```
Module No1.
Import Unicode.Utf8. (*We first give the axioms of Principia for the prop
ositional calculus in *1.*)

Axiom MP1_1 : ∀  P Q : Prop,
  (P → Q) → P → Q. (*Modus ponens*)

  (**1.11 ommitted: it is MP for propositions containing variables. Likewi
se, ommitted the well-formedness rules 1.7, 1.71, 1.72*)

Axiom Taut1_2 : ∀ P : Prop,
  P ∨ P→ P. (*Tautology*)

Axiom Add1_3 : ∀ P Q : Prop,
  Q → P ∨ Q. (*Addition*)

Axiom Perm1_4 : ∀ P Q : Prop,
  P ∨ Q → Q ∨ P. (*Permutation*)

Axiom Assoc1_5 : ∀ P Q R : Prop,
  P ∨ (Q ∨ R) → Q ∨ (P ∨ R).

Axiom Sum1_6: ∀ P Q R : Prop,
  (Q → R) → (P ∨ Q → P ∨ R). (*These are all the propositional axioms of P
rincipia Mathematica.*)

Axiom Impl1_01 : ∀ P Q : Prop,
  (P → Q) = (~P ∨ Q). (*This is a definition in Principia: there → is a define
d sign and ∨, ~ are primitive ones. So we will use this axiom to switch bet
ween disjunction and implication.*)

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).
  replace (~P ∨ ~P) with (P → ~P).
  apply MP1_1.
  apply Impl1_01.
Qed.

Theorem n2_02 : ∀ P Q : Prop,
  Q → (P → Q).
Proof. intros P Q.
  specialize Add1_3 with (~P) Q.
  replace (~P ∨ Q) with (P → Q).
  apply (MP1_1 Q (P → Q)).
  apply Impl1_01.
Qed.

Theorem n2_03 : ∀ P Q : Prop,
  (P → ~Q) → (Q → ~P).
Proof. intros P Q.
  specialize Perm1_4 with (~P) (~Q).
  replace (~P ∨ ~Q) with (P → ~Q).
  replace (~Q ∨ ~P) with (Q → ~P).
  apply (MP1_1 (P → ~Q) (Q → ~P)).
  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.
 replace (~Q ∨ R) with (Q → R).
 replace (~P ∨ (Q → R)) with (P → (Q → R)).
 replace (~P ∨ R) with (P → R).
 replace (~Q ∨ (P → R)) with (Q → (P → R)).
 apply (MP1_1 (P → Q → R) (Q → P → R)).
 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.
 replace (~P ∨ Q) with (P → Q).
 replace (~P ∨ R) with (P → R).
 apply (MP1_1 (Q → R) ((P → Q) → (P → R))).
 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.
  specialize MP1_1 with ((Q → R) → (P → Q) → P → R) ((P → Q) → ((Q → R
) → (P → R))).
  intros MP1_1.
  apply MP1_1.
  apply Comm2_04.
  apply Syll2_05.
Qed.

Theorem n2_07 : ∀ P : Prop,
  P → (P ∨ P).
Proof. intros P.
  specialize Add1_3 with P P.
  apply MP1_1.
Qed.

Theorem n2_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.
  specialize MP1_1 with ((P ∨ P) → P) (P → P).
  intros MP1_1.
  apply Syll2_05.
  apply Taut1_2.
  apply n2_07.
Qed.
```

```
Theorem n2_1 : ∀ P : Prop,
  (~P) ∨ P.
Proof. intros P.
  specialize n2_08 with P.
  replace (~P ∨ P) with (P → P).
  apply MP1_1.
  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 Abs2_01.
  apply Perm1_4.
  apply n2_1.
Qed.

Theorem n2_12 : ∀ P : Prop,
  P → ~~P.
Proof. intros P.
  specialize n2_11 with (~P).
  intros n2_11.
  rewrite Impl1_01.
  assumption.
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.
  apply Sum1_6.
  apply n2_12.
  apply n2_11.
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.
  rewrite Impl1_01.
  apply Perm1_4.
  apply n2_13.
Qed.

Theorem Trans2_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.
  specialize n2_03 with (~P) (~Q).
  intros n2_03.
  specialize Syll2_05 with (~Q) (~~P) P.
  intros Syll2_05b.
```

```
  specialize Syll2_05 with (~P → Q) (~P → ~~Q) (~Q → ~~P).
  intros Syll2_05c.
  specialize Syll2_05 with (~P → Q) (~Q → ~~P) (~Q → P).
  intros Syll2_05d.
  apply Syll2_05d.
  apply Syll2_05b.
  apply n2_14.
  apply Syll2_05c.
  apply n2_03.
  apply Syll2_05a.
  apply n2_12.
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.

Ltac MP H1 H2 :=
  match goal with
    | [ H1 : ?P → ?Q, H2 : ?P |- _ ] => specialize (H1 H2)
end.

Theorem Trans2_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 n2_03 with P (~Q).
```

```
  intros n2_03a.
  MP n2_12a Syll2_05a.
  Syll Syll2_05a n2_03a S.
  apply S.
Qed.

Theorem Trans2_17 : ∀ P Q : Prop,
  (~Q → ~P) → (P → Q).
Proof. intros P Q.
  specialize n2_03 with (~Q) P.
  intros n2_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 n2_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.

Axiom n2_33 : ∀ P Q R : Prop,
 (P∨Q∨R)=((P∨Q)∨R). (*This definition makes the default left associatio
n. The default in Coq is right association, so this will need to be applied to
 underwrite some inferences.*)

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_05a Perm1_4a.
 specialize Sum1_6 with P Q R.
 intros Sum1_6a.
 Syll Sum1_6a Syll2_05a 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 Trans2_16 with P (P∨Q).
  intros Trans2_16a.
  MP n2_2 Trans2_16a.
```

```
  apply Trans2_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 Trans2_16 with Q (P∨Q).
  intros Trans2_16a.
  MP Add1_3a Trans2_16a.
  apply Trans2_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 Trans2_17 with Q P.
  intros Trans2_17a.
  Syll n2_52a Trans2_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.
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_05a Perm1_4b.
  Syll Syll2_05a 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 explicitly!*)
  Syll Assoc1_5a n2_31a Sa.
  specialize n2_32 with P Q R.
  intros n2_32a. (*not cited explicitly!*)
  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 n2_02 with (~P) ((P∨Q→R)→(Q→R)).
  intros n2_02a. (*Not mentioned! Greg's suggestion per the BRS list in Ju
ne 25, 2017.*)
  MP Syll2_06a n2_02a.
