LOGIQUES MATHEMATIQUES

ENSIBS Informatique et cybersécurité
3ème année

2023-2024

[Mazen EL-SAYED](mailto:mazen.el-sayed@univ-ubs.fr)
mazen.el-sayed@univ-ubs.fr

Contenu du cours

– Logique des propositions
– Logique du 1er ordre
– Outillage : Prover9

Organisation du cours

- 28h de cours (CM/TD/TP)
- 2 DS

Pas de prérequis, mais rythme élevé, présence impérative à tous les cours,

LOGIQUE Introduction - Motivations

- **Les logiques constituent l'un des piliers majeurs de l'IA**
  ≈ approche **formelle** (par opposition aux approches quantitatives/statistiques, i.e. machine learning)

- **Les logiques sont utilisées pour :**
  - **Modéliser de manière formelle des « objets » de l'univers étudié**
  - **Raisonner** sur les modèles de l'univers étudié

- La logique est à la base de l'étude des **raisonnements**, c'est à dire des inférences que l'on peut faire à partir des modèles

- **Bénéfices à l'utilisation des formalismes logiques :**
  - Modèles précis et démarche rigoureuse
  - Possibilité d'automatiser au moins partiellement l'analyse et la vérification des modèles (la démonstration)

LOGIQUE Introduction - Motivations

- **Les logiques classiques**
  - Logique propositionnelle
  - Logique de prédicat

- **Les logiques non classiques**
  - Logiques multivalentes
  - Logique temporelle
  - Logique floue,
  - …

Introduction - Motivations (II)
Logique des propositions

Considérons la situation décrite par les affirmations suivantes :

1. **Si** le train arrive en retard et s'il n'y a pas de taxis à la gare **alors** le conférencier arrive en retard

2. Le conférencier n'est pas en retard

3. Le train est arrivé en retard

Et la déduction suivante :

Donc il y avait des taxis à la gare

Introduction - Motivations (III)
Logique des propositions

Pourquoi peut-on en déduire qu'il y avait des taxis à la gare ?

<u>Premièrement</u>, si l'on considère l'affirmation 1 et l'affirmation 3 ensemble, on peut affirmer que s'il n'y avait pas eu de taxis à la gare, le conférencier serait arrivé en retard.

<u>Deuxièmement</u>, cette dernière affirmation n'est compatible avec le fait 2 que s'il y avait des taxis à la gare.

Il est donc consistant de déduire qu'il y avait des taxis à la gare.

Introduction - Motivations (IV)
Logique des propositions

Comment peut-on formaliser la situation décrite précédemment dans un langage mathématique moins ambigu que le langage naturel ?

En utilisant le langage de la logique des propositions :

- Soit p la proposition " le train arrive en retard "
- Soit q la proposition " il y a des taxis à la gare "
- Soit r la proposition " le conférencier est en retard "

Alors nous pouvons formaliser la situation décrite par la formule logique suivante :

$$((p \wedge \neg q) \rightarrow r) \wedge (\neg r \wedge p)$$

Et nous pouvons formaliser la déduction par :

$$((p \wedge \neg q) \rightarrow r) \wedge (\neg r \wedge p) \models q$$

où $\models$ se lit " q est une conséquence logique de $((p \wedge \neg q) \rightarrow r) \wedge (\neg r \wedge p)$

Chapitre 1

La logique propositionnelle

Plan

- Syntaxe
- Sémantique
- Procédure de décision
- Déduction naturelle
- Principes de résolution

Logique propositionnelle

- Qu’est ce que la logique propositionnelle
    - Un langage formel pour exprimer des connaissances
    - La forme la plus simple des logiques mathématiques
    - Un langage symbolique pour décrire des propositions et de raisonner avec.
- Qu’est-ce qu’une proposition?
    - Une proposition (ou assertion) est une phrase déclarative qui peut-être soit vraie soit fausse (mais pas les deux)

La logique propositionnelle – syntaxe

- **Définition :**
  - Une **proposition** est un énoncé qui est soit VRAI (T, true, V, 1) soit FAUX ($\perp$, F, false, 0).
- **Exemples :**
  - **10 + 11 = 21 :** une proposition qui est vraie en arithmétique décimale (fausse en arithmétique binaire 10+11=101).
  - **La lune est un soleil :** une proposition qui est fausse.
  - **7 est un nombre premier** : une proposition qui est vraie.
  - **Il pleuvra après demain :** une proposition décidable à terme.

Votez Segozi : n'est pas une proposition.

Ingrédients d'un langage logique

- **Syntaxe**
  - Vocabulaire : Un ensemble de symboles (dits propositionnels), deux constantes et de connecteurs
  - Un langage est obtenu à partir des règles qui combinent ces symboles

- **Sémantique**
  - Sens associé aux symboles et connecteurs
  - Tables de vérité

- **Procédure d'inférence**
  - Ce sont des règles qui permettent de dériver de nouvelles propositions à partir des propositions considérées comme vraies.

La logique propositionnelle
Syntaxe - Construction des formules (I)

Le *vocabulaire du langage de la logique propositionnelle* est composé:

- d'un ensemble, fini ou dénombrable, de propositions notées $p, q, r, \ldots$
  Dans la suite, nous notons un tel ensemble de propositions par les lettres P, Q,...;
- de deux constantes : vrai (noté $\top$) et faux (noté $\perp$);
- d'un ensemble de connecteurs logiques : et (noté $\wedge$), ou (noté $\vee$), non (noté $\neg$), implique (noté $\rightarrow$), équivalent (noté $\leftrightarrow$);
- les parenthèses: (,).

Logique propositionnelle
Syntaxe – écritures possibles

Ecriture acceptée pour les opérateurs

$\rightarrow$, $\Rightarrow$, $\supset$: Implication matérielle
$\leftrightarrow$, $\Leftrightarrow$, $\equiv$: Equivalence

Les connecteurs logiques

- **Négation**
  - Si ”p” désigne une proposition alors on note ”$\neg$ p” comme la négation de ”p”
  - ” $\neg$ p” signifie : il n’est vrai que ”p”
  - Il n’est pas vrai que ”8 est un nombre premier”
- **Conjonction**
  - Si ”p” et ”q” désignent deux propositions, alors on note ”p$\wedge$q”, pour exprimer le fait que les deux assertions p et q sont toutes les deux vraies.
- **Disjonction**
  - Si ”p” et ”q” désignent deux propositions, alors on note ”p$\vee$q”, pour exprimer le fait que au moins l’une des deux assertions p et q est vraie (elles peuvent être vraies toutes les deux).

Les connecteurs logiques

- **Implication matérielle**
  - Très importante pour exprimer des informations conditionnelles
  - Si ”p” et ”q” désignent deux propositions, alors on note ”$p \rightarrow q$”, pour exprimer le fait que si p est vraie alors q est également vraie.
  - L'implication matérielle ”$p \rightarrow q$” est logiquement équivalente à ” $\neg p \vee q$”

Les connecteurs logiques

- **Equivalence logique**

  - Si ”p” et ”q” désignent deux propositions alors on note ”$p \leftrightarrow q$”, pour exprimer le fait que les deux propositions p et q sont équivalentes. En d’autres termes, p est vraie ssi q est vraie

  - ”$p \leftrightarrow q$” est vue comme une conjonction de ”$p \rightarrow q$” et ”$q \rightarrow p$” (c-à-d $(p \rightarrow q) \wedge (q \rightarrow p)$)

Le langage

- **Le langage : ensemble de formules propositionnelles**

L'ensemble des formules propositionnelles (formules propositionnelles bien formées) est défini par les règles suivantes

1. Si φ est une formule propositionnelle alors $\neg\varphi$ est une formule propositionnelle.
2. Si φ et ψ sont des formules propositionnelles alors $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$ et $\varphi \leftrightarrow \psi$ sont des formules propositionnelles.
3. Les formules propositionnelles sont obtenues uniquement en appliquant les règles (1)-(2) un certain nombre de fois.

La logique propositionnelle
Syntaxe - Construction des formules (II)

Les *formules de la logique propositionnelle* respectent la règle de formation suivante:

$$\phi ::= \top \mid \bot \mid p \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2$$

$$\mid \phi_1 \rightarrow \phi_2 \mid \phi_1 \leftrightarrow \phi_2 \mid \neg\phi_1 \mid (\phi_1)$$

Où $\top$, $\bot$ sont respectivement les constantes vrai et faux, $p \in P$ est une proposition et ϕ_1, ϕ_2 sont des formules propositionnelles bien formées. L'ensemble des formules propositionnelles bien formées sera noté FormulesProp.

La logique propositionnelle
Quelques exemples de formules

Voici quelques exemples de formules et leur lecture intuitive:

- la formule propositionnelle "$p \rightarrow (q \wedge r)$", peut-être lue de la façon suivante "p implique q et r", ou peut-être également lu comme "si p est vrai alors q et r doivent être vrais;

- la formule propositionnelle "$\neg(p \wedge q)$", peut-être lue de la facon suivante "il est faux que p et q soient vrais (en même temps)".

Exemple de modélisation

1. Jean et Pierre prirent le café et Gustave fit de même.
2. Jean prit le café, et Pierre ou Gustave aussi
3. Jean et Pierre ont dîné tous les deux, ou bien Jean et Gustave prirent le café
4. Jean a dîné, ainsi que Gustave ou Pierre
5. Si Paul réussit le cours alors Chantal et Gustave aussi

J : Jean prit le café
P : Pierre prit le café
G : Gustave prit le café
D : Jean a dîné
E : Gustave a dîné
F : Pierre a dîné
R: Paul réussit le cours
C : Chantal réussit le cours, T : Gustave réussit le cours

1. $J \wedge P \wedge G$.
2. $J \wedge (P \vee G)$
3. $(D \wedge F) \vee (J \wedge G)$
4. $D \wedge (E \vee F)$
5. $R \rightarrow (C \wedge T)$

La logique propositionnelle
Ambiguïtés (I)

L'utilisation de la notation infixe s'accompagne de problèmes de parsing:

$$
\begin{aligned}
& e_1\, op_1 e_2\, op_2\, e_3 \\
=^? \ & (e_1\, op_1 e_2) op_2\, e_3 \\
=^? \ & e_1\, op_1 (e_2 op_2 e_3)
\end{aligned}
$$

La logique propositionnelle
Ambiguïtés (II)

Pour lever les ambiguïtés, on utilise les parenthèses ou des règles de priorité entre opérateurs :

- si op_1 a une plus grande précédence que op_2 alors

$$e_1 op_1 e_2 op_2 e_3$$

est équivalent à $((e_1 op_1 e_2) op_2 e_3)$

- si op est associf à gauche alors

$$e_1 ope_2 ope_3$$

est équivalent à $((e_1 ope_2) ope_3)$

Une fois ces règles fixées, à chaque formule correspond un et un seul arbre de parsing.

La logique propositionnelle
Règles de précédence

Ordre de précédence sur les opérateurs :

$$\neg$$
$$\wedge$$
$$\vee$$
$$\rightarrow$$
$$\leftrightarrow$$

et associativité à gauche. (sauf l'implication)

L’arbre de parsing

•L’arbre de parsing (arbre d'analyse syntaxique) **représente la formule et ses sous formules sous une forme arborescente**. Les nœuds non-terminaux sont étiquetés par les opérateurs ou connecteurs logiques utilisés et les feuilles par des littéraux.

