$$
\begin{array}{rcl}
\phi \rightarrow \psi & \Rightarrow & (\neg\phi) \vee \psi \\
\neg\neg\phi & \Rightarrow & \phi \\
\neg(\phi \wedge \psi) & \Rightarrow & (\neg\phi) \vee (\neg\psi) \\
\neg(\phi \vee \psi) & \Rightarrow & (\neg\phi) \wedge (\neg\psi) \\
(\phi \wedge \psi) \vee \theta & \Rightarrow & (\phi \vee \theta) \wedge (\psi \vee \theta) \\
\theta \vee (\phi \wedge \psi) & \Rightarrow & (\theta \vee \phi) \wedge (\theta \vee \psi) \\
\top \vee \phi & \Rightarrow & \top \\
\phi \vee \top & \Rightarrow & \top \\
\phi \wedge \top & \Rightarrow & \phi \\
\top \wedge \phi & \Rightarrow & \phi \\
\bot \vee \phi & \Rightarrow & \phi \\
\phi \vee \bot & \Rightarrow & \phi \\
\phi \wedge \bot & \Rightarrow & \bot \\
\bot \wedge \phi & \Rightarrow & \bot \\
\neg\top & \Rightarrow & \bot \\
\neg\bot & \Rightarrow & \top
\end{array}
$$

FIGURE 2.2 – Règles de mise en forme clausale

2.3 Forme clausale

On considère les règles de simplification de la figure 2.2.

Dans les règles de la figure 2.2, ϕ, ψ, θ sont des *variables logiques* : elles peuvent être remplacées par n'importe quelle formule de $\mathcal{F}_0(\mathcal{P})$.

De plus, les règles peuvent être appliquées dans n'importe quel contexte. Plus formellement, un *contexte* est une formule de $\mathcal{F}_0(\mathcal{P} \cup \{\Box\})$ où $\Box \notin \mathcal{P}$, qui ne comporte qu'une seule occurrence de $\Box$. Si C est un contexte et $\phi \in \mathcal{F}_0(\mathcal{P})$, $C[\phi]$ est la formule de $\mathcal{F}_0(\mathcal{P})$ obtenue en repmplaçant $\Box$ dans C par ϕ. La relation $\Rightarrow$ est alors la (plus petite) relation binaire sur $\mathcal{F}_0(\mathcal{P})$ qui contient toutes les instances des règles de la figure 2.2 et telle que, pour tout contexte C, si $\phi \Rightarrow \psi$, alors $C[\phi] \Rightarrow C[\psi]$.

Proposition 2.3.1 *Les règles de la figure 2.2 transforment des formules en des formules logiquement équivalentes.*

Proposition 2.3.2 *Les règles de la figure 2.2 se terminent (quel que soit l'ordre dans lequel elles sont appliquées) : il n'existe aucune suite infinie $\phi_n, n \in \mathbb{N}$ de formules de $\mathcal{F}_0(\mathcal{P})$ telle que, pour tout i, $\phi_i \Rightarrow \phi_{i+1}$.*

Ce résultat reste vrai si les règles sont appliquées modulo l'associativité-commutativité de $\wedge, \vee$: si une (instance de) règle s'applique à ϕ et ψ est identique à ϕ modulo l'associativité-commutativité de $\wedge, \vee$, elle s'applique à ψ, avec le même résultat.

Preuve:
On interprète comme suit les formules dans les entiers :

— $f(\bot) = f(\top) \stackrel{\text{def}}{=} 2$

— $f(P) \stackrel{\text{def}}{=} 2$ si P est une variable propositionnelle.
— $f(\phi \wedge \psi) \stackrel{\text{def}}{=} g_1(f(\phi), f(\psi))$ où $g_1(x,y) \stackrel{\text{def}}{=} x+y+1$
— $f(\phi \vee \psi) \stackrel{\text{def}}{=} g_2(f(\phi), f(\psi))$ où $g_2(x,y) \stackrel{\text{def}}{=} x \times y$
— $f(\neg\phi) \stackrel{\text{def}}{=} g_3(f(\phi))$ où $g_3(x) \stackrel{\text{def}}{=} 2^x$
— $f(\phi \to \psi) \stackrel{\text{def}}{=} g_4(f(\phi), f(\psi))$ où $g_4(x,y) \stackrel{\text{def}}{=} 2^{1+x+y}$

L'interprétation des formules est ensuite compatible avec l'associativité-commutativité de $\wedge, \vee$: $f(\phi \wedge \psi) = f(\psi \wedge \phi)$, $f(\psi \vee \phi) = f(\phi \vee \psi)$ et $f(\phi \wedge (\psi \wedge \theta)) = f((\phi \wedge \psi) \wedge \theta)$, $f(\phi \vee (\psi \vee \theta)) = f((\phi \vee \psi) \vee \theta)$. f est donc bien définie, indépendemment du représentant choisi, dans la classe d'équivalence modulo associativité et commutativité.

