3.6 La résolution dans la logique du premier ordre

3.6.1 Motivation

Un premier essai de transférer la résolution propositionnelle vers la logique du premier ordre est comme suit : Étant donné un ensemble fini S de formules, on peut le transformer en un ensemble de clauses universelles S', et puis construire l'ensemble des ces instances de Herbrand $H(S')$. Par le Théorème 3.5.4 et la complétude réfutationelle de la résolution propositionnelle, S est non-satisfaisable si et seulement si $H(S') \vdash_R \square$. Malheureusement, cela ne nous donne pas directement une énumération récursive des ensembles finis non-satisfaisables de formules puisqu'il s'agit d'un ensemble infini de clauses, et il n'est pas évident comment en général un algorithme peut travailler sur un ensemble infini de clauses.

En utilisant le fait qu'il y a des stratégies terminantes et complètes de la résolution propositionnelle on peut quand-même tirer de ce premier essai une énumération récursive des ensembles finis non-satisfaisables de formules : Étant donné S, on construit une énumération effective $s_1, s_2, \ldots$ des instances de Herbrand de S. Soit $\vdash_{RS}$ la déduction par résolution binaire et factorisation selon la stratégie complète et terminante. On lance l'algorithme suivant :

```
T := ∅ ;
n := 0 ;
while □ ∉ T do begin
   T := T ∪ { s_n } ;
   n := n + 1 ;
   while there exists a w ∉ T such that T ⊢_RS w do
       T := T ∪ { w } ;
end ;
print "non satisfaisable" ;
```

Puisque la stratégie est terminante, la boucle intérieure termine toujours après un nombre fini d'itérations. On dit que la boucle intérieure calcule une *saturation* par $\vdash_{RS}$. Dans quelque sens, cet algorithme est déjà un peu plus « intelligent » que l'algorithme présenté à la fin de la section précédente puisque, à chaque itération de la boucle principale, on réutilise le travail fait pendant les itérations précédentes.

La clef pour faire mieux est l'observation que, étant donnée une clause universelle U, son ensemble d'instances de Herbrand est un ensemble très régulier. L'idée est donc de chercher une représentation finie de cet ensemble infini de clauses. La représentation qui vient à l'esprit est d'utiliser une clause universelle en S' comme une représentation de l'ensemble infini $H(S')$. Puis, étant données deux clauses universelles C_1 et C_2, on cherche à exécuter sur un coup un pas de résolution sur toutes les instances de Herbrand de C_1 et toutes les instances de Herbrand de C_2, et de représenter l'ensemble résultant de clauses par une clause du premier ordre.

Pour illustrer cette idée, supposons données deux clauses du premier ordre $C_1 = \forall \bar{x}(P(s) \vee L)$ et $C_2 = \forall \bar{y}(\neg P(t) \vee K)$. On cherche une clause du premier ordre R, telle que les instances de Herbrand de R sont exactement les clauses propositionnelles qu'on peut obtenir par résolution propositionnelle sur P entre une instance de Herbrand de C_1 et une instance de Herbrand de C_2.

On a donc

$$\begin{aligned} H(C_1) &= \{(P(s) \vee L)\sigma \mid \sigma \in \Sigma_g\} \\ H(C_2) &= \{(\neg P(t) \vee K)\tau \mid \tau \in \Sigma_g\} \end{aligned}$$

Un pas de résolution est possible ssi $s\sigma = t\tau$. L'ensemble des clauses closes qu'on peut obtenir par un pas de résolution est donc

$$\{L\sigma \vee K\tau \mid \sigma, \tau \in \Sigma_g, s\sigma = t\tau\} \tag{3.1}$$