•L'utilisation d'un arbre d'analyse syntaxique (ou arbre de parsing) pour une formule propositionnelle est bénéfique pour plusieurs raisons :

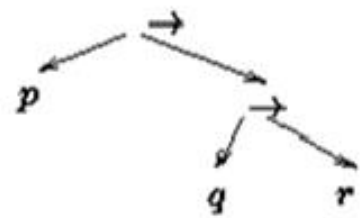
- **Facilité de conversion** : Un arbre d'analyse syntaxique **fournit une représentation intermédiaire utile pour convertir la formule propositionnelle dans d'autres formats** ou langages. Par exemple, vous pouvez facilement convertir la formule propositionnelle en forme normale conjonctive (FNC) ou en forme normale disjonctive (FND) à partir de l'arbre d'analyse syntaxique.
- **Analyse et vérification syntaxique** : L'utilisation d'un arbre d'analyse syntaxique **permet de vérifier la syntaxe de la formule propositionnelle**. Vous pouvez détecter et signaler les erreurs syntaxiques telles que les parenthèses non équilibrées, les opérateurs mal placés, ou les opérandes manquants.
- …

La logique propositionnelle
Arbre correspondant à une formule

La formule $p \rightarrow q \rightarrow r$ est donc équivalente à

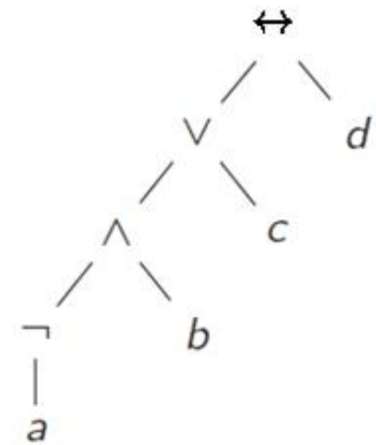
$$(p \rightarrow (q \rightarrow r))$$

et donc son arbre de parsing est :



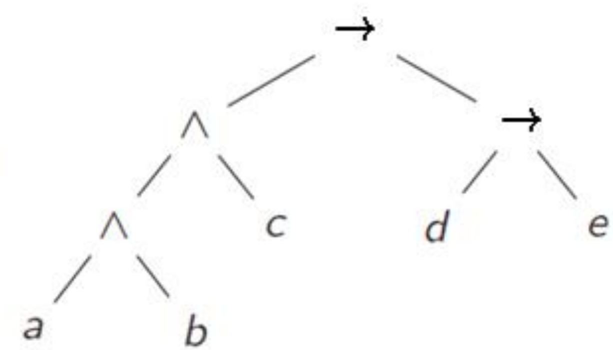
La logique propositionnelle
Arbre correspondant à une formule

$\neg a \wedge b \vee c \leftrightarrow d$
$(((\neg a) \wedge b) \vee c) \leftrightarrow d$ dénotent



- **Associativité** : $\wedge$, $\vee$ et $\leftrightarrow$ associent à *gauche* et $\rightarrow$ associe à *droite* :

$a \wedge b \wedge c \rightarrow d \rightarrow e$
$((a \wedge b) \wedge c) \rightarrow (d \rightarrow e)$ dénotent



La logique propositionnelle
Profondeur d'une formule

La profondeur d'une formule ϕ est la profondeur de l'arbre A_ϕ associé à la formule.

La profondeur d'une formule ϕ est définie de manière inductive comme suit :

- cas de base : si $\phi = p$ où p est une proposition alors $Prof(\phi) = 0$;
- cas inductif : si $\phi = (\phi_1)op(\phi_2)$ alors $Prof(\phi) = 1 + \max(Prof(\phi_1), Prof(\phi_2))$

Nous notons $\mathsf{FormulesProp}_i$ les formules propositionnnelles de profondeur i. Notons que

$$\mathsf{FormulesProp} = \bigcup_{i \geq 0} \mathsf{FormulesProp}_i$$

Logique des propositions
- Formes normales –

Toute formule bien formée peut être exprimée selon les deux formes normales suivantes :

- **Forme normale disjonctive (FND) : disjonctions de conjonctions entre propositions atomiques**

Exemple : $(\neg p \wedge q) \vee (r \wedge \neg t) \vee (q \wedge t)$

- **Forme normale conjonctive (FNC) : conjonctions de disjonctions entre propositions atomiques**

Exemple : $(\neg p \vee q) \wedge (r \vee t) \wedge (t \vee p)$

La logique propositionnelle

- Syntaxe
- **Sémantique**
- Procédure de décision
- Déduction naturelle
- Principes de résolution

La logique propositionnelle
La sémantique

- Utilisation de **l'interprétation** qui assigne des valeurs de vérité aux propositions atomiques
- La sémantique attribue une signification aux connecteurs.
  - Définir les **tables de vérités**
- La sémantique attribue une signification aux expressions.
- Elle est **compositionnelle** : la signification d'une formule est fonction de celle de ses constituants.
- Etudier la **satisfaisabilité** et la **validité** des formules
- La **conséquence logique** entre formules

La logique propositionnelle
La sémantique (I)

Etant donné un ensemble de proposotion P, une *fonction d'interprétation* V pour P est une fonction du type

$$V : P \rightarrow \{\text{vrai}, \text{faux}\},$$

V(p) = vrai ou faux

C'est une fonction qui assigne à chaque proposition de P la valeur vrai ou la valeur faux. L'ensemble des fonctions d'interprétation propositionnelle sera noté FonctInterProp.

Dans la suite, nous utiliserons parfois le terme *valuation* à la place de fonction d'interprétation.

La logique propositionnelle
La sémantique (II)

La *valeur de vérité* d'une formule propositionnelle ϕ contenant l'ensemble P de propositions, évaluée avec la fonction d'interprétation V, est notée $[\![\phi]\!]_V$. Formellement,

$[\![\cdot]\!]_. :$ FormulesProp$\times$ FonctInterProp $\rightarrow$ {vrai, faux}.

La fonction $[\![\phi]\!]_V$ est définie par induction sur la syntaxe de ϕ de la façon suivante:

- $\phi = p$, dans ce cas, $[\![\phi]\!]_V = V(p)$.

- $\phi = \neg\phi_1$, $[\![\phi]\!]_V = \begin{cases} \text{faux} & \underline{\text{si}}\ [\![\phi_1]\!]_V = \text{vrai} \\ \text{vrai} & \underline{\text{si}}\ [\![\phi_1]\!]_V = \text{faux} \end{cases}$

- $\phi = \phi_1 \vee \phi_2$,

$$[\![\phi]\!]_V = \text{vrai}\ \underline{\text{ssi}}\quad \begin{array}{l} [\![\phi_1]\!]_V = \text{vrai} \\ \underline{\text{ou}}\ [\![\phi_2]\!]_V = \text{vrai} \end{array}$$

- $\phi = \phi_1 \wedge \phi_2$,

  $[\![\phi]\!]_V =$ vrai ssi $[\![\phi_1]\!]_V =$ vrai
  et $[\![\phi_2]\!]_V =$ vrai

- $\phi = \phi_1 \rightarrow \phi_2$,

  $[\![\phi]\!]_V =$ vrai ssi $[\![\phi_1]\!]_V =$ faux
  ou $[\![\phi_2]\!]_V =$ vrai

- $\phi = \phi_1 \leftrightarrow \phi_2$,

  $[\![\phi]\!]_V =$ vrai ssi
  $[\![\phi_1]\!]_V =$ faux et $[\![\phi_2]\!]_V =$ faux
  ou $[\![\phi_1]\!]_V =$ vrai et $[\![\phi_2]\!]_V =$ vrai

Nous notons $V \Vdash \phi$ si et seulement si $[\![\phi]\!]_V =$ vrai.

La logique propositionnelle
La sémantique (III)

- L'information contenue dans la définition précédente est généralement écrite sous forme d'une table, appelée table de vérité

p	$\neg p$	(p)
vrai	faux	vrai
faux	vrai	faux

- Cette table permet de calculer la valeur de vérité d'une formule propositionnelle en fonction de celle des propositions atomiques qui la composent.

La logique propositionnelle
La sémantique (IV)

- La table de vérité du connecteur binaire $\wedge$

p	q	$p \wedge q$
vrai	vrai	vrai
vrai	faux	faux
faux	vrai	faux
faux	faux	faux

- Une interprétation V affecte une valeur de vérité à chacune des propositions atomiques (ou variables propositionnelles)
- Chaque ligne de la table de vérité correspond à une interprétation possible V.
  - Première ligne : $V(p)=vrai, V(q)=vrai \Rightarrow [[p\wedge q]]_V=vrai \Rightarrow V \Vdash p\wedge q$
  - Deuxième ligne : $V(p)=vrai, V(q)=faux \Rightarrow [[p\wedge q]]_V =faux \Rightarrow V \nVdash p\wedge q$,
  - …
- Le nombre d'interprétations possibles est égal au 2^n si n est le nombre de proposition atomiques

La logique propositionnelle
La sémantique (IV)

- La table de vérité de $\vee$

p	q	$p \vee q$
vrai	vrai	vrai
vrai	faux	vrai
faux	vrai	vrai
faux	faux	faux

La logique propositionnelle
La sémantique (IV)

- La table de vérité des connecteurs $\rightarrow$ et $\leftrightarrow$

p	q	$p \rightarrow q$	$p \leftrightarrow q$
vrai	vrai	vrai	vrai
vrai	faux	faux	faux
faux	vrai	vrai	faux
faux	faux	vrai	vrai

- Deux formules sont équivalentes ssi elles prennent les mêmes valeurs de vérité pour toutes les interprétations possibles
- Démontrer en utilisant la table de vérité que les deux formules suivantes sont sémantiquement équivalentes : $p \leftrightarrow q$ et $(p\rightarrow q) \wedge (q\rightarrow p)$

p	q	$p\leftrightarrow q$	$p\rightarrow q$	$q\rightarrow p$	$(p\rightarrow q) \wedge (q\rightarrow p)$
V	V	V	V	V	V
V	F	F	F	V	F
F	V	F	V	F	F
F	F	V	V	V	V

Propriétés algébriques

En utilisant les tables de vérités on peut montrer les propriétés suivantes :

Commutativité	$\varphi \vee \psi \equiv \psi \vee \varphi$, $\varphi \wedge \psi \equiv \psi \wedge \varphi$
Associativité	$(\varphi \vee \psi) \vee \theta \equiv \varphi \vee (\psi \vee \theta)$, $(\varphi \wedge \psi) \wedge \theta \equiv \varphi \wedge (\psi \wedge \theta)$
Distributivité	$\varphi \vee (\psi \wedge \theta) \equiv (\varphi \vee \psi) \wedge (\varphi \vee \theta)$, $\varphi \wedge (\psi \vee \theta) \equiv (\varphi \wedge \psi) \vee (\varphi \wedge \theta)$
Identité	$\varphi \vee \bot \equiv \varphi$, $\psi \wedge \top \equiv \psi$
Zéro	$\varphi \vee \top \equiv \top$, $\varphi \wedge \bot \equiv \bot$
Itempotence	$\varphi \vee \varphi \equiv \varphi$, $\varphi \wedge \varphi \equiv \varphi$
Absorption	$\varphi \vee (\varphi \wedge \psi) \equiv \varphi$, $\varphi \wedge (\varphi \vee \psi) \equiv \varphi$
Complémentation	$\varphi \vee \neg\varphi \equiv \top$, $\varphi \wedge \neg\varphi \equiv \bot$
Double négation	$\neg\neg\varphi \equiv \varphi$
de Morgan	$\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$, $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$
Définition	$\varphi \Rightarrow \psi \equiv \neg\varphi \vee \psi$, $\varphi \Leftrightarrow \psi \equiv (\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$

La logique propositionnelle
La sémantique (VI)

Voici deux définitions importantes :

- Une formule propositionnelle ϕ est *valide* si et seulement si pour toute fonction d'interprétation V pour les propositions de ϕ, on a $V \Vdash \phi$.

- Une formule propositionnelle ϕ est *satisfaisable* si et seulement si il existe une fonction d'interprétation V pour les propositions de ϕ, telle que $V \Vdash \phi$.

$V \Vdash \phi$ **V satisfait ϕ ; V est un modèle de ϕ**

$V \nVdash \phi$ **V ne satisfait pas ϕ ; V est un contre-modèle de ϕ**

Une conséquence directe de ces deux définitions est que :

Une formule propositionnelle ϕ est valide ssi sa negation $\neg\phi$ n'est pas satisfaisable.

