q.
  intros n2_53a.
  specialize n2_6 with P Q.
  intros n2_6a.
  Syll n2_53a n2_6a S.
  apply S.
Qed.

Theorem n2_621 : ∀ P Q : Prop,
  (P → Q) → ((P ∨ Q) → Q).
Proof. intros P Q.
  specialize n2_62 with P Q.
  intros n2_62a.
  specialize Comm2_04 with (P ∨ Q) (P→Q) Q.
  intros Comm2_04a.
  MP Comm2_04a n2_62a.
  apply Comm2_04a.
Qed.

Theorem n2_63 : ∀ P Q : Prop,
  (P ∨ Q) → ((¬P ∨ Q) → Q).
Proof. intros P Q.
  specialize n2_62 with P Q.
  intros n2_62a.
  replace (¬P∨Q) with (P→Q).
  apply n2_62a.
  apply Impl1_01.
Qed.

Theorem n2_64 : ∀ P Q : Prop,
  (P ∨ Q) → ((P ∨ ¬Q) → P).
Proof. intros P Q.
  specialize n2_63 with Q P.
  intros n2_63a.
  specialize Perm1_4 with P Q.
  intros Perm1_4a.
  Syll n2_63a Perm1_4a Ha.
  specialize Syll2_06 with (P∨¬Q) (¬Q∨P) P.
```

```
  intros Syll2_06a.
  specialize Perm1_4 with P (¬Q).
  intros Perm1_4b.
  MP Syll2_06a Perm1_4b.
  Syll Syll2_06a Ha S.
  apply S.
Qed.

Theorem n2_65 : ∀ P Q : Prop,
  (P → Q) → ((P → ¬Q) → ¬P).
Proof. intros P Q.
  specialize n2_64 with (¬P) Q.
  intros n2_64a.
  replace (¬P∨Q) with (P→Q) in n2_64a.
  replace (¬P∨¬Q) with (P→¬Q) in n2_64a.
  apply n2_64a.
  apply Impl1_01.
  apply Impl1_01.
Qed.

Theorem n2_67 : ∀ P Q : Prop,
  ((P ∨ Q) → Q) → (P → Q).
Proof. intros P Q.
  specialize n2_54 with P Q.
  intros n2_54a.
  specialize Syll2_06 with (¬P→Q) (P∨Q) Q.
  intros Syll2_06a.
  MP Syll2_06a n2_54a.
  specialize n2_24 with  P Q.
  intros n2_24.
  specialize Syll2_06 with P (¬P→Q) Q.
  intros Syll2_06b.
  MP Syll2_06b n2_24a.
  Syll Syll2_06b Syll2_06a S.
  apply S.
Qed.

Theorem n2_68 : ∀ P Q : Prop,
  ((P → Q) → Q) → (P ∨ Q).
Proof. intros P Q.
  specialize n2_67 with (¬P) Q.
  intros n2_67a.
```

```
  replace (¬P∨Q) with (P→Q) in n2_67a.
  specialize n2_54 with P Q.
  intros n2_54a.
  Syll n2_67a n2_54a S.
  apply S.
  apply Impl1_01.
Qed.

Theorem n2_69 : ∀ P Q : Prop,
  ((P → Q) → Q) → ((Q → P) → P).
Proof. intros P Q.
  specialize n2_68 with P Q.
  intros n2_68a.
  specialize Perm1_4 with P Q.
  intros Perm1_4a.
  Syll n2_68a Perm1_4a Sa.
  specialize n2_62 with Q P.
  intros n2_62a.
  Syll Sa n2_62a Sb.
  apply Sb.
Qed.

Theorem n2_73 : ∀ P Q R : Prop,
  (P → Q) → (((P ∨ Q) ∨ R) → (Q ∨ R)).
Proof. intros P Q R.
  specialize n2_621 with P Q.
  intros n2_621a.
  specialize n2_38 with R (P∨Q) Q.
  intros n2_38a.
  Syll n2_621a n2_38a S.
  apply S.
Qed.

Theorem n2_74 : ∀ P Q R : Prop,
  (Q → P) → ((P ∨ Q) ∨ R) → (P ∨ R).
Proof. intros P Q R.
  specialize n2_73 with Q P R.
  intros n2_73a.
  specialize Assoc1_5 with P Q R.
  intros Assoc1_5a.
  specialize n2_31 with Q P R.
  intros n2_31a. (*not cited*)
```

```
  Syll Assoc1_5a n2_31a Sa.
  specialize n2_32 with P Q R.
  intros n2_32a. (*not cited*)
  Syll n2_32a Sa Sb.
  specialize Syll2_06 with ((P∨Q)∨R) ((Q∨P)∨R) (P∨R).
  intros Syll2_06a.
  MP Syll2_06a Sb.
  Syll n2_73a Syll2_05a H.
  apply H.
Qed.

Theorem n2_75 : ∀ P Q R : Prop,
  (P ∨ Q) → ((P ∨ (Q → R)) → (P ∨ R)).
Proof. intros P Q R.
  specialize n2_74 with P (¬Q) R.
  intros n2_74a.
  specialize n2_53 with Q P.
  intros n2_53a.
  Syll n2_53a n2_74a Sa.
  specialize n2_31 with P (¬Q) R.
  intros n2_31a.
  specialize Syll2_06 with (P∨(¬Q)∨R)((P∨(¬Q))∨R) (P∨R).
  intros Syll2_06a.
  MP Syll2_06a n2_31a.
  Syll Sa Syll2_06a Sb.
  specialize Perm1_4 with P Q.
  intros Perm1_4a. (*not cited*)
  Syll Perm1_4a Sb Sc.
  replace (¬Q∨R) with (Q→R) in Sc.
  apply Sc.
  apply Impl1_01.
Qed.

Theorem n2_76 : ∀ P Q R : Prop,
  (P ∨ (Q → R)) → ((P ∨ Q) → (P ∨ R)).
Proof. intros P Q R.
  specialize n2_75 with P Q R.
  intros n2_75a.
  specialize Comm2_04 with (P∨Q) (P∨(Q→R)) (P∨R).
  intros Comm2_04a.
  apply Comm2_04a.
  apply n2_75a.
```

```
Qed.

Theorem n2_77 : ∀ P Q R : Prop,
  (P → (Q → R)) → ((P → Q) → (P → R)).
Proof. intros P Q R.
  specialize n2_76 with (¬P) Q R.
  intros n2_76a.
  replace (¬P∨(Q→R)) with (P→Q→R) in n2_76a.
  replace (¬P∨Q) with (P→Q) in n2_76a.
  replace (¬P∨R) with (P→R) in n2_76a.
  apply n2_76a.
  apply Impl1_01.
  apply Impl1_01.
  apply Impl1_01.
Qed.

Theorem n2_8 : ∀ Q R S : Prop,
  (Q ∨ R) → ((¬R ∨ S) → (Q ∨ S)).
Proof. intros Q R S.
  specialize n2_53 with R Q.
  intros n2_53a.
  specialize Perm1_4 with Q R.
  intros Perm1_4a.
  Syll Perm1_4a n2_53a Ha.
  specialize n2_38 with S (¬R) Q.
  intros n2_38a.
  Syll H n2_38a Hb.
  apply Hb.
Qed.

Theorem n2_81 : ∀ P Q R S : Prop,
  (Q → (R → S)) → ((P ∨ Q) → ((P ∨ R) → (P ∨ S))).
Proof. intros P Q R S.
  specialize Sum1_6 with P Q (R→S).
  intros Sum1_6a.
  specialize n2_76 with P R S.
  intros n2_76a.
  specialize Syll2_05 with (P∨Q) (P∨(R→S)) ((P∨R)→(P∨S)).
  intros Syll2_05a.
  MP Syll2_05a n2_76a.
  Syll Sum1_6a Syll2_05a H.
  apply H.
```

```
Qed.

Theorem n2_82 : ∀ P Q R S : Prop,
  (P ∨ Q ∨ R)→((P ∨ ¬R ∨ S)→(P ∨ Q ∨ S)).
Proof. intros P Q R S.
  specialize n2_8 with Q R S.
  intros n2_8a.
  specialize n2_81 with P (Q∨R) (¬R∨S) (Q∨S).
  intros n2_81a.
  MP n2_81a n2_8a.
  apply n2_81a.
Qed.

Theorem n2_83 : ∀ P Q R S : Prop,
  (P→(Q→R))→((P→(R→S))→(P→(Q→S))).
Proof. intros P Q R S.
  specialize n2_82 with (¬P) (¬Q) R S.
  intros n2_82a.
  replace (¬Q∨R) with (Q→R) in n2_82a.
  replace (¬P∨(Q→R)) with (P→Q→R) in n2_82a.
  replace (¬R∨S) with (R→S) in n2_82a.
  replace (¬P∨(R→S)) with (P→R→S) in n2_82a.
  replace (¬Q∨S) with (Q→S) in n2_82a.
  replace (¬Q∨S) with (Q→S) in n2_82a.
  replace (¬P∨(Q→S)) with (P→Q→S) in n2_82a.
  apply n2_82a.
  apply Impl1_01.
  apply Impl1_01.
  apply Impl1_01.
  apply Impl1_01.
  apply Impl1_01.
  apply Impl1_01.
  apply Impl1_01.
Qed.

Theorem n2_85 : ∀ P Q R : Prop,
  ((P ∨ Q) → (P ∨ R)) → (P ∨ (Q → R)).
Proof. intros P Q R.
  specialize Add1_3 with P Q.
  intros Add1_3a.
  specialize Syll2_06 with Q (P∨Q) R.
  intros Syll2_06a.
```

```
  MP Syll2_06a Add1_3a.
  specialize n2_55 with P R.
  intros n2_55a.
  specialize Syll2_05 with (P∨Q) (P∨R) R.
  intros Syll2_05a.
  Syll n2_55a Syll2_05a Ha.
  specialize n2_83 with (¬P) ((P∨Q)→(P∨R)) ((P∨Q)→R) (Q→R).
  intros n2_83a.
  MP n2_83a Ha.
  specialize Comm2_04 with (¬P) (P∨Q→P∨R) (Q→R).
  intros Comm2_04a.
  Syll Ha Comm2_04a Hb.
  specialize n2_54 with P (Q→R).
  intros n2_54a.
  specialize Simp2_02 with (¬P) ((P∨Q→R)→(Q→R)).
  intros Simp2_02a. (*Not cited*)
      (*Greg's suggestion per the BRS list on June 25, 2017.*)
  MP Syll2_06a Simp2_02a.
  MP Hb Simp2_02a.
  Syll Hb n2_54a Hc.
  apply Hc.
Qed.

Theorem n2_86 : ∀ P Q R : Prop,
  ((P → Q) → (P → R)) → (P → (Q →  R)).
Proof. intros P Q R.
  specialize n2_85 with (¬P) Q R.
  intros n2_85a.
  replace (¬P∨Q) with (P→Q) in n2_85a.
  replace (¬P∨R) with (P→R) in n2_85a.
  replace (¬P∨(Q→R)) with (P→Q→R) in n2_85a.
  apply n2_85a.
  apply Impl1_01.
  apply Impl1_01.
  apply Impl1_01.
Qed.

End No2.

Module No3.

Import No1.
```

```
Import No2.


Theorem Prod3_01 : ∀ P Q : Prop,
  (P ∧ Q) = (¬(¬P ∨ ¬Q)).
Proof. intros P Q.
  apply propositional_extensionality.
  split.
  specialize or_not_and with (P) (Q).
  intros or_not_and.
  specialize Transp2_03 with (¬P ∨ ¬Q) (P ∧ Q).
  intros Transp2_03.
  MP Transp2_03 or_not_and.
  apply Transp2_03.
  specialize not_and_or with (P) (Q).
  intros not_and_or.
  specialize Transp2_15 with (P ∧ Q) (¬P ∨ ¬Q).
  intros Transp2_15.
  MP Transp2_15 not_and_or.
  apply Transp2_15.
Qed.
(*This is a notational definition in Principia;
  it is used to switch between "∧" and "¬∨¬".*)

(*Axiom Abb3_02 : ∀ P Q R : Prop,
  (P → Q → R) = ((P → Q) ∧ (Q → R)).*)
  (*Since Coq forbids such strings as ill-formed, or
  else automatically associates to the right,
  we leave this notational axiom commented out.*)

Theorem Conj3_03 : ∀ P Q : Prop, P → Q → (P∧Q).
Proof. intros P Q.
  specialize n2_11 with (¬P∨¬Q). intros n2_11a.
  specialize n2_32 with (¬P) (¬Q) (¬(¬P ∨ ¬Q)). intros n2_32a.
  MP n2_32a n2_11a.
  replace (¬(¬P∨¬Q)) with (P∧Q) in n2_32a.
  replace (¬Q ∨ (P∧Q)) with (Q→(P∧Q)) in n2_32a.
  replace (¬P ∨ (Q → (P∧Q))) with (P→Q→(P∧Q)) in n2_32a.
  apply n2_32a.
  apply Impl1_01.
  apply Impl1_01.
  apply Prod3_01.
```

```
Qed.
(*3.03 is permits the inference from the theoremhood
    of P and that of Q to the theoremhood of P and Q. So:*)

Ltac Conj H1 H2 :=
  match goal with
    | [ H1 : ?P, H2 : ?Q |- _ ] =>
      assert (P ∧ Q)
end.

Theorem n3_1 : ∀ P Q : Prop,
  (P ∧ Q) → ¬(¬P ∨ ¬Q).
Proof. intros P Q.
  replace (¬(¬P∨¬Q)) with (P∧Q).
  specialize Id2_08 with (P∧Q).
  intros Id2_08a.
  apply Id2_08a.
  apply Prod3_01.
Qed.

Theorem n3_11 : ∀ P Q : Prop,
  ¬(¬P ∨ ¬Q) → (P ∧ Q).
Proof. intros P Q.
  replace (¬(¬P∨¬Q)) with (P∧Q).
  specialize Id2_08 with (P∧Q).
  intros Id2_08a.
  apply Id2_08a.
  apply Prod3_01.
Qed.

Theorem n3_12 : ∀ P Q : Prop,
  (¬P ∨ ¬Q) ∨ (P ∧ Q).
Proof. intros P Q.
  specialize n2_11 with (¬P∨¬Q).
  intros n2_11a.
  replace (¬(¬P∨¬Q)) with (P∧Q) in n2_11a.
  apply n2_11a.
  apply Prod3_01.
Qed.

Theorem n3_13 : ∀ P Q : Prop,
  ¬(P ∧ Q) → (¬P ∨ ¬Q).
```

```
Proof. intros P Q.
  specialize n3_11 with P Q.
  intros n3_11a.
  specialize Transp2_15 with (¬P∨¬Q) (P∧Q).
  intros Transp2_15a.
  MP Transp2_15a n3_11a.
  apply Transp2_15a.
Qed.

Theorem n3_14 : ∀ P Q : Prop,
  (¬P ∨ ¬Q) → ¬(P ∧ Q).
Proof. intros P Q.
  specialize n3_1 with P Q.
  intros n3_1a.
  specialize Transp2_16 with (P∧Q) (¬(¬P∨¬Q)).
  intros Transp2_16a.
  MP Transp2_16a n3_1a.
  specialize n2_12 with (¬P∨¬Q).
  intros n2_12a.
  Syll n2_12a Transp2_16a S.
  apply S.
Qed.

Theorem n3_2 : ∀ P Q : Prop,
  P → Q → (P ∧ Q).
Proof. intros P Q.
  specialize n3_12 with P Q.
  intros n3_12a.
  specialize n2_32 with (¬P) (¬Q) (P∧Q).
  intros n2_32a.
  MP n3_32a n3_12a.
  replace (¬Q ∨ P ∧ Q) with (Q→P∧Q) in n2_32a.
  replace (¬P ∨ (Q → P ∧ Q)) with (P→Q→P∧Q) in n2_32a.
  apply n2_32a.
  apply Impl1_01.
  apply Impl1_01.
Qed.

Theorem n3_21 : ∀ P Q : Prop,
  Q → P → (P ∧ Q).
Proof. intros P Q.
  specialize n3_2 with P Q.
```

```
  intros n3_2a.
  specialize Comm2_04 with P Q (P∧Q).
  intros Comm2_04a.
  MP Comm2_04a n3_2a.
  apply Comm2_04a.
Qed.

Theorem n3_22 : ∀ P Q : Prop,
  (P ∧ Q) → (Q ∧ P).
Proof. intros P Q.
  specialize n3_13 with Q P.
  intros n3_13a.
  specialize Perm1_4 with (¬Q) (¬P).
  intros Perm1_4a.
  Syll n3_13a Perm1_4a Ha.
  specialize n3_14 with P Q.
  intros n3_14a.
  Syll Ha n3_14a Hb.
  specialize Transp2_17 with (P∧Q) (Q ∧ P).
  intros Transp2_17a.
  MP Transp2_17a Hb.
  apply Transp2_17a.
Qed.

Theorem n3_24 : ∀ P : Prop,
  ¬(P ∧ ¬P).
Proof. intros P.
  specialize n2_11 with (¬P).
  intros n2_11a.
  specialize n3_14 with P (¬P).
  intros n3_14a.
  MP n3_14a n2_11a.
  apply n3_14a.
Qed.

Theorem Simp3_26 : ∀ P Q : Prop,
  (P ∧ Q) → P.
Proof. intros P Q.
  specialize Simp2_02 with Q P.
  intros Simp2_02a.
  replace (P→(Q→P)) with (¬P∨(Q→P)) in Simp2_02a.
  replace (Q→P) with (¬Q∨P) in Simp2_02a.
```

```
  specialize n2_31 with (¬P) (¬Q) P.
  intros n2_31a.
  MP n2_31a Simp2_02a.
  specialize n2_53 with (¬P∨¬Q) P.
  intros n2_53a.
  MP n2_53a Simp2_02a.
  replace (¬(¬P∨¬Q)) with (P∧Q) in n2_53a.
  apply n2_53a.
  apply Prod3_01.
  rewrite <- Impl1_01.
  reflexivity.
  rewrite <- Impl1_01.
  reflexivity.
Qed.

Theorem Simp3_27 : ∀ P Q : Prop,
  (P ∧ Q) → Q.
Proof. intros P Q.
  specialize n3_22 with P Q.
  intros n3_22a.
  specialize Simp3_26 with Q P.
  intros Simp3_26a.
  Syll n3_22a Simp3_26a S.
  apply S.
Qed.

Theorem Exp3_3 : ∀ P Q R : Prop,
  ((P ∧ Q) → R) → (P → (Q → R)).
Proof. intros P Q R.
  specialize Id2_08 with ((P∧Q)→R).
  intros Id2_08a. (*This theorem isn't needed.*)
  replace (((P ∧ Q) → R) → ((P ∧ Q) → R)) with
    (((P ∧ Q) → R) → (¬(¬P ∨ ¬Q) → R)) in Id2_08a.
  specialize Transp2_15 with (¬P∨¬Q) R.
  intros Transp2_15a.
  Syll Id2_08a Transp2_15a Sa.
  specialize Id2_08 with (¬R → (¬P ∨ ¬Q)).
  intros Id2_08b. (*This theorem isn't needed.*)
  Syll Sa Id2_08b Sb.
  replace (¬P ∨ ¬Q) with (P → ¬Q) in Sb.
  specialize Comm2_04 with (¬R) P (¬Q).
  intros Comm2_04a.
```

```
  Syll Sb Comm2_04a Sc.
  specialize Transp2_17 with Q R.
  intros Transp2_17a.
  specialize Syll2_05 with P (¬R → ¬Q) (Q → R).
  intros Syll2_05a.
  MP Syll2_05a Transp2_17a.
  Syll Sa Syll2_05a Sd.
  apply Sd.
  rewrite <- Impl1_01.
  reflexivity.
  rewrite Prod3_01.
  reflexivity.
Qed.

Theorem Imp3_31 : ∀ P Q R : Prop,
  (P → (Q → R)) → (P ∧ Q) → R.
Proof. intros P Q R.
  specialize Id2_08 with (P → (Q → R)).
  intros Id2_08a. (*This use of Id2_08 is redundant.*)
  replace ((P → (Q → R))→(P → (Q → R))) with
    ((P → (Q → R))→(¬P ∨ (Q → R))) in Id2_08a.
  replace (¬P ∨ (Q → R)) with
    (¬P ∨ (¬Q ∨ R)) in Id2_08a.
  specialize n2_31 with (¬P) (¬Q) R.
  intros n2_31a.
  Syll Id2_08a n2_31a Sa.
  specialize n2_53 with (¬P∨¬Q) R.
  intros n2_53a.
  replace (¬(¬P∨¬Q)) with (P∧Q) in n2_53a.
  Syll n2_31a n2_53a Sb.
  apply Sb.
  apply Prod3_01.
  rewrite Impl1_01.
  reflexivity.
  rewrite <- Impl1_01.
  reflexivity.
Qed.
(*The proof sketch cites Id2_08, but
    we did not seem to need it.*)

Theorem Syll3_33 : ∀ P Q R : Prop,
  ((P → Q) ∧ (Q → R)) → (P → R).
```

```
Proof. intros P Q R.
  specialize Syll2_06 with P Q R.
  intros Syll2_06a.
  specialize Imp3_31 with (P→Q) (Q→R) (P→R).
  intros Imp3_31a.
  MP Imp3_31a Syll2_06a.
  apply Imp3_31a.
Qed.

