Calul Propositionnel
Résumé de cours d'après les transparents

IUP MIAGE, Université de Nantes
(C. Retoré `http://perso.wanadoo.fr/retore/christian`)

6 octobre 2001

Table des matières

1 Présentation

1.1 Pourquoi un cours de logique?

Nombreuses utilisations de la logique en informatique

- intérêt immédiat en algo: écrire des conditionnelles et des conditions d'arrêt correctes . . .
- conception de circuits (portes "et" "ou" "non" réalisant une fonction de $\{0,1\}^n$ dans $\{0,1\}^p$).
- preuves de programmes
  montrer qu'un programme satisfait une propriété (formule logique)
  Exemple: fonction de recherche d'un entier dans un tableau trié qui rend "présent" ou "absent".
  Correction du programme:
  hypothèse: le tableau en entrée est trié
  conclusion: on a "présent" si et seulement si la valeur est dans le tableau, et "absent" sinon.
- bases de données
  On utilise des propriétés logiques, par exemple, si les prédicats de base sont "être le grand-père de", "être l'oncle de", etc. :
  chercher "un garçon de 9 ans ayant un oncle de 32 ans et un grand père paternel de 70 ans"
  revient à chercher "un homme de 32 ans ayant un père de 70 ans et un neveu de 9 ans"
  La logique permet de produire toutes ces formulations équivalentes, étant donné les équivalences de base "x est le neveu de y"$\equiv$ "y est l'oncle de x". Certaines formulations sont plus faciles à chercher que d'autres.
- CaML (programmation fonctionnelle typée)
  type = formule logique
  programme = démonstration formelle
  évalutation = normalisation de la démonstration
- ProLog (programmation logique)
  description du problème = axiomes logiques
  résolution du problème = construction d'une démonstration

1.2 La logique: philosophie, mathématique et informatique

Historiquement: dégager les principes du raisonnement correct.
Antiquité et Moyen-Age (scholastique) syllogismes, quantification, logique modale
Leibniz, Boole, de Morgan (remplacer la déduction par un calcul)
20^e mathématisation de la logique
(1) théorie des ensembles
(2) théorie des modèles
(3) théorie de la démonstration
(4) théorie de la récursion, calculabilité et complexité
(3) et (4) mathématiques pour l'informatique

1.3 Exemples de syllogismes

- Modus ponens:
  De "si A alors B"
  et "A"
  on déduit "B".
- Modus tollens:
  De "si A alors B"
  et "non B"
  on déduit "non A"

1.4 Une notion clef: syntaxe/sémantique

SYNTAXE $\xrightarrow{\text{interprétation}}$ SEMANTIQUE

Enoncés $\xrightarrow{\text{interprétation}}$ valeurs de vérité
dans un monde possible

La logique s'occupe de la vérité indépendamment du contenu des énoncés.
Par exemple
"si (il pleut) alors (la chaussée est mouillée)"
peut être vrai ou faux (si la chaussée est celle d'un tunel). Par contre
"si (il pleut) alors ((il pleut) ou (il neige))"
sera vrai quelles que soient les circonstances.

Coté syntaxe: démonstrations formelles qui produisent exactement les énoncés qui sont toujours vrais.

1.5 Logique classique

Seulement deux valeurs de vérité (vrai: 1 et faux: 0).

Dans une interprétation toute proposition devient vraie ou fausse.

C'est abusif: "123 est grand" n'est ni vrai ni faux; mais cette vision simpliste suffit pour bon nombre d'applications.

Sinon il existe des logiques multivalués, modales, temporelles etc.

2 Le langage du calcul des propositions

2.1 Langage du calcul des propositions: définition

Les propositions élémentaires (ou atomiques) sont des phrases simples dont on peut déterminer dans un contexte (ou interprétation) si elles sont vraies ou fausses.

Pour nous les propositions élementaires ou atomiques seront des lettres: p,q, ... On supposera qu'on en a un ensemble fini ou dénombrable de propositions atomiques.

Le vocabulaire varie:

Sont synonymes:
proposition atomique
proposition élementaire
formule atomique (du calcul propositionnel)
variable propositionnelle

Sont synonymes:
formule (du calcul propositionnel)
proposition (complexe)

Les propositions ou formules sont définies ainsi:

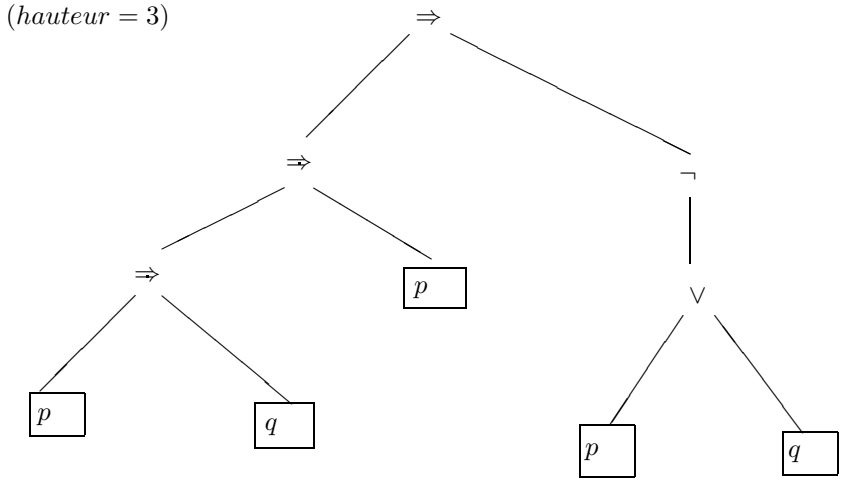
1. Les propositions atomiques sont des propositions.
2. Si X est une proposition, alors $(\neg X)$ ("non X", la négation de X) est une proposition.
3. Si X et Y sont des propositions, alors
   (a) $(X \wedge Y)$ ("X et Y") conjonction
   (b) $(X \vee Y)$ ("X ou Y", ou inclusif) disjonction
   (c) $(X \Rightarrow Y)$ ("si X alors Y") implication
   (d) $(X \Leftrightarrow Y)$ ("X si et seulement si Y") équivalence

   sont des propositions.
4. **Rien d'autre n'est une proposition.**

2.2 Les formules: arbres et expressions parenthésées

Contre-exemples: $p \vee q \wedge r$, $p \Rightarrow qr$

Exemple: $(((p \Rightarrow q) \Rightarrow p) \Rightarrow (\neg(p \vee q)))$



2.3 Sous-formules

Les sous-formules immédiates de $(X \wedge Y)$, $(X \vee Y)$, $(X \Rightarrow Y)$ et $(X \Leftrightarrow Y)$ sont X et Y.

La seule sous-formule immédiate de $(\neg X)$ est X.

Une formule Y est sous formule d'une formule X

si $Y = X$ ou

s'il existe un entier n et des formules

$$Y = Y_0, Y_1, \ldots, Y_n = X$$

telle que Y_i soit sous-formule immédiate de Y_{i+1}.

On peut aussi définir inductivement l'ensemble $SF(A)$ des sous-formules d'une formule A:

– Si $A = p$ une proposition atomique:

$$SF[A] = SF[p] = \{p\}$$

– Si $A = (\neg X)$ alors

$$SF[A] = SF[(\neg X)] = \{A\} \cup SF[X]$$

– Si $A = (X \bullet Y)$ où $\bullet$ peut être n'importe quel connecteur binaire ($\vee$,$\wedge$,$\Rightarrow$,$\Leftrightarrow$) les sous-formules de A sont:

$$SF[A] = SF[(X \bullet Y)] = \{(X \bullet Y)\} \cup SF[X] \cup SF[Y]$$

Les sous-formules d'une formule donnée s'obtiennent en choisissant un noeud dans l'arbre, et en prenant l'arbre en dessous.

C'est aussi une formule allant d'une parenthèse ouvrante à la fermante qui lui correspond, ou une proposition atomique de la formule considérée. Exemple: quelles sont les sous-formules de

$$(((p \Rightarrow q) \Rightarrow p) \Rightarrow (\neg(p \vee q)))$$

$(((p \Rightarrow q) \Rightarrow p) \Rightarrow (\neg(p \vee q)))$
$(p \vee q)$
$(\neg(p \vee q))$
$((p \Rightarrow q) \Rightarrow p)$
$(p \Rightarrow q)$
p (trois occurrences)
q (deux occurrences)

3 Sémantique du calcul des propositions

3.1 Valuation (ou interprétation)

Une valuation c'est la donnée, pour chaque proposition atomique de sa valeur de vérité.

Supposons connue la valeur de chaque proposition atomique.

Une telle valuation s'étend de manière unique à toutes les formules: en effet la valeur de vérité d'une formule complexe n'est fonction que de ses sous-formules (ce qui est une idéalisation). Elle se calcule au moyen des tables de vérité, qui permettent de calculer de proche en proche la valeur de la formule, en suivant les étapes de construction de la formule, c'est-à-dire en calculant la valeur des sous-formules.

