Références

- BENZAKEN, *Systèmes formels: Introduction à la logique et à la théorie des langages*, Masson 1991
- BEAUQUIER, BERSTEL, CHRETIENNE, *Éléments d'algorithmique*, Masson 1992
- AHO, ULLMAN, *Concepts fondamentaux de l'informatique*, Dunod 1993
- BELLOT, SAKAROVITCH, *Logique et automates*, Ellipses 1998
- J. HOPCROFT et J. ULLMAN, Introduction to automata theory, languages and computation, Addison-Wesley.
- J. STERN, Fondements mathématiques de l'informatique, McGraw-Hill.
- JACQUEMIN, *Logique et mathématiques pour l'informatique et l'IA*, Masson 1994
- LASSAIGNE, ROUGEMONT, *Logique et fondements de l'informatique*, Hermés 1993

Logique mathématique pour lôinformatique

Contenu

- Logique propositionnelle
  - Syntaxe, sémantique de la logique propositionnelle
  - Valuation, validité, satisfaisabilité
  - Tables de vérité
  - Forme normales
  - Méthodes et systèmes formels de preuve
  - Calcul naturel des propositions
- Logique des prédicats
  - Syntaxe et sémantique de la logique du premier ordre
  - Principes généraux du premier ordre
  - Systèmes formels du premier ordre

Logique ?

- Logique = formalisation du raisonnement dans le but de distinguer raisonnements corrects et incorrects
- Raisonnement humain = ensemble des processus intellectuels conduisant de faits à une conclusion
  - Caractéristique: difficile à vérifier, et donc à suivre
  - Ex: ~~S’il pleut je prends mon parapluie, Il ne pleut pas si le chien du voisin est dehors; Lorsqu’il est dehors, ce chien aboie. Donc, lorsque le chien du voisin aboie, je ne prends pas mon parapluie.~~ **Le chien peut aboyer dedans aussi !**
- Raisonnement formel = suite de propositions enchaînées selon des règles précises conduisant de prémisses à une conclusion
  - Caractéristique: possible de valider, automatisable
  - Ex: x=0 et y=x => y=0

Historique

- Aristote (-350): la logique comme la science de distinguer le vrai du faux
- Leibniz (1680): la « caractéristique universelle », sorte de langage pour passer du vrai au vrai
- Boole (1850): algébrisation de la logique 0=Faux, 1=Vrai
- Frege/Peano/Russel (~1900): logique comme fondement indubitable et justification des mathématiques (logistique)
- Gödel (~1930): impossibilité de formaliser complètement l'arithmétique (et donc les mathématiques, donc le raisonnement en général)

Utilité

Å Raisonnement automatique

Å Conception/vérification de circuits numériques

Å Preuve de programmes

Å Extraction de connaissances

Å Systèmes experts

Å é

Et des liens avec les théories :

Å des langages formels (formalisation du langage)

Å de la calculabilité (limites de lôinformatique)

Å de la complexité (efficacité de lôinformatique)

Les logiques

- Logiques classiques:
  - Logique dôordre ôzéroô (calcul des propositions)
  - Logique de premier ordre (calcul des prédicats)
  - Logiques dôordres supérieurs
- Logiques non-classiques
  - Logique modale
  - Logique floue
  - Logique non-monotone
  - Logique temporelle, directionnelle, é

Dans ce cours, on ne parlera que de logique des propositions et de la logique des prédicats

Syntaxe

- La logique propositionnelle est composée de formules qui vérifient une certaine syntaxe
- On peut définir cette syntaxe par une grammaire :
  - $V_T = V \cup \{$ó¬ô, ó∧ô, ó∨ô, ó⇒ô, ó⇔ô, ó(ô, ó)ô$\}$
  - $V_N = \{$óPropositionô, óAtomeô, óConnecteurô$\}$
  - S = óPropositionô
  - R = { Proposition → Atome | (Proposition) | ¬ Proposition
    | Proposition Connecteur Proposition
    Connecteur → ∧ | ∨ | ⇒ | ⇔
    Atome → x_1 | x_2 | é | x_n (avec $x_i \in V$)
    }

Syntaxe

- Propriétés :
  - V est un ensemble de symboles qui ne contient pas {ó¬ô, ó∧ô, ó∨ô, ó⇒ô, ó⇔ô, ó(ô, ó)ô} ni aucun symbole de V_N
  - Les symboles de V sont appelés **variables propositionnelles**
  - Les symboles ó¬ô, ó∧ô, ó∨ô, ó⇒ô et ó⇔ô sont appelés **connecteurs logiques**; ó¬ô est un connecteur **unaire**; ó∧ô, ó∨ô, ó⇒ô et ó⇔ô sont des connecteurs **binaires**
- Notation :
  - Variables propositionnelles : **V={a,b,c,é ,z,A,B,é ,Z}**
  - Formules logique : Φ ou Ψ, avec indices éventuels
- Exemples de formules syntaxiquement correctes :

$$\Phi_1 = a \Rightarrow b \wedge (a \vee b)$$

$$\Phi_2 = (\text{beau} \Rightarrow \text{sors}) \wedge \neg\, \text{sors} \Rightarrow \neg\, \text{beau}$$

Sémantique

Å La syntaxe indique si une formule est bien formée, pas si elle représente un raisonnement correct

Å On va donc associer une signification aux formules de la logique propositionnelle

