TD 11 : Théorème de Herbrand

lionel.rieg@ens-lyon.fr

Exercice 1 ***Axiome atomique***

Dans le système LK réversible vu la semaine dernière, les axiomes $\overline{\Gamma, A \vdash A, \Delta}$ portent sur des formules A arbitraires. Montrer qu'on peut se retreindre au cas où A doit être une formule atomique.

Exercice 2 ***Forme normale d'une preuve***

Dans la recherche automatique de preuves, on a intérêt à réduire au maximum l'espace de recherche donc à rechercher des preuves avec le plus de structure possible. Ainsi, l'élimination (ou la remontée) des coupures et la réversibilité participent à cet effort. Dans cet exercice, on va voir qu'on peut combiner ces résultats pour obtenir une forme très contrainte sur les preuves tout en restant complet. Le système de preuves que l'on va considérer est le système réversible vu la semaine dernière. On ne s'intéresse qu'aux formules en forme prénexe et on recherche des preuves sous la forme suivante :

- une partie réversible sans affaiblissement portant sur des formules sans quantificateur et avec axiome atomique,
- puis une partie avec des affaiblissements sur des formules quantifiées,
- enfin, une partie avec des introductions de quantificateurs et des contractions.

Noter qu'on travaille toujours à échange près.

1. Quel est l'intérêt de ne considérer que des formules en forme prénexe ?
2. On considère une preuve d'un séquent ne contenant que des formules en forme prénexe. Montrer que l'on peut faire remonter les affaiblissements au dessus des règles d'introduction des quantificateurs.
3. Pourquoi peut-on éviter les affaiblissements dans la partie réversible ?

Théorème de Herbrand

L'énoncé auquel on va s'intéresser ici n'est pas celui que vous avez vu en cours :

> Une formule $\exists \vec{x}.\, F(\vec{x})$ (avec F sans quantificateur) est vraie
> si et seulement si elle possède une disjonction de Herbrand.

On va plutôt s'intéresser à l'équivalence entre des variantes des deux formules. Plus précisément, on part de la négation de $\exists \vec{x}.\, F(\vec{x})$: une formule universelle U (c'est à dire de la forme $U = \forall \vec{x}.\, F'(\vec{x})$ où $F' = \neg F$ est sans quantificateur). On va supposer qu'elle n'est pas satisfiable :

$$\text{pour toute interprétation } \mathcal{I},\ \mathcal{I} \text{ ne valide pas } U \quad \textit{i.e.} \quad \forall \mathcal{I}.\, \mathcal{I} \not\models U$$

Pour le membre droit, on considère « la négation d'une disjonction de Herbrand », c'est-à-dire une conjonction d'instances closes de $F' = \neg F$ qu'on suppose non valide (c'est-à-dire valide dans aucune interprétation). On montrera alors qu'elle est équivalente à $\forall \mathcal{I}.\, \mathcal{I} \not\models U$.

$$\begin{aligned}
\exists \vec{x}.\, F(\vec{x}) \text{ est valide} &\overset{1}{\iff} \text{pour toute interprétation } \mathcal{I},\ \mathcal{I} \not\models U \\
&\overset{2}{\iff} \text{il existe } t_1, \ldots, t_n \text{ tels que } F'(t_1) \wedge \ldots \wedge F'(t_n) \text{ n'est valide dans aucune interprétation} \\
&\overset{3}{\iff} \text{il existe } t_1, \ldots, t_n \text{ tels que } F(t_1) \vee \ldots \vee F(t_n) \text{ est valide}
\end{aligned}$$

Exercice 3 *Sens faciles*

1. Montrer l'équivalence 1.
2. Montrer l'équivalence 3.
3. Montrer le sens réciproque de l'équivalence 2.

Ne reste à montrer que le sens direct de l'équivalence 2, à savoir

Si pour toute interprétation $\mathcal{I}$, $\mathcal{I} \not\models U$, alors il existe $t_1, \ldots, t_n$ tels que $\not\models F'(t_1) \wedge \ldots \wedge F'(t_n)$.