3.2 Valuation: tables de vérité

X	Y	$(X \wedge Y)$
1	1	1
1	0	0
0	1	0
0	0	0

X	Y	$(X \vee Y)$
1	1	1
1	0	1
0	1	1
0	0	0

X	$(\neg X)$
1	0
0	1

X	Y	$(X \Rightarrow Y)$
1	1	1
1	0	0
0	1	1
0	0	1

X	Y	$(X \Leftrightarrow Y)$
1	1	1
1	0	0
0	1	0
0	0	1

3.3 Commentaires sur le "sens" des connecteurs

Le "et" noté $(X \wedge Y)$ est commutatif. Ce n'est pas toujours le cas en français: "il se gare et (il) descend de voiture"

Le "ou" noté $X \vee Y$ est inclusif.

Exemple: les étudiant titulaires du DEUG ou qui ont réussit un concours d'entrée peuvent s'inscirre en licence. (on ne va pas refuser ceux qui ont les deux titres!).

Contre-exemple: en français le "ou" peut aussi être exclusif, comme dans "fromage ou dessert".

L'implication $X \Rightarrow Y$ signifie seulement que quand X est vrai, Y l'est aussi, c'est une abbréviation pour

$$(\neg X) \vee Y$$

Attention: du point de vue formel étudié ici il n'y a pas de causalité. Par exemple

"si mon fils embête sa soeur, alors il est privé de foot"

est vrai même si

"je le prive de foot" alors qu' "il n'embête pas sa soeur".

Souvent en français l'implication est en fait une équivalence: dans l'exemple ci-dessus mon fils croira que si "il n'embête pas sa soeur", alors "il n'est pas privé de foot".

"si éléphants volent, alors Paris est en France" et "si les mouettes volent, Paris est en France" sont valides.

L'équivalence $X \Leftrightarrow Y$ est juste une abbréviation pour $((X \Rightarrow Y) \wedge (Y \Rightarrow X))$, c'est à dire

$$((X \wedge Y) \vee ((\neg X) \wedge (\neg Y)))$$

Il n'y a pas non plus de relation de cause à effet entre le fait que X et Y soient toujours vrais simultannément, par exemple:

"(Paris est en France) si et seulement si (les mouettes volent)"

"(les éléphants volent) si et seulement si (les poules ont des dents)"

sont toutes deux valides.

3.4 Les pièges de l'interprétation standard

Le logique se contente de dire:

une formule est toujours vraie (par ex. $(p \Rightarrow p)$)

une formule est toujours fausse (par ex. $(q \wedge (\neg q))$)

ou qu'il existe des valuations qui la rendent vraie et d'autres qui la rendent fausse (par ex. $p \Rightarrow q$).

Lorsqu'on connaît le sens des propositions atomiques, on est tenté d'en dire plus, c'est-à-dire d'admettre implicitement la vérité d'autres formules que celles qu'on considère. C'est à éviter.

3.5 Formule (in)consistante et universellement valide

Une formule est *consistante* (satisfiable) s'il existe une valuation qui la rende vraie.

Un formule est *inconsistante* si elle est fausse pour toute valuation.

Une formule est *universellement valide* si elle est vraie pour toute valuation. Elle est alors appelée une tautologie ou un théorème du calcul des propositions.

3.6 Ensembles de formules (in)consistant

Un ensemble de formules $\mathcal{E}$ se traite comme la conjonction de toutes les formules le composant.

Différence: $\mathcal{E}$ peut être infini, tandis qu'une formule ne comporte qu'un nombre fini de conjonction.

Un ensemble de formules $\mathcal{E}$ est dit *consistant* s'il existe une valuation qui rende *simultanément* vraies toutes les formules de l'ensemble.

L'ensemble de formules $\mathcal{E}$ est dit *inconsistant* si pour toute valuation au moins une des formule est fausse.

Si $\mathcal{E}$ fini, la consistance (resp. l'inconsistance) de $\mathcal{E}$ est identique à la consistance (resp. l'inconsistance) de la conjonction des formules de $\mathcal{E}$.

3.7 Comment énumérer les valuations

Pour voir si une formule F est consistante, inconsistante, ou universellement valide, il faut lister toutes les valuations et calculer à chaque fois la valeur de F.

Remarque importante: *la valeur de F ne dépend que des propositions atomiques figurant dans F* Même si les propositions atomiques forment un ensemble infini, seul un nombre *fini* figurent dans F. Il suffit donc de décrire toutes les valuations sur les

propositions atomiques figurant dans F, et s'il y a n propositions atomiques dans F cela fait 2^n valuations à considérer.

Exemple: $F = ((p \Rightarrow q) \Rightarrow p) \Rightarrow r$

p peut valoir 0 ou 1, dans chacun de ces deux cas, q peut valoir 0 ou 1, et dans chacun de ces quatre cas r peut valoir 0 ou 1. On dresse un arbre ou un tableau de tous les cas possibles ($8 = 2^3$) . Á l'aide des tables de vérité on calcule la valeur des sous-formules de F dans chaque cas.

3.8 Implication et équivalence sémantique

Une formule X en implique une autre Y si chaque fois qu'une valuation rend X vraie, Y prend aussi la valeur vraie. Dans ce cas, la formule $(X \Rightarrow Y)$ est une tautologie.

Une formule X est équivalente à une autre Y si et seulement si, pour toute valuation X et Y prennent la même valeur de vérité. Dans ce cas, la formule $(X \Leftrightarrow Y)$ est une tautologie.

Une formule X est dite *conséquence d'un ensemble de formules* $\mathcal{E}$, si toute valuation qui rend simultanément vraies toutes les formules de $\mathcal{E}$ rend vraie X.

Exemple $X = r$ est conséquence de

$$\mathcal{E} = \{p,(p \Rightarrow q),(q \Rightarrow r)\}$$

4 L'algorithme de Quine

4.1 Justification et plan du cours sur cet algorithme

Du point de vue informatique il importe non seulement de comprendre les notions de formule d'interprétation etc. mais aussi de décrire des algorithmes qui permettent de vérifier efficacement si une formule est consistante, inconsistante ou uniformément valide — par exemple pour vérifier par machine la correction d'un programme, ou d'un protocole de communication sécurisé.

Pour écrire de tels algorithmes il faut que les techniques utilisées soit suffisamment précises, prêtes à être implantées, c.-à-d.formaliser et faire appel à des opérations très simples sur les formules.

Pour présenter un algorithme assez efficace de décision, nous allons procéder comme suit, les trois premiers points étant des ingrédients de l'algorithme:

- introduction des constantes logiques $\top$ et $\bot$.
- substitution d'une formule à une proposition atomique et équivalence sémantique
- remplacement d'une sous-formule par une sous formule équivalente et équivalence sémantique
- l'algorithme de Quine / exemple
- preuve de la terminaison de l'algorithme
- preuve de la correction de l'algorithme

4.2 Un exemple justifiant la méthode

Le raisonnement que voici est-il correct?

S'il est venu $s$eul, il a pris le $b$us ou le $t$rain. S'il a pris le $b$us ou son $a$utomobile alors il est arrivé en $r$etard et a $m$anqué la réunion. Il n'est pas arrivé en $r$etard.

Donc s'il est venu $s$eul il a pris le $t$rain.

Cet énoncé se formalise ainsi:

s: il est venu $s$eul
b: il a pris le $b$us
t: il a pris le $t$rain
a: il a pris son $a$utomobile
r: il est arrivé en $r$etard
m: il a $m$anqué la réunion

$$\Big(\big(s \Rightarrow (b \vee t)\big) \wedge \big((b \vee a) \Rightarrow (r \wedge m)\big) \wedge \neg r\Big) \Rightarrow (s \Rightarrow t)$$

Il faut une table de vérité, de $2^6 = 64$ lignes et 14 colonnes!

4.3 Les constantes $\top$ et $\bot$

On notera désormais $F \equiv G$ l'équivalence sémantique de F et G: pour toute valuation v, $v(F) = v(G)$.

On adjoint au langage deux constantes spéciales $\top$ et $\bot$ dont la particularité est la suivante: pour toute valuation, $v(\bot) = 0$ et $v(\top) = 1$.

Les formules sont définies comme précédemment, en ajoutant la clause: une constante est une proposition.

Ainsi par exemple $(p \Rightarrow \top)$, $q \vee (p \vee \top)$ etc. sont des formules.

D'après la définition de $\equiv$ on a:

Propriété 1 *F est universellement valide*
si et seulement si $F \equiv \top$
F est inconsistante si et seulement si $F \equiv \bot$.

4.4 Substitution d'une formule à une proposition atomique

Soit p une proposition atomique, autre que $\top$ ou $\bot$.

Soit F et X deux formules. La formule $F[X/p]$ est s'obtient en remplaçant *toutes* les occurrences de p par X.