On montre d'abord par récurrence sur ϕ que $f(\phi) \geq 2$.

Ensuite, toutes les fonctions g_i sont strictement croissantes dans leurs deux arguments pour $x, y \geq 2$. Il en résulte que, si $f(\phi) > f(\psi)$, alors $f(C[\phi]) > f(C[\psi])$ pour tout contexte C.

Il suffit ainsi de démontrer que, pour toutes les formules ϕ, ψ, chacune des règles fait décroître f. Pour plus de clarté, on notera $x = f(\phi)$, $y = f(\psi)$, $z = f(\theta)$ dans ce qui suit.

— $f(\phi \to \psi) = 2^{1+x+y}$ et $f(\neg\phi \vee \psi) = 2^x \times y$. Mais, pour $y \geq 0$, $2^y > y$ donc $2^{1+x+y} > 2^x \times y$.
— $f(\neg\neg\phi) = 2^{2^x} > x$
— $f(\neg(\phi \wedge \psi)) = 2^{x+y+1} > f(\neg\phi \vee \neg\psi) = 2^x \times 2^y$
— $f(\neg(\phi \vee \psi)) = 2^{x \times y}$ et $f(\neg\phi \wedge \neg\psi) = 2^x + 2^y + 1$. Or, pour $x, y \geq 2$, $x \times y \geq x + y$, donc $2^{x \times y} \geq 2^x \times 2^y \geq 2^x + 2^y \times (2^x - 1)$. Mais $2^x - 1 > 2$ pour $x \geq 2$, donc $2^{x \times y} > 2^x + 2 \times 2^y > 2^x + 2^y + 1$.
— $f((\phi \wedge \psi) \vee \theta) = f(\theta \vee (\phi \wedge \psi)) = (x+y+1) \times z$ et $f((\phi \vee \theta) \wedge (\psi \vee \theta)) = x \times z + y \times z + 1$ qui est strictement inférieur à $x \times z + y \times z + z$ pour $z \geq 2$.
— Les autre cas sont immédiats.

□

Une *forme irréductible* d'une formule ϕ est une formule ψ telle que $\phi \Rightarrow \cdots \Rightarrow \psi$ et aucune règle ne s'applique à ψ.

Definition 2.3.1 *Une formule est en* forme normale conjonctive *si elle est irréductible pour les règles de la figure 2.2.*

Definition 2.3.2 *Un* littéral *est une variable propositionnelle ou la négation d'une variable propositionnelle.*

Une clause *est une disjonction de littéraux ou bien $\perp$.*

Proposition 2.3.3 *Toute formule en forme normale conjonctive est une conjonctions de clauses ou bien $\top$.*

Definition 2.3.3 *Une* forme clausale *d'une formule $\phi \in \mathcal{F}_0(\mathcal{P})$ est une formule en forme normale conjonctive, logiquement équivalente à ϕ.*

Exercice 21 (3)
Donner un exemple de formule qui se réduit par le système de réécriture de la figure 2.2 en deux formes clausales distinctes.

L'exercice suivant montre que les formes clausales peuvent inévitablement conduire à une croissance exponentielle de la formule.

Exercice 22 (6)
Si $\phi \in \mathcal{F}_0(\mathcal{P})$, on note $\tau(\phi)$ la taille minimale d'une forme clausale de ϕ.

1. Donner un exemple d'une famille de formules ϕ_n telles que $\lim_{n\to+\infty} |\phi_n| = +\infty$ et $\lim_{n\to+\infty} \frac{\tau(\phi_n)}{\sqrt{2^{|\phi_n|}}} > 0$.
2. Montrer que, pour toute formule ϕ, $\tau(\phi) < |\phi| \times 2^{\frac{|\phi|+5}{2}}$

$$\text{Résolution binaire} \quad \frac{\neg A \vee C \quad A \vee C'}{C \vee C'}$$

$$\text{Factorisation binaire} \quad \frac{L \vee L \vee C}{L \vee C}$$

FIGURE 2.3 – Règle de résolution

2.4 Résolution

Dans cette partie, on ne considère que des formes clausales.

Décider de la satisfaisabilité d'une formule en forme clausale n'est pas (algorithmiquement) facile. C'est un problème NP-complet (voir calculabilité).

Les règles d'inférence de la figure 2.3 ont en prémisse une ou deux clauses et en conclusion une clause. Dans ces règles, C est ou bien une disjonction de littéraux, ou bien la clause vide $\perp$ (on suppose que $C \vee \perp = C$) et L est un littéral.