=> Il faut se munir dôune interprétation des symboles qui permettra dôassocier une sémantique à la syntaxe : côest le **calcul booléen** (ou propositionnel)

Sémantique

- Interprétation des symboles de la logique propositionnelle :
  - $\forall x \in V$, x est une variable propositionnelle qui peut prendre deux valeurs: **vrai** (noté **V**) ou **faux** (noté **F**)
  - $\neg$ signifie « **non** » (**négation**)
  - $\wedge$ signifie « **et** » (**conjonction**)
  - $\vee$ signifie « **ou** » (**disjonction**)
  - $\Rightarrow$ signifie « **implique** » (**implication**)
  - $\Leftrightarrow$ signifie « **équivalent** » (**équivalence**)
  - les "(" et ")" servent seulement à regrouper les termes

<u>Ex:</u> $((a \Rightarrow b) \wedge (\neg a \vee b)) \Leftrightarrow (a \Rightarrow b)$

signifie « a implique b et non a ou b est équivalent à a implique b »

Sémantique

- Pour lever les ambiguïtés d'interprétation des formules, on utilise les parenthèses, et la priorité des connecteurs suivante :

non > et > ou > implique = équivalent

$$\neg \;>\; \wedge \;>\; \vee \;>\; \Rightarrow \;=\; \Leftrightarrow$$

Ex:

$a \wedge \neg b \Leftrightarrow \neg c \vee d \;\equiv\; (a \wedge (\neg b)) \Leftrightarrow ((\neg c) \vee d)$

$\neg a \wedge b \Leftrightarrow \neg c \Rightarrow \neg d \wedge e \;\equiv\; (((\neg a) \wedge b) \Leftrightarrow (\neg c)) \Rightarrow ((\neg d) \wedge e)$?

$\equiv ((\neg a) \wedge b) \Leftrightarrow ((\neg c) \Rightarrow ((\neg d) \wedge e))$?

=> Les parenthèses sont utiles !

Raisonnement → Formule logique

Å Raisonnement en langue naturelle → formule :

Å Repérer les connecteurs entre les propositions P et Q:

Å P [et / mais / , / . / ;] Q $\leftrightarrow$ $P \wedge Q$

Å P [ou / sinon] Q $\leftrightarrow$ $P \vee Q$

Å Si P alors Q / Q si P/ P donc Q / Lorsque P, Q $\leftrightarrow$ $P \Rightarrow Q$

Å P si et seulement si Q $\leftrightarrow$ $P \Leftrightarrow Q$

Å Transformer les propositions en variables propositionnelles
Attention : minimiser le nombre de variables en utilisant la négation

Ex : « sôil fait beau alors je sors; il pleut; donc je reste chez moi »
Variables propositionnelles: P = «il fait beau», Q=«je sors»
Formulation du raisonnement: $(P\Rightarrow Q)\wedge\neg P\Rightarrow\neg Q$

Variables propositionnelles

- Une variable propositionnelle représente donc un élément de raisonnement qui peut-être vrai ou faux
- On appelle **valuation** d'une variable propositionnelle P le fait de donner une valeur à la variable. On note **P=V** et **P=F**
- **Var(Φ)** est l'ensemble des variables propositionnelles de Φ
- On appelle **interprétation** de Φ une valuation de toutes ses variables Var(Φ)
- La **valeur de vérité** d'une formule Φ dépend de l'interprétation choisie. Pour la déterminer, on utilise les **tables de vérité** qui à tout connecteur associent une valeur de vérité dépendant des valuations des formules connectées

Tables de vérité des connecteurs

- Tables de vérité des connecteurs de la logique propositionnelle :

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
F	F	V	F	F	V	V
F	V	V	F	V	V	F
V	F	F	F	V	F	F
V	V	F	V	V	V	V

Évaluation dôune formule

Å Étant donnée une interprétation I de Φ, on peut calculer sa **valeur de vérité $I(\Phi)$** en évaluant récursivement ses termes à partir des atomes

Ex:

Évaluons la formule $\Phi = a \Rightarrow b \wedge (a \vee b)$ pour lôinterprétation $I=\{a=V, b=F\}$

a	b	$a \vee b$	$b \wedge (a \vee b)$	$a \Rightarrow b \wedge (a \vee b)$
V	F	V	F	F

La valeur de vérité de Φ pour cette interprétation est faux

Tables de vérité

- La **table de vérité T(Φ)** d'une formule Φ réunit l'ensemble des valeurs de vérité de Φ pour toutes ses interprétations possibles

  Ex: La table de vérité de la formule $\Phi = a \Rightarrow b \wedge (a \vee b)$ est

a	b			$a \Rightarrow b \wedge (a \vee b)$
F	F			V
F	V			V
V	F			F
V	V			V

- Une formule Φ à *n* variables admet 2^n interprétations distinctes; sa table de vérité comporte donc 2^n lignes

Propriétés des formules

- Deux formules Φ et Ψ sont **synonymes** (noté $\Phi \equiv \Psi$) ssi
  - $Var(\Phi) = Var(\Psi) = Var$ même ensemble de variables
  - $\forall$ I sur Var, $I(\Phi) = I(\Psi)$ même valeur de vérité pour toute interprétation

- Une formule Φ est
  - **satisfaisable** ssi $\exists I, I(\Phi)=V$ une interprétation la rend vraie
  - **réfutable** ssi $\exists I, I(\Phi)=F$ une interprétation la rend fausse
  - une **tautologie** ssi $\forall I, I(\Phi)=V$ toute interprétation la rend vraie
  - une **antilogie** ssi $\forall I, I(\Phi)=F$ toute interprétation la rend fausse