Exemple:

$$((p \wedge \top) \vee (p \Rightarrow q))[(p \Rightarrow q)/p]$$
$$= (((p \Rightarrow q) \wedge \top) \vee ((p \Rightarrow q) \Rightarrow q))$$

Plus formellement:

- F est une constante ($\top$ ou $\bot$) alors $F[X/p] = F$.
- F est une proposition atomique.
  - Si $F = q \neq p$ alors $F[X/p] = F$
  - Si $F = p$ alors $F[X/p] = X$
- Si $F = (\neg G)$ alors $F[X/p] = (\neg G[X/p])$.
- Si $F = (G \bullet H)$
  alors $F[X/p] = (G[X/p] \bullet H[X/p])$

$\bullet$ désigne n'importe quel connecteur binaire
$\vee, \wedge, \Rightarrow, \Leftrightarrow$

4.4.1 Substitution simultanée de formules à des propositions atomiques

On peut aussi effectuer simultanément la substitution de X à p, de Y à q etc. ce qui est noté $F[X/p; Y/q]$. La définition inductive est la même.

Exemple: $((p \wedge \top) \vee (p \Rightarrow q))[(p \Rightarrow q)/p; p/q] = (((p \Rightarrow q) \wedge \top) \vee ((p \Rightarrow q) \Rightarrow p))$

Attention: en général $F[X/p][Y/q] \neq F[X/p; Y/q]$. En effet X et Y peuvent contenir les propositions atomiques p et q.

Exemple: $(p \Rightarrow q)[q/p][p/q] = (p \Rightarrow p)$
et $(p \Rightarrow q)[q/p; p/q] = (q \Rightarrow p)$.

Si a et b sont des propositions atomiques *ne figurant ni dans F ni dans X ni dans Y* alors on a

$$F[X/p; Y/q] = F[a/p][b/q][X/a][Y/b]$$

On traiterait de même le cas de n substitutions simultanées.

4.4.2 Propriétés de la substitution

[Expliquées mais non démontrées - exercice possible]

Propriété 2 *Si F est universellement valide ($F \equiv \top$) alors pour toute proposition atomique p et toute formule X on a $F[X/p] \equiv \top$ c.-à-d.$F[X/p]$ est universellement valide.*

Si F est inconsistante ($F \equiv \bot$) alors pour toute proposition atomique p et toute formule X on a $F[X/p] \equiv \bot$ c.-à-d.$F[X/p]$ est inconsistante.

Propriété 3 *Pour toute proposition atomique p on a:*
($F[\top/p] \equiv \top$ et $F[\bot/p] \equiv \top$) si et seulement si $F \equiv \top$.
($F[\top/p] \equiv \bot$ et $F[\bot/p] \equiv \bot$) si et seulement si $F \equiv \bot$.
F est consistante si et seulement si l'une au moins des deux formules $F[\top/p]$ et $F[\bot/p]$ est consistante.

4.5 Remplacement d'une occurrence d'une sous formule par une formule équivalente

Attention: c'est différent de la substitution.
On note $\equiv$ l'équivalence sémantique entre sous-formules.
On note $\bullet$ un connecteur binaire quelconque, c.-à-d.$\wedge, \vee, \Leftrightarrow, \Rightarrow$.
$F[X := X']$ désigne la formule obtenue à partir de F en remplaçant l'occurrence de X par la formule X'.

Propriété 4 *Soit F une formule contenant une occurrence d'une sous formule X.*
Soit X' une formule équivalente à X, c.-à-d.$X \equiv X'$.
Alors on a $F[X := X'] \equiv X$.

La propriété 4 utilise les deux suivantes:

Propriété 5 *Si $X \equiv X'$ alors $(\neg X) \equiv (\neg X')$ et pour toute formule Z, $(X \bullet Z) \equiv (X' \bullet Z)$ et $(Z \bullet X) \equiv (Z \bullet X')$*

Propriété 6 *Si l'occurence de X dans F est une occurrence de X dans G, alors*
- *si $F = (\neg G)$,*
  on a $F[X := X'] \equiv (\neg G[X := X'])$
- *si $F = (G \bullet L)$,*
  on a $F[X := X'] = (G[X := X') \bullet L$
- *si $F = (L \bullet G)$,*
  on a $F[X := X'] = (L \bullet G[X := X'])$

Récurrence sur le nombre c de connecteurs dans F mais pas dans X.
Remarquons pour commencer que si $c = 0$, c.-à-d.$F = X$ alors $F[X := X'] = X'$ et on a bien $X' = F[X := X'] \equiv F = X$.
On peut donc supposer dans la suite que $F[X]$ n'est pas X.

- Si $F = (\neg G)$, comme X n'est pas F (déjà vu), on peut supposer que X est une sous-formule de G. G ayant unconnecteur de moins que F en dehors de X on a $G[X := X'] \Leftrightarrow G$ par hypothèse de récurrence. En vertu de 5 on a $(\neg G[X := X']) \equiv (\neg G)$, c'est-à-dire $F[X := X'] \equiv F$ en vertu de 6.
- Si $F = (G \bullet H)$, comme X n'est pas F (déjà vu), X est une occurrence de la formule X dans G ou dans H.
  - Si X est une sous-formule de G, comme G a moins de connecteurs que F en dehors de X, on a $G[X := X'] \equiv G$ par hypothèse de récurrence. En vertu de 5 on a $(G[X := X'] \bullet H) \equiv (G \bullet H)$, d'où $F[X := X'] \equiv F$ en vertu de 6.
  - Si X est une sous-formule de H, comme H a moins de connecteurs que F, on a $H[X := X'] \Leftrightarrow H$ par hypothèse de récurrence. En vertu de 5 on a $(G \bullet H[X := X']) \equiv (G \bullet H)$, d'où $F[X := X'] \equiv F$ en vertu de 6.

4.6 Quelques équivalences utiles

entre connecteurs $(X \Rightarrow Y) \equiv ((\neg X) \vee Y)$
$(X \Leftrightarrow Y) \equiv ((X \Rightarrow Y) \wedge (Y \Rightarrow X))$

associativité $(X \wedge (Y \wedge Z)) \equiv ((X \wedge Y) \wedge Z)$
$(X \vee (Y \vee Z)) \equiv ((X \vee Y) \vee Z)$

commutativité $(X \vee Y) \equiv (Y \vee X)$
$(X \wedge Y) \equiv (Y \wedge X)$

distributivité $(X \wedge (Y \vee Z)) \equiv ((X \wedge Y) \vee (X \wedge Z))$
$(X \vee (Y \wedge Z)) \equiv ((X \vee Y) \wedge (X \vee Z))$

Lois de de Morgan $(\neg(\neg X)) \equiv X$
$(\neg(X \wedge Y)) \equiv ((\neg X) \vee (\neg Y))$.
$(\neg(X \vee Y)) \equiv ((\neg X) \wedge (\neg Y))$.

Règles de simplification de Quine

$(\neg\bot)$	$\equiv$	$\top$	$(\neg\top)$	$\equiv$	$\bot$
$(\top \Rightarrow X)$	$\equiv$	X	$(X \Rightarrow \top)$	$\equiv$	$\top$
$(\bot \Rightarrow X)$	$\equiv$	$\top$	$(X \Rightarrow \bot)$	$\equiv$	$(\neg X)$
$(\top \vee X)$	$\equiv$	$\top$	$(\bot \vee X)$	$\equiv$	X
$(X \vee \top)$	$\equiv$	$\top$	$(X \vee \bot)$	$\equiv$	X
$(X \wedge \top)$	$\equiv$	X	$(\bot \wedge X)$	$\equiv$	$\bot$
$(\top \wedge X)$	$\equiv$	X	$(X \wedge \bot)$	$\equiv$	$\bot$
$(\top \Leftrightarrow X)$	$\equiv$	X	$(\bot \Leftrightarrow X)$	$\equiv$	$(\neg X)$
$(X \Leftrightarrow \top)$	$\equiv$	X	$(X \Leftrightarrow \bot)$	$\equiv$	$(\neg X)$

On remarque que le second membre de l'équivalence a toujours une taille plus petite que le premier [taille = (nombre de connecteurs) + (nombre de $\top$ et $\bot$)].

Ces équivalences sont la base l'algorithme qui suit — avec les prop. 3 et 4.

4.7 L'algorithme de Quine

L'algorithme de Quine permet de vérifier rapidement si une formule est universellement valide, inconsistante, ou consistante en construisant un arbre dont les noeuds sont des formules.

Algorithme 7 (Quine) *1. Initialisation: l'arbre réduit à la formule à vérifier.*

2. *Si l'un des schémas de simplification s'applique à l'une des feuilles, adjoindre à l'arbre une feuille dont la formule est la formule obtenue par simplification, et recommencer 2.*
3. *Si l'une des feuilles contient une proposition atomique autre que $\top$ et $\bot$ en choisir une (par exemple la plus fréquente), disons p, et dessous chaque feuille F_i rajouter deux feuilles $F_i[\bot/p]$ et $F_i[\top/p]$, et retourner en 2.*

4.7.1 Terminaison de l'algorithme de Quine

L'algorithme termine.

On ne peut boucler sur 2 car la taille des formules inscrites sur les feuilles diminue à chaque simplification.

Chaque fois que l'on passe en 3 le nombre de propositions atomiques présentes dans au moins une feuille diminue.