Preuve. On applique simplement les définitions. Par définition ϕ est valide, si pour toute fonction d'interprétation V, $V \Vdash \phi$,

ce qui est équivalent à $V \nVdash \neg\phi$,

par définition de la sémantique de $\neg$ et donc si ϕ est valide,

il n'existe pas de fonction d'interprétation V telle que $V \Vdash \neg\phi$ et donc $\neg\phi$ n'est pas satisfaisable.

L'autre direction est laissée comme exercice.

La logique propositionnelle
La sémantique (VII)

Une formule propositionnelle ψ est une *conséquence logique* d'un ensemble de formules propositionnelles $\{\phi_1, \ldots, \phi_n\}$ si et seulement si pour toute fonction d'interprétation V, telle que $V \Vdash \phi_i$ pour tout i, $1 \leq i \leq n$, on a également $V \Vdash \psi$, ce fait est noté

$$\phi_1, \ldots, \phi_n \models \psi.$$

Deux formules ϕ et ψ sont *equivalentes* si et seulement si on a $\phi \models \psi$ et $\psi \models \phi$.

Méthode de la Table de vérité

La méthode de la table de vérité

Toutes les interprétations possibles V

p_1	p_2	p_3	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ψ
V	V	v	V	V	V	V	V
V	V	F	V	F	V	F	F
F	F	V	V	V	V	V	V
V	F	F	F	V	F	V	F
F	V	V	F	F	F	F	F
F	V	F	V	V	V	V	V
F	F	V	V	V	F	F	F
F	F	F	V	V	V	V	V

- Une formule est valide si elle est vraie dans toutes les interprétations (les 8 lignes)
- Une formule est satisfaisable s'il existe une interprétation tel que la formule est vraie
- Elle non satisfaisable si elle n'est pas vraie dans aucune ligne
- Pour démontrer que $\phi_{1,} \ldots \phi_4 \models \psi$ on doit démontrer que:

$\forall$V une interprétation, si V satisfait tous les ϕ_i ($V \Vdash \phi_i$ ou $[[\phi_i]]_V$ = vrai) alors V doit satisfaire ψ (c-à-d $[[\psi]]_V$ = vrai)

Logique des propositions
- Formes normales –

Toute formule bien formée peut être exprimée selon les deux formes normales suivantes :

- **Forme normale disjonctive (FND) : disjonctions de conjonctions entre propositions atomiques**

Exemple : $(\neg p \wedge q) \vee (r \wedge \neg t) \vee (q \wedge t)$

- **Forme normale conjonctive (FNC) : conjonctions de disjonctions entre propositions atomiques**

Exemple : $(\neg p \vee q) \wedge (r \vee t) \wedge (t \vee p)$

Mise sous forme normale disjonctive

En utilisant :

- la définition de $\Rightarrow$ et $\Leftrightarrow$:

$$\varphi \Rightarrow \psi \equiv \neg\varphi \vee \psi,\ \varphi \Leftrightarrow \psi \equiv (\neg\varphi \vee \psi) \wedge (\varphi \vee \neg\psi)$$

- on pousse les négations au niveau des variables :

$$\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi,\ \neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi,\ \neg\neg\varphi \equiv \varphi$$

- la distributivité de $\wedge$ par rapport au $\vee$ permet de terminer :

$$(\varphi \vee \psi) \wedge \theta \equiv (\varphi \wedge \theta) \vee (\psi \wedge \theta)$$

Mise sous forme normale disjonctive – exemple

- Éliminer $\Rightarrow$ et $\Leftrightarrow$ avec leur définition,
- Pousser les négations jusqu'au variable
- Utiliser la distributivité de $\wedge$ par rapport au $\vee$.

$$
\begin{aligned}
\varphi = &\neg(a \Rightarrow b) \vee \neg(c \vee \neg(d \Rightarrow a)) \\
&\neg(\neg a \vee b) \vee \neg(c \vee \neg(\neg d \vee a)) \\
&(\neg\neg a \wedge \neg b) \vee \neg(c \vee \neg(\neg d \vee a)) \\
&(a \wedge \neg b) \vee \neg(c \vee \neg(\neg d \vee a)) \\
&(a \wedge \neg b) \vee (\neg c \wedge \neg\neg(\neg d \vee a)) \\
&(a \wedge \neg b) \vee (\neg c \wedge (\neg d \vee a)) \\
&(a \wedge \neg b) \vee ((\neg c \wedge \neg d) \vee (\neg c \wedge a))) \\
&(a \wedge \neg b) \vee (\neg c \wedge \neg d) \vee (\neg c \wedge a)
\end{aligned}
$$

NB : Ce processus peut donner une formule exponentiellement plus grande que celle d'origine.

Mise sous forme normale conjonctive

En utilisant :

- la définition de $\Rightarrow$ et $\Leftrightarrow$:

$$\varphi \Rightarrow \psi \equiv \neg\varphi \vee \psi, \; \varphi \Leftrightarrow \psi \equiv (\neg\varphi \vee \psi) \wedge (\varphi \vee \neg\psi)$$

- on pousse les négations au niveau des variables :

$$\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi, \; \neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi, \; \neg\neg\varphi \equiv \varphi$$

- la distributivité de $\vee$ par rapport au $\wedge$ permet de terminer :

$$(\varphi \wedge \psi) \vee \theta \equiv (\varphi \vee \theta) \wedge (\psi \vee \theta)$$

Obtenir le FND à partir d'une table de vérité

- Obtenir le FND d'une formule à partir de sa table de vérité
- Algorithme
  - Pour chaque ligne ayant comme valeur de vérité vrai (1), écrire la conjonction des littéraux, $C_i = (l_i \wedge \ldots \wedge l_r)$ de la façon suivante :
    - si les variables sont $p_1, \ldots, p_r$ et la variable p_i vaut vrai (1), le littéral l_i est p_i .
    - Sinon si la variable a pour valeur faux (0), le littéral l_i est $\neg p_i$
  - Prendre la disjonction $(C_1 \vee \ldots \vee C_k)$ de toutes les C_i correspondant aux lignes de la table qui ont pour valeur vrai (1).

FND – exemple

Pour chaque ligne ayant pour valeur 1, on écrit la conjonction des littéraux selon les règles décrites. On obtient :

x	y	z	$f(x,y,z)$	
0	0	0	1	$\neg x \wedge \neg y \wedge \neg z$
0	0	1	0	
0	1	0	0	
0	1	1	1	$\neg x \wedge y \wedge z$
1	0	0	0	
1	0	1	1	$x \wedge \neg y \wedge z$
1	1	0	0	
1	1	1	1	$x \wedge y \wedge z$

On prend alors la disjonction de toutes ces lignes pour obtenir la forme normale disjonctive

$$f(x,y,z) = (\neg x \wedge \neg y \wedge \neg z) \vee (\neg x \wedge y \wedge z) \vee (x \wedge \neg y \wedge z) \vee (x \wedge y \wedge z)$$

Obtenir le FNC à partir d'une table de vérité

- Obtenir le FNC d'une formule à partir de sa table de vérité
- Algorithme
  - Pour chaque ligne ayant comme valeur de vérité Faux (0), écrire la disjonction des littéraux, $D_i = (l_i \vee \ldots \vee l_r)$ de la façon suivante :
    - si les variables sont $p_1, \ldots, p_r$ et la variable p_i vaut faux (0), le littéral l_i est p_i .
    - Sinon si la variable a pour valeur vrai (1), le littéral l_i est $\neg p_i$
  - Prendre la conjonction $(D_1 \wedge \ldots \wedge D_k)$ de toutes les clauses D_i correspondant aux lignes de la table qui ont pour valeur faux (0).

FNC – exemple

x	y	z	$f(x,y,z)$	
0	0	0	0	$x \vee y \vee z$
0	0	1	1	
0	1	0	1	
0	1	1	0	$x \vee \neg y \vee \neg z$
1	0	0	1	
1	0	1	0	$\neg x \vee y \vee \neg z$
1	1	0	1	
1	1	1	0	$\neg x \vee \neg y \vee \neg z$

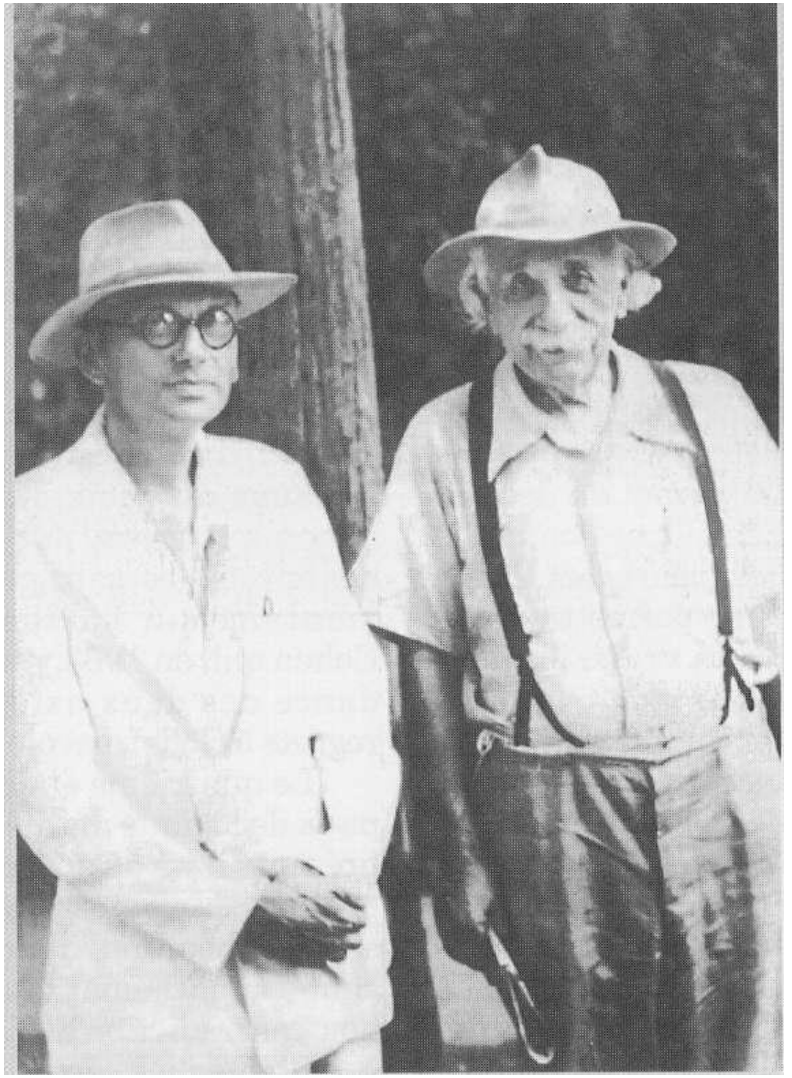
On prend alors la conjonction de toutes ces lignes pour obtenir la forme normale conjonctive

$$f(x,y,z) = (x \vee y \vee z) \wedge (x \vee \neg y \vee \neg z) \wedge (\neg x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee \neg z)$$

La logique propositionnelle

- Syntaxe
- Sémantique
- **Procédure de décision**
- Déduction naturelle
- Principes de résolution

La logique des propositions
- Procédure de décision –
Tableaux sémantiques ou de Beth



La logique propositionnelle Procédure de décision - Tableaux sémantiques (I)

Algorithme pour établir la satisfaisabilité/validité de formules de la logique propositionnelle.

Le principe est très simple : pour prouver la satisfaisabilité d'une formule de la logique propositionnelle, on cherche systématiquement un modèle pour cette formule.

La logique propositionnelle
Procédure de décision - Tableaux sémantiques (II)

On a besoin de quelques nouvelles définitions:

Une formule propositionnelle ϕ est un *littéral* si et seulement si ϕ est une proposition ou la négation d'une proposition.

Deux formules ϕ et $\neg\phi$ sont des *formules complémentaires*.

