TD 12 : Résolution

lionel.rieg@ens-lyon.fr

1 Quelques rappels sur Friedman (*cf.* DM 2)

Exercice 1 ***Corrigé du DM 2***

On va corriger ici les questions qui ont posées le plus de difficultés dans le DM 2.

1. Rappeler la définition de la R-traduction.
2. Prouver que $\neg_R\neg_R A^{\neg_R} \vdash_{NJ} A^{\neg_R}$. On ne traitera que les cas où A est une conjonction ou une implication.
3. Prouver que si $\Gamma \vdash_{NK} A$ et si les variables libres de R n'apparaissent pas dans cette dérivation, alors $\Gamma^{\neg_R} \vdash_{NJ} A^{\neg_R}$, où $\Gamma^{\neg_R}$ est l'application de $(\,\cdot\,)^{\neg_R}$ à toutes les formules de Γ. On ne traitera que les cas des règles $\vee_e$ et $\exists_e$.
4. Prouver la R-traduction du schéma de Leibniz dans HA.
5. Prouver que $PA \vdash \forall x \exists y\,(a \doteq b)$ entraîne $HA \vdash \forall x \exists y\,(a \doteq b)$.

2 Résolution propositionnelle

On va montrer la correction et la complétude de la méthode de résolution. Pour rappel, voici la règle de résolution :

$$\frac{C_1 \qquad C_2}{(C_1 \setminus \{a\}) \cup (C_2 \setminus \{\overline{a}\})}\; {\scriptstyle a \in C_1, \overline{a} \in C_2}$$

La clause $C := (C_1 \setminus \{a\}) \cup (C_2 \setminus \{\overline{a}\})$ est appelée la *résolvante* (sur a) des clauses C_1 et C_2.

Un *arbre de résolution* pour un ensemble de clause S est un arbre binaire tel que :

- les nœuds internes et les feuilles sont étiquetés par des clauses,
- les clauses qui étiquettent les feuilles sont dans S,
- la clause qui étiquette un nœud interne est la résolvante des clauses qui étiquettent ses fils

Un *arbre de réfutation* est un arbre de résolution dont la clause associée à la racine est la clause vide $\square$.

Exercice 2 ***Correction de la méthode de résolution***

On va montrer le théorème suivant :

Théorème (Correction)
Si S possède un arbre de réfutation, alors S est insatisfiable.

1. Montrer que la règle de résolution préserve la validité : dans tout modèle $\mathcal{M}$, si $\mathcal{M} \models C_1$ et $\mathcal{M} \models C_2$, alors $\mathcal{M} \models C$.
2. En déduire le théorème.

Exercice 3 ***Complétude de la règle de résolution***

On va démontrer la réciproque du théorème précédent.

Théorème (Complétude)
Si S est insatisfiable, alors S possède un arbre de réfutation.

1. Montrer que s'il existe L et C tels que $L, \overline{L} \in C \in S$, alors $S' = S \setminus \{C\}$ est satisfiable si et seulement si S l'est. De plus, tout arbre de résolution pour S' est un arbre de résolution pour S.
2. Montrer que s'il existe L et C tels que $L \in C \in S$ et $\overline{L}$ n'apparaît pas dans S, alors $S' = S \setminus \{C\}$ est satisfiable si et seulement si S l'est. De plus, tout arbre de résolution pour S' est un arbre de résolution pour S.
3. Pourquoi peut-on supposer que S est fini ?

On suppose que S est fini. La démonstration va alors se faire par récurrence sur le nombre de formules atomiques (aussi appelées *atomes*) distinctes présentes dans S.

4. Démontrer le cas de base de la récurrence.

Pour l'étape inductive, grâce aux questions 1 et 2, on peut supposer que tous les atomes apparaissent positivement et négativement dans S et qu'aucune clause ne contient à la fois L et $\overline{L}$. Prenons un atome a de S. On construit S' comme la réunion des deux ensembles suivants :

$$\{C \mid \exists C_1 \in S \, \exists C_2 \in S. \, C \text{ est la résolvante sur } a \text{ de } C_1 \text{ et } C_2\} \qquad \text{et} \qquad \{C \in S \mid a \notin C \land \overline{a} \notin C\}$$

5. En utilisant S', démontrer le cas inductif.

3 Résolution au premier ordre

À présent, on va passer au cas du premier ordre sans égalité. Dans ce cadre, les clauses peuvent contenir des variables. La règle de résolution est généralisée comme suit :

$$\frac{C_1 \qquad C_2}{[(C_1\rho_1 \setminus D_1\rho_1) \cup (C_2\rho_2 \setminus D_2\rho_2)]\sigma}$$

où :
- $D_1 \subseteq C_1$ et $D_2 \subseteq C_2$,
- les littéraux de D_1 sont tous positifs et ceux de D_2 sont tous négatifs,
- ρ_1 et ρ_2 sont des renommages des variables de C_1 et C_2 de façon à les rendre disjointes,
- σ est un mgu de l'ensemble $D_1\rho_1 \cup D_2\rho_2$ où l'on ignore les signes des littéraux.

On dit alors que C est la résolvante de C_1 et C_2 par rapport à $(D_1, D_2, \rho_1, \rho_2, \sigma)$.

Théorème (Correction)
Si S possède un arbre de réfutation, alors S est insatisfiable.

Lemme (Relèvement)
Si S est un ensemble fini de clause et S_g est un ensemble fini d'instances close de S et si T_g est un arbre de résolution pour S_g, alors il existe un arbre de résolution T pour S tel que la racine de T_g soit une instance close de celle de T.

Théorème (Complétude)
Si S est insatisfiable, alors S possède un arbre de réfutation.

Exercice 4 *Démonstrations*

1. En utilisant la même méthode que pour le cas propositionnel, montrer le théorème de correction.
2. En utilisant le théorème de Herbrand, la complétude de la résolution propositionnelle et le lemme de relèvement, montrer le théorème de complétude.
3. Montrer le lemme de relèvement.