Lorsque l'algorithme est fini, toutes les feuilles sont $\top$ ou $\bot$. En effet, il n'y a plus de propositions atomiques autres que $\top$ et $\bot$, (sinon 3 s'appliquerait) et toutes les fomules écrites à partir de $\top$ et $\bot$ se simplifient en $\top$ ou $\bot$ par les règles de simplification.

4.7.2 Correction de l'algorithme de Quine

Soit $\mathcal{A}$ un arbre de Quine terminé, dont la racine est la formule F.

1. toutes les feuilles de $\mathcal{A}$ sont $\top$, si et seulement si F est universellement valide,
2. toutes les feuilles de $\mathcal{A}$ sont $\bot$, si et seulement si F est inconsistante
3. une feuille de $\mathcal{A}$ au moins est $\top$ si et seulement si F est consistante.

On montrera cela au paragraph 4.9 après avoir traité un exemple.

4.8 Exemple

Soit F la formule du pargraphe 4.2.

$\Big(\big(s \Rightarrow (b \vee t)\big) \wedge \big((b \vee a) \Rightarrow (r \wedge m)\big) \wedge \neg r\Big) \Rightarrow (s \Rightarrow t)$

Pas de règle de simplification possible: bifurcation par ex. sur s

$F[\top/s] =$

$\Big(\big(\top \Rightarrow (b \vee t)\big) \wedge \big((b \vee a) \Rightarrow (r \wedge m)\big) \wedge \neg r\Big) \qquad \Rightarrow (\top \Rightarrow t)$

$F[\bot/s] =$

$\Big(\big(\bot \Rightarrow (b \vee t)\big) \wedge \big((b \vee a) \Rightarrow (r \wedge m)\big) \wedge \neg r\Big) \qquad \Rightarrow (\bot \Rightarrow t)$

simplification des feuilles:

$F[\top/s][\top \Rightarrow (b \vee t) := (b \vee t] =$

$\Big((b \vee t) \wedge \big((b \vee a) \Rightarrow (r \wedge m)\big) \wedge \neg r\Big) \Rightarrow t \quad = G$

(plus de simplification possible)

$F[\bot/s][(\bot \Rightarrow t) := \top] =$

$\Big(\big(\bot \Rightarrow (b \vee t)\big) \wedge \big((b \vee a) \Rightarrow (r \wedge m)\big) \wedge \neg r\Big) \Rightarrow \top$

encore une simplification $[\ldots \Rightarrow \top := \top]$ done:

$F[\bot/s][(\bot \Rightarrow t) := \top][\ldots \Rightarrow \top := \top] = \top$

La branche $F[\bot/s]$ est finie.

bifurcation par ex. sur b:

$G[\top/b] =$

$\Big((\top \vee t) \wedge \big((\top \vee a) \Rightarrow (r \wedge m)\big) \wedge \neg r\Big) \Rightarrow (\top \Rightarrow t)$

$G[\bot/b] =$

$\Big((\bot \vee t) \wedge \big((\bot \vee a) \Rightarrow (r \wedge m)\big) \wedge \neg r\Big) \Rightarrow (\top \Rightarrow t)$

simplification des feuilles:

$G[\top/b] \rightarrow$

$\Big(\top \wedge \big(\top \Rightarrow (r \wedge m)\big) \wedge \neg r\Big) \Rightarrow t$

$\overrightarrow{\Big((r \wedge m) \wedge \neg r\Big) \Rightarrow t \quad = H}$

$G[\bot/b] \rightarrow$

$\Big(t \wedge \big(a \Rightarrow (r \wedge m)\big) \wedge \neg r\Big) \Rightarrow t \quad = K$

bifurcation des deux feuilles suivant r
donc quatre feuilles ensuite

$H[\top/r] = \Big((\top \wedge m) \wedge \neg\top\Big) \Rightarrow t$

$H[\bot/r] = \Big((\bot \wedge m) \wedge \neg\bot\Big) \Rightarrow t$

$K[\top/r] = \Big(t \wedge \big(a \Rightarrow (\top \wedge m)\big) \wedge \neg\top\Big) \Rightarrow t$

$K[\bot/r] = \Big(t \wedge \big(a \Rightarrow (\bot \wedge m)\big) \wedge \neg\bot\Big) \Rightarrow t$

simplification des quatre feuilles:

$H[\top/r] \rightarrow (m \wedge \bot) \Rightarrow t \rightarrow \bot \Rightarrow t \rightarrow \top$

branche terminée.

$H[\bot/r] \rightarrow \Big(\bot \wedge \top\Big) \Rightarrow t \rightarrow \bot \Rightarrow t \rightarrow \top$

branche terminée

$K[\top/r] \rightarrow \Big(t \wedge \big(\top\big) \wedge \bot\Big) \Rightarrow t$
$\rightarrow \bot \Rightarrow t$
$\rightarrow \top$

branche terminée

$K[\bot/r] \rightarrow \Big(t \wedge \big(a \Rightarrow \bot\big) \wedge \top\Big) \Rightarrow t$
$\rightarrow \Big(t \wedge \big(\neg a\big)\Big) \Rightarrow t \quad = L$

bifurcation sur t dans la seule feuille restante L:

$L[\top/t] = \Big(\top \wedge \big(\neg a\big)\Big) \Rightarrow \top$

$L[\bot/t] = \Big(\bot \wedge \big(\neg a\big)\Big) \Rightarrow \bot$

simplification des deux feuilles

$L[\top/t] \rightarrow \top$

branche terminée.

$L[\bot/t] \rightarrow \bot \Rightarrow \bot \rightarrow \top$

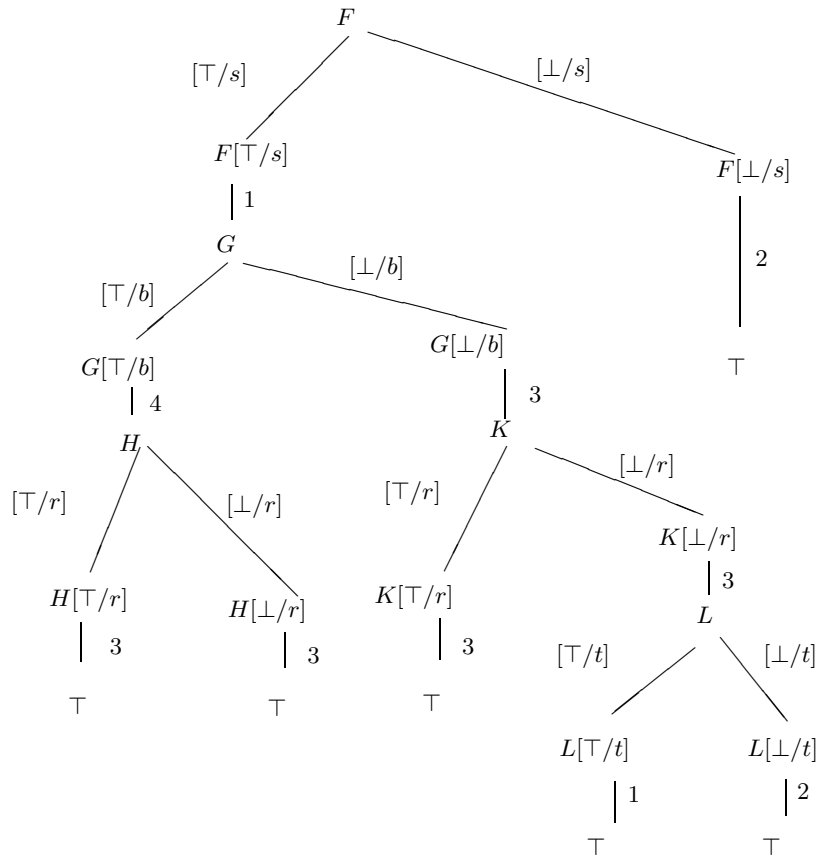
branche terminée

Toutes les branches se terminent par $\top$ c'est une formule universellement valide.

La structure de l'arbre est la suivante.

On a représenté une série de simplifications par une seule arête, avec le nombre de simplifications.

On y arrive en 35 étapes élémentaires (substitution, simplifications), à comparer avec 894 cases de la table de vérité.



4.9 Correction de l'algorithme de Quine

On montre que pour tout noeud F dans l'arbre de Quine terminé on a:

- si le sous-arbre en dessous du noeud F n'a que des feuilles $\top$, alors $F \equiv \top$
- si le sous-arbre en dessous du noeud F n'a que des feuilles $\bot$, alors $F \equiv \bot$
- si le sous-arbre en dessous du noeud F a au moins une feuille $\top$ alors F est consistante.

On procède par récurrence sur la distance maximum du noeud F à une feuille située en dessous de lui.

4.9.1 Correction de l'algorithme de Quine (hauteur 1)

Si l'arbre est réduit à une feuille, comme l'algorithme est terminé, il n'y pas de proposition atomique (sinon 3 s'appliquerait) et la formule est $\top$ ou $\bot$ (sinon 2 s'appliquerait). Si c'est $\top$, la formule est universellement valide et toutes les feuilles situées en dessous sont $\top$, et si c'est $\bot$, la formule est inconsistante et toutes les feuilles en dessous sont $\bot$.

