Théorie de la démonstration

Michel Parigot & Paul Rozière

25 mars 2011
(1180 –version provisoire)

Ce polycopié est incomplet.

Dans le chapitre I manque essentiellement une preuve de normalisation de la déduction naturelle, et le théorème du séquent median.

Dans le chapitre II, les paragraphes sur la résolution, la méthode inverse, la forme normale structurelle ne sont pas rédigés, plusieurs preuves et apragraphes à compléter.

Le chapitre III (λ-calcul, déduction naturelle et λ-calcul typé, normalisation forte de la déduction naturelle, système T, logique du second ordre ...) n'est pas rédigé, en dehors de la normalisation forte du système T (fichier à part), et de notes de cours sur la déduction naturelle à plusieurs conclusions (fichier à part).

Chapitre 1

SYSTÈMES DE DEDUCTION

1.1 Les systèmes axiomatiques.

Les systèmes axiomatiques fournissent une formalisation très simple de la logique : on se donne un petit nombre de vérités élémentaires (les axiomes) et de constructions qui préservent la vérité (les règles), l'objectif étant d'atteindre ainsi toutes les vérités (problème de la complétude). Ces systèmes ne visent pas à rendre compte de la notion habituelle de démonstration mais seulement de la démontrabilité : on définit une notion de démonstration artificielle qui coïncide uniquement au plan extensionnel avec la notion uselle de démonstration (elle permet de démontrer les mêmes choses mais pas nécessairement de la même manière). En fait, on considère la logique comme une théorie mathématique particulière qu'il s'agit d'axiomatiser.

Aucun choix particulier pour le axiomes et les règles ne s'impose vraiment (il y en a à peu près autant que de livres de logique). On va donner, pour la logique classique, un système "extrême", qui ne possède qu'une seule règle de déduction, le modus ponens. Pour simplifier, on ne considère comme opérateurs logiques primitifs que $\rightarrow, \neg$ et $\forall$; les autres opérateurs sont définis de la façon usuelle à partir $\rightarrow, \neg$ et $\forall$.

Axiomes.
On prend toutes les *généralisations* des instances des schémas d'axiomes suivants (c'est-à-dire les instances de ces schémas d'axiomes, précédés de quantificateurs universels, de façon à obtenir une formule close) :

(A1) $A \rightarrow (B \rightarrow A)$ [1]
(A2) $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$
(A3) $(\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B)$
(B1) $\forall x A \rightarrow A[t/x]$
(B2) $\forall x(A \rightarrow B) \rightarrow (\forall x A \rightarrow \forall x B)$
(B3) $A \rightarrow \forall x A$ (si x n'a pas d'occurrence libre dans A)

Règle.
On prend le modus ponens : de A et $A \rightarrow B$ on déduit B.

Notion de déduction.
Soient Γ un ensemble de formules et A une formule. Une *déduction* sous les hypothèses Γ est une suite finie $A_0, \ldots, A_n$ de formules telles que pour tout $i \leq n$, l'une des conditions suivantes est réalisée :

(a) A_i est un axiome,
(b) A_i est une hypothèse (i.e. $A_i \in \Gamma$),
(c) A_i est obtenue par modus ponens à partir de A_j et A_k pour des $j, k < i$ (i.e. il existe $j, k < i$ tels que $A_k = A_j \rightarrow A_i$).

Une *déduction* de A sous les hypothèses Γ est une déduction sous les hypothèses Γ dont la dernière formule est A. On dit que A est *déductible* de Γ s'il existe une déduction de A sous les hypothèses Γ (notation $\Gamma \vdash A$) et que A est un *valide* si elle déductible de l'ensemble vide (notation $\vdash A$). Deux formules A et B sont *équivalentes* si A est déductible de B et B déductible de A (notation $A \equiv B$).

Exemple. Déduction de $Ns(s(y))$ sous les hypothèses $\forall x(Nx \rightarrow Ns(x)), Ny$:

1. $\forall x(Nx \rightarrow Ns(x)) \rightarrow (Ny \rightarrow Ns(y))$ instance de B1

1. Par exemple si A et B sont des formules ayant x pour seule variable libre $\forall x(A \rightarrow (B \rightarrow A))$ est un axiome.

2. $\forall x(Nx \to Ns(x))$ hypothèse
3. $Ny \to Ns(y)$ modus ponens 1/2
4. Ny hypothèse
5. $Ns(y)$ modus ponens 3/4
6. $\forall x(Nx \to Ns(x)) \to (Ns(y) \to Ns(s(y)))$ instance de B1
7. $Ns(y) \to Ns(s(y))$ modus ponens 6/2
8. $Ns(s(y))$ modus ponens 7/5

Les déductions se représentent naturellement sous forme d'arbres : la racine représente la conclusion, les feuilles représentent les hypothèses et les instances des axiomes utilisées, et les noeuds intermédiaires représentent les applications du modus ponens, écrit sous la forme

$$\frac{A \to B \quad A}{B}$$

Ainsi la démonstration de l'exemple précédent peut elle être représentée sous forme de l'arbre suivant :

$$\dfrac{\dfrac{\forall x(Nx \to Ns(x)) \to (Ns(y) \to Ns(s(y))) \quad \forall x(Nx \to Ns(x))}{Ns(y) \to Ns(s(y))} \quad \dfrac{\dfrac{\forall x(Nx \to Ns(x)) \to (Ny \to Ns(y)) \quad \forall x(Nx \to Ns(x))}{Ny \to Ns(y)} \qquad Ny}{Ns(y)}}{Ns(s(y))}$$

On va montrer en trois exemples quelques "défauts" apparents des systèmes axiomatiques du point de vue de la théorie de la démonstration.

Exemple 1 (le sens des démonstrations)

Voyons comment on démontre une vérité élémentaire comme $A \to A$ dans ce système.

1. $(A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))$ instance de A2
2. $A \to ((A \to A) \to A)$ instance de A1
3. $(A \to (A \to A)) \to (A \to A)$ modus ponens 1/2
4. $A \to (A \to A)$ instance de A1
5. $A \to A$ modus ponens 3/4

La même démonstration devient peut-être un plus lisible sous forme d'arbre, mais elle demeure aussi peu intelligible :

$$\dfrac{\dfrac{(A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A)) \quad A \to ((A \to A) \to A)}{(A \to (A \to A)) \to (A \to A)} \qquad A \to (A \to A)}{A \to A}$$

Commentaire.
On a le sentiment que cette démonstration est arbitraire, qu'elle ne nous dit rien sur $A \to A$. Dans ce système les démonstrations ne donnent pas de renseignement direct sur ce qui est démontré : elle ne remplissent pas l'une des fonctions qu'on attribue généralement aux démonstrations, à savoir nous expliquer "pourquoi" ce qu'on a démontré est vrai. De plus il

semble difficile, de prime abord, de prendre les démonstrations comme objet d'étude pour faire une "théorie des démonstrations" : comment définir par exemple une notion de démonstration intrinsèque (i.e. qui n'utilise pas de moyens "extérieurs" à ce qui est démontré) ?

Exemple 2 (la formalisation du raisonnement)

Le théorème de la déduction : *si* $\Gamma, A \vdash B$, *alors* $\Gamma \vdash A \to B$, se démontre par récurrence sur la longueur de la déduction de B sous les hypothèses Γ, A comme suit :
(a) B est un axiome ou $B \in \Gamma$.
On a une déduction de $A \to B$ sous les hypothèses Γ

$$\frac{B \to (A \to B) \quad B}{A \to B}$$

(b) $B = A$.
On a montré dans l'exemple précédent que $A \to A$ est un théorème.
(c) B est obtenu par modus ponens à partir de C et $C \to B$.
Les formules C et $C \to B$ étant déductibles de Γ et A, on a par hypothèse de récurrence des déductions d_1 de $A \to C$ et d_2 de $A \to (C \to B)$ sous les hypothèses Γ. On forme une déduction de $A \to B$ sous les hypothèses Γ comme suit,

$$\dfrac{\dfrac{(A \to (C \to B)) \to ((A \to C) \to (A \to B)) \quad A \to (C \to B)}{(A \to C) \to (A \to B)} \qquad \begin{array}{c}\vdots \\ d_1 \\ \vdots \\ A \to C\end{array}}{A \to B} \qquad \begin{array}{c}\vdots \\ d_2 \\ \vdots\end{array}$$

Commentaire.

Le théorème de la déduction qui est une figure élémentaire du raisonnement : *pour démontrer* $A \to B$, *on démontre* B *sous l'hypothèse* A, apparait ici uniquement comme un métathéorème. Ce système ne rend donc pas compte de la notion usuelle de démonstration. Il considère la logique comme une théorie mathématique particulière : un corps de vérités logiques axiomatisé, et non comme une formalisation des lois du raisonnement.

Exemple 3 (la recherche des démonstrations)

Une démonstration se présente sous forme d'un arbre dont tous les noeuds qui ne sont pas des feuilles sont des applications du modus ponens. Mais quand on cherche une démonstration d'un théorème, ce qu'on a c'est la racine de l'arbre, l'arbre étant à construire en partant de cette racine. Dans ce processus chaque étape de la démonstration représente une invention. En effet dans les applications du modus ponens

$$\frac{A \to B \quad A}{B}$$

on possède le B et on doit inventer un A tel que A et $A \to B$ soient eux-mêmes des théorèmes.

En fait cette méthode de démonstration est impraticable et d'ailleurs la première chose qu'on fait en pratique c'est de prouver des "règles dérivées" – comme le théorème de la déduction – pour quitter ce système et retrouver le mode de raisonnement habituel.

Exercice 1 Prouver la règle de généralisation : si $\Gamma \vdash A$ et x est une variable qui n'a pas d'occurrence libre dans les formules de Γ, alors $\Gamma \vdash \forall x A$. [1]

On ne peut terminer cette revue critique (et un peu superficielle) des systèmes axiomatiques sans en évoquer brièvement quelques avantages. Le plus évident est l'extrême simplicité de la notion de démonstration (à comparer avec celles que fournissent les autres systèmes présentés dans ce chapître) qui en fait un outil métamathématique intéressant. Cette simplicité s'explique principalement par une propriété que les autres systèmes n'ont pas : à aucun moment de la construction d'une déduction, on n'est amené à travailler sur les hypothèses de la déduction (en particulier les hypothèses d'une "sous-déduction" sont toujours des hypothèses de la déduction elle-même). Cette propriété, qui pourrait passer du point de vue logique pour une bizarrerie (c'est elle en particulier qui empêche d'écrire les preuves "naturellement"), se trouve être aussi particulièrement intéressante du point de vue algorithmique : elle donne en effet une façon originale de calculer sans recours à des variables. [2]

1.2 La déduction naturelle.

Dans ce chapître Γ et Γ' dénotent des suites non ordonnées [3] ; on désigne par Γ, Γ' la concaténation des suites Γ et Γ' et par Γ, A l'adjonction de la formule A à Γ ; on confond la suite réduite à la formule A avec la formule A elle-même.

1.2.1 Les figures élémentaires du raisonnement.

La déduction naturelle est essentiellement une codification de la pratique de la déduction en mathématiques : elle décompose le raisonnement en étapes élémentaires qui correspondent au maniement des différents symboles logiques. Les étapes élémentaires sont représentées par des règles de la forme

$$\frac{A_1 \ldots . A_n}{B}$$

permettant de déduire une conclusion B à partir de prémisses $A_1, \ldots, A_n$.

A chaque symbole logique correspondent deux groupes de règles ayant un rôle symétrique :

(i) les règles d'*introduction* qui permettent de "construire" des énoncés : par exemple "de A et B on conclut $A \wedge B$", qui a A et B pour prémisses et $A \wedge B$ pour conclusion.

(ii) les règles d'*élimination* qui permettent d'"utiliser" des énoncés : par exemple, "de $A \wedge B$ on conclut A", qui a $A \wedge B$ pour prémisse et A pour conclusion.

Commençons par recenser les différentes figures du raisonnement associées usuellement aux symboles logiques.

Conjonction :
- *pour démontrer $A \wedge B$, on démontre A et on démontre B ;*
- *de $A \wedge B$ on conclut A et on conclut B.*

Implication :
- *pour démontrer $A \to B$, on démontre B sous l'hypothèse A ;*
- *de $A \to B$ et A, on conclut B.*

Négation :
- *pour démontrer $\neg A$, on démontre une contradiction sous l'hypothèse A ;*
- *de A et $\neg A$, on conclut une contradiction.*

Quantification universelle :
- *pour démontrer $\forall x A$, on démontre $A[y/x]$ pour un y arbitraire* (y arbitraire signifie qu'on ne fait aucune hypothèse sur y ; formellement cela se traduit par la restriction : y n'a pas d'occurrence libre dans les hypothèses et A) ;
- *de $\forall x A$, on conclut $A[t/x]$ pour n'importe quel terme t.*

Disjonction :
- *pour démontrer $A \vee B$, on démontre A ou on démontre B ;*
- *de $A \vee B$ on conclut C dès lors qu'on possède des démonstrations de C sous l'hypothèse A et de C sous l'hypothèse B.*

Quantification existentielle : [4]
- *pour démontrer $\exists x A$, on démontre $A[t/x]$ pour un certain terme t ;*
- *de $\exists x A$ on conclut C, dès lors qu'on possède une démonstration de C sous l'hypothèse $A[y/x]$, pour un y arbitraire* (ici y arbitraire signifie que ni C, ni A, ni les hypothèses de la démonstration de C autres que $A[y/x]$ ne dépendent de y).

1.2.2 Systèmes de déduction naturelle.

Il y a plusieurs façons équivalentes de formaliser les figures élémentaires du raisonnement qu'on vient de recenser. On peut soit considérer qu'on raisonne directement sur des *formules*, soit qu'on se place à un niveau au dessus et qu'on raisonne sur des *séquents*, i.e. des expressions de la forme $\Gamma \vdash A$ (A est démontrable sous les hypothèses Γ). La première façon est assurément la plus satisfaisante, mais la seconde permet de définir plus facilement la notion formelle de démonstration, et c'est donc elle qu'on va considérer en premier.

a) Déduction de séquents

Le système a pour axiomes les séquents $A \vdash A$ et possède deux sortes de règles : des règles logiques (règles d'introduction et d'élimination pour chaque connecteur) et des règles structurelles qui permettent de manipuler les hypothèses.

Les **règles logiques** correspondent exactement aux figures élémentaires du raisonnement recensées précédemmment. Ainsi, dans le cas de l'implication, on écrira les deux règles suivantes :

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B} \to_i \qquad\qquad \frac{\Gamma \vdash A \to B \quad \Gamma' \vdash A}{\Gamma, \Gamma' \vdash B} \to_e$$

Avec la règle $\to_i$ apparaît une propriété importante de la déduction naturelle, qui n'existait pas dans les systèmes axiomatiques : l'application d'une règle peut faire disparaître des hypothèses. De plus pour que les système soit complet, il faut admettre que la règle puisse s'appliquer à une hypothèse qui n'apparaît pas aussi bien qu'à plusieurs occurrences de la même hypothèse.

C'est pour cela que le système comporte des **règles structurelles** qui permettent de décrire précisément la "gestion" des hypothèses : l'affaiblissement permet d'ajouter de nouvelles hypothèses, et la contraction permet de confondre deux occurrences différentes d'une même hypothèse.

L'ensemble des règles est donné dans le tableau suivant. La formalisation des règles pour la négation nécessite l'usage d'une constante de proposition $\bot$ (appellée "faux") représentant la contradiction.

AXIOME

$$\frac{}{A \vdash A}\ \text{ax.}$$

REGLES LOGIQUES

$$\frac{\Gamma \vdash A \quad \Gamma' \vdash B}{\Gamma, \Gamma' \vdash A \wedge B}\ \wedge_i \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\ \wedge_{eg} \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\ \wedge_{ed}$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\ \to_i \qquad \frac{\Gamma \vdash A \to B \quad \Gamma' \vdash A}{\Gamma, \Gamma' \vdash B}\ \to_e$$

$$\frac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}\ \neg_i \qquad \frac{\Gamma \vdash \neg A \quad \Gamma' \vdash A}{\Gamma, \Gamma' \vdash \bot}\ \neg_e$$

$$\frac{\Gamma \vdash A[y/x]}{\Gamma \vdash \forall x A}\ \forall_i\ (*) \qquad \frac{\Gamma \vdash \forall x A}{\Gamma \vdash A[t/x]}\ \forall_e$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ \vee_{ig} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ \vee_{id} \qquad \frac{\Gamma \vdash A \vee B \quad \Gamma', A \vdash C \quad \Gamma'', B \vdash C}{\Gamma, \Gamma', \Gamma'' \vdash C}\ \vee_e$$

$$\frac{\Gamma \vdash A[t/x]}{\Gamma \vdash \exists x A}\ \exists_i \qquad \frac{\Gamma \vdash \exists x A \quad \Gamma', A[y/x] \vdash C}{\Gamma, \Gamma' \vdash C}\ \exists_e\ (**)$$

(*) La règle $\forall_i$ ne vaut que si y n'a pas d'occurrences libres dans Γ, A.
(**) La règle $\exists_e$ ne vaut que si y n'a pas d'occurrences libres dans Γ', A, C.

REGLES STRUCTURELLES

$$\frac{\Gamma \vdash B}{\Gamma, A \vdash B}\ w \qquad \frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B}\ c$$

TABLE 1.1 – Règles de déduction naturelle

Commentaire. Les règles structurelles sont très importantes au plan algorithmique, si l'on pense qu'une démonstration de $\Gamma \vdash A$ est un algorithme dont les entrées sont Γ et la sortie A. L'affaiblissement représente le cas d'une entrée qu'on n'utilise pas (effacement) et la contraction, celui d'une entrée qu'on utilise plusieurs fois (duplication). On aurait pu considérer une autre règle structurelle, appelée échange, qui permet de changer l'ordre des hypothèses (cela suppose bien sûr que les hypothèses soient organisées en suites ordonnées). Mais son ajout augmente grandement la complexité syntaxique du système sans présenter d'intérêt évident.

Dès lors qu'on ne s'intéresse pas au contenu algorithmique des démonstrations, mais seulement à ce qu'elles démontrent, on peut considérer que les contractions sont faites systématiquement ; autrement dit, on peut considérer que les hypothèses forment un ensemble (on verra au chapître III que cette vision ensembliste ne convient pas du point de vue algorithmique [5]).

Notion de déduction.
Une *déduction* dans ce système est une suite finie $S_0, ..., S_n$ de séquents tels que pour tout $i \leq n$, l'une des conditions suivantes est réalisée :

(a) S_i est un axiome,

(b) S_i est obtenue par application d'une des règles à des séquents S_j pour des $j < i$.

Une formule A est *déductible* de Γ, s'il existe une déduction qui se termine par le séquent $\Gamma \vdash A$ (notation $\Gamma \vdash_m A$). Une formule est un *théorème* si elle est déductible de la suite vide (notation $\vdash_m A$).
Comme dans le cas du système axiomatique, les déductions se représentent naturellement sous forme d'arbres.

Conventions.
(i) Dans les règles $\forall i$ et $\exists e$, la variable y s'appelle le *paramètre propre* de la règle. On supposera toujours qu'un paramètre propre n'apparaît jamais hors de la sous-déduction qui se termine par la règle dont il est le paramètre propre.
(ii) On supposera que la négation n'est pas un symbole primitif, et que $\neg A$ est défini comme $A \rightarrow \bot$. Les règles $\neg i$ et $\neg e$ deviennent alors des cas particuliers de $\rightarrow i$ et $\rightarrow e$. Inversement si on prend $\neg$ comme symbole primitif, alors on peut démontrer l'équivalence de $\neg A$ et $A \rightarrow \bot$.
(iii) On appelle *formule principale* d'une règle, la formule qui contient le connecteur qu'on introduit ou élimine par cette règle (la formule principale d'une introduction apparaît donc dans la conclusion et la formule principale d'une élimination, dans l'une des prémisses).
(iv) Dans les dérivations concrètes, les règles structurelles ne sont pas écrites explicitement : elles sont "intégrées" dans les règles logiques ; en outre les barres des axiomes sont omises.

Exemples.
L'exemple (a) contient un affaiblissement sur B et l'exemple (b) une contraction sur A ; dans l'exemple (c), on a pris comme paramètre de la règle $\forall i$ la variable x elle-même, ce qu'on peut presque toujours faire.

(a) $A \rightarrow (B \rightarrow A)$

$$\dfrac{\dfrac{A \vdash A}{A \vdash B \rightarrow A}\, w+\rightarrow_i}{\vdash A \rightarrow (B \rightarrow A)}\, \rightarrow_i$$

(b) $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$

$$\dfrac{\dfrac{\dfrac{\dfrac{A \to (B \to C) \vdash A \to (B \to C) \quad A \vdash A}{A \to (B \to C), A \vdash B \to C} \to_e \quad \dfrac{A \to B \vdash A \to B \quad A \vdash A}{A \to B, A \vdash B} \to_e}{A \to (B \to C), A \to B, A \vdash C} c + \to_e}{A \to (B \to C), A \to B \vdash A \to C} \to_i}{A \to (B \to C) \vdash (A \to B) \to (A \to C)} \to_i}{\vdash (A \to (B \to C)) \to ((A \to B) \to (A \to C))} \to_i$$

(c) $\neg\exists x A \to \forall x \neg A$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{A \vdash \exists x A} \exists_i}{\neg\exists x A \vdash \neg\exists x A} \neg_e}{\neg\exists x A, A \vdash \bot} \neg_i}{\neg\exists x A \vdash \neg A} \forall_i}{\neg\exists x A \vdash \forall x \neg A} \to_i}{\vdash \neg\exists x A \to \forall x \neg A}$$

Remarque. Sans les conditions sur le paramètre y, les règles $\forall_i$ et $\exists_e$ deviennent fausses : on peut par exemple démontrer $\exists x Px \to \forall y Py$ et $\exists x \forall y((Px \to Py) \wedge (Py \to Px))$.
Commentaire. L'utilisation de règles structurelles constitue ce qu'on pourrait appeler une gestion "logique" des hypothèses, mais on pourrait aussi imaginer une gestion "naturelle" qui, au lieu de permettre des affaiblissements et des contractions à des endroits arbitraires, les limiterait à leur place "naturelle" : on ajoute une formule quand on en a besoin, et on confond deux occurrences à contracter dès qu'elles se rencontrent. Pour cela on n'a pas besoin de règles explicites, mais de conventions :
– Les règles qui font disparaître des hypothèses ($\to i$, $\neg i$, $\vee e$ et $\exists e$) sont applicables même dans le cas où ces hypothèses ne sont pas présentes.
– On "marque" les hypothèses (deux occurrences recoivent la même marque si et seulement si on veut les contracter) et les hypothèses marquées sont gérées comme des ensembles : ainsi deux occurrences de la même formule ayant la même marque sont confondues dès qu'elles apparaissent dans le même séquent.

La gestion logique est celle qui convient le mieux quand on prouve des théorèmes sur le système, et la gestion naturelle quand on utilise le système du point de vue algorithmique. En fait, il n'y guère de raisons d'utiliser la gestion naturelle avec la déduction de séquents, car cela correspond exactement à la déduction de formules.

b) Déduction de formules

Au lieu de manipuler des séquents, on construit directement des déductions à partir d'hypothèses en raisonnant sur les formules. L'étape de base, qui dans la version précédente était l'axiome $A \vdash A$, devient la déduction de A sous l'hypothèse A qu'on écrit simplement A. Ensuite on construit des déductions complexes en appliquant des règles logiques, qui sont par exemple pour la conjonction :

$$\dfrac{A \quad B}{A \wedge B} \wedge_i \qquad \dfrac{A \wedge B}{A} \wedge_{eg} \qquad \dfrac{A \wedge B}{A} \wedge_{ed}$$

L'introduction de la conjonction permet ainsi de construire une déduction de $A \wedge B$ à partir de déductions de A et B (les hypothèses de la déduction résultante étant exactement celles des déductions initiales). Plus généralement, les déductions se présentent sous forme d'arbres : les feuilles de l'arbre représentent les hypothèses, la racine de l'arbre représente la conclusion et les noeuds intermédiaires représentent la conclusion d'une règle dont les prémisses sont les noeuds immédiatement au dessus dans l'arbre.

Cette construction simple ne vaut que pour les règles qui ne font pas disparaître d'hypothèses. Dans le cas général, on doit pouvoir *mutifier* des hypothèses : une hypothèse d'une déduction initiale disparait dans la déduction résultante. Considérons par exemple le cas de l'introduction de l'implication : on doit passer d'une déduction de B sous l'hypothèse A à une déduction de $A \to B$ sans hypothèse ; on le fera en mutifiant A dans la déduction de B sous l'hypothèse A (formellement on remplace les occurrences de A par $[\![A]\!]$). On adoptera la représentation suivante [6]

$$\begin{array}{c} A \\ \vdots \\ d \\ \vdots \\ B \end{array} \qquad\qquad \begin{array}{c} [\![A]\!] \\ \vdots \\ d \\ \vdots \\ \dfrac{B}{A \to B} \end{array}$$

déduction d de B sous l'hypothèse A

déduction de $A \to B$ où l'hypothèse A de d a été mutifiée

Cependant l'opération de mutification de A, pour être adéquate, doit pouvoir ne s'appliquer qu'à certaines occurrences de l'hypothèse A dans la déduction (et même éventuellement aucune). On a donc besoin d'une version plus sophistiquée de cette opération, permettant de préciser quelles occurrences de la formule sont concernées par une mutification particulière (puisque des occurrences différentes de la même formule peuvent être mutifiées à des moments différents). On peut le faire en attachant un même entier à ces occurrences et à la règle appliquée :

$$\begin{array}{c} [\![A]\!]^n \\ \vdots \\ d \\ \vdots \\ \dfrac{B}{A \to B} \to_i n \end{array}$$

Cela revient à regrouper les occurrences d'une formule par paquets et à les lier dans les déductions comme on lie des variables dans une formule. Bien sûr le choix de l'entier n n'a aucune importance, celui-ci n'étant qu'un lieur. Lorsqu'on manipule des démonstrations, ces liaisons de variables posent les problèmes habituels de capture de variables (quand une occurrence libre rentre malencontreusement dans le champ d'un lieur).

Avec les conventions de notation précédentes, les règles de déduction naturelle deviennent les suivantes : Une déduction de A sous les hypothèses Γ devient dans ce cadre un arbre dont la conclusion est A et dont les hypothèses non mutifiées sont toutes dans Γ. Il faut noter que la définition de la notion de déduction est plus difficile qu'avec les séquents.

$$\frac{A \quad B}{A \wedge B}\ \wedge_i \qquad \frac{A \wedge B}{A}\ \wedge_{eg} \qquad \frac{A \wedge B}{B}\ \wedge_{ed}$$

$$\frac{\begin{array}{c}[\![A]\!]^n \\ \vdots\, d \,\vdots \\ B\end{array}}{A \to B}\ \to_i n \qquad \frac{A \to B \quad A}{B}\ \to_e$$

$$\frac{\begin{array}{c}[\![A]\!]^n \\ \vdots\, d \,\vdots \\ \bot\end{array}}{\neg A}\ \neg_i n \qquad \frac{\neg A \quad A}{\bot}\ \neg_e$$

$$\frac{A[y/x]}{\forall x A}\ \forall_i\ (*) \qquad \frac{\forall x A}{A[t/x]}\ \forall_e$$

$$\frac{A}{A \vee B}\ \vee_{ig} \qquad \frac{B}{A \vee B}\ \vee_{id} \qquad \frac{A \vee B \quad \begin{array}{c}[\![A]\!]^n \\ \vdots\, d_1 \,\vdots \\ C\end{array} \quad \begin{array}{c}[\![B]\!]^n \\ \vdots\, d_2 \,\vdots \\ C\end{array}}{C}\ \vee_e n$$

$$\frac{A[t/x]}{\exists x A}\ \exists_i \qquad \frac{\exists x A \quad \begin{array}{c}[\![A[y/x]]\!]^n \\ \vdots\, d \,\vdots \\ C\end{array}}{C}\ \exists_e\ n\ (*)$$

(*) La règle $\forall_i$ (resp. $\exists_e$) ne vaut que si y n'a pas d'occurrences libres dans A et les hypothèses (resp. A, C et les hypothèses de d autres que $A[y/x]$).

TABLE 1.2 – Règles de déduction naturelle – déduction de formules

Exercice 2 Définir formellement la notion de déduction dans ce système.

Exemple. (on pourra remarquer que les démonstrations sont un peu moins lourdes dans cette version de la déduction naturelle que dans la précédente)
(a) $A \to (B \to A)$

$$\dfrac{\dfrac{[\![A]\!]^2}{B \to A}\ {\scriptstyle \to_i 1}}{A \to (B \to A)}\ {\scriptstyle \to_i 2}$$

(b) $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$

$$\dfrac{\dfrac{\dfrac{\dfrac{[\![A \to (B \to C)]\!]^3 \quad [\![A]\!]^1}{B \to C}\ {\scriptstyle \to_e} \quad \dfrac{[\![A \to B]\!]^2 \quad [\![A]\!]^1}{B}\ {\scriptstyle \to_e}}{C}\ {\scriptstyle \to_e}}{A \to C}\ {\scriptstyle \to_i 1}}{(A \to B) \to (A \to C)}\ {\scriptstyle \to_i 2}}{(A \to (B \to C)) \to ((A \to B) \to (A \to C))}\ {\scriptstyle \to_i 3}$$

(c) $\neg\exists x A \to \forall x \neg A$

$$\dfrac{\dfrac{\dfrac{[\![\neg\exists x A]\!]^2 \quad \dfrac{[\![A]\!]^1}{\exists x A}\ {\scriptstyle \exists_i}}{\bot}\ {\scriptstyle \exists_e}}{\neg A}\ {\scriptstyle \neg_i 1}}{\forall x \neg A}\ {\scriptstyle \forall_i}}{\neg\exists x A \to \forall x \neg A}\ {\scriptstyle \to_i 2}$$

Commentaire. On voit apparaître sur ces exemples une stratégie de recherche de démonstrations extrêmement simple : partant de la conclusion, on détruit le connecteur principal (i.e. on considère que la règle précédente était une introduction) tant que cela est possible, puis on essaie d'obtenir la formule restante à partir des hypothèses en appliquant des éliminations. Cette stratégie donne une assez bonne idée de la manière de rechercher des démonstrations en déduction naturelle, au moins quand on ne considère que des formules avec $\to, \wedge$ et $\forall$. Comme le montre l'exemple suivant, la situation est différente quand le connecteur principal de la formule est un $\vee$: dans ce cas, il est préférable de ne pas détruire le connecteur principal (la raison est claire : il se peut très bien qu'on puisse prouver $\Gamma \vdash A \vee B$ sans qu'on puisse prouver ni $\Gamma \vdash A$ ni $\Gamma \vdash B$; en revanche prouver $\Gamma \vdash A \wedge B$ revient exactement à prouver $\Gamma \vdash A$ et $\Gamma \vdash B$).
Exemple.

$$\dfrac{\dfrac{[\![(A \wedge B) \vee C]\!]^3 \quad \dfrac{\dfrac{[\![A \wedge B]\!]^2}{A}\ {\scriptstyle \wedge_{eg}}}{A \vee C}\ {\scriptstyle \vee_{ig}} \quad \dfrac{[\![C]\!]^2}{A \vee C}\ {\scriptstyle \vee_{id}}}{A \vee C}\ {\scriptstyle \vee_e 2} \quad \dfrac{[\![(A \wedge B) \vee C]\!]^3 \quad \dfrac{\dfrac{[\![A \wedge B]\!]^1}{B}\ {\scriptstyle \wedge_{ed}}}{B \vee C}\ {\scriptstyle \vee_{ig}} \quad \dfrac{[\![C]\!]^1}{B \vee C}\ {\scriptstyle \vee_{ig}}}{B \vee C}\ {\scriptstyle \vee_e 1}}{(A \vee C) \wedge (B \vee C)}\ {\scriptstyle \wedge_i}}{((A \wedge B) \vee C) \to ((A \vee C) \wedge (B \vee C))}\ {\scriptstyle \to_i 3}$$

Exercice 3 Montrer
$\vdash_m \ A \to \neg\neg A$
$\vdash_m (A \to (B \to C)) \to ((A \wedge B) \to C)$
$\vdash_m \forall x(A \wedge B) \to (\forall xA \wedge \forall xB)$
$\vdash_m \forall x(A \to B) \to (A \to \forall xB)$ (si x n'a pas d'occurrence libre dans A)
$\vdash_m \exists x(A \vee B) \to (\exists xA \vee \exists xB)$

1.2.3 Le raisonnement non constructif

Les règles de déduction naturelle que nous avons vues jusqu'à maintenant ne rendent pas compte de tout le raisonnement mathématique. Ce sont les règles de la **logique minimale**. Les symboles $\vdash_m$ et $\equiv_m$ introduits au §2.2 désignent respectivement les relations de déductibilité et d'équivalence entre formules en logique minimale.

En fait les règles de la logique minimale ne disent rien sur la constante $\bot$ qui de ce fait ne se distingue pas d'une lettre de formule quelconque, comme le montre la proposition suivante.

Proposition 1.2.1 *Soient $A_1, ..., A_n, A, B$ des formules. Alors*

$$\textit{si } A_1, ..., A_n \vdash_m A, \textit{ alors } A_1[B/\ \bot], ..., A_n[B/\ \bot] \vdash_m A[B/\ \bot].$$

Démonstration. Récurrence immédiate sur la construction de la déduction $A_1, ..., A_n \vdash_m A$.

Pour pouvoir démontrer toutes les vérités logiques, on doit ajouter certaines règles pour le maniement de $\bot$.

