

Devoir maison

Preuve automatique en déduction naturelle propositionnelle

*Le devoir devra être rendu **le mardi 20 Novembre 2007**. Les sources Ocaml du devoir devront être envoyés par mail à David Pichardie pour cette date. Le devoir peut être réalisé par petits groupes, mais chaque étudiant doit rendre une copie personnelle en indiquant clairement ses collaborateurs (et/ou collaboratrices). Les sources Ocaml seront indentés et commentés **avec soin**. L'évaluation du devoir tiendra compte de la précision des arguments et de la présentation générale.*

Le but de ce devoir est de réaliser un outil de preuve automatique en déduction naturelle pour la logique propositionnelle. Nous nous appuyons pour cela sur la preuve de complétude (partielle) vue en cours.

Notations : Nous notons \mathcal{F} l'ensemble des formules de la logique propositionnelle comportant les connecteurs \neg , \wedge , \vee et \Rightarrow . Nous notons \mathcal{F}' l'ensemble des formules de la logique propositionnelle ne comportant que les connecteurs \neg et \Rightarrow . Nous notons simplement \vdash la relation \vdash_{NK} .

La preuve de complétude vue en cours démontre que pour toute formule F de \mathcal{F}' et tout ensemble fini Γ de formule de \mathcal{F}' , $\Gamma \models F$ implique $\Gamma \vdash F$.

Question 1. Proposez une transformation $\Phi \in \mathcal{F} \rightarrow \mathcal{F}'$ telle que pour tout formule $F \in \mathcal{F}$, $\{\Phi(F)\} \vdash F$ (que l'on peut noter $\Phi(F) \vdash F$).

Question 2. Complétez la preuve de complétude vue en cours pour démontrer la complétude de \vdash_{NK} sur \mathcal{F} .

Question 3. En vous inspirant de la preuve précédente, programmez une fonction Ocaml de type `formule -> arbreDePreuve` qui, pour toute tautologie F , renvoie un arbre de preuve valide de $\vdash F$. Justifiez la validité de votre algorithme et proposez des exemples significatifs d'utilisation de cette fonction.

Question 4. Les preuves construites par la fonction précédente ne sont pas toujours celles qu'un humain aurait proposées au premier abord. Discutez ce problème sur un exemple et proposez des techniques de transformation d'arbre de preuve permettant de rendre les preuves générées plus naturelles. Aucun programme Ocaml n'est requis ici.