Quand $\bar{x} \cap \bar{y} = \emptyset$, cet ensemble est le même que

$$\{(L \vee K)\nu \mid \nu \in \Sigma_g, s\nu = t\nu\} \tag{3.2}$$

Quand $\bar{x} \cap \bar{y} = \emptyset$ on peut supposer que $Dom(\sigma) \cap Dom(\tau) = \emptyset$. Les ensembles (3.1) et (3.2) sont alors égaux parce que
– étant donnée σ et τ en (3.1), on peut définir ν en (3.2) par $\nu = \sigma \circ \tau$ $(= \tau \circ \sigma)$;
– étant donnée τ in (3.2), on peut définir en (3.2) $\sigma = \nu \mid_{\bar{x}}$ et $\tau = \nu \mid_{\bar{y}}$.
Par contre, cette égalité est en général fausse quand $\bar{x}$ et $\bar{y}$ ne sont pas disjoints. Par exemple, les termes x et $f(x)$ ont des instances communes (par exemple, $f(a)$), mais il n'y a aucune substitution ν telle que $x\nu = f(x)\nu$.

3.6.2 Unification

Afin de mettre en œvre l'idée esquissée ci-dessus, il nous faut donc une méthode qui nous permet de représenter toutes les solutions d'une équation entre termes, ou plus généralement toutes les solutions d'une conjonction d'équations entre termes. C'est exactement le problème d'*unification*.

Nous rappelons que pour une substitution σ, son *domaine* est défini comme $Dom(\sigma) := \{x \mid \sigma(x) \neq x\}$.

La substitution obtenue par l'application d'abord de σ puis de τ est notée $\sigma \circ \tau$. Dans cette notation, on a donc que $t(\sigma \circ \tau) = (t\sigma)\tau$ pour tout terme t et toutes substitutions σ et τ.

Une unification σ est *idempotente* si $\sigma = \sigma \circ \sigma$.

Definition 3.6.1 *Un* problème d'unification *est soit* $\bot$*, soit une conjonction d'équations* $s_1 \stackrel{?}{=} t_1 \wedge \ldots \wedge s_n \stackrel{?}{=} t_n$ *avec* $s_i, t_i \in T(\mathcal{F}, \mathcal{X})$.

Une substitution σ *est un* unificateur *d'un problème d'unification* $s_1 \stackrel{?}{=} t_1 \wedge \ldots \wedge s_n \stackrel{?}{=} t_n$ *si, pour tout* i, $s_i\sigma = t_i\sigma$.

Une substitution μ *est un* unificateur le plus général *(angl. : most general unifier), abrégé* mgu, *d'un problème d'unification* $P = (s_1 \stackrel{?}{=} t_1 \wedge \ldots \wedge s_n \stackrel{?}{=} t_n)$ *si*

1. μ *est un unificateur de* P ;
2. μ *est idempotent ;*
3. *pour chaque unificateur* τ *de* P *il existe une substitution* ν *telle que* $\tau = \mu \circ \nu$

Attention, il existe dans la littérature des définitions d'un mgu qui sont légèrement différentes mais équivalentes à la notre, mais aussi des définitions légèrement différentes et pas équivalentes à la notre. En occurrence, on trouve aussi une définition qui n'exige pas que le mgu soit idempotent.

Proposition 3.6.1 *Une substitution σ est idempotente ssi $Dom(\sigma) \cap VIm(\sigma) = \emptyset$.*

Démonstration: Soit $y \in Dom(\sigma) \cap VIm(\sigma)$. Il existe alors $x \in Dom(\sigma)$ tel que $y \in Var(x\sigma)$. Puisque $y\sigma \neq y$, on obtient alors que $x\sigma\sigma \neq x\sigma$, donc σ n'est pas idempotent.

Soit $Dom(\sigma) \cap VIm(\sigma) = \emptyset$. Si $x \notin Dom(\sigma)$ alors $x\sigma = x$ et donc $x\sigma\sigma = x$. Si $x \in Dom(\sigma)$, alors $x\sigma\sigma = x\sigma$ puisque $Dom(\sigma) \cap Var(x\sigma) = \emptyset$. □