On note $E \vdash_R C$ lorsque la clause C peut être obtenue à partir de E. par une application d'un nombre quelconque de règles de la figure 2.3. De manière équivalente, $E \vdash C$ s'il existe un arbre dont les noeuds sont étiquetés par des clauses, la racine est étiquetée par C, les étiquettes des feuilles sont dans E et, chaque noeud qui n'est pas une feuille

— ou bien a un seul fils et, dans ce cas, son étiquette est obtenue par factorisation à partir de l'étiquette de son fils

— ou bien a deux fils et, dans ce cas, son étiquette est obtenue par résolution binaire à partir des étiquettes de ses deux fils.

La *taille* d'une preuve est le nombre de noeuds de l'arbre correspondant.

Exemple 2.4.1 Si $E = \{P \vee \neg Q \vee R, P \vee \neg R\}$ alors $E \vdash_R P \vee \neg Q$ et voici un arbre de preuve :

$$\dfrac{\dfrac{P \vee \neg Q \vee R \quad P \vee \neg R}{P \vee P \vee \neg Q}\,R}{P \vee \neg Q}\,F$$

Exercice 23 (2)
Montrer comment les règles de la figure 2.3 permettent de dériver la clause vide $\perp$ de l'ensemble de clauses

$$E = \{P \vee \neg Q, \neg P \vee \neg Q, P \vee Q, \neg P \vee Q\}$$

Exercice 24 (2)
Montrer qu'une même clause peut avoir plusieurs arbres de preuve distincts (à permutation près des fils).

Proposition 2.4.1 *Les règles de la figure 2.3 sont correctes.* ($\vdash_R \subseteq \models$).

Preuve:
Il suffit de montrer que, pour chacune des deux règles, lorsqu'une interprétation satisfait les prémisses, elle satisfait aussi la conclusion, puis de faire une récurrence sur la longueur de la preuve (taille de l'arbre de preuve). Les détails sont laissés en exercice. $\square$

Pour démontrer le théorème qui suit, on utilisera les *arbres sémantiques* que nous définissons maintenant formellement (après les avoir utilisés dans la preuve du théorème 2.2.1).

On suppose, ainsi que dans toute la suite, que l'ensemble des variables propositionnelles est dénombrable : $\mathcal{P} = \{P_i, i \in \mathbb{N}\}$. Une *interprétation partielle* est une application de $\{P_1, \ldots, P_n\}$ dans $\{0, 1\}$. $\{P_1, \ldots, P_n\}$ est alors le domaine de l'interprétation partielle. Les interprétations partielles sont ordonnées par prolongement : $I \leq J$ si $Dom(I) \subseteq Dom(J)$ et $\forall x \in Dom(I), I(x) = J(x)$. Si $Dom(I) \neq \mathcal{P}$, il existe exactement deux interprétations partielles I_0 et I_1 telles que $I_0, I_1 > I$ et $\forall J, J > I \Rightarrow J \geq I_0$ ou $J \geq I_1$. I_0 et I_1 sont les *successeurs* de I. Si $Dom(I) = \{P_1, \ldots, P_n\}$, I_0 est l'interprétation qui prolonge I par $I_0(P_{n+1}) = 0$. C'est le *fils droit* de I. I_1 prolonge I par $I_1(P_{n+1}) = 1$. C'est le *fils gauche* de I.

Une interprétation partielle I *falsifie* une clause C si toutes les variables propositionnelles de C sont dans le domaine de I et $I \not\models C$.

Si E est un ensemble de clauses, l'*arbre sémantique* $A(E)$ est défini comme suit. On confond les chemins finis de l'arbre, les noeuds de l'arbre et les interprétations partielles qui correspondent. On précise ci-dessous la correspondance.

- La racine correspond au chemin vide, c'est à dire à l'interprétation de domaine vide.
- Si N est un noeud de l'arbre correspondant à l'interprétation I de domaine $\{P_1, \ldots, P_n\}$,
  - Ou bien il existe une clause $C \in E$ telle que I falsifie C et N est un *noeud d'échec*. C'est alors une feuille de l'arbre et on l'étiquette par une clause de E falsifiée par I
  - Ou bien on n'est pas dans le premier cas et I est une interprétation totale sur $Var(t)$. Dans ce cas N est une feuille de l'arbre. C'est un *noeud de succès*.
  - Dans les autres cas, N a deux fils qui correspondent aux deux successeurs de I.

Exemple 2.4.2 Soit $E = \{P_1, \neg P_2 \vee \neg P_1, P_3 \vee \neg P_1\}$. L'arbre sémantique de E est représenté dans la figure 2.4. L'arbre comporte un noeud de succès (et un seul) qui correspond à la seule interprétation qui satisfait E.

