Quelques raffinements de la résolution propositionnelle

Jean Goubault-Larrecq

February 19, 1999

Le but de cette note est de montrer comment obtenir certains des raffinements usuels de la résolution — hyperrésolution, résolution sémantique, raffinements ordonnés — par des techniques à la Gentzen de réécriture de preuves dans $\mathbf{LK}_0$. Ces techniques s'étendront au premier ordre naturellement, par la technique standard de *relèvement* (lifting en anglais).

1 Résolution et preuves à base de coupures

Rappelons qu'une clause $\neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_m \vee B_1 \vee B_2 \vee \ldots \vee B_n$, où les A_i et les B_j sont atomiques, n'est qu'une notation commode pour le séquent atomique:

$$A_1, A_2, \ldots, A_m \longrightarrow B_1, B_2, \ldots, B_n$$

La règle de résolution :

$$\frac{C \vee A \quad C' \vee \neg A}{C \vee C'}$$

n'est alors rien d'autre que la règle de coupure Cut de $\mathbf{LK}_0$, restreinte aux séquents atomiques :

$$\frac{\Gamma \longrightarrow \Delta, A \quad \Gamma', A \longrightarrow \Delta'}{\Gamma, \Gamma' \longrightarrow \Delta, \Delta'}$$

Soit S un ensemble de clauses, et nommons $\mathbf{LK}_0 + S$ le système logique obtenue en ajoutant aux règles de $\mathbf{LK}_0$ les règles :

$$\frac{}{\Gamma \longrightarrow \Delta}$$

pour toute clause $\Gamma \longrightarrow \Delta$ dans S.

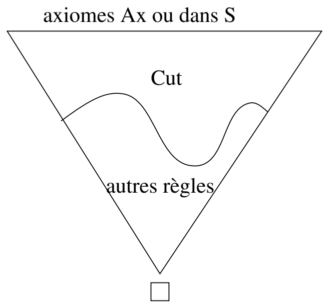
Toute dérivation par résolution à partir de S est ainsi une preuve d'un certain séquent dans $\mathbf{LK}_0 + S$, et toute réfutation par résolution est une preuve de la clause vide $\Box$, c'est-à-dire du séquent vide $\longrightarrow$ dans $\mathbf{LK}_0 + S$.

Mais il s'agit d'une preuve d'un format très spécial : la seule règle de $\mathbf{LK}_0$ qu'elle utilise est la coupure Cut. Ceci est étonnant, parce que c'est en quelque sorte un résultat inverse de l'élimination des coupures (qui dit que la seule règle inutile en $\mathbf{LK}_0$ serait justement Cut).

On peut en fait montrer la complétude de la résolution en effectuant les étapes suivantes (consulter la fin du chapitre "Résolution au premier ordre" pour plus de détails). Soit F une formule valide, S une forme clausale de $\neg F$ (qui est donc insatisfiable), on souhaite montrer que l'on peut déduire le séquent vide $\longrightarrow$ à partir de S en utilisant juste la règle de coupure. Pour ceci :

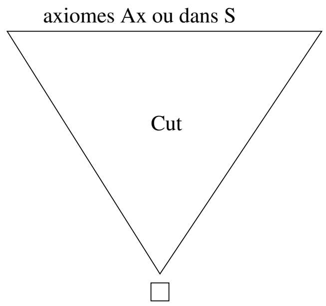
1. Comme S est insatisfiable, $S \longrightarrow$ est prouvable dans $\mathbf{LK}_0$ par la complétude de $\mathbf{LK}_0$; d'autre part, $\longrightarrow S$ est trivialement prouvable dans $\mathbf{LK}_0 + S$ (pas dans $\mathbf{LK}_0$, évidemment). En utilisant une coupure, on en déduit que le séquent vide est dérivable dans $\mathbf{LK}_0 + S$.

2. On applique maintenant les mouvements d'élimination des coupures (cf. tronc commun, ou bien le chapitre “Logique classique du premier ordre”). Ceci fait remonter les coupures le plus haut possible, et termine. On se retrouve alors avec une preuve de la forme :

   avec tous les axiomes de la forme Ax ou dans S en haut, des coupures en dessous et toutes les autres règles encore en-dessous. (Toutes les règles logiques, c'est-à-dire portant sur les connecteurs logiques, ont en effet nécessairement disparu, parce que les clauses de S sont des séquents *atomiques*.)

   Mais la conclusion étant le séquent vide, la dernière règle appliquée ne peut être qu'une coupure ou bien un séquent de S (si la clause vide est dans S). Dans tous les cas, il n'y a *pas* d'autre règle que Cut appliquée dans la partie basse de la preuve, et la preuve est donc de la forme :

3. Il ne reste plus qu'à montrer qu'on n'a jamais besoin de la règle Ax en haut de la preuve. Or aucune instance de Ax ne peut être la conclusion (clause vide) de la preuve, donc toute instance de Ax est forcément une prémisse de Cut. On transforme les cas où la formule axiome (celle qui est la même de chaque côté du séquent) n'est pas la formule de coupure :

$$\dfrac{\dfrac{}{\Gamma \longrightarrow \Delta, A} \qquad \dfrac{}{\Gamma', A, B \longrightarrow \Delta', B}}{\Gamma, \Gamma', B \longrightarrow \Delta, \Delta', B}\, Cut$$

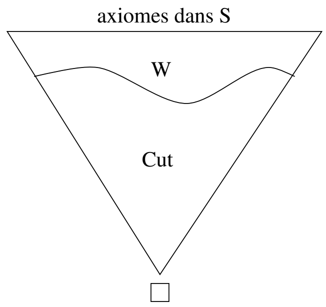
   en des instances de Ax (et symétriquement si A est à gauche du séquent de gauche). Et l'on transforme les cas où la formule axiome (celle qui est la même de chaque côté du séquent) est la formule de coupure :

$$\dfrac{\dfrac{}{\Gamma \longrightarrow \Delta, A} \qquad \dfrac{}{\Gamma', A \longrightarrow A, \Delta'}\,Ax}{\Gamma, \Gamma' \longrightarrow \Delta, \Delta', A}\,Cut$$

en :

$$\dfrac{\dfrac{}{\Gamma \longrightarrow \Delta, A}}{\Gamma, \Gamma' \longrightarrow \Delta, \Delta', A}\,W$$

(et symétriquement si A est à gauche du séquent de gauche), où W est une règle d'affaiblissement que l'on ajoute au calcul. Comme le processus termine, on obtient une preuve n'utilisant que les règles Cut, W, et les axiomes de S, de :



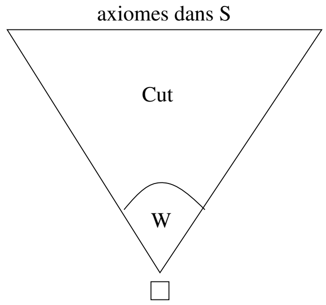
4. On observe maintenant que l'on peut permuter W et Cut de sorte à faire descendre les instances de W dans la preuve. Les instances de W qui introduisent une formule de coupure absorbent la coupure qui est en-dessous, par exemple :

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ \Gamma \longrightarrow \Delta\end{matrix}}{\Gamma, \Gamma' \longrightarrow \Delta, \Delta', A}\,W \qquad \begin{matrix}\vdots\\ \Gamma'', A \longrightarrow \Delta''\end{matrix}}{\Gamma, \Gamma', \Gamma'' \longrightarrow \Delta, \Delta', \Delta''}\,Cut \quad \rightsquigarrow \quad \dfrac{\begin{matrix}\vdots\\ \Gamma \longrightarrow \Delta\end{matrix}}{\Gamma, \Gamma', \Gamma'' \longrightarrow \Delta, \Delta', \Delta''}\,W$$

