形式化方法 Lab3

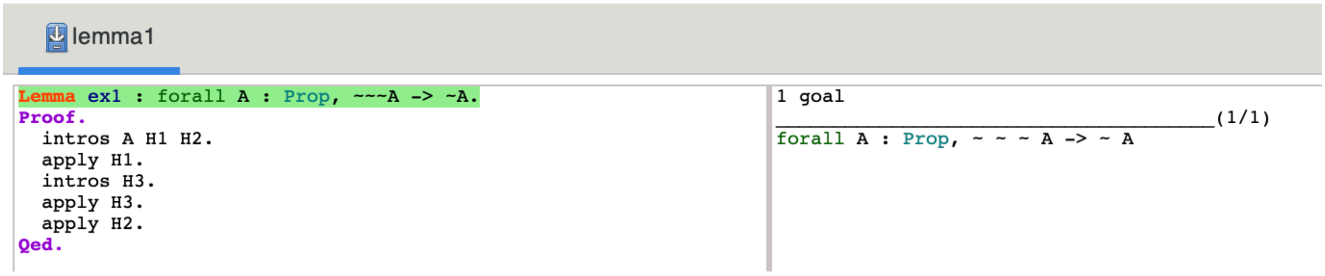
PB20111686 黄瑞轩

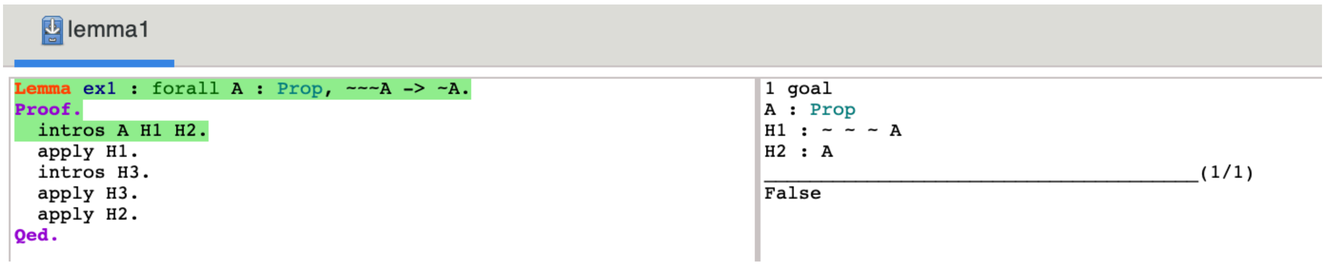
1 Ex1

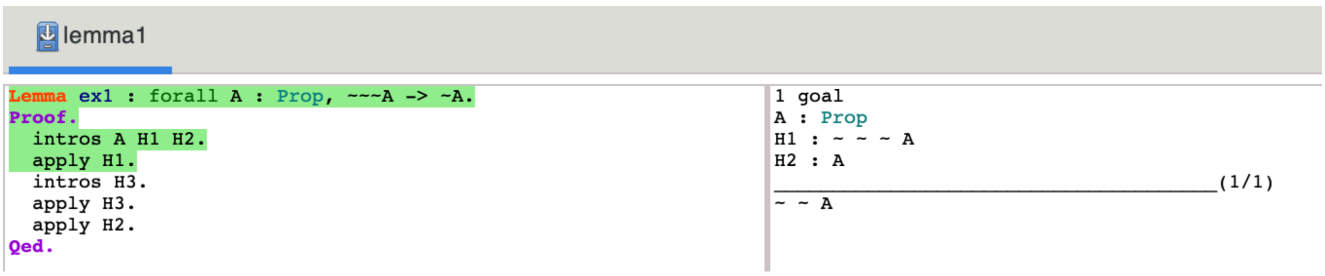
代码: ./code/lemma1.v

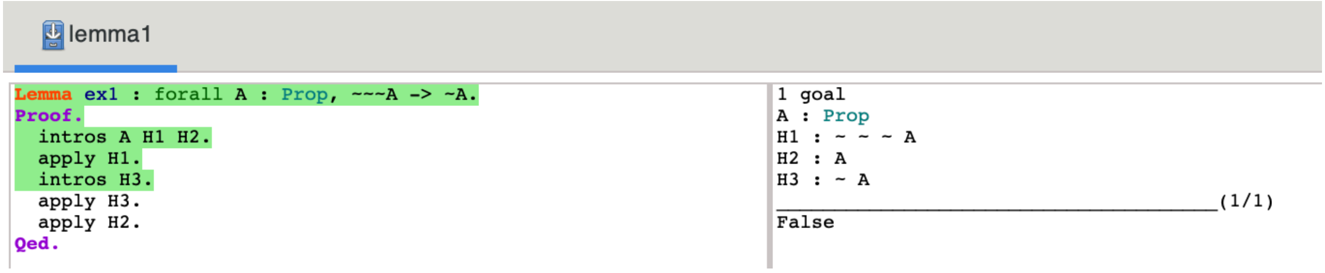
```
Lemma ex1 : forall A : Prop, ~~~A -> ~A.
Proof.
  intros A H1 H2.
  apply H1.
  intros H3.
  apply H3.
  apply H2.
Qed.
```