- Relation entre synonymie et équivalence :
  - $(\Phi \equiv \Psi) \Rightarrow (\Phi \Leftrightarrow \Psi)$ par définition
  - $(\Phi \Leftrightarrow \Psi) \not\Rightarrow (\Phi \equiv \Psi)$ car $Var(\Phi) \neq Var(\Psi)$ possible

Propriétés des formules

Å Exemple :

La formule $\Phi = a \Rightarrow b \wedge (a \vee b)$ dont la table de vérité est

a	b	$a \Rightarrow b \wedge (a \vee b)$	$a \Rightarrow b$
F	F	V	V
F	V	V	V
V	F	F	F
V	V	V	V

Å est satisfaisable (lôinterprétation {a=F,b=F} la rend vraie)

Å est réfutable (lôinterprétation {a=V,b=F} la rend fausse)

Å nôest pas une tautologie ni une antilogie

Å est synonyme de la formule $a \Rightarrow b$

Validité dÕun raisonnement

- Un raisonnement est **valide** ssi sa formulation en logique propositionnelle est une tautologie

  En effet, si sa formulation nÕest pas une tautologie, elle admet une interprétation qui la réfute. Une telle interprétation est un **contre-exemple** au raisonnement

- On peut donc déterminer la validité dÕun raisonnement en construisant la table de vérité de sa formulation logique

Validité dôun raisonnement

Å Exemple:

«Sôil pleut je prends mon parapluie; Il ne pleut pas si le chien du voisin est dehors; Lorsquôil est dehors, ce chien aboie. Donc, lorsque le chien du voisin aboie, je ne prends pas mon parapluie.»

Se formule

$\Phi = (p \Rightarrow u) \wedge (d \Rightarrow \neg p) \wedge (d \Rightarrow a) \Rightarrow (a \Rightarrow \neg u)$

dont la table de vérité est donnée ci-contre

p	u	d	a	$p \Rightarrow u$	$d \Rightarrow \neg p$	$d \Rightarrow a$	$a \Rightarrow \neg u$	Φ
é	é	é	é	é	é	é	é	V
F	**V**	**F**	**V**	**V**	**V**	**V**	**F**	**F**
é	é	é	é	é	é	é	é	V
F	**V**	**V**	**V**	**V**	**V**	**V**	**F**	**F**
é	é	é	é	é	é	é	é	V
V	**V**	**F**	**V**	**V**	**V**	**V**	**F**	**F**
é	é	é	é	é	é	é	é	V

Ce raisonnement nôest donc pas valide: il admet trois contre-exemples

Validité dôun raisonnement

Å Lôapproche par table de vérité a ses limites : 2^n lignes, ce nôest rapidement plus calculable, même par une machine:

  Å $2^5 = 32$, encore gérable à la main

  Å $2^{10} = 1024$, faisable en machine mais plus à la main

  Å $2^{30} = 1\ 073\ 741\ 824$, limite même en machine

  Å 2^{90} > temps en nanosecondes écoulé depuis la naissance de lôunivers (>13,7 milliards dôannées)

=> Il faut trouver dôautres approches de validation dôun raisonnement

  Å Ramener à un raisonnement plus simple équivalent

  Å Comparer à un raisonnement connu valide

  Å Utiliser des mécanismes de déduction automatique

Simplification de formules

- Simplification = remplacer une sous-formule par une sous-formule équivalente
- Pour toutes formules Φ et Ψ de la logique propositionnelle:
  - $\Phi \vee V \Leftrightarrow V$ (élément absorbant)
    $\Phi \wedge F \Leftrightarrow F$
  - $\Phi \vee F \Leftrightarrow \Phi$ (élément neutre)
    $\Phi \wedge V \Leftrightarrow \Phi$
  - $\neg\neg\Phi \Leftrightarrow \Phi$ (double négation)
  - $\Phi \vee \neg\Phi \Leftrightarrow V$ (tiers exclus)
    $\Phi \wedge \neg\Phi \Leftrightarrow F$
  - $\Phi \vee \Phi \Leftrightarrow \Phi$ (idempotence)
    $\Phi \wedge \Phi \Leftrightarrow \Phi$

Simplification de formules

- Pour toutes formules Φ, Ψ et Λ de la logique propositionnelle:
  - $\Phi \vee \Psi \Leftrightarrow \Psi \vee \Phi$ (commutativité)
    $\Phi \wedge \Psi \Leftrightarrow \Psi \wedge \Phi$
  - $\Phi \vee (\Psi \vee \Lambda) \Leftrightarrow (\Phi \vee \Psi) \vee \Lambda$ (associativité)
    $\Phi \wedge (\Psi \wedge \Lambda) \Leftrightarrow (\Phi \wedge \Psi) \wedge \Lambda$
  - $\Phi \vee (\Phi \wedge \Psi) \Leftrightarrow \Phi$ (absorption)
    $\Phi \wedge (\Phi \vee \Psi) \Leftrightarrow \Phi$
  - $\Phi \vee (\Psi \wedge \Lambda) \Leftrightarrow (\Phi \vee \Psi) \wedge (\Phi \vee \Lambda)$ (distributivité)
    $\Phi \wedge (\Psi \vee \Lambda) \Leftrightarrow (\Phi \wedge \Psi) \vee (\Phi \wedge \Lambda)$
  - $\neg(\Phi \vee \Psi) \Leftrightarrow \neg\Phi \wedge \neg\Psi$ (De Morgan)
    $\neg(\Phi \wedge \Psi) \Leftrightarrow \neg\Phi \vee \neg\Psi$
  - $\Phi \Rightarrow \Psi \Leftrightarrow \neg\Phi \vee \Psi$ (traduction de l’implication)
  - $(\Phi \Leftrightarrow \Psi) \Leftrightarrow (\Phi \Rightarrow \Psi) \wedge (\Psi \Rightarrow \Phi)$ (traduction de l’équivalence)
    $\Leftrightarrow (\Phi \wedge \Psi) \vee (\neg\Phi \wedge \neg\Psi)$

