```
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.
```