Et les instances de W qui n'introduisent pas la formule de coupure permutent avec la coupure qui est en-dessous :

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ \Gamma \longrightarrow \Delta, A\end{matrix}}{\Gamma, \Gamma' \longrightarrow \Delta, \Delta', A}\,W \qquad \begin{matrix}\vdots\\ \Gamma'', A \longrightarrow \Delta''\end{matrix}}{\Gamma, \Gamma', \Gamma'' \longrightarrow \Delta, \Delta', \Delta''}\,Cut \quad \rightsquigarrow \quad \dfrac{\dfrac{\begin{matrix}\vdots\\ \Gamma \longrightarrow \Delta\end{matrix} \quad \begin{matrix}\vdots\\ \Gamma'', A \longrightarrow \Delta''\end{matrix}}{\Gamma, \Gamma'' \longrightarrow \Delta, \Delta''}\,Cut}{\Gamma, \Gamma', \Gamma'' \longrightarrow \Delta, \Delta', \Delta''}\,W$$

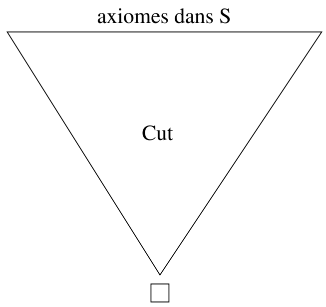
Il est facile de voir que le processus termine, et on aboutit à une preuve n'utilisant que Cut, W et les axiomes de S de la forme :



Mais les seules instances de W qui permettent de déduire le séquent vide (en bas) sont des instances de la forme:

$$\frac{\longrightarrow}{\longrightarrow}\, W$$

On peut donc toutes les supprimer, et il reste une preuve de la forme :



ce qui était la forme souhaitée.

On a ainsi prouvé de façon essentiellement syntaxique (modulo la preuve de complétude de $\mathbf{LK}_0$) que la résolution était une méthode complète de preuve.

Il y a cependant quelques détails à règler. Dans l'étape 1, par exemple, on a utilisé S de façon ambiguë soit comme une formule, soit comme un ensemble de séquents. Il faut être plus précis : S est une disjonction de conjonctions D_i, $1 \leq i \leq n$, où D_i est la formule $L_{i1} \vee \ldots \vee L_{in_i}$, les L_{ij} étant des littéraux. Soit C_i la clause (le séquent) que l'on a aussi décidé de noter $L_{i1} \vee \ldots \vee L_{in_i}$. Ce n'est pas $S \longrightarrow$ qui est prouvable, mais $D_1, \ldots, D_n \longrightarrow$; d'autre part, D_i est clairement prouvable à partir de $\mathbf{LK}_0 + S$ (partir de C_i), pour tout i : on en conclut que $\longrightarrow$ est prouvable en $\mathbf{LK}_0 + S$, en utilisant Cut n fois.

2 Tautologies, subsomption, clauses pures

Nous identifions dans la suite les notions de clauses et de séquents. Considérons les types suivants de clauses :

Une *tautologie* est une clause contenant à la fois un atome A et sa négation $\neg A$.

On dit qu'une clause C est *subsumée* par une clause C' si et seulement si $C = C' \vee C''$ (en abrégé, $C' \subseteq C$) pour une certaine clause C''. En particulier, C est impliquée par C'.

Une clause C est *pure* dans l'ensemble de clauses S si et seulement si C contient un littéral A tel que $\neg A$ n'apparaisse dans aucune clause de S, ou bien un littéral $\neg A$ tel que A n'apparaisse dans aucune clause de S.

Intuitivement, les tautologies, les clauses subsumées par d'autres clauses, les clauses pures sont sémantiquement inutiles ou redondantes. On peut donc déjà les effacer de l'ensemble initial de clauses S.

Ce qu'on peut se demander, c'est s'il est légal d'effacer de telles clauses lorsqu'elles se retrouvent générées par l'algorithme de résolution. En effet, on peut imaginer la situation suivante : sans effacer les clauses inutiles ou redondantes, supposons qu'il faille 3 étapes de résolution pour réfuter S. Fabriquons un résolvant et ajoutons-le à S, nous sommes à deux étapes de la clause vide. Maintenant supprimons par exemple une clause subsumée de S : il est envisageable que, bien que S reste insatisfiable, il nous faille maintenant 4 étapes de résolution pour réfuter S. Si le processus continue, nous engendrerons des ensembles de clauses qui, même s'ils sont tous insatisfiables, s'éloigneront petit à petit d'un ensemble contenant la clause vide, au lieu de s'en rapprocher.

Il est plus commode ici d'abandonner pour un temps la présentation de la résolution en terme de preuves formées uniquement avec des clauses de S et des coupures. Rappelons qu'une étape de résolution à partir de S est une transition :

$$S \to S \cup \{C \vee C'\}$$

où le résolvant $C \vee C'$ est obtenu à partir de deux clauses $C \vee A$ et $C' \vee \neg A$ de S.

Une dérivation par résolution de la clause C_0 à partir de S est une suite finie d'étapes de résolution :

$$S = S_0 \to S_1 \to \ldots \to S_n \ni C_0$$

Il s'agit d'une *réfutation de* S si C_0 est la clause vide.

Il est facile de voir que l'on peut toujours transformer une dérivation par résolution en ce sens en une dérivation par coupures au sens de la section 1 (par récurrence sur n), et réciproquement. L'unique différence est que les dérivations vues en tant que suites d'étapes peuvent être beaucoup plus courtes que les dérivations par coupures, parce qu'elles partagent systématiquement les sous-dérivations. Par exemple, considérons l'ensemble de clauses :

$$S_0 = \{A \vee D, A \vee \neg D, \neg A \vee B, \neg A \vee E, \neg B \vee \neg E\}$$

S_0 est insatisfiable : résoudre $A \vee D$ avec $A \vee \neg D$, pour obtenir A, ceci produit $S_1 = S_0 \cup \{A\}$. Puis résoudre A et $\neg A \vee B$, pour obtenir $S_2 = S_1 \cup \{B\}$; résoudre A encore avec $\neg A \vee E$, pour obtenir $S_3 = S_2 \cup \{E\}$; résoudre B avec $\neg B \vee \neg E$ pour obtenir $S_4 = S_3 \cup \{\neg E\}$, enfin résoudre avec E pour obtenir S_5 qui contient la clause vide. Mais une représentation par preuve avec coupures duplique la dérivation de A à partir de $A \vee D$ et $A \vee \neg D$:

$$\dfrac{\dfrac{\dfrac{A \vee D \quad A \vee \neg D}{A} \qquad \neg A \vee B}{B} \qquad \dfrac{\dfrac{\dfrac{A \vee D \quad A \vee \neg D}{A} \qquad \neg A \vee E}{E} \qquad \neg B \vee \neg E}{\neg B}}{\square}$$

Ceci étant dit, examinons l'effet de la subsomption :

Lemme 1 *Soient deux clauses C_1 et C'_1, et supposons que C_1 subsume C'_1. Pour tout résolvant C' entre C'_1 et C_2, C' est subsumée par C_1 ou par un résolvant de C_1 et C_2.*

De plus, l'unique clause qui subsume la clause vide est la clause vide.