4.9.2 Correction de l'algorithme de Quine (simplifications)

Si la règle sous le noeud considéré est une simplification de F à $F[X := X']$.

Par hypothèse de récurrence, la formule $F[X := X']$ est:

U universellement valide si et seulement si les feuilles en dessous de $F[X := X']$ sont toutes $\top$

I inconsistante si et seulement si les feuilles en dessous de $F[X := X']$ sont toutes $\bot$

C consistante si et seulement si une feuille en dessous de $F[X := X']$ est $\top$.

Comme les simplifications sont des équivalences, on a d'après la prop. 4 $F \equiv F[X := X']$. Les feuilles en dessous de F sont exactement celles en dessous de $F[X := X']$. F est donc U (resp. I, C) ssi $F[X := X']$ est U (resp. I, C)

4.9.3 Correction de l'algorithme de Quine (bifurcations)

S'il s'agit d'une bifurcation, disons sur la proposition atomique p, l'arbre de droite à pour racine $F[\top/p]$ et celui de gauche $F[\bot/p]$.

Formules universellement valides Si l'arbre sous F ne contient que des feuilles $\top$, alors l'arbre sous $F[\top/p]$ ne contient que des feuilles $\top$, et par hypothèse de récurrence on a $F[\top/p] \equiv \top$. De même l'arbre sous $F[\bot/p]$ ne contient que des feuilles $\top$, et par hypothèse de récurrence on a $F[\top/p] \equiv \top$. En vertu de la proposition 3 on a donc $F \equiv \top$.

Réciproquement, si $F \equiv \top$, on a, en vertu de la proposition 3 $F[\top/p] \equiv \top$ et $F[\bot/p] \equiv \top$. Par hypothèse de récurrence toutes les feuilles sous $F[\top/p]$ sont $\top$ ainsi que toutes les feuilles sous $F[\bot/p]$. Donc toutes les feuilles sous F sont $\top$.

Formules inconsistantes Même raisonnement avec des feuilles $\bot$.

Formules consistantes Si l'arbre sous F contient au moins une feuille $\top$, alors l'un des deux arbres $F[\top/p]$ ou $F[\bot/p]$ contient une feuille $\top$. L'une de ces deux formule est donc consistante, et en vertu de la prop. 3 il en est de même de F.

Si F est consistante, alors, d'après la prop. 3 l'une des deux formules $F[\top/p]$ ou $F[\bot/p]$ est consistante. Par hyptohèse de récurrence elle a donc au moins une feuille $\top$ en dessous d'elle, et donc F a cette même feuille $\top$ sous elle.

5 Formes normales disjonctives et conjonctives, canoniques ou non

5.1 Formes normales disjonctives et conjonctives

Toute formule F peut se tranformer en une formule équivalente qui est une conjonction de disjonctions de propositions atomiques et de négations de propositions atomiques.

On remplace$\Rightarrow$ et $\Leftrightarrow$ par leur définition en termes de $\vee$ et $\wedge$ — ce qui donne une formule équivalente en vertu de 4.

Les lois de de Morgan permettent de se ramener à des négations sur les propositions atomiques exclusivement — et un nombre quelconque de négations se ramène toujours à zéro ou une négation.

La distributivité, qui fonctionne dans les deux sens, permet alors soit d'avoir des conjonctions de disjonctions, soit des disjonctions de conjonctions.

EXEMPLE SUR UN ARBRE DE FORMULE.

Pour mettre une formule sous forme normale disjonctive on peut procéder comme pour développer une exrpession arithmétique avec $\times$ et $+$ — la distributivité est la même loi. C'est extrêmement long et compliqué, on verra une meilleure méthode ci-après.

Pour voir que cela se termine néanmoins, on procède par induction sur la hauteur de l'arbre de la formule avec des $\vee$ et des $\wedge$ à plusieurs composantes.

Si l'arbre commence par $\vee$ il suffit de transformer chaque sous arbre et de le remettre.

Si l'arbre commence par $\wedge$, disons que c'est $F = (A \vee B \vee C) \wedge X \wedge Y$ on considère sa branche la plus longue, disons qu'elle passe par la sous formule $(A \vee B \vee C)$ — A,B,C sont toutes des conjonctions, disons $A = (K \wedge L)$ $B = (M \wedge N)$, $C = (P \wedge Q)$. Par distributivité la formule F est équivalent à la disjonction de $K \wedge L \wedge X \wedge Y$, $M \wedge N \wedge X \wedge Y$ et $P \wedge Q \wedge X \wedge Y$.

Mais l'arbre de chacune de ces trois formules est moins haut que celui de F: par hypothèse de réccurrence on peut donc tranformer chacune en disjonction de conjonction, ce qui fournit un équivalent pour F.

5.2 Forme normale disjonctive et conjonctives canoniques

Etant donné la table de vérité de F, comment trouver une formule qui soit sémantiquement équivalente à F? En fait on peut demander à ce que F soit sous forme normale disjonctive ou conjonctive et que chaque proposition atomique figure dans chaque conjonction exactement une fois (et une telle écriture est unique).

Donc toute fonction booléenne (de $\{0,1\}^n$ dans $\{0,1\}$) correspond à une formule à n propositions atomiques, sous l'une de ces deux formes. Si l'une à k termes, l'autre en a $2^n - k$. On peut donc en particulier réaliser toute fonction booléenne par un circuit booléen avec des portes AND, OR et NOT.

Comme une fonction de $\{0,1\}^n$ dans $\{0,1\}^p$ correspond à p fonctions booléennes de $\{0,1\}^n$ dans $\{0,1\}$ on peut également réaliser des circuits à plusieurs sorties.

5.2.1 Forme normale disjonctive canonique

Pour chaque ligne sur laquelle F vaut 1, former une conjonction qui contient p si p vaut 1 et $\neg p$ si p vaut 0. On forme ensuite la disjonction de ces conjonctions. La

disjonction vaut 0 si et seulement si l'une des conjonction est vraie, c.-à-d.ssi on est sur l'une des lignes où F vaut 1.

5.2.2 Forme normale conjonctive canonique

Pour chaque ligne où F vaut 0, on forme la disjonction suivante: on met p si p vaut 0 sur cette ligne et $(\neg p)$ si p vaut 1 sur cette ligne. Cette disjonction vaut donc 1 si et seulement si on n'est pas dans cette ligne. On forme ensuite la conjonction de ces disjonctions. La conjonction vraie si et seulement si toutes les disjonctions valent 1, c.-à-d.ssi on n'est dans aucune des lignes où F vaut 0, c.-à-d.ssi F vaut 1.

p	q	r	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

$$
\begin{array}{lcccccccc}
F & \equiv & & ((\neg p) & \wedge & (\neg q) & \wedge & (\neg r)) \\
 & & \vee & ((\neg p) & \wedge & q & \wedge & (\neg r)) \\
 & & \vee & (p & \wedge & (\neg q) & \wedge & (\neg r)) \\
 & & \vee & (p & \wedge & (\neg q) & \wedge & r) \\
\hline
F & \equiv & & (p & \vee & q & \vee & (\neg r)) \\
 & & \wedge & (p & \vee & (\neg q) & \vee & (\neg r)) \\
 & & \wedge & ((\neg p) & \vee & (\neg q) & \vee & r) \\
 & & \wedge & ((\neg p) & \vee & (\neg q) & \vee & (\neg r))
\end{array}
$$

6 Un système déductif complet: le calcul des séquents

6.1 Présentation

On va présenter un système formel pour démontrer une formule, de sorte que les formules universellement valides soient exactement les formules démontrées.

Pour ce système on présentera un algorithme déterministe de recherche de démonstration qui en plus de dire si la formule est ou non une tautologie, fournit immédiatement

- toutes les valuations rendant la formule vraie
- toutes les valuations rendant la formule fausse
- une forme normale conjonctive
- une forme normale conjonctive

6.2 Séquents: définition formelle

On manipule des expressions appelées sequents:

$$A_1, \ldots, A_k \vdash B_1, \ldots, B_l$$

- $A_1, \ldots, A_k$ sont des formules appelées hypothèses du séquent. Si $k = 0$, le séquent n'a pas d'hypothèse.
- $B_1, \ldots, B_l$ sont des formules appelées conclusions du séquent. Si $l = 0$, le séquent n'a pas de conclusion.

6.3 Signification d'un séquent

Un séquent $A_1, \ldots, A_k \vdash B_1, \ldots, B_l$
signifie que
la conjonction des $A_1, \ldots, A_k$ entraîne la disjonction des $B_1, \ldots, B_l$
ou encore
si toutes les formules de $A_1, \ldots, A_k$ sont vraies alors l'une au moins des formules $B_1, \ldots, B_l$ est vraie.