Theorem Syll3_34 : ∀ P Q R : Prop,
  ((Q → R) ∧ (P → Q)) → (P → R).
Proof. intros P Q R.
  specialize Syll2_05 with P Q R.
  intros Syll2_05a.
  specialize Imp3_31 with (Q→R) (P→Q) (P→R).
  intros Imp3_31a.
  MP Imp3_31a Syll2_05a.
  apply Imp3_31a.
Qed.

Theorem Ass3_35 : ∀ P Q : Prop,
  (P ∧ (P → Q)) → Q.
Proof. intros P Q.
  specialize n2_27 with P Q.
  intros n2_27a.
  specialize Imp3_31 with P (P→Q) Q.
  intros Imp3_31a.
  MP Imp3_31a n2_27a.
  apply Imp3_31a.
Qed.

Theorem Transp3_37 : ∀ P Q R : Prop,
  (P ∧ Q → R) → (P ∧ ¬R → ¬Q).
Proof. intros P Q R.
  specialize Transp2_16 with Q R.
  intros Transp2_16a.
  specialize Syll2_05 with P (Q→R) (¬R→¬Q).
  intros Syll2_05a.
  MP Syll2_05a Transp2_16a.
  specialize Exp3_3 with P Q R.
  intros Exp3_3a.
  Syll Exp3_3a Syll2_05a Sa.
```

```
  specialize Imp3_31 with P (¬R) (¬Q).
  intros Imp3_31a.
  Syll Sa Imp3_31a Sb.
  apply Sb.
Qed.

Theorem n3_4 : ∀ P Q : Prop,
  (P ∧ Q) → P → Q.
Proof. intros P Q.
  specialize n2_51 with P Q.
  intros n2_51a.
  specialize Transp2_15 with (P→Q) (P→¬Q).
  intros Transp2_15a.
  MP Transp2_15a n2_51a.
  replace (P→¬Q) with (¬P∨¬Q) in Transp2_15a.
  replace (¬(¬P∨¬Q)) with (P∧Q) in Transp2_15a.
  apply Transp2_15a.
  apply Prod3_01.
  rewrite <- Impl1_01.
  reflexivity.
Qed.

Theorem n3_41 : ∀ P Q R : Prop,
  (P → R) → (P ∧ Q → R).
Proof. intros P Q R.
  specialize Simp3_26 with P Q.
  intros Simp3_26a.
  specialize Syll2_06 with (P∧Q) P R.
  intros Syll2_06a.
  MP Simp3_26a Syll2_06a.
  apply Syll2_06a.
Qed.

Theorem n3_42 : ∀ P Q R : Prop,
  (Q → R) → (P ∧ Q → R).
Proof. intros P Q R.
  specialize Simp3_27 with P Q.
  intros Simp3_27a.
  specialize Syll2_06 with (P∧Q) Q R.
  intros Syll2_06a.
  MP Syll2_06a Simp3_27a.
  apply Syll2_06a.
```

```
Qed.

Theorem Comp3_43 : ∀ P Q R : Prop,
  (P → Q) ∧ (P → R) → (P → Q ∧ R).
Proof. intros P Q R.
  specialize n3_2 with Q R.
  intros n3_2a.
  specialize Syll2_05 with P Q (R→Q∧R).
  intros Syll2_05a.
  MP Syll2_05a n3_2a.
  specialize n2_77 with P R (Q∧R).
  intros n2_77a.
  Syll Syll2_05a n2_77a Sa.
  specialize Imp3_31 with (P→Q) (P→R) (P→Q∧R).
  intros Imp3_31a.
  MP Sa Imp3_31a.
  apply Imp3_31a.
Qed.

Theorem n3_44 : ∀ P Q R : Prop,
  (Q → P) ∧ (R → P) → (Q ∨ R → P).
Proof. intros P Q R.
  specialize Syll3_33 with (¬Q) R P.
  intros Syll3_33a.
  specialize n2_6 with Q P.
  intros n2_6a.
  Syll Syll3_33a n2_6a Sa.
  specialize Exp3_3 with (¬Q→R) (R→P) ((Q→P)→P).
  intros Exp3_3a.
  MP Exp3_3a Sa.
  specialize Comm2_04 with (R→P) (Q→P) P.
  intros Comm2_04a.
  Syll Exp3_3a Comm2_04a Sb.
  specialize Imp3_31 with (Q→P) (R→P) P.
  intros Imp3_31a.
  Syll Sb Imp3_31a Sc.
  specialize Comm2_04 with (¬Q→R) ((Q→P)∧(R→P)) P.
  intros Comm2_04b.
  MP Comm2_04b Sc.
  specialize n2_53 with Q R.
  intros n2_53a.
  specialize Syll2_06 with (Q∨R) (¬Q→R) P.
```

```
  intros Syll2_06a.
  MP Syll2_06a n2_53a.
  Syll Comm2_04b Syll2_06a Sd.
  apply Sd.
Qed.

Theorem Fact3_45 : ∀ P Q R : Prop,
  (P → Q) → (P ∧ R) → (Q ∧ R).
Proof. intros P Q R.
  specialize Syll2_06 with P Q (¬R).
  intros Syll2_06a.
  specialize Transp2_16 with (Q→¬R) (P→¬R).
  intros Transp2_16a.
  Syll Syll2_06a Transp2_16a Sa.
  specialize Id2_08 with (¬(P→R)→¬(Q→¬R)).
  intros Id2_08a.
  Syll Sa Id2_08a Sb.
  replace (P→¬R) with (¬P∨¬R) in Sb.
  replace (Q→¬R) with (¬Q∨¬R) in Sb.
  replace (¬(¬P∨¬R)) with (P∧R) in Sb.
  replace (¬(¬Q∨¬R)) with (Q∧R) in Sb.
  apply Sb.
  apply Prod3_01.
  apply Prod3_01.
  rewrite <- Impl1_01.
  reflexivity.
  rewrite <- Impl1_01.
  reflexivity.
Qed.

Theorem n3_47 : ∀ P Q R S : Prop,
  ((P → R) ∧ (Q → S)) → (P ∧ Q) → R ∧ S.
Proof. intros P Q R S.
  specialize Simp3_26 with (P→R) (Q→S).
  intros Simp3_26a.
  specialize Fact3_45 with P R Q.
  intros Fact3_45a.
  Syll Simp3_26a Fact3_45a Sa.
  specialize n3_22 with R Q.
  intros n3_22a.
  specialize Syll2_05 with (P∧Q) (R∧Q) (Q∧R).
  intros Syll2_05a.
```

```
  MP Syll2_05a n3_22a.
  Syll Sa Syll2_05a Sb.
  specialize Simp3_27 with (P→R) (Q→S).
  intros Simp3_27a.
  specialize Fact3_45 with Q S R.
  intros Fact3_45b.
  Syll Simp3_27a Fact3_45b Sc.
  specialize n3_22 with S R.
  intros n3_22b.
  specialize Syll2_05 with (Q∧R) (S∧R) (R∧S).
  intros Syll2_05b.
  MP Syll2_05b n3_22b.
  Syll Sc Syll2_05b Sd.
  clear Simp3_26a. clear Fact3_45a. clear Sa.
    clear n3_22a. clear Fact3_45b.
    clear Syll2_05a. clear Simp3_27a.
    clear Sc. clear n3_22b. clear Syll2_05b.
  Conj Sb Sd.
  split.
  apply Sb.
  apply Sd.
  specialize n2_83 with ((P→R)∧(Q→S)) (P∧Q) (Q∧R) (R∧S).
  intros n2_83a. (*This with MP works, but it omits Conj3_03.*)
  specialize Imp3_31 with (((P→R)∧(Q→S))→((P∧Q)→(Q∧R)))
    (((P→R)∧(Q→S))→((Q∧R)→(R∧S)))
    (((P→R)∧(Q→S))→((P∧Q)→(R∧S))).
  intros Imp3_31a.
  MP Imp3_31a n2_83a.
  MP Imp3_31a H.
  apply Imp3_31a.
Qed.

Theorem n3_48 : ∀ P Q R S : Prop,
  ((P → R) ∧ (Q → S)) → (P ∨ Q) → R ∨ S.
Proof. intros P Q R S.
  specialize Simp3_26 with (P→R) (Q→S).
  intros Simp3_26a.
  specialize Sum1_6 with Q P R.
  intros Sum1_6a.
  Syll Simp3_26a Sum1_6a Sa.
  specialize Perm1_4 with P Q.
  intros Perm1_4a.
```

```
  specialize Syll2_06 with (P∨Q) (Q∨P) (Q∨R).
  intros Syll2_06a.
  MP Syll2_06a Perm1_4a.
  Syll Sa Syll2_06a Sb.
  specialize Simp3_27 with (P→R) (Q→S).
  intros Simp3_27a.
  specialize Sum1_6 with R Q S.
  intros Sum1_6b.
  Syll Simp3_27a Sum1_6b Sc.
  specialize Perm1_4 with Q R.
  intros Perm1_4b.
  specialize Syll2_06 with (Q∨R) (R∨Q) (R∨S).
  intros Syll2_06b.
  MP Syll2_06b Perm1_4b.
  Syll Sc Syll2_06a Sd.
  specialize n2_83 with ((P→R)∧(Q→S)) (P∨Q) (Q∨R) (R∨S).
  intros n2_83a.
  MP n2_83a Sb.
  MP n2_83a Sd.
  apply n2_83a.
Qed.

End No3.

Module No4.

Import No1.
Import No2.
Import No3.

Theorem Equiv4_01 : ∀ P Q : Prop,
  (P ↔ Q) = ((P → Q) ∧ (Q → P)).
  Proof. intros P Q.
  apply propositional_extensionality.
  specialize iff_to_and with P Q.
  intros iff_to_and.
  apply iff_to_and.
  Qed.
  (*This is a notational definition in Principia;
  it is used to switch between "↔" and "→∧←".*)

(*Axiom Abb4_02 : ∀ P Q R : Prop,
```

```
  (P ↔ Q ↔ R) = ((P ↔ Q) ∧ (Q ↔ R)).*)
  (*Since Coq forbids ill-formed strings, or else
  automatically associates to the right, we leave
  this notational axiom commented out.*)

Ltac Equiv H1 :=
  match goal with
    | [ H1 : (?P→?Q) ∧ (?Q→?P) |- _ ] =>
      replace ((P→Q) ∧ (Q→P)) with (P↔Q) in H1
end.

Theorem Transp4_1 : ∀ P Q : Prop,
  (P → Q) ↔ (¬Q → ¬P).
Proof. intros P Q.
  specialize Transp2_16 with P Q.
  intros Transp2_16a.
  specialize Transp2_17 with P Q.
  intros Transp2_17a.
  Conj Transp2_16a Transp2_17a.
  split.
  apply Transp2_16a.
  apply Transp2_17a.
  Equiv H.
  apply H.
  apply Equiv4_01.
Qed.

Theorem Transp4_11 : ∀ P Q : Prop,
  (P ↔ Q) ↔ (¬P ↔ ¬Q).
Proof. intros P Q.
  specialize Transp2_16 with P Q.
  intros Transp2_16a.
  specialize Transp2_16 with Q P.
  intros Transp2_16b.
  Conj Transp2_16a Transp2_16b.
  split.
  apply Transp2_16a.
  apply Transp2_16b.
  specialize n3_47 with (P→Q) (Q→P) (¬Q→¬P) (¬P→¬Q).
  intros n3_47a.
  MP n3_47 H.
  specialize n3_22 with (¬Q → ¬P) (¬P → ¬Q).
```

```
  intros n3_22a.
  Syll n3_47a n3_22a Sa.
  replace ((P → Q) ∧ (Q → P)) with (P↔Q) in Sa.
  replace ((¬P → ¬Q) ∧ (¬Q → ¬P)) with (¬P↔¬Q) in Sa.
  clear Transp2_16a. clear H. clear Transp2_16b.
   clear n3_22a. clear n3_47a.
  specialize Transp2_17 with Q P.
  intros Transp2_17a.
  specialize Transp2_17 with P Q.
  intros Transp2_17b.
  Conj Transp2_17a Transp2_17b.
  split.
  apply Transp2_17a.
  apply Transp2_17b.
  specialize n3_47 with (¬P→¬Q) (¬Q→¬P) (Q→P) (P→Q).
  intros n3_47a.
  MP n3_47a H.
  specialize n3_22 with (Q→P) (P→Q).
  intros n3_22a.
  Syll n3_47a n3_22a Sb.
  clear Transp2_17a. clear Transp2_17b. clear H.
      clear n3_47a. clear n3_22a.
  replace ((P → Q) ∧ (Q → P)) with (P↔Q) in Sb.
  replace ((¬P → ¬Q) ∧ (¬Q → ¬P)) with (¬P↔¬Q) in Sb.
  Conj Sa Sb.
  split.
  apply Sa.
  apply Sb.
  Equiv H.
  apply H.
  apply Equiv4_01.
  apply Equiv4_01.
  apply Equiv4_01.
  apply Equiv4_01.
  apply Equiv4_01.
Qed.

Theorem n4_12 : ∀ P Q : Prop,
  (P ↔ ¬Q) ↔ (Q ↔ ¬P).
  Proof. intros P Q.
    specialize Transp2_03 with P Q.
    intros Transp2_03a.
```

```
    specialize Transp2_15 with Q P.
    intros Transp2_15a.
    Conj Transp2_03a Transp2_15a.
    split.
    apply Transp2_03a.
    apply Transp2_15a.
    specialize n3_47 with (P→¬Q) (¬Q→P) (Q→¬P) (¬P→Q).
    intros n3_47a.
    MP n3_47a H.
    specialize Transp2_03 with Q P.
    intros Transp2_03b.
    specialize Transp2_15 with P Q.
    intros Transp2_15b.
    Conj Transp2_03b Transp2_15b.
    split.
    apply Transp2_03b.
    apply Transp2_15b.
    specialize n3_47 with (Q→¬P) (¬P→Q) (P→¬Q) (¬Q→P).
    intros n3_47b.
    MP n3_47b H0.
    clear Transp2_03a. clear Transp2_15a. clear H.
      clear Transp2_03b. clear Transp2_15b. clear H0.
    Conj n3_47a n3_47b.
    split.
    apply n3_47a.
    apply n3_47b.
    rewrite <- Equiv4_01 in H.
    rewrite <- Equiv4_01 in H.
    rewrite <- Equiv4_01 in H.
    apply H.
Qed.

Theorem n4_13 : ∀ P : Prop,
  P ↔ ¬¬P.
  Proof. intros P.
  specialize n2_12 with P.
  intros n2_12a.
  specialize n2_14 with P.
  intros n2_14a.
  Conj n2_12a n2_14a.
  split.
  apply n2_12a.
```

```
  apply n2_14a.
  Equiv H.
  apply H.
  apply Equiv4_01.
Qed.

Theorem n4_14 : ∀ P Q R : Prop,
  ((P ∧ Q) → R) ↔ ((P ∧ ¬R) → ¬Q).
Proof. intros P Q R.
specialize Transp3_37 with P Q R.
intros Transp3_37a.
specialize Transp3_37 with P (¬R) (¬Q).
intros Transp3_37b.
Conj Transp3_37a Transp3_37b.
split. apply Transp3_37a.
apply Transp3_37b.
specialize n4_13 with Q.
intros n4_13a.
specialize n4_13 with R.
intros n4_13b.
replace (¬¬Q) with Q in H.
replace (¬¬R) with R in H.
Equiv H.
apply H.
apply Equiv4_01.
apply propositional_extensionality.
apply n4_13b.
apply propositional_extensionality.
apply n4_13a.
Qed.

Theorem n4_15 : ∀ P Q R : Prop,
  ((P ∧ Q) → ¬R) ↔ ((Q ∧ R) → ¬P).
  Proof. intros P Q R.
  specialize n4_14 with Q P (¬R).
  intros n4_14a.
  specialize n3_22 with Q P.
  intros n3_22a.
  specialize Syll2_06 with (Q∧P) (P∧Q) (¬R).
  intros Syll2_06a.
  MP Syll2_06a n3_22a.
  specialize n4_13 with R.
```

```
  intros n4_13a.
  replace (¬¬R) with R in n4_14a.
  rewrite Equiv4_01 in n4_14a.
  specialize Simp3_26 with ((Q ∧ P → ¬R) → Q ∧ R → ¬P)
      ((Q ∧ R → ¬P) → Q ∧ P → ¬R).
  intros Simp3_26a.
  MP Simp3_26a n4_14a.
  Syll Syll2_06a Simp3_26a Sa.
  specialize Simp3_27 with ((Q ∧ P → ¬R) → Q ∧ R → ¬P)
      ((Q ∧ R → ¬P) → Q ∧ P → ¬R).
  intros Simp3_27a.
  MP Simp3_27a n4_14a.
  specialize n3_22 with P Q.
  intros n3_22b.
  specialize Syll2_06 with (P∧Q) (Q∧P) (¬R).
  intros Syll2_06b.
  MP Syll2_06b n3_22b.
  Syll Syll2_06b Simp3_27a Sb.
  clear n4_14a. clear n3_22a. clear Syll2_06a.
      clear n4_13a. clear Simp3_26a. clear n3_22b.
      clear Simp3_27a. clear Syll2_06b.
  Conj Sa Sb.
  split.
  apply Sa.
  apply Sb.
  Equiv H.
  apply H.
  apply Equiv4_01.
  apply propositional_extensionality.
  apply n4_13a.
Qed.

Theorem n4_2 : ∀ P : Prop,
  P ↔ P.
  Proof. intros P.
  specialize n3_2 with (P→P) (P→P).
  intros n3_2a.
  specialize Id2_08 with P.
  intros Id2_08a.
  MP n3_2a Id2_08a.
  MP n3_2a Id2_08a.
  Equiv n3_2a.
```

```
  apply n3_2a.
  apply Equiv4_01.
Qed.

Theorem n4_21 : ∀ P Q : Prop,
  (P ↔ Q) ↔ (Q ↔ P).
  Proof. intros P Q.
  specialize n3_22 with (P→Q) (Q→P).
  intros n3_22a.
  replace ((P → Q) ∧ (Q → P)) with (P↔Q) in n3_22a.
  replace ((Q → P) ∧ (P → Q)) with (Q↔P) in n3_22a.
  specialize n3_22 with (Q→P) (P→Q).
  intros n3_22b.
  replace ((P → Q) ∧ (Q → P)) with (P↔Q) in n3_22b.
  replace ((Q → P) ∧ (P → Q)) with (Q↔P) in n3_22b.
  Conj n3_22a n3_22b.
  split.
  apply n3_22a.
  apply n3_22b.
  Equiv H.
  apply H.
  apply Equiv4_01.
  apply Equiv4_01.
  apply Equiv4_01.
  apply Equiv4_01.
  apply Equiv4_01.
Qed.

Theorem n4_22 : ∀ P Q R : Prop,
  ((P ↔ Q) ∧ (Q ↔ R)) → (P ↔ R).
Proof. intros P Q R.
  specialize Simp3_26 with (P↔Q) (Q↔R).
  intros Simp3_26a.
  specialize Simp3_26 with (P→Q) (Q→P).
  intros Simp3_26b.
  replace ((P→Q) ∧ (Q→P)) with (P↔Q) in Simp3_26b.
  Syll Simp3_26a Simp3_26b Sa.
  specialize Simp3_27 with (P↔Q) (Q↔R).
  intros Simp3_27a.
  specialize Simp3_26 with (Q→R) (R→Q).
  intros Simp3_26c.
  replace ((Q→R) ∧ (R→Q)) with (Q↔R) in Simp3_26c.
```

```
  Syll Simp3_27a Simp3_26c Sb.
  specialize n2_83 with ((P↔Q)∧(Q↔R)) P Q R.
  intros n2_83a.
  MP n2_83a Sa.
  MP n2_83a Sb.
  specialize Simp3_27 with (P↔Q) (Q↔R).
  intros Simp3_27b.
  specialize Simp3_27 with (Q→R) (R→Q).
  intros Simp3_27c.
  replace ((Q→R) ∧ (R→Q)) with (Q↔R) in Simp3_27c.
  Syll Simp3_27b Simp3_27c Sc.
  specialize Simp3_26 with (P↔Q) (Q↔R).
  intros Simp3_26d.
  specialize Simp3_27 with (P→Q) (Q→P).
  intros Simp3_27d.
  replace ((P→Q) ∧ (Q→P)) with (P↔Q) in Simp3_27d.
  Syll Simp3_26d Simp3_27d Sd.
  specialize n2_83 with ((P↔Q)∧(Q↔R)) R Q P.
  intros n2_83b.
  MP n2_83b Sc. MP n2_83b Sd.
  clear Sd. clear Sb. clear Sc. clear Sa. clear Simp3_26a.
    clear Simp3_26b. clear Simp3_26c. clear Simp3_26d.
    clear Simp3_27a. clear Simp3_27b. clear Simp3_27c.
    clear Simp3_27d.
  Conj n2_83a n2_83b.
  split.
  apply n2_83a.
  apply n2_83b.
  specialize Comp3_43 with ((P↔Q)∧(Q↔R)) (P→R) (R→P).
  intros Comp3_43a.
  MP Comp3_43a H.
  replace ((P→R) ∧ (R→P)) with (P↔R) in Comp3_43a.
  apply Comp3_43a.
  apply Equiv4_01.
  apply Equiv4_01.
  apply Equiv4_01.
  apply Equiv4_01.
  apply Equiv4_01.
Qed.

