Cours de Logique*

Vers la Résolution : Mise en Forme Clausale

ENS Cachan, Informatique, L3

30 mars 2016

Nous traitons ici de la mise en forme clausale (et des résultats associés) en calcul des prédicats, dont les étapes sont :

— Mise en forme prénexe : plus de quantificateur sous les autres connecteurs logiques (Section 1).

— Mise en forme normale négative : plus d'implication, plus de connecteurs logiques sous les négations (Section 2).

— Skolémisation : élimination des quantificateurs (Section 3).

— Mise en forme clausale : plus de conjonction sous les disjonctions (Section 4).

Les clauses sont à la base d'un système de (recherche de) preuve par contradiction appellé *résolution*. Il est donc important de pouvoir ramener tout ensemble de formules closes à un ensemble de clauses. Nous présentons brièvement les règles de la preuve par résolution à la fin de ces notes (Section 5) sans en donner pour l'instant de preuve de complétude.

1 Mise en forme prénexe

Définition 1.1 (Forme prénexe) *Une formule est* en forme prénexe *si elle est de la forme*

$$Q_1 x_1 \ldots Q_n x_n.\ \varphi$$

où $Q_i \in \{\forall, \exists\}$ *pour tout* $i = 1, \ldots, n$ *et* φ *est sans quantificateurs.*

On considère les règles de transformation suivantes.

$$\begin{array}{rcll}
(Qx.\ \varphi) * \psi & \leadsto & Qx.\ (\varphi * \psi) & \text{si } x \notin \mathsf{VL}(\psi),\ \text{ et } * \in \{\wedge, \vee\} \\
\psi * (Qx.\ \varphi) & \leadsto & Qx.\ (\psi * \varphi) & \text{si } x \notin \mathsf{VL}(\psi),\ \text{ et } * \in \{\wedge, \vee, \Rightarrow\} \\
\neg\exists x.\ \varphi & \leadsto & \forall x.\ \neg\varphi & \\
\neg\forall x.\ \varphi & \leadsto & \exists x.\ \neg\varphi & \\
(\forall x.\ \varphi) \Rightarrow \psi & \leadsto & \exists x.\ (\varphi \Rightarrow \psi) & \text{si } x \notin \mathsf{VL}(\psi) \\
(\exists x.\ \varphi) \Rightarrow \psi & \leadsto & \forall x.\ (\varphi \Rightarrow \psi) & \text{si } x \notin \mathsf{VL}(\psi)
\end{array}$$

Exemple 1.1 On peut appliquer ces transformations de deux façons différentes à la formule ci-dessous.

*Ces notes de cours sont presque exactement reprises des notes du MOOC "Introduction à la logique informatique (partie 2)", réalisé par David Baelde, Hubert Comon et Étienne Lozes.

$$\big(\forall x.\ B(x)\big) \Rightarrow \big(\forall x.\ B(x)\big)$$

$$\equiv_\alpha$$

$$\big(\forall x.\ B(x)\big) \Rightarrow \big(\forall y.\ B(y)\big)$$

$$\exists x.\ \Big(B(x) \Rightarrow \big(\forall y.\ B(y)\big)\Big) \qquad \forall y.\ \Big(\big(\forall x.\ B(x)\big) \Rightarrow B(y)\Big)$$

$$\exists x. \forall y.\ \big(B(x) \Rightarrow B(y)\big) \qquad \forall y. \exists x.\ \big(B(x) \Rightarrow B(y)\big)$$

On étend la notion d'équivalence logique aux formules ayant des variables libres : deux formules φ, ψ sont logiquement équivalentes si pour toute structure $\mathcal{S}$, pour toute $\mathcal{S}$-affectation σ interprétant les variables libres de ces formules, $\mathcal{S}, \sigma \models \varphi$ ssi $\mathcal{S}, \sigma \models \psi$.