  MP Hb n2_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.

Axiom Prod3_01 : ∀ P Q : Prop,
  (P ∧ Q) = ~(~P ∨ ~Q).

Axiom Abb3_02 : ∀ P Q R : Prop,
  (P→Q→R)=(P→Q)∧(Q→R).

Theorem Conj3_03 : ∀ P Q : Prop, P → Q → (P∧Q). (*3.03 is a derived rule
permitting an inference from the theoremhood of P and that of Q to that
of P and 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.

Theorem n3_1 : ∀ P Q : Prop,
  (P ∧ Q) → ~(~P ∨ ~Q).
Proof. intros P Q.
  replace (~(~P∨~Q)) with (P∧Q).
  specialize n2_08 with (P∧Q).
```

```
  intros n2_08a.
  apply n2_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 n2_08 with (P∧Q).
  intros n2_08a.
  apply n2_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 Trans2_15 with (~P∨~Q) (P∧Q).
  intros Trans2_15a.
```

```
  MP Trans2_16a n3_11a.
  apply Trans2_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 Trans2_16 with (P∧Q) (~(~P∨~Q)).
  intros Trans2_16a.
  MP Trans2_16a n3_1a.
  specialize n2_12 with (~P∨~Q).
  intros n2_12a.
  Syll n2_12a Trans2_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 Trans2_17 with (P∧Q) (Q ∧ P).
  intros Trans2_17a.
  MP Trans2_17a Hb.
  apply Trans2_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 n2_02 with Q P.
  intros n2_02a.
  replace (P→(Q→P)) with (~P∨(Q→P)) in n2_02a.
  replace (Q→P) with (~Q∨P) in n2_02a.
  specialize n2_31 with (~P) (~Q) P.
  intros n2_31a.
  MP n2_31a n2_02a.
  specialize n2_53 with (~P∨~Q) P.
  intros n2_53a.
  MP n2_53a n2_02a.
  replace (~(~P∨~Q)) with (P∧Q) in n2_53a.
  apply n2_53a.
  apply Prod3_01.
  replace (~Q∨P) with (Q→P).
  reflexivity.
  apply Impl1_01.
  replace (~P∨(Q→P)) with (P→Q→P).
  reflexivity.
  apply Impl1_01.
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 Trans2_15 with (~P∨~Q) R.
  intros Trans2_15a.
  replace (~R→(~P∨~Q)) with (~R→(P→~Q)) in Trans2_15a.
  specialize Comm2_04 with (~R) P (~Q).
  intros Comm2_04a.
  Syll Trans2_15a Comm2_04a Sa.
  specialize Trans2_17 with Q R.
  intros Trans2_17a.
  specialize  Syll2_05 with P (~R→~Q) (Q→R).
  intros Syll2_05a.
  MP Syll2_05a Trans2_17a.
  Syll Sa Syll2_05a Sb.
  replace (~(~P∨~Q)) with (P∧Q) in Sb.
  apply Sb.
  apply Prod3_01.
  replace (~P∨~Q) with (P→~Q).
  reflexivity.
  apply Impl1_01.
```

```
Qed.

Theorem Imp3_31 : ∀ 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_53 with (~P∨~Q) R.
  intros n2_53a.
  Syll n2_31a n2_53a S.
  replace (~Q∨R) with (Q→R) in S.
  replace (~P∨(Q→R)) with (P→Q→R) in S.
  replace (~(~P∨~Q)) with (P∧Q) in S.
  apply S.
  apply Prod3_01.
  apply Impl1_01.
  apply Impl1_01.
Qed.

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 n3_37 : ∀ P Q R : Prop,
  (P ∧ Q → R) → (P ∧ ~R → ~Q).
Proof. intros P Q R.
  specialize Trans2_16 with Q R.
  intros Trans2_16a.
  specialize Syll2_05 with P (Q→R) (~R→~Q).
  intros Syll2_05a.
  MP Syll2_05a Trans2_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 Trans2_15 with (P→Q) (P→~Q).
  intros Trans2_15a.
  MP Trans2_15a n2_51a.
  replace (P→~Q) with (~P∨~Q) in Trans2_15a.
  replace (~(~P∨~Q)) with (P∧Q) in Trans2_15a.
  apply Trans2_15a.
  apply Prod3_01.
  replace (~P∨~Q) with (P→~Q).
  reflexivity.
  apply Impl1_01.
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_05a 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 Trans2_16 with (Q→~R) (P→~R).
```

```
  intros Trans2_16a.
  Syll Syll2_06a Trans2_16a S.
  replace (P→~R) with (~P∨~R) in S.
  replace (Q→~R) with (~Q∨~R) in S.
  replace (~(~P∨~R)) with (P∧R) in S.
  replace (~(~Q∨~R)) with (Q∧R) in S.
  apply S.
  apply Prod3_01.
  apply Prod3_01.
  replace (~Q∨~R) with (Q→~R).
  reflexivity.
  apply Impl1_01.
  replace (~P∨~R) with (P→~R).
  reflexivity.
  apply Impl1_01.
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.
  specialize n2_83 with ((P→R)∧(Q→S)) (P∧Q) (Q∧R) (R∧S).
  intros n2_83a.
  MP n2_83a Sb.
  MP n2_83 Sd.
  apply n2_83a.
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.

Axiom Equiv4_01 : ∀ P Q : Prop,
  (P↔Q)=((P→Q) ∧ (Q→P)). (*n4_02 defines P iff Q iff R as P iff Q AND Q if
f R.*)

Axiom EqBi : ∀ P Q : Prop,
  (P=Q) ↔ (P↔Q).

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

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

Theorem Trans4_1 : ∀ P Q : Prop,
  (P → Q) ↔ (~Q → ~P).
Proof. intros P Q.
  specialize Trans2_16 with P Q.
  intros Trans2_16a.
  specialize Trans2_17 with P Q.
  intros Trans2_17a.
```

```
  Conj Trans2_16a Trans2_17a.
  split.
  apply Trans2_16a.
  apply Trans2_17a.
  Equiv H.
  apply H.
  apply Equiv4_01.
Qed.

Theorem Trans4_11 : ∀ P Q : Prop,
  (P ↔ Q) ↔ (~P ↔ ~Q).
Proof. intros P Q.
  specialize Trans2_16 with P Q.
  intros Trans2_16a.
  specialize Trans2_16 with Q P.
  intros Trans2_16b.
