Institut Galilée | Département d'informatique
2002 | MM

TRAVAUX DIRIGÉS DE LOGIQUE

Ceci est un recueil d'exercices de base, utilisés en TD, de problèmes qui peuvent aussi servir à rédiger des énoncés de devoirs, et de notes d'ordre pédagogique, dépourvues de prétention didactique et d'originalité; par contre et en revanche, il manque d'exemples : il faut les produire face à son public, en n'oubliant jamais que les plus simples passent généralement pour simplistes et les autres pour incompréhensibles!

PETIT MANUEL DE LOGIQUE NAÏVE À L'USAGE DES FAMILLES.

Un étudiant arrivant en troisième année d'université n'est pas sans avoir déjà eu l'occasion de raisonner plusieurs fois, avec plus ou moins de bonheur; mais, le paradoxe le plus apparent de l'étude de la logique est que l'on y est toujours en train de raisonner. Ceci suppose que l'on ait déjà une logique présente à l'esprit, procédant du simple bon sens, légèrement instrumentalisé.
La logique naïve dont il est question ici, et dont l'usage est recommandé en toute circonstance, est justement une tentative d'instrumentalisation de ce *bon sens si bien partagé* ... où l'on pourra reconnaître une présentation informelle d'un système de déduction fort bien qualifiée, par Gentzen soi–même, de naturelle.

Les mots sont d'abord pris dans leur sens commun, mais l'usage répété et systématique de certains d'entre eux ne manquera pas de préciser ce sens de façon utile pour la suite.

- La logique naïve s'applique à démontrer **des énoncés** par le moyen de déductions, qui préservent la vérité : un énoncé déduit à partir d'énoncés vrais, est lui–même vrai. On choisit donc tout d'abord une collection d'énoncés de base A, B, ... (dont la nature dépend du domaine considéré) à partir desquels on construira d'autres énoncés, par des opérations grammaticales. En dehors des énoncés démontrés à la faveur de précédents exercices (qui sont des acquis définitifs) et des ressources propres au domaine auquel on s'intéresse (qui sont des énoncés vrais dans le domaine en question), les ressources disponibles à un moment donné d'une déduction sont des énoncés de deux ordres :
  – des hypothèses posées temporairement;
  – les énoncés que l'on en a déduit.

Une ressource peut être utilisée autant de fois que l'on veut, voire pas du tout.

Une déduction n'est une démonstration que lorsqu'elle ne dispose d'aucune hypothèse temporaire.

- Les déductions élémentaires qui sont proposées ici (sous une forme schématique pour des raisons de commodité), montrent comment on peut utiliser des ressources disponibles, figurées au dessus de la barre, pour en déduire une nouvelle, figurée dessous. Pris isolément, chacun des schémas en question (sauf peut–être ceux d'entre eux qui disposent de *ressources occultes*) sera d'abord regardé comme l'expression d'une trivialité; mais c'est leur ensemble, et l'usage cumulatif qu'on en fait, qui a de l'intérêt et n'est pas toujours facile à maîtriser : on passe encore du simpliste à l'incompréhensible, comme d'habitude ...

Voici la liste des opérations grammaticales qui permettent de construire de nouveaux énoncés à partir d'énoncés déjà connus et des commentaires sur les schémas de déduction qui leur sont attachés.

Il ne faut évidemment pas utiliser les schémas sous cette forme dans une démonstration, mais les rédiger, en un français (plus ou moins) digeste, et ce n'est pas la moindre affaire.

Schémas "propositionnels" naïfs

	$$\dfrac{\mathbf{L'absurde}}{A}$$
$$\dfrac{A \quad B}{A \textbf{ et } B}$$	$$\dfrac{A \textbf{ et } B}{A} \qquad \dfrac{A \textbf{ et } B}{B}$$
$$\dfrac{A}{A \textbf{ ou } B} \qquad \dfrac{B}{A \textbf{ ou } B}$$	$$\dfrac{A \textbf{ ou } B \quad \begin{matrix}[\![A]\!]\\ \vdots \\ C\end{matrix} \quad \begin{matrix}[\![B]\!]\\ \vdots \\ C\end{matrix}}{C}$$
$$\dfrac{\begin{matrix}[\![A]\!]\\ \vdots \\ B\end{matrix}}{A \Rightarrow B}$$	$$\dfrac{A \quad A \Rightarrow B}{B}$$

• $\mathbf{L'absurde}$ est un énoncé qui se présente dans tous les domaines! et qui est faux, absolument. Le schéma qui lui est directement attaché signifie que l'on peut en déduire n'importe quel énoncé.

• Lorsque A et B sont des énoncés, leur *conjonction* A **et** B est un énoncé. Les trois schémas qui sont directement attachés à la conjonction sont sans surprise.

• Lorsque A et B sont des énoncés, leur *disjonction* A **ou** B est un énoncé : cette disjonction ne doit pas être comprise dans un sens exclusif mais comme le barbarisme post–moderne *et/ou* souvent utilisé de nos jours. L'un des trois schémas qui lui sont attachés demande des explications :

$$\boxed{\dfrac{A \textbf{ ou } B \quad \begin{matrix}[\![A]\!]\\ \vdots \\ C\end{matrix} \quad \begin{matrix}[\![B]\!]\\ \vdots \\ C\end{matrix}}{C}}$$

il décrit le raisonnement *par disjonction des cas* : il permet, de déduire un énoncé C d'une ressource de la forme A **ou** B. Il peut se rédiger en deux temps, et se comprendre de la façon suivante :
— **Si** A (adjonction temporaire de l'hypothèse A), ... et il faut déduire C : ce travail accompli, l'hypothèse A est éliminée des ressources, ainsi que les énoncés dont la déduction dépend de cette hypothèse;
— **Si** B (adjonction temporaire de l'hypothèse B), ... et il faut déduire C : ce travail accompli, l'hypothèse B est éliminée des ressources, ainsi que les énoncés dont la déduction dépend de cette hypothèse.

Ceci fait, on n'utilise que la ressource A **ou** B pour déduire C.

• Lorsque A et B sont des énoncés, *l'implication* $A \Rightarrow B$ est un énoncé : l'usage d'un verbe, dans l'expression A **implique** B (au lieu d'une conjonction de coordination comme dans les cas précédents) est certainement à l'origine de nombreux malentendus. Le schéma de droite, traditionnellement appelé *modus ponens*, est la figure la plus célèbre du monde du raisonnement. L'autre schéma mérite quelques explications :

$$\boxed{\dfrac{\begin{array}{c}[\![A]\!]\\ \vdots \\ B\end{array}}{A \Rightarrow B}}$$

il est essentiel au point qu'on l'appelle généralement *la déduction*, ce qui n'est pas peu dire. Il peut se rédiger de la façon suivante :
— **Supposons** A (adjonction temporaire de l'hypothèse A),... il faut alors déduire B.
Lorsque ce travail est accompli, l'hypothèse A est éliminée des ressources, ainsi que les énoncés dont la déduction dépend de cette hypothèse; et l'on a bien déduit l'énoncé $A \Rightarrow B$ sans usage propre d'une ressource.

Ce raisonnement qui paraît naturel à ceux qui l'ont assimilé, peut produire des résultats fort gracieux, lorsqu'il tombe de mains moins habiles!

- Lorsque A est un énoncé, sa *négation* $\mathbf{non}\ A$ désigne l'énoncé $A \Rightarrow \mathbf{L'absurde}$.

De A et $\mathbf{non}\ A$ on peut évidemment déduire $\mathbf{L'absurde}$, par application du *modus ponens*.

$$A\ \mathbf{ou}\ \mathbf{non}\ A$$

Le principe du tiers exclu, selon lequel cet énoncé est vrai, quel que soit l'énoncé A, est nécessaire pour donner son caractère "classique" à la logique. Même lorsque ce principe est vérifié par les énoncés de bases, il n'est pas possible de l'étendre à tous les autres, par application des schémas. Montrons comment le tiers exclu permet de justifier deux schémas d'usage courant :

$$\boxed{\dfrac{\begin{array}{c}[\![\mathbf{non}\ A]\!]\\ \vdots \\ \mathbf{L'absurde}\end{array}}{A}}$$

Le raisonnement par l'absurde, dont le schéma est présenté ici, est une conséquence du principe du tiers exclu. En effet, si l'on sait déduire $\mathbf{L'absurde}$ sous l'hypothèse $\mathbf{non}\ A$ alors, on peut faire le raisonnement par cas suivant :
— Si A alors A!
— Si $\mathbf{non}\ A$ alors on sait déduire $\mathbf{L'absurde}$, d'où l'on peut déduire n'importe quel énoncé, en particulier A.