Solutions

▸ Exercice 1

1. la R-traduction est défini inductivement par

$$
\begin{array}{llll}
\bot^{\neg R} := R & (a \doteq b)^{\neg R} := \neg_R\neg_R(a \doteq b) & (A \wedge B)^{\neg R} := A^{\neg R} \wedge B^{\neg R} & (\forall x A)^{\neg R} := \forall x A^{\neg R} \\
\top^{\neg R} := \top & (A \Rightarrow B)^{\neg R} := A^{\neg R} \to B^{\neg R} & (A \vee B)^{\neg R} := \neg_R\neg_R(A^{\neg R} \wedge B^{\neg R}) & (\exists x A)^{\neg R} := \neg_R\neg_R(\exists x A^{\neg R})
\end{array}
$$

2. On rappelle que la règle d'affaiblissement est admissible dans NJ.

   Pour $A \wedge B$:

$$
\begin{prooftree}
\AxiomC{\text{HI sur } A}
\noLine
\UnaryInfC{\neg_R\neg_R A^{\neg R} \vdash A^{\neg R}}
\LeftLabel{\text{Weak}}
\doubleLine
\UnaryInfC{\neg_R\neg_R(A^{\neg R} \wedge B^{\neg R}), \neg_R\neg_R A^{\neg R} \vdash A^{\neg R}}
\LeftLabel{\to_i}
\UnaryInfC{\neg_R\neg_R(A^{\neg R} \wedge B^{\neg R}) \vdash \neg_R\neg_R A^{\neg R} \to A^{\neg R}}
\AxiomC{}
\LeftLabel{\text{Ax}}
\UnaryInfC{\dots}
\AxiomC{}
\LeftLabel{\text{Ax}}
\UnaryInfC{\dots}
\AxiomC{}
\RightLabel{\text{Ax}}
\UnaryInfC{\neg_R\neg_R(A^{\neg R} \wedge B^{\neg R}), \neg_R A^{\neg R}, A^{\neg R} \wedge B^{\neg R} \vdash A^{\neg R} \wedge B^{\neg R}}
\RightLabel{\wedge_e^1}
\UnaryInfC{\neg_R\neg_R(A^{\neg R} \wedge B^{\neg R}), \neg_R A^{\neg R}, A^{\neg R} \wedge B^{\neg R} \vdash A^{\neg R}}
\RightLabel{\to_e}
\BinaryInfC{\neg_R\neg_R(A^{\neg R} \wedge B^{\neg R}), \neg_R A^{\neg R}, A^{\neg R} \wedge B^{\neg R} \vdash R}
\RightLabel{\to_i}
\UnaryInfC{\neg_R\neg_R(A^{\neg R} \wedge B^{\neg R}), \neg_R A^{\neg R} \vdash \neg_R(A^{\neg R} \wedge B^{\neg R})}
\RightLabel{\to_e}
\BinaryInfC{\neg_R\neg_R(A^{\neg R} \wedge B^{\neg R}), \neg_R A^{\neg R} \vdash R}
\RightLabel{\to_i}
\UnaryInfC{\neg_R\neg_R(A^{\neg R} \wedge B^{\neg R}) \vdash \neg_R\neg_R A^{\neg R}}
\RightLabel{\to_e}
\BinaryInfC{\neg_R\neg_R(A^{\neg R} \wedge B^{\neg R}) \vdash A^{\neg R}}
\AxiomC{\text{idem}}
\LeftLabel{\wedge_i}
\BinaryInfC{\neg_R\neg_R(A^{\neg R} \wedge B^{\neg R}) \vdash A^{\neg R} \wedge B^{\neg R}}
\end{prooftree}
$$

   Pour $A \to B$:

$$
\begin{prooftree}
\AxiomC{\text{HI sur } B}
\noLine
\UnaryInfC{\neg_R\neg_R B^{\neg R} \;\vdash\; B^{\neg R}}
\RightLabel{\text{Weak}}
\doubleLine
\UnaryInfC{\Gamma, \neg_R\neg_R B^{\neg R} \;\vdash\; B^{\neg R}}
\RightLabel{\to_i}
\UnaryInfC{\Gamma \;\vdash\; \neg_R\neg_R B^{\neg R} \to B^{\neg R}}
\AxiomC{}
\LeftLabel{\text{Ax}}
\UnaryInfC{\dots}
\AxiomC{}
\LeftLabel{\text{Ax}}
\UnaryInfC{\dots}
\AxiomC{}
\LeftLabel{\text{Ax}}
\UnaryInfC{\dots}
\AxiomC{}
\RightLabel{\text{Ax}}
\UnaryInfC{\Gamma, \neg_R B^{\neg R}, A^{\neg R} \to B^{\neg R} \;\vdash\; A^{\neg R}}
\RightLabel{\to_e}
\BinaryInfC{\Gamma, \neg_R B^{\neg R}, A^{\neg R} \to B^{\neg R} \;\vdash\; B^{\neg R}}
\RightLabel{\to_e}
\BinaryInfC{\Gamma, \neg_R B^{\neg R}, A^{\neg R} \to B^{\neg R} \;\vdash\; R}
\RightLabel{\to_i}
\UnaryInfC{\Gamma, \neg_R B^{\neg R} \;\vdash\; \neg_R(A^{\neg R} \to B^{\neg R})}
\RightLabel{\to_e}
\BinaryInfC{\Gamma, \neg_R B^{\neg R} \;\vdash\; R}
\RightLabel{\to_i}
\UnaryInfC{\Gamma \;\vdash\; \neg_R\neg_R B^{\neg R}}
\RightLabel{\to_e}
\BinaryInfC{\neg_R\neg_R(A^{\neg R} \to B^{\neg R}), A^{\neg R} \;\vdash\; B^{\neg R}}
\RightLabel{\to_i}
\UnaryInfC{\neg_R\neg_R(A^{\neg R} \to B^{\neg R}) \;\vdash\; A^{\neg R} \to B^{\neg R}}
\end{prooftree}
$$

   où $\Gamma := \neg_R\neg_R(A^{\neg R} \to B^{\neg R}), A^{\neg R}$.