Simplification de formules

Å Toutes ses règles peuvent être démontrées en établissant les tables de vérité des formules équivalentes

<u>Ex:</u> Démontrons que $\neg(\Phi \vee \Psi) \Leftrightarrow \neg\Phi \wedge \neg\Psi$

Φ	Ψ	$\neg(\Phi \vee \Psi)$	$\neg\Phi \wedge \neg\Psi$
F	F	V	V
F	V	F	F
v	F	F	F
V	v	f	F

Å Exemple de simplification :

La formule $(\neg a \wedge b) \wedge \neg(a \Rightarrow b) \Rightarrow (a \vee b)$

se simplifie en V

Simplification de formules

- A lôaide des règles de simplification, on peut exprimer une formule en nôutilisant quôun sous-ensemble des connecteurs logiques :
  - $\{\vee,\neg\}$ (formulation disjonctive pure)
  - $\{\wedge,\neg\}$ (formulation conjonctive pure)
  - $\{\Rightarrow,\neg\}$ (formulation implicative pure)
  - $\{\vee,\wedge,\neg\}$ (formulation simple)

Ex: la proposition $a \wedge b \Rightarrow \neg c \vee d$ se reformule

- $\neg a \vee \neg b \vee \neg c \vee d$ en disjonctif pur
- $\neg(a \wedge b \wedge c \wedge \neg d)$ en conjonctif pur
- $\neg(a \Rightarrow \neg b) \Rightarrow (c \Rightarrow d)$ en implicatif pur
- $\neg(a \wedge b) \vee \neg c \vee d$ en simple

Formes normales

∀ Φ est en **forme normale conjonctive** (**FNC**) ssi cêest une conjonction de disjonctions dâatomes

$$\Phi = (a_1 \vee \text{é} \quad \vee a_n \vee \neg a_{n+1} \vee \text{é} \quad \vee \neg a_{n+m})$$
$$\wedge (b_1 \vee \text{é} \quad \vee b_p \vee \neg b_{p+1} \vee \text{é} \quad \vee \neg b_{p+q})$$
$$\wedge \text{ é}$$

∀ Φ est en **forme normale disjonctive** (**FND**) ssi cêest une disjonction de conjonctions dâatomes

$$\Phi = (a_1 \wedge \text{é} \quad \wedge a_n \wedge \neg a_{n+1} \wedge \text{é} \quad \wedge \neg a_{n+m})$$
$$\vee (b_1 \wedge \text{é} \quad \wedge b_p \wedge \neg b_{p+1} \wedge \text{é} \quad \wedge \neg b_{p+q})$$
$$\vee \text{ é}$$

Formes normales

Théorème: toute formule admet une formulation synonyme en FND et en FNC

Preuve : par simplification :

1. Traduction des équivalences et des implications
2. Distribution des négations (lois de De Morgan et double négation)
3. Distribution des conjonctions (resp. disjonctions)

=> On obtient une forme normale disjonctive (resp. conjonctive)

Attention : toutes les négations doivent être distribuées jusqu'aux atomes

Ex: la formule $(a \Rightarrow b) \Rightarrow (c \Rightarrow d)$ s'écrit

Å $(a \wedge \neg b) \vee (\neg c) \vee (d)$ en FND

Å $(a \vee d \vee \neg c) \wedge (d \vee \neg c \vee \neg b)$ en FNC

Formes normales (FND)

- Calcul d'une FND à partir des tables de vérité :
  - Soit $Var(\Phi)=\{a_1, a_2, \ldots, a_n\}$
  - À chaque interprétation

    $$I = \{a_{i1}=V, \ldots, a_{ik}=V, a_{ik+1}=F, \ldots, a_{in}=F\}$$

    telle que $I(\Phi)=V$, on associe une proposition

    $$\Phi_I = (a_{i1} \wedge \ldots \wedge a_{ik} \wedge \neg a_{ik+1} \wedge \ldots \wedge \neg a_{in})$$
  - Alors $\Psi = \bigvee_{I(\Phi)=V} \Phi_I$ est une FND de Φ :
    - Ψ est en FND par construction
    - $\Psi \equiv \Phi$ car $Var(\Psi)=Var(\Phi)$ et $\forall I,\ I(\Psi)=I(\Phi)$
- C'est la **FND canonique** : $\forall v \in Var(\Phi),\ v \in Var(\Phi_I)$

Formes normales (FND)