Conséquemment on appellera *formule associée au séquent* la formule

$$(A_1 \wedge \cdots \wedge A_k) \Rightarrow (B_1 \vee \cdots \vee B_l)$$

[Si $k = 0$ alors $(A_1 \wedge \cdots \wedge A_k) = \top$ et si $l = 0$ alors $(B_1 \vee \cdots \vee B_l) = \bot$]

Etant donnée une valuation, on dira dira qu'un séquent est vrai ou faux si et seulement si la formule associée l'est pour cette valuation.

6.4 Démonstrations

On construit des preuves qui sont des arbres.

- Les feuilles sont des axiomes
- Les branchements sont des règles du calcul des séquents.

Nous utiliserons les connecteurs $\vee, \wedge, \Rightarrow, \neg$.

Les axiomes sont tous les séquents de la forme: $\Gamma \vdash \Delta$ tel que Γ et Δ sont des suites de formules atomiques qui contiennent une même proposition atomique. [On

peut également autoriser comme axiomes les séquents $\Gamma \vdash \Delta$ tel que Γ et Δ comportent une proposition (non nécessairement atomique) en commun.]

Les majuscules grecques désignent des suites de formules.

Règles

$$\frac{\Gamma,A,B,\Delta \vdash \Theta}{\Gamma,(A \wedge B),\Delta \vdash \Theta}\wedge_g \qquad \frac{\Theta \vdash \Gamma,A,\Delta \qquad \Theta \vdash \Gamma,B,\Delta}{\Theta \vdash \Gamma,(A \wedge B),\Delta}\wedge_d$$

$$\frac{\Gamma,A,\Delta \vdash \Theta \qquad \Gamma,B,\Delta \vdash \Theta}{\Gamma,(A \vee B),\Delta \vdash \Theta}\vee_g \qquad \frac{\Theta \vdash \Gamma,A,B,\Delta}{\Theta \vdash \Gamma,(A \vee B),\Delta}\vee_d$$

$$\frac{\Gamma,\Delta \vdash A,\Theta \qquad B,\Gamma,\Delta \vdash \Theta}{\Gamma,(A \Rightarrow B),\Delta \vdash \Theta}\Rightarrow_g \qquad \frac{A,\Gamma \vdash B,\Delta,\Theta}{\Gamma \vdash \Delta,(A \Rightarrow B),\Theta}\Rightarrow_d$$

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

6.5 Exemples de démonstrations

$$\dfrac{\dfrac{\dfrac{\dfrac{axiome}{\mathbf{q},p,p,s \vdash \mathbf{q},t,t}}{q,p,(\neg q),p,s \vdash t,t}\neg_g}{q,p,(\neg q),(p \wedge s) \vdash t,t}\wedge g \qquad \dfrac{\dfrac{\dfrac{axiome}{r,\mathbf{q},p,p,s \vdash \mathbf{q},t}}{r,q,p,(\neg q),p,s \vdash t}\neg_g}{r,q,p,(\neg q),(p \wedge s) \vdash t}\wedge g}{q,p,(\neg q),(t \Rightarrow r),(p \wedge s) \vdash t}\Rightarrow_g$$

Pour vérifier la correction du raisonnement du paragraphe 4.2 il faut démontrer la formule:

$$\Big(\big(s \Rightarrow (b \vee t)\big) \wedge \big((b \vee a) \Rightarrow (r \wedge m)\big) \wedge \neg r\Big) \Rightarrow (s \Rightarrow t)$$

ce qui se fait ainsi:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{t,s\vdash b,a,t,r \quad b,s\vdash b,a,t,r}{(b\vee t),s\vdash b,a,t,r}\vee_g \quad s\vdash s,b,a,t,r}{(s\Rightarrow(b\vee t)),s\vdash b,a,t,r}\Rightarrow_g}{(s\Rightarrow(b\vee t)),s\vdash(b\vee a),t,r}\vee_d \quad \dfrac{\dfrac{\dfrac{t,r,m,s\vdash t,r \quad b,r,m,s\vdash t,r}{b\vee t,r,m,s\vdash t,r}\vee_g \quad r,m,s\vdash s,t,r}{(s\Rightarrow(b\vee t)),r,m,s\vdash t,r}\Rightarrow_g}{(s\Rightarrow(b\vee t)),(r\wedge m),s\vdash t,r}\wedge_g}{(s\Rightarrow(b\vee t)),((b\vee a)\Rightarrow(r\wedge m)),s\vdash t,r}\Rightarrow_g}{(s\Rightarrow(b\vee t)),((b\vee a)\Rightarrow(r\wedge m)),\neg r,s\vdash t}\neg_g}{(s\Rightarrow(b\vee t)),((b\vee a)\Rightarrow(r\wedge m)),\neg r\vdash s\Rightarrow t}\Rightarrow_d}{(s\Rightarrow(b\vee t))\wedge((b\vee a)\Rightarrow(r\wedge m))),\neg r\vdash s\Rightarrow t}\wedge_g}{((s\Rightarrow(b\vee t))\wedge((b\vee a)\Rightarrow(r\wedge m)))\wedge\neg r)\vdash s\Rightarrow t}\wedge_g}{\vdash(((s\Rightarrow(b\vee t))\wedge((b\vee a)\Rightarrow(r\wedge m)))\wedge\neg r)\Rightarrow(s\Rightarrow t)}\Rightarrow_d$$

6.6 Validité du calcul des séquents

6.6.1 Validité des axiomes

Remarquons que les axiomes corrrespondent à des formules universellement valides, c.-à-d.

Propriété 8 *Un axiome*

$$a_1,\ldots,a_k\vdash b_1,\ldots,b_l$$

avec $a_i=b_j$ (pour un i et un j t.q. $1\leq i\leq k$ et $1\leq j\leq l$) a une formule associée universellement valide — $v((a_1\wedge\ldots\wedge a_k)\Rightarrow(b_1\vee\cdots\vee b_l))=1$ pour toute valuation v.

En effet,
— si $v(a_i)=v(b_j)=1$ alors $v(b_1\vee\cdots\vee b_l)=1$ et donc $v((a_1\wedge\cdots\wedge a_k)\Rightarrow(b_1\vee\cdots\vee b_l))=1$.
— si $v(a_i)=v(b_j)=0$ alors $v(a_1\wedge\ldots\wedge a_k)=0$ et donc $v((a_1\wedge\ldots\wedge a_k)\Rightarrow(b_1\vee\cdots\vee b_l))=1$.

6.6.2 Validité des règles

Puisqu'il s'agit d'un système déductif, il faut que pour toute valuation:

Propriété 9 *Etant donnée une valuation v,*
si tous les séquents prémisses de la règle R (1 ou 2 suivant R) valent 1,
alors le séquent conclusion de R vaut 1 *aussi.*

Cettte propriété, plus la validité des axiomes (prop. 8), entraîne que tous les séquents démontrables sont vrais pour toute valuation. En particulier si $\vdash F$ est démontrable alors F est une tautologie.

Validité la règle de l'implication à droite Cette règle consiste à dire que si "on a montré B avec une hypothèse A", alors "on a montré $A\Rightarrow B$ *sans* l'hypothèse A."

$$\dfrac{A,\Gamma\vdash B,\Delta,\Theta \quad \mathbf{P}}{\Gamma\vdash\Delta,(A\Rightarrow B),\Theta \quad \mathbf{C}}\Rightarrow_d$$

Soit v une valuation. Si $v(\mathbf{P})=1$ alors l'une des formules de B,Δ,Θ vaut 1 (a) ou l'une des formules de A,Γ vaut 0 (b).

(a) Dans le premier cas, si $v(B) = 1$ alors $v(A \Rightarrow B) = 1$ et l'une des formules de Δ,$(A \Rightarrow B)$,Θ vaut 1, donc **C** vaut 1. Si $v(B) = 0$ alors une formule de Δ,Θ vaut 1 et donc une formule de Δ,$(A \Rightarrow B)$,Θ vaut 1 et finalement C vaut 1.

(b) Dans le second cas, si $v(A) = 0$ alors $v(A \Rightarrow B) = 1$ et l'une des formules de Δ,$(A \Rightarrow B)$,Θ vaut 1, donc **C** vaut 1. Sinon, une des formules de Γ vaut 0, et par conséquent **C** vaut 1.

Validité de la règle de l'implication à gauche Cette règle est moins intuitive.

$$\frac{\Gamma,\Delta \vdash A,\Theta \quad \textbf{P1} \qquad B,\Gamma,\Delta \vdash \Theta \quad \textbf{P2}}{\Gamma,(A \Rightarrow B),\Delta \vdash \Theta \quad \textbf{C}} \Rightarrow_g$$

Soit v une valuation, supposons que $v(\textbf{P1}) = 1$ et $v(\textbf{P2}) = 1$. On sait donc que

(a) (l'une des formules de Γ,Δ vaut 0 (a1) OU l'une des formules de A,Θ vaut 1 (a2))

ET (b)(l'une des formules de B,Γ,Δ vaut 0 (b1) ou l'une des formule de Θ vaut 1 (b2)).

Si l'une des formule de Θ vaut 1 alors **C** vaut 1.

Si l'une des formules de Γ,Δ vaut 0, alors **C** vaut 0.