Lemme 2.4.1 *Si E est un ensemble de clauses, alors E est satisfaisable si et seulement si ou bien $A(E)$ contient un noeud de succès, ou bien $A(E)$ contient un chemin infini.*

Preuve:

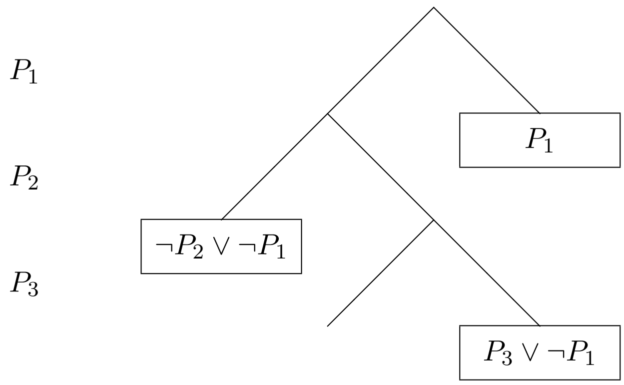

FIGURE 2.4 – Arbre sémantique de l'ensemble E de l'exemple 2.4.2

Si E est satisfaisable, alors l'interprétation I qui satisfait E définit ou bien un chemin conduisant à un noeud de succès (cas où $\mathcal{P}$ est fini) ou bien un chemin infini de $A(E)$ (cas où $\mathcal{P}$ est infini).

Réciproquement, si $A(E)$ contien un noeud de succès, alors l'interprétation I qui lui correspond est totale et ne falsifie aucune clause de E et donc $I \models E$. Si $A(E)$ contient un chemin infini, il existe une suite infinie d'interprétations partielles I_n (les noeuds de ce chemin) telles que $Dom(I_n) = \{P_1, \ldots, P_n\}$, $I_n < I_{n+1}$ et, pour tout n, I_n ne falsifie aucune clause de E. On définit alors I par $I(P_i) = I_i(P_i)$ pour tout i. Pour toute clause $C \in E$, si n_C est l'indice maximal d'une variable propositionnelle de C, $I_{n_C} \models C$ par construction, et, puisque I_{n_C} prolonge I_k pour $k \leq n_C$, I et I_C coïncident sur le domaine de I_{n_C}. Il en résulte que $I \models C$. $\square$

Exercice 25 (2)
Construire l'arbre sémantique associé à $E = \{\neg P_2, P_1 \vee P_2 \vee P_3, P_1 \vee \neg P_3, \neg P_1 \vee P_2 \vee \neg P_3, \neg P_1 \vee P_3\}$ E est-il satisfaisable ? Pourquoi ?

Théorème 2.4.1 (Complétude réfutationnelle) *Un ensemble de clauses E est insatisfaisable si et seulement si $E \vdash_R \bot$.*

Preuve:
Si $E \vdash_R \bot$, alors E est insatisfaisable par le résultat de correction.

Supposons maintenant que E est insatisfaisable et montrons que $E \vdash_R \bot$.

Soit E^* l'ensemble des clauses C telles que $E \vdash_R C$. Comme $E \subseteq E^*$ est insatisfaisable, il en est de même pour E^*. Donc, d'après le lemme 2.4.1, $A(E^*)$ ne contient ni noeud de succès, ni chemin infini. Raisonnons alors par l'absurde et supposons que $A(E^*)$ n'est pas réduit à la racine. Soit alors N un noeud maximal ayant deux fils. Ces deux fils sont des feuilles, par maximalité, donc des noeuds d'échec. Soit I l'interprétation correspondant à N et I_0, I_1 ses deux successeurs, obtenus en interprétant la variable $P \notin Dom(I)$. Il existe des clauses de E^* qui sont falsifiées respectivement par I_0 et I_1. Soient $C_0, C_1 \in E^*$

de taille minimale qui sont falsifiées respectivement par I_0 et I_1. Comme I ne falsifie ni C_0 ni C_1, $C_0 = P \vee C_0'$ et $C_1 = \neg P \vee C_1'$. De plus si $C_0' = P \vee C_0''$, alors, par factorisation, $C_0 \vdash_R P \vee C_0''$ et I_0 falsifie $P \vee C_0''$, ce qui contredit la minimalité de C_0. De plus C_0' ne peut s'écrire $\neg P \vee C_0''$, car I_0 falsifie C_0. Il en résulte que $Var(C_0') \subseteq Dom(I)$. De même, $Var(C_1') \subseteq Dom(I)$. Par résolution, $C_0, C_1 \vdash_R C_0' \vee C_1'$, donc $C_0' \vee C_1' \in E^*$. De plus I_0 falsifie C_0', donc I falsifie C_0' et, de même, I_1 falsifie C_1', donc I falsifie C_1'. Il en résulte que I falsifie $C_0' \vee C_1'$ et donc que N est un noeud d'échec. Absurde.