Å Exemple

Calculons la FND canonique de la formule $(a \Rightarrow b) \Rightarrow c$

a	b	c	$a \Rightarrow b$	$(a \Rightarrow b) \Rightarrow c$	
F	F	F	V	F	
F	F	V	V	V	$\rightarrow (\neg a \wedge \neg b \wedge c)$
F	V	F	V	F	
F	V	V	V	V	$\rightarrow (\neg a \wedge b \wedge c)$
V	F	F	F	V	$\rightarrow (a \wedge \neg b \wedge \neg c)$
V	F	V	F	V	$\rightarrow (a \wedge \neg b \wedge c)$
V	V	F	V	F	
V	V	V	V	V	$\rightarrow (a \wedge b \wedge c)$

Donc la FND canonique est :

$(\neg a \wedge \neg b \wedge c) \vee (\neg a \wedge b \wedge c) \vee (a \wedge \neg b \wedge \neg c) \vee (a \wedge \neg b \wedge c) \vee (a \wedge b \wedge c)$

Formes normales (FND)

Théorème : deux formules Φ et Ψ sont synonymes ssi elles ont même FND canonique (aux règles de commutativité près)

Preuve :

- même FND => synonymes par définition :
  - même FND => mêmes ensembles de variables
  - même FND => même valeur pour toute interprétation
- synonymes => même FND canonique par construction à partir des tables de vérité de Φ et Ψ :
  - $\Phi \equiv \Psi$, donc $Var(\Phi) = Var(\Psi)$ et $\forall I$, $I(\Phi) = I(\Psi)$
  - Donc Φ et Ψ ont même table de vérité (à permutation des lignes/colonnes près)
  - Donc on peut construire la même FND canonique pour Φ et Ψ

Formes normales (FND)

- Exemple : $(a \Rightarrow b) \Rightarrow c \equiv (\neg c \Rightarrow a) \wedge (b \Rightarrow c)$?
  - FND canonique de $(a \Rightarrow b) \Rightarrow c$:

    $(\neg a \wedge \neg b \wedge c) \vee (\neg a \wedge b \wedge c) \vee (a \wedge \neg b \wedge \neg c) \vee (a \wedge \neg b \wedge c) \vee (a \wedge b \wedge c)$
  - FND canonique de $(\neg c \Rightarrow a) \wedge (b \Rightarrow c)$:

a	b	c	$(\neg c \Rightarrow a) \wedge (b \Rightarrow c)$
F	F	F	F
F	F	V	V
F	V	F	F
F	V	V	v
V	F	F	v
V	F	V	V
V	V	F	F
V	V	V	V

→ $(\neg a \wedge \neg b \wedge c) \vee (\neg a \wedge b \wedge c) \vee (a \wedge \neg b \wedge \neg c) \vee (a \wedge \neg b \wedge c) \vee (a \wedge b \wedge c)$

Même FND canonique, donc formules synonymes

Formes normales (FND)

- Problème: comparer 2 formules par leurs FND canoniques nécessite de construire les tables de vérité des deux formules => on n'a rien gagné ?

- Solution: construire la FND canonique sans passer par la table de vérité :
  - Passer la formule en FND au moyen des règles de simplification
  - Passer la FND en FND canonique par **complétion**

Formes normales (FND)

Å Complétion dôune FND en FND canonique :

Å Soit $\Psi = \Psi_1 \vee \text{é} \vee \Psi_n$ une FND de la formule Φ

Å Pour chaque $\Psi_i = (a_{i1} \wedge \text{é} \wedge \neg a_{ij})$,
pour chaque $b \in \mathrm{Var}(\Phi) \setminus \mathrm{Var}(\Psi_i)$, on remplace Ψ_i par

$\Psi_{i1} = (a_{i1} \wedge \text{é} \wedge \neg a_{ij} \wedge b)$

$\Psi_{i2} = (a_{i1} \wedge \text{é} \wedge \neg a_{ij} \wedge \neg b)$

Å La synonymie avec Φ est préservée par les règles

Å de lôélément neutre : $\Psi_i \Leftrightarrow \Psi_i \wedge \mathrm{V}$

Å du tiers exclus : $\Psi_i \wedge \mathrm{V} \Leftrightarrow \Psi_i \wedge (b \vee \neg b)$

Å et de distributivité : $\Psi_i \wedge (b \vee \neg b) \Leftrightarrow (\Psi_i \wedge b) \vee (\Psi_i \wedge \neg b)$

Formes normales (FND)

- <u>Exemple :</u> FND canonique de $\Phi = (a \Rightarrow b) \Rightarrow c$?
  - FND : $(a \wedge \neg b) \vee c = \Psi_1 \vee \Psi_2$
  - Complétion :
    - $Var(\Phi) \setminus Var(\Psi_1) = \{c\}$

      $\rightarrow \Psi_1 \Leftrightarrow a \wedge \neg b \wedge V$

      $\Leftrightarrow a \wedge \neg b \wedge (c \vee \neg c)$

      $\Leftrightarrow \underline{\mathbf{(a \wedge \neg b \wedge c)}} \vee (a \wedge \neg b \wedge \neg c)$

    - $Var(\Phi) \setminus Var(\Psi_2) = \{a,b\}$

      $\rightarrow \Psi_2 \Leftrightarrow c \wedge V \wedge V$

      $\Leftrightarrow c \wedge (a \vee \neg a) \wedge (b \vee \neg b)$

      $\Leftrightarrow (a \wedge b \wedge c) \vee \underline{\mathbf{(a \wedge \neg b \wedge c)}} \vee (\neg a \wedge b \wedge c) \vee (\neg a \wedge \neg b \wedge c)$