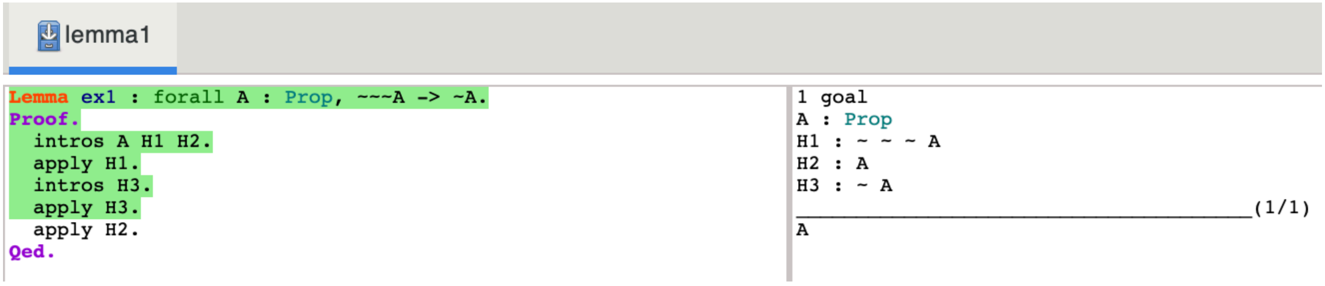
证明步骤:

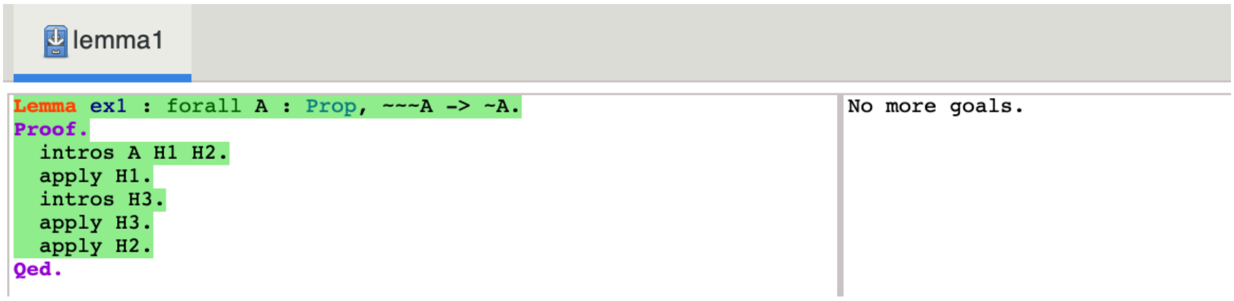












2 Ex2

代码：./code/lemma2.v

```
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.
```

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
______________________________________(1/1)
forall A B : Prop, A \/ B -> ~ (~ A /\ ~ B)
```

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
______________________________________(1/1)
~ (~ A /\ ~ B)
```

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

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
______________________________________(1/1)
False
```

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
______________________________________(1/2)
False
______________________________________(2/2)
False
```

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_true : A
not_A : A -> False
not_B : B -> False
______________________________________(1/1)
False
```

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_true : False
not_A : A -> False
not_B : B -> False
______________________________________(1/1)
False
```

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.
```

```
This subproof is complete, but there are some unfocused goals:
______________________________________(1/1)
False
```

lemma2

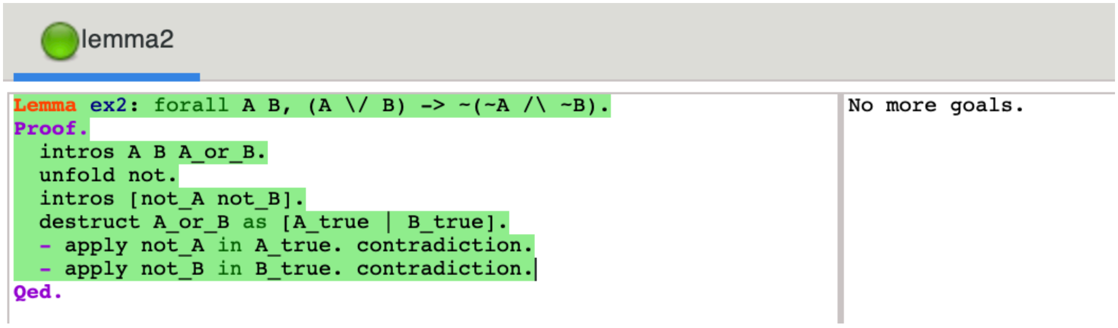
```
Lemma ex2: forall A B, (A \/ B) -> ~(~A /\ ~B).
Proof.
  intros A B A_or_B.
  unfold not.
  intros [not_A not_B].
  destruct A_or_B as [A_true | B_true].
  - apply not_A in A_true. contradiction.
  - apply not_B in B_true. contradiction.
Qed.
```

```
1 goal
A, B : Prop
B_true : B
not_A : A -> False
not_B : B -> False
______________________________________(1/1)
False
```

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
B_true : False
not_A : A -> False
not_B : B -> False
______________________________________(1/1)
False
```

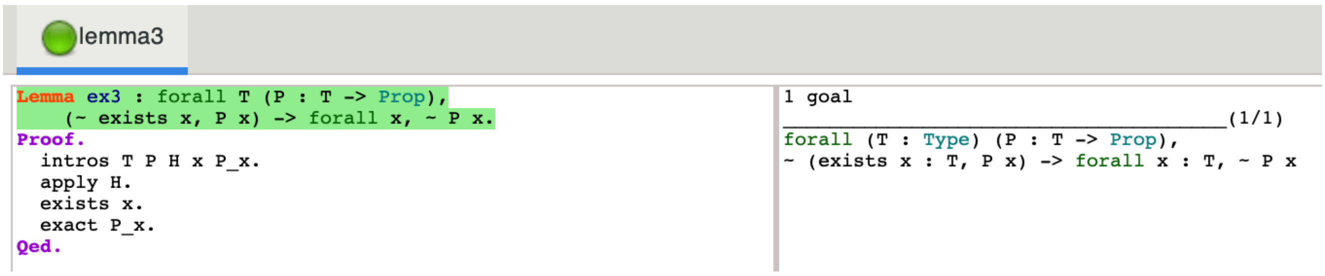


3 Ex3

代码：./code/lemma3.v

```
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.
```

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
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.
```