Pense-bête

Symbole	Introduction	Elimination
\forall	intro ou intros	apply $\langle \texttt{Hyp} \rangle$
\Rightarrow	intro ou intros	$\mathtt{apply}\; \langle \mathtt{Hyp} \rangle$
_	intro	$\mathtt{destruct}\ \langle \mathtt{Hyp} \rangle$
٨	split	destruct (Hyp) as (H1,H2)
V	left ou right	destruct (Hyp) as [H1 H2]
3	exists (term)	destruct (Hyp) as (x,H)
Т	trivial	
		exfalso ou destruct $\langle { t Hyp} \rangle$
$t_1 = t_2$	reflexivity	rewrite <- $\langle \text{Hyp1} \rangle$ in $\langle \text{Hyp2} \rangle$

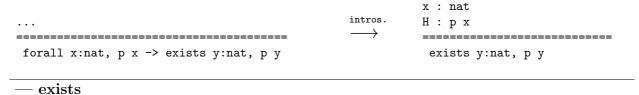
Les parties en gris sont optionnelles et permettent de nommer les hypothèses que l'on va introduire dans l'environnement. Dans le cas de la réécriture d'une égalité, la flèche vers la gauche permet d'inverser le sens de réécriture et le in permet de réécrire l'égalité dans une hypothèse.

Concept	Tactique	Utilisation
hypothèse	assumption	permet de conclure quand la conclusion du but courant est également une hypothèse de ce but.
étape	assert (Form) as H	permet d'introduire un résultat intermédiaire $\langle Form \rangle$ qu'on devra prouver avant de l'avoir comme nouvelle hypothèse.
définition	unfold (def) in (Hyp)	déplie la définition $\langle def \rangle$.
calcul	$oxed{ extstyle simpl in $\langle ext{Hyp} angle}$	permet d'effectuer un calcul (addition, concaténation)
inductif	apply <règle></règle>	introduction correspondant à une règle de construction
	$ig $ induction $\langle ext{Hyp} angle$	application du principe d'induction
	inversion $\langle \mathtt{Hyp} \rangle$	raisonnement par cas
arithmétique	omega	résolution d'(in)équations entre entiers.

Détails des tactiques

— intros

La tactique intros applique la règle d'introduction correspondant au connecteur ou quantificateur de la conclusion du but courant. Et elle recommencera, autant de fois que possible, sur la conclusion du but obtenu. Par exemple, si la conclusion du but courant est : $\forall x, p(x) \Rightarrow \exists y, p(y)$, a tactique intros va introduire la variable x ainsi que l'hypothèse p x.



Pour prouver une formule quantifiée existentiellement comme exists y:nat, p y, l'utilisateur doit fournir à Coq à la fois le *témoin* et la preuve que ce témoin vérifie le prédicat p (ce qui correspond à la règle d'introduction de \exists). La tactique exists permet de faire cela. Dans le but obtenu précédemment, on peut instancier y par x dans

la formule que l'on cherche à prouver à l'aide de la commande exists x. Il restera alors à montrer p x.

— assumption

La tactique assumption correspond à la règle axiome de la déduction naturelle. On peut donc l'utiliser pour finir la preuve quand la conclusion du but courant se trouve dans les hypothèses.

— apply

La tactique apply permet d'utiliser une formule que l'on a en hypothèse. Par exemple, si la conclusion du but courant est une formule $\mathbb Q$ et que l'on a en hypothèse une formule $\mathbb P \to \mathbb Q$ (nommée $\mathbb H$), alors on peut appliquer cette hypothèse grâce à la commande apply $\mathbb H$. Il restera alors à prouver $\mathbb P$.

D'une manière plus générale, la tactique apply peut s'utiliser sur toute hypothèse H de la forme $\forall x_1 \dots \forall x_k, P_1 \Rightarrow \dots P_n \Rightarrow Q'$. Il est parfois nécessaire de préciser avec quelles valeurs il faut instancier les différentes variables x_i .

Ceci peut se faire à l'aide de la commande apply H with $(x_1 := y_1)$ (...) $(x_n := y_n)$.

— destruct

La tactique destruct permet d'éliminer des conjonctions, disjonctions, négations, contradictions et existentiels en hypothèse. Si c'est une conjonction H:P1 /\ P2 on se retrouve avec deux hypothèses au lieu d'une seule : une pour P1 et l'autre pour P2.

H1 : P1

Si c'est une disjonction P1 \/ P2, on obtient deux sous-buts avec uniquement P1 ou P2 en hypothèse.