Aussagenlogik

Fitch-Regel	Coq-Taktik
$\wedge$ I	`split.`
$\wedge$ E_1	`destruct H as [H0 _]; exact H0.`
$\wedge$ E_2	`destruct H as [_ H0]; exact H0.`
$\vee$ I_1	`left.`
$\vee$ I_2	`right.`
$\vee$ E	`destruct H as [L|R]. apply H0; exact L. apply H1; exact R.`
$\rightarrow$ I	`intro.`
$\rightarrow$ E	`apply H; exact H0.`
$\neg$ I	`intro.`
$\neg$ E	`apply NNPP.`
$\bot$ I	`apply H.`
$\bot$ E	`contradiction.`

Prädikatenlogik

Fitch-Regel	Coq-Taktik
$\forall I$	`intro x0.`
$\forall E$	`apply H.`
$\exists I$	`exists t.`
$\exists E$	`destruct H.`
$= I$	`reflexivity.`
$= E$	`rewrite H.`

Es is gelegentlich hilfreich, während eines Beweises von Hand benannte Unterziele einzuführen, die man, wenn sie bewiesen sind, als zusätzliche Annahmen verwenden darf. Dies geschieht mittels `assert (<Formel>) as <Name>`.