En ajoutant la règle d'absurdité intuitionniste $\bot_e$

$$\frac{\bot}{A}\ \bot_e \quad (\text{ou } \frac{\Gamma \vdash \bot}{\Gamma \vdash A}\ \bot_e \text{ dans la déduction de séquents})$$

on obtient la **logique intuitionniste**. On définit de façon évidente les notions de déduction et de déductibilité intuitionnistes (on utilise les symboles $\vdash_i$ et $\equiv_i$ pour la déductibilité et l'équivalence en logique intuitionniste). La règle $\bot_i$ n'est pas démontrable en logique minimale : sinon, comme en logique minimale $\bot$ joue le rôle d'une formule quelconque, on pourrait tout aussi bien démontrer $\neg \bot \to A$ et donc A, pour une formule A arbitraire ! Voici un exemple concret où la règle $\bot_i$ est nécessaire : $(\neg\neg A \to \neg\neg B) \to \neg\neg(A \to B)$ est une formule démontrable en logique intuitionniste mais pas en logique minimale (cf remarque 2.4. ? [7]).
On considère généralement que l'absurdité intuitionniste est un raisonnement constructif, bien que son caractère constructif soit difficile à expliciter.
Exemple. La formule $\neg\neg(\neg\neg A \to A)$ est démontrable en logique intuitionniste mais pas en

logique minimale. Voici la démonstration en logique intuitionniste.

$$\dfrac{\dfrac{[\![\neg(\neg\neg A\to A)]\!]^3 \qquad \dfrac{\dfrac{\dfrac{[\![\neg\neg A]\!]^2 \qquad \dfrac{\dfrac{[\![\neg(\neg\neg A\to A)]\!]^3 \qquad \dfrac{[\![A]\!]^1}{\neg\neg A\to A}\,{\scriptstyle \to_i}}{\bot}\,{\scriptstyle \neg_e}}{\neg A}\,{\scriptstyle \neg_i\ 1}}{\bot}\,{\scriptstyle \neg_e}}{A}\,{\scriptstyle \bot_e}}{\neg\neg A\to A}\,{\scriptstyle \to_i\ 2}}{\bot}\,{\scriptstyle \neg_e}}{\neg\neg(\neg\neg A\to A)}\,{\scriptstyle \neg_i\ 3}$$

En ajoutant la règle d'absurdité classique $\bot_c$

$$\begin{array}{c}[\![\neg A]\!]\\ \vdots\\ \dfrac{\bot}{A}\,{\scriptstyle\bot_c}\end{array} \qquad \left(\text{ou } \dfrac{\Gamma,\neg A\vdash\bot}{\Gamma\vdash A}\,{\scriptstyle\bot_c}\ \text{ dans la déduction de séquents}\right)$$

on obtient la **logique classique**. On définit de façon évidente les notions de déduction et de déductibilité classiques (on utilise les symboles $\vdash_c$ et $\equiv_c$ pour la déductibilité et l'équivalence en logique classique). On remarque que $\bot_i$ se déduit de $\bot_c$ (en fait c'est le cas particulier où l'hypothèse $\neg A$ n'apparait pas) :

$$\dfrac{\dfrac{\Gamma\vdash\bot}{\Gamma,\neg A\vdash\bot}\,{\scriptstyle w}}{\Gamma\vdash A}\,{\scriptstyle\bot_c}$$

On montrera plus tard que $\bot_c$ n'est pas démontrable en logique intuitionniste. Le tiers-exclu $A\vee\neg A$ fournit un exemple concret de formule démontrable en logique classique mais pas en logique intuitionniste.

Remarque. En logique intuitionniste tous les connecteurs sont indépendants les uns des autres (cf chapître II). En revanche, en logique classique on peut tout définir à partir de $\neg$, $\wedge$ et $\forall$, par exemple.

Exemple. La règle $\bot_c$ permet de démontrer $\neg\forall xA\to\exists x\neg A$

$$\dfrac{\dfrac{\dfrac{[\![\neg\forall xA]\!]^3 \qquad \dfrac{\dfrac{\dfrac{[\![\neg\exists x\neg A]\!]^2 \qquad \dfrac{[\![\neg A]\!]^1}{\exists x\neg A}\,{\scriptstyle\exists_i}}{\bot}\,{\scriptstyle\neg_e}}{A}\,{\scriptstyle\bot_c\ 1}}{\forall xA}\,{\scriptstyle\forall_i}}{\bot}\,{\scriptstyle\neg_e}}{\exists x\neg A}\,{\scriptstyle\bot_c\ 2}}{\neg\forall xA\to\exists x\neg A}\,{\scriptstyle\to_i\ 3}$$

Exercice 4 Montrer

$\vdash_c (A \vee B) \leftrightarrow \neg(\neg A \wedge \neg B)$

$\vdash_c \exists x A \leftrightarrow \neg\forall x \neg A$

$\vdash_m \neg(A \vee B) \leftrightarrow (\neg A \vee \neg B)$

$\vdash_m \neg\exists x A \leftrightarrow \forall x \neg A$

Exercice 5 (i) Montrer qu'en logique intuitionniste, on peut restreindre la règle $\perp_i$ au cas où A est atomique.
(ii) Montrer qu'en logique classique avec comme seuls opérateurs primitifs $\wedge, \rightarrow, \forall$ et $\perp$, on peut restreindre la règle $\perp_c$ au cas où A est atomique. Pour se convaincre du fait que ça ne marche pas en général, on pourra considérer l'exemple de $A \vee \neg A$.

Exercice 6 Montrer l'équivalence de la déduction naturelle classique avec le système axiomatique présenté au début.

Commentaire. La logique classique et la logique intuitionniste n'ont pas les mêmes prétentions : la logique classique rend compte de la vérité ($\vdash_c \ A$ est à interpréter comme "A est vrai") alors que la logique intuitionniste rend compte de l'accès qu'on a à la vérité ($\vdash_i \ A$ est à interpréter comme "je sais que A "). Voici un exemple bien connu de démonstration en logique classique d'un énoncé, par ailleurs démontrable en logique intuitionniste, qui donne une bonne idée de la différence. Considérons l'énoncé "il existe a, b nombres irrationnels tels que a^b soit rationnel". Voici une démonstration utilisant le tiers-exclu (on suppose connu le fait que $\sqrt{2}$ est irrationnel) : si $\sqrt{2}^{\sqrt{2}}$ est rationnel on prend $a = \sqrt{2}$ et $b = \sqrt{2}$; sinon on prend $a = \sqrt{2}^{\sqrt{2}}$ et $b = \sqrt{2}$ et on a $a^b = 2$ qui est rationnel ; dans tous les cas, a^b est bien rationnel. Cette démonstration en logique classique ne permet pas d'exhiber de couple (a, b) ayant la propriété requise, et pour en exhiber un, il faut une démonstration beaucoup plus compliquée.

La prétention de la logique intuitionniste est précisément de formaliser les preuves constructives. On verra au chapître II que la logique intuitionniste a les propriétés de disjonction (si $\vdash_i A \vee B$, alors $\vdash_i A$ ou $\vdash_i B$) et d'existence (si $\vdash_i \exists x A$, alors $\vdash_i A[t/x]$ pour un certain terme t). On verra au chapître III comment on peut extraire des algorithmes de démonstrations en logique intuitionniste. [8]

Il y a plusieurs façons d'obtenir la logique classique en ajoutant des schémas d'axiomes :

Logique intuitionniste $+ A \vee \neg A$ (tiers exclu)

Logique intuitionniste $+ (\neg A \rightarrow A) \rightarrow A$ (loi de Peirce)

Logique minimale $+ \neg\neg A \rightarrow A$ (élimination des doubles négations)

Logique minimale $+ (\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B)$ (contraposition)

Montrons d'abord chacun de ces schémas en logique classique :

Tiers exclu

$$\dfrac{\dfrac{\dfrac{\dfrac{[\![\neg(A\vee\neg A)]\!]^2 \qquad \dfrac{[\![A]\!]^1}{A\vee\neg A}\vee_{ig}}{\bot}\neg_e}{\neg A}\neg_i\,1}{A\vee\neg A}\vee_{id}}{\bot}\neg_e \qquad [\![\neg(A\vee\neg A)]\!]^2}{A\vee\neg A}\perp_c 2$$

Loi de peirce

$$\dfrac{\dfrac{[\![\neg A]\!]^1 \qquad \dfrac{[\![\neg A\to A]\!]^2 \quad [\![\neg A]\!]^1}{A}\to_e}{\bot}\neg_e}{A}\perp_c 1}{(\neg A\to A)\to A}\to_i 2$$

Elimination des doubles négations

$$\dfrac{\dfrac{[\![\neg\neg A]\!]^2 \quad [\![\neg A]\!]^1}{\bot}\neg_e}{A}\perp_c 1}{\neg\neg A\to A}\to_i 2$$

Contraposition

$$\dfrac{\dfrac{\dfrac{[\![\neg B\to\neg A]\!]^3 \qquad [\![\neg B]\!]^1}{\neg A}\to_e \qquad [\![A]\!]^2}{\bot}\neg_e}{B}\perp_c 1}{A\to B}\to_i 2}{(\neg B\to\neg A)\to(A\to B)}\to_i 3$$

Montrons maintenant la règle de l'absurdité classique à partir des autres schémas :

Tiers exclu

$$\dfrac{A\vee\neg A \qquad \dfrac{\begin{array}{c}[\![\neg A]\!]^1\\ \vdots\\ \bot\end{array}}{A}\perp_e \qquad [\![A]\!]^1}{A}\vee_e 1$$

Loi de Peirce

$$\dfrac{(\neg A \to A) \to A \qquad \dfrac{\dfrac{[\![\neg A]\!]^1}{\vdots}}{\dfrac{\dfrac{\bot}{A}\,\bot_e}{\neg A \to A}\,\to_i 1}}{A}\,\to_e$$

Elimination des doubles négations

$$\dfrac{\neg\neg A \to A \qquad \dfrac{\dfrac{[\![\neg A]\!]^1}{\vdots}}{\dfrac{\bot}{\neg\neg A}\,\to_i 1}}{A}\,\to_e$$

Contraposition

$$\dfrac{(\neg A \to \neg\neg\neg A) \to (\neg\neg A \to A) \qquad \dfrac{\dfrac{\dfrac{[\![\neg\neg\neg A]\!]^1 \quad [\![\neg A]\!]^2}{\bot}\,\neg_e}{\neg\neg\neg A}\,\neg_i 1}{\neg A \to \neg\neg\neg A}\,\to_i 2}{\dfrac{\neg\neg A \to A \qquad \dfrac{\dfrac{[\![\neg A]\!]^3}{\vdots}}{\dfrac{\bot}{\neg\neg A}\,\neg_i 3}}{A}\,\to_e}\,\to_e$$

Exercice 7 Montrer
$\vdash_m (A \vee \neg A) \to ((\neg A \to A) \to A)$
$\vdash_m (\neg\neg A \to A) \to ((\neg A \to A) \to A)$

Remarque. On a montré l'équivalence au dessus de la logique minimale de certains schémas d'axiomes ; cela ne signifie évidemment pas que les énoncés correspondants sont équivalents en logique minimale : par exemple, $((\neg A \to A) \to A) \to (A \vee \neg A)$ et $(\neg\neg A \to A) \to (A \vee \neg A)$ ne sont pas démontrables en logique intuitionniste (cf chapître II).

1.2.4 De la logique intuitionniste à la logique classique

La logique intuitionniste diffère de la logique classique en ce qu'elle n'admet que les raisonnements constructifs. Cependant au plan syntaxique la différence est simple puisqu'il s'agit seulement de savoir si on peut remplacer $\neg\neg A$ par A. En fait, on peut transformer toute démonstration classique en démonstration intuitionniste au moyen d'une **traduction de Gödel**, qui consiste à placer des "$\neg\neg$" devant les formules atomiques, les $\vee$ et les $\exists$. Bien sûr, cette transformation ne conserve pas la conclusion de la démonstration : la nouvelle conclusion est seulement équivalente à l'ancienne en logique classique.

Appelons **stable** (par double négation) une formule telle que $\neg\neg A \vdash_m A$. Si A est stable, on a en fait $\neg\neg A \equiv_m A$.

Lemme 1.2.2 *La constante $\bot$ est stable, et pour toute formule A, $\neg A$ est stable.*

Démonstration.

$$\dfrac{\dfrac{\dfrac{[\![\bot]\!]^1}{\neg\bot}\,\neg_i\,1 \quad \neg\bot}{\bot}\,\neg_e \qquad \dfrac{\neg\neg\neg A \qquad \dfrac{\dfrac{\dfrac{[\![\neg A]\!]^1 \qquad [\![A]\!]^2}{\bot}\,\neg_e}{\neg\neg A}\,\neg_i\,1}{\bot}\,\neg_e}{\neg A}\,\neg_i\,2$$

Lemme 1.2.3 *(déplacement des doubles négations à l'intérieur des $\wedge, \to$ et $\forall$)*
(a) $\neg\neg(A \wedge B) \vdash_m \neg\neg A \wedge \neg\neg B$
(b) $\neg\neg(A \to B) \vdash_m \neg\neg A \to \neg\neg B$
(c) $\neg\neg\forall x A \vdash_m \forall x \neg\neg A$

Démonstration.
(a) $\neg\neg(A \wedge B) \vdash_m \neg\neg A \wedge \neg\neg B$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\![A \wedge B]\!]^1}{A}\,\wedge_{eg} \quad [\![\neg A]\!]^2}{\bot}\,\neg_e}{\neg(A \wedge B)}\,\neg_i\,1 \qquad \neg\neg(A \wedge B)}{\bot}\,\neg_e}{\neg\neg A}\,\neg_i\,2 \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\![A \wedge B]\!]^3}{B}\,\wedge_{ed} \quad [\![\neg B]\!]^4}{\bot}\,\neg_e}{\neg(A \wedge B)}\,\neg_i\,3 \qquad \neg\neg(A \wedge B)}{\bot}\,\neg_e}{\neg\neg B}\,\neg_i\,4}{\neg\neg A \wedge \neg\neg B}\,\wedge_i$$

(b) $\neg\neg(A \to B) \vdash_m \neg\neg A \to \neg\neg B$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\![A \to B]\!]^1 \quad [\![A]\!]^2}{B}\,\to_e \quad [\![\neg B]\!]^3}{\bot}\,\neg_e}{\neg(A \to B)}\,\neg_i\,1 \qquad \neg\neg(A \to B)}{\bot}\,\neg_e}{\neg A}\,\neg_i\,2 \qquad [\![\neg\neg A]\!]^4}{\bot}\,\neg_e}{\neg\neg B}\,\neg_i\,3}{\neg\neg A \to \neg\neg B}\,\to_i\,4$$

(c) $\neg\neg\forall x A \vdash_m \forall x \neg\neg A$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\![\forall x A]\!]^1}{A}\,\forall_e \quad [\![\neg A]\!]^2}{\bot}\,\neg_e}{\neg\forall x A}\,\neg_i\,1 \qquad \neg\neg\forall x A}{\bot}\,\neg_e}{\neg\neg A}\,\neg_i\,2}{\forall x \neg\neg A}\,\forall_i$$

Exercice 8
(a) $A \to B \vdash_m \neg B \to \neg A$
(b) $\neg\neg B \to \neg\neg A \vdash_m \neg A \to \neg B$

Définition 1.2.4 La traduction de Gödel est une application qui à toute formule A associe une formule A° classiquement équivalente à A définie par [9] :

$\bot^\circ = \bot$
$A^\circ = \neg\neg A$ si A est atomique
$(A \wedge B)^\circ = A^\circ \wedge B^\circ$
$(A \to B)^\circ = A^\circ \to B^\circ$
$(\forall x A)^\circ = \forall x (A^\circ)$
$(A \vee B)^\circ = \neg\neg(A^\circ \vee B^\circ)$
$(\exists x A)^\circ = \neg\neg\exists x (A^\circ)$

On remarquera que $(\neg A)^\circ = \neg A^\circ$.

Proposition 1.2.5 *Pour toute formule A, A° est stable par double négation.*

Démonstration. On procède par récurrence sur la complexité de A°. Si A° est $\bot$ ou commence par une négation, alors le résultat vient de 1.2.2. Si le connecteur principal de A° est $\wedge$, $\to$ ou $\forall$, on utilise 1.2.3 : supposons par exemple que A° soit $C^\circ \wedge D^\circ$; par 1.2.3 on a $\neg\neg(C^\circ \wedge D^\circ) \vdash_m \neg\neg C^\circ \wedge \neg\neg D^\circ$; mais par hypothèse de récurrence, $\neg\neg C^\circ \equiv_m C^\circ$ et $\neg\neg D^\circ \equiv_m D^\circ$ et donc $\neg\neg(C^\circ \wedge D^\circ) \vdash_m C^\circ \wedge D^\circ$.

Proposition 1.2.6 *Si $A_1, ..., A_n \vdash_c A$, alors $A_1^\circ, ..., A_n^\circ \vdash_m A^\circ$. Plus précisément, il existe un algorithme qui permet de transformer toute déduction classique d de A sous les hypothèses $A_1, ..., A_n$ en une déduction minimale d° de A° sous les hypothèses $A_1^\circ, ..., A_n^\circ$.*

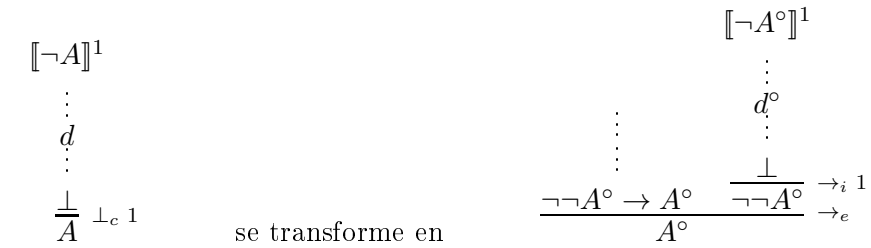
Démonstration. On construit d° par récurrence sur la construction de d. Pour une déduction de longueur 1, le résultat est évident.
Si la dernière règle concerne $\wedge$, $\to$ ou $\forall$, la construction n'utilise que l'hypothèse de récurrence et la définition de A°. Par exemple si d_1 et d_2 se transforment en d_1° et d_2°, alors

$$\dfrac{\begin{array}{cc} d_1 & d_2 \\ \vdots & \vdots \\ A & B \end{array}}{A \wedge B}\ \wedge_i \qquad \text{se transforme en} \qquad \dfrac{\begin{array}{cc} d_1^\circ & d_2^\circ \\ \vdots & \vdots \\ A^\circ & B^\circ \end{array}}{A^\circ \wedge B^\circ}\ \wedge_i$$

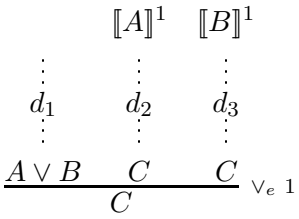
Si la dernière règle est $\vee_i$ ou $\exists_i$, on construit en plus une double négation : par exemple,

$$\dfrac{\begin{array}{c} d \\ \vdots \\ A \end{array}}{A \vee B}\ \vee_{ig} \qquad \text{se transforme en} \qquad \dfrac{\begin{array}{cc} \vdots & d^\circ \\ & \vdots \\ (A^\circ \vee B^\circ) \to \neg\neg(A^\circ \vee B^\circ) & \dfrac{A^\circ}{A^\circ \vee B^\circ}\ \vee_{ig} \end{array}}{\neg\neg(A^\circ \vee B^\circ)}\ \to_e$$

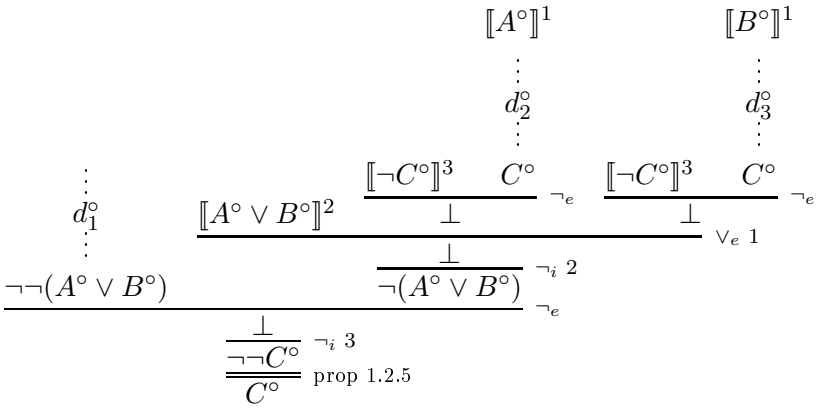
Si la dernière règle est $\bot_c$, on utilise la proposition 1.2.5 pour enlever une double négation :



Si la dernière règle est $\vee_e$ ou $\exists_e$, on utilise la proposition 1.2.5 pour enlever une double négation :



se transforme en



Exercice 9 On définit une autre traduction $^\bullet$ de la logique classique dans la logique minimale par [10] :

$\bot^\bullet = \bot$
$A^\bullet = \neg\neg A$ si A est atomique
$(A \wedge B)^\bullet = A^\bullet \wedge B^\bullet$
$(A \to B)^\bullet = A^\bullet \to B^\bullet$
$(\forall x A)^\bullet = \forall x (A^\bullet)$
$(A \vee B)^\bullet = \neg(\neg A^\bullet \wedge \neg B^\bullet)$
$(\exists x A)^\bullet = \neg\forall x (\neg A^\bullet)$.

Montrer que cette traduction satisfait aussi la proposition 2.4.4.

Lemme 1.2.7 *(déplacement des $\neg\neg$ à l'extérieur des $\wedge$, $\rightarrow$ et $\forall$)*
(a) $\neg\neg A \wedge \neg\neg B \vdash_m \neg\neg(A \wedge B)$
(b) $\neg\neg A \rightarrow \neg\neg B \vdash_i \neg\neg(A \rightarrow B)$
(c) $\forall x \neg\neg A \vdash_c \neg\neg \forall x A$

Démonstration.
(a) $\neg\neg A \wedge \neg\neg B \vdash_m \neg\neg(A \wedge B)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\neg\neg A \wedge \neg\neg B}{\neg\neg B}\,\wedge_{ed} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\neg\neg A \wedge \neg\neg B}{\neg\neg A}\,\wedge_{eg} \qquad \dfrac{\dfrac{[\![\neg(A\wedge B)]\!]^3 \qquad \dfrac{[\![A]\!]^1 \quad [\![B]\!]^2}{A\wedge B}\,\wedge_i}{\bot}\,\neg_e}{\neg A}\,\neg_i\,1}{\bot}\,\neg_e}{\neg B}\,\neg_i\,2}{\bot}\,\neg_e}{\neg\neg(A\wedge B)}\,\neg_i\,3$$

(b) $\neg\neg A \rightarrow \neg\neg B \vdash_i \neg\neg(A \rightarrow B)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\neg\neg A \rightarrow \neg\neg B \qquad \dfrac{\dfrac{[\![\neg A]\!]^2 \quad [\![A]\!]^3}{\bot}\,\neg_e}{\neg\neg A}\,\neg_i\,2}{\neg\neg B}\,\rightarrow_e \qquad \dfrac{\dfrac{[\![\neg(A\rightarrow B)]\!]^4 \qquad \dfrac{[\![B]\!]^1}{A\rightarrow B}\,\rightarrow_i}{\bot}\,\neg_e}{\neg B}\,\neg_i\,1}{\bot}\,\neg_e}{B}\,\bot_e}{A\rightarrow B}\,\rightarrow_i\,3 \qquad [\![\neg(A\rightarrow B)]\!]^4}{\bot}\,\neg_e}{\neg\neg(A\rightarrow B)}\,\neg_i\,4$$

(c) $\forall x \neg\neg A \vdash_c \neg\neg \forall x A$

$$\dfrac{\dfrac{\dfrac{\dfrac{\forall x\neg\neg A}{\neg\neg A}\,\forall_e \qquad [\![\neg A]\!]^1}{\bot}\,\neg_e}{A}\,\bot_c\,1 \qquad [\![\neg\forall x A]\!]^2}{\bot}\,\neg_e}{\neg\neg\forall x A}\,\neg_i\,2 \qquad \dfrac{A}{\forall x A}\,\forall_i$$

Remarque. Les résultats du lemme précédent ne peuvent pas être améliorés car $(\neg\neg A \rightarrow \neg\neg B) \rightarrow \neg\neg(A \rightarrow B)$ n'est pas démontrable en logique minimale et $\forall x \neg\neg A \rightarrow \neg\neg \forall x A$ n'est pas démontrable en logique intuitionniste (cf chapître II [11]).
Voici un argument permettant de déduire que $(\neg\neg A \rightarrow \neg\neg B) \rightarrow \neg\neg(A \rightarrow B)$ n'est pas démontrable en logique minimale, du fait que $(\neg A \rightarrow A) \rightarrow A$ n'est pas démontrable en logique intuitionniste : si $\neg\neg A \rightarrow \neg\neg B \vdash_m \neg\neg(A \rightarrow B)$, alors en substituant A à $\bot$ et $\bot$ à B on obtient par 1.2.1 $((A \rightarrow A) \rightarrow A) \rightarrow ((\bot\rightarrow A) \rightarrow A) \vdash_m ((A \rightarrow\bot) \rightarrow A) \rightarrow A$; comme l'hypothèse est démontrable en logique minimale, la conclusion aussi, c'est à dire $\vdash_m (\neg A \rightarrow A) \rightarrow A$.

Lemme 1.2.8
(a) $\neg(A \vee B) \equiv_m \neg A \wedge \neg B$
(b) $\neg\exists x A \equiv_m \forall x \neg A$

Démonstration. On montre la partie (a) et on laisse (b) en exercice.
(i) $\neg A \wedge \neg B \vdash_m \neg(A \vee B)$

$$\dfrac{\dfrac{\dfrac{\neg A \wedge \neg B}{\neg A}\wedge_{eg} \quad [\![A]\!]^1}{\bot}\neg_e \qquad \dfrac{\dfrac{\neg A \wedge \neg B}{\neg B}\wedge_{ed} \quad [\![B]\!]^1}{\bot}\neg_e \qquad [\![A \vee B]\!]^2}{\dfrac{\bot}{\neg(A \vee B)}\neg_i\ 2}\vee_e\ 1$$

(ii) $\neg(A \vee B) \vdash_m \neg A \wedge \neg B$

$$\dfrac{\dfrac{\neg(A \vee B) \quad \dfrac{[\![A]\!]^2}{A \vee B}\vee_{ig}}{\dfrac{\bot}{\neg A}\neg_i\ 2}\neg_e \qquad \dfrac{\neg(A \vee B) \quad \dfrac{[\![B]\!]^1}{A \vee B}\vee_{id}}{\dfrac{\bot}{\neg B}\neg_i\ 1}\neg_e}{\neg A \wedge \neg B}\wedge_i$$

Exercice 10
$\neg\neg(A \to B) \equiv_m \ A \to \neg\neg B$

Lemme 1.2.9 *Pour toute formule C du calcul propositionnel, $C^\circ \equiv_i \neg\neg C$.*

Démonstration. On procède par récurrence sur la complexité de C.
(1) Si $C = \bot$, alors $C^\circ = \bot$ et le résultat vient de 1.2.2.
(2) Si C est une formule atomique distincte de $\bot$, alors $C^\circ = \neg\neg C$ et le résultat est trivial.
(3) Si $C = A \wedge B$, alors $C^\circ = A^\circ \wedge B^\circ$. Par hypothèse de récurrence, $A^\circ \equiv_i \neg\neg A$, $B^\circ \equiv_i \neg\neg B$ et donc $C^\circ \equiv_i \neg\neg A \wedge \neg\neg B$; finalement par 1.2.3 et 1.2.7, on a $C^\circ \equiv_i \neg\neg(A \wedge B)$.
(4) Si $C = A \to B$, alors $C^\circ = A^\circ \to B^\circ$. Par hypothèse de récurrence, $A^\circ \equiv_i \neg\neg A$, $B^\circ \equiv_i \neg\neg B$ et donc $C^\circ \equiv_i \neg\neg A \to \neg\neg B$; finalement par 1.2.3 et 1.2.7, on a $C^\circ \equiv_i \neg\neg(A \to B)$.
(5) Si $C = A \vee B$, alors $C^\circ = \neg\neg(A^\circ \vee B^\circ)$. Par hypothèse de récurrence, $A^\circ \equiv_i \neg\neg A$, $B^\circ \equiv_i \neg\neg B$ et donc $C^\circ \equiv_i \neg\neg(\neg\neg A \vee \neg\neg B) \equiv_i \neg(\neg\neg\neg A \wedge \neg\neg\neg B) \equiv_i \neg(\neg A \wedge \neg B) \equiv_i \neg\neg(A \vee B)$ (les équivalences utilisent respectivement 1.2.8, 1.2.2 et 1.2.8).

Remarque. La formule $C^\circ \to \neg\neg C$ n'est pas démontrable en logique mininale, car $(\neg\neg A \to \neg\neg B) \to \neg\neg(A \to B)$ ne l'est pas (cf remarque précédente).

Proposition 1.2.10 *(Théorème de Glivenko) Pour toute formule C du calcul propositionnel,*

$$\vdash_c \ C \text{ si et seulement } \vdash_i \neg\neg C$$

Démonstration. Par 1.2.6 et 1.2.9.

Commentaire La démonstration de la proposition précédente repose sur la possibilité de faire passer les doubles négations à l'extérieur. On ne peut pas prouver le résultat pour la logique minimale car pour faire passer les doubles négations à l'extérieur d'un $\rightarrow$, on a besoin de la logique intuitionniste ; on ne peut pas l'étendre au calcul des prédicats car pour faire passer les doubles négations à l'extérieur d'un $\forall$, on a besoin de la logique classique (cf 1.2.7 et remarque suivante).
Pour justifier cela formellement on peut montrer (cf chapître II [12]) que $\neg\neg(\neg\neg A \rightarrow A)$ n'est pas démontrable en logique minimale et que $\neg\neg\forall x(Rx \vee \neg Rx)$ n'est pas démontrable en logique intuitionniste (bien sûr $\neg\neg A \rightarrow A$ et $\forall x(Rx \vee \neg Rx)$ sont démontrables en logique classique).

Exercice 11 Montrer que pour toute formule A du calcul des prédicats, A est démontrable en logique classique si et seulement si $\neg\neg A$ est démontrable en logique intuitionniste augmentée du schéma d'axiome $\forall x \neg\neg A \rightarrow \neg\neg \forall x A$.

Exercice 12
(a) $\vdash_m \neg\neg(A \vee \neg A)$
(b) $\vdash_m \neg\neg((\neg A \rightarrow A) \rightarrow A)$
(c) $\vdash_i \neg\neg(\neg\neg A \rightarrow A)$
(d) $\vdash_i \neg\neg((\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B))$

1.2.5 La notion de coupure en déduction naturelle

En mathématiques, on fait souvent usage de démonstrations "indirectes", qui prennent généralement la forme suivante : pour déduire B de Γ, on fait appel à une propriété A qu'on sait déduire de Γ (A est éventuellement beaucoup plus compliquée que B) et dont on démontre qu'elle entraîne B, au lieu de déduire directement B de Γ. Ce type de démonstration repose sur la transitivité de la déduction : de $\Gamma \vdash A$ et $\Gamma, A \vdash B$ on déduit $\Gamma \vdash B$.

En déduction naturelle la transitivité n'est pas tout à fait une figure élémentaire de raisonnement, elle s'obtient en deux étapes : on doit d'abord enregistrer le fait que A entraîne B sous forme d'un lemme $A \rightarrow B$, puis de A et $A \rightarrow B$ déduire B. La situation devient donc la suivante :

$$\dfrac{\dfrac{\begin{array}{c}[\![A]\!]^1 \\ \vdots \\ d_2 \\ \vdots \\ B\end{array}}{A \rightarrow B}\,{\scriptstyle \rightarrow_i 1} \qquad \begin{array}{c} \\ \\ d_1 \\ \vdots \\ A\end{array}}{B}\,{\scriptstyle \rightarrow_e}$$

Cette situation, où une $\rightarrow_i$ est directement suivie de la $\rightarrow_e$ correspondante, s'appelle une coupure. Plus généralement une **coupure** est la succession d'une règle d'introduction et d'une règle d'élimination ayant toutes deux la même formule principale C, appelée **formule de coupure** (on parlera aussi de coupure sur C). Chaque connecteur peut donc donner lieu à une sorte particulière de coupure ; voici par exemple les cas du $\wedge$ et du $\forall$:

$$\dfrac{\dfrac{\begin{array}{cc} d_1 & d_2 \\ \vdots & \vdots \\ A & B \end{array}}{A \wedge B}\ \wedge_{eg}}{A}\ \wedge_i \qquad\qquad \dfrac{\begin{array}{c} d \\ \vdots \\ A[t/x] \end{array}}{\dfrac{\forall x A}{A[t/x]}\ \forall_i}\ \forall_e$$

Les coupures constituent des détours dans les démonstrations. Voici un exemple : supposons qu'on cherche à démontrer que "$(\neg P \to P) \to P$ est un théorème du calcul propositionnel" (énoncé B) ; une manière de procéder est de faire appel au théorème de complétude qui dit que "les énoncés vrais dans tous les modèles sont des théorèmes" (énoncé A), dont on peut déduire facilement que $(\neg P \to P) \to P$ est un théorème. On aurait pu procéder plus directement et construire une déduction de $(\neg P \to P) \to P$ dans le système formel [13]. Mais l'intérêt mathématique des coupures c'est justement de permettre l'utilisation d'énoncés généraux. Elles représentent la partie intelligente (non mécanique) de l'activité mathématique : au lieu de refaire dans chaque cas particulier une démonstration particulière, on a remarqué un phénomène général derrière tous les cas particuliers, qu'on a démontré une fois pour toutes, et qu'on se contente d'appliquer ensuite dans ces cas particuliers.

Il y a une façon évidente d'éliminer une coupure sur une formule $A \to B$ dans une démonstration. Au lieu de conclure B à partir de A et de $A \to B$, on peut prendre la démonstration de B sous l'hypothèse A et remplacer dans cette démonstration toutes les occurrences de l'hypothèse A par la démonstration de A, ce qui fournit une démonstration de A qui ne contient pas de coupure sur $A \to B$. Cette "élimination" des coupures attire quelques remarques.

D'abord l'élimination d'une coupure peut créer de nouvelles coupures (dans le cas où la démonstration de A se termine par une introduction, et où l'hypothèse A est la formule principale d'une élimination dans la déduction de B sous l'hypothèse A) et dupliquer des coupures existantes (dans le cas où la démonstration de A contient des coupures et où l'hypothèse A est utilisée plusieurs fois), de sorte qu'il n'est pas évident qu'on puisse éliminer toutes les coupures. On verra au chapître suivant qu'il est possible de transformer toute démonstration en une démonstration sans coupure en utilisant ce procédé d'élimination.