On en conclut que l'arbre sémantique de E^* est réduit à la racine : la racine est un noeud d'échec, ce qui ne peut se produire que si $\bot \in E^*$, c'est à dire $E \vdash_R \bot$. □

Le résultat suivant montre que l'on peut obtenir le théorème de compacité comme corollaire au théorème de complétude.

Corollaire 2.4.1 *Si E est un ensemble de clauses insatisfaisables, alors E contient un sous-ensemble fini de clauses insatisfaisable.*

Preuve:
Si E est insatisfaisable, alors $E \vdash_R \bot$. Or la preuve correspondante est un arbre fini. Si E_0 est le sous-ensemble des clauses qui étiquettent des feuilles de l'arbre, $E_0 \subseteq E$, E_0 est fini et E_0 est insatisfaisable. □

La *longueur* d'une preuve est le nombre de sous-arbres distincts dans cette preuve.

Une preuve Π est *sans boucle* si, pour toute clause C, tout sous-arbre de Π dont la racine est étiquetée par C ne contient pas lui-même de sous-arbre propre dont la racine est étiquetée par C.

Exercice 26 (6)
Soit $\mathcal{P}$ un ensemble fini de variables propositionnelles, de cardinal n. On appelle 2-clause toute clause contenant au plus deux littéraux.

1. Montrer que, si E est ensemble de 2-clauses insatisfaisable, toute preuve sans boucle de $\bot$ est de longueur polynômiale en n
2. Montrer par contre que la taille peut être exponentielle : donner un exemple d'ensemble de 2-clauses E tel que E est insatisfaisable et une preuve sans boucle de $\bot$ à partir de E dont la taille est exponentielle en n.
3. Donner un algorithme polynômial en n pour décider de la satisfaisabilité d'un ensemble de 2-clauses

Exercice 27 (6)
Soit $\mathcal{P}$ un ensemble de variables propositionnelles fini. Donner un ensemble de clauses E qui est insatisfaisable et tel qu'il existe une preuve de $\bot$ de taille exponentielle en $|E|$ et ne contenant aucune redondance (i.e. deux noeuds distincts de l'arbre de preuve qui ne sont pas des feuilles sont étiquetés par des formules distinctes).

$$\text{TE} \quad \frac{}{P \vee \neg P}$$

$$\text{A} \quad \frac{C}{C \vee L}$$

FIGURE 2.5 – Tiers exclu et affaiblissement

En fait, on peut généraliser ce résultat (mais il n'est pas demandé de le montrer) : il existe des ensembles de clauses insatisfaisables dont *aucune* preuve de contradiction n'est de taille polynômiale.

On considère maintenant, en plus des règles d'inférence de la figure 2.3, les deux règles d'inférence de la figure 2.5. Dans ces règles, C est une clause quelconque, L est un littéral quelconque et P est une variable propositionnelle quelconque. On note $\vdash$ la relation de déduction associée.

Théorème 2.4.2 (Complétude) *Si E est un ensemble de clauses et C est une clause. Alors $E \models C$, si et seulement si $E \vdash C$. (Autrement dit $\models = \vdash$).*

Preuve:
La correction des règles d'inférence de la figure 2.5 est une vérification de routine. Il en résulte, par récurrence sur la longueur de la preuve que $E \vdash C$ entraine $E \models C$.

Réciproquement, montrons, par récurrence sur la longueur de la preuve par résolution que, pour tout ensemble de clauses E, tous littéraux $L_1, \ldots, L_k$ et toute clause C,

$$E, L_1, \ldots, L_k \vdash C$$

entraine

$$E \vdash C \vee \overline{L_1} \vee \ldots \vee \overline{L_k}$$

— Si la preuve ne contient aucune application de règle d'inférence : ou bien il existe un i tel que $C = L_i$ ou bien $C \in E$. dans le premier cas, par la règle du tiers exclu $\vdash L_i \vee \overline{L_i}$, puis, par affaiblissements, $\vdash L_i \vee \overline{L_1} \vee \ldots \vee \overline{L_k}$. Dans le deuxième cas, $E \vdash C$ et, par affaiblissements, $E \vdash C \vee \overline{L_1} \vee \ldots \vee \overline{L_k}$.
— Sinon, au moins une règle d'inférence est appliquée. Considérons la dernière règle appliquée dans la preuve.
  — Si c'est le tiers exclu, $E, L_1, \ldots, L_k \vdash P \vee \neg P$, et on obtient une preuve de $E \vdash \overline{L_1} \vee \ldots \vee \overline{L_k} \vee P \vee \neg P$:

$$\dfrac{\dfrac{}{P \vee \neg P}\, TE}{\overline{L_1} \vee \ldots \vee \overline{L_k} \vee P \vee \neg P}\, Aff$$

  — Si c'est un affaiblissement : $E, L_1, \ldots, L_k \vdash C_1$ et $C = C_1 \vee L$. Par hypothèse de récurrence, $E \vdash C_1 \vee \overline{L_1} \ldots \vee \overline{L_k}$. Par affaiblissement, $E \vdash C_1 \vee \overline{L_1} \ldots \vee \overline{L_k} \vee L$.