Theorem n4_24 : ∀ P : Prop,
  P ↔ (P ∧ P).
```

```
  Proof. intros P.
  specialize n3_2 with P P.
  intros n3_2a.
  specialize n2_43 with P (P ∧ P).
  intros n2_43a.
  MP n3_2a n2_43a.
  specialize Simp3_26 with P P.
  intros Simp3_26a.
  Conj n2_43a Simp3_26a.
  split.
  apply n2_43a.
  apply Simp3_26a.
  Equiv H.
  apply H.
  apply Equiv4_01.
Qed.

Theorem n4_25 : ∀ P : Prop,
  P ↔ (P ∨ P).
Proof. intros P.
  specialize Add1_3 with P P.
  intros Add1_3a.
  specialize Taut1_2 with P.
  intros Taut1_2a.
  Conj Add1_3a Taut1_2a.
  split.
  apply Add1_3a.
  apply Taut1_2a.
  Equiv H. apply H.
  apply Equiv4_01.
Qed.

Theorem n4_3 : ∀ P Q : Prop,
  (P ∧ Q) ↔ (Q ∧ P).
Proof. intros P Q.
  specialize n3_22 with P Q.
  intros n3_22a.
  specialize n3_22 with Q P.
  intros n3_22b.
  Conj n3_22a n3_22b.
  split.
  apply n3_22a.
```

```
  apply n3_22b.
  Equiv H. apply H.
  apply Equiv4_01.
Qed.

Theorem n4_31 : ∀ P Q : Prop,
  (P ∨ Q) ↔ (Q ∨ P).
  Proof. intros P Q.
    specialize Perm1_4 with P Q.
    intros Perm1_4a.
    specialize Perm1_4 with Q P.
    intros Perm1_4b.
    Conj Perm1_4a Perm1_4b.
    split.
    apply Perm1_4a.
    apply Perm1_4b.
    Equiv H. apply H.
    apply Equiv4_01.
Qed.

Theorem n4_32 : ∀ P Q R : Prop,
    ((P ∧ Q) ∧ R) ↔ (P ∧ (Q ∧ R)).
  Proof. intros P Q R.
    specialize n4_15 with P Q R.
    intros n4_15a.
    specialize Transp4_1 with P (¬(Q ∧ R)).
    intros Transp4_1a.
    replace (¬¬(Q ∧ R)) with (Q ∧ R) in Transp4_1a.
    replace (Q ∧ R→¬P) with (P→¬(Q ∧ R)) in n4_15a.
    specialize Transp4_11 with (P ∧ Q → ¬R) (P → ¬(Q ∧ R)).
    intros Transp4_11a.
    replace ((P ∧ Q → ¬R) ↔ (P → ¬(Q ∧ R))) with
        (¬(P ∧ Q → ¬R) ↔ ¬(P → ¬(Q ∧ R))) in n4_15a.
    replace (P ∧ Q → ¬R) with
        (¬(P ∧ Q ) ∨ ¬R) in n4_15a.
    replace (P → ¬(Q ∧ R)) with
        (¬P ∨ ¬(Q ∧ R)) in n4_15a.
    replace (¬(¬(P ∧ Q) ∨ ¬R)) with
        ((P ∧ Q) ∧ R) in n4_15a.
    replace (¬(¬P ∨ ¬(Q ∧ R))) with
        (P ∧ (Q ∧ R )) in n4_15a.
    apply n4_15a.
```

```
    apply Prod3_01.
    apply Prod3_01.
    rewrite Impl1_01.
    reflexivity.
    rewrite Impl1_01.
    reflexivity.
    replace (¬(P ∧ Q → ¬R) ↔ ¬(P → ¬(Q ∧ R))) with
        ((P ∧ Q → ¬R) ↔ (P → ¬(Q ∧ R))).
    reflexivity.
    apply propositional_extensionality.
    apply Transp4_11a.
    apply propositional_extensionality.
    apply Transp4_1a.
    apply propositional_extensionality.
    specialize n4_13 with (Q ∧ R).
    intros n4_13a.
    apply n4_13a.
Qed.
    (*Note that the actual proof uses n4_12, but
        that transposition involves transforming a
        biconditional into a conditional. This citation
        of the lemma may be a misprint. Using
        Transp4_1 to transpose a conditional and
        then applying n4_13 to double negate does
        secure the desired formula.*)

Theorem n4_33 : ∀ P Q R : Prop,
  (P ∨ (Q ∨ R)) ↔ ((P ∨ Q) ∨ R).
  Proof. intros P Q R.
    specialize n2_31 with P Q R.
    intros n2_31a.
    specialize n2_32 with P Q R.
    intros n2_32a.
    Conj n2_31a n2_32a.
    split.
    apply n2_31a.
    apply n2_32a.
    Equiv H.
    apply H.
    apply Equiv4_01.
Qed.
```

```
Theorem Abb4_34 : ∀ P Q R : Prop,
  (P ∧ Q ∧ R) = ((P ∧ Q) ∧ R).
  Proof. intros P Q R.
  apply propositional_extensionality.
  specialize n4_21 with ((P ∧ Q) ∧ R) (P ∧ Q ∧ R).
  intros n4_21.
  replace (((P ∧ Q) ∧ R ↔ P ∧ Q ∧ R) ↔ (P ∧ Q ∧ R ↔ (P ∧ Q) ∧ R))
    with ((((P ∧ Q) ∧ R ↔ P ∧ Q ∧ R) → (P ∧ Q ∧ R ↔ (P ∧ Q) ∧ R))
    ∧ ((P ∧ Q ∧ R ↔ (P ∧ Q) ∧ R) → ((P ∧ Q) ∧ R ↔ P ∧ Q ∧ R)))
    in n4_21.
  specialize Simp3_26 with
    (((P ∧ Q) ∧ R ↔ P ∧ Q ∧ R) → (P ∧ Q ∧ R ↔ (P ∧ Q) ∧ R))
    ((P ∧ Q ∧ R ↔ (P ∧ Q) ∧ R) → ((P ∧ Q) ∧ R ↔ P ∧ Q ∧ R)).
  intros Simp3_26.
  MP Simp3_26 n4_21.
  specialize n4_32 with P Q R.
  intros n4_32.
  MP Simp3_26 n4_32.
  apply Simp3_26.
  apply Equiv4_01.
Qed.

Theorem n4_36 : ∀ P Q R : Prop,
  (P ↔ Q) → ((P ∧ R) ↔ (Q ∧ R)).
Proof. intros P Q R.
  specialize Fact3_45 with P Q R.
  intros Fact3_45a.
  specialize Fact3_45 with Q P R.
  intros Fact3_45b.
  Conj Fact3_45a Fact3_45b.
  split.
  apply Fact3_45a.
  apply Fact3_45b.
  specialize n3_47 with (P→Q) (Q→P)
      (P ∧ R → Q ∧ R) (Q ∧ R → P ∧ R).
  intros n3_47a.
  MP n3_47 H.
  replace  ((P → Q) ∧ (Q → P)) with (P↔Q) in n3_47a.
  replace ((P ∧ R → Q ∧ R) ∧ (Q ∧ R → P ∧ R)) with
      (P ∧ R ↔ Q ∧ R) in n3_47a.
  apply n3_47a.
  apply Equiv4_01.
```

```
  apply Equiv4_01.
Qed.

Theorem n4_37 : ∀ P Q R : Prop,
  (P ↔ Q) → ((P ∨ R) ↔ (Q ∨ R)).
Proof. intros P Q R.
  specialize Sum1_6 with R P Q.
  intros Sum1_6a.
  specialize Sum1_6 with R Q P.
  intros Sum1_6b.
  Conj Sum1_6a Sum1_6b.
  split.
  apply Sum1_6a.
  apply Sum1_6b.
  specialize n3_47 with (P → Q) (Q → P)
      (R ∨ P → R ∨ Q) (R ∨ Q → R ∨ P).
  intros n3_47a.
  MP n3_47 H.
  replace  ((P → Q) ∧ (Q → P)) with (P↔Q) in n3_47a.
  replace ((R ∨ P → R ∨ Q) ∧ (R ∨ Q → R ∨ P)) with
      (R ∨ P ↔ R ∨ Q) in n3_47a.
  replace (R ∨ P) with (P ∨ R) in n3_47a.
  replace (R ∨ Q) with (Q ∨ R) in n3_47a.
  apply n3_47a.
  apply propositional_extensionality.
  specialize n4_31 with Q R.
  intros n4_31a.
  apply n4_31a.
  apply propositional_extensionality.
  specialize n4_31 with P R.
  intros n4_31b.
  apply n4_31b.
  apply Equiv4_01.
  apply Equiv4_01.
Qed.

Theorem n4_38 : ∀ P Q R S : Prop,
  ((P ↔ R) ∧ (Q ↔ S)) → ((P ∧ Q) ↔ (R ∧ S)).
Proof. intros P Q R S.
  specialize n3_47 with P Q R S.
  intros n3_47a.
  specialize n3_47 with R S P Q.
```

```
intros n3_47b.
Conj n3_47a n3_47b.
split.
apply n3_47a.
apply n3_47b.
specialize n3_47 with ((P→R) ∧ (Q→S))
    ((R→P) ∧ (S→Q)) (P ∧ Q → R ∧ S) (R ∧ S → P ∧ Q).
intros n3_47c.
MP n3_47c H.
specialize n4_32 with (P→R) (Q→S) ((R→P) ∧ (S → Q)).
intros n4_32a.
replace (((P → R) ∧ (Q → S)) ∧ (R → P) ∧ (S → Q)) with
    ((P → R) ∧ (Q → S) ∧ (R → P) ∧ (S → Q)) in n3_47c.
specialize n4_32 with (Q→S) (R→P) (S → Q).
intros n4_32b.
replace ((Q → S) ∧ (R → P) ∧ (S → Q)) with
    (((Q → S) ∧ (R → P)) ∧ (S → Q)) in n3_47c.
specialize n3_22 with (Q→S) (R→P).
intros n3_22a.
specialize n3_22 with (R→P) (Q→S).
intros n3_22b.
Conj n3_22a n3_22b.
split.
apply n3_22a.
apply n3_22b.
Equiv H0.
replace ((Q → S) ∧ (R → P)) with
    ((R → P) ∧ (Q → S)) in n3_47c.
specialize n4_32 with (R → P) (Q → S) (S → Q).
intros n4_32c.
replace (((R → P) ∧ (Q → S)) ∧ (S → Q)) with
    ((R → P) ∧ (Q → S) ∧ (S → Q)) in n3_47c.
specialize n4_32 with (P→R) (R → P) ((Q → S)∧(S → Q)).
intros n4_32d.
replace ((P → R) ∧ (R → P) ∧ (Q → S) ∧ (S → Q)) with
    (((P → R) ∧ (R → P)) ∧ (Q → S) ∧ (S → Q)) in n3_47c.
replace ((P→R) ∧ (R → P)) with (P↔R) in n3_47c.
replace ((Q → S) ∧ (S → Q)) with (Q↔S) in n3_47c.
replace ((P ∧ Q → R ∧ S) ∧ (R ∧ S → P ∧ Q)) with
    ((P ∧ Q) ↔ (R ∧ S)) in n3_47c.
apply n3_47c.
apply Equiv4_01.
```

```
  apply Equiv4_01.
  apply Equiv4_01.
  apply propositional_extensionality.
  apply n4_32d.
  replace ((R → P) ∧ (Q → S) ∧ (S → Q)) with
      (((R → P) ∧ (Q → S)) ∧ (S → Q)).
  reflexivity.
  apply propositional_extensionality.
  apply n4_32c.
  replace ((R → P) ∧ (Q → S)) with ((Q → S) ∧ (R → P)).
  reflexivity.
  apply propositional_extensionality.
  apply H0.
  apply Equiv4_01.
  apply propositional_extensionality.
  apply n4_32b.
  replace ((P → R) ∧ (Q → S) ∧ (R → P) ∧ (S → Q)) with
      (((P → R) ∧ (Q → S)) ∧ (R → P) ∧ (S → Q)).
  reflexivity.
  apply propositional_extensionality.
  apply n4_32a.
Qed.

Theorem n4_39 : ∀ P Q R S : Prop,
  ((P ↔ R) ∧ (Q ↔ S)) → ((P ∨ Q) ↔ (R ∨ S)).
Proof.  intros P Q R S.
  specialize n3_48 with P Q R S.
  intros n3_48a.
  specialize n3_48 with R S P Q.
  intros n3_48b.
  Conj n3_48a n3_48b.
  split.
  apply n3_48a.
  apply n3_48b.
  specialize n3_47 with ((P → R) ∧ (Q → S))
      ((R → P) ∧ (S → Q)) (P ∨ Q → R ∨ S) (R ∨ S → P ∨ Q).
  intros n3_47a.
  MP n3_47a H.
  replace ((P ∨ Q → R ∨ S) ∧ (R ∨ S → P ∨ Q)) with
      ((P ∨ Q) ↔ (R ∨ S)) in n3_47a.
  specialize n4_32 with ((P → R) ∧ (Q → S)) (R → P) (S → Q).
  intros n4_32a.
```

```
replace (((P → R) ∧ (Q → S)) ∧ (R → P) ∧ (S → Q)) with
    ((((P → R) ∧ (Q → S)) ∧ (R → P)) ∧ (S → Q)) in n3_47a.
specialize n4_32 with (P → R) (Q → S) (R → P).
intros n4_32b.
replace (((P → R) ∧ (Q → S)) ∧ (R → P)) with
    ((P → R) ∧ (Q → S) ∧ (R → P)) in n3_47a.
specialize n3_22 with (Q → S) (R → P).
intros n3_22a.
specialize n3_22 with (R → P) (Q → S).
intros n3_22b.
Conj  n3_22a n3_22b.
split.
apply n3_22a.
apply n3_22b.
Equiv H0.
replace ((Q → S) ∧ (R → P)) with
    ((R → P) ∧ (Q → S)) in n3_47a.
specialize n4_32 with (P → R) (R → P) (Q → S).
intros n4_32c.
replace ((P → R) ∧ (R → P) ∧ (Q → S)) with
    (((P → R) ∧ (R → P)) ∧ (Q → S)) in n3_47a.
replace ((P → R) ∧ (R → P)) with (P↔R) in n3_47a.
specialize n4_32 with (P↔R) (Q→S) (S→Q).
intros n4_32d.
replace (((P ↔ R) ∧ (Q → S)) ∧ (S → Q)) with
    ((P ↔ R) ∧ (Q → S) ∧ (S → Q)) in n3_47a.
replace ((Q → S) ∧ (S → Q)) with (Q ↔ S) in n3_47a.
apply n3_47a.
apply Equiv4_01.
replace ((P ↔ R) ∧ (Q → S) ∧ (S → Q)) with
    (((P ↔ R) ∧ (Q → S)) ∧ (S → Q)).
reflexivity.
apply propositional_extensionality.
apply n4_32d.
apply Equiv4_01.
apply propositional_extensionality.
apply n4_32c.
replace ((R → P) ∧ (Q → S)) with ((Q → S) ∧ (R → P)).
reflexivity.
apply propositional_extensionality.
apply H0.
apply Equiv4_01.
```

```
  replace ((P → R) ∧ (Q → S) ∧ (R → P)) with
      (((P → R) ∧ (Q → S)) ∧ (R → P)).
  reflexivity.
  apply propositional_extensionality.
  apply n4_32b.
  apply propositional_extensionality.
  apply n4_32a.
  apply Equiv4_01.
Qed.

Theorem n4_4 : ∀ P Q R : Prop,
  (P ∧ (Q ∨ R)) ↔ ((P∧ Q) ∨ (P ∧ R)).
Proof. intros P Q R.
  specialize n3_2 with P Q.
  intros n3_2a.
  specialize n3_2 with P R.
  intros n3_2b.
  Conj n3_2a n3_2b.
  split.
  apply n3_2a.
  apply n3_2b.
  specialize Comp3_43 with P (Q→P∧Q) (R→P∧R).
  intros Comp3_43a.
  MP Comp3_43a H.
  specialize n3_48 with Q R (P∧Q) (P∧R).
  intros n3_48a.
  Syll Comp3_43a n3_48a Sa.
  specialize Imp3_31 with P (Q∨R) ((P∧ Q) ∨ (P ∧ R)).
  intros Imp3_31a.
  MP Imp3_31a Sa.
  specialize Simp3_26 with P Q.
  intros Simp3_26a.
  specialize Simp3_26 with P R.
  intros Simp3_26b.
  Conj Simp3_26a Simp3_26b.
  split.
  apply Simp3_26a.
  apply Simp3_26b.
  specialize n3_44 with P (P∧Q) (P∧R).
  intros n3_44a.
  MP n3_44a H0.
  specialize Simp3_27 with P Q.
```

```
  intros Simp3_27a.
  specialize Simp3_27 with P R.
  intros Simp3_27b.
  Conj Simp3_27a Simp3_27b.
  split.
  apply Simp3_27a.
  apply Simp3_27b.
  specialize n3_48 with (P∧Q) (P∧R) Q R.
  intros n3_48b.
  MP n3_48b H1.
  clear H1. clear Simp3_27a. clear Simp3_27b.
  Conj n3_44a n3_48b.
  split.
  apply n3_44a.
  apply n3_48b.
  specialize Comp3_43 with (P ∧ Q ∨ P ∧ R) P (Q∨R).
  intros Comp3_43b.
  MP Comp3_43b H1.
  clear H1. clear H0. clear n3_44a. clear n3_48b.
      clear Simp3_26a. clear Simp3_26b.
  Conj Imp3_31a Comp3_43b.
  split.
  apply Imp3_31a.
  apply Comp3_43b.
  Equiv H0.
  apply H0.
  apply Equiv4_01.
Qed.

Theorem n4_41 : ∀ P Q R : Prop,
  (P ∨ (Q ∧ R)) ↔ ((P ∨ Q) ∧ (P ∨ R)).
Proof. intros P Q R.
  specialize Simp3_26 with Q R.
  intros Simp3_26a.
  specialize Sum1_6 with P (Q ∧ R) Q.
  intros Sum1_6a.
  MP Simp3_26a Sum1_6a.
  specialize Simp3_27 with Q R.
  intros Simp3_27a.
  specialize Sum1_6 with P (Q ∧ R) R.
  intros Sum1_6b.
  MP Simp3_27a Sum1_6b.
```

```
  clear Simp3_26a. clear Simp3_27a.
  Conj Sum1_6a Sum1_6a.
  split.
  apply Sum1_6a.
  apply Sum1_6b.
  specialize Comp3_43 with (P ∨ Q ∧ R) (P ∨ Q) (P ∨ R).
  intros Comp3_43a.
  MP Comp3_43a H.
  specialize n2_53 with P Q.
  intros n2_53a.
  specialize n2_53 with P R.
  intros n2_53b.
  Conj n2_53a n2_53b.
  split.
  apply n2_53a.
  apply n2_53b.
  specialize n3_47 with (P ∨ Q) (P ∨ R) (¬P → Q) (¬P → R).
  intros n3_47a.
  MP n3_47a H0.
  specialize Comp3_43 with (¬P) Q R.
  intros Comp3_43b.
  Syll n3_47a Comp3_43b Sa.
  specialize n2_54 with P (Q∧R).
  intros n2_54a.
  Syll Sa n2_54a Sb.
  clear Sum1_6a. clear Sum1_6b. clear H. clear n2_53a.
      clear n2_53b. clear H0. clear n3_47a. clear Sa.
      clear Comp3_43b. clear n2_54a.
  Conj Comp3_43a Sb.
  split.
  apply Comp3_43a.
  apply Sb.
  Equiv H.
  apply H.
  apply Equiv4_01.
Qed.

Theorem n4_42 : ∀ P Q : Prop,
  P ↔ ((P ∧ Q) ∨ (P ∧ ¬Q)).
Proof. intros P Q.
  specialize n3_21 with P (Q ∨ ¬Q).
  intros n3_21a.
```

```
  specialize n2_11 with Q.
  intros n2_11a.
  MP n3_21a n2_11a.
  specialize Simp3_26 with P (Q ∨ ¬Q).
  intros Simp3_26a. clear n2_11a.
  Conj n3_21a Simp3_26a.
  split.
  apply n3_21a.
  apply Simp3_26a.
  Equiv H.
  specialize n4_4 with P Q (¬Q).
  intros n4_4a.
  replace (P ∧ (Q ∨ ¬Q)) with P in n4_4a.
  apply n4_4a.
  apply propositional_extensionality.
  apply H.
  apply Equiv4_01.
Qed.

Theorem n4_43 : ∀ P Q : Prop,
  P ↔ ((P ∨ Q) ∧ (P ∨ ¬Q)).
Proof. intros P Q.
  specialize n2_2 with P Q.
  intros n2_2a.
  specialize n2_2 with P (¬Q).
  intros n2_2b.
  Conj n2_2a n2_2b.
  split.
  apply n2_2a.
  apply n2_2b.
  specialize Comp3_43 with P (P∨Q) (P∨¬Q).
  intros Comp3_43a.
  MP Comp3_43a H.
  specialize n2_53 with P Q.
  intros n2_53a.
  specialize n2_53 with P (¬Q).
  intros n2_53b.
  Conj n2_53a n2_53b.
  split.
  apply n2_53a.
  apply n2_53b.
  specialize n3_47 with (P∨Q) (P∨¬Q) (¬P→Q) (¬P→¬Q).
```

```
  intros n3_47a.
  MP n3_47a H0.
  specialize n2_65 with (¬P) Q.
  intros n2_65a.
  replace (¬¬P) with P in n2_65a.
  specialize Imp3_31 with (¬P → Q) (¬P → ¬Q) (P).
  intros Imp3_31a.
  MP Imp3_31a n2_65a.
  Syll n3_47a Imp3_31a Sa.
  clear n2_2a. clear n2_2b. clear H. clear n2_53a.
    clear n2_53b. clear H0. clear n2_65a.
    clear n3_47a. clear Imp3_31a.
  Conj Comp3_43a Sa.
  split.
  apply Comp3_43a.
  apply Sa.
  Equiv H.
  apply H.
  apply Equiv4_01.
  apply propositional_extensionality.
  specialize n4_13 with P.
  intros n4_13a.
  apply n4_13a.
Qed.