  Conj Trans2_16a Trans2_16b.
  split.
  apply Trans2_16a.
  apply Trans2_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 Trans2_16a. clear H. clear Trans2_16b. clear n3_22a. clear n3_47a.
  specialize Trans2_17 with Q P.
  intros Trans2_17a.
  specialize Trans2_17 with P Q.
```

```
  intros Trans2_17b.
  Conj Trans2_17a Trans2_17b.
  split.
  apply Trans2_17a.
  apply Trans2_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 Trans2_17a. clear Trans2_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 n2_03 with P Q.
    intros n2_03a.
```

```
  specialize Trans2_15 with Q P.
  intros Trans2_15a.
  Conj n2_03a Trans2_15a.
  split.
  apply n2_03a.
  apply Trans2_15a.
  specialize n3_47 with (P→~Q) (~Q→P) (Q→~P) (~P→Q).
  intros n3_47a.
  MP n3_47a H.
  specialize n2_03 with Q P.
  intros n2_03b.
  specialize Trans2_15 with P Q.
  intros Trans2_15b.
  Conj n2_03b Trans2_15b.
  split.
  apply n2_03b.
  apply Trans2_15b.
  specialize n3_47 with (Q→~P) (~P→Q) (P→~Q) (~Q→P).
  intros n3_47b.
  MP n3_47b H0.
  clear n2_03a. clear Trans2_15a. clear H. clear n2_03b. clear Trans2_15b
. clear H0.
  replace ((P → ¬ Q) ∧ (~Q → P)) with (P↔~Q) in n3_47a.
  replace ((Q → ~P) ∧ (~P → Q)) with (Q↔~P) in n3_47a.
  replace ((P → ¬ Q) ∧ (~Q → P)) with (P↔~Q) in n3_47b.
  replace ((Q → ~P) ∧ (~P → Q)) with (Q↔~P) in n3_47b.
  Conj n3_47a n3_47b.
  split.
  apply n3_47a.
  apply n3_47b.
  Equiv H.
  apply H.
```

```
   apply Equiv4_01.
   apply Equiv4_01.
   apply Equiv4_01.
   apply Equiv4_01.
   apply Equiv4_01.
  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 n3_37 with P Q R.
intros n3_37a.
specialize n3_37 with P (~R) (~Q).
intros n3_37b.
Conj n3_37a n3_37b.
split. apply n3_37a.
```

```
apply n3_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 EqBi.
apply n4_13b.
apply EqBi.
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.
 split.
 apply Sa.
 apply Sb.
 apply EqBi.
 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 n2_08 with P.
 intros n2_08a.
 MP n3_2a n2_08a.
 MP n3_2a n2_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.
 specialize Equiv4_01 with P Q.
 intros Equiv4_01a.
 replace ((P → Q) ∧ (Q → P)) with (P↔Q) in n3_22a.
 specialize Equiv4_01 with Q P.
 intros Equiv4_01b.
 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 Equiv4_01b.
 apply n3_22b.
 split.
 apply n3_22a.
 apply n3_22b.
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. cl
ear 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 Trans4_1 with P (~(Q ∧ R)).
  intros Trans4_1a.
  replace (~~(Q ∧ R)) with (Q ∧ R) in Trans4_1a.
  replace (Q ∧ R→~P) with (P→~(Q ∧ R)) in n4_15a.
  specialize Trans4_11 with (P ∧ Q → ¬ R) (P → ¬ (Q ∧ R)).
  intros Trans4_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 EqBi.
  apply Trans4_11a.
  apply EqBi.
  apply Trans4_1a.
  apply EqBi.
  apply n4_13.
  Qed. (*Note that the actual proof uses n4_12, but that transposition inv
olves transforming a biconditional into a conditional. This way of doing it
 - using Trans4_1 to transpose a conditional and then applying n4_13 to d
ouble negate - is easier without a derived rule for replacing a bicondition
al with one of its equivalent implications.*)

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.
  split. apply n2_31a.
  apply n2_32a.
 Qed.
```

```
 Axiom n4_34 : ∀ P Q R : Prop,
 P ∧ Q ∧ R = ((P ∧ Q) ∧ R). (*This axiom ensures left association of bracke
ts. Coq's default is right association. But Principia proves associativity of l
ogical product as n4_32. So in effect, this axiom gives us a derived rule th
at allows us to shift between Coq's and Principia's default rules for brack
ets of logical products.*)

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_4
7a.
 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_4
7a.
  replace (R ∨ P) with (P ∨ R) in n3_47a.
  replace (R ∨ Q) with (Q ∨ R) in n3_47a.
  apply n3_47a.
  apply EqBi.
  apply n4_31.
  apply EqBi.
  apply n4_31.
  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 n
3_47c.
  apply n3_47c.
  apply Equiv4_01.
  apply Equiv4_01.
  apply Equiv4_01.
  apply EqBi.
  apply n4_32d.
  replace ((R → P) ∧ (Q → S) ∧ (S → Q)) with (((R → P) ∧ (Q → S)) ∧ (S → Q
)).
  reflexivity.
  apply EqBi.
  apply n4_32c.
  replace ((R → P) ∧ (Q → S)) with ((Q → S) ∧ (R → P)).
  reflexivity.
  apply EqBi.
  apply H0.
  apply Equiv4_01.
  apply EqBi.
  apply n4_32b.
  replace ((P → R) ∧ (Q → S) ∧ (R → P) ∧ (S → Q)) with (((P → R) ∧ (Q → S)
) ∧ (R → P) ∧ (S → Q)).
  reflexivity.
  apply EqBi.
  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 n
3_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 EqBi.
 apply n4_32d.
 apply Equiv4_01.
 apply EqBi.
 apply n4_32c.
 replace ((R → P) ∧ (Q → S)) with ((Q → S) ∧ (R → P)).
 reflexivity.
 apply EqBi.
 apply H0.
 apply Equiv4_01.
 replace ((P → R) ∧ (Q → S) ∧ (R → P)) with (((P → R) ∧ (Q → S)) ∧ (R → P
)).
 reflexivity.
 apply EqBi.
```

```
  apply n4_32b.
  apply EqBi.
  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 Si
mp3_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_6b.
  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.
  split.
  apply Comp3_43a.
  apply Sb.
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 EqBi.
  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. cl
ear 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 EqBi.
  apply n4_13.
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 n2_08 with P.
    intros n2_08a.
    specialize Simp3_26 with P Q.
    intros Simp3_26a.
    Conj n2_08a Simp3_26a.
    split.
    apply n2_08a.
    apply Simp3_26a.
```

```
  specialize n3_44 with P P (P ∧ Q).
  intros n3_44a.