  - Dôù la FND canonique de Φ :

$(a \wedge \neg b \wedge c) \vee (a \wedge \neg b \wedge \neg c) \vee (a \wedge b \wedge c) \vee (\neg a \wedge b \wedge c) \vee (\neg a \wedge \neg b \wedge c)$

Comparer deux formules (FND)

Pour comparer deux formules, on dispose de moyen sémantiques et syntaxiques :

- Sémantique : comparaison des tables de vérité
- Syntaxique : simplification ; comparaison des FND canoniques

Formes normales (FNC)

- Calcul d'une FNC à partir des tables de vérité :
  - Soit $Var(\Phi)=\{a_1, a_2, \text{é} , a_n\}$
  - À chaque interprétation

    $$I = \{a_{i1}=V, \text{é} , a_{ik}=V, a_{ik+1}=F , \text{é} , a_{in}=F\}$$

    telle que $I(\Phi)=F$, on associe une proposition

    $$\Phi_I = (\neg\, a_{i1} \vee \text{é} \vee \neg\, a_{ik} \vee a_{ik+1} \vee \text{é} \vee a_{in})$$
  - Alors $\Psi = V_{I(\Phi)=F}\ \Phi_I$ est une FNC de Φ :
    - Ψ est en FNC par construction
    - $\Psi \equiv \Phi$ car $Var(\Psi)=Var(\Phi)$ et $\forall I,\ I(\Psi)= I(\Phi)$
- C'est la **FNC canonique** : $\forall v \in Var(\Phi),\ v \in Var(\Phi_I)$

Formes normales (FNC)

Å Exemple

Calculons la FNC canonique de la formule $(a \Rightarrow b) \Rightarrow c$

a	b	c	$a \Rightarrow b$	$(a \Rightarrow b) \Rightarrow c$	
F	F	F	V	F	→ $(\neg a \wedge \neg b \wedge \neg c)$
F	F	V	V	V	
F	V	F	V	F	→ $(\neg a \wedge b \wedge \neg c)$
F	V	V	V	V	
V	F	F	F	V	
V	F	V	F	V	
V	V	F	V	F	→ $(a \wedge b \wedge \neg c)$
V	V	V	V	V	

FNC canonique est : $(a \vee b \vee c) \wedge (a \vee \neg b \vee c) \wedge (\neg a \vee \neg b \vee c) =$

$\neg((\neg a \wedge \neg b \wedge \neg c) \vee (\neg a \wedge b \wedge \neg c) \vee (\neg a \wedge \neg b \wedge \neg c)) =$

Formes normales (FNC)

Théorème : deux formules Φ et Ψ sont synonymes si et seulement si elles ont même FNC canonique (aux règles de commutativité près)

Preuve :

- même FNC => synonymes par définition :
  - même FNC => mêmes ensembles de variables
  - même FNC => même valeur pour toute interprétation
- synonymes => même FNC canonique par construction à partir des tables de vérité de Φ et Ψ :
  - $\Phi \equiv \Psi$, donc $Var(\Phi) = Var(\Psi)$ et $\forall I, I(\Phi) = I(\Psi)$
  - Donc Φ et Ψ ont même table de vérité (à permutation des lignes/colonnes près)
  - Donc on peut construire la même FNC canonique pour Φ et Ψ

Formes normales (FNC)

Å Exemple : $(a \Rightarrow b) \Rightarrow c \equiv (\neg c \Rightarrow a) \wedge (b \Rightarrow c)$?

Å FND canonique de $(a \Rightarrow b) \Rightarrow c$:

$(a \vee b \vee c) \wedge (a \vee \neg b \vee c) \wedge (\neg a \vee \neg b \vee c)$

Å FNC canonique de $(\neg c \Rightarrow a) \wedge (b \Rightarrow c)$:

a	b	c	$(\neg c \Rightarrow a) \wedge (b \Rightarrow c)$
F	F	F	F
F	F	V	V
F	V	F	F
F	V	V	v
V	F	F	v
V	F	V	V
V	V	F	F
V	V	V	V

→ $(a \vee b \vee c) \wedge (a \vee \neg b \vee c)$
$\wedge (\neg a \vee \neg b \wedge c)$

Même FNC canonique,
donc formules synonymes

Méthodes et systèmes formels de preuve

- Méthode de Quine : Elle est basée sur les transformations équivalentes utilisant les équivalences de la liste1 suivante ($V \equiv 1$ et $F \equiv 0$) :
  - $\neg 0 \equiv 1$, $\quad \neg 1 \equiv 0$,
  - $\Phi \wedge 0 \equiv 0$, $\quad \Phi \wedge 1 \equiv \Phi$,
  - $\Phi \vee 1 \equiv 1$, $\quad \Phi \vee 0 \equiv \Phi$,
  - $1 \Rightarrow \Phi \equiv \Phi$, $\quad 0 \Rightarrow \Phi \equiv 1$,
  - $\Phi \Leftrightarrow 1 \equiv \Phi$, $\quad \Phi \Leftrightarrow 0 \equiv \neg\Phi$,

  - Soient $I_{A=1}$ (X) = I(X) si X $\neq$ A sinon 1 et $I_{A=0}$ (X) = I(X) si X $\neq$ A sinon 0

Méthodes et systèmes formels de preuve

- <u>Méthode de Quine :</u> $\Phi \equiv>_{liste1} \Psi$: Ψ est le résultat des transformations droites de Φ par équivalences de la liste1 jusqu'à saturation.
- Quine(Φ)

$\Phi(A\backslash 1) \equiv>_{liste1} \Phi_1$

Quine(Φ_1)

$\Phi(A\backslash 0) \equiv>_{liste1} \Phi_2$

Quine(Φ_2)

Méthodes et systèmes formels de preuve

- Proposition 1: (1) La méthode converge sur toute formule Φ : toute branche la réduit à 1 ou à 0;

(2) $\models \Phi$ ($\mathrm{CONTR}(\Phi)$) si et seulement si toute branche réduit Φ à 1 (respectivement à 0).