Exemple 3.6.1

1. Le problème d'unification $x \stackrel{?}{=} g(y,z) \wedge f(x) \stackrel{?}{=} f(a)$ n'a pas d'unificateur.
2. Le problème d'unification $x \stackrel{?}{=} f(x)$ n'a pas d'unificateur.
3. Soit le problème d'unification $f(x) \stackrel{?}{=} f(g(y))$. Les substitutions suivantes sont des unificateurs :

$$\begin{aligned} \sigma_1 = &\quad \{x \to g(y)\} \\ \sigma_2 = &\quad \{x \to g(a); y \to a\} \end{aligned}$$

   σ_1 est un mgu, σ_2 ne l'est pas. En fait, $\sigma_2 = \sigma_1 \circ \{y \to a\}$, mais il n'y a aucune substitution ν telle que $\sigma_1 = \sigma_2 \circ \nu$.
4. Soit le problème d'unification

$$f(x,x) \stackrel{?}{=} f(x,y) \wedge g(x) \stackrel{?}{=} g(g(z)) \wedge f(z,a) \stackrel{?}{=} f(z',a)$$

   Les substitutions suivantes sont toutes des mgu :

$$\{x \to g(z'); y \to g(z'); z \to z'\}$$
$$\{x \to g(z); y \to g(z); z' \to z\}$$

5. Soit le problème d'unification $g(x) \stackrel{?}{=} g(g(z)) \wedge f(a,z) \stackrel{?}{=} f(a,y)$. Les substitutions suivantes sont des unificateurs :

$$\begin{aligned} \sigma_1 = &\quad \{x \to g(z); y \to z\} \\ \sigma_2 = &\quad \{x \to g(w); y \to w; z \to w; w \to y\} \end{aligned}$$

   σ_1 est un mgu, en fait $\sigma_2 = \sigma_1 \circ \{z \to w, w \to y\}$. On aussi que $\sigma_1 = \sigma_2 \circ \{w \to z; y \to w\}$, mais σ_2 n'est pas idempotent (parce que $w \in Dom(\sigma_2) \cap VIm(\sigma_2)$) donc pas un mgu.

Proposition 3.6.2 *Soit μ un mgu d'un problème d'unification P. Alors pour toute substitution σ :*

$$\sigma \textit{ est un unificateur de } P \quad \Leftrightarrow \quad \sigma = \mu \circ \sigma$$

Démonstration: Selon la définition d'un mgu, $\mu \circ \sigma$ est un unificateur de P.

Soit σ un unificateur de P. Il existe donc une substitution τ telle que $\sigma = \mu \circ \tau$. Puisque μ est idempotent, $\sigma = \mu \circ \tau = \mu \circ \mu \circ \tau = \mu \circ \sigma$. □

$$\text{Decompose} \quad \frac{\phi \wedge f(s_1, \ldots, s_n) \stackrel{?}{=} f(t_1, \ldots, t_n)}{\phi \wedge s_1 \stackrel{?}{=} t_1 \wedge \ldots \wedge s_n \stackrel{?}{=} t_n}$$

$$\text{Clash} \quad \frac{\phi \wedge f(s_1, \ldots, s_n) \stackrel{?}{=} g(t_1, \ldots, t_m)}{\bot} \qquad \text{si } f \neq g$$

$$\text{Trivial} \quad \frac{\phi \wedge s \stackrel{?}{=} s}{\phi}$$

$$\text{Replace} \quad \frac{\phi \wedge x \stackrel{?}{=} t}{\phi[x \mapsto t] \wedge x \stackrel{?}{=} t} \qquad \text{si } x \in \mathrm{Var}(\phi) \text{ et } x \notin \mathrm{Var}(t)$$

$$\text{Occur Check} \quad \frac{\phi \wedge x \stackrel{?}{=} t}{\bot} \qquad \text{si } x \in \mathrm{Var}(t) \text{ et } t \notin \mathcal{X}$$

FIG. 3.3 – Règles d'unification.

Exercice 74
Montrer que le mgu est unique à renommage près :
Soient μ et ν deux mgus d'un problème d'unification P. Montrer qu'il y a un *renommage* ρ tel que $\mu = \nu \circ \rho$.