ATTENTION. Si après avoir temporairement posé l'hypothèse A on est capable de déduire $\mathbf{L'absurde}$, on pourra en conclure $A \Rightarrow \mathbf{L'absurde}$, c'est–à–dire $\mathbf{non}\ A$: ceci tient à la définition de la négation, mais n'a rien à voir avec un quelconque raisonnement par l'absurde!

$$\boxed{\dfrac{\begin{array}{c}[\![\mathbf{non}\ A]\!]\\ \vdots \\ B\end{array}}{A\ \mathbf{ou}\ B}}$$

Enfin voici le schéma d'un raisonnement qui peut être utile pour démontrer un énoncé de la forme $A\ \mathbf{ou}\ B$, qui est encore une conséquence du principe du tiers exclu. En effet, si l'on sait déduire B sous l'hypothèse $\mathbf{non}\ A$ on peut faire le raisonnement par cas suivant :
— Si A alors on a aussi $A\ \mathbf{ou}\ B$;
— Si $\mathbf{non}\ A$ alors on sait déduire B, donc aussi $A\ \mathbf{ou}\ B$.

Les énoncés que l'on est couramment conduit à manipuler sont des prédicats, disons, des phrases, qui expriment des relations entre des individus (dont la nature dépend du domaine considéré).

On utilise des *variables* x, y, ... pour désigner des individus de façon générique (par exemple, dans l'expression "soit x un entier ... "); mais il peut aussi intervenir des individus particuliers (par exemple, l'entier 1) ou bien calculés à partir d'autres (par exemple, les entiers de la forme $(x+1) \times x$). Nous utiliserons des lettres comme t, u, ... pour désigner des individus de l'une quelconque des sortes précédentes.

Pour insister sur le fait qu'un énoncé A dépend éventuellement de la variable x, nous utiliserons la notation $A[x]$ et, si t est l'expression d'un individu, $A[t]$ s'obtiendra en remplaçant x par t partout où il se trouve dans A.

Une quantification est une opération qui fait disparaître une variable : si $\mathbf{Q}x$ est un quantificateur sur x, $\mathbf{Q}xA[x]$ est **un énoncé qui ne dépend plus de** x. On dit de façon imagée que toute apparition de x qui est à la portée d'un quantificateur $\mathbf{Q}x$ est *muette* ou *liée*.

ATTENTION. A proprement parler, on ne peut pas prétendre qu'un énoncé qui dépend de variables soit vrai! Si nous insinuons tout de même que $A[x]$ est vrai, c'est pour dire qu'on obtient toujours un énoncé vrai lorsque l'on remplace x par un individu particulier.

Voici les quantifications et les schémas qui s'y rapportent.

• Lorsque $A[x]$ est un énoncé dépendant éventuellement de la variable x alors $\forall x A[x]$ est un énoncé ne dépendant plus de la variable x et qui se lit, se comprend et même peut s'écrire ou bien *pour tout* x $A[x]$ ou bien $A[x]$ *quel que soit* x.

$$\boxed{\frac{A[x]}{\forall x A[x]}}$$

Ce schéma de *généralisation* ne peut s'appliquer que lorsqu'aucune des hypothèses actuellement disponibles ne dépend de x, car il faut évidemment que x soit quelconque pour que la déduction soit correcte! En vue de l'application de ce schéma on sera conduit à écrire quelque chose comme :
— soit x quelconque . . .

$$\boxed{\frac{\forall x A[x]}{A[t]}}$$

Ce schéma *de spécialisation* exprime qu'un énoncé vrai pour tout individu x l'est aussi pour l'individu *particulier* désigné par t.
On se souviendra cependant que l'expression t ne doit pas dépendre d'une variable qui deviendrait muette dans l'énoncé $A[t]$.

• Lorsque $A[x]$ est un énoncé dépendant éventuellement de la variable x alors $\exists x A[x]$ est un énoncé ne dépendant plus de la variable x et qui se lit, se comprend et même peut s'écrire *il existe* x *tel que* $A[x]$.

$$\boxed{\frac{A[t]}{\exists x A[x]}}$$

La signification de ce schéma est claire : connaissant un individu particulier t qui permet d'affirmer l'énoncé $A[t]$, on est bien en droit d'en déduire $\exists x A[x]$!
On se souviendra cependant que l'expression t ne doit pas dépendre d'une variable qui deviendrait muette dans l'énoncé $A[t]$.

$$\boxed{\frac{\begin{array}{cc} & [\![A[x]]\!] \\ & \vdots \\ \exists x A[x] & B \end{array}}{B}}$$

Ce schéma peut se rédiger de la façon suivante :
— **Si** x **est tel que** $A[x]$ (adjonction temporaire de l'hypothèse $A[x]$),. . . il faut alors déduire B.
Lorsque ce travail est accompli, l'hypothèse $A[x]$ est éliminée des ressources, ainsi que les énoncés dont la déduction dépend de cette hypothèse; et l'on a bien déduit l'énoncé B de la seule ressource $\exists x A[x]$. L'individu désigné par x est caractérisé par le fait qu'il vérifie l'hypothèse $A[x]$, il ne pouvait pas apparaître avant cette hypothèse, ni ne peut survivre à sa disparition : aucune des hypothèses qui sont encore disponibles maintenant, ni B elle–même, ne peuvent donc dépendre de la variable x.

REMARQUE. La restriction sur l'usage des expressions t dans deux des schémas ne peut se justifier ici que par le fameux bon sens dont il a été question au début, mais on peut voir facilement qu'elle est nécessaire, en observant un contre–exemple.

Considérons l'énoncé $A[x]$ qui s'écrit $\forall y(y = x)$, par exemple dans le domaine des entiers, t l'expression $x+y$, alors $A[t]$ est égal à $\forall y(y = x+y)$. Si l'on ne tient pas compte du fait que t ne vérifie pas la restriction relativement à $A[x]$, on peut "démontrer" l'énoncé $\exists x \forall y(y = x+y) \Rightarrow \exists x \forall y(y = x)$ de la manière suivante :

supposons $\exists x \forall y(y = x+y)$, si x est tel que $\forall y(y = x+y)$ on peut en "déduire" $\exists x \forall y(y = x)$, ce qui permet de conclure.

Or, cette implication est fausse car on a bien $m = 0+m$ pour tout entier m mais, il n'existe pas d'entier n tel que $m = n$ pour tout m car, il y a plusieurs entiers!

Exercice 0. Rédaction de démonstrations.

L'équivalence $A \Leftrightarrow B$ désigne l'énoncé $(A \Rightarrow B)$ **et** $(B \Rightarrow A)$ qui signifie que A et B sont synonymes. La démonstration d'une équivalence nécessite la démonstration de deux implications.

Rédiger des démonstrations d'implications et d'équivalences bien connues, par exemple celles qui relient certaines opérations par le truchement de la négation.

Dans cet exercice, on évitera tout usage involontaire du principe du tiers exclu, et tout raisonnement abusif par l'absurde : d'une façon générale, on signale explicitement que l'on s'apprête à faire un raisonnement par l'absurde, car *il n'est pas si naïf que ça*, comme disait le grand lapin blanc.

L'INDUCTION.

On est souvent conduit, et spécialement en logique, à considérer des expressions appelées *termes* ou *formules*, qui sont construites à partir de symboles dépendant du domaine considéré.

Chacun de ces symboles admet une *arité*, c'est–à–dire un entier naturel qui indique le nombre d'arguments auxquels il s'applique :

– les symboles d'arité 0 désignent des individus, par exemple : les variables et les constantes de toute nature;
– les autres, des relations ou des fonctions, et sont appelés *opérateurs*, par exemple : les opérateurs arithmétiques et logiques, ...

Les expressions définies par un tel système, sont construites en appliquant un symbole à des expressions déjà construites (le nombre de ces expressions doit être égal à l'arité du symbole).

Cette phrase *bonne enfant* cache trop bien la difficulté du sujet : la définition d'une classe d'objets par **induction** (ou même par **récurrence**) et **la récursivité**, au sens informatique du terme, et le type de raisonnements qui leur est attaché. L'expérience montre que nos étudiants en informatique, sont généralement réfractaires à un tel feu, et qu'ils lui préfèrent les points de suspension (signe de ponctuation qui tend à s'énoncer "trois petits points", ce qui est déjà très révélateur d'une déperdition de sens!) ; or, il n'existe pas de langage de programmation comprenant lesdits points dans leur syntaxe de façon active, sauf à les définir ... par exemple, dans le cas simple d'une liste $(x_1, \ldots, x_n)$, par la récurrence suivante, sur l'entier naturel n :

– $(x_1, \ldots, x_0) = ()$, c'est–à–dire, la liste vide, base de la construction de toute liste;
– $(x_1, \ldots, x_{n+1}) = ((x_1, \ldots, x_n), x_{n+1})$ obtenue par adjonction de x_{n+1} à la fin de la liste $(x_1, \ldots, x_n)$.