Définition : Un modèle d'un ensemble Γ de formules : une interprétation I telle que Φ^I =1 pour tout $\Phi \in \Gamma$. Notation : $I \models \Gamma$.

Remarque : Les diagrammes de Quine décrivent explicitement les modèles des formules et souvent ils sont plus compacts que les tables de vérité.

Méthodes et systèmes formels de preuve

Å <u>Exemple :</u>

Prouver par Quine : $\models A\wedge \neg B \Rightarrow (\neg A \Rightarrow C\vee A)$

$$\Phi = A\wedge \neg B \Rightarrow (\neg A \Rightarrow C\vee A)$$

A = 1 A=0

$1\wedge \neg B \Rightarrow (\neg 1 \Rightarrow C\vee 1)$ $0\wedge \neg B \Rightarrow (\neg 0 \Rightarrow C\vee 0)$

$1\wedge \neg B \Rightarrow 1$ $0 \Rightarrow (\neg 0 \Rightarrow C\vee 0)$

1 1

Déduction par réfutation

- Principe de réfutation : Soit une relation binaire $\Rightarrow_\alpha$ entre les ensembles de formules. Supposons que $\Gamma_1 \Rightarrow_\alpha \Gamma_2$ implique $\Gamma_1 \models \Gamma_2$ pour tout Γ_1, Γ_2.Pour prouver que $\Gamma \models \Phi$, il suffit d'établir une suite d'ensembles de formules :

Γ U $\{\neg\Phi\} \models \Gamma_0 \Rightarrow_\alpha .... \Rightarrow_\alpha \Gamma_n$, où Γ_n est non-cohérent.

Alors Γ U $\{\neg\Phi\}$ l'est aussi et selon le principe de cohérence, $\Gamma \models \Phi$.

- Proposition 2 (Principe de cohérence) :

$\Gamma \models \Phi$ si et seulement si Γ U $\{\neg\Phi\}$ est non-cohérent;

Déduction par réfutation

Å 1) Méthode des tableaux :

$\wedge$-formules	Membres de < ϕ_1, ϕ_2>
$\phi_1 \wedge \phi_2$	ϕ_1, ϕ_2
$\phi_1 \Leftrightarrow \phi_2$	$\phi_1 \Rightarrow \phi_2$, $\phi_2 \Rightarrow \phi_1$
$\neg(\phi_1 \vee \phi_2)$	$\neg\phi_1, \neg\phi_2$
$\neg(\phi_1 \Rightarrow \phi_2)$	$\phi_1, \neg\phi_2$

Déduction par réfutation

$\vee$ -formules	Membres de < $\phi_1 \vee \phi_2$>
$\phi_1 \vee \phi_2$	ϕ_1, ϕ_2
$\phi_1 \Rightarrow \phi_2$	$\neg\phi_1, \phi_2$
$\neg(\phi_1 \wedge \phi_2)$	$\neg\phi_1, \neg\phi_2$
$\neg(\phi_1 \Leftrightarrow \phi_2)$	$\neg(\phi_1 \Rightarrow \phi_2)$, $\neg(\phi_2 \Rightarrow \phi_1)$

Déduction par réfutation

Å Tableau : Un ensemble de formules T = $\{B_1$,é , $B_\kappa\}$

Å Branche : Une liste de formules B=$\{\phi_1$,é , $\phi_l\}$. B est fermée : $\Psi, \neg\Psi \in B$ pour une Ψ.

Å **Méthode :** Pour prouver que $\Gamma \vdash_t \phi$

(1) Initialiser le premier tableau $T_0 = \{[\Gamma, \neg\phi]\}$.

(2) Etablir une dérivation $T_0 \Rightarrow_t \ldots\ldots \Rightarrow_t T_n$.

Un pas $T_{n-1} \Rightarrow_t T_n$ (n>=1) : côest soit

- une extension dôune branche $B \in T_{n-1}$ en utilisant une $\wedge$-formule $<\phi_1, \phi_2> \in B$.

- une ramification dôune branche $B \in T_{n-1}$ en utilisant $\vee$-formule $<\phi_1, \phi_2> \in B$.

(3) $\Gamma \vdash_t \phi$: dans cette dérivation, le dernier tableau T_n est fermé (toute sa branche est fermée).