Theorem n4_44 : ∀ P Q : Prop,
  P ↔ (P ∨ (P ∧ Q)).
  Proof. intros P Q.
    specialize n2_2 with P (P∧Q).
    intros n2_2a.
    specialize Id2_08 with P.
    intros Id2_08a.
    specialize Simp3_26 with P Q.
    intros Simp3_26a.
    Conj Id2_08a Simp3_26a.
    split.
    apply Id2_08a.
    apply Simp3_26a.
    specialize n3_44 with P P (P ∧ Q).
    intros n3_44a.
    MP n3_44a H.
    clear H. clear Id2_08a. clear Simp3_26a.
```

```
    Conj n2_2a n3_44a.
    split.
    apply n2_2a.
    apply n3_44a.
    Equiv H.
    apply H.
    apply Equiv4_01.
Qed.

Theorem n4_45 : ∀ P Q : Prop,
  P ↔ (P ∧ (P ∨ Q)).
  Proof. intros P Q.
  specialize n2_2 with (P ∧ P) (P ∧ Q).
  intros n2_2a.
  replace (P ∧ P ∨ P ∧ Q) with (P ∧ (P ∨ Q)) in n2_2a.
  replace (P ∧ P) with P in n2_2a.
  specialize Simp3_26 with P (P ∨ Q).
  intros Simp3_26a.
  Conj n2_2a Simp3_26a.
  split.
  apply n2_2a.
  apply Simp3_26a.
  Equiv H.
  apply H.
  apply Equiv4_01.
  specialize n4_24 with P.
  intros n4_24a.
  apply propositional_extensionality.
  apply n4_24a.
  specialize n4_4 with P P Q.
  intros n4_4a.
  apply propositional_extensionality.
  apply n4_4a.
Qed.

Theorem n4_5 : ∀ P Q : Prop,
  P ∧ Q ↔ ¬(¬P ∨ ¬Q).
  Proof. intros P Q.
    specialize n4_2 with (P ∧ Q).
    intros n4_2a.
    rewrite Prod3_01.
    replace (¬(¬P ∨ ¬Q)) with (P ∧ Q).
```

```
    apply n4_2a.
    apply Prod3_01.
Qed.

Theorem n4_51 : ∀ P Q : Prop,
  ¬(P ∧ Q) ↔ (¬P ∨ ¬Q).
  Proof. intros P Q.
    specialize n4_5 with P Q.
    intros n4_5a.
    specialize n4_12 with (P ∧ Q) (¬P ∨ ¬Q).
    intros n4_12a.
    specialize Simp3_26 with
      ((P ∧ Q ↔ ¬(¬P ∨ ¬Q)) → (¬P ∨ ¬Q ↔ ¬(P ∧ Q)))
      ((¬P ∨ ¬Q ↔ ¬(P ∧ Q)) → ((P ∧ Q ↔ ¬(¬P ∨ ¬Q)))).
    intros Simp3_26a.
    replace ((P ∧ Q ↔ ¬(¬P ∨ ¬Q)) ↔ (¬P ∨ ¬Q ↔ ¬(P ∧ Q)))
      with (((P ∧ Q ↔ ¬(¬P ∨ ¬Q)) → (¬P ∨ ¬Q ↔ ¬(P ∧ Q)))
      ∧ ((¬P ∨ ¬Q ↔ ¬(P ∧ Q)) → ((P ∧ Q ↔ ¬(¬P ∨ ¬Q)))))
      in n4_12a.
    MP Simp3_26a n4_12a.
    MP Simp3_26a n4_5a.
    specialize n4_21 with (¬(P ∧ Q)) (¬P ∨ ¬Q).
    intros n4_21a.
    specialize Simp3_27 with
    (((¬(P ∧ Q) ↔ ¬P ∨ ¬Q)) → ((¬P ∨ ¬Q ↔ ¬(P ∧ Q))))
    (((¬P ∨ ¬Q ↔ ¬(P ∧ Q))) → ((¬(P ∧ Q) ↔ ¬P ∨ ¬Q))).
    intros Simp3_27a.
    MP Simp3_27a n4_21a.
    MP Simp3_27a Simp3_26a.
    apply Simp3_27a.
    apply Equiv4_01.
Qed.

Theorem n4_52 : ∀ P Q : Prop,
  (P ∧ ¬Q) ↔ ¬(¬P ∨ Q).
  Proof. intros P Q.
    specialize n4_5 with P (¬Q).
    intros n4_5a.
    replace (¬¬Q) with Q in n4_5a.
    apply n4_5a.
    specialize n4_13 with Q.
    intros n4_13a.
```

```
    apply propositional_extensionality.
    apply n4_13a.
Qed.

Theorem n4_53 : ∀ P Q : Prop,
  ¬(P ∧ ¬Q) ↔ (¬P ∨ Q).
  Proof. intros P Q.
    specialize n4_52 with P Q.
    intros n4_52a.
    specialize n4_12 with (P ∧ ¬Q) ((¬P ∨ Q)).
    intros n4_12a.
    replace ((P ∧ ¬Q ↔ ¬(¬P ∨ Q)) ↔ (¬P ∨ Q ↔ ¬(P ∧ ¬Q)))
      with (((P ∧ ¬Q ↔ ¬(¬P ∨ Q)) → (¬P ∨ Q ↔ ¬(P ∧ ¬Q)))
      ∧ ((¬P ∨ Q ↔ ¬(P ∧ ¬Q)) → (P ∧ ¬Q ↔ ¬(¬P ∨ Q))))
      in n4_12a.
    specialize Simp3_26 with
      ((P ∧ ¬Q ↔ ¬(¬P ∨ Q)) → (¬P ∨ Q ↔ ¬(P ∧ ¬Q)))
      ((¬P ∨ Q ↔ ¬(P ∧ ¬Q)) → (P ∧ ¬Q ↔ ¬(¬P ∨ Q))).
    intros Simp3_26a.
    MP Simp3_26a n4_12a.
    MP Simp3_26a n4_52a.
    specialize n4_21 with (¬(P ∧ ¬Q)) (¬P ∨ Q).
    intros n4_21a.
    replace ((¬(P ∧ ¬ Q) ↔ ¬P ∨ Q) ↔ (¬P ∨ Q ↔ ¬(P ∧ ¬Q)))
      with (((¬(P ∧ ¬ Q) ↔ ¬P ∨ Q) → (¬P ∨ Q ↔ ¬(P ∧ ¬Q)))
      ∧ ((¬P ∨ Q ↔ ¬(P ∧ ¬Q)) → (¬(P ∧ ¬ Q) ↔ ¬P ∨ Q)))
      in n4_21a.
    specialize Simp3_27 with
      ((¬(P ∧ ¬ Q) ↔ ¬P ∨ Q) → (¬P ∨ Q ↔ ¬(P ∧ ¬Q)))
      ((¬P ∨ Q ↔ ¬(P ∧ ¬Q)) → (¬(P ∧ ¬ Q) ↔ ¬P ∨ Q)).
    intros Simp3_27a.
    MP Simp3_27a n4_21a.
    MP Simp3_27a Simp3_26a.
    apply Simp3_27a.
    apply Equiv4_01.
    apply Equiv4_01.
Qed.

Theorem n4_54 : ∀ P Q : Prop,
  (¬P ∧ Q) ↔ ¬(P ∨ ¬Q).
  Proof. intros P Q.
    specialize n4_5 with (¬P) Q.
```

```
    intros n4_5a.
    specialize n4_13 with P.
    intros n4_13a.
    replace (¬¬P) with P in n4_5a.
    apply n4_5a.
    apply propositional_extensionality.
    apply n4_13a.
Qed.

Theorem n4_55 : ∀ P Q : Prop,
  ¬(¬P ∧ Q) ↔ (P ∨ ¬Q).
  Proof. intros P Q.
    specialize n4_54 with P Q.
    intros n4_54a.
    specialize n4_12 with (¬P ∧ Q) (P ∨ ¬Q).
    intros n4_12a.
    replace (¬P ∧ Q ↔ ¬(P ∨ ¬Q)) with
        (P ∨ ¬Q ↔ ¬(¬P ∧ Q)) in n4_54a.
    replace (P ∨ ¬Q ↔ ¬(¬P ∧ Q)) with
        (¬(¬P ∧ Q) ↔ (P ∨ ¬Q)) in n4_54a.
    apply n4_54a.
    specialize n4_21 with (¬(¬P ∧ Q)) (P ∨ ¬Q).
    intros n4_21a. (*Not cited*)
    apply propositional_extensionality.
    apply n4_21a.
    apply propositional_extensionality.
    replace ((P∨¬Q↔¬(¬P∧Q))↔(¬P∧Q↔¬(P∨¬Q)))
      with ((¬P∧Q↔¬(P∨¬Q))↔(P∨¬Q↔¬(¬P∧Q))).
    apply n4_12a.
    apply propositional_extensionality.
    specialize n4_21 with (P∨¬Q↔¬(¬P∧Q))
    (¬P∧Q↔¬(P∨¬Q)).
    intros n4_21b.
    apply n4_21.
Qed.

Theorem n4_56 : ∀ P Q : Prop,
  (¬P ∧ ¬Q) ↔ ¬(P ∨ Q).
  Proof. intros P Q.
    specialize n4_54 with P (¬Q).
    intros n4_54a.
    replace (¬¬Q) with Q in n4_54a.
```

```
    apply n4_54a.
    apply propositional_extensionality.
    specialize n4_13 with Q.
    intros n4_13a.
    apply n4_13a.
Qed.

Theorem n4_57 : ∀ P Q : Prop,
  ¬(¬P ∧ ¬Q) ↔ (P ∨ Q).
  Proof. intros P Q.
    specialize n4_56 with P Q.
    intros n4_56a.
    specialize n4_12 with (¬P ∧ ¬Q) (P ∨ Q).
    intros n4_12a.
    replace (¬P ∧ ¬Q ↔ ¬(P ∨ Q)) with
        (P ∨ Q ↔ ¬(¬P ∧ ¬Q)) in n4_56a.
    replace (P ∨ Q ↔ ¬(¬P ∧ ¬Q)) with
        (¬(¬P ∧ ¬Q) ↔ P ∨ Q) in n4_56a.
    apply n4_56a.
    specialize n4_21 with (¬(¬P ∧ ¬Q)) (P ∨ Q).
    intros n4_21a.
    apply propositional_extensionality.
    apply n4_21a.
    replace ((¬P∧¬Q↔¬(P∨Q))↔(P∨Q↔¬(¬P∧¬Q))) with
        ((P∨Q↔¬(¬P∧¬Q))↔(¬P∧¬Q↔¬(P∨Q))) in n4_12a.
    apply propositional_extensionality.
    apply n4_12a.
    apply propositional_extensionality.
    specialize n4_21 with
        (P ∨ Q ↔ ¬(¬P ∧ ¬Q)) (¬P ∧ ¬Q ↔ ¬(P ∨ Q)).
    intros n4_21b.
    apply n4_21b.
Qed.

Theorem n4_6 : ∀ P Q : Prop,
  (P → Q) ↔ (¬P ∨ Q).
  Proof. intros P Q.
    specialize n4_2 with (¬P∨ Q).
    intros n4_2a.
    rewrite Impl1_01.
    apply n4_2a.
Qed.
```

```
Theorem n4_61 : ∀ P Q : Prop,
  ¬(P → Q) ↔ (P ∧ ¬Q).
  Proof. intros P Q.
  specialize n4_6 with P Q.
  intros n4_6a.
  specialize Transp4_11 with (P→Q) (¬P∨Q).
  intros Transp4_11a.
  specialize n4_52 with P Q.
  intros n4_52a.
  replace ((P → Q) ↔ ¬P ∨ Q) with
      (¬(P → Q) ↔ ¬(¬P ∨ Q)) in n4_6a.
  replace (¬(¬P ∨ Q)) with (P ∧ ¬Q) in n4_6a.
  apply n4_6a.
  apply propositional_extensionality.
  apply n4_52a.
  replace (((P→Q)↔¬P∨Q)↔(¬(P→Q)↔¬(¬P∨Q))) with
      ((¬(P→Q)↔¬(¬P∨Q))↔((P→Q)↔¬P∨Q)) in Transp4_11a.
  apply propositional_extensionality.
  apply Transp4_11a.
  apply propositional_extensionality.
  specialize n4_21 with (¬(P→Q)↔¬(¬P∨Q))
      ((P→Q)↔(¬P∨Q)).
  intros n4_21a.
  apply n4_21a.
Qed.

Theorem n4_62 : ∀ P Q : Prop,
  (P → ¬Q) ↔ (¬P ∨ ¬Q).
  Proof. intros P Q.
    specialize n4_6 with P (¬Q).
    intros n4_6a.
    apply n4_6a.
Qed.

Theorem n4_63 : ∀ P Q : Prop,
  ¬(P → ¬Q) ↔ (P ∧ Q).
  Proof. intros P Q.
    specialize n4_62 with P Q.
    intros n4_62a.
    specialize Transp4_11 with (P → ¬Q) (¬P ∨ ¬Q).
    intros Transp4_11a.
```

```
    specialize n4_5 with P Q.
    intros n4_5a.
    replace (¬(¬P ∨ ¬Q)) with (P ∧ Q) in Transp4_11a.
    replace ((P → ¬Q) ↔ ¬P ∨ ¬Q) with
        ((¬(P → ¬Q) ↔ P ∧ Q)) in n4_62a.
    apply n4_62a.
    replace (((P→¬Q)↔¬P∨¬Q)↔(¬(P→¬Q)↔P∧Q)) with
        ((¬(P→¬Q)↔P∧Q)↔((P→¬Q)↔¬P∨¬Q)) in Transp4_11a.
    apply propositional_extensionality.
    apply Transp4_11a.
    specialize n4_21 with
        (¬(P → ¬Q) ↔ P ∧ Q) ((P → ¬Q) ↔ ¬P ∨ ¬Q).
    intros n4_21a.
    apply propositional_extensionality.
    apply n4_21a.
    apply propositional_extensionality.
    apply n4_5a.
Qed.

Theorem n4_64 : ∀ P Q : Prop,
  (¬P → Q) ↔ (P ∨ Q).
  Proof. intros P Q.
    specialize n2_54 with P Q.
    intros n2_54a.
    specialize n2_53 with P Q.
    intros n2_53a.
    Conj n2_54a n2_53a.
    split.
    apply n2_54a.
    apply n2_53a.
    Equiv H.
    apply H.
    apply Equiv4_01.
Qed.

Theorem n4_65 : ∀ P Q : Prop,
  ¬(¬P → Q) ↔ (¬P ∧ ¬Q).
  Proof. intros P Q.
  specialize n4_64 with P Q.
  intros n4_64a.
  specialize Transp4_11 with(¬P → Q) (P ∨ Q).
  intros Transp4_11a.
```

```
  specialize n4_56 with P Q.
  intros n4_56a.
  replace (((¬P→Q)↔P∨Q)↔(¬(¬P→Q)↔¬(P∨Q))) with
      ((¬(¬P→Q)↔¬(P∨Q))↔((¬P→Q)↔P∨Q)) in Transp4_11a.
  replace ((¬P → Q) ↔ P ∨ Q) with
      (¬(¬P → Q) ↔ ¬(P ∨ Q)) in n4_64a.
  replace (¬(P ∨ Q)) with (¬P ∧ ¬Q) in n4_64a.
  apply n4_64a.
  apply propositional_extensionality.
  apply n4_56a.
  apply propositional_extensionality.
  apply Transp4_11a.
  apply propositional_extensionality.
  specialize n4_21 with (¬(¬P → Q)↔¬(P ∨ Q))
      ((¬P → Q)↔(P ∨ Q)).
  intros n4_21a.
  apply n4_21a.
Qed.

Theorem n4_66 : ∀ P Q : Prop,
  (¬P → ¬Q) ↔ (P ∨ ¬Q).
  Proof. intros P Q.
  specialize n4_64 with P (¬Q).
  intros n4_64a.
  apply n4_64a.
Qed.

Theorem n4_67 : ∀ P Q : Prop,
  ¬(¬P → ¬Q) ↔ (¬P ∧ Q).
  Proof. intros P Q.
  specialize n4_66 with P Q.
  intros n4_66a.
  specialize Transp4_11 with (¬P → ¬Q) (P ∨ ¬Q).
  intros Transp4_11a.
  replace ((¬P → ¬Q) ↔ P ∨ ¬Q) with
      (¬(¬P → ¬Q) ↔ ¬(P ∨ ¬Q)) in n4_66a.
  specialize n4_54 with P Q.
  intros n4_54a.
  replace (¬(P ∨ ¬Q)) with (¬P ∧ Q) in n4_66a.
  apply n4_66a.
  apply propositional_extensionality.
  apply n4_54a.
```

```
  replace (((¬P→¬Q)↔P∨¬Q)↔(¬(¬P→¬Q)↔¬(P∨¬Q))) with
      ((¬(¬P→¬Q)↔¬(P∨¬Q))↔((¬P→¬Q)↔P∨¬Q)) in Transp4_11a.
  apply propositional_extensionality.
  apply Transp4_11a.
  apply propositional_extensionality.
  specialize n4_21 with (¬(¬P → ¬Q)↔¬(P ∨ ¬Q))
     ((¬P → ¬Q)↔(P ∨ ¬Q)).
  intros n4_21a.
  apply n4_21a.
Qed.

  (*Return to this proof.*)
  (*We did get one half of the ↔.*)
Theorem n4_7 : ∀ P Q : Prop,
  (P → Q) ↔ (P → (P ∧ Q)).
  Proof. intros P Q.
  specialize Comp3_43 with P P Q.
  intros Comp3_43a.
  specialize Exp3_3 with
      (P → P) (P → Q) (P → P ∧ Q).
  intros Exp3_3a.
  MP Exp3_3a Comp3_43a.
  specialize Id2_08 with P.
  intros Id2_08a.
  MP Exp3_3a Id2_08a.
  specialize Simp3_27 with P Q.
  intros Simp3_27a.
  specialize Syll2_05 with P (P ∧ Q) Q.
  intros Syll2_05a.
  MP Syll2_05a Simp3_27a.
  clear Id2_08a. clear Comp3_43a. clear Simp3_27a.
  Conj Syll2_05a Exp3_3a.
  split.
  apply Exp3_3a.
  apply Syll2_05a.
  Equiv H.
  apply H.
  apply Equiv4_01.
Qed.

Theorem n4_71 : ∀ P Q : Prop,
  (P → Q) ↔ (P ↔ (P ∧ Q)).
```

```
  Proof. intros P Q.
  specialize n4_7 with P Q.
  intros n4_7a.
  specialize n3_21 with (P→(P∧Q)) ((P∧Q)→P).
  intros n3_21a.
  replace ((P → P ∧ Q) ∧ (P ∧ Q → P)) with
      (P↔(P ∧ Q)) in n3_21a.
  specialize Simp3_26 with P Q.
  intros Simp3_26a.
  MP n3_21a Simp3_26a.
  specialize Simp3_26 with (P→(P∧Q)) ((P∧Q)→P).
  intros Simp3_26b.
  replace ((P → P ∧ Q) ∧ (P ∧ Q → P)) with
      (P↔(P ∧ Q)) in Simp3_26b. clear Simp3_26a.
  Conj n3_21a Simp3_26b.
  split.
  apply n3_21a.
  apply Simp3_26b.
  Equiv H.
  clear n3_21a. clear Simp3_26b.
  Conj n4_7a H.
  split.
  apply n4_7a.
  apply H.
  specialize n4_22 with (P → Q) (P → P ∧ Q) (P ↔ P ∧ Q).
  intros n4_22a.
  MP n4_22a H0.
  apply n4_22a.
  apply Equiv4_01.
  apply Equiv4_01.
  apply Equiv4_01.
Qed.