La logique propositionnelle
Procédure de décision - Tableaux sémantiques (III)

Considérons la formule $\phi = p \wedge (\neg q \vee \neg p)$

Essayons de construire systématiquement une fonction d'interprétation V telle que

$$[\![\phi]\!]_V = \text{vrai}$$

Par définition on a

$$[\![\phi]\!]_V = \text{vrai}$$
$$\underline{\text{ssi}}$$
$$[\![p]\!]_V = \text{vrai} \quad \underline{\text{et}} \quad [\![\neg q \vee \neg p]\!]_V = \text{vrai}$$

et

$$[\![\neg q \vee \neg p]\!]_V = \text{vrai}$$
$$\underline{\text{ssi}}$$
$$[\![\neg q]\!]_V = \text{vrai} \quad \underline{\text{ou}} \quad [\![\neg p]\!]_V = \text{vrai}$$

et donc,

$$[\![\phi]\!]_V = \text{vrai}$$
$$\underline{\text{ssi}}$$
$$\left([\![p]\!]_V = \text{vrai } \underline{\text{et}} \ [\![\neg q]\!]_V = \text{vrai}\right)$$
$$\underline{\text{ou}}\left([\![p]\!]_V = \text{vrai } \underline{\text{et}} \ [\![\neg p]\!]_V = \text{vrai}\right) \quad \text{impossible}$$

Pour ϕ, on construit la fonction d'interprétation V de la façon suivante :

- $V(p) = \text{vrai}$;
- $V(q) = \text{faux}$.

et on a bien que $V \Vdash \phi$.

La logique propositionnelle Procédure de décision - Tableaux sémantiques (IV)

Nous avons réduit le problème de satisfaction de ϕ à un problème de satisfaction d'ensembles de littéraux.

Un ensemble S de littéraux est satisfaisable <u>ssi</u> il ne contient pas une paire de littéraux complémentaires.

Vu que chaque formule contient un ensemble fini de propositions, et donc un ensemble fini de littéraux, il y a un nombre fini d'ensembles de littéraux.

La logique propositionnelle
Procédure de décision - Tableaux sémantiques (V)

Un autre exemple : $\phi = (p \vee q) \wedge (\neg p \wedge \neg q)$.

Par définition, on

$$\llbracket \phi \rrbracket_V = \text{vrai}$$
$$\underline{\text{ssi}}$$
$$\llbracket p \vee q \rrbracket_V = \text{vrai } \underline{\text{et}} \;\; \llbracket \neg p \wedge \neg q \rrbracket_V = \text{vrai}$$

et donc,

$$\llbracket \phi \rrbracket_V = \text{vrai}$$
$$\underline{\text{ssi}}$$
$$\llbracket p \vee q \rrbracket_V = \text{vrai } \underline{\text{et}} \;\; \llbracket \neg p \rrbracket_V = \text{vrai}$$
$$\underline{\text{et}} \;\; \llbracket \neg q \rrbracket_V = \text{vrai}$$

et donc,

$$
\begin{gathered}
[\![\phi]\!]_V = \text{vrai} \\
\underline{\text{ssi}} \\
\text{soit}\ \big([\![p]\!]_V = \text{vrai}\ \ \underline{\text{et}}\ \ [\![\neg p]\!]_V = \text{vrai} \\
\underline{\text{et}}\ \ [\![\neg q]\!]_V = \text{vrai}\big) \\
\underline{\text{ou}}\ \big([\![q]\!]_V = \text{vrai}\ \ \underline{\text{et}}\ \ [\![\neg p]\!]_V = \text{vrai} \\
\underline{\text{et}}\ \ [\![\neg q]\!]_V = \text{vrai}\big)
\end{gathered}
$$

Vu que les deux ensembles de littéraux ont des pairs complémentaires, la formule ϕ n'est donc pas satisfaisable.

La logique propositionnelle Procédure de décision - Tableaux sémantiques (VI)

- L’information présentée dans les développements précédents est plus facilement présentée sous forme d’un arbre (tableau sémantique ou de Beth).

- Une méthode systématique pour trouver les modèles d’une formule

- Construction de l'arbre :

  - Mettre l'ensemble contenant la formule comme la racine de cet arbre
  - Appliquer le pas de simplification sur chaque nœud de la façon suivante :
    - Décomposer une formule de l’ensemble de formules (une seule à la fois).
      - Créer un fils si α-règle ($\wedge$, …)
      - Créer deux fils si β-règle ($\vee$, …)
  - Répéter jusqu’à ce que tous les ensembles contiennent des littéraux

- Un autre exemple : $\varphi = p \wedge (\neg q \vee \neg p)$

$$\begin{tikzcd} & \{p \wedge (\neg q \vee \neg p)\} \arrow[d] & \\ & \{p, (\neg q \vee \neg p)\} \arrow[dl] \arrow[dr] & \\ \{p, \neg q\} & & \{p, \neg p\} \end{tikzcd}$$

- φ est satisfaisable puisque l'ensemble $\{p, \neg q\}$ l'est. En effet, il ne contient pas des propositions complémentaires.

Exemple : $\phi = (p \vee q) \wedge (\neg p \wedge \neg q)$

$$
\begin{tikzcd}
& \{(p \vee q) \wedge (\neg p \wedge \neg q)\} \arrow[d] & \\
& \{(p \vee q), (\neg p \wedge \neg q)\} \arrow[d] & \\
& \{(p \vee q), \neg p, \neg q\} \arrow[dl] \arrow[dr] & \\
\{p, \neg p, \neg q\} & & \{q, \neg p, \neg q\}
\end{tikzcd}
$$

La logique propositionnelle
Procédure de décision - Tableaux sémantiques (VII)

Remarques :

- quand on applique le pas de simplification à un noeud où on sélectionne une conjonction, alors on obtient un seul fils, on appelle ces règles, des α-règles;

- quand on applique le pas de simplification à un noeud où on sélectionnne une disjction, alors on obtient deux fils, on appelle ces règles, des β-règles.

La logique propositionnelle Procédure de décision - Tableaux sémantiques (VIII)

α	α_1	α_2
$\neg\neg\phi$	ϕ	
$\phi_1 \wedge \phi_2$	ϕ_1	ϕ_2
$\neg(\phi_1 \vee \phi_2)$	$\neg\phi_1$	$\neg\phi_2$
$\neg(\phi_1 \rightarrow \phi_2)$	ϕ_1	$\neg\phi_2$
$\phi_1 \leftrightarrow \phi_2$	$\phi_1 \rightarrow \phi_2$	$\phi_2 \rightarrow \phi_1$

α-règles.

Remarque : toutes les formules α peuvent être considérées (sont équivalentes) à des conjonctions. Par exemple :

$$\neg(\phi_1 \vee \phi_2) \equiv \neg\phi_1 \wedge \neg\phi_2$$

La logique propositionnelle
Procédure de décision - Tableaux sémantiques (IX)

β	β_1	β_2
$\phi_1 \vee \phi_2$	ϕ_1	ϕ_2
$\neg(\phi_1 \wedge \phi_2)$	$\neg\phi_1$	$\neg\phi_2$
$\phi_1 \rightarrow \phi_2$	$\neg\phi_1$	ϕ_2
$\neg(\phi_1 \leftrightarrow \phi_2)$	$\neg(\phi_1 \rightarrow \phi_2)$	$\neg(\phi_2 \rightarrow \phi_1)$

β-règles.

Remarque : toutes les formules β peuvent être considérées (sont équivalentes) à des disjonctions. Par exemple :

$$\neg(\phi_1 \wedge \phi_2) \equiv \neg\phi_1 \vee \neg\phi_2$$

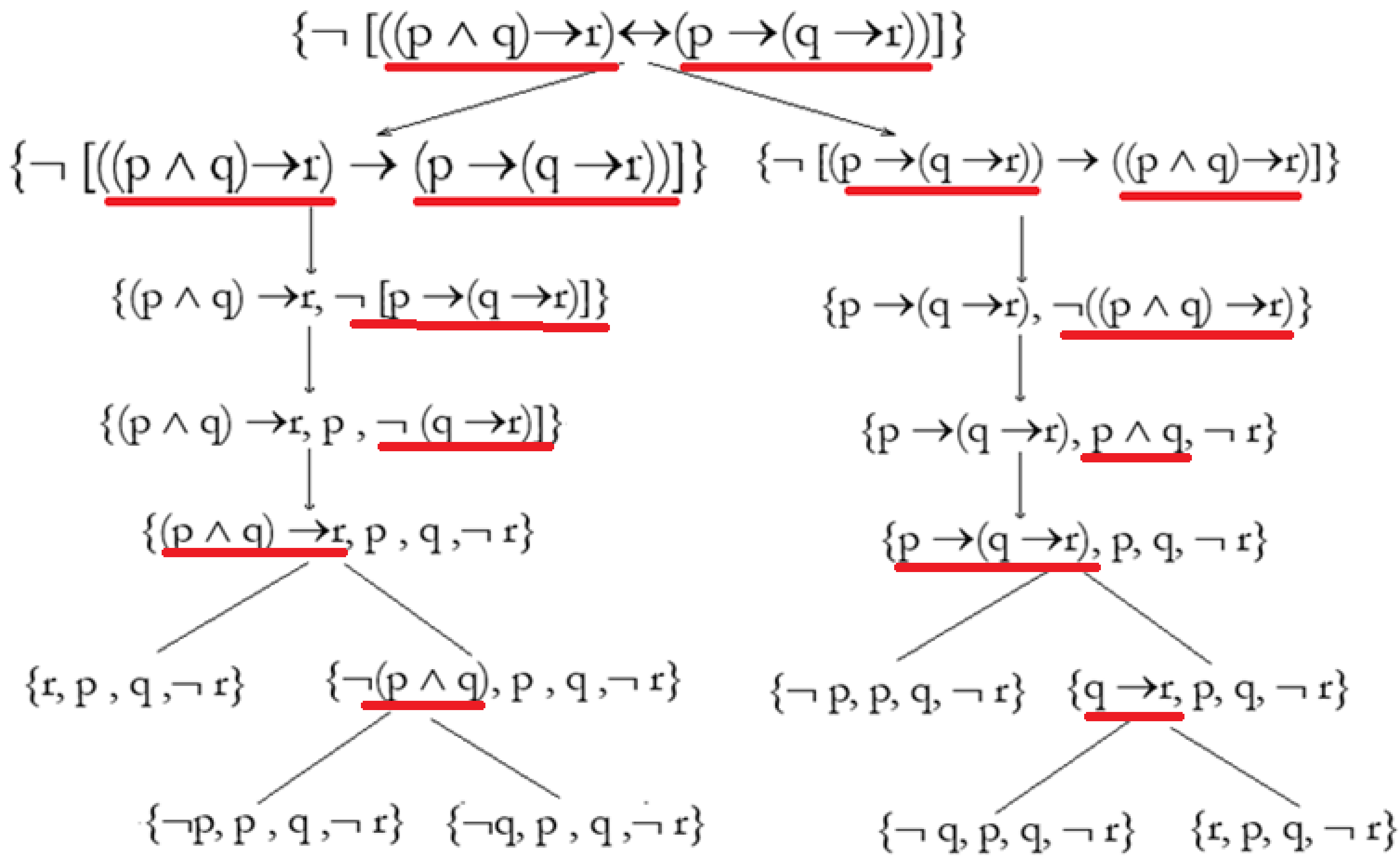
β	β_1	β_2
$\phi_1 \vee \phi_2$	ϕ_1	ϕ_2
$\neg(\phi_1 \wedge \phi_2)$	$\neg\phi_1$	$\neg\phi_2$
$\phi_1 \rightarrow \phi_2$	$\neg\phi_1$	ϕ_2
$\neg(\phi_1 \leftrightarrow \phi_2)$	$\neg(\phi_1 \rightarrow \phi_2)$	$\neg(\phi_2 \rightarrow \phi_1)$

β-règles.