3. Pour la règle $\vee_e$:

   Par hypothèse d'induction, on possède des dérivations π_1^*, π_2^* et π_3^* de $\Gamma^{\neg R} \vdash (A \vee B)^{\neg R} \equiv \Gamma^{\neg R} \vdash \neg_R\neg_R A^{\neg R} \vee B^{\neg R}$, $\Gamma^{\neg R}, A^{\neg R} \vdash C^{\neg R}$ et $\Gamma^{\neg R}, B^{\neg R} \vdash C^{\neg R}$ respectivement.

$$
\begin{prooftree}
\AxiomC{\text{Question 2}}
\noLine
\UnaryInfC{\neg_R\neg_R C^{\neg R} \vdash C^{\neg R}}
\doubleLine
\UnaryInfC{\Gamma^{\neg R}, \neg_R\neg_R C^{\neg R} \vdash C^{\neg R}}
\AxiomC{\pi_1^*}
\noLine
\UnaryInfC{\neg_R C^{\neg R} \vdash \neg_R\neg_R(A^{\neg R} \vee B^{\neg R})}
\doubleLine
\UnaryInfC{\Gamma^{\neg R}, \neg_R C^{\neg R} \vdash \neg_R\neg_R(A^{\neg R} \vee B^{\neg R})}
\AxiomC{}
\UnaryInfC{\Gamma' \vdash \neg_R C^{\neg R}}
\AxiomC{}
\UnaryInfC{\Gamma' \vdash A^{\neg R} \vee B^{\neg R}}
\AxiomC{\pi_2^*}
\noLine
\UnaryInfC{\Gamma^{\neg R}, A^{\neg R} \vdash C^{\neg R}}
\doubleLine
\UnaryInfC{\Gamma', A^{\neg R} \vdash C^{\neg R}}
\AxiomC{\pi_3^*}
\noLine
\UnaryInfC{\Gamma^{\neg R}, B^{\neg R} \vdash C^{\neg R}}
\doubleLine
\UnaryInfC{\Gamma', B^{\neg R} \vdash C^{\neg R}}
\RightLabel{\vee_e}
\TrinaryInfC{\Gamma' \vdash C^{\neg R}}
\RightLabel{\to_e}
\BinaryInfC{\Gamma' \vdash R}
\RightLabel{\to_i}
\UnaryInfC{\Gamma^{\neg R}, \neg_R C^{\neg R} \vdash \neg_R(A^{\neg R} \vee B^{\neg R})}
\RightLabel{\to_e}
\BinaryInfC{\Gamma^{\neg R}, \neg_R C^{\neg R} \vdash R}
\RightLabel{\to_i}
\UnaryInfC{\Gamma^{\neg R} \vdash \neg_R\neg_R C^{\neg R}}
\BinaryInfC{\Gamma^{\neg R} \vdash C^{\neg R}}
\end{prooftree}
$$

   où $\Gamma' := \Gamma^{\neg R}, \neg_R C^{\neg R}, A^{\neg R} \vee B^{\neg R}$.

   Pour la règle $\exists_e$:

   Par hypothèse d'induction, on possède des dérivations π_1^* et π_2^* de $\Gamma^{\neg R} \vdash (\exists x A)^{\neg R} \equiv \Gamma^{\neg R} \vdash \neg_R\neg_R(\exists x.A^{\neg R})$ et $\Gamma^{\neg R}, A^{\neg R} \vdash B^{\neg R}$ respectivement. On sait également que $x \notin FV(\Gamma, B)$.

$$
\begin{prooftree}
\AxiomC{Question 2}
\noLine
\UnaryInfC{$\neg_R\neg_R B^{\neg R} \vdash B^{\neg R}$}
\doubleLine
\UnaryInfC{$\Gamma^{\neg R}, \neg_R\neg_R B^{\neg R} \vdash B^{\neg R}$}
\LeftLabel{$\to_i$}
\UnaryInfC{$\Gamma^{\neg R} \vdash \neg_R\neg_R B^{\neg R} \to B^{\neg R}$}
\AxiomC{π_1^*}
\noLine
\UnaryInfC{$\Gamma^{\neg R} \vdash \neg_R\neg_R(\exists x\, A^{\neg R})$}
\doubleLine
\UnaryInfC{$\Gamma^{\neg R}, \neg_R B^{\neg R} \vdash \neg_R\neg_R(\exists x\, A^{\neg R})$}
\AxiomC{}
\LeftLabel{Ax}
\UnaryInfC{$\Gamma'' \vdash \neg_R B^{\neg R}$}
\AxiomC{π_2^*}
\noLine
\UnaryInfC{$\Gamma^{\neg R}, A^{\neg R} \vdash B^{\neg R}$}
\doubleLine
\UnaryInfC{$\Gamma'', A^{\neg R} \vdash B^{\neg R}$}
\AxiomC{}
\RightLabel{Ax}
\UnaryInfC{$\Gamma'' \vdash \exists x\, A^{\neg R}$}
\RightLabel{$\exists_e$}
\BinaryInfC{$\Gamma'' \vdash B^{\neg R}$}
\RightLabel{$\to_e$}
\BinaryInfC{$\Gamma'' \vdash R$}
\RightLabel{$\to_i$}
\UnaryInfC{$\Gamma^{\neg R}, \neg_R B^{\neg R} \vdash \neg_R(\exists x\, A^{\neg R})$}
\RightLabel{$\to_e$}
\BinaryInfC{$\Gamma^{\neg R}, \neg_R B^{\neg R} \vdash R$}
\RightLabel{$\to_i$}
\UnaryInfC{$\Gamma^{\neg R} \vdash \neg_R\neg_R B^{\neg R}$}
\RightLabel{$\to_e$}
\BinaryInfC{$\Gamma^{\neg R} \vdash B^{\neg R}$}
\end{prooftree}
$$

où $\Gamma'' := \Gamma^{\neg R}, \neg_R B^{\neg R}, \exists x\, A^{\neg R}$. Pour la règle $\exists_e$, il faut s'assurer que $x \notin FV(\Gamma^{\neg R}, \neg_B B^{\neg R}, B^{\neg R}) = FV(\Gamma, B, R)$. On a déjà $x \notin FV(\Gamma, B)$ donc il suffit d'avoir $x \notin FV(R)$, qui est une hypothèse.