Theorem n4_72 : ∀ P Q : Prop,
  (P → Q) ↔ (Q ↔ (P ∨ Q)).
  Proof. intros P Q.
  specialize Transp4_1 with P Q.
  intros Transp4_1a.
  specialize n4_71 with (¬Q) (¬P).
  intros n4_71a.
  Conj Transp4_1a n4_71a.
  split.
```

```
apply Transp4_1a.
apply n4_71a.
specialize n4_22 with
    (P→Q) (¬Q→¬P) (¬Q↔¬Q ∧ ¬P).
intros n4_22a.
MP n4_22a H.
specialize n4_21 with (¬Q) (¬Q ∧ ¬P).
intros n4_21a.
Conj n4_22a n4_21a.
split.
apply n4_22a.
apply n4_21a.
specialize n4_22 with
    (P→Q) (¬Q ↔ ¬Q ∧ ¬P) (¬Q ∧ ¬P ↔ ¬Q).
intros n4_22b.
MP n4_22b H0.
specialize n4_12 with (¬Q ∧ ¬P) (Q).
intros n4_12a.
Conj n4_22b n4_12a.
split.
apply n4_22b.
apply n4_12a.
specialize n4_22 with
    (P → Q) ((¬Q ∧ ¬P) ↔ ¬Q) (Q ↔ ¬(¬Q ∧ ¬P)).
intros n4_22c.
MP n4_22b H0.
specialize n4_57 with Q P.
intros n4_57a.
replace (¬(¬Q ∧ ¬P)) with (Q ∨ P) in n4_22c.
specialize n4_31 with P Q.
intros n4_31a.
replace (Q ∨ P) with (P ∨ Q) in n4_22c.
apply n4_22c.
apply propositional_extensionality.
apply n4_31a.
apply propositional_extensionality.
replace (¬(¬Q ∧ ¬P) ↔ Q ∨ P) with
    (Q ∨ P ↔¬(¬Q ∧ ¬P)) in n4_57a.
apply n4_57a.
apply propositional_extensionality.
specialize n4_21 with (Q ∨ P) (¬(¬Q ∧ ¬P)).
intros n4_21b.
```

```
  apply n4_21b.
Qed.

Theorem n4_73 : ∀ P Q : Prop,
  Q → (P ↔ (P ∧ Q)).
  Proof. intros P Q.
  specialize Simp2_02 with P Q.
  intros Simp2_02a.
  specialize n4_71 with P Q.
  intros n4_71a.
  replace ((P → Q) ↔ (P ↔ P ∧ Q)) with
      (((P→Q)→(P↔P∧Q))∧((P↔P∧Q)→(P→Q))) in n4_71a.
  specialize Simp3_26 with
      ((P → Q) → P ↔ P ∧ Q) (P ↔ P ∧ Q → P → Q).
  intros Simp3_26a.
  MP Simp3_26a n4_71a.
  Syll Simp2_02a Simp3_26a Sa.
  apply Sa.
  apply Equiv4_01.
Qed.

Theorem n4_74 : ∀ P Q : Prop,
  ¬P → (Q ↔ (P ∨ Q)).
  Proof. intros P Q.
  specialize n2_21 with P Q.
  intros n2_21a.
  specialize n4_72 with P Q.
  intros n4_72a.
  replace (P → Q) with (Q ↔ P ∨ Q) in n2_21a.
  apply n2_21a.
  apply propositional_extensionality.
  replace ((P → Q) ↔ (Q ↔ P ∨ Q)) with
      ((Q ↔ P ∨ Q) ↔ (P → Q)) in n4_72a.
  apply n4_72a.
  apply propositional_extensionality.
  specialize n4_21 with (Q↔(P ∨ Q)) (P → Q).
  intros n4_21a.
  apply n4_21a.
Qed.

Theorem n4_76 : ∀ P Q R : Prop,
  ((P → Q) ∧ (P → R)) ↔ (P → (Q ∧ R)).
```

```
  Proof. intros P Q R.
  specialize n4_41 with (¬P) Q R.
  intros n4_41a.
  replace (¬P ∨ Q) with (P→Q) in n4_41a.
  replace (¬P ∨ R) with (P→R) in n4_41a.
  replace (¬P ∨ Q ∧ R) with (P → Q ∧ R) in n4_41a.
  replace ((P → Q ∧ R) ↔ (P → Q) ∧ (P → R)) with
      ((P → Q) ∧ (P → R) ↔ (P → Q ∧ R)) in n4_41a.
  apply n4_41a.
  apply propositional_extensionality.
  specialize n4_21 with ((P → Q) ∧ (P → R)) (P → Q ∧ R).
  intros n4_21a.
  apply n4_21a.
  apply Impl1_01.
  apply Impl1_01.
  apply Impl1_01.
Qed.

Theorem n4_77 : ∀ P Q R : Prop,
  ((Q → P) ∧ (R → P)) ↔ ((Q ∨ R) → P).
  Proof. intros P Q R.
  specialize n3_44 with P Q R.
  intros n3_44a.
  specialize n2_2 with Q R.
  intros n2_2a.
  specialize Add1_3 with Q R.
  intros Add1_3a.
  specialize Syll2_06 with Q (Q ∨ R) P.
  intros Syll2_06a.
  MP Syll2_06a n2_2a.
  specialize Syll2_06 with R (Q ∨ R) P.
  intros Syll2_06b.
  MP Syll2_06b Add1_3a.
  Conj Syll2_06a Syll2_06b.
  split.
  apply Syll2_06a.
  apply Syll2_06b.
  specialize Comp3_43 with ((Q ∨ R)→P)
    (Q→P) (R→P).
  intros Comp3_43a.
  MP Comp3_43a H.
  clear n2_2a. clear Add1_3a. clear H.
```

```
    clear Syll2_06a. clear Syll2_06b.
  Conj n3_44a Comp3_43a.
  split.
  apply n3_44a.
  apply Comp3_43a.
  Equiv H.
  apply H.
  apply Equiv4_01.
Qed.

Theorem n4_78 : ∀ P Q R : Prop,
  ((P → Q) ∨ (P → R)) ↔ (P → (Q ∨ R)).
  Proof. intros P Q R.
  specialize n4_2 with ((P→Q) ∨ (P → R)).
  intros n4_2a.
  replace (((P→Q)∨(P→R))↔((P→Q)∨(P→R))) with
      (((P → Q) ∨ (P → R))↔((¬P ∨ Q) ∨ ¬P ∨ R)) in n4_2a.
  specialize n4_33 with (¬P) Q (¬P ∨ R).
  intros n4_33a.
  replace ((¬P ∨ Q) ∨ ¬P ∨ R) with
      (¬P ∨ Q ∨ ¬P ∨ R) in n4_2a.
  specialize n4_31 with (¬P) Q.
  intros n4_31a.
  specialize n4_37 with (¬P∨Q) (Q ∨ ¬P) R.
  intros n4_37a.
  MP n4_37a n4_31a.
  replace (Q ∨ ¬P ∨ R) with
      ((Q ∨ ¬P) ∨ R) in n4_2a.
  replace ((Q ∨ ¬P) ∨ R) with
      ((¬P ∨ Q) ∨ R) in n4_2a.
  specialize n4_33 with (¬P) (¬P∨Q) R.
  intros n4_33b.
  replace (¬P ∨ (¬P ∨ Q) ∨ R) with
      ((¬P ∨ (¬P ∨ Q)) ∨ R) in n4_2a.
  specialize n4_25 with (¬P).
  intros n4_25a.
  specialize n4_37 with
      (¬P) (¬P ∨ ¬P) (Q ∨ R).
  intros n4_37b.
  MP n4_37b n4_25a.
  replace (¬P ∨ ¬P ∨ Q) with
      ((¬P ∨ ¬P) ∨ Q) in n4_2a.
```

```
  replace (((¬P ∨ ¬P) ∨ Q) ∨ R) with
      ((¬P ∨ ¬P) ∨ Q ∨ R) in n4_2a.
  replace ((¬P ∨ ¬P) ∨ Q ∨ R) with
      ((¬P) ∨ (Q ∨ R)) in n4_2a.
  replace (¬P ∨ Q ∨ R) with
      (P → (Q ∨ R)) in n4_2a.
  apply n4_2a.
  apply Impl1_01.
  apply propositional_extensionality.
  apply n4_37b.
  apply Abb2_33.
  replace ((¬P ∨ ¬P) ∨ Q) with (¬P ∨ ¬P ∨ Q).
  reflexivity.
  apply Abb2_33.
  replace ((¬P ∨ ¬P ∨ Q) ∨ R) with
      (¬P ∨ (¬P ∨ Q) ∨ R).
  reflexivity.
  apply propositional_extensionality.
  apply n4_33b.
  apply propositional_extensionality.
  apply n4_37a.
  replace ((Q ∨ ¬P) ∨ R) with (Q ∨ ¬P ∨ R).
  reflexivity.
  apply Abb2_33.
  apply propositional_extensionality.
  apply n4_33a.
  rewrite <- Impl1_01.
  rewrite <- Impl1_01.
  reflexivity.
Qed.

Theorem n4_79 : ∀ P Q R : Prop,
  ((Q → P) ∨ (R → P)) ↔ ((Q ∧ R) → P).
  Proof. intros P Q R.
    specialize Transp4_1 with Q P.
    intros Transp4_1a.
    specialize Transp4_1 with R P.
    intros Transp4_1b.
    Conj Transp4_1a Transp4_1b.
    split.
    apply Transp4_1a.
    apply Transp4_1b.
```

```
specialize n4_39 with
    (Q→P) (R→P) (¬P→¬Q) (¬P→¬R).
intros n4_39a.
MP n4_39a H.
specialize n4_78 with (¬P) (¬Q) (¬R).
intros n4_78a.
rewrite Equiv4_01 in n4_78a.
specialize Simp3_26 with
  (((¬P→¬Q)∨(¬P→¬R))→(¬P→(¬Q∨¬R)))
  ((¬P→(¬Q∨¬R))→((¬P→¬Q)∨(¬P→¬R))).
intros Simp3_26a.
MP Simp3_26a n4_78a.
specialize Transp2_15 with P (¬Q∨¬R).
intros Transp2_15a.
specialize Simp3_27 with
  (((¬P→¬Q)∨(¬P→¬R))→(¬P→(¬Q∨¬R)))
  ((¬P→(¬Q∨¬R))→((¬P→¬Q)∨(¬P→¬R))).
intros Simp3_27a.
MP Simp3_27a n4_78a.
specialize Transp2_15 with (¬Q∨¬R) P.
intros Transp2_15b.
specialize Syll2_06 with ((¬P→¬Q)∨(¬P→¬R))
  (¬P→(¬Q∨¬R)) (¬(¬Q∨¬R)→P).
intros Syll2_06a.
MP Syll2_06a Simp3_26a.
MP Syll2_06a Transp2_15a.
specialize Syll2_06 with (¬(¬Q∨¬R)→P)
  (¬P→(¬Q∨¬R)) ((¬P→¬Q)∨(¬P→¬R)).
intros Syll2_06b.
MP Syll2_06b Trans2_15b.
MP Syll2_06b Simp3_27a.
Conj Syll2_06a Syll2_06b.
split.
apply Syll2_06a.
apply Syll2_06b.
Equiv H0.
clear Transp4_1a. clear Transp4_1b. clear H.
  clear Simp3_26a. clear Syll2_06b. clear n4_78a.
  clear Transp2_15a. clear Simp3_27a.
  clear Transp2_15b. clear Syll2_06a.
Conj n4_39a H0.
split.
```

```
    apply n4_39a.
    apply H0.
    specialize n4_22 with ((Q→P)∨(R→P))
      ((¬P→¬Q)∨(¬P→¬R)) (¬(¬Q∨¬R)→P).
    intros n4_22a.
    MP n4_22a H.
    specialize n4_2 with (¬(¬Q∨¬R)→P).
    intros n4_2a.
    Conj n4_22a n4_2a.
    split.
    apply n4_22a.
    apply n4_2a.
    specialize n4_22 with ((Q→P)∨(R→P))
    (¬(¬Q∨¬R)→P) (¬(¬Q∨¬R)→P).
    intros n4_22b.
    MP n4_22b H1.
    rewrite <- Prod3_01 in n4_22b.
    apply n4_22b.
    apply Equiv4_01.
Qed.

Theorem n4_8 : ∀ P : Prop,
  (P → ¬P) ↔ ¬P.
  Proof. intros P.
    specialize Abs2_01 with P.
    intros Abs2_01a.
    specialize  Simp2_02 with P (¬P).
    intros Simp2_02a.
    Conj Abs2_01a Simp2_02a.
    split.
    apply Abs2_01a.
    apply Simp2_02a.
    Equiv H.
    apply H.
    apply Equiv4_01.
Qed.

Theorem n4_81 : ∀ P : Prop,
  (¬P → P) ↔ P.
  Proof. intros P.
    specialize n2_18 with P.
    intros n2_18a.
```

```
    specialize  Simp2_02 with (¬P) P.
    intros Simp2_02a.
    Conj n2_18a Simp2_02a.
    split.
    apply n2_18a.
    apply Simp2_02a.
    Equiv H.
    apply H.
    apply Equiv4_01.
Qed.

Theorem n4_82 : ∀ P Q : Prop,
  ((P → Q) ∧ (P → ¬Q)) ↔ ¬P.
  Proof. intros P Q.
    specialize n2_65 with P Q.
    intros n2_65a.
    specialize Imp3_31 with (P→Q) (P→¬Q) (¬P).
    intros Imp3_31a.
    MP Imp3_31a n2_65a.
    specialize n2_21 with P Q.
    intros n2_21a.
    specialize n2_21 with P (¬Q).
    intros n2_21b.
    Conj n2_21a n2_21b.
    split.
    apply n2_21a.
    apply n2_21b.
    specialize Comp3_43 with (¬P) (P→Q) (P→¬Q).
    intros Comp3_43a.
    MP Comp3_43a H.
    clear n2_65a. clear n2_21a.
      clear n2_21b. clear H.
    Conj Imp3_31a Comp3_43a.
    split.
    apply Imp3_31a.
    apply Comp3_43a.
    Equiv H.
    apply H.
    apply Equiv4_01.
Qed.

Theorem n4_83 : ∀ P Q : Prop,
```

```
  ((P → Q) ∧ (¬P → Q)) ↔ Q.
  Proof. intros P Q.
  specialize n2_61 with P Q.
  intros n2_61a.
  specialize Imp3_31 with (P→Q) (¬P→Q) (Q).
  intros Imp3_31a.
  MP Imp3_31a n2_61a.
  specialize Simp2_02 with P Q.
  intros Simp2_02a.
  specialize Simp2_02 with (¬P) Q.
  intros Simp2_02b.
  Conj Simp2_02a Simp2_02b.
  split.
  apply Simp2_02a.
  apply Simp2_02b.
  specialize Comp3_43 with Q (P→Q) (¬P→Q).
  intros Comp3_43a.
  MP Comp3_43a H.
  clear n2_61a. clear Simp2_02a.
    clear Simp2_02b. clear H.
  Conj Imp3_31a Comp3_43a.
  split.
  apply Imp3_31a.
  apply Comp3_43a.
  Equiv H.
  apply H.
  apply Equiv4_01.
Qed.

Theorem n4_84 : ∀ P Q R : Prop,
  (P ↔ Q) → ((P → R) ↔ (Q → R)).
  Proof. intros P Q R.
    specialize Syll2_06 with P Q R.
    intros Syll2_06a.
    specialize Syll2_06 with Q P R.
    intros Syll2_06b.
    Conj Syll2_06a Syll2_06b.
    split.
    apply Syll2_06a.
    apply Syll2_06b.
    specialize n3_47 with
        (P→Q) (Q→P) ((Q→R)→P→R) ((P→R)→Q→R).
```

```
    intros n3_47a.
    MP n3_47a H.
    replace ((P→Q) ∧ (Q → P)) with
        (P↔Q) in n3_47a.
    replace (((Q→R)→P→R)∧((P→R)→Q→R)) with
        ((Q → R) ↔ (P → R)) in n3_47a.
    replace ((Q → R) ↔ (P → R)) with
        ((P→ R) ↔ (Q → R)) in n3_47a.
    apply n3_47a.
    apply propositional_extensionality.
    specialize n4_21 with (P→R) (Q→R).
    intros n4_21a.
    apply n4_21a.
    apply Equiv4_01.
    apply Equiv4_01.
Qed.

Theorem n4_85 : ∀ P Q R : Prop,
  (P ↔ Q) → ((R → P) ↔ (R → Q)).
  Proof. intros P Q R.
  specialize Syll2_05 with R P Q.
  intros Syll2_05a.
  specialize Syll2_05 with R Q P.
  intros Syll2_05b.
  Conj Syll2_05a Syll2_05b.
  split.
  apply Syll2_05a.
  apply Syll2_05b.
  specialize n3_47 with
      (P→Q) (Q→P) ((R→P)→R→Q) ((R→Q)→R→P).
  intros n3_47a.
  MP n3_47a H.
  replace ((P→Q) ∧ (Q → P)) with (P↔Q) in n3_47a.
  replace (((R→P)→R→Q)∧((R→Q)→R→P)) with
      ((R → P) ↔ (R → Q)) in n3_47a.
  apply n3_47a.
  apply Equiv4_01.
  apply Equiv4_01.
Qed.

Theorem n4_86 : ∀ P Q R : Prop,
  (P ↔ Q) → ((P ↔ R) ↔ (Q ↔ R)).
```

```
  Proof. intros P Q R.
  specialize n4_22 with  Q P R.
  intros n4_22a.
  specialize Exp3_3 with (Q↔P) (P↔R) (Q↔R).
  intros Exp3_3a. (*Not cited*)
  MP Exp3_3a n4_22a.
  specialize n4_22 with  P Q R.
  intros n4_22b.
  specialize Exp3_3 with (P↔Q) (Q↔R) (P↔R).
  intros Exp3_3b.
  MP Exp3_3b n4_22b.
  clear n4_22a. clear n4_22b.
  replace (Q↔P) with (P↔Q) in Exp3_3a.
  Conj Exp3_3a Exp3_3b.
  split.
  apply Exp3_3a.
  apply Exp3_3b.
  specialize Comp3_43 with (P↔Q)
      ((P↔R)→(Q↔R)) ((Q↔R)→(P↔R)).
  intros Comp3_43a. (*Not cited*)
  MP Comp3_43a H.
  replace (((P↔R)→(Q↔R))∧((Q↔R)→(P↔R)))
    with ((P↔R)↔(Q↔R)) in Comp3_43a.
  apply Comp3_43a.
  apply Equiv4_01.
  apply propositional_extensionality.
  specialize n4_21 with P Q.
  intros n4_21a.
  apply n4_21a.
Qed.

Theorem n4_87 : ∀ P Q R : Prop,
  (((P ∧ Q) → R) ↔ (P → Q → R)) ↔
      ((Q → (P → R)) ↔ (Q ∧ P → R)).
  Proof. intros P Q R.
  specialize Exp3_3 with P Q R.
  intros Exp3_3a.
  specialize Imp3_31 with P Q R.
  intros Imp3_31a.
  Conj Exp3_3a Imp3_31a.
  split.
  apply Exp3_3a.
```

```
apply Imp3_31a.
Equiv H.
specialize Exp3_3 with Q P R.
intros Exp3_3b.
specialize Imp3_31 with Q P R.
intros Imp3_31b.
Conj Exp3_3b Imp3_31b.
split.
apply Exp3_3b.
apply Imp3_31b.
Equiv H0.
specialize Comm2_04 with P Q R.
intros Comm2_04a.
specialize Comm2_04 with Q P R.
intros Comm2_04b.
Conj Comm2_04a Comm2_04b.
split.
apply Comm2_04a.
apply Comm2_04b.
Equiv H1.
clear Exp3_3a. clear Imp3_31a. clear Exp3_3b.
    clear Imp3_31b. clear Comm2_04a.
    clear Comm2_04b.
replace (P∧Q→R) with (P → Q → R).
replace (Q∧P→R) with (Q → P → R).
replace (Q→P→R) with (P → Q → R).
specialize n4_2 with
    ((P → Q → R) ↔ (P → Q → R)).
intros n4_2a.
apply n4_2a.
apply propositional_extensionality.
apply H1.
replace (Q→P→R) with (Q∧P→R).
reflexivity.
apply propositional_extensionality.
apply H0.
replace (P→Q→R) with (P∧Q→R).
reflexivity.
apply propositional_extensionality.
apply H.
apply Equiv4_01.
apply Equiv4_01.
```

```
  apply Equiv4_01.
Qed.

End No4.

Module No5.

Import No1.
Import No2.
Import No3.
Import No4.

Theorem n5_1 : ∀ P Q : Prop,
  (P ∧ Q) → (P ↔ Q).
  Proof. intros P Q.
  specialize n3_4 with P Q.
  intros n3_4a.
  specialize n3_4 with Q P.
  intros n3_4b.
  specialize n3_22 with P Q.
  intros n3_22a.
  Syll n3_22a n3_4b Sa.
  clear n3_22a. clear n3_4b.
  Conj n3_4a Sa.
  split.
  apply n3_4a.
  apply Sa.
  specialize n4_76 with (P∧Q) (P→Q) (Q→P).
  intros n4_76a. (*Not cited*)
  replace ((P∧Q→P→Q)∧(P∧Q→Q→P)) with
      (P ∧ Q → (P → Q) ∧ (Q → P)) in H.
  replace ((P→Q)∧(Q→P)) with (P↔Q) in H.
  apply H.
  apply Equiv4_01.
  replace (P ∧ Q → (P → Q) ∧ (Q → P)) with
      ((P ∧ Q → P → Q) ∧ (P ∧ Q → Q → P)).
  reflexivity.
  apply propositional_extensionality.
  apply n4_76a.
Qed.