α	α_1	α_2
$\neg\neg\phi$	ϕ	
$\phi_1 \wedge \phi_2$	ϕ_1	ϕ_2
$\neg(\phi_1 \vee \phi_2)$	$\neg\phi_1$	$\neg\phi_2$
$\neg(\phi_1 \rightarrow \phi_2)$	ϕ_1	$\neg\phi_2$
$\phi_1 \leftrightarrow \phi_2$	$\phi_1 \rightarrow \phi_2$	$\phi_2 \rightarrow \phi_1$

α-règles.

- Un autre exemple : $\varphi = \neg\,[((p \wedge q)\rightarrow r)\leftrightarrow(p \rightarrow(q \rightarrow r))]$



- φ est non satisfaisable puisque tous les ensembles des littéraux sont non satisfaisables

La logique propositionnelle
Procédure de décision - Tableaux sémantiques (X)

Algorithme de construction d'un tableau sémantique, quelques notations :

- un arbre représentant le tableau sémantique d'une formule ϕ sera noté T_ϕ;
- l, n, m représentent des noeuds d'un arbre;
- Lab(l) dénote l'étiquette du noeud l, c'est un ensemble de formules;
- Status(l) est le status du noeud l, ce status est – pour tout noeud qui n'est pas une feuille et Status(l) $\in \{ouvert, ferme\}$ si l est une feuille;
- Q, R dénotent des ensembles de noeuds.

La logique propositionnelle Procédure de décision - Tableaux sémantiques (XI)

Algorithme :

- Initialisation : $R := \{l\}$; Lab$(l) := \{\phi\}$;

- Itérer tant que $R \neq \emptyset$: choisir $l \in R$:

  - **si** Lab(l) est un ensemble de littéraux **alors** :

    * Si Lab(l) contient une paire de littéraux complémentaires : Status(l) := ferme et $R := R \setminus \{l\}$;

    * Si Lab(l) ne contient pas de paires de littéraux complémentaires : Status(l) := ouvert et $R := R \setminus \{l\}$;

- **si** Lab(l) n'est pas un ensemble de littéraux **alors** : choisir $\psi \in$ Lab(l),
  * si ψ est une α-formule alors créer m fils de l et :
    - $R := R \cup \{m\} \setminus \{l\}$;
    - $\mathrm{Lab}(m) := (\mathrm{Lab}(l) \setminus \{\psi\}) \cup \{\alpha_1, \alpha_2\}$;
  * si ψ est une β-formule alors créer m et n fils de l et :
    - $R := R \cup \{m, n\} \setminus \{l\}$;
    - $\mathrm{Lab}(m) := (\mathrm{Lab}(l) \setminus \{\psi\}) \cup \{\beta_1\}$;
    - $\mathrm{Lab}(n) := (\mathrm{Lab}(l) \setminus \{\psi\}) \cup \{\beta_2\}$;

La logique propositionnelle
Procédure de décision - Tableaux sémantiques (XI)

Un tableau dont la construction est terminée est appelé un *tableau complet*.

Un tableau complet est *fermé* si toutes ses feuilles sont étiquettées avec *fermé*, il est dit *ouvert* sinon.

La logique propositionnelle Procédure de décision - Tableaux sémantiques (XII)

Théorème : Si T_ϕ est le tableau complet de la formule ϕ, alors ϕ est non satisfaisable ssi T_ϕ est fermé.

Avant de prouver la correction de l'algorithme par rapport au théorème ci-dessus observons quelques conséquences de ce théorème :

- ϕ est satisfaisable <u>ssi</u> T_ϕ est ouvert;
- ϕ est valide <u>ssi</u> $T_{\neg\phi}$ est fermé.

La logique propositionnelle Procédure de décision - Tableaux sémantiques (XII)

Correction de l'algorithme : qu'est-ce que cela veut dire ?

- *terminaison* : l'algorithme termine pour toutes les formules de la logique propositionnelle;
- *adéquation* : si l'algorithme construit un tableau qui ferme pour une formule ϕ alors la formule ϕ est non satisfaisable;
- *complétude* : pour toute formule ϕ non satisfaisable, l'algorithme construit un tableau fermé.

La logique propositionnelle Procédure de décision - Tableaux sémantiques (XIII)

Terminaison de l'algorithme :

On définit une fonction de "poids" qui associe à chaque ensemble de formules un nombre naturel, on note Poids(l) le poids de l'étiquette du noeuds l.

On montre que si m est un fils de l alors

$$\text{Poids}(m) < \text{Poids}(l).$$

Vu que Poids() est une fonction naturelle alors la profondeur de tout tableau sémantique est finie et donc l'algorithme termine. (preuve complète donnée au cours.)

La logique propositionnelle
Rappel : l'induction mathématique (I)

Il est bien connu que pour $n \geq 1$, on a

$$1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}$$

Pour démontrer que cette égalité, notée $M(n)$, est valide pour n'importe quel $n \geq 1$, on applique le *principe d'induction mathématique*.

La logique propositionnelle
Rappel : l'induction mathématique (II)

SI

- Cas de base : on établit que la propriété est vraie pour $n = 1$, c'est-à-dire que l'on prouve que $M(1)$ est vraie.

- Cas inductif : en faisant l'hypothèse que la propriété est vraie pour $n = i$, on établit que la propriété est également vraie pour $n = i + 1$.

ALORS

Le principe d'induction affirme que $M(n)$ est vraie pour tout $n \geq 1$.

La logique propositionnelle Procédure de décision - Tableaux sémantiques (XIII)

Définition de la fonction Poids(Ψ). On définit d'abord cette fonction pour une formule ψ par induction sur la structure de la formule.

- $\text{Poids}(p) = 1$.
- $\text{Poids}(\neg\psi) = 1 + \text{Poids}(\psi)$;
- $\text{Poids}(\psi_1 \wedge \psi_2) = 2 + \text{Poids}(\psi_1) + \text{Poids}(\psi_2)$;
- $\text{Poids}(\psi_1 \vee \psi_2) = 2 + \text{Poids}(\psi_1) + \text{Poids}(\psi_2)$;
- $\text{Poids}(\psi_1 \rightarrow \psi_2) = 2 + \text{Poids}(\psi_1) + \text{Poids}(\psi_2)$;
- $\text{Poids}(\psi_1 \leftrightarrow \psi_2) = 2 \times (\text{Poids}(\psi_1) + \text{Poids}(\psi_2)) + 5$

et finalement, $\text{Poids}(\Psi) = \sum_{\psi \in \Psi} \text{Poids}(\psi)$.

La logique propositionnelle Procédure de décision - Tableaux sémantiques (XIII)

Propriétés :

- la fonction Poids() attribue un poids fini et strictement positif à chaque formule;
- le poids de l'étiquette d'un noeud l est toujours strictement supérieur aux poids des étiquette de ses fils.

La logique propositionnelle
Procédure de décision - Tableaux sémantiques (XIV)

Adéquation de l'algorithme :

On doit montrer que "Si T_ϕ est fermé alors ϕ est non satisfaisable".

On montre une propriété plus forte : "Si un sous-arbre de racine l est fermé alors l'ensemble de formules $\mathsf{Lab}(l)$ est non satisfaisable". Pour établir cette propriété, on raisonne par induction sur la profondeur du sous-arbre. (preuve complète donnée au cours.)

La logique propositionnelle
Procédure de décision - Tableaux sémantiques (XIV)

Lemme: Si un sous-arbre de racine l est fermé alors $\mathsf{Lab}(l)$ est non satisfaisable.

Preuve: Par induction sur la profondeur du sous-arbre.

CB: $Prof(l) = 0$. Alors $\mathsf{Lab}(l)$ est un ensemble de littéraux. Si l est fermé, $\mathsf{Lab}(l)$ contient une paire de littéraux complémentaires et donc $\mathsf{Lab}(l)$ est non satisfaisable.

CI: dans ce cas $\mathsf{Prof}(l) > 0$, et l'algorithme a sélectionné $\psi \in \mathsf{Lab}(l)$. Deux cas sont possibles :

1. ψ est une α-formule. Traitons le cas $\psi \equiv \neg\neg\phi$. Dans ce cas, l a une fils m.

On a $\mathsf{Lab}(m) = \mathsf{Lab}(l) \setminus \{\neg\neg\phi\} \cup \{\phi\}$. On sait que le sous-arbre de racine m est fermé (dans le cas contraire l ne pourrait pas l'être). Par **HI**, on a que $\mathsf{Lab}(m)$ est non satisfaisable et donc pour toute valuation V, on a soit $V \not\models \mathsf{Lab}(l)\setminus\{\phi\}$ ou $V \not\models \neg\neg\phi$. Donc pour chaque valuation V on sait que $V \not\models \mathsf{Lab}(l)\setminus\{\phi\}$ ou $V \not\models \phi$. Par conséquent, nous avons bien que pour toute valuation V, $V \not\models \mathsf{Lab}(l)$, et $\mathsf{Lab}(l)$ est non satisfaisable.

2. ψ est une β-formule. Traitons le cas $\psi \equiv \phi_1 \vee \phi_2$. Dans ce cas, l a deux fils m et n. On a $\mathsf{Lab}(m) = \mathsf{Lab}(l) \setminus \{\phi_1 \vee \phi_2\} \cup \{\phi_1\}$ et $\mathsf{Lab}(n) = \mathsf{Lab}(l) \setminus \{\phi_1 \vee \phi_2\} \cup \{\phi_2\}$. Par **HI** on sait que $\mathsf{Lab}(m)$ et $\mathsf{Lab}(n)$ sont non satisfaisables. Donc toute valuation V qui satisfait $\mathsf{Lab}(l) \setminus \{\phi_1 \vee \phi_2\}$ ne satisfait ni ϕ_1 ni ϕ_2, et donc ne satisfait pas $\phi_1 \vee \phi_2$. On a donc bien que $\mathsf{Lab}(l)\setminus\{\phi_1\vee\phi_2\}\cup \{\phi_1 \vee \phi_2\} = \mathsf{Lab}(l)$ est non satisfaisable.

La logique propositionnelle
Procédure de décision - Tableaux sémantiques (XV)

Complétude de l'algorithme :

On doit montrer que "Si ϕ est non satisfaisable alors T_ϕ est fermé".

Prouver cette propriété revient à établir la contraposée : "Si T_ϕ est ouvert alors la formule ϕ est satisfaisable".

Pour établir cela, nous avons besoin de quelques notions supplémentaires.

La logique propositionnelle Procédure de décision - Tableaux sémantiques (XVI)

Un ensemble de formules Ψ (on considère des littéraux, conjonctions ou disjonctions) a la *propriété de Hintikka* ssi les trois propriétés suivantes sont vérifiées :

1. pour toute proposition p, on a soit $p \notin \Psi$ ou $\neg p \notin \Psi$;

2. si $\{\phi_1 \vee \phi_2\} \in \Psi$ alors on a $\phi_1 \in \Psi$ ou $\phi_2 \in \Psi$;

3. si $\{\phi_1 \wedge \phi_2\} \in \Psi$ alors on a $\phi_1 \in \Psi$ et $\phi_2 \in \Psi$.

Propriété des ensembles de Hintikka :

Tout ensemble de formules Ψ qui a la propriété de Hintikka est satisfaisable. (preuve complète donnée au cours.)

Lemme Tout ensemble de formules Ψ qui a la propriété de Hintikka est satisfaisable.

Preuve. On construit une valuation V de la façon suivante :

$$V(p) = Vrai \text{ si } p \in \Psi$$

$$V(p) = Faux \text{ si } \neg p \in \Psi$$

$$\text{et (arbitrairement) } V(p) = Vrai$$
$$\text{si } p \notin \Psi \text{ et } \neg p \notin \Psi.$$

Montrons que toute formule $\phi \in \Psi$ est satisfaite par V.

On raisonne par induction sur la structure de ϕ :

CB :
$$\begin{cases} \phi = p, [\phi]_V = V(p) = Vrai & (p \in \Psi) \\ \phi = \neg p, [\phi]_V = \neg V(p) = Vrai & (\neg p \in \Psi) \end{cases}$$

CI :

- $\phi = \phi_1 \vee \phi_2$.
  Par définition des ensembles de Hintikka on sait que soit $\phi_1 \in \Psi$, soit $\phi_2 \in \Psi$. Par **HI** dans les 2 cas on a $[\phi_1]_V = [\phi_2]_V = Vrai$. Donc $[\phi_1 \vee \phi_2]_V = Vrai$.