Lemme 1.1 (Correction des règles) *Si $\varphi \rightsquigarrow \psi$, alors φ et ψ sont logiquement équivalentes.*

Preuve:
Comme l'équivalence logique est une congruence, il suffit de prouver ce lemme lorsque la réécriture a lieu à la racine de la formule. On raisonne par analyse de cas. Considérons la règle

$$\underbrace{(\exists x.\ \varphi_1) \vee \varphi_2}_{=\phi} \ \rightsquigarrow\ \underbrace{\exists x.\ (\varphi_1 \vee \varphi_2)}_{=\psi}.$$

On veut montrer que pour tout $\mathcal{S}, \sigma$, on a $\mathcal{S}, \sigma \models \phi$ ssi $\mathcal{S}, \sigma \models \psi$. Soit $\mathcal{S}, \sigma$ fixés.

— Supposons que $\mathcal{S}, \sigma \models \varphi$. Alors $\mathcal{S}, \sigma \models \exists x.\ \varphi_1$ ou $\mathcal{S}, \sigma \models \varphi_2$.

Si	$\mathcal{S}, \sigma \models \exists x.\ \varphi_1$	(1^{er} cas)
alors	il existe $a \in D_{\mathcal{S}}$ tel que $\mathcal{S}, \sigma \uplus \{x \mapsto a\} \models \varphi_1$	(def. de $\models$)
donc	il existe $a \in D_{\mathcal{S}}$ tel que $\mathcal{S}, \sigma \uplus \{x \mapsto a\} \models \varphi_1 \vee \varphi_2$	(def. de $\models$)
donc	$\mathcal{S}, \sigma \models \exists x.\ (\varphi_1 \vee \varphi_2)$	(def. de $\models$)
Si	$\mathcal{S}, \sigma \models \varphi_2$	($2^{\text{ème}}$ cas)
alors	il existe $a \in D_{\mathcal{S}}. \mathcal{S}, \sigma \uplus \{x \mapsto a\} \models \varphi_2$	($D_{\mathcal{S}} \neq \emptyset$)
donc	il existe $a \in D_{\mathcal{S}}$ tel que $\mathcal{S}, \sigma \uplus \{x \mapsto a\} \models \varphi_1 \vee \varphi_2$	(def. de $\models$)
donc	$\mathcal{S}, \sigma \models \exists x.\ (\varphi_1 \vee \varphi_2)$	(def. de $\models$)

Dans les deux cas, $\mathcal{S}, \sigma \models \psi$.

— Supposons maintenant que $\mathcal{S}, \sigma \models \psi$. Alors il existe $a \in D_{\mathcal{S}}$ tel que $\mathcal{S}, \sigma \uplus \{x \mapsto a\} \models \varphi_1 \vee \varphi_2$ (par def de $\models$).

Si	$\mathcal{S}, \sigma \uplus \{x \mapsto a\} \models \varphi_1$	(premier cas)
alors	$\mathcal{S}, \sigma \models \exists x.\ \varphi_1$	(par def. de $\models$)
donc	$\mathcal{S}, \sigma \models (\exists x.\ \varphi_1) \vee \varphi_2$	(par def. de $\models$)
Si	$\mathcal{S}, \sigma \uplus \{x \mapsto a\} \models \varphi_2$	(deuxième cas)
alors	$\mathcal{S}, \sigma \models \varphi_2$	($x \notin \mathsf{VL}(\varphi_2)$)
donc	$\mathcal{S}, \sigma \models (\exists x.\ \varphi_1) \vee \varphi_2$	(par def. de $\models$)

Dans les deux cas, $\mathcal{S}, \sigma \models \varphi$.

Les autres règles se traitent de façon similaire, voire parfois plus simplement puisque sans utiliser l'hypothèse que $D_{\mathcal{S}} \neq \emptyset$. $\square$

Théorème 1.1 *Pour toute formule φ, on peut calculer une formule φ' logiquement équivalente à φ et en forme prénexe.*

2 Forme normale négative

Définition 2.1 (Littéral) *Un* littéral *est une formule de la forme $P(t_1, \dots t_n)$ ou sa négation.*

Définition 2.2 (Forme normale négative) *Une formule φ est en* forme normale négative *si*

1. *φ ne contient aucune implication, et*
2. *les seules sous-formules de φ de la forme $\neg\psi$ sont les littéraux.*

Exemple 2.1 La formule $\exists x.\big((\neg P(x)) \vee \exists y.Q(x,y)\big)$ est en forme normale négative, mais les formules $\exists x.\neg\big(P(x) \vee \exists y.Q(x,y)\big)$ et $\exists x.P(x) \Rightarrow \exists y.Q(x,y)$ ne le sont pas.

