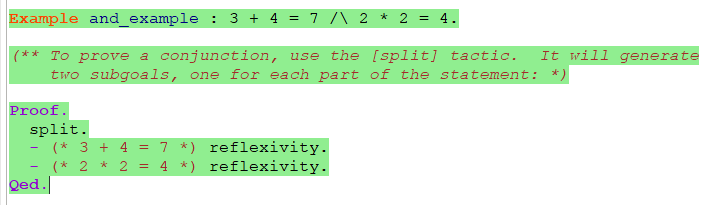
整理2

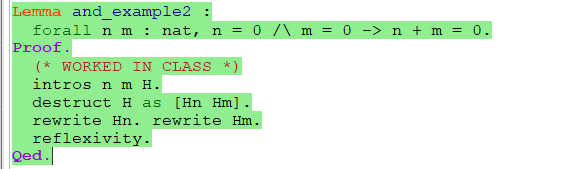
1. Logic.v

Conjunction:



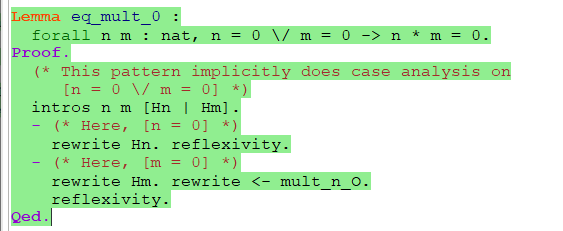
证明时，用split分开，分别证明两部分

当前提中有Conjunction 时，用destruct H as [HA HB]，HA表示A为真时，HB表示B为真时

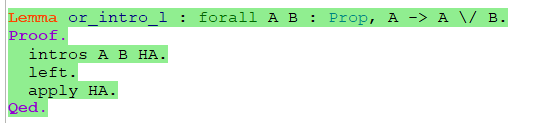


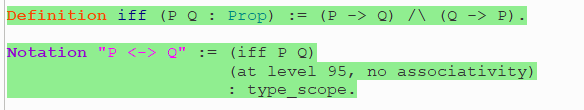
Disjunction：

当前提中有disjunction时，用destruct H as [HA | HB]，分别在HA和HB的前提下证明结论



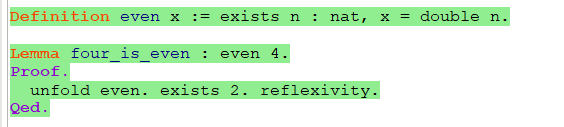
当disjunction在结论中时，用left或right选择一个证明



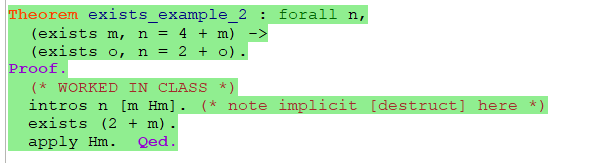
逻辑等价：  


两个命题的合取，用split分开

存在量词：

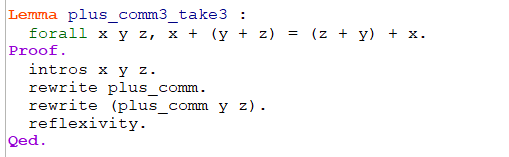


如果说一个 T 类型的变量 x 满足属性 P，则记为exists x:T, P。  
为了证明形如exists x:T,P.类型的命题，必须要证明对于某个特定的x，P属性成立。这称为存在的见证：



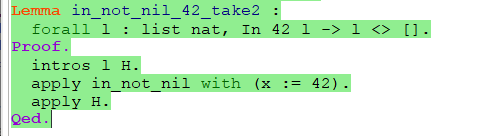
如果假设中有exists x，P ，可以用destruct来获得一个x和在x下成立的P

将定理作为参数：（将定理当作函数使用）



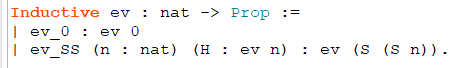
一个更优雅的替代方法是将plus\_comm直接应用于要实例化它的参数，这与将多态函数应用于类型参数的方式几乎相同。

apply with：与前面的用法差不多



1. IndProp

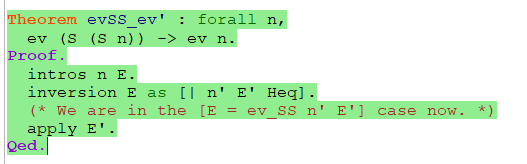
偶数的递归定义：



规则ev\_0表示 0是偶数

规则ev\_SS n 表示若n是偶数，则S(S n) 是偶数

inversion:



反演策略可以检测到（1）第一种情况（n = 0）不适用，以及（2）出现在ev\_SS情况下的n'必须与n相同。 它具有类似于destruct的“ as”变体，允许我们分配名称而不是由Coq选择它们。

与destruct不同：舍弃明显不匹配他的基情况

remember:

remember e as x

1. 将所有变量e替换为x
2. 在前提中加入x=e

Imp.v

- [clear H]: Delete hypothesis [H] from the context.

- \*[subst x]: For a variable [x], find an assumption [x = e] or

[e = x] in the context, replace [x] with [e] throughout the

context and current goal, and clear the assumption.

- [subst]: Substitute away \_all\_ assumptions of the form [x = e]

or [e = x] (where [x] is a variable).

- [rename... into...]: Change the name of a hypothesis in the

proof context. For example, if the context includes a variable

named [x], then [rename x into y] will change all occurrences

of [x] to [y].

- \*[assumption]: Try to find a hypothesis [H] in the context that

exactly matches the goal; if one is found, solve the goal.

- [contradiction]: Try to find a hypothesis [H] in the current

context that is logically equivalent to [False]. If one is

found, solve the goal.

- \*[constructor]: Try to find a constructor [c] (from some

[Inductive] definition in the current environment) that can be

applied to solve the current goal. If one is found, behave

like [apply c].