- $\phi = \phi_1 \wedge \phi_2$. On sait que $\phi_1 \in \Psi$ et $\phi_2 \in \Psi$. Par **HI** on a $[\phi_1]_V = [\phi_2]_V = Vrai$ et donc $[\phi_1 \wedge \phi_2]_V = Vrai$.

La logique propositionnelle Procédure de décision - Tableaux sémantiques (XVII)

On appelle l'ensemble des noeuds que l'on parcourt depuis la racine d'un arbre jusqu'à une feuille l la *branche* de l, on note $\mathsf{Branche}(A, l)$ la branche qui mène de la racine de A à la feuille l.

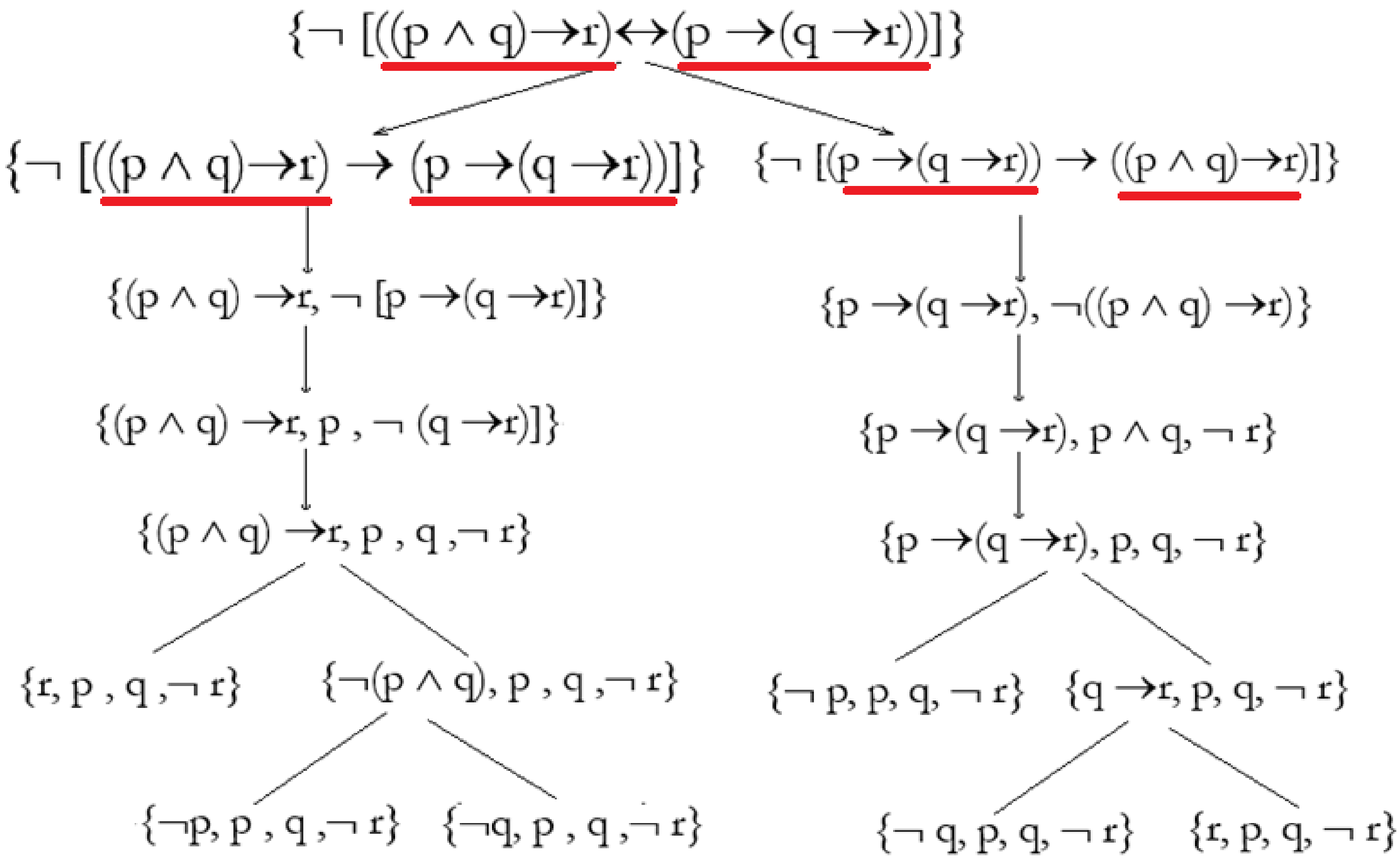
On a le lemme suivant :

Soit l une feuille ouverte de T_ϕ alors

$$\bigcup_{m \in \mathsf{Branche}(A,l)} \mathsf{Lab}(m)$$

est un ensemble de formules qui a la propriété de Hintikka. (preuve complète donnée au cours.)

- Un autre exemple : $\varphi = \neg\, [((p \wedge q) \rightarrow r) \leftrightarrow (p \rightarrow (q \rightarrow r))]$



- φ est non satisfaisable puisque tous les ensembles des littéraux sont non satisfaisables

Preuve. on montre que chaque propriété est vérifiée :

1. Pour tout p, $p \notin \Psi$ ou $\neg p \notin \Psi$. En effet l est une feuille ouverte et donc elle ne contient pas de littéraux complémentaires.

2. Si $\phi_1 \vee \phi_2 \in \Psi$ alors on a appliqué une β-règle et soit $\phi_1 \in \Psi$ ou $\phi_2 \in \Psi$.

3. Si $\phi_1 \wedge \phi_2 \in \Psi$ alors on a appliqué une α-règle et $\phi_1 \in \Psi$ ainsi que $\phi_2 \in \Psi$.

La logique propositionnelle
Procédure de décision - Tableaux sémantiques (XVIII)

On obtient la preuve de complétude par le raisonnement suivant :

Si T_ϕ est ouvert alors il contient une feuille l qui est ouverte et donc

$$\bigcup_{m \in \mathsf{Branche}(A,l)} \mathsf{Lab}(m)$$

est un ensemble de formules qui a la propriété de Hintikka. Donc toutes les formules de

$$\bigcup_{m \in \mathsf{Branche}(A,l)} \mathsf{Lab}(m)$$

sont satisfaisables et en particulier

$$\phi \in \bigcup_{m \in \mathsf{Branche}(A,l)} \mathsf{Lab}(m).$$

La logique propositionnelle

- Syntaxe
- Sémantique
- Procédure de décision
- **Déduction naturelle**
- Principes de résolution

La logique propositionnelle
Déduction naturelle (I)

L'algorithme introduit précédement ne convient pas bien pour faire des raisonnements "à la main" sur des formules propositionnelles. Quand on fait des preuves en mathématique, on utilise des règles qui nous permettent, pas par pas, de déduire des conclusions à partir de prémisses.

La *déduction naturelle* est une formalisation de ce type de raisonnements.

La logique propositionnelle
Déduction naturelle (II)

Supposons données un ensemble de formules $\phi_1, \ldots, \phi_n$ que nous appelons *prémisses* et une formule ψ que nous appelons *conclusion*. Et que l'on désire montrer que ψ peut être dérivée de $\phi_1, \ldots, \phi_n$, on notera cette intension

$$\phi_1, \ldots, \phi_n \vdash \psi$$

cette dernière expression est appelé un *séquent*.

Cette notion de dérivation syntaxique sera intimement liée à la notion sémantique de conséquence logique : $\models$, en effet, on aura :

$$\phi_1, \ldots, \phi_n \vdash \psi \ \underline{\text{ssi}} \ \phi_1, \ldots, \phi_n \models \psi$$

La logique propositionnelle
Déduction naturelle (III)

Plusieurs règles peuvent etre appliquées dans la déduction naturelle

Règle pour la conjonction:

- Introduction :

$$\frac{\phi \quad \psi}{\phi \wedge \psi} \wedge_i$$

- Elimination :

$$\frac{\phi \wedge \psi}{\phi} \wedge_{e_1} \qquad \frac{\phi \wedge \psi}{\psi} \wedge_{e_2}$$

La logique propositionnelle
Déduction naturelle (IV)

Exemples d'utilisation de ces règles pour la conjonction :

- $p \wedge q, r \vdash^{?} q \wedge r$.

1	$p \wedge q$	*prémisse*
2	r	*prémisse*
3	q	$\wedge_{e_2}, 1$
4	$q \wedge r$	$\wedge_i, 3, 2$

- $(p \wedge q) \wedge r, r \wedge t \vdash q \wedge t$;

- $(p \wedge q) \wedge r \vdash p \wedge (q \wedge r)$.

Exemple

- $(p\wedge q)\wedge r, r\wedge t \vdash q \wedge t$

1	$(p\wedge q) \wedge r$	prémisse
2	$r\wedge t$	prémisse
3	$(p\wedge q)$	$\wedge_{e1}$, 1
4	q	$\wedge_{e2}$, 3
5	t	$\wedge_{e2}$, 2
6	$q \wedge t$	$\wedge_{i}$, 4,5

- $(p\wedge q)\wedge r, p\wedge q \vdash p\wedge(q \wedge r)$

1	$(p\wedge q)\wedge r$	prémisse
2	$p\wedge q$	prémisse
3	r	$\wedge_{e2}$, 1
4	q	$\wedge_{e2}$, 2
5	$q \wedge r$	$\wedge_{i}$, 3,4
6	p	$\wedge_{e1}$, 2
7	$p\wedge(q \wedge r)$	$\wedge_{i}$, 5,6

La logique propositionnelle
Déduction naturelle (IV)

Règle pour la (double) négation :

- Introduction :

$$\frac{\phi}{\neg\neg\phi}\neg\neg_i$$

- Elimination :

$$\frac{\neg\neg\phi}{\phi}\neg\neg_e$$

Exemple : $p, \neg\neg(q \wedge r) \vdash \neg\neg p \wedge r$

- $p, \neg\neg(q \wedge r) \vdash \neg\neg p \wedge r$

1	p	prémisse
2	$\neg\neg(q \wedge r)$	prémisse
3	$(q \wedge r)$	$\neg\neg_e$, 2
4	r	$\wedge_{e2}$, 3
5	$\neg\neg p$	$\neg\neg_i$, 1
6	$\neg\neg p \wedge r$	$\wedge_i$, 4,5

La logique propositionnelle
Déduction naturelle (V)

Règle pour l'élimination de l'implication (Modus Ponens):

$$\frac{\phi \quad \phi \rightarrow \psi}{\psi} \rightarrow_e$$

Illustration :

1. il pleut;

2. si il pleut alors la route est mouillée.

alors on peut déduire que la route est mouillée.

La logique propositionnelle
Déduction naturelle (VI)

Etant données les prémisses

1. p;
2. $p \to q$;
3. $p \to (q \to r)$

Peut-on déduire r ?

1	p	prémisse
2	$p \to q$	prémisse
3	$p \to (q \to r)$	prémisse
4	q	$\to_e$,1 ,2
5	$q \to r$	$\to_e$,1 ,3
6	r	$\to_e$,4 ,5

La logique propositionnelle
Déduction naturelle (VII)

Supposons que $p \to q$ et $\neg q$ soient vraies, si p est vrai alors par la règle du *modus ponens* on peut dériver q ce qui est en contradiction avec le fait que $\neg q$ est vrai, donc on en déduit que $\neg p$ est vrai. Ce raisonnement est résumé dans la règle suivante appelée règle de *Modus Tollens* :

$$\frac{\phi \to \psi \qquad \neg\psi}{\neg\phi} MT$$

La logique propositionnelle
Déduction naturelle (VIII)

Prouvons le séquent suivant :

$$p \rightarrow (q \rightarrow r), p, \neg r \vdash \neg q$$

1	$p \rightarrow (q \rightarrow r)$	*prémisse*
2	p	*prémisse*
3	$\neg r$	*prémisse*
4	$q \rightarrow r$	$\rightarrow_e$1, 2
5	$\neg q$	MT4, 3

La logique propositionnelle
Déduction naturelle (XIII)

Règle pour l'introduction de l'implication :

$$\frac{\begin{array}{ll} \phi & \text{hyp.} \\ \cdot & \\ \cdot & \\ \cdot & \\ \psi & \text{fin hyp.} \end{array}}{\phi \to \psi} \to_i$$

La logique propositionnelle
Déduction naturelle (XII)

Le raisonnement à partir d'hypothèses :

On sait que $p \to q$ est vrai. Si on suppose (on fait l'hypothèse) que q est faux alors on peut déduire grâce à la règle MT que p est faux. Donc en supposant $\neg q$ on peut déduire $\neg p$ donc "$\neg q$ implique $\neg p$", exprimé de manière symbolique $\neg q \to \neg p$ est déduit.