Sauf dans les cas simples et statiques, il est préférable d'utiliser des définitions inductives et de faire des raisonnements par induction, quitte à paraître pesant. Il y a peu de questions en logique de base que l'on puisse sérieusement traiter autrement, surtout dans le cadre d'une licence d'informatique (on pourrait admettre plus facilement un certain laxisme devant un public mathématicien!).

Le cadre d'une induction est le suivant :

Soit E un ensemble muni d'une relation d'ordre strict **bien fondé, c'est–à–dire, sans suite décroissante infinie** (des algébristes évoqueraient Emil Artin ou Amalie Nœther), que l'on notera $\prec$.

Voici des exemples :

1) La relation $<$ sur l'ensemble $\mathbf{N}$ des entiers naturels, est un ordre strict bien fondé;
2) Toute application $h : E \rightarrow \mathbf{N}$ permet de définir la relation suivante sur E : $b \prec a$ ssi $h(b) < h(a)$, qui est un ordre strict bien fondé;
3) L'ensemble T des termes construits avec un ensemble P des symboles d'arité 0 et, pour fixer les idées, un symbole $\mathbf{u}$ d'arité 1 et un symbole $\mathbf{d}$ d'arité 2. Alors la relation $\prec$ définie par transitivité à partir de :
   - $t \prec \mathbf{u}\, t$, quel que soit $t \in T$;
   - $t_1 \prec \mathbf{d}\, t_1\, t_2$ et $t_2 \prec \mathbf{d}\, t_1\, t_2$, quels que soient $t_1 \in T$ et $t_2 \in T$;

   est un ordre strict bien fondé sur T;
4) La relation "*B est un sous–arbre strict de A*" est un ordre strict bien fondé sur tout ensemble d'arbres finis, qui contient tous les sous–arbres de ses éléments.

Le Principe d'induction.

Soit E un ensemble muni d'une relation d'ordre strict bien fondé $\prec$ et soit $P(x)$ l'énoncé d'une propriété des éléments de E, alors :

Lorsque pour tout $x \in E$, on peut déduire $P(x)$ de **l'hypothèse d'induction**

(HI) : pour tout $y \in E$, $y \prec x$ implique $P(y)$,

on peut en conclure $P(x)$ quel que soit $x \in E$.

On rédige généralement de la façon suivante :

Soit $x \in E$ et supposons que l'on ait $P(y)$ pour tout $y \prec x$, ... et il faut déduire $P(x)$.

Poser une hypothèse d'induction n'est pas exprimer sa croyance en un miracle, ou espérer un don du ciel! mais, en termes informatiques, faire des appels récursifs, sur des objets strictement plus petits que celui auquel on s'intéresse actuellement; ces appels ne manqueront certainement pas d'en faire à leur tour, jusqu'à ce que ce jeu se termine, puisque l'ordre est bien fondé.

Lorsque x est minimal pour $\prec$, l'hypothèse d'induction ne dit rien et il faut bien se résoudre à démontrer $P(x)$ dans ce cas!

*
*   *

Dans ce qui suit, des rappels seront faits de temps en temps, mais l'essentiel devra être puisé dans le cours lui–même!

CALCUL DES PROPOSITIONS.

Considérons un ensemble E muni des opérations définies par les applications :

$$\begin{aligned} neg &: E \to E \\ conj &: E \times E \to E \\ disj &: E \times E \to E \\ impl &: E \times E \to E \end{aligned}$$

La propriété de lecture unique s'applique de la façon suivante :

Pour toute application $f : \mathcal{P} \to E$ on peut construire une et une seule application $\underline{f} : \mathcal{F} \to E$ vérifiant les cinq conditions suivantes :

$$\underline{f}(p) = f(p)$$

pour toute variable propositionnelle p, et

$$\begin{aligned} \underline{f}(\neg A) &= neg(\underline{f}(A)) \\ \underline{f}((A \wedge B)) &= conj(\underline{f}(A), \underline{f}(B)) \\ \underline{f}((A \vee B)) &= disj(\underline{f}(A), \underline{f}(B)) \\ \underline{f}((A \to B)) &= impl(\underline{f}(A), \underline{f}(B)) \end{aligned}$$

pour toute formule A et toute formule B.

Ceci est une construction *par induction sur les formules* et l'application $\underline{f}$ ainsi construite s'appelle *l'extension de* f *aux formules* (relativement aux applications neg, $conj$, $disj$ et $impl$ en cause). L'existence et l'unicité d'une telle extension s'exprime souvent en disant que $\mathcal{F}$ est un objet *libre* dans sa catégorie.

En termes informatiques, cette construction correspond exactement à la *fonction récursive* qui s'écrit :

```
fonction f_souligne(X : formule) : E ;
    p : variable_propositionnelle ;
    A, B : formule ;
    selon la valeur de X faire
        p        : retourne f(p) ;
        ¬ A      : retourne neg(f_souligne(A)) ;
        (A ∧ B)  : retourne conj(f_souligne(A), f_souligne(B)) ;
        (A ∨ B)  : retourne disj(f_souligne(A), f_souligne(B)) ;
        (A → B)  : retourne impl(f_souligne(A), f_souligne(B)) ;
    fin
fin
```

La propriété de lecture unique assure que, pour chaque formule, un et un seul des cinq cas peut se présenter, et fournit un algorithme permettant, dans les quatre derniers, d'extraire les sous–formules nécessaires aux appels récursifs.

Exercice 1. Variables propositionnelles apparaissant dans une formule.

Donner une définition, par induction sur les formules, de l'ensemble $V(A)$ des variables propositionnelles qui apparaissent dans une formule A.

Il est clair, avec les notations ci–dessus, que $\underline{f}(A)$ ne dépend que des valeurs que prend f sur les éléments de $V(A)$.

Exercice 2. Des formules et des arbres.

$$\boxed{\begin{array}{c} * \\ * A_1 \ldots A_n * \\ * \end{array}}$$

Lorsque $n \neq 0$, où est–il naturel de poser un opérateur n–aire pour montrer qu'il s'applique à ses arguments et pas à d'autres?
Pour $n = 2$, qui est un cas très courant, la place la plus courante de l'opérateur se trouve entre ses arguments; mais l'écriture $A_1 * A_2$ est bien connue pour entraîner des ambiguïtés! et c'est la raison pour laquelle il est nécessaire de faire usage de parenthèses lorsque l'on utilise cette notation *en infixe*.

Dans le cas général, cette position centrale n'existe plus mais quatre positions se présentent, qui sont toutes excellentes : chacune d'elles, lorsqu'elle est systématiquement utilisée (c'est–à–dire pour tous les symboles), conduit à une écriture des termes qui n'est pas ambiguë.

– Poser l'opérateur au dessus (resp. en dessous) de ses arguments donne la *représentation arborescente* avec racine en haut (resp. en bas). Dans ces cas, le dessin de liens entre l'opérateur et ses arguments, n'est pas indispensable mais facilite grandement la lecture.

a) Donner une construction, par induction sur les formules, de l'arbre (avec, par exemple, la racine en haut) $\mathbf{a}(X)$ associé à une formule X.

b) Après avoir caractérisé les arbres obtenus par la construction précédente, donner une définition, par induction sur les arbres, de la formule $\mathbf{f}(\mathbf{A})$ associée à un arbre $\mathbf{A}$ convenable.

– Poser l'opérateur devant (resp. derrière) ses arguments donne l'écriture *polonaise préfixe* (resp. *suffixe*) : les qualités de l'écriture polonaise sont l'objet d'exercices classiques mais difficiles;

c) Donner une définition, par induction sur les formules, de la formule polonaise (par exemple préfixe) $\mathbf{p}(X)$ associé à une formule X.

La caractérisation des formules polonaises est un peu délicate et ne sera pas tentée ici!

Exercice 3.

a) Donner une définition, par induction sur les formules, des applications à valeurs entières suivantes :

$l(X) =$ la longueur de la formule X (compter tous les caractères, parenthèses comprises);
$n(X) =$ le nombre d'occurrences de symboles de négation intervenant dans la formule X;
$b(X) =$ le nombre d'occurrences de symboles binaires intervenant dans la formule X.

b) En déduire, en raisonnant par induction, que l'on a $l(X) = 4b(X) + n(X) + 1$ pour toute formule X.