— Si c'est une factorisation : $C = L \vee C_1$ et $E, L_1, \ldots, L_k \vdash L \vee L \vee C_1$. Par hypothèse de récurrence,

$$E \vdash L \vee L \vee C_1 \vee \overline{L_1} \vee \ldots \vee \overline{L_k}$$

et, par factorisation,

$$L \vee L \vee C_1 \vee \overline{L_1} \vee \ldots \vee \overline{L_k} \vdash C \vee \overline{L_1} \vee \ldots \vee \overline{L_k}.$$

— Si c'est une résolution : $C = C_1 \vee C_2$ et $E, L_1, \ldots, L_k \vdash C_1 \vee P$, $E, L_1, \ldots, L_k \vdash C_2 \vee \neg P$. Par hypothèse de récurrence (appliquée deux fois) :

$$E \vdash C_1 \vee P \vee \overline{L_1} \vee \ldots \vee \overline{L_k}$$

et

$$E \vdash C_2 \vee \neg P \vee \overline{L_1} \vee \ldots \vee \overline{L_k}$$

Par résolution, on obtient alors

$$E \vdash C_1 \vee C_2 \vee \overline{L_1} \vee \ldots \vee \overline{L_k} \vee \overline{L_1} \vee \ldots \vee \overline{L_k}$$

puis, par factorisations, $E \vdash C \vee \overline{L_1} \vee \ldots \vee \overline{L_k}$.

On applique maintenant ce résultat avec $C = \perp$: pour une clause $C_0 = \overline{L_1} \vee \ldots \vee \overline{L_k}$ arbitraire, $E, L_1, \ldots, L_k \vdash \perp$ entraine $E \vdash C_0$.

Si maintenant $E \models C_0$, alors $E, L_1, \ldots, L_k$ est insatisfaisable et, par le théorème 2.4.1, $E, L_1, \ldots, L_k \vdash_R \perp$ et donc $E, L_1, \ldots, L_k \vdash \perp$.

D'après ce que nous venons de voir, cela entraine $E \vdash C_0$. □

Exercice 28 (3)

On considère ici un raffinement de la résolution. On définit un ordre sur les littéraux de la manière suivante : $L > L'$ si la variable propositionnelle de L a un indice strictement plus grand que la variable propositionnelle de L'. (autrement dit, on étend l'ordre sur les variables propositionnelles aux littéraux). On restreint alors l'application de la résolution binaire

$$\frac{P \vee C \qquad \neg P \vee C'}{C \vee C'}$$

au cas où P est un littéral maximal de $P \vee C$ et $\neg P$ est un littéral maximal de $\neg P \vee C'$. De même, la règle de factorisation

$$\frac{L \vee L \vee C}{L \vee C}$$

est restreinte au cas où L est maximal dans $L \vee C$.

Montrer qu'avec ces restrictions, l'ensemble de règles d'inférence est encore réfutationnellement complet.

Exercice 29 (3)
On considère le système d'inférence constitué de l'unique règle :

$$\frac{P \vee P \ldots \vee P \vee C \qquad \neg P \vee \ldots \vee \neg P \vee C'}{C \vee C'}$$

Montrer que cette règle est (à elle seule) réfutationellement complète.

Exercice 30 (3)
On dira qu'une clause C *subsume* une clause C' si $C \models C'$.

On considère la stratégie de résolution+factorisation suivante : on n'applique une règle d'inférence que lorsqu'aucune des prémisses n'est subsumée par une clause différente ancêtre dans l'arbre de preuve.

Montrer que cette stratégie est réfutationnellement complète.

Definition 2.4.1 *On appelle* clause de Horn *une clause qui contient au plus un littéral positif.*

Exercice 31 (6)
On considère ici un autre raffinement de la résolution : la règle de résolution est restreinte au cas où l'une des prémisse au moins est dans E (l'ensemble de clauses initial). Cette stratégie est dite *input*.

1. Montrer que cette stratégie n'est pas réfutationnellement complète en général.
2. Montrer qu'elle est réfutationnellement complète lorsque E est un ensemble de clauses de Horn

Exercice 32 (5)
Donner un algorithme polynômial qui permet, étant donné un ensemble fini de clauses de Horn, de dire s'il est satisfaisable ou non.