Preuve : Posons $C_1' = C_1 \vee C'$. Supposons que la formule de coupure intervienne en tant que $\neg A$ dans C_2 (le cas A est similaire), et en tant que A dans C_1'. Alors C_2 est de la forme $C_2' \vee \neg A$. Si A est dans C', alors C' est de la forme $C'' \vee A$, donc $C' = C_1 \vee C'' \vee C_2'$ est subsumée par C_1. Sinon, alors C_1 est de la forme $C'' \vee A$, et $C' = C'' \vee C' \vee C_2'$, qui est subsumée par le résolvant $C'' \vee C_2'$ de $C_1 = C'' \vee A$ et de $C_2 = C_2' \vee \neg A$. $\square$

Il s'ensuit :

Corollaire 2 *Soit S et S' deux ensembles de clauses. Supposons que toute clause C de S est dans S' ou est subsumée par une clause de S'.*

Si la clause vide est dérivable à partir de S en n étapes de résolution, alors elle est dérivable à partir de S' en au plus n étapes de résolution.

Preuve : Par récurrence sur n. Si $n = 0$, alors la clause vide est dans S, donc par hypothèse elle est dans S' ou elle est subsumée par une clause de S'; dans chaque cas, la clause vide doit déjà être dans S'. Sinon, supposons que le résultat soit valide pour $n-1$ étapes, et que nous puissions dériver la clause vide de S en n étapes, $n > 0$. Examinons la première étape de résolution dans la dérivation : nous résolvons deux clauses C_1 et C_2 pour produire le résolvant C.

Par hypothèse, C_1 est subsumée par une clause C_1' de S' (éventuellement C_1 elle-même), et C_2 est subsumée par une clause C_2' de S'. Par le lemme 1, C est subsumée par une clause C', qui peut être C_1', C_2' ou un résolvant de C_1' et C_2'. Alors, $S \cup \{C\}$ et $S' \cup \{C'\}$ sont deux ensembles de clauses vérifiant les conditions du théorème, et nous pouvons dériver la clause vide en $n-1$ étapes de $S \cup \{C\}$. Par hypothèse de récurrence, nous pouvons dériver la clause vide en au plus $n-1$ étapes de $S' \cup \{C'\}$, donc en au plus n étapes de S'. $\square$

Nous pouvons ainsi conclure qu'il est légal de supprimer les clauses subsumées. Appelons *distance* de S à la clause vide la plus petite longueur n d'une réfutation à partir de S. Supposons S_0 à distance n de la clause vide. Supprimer toutes les clauses subsumées de S_0, une à une (pas en parallèle !), produit un sous-ensemble S_0' de S_0 qui est à distance au plus n de la clause vide, par le corollaire 2. Donc soit la clause vide est dans S_0', donc dans S_0, soit il existe une étape de résolution menant de S_0' à un ensemble de clauses S_1 à distance au plus $n-1$ de la clause vide. Par récurrence sur n, nous montrons ainsi facilement que pour toute stratégie d'élimination des clauses subsumées, la clause vide est toujours accessible en au plus n étapes de résolution.

On peut presque voir pourquoi on peut éliminer les clauses subsumées en format par coupures. En particulier, le cas suivant est particulièrement clair :

$$\begin{array}{c} \vdots \\ C \\ \vdots \\ C' \end{array} \quad \rightsquigarrow \quad \begin{array}{c} \vdots \\ \dfrac{C}{C'}\,W \end{array}$$

lorsque C, subsumée par C', est obtenue comme conclusion d'une sous-preuve de celle qui conclut C'. Comme on peut permuter les W vers le bas sans augmenter la taille de la preuve, et comme il ne peut subsister d'instances de W en bas de la preuve, cette opération diminue strictement la taille de la preuve. Mais cette explication ne fonctionne que si C est au-dessus de C', pas à côté par exemple.

Quant aux tautologies, nous avons :

Lemme 3 *Soit C une tautologie. Alors tout résolvant de C avec n'importe quelle clause C' est une tautologie ou est subsumée par C'.*

Preuve : Soit $C = C_1 \vee A \vee \neg A$. Tout résolvant entre C et C' de formule de coupure différente de A contient à la fois A et $\neg A$, et est donc une tautologie. Les résolutions portant sur A sont de la forme :

$$\frac{C_1 \vee A \vee \neg A \quad C' = C_2 \vee \neg A}{C_1 \vee C_2 \vee \neg A}$$

ou

$$\frac{C_1 \vee A \vee \neg A \quad C' = C_2 \vee A}{C_1 \vee C_2 \vee A}$$

et fournissent donc des clauses subsumées par C' dans les deux cas. □

Comme aucune tautologie n'est la clause vide, et que tout résolvant d'une tautologie avec une clause est une clauses subsumée et peut donc être supprimé, il n'est nécessaire de résoudre avec une tautologie nulle part dans une dérivation minimale. On peut donc toujours supprimer toutes les tautologies.

Ceci peut aussi se voir en format de coupures, dans la mesure ou le lemme 3 dit que l'on peut remplacer toute coupure avec une tautologie par une instance de Ax ou par une instance de W. On peut alors rejouer les mouvements de preuves correspondants présentés en section 1.

Pour les clauses pures, c'est encore plus simple :

Lemme 4 *Tout résolvant d'une clause pure avec n'importe quelle clause est pure.*

Preuve : Toute résolution entre une clause où A (resp. $\neg A$) est pur avec une autre clause doit opérer sur un autre littéral B. Le résolvant binaire produit doit donc contenir le littéral A (resp. $\neg A$), et est donc pur. □

Encore une fois, aucune clause pure n'est vide, et on peut donc les supprimer.

3 Hyperrésolution

Nous revenons à partir de maintenant à une présentation des dérivations par résolution sous forme d'arbres de preuves par coupures.

Une réfutation par résolution, c'est-à-dire une preuve de la clause vide n'utilisant que la règle Cut à partir de l'ensemble S de clauses, est un objet très malléable que l'on peut déformer à volonté.

L'idée pour améliorer l'efficacité de la recherche de preuve, est de déformer systématiquement les réfutations dans l'espoir d'arriver à un format canonique de réfutation, de même que l'élimination des coupures fournissait les preuves sans coupure comme formes canoniques des preuves de $\mathbf{LK}_0$. Ceci mène à chercher des restrictions de la règle de résolution qui préservent la complétude. On les appelle des *raffinements* de la résolution.

Le premier que nous allons étudier est l'*hyperrésolution positive* : c'est le raffinement de la résolution où l'une des prémisses de la règle de résolution est contrainte à être une clause *positive*, c'est-à-dire une clause de la forme $A_1 \vee \ldots \vee A_n$ (aucune négation). C'est donc la règle :

$$\frac{\overbrace{C \vee A}^{\text{positive}} \quad C' \vee \neg A}{C \vee C'}$$

Intuitivement, une clause positive C est un *fait* connu, exprimé sous forme de disjonction de faits élémentaires, les atomes. Une clause non positive $\neg A_1 \vee \ldots \vee \neg A_m \vee B_1 \vee \ldots \vee B_n$, $m \geq 1$, est alors une *règle de déduction* $A_1 \wedge \ldots \wedge A_m \Rightarrow B_1 \vee \ldots \vee B_n$. La règle d'hyperrésolution positive exprime alors que l'on peut se contenter de dériver de nouvelles clauses à partir de faits, ce en utilisant les règles, pour aboutir au fait vide. On n'a jamais besoin d'inférer des règles dérivées d'abord purement à partir d'autres règles.