  MP n3_44a H.
  clear H. clear n2_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.
 split.
 apply n2_2a.
 apply Simp3_26a.
 apply EqBi.
 apply n4_24.
 apply EqBi.
 apply n4_4.
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.
  replace ((P ∧ Q ↔ ¬ (¬ P ∨ ¬ Q)) ↔ (¬ P ∨ ¬ Q ↔ ¬ (P ∧ Q))) with ((P ∧ Q
 ↔ ¬ (¬ P ∨ ¬ Q)) = (¬ P ∨ ¬ Q ↔ ¬ (P ∧ Q))) in n4_12a.
  replace (P ∧ Q ↔ ¬ (¬ P ∨ ¬ Q)) with (¬ P ∨ ¬ Q ↔ ¬ (P ∧ Q)) in n4_5a.
  replace (¬ P ∨ ¬ Q ↔ ¬ (P ∧ Q)) with (~(P ∧ Q) ↔ (~P ∨ ~Q)) in n4_5a.
  apply n4_5a.
  specialize n4_21 with (¬ (P ∧ Q)) (¬ P ∨ ¬ Q).
  intros n4_21a.
  apply EqBi.
  apply n4_21.
  apply EqBi.
  apply EqBi.
 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 EqBi.
  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))) in n4_12a.
  replace (P ∧ ¬ Q ↔ ¬ (¬ P ∨ Q)) with (¬ P ∨ Q ↔ ¬ (P ∧ ¬ Q)) in n4_52a.
  replace (¬ P ∨ Q ↔ ¬ (P ∧ ¬ Q)) with (~(P ∧ ~Q) ↔ (~P ∨ Q)) in n4_52a.
  apply n4_52a.
  specialize n4_21 with (¬ (P ∧ ¬ Q)) (¬ P ∨ Q).
  intros n4_21a.
  apply EqBi.
  apply n4_21a.
  apply EqBi.
  apply EqBi.
 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 EqBi.
  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.
  apply EqBi.
  apply n4_21a.
  replace ((¬ P ∧ Q ↔ ¬ (P ∨ ¬ Q)) ↔ (P ∨ ¬ Q ↔ ¬ (¬ P ∧ Q))) with ((¬ P ∧
 Q ↔ ¬ (P ∨ ¬ Q)) = (P ∨ ¬ Q ↔ ¬ (¬ P ∧ Q))) in n4_12a.
  rewrite n4_12a.
  reflexivity.
  apply EqBi.
  apply EqBi.
```

```
 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 EqBi.
  apply n4_13.
 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 EqBi.
  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 EqBi.
  apply n4_12a.
  apply EqBi.
```

```
  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 Trans4_11 with (P→Q) (~P∨Q).
 intros Trans4_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 EqBi.
 apply n4_52a.
 replace (((P → Q) ↔ ¬ P ∨ Q) ↔ (¬ (P → Q) ↔ ¬ (¬ P ∨ Q))) with ((¬ (P →
Q) ↔ ¬ (¬ P ∨ Q)) ↔ ((P → Q) ↔ ¬ P ∨ Q)) in Trans4_11a.
 apply EqBi.
 apply Trans4_11a.
```

```
  apply EqBi.
  apply n4_21.
  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 Trans4_11 with (P → ¬ Q) (¬ P ∨ ¬ Q).
    intros Trans4_11a.
    specialize n4_5 with P Q.
    intros n4_5a.
    replace (¬ (¬ P ∨ ¬ Q)) with (P ∧ Q) in Trans4_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 Trans4_11a.
    apply EqBi.
    apply Trans4_11a.
    specialize n4_21 with (¬ (P → ¬ Q) ↔ P ∧ Q) ((P → ¬ Q) ↔ ¬ P ∨ ¬ Q).
    intros n4_21a.
    apply EqBi.
    apply n4_21a.
```

```
    apply EqBi.
    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 Trans4_11 with(¬ P → Q) (P ∨ Q).
  intros Trans4_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 Trans4_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 EqBi.
 apply n4_56a.
 apply EqBi.
 apply Trans4_11a.
 apply EqBi.
 apply n4_21.
 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 Trans4_11 with (¬ P → ¬ Q) (P ∨ ¬ Q).
 intros Trans4_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 EqBi.
```

```
 apply n4_54a.
 replace (((¬ P → ¬ Q) ↔ P ∨ ¬ Q) ↔ (¬ (¬ P → ¬ Q) ↔ ¬ (P ∨ ¬ Q))) with (
(¬ (¬ P → ¬ Q) ↔ ¬ (P ∨ ¬ Q)) ↔ ((¬ P → ¬ Q) ↔ P ∨ ¬ Q)) in Trans4_11a.
 apply EqBi.
 apply Trans4_11a.
 apply EqBi.
 apply n4_21.
 Qed.

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 n2_08 with P.
 intros n2_08a.
 MP Exp3_3a n2_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_26a.
 clear n2_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. clea
r 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 Trans4_1 with P Q.
  intros Trans4_1a.
  specialize n4_71 with (~Q) (~P).
  intros n4_71a.
  Conj Trans4_1a n4_71a.
  split.
  apply Trans4_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 EqBi.
  apply n4_31a.
  apply EqBi.
  replace (¬ (¬ Q ∧ ¬ P) ↔ Q ∨ P) with (Q ∨ P ↔¬ (¬ Q ∧ ¬ P)) in n4_57a.
  apply n4_57a.
  apply EqBi.
  apply n4_21.
  Qed.

Theorem n4_73 : ∀ P Q : Prop,
  Q → (P ↔ (P ∧ Q)).
  Proof. intros P Q.
  specialize n2_02 with P Q.
  intros n2_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 n2_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 EqBi.
 replace ((P → Q) ↔ (Q ↔ P ∨ Q)) with ((Q ↔ P ∨ Q) ↔ (P → Q)) in n4_72
a.
 apply n4_72a.
 apply EqBi.
 apply n4_21.
 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 EqBi.
 apply n4_21.
 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.
 split.
 apply n3_44a.
 split.
 specialize n2_2 with Q R.
 intros n2_2a.
 Syll n2_2a H Sa.
 apply Sa.
 specialize Add1_3 with Q R.
 intros Add1_3a.
 Syll Add1_3a H Sb.
 apply Sb.
 Qed. (*Note that we used the split tactic on a conditional, effectively intr
oducing an assumption for conditional proof. It remains to prove that (A
vB)→C and A→(AvB) together imply A→C, and similarly that (AvB)→C an
d B→(AvB) together imply B→C. This can be proved by Syll, but we need
a rule of replacement in the context of ((AvB)→C)→(A→C)/\(B→C).*)
```

```
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 EqBi.