Proposition 2.1 *Pour toute formule φ, on peut calculer une formule φ' en forme normale négative logiquement équivalente à φ.*

Preuve:
On considère les règles de réécriture suivantes.

$$\begin{array}{rcl}
\varphi \Rightarrow \psi & \rightsquigarrow & \neg\varphi \vee \psi \\
\neg(\forall x.\ \varphi) & \rightsquigarrow & \exists x.\neg\varphi \\
\neg(\exists x.\ \varphi) & \rightsquigarrow & \forall x.\neg\varphi \\
\neg(\varphi \vee \psi) & \rightsquigarrow & (\neg\varphi) \wedge (\neg\psi) \\
\neg(\varphi \wedge \psi) & \rightsquigarrow & (\neg\varphi) \vee (\neg\psi) \\
\neg\neg\varphi & \rightsquigarrow & \varphi
\end{array}$$

On laisse en exercice la preuve que toute suite de réécritures $\varphi_0 \rightsquigarrow \varphi_1 \rightsquigarrow \dots$ est finie. On observe aisément que, si $\varphi \not\rightsquigarrow$, alors φ est en forme normale négative. Enfin, si $\varphi \rightsquigarrow \psi$, alors φ et ψ sont logiquement équivalentes.

Ainsi, si φ est une formule arbitraire et si φ' est obtenue à partir de φ par une suite maximale de réécritures, alors φ' est en forme normale négative et logiquement équivalente à φ. $\square$

3 Skolémisation

La partie la plus délicate de la mise en forme clausale est l'élimination des quantificateurs existentiels, au prix de l'introduction de nouveau symboles : c'est la skolémisation.

Naturellement, nous allons utiliser ici la possibilité de travailler sur des formules en forme normale négative — sans cela, la distinction entre quantificateurs existentiels et universels n'aurait de sens que si on comptait le nombre de négations (et d'implications) au dessus des quantificateurs.

3.1 Symboles de Skolem

On se donne un ensemble de symboles de fonctions $\mathcal{F}$ et un ensemble de symboles de prédicats $\mathcal{P}$. Pour chaque $(\mathcal{F},\mathcal{P})$-formule φ et pour chaque variable x, on se donne un nouveau symbole de fonction $f_{\varphi,x}$ d'arité $|\mathsf{VL}(\exists x.\varphi)|$. On note $\overline{\mathcal{F}}$ l'ensemble de symboles de fonction $\mathcal{F}$ augmenté de ces nouveaux symboles de fonction, autrement dit

$$\overline{\mathcal{F}} \;=\; \mathcal{F} \uplus \{f_{\varphi,x} \mid \varphi \text{ est une } (\mathcal{F},\mathcal{P})\text{-formule}, \ x \in X\}.$$

Enfin, pour tout φ, x, on se fixe une énumération $\vec{y}_{\varphi,x} = (y_1, \ldots, y_n)$ de $\mathsf{VL}(\exists x.\varphi)$.

Définition 3.1 (skolémisation) *Soit φ une $(\mathcal{F},\mathcal{P})$-formule. La skolémisée de φ, notée $\mathsf{Sk}(\varphi)$, est la $(\overline{\mathcal{F}},\mathcal{P})$-formule définie par récurrence sur la taille de φ comme suit.*

$$\begin{array}{lll}
\mathsf{Sk}(P(t_1,\ldots,t_n)) & = P(t_1,\ldots,t_n) & \\
\mathsf{Sk}(\neg\varphi) & = \neg\mathsf{Sk}(\varphi) & \\
\mathsf{Sk}(\varphi * \psi) & = \mathsf{Sk}(\varphi) * \mathsf{Sk}(\psi) & (* \in \{\vee, \wedge, \Rightarrow\}) \\
\mathsf{Sk}(\forall x.\varphi) & = \forall x.\mathsf{Sk}(\varphi) & \\
\mathsf{Sk}(\exists x.\varphi) & = \mathsf{Sk}(\varphi)\{x \mapsto f_{\varphi,x}(\vec{y}_{\varphi,x})\} &
\end{array}$$

En anticipant un peu sur la suite on remarquera que, même si $\mathsf{Sk}(\phi)$ est définie pour tout ϕ, il faudra supposer ϕ en forme normale négative pour obtenir le théorème de Skolem, qui dit que ϕ et $\mathsf{Sk}(\phi)$ sont équisatisfaisables.