Ensuite, l'élimination des coupures peut faire "exploser" la taille des démonstrations : si l'hypothèse A est utilisée n fois dans la déduction de B sous l'hypothèse A, on recopie n fois la démonstration de A, et au cours de ce processus on a pu dupliquer des coupures dont l'élimination provoquera le même phénomème ..etc. Considérons par exemple la démonstration d suivante de $(A \to A) \to (A \to A)$

$$\dfrac{\begin{array}{cc} d_2 & d_1 \\ \vdots & \vdots \\ ((A \to A) \to (A \to A)) \to ((A \to A) \to (A \to A)) & (A \to A) \to (A \to A) \end{array}}{(A \to A) \to (A \to A)}$$

où d_1 est la démonstration suivante de $(A \to A) \to (A \to A)$

$$\dfrac{\dfrac{[\![A \to A]\!]^2 \quad \dfrac{[\![A \to A]\!]^2 \quad [\![A]\!]^1}{A} \to_e}{A} \to_e}{\dfrac{A}{A \to A} \to_i 1} \to_i 2$$

et d_2 est la démonstration suivante de $((A \to A) \to (A \to A)) \to ((A \to A) \to (A \to A))$

$$\dfrac{\dfrac{[\![(A \to A) \to (A \to A)]\!]^2 \quad \dfrac{[\![(A \to A) \to (A \to A)]\!]^2 \quad \dfrac{[\![(A \to A) \to (A \to A)]\!]^2 \quad [\![A \to A]\!]^1}{A \to A}}{A \to A}}{\dfrac{A \to A}{(A \to A) \to (A \to A)} \; 1} \; 2}{((A \to A) \to (A \to A)) \to ((A \to A) \to (A \to A))}$$

La démonstration d contient une coupure dont l'élimination provoque la création de nouvelles coupures, qu'on peut aussi éliminer .... jusqu'à obtenir à la fin la démonstration sans coupure de $(A \to A) \to (A \to A)$ suivante de hauteur 11, qui utilise 8 fois l'hypothèse $A \to A$ (remarque : lors de l'élimination des coupures de cette démonstration, il faut faire attention à ne pas "mélanger" les liaisons introduites par les mutifications ; le plus simple pour éviter cela est de donner systématiquement de nouveaux numéros aux liaisons au cours des recopies) :

$$\dfrac{\dfrac{[\![A \to A]\!]^2 \quad \dfrac{[\![A \to A]\!]^2 \quad \dfrac{[\![A \to A]\!]^2 \quad \dfrac{[\![A \to A]\!]^2 \quad \dfrac{[\![A \to A]\!]^2 \quad \dfrac{[\![A \to A]\!]^2 \quad \dfrac{[\![A \to A]\!]^2 \quad \dfrac{[\![A \to A]\!]^2 \quad [\![A]\!]^1}{A}}{A}}{A}}{A}}{A}}{A}}{A}}{\dfrac{A}{A \to A} \; 1} \; 2}{(A \to A) \to (A \to A)}$$

Plus généralement, si on prend pour d_2 une démonstration de $((A \to A) \to (A \to A)) \to ((A \to A) \to (A \to A))$ qui utilise n fois l'hypothèse $(A \to A) \to (A \to A)$, alors d est une démonstration de hauteur $n + 3$ qui produit par élimination des coupures une démonstration de hauteur $2^n + 3$.

Mais l'élimination des coupures n'est pas seulement le remplacement de démonstrations courtes et intelligentes par des démonstrations longues et stupides, le tout se faisant à l'aide d'un procédé particulièrement ennuyeux. Les démonstrations sans coupures sont un *outil d'analyse métamathématique* puissant et la procédure d'élimination des coupures fournit un

modèle de calcul original en informatique, qui fera l'objet d'une grande partie de ce cours (cf chapître III).

Les démonstrations sans coupure sont des démonstrations "intrinsèques" : rien de ce qui est utilisé dans la démonstration n'est extérieur à la conclusion. Formellement cela signifie que toute les formules intervenant dans la démonstration sont des sous-formules de la conclusion : c'est ce qu'on appelle la **propriété de la sous-formule**. L'intérêt de cette propriété, du point de vue métamathématique, réside essentiellement dans le contrôle qu'elle permet sur les démonstrations : on peut espérer analyser toutes les démonstrations sans coupures d'un énoncé, alors qu'on ne peut analyser toutes les démonstrations d'un énoncé ; on obtient ainsi notamment des preuves de cohérence ($\bot$ n'ayant pas d'autre sous-formule qu'elle-même, il n'y a pas de démonstration sans coupure de $\bot$).

Proposition 1.2.11 *Les démonstrations sans coupures en logique minimale dans le fragment $\to$, $\wedge$, $\forall$ ont la propriété de la sous-formule : si d est une démonstration sans coupure de A sous les hypothèses Γ, alors toutes les formules figurant dans d sont des sous-formules de A ou de formules de Γ.*

Démonstration. Soit d une démonstration sans coupure de A sous les hypothèse Γ. Supposons que d ne possède pas la propriété de la sous-formule et soit C une formule de complexité maximum parmi les formules figurant dans d qui ne sont pas sous-formules de A ou de formules de Γ. Dans ce cas, C est conclusion d'une règle R_1 et prémisse d'une règle R_2 (par maximalité, C ne peut être une hypothèse mutifiée). Dans les règles pour $\to$, $\wedge$, et $\forall$, toutes les formules sont sous-formules de la formule principale ; comme C est maximale, il s'en suit que C est formule principale de R_1 et R_2. Cela signifie que C est une formule de coupure.

Remarque. Pour obtenir les mêmes propriétés en présence des autres connecteurs comme $\vee$, il faut une notion de coupure un peu plus compliquée (on a besoin en plus de coupures dites commutatives). Les règles d'absurdité posent d'autres problèmes que nous aborderons plus tard.
14

1.3 Calcul des séquents.

Le calcul des séquents chasse sur les mêmes terres que la déduction naturelle, mais avec d'autres armes : il a aussi pour ambition de rendre compte de la notion de démonstration, mais pour mieux en dégager les propriétés mathématiques, il n'en retient que l'essentiel, perdant au passage son caractère "naturel".

La différence principale avec la déduction naturelle réside dans le remplacement des règles d'élimination par des règles d'introduction à gauche, ce qui permet d'avoir un système complétement symétrique où les deux groupes de règles d'un même connecteur (introductions à gauche et à droite) se "répondent" exactement.

En fait, le calcul des séquents ne s'éloigne pas seulement de la notion de démonstration usuelle, il s'éloigne de la notion de démonstration tout court : contrairement à la déduction naturelle, il ne permet de travailler qu'au niveau des "séquents" ; il n'y a pas vraiment de déduction de "formule" correspondante.

Ainsi le calcul des séquents apparaît-il plus comme un formalisme permettant de décrire des constructions de démonstrations plutôt que de les construire (les règles structurelles prennent notamment une grande importance).

1.3.1 Les séquents.

En logique classique, les séquents deviennent symétriques, c'est à dire qu'un séquent possède non seulement plusieurs formules à gauche du signe " $\vdash$ ", mais aussi plusieurs formules à droite de celui-ci. C'est probablement surtout ceci qui éloigne le calcul des séquents de la notion de démonstration usuelle.

On verra que le calcul des séquents intuitionniste manipule essentiellement les mêmes séquents que la déduction naturelle, et est du même coup beaucoup plus proche de cette dernière et de la notion de démonstration usuelle que ne l'est le calcul des séquents classique.

Un *séquent*, est une expression de la forme $\Gamma \vdash \Delta$ où Γ et Δ sont des suites finies non ordonnées de formules éventuellement vide[2]. [15] On peut également trouver des définitions de séquents où Γ et Δ sont des suites finies ordonnées de formules. Il faut alors ajouter à la logique une règle dite *d'échange* qui permet de permuter les formules du séquent à gauche et à droite.[3]

Dans le séquent $\Gamma \vdash \Delta$, on désigne Γ par partie *droite du séquent*, Δ par partie *gauche du séquent*.

Comme un tel séquent a plusieurs formules à droite, on ne peut distinguer a priori une conclusion parmi celles-ci. Pour fixer les choses définissons l'interprétation d'un séquent en logique classique, qui est une extension de la relation de satisfaction pour les formules.

Définition 1.3.1 Un séquent $\Gamma \vdash \Delta$ exprimé dans le langage $\mathcal{L}$ est valide dans une $\mathcal{L}$-structure $\mathcal{M}$ (ou satisfait par cette structure), quand, pour toutes substitution σ de ses variables libres par des éléments de $\mathcal{M}$, si toutes les formules de $\sigma(\Gamma)$ sont valides dans $\mathcal{M}$ alors au moins l'une des formules de $\sigma(\Delta)$ est valide.

2. une suite non ordonnée de formules peut être formalisée comme la donnée d'une application d'un ensemble fini d'indices dans l'ensemble des formules. Quand cela sera nécessaire on notera l'indice de la formule en exposant

3. Il est également possible de trouver des définitions de séquent où Γ et Δ sont des ensembles de formules, mais cela n'est adéquat que si l'on s'intéresse plus à la prouvabilité qu'aux preuves elles-mêmes.

Remarquons que l'interprétation du séquent $A_1, \ldots, A_n \vdash B_1, \ldots, B_n$ est celle de la formule $(A_1 \wedge \cdots \wedge A_n) \to (B_1 \vee \cdots \vee B_n)$, ou encore de la formule $A_1 \to \cdots \to A_n \to \neg B_1 \to \ldots \neg B_n \to \bot$ [4]. Pour interpréter "naturellement" une preuve en calcul des séquents, il faudrait probablement distinguer l'une des formules à droite comme la formule conclusion et les autres formules à droites comme des hypothèses négatives, par exemple interpréter le séquent ci-dessus comme $A_1 \to \cdots \to A_n \to \neg B_2 \to \ldots \neg B_n \to B_1$. Le raisonnement par l'absurde apparaît alors comme le changement de choix de formule à droite "active", c'est à dire considérée comme la conclusion courante. Comme le calcul des séquents ne traite pas de ce choix de formule conclusion, il gère implicitement [5] le raisonnement par l'absurde.

Par contre, la symétrie du séquent permet de donner un calcul complètement symétrique et uniforme pour les connecteurs, à la différence de la déduction naturelle.

1.3.2 Les règles du calcul des séquents.

Le calcul des séquents conserve les règles d'introduction de la déduction naturelle qui s'appellent alors *règles droites*. Les règles d'éliminations sont remplacées par des *règles gauches* qui agissent sur la partie gauche du séquent.

IL y a plusieurs façon de comprendre et de reconstruire les règles du calcul des séquents, en voici une qui se sert de la déduction naturelle. En effet dans le cas où les séquents ont au plus une formule à droite, les règles gauches, vue de bas en haut, peuvent être interprétées comme des procédés de construction de preuves en déduction naturelle "en descendant" (vers la conclusion). Ainsi supposons que nous ayons à prouver le séquent $\Gamma, A \to B \vdash C$. Pour construire une preuve de C sous hypothèse $\Gamma, A \to B$ en déduction naturelle, nous pouvons utiliser le procédé décrit ainsi : « Si j'ai une preuve de A sous hypothèses Γ, pour prouver C sous hypothèses Γ, il suffit de prouver C sous hypothèses Γ, B ». Ceci correspond à la règle gauche du connecteur $\to$ en calcul des séquents :

$$\frac{\Gamma, B \vdash C \quad \Gamma \vdash A}{\Gamma, A \to B \vdash C} \to_g$$

dans laquelle on interprète la partie gauche comme : « les hypothèses et ce qui a déjà été prouvé », la formule à droite comme « ce que l'on cherche à prouver ». On systématisera ceci pour traduire la déduction naturelle en calcul des séquents au § 1.3.4 page 37.

Le système de règles du tableau 1.3 page suivante est complet pour la logique classique.

Règle de coupure.

À cause du remplacement des règles d'élimination par des règles d'introduction à gauche, les règles logiques n'engendrent directement aucune coupure : on ne peut pas en particulier disposer de la transitivité au niveau des déductions. On ajoute donc une règle de coupure explicite qui remplace les diverses coupures associées aux opérateurs logiques qui apparaissent en déduction naturelle.

4. On note $A \to B \to C$ pour $A \to (B \to C)$.
5. ce qui l'éloigne très clairement de la démonstration usuelle !

On adopte comme connecteurs $\neg$, $\rightarrow$, $\wedge, \vee$, comme quantificateurs $\forall$, $\exists$. L'absurde $\bot$ n'est pas un symbole primitif. Le séquent $\Gamma \vdash$ correspond à $\Gamma \vdash \bot$.

Règles axiome / coupure

$$\frac{}{A \vdash A}\ \text{ax.} \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma', A \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\ \textit{cut}$$

Règles logiques

conjonction

$$\frac{\Gamma, A \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ \wedge_{g_d} \qquad \frac{\Gamma, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ \wedge_{g_g} \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \wedge B, \Delta, \Delta'}\ \wedge_d$$

disjonction

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \vee B \vdash \Delta, \Delta'}\ \vee_g \qquad \frac{\Gamma \vdash A, \Delta}{\Gamma \vdash A \vee B, \Delta}\ \vee_{d_g} \qquad \frac{\Gamma \vdash B, \Delta}{\Gamma \vdash A \vee B, \Delta}\ \vee_{d_d}$$

implication

$$\frac{\Gamma, B \vdash \Delta \quad \Gamma' \vdash A, \Delta'}{\Gamma, \Gamma', A \rightarrow B \vdash \Delta, \Delta'}\ \rightarrow_g \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash A \rightarrow B, \Delta}\ \rightarrow_{d_g} \qquad \frac{\Gamma \vdash B, \Delta}{\Gamma \vdash A \rightarrow B, \Delta}\ \rightarrow_{d_d}$$

négation

$$\frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}\ \neg_g \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}\ \neg_d$$

quantification universelle

$$\frac{A[t/x], \Gamma \vdash \Delta}{\forall x A, \Gamma \vdash \Delta}\ \forall_g \qquad \frac{\Gamma \vdash A[y/x], \Delta}{\Gamma \vdash \forall x A, \Delta}\ \forall_d\ (*)$$

quantification existentielle

$$\frac{\Gamma, A[y/x] \vdash \Delta}{\Gamma, \exists x A \vdash \Delta}\ \exists_g\ (*) \qquad \frac{\Gamma \vdash A[t/x], \Delta}{\Gamma \vdash \exists x A, \Delta}\ \exists_d$$

$(*)$ *Restriction* : y n'a pas d'occurrence libre dans Γ, A, Δ.

Règles structurelles

Affaiblissement

$$\frac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta}\ w_g \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash A, \Delta}\ w_d$$

Contraction

$$\frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta}\ c_g \qquad \frac{\Gamma \vdash A, A, \Delta}{\Gamma \vdash A, \Delta}\ c_d$$

Table 1.3 – Règles du calcul des séquents

Conventions.

i. Dans les règles $\forall_d$ et $\exists_g$ la variable y s'appelle le *paramètre propre* de la règle. On rappelle la restriction $(*)$: y ne doit pas apparaître dans le séquent conclusion de la règle. On supposera de plus qu'un paramètre propre n'apparaît jamais hors de la sous-déduction qui se termine par la règle dont il est le paramètre propre, ce qui évitera de se soucier de la conservation de la restriction $(*)$ lors de l'élimination des coupures (voir § 1.5.1 page 55).

ii. Dans une règle les formules (en fait les occurrences de formule) distinguées (écrites en lettres latines capitales) sont dites *actives*, et les autres forment le *contexte*. On appelle *formule principale* d'une règle, la formule active de la conclusion de cette règle ; dans un axiome les deux formules sont considérées comme principales.

Définition 1.3.2 Une *déduction* en calcul des séquents est une suite finie $S_0, \dots S_n$ tels que pour tout $i \leq n$, l'une des conditions suivantes au moins est réalisée :

i. S_i est un axiome,

ii. S_i est obtenu par application d'une des règles à des séquents S_j pour des $j < i$.

Une formule A est *déductible* de Γ, s'il existe une déduction qui se termine par le séquent $\Gamma \vdash A$; on dit qu'une formule est un *théorème* si elle est close et déductible de la suite vide.

Exemples.

On reprend des exemples vus précédemment en déduction naturelle.

1. $A \to (B \to A)$

$$\dfrac{\dfrac{\dfrac{A \vdash A}{A \vdash B \to A}\ {\scriptstyle \to_d + w_d}}{\vdash A \to (B \to A)}\ {\scriptstyle \to_d}}$$

2. $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{B \vdash B \quad C \vdash C}{B \to C, B \vdash C}\ {\scriptstyle \to_g}}{A \vdash A \quad B \to C, A \to B, A \vdash C}\ {\scriptstyle \to_g}}{A \to (B \to C), A \to B, A \vdash C}\ {\scriptstyle \to_g + c_g}}{A \to (B \to C), A \to B \vdash A \to C}\ {\scriptstyle \to_d}}{A \to (B \to C) \vdash (A \to B) \to (A \to C)}\ {\scriptstyle \to_d}}{\vdash (A \to (B \to C)) \to ((A \to B) \to (A \to C))}\ {\scriptstyle \to_d}$$

3. $\neg\neg A \to A$

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{\vdash A, \neg A}\ {\scriptstyle \neg_d}}{\neg\neg A \vdash A}\ {\scriptstyle \neg_g}}{\vdash \neg\neg A \to A}\ {\scriptstyle \to_d}}$$

4. $A \vee \neg A$

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{\vdash A, \neg A}\ {\scriptstyle \neg_d}}{\vdash A \vee \neg A, A}\ {\scriptstyle \vee_d}}{\vdash A \vee \neg A}\ {\scriptstyle \vee_d + c_d}}$$

5. $(\neg B \to \neg A) \to (A \to B)$

$$\dfrac{\dfrac{\dfrac{\dfrac{B \vdash B}{\vdash B, \neg B}\,\neg_d \quad \dfrac{A \vdash A}{A, \neg A \vdash}\,\neg_g}{\neg B \to \neg A, A \vdash B}\,\to_g}{\neg B \to \neg A \vdash A \to B}\,\to_d}{\vdash (\neg B \to \neg A) \to (A \to B)}\,\to_d$$

6. $(\neg A \to A) \to A$

$$\dfrac{\dfrac{\dfrac{A \vdash A}{\vdash A, \neg A}\,\neg_d \quad A \vdash A}{\neg A \to A \vdash A}\,\to_g + c_d}{\vdash (\neg A \to A) \to A}\,\to_d$$

Dans l'exemple 1 page précédente, qui ne comporte que des règles droites, la démonstration en calcul des séquents correspond exactement à la démonstration en déduction naturelle. Dans l'exemple 2 page précédente, on peut remarquer une inversion du sens de la démonstration due au remplacement des éliminations par des introductions à gauche : en déduction naturelle la règle de formule principale $B \to C$ précède nécessairement la règle de formule principale $B \to C$, alors qu'en calcul des séquents on a nécessairement l'ordre inverse. Les formules des exemples 3 page précédente, 4 page précédente, 5 et 6 ne sont pas démontrables en logique intuitionniste ; le recours à la règle d'absurdité classique est ici remplacé par l'utilisation de séquents avec deux formules à droite.

Une formulation alternative des règles du calcul des séquents.

Le calcul des séquents admet diverses présentations. La présentation précédente convient à une écriture des règles de haut en bas, elle est adaptée à l'écriture des démonstrations.

Si on a en tête la recherche des démonstrations par analyse de la formule à prouver (écriture des règles de bas en haut), on aura tendance à écrire les règles pour les connecteurs binaires d'une autre manière, en recopiant systématiquement le contexte du séquent conclusion dans les prémisses de la règle pour les règles logiques binaires, et en « superposant » les deux règles unaires. Ainsi pour le connecteur $\to$ les règles deviennent les suivantes :

$$\dfrac{\Gamma, B \vdash \Delta \quad \Gamma \vdash A, \Delta}{\Gamma, A \to B \vdash \Delta}\,\to_g \qquad \dfrac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \to B, \Delta}\,\to_d$$

Dans le cas de la règle $\to_g$, la formulation "montante" correspond d'ailleurs à la façon usuelle de raisonner.

Ces deux formulations des règles du calcul des séquents – que l'on appelle "descendante" et "montante" – sont équivalentes en présence des règles structurelles.

La formulation "montante" s'obtient à partir de la formulation "descendante" à l'aide de contractions :

$$\dfrac{\dfrac{\Gamma, B \vdash \Delta \quad \Gamma \vdash A, \Delta}{\Gamma, \Gamma, A \to B \vdash \Delta, \Delta}}{\Gamma A \to B \vdash \Delta}\,c^* \qquad \dfrac{\dfrac{\dfrac{\Gamma, A \vdash B, \Delta}{\Gamma, A \vdash A \to B, \Delta}}{\Gamma \vdash A \to B, A \to B, \Delta}}{\Gamma \vdash A \to B, \Delta}\,c_d$$

La formulation "descendante" s'obtient à partir de la formulation "montante" à l'aide d'affaiblissements :

$$\dfrac{\dfrac{\Gamma_1, B \vdash \Delta_1}{\Gamma_1, \Gamma_2, B \vdash \Delta_1, \Delta_2}\, w^* \quad \dfrac{\Gamma_2 \vdash A, \Delta_2}{\Gamma_1, \Gamma_2 \vdash A, \Delta_1, \Delta_2}\, w^*}{\Gamma_1, \Gamma_2, A \to B \vdash \Delta_1, \Delta_2} \qquad \dfrac{\dfrac{\Gamma, A \vdash \Delta}{\Gamma, A \vdash B, \Delta}\, w_d}{\Gamma \vdash A \to B, \Delta} \quad \dfrac{\dfrac{\Gamma \vdash B, \Delta}{\Gamma, A \vdash B, \Delta}\, w_g}{\Gamma \vdash A \to B, \Delta}$$

Dans la formulation "montante" on supprime généralement les règles d'affaiblissement en les intégrant dans les axiomes : on prend comme axiomes les séquents $\Gamma \vdash \Delta$ avec Γ et Δ ayant au moins une formule en commun. La manière dont les règles sont écrites fait qu'elles intègrent aussi des contractions (les contextes sont systématiquement contractés) ; cependant la règle de contraction demeure indispensable en calcul des prédicats (cf § 1.3.3 et la preuve de complétude § 1.4 page 48). Cette formulation "montante" du calcul des séquents correspond à une méthode de recherche automatique de preuves appelée *méthode des tableaux* qui procède par analyse de la formule.

Dans le cas propositionnel, on peut réinterpréter cette formulation du calcul des séquents comme une recherche systématique d'une valuation qui rend faux le séquent conclusion (donc les formules de droite vraies et celles de gauche fausses), ce qui est la présentation usuelle de la méthode des tableaux. Une branche d'une déduction partielle correspond à une valuation. L'axiome indique l'impossibilité de trouver une telle valuation sur la branche considérée. Une preuve, dont toutes les branches se termine par des axiomes, peut être vue comme un arbre de recherche systématique de valuations falsifiant le séquent conclusion avec échec dans tous les cas.

Cette interprétation se généralise au calcul des prédicats (voir § 1.4 page 48).

La formulation descendante correspond également à une méthode de recherche automatique de preuves, qui procède par *déduction*, appelée *méthode inverse* par opposition à la méthode des tableaux. Une méthode de recherche analogue est la *résolution*.

Ces deux formulations sont cohérentes d'un point de vue "statique" : écriture ou recherche de preuves. Si l'on s'intéresse à la procédure d'élimination des coupures, c'est à dire la "dynamique" on adoptera des formulations mixtes, plus cohérentes du point de vue de la gestion des règles structurelles (voir § 1.5 page 55).

1.3.3 Structure des déductions en calcul des séquents.

Les considérations de ce paragraphe sont essentiellement destinées à permettre une meilleure compréhension de la structure des déductions en calcul des séquents. Il faut cependant garder à l'esprit qu'elles sont hautement dépendantes de la façon dont les règles sont formulées et qu'elles ne valent pas pour le calcul des séquents intuitionniste qui sera introduit au paragraphe 1.3.5 page 40.

La première propriété structurelle remarquable des déductions en calcul des séquents – qui le distingue des autres systèmes, en particulier de la déduction naturelle – est la possibilité d'inverser l'ordre d'application de certaines règles. En voici un exemple :

$$\dfrac{\dfrac{\dfrac{A \vdash A}{A \wedge B \vdash A}\, \wedge_g}{A \wedge B \vdash A \vee B}\, \vee_d}{\vdash (A \wedge B) \to (A \vee B)}\, \to_d$$

Dans cette déduction, la règle $\rightarrow_d$ est forcément la dernière car sa formule principale contient comme sous-formules les formules principales des autres règles ; en revanche $\wedge_d$ et $\vee_g$ peuvent être inversées, ce qui donne la déduction suivante :

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{A \vdash A \vee B}\,\vee_d}{A \wedge B \vdash A \vee B}\,\wedge_g}{\vdash (A \wedge B) \rightarrow (A \vee B)}\,\rightarrow_d}$$

De façon analogue, toutes les règles du calcul des séquents propositionnel commutent, sauf quand la formule principale de la plus haute est prémisse secondaire de la plus basse. Cette commutation peut engendrer des duplications et se fait alors aux règles structurelles près (suivant la version envisagée). Par exemple :

$$\dfrac{\dfrac{\Gamma_1 \vdash \Delta_1, A, C \quad \Gamma_2 \vdash \Delta_2, B, C}{\Gamma_1, \Gamma_2 \vdash A \wedge B, \Delta_1, \Delta_2, C}\,\wedge_d}{\Gamma_1, \Gamma_2 \vdash A \wedge B, \Delta_1, \Delta_2, C \vee D}\,\vee_d$$

devient en commutant les règles $\vee_d$ et $\wedge_d$

$$\dfrac{\dfrac{\Gamma_1 \vdash \Delta_1, A, C}{\Gamma_1 \vdash \Delta_1, A, C \vee D}\,\vee_d \quad \dfrac{\Gamma_2 \vdash \Delta_2, B, C}{\Gamma_2 \vdash \Delta_2, B, C \vee D}\,\vee_d}{\Gamma_1, \Gamma_2 \vdash A \wedge B, \Delta_1, \Delta_2, C \vee D}\,\wedge_d + c_d$$

Cette propriété devient fausse en calcul des séquents du premier ordre. En effet la commutation de règles pourrait violer la condition de paramètre propre (notée $(*)$ ci-dessus) pour les règles $\forall_g$ et $\exists_d$ comme dans la preuve suivante de $\vdash \forall x \rightarrow \forall x A x$ où l'ordre des règles est contraint :

$$\dfrac{\dfrac{\dfrac{\dfrac{Ax \vdash Ax}{\forall x Ax \vdash xAx}\,\exists_g}{\forall x Ax \vdash \forall x Ax}\,\forall_d}{\vdash \forall x Ax \rightarrow \forall x Ax}\,\rightarrow_d}$$

L'exemple 1.3.3 page ci-contre donne un exemple plus interéssant de non commutation entre règle $\forall_d$ et $\exists_d$.

Ceci a des conséquences sur la recherche de preuves. Quand toutes les règles commutent entre elles, et que l'on cherche une preuve à partir de la conclusion, en version montante, on n'a jamais à se repentir d'un choix de règles. En particulier si l'on considère la version montante du calcul des séquents, le calcul est complet pour la logique classique propositionnelle, même sans les règles structurelles : l'affaiblissement est intégré au niveau des axiomes et la contraction est intégrée dans les règles logiques.

Cette propriété devient fausse en logique du premier ordre, la contraction étant nécessaire en présence de quantificateurs ($\exists x \forall y (Px \rightarrow Py)$ est un exemple de formule qui n'est pas démontrable sans la règle de contraction). En fait les seules contractions concernent les $\forall$ à gauche et les $\exists$ à droite et il suffit d'intégrer la contraction dans les règles $\forall_g$ et $\exists_d$ comme suit :

$$\dfrac{\Gamma, A[t/x], \forall x A \vdash \Delta}{\Gamma \vdash \forall x A, \Delta}\,\forall_g \quad \dfrac{\Gamma \vdash A[t/x], \exists x A, \Delta}{\Gamma \vdash \exists x A, \Delta}\,\exists_d$$

Exemple.

$$
\begin{array}{cl}
Pa, Pb \vdash Px, Pa & \\ \cline{1-1}
Pa \vdash Px, Pb \to Pa & \to_d \\ \cline{1-1}
\vdash Pa \to Px, Pb \to Pa & \to_d \\ \cline{1-1}
\vdash Pa \to Px, \forall y(Py \to Pa) & \forall_d \\ \cline{1-1}
\vdash Pa \to Px, \exists x \forall y(Py \to Px) & \exists_d \\ \cline{1-1}
\vdash \forall y(Py \to Px), \exists x \forall y(Py \to Px) & \forall_d \\ \cline{1-1}
\vdash \exists x \forall y(Py \to Px) & \exists_d
\end{array}
$$

1.3.4 Traduction de la déduction naturelle en calcul des séquents.

Traduction de preuves normales en preuves sans coupures.

On peut formaliser les indications données en début de paragraphe 1.3 page 30 pour obtenir une traduction de la déduction naturelle intuitionniste en calcul des séquents qui transforme une preuve normale en une preuve sans coupures. On va se restreindre aux règles d'introduction et d'élimination du connecteur $\to$, il n'est pas très difficile d'étendre cette traduction aux connecteurs $\wedge$ et $\forall$.

On va définir une fonction φ qui associe à d, une preuve *normale* en déduction naturelle de $\Gamma \vdash C$, $\varphi(d)$, une preuve en calcul des séquents *sans coupures* de $\Gamma \vdash C$. Définissons φ par induction sur la hauteur de la preuve. Si la dernière règle est une règle d'introduction, on la traduit en la règle droite qui est identique.

Supposons maintenant que la dernière règle de la preuve d de $\Gamma \vdash C$ soit une règle d'élimination. Alors, comme la preuve est normale, toutes les règles sur la branche principale de d sont des règles d'élimination. On peut donc supposer que la preuve d a la forme suivante :

$$
\dfrac{
\begin{array}{c}
\dfrac{
\begin{array}{c}
\dfrac{A_1 \to A_2 \to \cdots \to A_i \to \cdots \to A_n \to C \quad \begin{array}{c}\Gamma_1\\ \vdots\\ d_1\\ \vdots\\ A_1\end{array}}{A_2 \to \cdots \to A_i \to \cdots \to A_n \to C}\,\to_e \\
\ddots \\
A_i \to \cdots \to A_n \to C
\end{array}
\quad \begin{array}{c}\Gamma_i\\ \vdots\\ d_i\\ \vdots\\ A_i\end{array}}{A_{i+1} \to \cdots \to A_n \to C}\,\to_e \\
\ddots \\
A_n \to C
\end{array}
\quad \begin{array}{c}\Gamma_n\\ \vdots\\ d_n\\ \vdots\\ A_n\end{array}}{C}\,\to_e
$$

La formule principale de la plus haute règle d'élimination, soit $A_1 \to \cdots \to \cdots A_n \to C$ apparaît nécessairement dans Γ, et on obtient la traduction suivante de d en calcul des séquents, connaissant celles des d_i :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\varphi(d_n)}\ \vdots}{\dfrac{C \vdash C \quad \Gamma_n \vdash A_n}{A_n \to C, \Gamma_n \vdash C}\ \to_g}\ \vdots}{A_{i+1} \to \cdots \to A_n \to C, \Gamma_n, \ldots, \Gamma_{i+1} \vdash C \quad \Gamma_i \vdash A_i}\ \to_g \quad \dfrac{\vdots}{\varphi(d_i)}}{A_i \to \cdots \to A_n \to C, \Gamma_n, \ldots, \Gamma_i \vdash C}\ \vdots \quad \dfrac{\vdots}{\varphi(d_1)}}{A_2 \to \cdots \to A_i \to \cdots \to A_n \to C, \Gamma_n, \ldots, \Gamma_2 \vdash C \qquad \Gamma_1 \vdash A_n}\ \to_g}{A_1 \to A_2 \to \cdots \to A_i \to \cdots \to A_n \to C, \Gamma_n, \ldots, \Gamma_1 \vdash C}$$

(avec $\Gamma = A_1 \to A_2 \to \cdots \to A_i \to \cdots \to A_n \to C, \Gamma_n, \ldots, \Gamma_1$)

La preuve traduite en calcul des séquents lue de "bas en haut" peut être comprise comme une description de la construction de la preuve originale de déduction naturelle, en procédant de "bas en haut" pour les règles d'introduction et de "haut en bas" pour les règles d'élimination. Ceci correspond à la stratégie de recherche d'une preuve normale en déduction naturelle décrite au § 1.2.2 page 15.

On peut généraliser cette traduction aux preuves de déduction naturelle quelconques, en procédant par induction sur le nombre de coupures. Une coupure de déduction naturelle se traduit à l'aide de la règle de coupure du calcul :

$$\dfrac{\dfrac{\Gamma_1, [\![A]\!]}{\vdots}\ d_1 \quad \dfrac{\Gamma_2}{\vdots}\ d_2}{\dfrac{B}{A \to B}\ \to_i \quad A}\ \to_e \qquad \rightsquigarrow \qquad \dfrac{\dfrac{\vdots}{\varphi(d_1)} \quad \dfrac{\vdots}{\varphi(d_2)}}{\Gamma_1, A \vdash B \quad \Gamma_2 \vdash A}\ cut$$
$$B \qquad\qquad \Gamma_1, \Gamma_2 \vdash B$$

Exercice 13 **1.** Préciser la définition de la traduction φ pour les preuves quelconques indiquée ci-dessus.

2. Étendre φ au fragement $\to, \wedge, \forall$ pour les preuves normales en déduction naturelle minimale, puis pour les preuves quelconques, de façon qu'une preuve normale soit traduite en une preuve sans coupures.

Traduction de la déduction naturelle classique.

La traduction φ n'est pas locale, et on ne la généralisera pas facilement au raisonnement par l'absurde. Aussi va-t-on définir une traduction ϕ qui utilise systématiquement la règle de coupure pour les règles d'élimination.

Pour éviter d'introduire des règles de calcul des séquents sur la constante $\bot$, on supposera que les formules sont écrites avec les connecteurs et quantificateurs usuels, mais pas $\bot$. Remarquons que la déduction naturelle utilise tout de même localement la constante $\bot$ pour la

règle d'élimination de la négation. Cette restriction impose que $\bot$ n'est pas vraiment considérée comme une formule : seules les règles pour la négation et les règles d'absurdité classique et intuitionniste peuvent l'utiliser (Cette restriction n'a rien d'essentiel).

On va utiliser par commodité la notation de la déduction naturelle comme déduction de séquents. La fonction ϕ qui a une preuve de déduction naturelle de $\Gamma \vdash C$ associe une preuve en calcul des séquents de $\Gamma \vdash C$, si $C \neq \bot$, de $\Gamma \vdash$ si $C = \bot$, est définie par induction sur le nombre de règles. On vérifie lors de la construction que $\phi(d)$ n'utilise que des formules apparaissant dans d. En particulier les variables ne sont pas modifiées.

Les axiomes et les règles structurelles gauche sont traduites telles quelles. Les règles d'introduction sont traduites également tel quelles, c'est à dire par les règles droite correspondantes. Dans le cas de la règle $\forall_i$, on doit vérifier que la condition $(*)$ sur le paramètre propre est conservée par ϕ, ce qui découle immédiatement de ce que $\phi(d)$ n'utilise que des formules de d.