L'argument pour $p \to q \vdash \neg q \to \neg p$ peut être résumé de la façon suivante:

1	$p \to q$	prémisse
2	$\neg q$	hyp.
3	$\neg p$	MT1, 2, fin hyp.2
4	$\neg q \to \neg p$	$\to_i$, 2 − 3

La logique propositionnelle
Déduction naturelle (XIV)

Prouvons le séquent $\neg q \to \neg p \vdash p \to \neg\neg q$:

1	$\neg q \to \neg p$	prémisse
2	p	hyp.
3	$\neg\neg p$	$\neg\neg_i, 2$
4	$\neg\neg q$	$MT, 1, 3,$ fin hyp.2
5	$p \to \neg\neg q$	$\to_i, 2-4$

La logique propositionnelle
Déduction naturelle (XV)

Preuves de formules vraies sans prémisse :

$$\vdash p \to \neg\neg p$$

1	p	hyp.
2	$\neg\neg p$	$\neg\neg_i, 1,$ fin hyp.1
3	$p \to \neg\neg p$	$\to_i, 1-2$

Note : les formules que l'on peut établir sans prémisses sont les formules valides.

La logique propositionnelle
Déduction naturelle (XVI)

Prouvons le séquent suivant :

$$\vdash^? (q \to r) \to ((\neg q \to \neg p) \to (p \to r))$$

1	$q \to r$	hyp.
2	$\neg q \to \neg p$	hyp.
3	p	hyp.
4	$\neg\neg p$	$\neg\neg_i, 3$
5	$\neg\neg q$	$MT 2, 4$
6	q	$\neg\neg_e 5$
7	r	$\to_e 1, 6$, fin hyp.3
8	$p \to r$	$\to_i 3-7$, fin hyp.2
9	$(\neg q \to \neg p) \to (p \to r)$	$\to_i 2, 8$, fin hyp.1
10	$(q \to r)$ $\to ((\neg q \to \neg p) \to (p \to r))$	$\to_i 1, 9$

La logique propositionnelle
Déduction naturelle (XVII)

Règle pour introduire la disjonction :

$$\frac{\phi}{\phi \lor \psi} \lor i_1 \qquad\qquad \frac{\psi}{\phi \lor \psi} \lor i_2$$

La logique propositionnelle
Déduction naturelle (XVIII)

Comment éliminer la disjonction ?

Imaginons que l'on puisse :

1. établir une formule ϕ sous hypothèse que ψ_1 est vrai,

2. et que l'on puisse également établir ϕ sous hypothèse que ψ_2 est vrai,

3. et qu'il existe une preuve pour $\psi_1 \vee \psi_2$

alors on devrait pouvoir déduire ϕ.

Cette stratégie de preuve est formalisée dans la règle suivante :

$$\frac{\psi_1 \vee \psi_2 \quad \begin{array}{c}\psi_1\text{hyp.}\\ \cdot\\ \cdot\\ \cdot\\ \phi\text{fin hyp.}\end{array} \quad \begin{array}{c}\psi_2\text{hyp.}\\ \cdot\\ \cdot\\ \cdot\\ \phi\text{fin hyp.}\end{array}}{\phi} \vee_e$$

Attention : dans chacun des deux cas, on ne peut pas utiliser l'hypothèse temporaire faite pour l'autre cas, sauf si cette hypothèse a été établie avant.

La logique propositionnelle
Déduction naturelle (XIX)

Prouvons le séquent

$$p \vee q \vdash q \vee p$$

1	$p \vee q$	prémisse
2	p	hyp.
3	$q \vee p$	$\vee_{i_2}$, fin hyp.2
4	q	hyp.
5	$q \vee p$	$\vee_{i_1}$, fin hyp.4
6	$q \vee p$	$\vee_e 1, 2-3, 4-5$

La logique propositionnelle
Déduction naturelle (XX)

Règle de copie pour conclure un raisonnement sous-hypothèse.

$$\frac{\phi}{\phi} copie$$

Considérons le séquent :

$$\vdash p \rightarrow (q \rightarrow p)$$

Et la preuve suivante :

1	p	hyp.
2	q	hyp.
3	p	$copie$ 1, fin hyp.2
4	$q \rightarrow p$	$\rightarrow_i$ 2, 3, fin hyp.1
5	$p \rightarrow (q \rightarrow p)$	$\rightarrow_i$ 1 − 4

Déduction naturelle (XXI)

Règles pour la négation.

Les contradictions sont des formules de la forme:

$$\neg\phi \wedge \phi \text{ ou } \phi \wedge \neg\phi$$

Notons qu'aucune valuation ne satisfait une contradiction. Donc on a $\neg\Phi\wedge\Phi \models \psi$ pour n'importe quel ψ. Donc si la méthode de déduction est complète, on devrait avoir $\neg\Phi\wedge\Phi \vdash \psi$ pour n'importe quel ψ.

Le fait que l'on peut déduire tout à partir d'une contradiction est formalisé par la règle suivante :

$$\frac{\perp}{\phi}\perp_e$$

Le fait que $\perp$ represente une contradiction est formalisé par la règle suivante :

$$\frac{\phi \qquad \neg\phi}{\perp}\neg_e$$

Comment introduire une négation ?

Supposons que l'on fasse une hypothèse et que l'on arrive a déduire une contradiction, dans ce cas l'hypothèse est fausse. Ceci est formalisé par la règle de preuve suivante :

$$\frac{\begin{array}{ll} \phi & \text{hyp.} \\ . & \\ . & \\ . & \\ \perp & \text{fin hyp.} \end{array}}{\neg\phi}\neg_i$$

La logique propositionnelle
Déduction naturelle (XXI)

On est maintenant en mesure de prouver le sequent de l'exemple introductif (taxi-train-invité):

$$p \wedge \neg q \rightarrow r, \neg r, p \vdash q$$

1	$p \wedge \neg q \rightarrow r$	prémisse
2	$\neg r$	prémisse
3	p	prémisse
4	$\neg q$	hyp.
5	$p \wedge \neg q$	$\wedge_i 3, 4$
6	r	$\rightarrow_e 1, 5$
7	$\perp$	$\neg_e 6, 2,$ fin hyp.4
8	$\neg\neg q$	$\neg_i 4 - 7$
9	q	$\neg\neg_e 8$

La logique propositionnelle
Déduction naturelle (XXII)

Règles dérivées :

Notons que MT

$$\frac{\phi \to \psi \quad \neg\psi}{\neg\phi} MT$$

est une règle dérivée. En effet :

1	$\phi \to \psi$	prémisse
2	$\neg\psi$	prémisse
3	ϕ	hyp.
4	ψ	$\to_e 1, 3$
5	$\perp$	$\neg_e 4, 2,$ fin hyp.3
6	$\neg\phi$	$\neg_i 3 - 5$

La logique propositionnelle

- Syntaxe
- Sémantique
- Procédure de décision
- Déduction naturelle
- **Principes de résolution**

La logique propositionnelle
Résolution (I)

Rappel : Une formule propositionnelle ϕ est un *littéral* si et seulement si ϕ est une proposition ou la négation d'une proposition.

Donc, p et $\neg p$ sont des littéraux.

Une formule propositionnelle ϕ est une *clause* si $\phi = \psi_1 \vee \psi_2 \vee \ldots \vee \psi_n$ où $n \geq 1$ et chaque ψ_i, $1 \leq i \leq n$, est un littéral.

Par exemple, $p \vee \neg q \vee r$ est une clause.

La logique propositionnelle
Résolution (II)

Une clause

$$\neg a_1 \vee \neg a_2 \vee \ldots \vee \neg a_m \vee b_1 \vee b_2 \vee \ldots \vee b_n$$

où $a_1, a_2, \ldots, a_m, b_1, b_2, \ldots, b_n$ sont des propositions peut être écrite comme l'implication:

$$a_1 \wedge a_2 \wedge \ldots \wedge a_m \rightarrow b_1 \vee b_2 \vee \ldots \vee b_n$$

ou encore comme une paire d'ensembles (Γ, Δ) où $\Gamma = \{a_1, a_2, \ldots, a_m\}$ et $\Delta = \{b_1, b_2, \ldots, b_n\}$. Donc Γ contient l'ensemble des propositions qui apparaissent négativement dans la clause, alors que Δ contient l'ensemble des propositions qui apparaissent positivement dans la clause.

Pour toute formule ϕ il existe un ensemble fini de clauses equivalent.

La logique propositionnelle
Résolution (IV)

On dit qu'une clause (Γ, Δ) est

- *négative* si $\Delta = \emptyset$;
- *positive* si $\Gamma = \emptyset$;
- *vide* si $\Gamma = \Delta = \emptyset$;
- *tautologique* si $\Gamma \cap \Delta \neq \emptyset$.

La logique propositionnelle
Résolution (V)

La fonction d'interprétation V satisfait la clause $C = (\Gamma, \Delta)$ <u>si et seulement si</u> il existe $a \in \Gamma$ tel que $V(a) =$ faux <u>ou</u> si il existe $b \in \Delta$ tel que $V(b) =$ vrai. Formellement $[\![(\Gamma, \Delta)]\!]_V =$ vrai <u>ssi</u> $\exists a \in \Gamma : V(a) =$ faux <u>ou</u> $\exists a \in \Delta : V(a) =$ vrai.

Et donc,

Une clause vide n'est satisfaite par aucune fonction d'interprétation.

Une clause tautologique est satisfaite par toutes les fonctions d'interprétation.

La logique propositionnelle
Résolution (VI)

Nous étendons maintenant les notions de satisfaction, satisfaisabilité et validité aux ensembles de clauses:

Un ensemble de clauses S est satisfait par la fonction d'interprétation V si et seulement si on a que $V \Vdash C$ pour toute clause $C \in S$.

Un ensemble de clauses S est satisfaisable si et seulement si il existe une fonction d'interprétation qui satisfait S.

Un ensemble de clauses S est valide si et seulement si toute fonction d'interprétation V satisfait S.

La logique propositionnelle
Résolution (VII)

Et donc,

L'ensemble vide de clause est valide et donc satisfaisable.

Soit S un ensemble de clauses et C une clause tautologique, $S \cup \{C\}$ est satisfaisable (respectivement valide) si et seulement si S est satisfaisable (respectivement valide).

Ce dernier corollaire nous indique que lorsque l'on veut verifier la satisfaisabilité ou la validité d'un ensemble de clause, on peut toujours supprimer les clauses tautologiques.

La logique propositionnelle
Résolution (VIII) : Règle de coupure

Etant données deux clauses $C_1 = (\Gamma_1, \Delta_1)$ et $C_2 = (\Gamma_2, \Delta_2)$ et une proposition p tels que $p \in \Delta_1$ et $p \in \Gamma_2$, c'est-à-dire p apparait positivement dans C_1 et négativement dans C_2, on obtient la clause $C_3 = (\Gamma_3, \Delta_3)$ par coupure avec p où :

- $\Gamma_3 = \Gamma_1 \cup (\Gamma_2 \setminus \{p\})$;

- $\Delta_3 = (\Delta_1 \setminus \{p\}) \cup \Delta_2$.