Exemple 3.1 Soit φ la formule $\exists x \forall y \exists z.\ z * (x * y) \neq z$. On note c le symbole de Skolem d'arité 0 associé à x et $\forall y \exists z. z*(x*y) \neq z$, et f le symbole de Skolem d'arité 1 associé à z et $z*(x*y) \neq z$. Alors $\mathsf{Sk}(\varphi) = \forall y.\ f(c,y)*(c*y) \neq f(c,y)$.

3.2 Interprétation des symboles de Skolem

Il nous faut définir et étudier les deux foncteurs de structures suivants.

$$\begin{tikzcd}
\mathcal{S} \arrow[r, "\text{enrichissement}"] & \overline{\mathcal{S}} \\
\mathcal{T}_{|\mathcal{F}} & \mathcal{T} \arrow[l, "\text{restriction}"']
\end{tikzcd}$$

$(\mathcal{F},\mathcal{P})$-structures $\qquad\qquad$ $(\overline{\mathcal{F}},\mathcal{P})$-structures

Soit $\mathcal{S}$ une $(\mathcal{F}, \mathcal{P})$-structure. On veut construire à partir de $\mathcal{S}$ une $(\overline{\mathcal{F}}, \mathcal{P})$-structure $\overline{\mathcal{S}}$. Pour être tout à fait constructif, on a besoin de se donner une fonction $\mathsf{choix} : 2^{D_{\mathcal{S}}} \to D_{\mathcal{S}}$ telle que pour toute partie $T \neq \emptyset$ de $D_{\mathcal{S}}$, $\mathsf{choix}(T) \in T$. On peut supposer donnée une telle fonction en invoquant l'axiome du choix. De plus, comme $D_{\mathcal{S}} \neq \emptyset$, on peut se donner un élément $a_0 \in D_{\mathcal{S}}$ tel que $\mathsf{choix}(\emptyset) = a_0$.

Soit x, φ fixés, et soient $y_1, \ldots, y_n$ les variables libres de $\exists x.\varphi$. Pour tout $b_1, \ldots b_n \in D_{\mathcal{S}}$, on pose

$$T_{x,\varphi}(b_1, \ldots, b_n) := \left\{ a \in D_{\mathcal{S}} \mid \mathcal{S}, \{\vec{y} \mapsto \vec{b}, x \mapsto a\} \models \varphi \right\}$$

l'ensemble des témoins pour $\exists x.\varphi$ dans $\mathcal{S}$ avec l'affectation $\{\vec{y} \mapsto \vec{b}\}$.

On peut maintenant définir $\overline{\mathcal{S}}$:

- le domaine de $\overline{\mathcal{S}}$ est celui de $\mathcal{S}$, i.e. $D_{\overline{\mathcal{S}}} = D_{\mathcal{S}}$;
- $P_{\overline{\mathcal{S}}} = P_{\mathcal{S}}$ pour tout symbole de prédicat $P \in \mathcal{P}$
- $f_{\overline{\mathcal{S}}} = f_{\mathcal{S}}$ pour tout symbole de fonction "original" $f \in \mathcal{F}$
- enfin, pour $f \in \overline{\mathcal{F}} \setminus \mathcal{F}$ un symbole de Skolem associé à x et φ, on pose $f_{\overline{\mathcal{S}}}(\vec{b}) = \mathsf{choix}(T_{x,\varphi}(\vec{b}))$.

Lemme 3.1 *Soit $\mathcal{S}$ une $(\mathcal{F}, \mathcal{P})$-structure. Pour toute $(\mathcal{F}, \mathcal{P})$-formule φ et pour toute affectation σ telle que $Dom(\sigma) \supseteq \mathsf{VL}(\varphi)$,*

$$\mathcal{S}, \sigma \models \varphi \qquad \textit{ssi} \qquad \overline{\mathcal{S}}, \sigma \models \mathsf{Sk}(\varphi).$$

<u>Preuve:</u>
On raisonne par induction sur φ. Si φ est une formule atomique, c'est trivial. Si φ commence par un connecteur $\neg, \vee, \wedge, \Rightarrow$, c'est aussi trivial (pour la négation, noter que l'on montre une équivalence). Reste le cas des quantificateurs : le cas du $\forall$ est aussi trivial, mais ne fera pas de mal d'être détaillé pour s'en persuader, et le cas du $\exists$ utilise tout ce que l'on vient de définir.