Autres symboles d'usage courant

• Dans certaines circonstances, il est judicieux d'adopter deux nouveaux **symboles logiques primitifs d'arité** 0 : $\top$ (**le vrai**) et $\bot$ (**le faux**). Ce sont des formules (non–atomiques!) qui interviennent dans la construction générale des formules et pour lesquelles il faut ajouter les clauses $\underline{\delta}(\top) = 1$ et $\underline{\delta}(\bot) = 0$ à la définition de l'extension d'une distribution de valeurs de vérité δ.

Il est aussi possible, mais c'est moins intéressant, de les définir par $\top = (p_0 \vee \neg p_0)$ et $\bot = (p_0 \wedge \neg p_0)$, où p_0 est une variable propositionnelle fixée.

• **L'équivalence** est le symbole abréviateur $\leftrightarrow$ défini par $(A \leftrightarrow B) = ((A \to B) \wedge (B \to A))$. L'équivalence, dont l'utilité est indéniable, est trop "composée" pour prétendre à un statut d'opérateur logique primitif.

Exercice 4. Tautologies.

Si l'on utilise les applications

$$\begin{array}{rl} neg: & \{0,1\} \to \{0,1\} \text{ définie par } neg(x) = (1-x) \\ conj: & \{0,1\} \times \{0,1\} \to \{0,1\} \text{ définie par } conj(x,y) = xy \\ disj: & \{0,1\} \times \{0,1\} \to \{0,1\} \text{ définie par } disj(x,y) = x(1-y)+y = x+(1-x)y \\ impl: & \{0,1\} \times \{0,1\} \to \{0,1\} \text{ définie par } impl(x,y) = (1-x)+xy \end{array}$$

alors l'extension aux formules de toute distribution de valeurs de vérité $\delta : \mathcal{P} \to \{0,1\}$ est celle qui a été définie dans le cours.

On rappelle qu'une formule A est une *tautologie* lorsqu'elle est satisfaite par toute distribution de valeurs de vérité, c'est–à–dire lorsque $\underline{\delta}(A) = 1$ pour toute δ.

a) Montrer, par un calcul direct utilisant la définition de $\underline{\delta}$ à partir de δ, que les *formules choisies* sont des tautologies. Il pourra être utile d'observer que la propriété $x \in \{0,1\}$ est équivalente à $x(1-x) = 0$, c'est–à–dire à $x^2 = x$.

b) Montrer que les équivalences suivantes sont vraies, quelles que soient les formules A et B et la distribution de valeurs de vérité δ :

$$\begin{array}{l} \underline{\delta}((A \to B)) = 1 \text{ ssi } \underline{\delta}(A) \leq \underline{\delta}(B) \\ \underline{\delta}((A \leftrightarrow B)) = 1 \text{ ssi } \underline{\delta}(A) = \underline{\delta}(B) \end{array}$$

La dernière propriété peut servir à revisiter les *formules choisies* qui sont des équivalences!

On dit souvent que deux formules A et B sont *équivalentes* lorsque $(A \leftrightarrow B)$ est une tautologie, c'est–à–dire lorsque $\underline{\delta}(A) = \underline{\delta}(B)$ pour toute δ.

c) Montrer que la définition de $\underline{\delta}$ est en accord avec la logique naïve, c'est–à–dire que les équivalences suivantes sont vraies, quelles que soient les formules A et B et la distribution de valeurs de vérité δ :

$$\begin{array}{rl} \underline{\delta}(\neg A) = 1 & \text{ssi } \mathbf{non}\ \underline{\delta}(A) = 1 \qquad \text{(ce qui peut aussi s'écrire } \underline{\delta}(A) = 0!) \\ \underline{\delta}((A \wedge B)) = 1 & \text{ssi } \underline{\delta}(A) = 1 \ \mathbf{et}\ \underline{\delta}(B) = 1 \\ \underline{\delta}((A \vee B)) = 1 & \text{ssi } \underline{\delta}(A) = 1 \ \mathbf{ou}\ \underline{\delta}(B) = 1 \\ \underline{\delta}((A \to B)) = 1 & \text{ssi } \underline{\delta}(A) = 1 \Rightarrow \underline{\delta}(B) = 1 \end{array}$$

Utiliser cette "traduction" pour montrer que les *formules choisies* sont des tautologies.

Exercice 5. Substitutions.

L'ensemble $\mathcal{F}$ des formules est naturellement muni des opérations :

$$\begin{array}{rl} neg: & \mathcal{F} \to \mathcal{F} \text{ définie par } neg(A) = \neg A \\ conj: & \mathcal{F} \times \mathcal{F} \to \mathcal{F} \text{ définie par } conj(A,B) = (A \wedge B) \\ disj: & \mathcal{F} \times \mathcal{F} \to \mathcal{F} \text{ définie par } disj(A,B) = (A \vee B) \\ impl: & \mathcal{F} \times \mathcal{F} \to \mathcal{F} \text{ définie par } impl(A,B) = (A \to B) \end{array}$$

Soit maintenant $s : \mathcal{P} \to \mathcal{F}$ une application et désignons par $\underline{s}$ son extension aux formules.

a) Vérifier que pour toute formule A, la formule $\underline{s}(A)$ est obtenue en substituant $s(p)$ à chaque variable propositionnelle p apparaissant dans A : pour cette raison, on dit souvent qu'une application du type de s est *une substitution*.

b) Soit δ une distribution de valeurs de vérité et considérons la nouvelle distribution de valeurs de vérité δ' définie par $\delta'(p) = \underline{\delta}(s(p))$.

Montrer, par induction sur les formules, que l'on a $\underline{\delta}'(A) = \underline{\delta}(\underline{s}(A))$ pour toute formule A (pour comprendre ce qui se passe, il pourra être intéressant de considérer la représentation arborescente des formules).

En déduire qu'une substitution transforme une tautologie en une tautologie.

Formules choisies

1. $(A \to A)$

2. $(A \to (A \wedge A))$
3. $((A \wedge B) \to A)$
4. $((A \wedge \bot) \leftrightarrow \bot)$
5. $((A \wedge \top) \leftrightarrow A)$
6. $((A \wedge (B \wedge C)) \leftrightarrow ((A \wedge B) \wedge C))$
7. $((A \wedge B) \leftrightarrow (B \wedge A))$

8. $((A \vee A) \to A)$
9. $(A \to (A \vee B))$
10. $((A \vee \top) \leftrightarrow \top)$
11. $((A \vee (B \vee C)) \leftrightarrow ((A \vee B) \vee C))$
12. $((A \vee B) \leftrightarrow (B \vee A))$
13. $((A \vee \bot) \leftrightarrow A)$

14. $(\neg\neg A \leftrightarrow A)$

15. $(\neg(A \wedge B) \leftrightarrow (\neg A \vee \neg B))$
16. $(\neg(A \vee B) \leftrightarrow (\neg A \wedge \neg B))$
17. $(\neg(A \to B) \leftrightarrow (A \wedge \neg B))$
18. $(\neg A \leftrightarrow (A \to \bot))$
19. $(A \vee \neg A)$
20. $\neg(A \wedge \neg A)$

21. $((A \to B) \leftrightarrow (\neg A \vee B))$
22. $((A \to B) \leftrightarrow (\neg B \to \neg A))$
23. $((A \to (B \to C)) \leftrightarrow ((A \wedge B) \to C)$

24. $((A \wedge (B \vee C)) \leftrightarrow ((A \wedge B) \vee (A \wedge C)))$
25. $((A \vee (B \wedge C)) \leftrightarrow ((A \vee B) \wedge (A \vee C)))$
26. $((A \to (B \wedge C)) \leftrightarrow ((A \to B) \wedge (A \to C)))$
27. $((A \to (B \vee C)) \leftrightarrow ((A \to B) \vee (A \to C)))$
28. $((A \to (B \to C)) \leftrightarrow ((A \to B) \to (A \to C)))$

29. $(((A \wedge B) \to C) \leftrightarrow ((A \to C) \vee (B \to C)))$
30. $(((A \vee B) \to C) \leftrightarrow ((A \to C) \wedge (B \to C)))$

REMARQUE. Si A est une formule et p une variable propositionnelle, on dira souvent "*Soit $A[p]$ une formule dans laquelle p apparaît éventuellement*". Cette expression ne dit rien sur A! mais insiste sur l'intérêt que l'on porte à p et introduit une notation suggestive : si X est une formule, on peut en effet considérer la substitution définie par

$$s(q) = \begin{cases} X & \text{si } q = p, \\ q & \text{sinon.} \end{cases}$$

On note alors $A[X]$ au lieu de $\underline{s}(A)$. Il est évidemment possible d'adapter cette notation au cas d'une suite finie de variables.