4. On se place dans l'arithmétique de Heyting, c'est-à-dire NJ à laquelle on ajoute certains axiomes, notamment le schéma de Leibniz. On veut démontrer la traduction du schéma de Leibniz, *i.e.* construire une dérivation de $PA, \Gamma^{\neg R} \vdash \forall x \forall y.\, \neg_R\neg_R(a \doteq b) \to A^{\neg R}[x/z] \to A^{\neg R}[y/z]$. Puisque l'affaiblissement est admissible dans NJ, on peut se retreindre à construire une dérivation de $PA \vdash \forall x \forall y.\, \neg_R\neg_R(a \doteq b) \to A^{\neg R}[x/z] \to A^{\neg R}[y/z]$.

$$
\begin{prooftree}
\AxiomC{}
\LeftLabel{Ax}
\UnaryInfC{$\Gamma, \neg_R A^{\neg R}[y/z], a \doteq b \vdash \forall x \forall y. a \doteq b \to A^{\neg R}[x/z] \to A^{\neg R}[y/z]$}
\LeftLabel{$\forall_e$}
\UnaryInfC{$\Gamma, \neg_R A^{\neg R}[y/z], a \doteq b \vdash \forall y.\, a \doteq b \to A^{\neg R}[x/z] \to A^{\neg R}[y/z]$}
\LeftLabel{$\forall_e$}
\UnaryInfC{$\Gamma, \neg_R A^{\neg R}[y/z], a \doteq b \vdash a \doteq b \to A^{\neg R}[x/z] \to A^{\neg R}[y/z]$}
\AxiomC{}
\RightLabel{Ax}
\UnaryInfC{$\Gamma, \neg_R A^{\neg R}[y/z], a \doteq b \vdash a \doteq b$}
\LeftLabel{$\to_e$}
\BinaryInfC{$\Gamma, \neg_R A^{\neg R}[y/z], a \doteq b \vdash A^{\neg R}[x/z] \to A^{\neg R}[y/z]$}
\AxiomC{}
\RightLabel{Ax}
\UnaryInfC{$\Gamma, \neg_R A^{\neg R}[y/z], a \doteq b \vdash A^{\neg R}[x/z]$}
\BinaryInfC{$\Gamma, \neg_R A^{\neg R}[y/z], a \doteq b \vdash A^{\neg R}[y/z]$}
\end{prooftree}
$$

$$\vdots$$

$$
\begin{prooftree}
\AxiomC{Question 2}
\noLine
\UnaryInfC{$\neg_R\neg_R A^{\neg R}[y/z] \vdash A^{\neg R}[y/z]$}
\doubleLine
\LeftLabel{Weak}
\UnaryInfC{$\Gamma, \neg_R\neg_R A^{\neg R}[y/z] \vdash A^{\neg R}[y/z]$}
\LeftLabel{$\to_i$}
\UnaryInfC{$\Gamma \vdash \neg_R\neg_R A^{\neg R}[y/z] \to A^{\neg R}[y/z]$}
\AxiomC{}
\RightLabel{Ax}
\UnaryInfC{$\ldots \vdash \neg_R(a \doteq b)$}
\AxiomC{}
\RightLabel{Ax}
\UnaryInfC{$\ldots \vdash \neg_R A^{\neg R}[y/z]$}
\AxiomC{$\vdots$}
\noLine
\UnaryInfC{$\Gamma, \neg_R A^{\neg R}[y/z], a \doteq b \vdash A^{\neg R}[y/z]$}
\RightLabel{$\to_e$}
\BinaryInfC{$\Gamma, \neg_R A^{\neg R}[y/z], a \doteq b \vdash R$}
\RightLabel{$\to_i$}
\UnaryInfC{$\Gamma, \neg_R A^{\neg R}[y/z] \vdash \neg_R(a \doteq b)$}
\RightLabel{$\to_e$}
\BinaryInfC{$\Gamma, \neg_R A^{\neg R}[y/z] \vdash R$}
\RightLabel{$\to_i$}
\UnaryInfC{$\Gamma \vdash \neg_R\neg_R A^{\neg R}[y/z]$}
\RightLabel{$\to_e$}
\BinaryInfC{$\Gamma \vdash A^{\neg R}[y/z]$}
\RightLabel{$\to_i$}
\UnaryInfC{$HA, \neg_R\neg_R(a \doteq b) \vdash A^{\neg R}[x/z] \to A^{\neg R}[y/z]$}
\RightLabel{$\to_i$}
\UnaryInfC{$HA \vdash \neg_R\neg_R(a \doteq b) \to A^{\neg R}[x/z] \to A^{\neg R}[y/z]$}
\RightLabel{$\forall_i$}
\UnaryInfC{$HA \vdash \forall y.\, \neg_R\neg_R(a \doteq b) \to A^{\neg R}[x/z] \to A^{\neg R}[y/z]$}
\RightLabel{$\forall_i$}
\UnaryInfC{$HA \vdash \forall x \forall y.\, \neg_R\neg_R(a \doteq b) \to A^{\neg R}[x/z] \to A^{\neg R}[y/z]$}
\end{prooftree}
$$

où $\Gamma := HA, \neg_R\neg_R(a \doteq b), A^{\neg R}[x/z]$.

5. Pour simplifier, on ne va donner que les étapes du raisonnement sans construire explicitement l'arbre de dérivation. On pose $R := \exists y(a \doteq b)[z/x]$ avec z une variable fraîche. D'après la question précédente, on a une dérivation π de $HA \vdash \forall x \neg_R\neg_R \exists y \neg_R\neg_R(a \doteq b)$. Mais $\neg\exists x\, A$ est intuitionnistiquement équivalent à $\forall x \neg A$ (et la même preuve fonctionne pour $\neg_R$) donc on peut transformer π en une dérivation de $HA \vdash \forall x\, \neg_R \forall y \neg_R\neg_R\neg_R(a \doteq b)$. On peut simplifier la triple négation en une négation simple : $HA \vdash \forall x\, \neg_R \forall y \neg_R(a \doteq b)$. À présent, on instancie x avec z pour obtenir $HA \vdash \neg_R \forall y \neg_R(a \doteq b)$. Par ailleurs, $\vdash (a \doteq b)[z/x] \to \exists y\, (a \doteq b)[z/x]$ donc $\vdash \forall y\, (a \doteq b)[z/x] \to \exists y\, (a \doteq b)[z/x] \equiv \forall y\, \neg_R(a \doteq b)[z/x]$. En combinant ça avec la dérivation précédente, on obtient une dérivation de $HA \vdash R \equiv \exists y(a \doteq b)[z/x]$. Puisque z est une variable fraîche, on en déduit que $HA \vdash \forall z \exists y\, (a \doteq b)[z/x]$ qui donne par α-conversion $HA \vdash \forall x \exists y\, (a \doteq b)$.