Les règles d'élimination sont traduites à l'aide de la règle de coupure et de la règle gauche correspondante.

$$\dfrac{\begin{array}{cc}\vdots & \vdots\\ d_1 & d_2\\ \vdots & \vdots\\ \Gamma_1 \vdash A \to B & \Gamma_2 \vdash A\end{array}}{\Gamma_1, \Gamma_2 \vdash B}\ {\scriptstyle \to_e} \quad \rightsquigarrow \quad \dfrac{\begin{array}{cc}\vdots & \vdots\\ \phi(d_1) & \\ \vdots & \\ \Gamma_1 \vdash A \to B & \dfrac{\begin{array}{cc}\vdots & \vdots\\ B \vdash B & \Gamma_2 \vdash A\end{array}}{\Gamma_2, A \to B \vdash B}\ {\scriptstyle \to_g}\end{array}}{\Gamma_1, \Gamma_2 \vdash B}\ {\scriptstyle cut}$$

$$\dfrac{\begin{array}{c}\vdots\\ d\\ \vdots\\ \Gamma \vdash A \wedge B\end{array}}{\Gamma \vdash A}\ {\scriptstyle \wedge_{eg}} \quad \rightsquigarrow \quad \dfrac{\begin{array}{cc}\vdots & \\ \phi(d) & \\ \vdots & \\ \Gamma \vdash A \wedge B & \dfrac{A \vdash A}{A \wedge B \vdash A}\ {\scriptstyle \wedge_{g_g}}\end{array}}{\Gamma \vdash A}\ {\scriptstyle cut}$$

de même pour $\wedge_{ed}$.

$$\dfrac{\begin{array}{c}\vdots\\ d\\ \vdots\\ \Gamma \vdash \forall x A\end{array}}{\Gamma \vdash A[t/x]}\ {\scriptstyle \forall_e} \quad \rightsquigarrow \quad \dfrac{\begin{array}{cc}\vdots & \\ \phi(d) & \\ \vdots & \\ \Gamma \vdash \forall x A & \dfrac{A[t/x] \vdash A[t/x]}{\forall x A \vdash A[t/x]}\ {\scriptstyle \forall_g}\end{array}}{\Gamma \vdash A[t/x]}\ {\scriptstyle cut}$$

$$\dfrac{\begin{array}{ccc}\vdots & \vdots & \vdots\\ d_1 & \phi(d_2) & \phi(d_3)\\ \vdots & \vdots & \vdots\\ \Gamma_1 \vdash A \vee B & \Gamma_2, A \vdash C & \Gamma_3, B \vdash C\end{array}}{\Gamma_1, \Gamma_2, \Gamma_3 \vdash C}\ {\scriptstyle \vee_e} \quad \rightsquigarrow \quad \dfrac{\begin{array}{ccc}\vdots & \vdots & \vdots\\ \phi(d_1) & \phi(d_2) & \phi(d_3)\\ \vdots & \vdots & \vdots\\ \Gamma_1 \vdash A \vee B & \dfrac{\Gamma_2, A \vdash C \quad \Gamma_3, B \vdash C}{\Gamma_2, \Gamma_3, A \vee B \vdash C}\ {\scriptstyle \vee_g} & \end{array}}{\Gamma_1, \Gamma_2, \Gamma_3 \vdash C}\ {\scriptstyle cut}$$

$$\dfrac{\begin{array}{cc}\vdots & \vdots\\ d_1 & \phi(d_2)\\ \vdots & \vdots\\ \Gamma_1 \vdash \exists x A & \Gamma_2, A[y/x] \vdash C\end{array}}{\Gamma_1, \Gamma_2 \vdash C}\ \exists_e * \quad \rightsquigarrow \quad \dfrac{\begin{array}{cc}\vdots & \vdots\\ \phi(d_1) & \phi(d_2)\\ \vdots & \vdots\\ \Gamma_1 \vdash \exists x A & \dfrac{\Gamma_2, A[y/x] \vdash C}{\Gamma_2, \exists x A \vdash C}\ \exists_g *\end{array}}{\Gamma_1, \Gamma_2 \vdash C}\ cut$$

Dans ce dernier cas, la condition $(*)$ est vérifiée pour la traduction à cause de la condition analogue en déduction naturelle pour $\exists_e$, et car $\phi(d_2)$ n'utilise que des formules de d_2.

La règle d'absurdité intuitionniste se traduit par l'affaiblissement droit :

$$\dfrac{\begin{array}{c}\vdots\\ d\\ \vdots\\ \Gamma \vdash \bot\end{array}}{\Gamma \vdash C}\ \bot_e \quad \rightsquigarrow \quad \dfrac{\begin{array}{c}\vdots\\ d\\ \vdots\\ \Gamma \vdash\end{array}}{\Gamma \vdash C}\ w_d$$

La règle d'absurdité classique utilise la coupure et la règle $\neg_d$.

$$\dfrac{\begin{array}{c}\vdots\\ d\\ \vdots\\ \Gamma, \neg A \vdash \bot\end{array}}{\Gamma \vdash A}\ \bot_c \quad \rightsquigarrow \quad \dfrac{\begin{array}{cc}\vdots & \dfrac{}{A \vdash A}\ \neg_d\\ d & \\ \vdots & \\ \Gamma, \neg A \vdash & \vdash \boxed{\neg A, A}\end{array}}{\Gamma \vdash A}\ cut$$

Proposition 1.3.3 *La fonction ϕ définie ci-dessus transforme une preuve d en déduction naturelle de $\Gamma \vdash C$ en une preuve $\phi(d)$ en calcul des séquents de $\Gamma \vdash C$.*

On remarque que les seules règle dont la traduction ne nécessitent pas exactement une formule à droite sont les deux règles d'absurdité intuitionniste et classique ; la règle d'absurdité intuitionniste demande un affaiblissement à droite, la règle d'absurdité classique utilise deux formules à droite.

On en déduit immédiatement qu'un séquent $\Gamma \vdash C$ démontrable en logique intuitionniste est démontrable en calcul des séquents avec au plus une formule à droite et, que si de plus il est démontrable en logique minimale, il a une preuve en calcul des séquents qui n'utilise pas la règle d'affaiblissement à droite. On définira dans la section suivante le *calcul des séquents intuitionniste* qui est un calcul des séquents avec au plus une formule à droite. Pour que cette appellation soit complètement justifiée, on donnera au § 1.3.6 page 43 une traduction du calcul des séquents avec au plus une formule à droite en déduction naturelle intuitionniste.

1.3.5 Calcul des séquents intuitionniste.

La logique intuitionniste a une caractérisation particulièrement simple en calcul des séquents : on considère la version descendante du calcul dont les règles sont restreintes *aux séquents avec au plus une formule à droite*, la contraction à droite est considérée comme implicite.

La contraction droite implicite peut intervenir dans $\vee_g$, $\rightarrow_d$et $\neg_d$. Cependant une version un peu plus restrictive du système suffit :

- Pour $\neg_d$, on peut éliminer le cas où il y a une contraction sur la formule principale et prendre la règle :

$$\frac{\Gamma, A \vdash}{\Gamma \vdash \neg A}\ \neg_d$$

- Pour $\rightarrow_d$, on peut éliminer le cas $\rightarrow_{d_g}$ où il peut y avoir une contraction implicite sur la formule principale en prenant la règle "montante"

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B}\ \rightarrow_d$$

- Pour $\vee_g$, la contraction est inévitable mais on peut néanmoins prendre la version simplifiée suivante (où Δ a au plus un élément).

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma', B \vdash \Delta}{\Gamma, \Gamma', A \vee B \vdash \Delta}\ \vee_g$$

Cela donne le système de calcul des séquents intuitionniste du tableau 1.4 page suivante (avec Δ ayant au plus un élément).

Exercice 14 Vérifier que le calcul des séquents défini précédemment et le calcul des séquents classique restreint aux séquents avec au plus une seule formule à droite permettent bien de déduire les mêmes séquents. Montrer qu'il en va de même quand ces systèmes sont privés de la règle de coupure.

Remarque. Le calcul des séquents intuitionniste peut être formulé avec la constante $\bot$ à la place de la négation. Il suffit de remplacer dans les règles les séquents $\Gamma \vdash$ par $\Gamma \vdash \bot$. La règle $\neg_d$ devient alors un cas particulier de la règle $\rightarrow_d$, la règle $\neg_g$ internalise un affaiblisement à droite :

$$\frac{\Gamma \vdash A}{\Gamma, \neg A \vdash C}\ \neg_g$$

Le système résultant ne comporte que des séquents avec exactement une formule à droite.

Exemples Démonstrations en calcul des séquents intuitionniste.

1. $\vdash ((A \wedge B) \vee C) \rightarrow ((A \vee C) \wedge (B \vee C))$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{A \vdash A \vee C}\ \vee_d}{A \wedge B \vdash A \vee C}\ \wedge_g \quad \dfrac{\dfrac{C \vdash C}{C \vdash A \vee C}\ \vee_d}{}\ \vee_g}{(A \wedge B) \vee C \vdash A \vee C} \quad \dfrac{\dfrac{\dfrac{\dfrac{B \vdash B}{B \vdash B \vee C}\ \vee_d}{A \wedge B \vdash B \vee C}\ \wedge_g \quad \dfrac{\dfrac{C \vdash C}{C \vdash B \vee C}\ \vee_d}{}\ \vee_g}{(A \wedge B) \vee C \vdash B \vee C}}{(A \wedge B) \vee C \vdash (A \vee C) \wedge (B \vee C)}\ \wedge_d + c_g}{\vdash ((A \wedge B) \vee C) \rightarrow ((A \vee C) \wedge (B \vee C))}\ \rightarrow_d$$

2. $\vdash \neg \exists x A \rightarrow \forall x \neg A$

RÈGLES AXIOME / COUPURE

$$\frac{}{A \vdash A}\ \text{ax.} \qquad \frac{\Gamma \vdash A \quad \Gamma', A \vdash \Delta}{\Gamma, \Gamma' \vdash \Delta}\ \textit{cut}$$

RÈGLES LOGIQUES

conjonction

$$\frac{\Gamma, A \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ \wedge_{g_d} \qquad \frac{\Gamma, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ \wedge_{g_g} \qquad \frac{\Gamma \vdash A \quad \Gamma' \vdash B}{\Gamma, \Gamma' \vdash A \wedge B}\ \wedge_d$$

disjonction

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma', B \vdash \Delta}{\Gamma, \Gamma', A \vee B \vdash \Delta}\ \vee_g \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ \vee_{d_g} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ \vee_{d_d}$$

implication

$$\frac{\Gamma, B \vdash \Delta \quad \Gamma' \vdash A}{\Gamma, \Gamma', A \to B \vdash \Delta}\ \to_g \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\ \to_d$$

négation

$$\frac{\Gamma \vdash A}{\Gamma, \neg A \vdash}\ \neg_g \qquad \frac{\Gamma, A \vdash}{\Gamma \vdash \neg A}\ \neg_d$$

quantification universelle

$$\frac{\Gamma, A[t/x] \vdash \Delta}{\Gamma \vdash \forall x A, \Delta}\ \forall_g \qquad \frac{\Gamma \vdash A[y/x]}{\Gamma \vdash \forall x A}\ \forall_d\ (*)$$

quantification existentielle

$$\frac{\Gamma, A[y/x] \vdash \Delta}{\Gamma, \exists x A \vdash \Delta}\ \exists_g\ (*) \qquad \frac{\Gamma \vdash A[t/x]}{\Gamma \vdash \exists x A}\ \exists_d$$

$(*)$ *Restriction* : y n'a pas d'occurrence libre dans le séquent conclusion de la règle.

RÈGLES STRUCTURELLES

Affaiblissement

$$\frac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta}\ w_g \qquad \frac{\Gamma \vdash}{\Gamma \vdash A}\ w_d$$

Contraction

$$\frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta}\ c_g$$

Δ contient au plus une formule

TABLE 1.4 – Règles du calcul des séquents intuitionniste

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A[y/x] \vdash A[y/x]}{A[y/x] \vdash \exists x A}\,\exists_d}{\neg\exists x A, A[y/x] \vdash}\,\neg_g}{\neg\exists x A \vdash \neg A[y/x]}\,\neg_d}{\neg\exists x A \vdash \forall x \neg A}\,\forall_d}{\vdash \neg\exists x A \to \forall x \neg A}\,\to_d$$

3. $\vdash \neg\neg(A \vee \neg A)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{A \vdash A \vee \neg A}\,\vee_d}{\neg(A \vee \neg A), A \vdash}\,\neg_g}{\neg(A \vee \neg A) \vdash \neg A}\,\neg_d}{\neg(A \vee \neg A) \vdash A \vee \neg A}\,\vee_d}{\neg(A \vee \neg A) \vdash}\,\neg_g + c_g}{\vdash \neg\neg(A \vee \neg A)}\,\neg_d$$

Calcul des séquents minimal. On obtient un système de règles pour la logique minimale en prenant le calcul des séquents intuitionniste privé de l'affaiblissement droit. Les exemples de démonstrations données au paragraphe précédent sont en calcul des séquents minimal.

1.3.6 Traduction du calcul des séquents en déduction naturelle.

Traduction du calcul des séquents intuitionniste.

On va définir une fonction ψ qui associe à une preuve π en calcul des séquents intuitionniste du séquent $\Gamma \vdash C$, resp. $\Gamma \vdash$ une preuve en déduction naturelle du séquent $\Gamma \vdash C$, resp. $\Gamma \vdash \bot$. La preuve $\psi(\pi)$ est définie par induction sur π, et la définition vérifiera que $\psi(\pi)$ utilise les mêmes formules que π. On s'inspire de l'interprétation du calcul des séquents comme construction d'une preuve de déduction naturelle qui a été esquissée au début de la section 1.3 page 30.

On utilise la formulation de la déduction naturelle comme déduction de séquents. Dans la suite Δ contient au plus une formule, Si Δ contient exactement une formule Δ^* désigne celle-ci, si Δ est vide Δ^* désigne $\bot$.

Les règles droites et les règles structurelles à gauche sont traduites telles quelles, c'est à dire en ce qui concerne les règles droites renommées en la règle d'introduction correspondante.

La règle d'affaiblissement à droite est traduite par l'absurdité intuitionniste.

$$\dfrac{\begin{array}{c}\vdots\\ \pi \\ \vdots\end{array}}{\dfrac{\Gamma \vdash}{\Gamma \vdash C}\,w_d} \quad \rightsquigarrow \quad \dfrac{\begin{array}{c}\vdots\\ \psi(\pi) \\ \vdots\end{array}}{\dfrac{\Gamma \vdash \bot}{\Gamma \vdash C}\,\bot_e}$$

Les règles gauche du $\vee$ et du $\exists$ s'avèrent être des formes particulières des règles d'élimination correspondantes. On traduit donc trivialement ces règles en demandant à la prémisse principale de la règle d'élimination d'être un axiome :

$$\dfrac{\begin{array}{cc}\vdots\,\pi_1 & \vdots\,\pi_2\\ \Gamma_1, A \vdash \Delta & \Gamma_2, B \vdash \Delta\end{array}}{\Gamma_1, \Gamma_2, A \vee B \vdash C}\ \vee_g \quad \rightsquigarrow \quad \dfrac{A \vee B \vdash A \vee B \quad \begin{array}{cc}\vdots\,\psi(\pi_1) & \vdots\,\psi(\pi_2)\\ \Gamma_1, A \vdash \Delta^* & \Gamma_2, B \vdash \Delta^*\end{array}}{\Gamma_1, \Gamma_2 \vdash C}\ \vee_e$$

$$\dfrac{\begin{array}{c}\vdots\,\pi\\ \Gamma, A[y/x] \vdash C\end{array}}{\Gamma, \exists x A \vdash C}\ \exists_g * \quad \rightsquigarrow \quad \dfrac{\exists x A \vdash \exists x A \quad \begin{array}{c}\vdots\,\psi(\pi)\\ \Gamma, A[y/x] \vdash C\end{array}}{\Gamma \vdash C}\ \exists_e *$$

Pour les autres connecteurs la traduction n'est plus locale. On se sert de la propriété suivante des preuves de déduction naturelle : dans une preuve de $\Gamma, H \vdash C$, la formule H est introduite nécessairement par un axiome $H \vdash H$ ou par une règle d'affaiblissement.

Voyons la traduction de la règle $\rightarrow_g$:

$$\dfrac{\begin{array}{cc}\vdots\,\pi_1 & \vdots\,\pi_2\\ \Gamma_1, B \vdash \Delta & \Gamma_2 \vdash A\end{array}}{\Gamma_1, \Gamma_2, A \rightarrow B \vdash \Delta}\ \rightarrow_g \quad \rightsquigarrow \quad \begin{array}{c}\vdots\,\psi(\pi_2)\\ \dfrac{A \rightarrow B \vdash A \rightarrow B \quad \Gamma_2 \vdash A}{A \rightarrow B, \Gamma_2 \vdash B}\ \rightarrow_e\\ \vdots\,\psi(\pi_1)\\ \Gamma_1, \Gamma_2, A \rightarrow B \vdash \Delta^*\end{array}$$

La notation ci-dessus signifie que *toutes les occurrences de l'axiome* $B \vdash B$ dans la preuve $\psi(\pi_1)$ sont remplacées par la preuve de $\Gamma_2, A \rightarrow B \vdash B$ obtenue à partir de $\psi(\pi_2)$ et de $A \rightarrow B$, (en fait toutes les occurrences qui ont été contractées en l'occurrence de B dans la conclusion de $\psi(\pi_1)$). En particulier, si B est issue d'un affaiblissement la preuve $\psi(\pi_2)$ est simplement effacée, si B est issue d'une contraction elle est dupliquée.

Avec les mêmes conventions de notation on traduit $\wedge_g$ et $\forall_g$:

$$\dfrac{\begin{array}{c}\vdots\,\pi\\ \Gamma, A \vdash \Delta\end{array}}{\Gamma, A \wedge B \vdash \Delta}\ \wedge_{g_g} \quad \rightsquigarrow \quad \begin{array}{c}\dfrac{A \wedge B \vdash A \wedge B}{A \wedge B \vdash A}\ \wedge_{eg}\\ \vdots\,\psi(\pi)\\ \Gamma, A \wedge B \vdash \Delta^*\end{array}$$

La règle $\wedge_{g_d}$ se traite de façon identique, et pour la règle $\forall_g$:

$$\dfrac{\begin{array}{c}\vdots\\ \pi\\ \vdots\\ \Gamma, A[t/x] \vdash \Delta\end{array}}{\Gamma, \forall x A \vdash \Delta}\ \forall_g \qquad \rightsquigarrow \qquad \dfrac{\dfrac{}{\forall x A \vdash \forall x A}\ \forall_e}{\begin{array}{c}\forall x A \vdash A[t/x]\\ \vdots\\ \psi(\pi)\\ \vdots\\ \Gamma, \forall x A \vdash \Delta^*\end{array}}$$

Enfin la coupure se traduit, toujours avec la même convention (effacement ou duplication éventuels de la preuve $\phi(\pi_1)$) par :

$$\dfrac{\begin{array}{cc}\vdots & \vdots\\ \pi_1 & \pi_2\\ \vdots & \vdots\\ \Gamma_1 \vdash A & \Gamma_2, A \vdash \Delta\end{array}}{\Gamma_1, \Gamma_2 \vdash \Delta}\ cut \qquad \rightsquigarrow \qquad \begin{array}{c}\vdots\\ \psi(\pi_1)\\ \vdots\\ \Gamma_1 \vdash A\\ \vdots\\ \psi(\pi_2)\\ \vdots\\ \Gamma_1, \Gamma_2 \vdash \Delta^*\end{array}$$

Proposition 1.3.4 *La fonction ψ définie ci-dessus transforme une preuve π en calcul des séquents intuitionniste de $\Gamma \vdash C$, resp. $\Gamma \vdash$, en une preuve $\psi(\pi)$ en déduction naturelle intuitionniste de $\Gamma \vdash C$, resp. $\Gamma \vdash \bot$.*

On déduit de ce résultat et de la proposition 1.3.3 page 40 l'équivalence de la prouvabilité en calcul des séquents intuitionniste et en déduction naturelle intuitionniste :

Proposition 1.3.5 *Un séquent $\Gamma \vdash C$ est prouvable en déduction naturelle intuitionniste si et seulement s'il est prouvable en calcul des séquents intuitionniste ; le résultat est identique pour la logique minimale.*

Exercice 15 Montrer que si π est une preuve sans coupures, $\psi(\pi)$ est une preuve normale. Donner un exemple où la réciproque est fausse.

Exercice 16 On se restreint au fragment $\rightarrow, \wedge, \forall$. Montrer que, si φ est la traduction de la déduction naturelle définie au § 1.3.4 page 37 et étendue à la section 13 page 38, alors pour toute preuve normale en déduction naturelle minimale d on a $\psi(\varphi(d)) = d$. Trouver une preuve en calcul des séquents minimal sans coupures π telle que $\varphi(\psi(\pi)) \neq \pi$.

Traduction du calcul des séquents classique.

Le calcul des séquents classique ayant plusieurs formules à droite et la déduction naturelle une seule, on pourrait penser à choisir l'une d'entre elles comme conclusion courante, les autres formules à droites étant transformées en hypothèses négatives. Quand la formule principale d'une règle droite n'est pas la conclusion courante, il faut un raisonnement par l'absurde pour changer de conclusion. La contraction à droite est mimée à l'aide du raisonnement par l'absurde par la contraction à gauche sur la négation des formules contractées.

Une façon systématique de procéder est de traduire un séquent $\Gamma \vdash \Delta$ par le séquent $\Gamma, \neg\Delta \vdash \bot$, où $\neg\Delta$ est par convention $\{\neg A \ / \ A \in \Delta\}$.

Exercice 17 Montrer que si $\Gamma \vdash \Delta$ est dérivable en calcul des séquents classique, alors $\Gamma, \neg\Delta \vdash \bot$ est dérivable en déduction naturelle classique.

1.3.7 Quelques propriétés des preuves sans coupures en calcul des séquents.

On verra dans la section suivante qu'un séquent prouvable a une preuve sans coupures. Cela a des conséquences évidentes pour la recherche de preuves.

Étudions quelques propriétés des preuves sans coupures.

1.3.8 Propriété de la sous-formule.

On peut affiner la notion de sous-formule de façon à tenir compte de la façon suivante :

Définition 1.3.6 On définit inductivement $\mathcal{S}^+(F)$ et $\mathcal{S}^-(F)$ les ensembles des *sous-formules positives et négatives* d'une formule F donnée.

Pour α atomique

$\mathcal{S}^+(\alpha) = \{\alpha\}$ $\qquad$ $\mathcal{S}^-(\alpha) = \emptyset$

$\mathcal{S}^+(\neg A) = \{\neg A\} \cup \mathcal{S}^-(A)$ $\qquad$ $\mathcal{S}^-(\neg A) = \mathcal{S}^+(A)$

$\mathcal{S}^+(A \wedge B) = \{A \wedge B\} \cup \mathcal{S}^+(A) \cup \mathcal{S}^+(B)$ $\qquad$ $\mathcal{S}^-(A \wedge B) = \mathcal{S}^-(A) \cup \mathcal{S}^-(B)$

$\mathcal{S}^+(A \vee B) = \{A \vee B\} \cup \mathcal{S}^+(A) \cup \mathcal{S}^+(B)$ $\qquad$ $\mathcal{S}^-(A \vee B) = \mathcal{S}^-(A) \cup \mathcal{S}^-(B)$

$\mathcal{S}^+(A \to B) = \{A \to B\} \cup \mathcal{S}^-(A) \cup \mathcal{S}^+(B)$ $\qquad$ $\mathcal{S}^-(A \to B) = \mathcal{S}^+(A) \cup \mathcal{S}^-(B)$

$\mathcal{S}^+(\forall x A) = \{\forall x A\} \cup \bigcup_{y \text{ variable}} \mathcal{S}^+(A[y/x])$ $\qquad$ $\mathcal{S}^-(\forall x A) = \bigcup_{t \text{ terme}} \mathcal{S}^-(A[t/x])$

$\mathcal{S}^+(\exists x A) = \{\exists x A\} \cup \bigcup_{t \text{ terme}} \mathcal{S}^+(A[t/x])$ $\qquad$ $\mathcal{S}^-(\exists x A) = \bigcup_{y \text{ variable}} \mathcal{S}^-(A[y/x])$

Une sous-formule de F est une sous-formule positive ou négative de F.

Remarquons qu'une formule propositionnelle n'a qu'un nombre fini de sous-formules, mais que n'importe quel terme peut apparaître dans la sous-formule d'une formule contenant un quantificateur.

Le calcul des séquents sans coupures a clairement la propriété de la sous-formule, et on peut affiner celle-ci :

Proposition 1.3.7 *Dans une preuve sans coupures d'un séquent $\Gamma \vdash \Delta$ n'appararaissent que des séquents constitués de sous-formules des formules de Γ et Δ. De plus si ces sous-formules apparaissent à droite du signe $\vdash$, ce sont des sous-formules positives des formules de Δ ou*

négatives des formules de Γ *; si si ces sous-formules apparaissent à gauche du signe* $\vdash$*, ce sont des sous-formules négatives des formules de* Δ *ou positives des formules de* Γ*.*

Cette proposition se vérifie immédiatement règle par règle par induction sur la hauteur d'une preuve sans coupures.

La recherche de preuves sans coupures peut se faire "en remontant". La propriété de la sous-formule permet de construire une méthode de recherche de preuves "en descendant". Pour une preuve de $\vdash F$, on sait que l'on n'utilisera que des instances des règles du calcul des séquents pour des sous-formules de F, règles droites pour les sous-formules positives, gauches pour les sous-formules négatives. On peut procéder alors par "saturation" en appliquant ces règles à partir des axiomes jusqu'à trouver F. C'est, très grossièrement décrit, le principe d'une méthode de recherche automatique de preuve appellée *méthode inverse* due à Maslov, appellée ainsi par opposition à la méthode des tableaux.

1.3.9 Preuves en présence d'axiomes non logiques.

Tant qu'une théorie a un nombre fini d'axiomes, il est toujours possible de remplacer la preuve de F dans la théorie T par la preuve du séquent $T \vdash F$, mais ce n'est plus possible en présence d'une infinité d'axiomes.

On est donc amené à étendre la notion de déduction, en ajoutant au calcul des séquents des axiomes non logiques (qui sont des séquents). Cela a clairement un sens si les séquents axiomes sont composés de formules closes. Il est utile de manipuler aussi des axiomes constitués de formules non closes. Pour que cela ait un sens, on considère d'habitude qu'il s'agit de clôtures universelles, mais ici il s'agit de séquents et non de formules. On définit donc un *ensemble d'axiomes non logiques pour le calcul des séquents* comme un ensemble de séquents stable par substitution des variables libres par des termes quelconques du langage.

Concrètement on représentera de tels ensembles de séquents dans le langage $\mathcal{L}$ sans variables et étendu avec des "meta-variables" que l'on notera $?x$, $?y$, ... Si dans $\Gamma \vdash \Delta$ n'apparaissent que les variables $?x_1, \ldots, ?x_n$, on l'interprète par

$$\left\{\Gamma\left[\overline{t}/\overline{?x}\right] \vdash \Delta\left[\overline{t}/\overline{?x}\right] \ / \ t_1 \ldots t_n \text{ termes de } \mathcal{L}\right\} \ .$$

En présence d'axiomes non logiques, il n'est naturellement pas possible d'éliminer la règle de coupure. On a cependant une généralisation naturelle de preuve sans coupures : une preuve sans coupures autres que sur les formules des axiomes non logiques. On verra dans la section suivante qu'un séquent conséquence d'un ensemble d'axiomes non logiques au une preuve de ce type.

Ce résultat n'a d'intérêt que si cela représente une vraie limitation pour la règle de coupures. Par exemple dans l'arithmétique de Peano, le schéma de récurrence, que l'on peut exprimer ainsi :

$$A[0/x], \forall y(A[y/x] \to A[sy/x]) \vdash A \quad \text{pour une formule quelconque } A$$

fait intervenir toutes les formules du langage.

De même pour l'égalité si on utilise le schéma :

$$A[t/x],\ t = u \vdash A[u/x] \quad \text{pour une formule quelconque } A$$

mais dans ce dernier cas, il est possible de restreindre le schéma d'égalité aux formules atomiques.

Un cas particulier intéressant est la restriction aux séquents constitués de formules atomiques. En présence d'axiomes non logiques de cette forme, et pour prouver de tels séquents, on s'interessera aux preuves sans coupures autres que sur les atomes des axiomes non logiques, ce qui, vu la propriété de la sous-formule, revient à chercher des preuves n'utilisant que la règle de coupure sur les atomes des axiomes non logiques. On a ainsi un moyen de formaliser *méthode de résolution*, une méthode de recherche automatique de preuve due à Robinson. Il s'agit d'une méthode de recherche de preuve par saturation ("en descendant"). En général on cherche a prouver le séquent vide. Les séquents axiomes sont exprimés avec des meta-variables tel qu'indiqué ci-dessus. La règle de résolution s'applique directement à ces séquents, c'est la règle de coupure accompagnée d'un choix judicieux des instances de méta-variables :

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma', A' \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'} \ \sigma(A) = \sigma(A')$$

Note : cette façon de noter la résolution permet un traitement cohérent avec la calcul des séquents, mais n'est pas la plus usitée. Pour retrouver la notation usuelle, il suffit de remplacer les séquents constitués de formules atomiques avec méta-variables par des *clauses*, c'est à dire des disjonctions de *littéraux*, les littéraux étant des formules atomiques (partie droite du séquent) ou négations de formules atomiques (partie gauche du séquent). Les méta-variables sont remplacées par les variables du langages et les formules avec variables libres sont interprétées par leur clôture universelle. La règle de résolution s'exprime alors aisément dans ce formalisme.

1.4 Complétude en calcul des séquents : une preuve sémantique de l'existence d'une preuve sans coupures.

1.4.1 Préliminaires.

On va montrer directement l'existence d'une preuve sans coupures pour une formule universellement valide en logique classique.

Joint à la correction du calcul des séquents, ce résultat permet de démontrer qu'une formule démontrable en calcul des séquents est démontrable sans coupures. On donnera au § 1.5 page 55 des procédés de calcul qui permettent de montrer directement ce dernier résultat, aussi bien pour le calcul des séquents classique que pour le calcul des séquents intuitionniste ou minimal.

Il n'y a pas grande différence vis à vis d'une preuve de complétude de complétude usuelle, ou par exemple d'une preuve de complétude du calcul des séquents avec coupures, si ce n'est que pour définir un contre-modèle d'une formule F, en un certain sens on se contentera d'évaluer les sous-formules de F.

La méthode que nous allons utiliser pourrait aussi bien se formuler en méthode des tableaux. On peut essayer d'en donner l'intuition. Comme d'une part la plupart des règles commutent en calcul des séquents et que d'autre part on a la règle de contraction à droite et à gauche, il est possible de construire pour une formule F donnée un arbre de séquents potentiellement infini tel que chaque nœud corresponde à une règle du calcul des séquents et tel que si la formule F est prouvable alors une section finie de cet arbre est une preuve en calcul des séquents.

Quand F n'est pas prouvable l'arbre en question est nécessairement infini. Il possède, étant à branchement fini une branche infinie. Cette branche permet de construire un contre-modèle de F en prenant l'ensemble des termes comme ensemble de base et en validant toutes les formules qui apparaissent du côté gauche d'un séquent sur cette branche.

Cette méthode met l'accent sur l'importance des commutations de règles et de la règle de contraction pour la simplicité de la sémantique de la logique classique. Elle permettra a contrario de comprendre la plus grande complexité et la diversité des sémantique de la logique intuitionniste ou modale (comme la sémantique de Kripke, voir partie **??** page ??).

1.4.2 Le théorème de complétude.

Le langage choisi $\mathcal{L}$ est fini ou dénombrable. On se place en calcul des prédicats pur (sans égalité). On rappelle que l'on a défini la satisfaction d'un séquent quelconque (non nécessairement clos) voir définition 1.3.1 page 30.

Théorème 1.4.1 (complétude faible) *Un séquent Σ est démontrable en calcul des séquents sans coupures si et seulement si il est valide dans toutes les $\mathcal{L}$-structures.*

Comme un séquent contient un nombre fini de formules, ce théorème est plus faible que le théorème de complétude de Gödel. Il n'a pas pour conséquence le théorème de compacité du calcul des prédicats (voir [Cori-Lascar 88]).

On aura essentiellement la même preuve pour un théorème en présence d'axiomes non logiques (voir section précédente).

Théorème 1.4.2 (complétude) *Soit $\mathcal{T}$ un ensemble dénombrable d'axiomes non logiques pour le calcul des séquents (séquents stables par substitution de termes aux variables libres), alors le séquent Σ est conséquence de $\mathcal{T}$ en calcul des séquents avec coupure restreinte aux formules apparaissant dans les séquents de $\mathcal{T}$ si et seulement s'il est conséquence sémantique de $\mathcal{T}$, c'est à dire que toute $\mathcal{L}$-structure satisfaisant $\mathcal{T}$ satisfait Σ.*

Ce théorème a pour corollaire immédiat le théorème 1.4.1 qui est le cas particulier où $\mathcal{T}$ est vide.

Pour les applications que nous avons en vue la restriction aux ensembles d'axiomes $\mathcal{T}$ dénombrable ne pose pas de problèmes, mais le résultat reste vrai sans cette restriction. Par ailleurs la preuve qui suit va utiliser assez fortement la dénombrabilité.

La partie "correction" du théorème de complétude, à savoir que s'il existe une preuve de Σ en calcul des séquents sous hypothèses $\mathcal{T}$ alors Σ est conséquence sémantique de $\mathcal{T}$, est vraie sans restriction sur la règle de coupure et se vérifie immédiatement règle par règle par induction sur la hauteur de la preuve.

On va montrer la réciproque par contraposée.

1.4.3 Construction de l'arbre de recherche de preuve.

La première étape consiste à construire pour un séquent Σ un arbre éventuellement infini de recherche d'une preuve en calcul des séquents de Σ en présence des axiomes $\mathcal{T}$. Les règles $\forall_g$ et $\exists_d$ posent un problème particulier, puisqu'il existe une infinité de prémisses possibles pour ces règles. On se donne donc une énumération des termes du langage $\mathcal{L}$ soit $\{t_i \ / \ i \in \mathbb{N}\}$. La règle de coupure pose un problème analogue, on se donne une énumération des formules

apparaissant dans $\mathcal{T}$, soit $\{C_i \ / \ i \in \mathbb{N}\}$. Enfin pour les règles $\forall_d$ et $\exists_g$, on rappelle que le langage utilise une infinité de variables $\{x_i \ / \ i \in \mathbb{N}\}$.

Il s'agit maintenant de donner une construction de l'arbre qui vérifie que sur toute branche infinie de l'arbre de recherche de preuve :

- Chaque sous-formule du séquent conclusion ou des axiomes non logiques apparaît au moins une fois comme formule principale d'une règle ;
- chaque formule C_i apparaît au moins une fois comme formule principale d'une règle de coupure ;
- chaque chaque sous-formule positive existentielle d'un séquent de la branche (en particulier du séquent conclusion), apparaît alors une infinité de fois comme formule principale d'une règle $\exists_d$, chaque terme t_i apparaissant comme témoin au moins une fois ;
- chaque sous-formule négative existentielle d'un séquent de la branche (en particulier du séquent conclusion) apparaît une infinité de fois comme formule principale d'une règle $\forall_g$, chaque terme t_i apparaissant comme témoin au moins une fois.

La façon d'obtenir ceci n'a pas grande importance. Pour déterminer la règle à utiliser (en montant), on va d'abord considérer pour cette construction que le séquent $\Gamma \vdash \Delta$, est constitué de deux suites *ordonnées* Γ et Δ (et non des multi-ensembles) ; on va ensuite annoter le séquent par 4 chiffres, le premier indique s'il faut appliquer une règle gauche droite ou de coupure, le second indique l'indice de l'énumération des formules de coupure, le troisième indique l'indice de l'énumération des termes pour les règles $\forall_g$, le quatrième pour les règles $\exists_d$.

