

形式化方法 Lab3

PB20111686 黄瑞轩

1 Ex1

代码: ./code/lemma1.v

```
1 | Lemma ex1 : forall A : Prop, ~~~A -> ~A.
2 | Proof.
3 |   intros A H1 H2.
4 |   apply H1.
5 |   intros H3.
6 |   apply H3.
7 |   apply H2.
8 | Qed.
```

证明步骤:

lemma1

Lemma ex1 : forall A : Prop, ~~~A -> ~A.
Proof.
 intros A H1 H2.
 apply H1.
 intros H3.
 apply H3.
 apply H2.
Qed.

1 goal
----- (1/1)
forall A : Prop, ~ ~ ~ A -> ~ A

lemma1

Lemma ex1 : forall A : Prop, ~~~A -> ~A.
Proof.
 intros A H1 H2.
 apply H1.
 intros H3.
 apply H3.
 apply H2.
Qed.

1 goal
A : Prop
H1 : ~ ~ ~ A
H2 : A
----- (1/1)
False

lemma1

Lemma ex1 : forall A : Prop, ~~~A -> ~A.
Proof.
 intros A H1 H2.
 apply H1.
 intros H3.
 apply H3.
 apply H2.
Qed.

1 goal
A : Prop
H1 : ~ ~ ~ A
H2 : A
----- (1/1)
~ ~ A

lemma1	
<pre> Lemma ex1 : forall A : Prop, ~~~A -> ~A. Proof. intros A H1 H2. apply H1. intros H3. apply H3. apply H2. Qed. </pre>	<pre> 1 goal A : Prop H1 : ~ ~ ~ A H2 : A H3 : ~ A False (1/1) </pre>

lemma1	
<pre> Lemma ex1 : forall A : Prop, ~~~A -> ~A. Proof. intros A H1 H2. apply H1. intros H3. apply H3. apply H2. Qed. </pre>	<pre> 1 goal A : Prop H1 : ~ ~ ~ A H2 : A H3 : ~ A A (1/1) </pre>

lemma1	
<pre> Lemma ex1 : forall A : Prop, ~~~A -> ~A. Proof. intros A H1 H2. apply H1. intros H3. apply H3. apply H2. Qed. </pre>	<pre> No more goals. </pre>

2 Ex2

代码: ./code/lemma2.v

```

1 Lemma ex2 : forall A B, (A /\ B) -> ~(~A /\ ~B).
2 Proof.
3   intros A B A_or_B.
4   unfold not.
5   intros [not_A not_B].
6   destruct A_or_B as [A_true | B_true].
7   - apply not_A in A_true. contradiction.
8   - apply not_B in B_true. contradiction.
9   Qed.

```

证明步骤:

lemma2

Lemma ex2: forall A B, (A \ / B) -> ~(~A /\ ~B).

Proof.

```
intros A B A_or_B.
unfold not.
intros [not_A not_B].
destruct A_or_B as [A_true | B_true].
- apply not_A in A_true. contradiction.
- apply not_B in B_true. contradiction.
```

Qed.

1 goal

forall A B : Prop, A \ / B -> ~ [~ A /\ ~ B] (1/1)

lemma2

Lemma ex2: forall A B, (A \ / B) -> ~(~A /\ ~B).

Proof.

```
intros A B A_or_B.
unfold not.
intros [not_A not_B].
destruct A_or_B as [A_true | B_true].
- apply not_A in A_true. contradiction.
- apply not_B in B_true. contradiction.
```

Qed.

1 goal

A, B : Prop

A_or_B : A \ / B

~ [~ A /\ ~ B] (1/1)

lemma2

Lemma ex2: forall A B, (A \ / B) -> ~(~A /\ ~B).

Proof.

```
intros A B A_or_B.
unfold not.
intros [not_A not_B].
destruct A_or_B as [A_true | B_true].
- apply not_A in A_true. contradiction.
- apply not_B in B_true. contradiction.
```

Qed.

1 goal

A, B : Prop

A_or_B : A \ / B

(A -> False) /\ (B -> False) -> False (1/1)

lemma2

Lemma ex2: forall A B, (A \ / B) -> ~(~A /\ ~B).

Proof.

```
intros A B A_or_B.
unfold not.
intros [not_A not_B].
destruct A_or_B as [A_true | B_true].
- apply not_A in A_true. contradiction.
- apply not_B in B_true. contradiction.
```

Qed.

1 goal

A, B : Prop

A_or_B : A \ / B

not_A : A -> False

not_B : B -> False

False (1/1)

lemma2

Lemma ex2: forall A B, (A \ / B) -> ~(~A /\ ~B).

Proof.

```
intros A B A_or_B.
unfold not.
intros [not_A not_B].
destruct A_or_B as [A_true | B_true].
- apply not_A in A_true. contradiction.
- apply not_B in B_true. contradiction.
```

Qed.