On note

$$C_1, C_2 \vdash_p^c C_3$$

le fait que C_3 est déductible de C_1, C_2 par coupure sur la proposition p.

Exemple

- $C1 = p \vee \neg q \vee r \quad \Rightarrow \Gamma 1 = \{q\}, \Delta 1 = \{p, r\}$
- $C2 = \neg p \vee t \quad \Rightarrow \Gamma 2 = \{p\}, \Delta 2 = \{t\}$

$$C_1, C_2 \vdash_p^c C_3$$

- $\Gamma 3 = \Gamma 1 \cup (\Gamma 2 \setminus \{p\}) = \{q\}$
- $\Delta 3 = (\Delta 1 \setminus \{p\}) \cup \Delta 2 = \{r, t\}$

$\Rightarrow C3 = \neg q \vee r \vee t$

La logique propositionnelle
Résolution (IX) : Règle de coupure (suite)

Théorème : Si $C_3 = (\Gamma_3, \Delta_3)$ est une clause obtenue par la règle de coupure à partir des clauses $C_1 = (\Gamma_1, \Delta_1)$ et $C_2 = (\Gamma_2, \Delta_2)$ alors C_3 est une conséquence logique de C_1 et C_2.

si $C_1, C_2 \vdash_p^c C_3$ alors $C_1, C_2 \models C_3$.

La logique propositionnelle
Résolution (IX) : Règle de coupure (suite)

Théorème : Si $C_3 = (\Gamma_3, \Delta_3)$ est une clause obtenue par la règle de coupure à partir des clauses $C_1 = (\Gamma_1, \Delta_1)$ et $C_2 = (\Gamma_2, \Delta_2)$ alors C_3 est une conséquence logique de C_1 et C_2.

si $C_1, C_2 \vdash_p^c C_3$ alors $C_1, C_2 \models C_3$.

Preuve. Pour montrer que la règle de coupure est adéquate, nous devons montrer que si $C_1, C_2 \vdash_p^c C_3$ alors $C_1, C_2 \models C_3$. Par définition de $\models$, cela revient à montrer que pour toute fonction d'interprétatic V telle que $V \Vdash C_1$ et $V \Vdash C_2$, on a $V \Vdash C_3$. Par contraposition, cela revient à montrer que si $V \not\Vdash C_3$ alors soit $V \not\Vdash C_1$ ou $V \not\Vdash C_2$. Par définition de $\vdash_p^c$, nous savons que $\Gamma_3 = \Gamma_1 \cup (\Gamma_2 \setminus \{p\})$ et $\Delta_3 = (\Delta_1 \setminus \{p\}) \cup \Delta_3$. On sait également que si V ne satisfait pas C_3, alors

$$\forall q \in \Gamma_3 : V(q) = \text{vrai et } \forall q \in \Delta_3 : V(q) = \text{faux}$$

Par définition de Γ_3 et Δ_3, on a en particulier:

- $\forall q \in \Gamma_1 : V(q) =$ vrai;
- $\forall q \in \Gamma_2 \setminus \{p\} : V(q) =$ vrai;
- $\forall q \in \Delta_1 \setminus \{p\} : V(q) =$ faux;
- $\forall q \in \Delta_2 : V(q) =$ faux.

Il suffit maintenant de considérer les deux valeurs possible pour $V(p)$:

- <u>si</u> $V(p) =$ vrai <u>alors</u> on a $\forall q \in \Gamma_2 =$ V(q) = vrai donc on a $V \not\Vdash C_2$;
- <u>si</u> $V(p) =$ faux <u>alors</u> on a $\forall q \in \Delta_1 =$ V(q) = faux donc on a $V \not\Vdash C_1$;

♣

La logique propositionnelle
Résolution (X)

Soit S un ensemble de clauses et C une clause. Une preuve de C par coupure à partir de S est une suite finie de clauses $C_1, C_2, \ldots, C_n$ où $C_n = C$ et pour tout i, $1 \leq i \leq n$, on a :

- soit $C_i \in S$;
- ou il existe k, l, $1 \leq k < l < i$ et une proposition p tels que $C_k, C_l \vdash_p^c C_i$.

On note $S \vdash^c C$ le fait que C est déductible de S par coupure.

La logique propositionnelle
Résolution (XI)

Théorème : Pour tout ensemble de clauses S et pour toute clause C, si $S \vdash^c C$ alors $S \models C$.

Pour tout ensemble de clauses S et pour toute clause C tel que $S \models C$ alors on a $S \vdash^c C$.

La logique propositionnelle
Résolution (XII) : Complétude de la méthode de preuve par coupure

Une *réfutation* d'un ensemble S de clauses par coupure est une dérivation de la clause vide à partir de S.

Théorème : Si il existe un réfutation de S par coupure alors l'ensemble de clauses S est non satisfaisable.

La logique propositionnelle
Résolution (XIII)

Comment peut-on se servir de la notion de réfutation pour vérifier qu'une formule ϕ est une conséquence logique d'un ensemble fini de formules

$$\psi_1, \psi_2, \ldots, \psi_n \overset{?}{\models} \phi$$

Il suffit de procéder de la façon suivante:

1. transformer la conjonction $\psi_1 \wedge \psi_1 \wedge \ldots \wedge \psi_n$ en un ensemble fini de clauses S (cette transformation est toujours possible);

2. de la même façon, transformer $\neg\phi$ en un ensemble de clauses S';

3. déduire par coupure l'ensemble de tous les clauses D déductibles de $S \cup S'$.

La logique propositionnelle
Résolution (XIV)

Si la clause vide appartient à l'ensemble D alors l'ensemble D n'est pas satisfaisable et donc l'ensemble $S \cup S'$ ne l'est pas non plus, ce qui revient à dire que la formule

$\psi_1 \wedge \psi_2 \wedge \dots \wedge \psi_n \wedge \neg\phi$ n'est pas satisfaisable

et donc $\qquad \psi_1, \psi_2, \dots, \psi_n \dot{\models} \phi$

ϕ est une conséquence logique de l'ensemble $\{\psi_1, \psi_2, \dots, \psi_n\}$.

Exemple

Démontrez que $\alpha, \varphi \models \psi$ par la méthode de résolution propositionnelle où $\alpha = p \rightarrow (q \vee r)$, $\varphi = \neg r$ et $\psi = \neg q \rightarrow \neg p$.

Donc il faut demontrer que $\alpha \wedge \varphi \wedge \neg \psi$ est non satisfaisable.

1. Construire l'ensemble S de clauses des formules α et φ
   - $\alpha = p \rightarrow (q \vee r) = \neg p \vee (q \vee r) = \neg p \vee q \vee r$
   - $\varphi = \neg r$ (elle est déjà une clause)
   - $S = \{\neg p \vee q \vee r, \neg r\}$
2. Construire l'ensemble S' de clauses des formules $\neg \psi$
   - $\neg \psi = \neg (\neg q \rightarrow \neg p) = \neg (\neg \neg q \vee \neg p) = \neg q \wedge p$
   - $S' = \{\neg q , p\}$
3. $S \cup S' = \{\neg p \vee q \vee r, \neg r, \neg q , p\}$

Exemple

$C1 : \neg p \vee q \vee r$

$C2 : \neg r,$

$C3 : \neg q$

$C4 : p$

$C5 : q \vee r$ Res(C1, C4) $(C_1, C_4 \vdash_p^c q \vee r)$

$C6 : r$ Res(C3, C5) $(C_3, C_5 \vdash_q^c r)$

$C7 : \square$- Res(C2, C6) $(C_2, C_6 \vdash_r^c [])$

D contient la clause vide alors $S \cup S'$ est non satisfaisable
Cela veut dire que la formule $\alpha \wedge \varphi \wedge \neg \psi$
est non satisfaisable. Finalement, on peut dire que ψ est une conséquence logique de α et φ.

Résumé

On a étudié dans ce chapitre :

- Les formes normales conjonctives et disjonctives
- Etude de la satisfaisabilité (ou satisfiabilité), de la validité des formules, prouver qu’une formule est une conséquence logique d’autres formules:
  - La méthode de la table de vérité
  - La méthode du tableau sémantique ou de Beth
  - La méthode de déduction naturelle
  - Le principe de résolution

Résumé - La méthode de la table de vérité

La méthode de la table de vérité

Toutes les interprétations possibles V

p_1	p_2	p_3	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ψ
V	V	v	V	V	V	V	V
V	V	F	V	F	V	F	F
F	F	V	V	V	V	V	V
V	F	F	F	V	F	V	F
F	V	V	F	F	F	F	F
F	V	F	V	V	V	V	V
F	F	V	V	V	F	F	F
F	F	F	V	V	V	V	V

- Une formule est valide si elle est vraie dans toutes les interprétations (les 8 lignes)
- Une formule est satisfaisable s'il existe une interprétation tel que la formule est vraie
- Elle non satisfaisable si elle n'est pas vraie dans aucune ligne
- Pour démontrer que $\phi_{1,} \ldots \phi_4$ |= ψ on doit démontrer que:

  $\forall$V une interprétation, si V satisfait tous les ϕ_i ($V \Vdash \phi_i$ ou $[[\phi_i]]_V$ = vrai) alors V doit satisfaire ψ (c-à-d $[[\psi]]_V$ = vrai)

Résumé - La méthode du tableau sémantique (ou de Beth)

La méthode du tableau sémantique (ou de Beth)

- Pour étudier la validité et la satifaisabilité d'une formule
  - Une formule ϕ est non satisfaisable si le tableau T_ϕ est fermé
  - Une formule ϕ est valide si le tableau $T_{\neg\phi}$ est fermé

- Pour démontrer $\phi_1, \ldots \phi_n \models \psi$ on démontre que $\phi_1 \wedge \phi_2 \wedge \ldots \wedge \phi_n \wedge \neg\psi$ est non satisfaisable.

  - *adéquation* : si l'algorithme construit un tableau qui ferme pour une formule ϕ alors la formule ϕ est non satisfaisable;

  - *complétude* : pour toute formule ϕ non satisfaisable, l'algorithme construit un tableau fermé.

Résumé - La méthode de la déduction naturelle

La méthode de la déduction naturelle

- Démontrer les séquents (conséquence logique des formules) par une méthode syntaxique
  - On applique successivement des règles de dérivé ou d'inférence
- On a la théorème d'adéquation et de complétude

$$\phi_1, \ldots, \phi_n \vdash \psi \ \underline{\text{ssi}} \ \phi_1, \ldots, \phi_n \models \psi$$

Résumé - La méthode de résolution

La méthode de résolution

- Pour étudier la validité et la satisfaisabilité d'une formule
- On peut démontrer la conséquence logique des formules
  - Transformant la formule en forme normale conjonctive
    - Un ensemble de clauses
  - Appliquer la règle de coupure successivement sur les clauses
  - Si on a une réfutation alors la formule est non satisfaisable
- On a les théorèmes d'adéquation et de complétude

Théorème : Pour tout ensemble de clauses S et pour toute clause C, <u>si</u> $S \vdash^c C$ <u>alors</u> $S \models C$.

Pour tout ensemble de clauses S et pour toute clause C tel que $S \models C$ alors on a $S \vdash^c C$.

La logique propositionnelle
Rappel : l'induction mathématique (I)

Il est bien connu que pour $n \geq 1$, on a

$$1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}$$

Pour démontrer que cette égalité, notée $M(n)$, est valide pour n'importe quel $n \geq 1$, on applique le *principe d'induction mathématique*.

La logique propositionnelle
Rappel : l'induction mathématique (II)

SI

- Cas de base : on établit que la propriété est vraie pour $n = 1$, c'est-à-dire que l'on prouve que $M(1)$ est vraie.

- Cas inductif : en faisant l'hypothèse que la propriété est vraie pour $n = i$, on établit que la propriété est également vraie pour $n = i + 1$.

ALORS

Le principe d'induction affirme que $M(n)$ est vraie pour tout $n \geq 1$.