- Supposons $\varphi = \forall x.\psi$. Alors

$$\begin{array}{lll}
 & \mathcal{S}, \sigma \models \varphi & \\
\text{ssi} & \text{pour tout } a \text{ dans } D_{\mathcal{S}},\ \mathcal{S}, \sigma \uplus \{x \mapsto a\} \models \psi & \text{par def. de } \models \\
\text{ssi} & \text{pour tout } a \text{ dans } D_{\mathcal{S}},\ \overline{\mathcal{S}}, \sigma \uplus \{x \mapsto a\} \models \mathsf{Sk}(\psi) & \text{par induction} \\
\text{ssi} & \overline{\mathcal{S}}, \sigma \models \forall x.\mathsf{Sk}(\psi) & \text{par def. de } \models \\
\text{ssi} & \overline{\mathcal{S}}, \sigma \models \mathsf{Sk}(\varphi) & \text{par def. de } \mathsf{Sk}(.)
\end{array}$$

- Supposons $\varphi = \exists x.\psi$, et notons $b_i = \sigma(y_i)$.

$$\begin{array}{lll}
 & \mathcal{S}, \sigma \models \varphi & \\
\text{ssi} & T_{x,\psi}(\vec{b}) \neq \emptyset & \text{par def. de } T_{x,\psi} \\
\text{ssi} & f_{\overline{\mathcal{S}}}(\vec{b}) \in T_{x,\psi}(\vec{b}) & \text{par def. de } f_{\overline{\mathcal{S}}} \\
\text{ssi} & \mathcal{S}, \sigma \uplus \{x \mapsto f_{\overline{\mathcal{S}}}(\vec{b})\} \models \psi & \text{par def. de } T_{x,\psi} \\
\text{ssi} & \overline{\mathcal{S}}, \sigma \uplus \{x \mapsto f_{\overline{\mathcal{S}}}(\vec{b})\} \models \mathsf{Sk}(\psi) & \text{par induction} \\
\text{ssi} & \overline{\mathcal{S}}, \sigma \models \big(\mathsf{Sk}(\psi)\big)\{x \mapsto f(\vec{y})\} & \text{par le lemme de substitution} \\
\text{ssi} & \overline{\mathcal{S}}, \sigma \models \mathsf{Sk}(\varphi) & \text{par def de } \mathsf{Sk}(.)
\end{array}$$

$\square$

3.3 Foncteur de restriction

Contrairement au foncteur d'enrichissement, le foncteur de restriction ne préserve la satisfaction que pour certaines formules. C'est le cas notamment pour les formules en forme normale négative.

Lemme 3.2 *Soit $\mathcal{S}'$ une $(\overline{\mathcal{F}}, \mathcal{P})$-structure. Pour toute $(\mathcal{F}, \mathcal{P})$-formule φ en forme normale négative, et pour toute affectation σ telle que $Dom(\sigma) \supseteq \mathsf{VL}(\varphi)$,*

$$si \qquad \mathcal{S}', \sigma \models \mathsf{Sk}(\varphi) \qquad alors \qquad \mathcal{S}'_{|\mathcal{F}}, \sigma \models \varphi.$$

<u>Preuve:</u>
On raisonne par induction sur φ. Au contraire du lemme précédent dans lequel on prouvait une équivalence, nous ne prouvons ici qu'une implication. Le cas de base est maintenant celui des littéraux, puisque nous avons supposé la formule en forme normale négative.

La récurrence est facile pour les connecteurs $\wedge, \vee$ et les quantifications universelles. Noter qu'elle ne marcherait pas pour des négations ou des implications, pour lesquelles nous aurions besoin d'une hypothèse d'induction plus forte.

A nouveau le seul cas intéressant est celui du $\exists$, mais on détaillera aussi celui du $\forall$ pour s'en convaincre. Soit φ fixée, et supposons $\mathcal{S}', \sigma \models \varphi$.