Theorem n5_11 : ∀ P Q : Prop,
```

```
  (P → Q) ∨ (¬P → Q).
  Proof. intros P Q.
  specialize n2_5 with P Q.
  intros n2_5a.
  specialize n2_54 with (P → Q) (¬P → Q).
  intros n2_54a.
  MP n2_54a n2_5a.
  apply n2_54a.
Qed.
  (*The proof sketch cites n2_51,
      but this may be a misprint.*)

Theorem n5_12 : ∀ P Q : Prop,
  (P → Q) ∨ (P → ¬Q).
  Proof. intros P Q.
  specialize n2_51 with P Q.
  intros n2_51a.
  specialize n2_54 with ((P → Q)) (P → ¬Q).
  intros n2_54a.
  MP n2_54a n2_5a.
  apply n2_54a.
Qed.
  (*The proof sketch cites n2_52,
      but this may be a misprint.*)

Theorem n5_13 : ∀ P Q : Prop,
  (P → Q) ∨ (Q → P).
  Proof. intros P Q.
  specialize n2_521 with P Q.
  intros n2_521a.
  replace (¬(P → Q) → Q → P) with
      (¬¬(P → Q) ∨ (Q → P)) in n2_521a.
  replace (¬¬(P→Q)) with (P→Q) in n2_521a.
  apply n2_521a.
  apply propositional_extensionality.
  specialize n4_13 with (P→Q).
  intros n4_13a. (*Not cited*)
  apply n4_13a.
  rewrite <- Impl1_01.
  reflexivity.
Qed.
```

```
Theorem n5_14 : ∀ P Q R : Prop,
  (P → Q) ∨ (Q → R).
  Proof. intros P Q R.
  specialize Simp2_02 with P Q.
  intros Simp2_02a.
  specialize Transp2_16 with Q (P→Q).
  intros Transp2_16a.
  MP Transp2_16a Simp2_02a.
  specialize n2_21 with Q R.
  intros n2_21a.
  Syll Transp2_16a n2_21a Sa.
  replace (¬(P→Q)→(Q→R)) with
      (¬¬(P→Q)∨(Q→R)) in Sa.
  replace (¬¬(P→Q)) with (P→Q) in Sa.
  apply Sa.
  apply propositional_extensionality.
  specialize n4_13 with (P→Q).
  intros n4_13a.
  apply n4_13a.
  rewrite <- Impl1_01.
  reflexivity.
Qed.

Theorem n5_15 : ∀ P Q : Prop,
  (P ↔ Q) ∨ (P ↔ ¬Q).
  Proof. intros P Q.
  specialize n4_61 with P Q.
  intros n4_61a.
  replace (¬(P → Q) ↔ P ∧ ¬Q) with
      ((¬(P→Q)→P∧¬Q)∧((P∧¬Q)→¬(P→Q))) in n4_61a.
  specialize Simp3_26 with
      (¬(P → Q) → P ∧ ¬Q) ((P ∧ ¬Q) → ¬(P → Q)).
  intros Simp3_26a.
  MP Simp3_26a n4_61a.
  specialize n5_1 with P (¬Q).
  intros n5_1a.
  Syll Simp3_26a n5_1a Sa.
  specialize n2_54 with (P→Q) (P ↔ ¬Q).
  intros n2_54a.
  MP n2_54a Sa.
  specialize n4_61 with Q P.
  intros n4_61b.
```

```
replace ((¬(Q → P)) ↔ (Q ∧ ¬P)) with
    (((¬(Q→P))→(Q∧¬P))∧((Q∧¬P)→(¬(Q→P)))) in n4_61b.
specialize Simp3_26 with
    (¬(Q → P)→ (Q ∧ ¬P)) ((Q ∧ ¬P)→ (¬(Q → P))).
intros Simp3_26b.
MP Simp3_26b n4_61b.
specialize n5_1 with Q (¬P).
intros n5_1b.
Syll Simp3_26b n5_1b Sb.
specialize n4_12 with P Q.
intros n4_12a.
replace (Q↔¬P) with (P↔¬Q) in Sb.
specialize n2_54 with (Q→P) (P↔¬Q).
intros n2_54b.
MP n2_54b Sb.
clear n4_61a. clear Simp3_26a. clear n5_1a.
    clear n2_54a. clear n4_61b. clear Simp3_26b.
    clear n5_1b. clear n4_12a. clear n2_54b.
replace (¬(P → Q) → P ↔ ¬Q) with
    (¬¬(P → Q) ∨ (P ↔ ¬Q)) in Sa.
replace (¬¬(P→Q)) with (P→Q) in Sa.
replace (¬(Q → P) → (P ↔ ¬Q)) with
    (¬¬(Q → P) ∨ (P ↔ ¬Q)) in Sb.
replace (¬¬(Q→P)) with (Q→P) in Sb.
Conj Sa Sb.
split.
apply Sa.
apply Sb.
specialize n4_41 with (P↔¬Q) (P→Q) (Q→P).
intros n4_41a.
replace ((P → Q) ∨ (P ↔ ¬Q)) with
    ((P ↔ ¬Q) ∨ (P → Q)) in H.
replace ((Q → P) ∨ (P ↔ ¬Q)) with
    ((P ↔ ¬Q) ∨ (Q → P)) in H.
replace (((P↔¬Q)∨(P→Q))∧((P↔¬Q)∨(Q→P))) with
    ((P ↔ ¬Q) ∨ (P → Q) ∧ (Q → P)) in H.
replace ((P→Q) ∧ (Q → P)) with (P↔Q) in H.
replace ((P ↔ ¬Q) ∨ (P ↔ Q)) with
    ((P ↔ Q) ∨ (P ↔ ¬Q)) in H.
apply H.
apply propositional_extensionality.
apply n4_31.
```

```
  apply Equiv4_01.
  apply propositional_extensionality.
  apply n4_41a.
  apply propositional_extensionality.
  apply n4_31.
  apply propositional_extensionality.
  apply n4_31.
  apply propositional_extensionality.
  specialize n4_13 with (Q→P).
  intros n4_13a.
  apply n4_13a.
  rewrite <- Impl1_01.
  reflexivity.
  apply propositional_extensionality.
  specialize n4_13 with (P→Q).
  intros n4_13b.
  apply n4_13b.
  rewrite <- Impl1_01.
  reflexivity.
  apply propositional_extensionality.
  apply n4_12a.
  apply Equiv4_01.
  apply Equiv4_01.
Qed.

Theorem n5_16 : ∀ P Q : Prop,
  ¬((P ↔ Q) ∧ (P ↔ ¬Q)).
  Proof. intros P Q.
  specialize Simp3_26 with ((P→Q)∧ (P → ¬Q)) (Q→P).
  intros Simp3_26a.
  specialize Id2_08 with ((P ↔ Q) ∧ (P → ¬Q)).
  intros Id2_08a.
  replace (((P → Q) ∧ (P → ¬Q)) ∧ (Q → P)) with
      ((P→Q)∧((P→¬Q)∧(Q→P))) in Simp3_26a.
  replace ((P → ¬Q) ∧ (Q → P)) with
      ((Q → P) ∧ (P → ¬Q)) in Simp3_26a.
  replace ((P→Q) ∧ (Q → P)∧ (P → ¬Q)) with
      (((P→Q) ∧ (Q → P)) ∧ (P → ¬Q)) in Simp3_26a.
  replace ((P → Q) ∧ (Q → P)) with
      (P↔Q) in Simp3_26a.
  Syll Id2_08a Simp3_26a Sa.
  specialize n4_82 with P Q.
```

```
intros n4_82a.
replace ((P → Q) ∧ (P → ¬Q)) with (¬P) in Sa.
specialize Simp3_27 with
    (P→Q) ((Q→P)∧ (P → ¬Q)).
intros Simp3_27a.
replace ((P→Q) ∧ (Q → P)∧ (P → ¬Q)) with
    (((P→Q) ∧ (Q → P)) ∧ (P → ¬Q)) in Simp3_27a.
replace ((P → Q) ∧ (Q → P)) with
    (P↔Q) in Simp3_27a.
specialize Syll3_33 with Q P (¬Q).
intros Syll3_33a.
Syll Simp3_27a Syll2_06a Sb.
specialize Abs2_01 with Q.
intros Abs2_01a.
Syll Sb Abs2_01a Sc.
clear Sb. clear Simp3_26a. clear Id2_08a.
    clear n4_82a. clear Simp3_27a. clear Syll3_33a.
    clear Abs2_01a.
Conj Sa Sc.
split.
apply Sa.
apply Sc.
specialize Comp3_43 with
    ((P ↔ Q) ∧ (P → ¬Q)) (¬P) (¬Q).
intros Comp3_43a.
MP Comp3_43a H.
specialize n4_65 with Q P.
intros n4_65a.
replace (¬Q ∧ ¬P) with (¬P ∧ ¬Q) in n4_65a.
replace (¬P ∧ ¬Q) with
    (¬(¬Q→P)) in Comp3_43a.
specialize Exp3_3 with
    (P↔Q) (P→¬Q) (¬(¬Q→P)).
intros Exp3_3a.
MP Exp3_3a Comp3_43a.
replace ((P→¬Q)→¬(¬Q→P)) with
    (¬(P→¬Q)∨¬(¬Q→P)) in Exp3_3a.
specialize n4_51 with (P→¬Q) (¬Q→P).
intros n4_51a.
replace (¬(P → ¬Q) ∨ ¬(¬Q → P)) with
    (¬((P → ¬Q) ∧ (¬Q → P))) in Exp3_3a.
replace ((P→¬Q) ∧ (¬Q → P)) with
```

```
    (P↔¬Q) in Exp3_3a.
replace ((P↔Q)→¬(P↔¬Q)) with
    (¬(P↔Q)∨¬(P↔¬Q)) in Exp3_3a.
specialize n4_51 with (P↔Q) (P↔¬Q).
intros n4_51b.
replace (¬(P ↔ Q) ∨ ¬(P ↔ ¬Q)) with
    (¬((P ↔ Q) ∧ (P ↔ ¬Q))) in Exp3_3a.
apply Exp3_3a.
apply propositional_extensionality.
apply n4_51b.
rewrite <- Impl1_01.
reflexivity.
apply Equiv4_01.
apply propositional_extensionality.
apply n4_51a.
rewrite <- Impl1_01.
reflexivity.
apply propositional_extensionality.
apply n4_65a.
apply propositional_extensionality.
specialize n4_3 with (¬P) (¬Q).
intros n4_3a.
apply n4_3a.
apply Equiv4_01.
apply propositional_extensionality.
specialize n4_32 with (P→Q) (Q→P) (P→¬Q).
intros n4_32a.
apply n4_32a.
replace (¬P) with ((P → Q) ∧ (P → ¬Q)).
reflexivity.
apply propositional_extensionality.
apply n4_82a.
apply Equiv4_01.
apply propositional_extensionality.
specialize n4_32 with (P→Q) (Q→P) (P→¬Q).
intros n4_32b.
apply n4_32b.
apply propositional_extensionality.
specialize n4_3 with (Q→P) (P→¬Q).
intros n4_3b.
apply n4_3b.
replace ((P → Q) ∧ (P → ¬Q) ∧ (Q → P)) with
```

```
      (((P → Q) ∧ (P → ¬Q)) ∧ (Q → P)).
  reflexivity.
  apply propositional_extensionality.
  specialize n4_32 with (P→Q) (P→¬Q) (Q→P).
  intros n4_32a.
  apply n4_32a.
Qed.

Theorem n5_17 : ∀ P Q : Prop,
  ((P ∨ Q) ∧ ¬(P ∧ Q)) ↔ (P ↔ ¬Q).
  Proof. intros P Q.
  specialize n4_64 with Q P.
  intros n4_64a.
  specialize n4_21 with (Q∨P) (¬Q→P).
  intros n4_21a.
  replace ((¬Q→P)↔(Q∨P)) with
      ((Q∨P)↔(¬Q→P)) in n4_64a.
  replace (Q∨P) with (P∨Q) in n4_64a.
  specialize n4_63 with P Q.
  intros n4_63a.
  replace (¬(P → ¬Q) ↔ P ∧ Q) with
      (P ∧ Q ↔ ¬(P → ¬Q)) in n4_63a.
  specialize Transp4_11 with (P∧Q) (¬(P→¬Q)).
  intros Transp4_11a.
  replace (¬¬(P→¬Q)) with
      (P→¬Q) in Transp4_11a.
  replace (P ∧ Q ↔ ¬(P → ¬Q)) with
      (¬(P ∧ Q) ↔ (P → ¬Q)) in n4_63a.
  clear Transp4_11a. clear n4_21a.
  Conj n4_64a n4_63a.
  split.
  apply n4_64a.
  apply n4_63a.
  specialize n4_38 with
      (P ∨ Q) (¬(P ∧ Q)) (¬Q → P) (P → ¬Q).
  intros n4_38a.
  MP n4_38a H.
  replace ((¬Q→P) ∧ (P → ¬Q)) with
      (¬Q↔P) in n4_38a.
  specialize n4_21 with P (¬Q).
  intros n4_21b.
  replace (¬Q↔P) with (P↔¬Q) in n4_38a.
```

```
  apply n4_38a.
  apply propositional_extensionality.
  apply n4_21b.
  apply Equiv4_01.
  replace (¬(P ∧ Q) ↔ (P → ¬Q)) with
      (P ∧ Q ↔ ¬(P → ¬Q)).
  reflexivity.
  apply propositional_extensionality.
  apply Transp4_11a.
  apply propositional_extensionality.
  specialize n4_13 with (P→¬Q).
  intros n4_13a.
  apply n4_13a.
  apply propositional_extensionality.
  specialize n4_21 with (P ∧ Q) (¬(P→¬Q)).
  intros n4_21b.
  apply n4_21b.
  apply propositional_extensionality.
  specialize n4_31 with P Q.
  intros n4_31a.
  apply n4_31a.
  apply propositional_extensionality.
  apply n4_21a.
Qed.

Theorem n5_18 : ∀ P Q : Prop,
  (P ↔ Q) ↔ ¬(P ↔ ¬Q).
  Proof. intros P Q.
  specialize n5_15 with P Q.
  intros n5_15a.
  specialize n5_16 with P Q.
  intros n5_16a.
  Conj n5_15a n5_16a.
  split.
  apply n5_15a.
  apply n5_16a.
  specialize n5_17 with (P↔Q) (P↔¬Q).
  intros n5_17a.
  replace ((P ↔ Q) ↔ ¬(P ↔ ¬Q)) with
      (((P↔Q)∨(P↔¬Q))∧¬((P↔Q)∧(P↔¬Q))).
  apply H.
  apply propositional_extensionality.
```

```
  apply n5_17a.
Qed.

Theorem n5_19 : ∀ P : Prop,
  ¬(P ↔ ¬P).
  Proof. intros P.
  specialize n5_18 with P P.
  intros n5_18a.
  specialize n4_2 with P.
  intros n4_2a.
  replace (¬(P↔¬P)) with (P↔P).
  apply n4_2a.
  apply propositional_extensionality.
  apply n5_18a.
Qed.

Theorem n5_21 : ∀ P Q : Prop,
  (¬P ∧ ¬Q) → (P ↔ Q).
  Proof. intros P Q.
  specialize n5_1 with (¬P) (¬Q).
  intros n5_1a.
  specialize Transp4_11 with P Q.
  intros Transp4_11a.
  replace (¬P↔¬Q) with (P↔Q) in n5_1a.
  apply n5_1a.
  apply propositional_extensionality.
  apply Transp4_11a.
Qed.

Theorem n5_22 : ∀ P Q : Prop,
  ¬(P ↔ Q) ↔ ((P ∧ ¬Q) ∨ (Q ∧ ¬P)).
  Proof. intros P Q.
  specialize n4_61 with P Q.
  intros n4_61a.
  specialize n4_61 with Q P.
  intros n4_61b.
  Conj n4_61a n4_61b.
  split.
  apply n4_61a.
  apply n4_61b.
  specialize n4_39 with
      (¬(P → Q)) (¬(Q → P)) (P ∧ ¬Q) (Q ∧ ¬P).
```

```
  intros n4_39a.
  MP n4_39a H.
  specialize n4_51 with (P→Q) (Q→P).
  intros n4_51a.
  replace (¬(P → Q) ∨ ¬(Q → P)) with
      (¬((P → Q) ∧ (Q → P))) in n4_39a.
  replace ((P → Q) ∧ (Q → P)) with
      (P↔Q) in n4_39a.
  apply n4_39a.
  apply Equiv4_01.
  apply propositional_extensionality.
  apply n4_51a.
Qed.

Theorem n5_23 : ∀ P Q : Prop,
  (P ↔ Q) ↔ ((P ∧ Q) ∨ (¬P ∧ ¬Q)).
  Proof. intros P Q.
  specialize n5_18 with P Q.
  intros n5_18a.
  specialize n5_22 with P (¬Q).
  intros n5_22a.
  Conj n5_18a n5_22a.
  split.
  apply n5_18a.
  apply n5_22a.
  specialize n4_22 with (P↔Q) (¬(P↔¬Q))
    (P ∧ ¬¬Q ∨ ¬Q ∧ ¬P).
  intros n4_22a.
  MP n4_22a H.
  replace (¬¬Q) with Q in n4_22a.
  replace (¬Q ∧ ¬P) with (¬P ∧ ¬Q) in n4_22a.
  apply n4_22a.
  apply propositional_extensionality.
  specialize n4_3 with (¬P) (¬Q).
  intros n4_3a.
  apply n4_3a. (*with (¬P) (¬Q)*)
  apply propositional_extensionality.
  specialize n4_13 with Q.
  intros n4_13a.
  apply n4_13a.
Qed.
  (*The proof sketch in Principia offers n4_36.
```

```
    This seems to be a misprint. We used n4_3.*)

Theorem n5_24 : ∀ P Q : Prop,
  ¬((P ∧ Q) ∨ (¬P ∧ ¬Q)) ↔ ((P ∧ ¬Q) ∨ (Q ∧ ¬P)).
  Proof. intros P Q.
  specialize n5_22 with P Q.
  intros n5_22a.
  specialize n5_23 with P Q.
  intros n5_23a.
  replace ((P↔Q)↔((P∧ Q) ∨(¬P ∧ ¬Q))) with
      ((¬(P↔Q)↔¬((P∧ Q)∨(¬P ∧ ¬Q)))) in n5_23a.
  replace (¬(P↔Q)) with
      (¬((P ∧ Q) ∨ (¬P ∧ ¬Q))) in n5_22a.
  apply n5_22a.
  replace (¬((P ∧ Q)∨(¬P ∧ ¬Q))) with (¬(P↔Q)).
  reflexivity.
  apply propositional_extensionality.
  apply n5_23a.
  replace (¬(P ↔ Q) ↔ ¬(P ∧ Q ∨ ¬P ∧ ¬Q)) with
      ((P ↔ Q) ↔ P ∧ Q ∨ ¬P ∧ ¬Q).
  reflexivity.
  specialize Transp4_11 with
      (P↔Q) (P ∧ Q ∨ ¬P ∧ ¬Q).
  intros Transp4_11a.
  apply propositional_extensionality.
  apply Transp4_11a. (*Not cited*)
Qed.

Theorem n5_25 : ∀ P Q : Prop,
  (P ∨ Q) ↔ ((P → Q) → Q).
  Proof. intros P Q.
  specialize n2_62 with P Q.
  intros n2_62a.
  specialize n2_68 with P Q.
  intros n2_68a.
  Conj n2_62a n2_68a.
  split.
  apply n2_62a.
  apply n2_68a.
  Equiv H.
  apply H.
  apply Equiv4_01.
```

```
Qed.

Theorem n5_3 : ∀ P Q R : Prop,
  ((P ∧ Q) → R) ↔ ((P ∧ Q) → (P ∧ R)).
  Proof. intros P Q R.
  specialize Comp3_43 with (P ∧ Q) P R.
  intros Comp3_43a.
  specialize Exp3_3 with
      (P ∧ Q → P) (P ∧ Q →R) (P ∧ Q → P ∧ R).
  intros Exp3_3a. (*Not cited*)
  MP Exp3_3a Comp3_43a.
  specialize Simp3_26 with P Q.
  intros Simp3_26a.
  MP Exp3_3a Simp3_26a.
  specialize Syll2_05 with (P ∧ Q) (P ∧ R) R.
  intros Syll2_05a.
  specialize Simp3_27 with P R.
  intros Simp3_27a.
  MP Syll2_05a Simp3_27a.
  clear Comp3_43a. clear Simp3_27a.
      clear Simp3_26a.
  Conj Exp3_3a Syll2_05a.
  split.
  apply Exp3_3a.
  apply Syll2_05a.
  Equiv H.
  apply H.
  apply Equiv4_01.
Qed.

Theorem n5_31 : ∀ P Q R : Prop,
  (R ∧ (P → Q)) → (P → (Q ∧ R)).
  Proof. intros P Q R.
  specialize Comp3_43 with P Q R.
  intros Comp3_43a.
  specialize Simp2_02 with P R.
  intros Simp2_02a.
  specialize Exp3_3 with
      (P→R) (P→Q) (P→(Q ∧ R)).
  intros Exp3_3a. (*Not cited*)
  specialize n3_22 with (P → R) (P → Q). (*Not cited*)
  intros n3_22a.
```

```
  Syll n3_22a Comp3_43a Sa.
  MP Exp3_3a Sa.
  Syll Simp2_02a Exp3_3a Sb.
  specialize Imp3_31 with R (P→Q) (P→(Q ∧ R)).
  intros Imp3_31a. (*Not cited*)
  MP Imp3_31a Sb.
  apply Imp3_31a.
Qed.