▶ **Exercice 2**

1. Soit un modèle $\mathcal{M}$ dans lequel C_1 et C_2 sont valides. Montrons que $\mathcal{M} \models C$. Comme C_1 et C_2 sont des disjonctions, $\mathcal{M} \models C_i$ revient à dire qu'il existe $L_1 \in C_1$ et $L_2 \in C_2$ tel que $\mathcal{M} \models L_1$ et $\mathcal{M} \models L_2$. On en déduit que $L_1 \neq \overline{L}_2$ donc l'un des deux littéral apparaît dans C. Ainsi C est valide dans $\mathcal{M}$.
2. La contraposée de la question précédente nous dit que si la résolvante C n'est pas valide dans un modèle $\mathcal{M}$, alors C_1 ou C_2 n'est pas valide dans $\mathcal{M}$. Ainsi, à chaque nœud interne de l'arbre de réfutation de S, si la résolvante n'est pas valide, alors l'un des fils du nœud n'est pas valide.

   Prenons un modèle $\mathcal{M}$ quelconque. Comme la racine est la clause vide qui ne peut être valide dans aucun modèle, par induction sur la structure de l'arbre de réfutation, on montre que l'une des feuilles (qui dépend de $\mathcal{M}$) n'est pas valide dans $\mathcal{M}$, donc que l'une des clause de S ne l'est pas. Ceci entraîne que S n'est pas valide dans $\mathcal{M}$.

▶ **Exercice 3**

1. Comme $S' \subseteq S$, S satisfiable entraîne S' satisfiable. Réciproquement, la clause C contenant L et $\overline{L}$ est trivialement satisfiable et les autres clauses de S sont des clauses de S' donc si S' est satisfiable, S l'est également. Puisque $S' \subseteq S$, tout arbre de résolution de S' est un arbre de résolution de S (on n'impose pas que toutes les clauses de S apparaissent aux feuilles de l'arbre, un sous-ensemble suffit).
2. L'inclusion $S \subseteq S'$ donne le sens direct. Pour le sens réciproque, on se donne un modèle $\mathcal{M}$ validant S'. On modifie la valeur associée au littéral L pour qu'il soit vrai. Cela ne peut invalider une clause de S' car $\overline{L}$ n'y apparaît pas. Dans ce nouveau modèle $\mathcal{M}'$, on a $\mathcal{M}' \models S'$ et $\mathcal{M}' \models C$ (car $\mathcal{M}' \models L$) donc $\mathcal{M}' \models S$.
3. Puisque S est insatisfiable, il suffit d'utiliser le théorème de compacité pour se ramener au cas où S est fini.
4. Dans le cas de base, S ne contient aucun atome. Il n'y alors que deux possibilité : ou bien $S = \varnothing$, ou bien $S = \{\square\}$. L'hypothèse que S est insatisfiable permet d'éliminer le premier cas. L'arbre réduit à la feuille $\square$ est alors un arbre de réfutation de S.
5. Puisque S' contient un atome de moins que S, on veut utiliser l'hypothèse de récurrence sur S'. Pour cela, il faut démontrer que S' est insatisfiable.

   Remarquons tout d'abord que a n'apparaît pas dans S'. En effet, par l'absurde supposons que a apparaisse dans $C \in S'$. Alors C a été construit par la première méthode, donc est la résolvante sur a de deux clauses C_1 et C_2 de S. Pour que a subsiste après une étape de de résolution sur a, il faut qu'initialement a et $\overline{a}$ soit tous deux présents dans une clause. Ainsi C_1 ou C_2 contenait à la fois a et $\overline{a}$, ce qui contredit les hypothèses faites sur S.

   Pour montrer que S' est insatisfiable, prenons un modèle $\mathcal{M}$, supposons par l'absurde que $\mathcal{M} \models S'$ et montrons qu'il existe un modèle $\mathcal{M}'$ validant S. On constate que si a n'apparaît pas dans $C \in S$, alors $C \in S'$ et $\mathcal{M} \models C$. Reste alors à traiter le cas des formules qui contiennent l'atome a car leur validité peut dépendre de celle de a. Pour détailler ceci, on définit les deux sous-ensembles de S suivants, qui contiennent les clauses dont la validité dans $\mathcal{M}$ dépend de celle de a ou $\overline{a}$.

   $$S_1 = \{C \in S \mid a \in C \text{ et } \mathcal{M} \not\models C \setminus \{a\}\} \qquad S_2 = \{C \in S \mid \overline{a} \in C \text{ et } \mathcal{M} \not\models C \setminus \{\overline{a}\}\}\,.$$

   On va distinguer 4 cas, suivant que S_1 et/ou S_2 sont vides :
   - Si S_1 et S_2 sont vides, cela signifie que la validité dans $\mathcal{M}$ des clauses de S contenant l'atome a ne dépend pas de la validité de a. Les autres clauses n'en dépendent pas non plus (car elle ne contiennent pas a) donc $\mathcal{M} \models S$ et on prend alors $\mathcal{M}' := \mathcal{M}$.
   - Si S_1 et S_2 sont non vides, prenons $C_1 \in S_1$ et $C_2 \in S_2$. La résolvante C de C_1 et C_2 n'est pas valide dans $\mathcal{M}$ sans quoi $C_1 \setminus \{a\}$ ou $C_2 \setminus \{\overline{a}\}$ le serait. Mais $C \in S'$ par définition de S' donc $\mathcal{M} \not\models S'$, ce qui est absurde.
   - Si $S_1 \neq \varnothing$ et $S_2 = \varnothing$, on prend $\mathcal{M}'$ tel que $\mathcal{M} \models a$ et $\mathcal{M}'$ coïncide avec $\mathcal{M}$ partout ailleurs. On a alors $\mathcal{M}' \models S$:
     - si a n'apparaît pas dans $C \in S$, $C \in S'$ et $\mathcal{M}' \models C \iff \mathcal{M} \models C$ qui est vrai puisque $\mathcal{M} \models S'$.
     - si $a \in C \in S$, $\mathcal{M}' \models C$

– si $\overline{a} \in C \in S$, $\mathcal{M}' \models C \setminus \{\overline{a}\}$ (car S_2 est vide) donc $\mathcal{M}' \models C$.
– Ce cas est le symétrique du précédent en prenant $\mathcal{M}' \models \overline{a}$.