Exercice 33 (6)
Une clause est *négative* si elle ne contient que des littéraux négatifs. On se propose d'étudier la stratégie suivante de résolution, appelée *stratégie négative* : l'application de la règle de résolution est restreinte au cas où l'une des prémisse est négative. On notera $\vdash_\neg$ la relation de déduction associée.

1. Soit E un ensemble de clauses tel que toute clause de E contient au moins un littéral positif. Montrer que E est satisfaisable.
2. Soit $E = \{\neg P \vee Q, P \vee Q, P \vee \neg Q, \neg P \vee \neg Q\}$. Montrer que $E \vdash_\neg \bot$ (exhiber une preuve).
3. Si I, J sont des interprétations partielles, on note $I >_{lex} J$ lorsqu'il existe un entier $k \geq 1$ tel que
   (a) pour tout $j < k$, P_j est dans le domaine de I et dans le domaine de J et $I(P_j) = J(P_j)$
   (b) P_k est dans le domaine de I et dans le domaine de J et $I(P_k) = 1$ et $J(P_k) = 0$.

On note aussi $I \leq J$ l'ordre de prolongement des interprétations partielles.

Montrer que $\geq_{lex}$ est un ordre sur les interprétations partielles et que, pour toutes interprétations partielles, $I \leq J$ ou $J \leq I$ ou $I \leq_{lex} J$ ou $J \leq_{lex} I$

4. Soit A l'arbre sémantique d'un ensemble E de clauses. On suppose que A est fini, non vide et que toutes ses feuilles sont des noeuds d'échec. Montrer qu'il existe une unique interprétation partielle maximale pour $\geq_{lex}$ qui soit un noeud d'échec et ne falsifie aucune clause négative de E.
5. Montrer la complétude réfutationnelle de $\vdash_{\neg}$ par la méthode des arbres sémantiques

2.5 Stratégies de sélection

Une *fonction de sélection* est une application qui associe à toute clause $L_1 \vee \cdots \vee L_n$ (où $n \geq 1$) l'un des littéraux L_i. Étant donnée une fonction de sélection f, on considère la restriction suivante de la résolution :

$$(R_f) \qquad \frac{P \vee C \quad \neg P \vee C'}{C \vee C'} \qquad \text{Si } f(P \vee C) = P \ ET \ f(\neg P \vee C') = \neg P$$

La règle de factorisation binaire et la règle R_f définissent une relation de déduction $\vdash_f$ pour le calcul propositionnel en forme clausale.

Exercice 34 (5)

1. Montrer que $F + R_f$ n'est pas réfutationnellement complète : donner un exemple de fonction de sélection f et d'un ensemble de clauses $\mathcal{E}$ insatisfaisable tel que $\mathcal{E} \not\vdash_f \bot$.
2. Qu'en est il si on suppose que la fonction de sélection sélectionne toujours un littéral négatif quand c'est possible ?

Une clause *de Horn* est une clause contenant au plus un littéral positif.

Théorème 2.5.1 *Pour toute fonction de sélection f, R_f est réfutationellement compète pour les clauses de Horn.*

Exercice 35 (7)
Montrer le théorème ci-dessus, d'abord dans le cas où f sélectionne toujours un littéral négatif quand il y en a au moins un, puis dans le cas général.

Exercice 36 (3)
Soit $\mathcal{E}$ un ensemble de clauses insatisfaisable quelconque (pas nécessairement des clauses de Horn). Montrer qu'il existe une fonction de sélection telle que la résolution binaire avec stratégie de sélection + factorisation binaire permet de déduire la clause vide de $\mathcal{E}$.

Exercice 37
Soit $\mathcal{P}$ un ensemble dénombrable de variables propositionnelles. On notera $\sqsubseteq$ la relation sur les clauses définie par $C \sqsubseteq C'$ si $C \models C'$.

Soit $\geq$ un ordre total bien fondé sur les *littéraux*. On considère la restriction suivante de la rgle de résolution :

$$\frac{C \vee P \quad \neg P \vee C'}{C \vee C'} \qquad \text{Si } P \text{ est maximal dans } C \vee P \text{ ET } \neg P \text{ est maximal dans } \neg P \vee C'$$

Noter qu'il s'agit d'une généralisation de la résolution avec stratégie ordonnée puisqu'on peut avoir $P \geq Q \geq \neg P$ par exemple.

Si $\mathcal{E}$ est un ensemble de clauses, on note $\mathcal{E}^*$ l'ensemble des conséquences de $\mathcal{E}$ par factorisation et résolution suivant la stratégie $\mathcal{S}$ ci-dessus.