Theorem n5_32 : ∀ P Q R : Prop,
  (P → (Q ↔ R)) ↔ ((P ∧ Q) ↔ (P ∧ R)).
  Proof. intros P Q R.
  specialize n4_76 with P (Q→R) (R→Q).
  intros n4_76a.
  specialize Exp3_3 with P Q R.
  intros Exp3_3a.
  specialize Imp3_31 with P Q R.
  intros Imp3_31a.
  Conj Exp3_3a Imp3_31a.
  split.
  apply Exp3_3a.
  apply Imp3_31a.
  Equiv H.
  specialize Exp3_3 with P R Q.
  intros Exp3_3b.
  specialize Imp3_31 with P R Q.
  intros Imp3_31b.
  Conj Exp3_3b Imp3_31b.
  split.
  apply Exp3_3b.
  apply Imp3_31b.
  Equiv H0.
  specialize n5_3 with P Q R.
  intros n5_3a.
  specialize n5_3 with P R Q.
  intros n5_3b.
  replace (P→Q→R) with (P∧Q→R) in n4_76a.
  replace (P∧Q→R) with (P∧Q→P∧R) in n4_76a.
  replace (P→R→Q) with (P∧R→Q) in n4_76a.
  replace (P∧R→Q) with (P∧R→P∧Q) in n4_76a.
  replace ((P∧Q→P∧R)∧(P∧R→P∧Q)) with
      ((P∧Q)↔(P∧R)) in n4_76a.
```

```
  replace ((P∧Q↔P∧R)↔(P→(Q→R)∧(R→Q))) with
      ((P→(Q→R)∧(R→Q))↔(P∧Q ↔ P∧R)) in n4_76a.
  replace ((Q→R)∧(R→Q)) with (Q↔R) in n4_76a.
  apply n4_76a.
  apply Equiv4_01.
  apply propositional_extensionality.
  specialize n4_21 with
      (P→((Q→R)∧(R→Q))) ((P∧Q)↔(P∧R)).
  intros n4_21a.
  apply n4_21a. (*to commute the biconditional*)
  apply Equiv4_01.
  replace (P ∧ R → P ∧ Q) with (P ∧ R → Q).
  reflexivity.
  apply propositional_extensionality.
  apply n5_3b.
  apply propositional_extensionality.
  apply H0.
  replace (P ∧ Q → P ∧ R) with (P ∧ Q → R).
  reflexivity.
  apply propositional_extensionality.
  apply n5_3a.
  apply propositional_extensionality.
  apply H.
  apply Equiv4_01.
  apply Equiv4_01.
Qed.

Theorem n5_33 : ∀ P Q R : Prop,
  (P ∧ (Q → R)) ↔ (P ∧ ((P ∧ Q) → R)).
  Proof. intros P Q R.
    specialize n5_32 with P (Q→R) ((P∧Q)→R).
    intros n5_32a.
    replace
        ((P→(Q→R)↔(P∧Q→R))↔(P∧(Q→R)↔P∧(P∧Q→R)))
        with
        (((P→(Q→R)↔(P∧Q→R))→(P∧(Q→R)↔P∧(P∧Q→R)))
        ∧
        ((P∧(Q→R)↔P∧(P∧Q→R)→(P→(Q→R)↔(P∧Q→R))))
        in n5_32a.
    specialize Simp3_26 with
        ((P→(Q→R)↔(P∧Q→R))→(P∧(Q→R)↔P∧(P∧Q→R)))
        ((P∧(Q→R)↔P∧(P∧Q→R)→(P→(Q→R)↔(P∧Q→R)))).
```

```
    intros Simp3_26a. (*Not cited*)
    MP Simp3_26a n5_32a.
    specialize n4_73 with Q P.
    intros n4_73a.
    specialize n4_84 with Q (Q∧P) R.
    intros n4_84a.
    Syll n4_73a n4_84a Sa.
    replace (Q∧P) with (P∧Q) in Sa.
    MP Simp3_26a Sa.
    apply Simp3_26a.
    apply propositional_extensionality.
    specialize n4_3 with P Q.
    intros n4_3a.
    apply n4_3a. (*Not cited*)
    apply Equiv4_01.
Qed.

Theorem n5_35 : ∀ P Q R : Prop,
  ((P → Q) ∧ (P → R)) → (P → (Q ↔ R)).
  Proof. intros P Q R.
  specialize Comp3_43 with P Q R.
  intros Comp3_43a.
  specialize n5_1 with Q R.
  intros n5_1a.
  specialize Syll2_05 with P (Q∧R) (Q↔R).
  intros Syll2_05a.
  MP Syll2_05a n5_1a.
  Syll Comp3_43a Syll2_05a Sa.
  apply Sa.
Qed.

Theorem n5_36 : ∀ P Q : Prop,
  (P ∧ (P ↔ Q)) ↔ (Q ∧ (P ↔ Q)).
  Proof. intros P Q.
  specialize n5_32 with (P↔Q) P Q.
  intros n5_32a.
  specialize Id2_08 with (P↔Q).
  intros Id2_08a.
  replace (P↔Q→P↔Q) with
      ((P↔Q)∧P↔(P↔Q)∧Q) in Id2_08a.
  replace ((P↔Q)∧P) with (P∧(P↔Q)) in Id2_08a.
  replace ((P↔Q)∧Q) with (Q∧(P↔Q)) in Id2_08a.
```

```
  apply Id2_08a.
  apply propositional_extensionality.
  specialize n4_3 with Q (P↔Q).
  intros n4_3a.
  apply n4_3a.
  apply propositional_extensionality.
  specialize n4_3 with P (P↔Q).
  intros n4_3b.
  apply n4_3b.
  replace ((P ↔ Q) ∧ P ↔ (P ↔ Q) ∧ Q) with
      (P ↔ Q → P ↔ Q).
  reflexivity.
  apply propositional_extensionality.
  apply n5_32a.
Qed.
  (*The proof sketch cites Ass3_35 and n4_38,
    but the sketch was indecipherable.*)

Theorem n5_4 : ∀ P Q : Prop,
  (P → (P → Q)) ↔ (P → Q).
  Proof. intros P Q.
  specialize n2_43 with P Q.
  intros n2_43a.
  specialize Simp2_02 with (P) (P→Q).
  intros Simp2_02a.
  Conj n2_43a Simp2_02a.
  split.
  apply n2_43a.
  apply Simp2_02a.
  Equiv H.
  apply H.
  apply Equiv4_01.
Qed.

Theorem n5_41 : ∀ P Q R : Prop,
  ((P → Q) → (P → R)) ↔ (P → Q → R).
  Proof. intros P Q R.
  specialize n2_86 with P Q R.
  intros n2_86a.
  specialize n2_77 with P Q R.
  intros n2_77a.
  Conj n2_86a n2_77a.
```

```
  split.
  apply n2_86a.
  apply n2_77a.
  Equiv H.
  apply H.
  apply Equiv4_01.
Qed.

Theorem n5_42 : ∀ P Q R : Prop,
  (P → Q → R) ↔ (P → Q → P ∧ R).
  Proof. intros P Q R.
  specialize n5_3 with P Q R.
  intros n5_3a.
  specialize n4_87 with P Q R.
  intros n4_87a.
  replace ((P∧Q)→R) with (P→Q→R) in n5_3a.
  specialize n4_87 with P Q (P∧R).
  intros n4_87b.
  replace ((P∧Q)→(P∧R)) with
      (P→Q→(P∧R)) in n5_3a.
  apply n5_3a.
  specialize Imp3_31 with P Q (P∧R).
  intros Imp3_31b.
  specialize Exp3_3 with P Q (P∧R).
  intros Exp3_3b.
  Conj Imp3_31b Exp3_3b.
  split.
  apply Imp3_31b.
  apply Exp3_3b.
  Equiv H.
  apply propositional_extensionality.
  apply H.
  apply Equiv4_01.
  specialize Imp3_31 with P Q R.
  intros Imp3_31a.
  specialize Exp3_3 with P Q R.
  intros Exp3_3a.
  Conj Imp3_31a Exp3_3.
  split.
  apply Imp3_31a.
  apply Exp3_3a.
  Equiv H.
```

```
  apply propositional_extensionality.
  apply H.
  apply Equiv4_01.
Qed.

Theorem n5_44 : ∀ P Q R : Prop,
  (P→Q) → ((P → R) ↔ (P → (Q ∧ R))).
  Proof. intros P Q R.
  specialize n4_76 with P Q R.
  intros n4_76a.
  rewrite Equiv4_01 in n4_76a.
  specialize Simp3_26 with
    (((P→Q)∧(P→R))→(P→(Q∧R)))
    ((P→(Q∧R))→((P→Q)∧(P→R))).
  intros Simp3_26a.
  MP Simp3_26a n4_76a.
  specialize Simp3_27 with
    (((P→Q)∧(P→R))→(P→(Q∧R)))
    ((P→(Q∧R))→((P→Q)∧(P→R))).
  intros Simp3_27a.
  MP Simp3_27a n4_76a.
  specialize Simp3_27 with (P→Q) (P→Q∧R).
  intros Simp3_27d.
  Syll Simp3_27d Simp3_27a Sa.
  specialize n5_3 with (P→Q) (P→R) (P→(Q∧R)).
  intros n5_3a.
  rewrite Equiv4_01 in n5_3a.
  specialize Simp3_26 with
    ((((P→Q)∧(P→R))→(P→(Q∧R)))→
    (((P→Q)∧(P→R))→((P→Q)∧(P→(Q∧R)))))
    ((((P→Q)∧(P→R))→((P→Q)∧(P→(Q∧R))))
    →(((P→Q)∧(P→R))→(P→(Q∧R)))).
  intros Simp3_26b.
  MP Simp3_26b n5_3a.
  specialize Simp3_27 with
  ((((P→Q)∧(P→R))→(P→(Q∧R)))→
  (((P→Q)∧(P→R))→((P→Q)∧(P→(Q∧R)))))
  ((((P→Q)∧(P→R))→((P→Q)∧(P→(Q∧R))))
  →(((P→Q)∧(P→R))→(P→(Q∧R)))).
  intros Simp3_27b.
  MP Simp3_27b n5_3a.
  MP Simp3_26a Simp3_26b.
```

```
  MP Simp3_27a Simp3_27b.
  clear n4_76a. clear Simp3_26a. clear Simp3_27a.
    clear Simp3_27b. clear Simp3_27d. clear n5_3a.
  Conj Simp3_26b Sa.
  split.
  apply Sa.
  apply Simp3_26b.
  Equiv H.
  specialize n5_32 with (P→Q) (P→R) (P→(Q∧R)).
  intros n5_32a.
  rewrite Equiv4_01 in n5_32a.
  specialize Simp3_27 with
    (((P → Q) → (P → R) ↔ (P → Q ∧ R))
      → (P → Q) ∧ (P → R) ↔ (P → Q) ∧ (P → Q ∧ R))
    ((P → Q) ∧ (P → R) ↔ (P → Q) ∧ (P → Q ∧ R)
      → (P → Q) → (P → R) ↔ (P → Q ∧ R)).
  intros Simp3_27c.
  MP Simp3_27c n5_32a.
  replace (((P→Q)∧(P→(Q∧R)))↔((P→Q)∧(P→R)))
    with (((P→Q)∧(P→R))↔((P→Q)∧(P→(Q∧R)))) in H.
  MP Simp3_27c H.
  apply Simp3_27c.
  specialize n4_21 with
    ((P→Q)∧(P→R)) ((P→Q)∧(P→(Q∧R))).
  intros n4_21a.
  apply propositional_extensionality.
  apply n4_21a.
  apply Equiv4_01.
Qed.

Theorem n5_5 : ∀ P Q : Prop,
  P → ((P → Q) ↔ Q).
  Proof. intros P Q.
  specialize Ass3_35 with P Q.
  intros Ass3_35a.
  specialize Exp3_3 with P (P→Q) Q.
  intros Exp3_3a.
  MP Exp3_3a Ass3_35a.
  specialize Simp2_02 with P Q.
  intros Simp2_02a.
  specialize Exp3_3 with P Q (P→Q).
  intros Exp3_3b.
```

```
  specialize n3_42 with P Q (P→Q). (*Not cited*)
  intros n3_42a.
  MP n3_42a Simp2_02a.
  MP Exp3_3b n3_42a.
  clear n3_42a. clear Simp2_02a. clear Ass3_35a.
  Conj Exp3_3a Exp3_3b.
  split.
  apply Exp3_3a.
  apply Exp3_3b.
  specialize n3_47 with P P ((P→Q)→Q) (Q→(P→Q)).
  intros n3_47a.
  MP n3_47a H.
  replace (P∧P) with P in n3_47a.
  replace (((P→Q)→Q)∧(Q→(P→Q))) with
      ((P→Q)↔Q) in n3_47a.
  apply n3_47a.
  apply Equiv4_01.
  apply propositional_extensionality.
  specialize n4_24 with P.
  intros n4_24a. (*Not cited*)
  apply n4_24a.
Qed.

Theorem n5_501 : ∀ P Q : Prop,
  P → (Q ↔ (P ↔ Q)).
  Proof. intros P Q.
  specialize n5_1 with P Q.
  intros n5_1a.
  specialize Exp3_3 with P Q (P↔Q).
  intros Exp3_3a.
  MP Exp3_3a n5_1a.
  specialize Ass3_35 with P Q.
  intros Ass3_35a.
  specialize Simp3_26 with (P∧(P→Q)) (Q→P).
  intros Simp3_26a. (*Not cited*)
  Syll Simp3_26a Ass3_35a Sa.
  replace ((P∧(P→Q))∧(Q→P)) with
      (P∧((P→Q)∧(Q→P))) in Sa.
  replace ((P→Q)∧(Q→P)) with (P↔Q) in Sa.
  specialize Exp3_3 with P (P↔Q) Q.
  intros Exp3_3b.
  MP Exp3_3b Sa.
```

```
  clear n5_1a. clear Ass3_35a.
      clear Simp3_26a. clear Sa.
  Conj Exp3_3a Exp3_3b.
  split.
  apply Exp3_3a.
  apply Exp3_3b.
  specialize n4_76 with P (Q→(P↔Q)) ((P↔Q)→Q).
  intros n4_76a. (*Not cited*)
  replace ((P→Q→P↔Q)∧(P→P↔Q→Q)) with
      ((P→(Q→P↔Q)∧(P↔Q→Q))) in H.
  replace ((Q→(P↔Q))∧((P↔Q)→Q)) with
      (Q↔(P↔Q)) in H.
  apply H.
  apply Equiv4_01.
  replace (P→(Q→P↔Q)∧(P↔Q→Q)) with
      ((P→Q→P↔Q)∧(P→P↔Q→Q)).
  reflexivity.
  apply propositional_extensionality.
  apply n4_76a.
  apply Equiv4_01.
  replace (P∧(P→Q)∧(Q→P)) with
      ((P∧(P→Q))∧(Q→P)).
  reflexivity.
  apply propositional_extensionality.
  specialize n4_32 with P (P→Q) (Q→P).
  intros n4_32a. (*Not cited*)
  apply n4_32a.
Qed.

Theorem n5_53 : ∀ P Q R S : Prop,
  (((P ∨ Q) ∨ R) → S) ↔ (((P → S) ∧ (Q → S)) ∧ (R → S)).
  Proof. intros P Q R S.
  specialize n4_77 with S (P∨Q) R.
  intros n4_77a.
  specialize n4_77 with S P Q.
  intros n4_77b.
  replace (P ∨ Q → S) with
      ((P→S)∧(Q→S)) in n4_77a.
  replace ((((P→S)∧(Q→S))∧(R→S))↔(((P∨Q)∨R)→S))
      with
      ((((P∨Q)∨R)→S)↔(((P→S)∧(Q→S))∧(R→S)))
      in n4_77a.
```

```
  apply n4_77a.
  apply propositional_extensionality.
  specialize n4_21 with ((P ∨ Q) ∨ R → S)
      (((P → S) ∧ (Q → S)) ∧ (R → S)).
  intros n4_21a.
  apply n4_21a. (*Not cited*)
  apply propositional_extensionality.
  apply n4_77b.
Qed.

Theorem n5_54 : ∀ P Q : Prop,
  ((P ∧ Q) ↔ P) ∨ ((P ∧ Q) ↔ Q).
  Proof. intros P Q.
  specialize n4_73 with P Q.
  intros n4_73a.
  specialize n4_44 with Q P.
  intros n4_44a.
  specialize Transp2_16 with Q (P↔(P∧Q)).
  intros Transp2_16a.
  MP n4_73a Transp2_16a.
  specialize Transp4_11 with Q (Q∨(P∧Q)).
  intros Transp4_11a.
  replace (Q∧P) with (P∧Q) in n4_44a.
  replace (Q↔Q∨P∧Q) with
      (¬Q↔¬(Q∨P∧Q)) in n4_44a.
  replace (¬Q) with (¬(Q∨P∧Q)) in Transp2_16a.
  replace (¬(Q∨P∧Q)) with
      (¬Q∧¬(P∧Q)) in Transp2_16a.
  specialize n5_1 with (¬Q) (¬(P∧Q)).
  intros n5_1a.
  Syll Transp2_16a n5_1a Sa.
  replace (¬(P↔P∧Q)→(¬Q↔¬(P∧Q))) with
      (¬¬(P↔P∧Q)∨(¬Q↔¬(P∧Q))) in Sa.
  replace (¬¬(P↔P∧Q)) with (P↔P∧Q) in Sa.
  specialize Transp4_11 with Q (P∧Q).
  intros Transp4_11b.
  replace (¬Q↔¬(P∧Q)) with (Q↔(P∧Q)) in Sa.
  replace (Q↔(P∧Q)) with ((P∧Q)↔Q) in Sa.
  replace (P↔(P∧Q)) with ((P∧Q)↔P) in Sa.
  apply Sa.
  apply propositional_extensionality.
  specialize n4_21 with (P∧Q) P.
```

```
  intros n4_21a. (*Not cited*)
  apply n4_21a.
  apply propositional_extensionality.
  specialize n4_21 with (P∧Q) Q.
  intros n4_21b. (*Not cited*)
  apply n4_21b.
  apply propositional_extensionality.
  apply Transp4_11b.
  apply propositional_extensionality.
  specialize n4_13 with (P ↔ (P∧Q)).
  intros n4_13a. (*Not cited*)
  apply n4_13a.
  rewrite <- Impl1_01. (*Not cited*)
  reflexivity.
  apply propositional_extensionality.
  specialize n4_56 with Q (P∧Q).
  intros n4_56a. (*Not cited*)
  apply n4_56a.
  replace (¬(Q∨P∧Q)) with (¬Q).
  reflexivity.
  apply propositional_extensionality.
  apply n4_44a.
  replace (¬Q↔¬(Q∨P∧Q)) with (Q↔Q∨P∧Q).
  reflexivity.
  apply propositional_extensionality.
  apply Transp4_11a.
  apply propositional_extensionality.
  specialize n4_3 with P Q.
  intros n4_3a. (*Not cited*)
  apply n4_3a.
Qed.

Theorem n5_55 : ∀ P Q : Prop,
  ((P ∨ Q) ↔ P) ∨ ((P ∨ Q) ↔ Q).
  Proof. intros P Q.
  specialize Add1_3 with (P∧Q) (P).
  intros Add1_3a.
  replace ((P∧Q)∨P) with ((P∨P)∧(Q∨P)) in Add1_3a.
  replace (P∨P) with P in Add1_3a.
  replace (Q∨P) with (P∨Q) in Add1_3a.
  specialize n5_1 with P (P∨Q).
  intros n5_1a.
```

```
Syll Add1_3a n5_1a Sa.
specialize n4_74 with P Q.
intros n4_74a.
specialize Transp2_15 with P (Q↔P∨Q).
intros Transp2_15a. (*Not cited*)
MP Transp2_15a n4_74a.
Syll Transp2_15a Sa Sb.
replace (¬(Q↔(P∨Q))→(P↔(P∨Q))) with
    (¬¬(Q↔(P∨Q))∨(P↔(P∨Q))) in Sb.
replace (¬¬(Q↔(P∨Q))) with (Q↔(P∨Q)) in Sb.
replace (Q↔(P∨Q)) with ((P∨Q)↔Q) in Sb.
replace (P↔(P∨Q)) with ((P∨Q)↔P) in Sb.
replace ((P∨Q↔Q)∨(P∨Q↔P)) with
    ((P∨Q↔P)∨(P∨Q↔Q)) in Sb.
apply Sb.
apply propositional_extensionality.
specialize n4_31 with (P ∨ Q ↔ P) (P ∨ Q ↔ Q).
intros n4_31a.  (*Not cited*)
apply n4_31a.
apply propositional_extensionality.
specialize n4_21 with (P ∨ Q) P.
intros n4_21a. (*Not cited*)
apply n4_21a.
apply propositional_extensionality.
specialize n4_21 with (P ∨ Q) Q.
intros n4_21b. (*Not cited*)
apply n4_21b.
apply propositional_extensionality.
specialize n4_13 with (Q ↔ P ∨ Q).
intros n4_13a. (*Not cited*)
apply n4_13a.
rewrite <- Impl1_01.
reflexivity.
apply propositional_extensionality.
specialize n4_31 with P Q.
intros n4_31b.
apply n4_31b.
apply propositional_extensionality.
specialize n4_25 with P.
intros n4_25a. (*Not cited*)
apply n4_25a.
replace ((P∨P)∧(Q∨P)) with ((P∧Q)∨P).
```

```
  reflexivity.
  replace ((P∧Q)∨P) with (P∨(P∧Q)).
  replace (Q∨P) with (P∨Q).
  apply propositional_extensionality.
  specialize n4_41 with P P Q.
  intros n4_41a. (*Not cited*)
  apply n4_41a.
  apply propositional_extensionality.
  specialize n4_31 with P Q.
  intros n4_31c.
  apply n4_31c.
  apply propositional_extensionality.
  specialize n4_31 with P (P ∧ Q).
  intros n4_31d. (*Not cited*)
  apply n4_31d.
Qed.