Ainsi S' est insatisfiable et on peut lui appliquer l'hypothèse de récurrence. Reste à voir que l'existence d'un arbre de réfutation T' pour S' entraîne l'existence d'un arbre de réfutation T pour S.

Toute les formules de S' sont obtenues à partir de formules de S, soit par résolution, soit telle quelle. Il suffit donc d'ajouter une étape de résolution au niveau des feuilles de T' auxquelles sont associées des formules de S' obtenues par résolution sur a de deux formules de S. Les deux nouvelles feuilles sont alors des clauses de S et l'arbre T ainsi construit est un arbre de résolution pour S. Comme sa racine n'a pas changée, c'est un arbre de réfutation pour S.

▸ Exercice 4

On commence par rappeler quelques notes du cours de Colin à propos du théorème de Herbrand et des formes normales de Skolem et de Herbrand.

Preliminaries

We assume given a fixed first-order language **without equality** $\mathcal{L} = (\mathcal{V}, \Sigma, \Pi)$, with at least one constant, say c.

A Reminder on Herbrand's Theorem

Let A be a prenex sentence over $\mathcal{L}$. Thus A is of the form

$$\exists\vec{x}_1\forall\vec{y}_1 \cdots \exists\vec{x}_n\forall\vec{y}_n B[\vec{x}_1, \vec{y}_1, \dots, \vec{x}_n, \vec{y}_n]$$

where B is quantifier free. The formula B is called the *matrix* of A. Note that the above form is unique as soon as we require $|\vec{y}_1|, \dots, |\vec{x}_n| > 0$.

The *Herbrand normal form* of A is the formula A^H defined as follows.

– The language of A^H is
$$\mathcal{L}^H(A) \quad := \quad \mathcal{L} + \{\vec{\mathsf{f}}_i \mid 1 \leqslant i \leqslant n\}$$
where all $\vec{\mathsf{f}}_i$ are of arity $|\vec{x}_1| + \cdots + |\vec{x}_i|$.

– The formula A^H is
$$A^H \quad := \quad \exists\vec{x}_1 \cdots \exists\vec{x}_n B[\vec{x}_1, \vec{\mathsf{f}}_1(\vec{x}_1), \dots, \vec{x}_n, \vec{\mathsf{f}}_n(\vec{x}_1, \dots, \vec{x}_n)]$$

Using the Axiom of Choice (actually its contraposition), it is easy to see that

Lemma 1 *$\models A$ if and only if $\models A^H$.*

Note that in Lemma 1, the direction ($\Longrightarrow$) is direct, and it is direction ($\Longleftarrow$) which uses the contrapositive of the Axiom of Choice. Using the Completeness Theorem of first-order logic, we thus get

$$\vdash A \qquad \text{if and only if} \qquad \vdash A^H \tag{H}$$

Hebrand's Theorem is an effective approximation of (H). It says two things.

First, given a proof of A, one can effectively build a proof of a sentence $\tilde{A}$ of the form

$$\bigvee_{1\leqslant j\leqslant m} B[\vec{t}_{j,1}, \vec{\mathsf{f}}_1(\vec{t}_{j,1}), \dots, \vec{t}_{j,n}, \vec{\mathsf{f}}_n(\vec{t}_{j,1}, \dots, \vec{t}_{j,n})]$$

where $(\vec{t}_{j,1}, \ldots, \vec{t}_{j,n})_{1 \leqslant j \leqslant m}$ is a finite sequence of closed terms on the language of A^H. Such $\tilde{A}$ is called an *Hebrand disjunction* of A.

Hebrand's Theorem also says the converse : From a proof of an Hebrand disjunction of A, one can effectively build a proof of A.

Theorem 2 (Herbrand) *Given a sentence A in prenex form,*

(i) from a derivation of $\vdash A$, one can effectively build an Herbrand disjunction $\tilde{A}$ of A and a derivation of $\vdash \tilde{A}$,

(ii) from a derivation a $\vdash \tilde{A}$ of an Hebrand disjunction $\tilde{A}$ of A, one can effectively build a derivation of $\vdash A$.

For the resolution, we will use the following corollary to Herbrand's Theorem :

Corollary 3 *Let A be a prenex sentence of the form*

$$\exists x_1 \ldots \exists x_n B[x_1, \ldots, x_n] \qquad \text{with } B \text{ quantifier free}$$

Then A is valid ($\models A$) if and only if there are closed terms $(t_{j,i})_{1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m}$ of $\mathcal{L}$ such that

$$\models \bigvee_{1 \leqslant j \leqslant m} B[t_{j,1}, \ldots, t_{j,n}]$$

Skolem Normal Forms and Unsatisfiability

Let A be a prenex sentence over $\mathcal{L}$. The *Skolem Normal Form of A* is

$$A^S \quad := \quad \neg(\neg A)^H$$

In particular, if A is of the form

$$\forall \vec{x}_1 \exists \vec{y}_1 \cdots \forall \vec{x}_n \exists \vec{y}_n B[\vec{x}_1, \vec{y}_1, \ldots, \vec{x}_n, \vec{y}_n] \qquad \text{with } B \text{ quantifier free}$$

then

$$A^S \quad = \quad \forall \vec{x}_1 \cdots \forall \vec{x}_n B[\vec{x}_1, \vec{\mathsf{f}}_1(\vec{x}_1), \ldots, \vec{x}_n, \vec{\mathsf{f}}_n(\vec{x}_1, \ldots, \vec{x}_n)]$$

The Skolem normal form of A can be seen as an "unsatisfiability normal form" of A, while Herbrand normal forms can be seen as a "validity normal forms".