Par exemple, pour montrer que :

$$A \vee B \quad A \vee \neg B \quad \neg A \vee B \quad \neg A \vee \neg B$$

est insatisfiable, nous nous interdisons de résoudre $A \vee \neg B$ avec $\neg A \vee \neg B$. Par contre, nous partons du fait $A \vee B$, et déduisons les nouveaux faits A (par résolution avec $A \vee \neg B$ sur B), B (avec $\neg A \vee B$ sur A), $\neg B$ (par résolution entre le fait A et $\neg A \vee \neg B$), et enfin la clause vide (par résolution entre le fait B et la règle $\neg B$).

Pour montrer que cette règle est complète, prenons une réfutation par résolution quelconque, et transformons-la petit à petit en une réfutation par hyperrésolution positive.

Remarquons d'abord :

Lemme 5 *Soit π une réfutation par résolution à partir de S. Si π n'est pas obtenue par hyperrésolution positive, alors π contient un* rédex *de la forme :*

$$\dfrac{\begin{array}{c}\vdots\\ \overbrace{C \vee A}^{\text{positive}}\end{array} \qquad \dfrac{\begin{array}{c}\vdots\\ \overbrace{C_1 \vee B}^{\text{non positive}}\end{array} \qquad \begin{array}{c}\vdots\\ C_2 \vee \neg B\end{array}}{C_1 \vee C_2 = C' \vee \neg A}}{C \vee C'}$$

Preuve : Comme π n'est pas obtenue par hyperrésolution positive, elle contient une coupure entre deux clauses non positives. Choisissons une occurrence minimale d'une telle coupure, c'est-à-dire la plus basse possible dans π :

$$\dfrac{\begin{array}{c}\vdots\\ \overbrace{C_1 \vee B}^{\text{non positive}}\end{array} \qquad \begin{array}{c}\vdots\\ C_2 \vee \neg B\end{array}}{C_1 \vee C_2}$$

Comme $C_1 \vee B$ n'est pas positive, C_1 n'est pas positive et $C_1 \vee C_2$ non plus. En particulier, $C_1 \vee C_2$ n'est pas la conclusion de π, qui est la clause vide. Donc $C_1 \vee C_2$ est nécessairement prémisse d'une coupure dans π. Comme la coupure ci-dessus a été choisie minimale telle que ses deux prémisses soient non positives, $C_1 \vee C_2$ est donc nécessairement coupée avec une clause positive dans π. □

Nous définissons alors les règles de réécriture suivantes sur les rédex. Il y a trois cas, selon que $\neg A$, qui est dans $C_1 \vee C_2$, est dans C_1 uniquement, dans C_2 uniquement, ou dans C_1 et C_2 en même temps :

$$(1)\qquad \dfrac{\overset{\vdots\,\pi_0}{C \vee A} \quad \dfrac{\overset{\vdots\,\pi_1}{C'_1 \vee \neg A \vee B} \quad \overset{\vdots\,\pi_2}{C_2 \vee \neg B}}{C'_1 \vee C_2 \vee \neg A}}{C \vee C'_1 \vee C_2} \quad \rightarrow \quad \dfrac{\dfrac{\overset{\vdots\,\pi_0}{C \vee A} \quad \overset{\vdots\,\pi_1}{C'_1 \vee \neg A \vee B}}{C \vee C'_1 \vee B} \quad \overset{\vdots\,\pi_2}{C_2 \vee \neg B}}{C \vee C'_1 \vee C_2}$$

$$(2)\qquad \dfrac{\overset{\vdots\,\pi_0}{C \vee A} \quad \dfrac{\overset{\vdots\,\pi_1}{C_1 \vee B} \quad \overset{\vdots\,\pi_2}{C'_2 \vee \neg A \vee \neg B}}{C_1 \vee C'_2 \vee \neg A}}{C \vee C_1 \vee C'_2} \quad \rightarrow \quad \dfrac{\overset{\vdots\,\pi_1}{C_1 \vee B} \quad \dfrac{\overset{\vdots\,\pi_0}{C \vee A} \quad \overset{\vdots\,\pi_2}{C'_2 \vee \neg A \vee \neg B}}{C \vee C'_2 \vee \neg B}}{C \vee C_1 \vee C'_2}$$

$$(3)\qquad \dfrac{\overset{\vdots\,\pi_0}{C \vee A} \quad \dfrac{\overset{\vdots\,\pi_1}{C'_1 \vee \neg A \vee B} \quad \overset{\vdots\,\pi_2}{C'_2 \vee \neg A \vee \neg B}}{C'_1 \vee C'_2 \vee \neg A}}{C \vee C'_1 \vee C'_2} \quad \rightarrow \quad \dfrac{\dfrac{\overset{\vdots\,\pi_0}{C \vee A} \quad \overset{\vdots\,\pi_1}{C'_1 \vee \neg A \vee B}}{C \vee C'_1 \vee B} \quad \dfrac{\overset{\vdots\,\pi_0}{C \vee A} \quad \overset{\vdots\,\pi_2}{C'_2 \vee \neg A \vee \neg B}}{C \vee C'_2 \vee B}}{C \vee C'_1 \vee C'_2}$$

où il est sous-entendu que C est une clause positive, que dans la règle (2) C_1 n'est pas positive, et que dans toute sous-dérivation notée sous la forme :

$$\overset{\vdots\,\pi'}{C' \vee L_1 \vee \ldots \vee L_k}$$

les littéraux L_1, …, L_k n'apparaissent pas dans C'.

Si cette relation de réécriture termine, toute forme normale d'une réfutation par résolution est donc une réfutation par hyperrésolution positive à partir du même ensemble S de clauses de départ, par le lemme 5. Il ne reste qu'à montrer que ce système de réécriture termine (faiblement suffirait).

Pour ceci, on traduit les preuves dans un langage du premier ordre de signature f_n, $n \in \mathbb{N}$ (unaires) et a (constante) par $T_{hyper}(C) = a$ si C est un axiome dans S (une preuve sans prémisse), et sinon :

$$T_{hyper}\left(\dfrac{\overset{\vdots\,\pi_1}{C \vee A} \quad \overset{\vdots\,\pi_2}{C' \vee \neg A}}{C \vee C'}\right) = f_n(T_{hyper}(\pi_1), T_{hyper}(\pi_2))$$

où n est le nombre de littéraux négatifs (i.e., de la forme $\neg D$) dans la clause C.

Notons n_1 le nombre de littéraux négatifs de C_1, n_2 celui de C_2, n'_1 celui de C'_1, n'_2 celui de C'_2. Remarquons que dans le cas (2) $n_1 \geq 1$, car C_1 n'est pas positive : on peut donc poser $C_1 = C'_1 \vee \neg A'$ pour un certain A' et choisir $n_1 = n'_1 + 1$. (Ceci permet d'uniformiser la notation de la règle (2) ci-dessous par rapport aux autres.)