2 goals

A, B : Prop

A_true : A

not_A : A -> False

not_B : B -> False

False (1/2)

False

False (2/2)

lemma2

Lemma ex2: forall A B, (A \vee B) \rightarrow \neg (\neg A \wedge \neg B).

Proof.

```
intros A B A_or_B.
unfold not.
intros [not_A not_B].
destruct A_or_B as [A_true | B_true].
- apply not_A in A_true. contradiction.
- apply not_B in B_true. contradiction.
```

Qed.

1 goal

A, B : Prop

A_true : A

not_A : A \rightarrow False

not_B : B \rightarrow False

False (1/1)

lemma2

Lemma ex2: forall A B, (A \vee B) \rightarrow \neg (\neg A \wedge \neg B).

Proof.

```
intros A B A_or_B.
unfold not.
intros [not_A not_B].
destruct A_or_B as [A_true | B_true].
- apply not_A in A_true. contradiction.
- apply not_B in B_true. contradiction.
```

Qed.

1 goal

A, B : Prop

A_true : False

not_A : A \rightarrow False

not_B : B \rightarrow False

False (1/1)

lemma2

Lemma ex2: forall A B, (A \vee B) \rightarrow \neg (\neg A \wedge \neg B).

Proof.

```
intros A B A_or_B.
unfold not.
intros [not_A not_B].
destruct A_or_B as [A_true | B_true].
- apply not_A in A_true. contradiction.
- apply not_B in B_true. contradiction.
```

Qed.

This subproof is complete, but there are some unfocused goals:

False (1/1)

lemma2

Lemma ex2: forall A B, (A \vee B) \rightarrow \neg (\neg A \wedge \neg B).

Proof.

```
intros A B A_or_B.
unfold not.
intros [not_A not_B].
destruct A_or_B as [A_true | B_true].
- apply not_A in A_true. contradiction.
- apply not_B in B_true. contradiction.
```

Qed.

1 goal

A, B : Prop

B_true : B

not_A : A \rightarrow False

not_B : B \rightarrow False

False (1/1)

lemma2

Lemma ex2: forall A B, (A \vee B) \rightarrow \neg (\neg A \wedge \neg B).

Proof.

```
intros A B A_or_B.
unfold not.
intros [not_A not_B].
destruct A_or_B as [A_true | B_true].
- apply not_A in A_true. contradiction.
- apply not_B in B_true. contradiction.
```

Qed.

1 goal

A, B : Prop

B_true : False

not_A : A \rightarrow False

not_B : B \rightarrow False

False (1/1)

lemma2

```

Lemma ex2: forall A B, (A /\ B) -> ~(~A /\ ~B).
Proof.
  intros A B A_or_B.
  unfold not.
  intros [not_A not_B].
  destruct A_or_B as [A_true | B_true].
  - apply not_A in A_true. contradiction.
  - apply not_B in B_true. contradiction.
Qed.

```

No more goals.

3 Ex3

代码: ./code/lemma3.v

```

1 Lemma ex3 : forall T (P : T -> Prop),
2   (~ exists x, P x) -> forall x, ~ P x.
3 Proof.
4   intros T P H x P_x.
5   apply H.
6   exists x.
7   exact P_x.
8 Qed.

```

证明步骤:

lemma3

```

Lemma ex3 : forall T (P : T -> Prop),
  (~ exists x, P x) -> forall x, ~ P x.
Proof.
  intros T P H x P_x.
  apply H.
  exists x.
  exact P_x.
Qed.

```

1 goal
forall (T : Type) (P : T -> Prop),
~ (exists x : T, P x) -> forall x : T, ~ P x (1/1)

lemma3

```

Lemma ex3 : forall T (P : T -> Prop),
  (~ exists x, P x) -> forall x, ~ P x.
Proof.
  intros T P H x P_x.
  apply H.
  exists x.
  exact P_x.
Qed.

```

1 goal
T : Type
P : T -> Prop
H : ~ (exists x : T, P x)
x : T
P_x : P x
False (1/1)

lemma3

```

Lemma ex3 : forall T (P : T -> Prop),
  (~ exists x, P x) -> forall x, ~ P x.
Proof.
  intros T P H x P_x.
  apply H.
  exists x.
  exact P_x.
Qed.

```

```

1 goal
T : Type
P : T -> Prop
H : ~ (exists x : T, P x)
x : T
P_x : P x
_____ (1/1)
exists x0 : T, P x0

```

lemma3

```

Lemma ex3 : forall T (P : T -> Prop),
  (~ exists x, P x) -> forall x, ~ P x.
Proof.
  intros T P H x P_x.
  apply H.
  exists x.
  exact P_x.
Qed.

```

```

1 goal
T : Type
P : T -> Prop
H : ~ (exists x : T, P x)
x : T
P_x : P x
_____ (1/1)
P x

```

lemma3

```

Lemma ex3 : forall T (P : T -> Prop),
  (~ exists x, P x) -> forall x, ~ P x.
Proof.
  intros T P H x P_x.
  apply H.
  exists x.
  exact P_x.
Qed.

```

No more goals.