Problème 6. Fonctions booléennes et formes normales disjonctives

Une *fontion booléenne* d'arité n est une application $f : \{0,1\}^n \to \{0,1\}$: une telle fonction s'applique donc aux suites $(x_1, \ldots, x_n)$ formées de n valeurs 0 ou 1 et pour chacune de ces suites, $f(x_1, \ldots, x_n)$ prend la valeur 0 ou la valeur 1.

Le but de cet exercice est de montrer qu'une fonction booléenne est la "table de vérité" d'une formule : comme conséquence on obtiendra le calcul simple d'une *forme normale disjonctive* d'une formule dont on connaît la table de vérité.

a) Commençons par un peu d'algèbre (booléenne, comme il se doit).
Une fonction booléenne f étant donnée, il y a deux cas à considérer :

- si l'arité de f est 0, on a ou bien $f() = 0$ ou bien $f() = 1$;
- sinon, pour $f : \{0,1\}^{n+1} \to \{0,1\}$, on définit les deux fonctions booléennes f_0 et $f_1 : \{0,1\}^n \to \{0,1\}$ par :

$$f_0(x_1,\ldots,x_n) = f(x_1,\ldots,x_n,0)$$
$$f_1(x_1,\ldots,x_n) = f(x_1,\ldots,x_n,1)$$

Vérifier que l'on a

$$f(x_1,\ldots,x_n,x_{n+1}) = f_0(x_1,\ldots,x_n)\,\overline{x}_{n+1} + f_1(x_1,\ldots,x_n)\,x_{n+1}$$

(où, pour tout $x \in \{0,1\}, \overline{x}$ désigne l'expression $(1-x)$, que l'on a vraiment intérêt à laisser telle quelle) pour toute suite $(x_1,\ldots,x_n,x_{n+1}) \in \{0,1\}^{n+1}$.

Appelons *monôme booléen* un produit $y_1\ldots y_n$ dans lequel chaque i, y_i est ou bien x_i ou bien $\overline{x}_i$ (un tel produit vaut 1 lorsque $n=0$).
Montrer que toute fonction booléenne est égale à une somme de monômes booléens (éventuellement 0) : on pourra commencer par étudier la fonction dont la table est dressée ci–contre.

x_1	x_2	x_3	$f(x_1,x_2,x_3)$
0	0	0	1
1	0	0	0
0	1	0	0
1	1	0	1
0	0	1	1
1	0	1	0
0	1	1	0
1	1	1	0

b) Maintenant, on associe une formule $\tilde{f}$ du calcul des propositions à chaque fonction booléenne f, par récurrence sur l'arité de f, de la façon suivante :

- si cette arité est 0 alors, on pose
  $\tilde{f} = \bot$ lorsque $f() = 0$ et $\tilde{f} = \top$ lorsque $f() = 1$;
- sinon, pour $f : \{0,1\}^{n+1} \to \{0,1\}$ on pose
  $\tilde{f} = ((\tilde{f}_0 \wedge \neg p_{n+1}) \vee (\tilde{f}_1 \wedge p_{n+1}))$.

(On a évidemment adopté $\top$ et $\bot$ comme symboles primitifs.) Montrer que pour toute distribution de valeurs de vérité δ et toute fonction booléenne f d'arité n, on a $\underline{\delta}(\tilde{f}) = f(\delta(p_1),\ldots,\delta(p_n))$.

c) Soit A une formule dont on connaît la *table de vérité*. Appliquer ce qui précède pour calculer une *forme normale disjonctive* équivalente à A.

Rappels :

- *Un littéral* est une formule p ou $\neg p$ où p est une variable propositionnelle.
- *Une conjonction de littéraux* est une conjonction $l_1 \wedge \ldots \wedge l_n$ (où on néglige d'écrire les parenthèses) où chaque l_i est un littéral (cette conjonction vaut $\top$ lorsque $n=0$).
- Une forme normale disjonctive est une disjonction $c_1 \vee \ldots \vee c_n$ (où on néglige d'écrire les parenthèses) où chaque c_i est une conjonction de littéraux (cette disjonction vaut $\bot$ lorsque $n=0$).

Problème 7. Lemme d'interpolation.

Soient p et q deux variables propositionnelles.

a) Soit $A[p]$ une formule dans laquelle p apparaît éventuellement et considérons les deux formules $A_0 = A[\neg q]$ et $A_1 = A[q]$ obtenues par des substitutions.

Montrer que $(A[p] \to (A_0 \vee A_1))$ est une tautologie.

b) Soit de plus B une formule dans laquelle p **n'apparaît pas** et telle que $(A[p] \to B)$ soit une **tautologie**.

Montrer que $((A_0 \vee A_1) \to B)$ est une tautologie.

Indication. Si δ est une distribution de valeurs de vérité qui satisfait, par exemple, A_1, considérer la distribution de valeurs de vérité δ' qui coïncide avec δ sauf éventuellement en p, où l'on a $\delta'(p) = \delta(q)$.

c) Pour énoncer le lemme , il est commode d'adopter l'usage des symboles logiques primitifs $\top$ et $\perp$.

Montrer le *lemme d'interpolation* que voici :

Soient A et B des formules telles que $(A \to B)$ est une tautologie alors, il existe une formule C (dite "interpolante entre A et B") qui vérifie les propriétés suivantes :

- *$(A \to C)$ est une tautologie;*
- *$(C \to B)$ est une tautologie;*
- *toute variable propositionnelle apparaissant dans C apparaît aussi dans A et dans B.*

Lorsque A et B ont une variable commune, on peut faire un raisonnement par récurrence sur le nombre de variables propositionnelles apparaissant dans A mais pas dans B, en appliquant le résultat des questions précédentes.

Problème 8. Théorème de compacité.

Pour tout ensemble $\mathcal{A}$ de formules du calcul propositionnel, on pose les définitions suivantes :

- une distribution de valeurs de vérité δ *satisfait* $\mathcal{A}$ ssi $\underline{\delta}(A) = 1$ pour toute $A \in \mathcal{A}$;
- $\mathcal{A}$ *est satisfaisable* ssi il existe une distribution de valeurs de vérité δ qui satisfait $\mathcal{A}$;
- $\mathcal{A}$ *est finiment satisfaisable* ssi toute *partie finie* $\mathcal{B} \subseteq \mathcal{A}$ est satisfaisable (ceci signifie bien que, pour toute partie finie $\mathcal{B} \subseteq \mathcal{A}$, il y a une distribution de valeurs de vérité $\delta_{\mathcal{B}}$, **qui dépend de** $\mathcal{B}$ et qui satisfait $\mathcal{B}$).

Le but de cet exercice est la démonstration du *théorème de compacité* dont voici l'énoncé :

Pour tout ensemble $\mathcal{A}$ de formules du calcul propositionnel, les deux propriétés suivantes sont équivalentes :

1) $\mathcal{A}$ est satisfaisable,
2) $\mathcal{A}$ est finiment satisfaisable.

Il est clair que 1) implique 2); de même, 2) implique trivialement 1) lorsque $\mathcal{A}$ est fini (puisqu'alors, toute partie de $\mathcal{A}$ est finie). Il reste à démontrer que 2) implique 1) lorsque l'on ne suppose pas que $\mathcal{A}$ est fini.

Soit donc $\mathcal{A}$ un ensemble finiment satisfaisable.

a) Soit p une variable propositionnelle quelconque.

Montrer que l'un des deux ensembles $\mathcal{A} \cup \{p\}$ ou $\mathcal{A} \cup \{\neg p\}$ est finiment satisfaisable.

Indication : En supposant que $\mathcal{A} \cup \{p\}$ n'est pas finiment satisfaisable (c'est–à–dire, qu'il existe une partie finie $\mathcal{B} \subseteq \mathcal{A} \cup \{p\}$ qui n'est pas satisfaisable) montrer que $\mathcal{A} \cup \{\neg p\}$ est finiment satisfaisable.

b) Soit $P = \{p_1, \ldots, p_n, \ldots\}$ une énumération de l'ensemble des variables propositionnelles.