```

```
  apply n4_37b.
  apply n2_33.
  replace ((¬ P ∨ ¬ P) ∨ Q) with (¬ P ∨ ¬ P ∨ Q).
  reflexivity.
  apply n2_33.
  replace ((¬ P ∨ ¬ P ∨ Q) ∨ R) with (¬ P ∨ (¬ P ∨ Q) ∨ R).
  reflexivity.
  apply EqBi.
  apply n4_33b.
  apply EqBi.
  apply n4_37a.
  replace ((Q ∨ ¬ P) ∨ R) with (Q ∨ ¬ P ∨ R).
  reflexivity.
  apply n2_33.
  apply EqBi.
  apply n4_33a.
  replace (¬ P ∨ Q) with (P→Q).
  replace (¬ P ∨ R) with (P→R).
  reflexivity.
  apply Impl1_01.
  apply Impl1_01.
  Qed.

Theorem n4_79 : ∀ P Q R : Prop,
  ((Q → P) ∨ (R → P)) ↔ ((Q ∧ R) → P).
  Proof. intros P Q R.
    specialize Trans4_1 with Q P.
    intros Trans4_1a.
    specialize Trans4_1 with R P.
    intros Trans4_1b.
    Conj Trans4_1a Trans4_1b.
    split.
```

```
  apply Trans4_1a.
  apply Trans4_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.
  replace ((¬ P → ¬ Q) ∨ (¬ P → ¬ R)) with (¬ P → ¬ Q ∨ ¬ R) in n4_39a.
  specialize Trans2_15 with P (~Q ∨ ~R).
  intros Trans2_15a.
  replace (¬ P → ¬ Q ∨ ¬ R) with (¬ (¬ Q ∨ ¬ R) → P) in n4_39a.
  replace (~(~Q ∨ ~R)) with (Q ∧ R) in n4_39a.
  apply n4_39a.
  apply Prod3_01.
  replace (¬ (¬ Q ∨ ¬ R) → P) with (¬ P → ¬ Q ∨ ¬ R).
  reflexivity.
  apply EqBi.
  split.
  apply Trans2_15a.
  apply Trans2_15.
  replace (¬ P → ¬ Q ∨ ¬ R) with ((¬ P → ¬ Q) ∨ (¬ P → ¬ R)).
  reflexivity.
  apply EqBi.
  apply n4_78a.
 Qed.

Theorem n4_8 : ∀ P : Prop,
 (P → ~P) ↔ ~P.
 Proof. intros P.
  specialize Abs2_01 with P.
  intros Abs2_01a.
  specialize  n2_02 with P (~P).
```

```
  intros n2_02a.
  Conj Abs2_01a n2_02a.
  split.
  apply Abs2_01a.
  apply n2_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  n2_02 with (~P) P.
  intros n2_02a.
  Conj n2_18a n2_02a.
  split.
  apply n2_18a.
  apply n2_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 n2_02 with P Q.
  intros n2_02a.
  specialize n2_02 with (~P) Q.
  intros n2_02b.
  Conj n2_02a n2_02b.
  split.
  apply n2_02a.
  apply n2_02b.
  specialize Comp3_43 with Q (P→Q) (~P→Q).
  intros Comp3_43a.
  MP Comp3_43a H.
  clear n2_61a. clear n2_02a. clear n2_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 EqBi.
  apply n4_21.
  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.
  split.
  split.
  replace (P↔Q) with (Q↔P) in H.
  Conj H H0.
  split.
  apply H.
  apply H0.
  specialize n4_22 with  Q P R.
  intros n4_22a.
  MP n4_22a H1.
  replace (Q ↔ R) with ((Q→R) ∧ (R→Q)) in n4_22a.
  specialize Simp3_26 with (Q→R) (R→Q).
  intros Simp3_26a.
  MP Simp3_26a n4_22a.
  apply Simp3_26a.
  apply Equiv4_01.
  apply EqBi.
  apply n4_21.
  replace (P↔R) with (R↔P) in H0.
  Conj H0 H.
  split.
  apply H.
  apply H0.
  replace ((P ↔ Q) ∧ (R ↔ P)) with ((R ↔ P) ∧ (P ↔ Q)) in H1.
```

```
specialize n4_22 with R P Q.
intros n4_22a.
MP n4_22a H1.
replace (R ↔ Q) with ((R→Q) ∧ (Q→R)) in n4_22a.
specialize Simp3_26 with (R→Q) (Q→R).
intros Simp3_26a.
MP Simp3_26a n4_22a.
apply Simp3_26a.
apply Equiv4_01.
apply EqBi.
apply n4_3.
apply EqBi.
apply n4_21.
split.
Conj H H0.
split.
apply H.
apply H0.
specialize n4_22 with P Q R.
intros n4_22a.
MP n4_22a H1.
replace (P↔R) with ((P→R)∧(R→P)) in n4_22a.
specialize Simp3_26 with (P→R) (R→P).
intros Simp3_26a.
MP Simp3_26a n4_22a.
apply Simp3_26a.
apply Equiv4_01.
Conj H H0.
split.
apply H.
apply H0.
specialize n4_22 with P Q R.
```

```
 intros n4_22a.
 MP n4_22a H1.
 replace (P↔R) with ((P→R)∧(R→P)) in n4_22a.
 specialize Simp3_27 with (P→R) (R→P).
 intros Simp3_27a.
 MP Simp3_27a n4_22a.
 apply Simp3_27a.
 apply Equiv4_01.
 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 Co
mm2_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 EqBi.
 apply H1.
 replace (Q → P → R) with (Q ∧ P → R).
 reflexivity.
 apply EqBi.
 apply H0.
 replace (P → Q → R) with (P ∧ Q → R).
 reflexivity.
 apply EqBi.
 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.
  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 EqBi.
 apply n4_76a.
 Qed. (*Note that n4_76 is not cited, but it is used to move from ((a→b) ∧
(a→c)) to (a→ (b ∧ c)).*)

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.

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.

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 EqBi.
 apply n4_13.
 replace (~~ (P → Q) ∨ (Q → P)) with (~ (P → Q) → Q → P).
 reflexivity.
 apply Impl1_01.
 Qed. (*n4_13 is not cited, but is needed for double negation elimination.
*)

Theorem n5_14 : ∀ P Q R : Prop,
 (P → Q) ∨ (Q → R).