L'arbre de recherche, construit à partir de la racine, est alors entièrement déterminé d'une part par la donnée des règles suivantes du tableau 1.5 page suivante, avec la convention que si la règle axiome peut être utilisée, aucune autre règle ne s'applique, d'autre part par un ordre arbitraire sur les parties gauches et droite du séquent Σ avec l'annotation $0, 0, 0, 0$.

On internalise les contractions aux règles, il faut le faire explicitement pour les deux règles $\forall_g$ et $\exists_d$. Une contraction explicite peut-être nécessaire pour les autres règles afin de permettre la détection des axiomes non logiques (formules entre crochets).

Le séquent annoté racine (correspondant à Σ) s'écrit :

$$\Gamma_0 \vdash^{0,0,0,0} \Delta_0$$

Les formules notées entre crochets (contraction) sont présentes uniquement si elles sont identiques à l'une des formules C_i. En particulier si $\mathcal{S}$ est vide ces contractions sont inutiles.

On pourrait même ajouter les crochets à la syntaxe et convenir qu'il n'est de ne pas appliquer une règle à une formule entre crochets.

Tout cela n'a aucune importance pour le théorème de complétude visé. Bien-sûr on aurait pu faire la contraction systématiquement, mais la restriction donnée aura l'avantage de conduire à un semi-algorithme de recherche de preuves "plus efficace" (autant que l'on puisse parler d'efficacité pour un problème non décidable). Ici la principale source d'inefficacité du semi-algorithme induit est le choix d'énumérer tous les t_i pour les règles $\forall_g$ et $\exists_d$.

1.4.4 Propriétés de l'arbre de recherche de preuve.

En oubliant l'ordre dans la définition des séquents, les décorations et la règle d'échange qui devient l'identité, un arbre fini utilisant ces règles devient une preuve en calcul des séquents version montante, avec contraction explicite intégrée aux règles $\forall_g$ et $\exists_d$ et affaiblissements intégrés aux axiomes (logiques ou non).

$$\frac{}{\Gamma_1, \alpha, \Gamma_2 \vdash^{c,n,p,q} \Delta_1, \alpha, \Delta_2} \text{ Ax. } \alpha \text{ formule atomique} \qquad \frac{}{\Gamma' \vdash^{c,n,p,q} \Delta'} \text{ Ax. } \Gamma \subset \Gamma', \Delta \subset \Delta' \text{ et } \Gamma \vdash \Delta \in \mathcal{T}$$

$$\frac{\Gamma, A, B, [A \wedge B] \vdash^{2,n,p,q} \Delta}{A \wedge B, \Gamma \vdash^{1,n,p,q} \Delta} \wedge_g \qquad \frac{\Gamma \vdash^{1,n,p,q} \Delta, A, [A \wedge B] \quad \Gamma \vdash^{1,n,p,q} \Delta, B, [A \wedge B]}{\Gamma \vdash^{0,n,p,q} A \wedge B, \Delta} \wedge_d$$

$$\frac{\Gamma, A, [A \vee B] \vdash^{2,n,p,q} \Delta \quad \Gamma, B, [A \vee B] \vdash^{2,n,p,q} \Delta}{A \vee B, \Gamma \vdash^{1,n,p,q} \Delta} \vee_g \qquad \frac{\Gamma \vdash^{1,n,p,q} \Delta, A, B, [A \vee B]}{\Gamma \vdash^{0,n,p,q} A \vee B, \Delta} \vee_d$$

$$\frac{\Gamma, B, [A \to B] \vdash^{2,n,p,q} \Delta \quad \Gamma \vdash^{2,n,p,q}, [A \to B]\Delta, A}{A \to B, \Gamma \vdash^{1,n,p,q} \Delta} \to_g \qquad \frac{\Gamma, A \vdash^{1,n,p,q} \Delta, B, [A \to B]}{\Gamma \vdash^{0,n,p,q} A \to B, \Delta} \to_d$$

$$\frac{\Gamma, [\neg A] \vdash^{2,n,p,q} \Delta, A}{\neg A, \Gamma \vdash^{1,n,p,q} \Delta} \neg_g \qquad \frac{\Gamma, A \vdash^{1,n,p,q} \Delta, [\neg A]}{\Gamma \vdash^{0,n,p,q} \neg A, \Delta} \neg_d$$

$$\frac{\Gamma, A[t_q/x], \forall x A \vdash^{2,n,p,q+1} \Delta}{\forall x A, \Gamma \vdash^{1,n,p,q} \Delta} \forall_g \qquad \frac{\Gamma \vdash^{1,n,p,q} \Delta, A[x_i/x], [\forall x A]}{\Gamma \vdash^{0,n,p,q} \forall x A, \Delta} \forall_d\ (*)$$

$$\frac{\Gamma, A[x_i/x], [\exists x A] \vdash^{2,n,p,q} \Delta}{\exists x A, \Gamma \vdash^{1,n,p,q} \Delta} \exists_g\ (*) \qquad \frac{\Gamma \vdash^{1,n,p+1,q} \Delta, A[t_p/x], \exists x A}{\Gamma \vdash^{0,n,p,q} \exists x A, \Delta} \exists_d$$

$(*)$: i le plus petit j tel que x_j n'est pas encore apparue .

$$\frac{\Gamma, \alpha \vdash^{2,n,p,q} \Delta}{\alpha, \Gamma \vdash^{1,n,p,q} \Delta} \text{ échange à gauche } \alpha \text{ atomique} \qquad \frac{\Gamma \vdash^{1,n,p,q} \Delta, \alpha}{\Gamma \vdash^{0,n,p,q} \alpha, \Delta} \text{ échange à droite } \alpha \text{ atomique}$$

$$\frac{\vdash^{2,n,p,q} \Delta}{\vdash^{1,n,p,q} \Delta} \text{ partie gauche vide} \qquad \frac{\Gamma \vdash^{1,n,p,q}}{\Gamma \vdash^{0,n,p,q}} \text{ partie droite vide}$$

$$\frac{\Gamma \vdash^{0,n+1,p,q} \Delta, C_n \quad \Gamma, C_n \vdash^{0,n+1,p,q} \Delta}{\Gamma \vdash^{2,n,p,q} \Delta}\ cut$$

TABLE 1.5 – Construction de l'arbre de recherche de preuves

On a donc :

Lemme 1.4.3 *Si la hauteur de l'arbre de recherche de preuve de Σ sous hypothèses $\mathcal{T}$ est finie, alors il représente une preuve en calcul des séquents de Σ sous hypothèses $\mathcal{T}$.*

Démonstration. La seule règle qui à un séquent n'associe pas d'autre séquent est la règle axiome. Si l'arbre est fini toutes les feuilles sont des axiomes, et décrit une preuve de calcul des séquents. ■

Dans la suite on va appeller *chaîne de déduction* une branche de l'arbre de recherche de preuve, un peu plus précisément :

Définition 1.4.4 Une chaîne de déduction de Σ sous hypothèses $\mathcal{S}$ est une suite finie ou dénombrable de séquents $(\Sigma_i,\ i < \gamma)$, avec $\gamma \in \mathbb{N}$ ou $\gamma = \omega$ vérifiant :
- $\Sigma_0 = \Sigma$;
- Σ_{i+1} est l'une des prémisses de la seule règle de conclusion Σ_i dans la construction de l'arbre de recherche de preuve, quand on ajoute les décorations indiquées.

Remarque. Les règles ont été écrites de façon que la propriété suivante soit vraie :
Si $(\Sigma_i,\ i < \gamma)$ est une chaîne de déduction on a :

$$\text{si } i > j \text{ alors } \Sigma_j \text{ a pour conséquence } \Sigma_i$$

On peut maintenant donner un énoncé du lemme de König dans ce cas particulier :

Lemme 1.4.5 (König) *Si la hauteur de l'arbre est infinie, il existe une chaîne de déduction infinie.*

Démonstration. L'arbre de preuve est à branchement fini. Sachant que l'arbre est de hauteur infinie, on construit la chaîne de déduction par récurrence à partir de la racine, en choisissant dans le cas de deux prémisses une premisse racine d'un sous-arbre de hauteur infinie. Cette démonstration utilise l'axiome du choix dénombrable.

Pour terminer la preuve il nous faut démontrer le lemme suivant :

Lemme 1.4.6 *S'il existe une chaîne de déduction infinie, soit $(\Sigma_i)_{i\in\mathbb{N}}$ de Σ sous hypothèses $\mathcal{S}$, alors il existe une $\mathcal{L}$-structure $\mathcal{M}$ de base dénombrable vérifiant :*
- $\forall S \in \mathcal{S},\ \mathcal{M} \models S$;
- $\forall i \in \mathbb{N},\ \mathcal{M} \not\models \Sigma_i$.

Démonstration (complétude). On déduit des trois lemmes précédents le théorème de complétude. S'il n'y a pas de preuve en calcul des séquents de Σ sous hypothèses $\mathcal{S}$, en particulier il n'y en a pas dans le calcul des séquents défini au § 1.4.3 page 49 ci-dessus pour un ordre arbitraire sur les formules droite et gauche de Σ. D'après le lemme 1.4.3 l'arbre est donc de hauteur infinie, d'après le lemme 1.4.5 il existe une chaîne de déduction infinie, et d'après le lemme 1.4.6 cet ensemble définit un contre-modèle qui valide les séquents axiomes $\mathcal{S}$ et ne valide pas Σ. ■

Reste donc à démontrer le lemme 1.4.6 page ci-contre. On se donne pour les deux lemmes qui suivent une chaîne de déduction infinie $(\Sigma_i)_{i\in\mathbb{N}}$.

Le premier lemme correspond au prérecquis pour la construction de l'arbre de recherche de preuve annoncé au début du § 1.4.3 page 49 :

Lemme 1.4.7

1. *Si la formule F apparaît dans un Σ_i, alors il existe $j \geq i$ telle qu'elle apparaît en position active dans Σ_j.*
2. *Chaque formule des séquents de $\mathcal{S}$ apparaît soit à gauche soit à droite dans l'un des séquents Σ_i.*
3. *Si $\forall x F$ apparaît positivement, resp. $\exists x F$ négativement dans un séquent Σ_i, alors pour tout terme t_n il existe $j \geq i$ tel que $F[t_i/x]$ apparaît positivement, resp. négativement, dans Σ_j.*

Le second lemme donne la construction du contre-modèle :

Lemme 1.4.8 *Soit $\mathcal{M}$ la $\mathcal{L}$-structure d'ensemble de base $\{\overline{t_i} \ / \ i \in \mathbb{N}\}$, avec les symboles de fonction interprétés naturellement (f symbole de fonction d'arité p) :*

$$\overline{f\overline{t_1}\dots\overline{t_p}}^{\mathcal{M}} = \overline{ft_1\dots t_p}$$

et les symboles de prédicats par (P symbole de prédicat d'arité p) :

$$\mathcal{M} \models P\overline{u_1}\dots\overline{u_p} \text{ ssi il existe } i \in \mathbb{N} \text{ tel que } Pu_1\dots u_p \text{ apparaît à gauche dans } \Sigma_i \ .$$

Alors pour toute formule F à p variables libres $x_1, \dots, x_p$
- *Si F apparaît à gauche dans les Σ alors $\mathcal{M} \models F[\overline{x_1}/x_1, \dots, \overline{x_p}/x_p]$;*
- *Si F apparaît à droite dans les Σ alors $\mathcal{M} \not\models F[\overline{x_1}/x_1, \dots, \overline{x_p}/x_p]$;*

Remarquons que ce lemme ne fournit pas une évaluation de toutes les formules, justement parce que l'on a restreint l'usage de la règle de coupure.

Ces deux lemmes permettent de prouver le lemme 1.4.6 page ci-contre.

Démonstration (lemme 1.4.6 page précédente). En effet on choisit le modèle fournit par le lemme 1.4.8. Vu la définition de l'interprétation d'un séquent $\mathcal{M} \not\models \Sigma_i$, en particulier $\mathcal{M} \not\models \Sigma = \Sigma_0$.

Soit maintenant S un séquent de $\mathcal{S}$. Procédons par l'absurde. Supposons que $\mathcal{M} \not\models S$. Alors $\mathcal{M}$ ne valide pas l'une des instances close de S, soit S_0, qui appartient également à $\mathcal{S}$ par hypothèse. Posons $S_0 =_{def} A_1, \dots, A_n \vdash B_1 \dots B_p$. On a donc pour $j \in \{1, \dots, n\}$, $\mathcal{M} \models A_j$ et pour $k \in \{1, \dots, p\}$, $\mathcal{M} \not\models B_k$. Or d'après le lemme 1.4.7, les formules A_j et B_k apparaissent toutes dans les Σ_i, donc d'après le lemme 1.4.8, les A_j à gauche et les B_k à droite. Soit donc un entier m tel que :
- $\forall j \in \{1, \dots, n\}\ \exists i \leq m$ A_j apparaît à gauche dans Σ_i
- $\forall k \in \{1, \dots, p\}\ \exists i \leq m B_k$ apparaît à droite dans Σ_i

On a construit les règles de façon qu'alors
- $\forall j \in \{1, \dots, n\}$ A_j apparaît à gauche dans Σ_m

– $\forall k \in \{1, \ldots, p\}$ B_k apparaît à droite dans Σ_m

Ceci signifie que le séquent Σ_m est un axiome propre à affaiblissement près, et alors la chaîne de déduction $(\Sigma_i)_{i\in\mathbb{N}}$ ne serait pas infinie, ce qui contredit l'hypothèse. $\mathcal{M}$ est donc bien modèle de S. ■

Il reste à prouver les deux lemmes.

Démonstration (lemme 1.4.7). Les deux premières assertions découlent directement de la façon dont est construit l'arbre de recherche. Pour la troisième, on le montre par récurrence sur n en utilisant la contraction internalisée aux règles $\forall_g$ et $\exists_d$ et la première assertion. ■

Démonstration (lemme 1.4.8). Le résultat se montre pour chaque formule F d'un Σ_i par induction sur la structure de F.

- Pour les formules atomiques, c'est la définition du modèles pour les formules apparaissant à gauche, la conséquence de ce qu'aucun des Σ_i n'est un axiome pour les formules apparaissant à droite.
- Si le connecteur principal de la formule est propositionnel, on utilise la première assertion du lemme 1.4.7, et on le vérifie pour chaque règle.
- Pour une formule $\forall xG$ à droite dans un Σ_i, d'après le lemme 1.4.7, cette formule est en position active dans un Σ_j, pour $j \geq i$, la règle qui correspond est le $\forall_g$, supposons que $F[x_m/x]$ soit la prémisse. Par hypothèse d'induction $\mathcal{M} \not\models F[\overline{x_m}/x]$, donc $\mathcal{M} \not\models \forall xF$. On peut remarquer que le rôle tenu par les x_m est celui des témoins de Henkin dans une preuve de complétude à la Henkin (voir [Cori-Lascar 88]).
- les formules $\exists xG$ à gauche dans un Σ_i se traitent comme au cas précédent.
- Pour les formules $\forall xG$ à droite et $\exists xG$ à gauche, on utilise la troisième assertion du lemme 1.4.7. ■

Corollaire 1.4.9 (élimination des coupures) *Si un séquent Σ est prouvable en calcul des séquents avec coupures sous hypothèse un ensemble $\mathcal{S}$ de séquents stable par substitution, alors il est prouvable en calcul des séquents avec coupures restreintes aux formules apparaissant dans $\mathcal{S}$. En particulier, si $\mathcal{S}$ est vide, Σ est prouvable sans coupures.*

Démonstration. La règle de coupure est sémantiquement correcte, donc sous les hypothèses de l'énoncé, Σ est conséquence sémantique de $\mathcal{S}$, d'où le résultat d'après le théorème de complétude.

Avant de donner un certain nombre de conséquences du théorème de complétude, en particulier pour la preuve automatique, on va donner une preuve combinatoire directe de ce dernier énoncé.

1.5 Elimination des coupures en calcul des séquents.

1.5.1 Introduction.

Pour prouver l'élimination des coupures on va montrer qu'il est possible de "remonter" celles-ci au niveau des axiomes, et la coupure s'élimine alors trivialement (dans le cas des axiomes logiques) :

$$\dfrac{\begin{array}{c}\vdots\\ \pi\\ \vdots\end{array}}{\dfrac{\Gamma \vdash \Delta, A \quad A \vdash A}{\Gamma \vdash \Delta, A}\ cut} \quad \rightsquigarrow \quad \begin{array}{c}\vdots\\ \pi\\ \vdots\\ \Gamma \vdash \Delta, A\end{array}$$

Remarquons qu'en présence de séquents axiomes non logiques, on aura une coupure sur une formule du séquent axiome, dont on sait d'après le théorème de complétude qu'elles ne sont en général pas éliminables.

Pour "remonter" les coupures on utilise deux principes :

- la commutation de règles pour les règles dont la formule principale n'est pas la formule de coupure :

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\\ \pi_{11}\\ \vdots\end{array}}{\Gamma_{11} \vdash \Delta_{11}, A} \quad \dfrac{\begin{array}{c}\vdots\\ \pi_{12}\\ \vdots\end{array}}{\Gamma_{12} \vdash \Delta_{12}}\ R1}{\Gamma_1 \vdash \Delta_1, A} \quad \dfrac{\begin{array}{c}\vdots\\ \pi_2\\ \vdots\end{array}}{A, \Gamma_2 \vdash \Delta_2}\ R2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\ cut$$

$$\Big\downarrow$$

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\\ \pi_{11}\\ \vdots\end{array}}{\Gamma_{11} \vdash \Delta_{11}, A} \quad \dfrac{\begin{array}{c}\vdots\\ \pi_2\\ \vdots\end{array}}{A, \Gamma_2 \vdash \Delta_2}\ R2}{\Gamma_{11}, \Gamma_2 \vdash \Delta_{11}, \Delta_2}\ cut \quad \dfrac{\begin{array}{c}\vdots\\ \pi_{12}\\ \vdots\end{array}}{\Gamma_{12} \vdash \Delta_{12}}}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\ R1$$

- quand les deux occurrences de la formule de coupure sont chacune formule principale d'une règle logique, celles-ci sont alors la règle gauche et la règle droite d'un même connecteur ou quantificateur ; on fait disparaître ces deux règles logique, et on fait porter la coupure sur les prémisses des règles. Par exemple pour l'implication cela donne :

$$\dfrac{\dfrac{\dfrac{\vdots\,\pi_1}{\Gamma_1, A \vdash B, \Delta_1}}{\Gamma_1 \vdash A \to B, \Delta_1}\to_d \quad \dfrac{\dfrac{\vdots\,\pi_{21}}{B, \Gamma_{21} \vdash \Delta_{21}} \quad \dfrac{\vdots\,\pi_{22}}{\Gamma_{22} \vdash \Delta_{22}, A}}{A \to B, \Gamma_2 \vdash \Delta_2}\to_g}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\,cut$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\vdots\,\pi_1}{\Gamma_1, A \vdash B, \Delta_1} \quad \dfrac{\vdots\,\pi_{21}}{B, \Gamma_{21} \vdash \Delta_{21}}}{\Gamma_1, \Gamma_{21}, A \vdash \Delta_1, \Delta_{21}}\,cut \quad \dfrac{\vdots\,\pi_{22}}{\Gamma_{22} \vdash \Delta_{22}, A}}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\,cut$$

On appellera *réduction logique* ce type de transformation.

On sait (voir § 1.3.3 page 35) que la commutation de la règle de coupure avec une règle logique pourrait être "bloquée" dans une autre situation que celle qui correspond à la réduction logique : si elle violait la condition sur le paramètre propre, dans le cas d'une règle $\forall_d$ ou $\exists_g$. En effet la commutation peut modifier le contexte d'application de la règle :

$$\dfrac{\dfrac{\dfrac{\vdots\,\pi_1}{\Gamma_1 \vdash \Delta_1, B[y/x], A}\forall_d \quad \dfrac{\vdots\,\pi_2}{A, \Gamma_2 \vdash \Delta_2}\,R2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2, \forall x B}\,cut \quad \rightsquigarrow \quad \dfrac{\dfrac{\vdots\,\pi_1}{\Gamma_1 \vdash \Delta_1, B[y/x], A} \quad \dfrac{\vdots\,\pi_2}{A, \Gamma_2 \vdash \Delta_2}\,R2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2, B[y/x]}\,cut}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2, \forall x B}\forall_d$$

On a alors besoin de la convention adoptée au § 1.3.2 page 33, à savoir que y n'apparaît jamais hors de la sous-déduction qui se termine par la règle dont il est le paramètre propre.

La commutation avec des règles structurelles.

La commutation de la coupure avec une règle structurelle n'est pas innocente. La commutation avec l'affaiblissement correspond à l'effacement d'une des deux sous-preuves prémisses :

$$\dfrac{\dfrac{\vdots\,\pi_1}{\Gamma_1 \vdash \Delta_1, A} \quad \dfrac{\dfrac{\vdots\,\pi_2}{\Gamma_2 \vdash \Delta_2}\,w}{A, \Gamma_2 \vdash \Delta_2}}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\,cut \quad \rightsquigarrow \quad \dfrac{\vdots\,\pi_2}{\Gamma_2 \vdash \Delta_2}\,w^*$$

La commutation avec la règle de contraction induit une duplication d'une des deux sous-preuves prémisses :

$$\dfrac{\dfrac{\vdots\,\pi_1}{\Gamma_1 \vdash \Delta_1, A} \quad \dfrac{\vdots\,\pi_2}{\dfrac{A, A, \Gamma_2 \vdash \Delta_2}{A, \Gamma_2 \vdash \Delta_2}\, c_d}}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\, cut \quad \rightsquigarrow \quad \dfrac{\dfrac{\vdots\,\pi_1}{\Gamma_1 \vdash \Delta_1, A} \quad \dfrac{\vdots\,\pi_1 \quad \vdots\,\pi_2}{\dfrac{\Gamma_1 \vdash \Delta_1, A \quad A, A, \Gamma_2 \vdash \Delta_2}{A, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\, cut}}{\dfrac{\Gamma_1, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_1, \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\, c^*}\, cut$$

La principale difficulté de la preuve vient de la commutation avec la contraction. En effet on peut observer que pour les autres réductions il est évident qu'une mesure simple décroit, la somme des hauteurs des sous-arbres prémisses de la coupure pour les autres commutations, la complexité de la formule de coupure pour les réductions logiques. Ce n'est pas le cas pour la commutation avec la contraction.

De plus en logique classique, il est possible d'avoir une coupure entre deux formules issues chacune d'une contraction, l'une à droite, l'autre à gauche. Dans ce cas la commutation n'est pas un procédé suffisant pour conclure. Supposons en effet que dans le schéma ci-dessus la preuve π_1 se termine par une contraction. Le résultat de la réduction sera alors :

$$\dfrac{\dfrac{\dfrac{\vdots\,\pi'_1}{\dfrac{\Gamma_1 \vdash \Delta_1, A, A}{\Gamma_1 \vdash \Delta_1, A}\, c_d} \quad \dfrac{\vdots\,\pi'_1}{\dfrac{\Gamma_1 \vdash \Delta_1, A, A}{\Gamma_1 \vdash \Delta_1, A}\, c_d} \quad \dfrac{\vdots\,\pi_2}{A, A, \Gamma_2 \vdash \Delta_2}}{A, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\, cut}{\dfrac{\Gamma_1, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_1, \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\, c^*}\, cut$$

Si on fait commuter la plus haute occurrence de la règle de coupure ci-dessus avec la contraction racine de π_1, on obtient (l'ordre des suites de contractions, qui commutent entre elles, n'a pas d'importance) :

$$\dfrac{\dfrac{\dfrac{\vdots\,\pi'_1}{\dfrac{\Gamma_1 \vdash \Delta_1, A, A}{\Gamma_1 \vdash \Delta_1, A}\, c_d} \quad \dfrac{\dfrac{\vdots\,\pi'_1}{\Gamma_1 \vdash \Delta_1, A, A} \quad \dfrac{\vdots\,\pi_2}{A, A, \Gamma_2 \vdash \Delta_2}\, cut}{A, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2, A} \quad \dfrac{\vdots\,\pi_2}{A, A, \Gamma_2 \vdash \Delta_2}\, cut}{\dfrac{A, A, \Gamma_1, \Gamma_2, \Gamma_2 \vdash \Delta_1, \Delta_2, \Delta_2}{A, A, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\, c^*}\, c_g}{\dfrac{\Gamma_1, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_1, \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\, c^*}\, cut$$

qui est une preuve plus complexe, en quelque sens que ce soit, que la preuve originale à savoir :

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \pi'_1\\ \vdots\\ \Gamma_1 \vdash \Delta_1, A, A\end{array}}{\Gamma_1 \vdash \Delta_1, A}\, c_d \qquad \dfrac{\begin{array}{c}\vdots\\ \pi_2\\ \vdots\\ A, A, \Gamma_2 \vdash \Delta_2\end{array}}{A, \Gamma_2 \vdash \Delta_2}\, c_d}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\, cut$$

Avant de donner une solution à ce problème, nous allons énumérer quelques réductions logiques.

1.5.2 Les réductions logiques.

Nous avons déjà vu la réduction dans le cas de l'implication au paragraphe ci-dessus. Le choix a été fait de prendre la formulation montante pour la règle droite et descendante pour la règle gauche. Un choix homogène montant ou descendant est possible, mais nécessite alors l'usage des règles structurelles pour la réduction. Ce choix correspond à appelle la formulation multiplicative des règles du calcul des séquents en logique linéaire (voir [Girard-Lafont-Taylor 89]). Le choix symétrique (montant pour la règle gauche, descendant pour la règle droite) est la formulation additive, et la réduction ne nécessite pas non plus de règles structurelles (exercice). Ces formulations prennent un sens en logique linéaire où les règles structurelles sont gérées par des connecteurs particuliers.

En fait pour les connecteurs binaires, la réduction logique ne dépend pas du connecteur mais du choix des règles gauche et droite pour ces connecteurs. En logique classique, il est toujours possible d'adopter le choix effectué ci-dessus pour l'implication, par exemple pour la conjonction :

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \pi_{11}\\ \vdots\\ \Gamma_{11} \vdash \Delta_{11}, A\end{array} \quad \begin{array}{c}\vdots\\ \pi_{12}\\ \vdots\\ \Gamma_{12} \vdash \Delta_{12}, B\end{array}}{\Gamma_2 \vdash \Delta_2, A \wedge B}\, \wedge_d \qquad \dfrac{\begin{array}{c}\vdots\\ \pi_2\\ \vdots\\ A, B, \Gamma_1 \vdash \Delta_1\end{array}}{A \wedge B, \Gamma_1 \vdash \Delta_1}\, \wedge_g}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\, cut$$

$$\Downarrow$$

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \pi_{11}\\ \vdots\\ \Gamma_{11} \vdash \Delta_{11}, A\end{array} \quad \begin{array}{c}\vdots\\ \pi_2\\ \vdots\\ A, B, \Gamma_1 \vdash \Delta_1\end{array}}{B, \Gamma_1, \Gamma_{11} \vdash \Delta_1, \Delta_{11}}\, cut \qquad \begin{array}{c}\vdots\\ \pi_{12}\\ \vdots\\ \Gamma_{12} \vdash \Delta_{12}, B\end{array}}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\, cut$$

Remarquons que les choix de règles pour l'implication et la conjonction faits ci-dessus conviennent à la logique intuitionniste. Il suffit de particulariser au cas où il y a une seule formule à droite. On ne peut faire le même choix pour la disjonction. Si on fait le choix suivant, qui est assez naturel, on a besoin des règles structurelles :

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \pi_1\\ \vdots\end{array}}{\dfrac{\Gamma_1 \vdash A}{\Gamma_1 \vdash A \vee B}\ \vee_{d_g}} \qquad \dfrac{\begin{array}{c}\vdots\\ \pi_{21}\\ \vdots\end{array} \quad \begin{array}{c}\vdots\\ \pi_{22}\\ \vdots\end{array}}{\dfrac{A, \Gamma_{21} \vdash \Delta_{21} \quad B, \Gamma_{22} \vdash \Delta_{22}}{A \vee B, \Gamma_2 \vdash \Delta_2}\ \vee_g}\ cut}{\Gamma_1, \Gamma_2 \vdash \Delta_2}$$

$$\Big\downarrow$$

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\\ \pi_1\\ \vdots\end{array} \quad \begin{array}{c}\vdots\\ \pi_{21}\\ \vdots\end{array}}{\Gamma_1 \vdash A \quad A, \Gamma_{21} \vdash \Delta_{21}}\ cut}{\dfrac{\Gamma_1, \Gamma_{21} \vdash \Delta_{21}}{\Gamma_1, \Gamma_2 \vdash \Delta_2}\ w^*}$$

($\Delta_{21} \cup \Delta_{22} = \Delta_2$, dans le cas de la logique intuitionniste, Δ_2 contient au plus une formule).

La négation se traite encore plus simplement (exercice).

Pour les quantificateurs, comme en déduction naturelle, on a besoin de modifier une preuve par substitution d'un terme à une variable :

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \pi_1\\ \vdots\end{array}}{\dfrac{\Gamma_1 \vdash \Delta_1, A[y/x]}{\Gamma_1 \vdash \Delta_1, \forall x A}\ \forall_d} \qquad \dfrac{\begin{array}{c}\vdots\\ \pi_2\\ \vdots\end{array}}{\dfrac{A[t/x], \Gamma_2 \vdash \Delta_2}{\forall x A, \Gamma_2 \vdash \Delta_2}\ \forall_g}\ cut}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2} \qquad \rightsquigarrow \qquad \dfrac{\dfrac{\begin{array}{c}\vdots\\ \pi_1[t/y]\\ \vdots\end{array}}{\Gamma_1 \vdash \Delta_1, A[t/x]} \quad \dfrac{\begin{array}{c}\vdots\\ \pi_2\\ \vdots\end{array}}{A[t/x], \Gamma_2 \vdash \Delta_2}\ cut}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}$$

La correction de cette transformation est assurée par la condition sur le paramètre propre y, qui assure que $\Gamma_1[t/y] = \Gamma_1$, $\Delta_1[t/y] = \Delta_1$, et $A[y/x][t/y] = A[t/x]$.

La réduction dans le cas du connecteur existentiel est identique.

1.5.3 Le calcul des séquents intuitionniste.

Remarquons que nous pouvons déduire l'élimination des coupures du calcul des séquents intuitionniste dans le fragment $\rightarrow, \wedge, \forall$ en composant la traduction ψ du § 1.3.6 page 43, la normalisation de la preuve de déduction naturelle obtenue, et la traduction φ du § 1.3.4 page 37 qui transforme une preuve normale en une preuve sans coupures.

On va cependant donner une preuve directe. Pour gérer le problème posé par les contractions, on peut utiliser une généralisation de la règle de coupures, la multi-coupure :

$$\dfrac{\Gamma_1 \vdash A \quad \overbrace{A, \ldots, A}^{n}, \Gamma_2 \vdash \Delta}{\Gamma_1, \Gamma_2 \vdash \Delta}\ cut\times \qquad \begin{array}{l} n \geq 1 \\ \Delta \text{ contient au plus une formule} \end{array}$$

Cette règle peut s'obtenir comme n applications successives de la règle de coupure ordinaire, ou encore comme $n-1$ contractions suivie d'une règle de coupure.

On va maintenant démontrer le lemme suivant :

Lemme 1.5.1 *S'il existe une preuve sans coupures π_1 de $\Gamma_1 \vdash A$, une preuve sans coupures π_2 de $A, \ldots, A, \Gamma_2 \vdash \Delta$ (où Δ contient au plus une formule), alors il existe une preuve sans coupure de $\Gamma_1, \Gamma_2 \vdash \Delta$.*

Démonstration. On va montrer ce lemme par induction d'abord sur la complexité de A, puis sur $h(\pi_1) + h(\pi_2)$, où $h(\pi)$ désigne la hauteur de la preuve π *sans compter les règles structurelles*. On distingue à chaque fois suivant les dernières règles terminant π_1 et π_2. On utilise les réductions vues ci-dessus, ou des composéesde celles-ci.

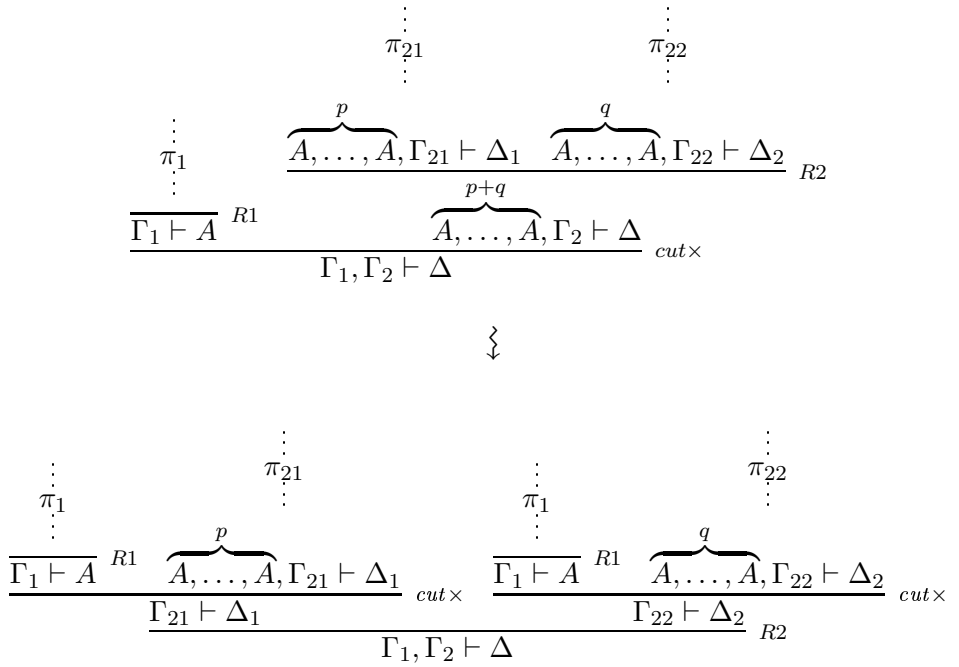
Si l'une de celles-ci est affaiblissement, le résultat suit par hypothèse de récurrence sur $h(\pi_1) + h(\pi_2)$ (voir ci-dessus).

Si l'une de celles-ci est une contraction La permutation de la règle de multi-coupure avec la contraction s'écrit maintenant (Δ contient au plus une formule) :

$$\dfrac{\Gamma_1 \vdash A \quad \overbrace{A, \ldots, A}^{n}, \Gamma_2 \vdash \Delta}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\, cut\times \quad \rightsquigarrow \quad \dfrac{\Gamma_1 \vdash A \quad \overbrace{A, \ldots, A, A}^{n+1}, \Gamma_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\, cut\times$$

et on obtient donc le résultat par hypothèse de récurrence sur $h(\pi_1) + h(\pi_2)$.