Un système de règles pour les problèmes d'unification est donné sur la Figure 3.3. Dans ces règles, ont suppose que le symbole $\wedge$ est associative et commutative. De plus, on suppose que le symbole d'égalité est commutative, c'est-à-dire on identifie $s \stackrel{?}{=} t$ avec $t \stackrel{?}{=} s$. On note $P \rightarrow_U P'$ quand on peut transformer par une des règles de la Figure 3.3 un problème d'unification P dans un problème d'unification P'.

Proposition 3.6.3 *Les règles de la Figure 3.3 sont correctes : Si $P \rightarrow_U P'$ alors P et P' ont les mêmes unificateurs.*

On dit qu'une variable x est *éliminée* dans un problème d'unification P quand P est de la forme $x \stackrel{?}{=} t \wedge P'$ telle que $x \notin \mathrm{Var}(t)$ et $x \notin \mathrm{Var}(P')$.

Proposition 3.6.4 *Les règles de la Figure 3.3 terminent : Il n'y a pas de séquence infinie $P_1 \rightarrow_U \ldots \rightarrow_U P_n \rightarrow_U \ldots$*

Nous définissons deux fonctions de mesure sur les problèmes d'unification :
– $\mu_1(P) := card(\{x \in \mathrm{Var}(P) \mid x \text{ n'est pas éliminée en } P\})$
– $\mu_2(s_1 \stackrel{?}{=} t_1 \wedge \ldots \wedge s_n \stackrel{?}{=} t_n) := \Sigma(|s_i| + |t_i|)$

Exercice 75
Est-il vrai que, si $P_1 \rightarrow_U P_2$ alors $\mu_1(P_2) < \mu_1(P_1)$? Est-il vrai que $\mu_2(P_2) < \mu_2(P_1)$? Est-il vrai que $\mu_1(P_2) + \mu_2(P_2) < \mu_1(P_1) + \mu_2(P_1)$?

Definition 3.6.2 *Un* quasi-ordre *est une relation binaire $\precsim$ sur un ensemble D qui est réflexive ($x \precsim x$ pour tout x) et transitive (pour tous x, y et z, $x \precsim y$ et $y \precsim z$ implique $x \precsim z$).*

Si $\precsim$ est un quasi-ordre, on note $\approx$ sa relation d'équivalence associée ($x \approx y$ ssi $x \precsim y$ et $y \precsim x$), et $\prec$ sa partie stricte ($x \prec y$ ssi $x \precsim y$ et $x \not\precsim y$).
Par exemple, si μ est une application de D vers $\mathbb{N}$, alors la relation $\precsim_\mu$ définie par $x \precsim y$ ssi $\mu(x) \leq \mu(y)$ est un quasi-ordre. On a $x \approx_\mu y$ ssi $\mu(x) = \mu(y)$, et $x \prec_\mu y$ ssi $\mu(x) < \mu(y)$.

Definition 3.6.3 *Un quasi-ordre $\precsim$ est* noethérien *(angl. : well-founded), s'il n'y a pas de séquence $x_1, x_2, \ldots$ infinie avec $x_{i+1} \prec x_i$ pour tout $i \geq 0$.*

Par exemple, un quasi-ordre $\precsim_\mu$ défini par une application $D \rightarrow \mathbb{N}$ est un quasi-ordre noethérien.

Exercice 76
Soient $\precsim_1$ et $\precsim_2$ deux quasi-ordres noethériens sur D. Montrer que le quasi-ordre suivant sur D est également noethérien :

$$d \precsim_{1,2} e \quad \Leftrightarrow \quad d \prec_1 e, \text{ ou } d \approx_1 e \text{ et } d \precsim_2 e$$

On appelle le quasi-ordre $\precsim_{1,2}$ la *combinaison lexicographique* des quasi-ordres $\precsim_1$ et $\precsim_2$.
Démonstration: (de Proposition 3.6.4) Nous utilisons le quasi-ordre noethérien qui est la combinaison lexicographique des deux quasi-ordres noethériens $\precsim_{\mu_1}$ et $\precsim_{\mu_2}$. On montre facilement que chaque application d'une règle d'unification sur un problème d'unification donne un problème plus petit :