 Proof. intros P Q R.
 specialize n2_02 with P Q.
 intros n2_02a.
 specialize Trans2_16 with Q (P→Q).
 intros Trans2_16a.
 MP Trans2_16a n2_02a.
 specialize n2_21 with Q R.
 intros n2_21a.
 Syll Trans2_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 EqBi.
 apply n4_13.
 replace (~~(P→Q)∨(Q→R)) with (~(P→Q)→(Q→R)).
 reflexivity.
 apply Impl1_01.
 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. c
lear 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 EqBi.
 apply n4_31.
 apply Equiv4_01.
 apply EqBi.
 apply n4_41a.
 apply EqBi.
 apply n4_31.
 apply EqBi.
 apply n4_31.
 apply EqBi.
 apply n4_13.
```

```
 replace (~~(Q → P) ∨ (P ↔ ~Q)) with (¬ (Q → P) → (P ↔ ~Q)).
 reflexivity.
 apply Impl1_01.
 apply EqBi.
 apply n4_13.
 replace (~~ (P → Q) ∨ (P ↔ ¬ Q)) with (¬ (P → Q) → P ↔ ¬ Q).
 reflexivity.
 apply Impl1_01.
 apply EqBi.
 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 n2_08 with ((P ↔ Q) ∧ (P → ~Q)).
 intros n2_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 n2_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 n2_08a. clear n4_82a. clear Simp3_27a. c
lear 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 E
xp3_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 Exp
3_3a.
apply Exp3_3a.
apply EqBi.
apply n4_51b.
replace (¬ (P ↔ Q) ∨ ¬ (P ↔ ¬ Q)) with (P ↔ Q → ¬ (P ↔ ¬ Q)).
reflexivity.
apply Impl1_01.
apply Equiv4_01.
apply EqBi.
apply n4_51a.
replace (¬ (P → ¬ Q) ∨ ¬ (¬ Q → P)) with ((P → ¬ Q) → ¬ (¬ Q → P)).
reflexivity.
apply Impl1_01.
apply EqBi.
apply n4_65a.
apply EqBi.
apply n4_3.
apply Equiv4_01.
apply EqBi.
apply n4_32.
replace (¬ P) with ((P → Q) ∧ (P → ¬ Q)).
reflexivity.
apply EqBi.
apply n4_82a.
apply Equiv4_01.
apply EqBi.
apply n4_32.
apply EqBi.
```

```
 apply n4_3.
 replace ((P → Q) ∧ (P → ¬ Q) ∧ (Q → P)) with (((P → Q) ∧ (P → ¬ Q)) ∧ (Q
→ P)).
 reflexivity.
 apply EqBi.
 apply n4_32.
 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 Trans4_11 with (P∧Q) (~(P→~Q)).
 intros Trans4_11a.
 replace (~~(P→~Q)) with (P→~Q) in Trans4_11a.
 replace (P ∧ Q ↔ ¬ (P → ¬ Q)) with (¬ (P ∧ Q) ↔ (P → ¬ Q)) in n4_63a.
 clear Trans4_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 EqBi.
 apply n4_21b.
 apply Equiv4_01.
 replace (¬ (P ∧ Q) ↔ (P → ¬ Q)) with (P ∧ Q ↔ ¬ (P → ¬ Q)).
 reflexivity.
 apply EqBi.
 apply Trans4_11a.
 apply EqBi.
 apply n4_13.
 apply EqBi.
 apply n4_21.
 apply EqBi.
 apply n4_31.
 apply EqBi.
 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 EqBi.
 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 EqBi.
 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 Trans4_11 with P Q.
 intros Trans4_11a.
 replace (~P↔~Q) with (P↔Q) in n5_1a.
 apply n5_1a.
```

```
 apply EqBi.
 apply Trans4_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 EqBi.
 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.
 specialize n4_13 with Q.
 intros n4_13a.
 replace (~(P↔~Q)) with ((P ∧ ~~Q) ∨ (~Q ∧ ~P)) in n5_18a.
 replace (~~Q) with Q in n5_18a.
 replace (~Q ∧ ~P) with (~P ∧ ~Q) in n5_18a.
 apply n5_18a.
 apply EqBi.
 apply n4_3. (*with (~P) (~Q)*)
 apply EqBi.
 apply n4_13a.
 replace (P∧~~Q∨~Q∧~P) with (~(P↔~Q)).
 reflexivity.
 apply EqBi.
 apply n5_22a.
 Qed. (*The proof sketch in Principia offers n4_36, but we found it far si
mpler to simply use the commutativity of conjunction (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 EqBi.
 apply n5_23a.
 replace (~(P ↔ Q) ↔ ~(P ∧ Q ∨ ~P ∧ ~Q)) with ((P ↔ Q) ↔ P ∧ Q ∨ ~P
∧ ~Q).
 reflexivity.
 specialize Trans4_11 with (P↔Q) (P ∧ Q ∨ ~P ∧ ~Q).
 intros Trans4_11a.
 apply EqBi.
 apply Trans4_11a.
 Qed. (*Note that Trans4_11 is not cited explicitly.*)

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.
  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. (*Note that Exp is not cited in the proof sketch, but seems necessar
y.*)

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 n2_02 with P R.
  intros n2_02a.
  replace ((P→Q) ∧ (P→R)) with ((P→R) ∧ (P→Q)) in Comp3_43a.
  specialize Exp3_3 with (P→R) (P→Q) (P→(Q ∧ R)).
```

```
 intros Exp3_3a.
 MP Exp3_3a Comp3_43a.
 Syll n2_02a Exp3_3a Sa.
 specialize Imp3_31 with R (P→Q) (P→(Q ∧ R)).
 intros Imp3_31a.
 MP Imp3_31a Sa.
 apply Imp3_31a.
 apply EqBi.
 apply n4_3. (*with (P→R)∧(P→Q)).*)
 Qed. (*Note that Exp, Imp, and n4_3 are not cited in the proof sketch.*)

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 EqBi.
apply n4_3. (*to commute the biconditional to get the theorem.*)
apply Equiv4_01.
replace (P ∧ R → P ∧ Q) with (P ∧ R → Q).
reflexivity.
apply EqBi.
apply n5_3b.
apply EqBi.
apply H0.
replace (P ∧ Q → P ∧ R) with (P ∧ Q → R).
reflexivity.
apply EqBi.
apply n5_3a.
apply EqBi.
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)))). (*Not cited
.*)
  intros Simp3_26a.