On considère la suite $(\mathcal{A}_n)_{n \in \mathbf{N}}$ d'ensembles définie par la récurrence :

$$\mathcal{A}_0 = \mathcal{A}$$

$$\mathcal{A}_{n+1} = \begin{cases} \mathcal{A}_n \cup \{p_{n+1}\} & \text{si } \mathcal{A}_n \cup \{p_{n+1}\} \text{ est finiment satisfaisable,} \\ \mathcal{A}_n \cup \{\neg p_{n+1}\} & \text{sinon.} \end{cases}$$

Montrer que chacun des $\mathcal{A}_n$ est finiment satisfaisable.

c) On définit, par récurrence, les ensembles l_n de littéraux suivants :

$$l_0 = \emptyset$$

$$l_{n+1} = \begin{cases} l_n \cup \{p_{n+1}\} & \text{si } p_{n+1} \in \mathcal{A}_{n+1}, \\ l_n \cup \{\neg p_{n+1}\} & \text{sinon.} \end{cases}$$

Montrer que $l_n \subseteq \mathcal{A}_n$ pour tout n.

d) On définit, par récurrence, la distribution de valeurs de vérité λ :

$$\lambda(p_n) = \begin{cases} 1 & \text{si } p_n \in \mathcal{A}_n, \\ 0 & \text{sinon.} \end{cases}$$

Montrer que $\underline{\lambda}(A) = 1$ pour toute $A \in \mathcal{A}$ et donc que $\mathcal{A}$ est satisfaisable.

On pourra considérer un entier n assez grand pour que toutes les variables apparaissant dans A soient parmi $\{p_1, \ldots, p_n\}$ et se souvenir que $l_n \cup \{A\} \subseteq \mathcal{A}_n$.

SYSTÈME LK (Calcul propositionnel classique).

Identité

$$\frac{}{p \vdash p}$$

Coupure

$$\frac{\Gamma \vdash \Delta; A \quad A, \Lambda \vdash \Pi}{\Gamma, \Lambda \vdash \Delta; \Pi}$$

Règles structurelles

(E_d)	$\dfrac{\Gamma \vdash \Delta}{\Gamma \vdash \sigma(\Delta)}$ pour toute permutation σ	$\dfrac{\Gamma \vdash \Delta}{\sigma(\Gamma) \vdash \Delta}$ pour toute permutation σ	(E_g)
(C_d)	$\dfrac{\Gamma \vdash \Delta; A; A}{\Gamma \vdash \Delta; A}$	$\dfrac{A, A, \Gamma \vdash \Delta}{A, \Gamma \vdash \Delta}$	(C_g)
(A_d)	$\dfrac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta; A}$	$\dfrac{\Gamma \vdash \Delta}{A, \Gamma \vdash \Delta}$	(A_g)

Règles logiques

$(\neg\_d)$	$\dfrac{A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta; \neg A}$	$\dfrac{\Gamma \vdash \Delta; A}{\neg A, \Gamma \vdash \Delta}$	$(\neg\_g)$
$(\wedge\_d)$	$\dfrac{\Gamma \vdash \Delta; A \quad \Lambda \vdash \Pi; B}{\Gamma, \Lambda \vdash \Delta; \Pi; (A \wedge B)}$	$\dfrac{A, B, \Gamma \vdash \Delta}{(A \wedge B), \Gamma \vdash \Delta}$	$(\wedge\_g)$
$(\vee\_d)$	$\dfrac{\Gamma \vdash \Delta; A; B}{\Gamma \vdash \Delta; (A \vee B)}$	$\dfrac{A, \Gamma \vdash \Delta \quad B, \Lambda \vdash \Pi}{(A \vee B), \Gamma, \Lambda \vdash \Delta; \Pi}$	$(\vee\_g)$
$(\rightarrow\_d)$	$\dfrac{A, \Gamma \vdash \Delta; B}{\Gamma \vdash \Delta; (A \rightarrow B)}$	$\dfrac{\Gamma \vdash \Delta, A \quad B, \Lambda \vdash \Pi}{(A \rightarrow B), \Gamma, \Lambda \vdash \Delta; \Pi}$	$(\rightarrow\_g)$

CALCUL DES SÉQUENTS PROPOSITIONNEL : LK.

Sauf mention explicite du contraire, ”prouvable” signifie ”prouvable dans LK”.

Les ”axiomes” sont figurés ici par une règle sans prémisse, **une preuve dans LK se présente sous la forme d'un arbre dont toutes les feuilles sont vides**. Rappelons qu'on dit d'une formule X que c'est un *théorème* ou plus simplement qu'elle est *valide*, lorsque $\vdash X$ est prouvable.

La propriété fondamentale est le **Théorème d'élimination des coupures pour LK**

Tout séquent prouvable admet une preuve n'utilisant aucune coupure.

Les coupures dont il est question dans cet énoncé sont dites *logiques*.

On peut considérer des arbres dont les embranchements sont bien des applications de règles de **LK**, mais dont les feuilles ne sont pas toutes vides : soit Σ un ensemble de séquents alors, un arbre dont les feuilles non vides sont des éléments de Σ s'appellera *une déduction à partir de* Σ. Le théorème d'élimination des coupures s'applique encore à ce cas mais, il ne dit rien sur les *coupures propres* (à Σ), c'est–à–dire, celles dont la formule coupée provient sans modification d'une formule d'un élément de Σ, dans l'une au moins des prémisses : *Tout séquent déductible d'un ensemble* Σ *admet une déduction dont les seules coupures sont propres.*

Exercice 9. Les axiomes d'identité et les substitutions.

Les axiomes d'identité sont posés pour les seules variables propositionnelles : montrer, par induction sur les formules, que le séquent $X \vdash X$ est prouvable. En déduire qu'une substitution transforme une formule valide en une formule valide.

Dans la pratique, on considère souvent tout séquent $X \vdash X$ comme étant un axiome d'identité!

Exercice 10.

Construire une preuve du séquent $\vdash X$ pour chacune des *formules choisies* X.

Exercice 11. Les séquents et les formules.

a) Montrer que les règles logiques unaires (c'est–à–dire à une seule prémisse) sont inversibles. Plus précisément, par exemple pour le connecteur $\wedge$, montrer que l'on peut déduire $A, B, \Gamma \vdash \Delta$ à partir de $(A \wedge B), \Gamma \vdash \Delta$.

b) Soit S un séquent distinct du séquent vide ($\vdash$) et soit $\Phi(S)$ l'ensemble des formules X telles que l'on peut déduire $\vdash X$ de S en n'appliquant que des règles logiques unaires et des échanges.

Montrer que, réciproquement, on peut déduire S de $\vdash X$, pour toute $X \in \Phi(S)$.

$,$	$\vdash$	$;$
$\wedge$	$\to$	$\vee$

Bien que ça ne soit pas très ”académique”, il est commode de penser à un séquent S comme à une représentation un peu assouplie de l'une quelconque des formules $X \in \Phi(S)$. La petite table ci–contre résume ces correspondances, si on se souvient de plus qu'une négation $\neg$ signale le passage d'un membre à l'autre du séquent. Notons au passage que l'utilisation du *point virgule* comme séparateur dans le membre de droite des séquents, qui n'est pas traditionnelle, permet d'écrire la table ci–contre et surtout, peut décourager les meilleurs esprits à faire des usages frauduleux de la coupure (que l'on n'a pas à publier ici).

Exercice 12. Inversiblité des règles logiques binaires.

Les règles logiques binaires (c'est–à–dire à deux prémisses) ne sont pas inversibles telles quelles car il est généralement impossible de répartir les contextes! Fort judicieusement, les règles structurelles sont là pour les modifier de telle façon que ce problème ne se pose plus. Plus précisément :

a) Montrer que l'on peut déduire $\Gamma \vdash \Delta; A$ et $\Gamma \vdash \Delta; B$ à partir de $\Gamma \vdash \Delta; (A \wedge B)$, et réciproquement (les contextes Γ et Δ sont les mêmes dans ces trois séquents).

Une conséquence simple de ce résultat est que l'ensemble des deux séquents $\vdash X$ et $\vdash Y$, et le séquent $\vdash (X \wedge Y)$ peuvent se déduire l'un de l'autre.

b) Enoncer et démontrer les résultats analogues pour les connecteurs $\vee$ et $\rightarrow$.

Exercice 13. Satisfaction des séquents.