	μ_1	μ_2
Decompose	$\leq$	$<$
Clash	$\leq$	$<$
Trivial	$\leq$	$<$
Replace	$<$	
Occur Check	$<$	

□

Proposition 3.6.5 *Soit P un problème d'unification tel qu'aucune règle de la Figure 3.3 s'applique à P. Alors soit P est $\bot$, soit P est de la forme $x_1 \stackrel{?}{=} t_1 \wedge \ldots \wedge x_n \stackrel{?}{=} t_n$ et les variables $x_1, \ldots, x_n$ sont éliminées en P.*

Proposition 3.6.6 *Soit P un problème d'unification de la forme $x_1 \stackrel{?}{=} t_1 \wedge \ldots \wedge x_n \stackrel{?}{=} t_n$, où les variables $x_1, \ldots, x_n$ sont éliminées en P. Alors la substitution $\mu := \{x_1 \rightarrow t_1; \ldots; x_n \rightarrow t_n\}$ est un mgu de P.*

Démonstration: Évidemment, μ est un unificateur de P. On peut montrer que, quand σ est un unificateur de P, alors $\sigma = \sigma \circ \mu$. □

Exercice 77
Compléter la preuve de la Proposition 3.6.6.

Théorème 3.6.1 *[Robinson] Il y a un algorithme qui, étant donné un problème d'unification P,*
– soit répond « non », et dans ce cas P n'a pas d'unificateur ;
– soit donne une substitution σ, et dans ce cas σ est un mgu de P.

Démonstration: Étant donné un problème d'unification P, on applique les règles de la Figure 3.3 dans un ordre quelconque. Par la Proposition 3.6.4, on obtient en temps fini un problème d'unification P' sur lequel aucune règle d'unification s'applique. Par la Proposition 3.6.3, P et P' ont les mêmes unificateurs. Par la Proposition 3.6.5 il y a deux possibilités :

1. P' est $\bot$. Par Proposition 3.6.3, P n'a pas d'unificateur.
2. P' est de la forme $x_1 \stackrel{?}{=} t_1 \wedge \ldots \wedge x_n \stackrel{?}{=} t_n$, et les variables $x_1, \ldots, x_n$ sont éliminées en P. Dans ce cas, $\sigma = \{x_1 \to t_1; \ldots, x_n \to t_n\}$ est un mgu de P', donc par Proposition 3.6.3 aussi de P. □

Exercice 78
Calculer un mgu pour les problèmes d'unification suivants :

1. $f(x, g(a, y)) \stackrel{?}{=} f(h(y), g(y, a)) \wedge g(x, h(y)) \stackrel{?}{=} g(z, z))$
2. $f(x, x) \stackrel{?}{=} f(g(y), z) \wedge h(z) \stackrel{?}{=} h(y)$

Si μ est un mgu de s et t, alors l'ensemble (3.2) est donc le même que

$$\{((L \vee K)\mu)\nu \mid \nu \in \Sigma_g\} \tag{3.3}$$

Le raisonnement pour la factorisation est analogue.

3.6.3 Résolution

On obtient donc les règles de résolution données sur la Figure 3.4. Dans les preuves par résolution, on n'écrit normalement pas les quantificateurs universelles. Pourtant, il ne faut pas oublier que chaque clause est implicitement universellement quantifiée. Une conséquence importante est qu'on doit toujours assurer, avant d'appliquer un pas de résolution à deux clauses, que leur ensembles de variables soient disjoints. Si ce n'est pas le cas on est obligé de renommer les variables dans une des clauses. Finalement, si S est un ensemble de clauses du premier ordre, on note $S \vdash_{R1} C$ quand la clause C est obtenue à partir de S par une application des règles de la Figure 3.4.