  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 EqBi.
  apply n4_3. (*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 n2_08 with (P↔Q).
 intros n2_08a.
 replace (P↔Q→P↔Q) with ((P↔Q)∧P↔(P↔Q)∧Q) in n2_08a.
 replace ((P↔Q)∧P) with (P∧(P↔Q)) in n2_08a.
 replace ((P↔Q)∧Q) with (Q∧(P↔Q)) in n2_08a.
 apply n2_08a.
 apply EqBi.
 apply n4_3.
 apply EqBi.
 apply n4_3.
 replace ((P ↔ Q) ∧ P ↔ (P ↔ Q) ∧ Q) with (P ↔ Q → P ↔ Q).
 reflexivity.
 apply EqBi.
 apply n5_32a.
```

```
 Qed. (*The proof sketch cites Ass3_35 and n4_38. Since I couldn't deciph
er how that proof would go, I used a different one invoking other theore
ms.*)

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 n2_02 with (P) (P→Q).
 intros n2_02a.
 Conj n2_43a n2_02a.
 split.
 apply n2_43a.
 apply n2_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 EqBi.
  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 EqBi.
  apply H.
  apply Equiv4_01.
  Qed. (*The law n4_87 is really unwieldy to use in Coq. It is actually easie
r to introduce the subformula of the importation-
exportation law required and apply that biconditional. It may be worthw
hile in later parts of PM to prove a derived rule that allows us to manipul
ate a biconditional's subformulas that are biconditionals.*)

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.
  replace ((P→Q)∧(P→R)↔(P→Q∧R)) with (((P→Q)∧(P→R)→(P→Q∧R))∧
((P→Q∧R)→(P→Q)∧(P→R))) in n4_76a.
  specialize Simp3_26 with ((P→Q)∧(P→R)→(P→Q∧R)) ((P→Q∧R)→(P→Q
)∧(P→R)).
  intros Simp3_26a. (*Not cited.*)
  MP Simp3_26a n4_76a.
  specialize Exp3_3 with (P→Q) (P→R) (P→Q∧R).
  intros Exp3_3a. (*Not cited.*)
  MP Exp3_3a Simp3_26a.
  specialize Simp3_27 with ((P→Q)∧(P→R)→(P→Q∧R)) ((P→Q∧R)→(P→Q
)∧(P→R)).
  intros Simp3_27a. (*Not cited.*)
  MP Simp3_27a n4_76a.
```

```
 specialize Simp3_26 with (P→R) (P→Q).
 intros Simp3_26b.
 replace ((P→Q)∧(P→R)) with ((P→R)∧(P→Q)) in Simp3_27a.
 Syll Simp3_27a Simp3_26b Sa.
 specialize n2_02 with (P→Q) ((P→Q∧R)→P→R).
 intros n2_02a. (*Not cited.*)
 MP n2_02a Sa.
 clear Sa. clear Simp3_26b. clear Simp3_26a. clear n4_76a. clear Simp3_2
7a.
 Conj Exp3_3a n2_02a.
 split.
 apply Exp3_3a.
 apply n2_02a.
 specialize n4_76 with (P→Q) ((P→R)→(P→(Q∧R))) ((P→(Q∧R))→(P→R)
).
 intros n4_76b.
 replace (((P→Q)→(P→R)→P→Q∧R)∧((P→Q)→(P→Q∧R)→P→R)) with ((
P→Q)→((P→R)→P→Q∧R)∧((P→Q∧R)→P→R)) in H.
 replace (((P→R)→P→Q∧R)∧((P→Q∧R)→P→R)) with ((P→R)↔(P→Q∧R)
) in H.
 apply H.
 apply Equiv4_01.
 replace ((P→Q)→((P→R)→P→Q∧R)∧((P→Q∧R)→P→R)) with (((P→Q)→
(P→R)→P→Q∧R)∧((P→Q)→(P→Q∧R)→P→R)).
 reflexivity.
 apply EqBi.
 apply n4_76b.
 apply EqBi.
 apply n4_3. (*Not cited.*)
 apply Equiv4_01.
 Qed. (*This proof does not use either n5_3 or n5_32. It instead uses four
propositions not cited in the proof sketch, plus a second use of n4_76.*)
```

```
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 n2_02 with P Q.
  intros n2_02a.
  specialize Exp3_3 with P Q (P→Q).
  intros Exp3_3b.
  specialize n3_42 with P Q (P→Q). (*Not mentioned explicitly.*)
  intros n3_42a.
  MP n3_42a n2_02a.
  MP Exp3_3b n3_42a.
  clear n3_42a. clear n2_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 EqBi.
  apply n4_24. (*with P.*)
  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 EqBi.
  apply n4_76a.
  apply Equiv4_01.
  replace (P∧(P→Q)∧(Q→P)) with ((P∧(P→Q))∧(Q→P)).
  reflexivity.
  apply EqBi.
  apply n4_32. (*Not cited.*)
  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 EqBi.
  apply n4_3. (*Not cited explicitly.*)
  apply EqBi.
  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 Trans2_16 with Q (P↔(P∧Q)).
intros Trans2_16a.
MP n4_73a Trans2_16a.
specialize Trans4_11 with Q (Q∨(P∧Q)).
intros Trans4_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 Trans2_16a.
replace (~(Q∨P∧Q)) with (~Q∧~(P∧Q)) in Trans2_16a.
specialize n5_1 with (~Q) (~(P∧Q)).
intros n5_1a.
Syll Trans2_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 Trans4_11 with Q (P∧Q).
intros Trans4_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 EqBi.
apply n4_21. (*Not cited.*)
apply EqBi.
apply n4_21.
apply EqBi.
apply Trans4_11b.
apply EqBi.
apply n4_13. (*Not cited.*)
```

```
  replace (~~(P↔P∧Q)∨(~Q↔~(P∧Q))) with (~(P↔P∧Q)→~Q↔~(P∧Q)
).
  reflexivity.
  apply Impl1_01. (*Not cited.*)
  apply EqBi.
  apply n4_56. (*Not cited.*)
  replace (~(Q∨P∧Q)) with (~Q).
  reflexivity.
  apply EqBi.
  apply n4_44a.
  replace (~Q↔~(Q∨P∧Q)) with (Q↔Q∨P∧Q).
  reflexivity.
  apply EqBi.
  apply Trans4_11a.
  apply EqBi.
  apply n4_3. (*Not cited.*)
  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 Trans2_15 with P (Q↔P∨Q).
```

```
  intros Trans2_15a. (*Not cited.*)
  MP Trans2_15a n4_74a.