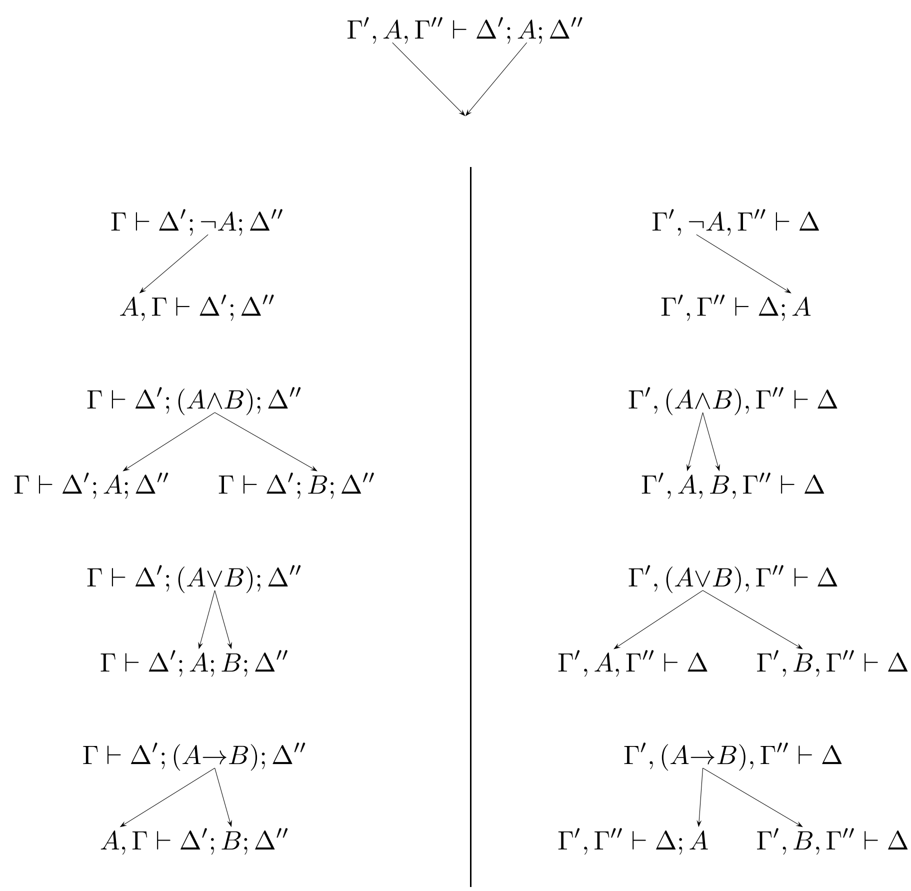
On dit qu'une distribution de valeurs de vérité δ *satisfait* le séquent $\Gamma \vdash \Delta$ ssi lorsque δ satisfait **toutes** les formules de Γ, alors δ satisfait aussi **au moins une** formule de Δ. Par exemple, un séquent de la forme $X \vdash X$ est satisfait par toute distribution de valeurs de vérité, mais *le séquent vide* $\vdash$ ne l'est par aucune.

a) Montrer que lorsque δ satisfait la ou les prémisses d'une règle du système LK alors elle satisfait aussi sa conclusion (on n'oubliera pas le cas de la coupure).

REMARQUE. Ceci implique évidemment que si δ satisfait un ensemble Σ de séquents, alors elle satisfait aussi tout séquent que l'on peut en déduire.

b) Transposer et démontrer les résultats d'inversibilité des règles logiques, en termes de satisfaction d'ensembles de séquents.

Un système de décomposition pour LK



Le système de décomposition pour LK représente de façon graphique les résultats d'inversibilité des règles démontrés dans les exercices précédents. L'application de l'une des règles fait disparaître

- ou bien un séquent (la première règle);
- ou bien un opérateur logique.

Si donc on applique ces règles **tant qu'il est possible**, on transforme un séquent S en un ensemble, que l'on notera $\mathbf{Dec}(S)$, de séquents *atomiques*, c'est–à–dire, constitués uniquement de variables propositionnelles (il est clair que $\mathbf{Dec}(S)$ n'est pas défini de façon univoque car il peut dépendre de l'ordre dans lequel on a appliqué les règles de décomposition).

Dans la pratique, on ajoute les règles suivantes, que l'on peut considérer, si l'on veut, comme des contractions. Par contre, les règles de réduction sont écrites de façon à éviter l'application de tout échange, afin d'écarter le risque fatal de rentrer dans une suite sans fin d'applications de cette opération.

"Contractions"

$$\begin{array}{c} \Gamma \vdash \Delta; A; \Delta'; A; \Delta'' \\ \downarrow \\ \Gamma \vdash \Delta; A; \Delta'; \Delta'' \end{array} \qquad\qquad \begin{array}{c} \Gamma, A, \Gamma', A, \Gamma'' \vdash \Delta \\ \downarrow \\ \Gamma, \Gamma', A, \Gamma'' \vdash \Delta \end{array}$$

Problème 14. Formes normales conjonctives et méthode de résolution.

a) Montrer comment calculer, à partir de $\mathbf{Dec}(\ \vdash X)$, *une forme normale conjonctive* (duale d'une forme normale disjonctive) équivalente à la formule X.

La pratique montre rapidement que ce calcul d'une forme normale conjonctive de X est plus aisé que le calcul "classique" et même, que $\mathbf{Dec}(\ \vdash X)$ est bien plus maniable que cette forme normale!

b) Montrer qu'un séquent S est prouvable ssi $\mathbf{Dec}(S) = \emptyset$.

REMARQUE. Ce résultat donne un **algorithme** de prouvabilité du calcul propositionnel, mais cet algorithme est d'allure exponentielle, car un arbre de décomposition admet des embranchements binaires. En fait, le problème est connu pour être *NP*–complet.

c) Montrer, en appliquant le théorème de complétude, que si l'on peut déduire le séquent vide $\vdash$ à partir de $\mathbf{Dec}(X \vdash\ )$ alors la formule X est valide.

Cette façon, assez indirecte, de montrer qu'une formule est valide est connue sous le nom de **résolution** : elle sera surtout utile dans le cas du calcul des prédicats, pour lequel elle a donné lieu à de nombreux travaux dans les milieux informatiques.

REMARQUE. D'après ce qui a été dit, au début de cette section, au sujet du théorème d'élimination des coupures, une déduction sans coupure logique du séquent vide $\vdash$ à partir de $\mathbf{Dec}(X \vdash\ )$ peut contenir des coupures propres à cet ensemble.

En partant de l'idée que, lors d'une telle déduction, **toute formule doit disparaître**, il est facile de voir que les seules règles utiles sont :

– des coupures;
– des contractions;

et, pour mettre les formules dans la position nécessaire à l'application de ces règles, des échanges.

d) Montrer par résolution que les *formules choisies* sont valides.

Exercice 15.

Soit X une formule.

a) Montrer, par induction sur les déductions, que si l'on peut déduire le séquent $\Gamma \vdash \Delta$ de $\vdash X$ alors $X, \Gamma \vdash \Delta$ est prouvable.

b) Donner une justification de la méthode de résolution n'utilisant que la notion de prouvabilité (alors que la justification précédente utilise aussi la notion de satisfaisabilité).

Problème 16. Théorème de complétude.

Le but de ce problème est de redémontrer le *Théorème de complétude* pour le calcul des propositions :

Si A est une tautologie, alors $\vdash A$ est prouvable dans **LK**.

Posons d'abord une définition : soient A une formule et δ une distribution de valeurs de vérité, alors, la formule A^δ est définie par :

$$A^\delta = \begin{cases} A & \text{si } \underline{\delta}(A) = 1, \\ \neg A & \text{sinon.} \end{cases}$$

a) Montrer, par induction sur la formule A, que le séquent $p_1^\delta, \ldots, p_m^\delta \vdash A^\delta$ est prouvable dans **LK**, quelle que soit δ, lorsque les variables propositionnelles apparaissant dans A sont parmi $p_1, \ldots, p_m$.

Indication. On simplifiera sensiblement la démonstration en montrant que, pour tout opérateur binaire $* \in \{\wedge, \vee, \rightarrow\}$, le séquent $A^\delta, B^\delta \vdash (A * B)^\delta$ est prouvable dans **LK**.

b) Soit A **une tautologie** dont les variables propositionnelles sont parmi $p_1, \ldots, p_m$.

Montrer que le séquent $p_1^\delta, \ldots, p_i^\delta \vdash A$ est prouvable dans **LK** pour tout entier naturel $i \leq m$, quelle que soit δ. En déduire le théorème de complétude.

Le vrai et le faux

Lorsque l'on veut adopter $\top$ et $\bot$ comme symboles logiques primitifs, il faut compléter **LK** par les règles :

$(\top\_d)$ $\quad \dfrac{}{\vdash \top}$ $\qquad\qquad$ $\dfrac{}{\bot \vdash}$ $\quad (\bot\_g)$

Problème 17. Lemme d'interpolation.