Lemma 4 *A is unsatisfiable (i.e. for all model $\mathcal{M}$ of $\mathcal{L}$, $\mathcal{M} \not\models A$) if and only if A^S is unsatisfiable.*

Démonstration. By Lemma 1, since A is unsatisfiable if and only if $\neg A$ is valid. ■

In particular, Corollary 3 gives :

Corollary 5 *Let A be a prenex sentence of the form*

$$\forall x_1 \ldots \forall x_n B[x_1, \ldots, x_n] \qquad \text{with } B \text{ quantifier free}$$

Then A is unsatisfiable if and only if there are closed terms $(t_{j,i})_{1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m}$ of $\mathcal{L}$ such that

$$\text{for all model } \mathcal{M} \text{ of } \mathcal{L}, \quad \mathcal{M} \not\models \bigwedge_{1 \leqslant j \leqslant m} B[t_{j,1}, \ldots, t_{j,n}]$$

Clauses

A *literal* is either an atomic formula of the negation of an atomic formula. A formula is in *conjunctive normal form* if it is a conjunction of disjunctions of literals.
Let now A be a prenex sentence of the form

$$\forall \vec{x}_1 \exists \vec{y}_1 \cdots \forall \vec{x}_n \exists \vec{y}_n B[\vec{x}_1, \vec{y}_1, \ldots, \vec{x}_n, \vec{y}_n]$$

where B is in conjunctive normal form. Then, since universal quantifications commute over conjunctions, A^S is equivalent to a formula of the form

$$(\forall \vec{z}_1 C_1) \wedge \cdots \wedge (\forall \vec{z}_m C_m)$$

Definition 6 – *A* clause *is a set of literals.*
– *Let $\mathcal{M}$ be a model of $\mathcal{L}$. A clause C is* valid in *$\mathcal{M}$, notation $\mathcal{M} \models C$, if for all assignment $v : \mathcal{V} \to \mathcal{M}$ there is a literal $L \in C$ such that $(\mathcal{M}, v) \models L$.*
– *A set of clauses S is satisfiable if there is a model $\mathcal{M}$ of $\mathcal{L}$ such that every clause of S is valid in $\mathcal{M}$.*

Note that the empty clause $\square$ is valid in no model.
Consider now a formula A^S equivalent to a formula

$$(\forall \vec{z}_1 C_1) \wedge \cdots \wedge (\forall \vec{z}_m C_m)$$

as above, where each C_j is a disjunction of literals :

$$C_j \quad = \quad L_{j,1} \vee \cdots \vee L_{j,k_j}$$

We can assume the vectors of variables $(\vec{z}_j)_{1 \leqslant j \leqslant m}$ to be pairwise disjoints. For each $j \in \{1, \ldots, m\}$ consider the clause

$$\hat{C} := \{L_{j,l} \mid 1 \leqslant l \leqslant k_j\}$$

Lemma 7 *A^S is unsatisfiable if and only if the set of clauses $\{\hat{C}_1, \ldots, \hat{C}_m\}$ is unsatisfiable.*

A *ground instance* of a clause C is a clause of the form $C\sigma$ where σ is a substitution such that $\mathrm{FV}(C) \subseteq \mathrm{dom}(\sigma)$ and $\sigma(x)$ is closed for all $x \in \mathrm{FV}(C)$.
We thus get the following consequence of Corollary 5 :

Corollary 8 *A set of clause S is unsatisfiable is and only if there is a set S_g of ground instances of clauses of S such that S_g is unsatisfiable.*

Avant de reprendre les questions de l'exercice, rappelons qu'au premier ordre, la validité d'une formule dépend de la façon dont on interprète ses variables libres. Ainsi, la validité de la clause C dans le modèle $\mathcal{M}$ avec la valuation des variables libres ρ s'écrit $\mathcal{M} \models_\rho C$ où ρ est une fonction partielle de l'ensemble des variables dans $\mathcal{M}$ qui sert à interpréter les variables libres de $\mathcal{M}$. La validité dans un modèle $\mathcal{M}$ est alors définie comme la validité pour toute les valuations ρ possibles :

$$\mathcal{M} \models C \iff \forall \rho. \mathcal{M} \models_\rho$$

1. La démonstration de la correction de la résolution au premier ordre est similaire au cas propositionnel. Comme précédemment, on commence par démontrer le lemme suivant

   Lemme (Préservation de la validité par résolution)
   Soit $\dfrac{C_1 \quad C_2}{C}$ *une application de la règle de résolution. Si* $\mathcal{M} \models C_1$ *et* $\mathcal{M} \models C_2$*, alors* $\mathcal{M} \models C$*.*

   Démonstration. Soient D_1, D_2, ρ_1, ρ_2 et σ comme dans la règle de résolution entre C_1 et C_2 pour obtenir C. On se donne ρ et on veut montrer que $\mathcal{M} \models_\rho C$. On note $\{x_1, \dots, x_n\}$ et $\{y_1, \dots, y_m\}$ les variables qui apparaissent dans $D_1\rho_1$ et dans $D_1\rho_2$ respectivement. Ces deux ensembles sont disjoints par hypothèses sur ρ_1 et ρ_2. On peut alors définir

   $$\nu'(z) := \begin{cases} \nu(\sigma(x_i) & \text{si } z = x_i \\ \nu(\sigma(y_j) & \text{si } z = y_j \\ \nu(z) & \text{sinon} \end{cases}$$

   Le fait que $\mathcal{M} \models C_1$ et $\mathcal{M} \models C_2$ nous donne que $\mathcal{M} \models_{\nu' \circ \rho_1} C_1$ et $\mathcal{M} \models_{\nu' \circ \rho_2} C_2$. Ceci signifie qu'il existe L_1 et L_2 tels que $\mathcal{M} \models_{\nu' \circ \rho_1} L_1$ et $\mathcal{M} \models_{\nu' \circ \rho_2} L_2$ donc $\mathcal{M} \models_{\nu'} L_1\rho_1$ et $\mathcal{M} \models_{\nu'} L_2\rho_2$. On va montrer que $\mathcal{M} \models_\nu L_1\rho_1\sigma$, $\mathcal{M} \models_\nu L_2\rho_2\sigma$ et que $L_1\rho_1\sigma \in C$ ou $L_2\rho_2\sigma \in C$, ce qui nous permettra d'en déduire que $\mathcal{M} \models_\nu C$.