Lemme 3.6.1 *Les règles de la résolution du premier ordre sont correctes : Si $S \vdash_{R1} C$ alors C est une conséquence logique de S.*

Lemme 3.6.2 *(Lemme de relèvement)*
– Soient C_1 et C_2 des clauses universelles closes, $D_1 \in H(C_1)$, $D_2 \in H(C_2)$, et $\{D_1, D_2\} \vdash_R D_3$ par résolution binaire. Alors il existe une clause universelle close C_3 telle que $\{C_1, C_2\} \vdash_{R1} C_3$ par résolution binaire et $D_3 \in H(C_3)$.

$$\text{Résolution binaire} \quad \frac{\forall \bar{x}(\neg P(\bar{s}) \vee L) \quad \forall \bar{y}(P(\bar{t}) \vee K)}{\forall z((L \vee K)\sigma)} \qquad \begin{array}{l} \sigma \text{ est un mgu de } \bar{s} \stackrel{?}{=} \bar{t} \\ \bar{x} = \mathrm{Var}(\neg P(\bar{s}) \vee L) \\ \bar{y} = \mathrm{Var}(P(\bar{t}) \vee K) \\ \bar{x} \cap \bar{y} = \emptyset \\ \bar{z} = \mathrm{Var}((L \vee K)\sigma) \end{array}$$

$$\text{Factorisation} \quad \frac{\forall \bar{x}(P(\bar{s}) \vee P(\bar{t}) \vee L)}{\forall \bar{z}((P(\bar{t}) \vee L)\sigma)} \qquad \begin{array}{l} \sigma \text{ est un mgu de } \bar{s} \stackrel{?}{=} \bar{t} \\ \bar{x} = \mathrm{Var}(P(\bar{s}) \vee P(\bar{t}) \vee L) \\ \bar{z} = \mathrm{Var}((P(\bar{t}) \vee L)\sigma) \end{array}$$

FIG. 3.4 – Règles de résolution pour la logique du premier ordre.

– *Soit C_1 une clause universelle close, $D_1 \in H(C_1)$, et $\{D_1\} \vdash_R D_2$ par factorisation binaire. Alors il existe une clause universelle close C_2 telle que $\{C_1\} \vdash_{R1} C_2$ et $D_2 \in H(C_2)$.*

Démonstration: Soient
– $C_1 = \forall \bar{x}(P(\bar{s}) \vee L)$
– $C_2 = \forall \bar{y}(\neg P(\bar{t}) \vee K)$
– $\bar{x} \cap \bar{y} = \emptyset$
– $D_1 = (P(\bar{s}) \vee L)\sigma$
– $D_2 = (\neg P(\bar{t}) \vee K)\sigma$
– $\bar{s}\sigma = \bar{t}\sigma$
Résolution binaire sur $\{C_1, C_2\}$ nous donne une clause universelle $C_3 := \forall z(L \vee K)\mu$, où $\bar{z} \subseteq \bar{x} \cup \bar{y}$, et μ est un mgu de $\bar{s}$ et $\bar{t}$. Résolution binaire propositionnelle sur $\{D_1, D_2\}$ donne $D_3 := (L \vee K)\sigma$. Puisque σ est un unificateur de $\bar{s}$ et $\bar{t}$, et puisque μ est un mgu de $\bar{s}$ et $\bar{t}$, il y a une substitution ν telle que $\sigma = \nu \circ \mu$. Donc, $(L \vee K)\sigma = ((L \vee K)\mu)\nu$, donc $D_3 = (L \vee K)\sigma \in H(C_3)$.

La preuve pour la factorisation est analogue. □

Lemme 3.6.3 *Soit S un ensemble de clauses universelles closes. Si $H(S) \vdash_R^* D$, alors il existe une clause universelle close C telle que $S \vdash_{R1}^* C$ et $D \in H(C)$.*