La question de l'inversion de la règle de coupure ne se pose pas car, lorsque l'on ne connaît que la conclusion d'une telle règle, on ignore tout d'une éventuelle formule de coupure A ... Cependant, on peut montrer que tout séquent prouvable admet une preuve se terminant par une coupure, sur la formule de laquelle on a un certain contrôle!

Pour énoncer le résultat proprement, on adopte $\top$ et $\bot$ comme symboles logiques primitifs, avec les règles ci–dessus.

a) Plus précisément, montrer que pour tout séquent prouvable $\Gamma_1, \Gamma_2 \vdash \Delta_1; \Delta_2$ il existe une formule C vérifiant les propriétés suivantes :

– $\Gamma_1 \vdash \Delta_1; C$ est prouvable;
– $C, \Gamma_2 \vdash \Delta_2$ est prouvable;
– toute variable propositionnelle apparaissant dans C apparaît aussi dans $\Gamma_1 \vdash \Delta_1$ et dans $\Gamma_2 \vdash \Delta_2$.

La démonstration se fait par induction sur les preuves sans coupure : le nombre de cas à considérer est assez important!

b) En déduire le lemme d'interpolation qui a déjà été démontré dans un exercice précédent par d'autres moyens.

Logique intuitionniste et système LJ.

A côté de la logique classique, objet principal du cours, se trouve la *logique intuitionniste* qui, elle, n'exclut pas le tiers, c'est–à–dire dans laquelle la formule $(A \vee \neg A)$ n'est généralement pas prouvable. La restriction essentielle qui distingue les séquents du système intuitionniste **LJ** de ceux du système **LK** porte sur la forme des séquents $\Gamma \vdash A$ où le second membre A est réduit à une formule et une seule.

Les formules du calcul des propositions intuitionniste sont construites, à partir d'un ensemble de variables propositionnelles et de la constante **0**, par application des symboles binaires $\wedge$, $\vee$ et $\rightarrow$.

La négation intuitionniste est un symbole abréviateur défini par $\neg A = (A \rightarrow \mathbf{0})$. L'introduction de ce symbole permet de considérer une formule classique comme étant aussi une formule intuitionniste : **une formule intervenant dans les deux systèmes sera nécessairement classique** mais, il y a évidemment des formules intuitionnistes qui ne sont pas classiques!

L'introduction de **0** a pour rôle essentiel de simplifier l'écriture de la règle $(\vee\_g)$ pour la disjonction à gauche : on aura intérêt à regarder attentivement les règles relatives à la disjonction!

On admettra le **Théorème d'élimination des coupures pour LJ** (mais il est intéressant de regarder comment se transposent les cas–clefs dont il a été question dans le cours au sujet de **LK**) :

Tout séquent prouvable dans **LJ** *y admet une preuve n'utilisant aucune coupure.*

Problème 18. Calcul des propositions intuitionniste.

a) Construire une preuve dans **LJ** de chacun des séquents suivants :

$X \vdash \neg\neg X$
$\neg\neg\neg X \vdash \neg X$
$\neg\neg(X \wedge Y) \vdash (\neg\neg X \wedge \neg\neg Y)$
$\neg\neg(X \rightarrow Y) \vdash (\neg\neg X \rightarrow \neg\neg Y)$

b) Les axiomes de l'identité sont posés pour les seules variables propositionnelles : montrer, par induction sur les formules, que le séquent $X \vdash X$ est prouvable dans **LJ** quelle que soit X.

c) Montrer, par induction sur les preuves, la propriété suivante pour toute Γ constituée de formules classiques et toute formule classique A :

si $\Gamma \vdash A$ (resp. $\Gamma \vdash \mathbf{0}$) est prouvable dans **LJ** alors $\Gamma \vdash A$ (resp. $\Gamma \vdash$) est prouvable dans **LK**.

d) La réciproque du résultat précédent n'est pas vraie! Par exemple, le séquent $\neg\neg X \vdash X$ n'est pas prouvable pour toute formule X (faire une tentative de preuve en prenant une variable propositionnelle). Voici la propriété essentielle de la logique intuitionniste (*constructivité* de la disjonction intuitionniste) :

Montrer que si $\vdash (X \vee Y)$ est prouvable dans **LJ** alors l'un des deux séquents $\vdash X$ ou $\vdash Y$ l'est aussi.

En déduire par exemple que $\vdash (X \vee (X \rightarrow Y))$ et $\vdash (X \vee \neg X)$ ne sont généralement pas prouvables dans **LJ** (alors que de tels séquents le sont toujours dans **LK**).

e) Etudier la prouvabilité dans **LJ** des implications constituant les *formules choisies* 15 à 23 (sauf évidemment 18!).

Problème 19. Traduction par double négation (Gödel).

Une formule A est dite *spéciale* ssi le séquent $\neg\neg A \vdash A$ est prouvable dans **LJ**. Par exemple, toute formule de la forme $\neg X$ est spéciale, d'après un résultat précédent.

a) Montrer que, lorsque A et B sont spéciales, les séquents suivants sont prouvables dans **LJ** :

$(\neg\neg A \wedge \neg\neg B) \vdash (A \wedge B)$
$(\neg\neg X \rightarrow \neg\neg B) \vdash (X \rightarrow B)$ (pour X quelconque)

b) On associe à toute formule classique A la formule ${}^sA$ (la notation traditionnelle $A^{\neg\neg}$ n'est faite pour simplifier ni l'écriture ni la lecture!), définie par l'induction suivante :

SYSTÈME LJ (Calcul propositionnel intuitionniste).

Les séquents ont la forme $\Gamma \vdash A$ *où* A *est* **une seule** *formule.*

Identité

$$\frac{}{p \vdash p}$$

Coupure

$$\frac{\Gamma \vdash A \quad A, \Lambda \vdash B}{\Gamma, \Lambda \vdash B}$$

Règles structurelles

$$\frac{\Gamma \vdash B}{A, \Gamma \vdash B} \quad (A\_g)$$

$$\frac{A, A, \Gamma \vdash B}{A, \Gamma \vdash B} \quad (C\_g)$$

$$\frac{\Gamma \vdash C}{\sigma(\Gamma) \vdash C} \quad (E\_g)$$

pour toute permutation σ

Règles logiques

$$\frac{}{\mathbf{0} \vdash A} \quad (\mathbf{0}\_g)$$

$$(\wedge\_d) \quad \frac{\Gamma \vdash A \quad \Lambda \vdash B}{\Gamma, \Lambda \vdash (A \wedge B)} \qquad\qquad \frac{A, B, \Gamma \vdash C}{(A \wedge B), \Gamma \vdash C} \quad (\wedge\_g)$$

$$(\vee B\_d) \quad \frac{\Gamma \vdash A}{\Gamma \vdash (A \vee B)}$$

$$\frac{A, \Gamma \vdash C \quad B, \Lambda \vdash C}{(A \vee B), \Gamma, \Lambda \vdash C} \quad (\vee\_g)$$

$$(A\vee\_d) \quad \frac{\Gamma \vdash B}{\Gamma \vdash (A \vee B)}$$

$$(\rightarrow\_d) \quad \frac{A, \Gamma \vdash B}{\Gamma \vdash (A \rightarrow B)} \qquad\qquad \frac{\Gamma \vdash A \quad B, \Lambda \vdash C}{(A \rightarrow B), \Gamma, \Lambda \vdash C} \quad (\rightarrow\_g)$$

On définit la négation par $\neg A = (A \rightarrow \mathbf{0})$.

${}^s p = \neg\neg p$ pour toute variable propositionnelle p
${}^s\neg A = \neg\, {}^s A$
${}^s(A \wedge B) = ({}^s A \wedge {}^s B)$
${}^s(A \vee B) = \neg\neg({}^s A \vee {}^s B)$
${}^s(A \to B) = ({}^s A \to {}^s B)$

Montrer que ${}^s A$ est spéciale pour toute formule classique A.

c) Montrer que lorsque le séquent $\Gamma \vdash \Delta$ est prouvable dans **LK** alors $\neg\, {}^s\Delta, \Gamma \vdash \mathbf{0}$ est prouvable dans **LJ**. (Pour $\Delta = B_1; \ldots; B_n$, on a posé $\neg\, {}^s\Delta = \neg\, {}^s B_1, \ldots, \neg\, {}^s B_n$.)

d) Montrer que $\vdash A$ *est prouvable dans* **LK** *ssi* $\vdash {}^s A$ *est prouvable dans* **LJ**, quelle que soit la formule classique A.