1. Montrer que $\sqsubseteq$ est une relation d'ordre bien fondée sur l'ensemble des clauses.

2. Montrer que la stratégie négative est un cas particulier de la stratégie ci-dessus (rappel : la stratégie négative consiste à ne faire de résolution que sur des clauses dont l'une des prémisses ne contient que des littéraux négatifs).
3. La stratégie *unitaire* consiste à restreindre la règle de résolution au cas où l'une des prémisses est réduite à un littéral.

   Si $\mathcal{E}$ est un ensemble de clauses, soit $\mathcal{E}_U$ l'ensemble des clauses que l'on peut déduire de $\mathcal{E}^*$ par la stratégie unitaire et $\mathcal{E}_S$ l'ensemble des clauses minimales de $\mathcal{E}_U$ pour l'ordre $\sqsubseteq$.

   Montrer que, si $\mathcal{E}^*$ ne contient pas $\perp$, alors $\mathcal{E}_S$ ne contient pas $\perp$ et $\mathcal{E}_S^* = \mathcal{E}_S$.
4. Montrer que, si $\mathcal{E}^*$ ne contient ni $\perp$ ni clause unitaire, et L est un littéral minimal de $\mathcal{E}*$, alors $(\mathcal{E}^* \cup \{\overline{L}\})^* = \mathcal{E}^* \cup \{\overline{L}\}$.
5. Montrer que la stratégie $\mathcal{S}$ est réfutationellement complète.

Exercice 38 (5)

$\mathcal{L}$ est un ensemble fini d'entiers, appelés étiquettes. Un *littéral étiqueté* est une paire d'un littéral et d'un élément de $\mathcal{L}$, noté $L \text{ et } e$. Une *clause étiquetée* est une disjonction de littéraux étiquetés. Comme d'habitude, la disjonction vide est notée $\perp$. La sémantique d'une clause étiquetée est la même que celle de la clause à laquelle on a retiré les étiquettes. Une *fonction de sélection s* est une application qui associe à toute clause étiquetée un sous-ensemble des littéraux de c. Dans cette partie, on considère les deux règles d'inférence suivantes :

$$R \quad \frac{L \text{ et } e \vee C \quad \overline{L} \text{ et } e' \vee C'}{C \vee C'} \quad \texttt{Si} \begin{cases} L \text{ et } e \in s(L \text{ et } e \vee C) \\ \overline{L} \text{ et } e' \in s(\overline{L} \text{ et } e' \vee C') \end{cases}$$

$$F \quad \frac{L \text{ et } e \vee L \text{ et } e' \vee C}{L \text{ et } e \vee C} \quad \texttt{Si} L \text{ et } e' \in s(L \text{ et } e \vee L \text{ et } e' \vee C)$$

Soit C un ensemble de clauses. On affecte à chaque formule littéral de chaque clause de C une étiquette dans un ensemble fini (on peut affecter des étiquettes différentes à un même littéral apparaissant dans deux clauses différentes). C est ainsi considéré comme un ensemble de clauses étiquetées. Soit $\geq$ un ordre sur les littéraux étiquetés. On considère la fonction de sélection suivante : $s(c)$ est l'ensemble des littéraux $L \text{ et } e$ tels que $L \text{ et } e$ est maximal dans c. On note S_e cette stratégie (paramétrée par $\geq$ et l'étiquetage).

On suppose d'abord que $\geq$ est un ordre total bien fondé. À une clause $C = L_1 \text{ et } a_1 \vee \cdots \vee L_n \text{ et } a_n$ on associe le multi-ensemble des littéraux étiquetés $m(C) = \{L_1 \text{ et } a_1, \ldots, L_n \text{ et } a_n\}$. Les clauses sont ainsi ordonnées par l'extension multi-ensemble de $\geq$. Si $\mathcal{S}$ est un ensemble de clauses et $L \text{ et } a$ est un littéral étiqueté, on note $\mathcal{S}(L)$ l'ensemble des clauses C ne contenant aucun littéral $L \text{ et } b$ et telles que $C \in \mathcal{S}$ ou bien $C \vee \overline{L} \text{ et } b \in \mathcal{S}$ pour au moins un b. (Autrement dit, on remplace L par $\top$ dans les clauses de $\mathcal{S}$ et on simplifie). Si $\mathcal{E}$ est un ensemble de clauses, on note de plus $\mathcal{E}^*$ l'ensemble des clauses déductibles de $\mathcal{E}$ par la stratégie S_e. Montrer que, si $\perp \notin \mathcal{E}^*$, C est une clause minimale de $\mathcal{E}^*$ et L est un littéral maximal de C, alors $\perp \notin (\mathcal{E}^*(L))^*$. Montrer la complétude réfutationnelle de la stratégie S_e sur les clauses étiquetées.