Coq 中基于 rewrite 的证明

在先前证明单调函数性质的过程中，我们经常会要通过参数之间的大小关系推导出函数值之间的大小关系。例如，当 f 是一个单调函数时，可以由 $x \leqslant y$ 推出 $f(x) \leqslant f(y)$。对于一般的函数，我们也可以由参数相等，推出函数值相等，即由 $x = y$ 推出 $f(x) = f(y)$。下面的证明中就要用到这一性质。

```
Definition is_fixpoint (f: Z -> Z) (x: Z): Prop :=
  f x = x.
```

```
Theorem fixpoint_self_comp: forall f x,
  is_fixpoint f x ->
  is_fixpoint (Zcomp f f) x.
Proof.
  unfold is_fixpoint, Zcomp.
  intros.
  (** 此时需要利用前提_[H: f x = x]_证明_[f (f x) = x]_。*)
  rewrite H.
  rewrite H.
  reflexivity.
Qed.
```

在数学上，如果 `f x = x`，那么我们就称 `x` 是函数 `f` 的一个不动点。上面的定理证明了，如果 `x` 是 `f` 的不动点，那么 `x` 也是 `f` 与自身复合结果的不动点。在这一证明中，前提 `H` 是命题 `f x = x`。证明指令 `rewrite H` 的效果是将结论中的 `f x` 替换为 `x`，因此，第一次使用该指令后，原先需要证明的 `f (f x) = x` 规约为了 `f x = x`。这一步背后的证明实际就用到了“只要函数 `f` 的参数不变，那么函数值也不变”这条性质。

Coq 证明脚本 1. 利用等式做证明的 rewrite 指令. 有时，这个 `a` 可能出现了多次，但是我们只希望将其中的若干个 `a` 而不是全部的 `a` 都替换成 `b`，此时可以在 rewrite 指令中增加 `at`，即使用 `rewrite H at ...` 或 `rewrite H at ... in ...` 指明需要进行替换的位置。例如，当前提 `H` 为 `x = f x`，待证明结论为 `x = f (f x)` 时，

`rewrite H` 与 `rewrite H at 1, 2`

都会将结论变为 `f x = f (f (f x))`；

`rewrite H at 1`

会将结论变为 `f x = f (f x)`；而

`rewrite H at 2`

会将结论变为 `x = f (f (f x))`。

Coq 允许 rewrite 使用的定理或前提中有 `forall` 概称量词，用户可以手动地填入这些 `forall` 约束的变量或由 Coq 自动填入这些变量。例如，当前提 `H1` 与待证明结论分别为：

```
H1: forall x y: Z, g x y = g y x
结论: g 3 5 = 6
```

时，`rewrite H1`、`rewrite (H1 3)` 或 `rewrite (H1 3 5)` 都会将待证明结论变换为 `g 5 3 = 6`。Coq 还允许 rewrite 使用的定理或前提中带有附加条件，例如当前提 `H2` 与待证明结论分别为：

```
H2: forall x: Z, x <= 0 -> h x = 0
结论: h (h (-5)) = 0
```

时，用户可以直接指明如何证明这一附加条件，使用 `rewrite (H2 (-5) ltac:(lia))` 完成重写替换；用户也可以不直接指明，例如 `rewrite H2` 或者 `rewrite (H2 (-5))` 会将原证明目标规约为两个新的证明目标：第一个证明目标中结论变为 `h 0 = 0`，第二个证明目标为需要补充证明的附加条件 `-5 <= 0`。如果使用的定理或前提带有超过一个附加条件，那么 `rewrite` 指令就会生成超过两个证明目标。如果产生的所有附加条件都可以用一条证明脚本完成证明，那么可以在 rewrite 指令中加入 `by`。上面例子中，`rewrite H2 by lia` 可以直接将结论变为 `h 0 = 0` 并不再额外产生附加的证明条件。

最后，Coq 中不仅允许将等式左侧的内容替换为等式右侧的内容，也允许使用 `<-` 进行反向操作。例如，当 `H` 具有形式 `a = b` 时，`rewrite <- H` 会将结论中的 `b` 替换为 `a`。同时，Coq 也允许在一条 rewrite 指令中，按指定顺序连续进行多次替换，例如 `rewrite H1, H2` 表示先 `rewrite H1` 再 `rewrite H2`。

下面关于不动点简单性质的证明需要我们灵活使用 rewrite 指令。

```
Example fixpoint_self_comp23: forall f x,
  is_fixpoint (Zcomp f f) x ->
  is_fixpoint (Zcomp f (Zcomp f f)) x ->
  is_fixpoint f x.
Proof.
  unfold is_fixpoint, Zcomp.
  intros.
  rewrite H in H0.
  rewrite H0.
  reflexivity.
Qed.
```

习题 1. 请在 Coq 中证明下面关于结合律的性质。

```
Definition power2 (f: Z -> Z -> Z) (x: Z): Z :=
  f x x.
```

```
Definition power4 (f: Z -> Z -> Z) (x: Z): Z :=
  f x (f x (f x x)).
```

```
Fact power2_power2: forall f a,
  assoc f ->
  power2 f (power2 f a) = power4 f a.
(* 请在此处填入你的证明，以_[Qed]_结束。 *)
```

习题 2. 请在 Coq 中证明下面关于群的性质。提示：`f x (g x) = f 1 (f x (g x))` 而 `f (g (g x)) (g x) = 1`。

```
Fact group_basic: forall (f: Z -> Z -> Z) (g: Z -> Z),
  assoc f ->
  (forall x, f 1 x = x) ->
  (forall x, f (g x) x = 1) ->
  (forall x, f x (g x) = 1).
(* 请在此处填入你的证明，以_[Qed]_结束。 *)
```