   Puisque $\mathcal{M} \models_{\nu'} L_i\rho_i$, il suffit de montrer que $\nu' = \nu \circ \sigma$ pour avoir $\mathcal{M} \models_{\nu\circ\sigma} L_i\rho_i$ qui est équivalent à $\mathcal{M} \models_\nu L_1\rho_1\sigma$. Montrons donc que pour toute variable z, $\nu'(z) = \nu(\sigma(z))$. Par définition, c'est déjà le cas pour $x_1, \dots, x_n$ et $y_1, \dots, y_m$. Prenons une variable z de $L_1\rho_1$ qui ne soit pas dans l'image de ρ_1 ou de ρ_2. De ce fait, z n'est pas dans le domaine de σ donc $\nu'(z) = \nu(z) = (\nu \circ \sigma)(z)$. De $\mathcal{M} \models_\nu L_1\rho_1\sigma$ et $\mathcal{M} \models_\nu L_2\rho_2\sigma$ on déduit que $L_1\rho_1\sigma \neq \overline{L_2\rho_2\sigma}$ donc $L_1\rho_1$ et $L_2\rho_2$ ne peuvent être tous les deux dans $D_1\rho_1 \cup D_2\rho_2$. Ainsi, l'un au moins se retrouve dans C et on peut conclure. ∎

   Le reste de la preuve est identique au cas propositionnel.

2. Par compacité, on peut supposer S fini. Si S est insatisfiable, alors d'après le théorème de Herbrand, il existe un ensemble S_g d'instances closes de clauses de S qui est insatisfiable. On peut appliquer la complétude de la résolution propositionnelle pour dire que S_g possède un arbre de réfutation T_g. Par le lemme de relèvement, cela nous donne un arbre de résolution T pour S tel que la racine de T_g est une instance de la racine de T. Mais comme la racine de T_g est $\square$, celle de T est également $\square$.

3. Pour démontrer le lemme de relèvement, on va comme précédemment utiliser un lemme intermédiaire qui relie la règle de résolution propositionnelle et celle au premier ordre.

   Lemme
   Soient C_1 et C_2 deux clauses. Si $C_1\sigma_1$ et $C_2\sigma_2$ en sont deux instances closes, alors toute résolvante de $C_1\sigma_1$ et $C_2\sigma_2$ s'écrit $C\sigma$ où C est une résolvante de C_1 et C_2.

   Démonstration. Notons $C' = (C_1\sigma_1 \setminus \{L\}) \cup (C_2\sigma_2 \setminus \{\overline{L}\})$ la résolvante de $C_1\sigma_1$ et $C_2\sigma_2$. On s'intéresse aux littéraux de C_1 et C_2 qui s'instancient en L et $\overline{L}$: $D_1 = \{L_1 \in C_1 \mid L_1\sigma_1 = L\}$ et $D_1 = \{L_2 \in C_2 \mid L_2\sigma_2 = L\}$. Soient ρ_1 et ρ_2 des renommages disjoints des variables de C_1 et C_2. Comme les images de ρ_1 et ρ_2 sont disjointes, $\sigma'_1 := \sigma_1 \circ \rho_1^{-1}$ et $\sigma'_2 := \sigma_2 \circ \rho_2^{-1}$ sont à supports disjoints et vérifient que $\forall L_1 \in D_1. L_1\rho_1\sigma'_1 = L_1(\sigma'_1 \circ \rho_1) = L_1\sigma_1 = L$ et $\forall L_2 \in D_2. L_2\rho_2\sigma'_2 = L_2(\sigma'_2 \circ \rho_2) = L_2\sigma_2 = \overline{L}$. On définit alors $\sigma' = \sigma'_1 \cup \sigma'_2$. C'est un unificateur de $D_1\rho_1 \cup D_2\rho_2$ lorsque l'on ignore les signes des littéraux : $(D_1\rho_1 \cup D_2\rho_2)\sigma' = \{L, \overline{L}\}$. On prend σ le mgu de $D_1\rho_1 \cup D_2\rho_2$ (aux signes près) qui nous permet d'écrire $\sigma' = \theta \circ \sigma$. Soit C la résolvante de C_1 et C_2 par rapport à $(D_1, D_2, \rho_1, \rho_1, \sigma)$: $C = [(C_1\rho_1 \setminus D_1\rho_1) \cup (C_2\rho_2 \setminus D_2\rho_2)]\sigma$. On a alors $C' = C\theta$ car
   - $D_1\rho_1\sigma\theta = D_1\rho_1(\theta \circ \sigma) = D_1\rho_1\sigma' = D_1(\sigma' \circ \rho_1) = \{L\}$
   - $D_2\rho_2\sigma\theta = D_2\rho_2(\sigma' \circ \theta) = D_2\rho_2\sigma = D_1(\sigma' \circ \rho_2) = \{\overline{L}\}$
   - $C_1\rho_1\sigma\theta = C_1\rho_1(\sigma \circ \theta) = C_1\rho_1\sigma' = C_1(\sigma' \circ \rho_1) = C_1\sigma_1$

– $C_2\rho_2\sigma\theta = C_2\rho_2(\sigma \circ \theta) = C_2\rho_2\sigma' = C_2(\sigma' \circ \rho_2) = C_2\sigma_2$ ■

Une fois qu'on a le lemme, on va faire une induction sur l'arbre T_g :

– Si T_g est une feuille d'étiquette C_g, comme c'est un arbre de résolution pour S_g, on a $C_g \in S_g$ donc C_g est une instance close d'une clause C de S. On pose alors pour T l'arbre réduit à une feuille d'étiquette C. C'est un arbre de résolution pour S puisque $S \in C$ et l'étiquette C_g de la racine de T_g est une instance close de C, l'étiquette de la racine de T.
– Si T_g est un nœud interne d'étiquette C_g dont les sous-arbres gauches et droits sont ${T_g}^1$ et ${T_g}^2$, ces sous-arbres sont des arbres de résolution pour S_g. Par hypothèse d'induction sur ${T_g}^1$ et ${T_g}^2$, il existe T^1 et T^2 arbres de résolution pour S dont les racines C^1 et C^2 s'instancient en ${C^1}_g$ et ${C^2}_g$, les racines de ${T_g}^1$ et ${T_g}^2$: ${C^1}_g = C^1\sigma_1$ et ${C^2}_g = C^2\sigma_2$. Comme C_g est la résolvante de $C^1\sigma_1$ et $C^2\sigma_2$, d'après le lemme, C_g s'écrit $C\sigma$ où C est la résolvante de C^1 et C^2. On recolle alors les arbres T^1 et T^2 en T avec un nœud interne étiqueté par C. Les feuilles de T sont celles de T^1 et T^2 qui sont toutes dans S. De plus la racine de C_g de T_g est une instance de la racine C de T, ce qui conclut la preuve.