La difficulté est de s'assurer que la conjonction contient suffisamment d'information pour couvrir toutes les interprétations $\mathcal{I}$ possibles. Ces interprétations sont caractérisées par leurs valeurs sur les formules atomiques (le reste étant défini par les tables de vérité des connecteurs). Ainsi, pour s'assurer de couvrir toutes les interprétations, il suffit de couvrir toutes les fonctions de l'ensemble des instances closes des formules atomiques (la *base de Herbrand* $\mathcal{B}$) dans les booléens.

Ainsi, chaque instance close f_i de F' dans la conjonction correspondra à une famille d'interprétations qui ont en commun la partie finie qui permet d'invalider f_i. Couvrir toutes les interprétations revient alors à s'assurer que toute interprétation possède une partie finie qui convient à l'un des f_i. On va voir que cela revient à montrer l'existence d'un *arbre de Herbrand*.

Définition *Arbre de Herbrand*

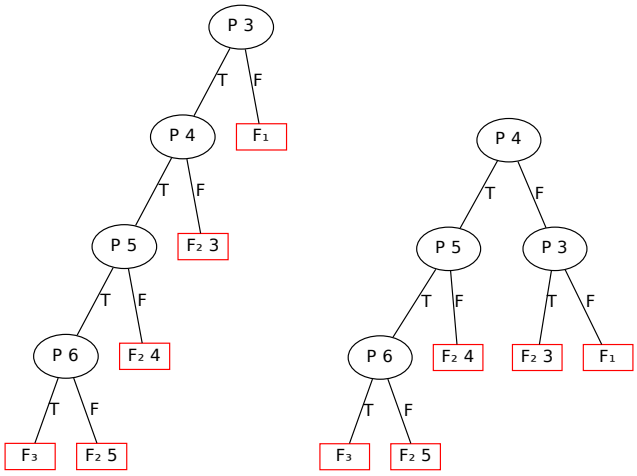
Un *arbre de Herbrand* est un arbre binaire dont :
- les nœuds internes sont étiquetés par des atomes,
- les branches correspondent à des interprétations partielles finies,
- les feuilles contiennent des contradictions (des f_i).

Par exemple, pour $F = \neg P\,3 \vee (P\,x \wedge \neg P\,(x+1) \vee P\,6$, une disjonction de Herbrand est

$$\neg P\,3 \vee (P\,3 \wedge \neg P\,4) \vee (P\,4 \wedge \neg P\,5) \vee (P\,5 \wedge \neg P\,6) \vee P\,6\,.$$

On a alors $F' = F_1 \wedge F_2\,x \wedge F_3$ où
- $F_1 = P\,3$
- $F_2\,x = P\,x \to P\,(x+1)$
- $F_3 = \neg P\,6$

et voici ci-contre deux arbres de Herbrand :

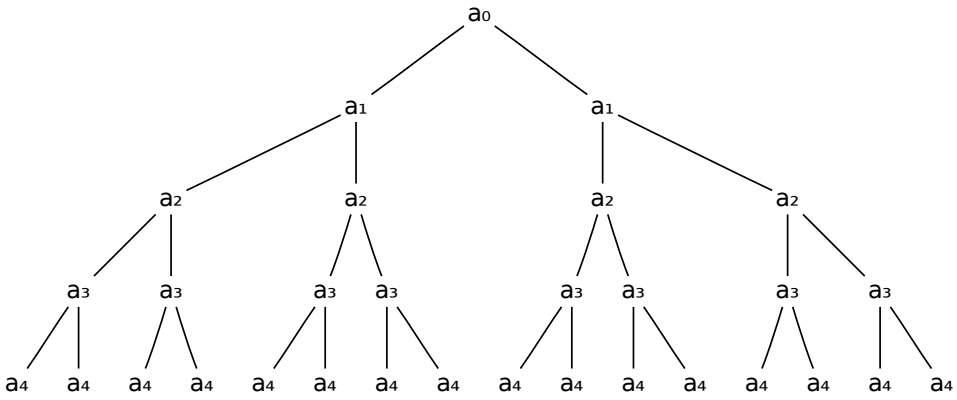


Exercice 4 ***Démonstration du théorème de Herbrand***

On veut démontrer la version du théorème de Herbrand présentée ci-avant.