La commutation avec une règle logique qui n'a pas pour formule principale une formule de coupure est maintenant disymétrique, suivant la prémisse de la règle de coupure sur laquelle elle porte. Dans le cas où la règle logique est au dessus de la prémisse où la formule de coupure apparaît à droite, c'est identique à ce qui a été montré en introduction. Dans l'autre cas c'est très similaire :



(Δ, Δ_1 et Δ_2 contiennent au plus une formule).

et à chaque fois on a le résultat par hypothèse de récurrence sur $h(\pi_1)+h(\pi_2)$.

Pour la réduction logique, les schémas vu en introduction et au § 1.5.2 page 58 n'éliminent qu'une occurrence de la formule introduite. Ceci convient si la multi-coupure est en fait une coupure, et le résultat suit par hypothèse d'induction sur la complexité de la formule de coupure.

On a donc besoin d'ajouter une commutation de la coupure originale. Par exemple pour l'implication (on désigne par $[F]^n$ une suite de n occurrences de la formule F) :

$$\dfrac{\dfrac{\dfrac{\vdots\,\pi_1}{\Gamma_1, B \vdash C}}{\Gamma_1 \vdash B \to C}\,{\to_d} \quad \dfrac{\vdots\,\pi_{21} \qquad \vdots\,\pi_{22}}{\dfrac{C, [B \to C]^p, \Gamma_{21} \vdash \Delta \quad \Gamma_{22}, [B \to C]^q \vdash B}{[B \to C]^{n+1}, \Gamma_2 \vdash \Delta}\,{\to_g}}\,{cut\times}}{\Gamma_1, \Gamma_2 \vdash \Delta}$$

$$\Big\downarrow$$

$$\dfrac{\dfrac{\dfrac{\vdots\,\pi_1}{\Gamma_1, B \vdash C} \quad \dfrac{\dfrac{\vdots\,\pi_1}{\Gamma_1, B \vdash C}\,{\to_d} \qquad \vdots\,\pi_{21}}{\dfrac{\Gamma_1 \vdash B \to C \qquad C, [B \to C]^p, \Gamma_{21} \vdash \Delta}{C, \Gamma_1, \Gamma_{21} \vdash \Delta}\,{cut}}\,{cut\times}}{\Gamma_1, \Gamma_1, \Gamma_{21}, B \vdash \Delta} \quad \dfrac{\dfrac{\vdots\,\pi_1}{\Gamma_1, B \vdash C}\,{\to_d} \qquad \vdots\,\pi_{22}}{\dfrac{\Gamma_1 \vdash B \to C \qquad \Gamma_{22}, [B \to C]^q \vdash B}{\Gamma_1, \Gamma_{22} \vdash B}\,{cut}}\,{cut\times}}{\dfrac{\Gamma_1, \Gamma_1, \Gamma_1, \Gamma_2 \vdash \Delta}{\Gamma_1, \Gamma_2 \vdash \Delta}\,{c^*}}$$

(Δ contient au plus une formule, $n = p + q$)

Dans ce cas le résultat du lemme suit par hypothèse de récurrence sur le somme des hauteurs des preuves prémisses pour les deux multi-coupures les plus hautes, puis par hypothèse d'induction sur la complexité de la formule de coupure pour les deux dernières coupures.

Les autres réductions logiques se font de façon similaire. ■

On peut maintenant démontrer l'élimination des coupures :

Théorème 1.5.2 *Si un séquent est démontrable en calcul des séquents intuitionniste, il est démontrable en calcul des séquents intuitionniste sans la règle de coupure.*

Démonstration. On démontre le résultat par récurrence sur le nombre de coupures dans la preuve π d'un séquent $\Gamma \vdash \Delta$. Si π contient au moins une coupure, on choisit parmi celles-ci l'une des plus hautes dans π. On peut appliquer le lemme précédent aux deux sous-preuves prémissses : on obtient ainsi une preuve π' de $\Gamma \vdash \Delta$ qui contient une coupure de moins que π d'où le résultat par récurrence. ■

Si on compare la preuve d'élimination des coupures précédente à la preuve de normalisation faible de la déduction naturelle (voir théorème **??** page ??), on constate que cette dernière preuve utilise des transformations *locales* à la preuve : on obtient la preuve sans coupures par permutations successives de la règle de multi-coupure avec les autres règles. La preuve de normalisation de la déduction naturelle utilise des transformations *globales* de la preuve, en ce qui concerne par exemple l'implication : il s'agit de remplacer certaines occurrences de l'axiome $A \vdash A$ dans une preuve de $\Gamma_1 \vdash A \to B$, par une preuve de $\Gamma_2 \vdash A$. Cela n'a cependant rien d'intrinsèque à chaque fois, même si c'est probablement un peu plus facile de le présenter ainsi. On pourrait en particulier donner des transformations globales des preuves de calcul des séquents pour l'élimination des coupures.

Une différence importante est qu'une stratégie de réduction de la preuve comme celle qui est adoptée dans la preuve d'élimination des coupures ci-dessus est indispensable, même dans le cas intuitionniste. Par contre en déduction naturelle toute composition des réductions élémentaires aboutit (théorème de normalisation forte **??** page ??), et aboutit à la même preuve (propriété de Church-Rosser **??** page ??).

1.5.4 Une gestion plus stricte des contractions.

La preuve d'élimination des coupures du § précédent utilise des transformations locales, et gère la contraction en l'intégrant à la règle de coupure (multi-coupure). Une autre solution, qui permet de conserver la règle de coupure usuelle, est de gérer plus strictement la contraction en l'intégrant aux règles logiques, mais pas systématiquement comme dans la version montante du calcul des séquents. Pour cela on revient à la définition des séquents comme couple de multi-ensembles. On rappelle que les multi-ensembles peuvent être formalisés comme des applications d'un ensemble fini d'index dans l'ensemble des formules. On peut représenter cette application comme un ensemble de couples que l'on notera F^x, où x est un index et F une formule. Par définition, deux formules différentes ne peuvent avoir le même index. On choisit comme règles du calcul des séquents les règles de la version descendante suivie de la contraction systématique des formules de même index.

Par exemple la règle $\rightarrow_g$ aura, entre autres, les instances suivantes :

$$\frac{\Gamma, (A \rightarrow B)^x, B^y \vdash \Delta \quad \Gamma', (A \rightarrow B)^x \vdash A^\alpha, \Delta'}{\Gamma, \Gamma', (A \rightarrow B)^x \vdash \Delta, \Delta'} \rightarrow_g$$

$$\frac{\Gamma, (A \rightarrow B)^x, B^y \vdash \Delta \quad \Gamma', (A \rightarrow B)^{x'} \vdash A^\alpha, \Delta'}{\Gamma, \Gamma', (A \rightarrow B)^x, (A \rightarrow B)^{x'}, (A \rightarrow B)^z \vdash \Delta, \Delta'} \rightarrow_g$$

à chaque fois la "," à gauche ou à droite signifie l'union ensembliste sur les formules indexées.

La contraction apparaît dans un ce système de deux façons : internalisée dans le contexte d'une règle binaire, ou sur la formule principale d'une règle logique.

Par ailleurs cette façon de gérer la contraction a un sens du point de vue de l'interprétation des preuves comme programmes. Par exemple les index des formules à gauche correspondent à des variables du λ-calcul.

On va tout d'abord montrer qu'il est toujours possible de transcrire une preuve du calcul des séquents en une preuve avec contractions implicites.

Lemme 1.5.3 *S'il existe une preuve de $\Gamma, A^x, A^y \vdash \Delta$, resp. de $\Gamma \vdash B^\alpha, B^\beta$ en calcul des séquents avec contractions implicite, alors il existe une preuve dans ce calcul ayant la même structure du séquent $\Gamma, A^x \vdash \Delta$, resp. $\Gamma \vdash B^\alpha, \Delta$.*

Démonstration. Par induction sur la hauteur de la preuve, en regardant la dernière règle utilisée. Cela se vérifie règle par règle sans difficultés. ■

Lemme 1.5.4 *S'il existe une preuve en calcul des séquents usuel (par exemple version descendante) du séquent $\Gamma \vdash \Delta$, où Γ et Δ sont des multi-ensembles notés comme des ensembles de formules indexées, alors il existe une preuve du même séquent en calcul des séquents avec contraction implicite, et cette preuve a la même structure aux règles de contractions près.*

Démonstration. Par récurrence sur la hauteur de la preuve, en regardant la dernière règle utilisée. Si celle-ci est une contraction, on utilise le lemme précédent. ■

Remarquons que la transformation sur la preuve induite par ce dernier lemme consiste à faire permuter les contractions jusqu'à ce que, soit elles disparaissent car partagées dans les contextes d'une règle binaire, soit l'une des formules contractée soit crée par une règle logique.

On peut donner maintenant une variante de la démonstration précédente d'élimination des coupures pour le calcul des séquents intuitionniste. Il suffit de réénoncer le lemme 1.5.1 page 60 pour la règle de coupure (et non de multi-coupure) dans le calcul des séquents avec contraction implicite.

Lemme 1.5.5 *Dans le calcul des séquents avec contraction implicite, s'il existe une preuve sans coupures π_1 de $\Gamma_1 \vdash A$, une preuve sans coupures π_2 de $A, \Gamma_2 \vdash \Delta$ (où Δ contient au plus une formule), alors il existe une preuve sans coupure de $\Gamma_1, \Gamma_2 \vdash \Delta$.*

Démonstration. La démonstration est quasi-identique à celle du lemme 1.5.1 page 60. La seule règle structurelle est désormais l'affaiblissement. Quand les dernières règles de π_1 et π_2 sont des règles logiques sur A, on gère la contraction éventuellement internalisée exactement de la même façon, et donc la démonstration se fait encore par induction sur la complexité de la formule de coupures, puis sur la somme des hauteurs des preuves π_1 et π_2 sans compter les affaiblissements. ■

1.5.5 Le calcul des séquents classique.

La preuve pour le calcul des séquents classique se fait de façon similaire. L'une ou l'autre des méthodes utilisée sont envisageables. On va utiliser la seconde. Pour gérer le cas d'une coupure sur deux formules contractées, qui ne pouvait se produire en calcul des séquents intuitionniste, il faut répéter le procédé utilisé (commutation + réduction logique) sur chacune des prémisses.

Lemme 1.5.6 *Dans le calcul des séquents avec contraction implicite, s'il existe une preuve sans coupures π_1 de $\Gamma_1 \vdash A, \Delta_1$, une preuve sans coupures π_2 de $A, \Gamma_2 \vdash \Delta_2$, alors il existe une preuve sans coupure de $\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2$.*

Démonstration. Par induction sur la complexité de A, la formule de coupure, puis par récurrence sur la somme des hauteurs des preuves π_1 et π_2 comptées sans règles structurelles (c.a.d. pour ce calcul des séquents sans affaiblissements). On regarde les dernières règles de π_1 et π_2. Dans le cas où l'une d'entre elle n'est pas une règle logique de formule principale A, on commute la coupure et cette règle et le résultat suit par hypothèse de récurrence sur $h(\pi_1) + h(\pi_2)$ (c'est identique à ce qui est fait dans la preuve du lemme 1.5.1 page 60).

Dans le cas où la dernière règle de π_1 comme la dernière règle de π_2 sont de formule principale A, on doit détailler symbole par symbole. On va traiter l'implication, dans le "pire des cas" où la règle $\rightarrow_d$ internalise une contraction, et la règle $\rightarrow_g$ deux contractions. La preuve

se présente ainsi (pour simplifier, on ne note pas les indices des prémisses secondaires) :

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \pi_1\\ \vdots\end{array}}{\Gamma_1, B \vdash C, (B \to C)^\alpha, \Delta_1}\ {\scriptstyle \to_d}\ \dfrac{\Gamma_1 \vdash (B \to C)^\alpha, \Delta_1}{\Gamma_1 \vdash (B \to C)^\alpha, \Delta_1} \qquad \dfrac{\dfrac{\begin{array}{c}\vdots\\ \pi_{21}\\ \vdots\end{array}}{C, (B \to C)^x, \Gamma_{21} \vdash \Delta_{21}} \quad \dfrac{\begin{array}{c}\vdots\\ \pi_{22}\\ \vdots\end{array}}{\Gamma_{22}, (B \to C)^x \vdash B, \Delta_{22}}\ {\scriptstyle \to_g}}{(B \to C)^x, \Gamma_2 \vdash \Delta_2}\ {\scriptstyle cut}}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}$$

$(\Gamma_2 = \Gamma_{21} \cup \Gamma_{22},\ \Delta_2 = \Delta_{21} \cup \Delta_{22})$

on utilise la méthode des *coupures croisées*, c'est à dire que l'on commence par une commutation de la coupure sur la prémisse où la formule de coupure apparaît à gauche (comme dans le cas intuitionniste) :

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \pi_1\\ \vdots\end{array}}{\Gamma_1, B \vdash C, (B \to C)^\alpha, \Delta_1}\ {\scriptstyle \to_d}\ \dfrac{\Gamma_1 \vdash (B \to C)^\alpha, \Delta_1}{\Gamma_1 \vdash (B \to C)^\alpha, \Delta_1} \qquad \dfrac{\begin{array}{c}\vdots\\ \pi_{21}\\ \vdots\end{array}}{C, (B \to C)^x, \Gamma_{21} \vdash \Delta_{21}}}{C, \Gamma_1, \Gamma_{21} \vdash \Delta_1, \Delta_{21}}\ {\scriptstyle cut}$$

et

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \pi_1\\ \vdots\end{array}}{\Gamma_1, B \vdash C, (B \to C)^\alpha, \Delta_1}\ {\scriptstyle \to_d}\ \dfrac{\Gamma_1 \vdash (B \to C)^\alpha, \Delta_1}{\Gamma_1 \vdash (B \to C)^\alpha, \Delta_1} \qquad \dfrac{\begin{array}{c}\vdots\\ \pi_{22}\\ \vdots\end{array}}{(B \to C)^x, \Gamma_{22} \vdash B, \Delta_{22}}}{\Gamma_1, \Gamma_{22} \vdash B, \Delta_1, \Delta_{22}}\ {\scriptstyle cut}$$

soient π'_1 et π'_2 ces deux preuves

puis sur la prémisse où la formule apparaît à droite (ce qui ne se pouvait se produire dans le cas intuitionniste) :

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \pi_1\\ \vdots\end{array}}{\Gamma_1, B \vdash C, (B \to C)^\alpha, \Delta_1} \qquad \dfrac{\dfrac{\begin{array}{c}\vdots\\ \pi_{21}\\ \vdots\end{array}}{C, (B \to C)^x, \Gamma_{21} \vdash \Delta_{21}} \quad \dfrac{\begin{array}{c}\vdots\\ \pi_{22}\\ \vdots\end{array}}{\Gamma_{22}, (B \to C)^x \vdash B, \Delta_{22}}\ {\scriptstyle \to_g}}{(B \to C)^x, \Gamma_2 \vdash \Delta_2}\ {\scriptstyle cut}}{\Gamma_1, \Gamma_2, B \vdash C, \Delta_1, \Delta_2}$$

soit π'' cette preuve

et on compose par des coupures sur les sous-formules immédiates de la formule de coupure :

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \pi''\\ \vdots\end{array} \qquad \begin{array}{c}\vdots\\ \pi'_1\\ \vdots\end{array}}{\Gamma_1,\Gamma_2,B\vdash C,\Delta_1,\Delta_2 \quad C,\Gamma_1,\Gamma_{21}\vdash\Delta_1,\Delta_{21}}\ cut \qquad \begin{array}{c}\vdots\\ \pi'_2\\ \vdots\end{array}}{\Gamma_1,\Gamma_2,B\vdash\Delta_1,\Delta_2 \qquad \Gamma_1,\Gamma_{22}\vdash B,\Delta_1,\Delta_{22}}\ cut$$

Les preuves π'_1, π'_2 et π'' peuvent être remplacées par ds preuves sans coupures par hypothèse de récurrence sur la somme des hauteurs des preuves prémisses. La dernière preuve est obtenue par coupure sur des formules de complexité inférieure à $A = B \to C$, donc se transforme en une preuve sans coupures par induction sur la complexité de A.

Les autres connecteurs binaires se traitent de façon identiques, la négation et les quentificateurs se traitent de façon analogue (un peu plus simplement car les règles sont unaires). ■

Théorème 1.5.7 (Élimination des coupures.) *Si un séquent $\Gamma \vdash \Delta$ a une preuve en calcul des séquents classique, alors il a une preuve sans coupures en calcul des séquents, ce quelquesoit la variante du calcul des séquents parmi celles introduites.*

Démonstration. On le prouve tout d'abord pour le calcul des séquents avec contraction implicite par récurrence sur le nombre de coupures dans la preuve. Il suffit d'éliminer une coupure de hauteur maximale par le lemme précédent, puis on conclut par hypothèse de récurrence. Pour étendre le résultat au calcul des séquents version descendante, on applique le lemme 1.5.4 page 63. Les autres versions du calcul des séquents se déduisent de celle-ci par règles structurelles. ■

On peut étendre ce résultat en présence d'axiomes propres, en reprenant la même démonstration. On obtient ainsi directement le corollaire 1.4.9 page 54 du théorème de complétude.

1.6 Premières applications de l'existence d'une preuve sans coupures.

1.6.1 Applications en logique intuitionniste.

Propriété de disjonction et propriété d'existence.

$$\vdash_i A \vee B \text{ ssi } \vdash_i A \vee \vdash_i B$$

$$\vdash \exists x A \text{ ssi il existe un terme } t \text{ tel que } \vdash_i A[t/x]$$

[*à compléter*]

Décidabilité du calcul des séquents intuitionniste.

Décidabilité par recherche d'une preuve sans coupures ni "redondances".
[*à compléter*]

1.6.2 Applications en logique classique.

La complétude de la méthode de coupures (résolution).

commutation des coupures entre elles, complétude de la stratégie linéaire par entrées.
[*à compléter*]

Le théorème du séquent médian.

Application au théorème de Herbrand.
[*à compléter*]

Chapitre 2

Applications à la preuve automatique.

2.1 Le calcul propositionnel.

Le calcul propositionnel classique est décidable. Les méthodes de décision que nous allons étudier ne sont pas spécialement efficaces en calcul propositionnel, mais se généralisent de façon utile au calcul des prédicats.

2.1.1 Méthode des tableaux.

En calcul propositionnel, la méthode des tableaux est une variante syntaxique du calcul des séquents version montante. On peut la voir comme une version un peu optimisée en espace et donc plus adéquate pour l'implémentation. Essentiellement, la différence est que l'on ne recopie pas les formules. Le séquent en un point de l'arbre est donc l'ensemble des formules sur la section de branche de la racine de la preuve en ce point.

[à compléter]

2.1.2 Méthode des connexions.

La méthode des connexions est encore une version optimisée en espace du calcul des séquents version montante, avec axiomes sur des formules atomiques. On "partage" des formules entre des séquents, et pas forcément entre des séquents qui sont sur une même branche (comme en méthode des tableaux), et on quotiente par les commutations de règles.

Comme en méthode des tableaux, on attribue un signe aux formules, le signe "$+$" correspond aux formules à droite dans un séquent, le signe "$-$" au formules à gauche. On appelle *littéral* une formule atomique signée.

On définit d'abord la *matrice* d'une formule signée F quelconque.

Définition 2.1.1 Les matrices de F^+, et F^-, soient $M(F^+)$ et $M(F^-)$, sont définies par induction sur F :

i. Pour $F = \alpha$ atomique,

$$M(\alpha^+) = \alpha^+ \quad M(\alpha^-) = \alpha^-;$$

ii. Pour $F = \neg A$,

$$M((\neg A)^+) = M(A^-) \quad M((\neg A)^-) = M(A^+)$$

iii. pour $F = A \wedge B$,

$$M((A \wedge B)^+) = \begin{matrix} M(A^+) \\ M(B^+) \end{matrix} \qquad M((A \wedge B)^-) = \; M(A^-)\; M(B^-)$$

iv. pour $F = A \vee B$,

$$M((A \vee B)^+) = \; M(A^+)\; M(B^+) \qquad M((A \vee B)^-) = \begin{matrix} M(A^-) \\ M(B^-) \end{matrix}$$

v. pour $F = A \vee B$,

$$M((A \to B)^+) = \; M(A^-)\; M(B^+) \qquad M((A \to B)^-) = \begin{matrix} M(A^+) \\ M(B^-) \end{matrix}$$

Les deux opérations de juxtaposition, verticalement ou horizontalement, sont considérées à associativité près.

La matrice d'une formule F est la matrice de F^+. La matrice d'un séquent $A_1, \ldots, A_n \vdash B_1, \ldots, B_p$ est :

$$M(A_1^-) \ldots M(A_n^-) M(B_1^+) \ldots M(B_p^+) \; .$$

L'opération de juxtaposition horizontale correspond à la "," du calcul des séquent, elle est utilisée pour ce qui correspond aux règles unaires du calcul des séquents. L'opération de juxtaposition verticale correspond aux règles binaires du calcul des séquents. On s'aperçoit alors que l'écriture de la matrice d'une formule correspond à l'écriture d'un arbre de calcul des séquents, l'arborescence correspondant à des "emboîtements" successifs.

On peut aussi interpréter la matrice d'une formule F comme une formule équivalente utilisant uniquement la disjonction (juxtaposition horizontale) et la conjonction (juxtapostion verticale) sur les littéraux. Ce n'est pas en général une forme normale : la matrice respecte la structure des connecteurs binaires de la formule F.

Exemple. Matrice de $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$:

$$\begin{pmatrix} A^+ \\ B^+ \\ C^- \end{pmatrix} \; \begin{pmatrix} A^+ \\ B^- \end{pmatrix} \; A^- \; C^+$$

Matrice de $(A \to B) \to A) \to A$:

$$\begin{pmatrix} A^- & B^+ \\ A^- & \end{pmatrix} \; A^+$$

On définit maintenant la notion qui correspond aux feuilles d'un arbre de recherche de preuves en calcul des séquents :

Définition 2.1.2 Un chemin est une suite finie d'atomes signés. On définit l'ensemble des *chemins* qui traversent une matrice, par induction sur la structure de la matrice :

i. pour un atome α^*, $* \in \{+, -\}$:

$$\mathcal{C}(\alpha^*) = \alpha^* \; ;$$

ii.

$$\mathcal{C}(A_1 \dots A_n) = \mathcal{C}(A_1) \times \cdots \times \mathcal{C}(A_n) \ ;$$

iii.

$$\mathcal{C}\begin{pmatrix} A_1 \\ \vdots \\ A_n \end{pmatrix} = \mathcal{C}(A_1) \cup \cdots \cup \mathcal{C}(A_n) \ ;$$

On peut aussi interpréter l'ensemble des chemins d'une matrice comme la forme normale conjonctive/disjonctive de la formule correspondante.

Exemple. La matrice de $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$ donnée ci-dessus est traversée par 6 chemins :

$$\begin{array}{lll} A^+, A^+, A^-, C^+ & A^+, B^-, A^-, C^+ & B^+, A^+, A^-, C^+ \\ B^+, B^-, A^-, C^+ & C^-, A^+, A^-, C^+ & C^-, B^-, A^-, C^+ \end{array}$$

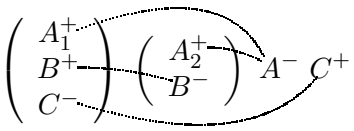
La matrice de $(A \to B) \to A) \to A$ donnée ci-dessus est traversée par deux chemins :

$$A^-, B^+, A^+ \quad \text{et} \quad A^-, A^+$$

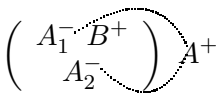
Enfin on définit la notion qui correspond aux séquents axiomes :

Définition 2.1.3 On appelle *connexion* d'une matrice un couple d'occurrences dans cette matrice du même atome sur un chemin l'une positive, l'autre négative. Un *ensemble complet de connexions* d'une matrice M est un ensemble de connexions tel que l'une d'entre elles apparaisse sur tout chemin traversant M.

Exemple. La matrice de $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$ a un ensemble complet de 5 connexions qui est $\{(A_1^+, A^-), (A_2^+, A^-), (B^+, B^-), (C^-, C^+)\}$ (on note A_1 et A_2 les deux occurrences positives de A) comme le montre le schéma :



La matrice de $(A \to B) \to A) \to A$ a un ensemble complet de 2 connexions $\{(A_1^-, A^+), (A_2^-, A^+)\}$ comme le montre le schéma :



Que l'on interprète la matrice comme une formule ou comme un arbre de calcul des séquents, on a immédiatement :

Proposition 2.1.4 *Une formule, ou un séquent est démontrable si et seulement si sa matrice possède un ensemble complet de connexions.*

L'essentiel est en fait que cette écriture suggère d'autres façons de rechercher la preuve qu'en calcul des séquents ou méthode des tableaux, par exemple en commençant par les connexions (c'est à dire les axiomes). La méthode qui consisterait à calculer l'ensemble de chemins puis à vérifier que chacun d'eux contient une connexion serait une version pauvre de la méthode des tableaux.

2.1.3 La résolution.

En calcul propositionnel, la résolution se réduit à utiliser la coupure. La règle de coupure est complète pour les séquents constitués d'atomes. [à compléter]

2.1.4 Méthode inverse et forme normale structurelle.

[à compléter]

2.2 Unification.

Le calcul des prédicats est indécidable. On peut cependant s'interesser à des algorithmes de semi-décision, c'est à dire des méthodes qui étant donné une formule termineront (théoriquement) si celle-ci est prouvable. Si l'on inspecte la preuve du théorème de complétude, en particulier la construction de l'arbre de recherche de preuve du § 1.4.3 page 49, et que l'on se restreint au cas sans axiomes non logiques (complétude faible) on observe que la non terminaison de l'algorithme vient des règles $\forall_g$ et $\exists_d$. On peut distinguer deux problèmes : la contraction, qui la vraie source d'indécidabilité et qui n'est pas réductible, et l'énumération de tous les termes du langage pour capturer toutes les possibilités d'application de ces règles. L'essai de tous les termes du langage est impraticable même sur ordinateur. On va remplacer ceci par un autre mécanisme : il s'agit de retarder les choix des termes pour les règles $\forall_g$ et $\exists_d$ par l'introduction de meta-variables. On fait le choix en recherchant à mettre en évidence un axiome logique : supposons que les termes t_i et t'_i contiennent des meta-variables,

$$\Gamma, A[t_1/x_1, \ldots, t_n/x_n] \vdash A[t'_1/x_1, \ldots, t'_n/x_n], \Delta$$

est un axiome pour une substitution σ sur les meta-variables à valeurs dans les termes du langage telle que :

$$\sigma(t_1) = \sigma(t'_1), \ldots, \sigma(t_n) = \sigma(t'_n) \ .$$

La recherche d'une telle substition s'appelle *l'unification*, elle est décidable. Il s'agira donc de construire un arbre de recherche de preuves (potentiellement infini à cause de la contraction) en repérant de tels séquents, on peut se restreindre aux formules A atomiques, et en utilisant l'unification. Cette méthode sera précisée aux § 2.3 page 84 et 2.4 page 89.

2.2.1 Premières définitions.

Soit un langage L du premier ordre contenant des variables et des symboles de fonction (dont des constantes) [1] *unifier* un ensemble fini S de couples de termes

$$S = \{(t_1, t'_1), \ldots, (t_n, t'_n)\}$$

c'est trouver une substitution s à support fini (c.a.d. l'identité sauf pour un nombre fini de variables), définie des variables dans les termes, telle que

1. Il s'agit de l'unification du premier ordre : il n'y a pas de variables de fonctions. L'unification d'ordre supérieur peut se définir en λ-calcul typé et autorise la substitution sur des variables de fonctions. Elle n'est pas décidable en général.

$$s(t_1) = s(t'_1), \ldots, s(t_n) = s(t'_n).$$

On appelle *support* d'une telle substitution σ et on le note $support(\sigma)$ l'ensemble fini des variables x telles que $\sigma(x) \neq x$.

Dans ce paragraphe un système désigne un ensemble fini de couples de termes de L.

Un *unificateur principal* de S est une substitution à support fini σ vérifiant que si S est unifiable et que s est un unificateur de S, alors il existe une substitution α telle que $s = \alpha \circ \sigma$.

Remarquons que Si σ et σ' sont deux unificateurs principaux d'un même système S, ils sont identiques à renommage de variables près.

Plus précisément, appelons *renommage* une une substitution à support fini dont les images sont des variables et qui est bijective des variables dans les variables.

Fait 2.2.1 *Si σ est un unificateur principal de S, alors σ' est un unificateur principal de S si et seulement si il existe un renommage α tel que $\sigma' = \alpha \circ \sigma$.*

Démonstration. exercice.

En pratique, lors de l'unification, d'une façon ou d'une autre, on essaye de « quotienter » par les renommages de variable. On peut manipuler par exemple des couples constitués d'un ensemble de variables et d'un terme substituable à chacune de ces variables.

Nous montrerons dans la suite (voir proposition 2.2.22 page 81) la proposition suivante.

Proposition 2.2.2 *Un système S est unifiable si et seulement s'il existe un unificateur principal pour S.*

Un algorithme d'unification fournira un unificateur principal dans le cas où le système est unifiable.

2.2.2 Algorithme d'unification.

L'algorithme d'unification étudié ici serait assez proche de l'algorithme original d'Herbrand. Une version moderne, avec beaucoup d'améliorations au point de vue efficacité, se trouve dans l'article [Martelli Montanari 82].

Une idée simple pour trouver un unificateur serait de « superposer » les termes à unifier. Quand il faut identifier deux symboles de constante ou fonction différents, l'unification échoue. Sinon et si les termes ne sont pas identiques, un terme doit être unifié à une variable, on effectue alors la substitution induite ce qui élimine cette variable et on recommence. La substitution n'est pas possible, soit quand on doit identifier deux constantes ou symboles de fonctions distincts (*clash*), soit quand on doit substituer à une variable un terme qui contient cette variable (*test d'occurrence*, ou *occur-check*)

Le défaut, au moins théorique, de la méthode grossièrement esquissée ci-dessus apparaît sur les deux exemples ci-dessous :

$$(x_0, fx_1x_1), (x_1, fx_2x_2), \ldots, (x_{n-1}, fx_nx_n), (x_n, fx_0x_0)$$

$$(x_0, fx_1x_1), (x_1, fx_2x_2), \ldots, (x_{n-1}, fx_nx_n), (x_n, fx_{n+1}x_{n+1})$$

Le premier de ces systèmes n'est pas unifiable (occur-check), le second l'est. Mais cela apparaîtra par la méthode esquissée ci-dessus après avoir effectué n substitutions et calculé un terme de taille 2^n.

Par ailleurs le test d'occurrence est la seule raison qui fasse échouer l'unification du premier système, et le second système définit une substitution par compositions successives. On va donc accepter une telle forme comme solution de l'unification. Du coup le test d'occurrence devient un peu moins immédiat.

Systèmes réduits, systèmes résolus.

Une substitution à support fini est décrite par un ensemble fini de couples dont les premiers membres sont des variables distinctes deux à deux, les second membres des termes sans occurrences de la variable qui est premier membre du même couple :

$$s = \{(x_1, t_1), \ldots, (x_n, t_n)\} \text{ avec } i \neq j \Rightarrow x_i \neq x_j.$$

On interprète s comme la substitution *simultanée* des t_i aux x_i.

Pour unifier un système on va procéder par réduction du système à une forme analogue, qui ne décrira pas cependant une substitution simultanée mais une composée de substitutions élémentaires.

On va donc commencer par travailler sur des systèmes dont le membre gauche est une variable.

La définition suivante isole des systèmes dont l'unification ne peut échouer que par le test d'occurrence (voir le fait 2.2.7 page ci-contre qui suit).

Définition 2.2.3 Un *système réduit* est un ensemble fini de couples (variable,terme) soit

$$S = \{(x_1, t_1), \ldots, (x_n, t_n)\} \text{ vérifiant } i \neq j \Rightarrow x_i \neq x_j.$$

Introduisons une relation d'équivalence naturelle associée à un système S quelconque. Nous noterons $var(S)$ pour l'ensemble des variables du langage L qui apparaissent dans L.

Définition 2.2.4 La relation $\sim_S$, est la plus petite relation sur $var(S)$ définie par :

- **i.** $x \sim_S x$;
- **ii.** si (x, y) appartient à S (y variable), $y \sim_S x$ et $x \sim_S y$;
- **iii.** si $x \sim_S y$ et $y \sim_S z$, alors $x \sim_S z$.

On a que

Fait 2.2.5 *Soit S un ensemble de couples.*

1. *La relation $\sim_S$ est une relation d'équivalence.*
2. *Si $x \sim_S x'$, tout unificateur s de S vérifie $s(x) = s(x')$;*

Démonstration. Le 1 découle de la définition. Le 2 est immédiat par induction sur la définition de $\sim_S$. ■

Voyons maintenant quelques propriétés de cette relation sur les systèmes réduits.

Lemme 2.2.6 *Soit S un système réduit. Pour toutes variables x et x', pour tout terme t, si $(x,t) \in S$ et $x \sim_S x'$ et $x \neq x'$, alors x' apparaît à gauche d'un couple de deux variables dans S.*

Démonstration. Par induction sur la définition de $\sim_S$. La clause i page précédente est exclue. Comme le système est réduit $(x,x') \in S$ est exclu, donc la clause ii page ci-contre donne $(x',x) \in S$. Dans le cas de la clause iii page précédente (transitivité), on a un y tel que $x \sim_S y$ et $y \sim_S x'$. Par hypothèse d'induction il existe z tel que $(y,z) \in S$, à nouveau par hypothèse d'induction x' apparaît à gauche d'un couple de deux variables dans S. ■

Fait 2.2.7 *Si S est un système réduit, si $(x,t) \in S$, $(x',t') \in S$ avec $x \neq x'$ et t et t' ne sont pas des variables, alors $x \not\sim_S x'$.*

Démonstration. D'après le lemme précédent, et comme chaque variable n'apparaît qu'une seule fois à gauche dans un système réduit. ■

Pour qu'un système réduit soit unifiable, il suffit de trouver un ordre dans lequel composer les substitutions élémentaires qu'il décrit. Pour cela on associe à un système S une relation binaire $<_S$ sur les variables. La définition qui suit ne suppose pas que S est réduit.