Il reste à examiner la valeur de **C** lorsque toutes les formules de Γ,Δ valent 1 (I) et toutes les formules de Θ valent 0 (II).

D'après (a) et (I), on a nécessairement (a2), et d'après (II) on a $v(A) = 1$.

D'après (b) et (II), on a nécessairement (b1), et d'après (I) $v(B) = 0$.

Par conséquent $v(A \Rightarrow B) = 0$, l'une des formules de Γ,$(A \Rightarrow B)$,Δ vaut 0, et **C** vaut donc 1.

6.6.3 Validité

Propriété 10 (validité) *Si un séquent est démontrable, alors la formule correspondante est une tautologie.*

Cette propriété se vérifie aisément par récurrence sur la hauteur de l'arbre de démonstration à partir des propriétés 8 et 9.

Considérons une valuation v.

Les axiomes correspondent à des formules vraies pour v (puisque ce sont des tautologies, cf. 8).

Or on sait d'après la prop. 9 que si les prémisses d'une règle sont vraies pour v alors la conclusion de la règle est vraie pour v.

Donc la formule associée au séquent conclusion de la démonstration est vraie pour v.

Comme l'argument fonctionne pour tout v, le séquent associé à la conclusion de la démonstration est vrai pour toute valuation.

6.7 Réversibilité des règles

Chaque règle R du calcul des séquents vérifie en outre la propriété suivante:

Propriété 11 *Etant donnée une valuation v, si le séquent conclusion de la règle R est vrai, alors les séquents prémisses de la règle R sont vrais.*

6.7.1 Reversibilité de la règle de l'implication à gauche

$$\frac{\Gamma,\Delta \vdash A,\Theta \quad \textbf{P1} \qquad B,\Gamma,\Delta \vdash \Theta \quad \textbf{P2}}{\Gamma,(A \Rightarrow B),\Delta \vdash \Theta \quad \textbf{C}} \Rightarrow_g$$

Soit v une valuation. Si le séquent conclusion de la règle est vrai pour v, c'est que l'une des formules de Γ,$(A \Rightarrow B)$,Δ est fausse ou que l'une des formules de Θ est vraie.

Si l'une des formules de Γ,Δ est fausse, alors **P1** est vrai, et **P2** aussi. Si l'une des formules de Θ est vraie, alors **P1** est vrai, et **P2** aussi. On se amène donc au cas où toutes les formules de Γ,Δ sont vraies, où toutes les formules de Θ sont fausses, et où $A \Rightarrow B$ est faux. On a donc A vrai et B faux. Comme A est vrai **P1** est vrai et comme B est faux, **P2** est vrai.

6.7.2 Reversibilité de la règle de l'implication à droite

$$\frac{A,\Gamma \vdash B,\Delta,\Theta \quad \mathbf{P}}{\Gamma \vdash \Delta,(A \Rightarrow B),\Theta \quad \mathbf{C}} \Rightarrow_d$$

Soit v une valuation. Si **C** est vrai, c'est que l'une des formules de Γ est fausse ou que l'une des formules de Δ,$(A \Rightarrow B)$,Θ est vraie.

Si l'une des formules de Γ est fausse, **P** est vrai. Si l'une des formules de Δ,Θ est vraie, **P** est vrai.

Si aucun de ces deux cas ne s'applique, c'est que $A \Rightarrow B$ est vrai. Si A est faux, **P** est vrai. Si A est vrai, B l'est aussi, et donc **P** est vrai.

6.8 Propriétés du calcul des séquents

En combinant les prop. 11 et 9 on obtient:

Propriété 12 *Soit v une valuation; alors pour toute règle R du calcul des séquents:*

(i) tous les séquents prémisses de la règle R (1 ou 2 suivant R) valent 1, *si et seulement si*

(ii) le séquent conclusion de R vaut 1.

si (i) alors (ii) est la prop. 9
si (ii) alors (i) est la contraposée de la prop. 11.

6.9 Un algorithme de démonstration automatique

Algorithme 13 *On construit un arbre de séquents de sorte que les branchements soient des règles. Si les feuilles sont des axiomes, alors la formule est démontrable, et sinon on peut produire une valuation qui rend la formule fausse.*

(a). L'arbre est initialisé au séquent à démontrer/réfuter.

(b). Si toutes les feuilles ne contiennent que des propositions atomiques, c'est fini.

(c). Si une feuille est un séquent qui contient une proposition composée, écrire en dessus les prémisses de la règle qui fabrique cette formule, et on retourne en (b)

Une fois choisie la formule à décomposer, les prémisses sont totalement déterminées, et ce choix n'influe pas sur le fait qu'on obtienne ou non une démonstration.

6.9.1 Propriétés de l'algorithme

Propriété 14 *L'algorithme appliqué à un un séquent* $a_1, \ldots, a_k \vdash b_1, \ldots, b_l$ *s'arrête*

(1) soit sur une démonstration et pour toute valuation $v((a_1, \ldots, a_k) \Rightarrow (b_1 \vee \cdots \vee b_l)) = 1$

(2) soit sur un arbre de séquents dont les branchements sont des règles, et dont au moins une feuille n'est pas un axiome, et il existe une valuation v *telle que* $v((a_1, \ldots, a_k) \Rightarrow (b_1 \vee \cdots \vee b_l)) = 0$

L'arbre construit ne peut avoir de branche infinie: le nombre de connecteurs dans une prémisse d'une règle est toujours moindre que le nombre de connecteurs dans la conclusion de la règle.

Lorsque l'algorithme est fini, toutes les feuilles sont des séquents ne contenant que des propositions atomiques,

(1) soit qui sont toutes des axiomes (et on aune démonstration)

(2) soit dont l'une au moins n'est pas un axiome

(1) Si toutes les feuilles sont des axiomes, on a une démonstration et la formule correspondant au séquent démontré est donc une tautologie d'après la prop. 10.

(2) Si l'une au moins des feuilles $p_1, p_2, \ldots, p_k \vdash q_1, q_2, \ldots, q_l$ n'est pas un axiome, c'est que $p_i \neq q_j$ pour $1 \leq i \leq k$ et $1 \leq j \leq l$. On définit une valuation v par $v(p_i) = 1$ pour $1 \leq i \leq k$ et $v(q_j) = 0$ pour $1 \leq j \leq l$ (ce n'est pas contradictoire: il n'y a pas de proposition atomique qui soit à la fois l'un des p_i et l'un des q_j). Pour cette valuation v la formule $(p_1 \wedge p_2 \wedge \cdots \wedge p_k) \Rightarrow (q_1 \vee q_2 \vee \cdots \vee q_l)$ associée au séquent $p_1, p_2, \ldots, p_k \vdash q_1, q_2, \ldots, q_l$ vaut 0. D'après la proposition 12 le séquent en dessous est faux, celui encore en dessous aussi, et donc le séquent racine (que l'on cherchait à démontrer) est faux pour cette valuation.

6.10 Complétude

Propriété 15 (complétude) *Si la formule correspondant à un séquent est une tautologie, alors le séquent est démontrable.*

En effet en lui appliquant l'algorithme 13 on obtient un arbre de séquents de type (1) ou de type (2). D'après la prop. 14 s'il était de type (2) il existerait une valuation qui rende faux ce séquent, il est donc de type (1) c.-à-d.démontrable.

En combinant cela avec la prop. 10 on a:

Propriété 16 *Une formule* F *est une tautologie si et seulement si* $\vdash F$ *est démontrable.*

6.11 Exemples d'utilisation de l'algorithme de recherche de démonstration

6.11.1 L'algorithme de recherche de démonstration pour un séquent démontrable

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathbf{q},p,p,s \vdash \mathbf{q},t,t}{q,p,(\neg q),p,s \vdash t,t}\,\neg_g}{q,p,(\neg q),(p \wedge s) \vdash t,t}\,\wedge g \qquad \dfrac{\dfrac{r,\mathbf{q},p,p,s \vdash \mathbf{q},t}{r,q,p,(\neg q),p,s \vdash t}\,\neg_g}{r,q,p,(\neg q),(p \wedge s) \vdash t}\,\wedge g}{q,p,(\neg q),(t \Rightarrow r),(p \wedge s) \vdash t}\,\Rightarrow_g$$

6.11.2 L'algorithme de recherche de démonstration pour une formule non démontrable

$$\dfrac{\dfrac{\dfrac{\dfrac{p \vdash r,s}{\vdash r,s,(\neg p)}\,\neg_d}{(\neg r) \vdash s,(\neg p)}\,\neg_g}{\vdash (\neg p),((\neg r) \Rightarrow s)}\,\Rightarrow_d \quad \dfrac{\dfrac{q \vdash r,s}{(\neg r),q \vdash s}\,\neg_g}{q \vdash ((\neg r) \Rightarrow s)}\,\Rightarrow_d}{((\neg p) \Rightarrow q) \vdash ((\neg r) \Rightarrow s)}\,\Rightarrow_g}{\vdash ((\neg p) \Rightarrow q) \Rightarrow ((\neg r) \Rightarrow s)}\,\Rightarrow_d$$

$$(((\neg p) \Rightarrow q) \Rightarrow ((\neg r) \Rightarrow s))$$
$$\equiv (((\neg q) \vee r \vee s) \wedge ((\neg p) \vee r \vee s))$$

6.11.3 L'algorithme de recherche de démonstration pour un séquent non démontrable

$$\dfrac{\dfrac{\dfrac{\vdash p,q}{(\neg p) \vdash q}\,\neg_g}{\vdash ((\neg p) \Rightarrow q)}\,\Rightarrow_d \quad \dfrac{\dfrac{r \vdash}{\vdash (\neg r)}\,\neg_d \quad s \vdash}{((\neg r) \Rightarrow s) \vdash}\,\Rightarrow_g}{(((\neg p) \Rightarrow q) \Rightarrow ((\neg r) \Rightarrow s)) \vdash}\,\Rightarrow_g$$

$$(((\neg p) \Rightarrow q) \Rightarrow ((\neg r) \Rightarrow s))$$
$$\equiv s \vee r \vee ((\neg p) \wedge (\neg q))$$

6.11.4 Forme normale conjonctive

L'algorithme 13 calcule une forme normale conjonctive d'une formule A en l'appliquant à $\vdash A$ — A est la formule associée au séquent $\vdash A$.