— Si $\varphi = \forall x.\psi$, alors

$$\begin{array}{lll} & \mathcal{S}', \sigma \models \forall x.\mathsf{Sk}(\psi) & \text{par def. de } \mathsf{Sk}(.) \\ \text{donc} & \text{pour tout } a \in D_{\mathcal{S}'},\ \mathcal{S}', \sigma \uplus \{x \mapsto a\} \models \mathsf{Sk}(\psi) & \text{par def. de } \models \\ \text{donc} & \text{pour tout } a \in D_{\mathcal{S}'},\ \mathcal{S}'_{|\mathcal{F}}, \sigma \uplus \{x \mapsto a\} \models \psi & \text{par induction} \\ \text{donc} & \mathcal{S}'_{|\mathcal{F}}, \sigma \models \forall x.\psi & \text{par def de } \models \end{array}$$

— Si $\varphi = \exists x.\psi$, et si $f = f_{\psi,x,\vec{y}}$, alors

$$\begin{array}{lll} & \mathcal{S}', \sigma \models \mathsf{Sk}(\psi)\{x \mapsto f(\vec{y}_{\varphi,x})\} & \text{par def. de } \mathsf{Sk}(.) \\ \text{donc} & \mathcal{S}', \sigma \uplus \{x \mapsto f_{\mathcal{S}'}(\vec{y}_{\varphi,x}\sigma)\} \models \mathsf{Sk}(\psi) & \text{par le lemme de substitution} \\ \text{donc} & \mathcal{S}'_{|\mathcal{F}}, \sigma \uplus \{x \mapsto f_{\mathcal{S}'}(\vec{y}_{\varphi,x}\sigma)\} \models \psi & \text{par induction} \\ \underline{\text{donc}} & \mathcal{S}'_{|\mathcal{F}}, \sigma \models \exists x.\psi & \text{par def de } \models \end{array}$$

□

On peut aussi montrer ce résultat en remplaçant l'hypothèse "en forme normale négative" par l'hypothèse "en forme prénexe", ce qui est souvent fait dans la littérature. Le point clé dans le lemme précédent est que les quantificateurs existentiels ne doivent pas apparaitre sous des négations, ce qui est vrai à la fois pour la forme normale négative et pour la forme prénexe.

Théorème 3.1 *Pour toute formule ϕ en forme normale négative, ϕ et $\mathsf{Sk}(\phi)$ sont équisatisfaisables.*

4 Mise en forme clausale

On rappelle qu'un littéral est une formule de la forme $P(t_1, \dots t_n)$ ou sa négation.

Définition 4.1 (Clause) *Une* clause *est une formule close (sans variables libres) en forme prénexe, universellement quantifiée, contenant une disjonction de littéraux,* i.e.*, une formule de la forme suivante (où $k, l, m \geq 0$) :*

$$\forall x_1, \dots, x_k.\ P_1(\vec{t_1}) \vee \dots \vee P_l(\vec{t_l}) \vee \neg P'_1(\vec{u}_1) \vee \dots \vee \neg P'_m(\vec{u}_m)$$

On pourra omettre d'écrire les quantificateurs universels quand il est clair que l'on fait référence à une clause.

Théorème 4.1 *Soit E un ensemble de formules closes. Il existe un ensemble $\overline{E}$ de clauses, fini et calculable lorsque E est fini et explicite, tel que E et $\overline{E}$ sont* équisatisfaisables *: E est satisfaisable ssi $\overline{E}$ l'est.*

<u>Preuve:</u>
Soit φ une formule close quelconque. On commence par la transformer en une formule φ_1 en forme normale négative. On pose ensuite $\varphi_2 = \mathsf{Sk}(\varphi_1)$, et on note $E_2 = \{\varphi_2 \mid \varphi \in E\}$. Si E est satisfaisable, et si $\mathcal{S}$ est un modèle de E, alors $\overline{\mathcal{S}}$ est un modèle de E_2 d'après le lemme 3.1. Réciproquement, si E_2 est satisfaisable, et si $\mathcal{T} \models E_2$, alors $\mathcal{T}_{|\mathcal{F}} \models E$ d'après le lemme 3.2. Donc E et E_2 sont équisatisfaisables.