Définition 2.2.8 La relation $<_S$ est la plus petite relation sur $var(S)$ vérifiant :

i. si (x,t) appartient à S et t n'est pas une variable et y apparaît dans t alors $y <_S x$

compatible avec $\sim_S$:

ii. si $x \sim_S x'$ et $x <_S y$, alors $x' <_S y$

iii. si $y \sim_S y'$ et $x <_S y$, alors $x <_S y'$.

et close par transitivité :

iv. si $x <_S y$ et $y <_S z$, alors $x <_S z$

Dans la suite $|t|$ désigne le nombre de symboles du terme t. On a :

Fait 2.2.9

1. *Si la relation $<_S$ est un ordre strict, et si $x <_S y$, alors $x \not\sim_S y$.*
2. *si $x <_S y$, alors pour toute unificateur s de S, $|s(x)| > |s(y)|$.*
3. *Soit $<_S /_{\sim_S}$ la relation induite sur $var(S)/_{\sim_S}$ par le passage au quotient de $var(S)$ par $\sim_S$. On alors que $<_S$ est un ordre strict ssi $<_S /_{\sim_S}$ est un ordre strict sur $var(S)/_{\sim_S}$.*

Démonstration. Le 1 découle directement de la définition. Pour le 2, on procède par induction sur la définition de l'ordre : pour la clause iv de la définition, cela découle de ce que t n'est pas une variable, contient y et $s(x) = s(t)$; pour les clauses ii et iii c'est une conséquence du 2.2.5 page précédente. 2 page ci-contre, et cela passe à la transitivité. En ce qui concerne le 3 : d'après la définition, pour x et y deux variables quelconques, si $[x]_{\sim_S}$ et $[y]_{\sim_S}$ désignent leurs classes d'équivalences, on a que $[x]_{\sim_S} <_S /_{\sim_S} [y]_{\sim_S}$ ssi $x <_S y$, d'où le résultat. ■

Fait 2.2.10 *Si S est un ensemble de couples tels que $<_S$ a un cycle, alors S n'est pas unifiable.*

Démonstration. Si $<_S$ n'est pas un ordre strict, on a un cycle : $x <_S x$, suivant 2.2.9 page précédente. 2 page précédente le système ne peut être unifiable. ■

On note $[x := t]$ la substitution définie par $s(x) = t$ et $s(y) = y$ pour $y \neq x$.

Proposition 2.2.11 *Soit S un système réduit. On suppose que la relation $<_S$ n'a pas de cycles. Soit ρ une substitution à image dans les variables qui choisit un représentant dans les classes modulo $\sim_S$, c.a.d. :*

$$x \sim_S \rho(x) \text{ et } x \sim_S y \Rightarrow \rho(x) = \rho(y)$$

Soit $<$ un ordre total strict sur $var(S)/_{\sim_S}$, qui prolonge $<_S /_{\sim_S}$. On note aussi $<$ l'ordre (partiel) induit sur $var(S)$. L'image par ρ de S s'écrit de façon unique comme

$$\rho(S) = \{(x_1, t_1), \ldots, (x_n, t_n)\} \cup I$$

où I ne contient que des couples de variables identiques, les t_i ne sont pas des variables et $x_i < x_j$ ssi $i < j$.

Alors :

$$\sigma = [x_1 := t_1] \circ \cdots \circ [x_n := t_n] \circ \rho$$

est un unificateur principal de S.

Démonstration. Vu la définition de α et de $\sim_S$, la substitution α unifie tous les couples de variable de S, en les transformant en couples de variables identiques. La substitution α ne peut identifier deux variables qui apparaissent à gauche dans des couples de S dont le membre droit n'est pas une variable d'après le fait 2.2.7 page précédente.

L'ordre $<$ étant strict sur les classes d'équivalences modulo $\sim_S$, il n'y a qu'une façon d'ordonner et d'indexer les variables $\alpha(x)$ de la façon indiquée.

Si σ' est un unificateur de $S' = \{(x_1, t_1), \ldots, (x_n, t_n)\}$ alors $\sigma' \circ \alpha$ est un unificateur de S. Posons $\sigma' = [x_1 := t_1] \circ \cdots \circ [x_n := t_n]$. Remarquons que $x_i = \alpha(x_i)$ et $t_i = \alpha(t_i)$.

Du fait de la condition sur l'ordre des variables, aucune des variables x_j pour $j \geq i$ n'apparaît dans $t_1, \ldots, t_i$, donc, en posant $s_i = [x_i := t_i]$,

$$\sigma'(S') = \{(s_1(x_1), t_1), (s_1 \circ s_2(x_2), s_1(t_1)), \ldots, (s_1 \circ \cdots \circ s_n(x_n), s_1 \circ \cdots \circ s_{n-1}(t_n))\}$$

et σ' est bien un unificateur de S' donc σ un unificateur de S.

pour montrer que σ est un unificateur principal. Il suffit de montrer que σ' est un unificateur principal de S'. En effet un unificateur s de S va identifier les variables d'une même classe modulo $\sim_S$. On a donc, si s' est la restriction de s à l'image par α du support de S, $s = s' \circ \alpha$. Donc si σ' est un unificateur principal de $\alpha(S)$, $\sigma' \circ \alpha$ est un unificateur principal de S.

Pour montrer que σ' est un unificateur principal de S', on utilise le lemme suivant.

Lemme 2.2.12 *Un système $\{(x, t)\} \cup S$ tel que x n'apparaît ni dans t ni dans S est unifiable par la substitution s ssi il existe une substitution s' telle que $s = s' \circ [x := t]$ et s' unifie S.*

Démonstration (lemme). On suppose que s unifie S. On pose, pour $y \neq x$ $s'(y) = s(y)$ et $s'(x) = x$. On a $s'(t) = s(t)$ et $s'(S) = s(S)$, car x n'apparaît ni dans t ni dans S. Donc s' unifie S, et $s(x) = s(t) = s'(t)$. La réciproque est évidente. ■

Démonstration (fin de la preuve de la proposition). On doit montrer que σ' qui est un unificateur de S', est un unificateur principal. On montre par récurrence sur $i \leq n$ que si s est un unificateur de S', alors $s = \sigma_i \circ [x_{n-i} := t_{n-i}] \circ \cdots \circ [x_n := t_n]$, où σ_i est un unificateur de $(x_1, t_1), \ldots, (x_{n-i}, t_{n-i})$.

Si $i = 0$, c'est évident.

On passe de i à $i+1$ par le lemme 2.2.12 page précédente. Le fait que l'ordre des variables prolonge $<_S$ assure que x_{n-i-1} n'apparaît ni dans t_{n-i-1} ni dans les t_j pour $j < n-i-1$.

On en déduit en posant $i = n$ que σ' est un unificateur principal de S'. ■

Proposition 2.2.13 *Un système réduit S est unifiable si et seulement si la relation $<_S$ est un ordre strict (n'a pas de cycles).*

Démonstration. Si $<_S$ est un ordre strict la proposition précédente donne un unificateur.

Si $<_S$ n'est pas un ordre strict, S n'est pas unifiable (fait 2.2.10 page 75). ■

Définition 2.2.14 Un *système résolu* est un système réduit S tel que $<_S$ est un ordre. Un unificateur principal défini tel que ci-dessus à la proposition 2.2.11 page précédente est appelé substitution définie par le système résolu.

Remarquons qu'une substitution définie par un système résolu dépend de la donnée de la substitution α et de l'ordre strict $<$ tels qu'en 2.2.11 page ci-contre, mais que , comme il s'agit d'un unificateur principal, elle est unique à renommage près.

Il est bien sûr décidable, mais de plus c'est algorithmiquement raisonnable (linéaire en temps et en espace, voir Knuth The Art of C. Pr., Vol 1, A. W. 1969 pp258-268, tri topologique), de tester si une relation finie donnée a un cycle.

Pour montrer l'existence d'un unificateur principal pour un système unifiable, il nous reste donc à montrer qu'un système est toujours équivalent pour l'unification à un système réduit. C'est l'objet de ce la section suivante.

On peut décomposer un algorithme d'unification en deux partie,

une première partie conduit à un système réduit et teste au passage les "clash" (deux termes à unifier ne commençant pas par le même symbole de fonction) ;

une deuxième partie où l'on teste si le système réduit est résolu, ("occur check").

Les deux parties ne sont pas nécessairement disjointes : dans l'algorithme de Martelli-Montanari ([Martelli Montanari 82]) qui est très utilisé, on teste l'acyclicité au cours de la réduction, ce qui est meilleur en cas d'échec par test d'occurrence. Il semble que l'on arrive à de meilleurs résultats théoriques avec un test d'acyclicité disjoint final. L'unification sans test d'occurrence revient à de l'unification avec termes infinis. En cas de suite importante d'unifications (prolog, résolution), il pourrait être beaucoup plus économique de faire un seul test d'acyclicité à la fin (cf. [Martelli Rossi 84]).

Réduction d'un système à unifier.

On définit deux types de réduction sur les systèmes. Le but est de transformer le système de départ en un système réduit ayant mêmes unificateurs. La relation $\triangleright$ est la clôture transitive des réductions définies ci-dessous.

Réduction 1 (simplification).

1. si $f \neq g$, alors $\{(ft_1 \dots tp, gt'_1 \dots t'_n)\} \cup S$ ne se réduit pas (CLASH)
2. $\{(ft_1 \dots tp, ft'_1 \dots t'_p)\} \cup S \triangleright_1 \{(t_1, t'_1), \dots, (t_p, t'_p)\} \cup S$
3. $\{(x, x)\} \cup S \triangleright_1 S$
4. $\{(t, x)\} \cup S \triangleright_1 \{(x, t)\} \cup S$, si t n'est pas une variable.

Remarquons que, dans les trois premiers cas, une étape de Réduction 1 fait décroître strictement la somme des longueurs des termes de S, et donc il ne peut y avoir de suite infinie d'étapes de cette réduction.

Fait 2.2.15 *Une suite de* Réduction 1 *est nécessairement finie.*

La solution qui avait été esquissée au début consiste à itérer autant que possible la simplification (Réduction 1), et la substitution d'une variable, le test d'occurrence étant fait à cette occasion :

$$\{(x, t)\} \cup S \triangleright \{(x, t)\} \cup S[t/x] \text{ où } x \text{ n'apparaît pas dans } t \qquad \textbf{subst}$$

Exercice 18 Montrer en utilisant cette méthode que si S est unifiable, S a un unificateur principal, qui est fourni par les règles de réduction ci-dessus.

On a vu que l'inconvénient de l'algorithme induit par les règles de simplification et la règle **subst** était un temps de calcul pouvant être exponentiel en la taille du système original. Pour éviter ceci, on va donc éviter de substituer :

Réduction 2 (superposition).

$$\{(x, t_1), (x, t_2)\} \cup S \triangleright_2 \{(x, t_1), (t_1, t_2)\} \cup S\,.$$

Il peut y avoir une suite infinie de telles réductions dès qu'elle est autorisée quand t_1 est une variable. Même en se restreignant aux cas où t_1 et t_2 ne sont pas des variables, il peut y avoir des suites infinies d'étapes de Réduction 2 et 1 comme le montre l'exemple qui suit.

$$(x, ffx), (x, fx) \triangleright_2 (x, fffx), (fx, ffx) \triangleright_1 (x, ffx), (x, fx) \dots$$

On note $\triangleright = \triangleright_1 \cup \triangleright_2$.

On vérifie immédiatement que

Fait 2.2.16 *Si $S \triangleright S'$ alors s est un unificateur de S si et seulement si s est un unificateur de S', ou encore ssi S est un unificateur de $S \cup S'$.*

On peut donc interrompre la réduction $S \triangleright S_1 \triangleright \dots \triangleright S_i$ dès détection d'un cycle pour la relation $<_{S_i}$ associée au système (fait 2.2.10 page 75), ou même pour la réunion des relations associées aux systèmes obtenus au cours de la réduction $S \cup S_1 \cup \dots \cup S_i$, ce qui est en fait plus simple : dans ce cas il suffit, au cours de la réduction, d'étendre la relation.

Par ailleurs, par définition d'un système réduit :

Fait 2.2.17 *Un système S est réduit si et seulement si aucune des 2 étapes de réductions définies ci-dessus ne peut s'appliquer à S.*

On peut remarquer que :

Fait 2.2.18 *si $S \triangleright_1^* S'$ alors $<_S \subset <_{S'}$*

Mais ceci est faux pour la RÉDUCTION 2.

Il nous reste à montrer que

Proposition 2.2.19 *Si S est unifiable, il existe un système réduit S' tel que $S \triangleright S'$. Alors une substitution σ est un unificateur de S si et seulement si elle est un unificateur de S'.*

Pour cela il suffit, d'après les faits 2.2.16 page ci-contre et 2.2.17, d'ajouter une condition pour l'application de l'étape RÉDUCTION 2 qui assure la terminaison d'une suite d'étapes de réductions. Il y a plusieurs façons d'assurer cette terminaison, qui conduisent à des algorithmes différents. On décrit deux solutions ci-dessous qui peuvent d'ailleurs être associées. La première consiste à faire le test d'occurrence au cours de la réduction, de façon à assurer la terminaison. La seconde consiste à se servir d'un ordre sur les termes pour restreindre l'usage de la deuxième réduction. Il est en fait plus efficace de modifier la deuxière réduction, et de travailler à équivalence $\ _S$. C'est ce qui est fait au § 2.2.2 auquel on peut directement se reporter pour une preuve de la proposition 2.2.19.

Première solution.

Une première méthode est de faire le test d'occurrence pendant la réduction, ce qui donne un argument de terminaison. L'algorithme induit n'a pas d'intérêt propre, mais la méthode peut être associée à d'autres plus efficaces.

Le principe est d'une part d'appliquer l'étape de RÉDUCTION 1 dès que possible, d'autre part d'appliquer l'étape de RÉDUCTION 2 pour une variable dont le nombre d'occurrences dans les membres gauches des couples n'augmentera pas au cours de la réduction. Précisons.

$ord(x, \prec) = \mathsf{card}\{z \ / \ z \not\succ x\}$
$occg_S(x) = \mathsf{card}\{(x', t) \ / \ (x', t) \in S \text{ et } x \sim_S x'\}$
$occ2(S) = \{x \ / \ occg_S(x) \geq 2\}$
$eq_S(x) = \mathsf{card}\{(x, y) \in S \ / \ y \in var(S)\} max(S, \prec) = \{x \ / \ x \text{ maximale pour } \prec \text{ dans } occ2(S)\}$
$mes(S, \prec) = (\sup_{x \in max(S)} ord(x, \prec), \sum_{x \in max(S)} occg_S(x), \sum_{x \in max(S)} eq_S(x)\})$

On a :

Lemme 2.2.20 *On suppose donnée une réduction de S vérifiant les conditions de la figure 2.1 page suivante. $(S_i)_i \leq n$ est la suite des systèmes obtenus au cours de la réduction. On pose $\prec_i = (\bigcup_{j=0}^{i} <_j)^*$. Supposons que $S_i \triangleright_2 S_{i+1} \triangleright_1^* S_k$. Alors $mes(S_i, \prec_i) < mes(S_k, \prec_k)$.*

Démonstration. Posons

$$S_i = \{(x, t_1), (x, t_2)\} \cup S' \triangleright_2 \{(x, t_1), (t_1, t_2)\} \cup S' = S_{i+1}\,.$$

On demande que dans une réduction

$$S = S_0 \triangleright S_1 \triangleright \cdots \triangleright S_n$$

Pour

$$S_i = \{(x, t_1), (x, t_2)\} \cup S' \triangleright_2 \{(x, t_1), (t_1, t_2)\} \cup S' = S_{i+1}\,.$$

tous les couples de S_i ont une variable comme membre gauche (la RÉDUCTION 1 n'est pas possible)
$(<_{S_1} \cup \cdots <_{S_i})^*$ est un ordre strict (sinon le système n'est pas unifiable OCCUR CHECK).
la variable x est maximale pour la relation $(<_{S_1} \cup \cdots <_{S_i})^*$ parmi les variables apparaissant au moins deux fois à gauche (existe quand $(<_{S_1} \cup \cdots <_{S_i})^*$ est un ordre),
t_1 n'est pas une variable.

FIGURE 2.1 – Conditions pour la RÉDUCTION 2 : première solution

Si t_2 est une variable, la troisième composante décroit.

Supposons que t_2 n'est pas une variable. Les étapes de décomposition (RÉDUCTION 1) sur S_{i+1} concernent nécessairement des sous-termes de t_1 et t_2 dans lesquels n'apparaissent que des variables inférieures à x pour $<_{S_i}$ donc pour $\prec_i$.

Soit y une variable telle que $y \prec_i x$ donc $y \prec_{i+1} x$. On a pour z une variable quelconque par transitivité $(z \succ_i x \Rightarrow z \succ_{i+1} y)$ donc $(z \not\succ_{i+1} y \Rightarrow z \not\succ_i x)$ donc $ord(y, \prec_{i+1}) \leq ord(x, \prec_i)$. Par ailleurs $x \not\succ_i x$ et $x \succ_{i+1} y$, donc on a que $ord(y, \prec_i) < ord(x, \prec_i)$.
Supposons maintenant que x soit la seule variable dans $max(S_i, \prec_i)$ et que x ait seulement deux occurrences à gauche dans S. Alors les variables de $max(S_k, \prec_k)$ sont des variables inférieures à x pour $\prec_i$ soient qu'elles soient dans t_1 ou t_2, soient qu'elles soient majorées par x dans S_i et d'après ce qui précède $mes(S_k, \prec_k) < mes(S_i, \prec_i)$ (première composante).
Supposons que x soit la seule variable dans $max(S_i, \prec_i)$ et que x ait plus de deux occurrences à gauche dans S. Alors x reste maximale et la seconde composante décroit.

Rest à traiter le cas où x n'est pas la seule variable dans $max(S_i, \prec_i)$. Dans ce cas à nouveau la seconde composante de a mesure décroit.

On en déduit le lemme suivant :

Lemme 2.2.21 *Une suite de réductions vérifiant les conditions indiquées à la figure 2.1 est nécessairement finie.*

Démonstration. Par l'absurde : supposons une suite infinie de réductions vérifiant les conditions indiquées à la figure 2.1. Cette suite contient nécessairement une infinité d'étapes de RÉDUCTION 2 d'après le fait 2.2.15 page 78 ce qui est impossible d'après le lemme précédent. ■

On est maintenant en mesure de prouver la proposition.

Démonstration (preuve de la proposition 2.2.19 page 79). étant donné S, il existe une suite finie de réductions d'origine S vérifiant les conditions indiquées à la figure 2.1 page ci-contre et qui est maximale d'après le lemme précédent. Le système S_0 obtenu est réduit. ■

L'algorithme d'unification induit est clair : on itère l'application de la RÉDUCTION 1 tant que c'est possible, puis la RÉDUCTION 2 sous les conditions de la figure 2.1 page précédente. On calcule l'ordre associé au système au cours de la réduction.

Pour que l'algorithme soit efficace, il faut modifier l'étape de RÉDUCTION 2 et traiter des ensembles de variables (équivalentes par $\sim_S$) et des ensembles de termes chacun substituable à ces variables (voir le § 2.2.2[Martelli Montanari 82]).

Proposition 2.2.22 *Le système S est unifiable, si et seulement s'il existe une réduction $S \triangleright^* S_0$ qui termine sur un système résolu. Une substitution définie par le système résolu est un unificateur principal de S.*

Démonstration. conséquence des propositions 2.2.19 page 79 et 2.2.13 page 77.

On a pour corollaire la proposition 2.2.2 page 73 annoncée en début de section.

Seconde solution

Cette solution n'est qu'esquissée.

Donnons maintenant un algorithme de réduction indépendant du test d'occurrence. Pour cela on donne une condition pour la deuxième réduction qui ne dépend pas de l'ordre.

$$\{(x, t_1), (x, t_2)\} \cup S' \triangleright_2 \{(x, t_1), (t_1, t_2)\} \cup S'$$

pour $|t_1| \leq |t_2|$ et $t_1 \neq x$, $t_2 \neq x$, et étant donné un ordre total arbitraire $<$ sur les variables, si $t_1 = y$, alors $x < y$.

FIGURE 2.2 – Conditions pour la RÉDUCTION 2 : deuxième solution

La terminaison est assurée essentiellement par décroissance de l'ordre lexicographique sur les suites des termes à gauche dans S rangés par ordre décroissant (cet ordre ne décroit pas strictement à chaque étape de RÉDUCTION 2 mais il ne peut y avoir de suite infinie de telles réductions). Ici l'ordre sur les termes est l'ordre sur la taille de ceux-ci, prolongé par l'ordre total strict arbitraire sur les variables de la figure 2.2.

Remarquons que l'on pourrait, en suivant ce deuxième algorithme, donner une version plus générale de la proposition 2.2.19 page 79, en enlevant la condition que S est unifiable.

La comparaison sur les longueurs n'a pas grand intérêt algorithmique. Il est plus efficace de modifier le deuxième étape de réduction de façon à utiliser un ordre qui respecte la structure des termes. On peut choisir (voir [Martelli Montanari 82], [Martelli Rossi 84])

RÉDUCTION 2'.

$$\{(x, t_1), (x, t_2)\} \cup S \triangleright \{(x, t), (t, t_1), (t, t_2)\} \cup S$$

où t est la *partie commune* de t_1 et t_2, si elle existe,

c'est à dire le terme obtenu en « superposant » les deux termes t_1 et t_2 et en prenant « l'intersection ». Si on ne peut trouver un tel terme, (t_1, t_2) n'est pas unifiable.

Le calcul de t se fait en fait en même temps que la réduction par simplification ($\triangleright_1$) de (t, t_1) et (t, t_2). Le résultat obtenu est appellé *frontière* de t_1, t_2.

Un algorithme comme celui de Martelli-Montanari (voir [Martelli Montanari 82]) calcule la partie commune et la frontière de plusieurs termes, le test d'occurrence étant fait à chaque étape de façon analogue à ce qui est indiqué en figure 2.1 page 80.

Une solution plus efficace pour la réduction d'un système à unifier.

On précise dans cette partie la solution à l'unification par calcul de partie commune et frontière de Martelli-Montanari ([Martelli Montanari 82]) esquissée ci-dessus.

On ne va plus travailler sur des couples de termes, mais sur des couples d'ensembles de variables et de multi-ensembles de termes.

La substitution σ unifie l'ensemble de termes $\{t_1, \ldots, t_n\}$) signifie que :

$$\sigma(t_1) = \cdots = \sigma(t_n) \ .$$

La substitution σ unifie $(\{x_1, \ldots, x_m\}, \{t_1, \ldots, t_n\})$ signifie qu'elle unifie $\{x_1, \ldots, x_m\} \cup \{t_1, \ldots, t_n\}$:

$$\sigma(x_1) = \cdots = \sigma(x_m) = \sigma(t_1) = \cdots = \sigma(t_n) \ .$$

Une substitution σ unifie un ensemble $\mathcal{S}$ de tels couples si elle unifie chacun des couples de $\mathcal{S}$. De la même façon on peut parler d'ensemble unifiable, d'unificateur le plus général et de système réduit :

Définition 2.2.23 Un tel système $\mathcal{S}$ est *réduit* quand :

i. si $(X, T) \in \mathcal{S}$ alors $T = \emptyset$ où $T = \{t\}$ est un singleton.

ii. si $(X, T) \in \mathcal{S}$ et $(X', T') \in \mathcal{S}$ alors $X \cap X' = \emptyset$.

iii. $(\emptyset, T) \notin \mathcal{S}$, (où T vide ou singleton quelconque).

Les deux notions de système unifiables sont équivalentes ; pour tous termes u, v on a :

$$(u, v) \text{ est unifiable} \quad \text{ssi} \quad (\emptyset, \{u, v\}) \text{ est unifiable}$$

et pour tout ensemble de variable $\{x_1, \ldots, x_m\}$ et tout multi-ensemble de termes $\{t_1, \ldots, t_n\}$, l'un des deux étant non vide, et $\alpha \in \{x_1, \ldots, x_m\} \cup \{t_1, \ldots, t_n\}$:

$$\begin{gathered}(\{x_1, \ldots, x_m\}, \{t_1, \ldots, t_n\}) \text{ est unifiable} \\ \text{ssi} \\ \{(x_1, \alpha), \ldots, (x_m, \alpha), (\alpha, t_1), \ldots, (\alpha, t_n)\} \text{ est unifiable.}\end{gathered}$$

Si de plus on choisit pour α une variable, un système réduit au nouveau sens est traduit en un système réduit en l'ancien sens. Pour la réciproque la traduction ci-dessus ne convient pas, il n'est pas très difficile d'en donner une (il faut regrouper les variables de S par classes d'équivalence pour $\sim_S$).

On définit un premier ensemble de règles de simplifications :

Préparation.

1. $\{(X, \{x\} \cup T)\} \cup \mathcal{S} \rhd_p \{(X \cup \{x\}, T)\} \cup \mathcal{S}$ pour x variable.
2. $\{(X, T), (X', T')\} \cup \mathcal{S} \underset{X \cap X' \neq \emptyset}{\rhd_p} \{(X \cup X', T \cup T')\} \cup \mathcal{S}$
3. $\{(\emptyset, \emptyset)\} \cup T \rhd_p T$.
4. $\{(\emptyset, \{t\})\} \cup T \rhd_p T$.

Les deux dernières règles sont là pour assurer que tous les cas de figure sont envisagés mais ne sont jamais appliquées sur des exemples "réalistes".

Clairement il ne peut y avoir une suite infinie de telles simplifications (le nombre de variables à gauche et le nombre de variables communes à plusieurs membres droits diminuent).

Le test de CLASH se fait dans le calcul de la partie commune et de la frontière pour un ensemble de termes. La partie commune de $\{t_1, \ldots, t_n\}$, notée $\mathrm{com}(\{t_1, \ldots, t_n\}$, est un terme, et la frontière de $\{t_1, \ldots, t_n\}$, notée $(\mathrm{frt}(\{t_1, \ldots, t_n\})$, est un ensemble de couples constitués d'un ensemble de variables et d'un multi-ensemble de termes. Elles sont définies par :

a. $\mathrm{com}(X \cup T) = x$, $\mathrm{frt}(X \cup T) = (X, T)$, où, X est un ensemble de variables, T un ensemble de termes qui ne sont pas des variables et x est "la plus petite variable" de X pour un ordre arbitraire prédéterminé sur les variables.

b. Pour $\{f\overline{t}, g\overline{u}\} \cup T)$, où T est un ensemble de termes et $f \neq g$, parties communes et frontières ne sont pas définies (CLASH).

c. Pour $T = \{ft_{1,1} \ldots t_{1,k}, \cdots, ft_{p,1} \ldots t_{p,k}\}$, un ensemble de termes qui ont tous même symbole principal :

$$\mathrm{com}(\{ft_{1,1} \ldots t_{1,k}, \cdots, ft_{p,1} \ldots t_{p,k}\}) = f\mathrm{com}(\{t_{1,1} \ldots t_{1,k}\}) \ldots \mathrm{com}(\{t_{p,1} \ldots t_{p,k}\})$$

$$\mathrm{frt}(\{ft_{1,1} \ldots t_{1,k}, \cdots, ft_{p,1} \ldots t_{p,k}\}) = \mathrm{frt}(\{t_{1,1} \ldots t_{1,k}\}) \cup \cdots \cup \mathrm{frt}(\{t_{p,1} \ldots t_{p,k}\})$$

On peut maintenant compléter les règles pour l'unification.

Superposition+Simplification.

$(X, T) \cup \mathcal{S} \rhd_s \{(X, \mathrm{com}(T))\} \cup \mathrm{frt}(T) \cup \mathcal{S}$ pour T possède au moins deux éléments.

On note $\rhd = \rhd_p \cup \rhd_s$. On vérifie facilement les deux faits suivants.

Fait 2.2.24 *Si $\mathcal{S} \rhd \mathcal{S}'$ alors σ unifie $\mathcal{S}$ si et seulement si σ unifie $\mathcal{S}'$.*

Fait 2.2.25 *Si aucune des règles $\rhd_s$ ou $\rhd_p$ ne s'applique à $\mathcal{S}$ alors $\mathcal{S}$ est réduit au sens défini ci-dessus (2.2.23 page ci-contre).*

Il reste à préciser l'enchaînement des règles de simplification $\rhd_p$ et $\rhd_s$ pour obtenir :

Proposition 2.2.26 *Un ensemble $\mathcal{S}$ de couples constitué d'un ensemble de variables et d'un multi-ensemble de termes est unifiable si et seulement s'il se simplifie par une succession de règles $\rhd_p$ et $\rhd_s$ en un système $\mathcal{S}'$ réduit (2.2.23 page précédente). Alors une substitution σ est un unificateur de $\mathcal{S}$ ssi elle est un unificateur de $\mathcal{S}'$.*

Démonstration. On convient de toujours appliquer prioritairement les règles $\triangleright_p$. Cela signifie en particulier que le calcul de partie commune et de frontière n'est effectué que pour un ensemble de termes T ne contenant pas de variables. On définit une mesure sur les termes qui consiste à compter le nombre de symboles qui ne sont pas des variables :

- $mes(x) = 0$ pour x variable ;
- $mes(ft_1 \dots t_n) = 1 + mes(t_1) + \cdots + mes(t_n)$ pour f un symbole de fonction d'arité n avec $n \in \mathbb{N}$.

On étend cette mesure aux ensembles de termes, et aux systèmes à unifier :

- $mes(\{t_1, \dots, t_n\}) = mes(t_1) + \cdots + mes(t_n)$;
- $mes(\{(X_1, T_1), \dots, (X_p, T_p)\} = mes(T_1) + \cdots + mes(T_p)$.

Les étapes $\triangleright_p$ conservent la mesure ou la font diminuer. Les étapes $\triangleright_s$ la font décroître. En effet quand T ne contient pas de variables, $mes(\mathrm{com}(T)) > 0$. Par ailleurs, si T a n éléments on vérifie facilement par induction sur la définition de la partie commune et de la frontière que :

$$mes(T) = n.mes(\mathrm{com}(T)) + mes(\mathrm{frt}(T))$$

comme la règle $\triangleright_s$ ne s'applique que pour T ayant au moins deux élements on en déduit que :

$$\text{si } \mathcal{S} \triangleright_s \mathcal{S}' \text{ alors } mes(\mathcal{S}) < mes(\mathcal{S}').$$

Dans une suite de réductions $\triangleright_p$ et $\triangleright_s$ avec priorité à $\triangleright_p$ il ne peut donc y avoir qu'un nombre fini d'étapes $\triangleright_s$. Comme par ailleurs il ne peut y avoir qu'un nombre fini d'étapes $\triangleright_p$ successives, une telle suite est finie, et d'après le fait 2.2.25 page précédente le système obtenu est réduit. On conclue par le fait 2.2.24 page précédente. ■

On obtient la proposition 2.2.19 page 79 comme corollaire de celle-ci. Il suffit d'utiliser les traductions indiquées en début de paragraphe.

2.3 Le calcul des prédicats : formes de Herbrand.

La méthode de recherche de preuves esquissée dans l'introduction du § 2.2 page 72 : utiliser des meta-variables dans les règles $\forall_g$ et $\exists_d$ pour reporter le choix du terme à la mise en évidence d'axiomes fait l'impasse sur le problème suivant : une fois la subtitution effectuée obtient-on une preuve correcte en calcul des séquents ? Cela va s'avérer faux en général à cause de la condition sur les paramètres propres dans les règles $\forall_d$ et $\exists_g$, comme le montre l'exemple suivant :

Dans ce § on va se restreindre au cas où la condition sur les paramètres propres ne peut pas poser de problèmes.

La façon la plus simple est de se restreindre aux séquents sans $\forall$ en position positive et $\exists$ en position négative, en simplifiant on se restreint à la prouvabilité des formules existentielles (non nécessairement closes) ou encore à la satisfaisaibilité (c'est à dire la non contradiction) des formules universelles.

On verra ensuite au § 2.4 page 89 comment traiter le cas général, soit en se ramenant à ce cas particulier, soit en restreignant l'unification.

2.3.1 Formes de Herbrand, formes de skolem.

En fait les $\forall$ en tête des formules à droite et les $\exists$ en tête de formules à gauche ne posent pas véritablement de problèmes, puisque l'on ne perd rien à supposer que dans une preuve d'un tel séquent, les règles correspondantes sont à la racine. Pour simplifier on va se restreindre aux formules prénexes. Il n' y a d'ailleurs essentiellement qu'une forme prénexe naturelle (en respectant la structure propositionnelle) d'une formule n'ayant soit que des $\forall$ négatifs, soit que des $\exists$ positifs, l'ordre des quantificateurs en tête, tous de même nature, n'ayant pas d'importance. On ne perd donc pas vraiment la structure de la formule originale par mise sous forme prénexe, ce qui n'est pas le cas en général.

Définition 2.3.1 Une formule prénexe de la forme :

$$\exists a_1 \dots \exists a_p \forall x_1, \dots, \forall x_n F \quad F \text{ propositionnelle}$$

est dite *en forme de Herbrand.*

Une formule sous forme prénexe de la forme :

$$\forall x_1, \dots, \forall x_n \exists a_1 \dots \exists a_p F \quad F \text{ propositionnelle}$$

est dite *en forme de Skolem.*

Un séquent est en forme de Herbrand quand toutes les formules à droite sont en forme de Herbrand et quand toutes les formules à gauche sont en forme de Skolem.

On peut appliquer le théorème du séquent médian aux séquents en forme de Herbrand. En particulier on peut montrer une forme plus précise suivante :

Proposition 2.3.2 *Si un séquent en forme de Herbrand est démontrable, il a une preuve en calcul des séquents version montante qui vérifie que :*

- **i.** *les affaiblissements sont au dessus de toutes les autres règles ;*
- **ii.** *les règles propositionnelles sont au dessus des règles sur les quantificateurs et des contractions ;*
- **iii.** *les règles $\forall_g$ et $\exists_d$ sont au dessus des contractions et des règles $\forall_d$ et $\exists_g$;*
- **iv.** *les règles de contraction sont au dessus des règles $\forall_d$ et $\exists_g$.*

Démonstration. On sait déjà (théorème du séquent médian) qu'il existe une preuve vérifiant les deux premières conditions. Ensuite il est toujours possible de faire commuter une règle $\forall_d$ ou $\exists_g$ vers le bas avec une règle $\exists_g$ ou $\forall_d$. Il est toujours possible de faire commuter les contractions avec les quantificateurs des deux façons nécessaires pour obtenir le résultat :

[*à compléter*]

Cela signifie qu'il suffit, pour contrôler les contractions, d'associer un entier à chaque formule qui est à droite et qui et qui contient au moins un quantificateur existentiel, ou qui est à gauche et qui contient au moins un quantificateur universel.

On va appeller *multiplicité* d'un séquent en forme de Herbrand une fonction qui associe un entier à toute première occurrence d'un quantificateur existentiel dans une formule à droite, et toute première occurrence d'un quantificateur universel dans une formule à gauche. On

$$\frac{}{\Gamma, Pt_1, \ldots t_n \vdash Pt'_1 \ldots t'_n, \Delta} \; P \text{ atomique, } \sigma \text{ unificateur de } (t_1, t'_1), \ldots, (t_n, t'_n)$$

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} \wedge_g \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta} \wedge_d$$

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta} \vee_g \qquad \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta} \vee_d$$

$$\frac{\Gamma, B \vdash \Delta \quad \Gamma \vdash A, \Delta}{\Gamma, A \to B \vdash \Delta} \to_g \qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \to B, \Delta} \to_d$$

$$\frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta} \neg_g \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta} \neg_d$$

$$\frac{A[?y/x], \Gamma \vdash \Delta}{\forall x A, \Gamma \vdash \Delta} \forall_g \qquad \frac{\Gamma \vdash A[y/x], \Delta}{\Gamma \vdash \forall x A, \Delta} \forall_d$$

$$\frac{\Gamma, A[y/x] \vdash \Delta}{\Gamma, \exists x A \vdash \Delta} \exists_g \qquad \frac{\Gamma \vdash A[?y/x], \Delta}{\Gamma \vdash \exists x A, \Delta} \exists_d$$

y est une variable du langage qui n'a pas d'occurrence libre dans Γ, A, Δ.