Via la traduction T_{hyper}, nous sommes donc ramenés à prouver que le système de réécriture suivant termine :

$$\begin{array}{lrcl}
(1) & f_0(t_0, f_{n'_1+1}(t_1, t_2)) & \rightarrow & f_{n'_1}(f_0(t_0, t_1), t_2) \\
(2) & f_0(t_0, f_{n'_1+1}(t_1, t_2)) & \rightarrow & f_{n'_1+1}(t_1, f_0(t_0, t_2)) \\
(3) & f_0(t_0, f_{n'_1+1}(t_1, t_2)) & \rightarrow & f_{n'_1}(f_0(t_0, t_1), f_0(t_0, t_2))
\end{array}$$

Lemme 6 *Le système de réécriture ci-dessus termine.*

Preuve : Nous définissons d'abord une réinterprétation des termes dans les entiers par :

$$[f_n(t_1, t_2)] = n.[t_1] + [t_2] \qquad [a] = 1$$

Nous remarquons que :

(a) pour tout t, $[t] > 0$ (une récurrence structurelle facile sur t);

(b) $[f_{n'_1+1}(t_1, t_2)] > [t_1]$ (car $n'_1 + 1 \geq 1$, et $[t_2] > 0$ par (a));

(c) $[f_{n'_1+1}(t_1, t_2)] > [t_2]$ (car $n'_1 + 1 \geq 1$, et $[t_1] \geq 1$ par (a));

(d) $[f_{n'_1+1}(t_1, t_2)] > [f_0(t_0, t_2)]$ pour tout terme t_0 (car $[f_0(t_0, t_2)] = [t_2]$, et (c));

(e) Notons $\Rightarrow$ la relation de réduction induite par les règles $(1)-(3)$; si $t \Rightarrow t'$, alors $[t] \geq [t']$ (en effet, c'est vrai si $t \to t'$ est l'une des règles, et l'interprétation est monotone : $[t_1] \geq [t'_1]$ et $[t_2] \geq [t'_2]$ impliquent $[f_n(t_1, t_2)] \geq [f_n(t'_1, t'_2)]$);

Soit SN l'ensemble des termes t fortement normalisants, c'est-à-dire tels qu'il n'existe pas de réduction infinie partant de t. Remarquons que pour tout $t \in SN$, on peut définir la plus grande longueur des réductions partant de t : il n'y a en effet qu'un nombre fini de telles réductions, par le lemme de König, puisque tout terme n'a qu'un nombre fini de réduits (en une étape) possibles. Remarquons que si $t \Rightarrow t'$, alors $\nu(t) > \nu(t')$.

Nous montrons maintenant que : (f) $f_n(t_0, t) \in SN$ pour tout $n \geq 1$ et tous $t_0, t \in SN$. En effet, toutes les réductions partant de $f_n(t_0, t)$ prennent place dans t_0 ou t, et sont donc finies. Formellement, on démontre (f) par récurrence sur $\nu(t_0) + \nu(t)$. Soit R une réduction arbitraire partant de $f_n(t_0, t)$. Si R est vide, elle est trivialement finie. Sinon, R est de la forme $f_n(t_0, t) \Rightarrow u$ suivi d'une réduction R'. Nous avons alors deux cas. Dans le premier cas, u est de la forme $f_n(t'_0, t)$, avec $t_0 \Rightarrow t'_0$, donc $\nu(t'_0) < \nu(t_0)$: nous pouvons donc appliquer l'hypothèse de récurrence et en déduire que R' est finie. Dans le second cas, u est de la forme $f_n(t_0, t')$ avec $t \Rightarrow t'$, donc $\nu(t') < \nu(t)$ et encore par hypothèse de récurrence R' est finie. Dans les deux cas, on conclut que R est finie. Comme R est arbitraire, $f_n(t_0, t) \in SN$.

Ensuite, nous avons : (g) $f_0(t_0, t) \in SN$ pour tous $t_0, t \in SN$. Ceci se démontre de façon similaire à (f), mais cette fois-ci par récurrence sur $([t], \nu(t_0) + \nu(t))$ ordonné lexicographiquement. Allons un peu plus vite : le seul cas qui change est celui où R est non vide, et est donc de la forme $f_0(t_0, t) \Rightarrow u$ suivi d'une réduction R'. Si $u = f_0(t'_0, t)$ avec $t_0 \Rightarrow t'_0$, R' est finie par récurrence. Si $u = f_0(t_0, t')$ avec $t \Rightarrow t'$, alors $[t'] \leq [t]$ par (e), et $\nu(t_0) + \nu(t') < \nu(t_0) + \nu(t)$, donc nous pouvons encore appliquer l'hypothèse de récurrence et conclure que R' est finie. Si u est un rédex, alors t est de la forme $f_{n'_1+1}(t_1, t_2)$, et nous avons trois sous-cas, correspondant chacun à une règle :

(1) par (b), $[t_1] < [t]$, donc nous pouvons appliquer l'hypothèse de récurrence et conclure que : (h.1.a) $f_0(t_0, t_1) \in SN$. D'autre part, puisque $t = f_{n'_1+1}(t_1, t_2) \in SN$, nous avons aussi : (h.1.b) $t_2 \in SN$. Si $n'_1 > 0$, alors on déduit de (f), (h.1.a) et (h.1.b) que $f_{n'_1}(f_0(t_0, t_1), t_2) \in SN$, donc que R' est finie. Si $n'_1 = 0$, alors par (c) $[t_2] < [t]$, et ceci nous permet de conclure que $f_{n'_1}(f_0(t_0, t_1), t_2) \in SN$ par hypothèse de récurrence, (h.1.a) et (h.1.b), donc encore une fois que R' est finie.

(2) par (c), $[t_2] < [t]$, donc nous pouvons appliquer l'hypothèse de récurrence et conclure que $f_0(t_0, t_2) \in SN$. Comme $t_1 \in SN$ puisque $t \in SN$, on en déduit par (f) que $f_{n'_1+1}(t_1, f_0(t_0, t_2)) \in SN$, donc que R' est finie.

(3) comme ci-dessus, par récurrence nous inférons que : (h.3.a) $f_0(t_0, t_1) \in SN$ et : (h.3.b) $f_0(t_0, t_2) \in SN$. Si $n'_1 > 0$, on en déduit par (f) que $f_{n'_1}(f_0(t_0, t_1), f_0(t_0, t_2))$ est dans SN, et donc que R' est finie. Sinon $n'_1 = 0$, mais par (d) $[f_0(t_0, t_2)] < [f_{n'_1+1}(t_1, t_2)]$, donc nous pouvons appliquer l'hypothèse de récurrence encore une fois pour inférer que $f_{n'_1}(f_0(t_0, t_1), f_0(t_0, t_2))$ est dans SN.

Dans tous les cas, R' est finie, donc R aussi. Comme R est arbitraire, il en découle que $f_0(t_0, t) \in SN$.