  Syll Trans2_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 EqBi.
  apply n4_31. (*Not cited.*)
  apply EqBi.
  apply n4_21. (*Not cited.*)
  apply EqBi.
  apply n4_21.
  apply EqBi.
  apply n4_13. (*Not cited.*)
  replace (~~(Q↔P∨Q)∨(P↔P∨Q)) with (~(Q↔P∨Q)→P↔P∨Q).
  reflexivity.
  apply Impl1_01.
  apply EqBi.
  apply n4_31.
  apply EqBi.
  apply n4_25. (*Not cited.*)
  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 EqBi.
  apply n4_41. (*Not cited.*)
  apply EqBi.
```

```
 apply n4_31.
 apply EqBi.
 apply n4_31.
 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 EqBi.
 apply n4_64a.
 apply Equiv4_01.
 apply Equiv4_01.
 Qed. (*A fair amount of manipulation was needed here to pull the releva
nt biconditional out of the biconditional of biconditionals.*)

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_7
4a.
 apply n4_74a.
 apply EqBi.
 apply n4_3. (*Not cited exlicitly.*)
 apply EqBi.
 apply n4_31. (*Not cited explicitly.*)
 apply EqBi.
 apply n4_3. (*Not cited explicitly.*)
```

```
 apply EqBi.
 apply n4_3. (*Not cited explicitly.*)
 replace (¬ Q ∧ P ↔ ¬ Q ∧ (Q ∨ P)) with (¬ Q → P ↔ Q ∨ P).
 reflexivity.
 apply EqBi.
 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 EqBi.
 apply n4_21. (*Not cited explicitly.*)
 apply EqBi.
 apply n4_3. (*Not cited explicitly.*)
 apply EqBi.
 apply n4_31. (*Not cited explicitly.*)
 apply EqBi.
 apply n4_31. (*Not cited explicitly.*)
 replace (¬ Q ∨ Q ∧ P) with (Q → Q ∧ P).
 reflexivity.
 apply EqBi.
 apply n4_6. (*Not cited explicitly.*)
```

```
 replace (¬ Q ∨ P) with (Q → P).
 reflexivity.
 apply EqBi.
 apply n4_6. (*Not cited explicitly.*)
 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 EqBi.
 apply n4_3. (*Not cited explicitly.*)
 apply EqBi.
 apply n4_21. (*Not cited explicitly.*)
 apply EqBi.
 apply n4_31. (*Not cited explicitly.*)
 apply EqBi.
 apply n4_31. (*Not cited explicitly.*)
 apply EqBi.
 apply n4_13. (*Not cited explicitly.*)
 Qed.

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

```
specialize n5_32 with (~R) (~P) (~Q).
intros n5_32a. (*Not cited.*)
replace (~R∧~P) with (~(R∨P)) in n5_32a.
replace (~R∧~Q) with (~(R∨Q)) in n5_32a.
replace ((~(R∨P))↔(~(R∨Q))) with ((R∨P)↔(R∨Q)) in n5_32a.
replace ((~P)↔(~Q)) with (P↔Q) in n5_32a.
replace (~R→(P↔Q)) with (~~R∨(P↔Q)) in n5_32a.
replace (~~R) with R in n5_32a.
replace (R∨P) with (P∨R) in n5_32a.
replace (R∨Q) with (Q∨R) in n5_32a.
replace ((R∨(P↔Q))↔(P∨R↔Q∨R)) with ((P∨R↔Q∨R)↔(R∨(P↔Q))) in
n5_32a.
apply n5_32a. (*Not cited.*)
apply EqBi.
apply n4_21. (*Not cited.*)
apply EqBi.
apply n4_31.
apply EqBi.
apply n4_31.
apply EqBi.
apply n4_13. (*Not cited.*)
replace (~~R∨(P↔Q)) with (~R→P↔Q).
reflexivity.
apply Impl1_01. (*Not cited.*)
apply EqBi.
apply Trans4_11. (*Not cited.*)
apply EqBi.
apply Trans4_11.
replace (~(R∨Q)) with (~R∧~Q).
reflexivity.
apply EqBi.
apply n4_56. (*Not cited.*)
```

```
  replace (~(R∨P)) with (~R∧~P).
  reflexivity.
  apply EqBi.
  apply n4_56.
  Qed. (*The proof sketch was indecipherable, but an easy proof was avail
able through n5_32.*)

Theorem n5_71 : ∀ P Q R : Prop,
  (Q → ~R) → (((P ∨ Q) ∧ R) ↔ (P ∧ R)).
  Proof. intros P Q R.
  specialize n4_4 with R P Q.
  intros n4_4a.
  specialize n4_62 with Q R.
  intros n4_62a.
  specialize n4_51 with Q R.
  intros n4_51a.
  replace (~Q∨~R) with (~(Q∧R)) in n4_62a.
  replace ((Q→~R)↔~(Q∧R)) with (((Q→~R)→~(Q∧R))∧(~(Q∧R)→(Q→
~R))) in n4_62a.
  specialize Simp3_26 with ((Q→~R)→~(Q∧R)) (~(Q∧R)→(Q→~R)).
  intros Simp3_26a.
  MP Simp3_26a n4_62a.
  specialize n4_74 with (Q∧R) (P∧R).
  intros n4_74a.
  Syll Simp3_26a n4_74a Sa.
  replace (R∧P) with (P∧R) in n4_4a.
  replace (R∧Q) with (Q∧R) in n4_4a.
  replace ((P∧R)∨(Q∧R)) with ((Q∧R)∨(P∧R)) in n4_4a.
  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 EqBi.
  apply n4_21. (*Not cited.*)
  apply EqBi.
  apply n4_3. (*Not cited.*)
  apply EqBi.
  apply n4_4a. (*Not cited.*)
  apply EqBi.
  apply n4_31. (*Not cited.*)
  apply EqBi.
  apply n4_3. (*Not cited.*)
  apply EqBi.
  apply n4_3. (*Not cited.*)
  apply Equiv4_01.
  apply EqBi.
  apply n4_51a.
  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 n
4_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 EqBi.
 apply n4_21. (*Not cited.*)
 replace (P→Q↔R) with ((P→Q→R)∧(P→R→Q)).
 reflexivity.
 apply EqBi.
 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_27b. clear Simp3_27c. clear Si
mp3_27d. clear Simp3_26a. clear Simp3_26b. clear Comp3_43a. clear Co
mp3_43b. clear Sa. clear Sc. clear Sd. clear H. clear n4_77a.
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 EqBi.
  apply n4_77a.
  apply Equiv4_01.
  apply Equiv4_01.
  apply Equiv4_01.
  Qed.

End No5.
```