$?y$ est une meta-variable qui n'a pas d'occurrence libre dans Γ, A, Δ.

$$\frac{\Gamma, \forall x_1 A[x_1/x], \ldots, \forall x_n A[x_n/x] \vdash \Delta}{\Gamma, \forall^n x A \vdash \Delta} c_g \qquad \frac{\Gamma \vdash \Delta, \exists x_1 A[x_1/x], \ldots, \exists x_n A[x_n/x]}{\Gamma \vdash \Delta, \exists^n x A} c_d$$

TABLE 2.1 – Recherche de preuves pour des formes de Herbrand.

note l'entier en indice du quantificateur correspondant. On peut réécrire la version montante du calcul des séquents en tenant compte de la multiplicité pour la contraction et en utilisant des meta-variables pour l'unification comme indiqué dans le tableau 2.1.

Du point de vue de l'unification, les variables du langage sont considérées comme des constantes, les meta-variables jouent le rôle des variables.

Corollaire 2.3.3 *Une séquent sous forme de Herbrand est démontrable ssi il existe une multiplicité pour ce séquent et une substitution σ telle que le séquent soit démontrable dans le système de règles du tableau 2.1.*

La démontrabilité d'un séquent dans ce système pour une multiplicité donnée est décidable. En effet, en dehors de la contraction, toutes les règles font décroître le nombre de symboles (vues du haut vers le bas) et on ne peut utiliser qu'un nombre fini de contractions. On construit donc en appliquant ces règles de bas en haut un arbre fini. Il reste à énumérer toutes les possibilités pour que les séquents feuilles de l'arbre obtenu soient des axiomes, il ne peut y en avoir qu'un nombre fini, et à utiliser dans chaque cas l'algorithme d'unification.

On obtient un algorithme de semi-décision en énumérant toutes les multiplicités.

2.3.2 Méthode des tableaux.

La méthode des tableaux est essentiellement le calcul des séquents version montante avec unification esquissé ci-dessus. On peut préférer retarder la contraction en utilisant les règles avec contraction internalisée du § 1.3.3. La notation est optimisée comme indiqué au § 2.1.1.

[*à compléter*]

2.3.3 Méthode des connexions.

Comme en calcul propositionnel, il s'agit essentiellement d'une version optimisée en espace du calcul des séquents version montante, plus précisément de celui de la table 2.1 page précédente.

Définition 2.3.4 On définit donc la matrice d'une formule signée avec multiplicité, par induction sur la structure de la formule, pour les connecteurs propositionnels, on reprend la définition 2.1.1 page 69 que l'on complète par :

i. $M((\forall x A)^+) = A[a/x]^+$ où a est une variable du langage (constante pour l'unification) "nouvelle" ;

ii. $M((\forall^n x A)^-) = A[?x_1/x]^- \dots A[?x_n/x]^-$ où les meta-variables $?x_i$ sont distinctes et "nouvelles" ;

iii. $M((\exists^n x A)^+) = A[?x_1/x]^+ \dots A[?x_n/x]^+$ où les meta-variables $?x_i$ sont distinctes et "nouvelles" ;

iv. $M((\exists x A)^-) = A[a/x]^-$ où a est une variable du langage (constante pour l'unification) "nouvelle".

On étend cette définition aux séquents comme en 2.1.1.

Les chemins sont définis de la même façon qu'en calcul propositionnel (2.1.2 page 70). Pour les connexions, on fait intervenir l'unification :

Définition 2.3.5 On appelle *connexion* d'une matrice un couple d'occurrences dans cette matrice de formules atomiques sur un chemin de signes opposés et de même connecteur principal, soit $(\alpha u_1 \dots u_p^+, \alpha v_1 \dots v_p^-)$. La substitution σ, définie sur les meta-variables à valeur dans les termes étendus avec meta-variables, unifie cette connexion quand elle unifie $\{(u_1, v_1), \dots, (u_p, v_p)\}$.

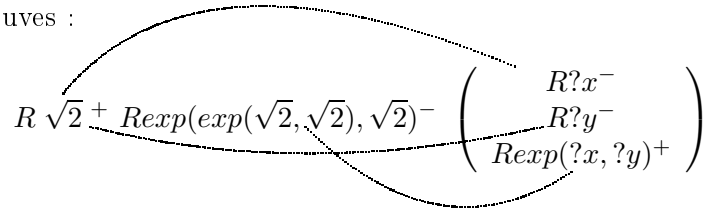
Un *ensemble complet de connexions* d'une matrice M est un ensemble de connexions tel que l'une d'entre elles apparaisse sur tout chemin traversant M. Un *ensemble complet de connexions est unifiable* s'il existe une substitution σ qui unifie toutes les connexions de cet ensemble.

Proposition 2.3.6 *Un séquent en forme de Herbrand est démontrable si et seulement si sa matrice possède un ensemble complet de connexions unifiable.*

Exemple. Le langage contient un symbole de prédicat unaire R (rationnel), un symbole de fonction binaire exp (exponentielle) et un symbole de constante $\sqrt{2}$. On veut démontrer le séquent :

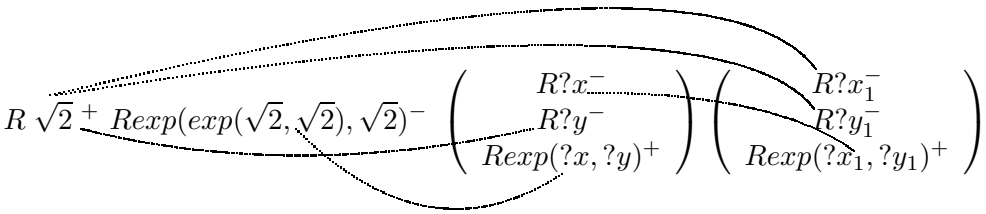
$$R\sqrt{2}, Rexp(exp(\sqrt{2},\sqrt{2}),\sqrt{2}) \vdash \exists x \exists y(\neg Rx \wedge \neg Ry \wedge Rexp(x,y))$$

Ce séquent est sous forme de Herbrand. La seule formule à laquelle une multiplicité peut être attribuée est la formule de droite. Si la multiplicité est de 1 on peut vérifier que la matrice ne fournit pas de preuves :



il n'y a qu'un ensemble complet de connexions : celui indiqué sur le schéma, et il n'est pas unifiable.

On va attribuer une multiplicité de 2 à la formule conclusion on obtient la matrice :



L'ensemble complet de connexions indiqué est unifiable par la substitution σ définie par :

$$\sigma(?x) = exp(\sqrt{2},\sqrt{2}),\ \sigma(?y) = \sigma(?x_1) = \sigma(?y_1) = \sqrt{2}$$

[*à compléter*]

2.3.4 Résolution.

[*à compléter*]

2.3.5 Méthode inverse, traduction en résolution.

[*à compléter*]

2.4 Le calcul des prédicats : cas général.

2.4.1 Skolemisation, herbrandisation.

[à compléter]

2.4.2 Substitutions idempotentes, ordre associé.

On rappelle que le support de la substitution σ, noté $support(\sigma)$, désigne l'ensemble fini des variables x telles que $\sigma(x) \neq x$.

Étant donné une substitution σ, il est toujours possible de trouver un renommage (voir paragraphe 2.2.1 page 72) α tel que, en posant $\sigma' = \alpha \circ \sigma$,

$$\forall x, y \in support(\sigma')\ y \text{ n'apparaît pas dans } \sigma'(x)\ .$$

Par ailleurs

Fait 2.4.1 *La substitution à support fini s est* idempotente, *c.a.d. $s \circ s = s$, ssi :*

$$\forall x, y \in support(s)\ \ y \textit{ n'apparaît pas dans } s(x)\ .$$

On a donc

Fait 2.4.2 *Un système S est unifiable ssi il a un unificateur principal idempotent.*

On peut, à une substitution s, associer un ensemble de variables $var(s)$ deux relations binaires sur les variables notée $\sim_s$ et $<_s$ qui sont l'ensemble $var(S)$, les relations binaires $\sim_S$ et $<_S$ où $S = \{(x, s(x))\ /\ x \in support(S)\}$.

On a immédiatement que :

Fait 2.4.3 *Si s est une substitution définie (voir 2.2.14 page 77) par un système résolu S alors, $var(s) \subset var(S)$:*

$$\forall x,\ y \in var(s)\ (y <_s x \Rightarrow y <_S x)\ .$$

par transitivité de $<_S$ et définition de s comme composée.

Il sera utile par la suite de se restreindre aux substitutions idempotentes, et on aurait pu le faire précédemment :

Proposition 2.4.4 *Une substitution est définie par un système résolu ssi elle est idempotente.*

Démonstration. Supposons que s soit une substitution idempotente de support $x_1, \ldots, x_n$, alors $\{(x_1, s(x_1)), \ldots, (x_n, s(x_n)\}$ est un système résolu qui définit la substitution s.
Réciproquement, supposons que s est une substitution définie par le système résolu S. Supposons que x_j apparaisse dans $s(x_i)$. Comme $<_s$ est inclus dans $<_S$ et que $<_S$ n'a pas de

cycle $i \neq j$. Supposons que $s(x_i)$ ne soit pas une variable, alors $x_j < x_i$. Vu l'ordre de composition des substitutions élémentaires pour définir S (voir proposition 2.2.11 page 76), qui doit respecter $<_S$, il faudrait que $x_j <_S x_j$ ce qui contredit que $<_S$ n'a pas de cycle.

Si maintenant $s(x_i)$ est une variable, $s(x_i) = x_j$. Alors par définition de la substitution s (proposition 2.2.11 page 76) $s(x_j) = x_j$. ■

2.4.3 Unification modulo une relation entre constantes et variables.

On considère le cas où l'on s'interdit, pour les substitutions cherchées, l'occurrence de certaines constantes dans l'image par la substitution d'une certaine variable. Cela intervient en recherche de preuves, les variables correspondent aux instanciations de quantificateurs existentiels à droite ou universels à gauche, certaines constantes aux instanciations de quantificateurs universels à droite ou existentiels à gauche, et dans ce dernier cas la « constante » en question ne doit pas apparaître libre dans le contexte, d'où ce genre de contraintes. On a besoin dans ce cas d'une extension très simple de la notion d'unification, pour laquelle les résultats des paragraphes précédents restent vrais.

Pour un système de couples S, on note $cst(S)$ les constantes de L qui apparaissent dans S. La relation $<_S$ est étendue à une relation entre $cst(S) \cup var(S)$ et $var(S)$ en reprenant la même définition qu'en 2.2.8 page 75. En particulier la clause i page 75 devient :

i' si (x, t) appartient à S, si t n'est pas une variable et si α apparaît dans t (α variable ou constante) alors $\alpha <_S x$

On clôt de façon identique par compatibilité avec $\sim_S$ et transitivité. La restriction de cette nouvelle relation $<_S$ à $var(S) \times var(S)$ est clairement la relation $<_S$ définie en 2.2.8 page 75, ce qui autorise à employer la même notation.

Ces définitions et notations sont étendues aux substitutions comme dans le paragraphe précédent, et on a de la même façon :

Fait 2.4.5 *Si s est une substitution définie par un système résolu S, alors pour toute variable ou constante α, pour toute variable x, ($\alpha <_s x \Rightarrow \alpha <_S x$).*

Définition 2.4.6 Soit $<_E$ une relation entre variables et constantes. Une substitution s est dite *compatible avec la relation* $<_E$ si et seulement si la relation $<_s \cup <_E$, définie sur $cst(s) \cup var(s)$ est sans cycles (la clôture transitive est un ordre).

Unifier modulo la relation $<_E$ un ensemble fini S de couples de termes,

$$S = \{(t_1, t'_1), \ldots, (t_n, t'_n)\}$$

c'est trouver un unificateur s compatible avec la relation $<_E$.

Notons $<_{sE}$, resp. $<_{SE}$, la clôture transitive de $<_E \cup <_s$, resp. $<_E \cup <_S$. (Dans une preuve $<_{sE}$ correspondra à l'ordre que doivent respecter les règles sur les quantificateurs, en remontant, voir la preuve de la proposition 2.5.2 page 96). Ces relations sont définies sur la réunion des constantes et des variables du langage.

On a immédiatement :

Fait 2.4.7 *Une substitution idempotente s unifie S* modulo $<_E$ *si et seulement si s unifie S et $<_{sE}$ est un ordre.*

Lemme 2.4.8 *On a, pour un système résolu S définissant s, une variable x et une constante ou une variable α,*

$$(\alpha <_{sE} x \Rightarrow \alpha <_{SE} x) \tag{1}$$

et si c est une constante

$$(c <_S x \Rightarrow c <_s x) \tag{2}$$

donc, pour α une constante ou une variable

$$(c <_{SE} \alpha \Rightarrow c <_{sE} \alpha)\,. \tag{3}$$

Démonstration. (1) évident.

(2) par définition de s à partir de S (c n'est pas substituable).

(3) On décompose $c <_{SE} d = \alpha$ (idem pour une variable) :

$$c = c_1 <_S x_1 <_E c_2 \cdots <_S x_n <_E c_n = d$$

D'après (2), $c <_{sE} d$. ■

Lemme 2.4.9 *Soit S un système résolu, s un unificateur principal défini par S. Alors la relation $<_{sE}$ est un ordre partiel si et seulement si la relation $<_{SE}$ est un ordre partiel.*

Démonstration. Il est évident que si $<_{sE}$ est un ordre, alors $<_{SE}$ est un ordre. Réciproquement, puisque $<_S$ est un ordre, un cycle contient au moins une constante soit c, $c <_{SE} c$, et d'après (3), $c <_{sE} c$. ■

Lemme 2.4.10 *Si s est une substitution idempotente telle que $<_{sE}$ a un cycle, alors pour toute substitution p, $p \circ s$ a un cycle .*

Démonstration. Si $<_{sE}$ a un cycle, comme s est une substitution idempotente, ce cycle contient nécessairement une constante. Il se décompose en

$c = c_1 <_s x_1 <_E c_2 \cdots <_s x_n <_E c_n = c$

Comme c_i est une constante, si c_i apparaît dans $s(x_i)$, alors c_i apparaît dans $p \circ s(x_i)$, i.e.

$$c_i <_s x_i \Rightarrow c_i <_{p \circ s} x_i$$ ■

Ce dernier lemme assure, pour l'unification modulo un ordre, qu'il suffit de chercher un unificateur principal.

Remarque. Soit s est une substitution idempotente qui unifie S. Alors $p \circ s$ est une substitution idempotente qui unifie S modulo $<_E$ si et seulement si la clôture transitive de $<_s \cup <_E \cup <_p$ est sans cycles.

Proposition 2.4.11 *Si $S \triangleright S'$ alors s est un unificateur de S modulo $<_E$ si et seulement si s est un unificateur de S' modulo $<_E$.*

Démonstration. Évident pour les deux étapes de réduction. ■

On déduit de cette proposition et des lemmes précédents la proposition qui suit.

Proposition 2.4.12 *Un système réduit S est unifiable modulo la relation $<_E$ si et seulement si la relation $<_{SE}$ est un ordre (partiel). Le système est dit alors résolu modulo l'ordre $<_E$. La substitution définie par ce système est un unificateur principal de S et un unificateur de S modulo l'ordre $<_E$.*

Il suffit donc, pour unifier un système modulo un ordre, de réduire le système par une des méthode décrit précédemment, puis, si S est le système réduit obtenu, de vérifier que la relation $<_{SE}$ est sans cycles.

On peut également réduire l'unification modulo un ordre à l'unification ordinaire par une méthode apparentée à la skolémisation.

2.4.4 Skolémisation.

On voit qu'unifier modulo un ordre consiste à ajouter à l'unification ordinaire une nouvelle possibilité d'échouer, en ne respectant pas cet ordre. La skolémisation consiste à traduire le système de façon à ce que l'échec de l'unification dû à l'ordre soit traduit en un échec de l'unification ordinaire par test d'occurrence. Pour cela on remplace les constantes par des termes où apparaissent toutes les variables inférieures à cette constante. Précisons.

On étend le langage avec de *nouveaux symboles de fonctions*, de façon qu'à chaque constante c apparaissant dans le graphe de $<_E$ soit associée une fonction f_c d'arité n le nombre de variables x tel que $x <_E c$. A chaque constante c apparaissant dans le graphe de $<_E$ on associe le terme $t_c = f_c(x_1, \ldots, x_n)$ où $x_1, \ldots, x_n$ sont exactement les n variables distinctes, vérifiant $x_1 <_E c, \ldots, x_n <_E c$. Sauf précision explicite, le même symbole de fonction peut être associé à 2 constantes différentes, mais f_c ne peut être un symbole du langage L de départ. S'il n'y a pas d'ambiguïté, on notera $\mathsf{sk}(S)$ le système obtenu à partir de S en remplaçant les constantes c par les termes t_c. Bien entendu $\mathsf{sk}(S)$ dépend non seulement de S mais aussi de $<_E$ et du choix des symboles de fonction f_c, et du choix d'un ordre sur les arguments des f_c.

Comme seules des constantes sont substituées, on a :

Fait 2.4.13 *Le système S est réduit ssi le système $\mathsf{sk}(S)$ est réduit.*

Proposition 2.4.14

1. *Un système réduit S est unifiable modulo l'ordre $<_E$ si et seulement si le système $\mathsf{sk}(S)$ est unifiable.*
2. *Si le système S est unifiable le système $\mathsf{sk}(S)$ est unifiable.*
3. *Si $\mathsf{sk}(S)$ est unifiable par une substitution s telle que, pour tout c et c', avec $t_c = f_c(x_1, \ldots, x_n)$, $t_{c'} = f_{c'}(x'_1, \ldots, x'_n)$ on a*

$$(f_c = f_{c'} \text{ et } s(x_1) = s(x'_1), \ldots, s(x_n) = s(x'_n)) \Rightarrow c = c'$$

alors S est unifiable.

Démonstration (2.4.14.1). Considérons une étape élémentaire dans la définition de $<_{\mathsf{sk}(S)}$, soit

$$y <_{\mathsf{sk}(S)} x \text{ avec } y \text{ apparaît dans } t \text{ et } (x, t) \in \mathsf{sk}(S)\,.$$

On en déduit que,

$$y <_S x \text{ ou } \exists c(c <_S x \text{ et y apparaît dans } t_c)\,,$$

ce qui entraîne :

$$(y <_S x \text{ ou } \exists c(c <_S x \text{ et } y <_E c)),$$

donc dans les deux cas

$$y <_{SE} x.$$

Par ailleurs $\sim_S = \sim_{\mathsf{sk}(S)}$, donc ce résultat passe à la compatibilité avec $\sim_S$. Il passe aussi à la transitivité et donc, si x et y sont des variables :

$$y <_{\mathsf{sk}(S)} x \Rightarrow y <_{SE} x,$$

et donc

$$(<_{\mathsf{sk}(S)} \text{ a un cycle }) \Rightarrow (<_{SE} \text{ a un cycle}),$$

donc

$$S \text{ unifiable modulo } E \Rightarrow \mathsf{sk}(S) \text{ unifiable.}$$

Réciproquement, si $<_{SE}$ a un cycle, il contient nécessairement une variable x, et peut se décomposer :

$$x = x_0 <_E c_1 <_S x_1 <_E c_2 \cdots <_S x_n = x$$

de $x_i <_E c_{i+1}$, on déduit x_i apparaît dans $t_{c_{i+1}}$, de $c_i <_S x_i$ on déduit donc $x_i <_{\mathsf{sk}(S)} x_{i+1}$, et donc $x <_{\mathsf{sk}(S)} x$. ■

Démonstration (2.4.14.2). La réduction conservant le caractère d'être unifiable et restant correcte par substitution aux constantes c des termes t_c (cf.). Si S est un système réduit $\mathsf{sk}(S)$ est un système réduit. ■

Démonstration (2.4.14.3 : indications). Par induction sur la taille de S. La condition donnée permet d'assurer que S est unifiable dans le cas (c, c'). ■

2.5 Preuve et unification, le théorème de Herbrand.

On utilise maintenant ce qui précède pour la recherche de preuve en calcul des séquents. L'unification ordinaire suffit pour traiter les séquents ne contenant que des formules purement universelles à gauche et purement existentielles à droite. Le cas général se ramène au cas précédent par skolemisation, ou le traite directement en utilisant l'unification modulo un ordre adéquat. Dans chacun des cas on peut donner une formulation du théorème de Herbrand.

On utilisera le théorème d'élimination des coupures pour un calcul des séquents version "montante" où les axiomes et affaiblissements sont restreints aux séquents constitués de formules atomiques, et les contractions explicites aux existentiels à droite, universels à gauche. On utilise même en fait une généralisation du théorème du séquent médian où les règles sur les quantificateurs apparaissent "le plus bas possible" dans la preuve.

On suppose que les variables liées par des quantificateurs distincts sont de noms distincts entre eux et de noms distincts de ceux des variables libres.

On définit une forme de Skolem d'une formule de façon usuelle. On définit la forme de Herbrand de même (analogue à la forme de Skolem en inversant les rôles des quantificateurs existentiels et universels). Une forme de Skolem d'un séquent est obtenu en prenant des formes de Skolem des formules à gauche et des formes de Herbrand des formules à droite.

On va tout d'abord définir une expansion de Herbrand d'une formule, rapidement c'est une formule propositionnelle obtenue par disjonction de copies de la "structure propositionnelle" de la formule originale.

expansion de Herbrand d'une formule.

Dans la recherche de preuve, le comportement d'un quantificateur ne dépend pas seulement de sa nature (existentiel ou universel), mais aussi de la place à laquelle apparaîtra la formule de constructeur principal ce quantificateur dans (à droite ou à gauche dans le séquent) quand la règle qui le concerne s'applique. Rappelons qu'une sous-formule ou un constructeur *apparaît en position positive dans un séquent* Σ, si elle apparaît en position positive dans une formule de la partie droite de Σ, ou en position négative dans une formule de la partie gauche de Σ, et qu'elle *apparaît en position négative* dans les autres cas, i.e. en position négative dans une formule droite de Σ, ou en position positive dans une formule gauche de Σ.

Nous dirons donc qu'un quantificateur est de *valeur universelle dans un séquent* Σ s'il est universel et apparaît en position positive dans Σ, ou s'il est existentiel et apparaît en position négative dans Σ. Nous dirons donc qu'un quantificateur est de *valeur existentielle dans un séquent* Σ dans les autres cas, i.e. s'il est existentiel et apparaît en position positive dans Σ, ou s'il est universel et apparaît en position négative dans Σ. Nous utiliserons le même vocabulaire pour la variable associée, sachant que nous avons pris la précaution de nommer différemment les variables liées différemment. Dans ce qui suit, Q_j désignera un quantificateur, $\forall_j$, resp. $\exists_j$, un quantificateur à valeur universelle, respectivement existentielle.

On associe bijectivement à chaque nom de variable quantifiée à valeur existentielle, une meta-variable de terme, et on appelle celle-ci des *variables existentielles* (ce ne sont pas des variables du langage).

On associe bijectivement à chaque nom de variable quantifiée à valeur universelle, une variable du langage distincte de toutes les variables libres déjà utilisées. On appelle celles-ci *variables universelles* (ce sont des variables du langage). Sous les conditions énoncées plus haut on peut choisir le même nom pour une variable universelle que pour la variable quantifiée associée et c'est ce que l'on fera dans la suite. Remarquons que, dans le cas des variables existentielles, il ne s'agit pas du même type d'objet. Par abus de langage on prendra cependant la même convention.

Du point de vue de l'unification, les variables et constantes du langage considéré (en particulier les variables universelles) sont des constantes, les variables existentielles sont des variables.

Soit Σ un séquent dont les variables quantifiés à valeur universelle sont $c_1, \ldots, c_p$, les variables quantifiées à valeur existentielle sont $x_1, \ldots, x_q$. On suppose de plus que cette énumération respecte l'ordre sur les quantificateurs induit par l'ordre structurel sur les formules : si le quantificateur liant c_j, respectivement x_j, apparaît dans la sous-formule de constructeur le quantificateur liant c_i, respectivement x_i, alors $i < j$ (on peut prendre par exemple l'ordre d'apparition dans le sens de la lecture).

Afin de gérer plusieurs copies de la « matrice propositionnelle » de F, on va attribuer un

index à chaque quantificateur de valeur existentielle, puis indexer les variables et constantes en fonction de cette indexation.

On suppose donc la donnée d'une fonction partielle ι des variables existentielles, dans les entiers, définie pour toutes celles qui apparaissent dans une formule F.

On définit à partir de ι une fonction des sous-formules G de F dans les suites finies d'entiers est la suite des $\iota(x_i)$ pour x_i une variable existentielle de G dans l'ordre de lecture. On la note également ι, en particulier, avec les notations introduites, $\iota(F) = \langle i(x_1), \ldots, i(x_q)\rangle$.

On définit par induction, pour une sous-formule G de F, la formule propositionnelle. G^ι :

1. Si G est atomique, alors $G^\iota = G$;

2. si $G = \neg H$, alors $G^\iota = \neg H^\iota$;

3. si $G = H\ c\ H'$, où c est un connecteur propositionnel binaire, alors $G^\iota = H^\iota\ c\ H^\iota$;

4. si $G = \forall_j c_j\ H$, alors $G^\iota = H[c_i^{\iota(H)}/c_i]^\iota$ (dans ce cas $\iota(H) = \iota(G)$) ;

5. si $G = \exists_j\ x_j\ H$, alors $G^\iota = H[x_j^{\iota(G)}/x_j]^\iota$.

On note $F^{\langle i_1,..,i_p\rangle}$ la formule F^ι, pour ι définie par $\iota(x_j) = i_j$
L'expansion d'ordre $\langle n_1, \ldots, n_p\rangle$ de F est l'ensemble de formules propositionnelles
$\{F^{\langle i_1,\ldots,i_q\rangle}$, pour $0 < i_1 \leq n_1, \ldots, 0 < i_l \leq n_q\}$
L'expansion d'ordre $\langle n_1, \ldots, n_p\rangle$ du séquent $\vdash F$ est le séquent

$$\vdash \{F^{\langle i_1,\ldots,i_p\rangle},\ \text{pour } 0 < i_1 \leq n_1, \ldots, 0 < i_l \leq n_p\}$$

On pourrait étendre cette notation à des séquents quelconques, a partir de maintenant, pour simplifier, on considère un séquent contenant une seule formule F à droite.

Théorème de Herbrand.

On étend l'appellation variables existentielles variables universelles aux variables indexées. Cette indexation sert à gérer les duplications de F. De plus du même coup on a des variables qui ont des noms toujours distincts des variables liées, et donc on peut s'abstraire des problèmes de capture de variable au cours de la reconstruction de la preuve (le problème des captures de variable existentielle n'est par ailleurs pas réellement problématique).

L'ordre structurel sur les formules induit un ordre strict naturel sur la réunion des variables existentielles et universelles dans une expansion, celui de la précédence sur les places des quantificateurs. Précisons. Soit $\overline{\Sigma}$ une expansion du séquent Σ. On pose $x_i^{\langle u\rangle} <_{\overline{\Sigma}} c_j^{\langle v\rangle}$ si $\langle u\rangle$ est une suite préfixe (au sens large) de $\langle v\rangle$ (ce qui a pour conséquence que le quantificateur $\forall_j c_i$ apparaît dans la sous-formule de constructeur principal $\exists_i x_i$).

On s'occupe des substitutions *idempotentes* définies sur les variables existentielles, à valeur dans l'ensemble des termes sur les variables union un ensemble de variables arbitraires (dont on a besoin pour reconstituer la substitution à partir de la preuve), les variables universelles sont donc bien considérées comme des constantes, du point de vue de la substitution.

On peut énoncer maintenant le théorème de Herbrand pour des formules quelconques.

Proposition 2.5.1 *Un séquent Σ est prouvable si et seulement si il existe une expansion $\overline{\Sigma}$ de Σ et une substitution s définie sur les variables existentielles à valeurs dans les termes, et compatible avec la relation $<_{\overline{\Sigma}}$ (cf 2.4.6 page 90) telle que*

$$s(\overline{\Sigma})\ \textit{est prouvable.}$$

On peut en donner une version plus précise en calcul des séquents.

Proposition 2.5.2 (HERBRAND – calcul des séquents) *Un séquent Σ est prouvable si et seulement s'il existe une expansion $\overline{\Sigma}$ d e Σ, et un ensemble complet de connections de $\overline{\Sigma}$ qui soit unifiable modulo la relation $<_{\overline{\Sigma}}$.*

Donnons maintenant, un énoncé du théorème de Herbrand plus usuel (avec skolemisation).

Proposition 2.5.3 (HERBRAND – Skolemisation) *Un séquent Σ est prouvable si et seulement s'il existe une expansion $\overline{\mathsf{sk}(\Sigma)}$ de la forme de skolem $\mathsf{sk}(\Sigma)$ de Σ, et un ensemble complet de connections de $\overline{\mathsf{sk}(\Sigma)}$ qui soit unifiable.*

Démonstration (Herbrand – calcul des séquents). S'il existe une preuve, on a une expansion, un ensemble complet de connections, une substitution s qui unifie ces connections, on suppose modulo renommage de variables, que c'est une substitution idempotente. A chaque règle existentielle on associe à la variable considérée son image par s. S'il y a un cycle :

$$c = c_1 <_s x_1 <_{\overline{\Sigma}} c_2 \cdots <_s x_n <_{\overline{\Sigma}} c_n = c$$

La règle sur x_i est en dessous de la règle sur c_{i+1}, car l'ordre des règles doit respecter celui des sous-formules. La règle sur c_i est en dessous de la règle sur x_i, en effet sinon, c_i serait libre dans le contexte lors de la règle sur c_i. Il y a une contradiction.

Réciproquement. S'il existe une telle expansion, un tel ensemble complet de connections une telle substitution, supposée idempotente, l'expansion associée de F est donc prouvable. On passe de F à son expansion, en remontant en appliquant n_i contractions dès apparition du quantificateur $\exists x_i$, en choisissant à chaque étape une variable ou constante minimale pour l'ordre $<_{s\Sigma}$ parmi celles restantes. Les règles sur les quantificateurs universels sont donc bien correctes. Il est possible de procéder ainsi car l'ordre des sous-formules et compatible avec l'ordre $<_{s\overline{\Sigma}}$. Si α est avant β, $\beta <_{s\overline{\Sigma}} \alpha$ est impossible. Distinguons les 4 cas possibles pour le couple (α, β).
(variable, constante), par définition de $<_{\overline{\Sigma}}$, $<_{s\overline{\Sigma}}$ aurait un cycle.
(variable,variable), supposons que x est avant y et $y <_{s\overline{\Sigma}} x$, alors il existe une variable universelle c tel que $y <_{\overline{\Sigma}} c <_{s\overline{\Sigma}} x$, mais comme x est avant y , x est avant c, ce qui est impossible (cas précédent).
(constante,variable), supposons que x est avant c et $c <_{s\overline{\Sigma}} x$, alors, il existe une variable existentielle y tel que $c <_{\overline{\Sigma}} y <_{s\overline{\Sigma}} x$. De x avant c on déduit x avant y, et on est ramené au cas précédent.
(constante,constante), supposons d avant c, et $c <_{s\overline{\Sigma}} d$, alors il existe x tel que $c <_{s\overline{\Sigma}} x <_{\overline{\Sigma}} d$, de d avant c on déduit x avant c, et on est ramené au cas précédent. ■

Démonstration (HERBRAND – skolemisation : indications). On se ramène au résultat précédent.
Supposons le séquent Σ prouvable. L'ensemble des variables et constantes est donné par l'expansion prouvable. Il suffit de montrer que la skolémisation est une skolémisation du système de termes associé aux connections de $\overline{\Sigma}$ pour $<_{\overline{\Sigma}}$. La clause 2 de la proposition 2.4.14 page 92.
Supposons maintenant que $\overline{\mathsf{sk}(\Sigma)}$ est prouvable.

Soit s la substitution donnée par la preuve du skolemisé. On construit l'index de l'expansion en prenant à chaque fois le cardinal de l'ensemble des classes d'équivalences pour la substitution s, dans les variables issue du même quantificateur. On construit l'expansion correspondante, en particulier, vu la définition de la skolemisation et de l'expansion du séquent :

$$(f_c = f_{c'} \text{ et } x_1 \sim_s x'_1, \ldots, x_n \sim_S x'_n) \Rightarrow c = c'$$

ce qui permet de conclure grâce à la clause 3 de la proposition 2.4.14 page 92.

2.5.1 Méthode des connexions dans le cas général.

Pour adapter la méthode utilisée pour les séquents de Herbrand au § 2.3.3 page 87, il suffit de remplacer l'unification par l'unification modulo un ordre comme indiqué au § précédent.
Exemple. On veut montrer $\vdash \exists x(Bx \to \forall c Bc)$. Pour une multiplicité de 1, la matrice est :

$$B?x^- \; Bc^+ \quad \text{conditions } :?x < c$$

Il y a évidemment un ensemble complet de connexions, il faut unifier par $\sigma(?x) = c$ ce qui n'est pas possible car alors $c <_\sigma ?x$, ce qui crée un cycle. Pour une multiplicité de 2 la matrice est :

$$B?x_1^- \; \overset{\frown}{Bc_1^+} \; B?x_2^- \; Bc_2^+ \quad \text{conditions : } ?x_1 < c_1, \; ?x_2 < c_2$$

L'ensemble complet de connexions (réduit à une connexion) indiqué est unifiable modulo l'ordre indiqué par $\sigma(?x_2) = c_1$, qui induit $c_1 <_\sigma ?x_2$.

[*à compléter*]

Bibliographie

[Bibel 82] W. Bibel, *Computationally improved versions of Herbrand's Theorem*, Proceedings of the Herbrand Symposium, Logic Colloquium 81, pp 11-28, North-Holland (1982).

[Girard 87] J.Y. Girard, *Proof Theory and Logical Complexity*, Bibliopolis (1987).

[Girard-Lafont-Taylor 89] J.Y. Girard – Yves Lafont – Paul Taylor, *Proofs and types*, Cambridge University Press (1989).

[Cori-Lascar 88] René Cori – Daniel Lascar, *Logique mathématique* Tomes I et II, Masson (88).

[Martelli Montanari 82] A. Martelli et U. Montanari, *An efficient Unification Algorithm*, ACM Transactions on Programming Languages and Systems, Vol. 4, No. 2, pp 252-282, (April 1982).

[Martelli Rossi 84] A. Martelli et G. Rossi, *Efficient Unification with Infinite Terms in Logic Programming*, Proceedings of the International Conference on Fifth Generation Computer Systems 1984, pp 202-209, (ICOT 1984).

[Wallen 90] L. Wallen, *Automated Deduction in Nonclassical Logics*, MIT Press 1990.

[*à compléter*]

Table des matières