Nous montrons maintenant que $t \in SN$ pour tout t, par récurrence structurelle sur t : si $t = a$, c'est évident, si t est de la forme $f_n(t_0, t')$, c'est une conséquence de (f) si $n \geq 1$ et de (g) si $n = 0$. □

Corollaire 7 *L'hyperrésolution positive est complète.*

Preuve : Soit S un ensemble de clauses insatisfiable. Il existe une réfutation π par résolution de S. Par le lemme 6, π a une forme normale π' pour $\rightarrow$, qui est une réfutation de S. De plus, par le lemme 5, π' est nécessairement une réfutation par hyperrésolution positive. □

4 Résolution ordonnée

Un autre raffinement classique de la résolution est la *résolution ordonnée*. Considérons, comme dans les BDD, que nous nous sommes donné un ordre total $<$ sur les atomes. Supposons que S est un ensemble insatisfiable de clauses, et numérotons les atomes libres dans S $A_1 < A_2 < \ldots < A_n$. Si nous reprenons la preuve sémantique de la complétude de la résolution, il est facile de voir que pour tout nœud d'inférence n étiqueté par A_j, de fils les nœuds d'échec n_1 et n_2 pour les clauses $C \vee A_j$ et $C' \vee \neg A_j$ respectivement, alors l'étape de résolution qui transforme n en un nouveau nœud d'échec :

$$\frac{C \vee A_j \quad C' \vee \neg A_j}{C \vee C'}$$

est telle que A_j est *maximal* dans C et dans C' : tous les A_i dans C et dans C' doivent être tels que $i < j$. La preuve de complétude de la résolution s'adapte ainsi directement pour montrer la complétude de la règle de *résolution ordonnée* :

$$\frac{C \vee A_j \quad C' \vee \neg A_j}{C \vee C'} \; A_j \text{ maximal dans } C \text{ et dans } C'$$

Par exemple, pour montrer que :

$$A \vee B \quad A \vee \neg B \quad \neg A \vee B \quad \neg A \vee \neg B$$

est insatisfiable, si on se fixe l'ordre $A < B$, ceci nous interdit de résoudre $A \vee B$ et $\neg A \vee B$. Par contre, on peut résoudre $A \vee B$ et $A \vee \neg B$ sur B, donnant A; puis $\neg A \vee B$ et $\neg A \vee \neg B$ sur B, donnant $\neg A$. Et maintenant A et $\neg A$ sur A, donnant la clause vide.

Il est aussi possible de montrer la complétude de la résolution ordonnée par transformation de preuves. C'est en fait beaucoup plus simple que pour l'hyperrésolution.

Comme pour le lemme 5, nous remarquons :

Lemme 8 *Soit π une réfutation par résolution à partir de S. Si π n'est pas obtenue par résolution ordonnée, alors π contient un* rédex *de la forme :*

$$\dfrac{\begin{matrix}\vdots\\ C \vee A\end{matrix} \quad \dfrac{\begin{matrix}\vdots\\ C_1 \vee B\end{matrix} \quad \begin{matrix}\vdots\\ C_2 \vee \neg B\end{matrix}}{C_1 \vee C_2 = C' \vee \neg A}}{C \vee C'} \quad ou \quad \dfrac{\dfrac{\begin{matrix}\vdots\\ C_1 \vee B\end{matrix} \quad \begin{matrix}\vdots\\ C_2 \vee \neg B\end{matrix}}{C_1 \vee C_2 = C' \vee A} \quad \begin{matrix}\vdots\\ C \vee \neg A\end{matrix}}{C \vee C'}$$

où B n'est pas maximal dans C_1 ou C_2, et A est maximal dans C et C'.

Preuve : Comme pour le lemme 5. Comme π n'est pas obtenue par résolution ordonnée, elle contient une coupure entre deux clauses sur un B non maximal. Choisissons une occurrence minimale d'une telle coupure, c'est-à-dire la plus basse possible dans π :

$$\dfrac{\begin{matrix}\vdots\\ C_1 \vee B\end{matrix} \quad \begin{matrix}\vdots\\ C_2 \vee \neg B\end{matrix}}{C_1 \vee C_2}$$

avec B non maximal dans C_1 ou C_2. En particulier, il existe un littéral supérieur à B dans C_1 ou dans C_2, donc dans $C_1 \vee C_2$. Donc $C_1 \vee C_2$ n'est pas la clause vide, et nous concluons comme au lemme 5. $\square$

Notons en particulier que $B < A$. En effet, B n'est pas maximal dans C_1 ou C_2, donc il existe un littéral supérieur à B dans $C_1 \vee C_2$. Mais A est maximal dans $C_1 \vee C_2$, donc $A > B$.

Comme en hyperrésolution, le littéral A ou $\neg A$ dans $C_1 \vee C_2$ peut provenir de C_1, de C_2, ou des deux. Ceci mène aux six règles de réduction $(1)-(3)$ (cf. section 3) plus les suivantes, obtenues par remplacement de A par $\neg A$ et réciproquement dans $(1)-(3)$, soumises à la condition que $B < A$:

$$(4) \quad \dfrac{\dfrac{\begin{matrix}\vdots\,\pi_1\\ C_1' \vee A \vee B\end{matrix} \quad \begin{matrix}\vdots\,\pi_2\\ C_2 \vee \neg B\end{matrix}}{C_1' \vee C_2 \vee A} \quad \begin{matrix}\vdots\,\pi_0\\ C \vee \neg A\end{matrix}}{C \vee C_1' \vee C_2} \quad \to \quad \dfrac{\dfrac{\begin{matrix}\vdots\,\pi_1\\ C_1' \vee A \vee B\end{matrix} \quad \begin{matrix}\vdots\,\pi_0\\ C \vee \neg A\end{matrix}}{C \vee C_1' \vee B} \quad \begin{matrix}\vdots\,\pi_2\\ C_2 \vee \neg B\end{matrix}}{C \vee C_1' \vee C_2}$$

$$(5) \quad \dfrac{\dfrac{\begin{matrix}\vdots\,\pi_1\\ C_1 \vee B\end{matrix} \quad \begin{matrix}\vdots\,\pi_2\\ C_2' \vee A \vee \neg B\end{matrix}}{C_1 \vee C_2' \vee A} \quad \begin{matrix}\vdots\,\pi_0\\ C \vee \neg A\end{matrix}}{C \vee C_1 \vee C_2'} \quad \to \quad \dfrac{\begin{matrix}\vdots\,\pi_1\\ C_1 \vee B\end{matrix} \quad \dfrac{\begin{matrix}\vdots\,\pi_2\\ C_2' \vee A \vee \neg B\end{matrix} \quad \begin{matrix}\vdots\,\pi_0\\ C \vee \neg A\end{matrix}}{C \vee C_2' \vee \neg B}}{C \vee C_1 \vee C_2'}$$

$$(6) \quad \dfrac{\dfrac{\begin{matrix}\vdots\,\pi_1\\ C_1' \vee A \vee B\end{matrix} \quad \begin{matrix}\vdots\,\pi_2\\ C_2' \vee A \vee \neg B\end{matrix}}{C_1' \vee C_2' \vee A} \quad \begin{matrix}\vdots\,\pi_0\\ C \vee \neg A\end{matrix}}{C \vee C_1' \vee C_2'} \quad \to \quad \dfrac{\dfrac{\begin{matrix}\vdots\,\pi_1\\ C_1' \vee A \vee B\end{matrix} \quad \begin{matrix}\vdots\,\pi_0\\ C \vee \neg A\end{matrix}}{C \vee C_1' \vee B} \quad \dfrac{\begin{matrix}\vdots\,\pi_2\\ C_2' \vee A \vee \neg B\end{matrix} \quad \begin{matrix}\vdots\,\pi_0\\ C \vee \neg A\end{matrix}}{C \vee C_2' \vee B}}{C \vee C_1' \vee C_2'}$$

