Preuves avec l'atelier B

3 janvier 2006

1 Interactif vs. automatique

À diverses occasions, l'Atelier B génère des obligations de preuves (PO). On peut afficher les preuves générées pour un module en cliquant sur le module, puis sélectionnant `Analysing .../ Show/Print PO`. Dans la fenêtre qui se présente, on active une des lignes et clique sur `Display`.

Pour prouver ("décharger") une obligation de preuve, on sélectionne l'une des options qui se trouvent sous le menu `Prove ...`.

La logique sur laquelle B est basée est indécidable. On ne peut donc pas s'attendre à trouver une méthode qui résout toutes les PO de manière automatique. L'Atelier B offre :

- plusieurs méthodes de preuve automatiques, qui appliquent des stratégies de recherche de preuve de plus en plus complexes (de `Automatic (force 0)...` à `Automatic (force 3)...`). La première est assez rapide, mais risque d'échouer souvent. La dernière peut être très lente sur certaines formules.
- un mode de preuve interactif, décrit plus en détail en Section 2. On le lance avec `Interactive`.

Le prouveur de B est décrit plus en détail dans le Manuel d'utilisateur[1], un résumé des commandes se trouve dans le Manuel de référence[2].

2 Le mode interactif

2.1 Fonctionnement

Après avoir lancé le mode interactif, une première fenêtre s'ouvre qui contient la liste des PO. En double-cliquant sur une PO, on commence la preuve interactive de cette PO. Ceci ouvre deux autres fenêtres :

- HYPOTHESIS : La liste des hypothèses du but actuel
- GOAL : Le but actuel à prouver. Dans la partie inférieure de la fenêtre, taper les commandes (Section 2.2). Tant que le but n'est pas encore prouvé, le but s'affiche en bleu. Une fois le but prouvé, il est affiché en vert.

[1] http://www.irit.fr/~Martin.Strecker/Teaching/Common/B_prover_user_manual.pdf

[2] http://www.irit.fr/~Martin.Strecker/Teaching/Common/B_prover_reference_manual.pdf

2.2 Commandes

- `dd` *(deduction)* : Le but actuel a la forme `P => Q`. La commande ajoute `P` aux hypothèses, il reste le but `Q`. Cette commande correspond à la règle de la déduction naturelle

$$\frac{\Gamma, P \vdash Q}{\Gamma \vdash P \Rightarrow Q}\ \mathrm{I}\Rightarrow$$

- `dc(F)` *(do cases)* : Si le but actuel est `G`, cette commande fait une distinction de cas pour la formule `F`, c.à.d introduit deux sous-buts `F => G` et `¬ F => G`. Cette commande correspond à la règle de la déduction naturelle

$$\frac{\Gamma \vdash F \Rightarrow G \quad \Gamma \vdash \neg F \Rightarrow G}{\Gamma \vdash G}\ (\mathrm{cut})$$

- `se(t)` *(suggest for exist)* : Le but actuel a la forme `# x. P(x)`. La variable `x` est remplacée par le terme `t`, on obtient le nouveau but `P(t)`. Cette commande correspond à la règle de la déduction naturelle

$$\frac{\Gamma \vdash P(t)}{\Gamma \vdash \exists x.P(x)}\ \mathrm{I}\exists$$

- `ph(t, !x. P(x))` *(particularize hypothesis)* : La formule `!x. P(x)` se trouve parmi les hypothèses actuelles. La variable `x` est remplacée par le terme `t`, la formule `P(t)` est ajoutée comme nouvelle hypothèse. Cette commande correspond à la règle du calcul des séquents :

$$\frac{\Gamma, \forall x.P(x), P(t) \vdash G}{\Gamma, \forall x.P(x) \vdash G}\ \forall\mathrm{L}$$

- `eh(l, r, f)` *(equality in hypothesis)* : Pourvu qu'il y a une hypothèse de la forme `l = r` ou `r = l` :
  - Si `f` est le mot clé `Goal` : remplace `l` par `r` dans le but actuel.
  - Si `f` est le mot clé `AllHyp` : remplace `l` par `r` dans toutes les hypothèses.

  Pour plus de variantes, voir le manuel de référence.
- `pr` *(prover)* : Appelle une procédure de preuve automatique, qui fait des simplifications propositionnelles et des simplifications arithmétiques. Voir le manuel de référence pour des paramètres admis par `pr`.
- `ap` *(arithmetic prover)* : Une procédure de preuve automatique pour des propositions arithmétiques. Parfois plus puissant que `pr`.
- `ba(n)` *(back)* : Fait `n` pas en arrière dans la preuve, où `n` est un nombre ≥ 1