Démonstration: Par induction sur la structure de la preuve $H(S) \vdash_R^* D$:
– si $D \in H(S)$ alors il existe un $C \in S$ tel que $D \in H(C)$.
– si le dernier pas de la preuve de D est une résolution binaire $\{D_1, D_2\} \vdash_R D$: Par hypothèse de récurrence, il existe des clauses universelles closes C_1, C_2 telles que $S \vdash_{R1}^* C_1$, $S \vdash_{R1}^* C_2$, $D_1 \in H(C_1)$ et $D_2 \in H(C_2)$. Par Lemme 3.6.2, il existe une clause universelle close C telle que $\{C_1, C_2\} \vdash_{R1} C$, donc $S \vdash_{R1}^* C$, et $D \in H(C)$.
– si le dernier pas de la preuve de D est une factorisation binaire : similaire. □

Théorème 3.6.2 *[Robinson] Soit S un ensemble de clauses universelles closes. S est non-satisfaisable si et seulement si $S \vdash_{R1}^* \square$.*

Démonstration: Par Lemme 3.6.1, si $S \vdash_{R1}^* \square$ alors S n'est pas satisfaisable.

Soit S non satisfaisable. Par Théorème 3.5.4, $H(S)$ n'est pas satisfaisable. Donc, par la complétude de la résolution propositionnelle, $H(S) \vdash_R^* \square$. Il suit, par Lemme 3.6.3, que $S \vdash_{R1}^* \square$ (remarquez que, si $\square \in H(C)$ alors $C = \square$). $\square$

Exemple 3.6.2 On considère l'ensemble des clauses universelles

(1) $P(x, x)$
(2) $\neg P(x, y) \vee P(x, s(y)))$
(3) $\neg P(f(x), s(s(f(x))))$

On en déduit par résolution

(4) $P(x, s(x))$ résolution binaire sur (1), (2)
(5) $P(x, (s(s(x)))$ résolution binaire sur (4), (2)
(6) $\square$ résolution binaire sur (5), (3)

Exercice 79
Soit l'ensemble de clauses universelles suivant :

(1) $\neg P(x) \vee Q(f(x), x)$
(2) $\neg P(x) \vee \neg Q(y, x) \vee R(y)$
(3) $\neg R(y) \vee \neg S(y) \vee \neg P(x) \vee \neg Q(y, x)$
(4) $\neg S(y) \vee R(y)$
(5) $P(a)$
(6) $\neg R(x) \vee S(x)$

En déduire par résolution $\square$.

Exercice 80
Soit S un ensemble fini de clauses universelles closes, et qui possède un plus petit modèle de Herbrand $\mathcal{H}$. Montrer que $\mathcal{H}$ est récursivement énumérable, c'est-à-dire que $P_{\mathcal{H}}$ est récursivement énumérable pour tous les symboles de prédicat P.

La *stratégie négative* consiste à restreindre l'application d'une résolution binaire au cas où une des deux prémisses ne contient que des littéraux négatives (l'application de la factorisation n'est pas restreinte). On note $S \vdash_{R1\neg}^* C$ quand une clause C peut être obtenue à partir d'un ensemble de clauses S par résolution binaire en stratégie négative.

Théorème 3.6.3 *La résolution binaire restreinte à la stratégie négative est réfutationnellement complète : Un ensemble S de clauses universelles closes est non satisfaisable si et seulement si $S \vdash_{R1\neg}^* \square$.*

Démonstration: Dans le cas de la logique propositionnelle, la stratégie négative est réfutationnellement complète (c'est une exercice dans le chapitre sur la résolution dans la logique propositionnelle). Puisque S est non-satisfaisable, il y a une preuve $H(S) \vdash_{R\neg}^* \square$. Relèvement (comme dans la preuve du lemme 3.6.3) donne une preuve par résolution du premier ordre qui suit la stratégie négative. $\square$

Exercice 81
Montrer, qu'en général la preuve la plus courte par résolution négative est exponentiellement plus longue que la preuve la plus courte par résolution :

Donner une famille $(S_n)_{n\geq 1}$ d'ensembles finis de clauses universelles, telle que toute réfutation de S_n par résolution négative a la longueur $2^n + c_1$ pour une constante c_1, mais qu'il existe une réfutation de S_n par résolution de longueur $n + c_2$ pour une constante c_2.