Pour montrer la terminaison, on utilise la traduction suivante T_{ord} définie par $T_{ord}(C) = a$ si C est un axiome dans S (une preuve sans prémisse), et :

$$T_{ord}\left(\dfrac{\begin{matrix}\vdots\,\pi_1\\ C \vee A\end{matrix} \quad \begin{matrix}\vdots\,\pi_2\\ C' \vee \neg A\end{matrix}}{C \vee C'}\right) = f_A(T_{ord}(\pi_1), T_{ord}(\pi_2))$$

où les symboles de fonction f_A sont indicés par les formules de coupure A. Via la traduction T_{ord}, nous sommes ramenés à montrer que le système de réécriture suivant termine :

$$\left.\begin{array}{lllll}
(1) & f_A(t_0, f_B(t_1,t_2)) & \to & f_B(f_A(t_0,t_1),t_2) \\
(2) & f_A(t_0, f_B(t_1,t_2)) & \to & f_B(t_1, f_A(t_0,t_2)) \\
(3) & f_A(t_0, f_B(t_1,t_2)) & \to & f_B(f_A(t_0,t_1), f_A(t_0,t_2)) \\
(4) & f_A(f_B(t_1,t_2),t_0) & \to & f_B(f_A(t_1,t_0),t_2) \\
(5) & f_A(f_B(t_1,t_2),t_0) & \to & f_B(t_1, f_A(t_2,t_0)) \\
(6) & f_A(f_B(t_1,t_2),t_0) & \to & f_B(f_A(t_1,t_0), f_A(t_2,t_0))
\end{array}\right\} (A > B)$$

Il suffit de prendre un rpo fondé sur la précédence $>$ définie par $f_A > f_B$ ssi $A > B$. Cette précédence est bien fondée sur l'ensemble fini des atomes libres dans S. Nous en concluons pour la seconde fois :

Théorème 9 *La résolution ordonnée est complète.*

5 Hyperrésolution semi-ordonnée

Nous pouvons tenter de combiner les restrictions de l'hyperrésolution et de la résolution ordonnée, mais le faire naïvement mène à l'échec. L'exemple :

$$A \vee B \quad A \vee \neg B \quad \neg A \vee B \quad \neg A \vee \neg B$$

avec $A < B$ ne permet de résoudre qu'avec le fait $A \vee B$ par hyperrésolution, et uniquement sur le littéral B par la contrainte d'ordre. On ne peut donc déduire que A (avec la règle $A \vee \neg B$) ou $A \vee \neg A$ (avec la règle $\neg A \vee \neg B$). Le fait A ne peut maintenant être utilisé qu'avec la règle $A \vee \neg A$ si nous voulons respecter les contraintes d'ordre, et ceci ne fait que régénérer le fait A. L'hyperrésolution positive ordonnée n'est donc pas complète.

Mais on peut quand même poser une contrainte d'ordre, du moment qu'elle ne porte que sur une des deux prémisses. La règle d'*hyperrésolution positive semi-ordonnée* à laquelle nous aboutissons est la suivante :

$$\frac{\overbrace{C \vee A}^{\text{positive, } A \text{ maximal dans } C} \qquad C' \vee \neg A}{C \vee C'}$$

Pour montrer la complétude, nous procédons comme pour l'hyperrésolution positive. Mais au lieu de partir d'une réfutation quelconque, nous partons d'une réfutation ordonnée. Les règles $(1)-(3)$ peuvent produire des réfutations qui ne sont plus ordonnées, par exemple (1) lorsque $A < B$ détruit la contrainte d'ordre; c'est normal, sinon nous pourrions prouver que l'hyperrésolution ordonnée est complète. Mais la propriété d'être positivement semi-ordonné est, elle, préservée :

Lemme 10 *On dit qu'une dérivation π par résolution est* positivement semi-ordonnée *si et seulement si toute coupure de π :*

$$\frac{C \vee A \quad C' \vee \neg A}{C \vee C'}$$

est telle que A est maximal dans C. (Noter que les dérivations par hyperrésolution positive semi-ordonnée sont exactement les dérivations par hyperrésolution positive qui sont positivement semi-ordonnées.)

Si π se réécrit en π' par l'une des règles $(1) - (3)$, alors π' est positivement semi-ordonnée.

Preuve : Par récurrence structurelle sur π, les seuls cas intéressants étant ceux où π est l'un des rédex possibles.

Règle (1) : comme π est positivement semi-ordonnée, A est maximal dans C, B est maximal dans $C'_1 \vee \neg A$. En particulier : (a) B est maximal dans C'_1, et : (b) $B > A$. Comme A est maximal dans C, (b) implique que B est maximal dans C; et par (a), nous déduisons : (c) B est maximal dans $C \vee C'_1$. Comme A est maximal dans C, et que π_0, π_1, π_2 sont positivement semi-ordonnées parce que π l'est, le côté droit de la règle est donc bien positivement semi-ordonné.

Règle (3) : raisonnement similaire.

Règle (2) : comme π est positivement semi-ordonnée, A est maximal dans C, B est maximal dans C_1, et π_0, π_1, π_2 sont positivement semi-ordonnées : le côté droit de la règle est donc bien positivement semi-ordonné. □

Théorème 11 *L'hyperrésolution positive semi-ordonnée est complète.*

Preuve : Soit S un ensemble de clauses insatisfiable. Par le théorème 9, il existe une réfutation π par résolution ordonnée de S, et donc en particulier par résolution semi-ordonnée. Par le lemme 6, π a une forme normale π' pour la relation de réécriture définie par les règles $(1) - (3)$, qui est une réfutation de S. Par le lemme 5, π' est une réfutation par hyperrésolution positive. Par le lemme 10, π' est positivement semi-ordonnée. □

6 Remarques

6.1 L'hyperrésolution fait plus

Soit C une clause positive (un fait) dérivable à partir de S par résolution. Alors C est aussi dérivable de S par hyperrésolution positive. En une phrase :

L'hyperrésolution positive dérive tous les faits dérivables.

Ceci généralise le corollaire 7, qui était le cas particulier où C était la clause vide. La seule chose à changer dans la démonstration de la section 3 est le lemme 5, qui doit être étendu à toute dérivation π d'une clause positive à partir de S; dans la preuve, la phrase "En particulier, $C_1 \vee C_2$ n'est pas la conclusion de π, qui est la clause vide", conséquence du fait que $C_1 \vee C_2$ n'est pas positive, doit être remplacée par "En particulier, $C_1 \vee C_2$ n'est pas la conclusion de π, qui est par hypothèse une clause positive".

6.2 Hyperrésolution négative

Il n'y a rien de magique à choisir des clauses positives dans l'hyperrésolution. En fait, on peut aussi considérer l'*hyperrésolution négative* :

$$\frac{C \vee A \quad \overbrace{C' \vee \neg A}^{\text{négative}}}{C \vee C'}$$

sachant qu'une clause négative est une clause de la forme $\neg A_1 \vee \ldots \vee \neg A_n$. Une telle clause est la négation d'un *but* $\neg(A_1 \wedge \ldots \wedge A_n)$, et si l'on peut voir l'hyperrésolution positive comme dérivant de nouveaux faits à partir des anciens (démonstration *en avant*), l'hyperrésolution négative peut être vue comme réduisant d'anciens buts à de nouveaux sous-buts (*en arrière*).

L'hyperrésolution négative, et l'hyperrésolution négative semi-ordonnée sont complètes aussi. Considérons en effet la transformation $S \mapsto S^-$ qui remplace dans S tout littéral négatif $\neg A$ par A et tout littéral positif A par $\neg A$. Si S est insatisfiable, alors S^- est aussi insatisfiable : en effet, si on avait $\rho \models S^-$, alors on aurait $\rho' \models S$, où ρ' envoie tout A vers la négation de $\rho(A)$. Donc S^- a une réfutation par hyperrésolution positive (resp. semi-ordonnée). Appliquons la transformation inverse $S \mapsto S^-$ à toute cette réfutation : il est facile de voir que l'on obtient ainsi une réfutation de S d'une part, et qu'il s'agit d'une réfutation par hyperrésolution négative (resp. semi-ordonnée), la transformation transformation en effet les clauses positives en négatives et vice-versa.