Theorem n5_6 : ∀ P Q R : Prop,
  ((P ∧ ¬Q) → R) ↔ (P → (Q ∨ R)).
  Proof. intros P Q R.
  specialize n4_87 with P (¬Q) R.
  intros n4_87a.
  specialize n4_64 with Q R.
  intros n4_64a.
  specialize n4_85 with P Q R.
  intros n4_85a.
  replace (((P∧¬Q→R)↔(P→¬Q→R))↔((¬Q→P→R)↔(¬Q∧P→R)))
       with
       ((((P∧¬Q→R)↔(P→¬Q→R))→((¬Q→P→R)↔(¬Q∧P→R)))
       ∧
       (((¬Q→P→R)↔(¬Q∧P→R))→(((P∧¬Q→R)↔(P→¬Q→R)))))
       in n4_87a.
  specialize Simp3_27 with
      (((P∧¬Q→R)↔(P→¬Q→R)→(¬Q→P→R)↔(¬Q∧P→R)))
      (((¬Q→P→R)↔(¬Q∧P→R)→(P∧¬Q→R)↔(P→¬Q→R))).
  intros Simp3_27a.
  MP Simp3_27a n4_87a.
  specialize Imp3_31 with (¬Q) P R.
  intros Imp3_31a.
  specialize Exp3_3 with (¬Q) P R.
  intros Exp3_3a.
  Conj Imp3_31a Exp3_3a.
```

```
  split.
  apply Imp3_31a.
  apply Exp3_3a.
  Equiv H.
  MP Simp3_27a H.
  replace (¬Q→R) with (Q∨R) in Simp3_27a.
  apply Simp3_27a.
  replace (Q ∨ R) with (¬Q → R).
  reflexivity.
  apply propositional_extensionality.
  apply n4_64a.
  apply Equiv4_01.
  apply Equiv4_01.
Qed.

Theorem n5_61 : ∀ P Q : Prop,
  ((P ∨ Q) ∧ ¬Q) ↔ (P ∧ ¬Q).
  Proof. intros P Q.
  specialize n4_74 with Q P.
  intros n4_74a.
  specialize n5_32 with (¬Q) P (Q∨P).
  intros n5_32a.
  replace (¬Q → P ↔ Q ∨ P) with
      (¬Q ∧ P ↔ ¬Q ∧ (Q ∨ P)) in n4_74a.
  replace (¬Q ∧ P) with (P ∧ ¬Q) in n4_74a.
  replace (¬Q ∧ (Q ∨ P)) with ((Q ∨ P) ∧ ¬Q) in n4_74a.
  replace (Q ∨ P) with (P ∨ Q) in n4_74a.
  replace (P ∧ ¬Q ↔ (P ∨ Q) ∧ ¬Q) with
      ((P ∨ Q) ∧ ¬Q ↔ P ∧ ¬Q) in n4_74a.
  apply n4_74a.
  apply propositional_extensionality.
  specialize n4_21 with ((P ∨ Q) ∧ ¬Q) (P ∧ ¬Q).
  intros n4_21a. (*Not cited*)
  apply n4_21a.
  apply propositional_extensionality.
  specialize n4_31 with P Q.
  intros n4_31a. (*Not cited*)
  apply n4_31a.
  apply propositional_extensionality.
  specialize n4_3 with (Q ∨ P) (¬Q).
  intros n4_3a. (*Not cited*)
  apply n4_3a.
```

```
  apply propositional_extensionality.
  specialize n4_3 with P (¬Q).
  intros n4_3b. (*Not cited*)
  apply n4_3b.
  replace (¬Q ∧ P ↔ ¬Q ∧ (Q ∨ P)) with
      (¬Q → P ↔ Q ∨ P).
  reflexivity.
  apply propositional_extensionality.
  apply n5_32a.
Qed.

Theorem n5_62 : ∀ P Q : Prop,
  ((P ∧ Q) ∨ ¬Q) ↔ (P ∨ ¬Q).
  Proof. intros P Q.
  specialize n4_7 with Q P.
  intros n4_7a.
  replace (Q→P) with (¬Q∨P) in n4_7a.
  replace (Q→(Q∧P)) with (¬Q∨(Q∧P)) in n4_7a.
  replace (¬Q∨(Q∧P)) with ((Q∧P)∨¬Q) in n4_7a.
  replace (¬Q∨P) with (P∨¬Q) in n4_7a.
  replace (Q∧P) with (P∧Q) in n4_7a.
  replace (P ∨ ¬Q ↔ P ∧ Q ∨ ¬Q) with
      (P ∧ Q ∨ ¬Q ↔ P ∨ ¬Q) in n4_7a.
  apply n4_7a.
  apply propositional_extensionality.
  specialize n4_21 with (P ∧ Q ∨ ¬Q) (P ∨ ¬Q).
  intros n4_21a. (*Not cited*)
  apply n4_21a.
  apply propositional_extensionality.
  specialize n4_3 with P Q.
  intros n4_3a. (*Not cited*)
  apply n4_3a.
  apply propositional_extensionality.
  specialize n4_31 with P (¬Q).
  intros n4_31a. (*Not cited*)
  apply n4_31a.
  apply propositional_extensionality.
  specialize n4_31 with (Q ∧ P) (¬Q).
  intros n4_31b. (*Not cited*)
  apply n4_31b.
  rewrite <- Impl1_01.
  reflexivity.
```

```
  rewrite <- Impl1_01.
  reflexivity.
Qed.

Theorem n5_63 : ∀ P Q : Prop,
  (P ∨ Q) ↔ (P ∨ (¬P ∧ Q)).
  Proof. intros P Q.
  specialize n5_62 with Q (¬P).
  intros n5_62a.
  replace (¬¬P) with P in n5_62a.
  replace (Q ∨ P) with (P ∨ Q) in n5_62a.
  replace ((Q∧¬P)∨P) with (P∨(Q∧¬P)) in n5_62a.
  replace (P ∨ Q ∧ ¬P ↔ P ∨ Q) with
      (P ∨ Q ↔ P ∨ Q ∧ ¬P) in n5_62a.
  replace (Q∧¬P) with (¬P∧Q) in n5_62a.
  apply n5_62a.
  apply propositional_extensionality.
  specialize n4_3 with (¬P) Q.
  intros n4_3a.
  apply n4_3a. (*Not cited*)
  apply propositional_extensionality.
  specialize n4_21 with (P∨Q) (P∨(Q∧¬P)).
  intros n4_21a. (*Not cited*)
  apply n4_21a.
  apply propositional_extensionality.
  specialize n4_31 with P (Q∧¬P).
  intros n4_31a. (*Not cited*)
  apply n4_31a.
  apply propositional_extensionality.
  specialize n4_31 with P Q.
  intros n4_31b. (*Not cited*)
  apply n4_31b.
  apply propositional_extensionality.
  specialize n4_13 with P.
  intros n4_13a. (*Not cited*)
  apply n4_13a.
Qed.

Theorem n5_7 : ∀ P Q R : Prop,
  ((P ∨ R) ↔ (Q ∨ R)) ↔ (R ∨ (P ↔ Q)).
  Proof. intros P Q R.
  specialize n4_74 with R P.
```

```
intros n4_74a.
specialize n4_74 with R Q.
intros n4_74b. (*Greg's suggestion*)
Conj n4_74a n4_74b.
split.
apply n4_74a.
apply n4_74b.
specialize Comp3_43 with
  (¬R) (P↔R∨P) (Q↔R∨Q).
intros Comp3_43a.
MP Comp3_43a H.
specialize n4_22 with P (R∨P) (R∨Q).
intros n4_22a.
specialize n4_22 with P (R∨Q) Q.
intros n4_22b.
specialize Exp3_3 with (P↔(R∨Q))
  ((R∨Q)↔Q) (P↔Q).
intros Exp3_3a.
MP Exp3_3a n4_22b.
Syll n4_22a Exp3_3a Sa.
specialize Imp3_31 with ((P↔(R∨P))∧
  ((R ∨ P) ↔ (R ∨ Q))) ((R∨Q)↔Q) (P↔Q).
intros Imp3_31a.
MP Imp3_31a Sa.
replace (((P↔(R∨P))∧((R ∨ P) ↔
    (R ∨ Q))) ∧ ((R∨Q)↔Q)) with
  ((P↔(R∨P))∧(((R ∨ P) ↔
    (R ∨ Q)) ∧ ((R∨Q)↔Q))) in Imp3_31a.
replace ((R ∨ P ↔ R ∨ Q) ∧ (R ∨ Q ↔ Q)) with
  ((R ∨ Q ↔ Q) ∧ (R ∨ P ↔ R ∨ Q)) in Imp3_31a.
replace ((P↔(R∨P)) ∧
    ((R ∨ Q ↔ Q) ∧ (R ∨ P ↔ R ∨ Q))) with
  (((P↔(R∨P)) ∧ (R ∨ Q ↔ Q)) ∧
    (R ∨ P ↔ R ∨ Q)) in Imp3_31a.
specialize Exp3_3 with
  ((P↔(R∨P))∧(R∨Q↔Q))
  (R ∨ P ↔ R ∨ Q) (P ↔ Q).
intros Exp3_3b.
MP Exp3_3b Imp3_31a.
replace (Q↔R∨Q) with (R∨Q↔Q) in Comp3_43a.
Syll Comp3_43a Exp3_3b Sb.
replace (R∨P) with (P∨R) in Sb.
```

```
replace (R∨Q) with (Q∨R) in Sb.
specialize Imp3_31 with (¬R) (P∨R↔Q∨R) (P↔Q).
intros Imp3_31b.
MP Imp3_31b Sb.
replace (¬R ∧ (P ∨ R ↔ Q ∨ R)) with
  ((P ∨ R ↔ Q ∨ R) ∧ ¬R) in Imp3_31b.
specialize Exp3_3 with
  (P ∨ R ↔ Q ∨ R) (¬R) (P ↔ Q).
intros Exp3_3c.
MP Exp3_3c Imp3_31b.
replace (¬R→(P↔Q)) with
  (¬¬R∨(P↔Q)) in Exp3_3c.
replace (¬¬R) with R in Exp3_3c.
specialize Add1_3 with P R.
intros Add1_3a.
specialize Add1_3 with Q R.
intros Add1_3b.
Conj Add1_3a Add1_3b.
split.
apply Add1_3a.
apply Add1_3b.
specialize Comp3_43 with (R) (P∨R) (Q∨R).
intros Comp3_43b.
MP Comp3_43b H0.
specialize n5_1 with (P ∨ R) (Q ∨ R).
intros n5_1a.
Syll Comp3_43b n5_1a Sc.
specialize n4_37 with P Q R.
intros n4_37a.
Conj Sc n4_37a.
split.
apply Sc.
apply n4_37a.
specialize n4_77 with (P ∨ R ↔ Q ∨ R)
  R (P↔Q).
intros n4_77a.
rewrite Equiv4_01 in n4_77a.
specialize Simp3_26 with
  ((R → P ∨ R ↔ Q ∨ R) ∧
    (P ↔ Q → P ∨ R ↔ Q ∨ R)
  → R ∨ (P ↔ Q) → P ∨ R ↔ Q ∨ R)
  ((R ∨ (P ↔ Q) → P ∨ R ↔ Q ∨ R)
```

```
        → (R → P ∨ R ↔ Q ∨ R) ∧
          (P ↔ Q → P ∨ R ↔ Q ∨ R)).
  intros Simp3_26a.
  MP Simp3_26 n4_77a.
  MP Simp3_26a H1.
  clear n4_77a. clear H1. clear n4_37a. clear Sa.
    clear n5_1a. clear Comp3_43b. clear H0.
    clear Add1_3a. clear Add1_3b. clear H. clear Imp3_31b.
    clear n4_74a. clear n4_74b. clear Comp3_43a.
    clear Imp3_31a. clear n4_22a. clear n4_22b.
    clear Exp3_3a. clear Exp3_3b. clear Sb. clear Sc.
  Conj Exp3_3c Simp3_26a.
  split.
  apply Exp3_3c.
  apply Simp3_26a.
  Equiv H.
  apply H.
  apply Equiv4_01.
  apply propositional_extensionality.
  apply n4_13. (*With R*)
  rewrite <- Impl1_01. (*With (¬R) (P↔Q)*)
  reflexivity.
  apply propositional_extensionality.
  apply n4_3. (*With (R ∨ Q ↔ R ∨ P) (¬R)*)
  apply propositional_extensionality.
  apply n4_31. (*With P R*)
  apply propositional_extensionality.
  apply n4_31. (*With Q R*)
  apply propositional_extensionality.
  apply n4_21. (*With (P ∨ R) (Q ∨ R)*)
  apply propositional_extensionality.
  apply n4_32. (*With (P ↔ R ∨ P) (R ∨ Q ↔ Q) (R ∨ P ↔ R ∨ Q)*)
  apply propositional_extensionality.
  apply n4_3. (*With (R ∨ Q ↔ Q) (R ∨ P ↔ R ∨ Q)*)
  apply propositional_extensionality.
  apply n4_21. (*To commute the biconditional.*)
  apply n4_32. (*With (P ↔ R ∨ P) (R ∨ P ↔ R ∨ Q) (R ∨ Q ↔ Q)*)
Qed.

Theorem n5_71 : ∀ P Q R : Prop,
  (Q → ¬R) → (((P ∨ Q) ∧ R) ↔ (P ∧ R)).
  Proof. intros P Q R.
```

```
specialize n4_62 with Q R.
intros n4_62a.
specialize n4_51 with Q R.
intros n4_51a.
specialize n4_21 with (¬(Q∧R)) (¬Q∨¬R).
intros n4_21a.
rewrite Equiv4_01 in n4_21a.
specialize Simp3_26 with
  ((¬(Q∧R)↔(¬Q∨¬R))→((¬Q∨¬R)↔¬(Q∧R)))
  (((¬Q∨¬R)↔¬(Q∧R))→(¬(Q∧R)↔(¬Q∨¬R))).
intros Simp3_26a.
MP Simp3_26a n4_21a.
MP Simp3_26a n4_51a.
clear n4_21a. clear n4_51a.
Conj n4_62a Simp3_26a.
split.
apply n4_62a.
apply Simp3_26a.
specialize n4_22 with
  (Q→¬R) (¬Q∨¬R) (¬(Q∧R)).
intros n4_22a.
MP n4_22a H.
replace ((Q→¬R)↔¬(Q∧R)) with
    (((Q→¬R)→¬(Q∧R))
    ∧
    (¬(Q∧R)→(Q→¬R))) in n4_22a.
specialize Simp3_26 with
    ((Q→¬R)→¬(Q∧R)) (¬(Q∧R)→(Q→¬R)).
intros Simp3_26b.
MP Simp3_26b n4_22a.
specialize n4_74 with (Q∧R) (P∧R).
intros n4_74a.
Syll Simp3_26a n4_74a Sa.
replace ((P∧R)∨(Q∧R)) with
    ((Q∧R)∨(P∧R)) in Sa.
replace ((Q∧R)∨(P∧R)) with (R∧(P∨Q)) in Sa.
replace (R∧(P∨Q)) with ((P∨Q)∧R) in Sa.
replace ((P∧R)↔((P∨Q)∧R)) with
    (((P∨Q)∧R)↔(P∧R)) in Sa.
apply Sa.
apply propositional_extensionality.
specialize n4_21 with ((P∨Q)∧R) (P∧R).
```

```
  intros n4_21a. (*Not cited*)
  apply n4_21a.
  apply propositional_extensionality.
  specialize n4_3 with (P∨Q) R.
  intros n4_3a.
  apply n4_3a. (*Not cited*)
  apply propositional_extensionality.
  specialize n4_4 with R P Q.
  intros n4_4a.
  replace ((Q∧R)∨(P∧R)) with ((P∧R)∨(Q∧R)).
  replace (Q ∧ R) with (R ∧ Q).
  replace (P ∧ R) with (R ∧ P).
  apply n4_4a. (*Not cited*)
  apply propositional_extensionality.
  specialize n4_3 with R P.
  intros n4_3a.
  apply n4_3a.
  apply propositional_extensionality.
  specialize n4_3 with R Q.
  intros n4_3b.
  apply n4_3b.
  apply propositional_extensionality.
  specialize n4_31 with (P∧R) (Q∧R).
  intros n4_31a. (*Not cited*)
  apply n4_31a.
  apply propositional_extensionality.
  specialize n4_31 with (Q∧R) (P∧R).
  intros n4_31b. (*Not cited*)
  apply n4_31b.
  apply Equiv4_01.
Qed.

Theorem n5_74 : ∀ P Q R : Prop,
  (P → (Q ↔ R)) ↔ ((P → Q) ↔ (P → R)).
  Proof. intros P Q R.
  specialize n5_41 with P Q R.
  intros n5_41a.
  specialize n5_41 with P R Q.
  intros n5_41b.
  Conj n5_41a n5_41b.
  split.
  apply n5_41a.
```

```
  apply n5_41b.
  specialize n4_38 with
      ((P→Q)→(P→R)) ((P→R)→(P→Q))
      (P→Q→R) (P→R→Q).
  intros n4_38a.
  MP n4_38a H.
  replace (((P→Q)→(P→R))∧((P→R)→(P→Q)))
    with ((P→Q)↔(P→R)) in n4_38a.
  specialize n4_76 with P (Q→R) (R→Q).
  intros n4_76a.
  replace ((Q→R)∧(R→Q)) with (Q↔R) in n4_76a.
  replace ((P→Q→R)∧(P→R→Q)) with
      (P→(Q↔R)) in n4_38a.
  replace (((P→Q)↔(P→R))↔(P→Q↔R)) with
      ((P→(Q↔R))↔((P→Q)↔(P→R))) in n4_38a.
  apply n4_38a.
  apply propositional_extensionality.
  specialize n4_21 with (P→Q↔R)
    ((P→Q)↔(P→R)).
  intros n4_21a. (*Not cited*)
  apply n4_21a.
  replace (P→Q↔R) with ((P→Q→R)∧(P→R→Q)).
  reflexivity.
  apply propositional_extensionality.
  apply n4_76a.
  apply Equiv4_01.
  apply Equiv4_01.
Qed.

Theorem n5_75 : ∀ P Q R : Prop,
  ((R → ¬Q) ∧ (P ↔ Q ∨ R)) → ((P ∧ ¬Q) ↔ R).
  Proof. intros P Q R.
  specialize n5_6 with P Q R.
  intros n5_6a.
  replace ((P∧¬Q→R)↔(P→Q∨R)) with
      (((P∧¬Q→R)→(P→Q∨R)) ∧
      ((P→Q∨R)→(P∧¬Q→R))) in n5_6a.
  specialize Simp3_27 with
      ((P∧¬Q→R)→(P→Q∨R))
      ((P→Q∨R)→(P∧¬Q→R)).
  intros Simp3_27a.
  MP Simp3_27a n5_6a.
```

```
specialize Simp3_26 with
  (P→(Q∨R)) ((Q∨R)→P).
intros Simp3_26a.
replace ((P→(Q∨R))∧((Q∨R)→P)) with
    (P↔(Q∨R)) in Simp3_26a.
Syll Simp3_26a Simp3_27a Sa.
specialize Simp3_27 with
  (R→¬Q) (P↔(Q∨R)).
intros Simp3_27b.
Syll Simp3_27b Sa Sb.
specialize Simp3_27 with
  (P→(Q∨R)) ((Q∨R)→P).
intros Simp3_27c.
replace ((P→(Q∨R))∧((Q∨R)→P)) with
    (P↔(Q∨R)) in Simp3_27c.
Syll Simp3_27b Simp3_27c Sc.
specialize n4_77 with P Q R.
intros n4_77a.
replace (Q∨R→P) with ((Q→P)∧(R→P)) in Sc.
specialize Simp3_27 with (Q→P) (R→P).
intros Simp3_27d.
Syll Sa Simp3_27d Sd.
specialize Simp3_26 with (R→¬Q) (P↔(Q∨R)).
intros Simp3_26b.
Conj Sd Simp3_26b.
split.
apply Sd.
apply Simp3_26b.
specialize Comp3_43 with
    ((R→¬Q)∧(P↔(Q∨R))) (R→P) (R→¬Q).
intros Comp3_43a.
MP Comp3_43a H.
specialize Comp3_43 with R P (¬Q).
intros Comp3_43b.
Syll Comp3_43a Comp3_43b Se.
clear n5_6a. clear Simp3_27a.
    clear Simp3_27c. clear Simp3_27d.
    clear Simp3_26a.  clear Comp3_43b.
    clear Simp3_26b. clear Comp3_43a.
    clear Sa. clear Sc. clear Sd. clear H.
    clear n4_77a. clear Simp3_27b.
Conj Sb Se.
```

```
  split.
  apply Sb.
  apply Se.
  specialize Comp3_43 with
    ((R→¬Q)∧(P↔Q∨R))
    (P∧¬Q→R) (R→P∧¬Q).
  intros Comp3_43c.
  MP Comp3_43c H.
  replace ((P∧¬Q→R)∧(R→P∧¬Q)) with
      (P∧¬Q↔R) in Comp3_43c.
  apply Comp3_43c.
  apply Equiv4_01.
  apply propositional_extensionality.
  apply n4_77a.
  apply Equiv4_01.
  apply Equiv4_01.
  apply Equiv4_01.
Qed.

End No5.
```