Si A est démontrable, $A \equiv p \vee (\neg p)$.

Si A n'est pas démontrable, d'après la propriété 12, une valuation rend A vrai exactement quand toutes les feuilles qui ne sont pas des axiomes sont vraies. Une telle feuille $p_1,p_2,\ldots,p_k \vdash q_1,q_2,\ldots,q_l$ est vraie lorsque la formule

$$(\neg p_1) \vee (\neg p_2) \vee \cdots \vee (\neg p_k) \vee q_1 \vee q_2 \vee \cdots \vee q_l$$

est vraie.

Donc A équivaut à la conjonction de toutes ces formules.

6.11.5 Forme normale disjonctive

L'algorithme 13 calcule une forme normale disjonctive d'une formule A en l'appliquant à $A \vdash$ — $\neg A$ est la formule associée au séquent $A \vdash$.

Si $A \vdash$ est démontrable, A est faux et $A \equiv p \wedge (\neg p)$.

Si $A \vdash$ n'est pas démontrable, d'après la propriété 12, une valuation rend A vrai (c.-à-d.$A \vdash$ faux) exactement quand l'une au moins des feuilles qui ne sont pas des axiomes est fausse. Une telle feuille $p_1,p_2,\ldots,p_k \vdash q_1,q_2,\ldots,q_l$ est fausse lorsque la formule

$$p_1 \wedge p_2 \wedge \cdots \wedge p_k \wedge (\neg q_1) \wedge (\neg q_2) \wedge \cdots \wedge (\neg q_l)$$

est fausse.

Donc A équivaut à la disjonction de toutes ces formules.

7 Résolution

7.1 Présentation de la méthode

On précente maintenant une méthode pour montrer qu'une formule est inconsistante (ou contradictoire). Pour cela on suppose que la formule est sous forme normale conjonctive.

On appelle *littéral* une proposition atomique ou la négation d'une proposition atomique :
p,$(\neg p)$,q,$(\neg q)$,s,$(\neg s)\ldots$

Une *clause* est une disjonction de littéraux :
$(p \vee (\neg q) \vee s)$,$(p \vee q \vee (\neg s)), \ldots$

Un ensemble de clause est compris comme la conjonction des clauses; c'est donc une formule en forme normale conjonctive, par ex.
$(p \vee (\neg q) \vee s) \wedge (p \vee q \vee (\neg s)) \wedge (s \vee (\neg t))$

Le principe de la résolution est l'équivalence suivante:

$[1] \quad (A \vee p) \wedge (B \vee (\neg p))$
$\equiv (A \vee p) \wedge (B \vee (\neg p)) \wedge (A \vee B) \quad [2]$

$[1]$ entraîne $[2]$. Il suffit de montrer que
$F = ((A \vee p) \wedge (B \vee (\neg p)) \Rightarrow (A \vee B)) \equiv \top$.
$F[\top/p] \equiv (((A \vee \top) \wedge (B \vee \bot)) \Rightarrow (A \vee B))$
$\equiv (B \Rightarrow (A \vee B)) \equiv \top$
$F[\bot/p] \equiv (((A \vee \bot) \wedge (B \vee \top)) \Rightarrow (A \vee B))$
$\equiv (A \Rightarrow (A \vee B)) \equiv \top$.
Donc $(F \Rightarrow (A \vee B)) \equiv \top$.

$[2]$ entraîne $[1]$ est une évidence.

Pour voir si un ensemble de clauses est contradictoire, on ajoute à l'ensemble de clauses de nouvelles clauses en appliquant le principe ci-dessus : si une clause C_1 contient p et une autre C_2 contient $(\neg p)$, on ajoute la clause C appelée *résolvant* et dont les termes sont ceux de C_1 moins p et ceux de C_2 moins $(\neg p)$.

Par exemple le résolvant de
$C_1 = (p \vee q \vee s \vee (\neg t))$ et $C_2 = (u \vee (\neg q) \vee v \vee (\neg t))$
est $C = (p \vee s \vee (\neg t) \vee u \vee v)$
— comme $(X \vee X) \equiv X$ il est inutile de répéter les littéraux communs.

D'après ce qui précède l'ensemble de clauses avant l'ajout d'un résolvant est équivalent à l'ensemble de clauses après ajout d'un résolvant.

S'il on obtient un résolvant vide à partir des clauses p et $(\neg p)$ on ajoute le résolvant $\bot$, et l'ensemble de clauses est clairement contradictoire.

7.2 Résolution — premier exemple

Soit la formule F (déjà vue)
$\Big(\big(s \Rightarrow (b \vee t)\big) \wedge \big((b \vee a) \Rightarrow (r \wedge m)\big) \wedge \neg r\Big) \Rightarrow (s \Rightarrow t)$

Pour montrer que F est une tautologie, on va monter que $(\neg F)$ est contradictoire.

$(\neg F) \equiv C_1 \wedge C_2 \wedge C_3 \wedge C_4 \wedge C_5 \wedge C_6 \wedge C_7 \wedge C_8$

où les clause C_i $1 \leq i \leq 8$ sont:

C_1	$=$	$((\neg s) \vee b \vee t)$	
C_2	$=$	$((\neg b) \vee r)$	
C_3	$=$	$((\neg b) \vee m)$	
C_4	$=$	$((\neg a) \vee r)$	
C_5	$=$	$((\neg a) \vee m)$	
C_6	$=$	$(\neg r)$	
C_7	$=$	s	
C_8	$=$	$(\neg t)$	
C_1	$=$	$((\neg s) \vee b \vee t)$	
C_2	$=$	$((\neg b) \vee r)$	
C_3	$=$	$((\neg b) \vee m)$	
C_4	$=$	$((\neg a) \vee r)$	
C_5	$=$	$((\neg a) \vee m)$	
C_6	$=$	$(\neg r)$	
C_7	$=$	s	
C_8	$=$	$(\neg t)$	
			Résolution:
C_9	$=$	$((\neg s) \vee t \vee r)$	résolvant de C_1 et C_2
C_{10}	$=$	$((\neg s) \vee t)$	résolvant de C_{10} et C_6
C_{11}	$=$	t	résolvant de C_{11} et C_7
C_{12}	$=$	$\bot$	résolvant de C_{12} et C_8
			contradiction trouvée

7.3 Résolution — deuxième exemple

Soit la formule $F = C_a \wedge C_b \wedge C_c$

C_a	$=$	$((\neg u) \vee v \vee w)$	
C_b	$=$	$(t \vee w)$	
C_c	$=$	$((\neg w) \vee u)$	
			Résolution
C_d	$=$	$v \equiv (u \vee (\neg u) \vee v)$	résolution de C_a et C_c
C_e	$=$	$(u \vee w)$	résolution de C_b et C_c
C_f	$=$	$(v \vee w)$	résolution de C_e et C_a
			plus de résolution
			clauses consistantes

7.4 Propriétés de l'algorithme de résolution

Cet algorithme termine toujours — grosso modo, quand on ajoute le résolvant de deux clauses, les résolutions qui apparaissent étaient déjà possibles.

Lorsqu'il est terminé, soit la clause vide $\bot$ a été obtenue et l'ensemble de clauses de départ est contradictoire — comme on l'a vu, les clauses ajoutées sont conséquences logiques des clauses de départ.

Soit il n'y a plus de résolutions possibles, sans que la clause vide ne soit apparu. L'ensemble de clauses de départ est alors consistant. S'en convaincre est plus difficile: il faut pour cela faire une correspondance entre les résolutions et les démonstrations dans une variante LKR du calcul des séquents avec des séquents de la forme $C_1, \ldots, C_n \vdash$ et une seule règle. On montre alors que les démonstrations du calcul des séquents usuel LK correspondent à des démonstrations dans LKR et réciproquement, et la complétude de LK donne le résultat. Cela nécessiterait plus d'heures de cours.