Toute formule $\varphi_2 \in E_2$ ne contient que des quantificateurs universels, et est en forme normale négative. Donc, si φ_3 est une forme prénexe de φ_2, φ_3 est de la forme $\forall \overline{x}.\psi$ où ψ est sans quantificateur. En appliquant la distributivité de $\wedge$ sur $\vee$, on peut remplacer la formule φ_3 par une formule φ_4 de la forme

$$\forall \overline{x}.\ \bigwedge_{i=1}^{n} \bigvee_{j=1}^{m} \ell_{i,j}$$

où $\ell_{i,j}$ sont des littéraux. Enfin, on peut réécrire φ_4 en la formule $\varphi_5 =$

$$\bigwedge_{i=1}^{n} \forall \overline{x}.\ \bigvee_{j=1}^{m} \ell_{i,j}$$

qui est une conjonction de clauses. On note $\mathsf{cl}(\varphi_2)$ l'ensemble de clauses ainsi défini, et on pose $\overline{E} = \bigcup\{\mathsf{cl}(\varphi_2) \mid \varphi_2 \in E_2\}$. Comme $\{\varphi_2\}$, $\{\varphi_3\}$, $\{\varphi_4\}$ et $\mathsf{cl}(\varphi_2)$ sont logiquement équivalents, $\overline{E}$ est logiquement équivalent à E_2, donc équisatisfaisable avec E. $\square$

Exemple 4.1 Considérons l'ensemble de formules

$$E = \left\{ \begin{array}{ll} \exists e.\forall x. & x * e = x \wedge e * x = x, \\ \forall x, y, z. & (x * y) * z = x * (y * z), \\ \forall x.\exists y.\forall z. & (x * y) * z = z, \\ \exists x.\forall y.\exists z. & z * (x * y) \neq z \end{array} \right\}.$$

Alors E est satisfaisable ssi l'ensemble de clauses

$$\overline{E} = \left\{ \begin{array}{ll} \forall x. & x * \epsilon = x, \\ \forall x. & \epsilon * x = x, \\ \forall x, y, z. & (x * z) * t = x * (z * t), \\ \forall x. \forall z. & (x * I(x)) * z = z, \\ \forall y. & f(y) * (c * y) \neq f(y) \end{array} \right\}$$

est satisfaisable.

Dans l'exemple 4.1 ci-dessus, pour éliminer les quantificateurs existentiels, on a introduit de nouveaux symboles de fonction ϵ, I, c, et f qui représentent les "fonctions de choix" associées aux quantificateurs qui ont été éliminés. Ces fonctions de choix dépendent des variables quantifiées universellement qui précèdent le quantificateur existentiel éliminé. Ce procédé de transformation de formule est appellé *skolémisation* en l'honneur du mathématicien norvégien Thoralf Skolem.

5 Résolution

Les règles de résolution et factorisation (binaires) sont données en figure 1. Dans les preuves par résolution, on n'écrit normalement pas les quantificateurs universels. Cependant, 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 leurs ensembles de variables sont 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 $S \vdash_{R+F} C$ quand la clause C est obtenue à partir de S par application des règles de la Figure 1.

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

$$\text{Factorisation} \qquad \frac{\forall \bar{x}(P(\bar{s}) \vee P(\bar{t}) \vee L)}{\forall \bar{z}((P(\bar{t}) \vee L)\sigma)} \qquad \left\{ \begin{array}{l} \sigma = \mathsf{mgu}(\bar{s} \stackrel{?}{=} \bar{t}) \\ \bar{x} = \mathrm{Var}(P(\bar{s}) \vee P(\bar{t}) \vee L) \\ \bar{z} = \mathrm{Var}((P(\bar{t}) \vee L)\sigma) \end{array} \right.$$

FIGURE 1 – Règles de résolution pour la logique du premier ordre.

Lemme 5.1 *Les règles de la résolution du premier ordre sont correctes : Si* $S \vdash_{R+F} C$ *alors* C *est une conséquence logique de* S*.*

Il est aisé de voir que la résolution n'est pas complète. On montrera cependant que ce système a une propriété proche :

Théorème 5.1 *les règles de la résolution du premier ordre sont réfutationnellement complètes : Pour tout ensemble de clauses S insatisfaisable, on a $S \vdash_{R+F} \bot$.*