6.3 Résolution sémantique

On peut généraliser la remarque 6.2 comme suit. Notons que les clauses positives sont exactement les clauses qui sont rendues fausses par l'affectation $\lambda A \cdot \bot$, et les clauses négatives sont exactement les clauses qui sont rendues fausses par l'affectation $\lambda A \cdot \top$. En général, donnons-nous une affectation ρ_0, et considérons la règle :

$$\frac{C \vee A \quad C' \vee \neg A}{C \vee C'}$$

où l'une des prémisses est contrainte à être fausse dans ρ_0. Cette règle s'appelle la *résolution sémantique*, ou ρ_0*-résolution*.

La résolution sémantique est complète, par un argument similaire à celui utilisé ci-dessus pour l'hyperrésolution négative. Considérons la transformation $S \mapsto S^{\rho_0}$ qui, pour tout A tel que $\rho_0(A) = \top$, transforme A en $\neg A$ et $\neg A$ en A, mais ne change pas les A tels que $\rho_0(A) = \bot$. Si S est insatifiable, alors S^{ρ_0} aussi; en effet, si on avait $\rho \models S^{\rho_0}$, alors l'interprétation ρ' envoyant A vers $\rho(A)$ si $\rho_0(A) = \bot$ et A vers $\overline{\neg}\rho(A)$ si $\rho_0(A) = \top$ serait telle que $\rho' \models S$. Donc S^{ρ_0} a une réfutation par hyperrésolution positive. Appliquons la transformation inverse $S \mapsto S^{\rho_0}$ à toute cette réfutation : il est facile de voir que l'on obtient ainsi une réfutation de S d'une part, et qu'il s'agit d'une réfutation par ρ_0-résolution d'autre part. En effet, toute clause positive $A_1 \vee \ldots \vee A_m \vee B_1 \vee \ldots \vee B_n$ (où sans perte de généralité on a choisi les A_i tels que $\rho_0(A_i) = \top$ et les B_j tels que $\rho_0(B_j) = \bot$) se traduit en $\neg A_1 \vee \ldots \vee \neg A_m \vee B_1 \vee \ldots \vee B_n$, qui est rendue fausse par ρ_0.

De façon similaire, la *résolution sémantique semi-ordonnée*, où l'une des prémisses est contrainte à être fausse dans ρ_0 et la formule de coupure doit être maximale dans l'autre prémisse, est encore complète.

6.4 Résolution ordonnée avec un ordre partiel

Nous avons montré que les restrictions d'ordre étaient complètes lorsque $<$ était un ordre total. En fait, elles fonctionnent toujours même lorsque $<$ est un ordre partiel. En effet, tout ordre partiel $<$ peut être plongé dans un ordre total $<^+$, autrement dit : il existe un ordre total $<^+$ tel que pour tous A, B, $A < B$ implique $A <^+ B$. (Ce résultat nécessite l'axiome du choix dans le cas général; ici, nous n'en avons besoin que sur des ensembles finis d'atomes, et une simple récurrence suffit.) Un littéral A ou $\neg A$ *maximal* dans une clause C est alors défini par : pour aucun B ou $\neg B$ dans C, on n'a $A < B$. Les notions de résolution ordonnée, hyperrésolution semi-ordonnée, etc., s'en trouvent conséquemment modifiées. Tout littéral maximal pour $<$ est alors aussi maximal pour $<^+$, et nous concluons à la complétude en observons que toute résolution ordonnée, hyperrésolution semi-ordonnée, etc., pour l'ordre $<^+$ en est encore une pour l'ordre $<$.

6.5 Tautologies, clauses pures et subsumées, suite

On peut montrer que l'élimination de clauses tautologiques, pures et subsumées est compatible avec tous les raffinements de la résolution présentés ici. Autrement dit, combiner les techniques préserve la complétude. Les preuves sont comme en section 2, en adaptant la notion de dérivation par résolution au cas des dérivations par hyperrésolution, par résolution ordonnée, etc.

Il existe des raffinements de la résolution pour lesquels l'élimination de tautologies, de clauses pures ou subsumées casse la complétude, comme la lock-résolution de Boyer.

6.6 Et la sémantique, dans tout ça ?

Bizarrement, nous avons prouvé que l'hyperrésolution, l'hyperrésolution semi-ordonnée et même la résolution sémantique étaient complètes par des moyens complètement syntaxiques. Par contre, nous avons été capables de montrer la complétude de la résolution ordonnée par un moyen sémantique (par arbres de décision). Il ne semble pas y avoir de démonstration sémantique de la complétude des méthodes précédentes.

6.7 Ensemble de support

Un cas particulier de la résolution sémantique est celui de la restriction dite de l'*ensemble de support* (*set-of-support strategy*). Supposons que l'ensemble de clauses S à réfuter soit de la forme $S_0 \cup S_1$, où S_0 est un ensemble non contradictoire de clauses. Le point est que l'on ne souhaite pas exhiber d'affectation satisfaisant S_0, mais il en existe une, nommons-la ρ_0. Ce cas peut se présenter par exemple lorsque S_0 est un ensemble d'instances d'une théorie modélisant un phénomène réel (l'univers fournissant le modèle ρ_0, mais ρ_0 est en général trop compliqué à décrire).

Si S est contradictoire, alors par la remarque 6.3, il existe une ρ_0-réfutation de S. Dans une telle réfutation sémantique, il est impossible que les deux prémisses d'une coupure soient prises dans S_0, puisque ρ_0 satisfait toutes les clauses de S_0. On peut donc restreindre la règle de résolution à ne résoudre qu'entre deux clauses, dont au plus une vient de S_0, et à rajouter le résolvant à l'ensemble S_1. (S_1 est appelé l'*ensemble de support*.)

Encore une fois, on peut se restreindre à ne résoudre que sur un littéral maximal dans toute clause de S_0. Mais la résolution par ensemble de support peut engendrer beaucoup plus de clauses que la ρ_0-résolution pour peu que l'on connaisse explicitement ρ_0, même avec la restriction ordonnée.

6.8 Résolution linéaire

Un autre raffinement important de la résolution est la *résolution linéaire*. Les transitions de cette forme de résolution sont de la forme :

$$S, C_0 \to S \cup \{C_1\}, C_1$$

où C_0 est une clause de S, C_1 est un résolvant de C_0 avec une clause C de S. C_0 est appelée la *clause centrale*, C est la *clause latérale*, et C_1 est la nouvelle *clause au sommet*. L'idée est que l'on choisit une clause dans l'ensemble initial S_0 de clauses, qui devient la clause au sommet, et on ne s'autorise à fabriquer la prochaine clause au sommet qu'en résolvant la précédente avec soit une clause de S_0 (une clause *input*) soit une clause précédemment générée (une clause *ancêtre*).

La résolution linéaire est complète, même dans sa variante semi-ordonnée (où la formule de coupure doit être maximale dans la clause latérale). Mais elle n'est pas compatible avec la stratégie de subsomption des nouvelles clauses, notamment.