1. Pourquoi la base de Herbrand $\mathcal{B}$ est-elle dénombrable ?

On énumère alors les éléments de $\mathcal{B}$ (appelés *atomes*) par $(a_i)_{i\in\mathbb{N}}$ et on les représente dans un arbre binaire infini :



2. À quoi correspond une branche infinie dans cet arbre ?
3. En utilisant le théorème de compacité, montrer que l'on peut couper une branche infinie de cet arbre à profondeur finie et avoir une contradiction sur la feuille.
4. Utiliser le lemme de Kőnig faible pour conclure.

Exercice 5 ***Une preuve sans le lemme de Kőnig***

La preuve précédente comporte deux défauts. Son défaut capital est de nécessiter un ordre *a priori* sur les atomes qui peut être très mal adapté à la formule U. Le second est d'utiliser le lemme de Kőnig qui masque le contenu calculatoire de la preuve. On va donc dans cet exercice faire une autre preuve, par contraposition, qui évite ces deux écueils.

1. Comment vérifier calculatoirement qu'un arbre est bien un arbre de Herbrand (pour une formule U fixée) ?
2. Écrire l'énoncé (la contraposée du théorème) que l'on veut démontrer.
3. Montrer que si p ne possède pas de sous-arbre de Herbrand, alors $p \cup \{a \mapsto true\}$ ou $p \cup \{a \mapsto false\}$ ne possèdent pas de sous arbre de Herbrand.
4. En itérant ce procédé, construire une interprétation.
5. Montrer que cette interprétation est un modèle de U.

Devoir Maison Coq pour le 20 décembre 2012

Vous devez formaliser en Coq la seconde preuve du théorème de Herbrand vue dans ce TD. Vous avez droit à la logique classique (module `Classical`) mais aucun autre axiome. Ainsi, un `Print Assumptions Herbrand` à la fin du fichier ne doit renvoyer que `classic : forall P : Prop, P \/ ~P`.

Voici les types qui vont apparaître dans la formalisation :

`atom`	les atomes (les instances close des formules atomiques)
`formula`	les formules sans quantificateur
`term`	les termes du premier ordre (que l'on passe à $U = \forall \vec{x}.\, F\ \vec{x}$)
`path`	les chemins dans un arbre binaire (= les interprétations finies)
`tree`	les arbres de Herbrand
`atom -> Prop`	les interprétations

Pour simplifier, on va supposer `atom` = `nat` et `term` = `nat`. On veillera par contre dans la formalisation à utiliser `atom` et `term` plutôt que `nat` pour maintenir la distinction. Par ailleurs, la théorie $U = \forall \vec{x}. F'(\vec{x})$ sera représentée par une fonction de `term` dans `formula`, c'est-à-dire on suppose qu'elle ne prend qu'un argument.

Vous trouverez un modèle sur la page des TD qui donne une méthode possible de démonstration. Vous êtes libre de ne pas le suivre mais le théorème que vous prouvez finalement doit être le même.

Quelques conseils pour finir :

- sur la page des TD, à côté du modèle `Herbrand.v` se trouve un fichier `OptBool.vo` qui définit des opérations sur le type `option bool` et prouve certains résultats qui peuvent vous être utiles ;
- on peut combiner plusieurs `destruct` en un seul en donnant un motif d'introduction : par exemple sur `H : A ∧ (B ∨ C)`, on peut faire `destruct H as [HA [HB | HC]]` ;
- les réécritures (tactiques `rewrite` et `setoid_rewrite`) valent mieux que des `destruct` car elles ne polluent pas le contexte (au besoin, on peut faire `Require Import Setoid` auparavant) ;
- une induction sur un prédicat inductif permet d'avoir directement la bonne forme pour ses arguments et les hypothèses d'induction ;
- ne pas hésiter à m'envoyer un mail si vous avez des questions sur ce DM.