Déduction par réfutation

Å <u>Exemple :</u> $\models (A \vee B) \Rightarrow ((A \Rightarrow P) \wedge (B \Rightarrow P) \Rightarrow P)$

1- $\neg((A \vee B) \Rightarrow ((A \Rightarrow P) \wedge (B \Rightarrow P) \Rightarrow P))$

2- $A \vee B$ [1: ext]

3- $\neg((A \Rightarrow P) \wedge (B \Rightarrow P) \Rightarrow P)$ [1: ext]

4- $(A \Rightarrow P) \wedge (B \Rightarrow P)$ [3: ext]

5- $\neg P$ [3: ext]

6- $A \Rightarrow P$ [4: ext]

7- $B \Rightarrow P$ [4: ext]

Déduction par réfutation

8- A [2: rm] 9- B [2: rm]

10-¬A [6:rm] 11-P [6:rm] 12-¬B [6:rm] 13- P[6:rm]

‘(8,10) ‘(5,11) ‘(9,12) ‘(5,13)

Exercice : Prouver que :

$\{(P\Rightarrow Q)\vee(R\Rightarrow S), P\wedge R\} \models (Q\vee S)$

Å <u>Théorème (Correction et complétude):</u>

$\bar{x}\bar{x}\bar{x}\bar{x}\Gamma \vdash_t \Phi$ si et seulement si $\Gamma \models \Phi$.

Déduction par réfutation

- Å <u>Méthode de résolution</u> (appliquée aux formes clausales) :
- Å Règle de résolution [A. Robinson] :

Pour deux forme clausales tout Γ_1, Γ_2, $\Gamma_1 \rightarrow_r \Gamma_2$

si $\Gamma_1 = \Gamma \cup \{Cl_1 \vee l, Cl_2 \vee \neg l\}$ et $\Gamma_2 = \Gamma_1 \cup \{Cl_1 \vee Cl_2\}$

($Cl_1 \vee Cl_2$:résolvante).

$\Gamma_1 \rightarrow_r \ldots\ldots \rightarrow_r \Gamma_n$, est une preuve si Γ_n est fermé.

- Å <u>Proposition 2(correction) :</u>

  Si $\Gamma_1 \rightarrow_r \Gamma_2$, alors $\Gamma_1 \models \Gamma_2$.

- Å <u>Corollaire :</u>

  Si $\Gamma_1 \rightarrow_r \ldots\ldots \rightarrow_r \Gamma_n$, est une preuve, alors Γ_1 est non-cohérent.

Déduction par réfutation

Å Méthode : Pour prouver $\Gamma \vdash_r \Phi$

(1) Former la Forme clausale initiale $\Gamma_0 = \Gamma \cup \{\neg\Phi\}$.

(2) Etablir une dérivation $\Gamma_0 \rightarrow_r \ldots\ldots \rightarrow_r \Gamma_n$.

(3) $\Gamma \vdash_r \Phi$, si cette dérivation est une preuve.

Exemple :

Prouver $\Gamma \vdash_r \Phi$ avec $\Gamma=\{(P \Rightarrow Q)\vee(R\Rightarrow S), P\wedge R\}$ et

$\Phi = \{ Q\vee S \}$

La forme clausale de $\Gamma \cup \{\neg\Phi\} = \{\neg P \vee Q\vee\neg R \vee S, P, R, \neg Q, \neg S\}$

Déduction par réfutation

Å Une preuve :

1- $\neg P \vee Q \vee \neg R \vee S$,

2- P,

3- R,

4- $\neg Q$,

5- $\neg S$

____

6– $Q \vee \neg R \vee S$ [1, 2]

7- $Q \vee S$ [3, 6]

8- S [4, 7]

9- ‘□□[5, 9]

<u>Théorème (Correction et complétude):</u>

$\bar{x}\bar{x}\bar{x}\bar{x}\Gamma \vdash_r \Phi$ si et seulement si $\Gamma \models \Phi$.

Déduction par réfutation

Å Exercice :

Pour $\Phi = (A \vee B) \Rightarrow ((A \Rightarrow P) \wedge (B \Rightarrow P) \Rightarrow P)$

Å Prouver que $\models \Phi$.

Conséquence logique

∀ Φ est une **conséquence logique** de Φ_1, é , Φ_n (noté **Φ_1, é , $\Phi_n \models \Phi$**) ssi $\Phi_1 \wedge$ é $\wedge\, \Phi_n \Rightarrow \Phi$ est une tautologie

- ∀ Φ_1, é , Φ_n sont les **prémisses**
- ∀ Φ est la **conséquence**

Å <u>Théorème:</u> Une formule Φ est une tautologie si et seulement si $\models \Phi$

<u>Preuve:</u> $\models \Phi$ signifie $\varnothing \models \Phi$ qui signifie $V \Rightarrow \Phi$ qui est vrai si et seulement si Φ est une tautologie

Conséquence logique

- Å <u>Théorème:</u> Des prémisses contradictoires ont pour conséquence nôimporte quelle formule

  <u>Preuve:</u> Φ_1, é , Φ_n contradictoires → $\Phi_1 \wedge$ é $\wedge\, \Phi_n \Leftrightarrow \mathrm{F}$

  donc Φ_1, é , $\Phi_n \models \Phi$ ssi $\mathrm{F} \Rightarrow \Phi$, c-a-d. quel que soit Φ

- Å <u>Théorème:</u> Si Φ_1, é , $\Phi_n \models \Phi$ alors Φ_1, é , $\Phi_{n-1} \models \Phi_n \Rightarrow \Phi$

  <u>Preuve:</u> Φ_1, é , $\Phi_n \models \Phi$ ssi $\Phi_1 \wedge$ é $\wedge\, \Phi_n \Rightarrow \Phi$

  $\Leftrightarrow \neg(\Phi_1 \wedge$ é $\wedge\, \Phi_n) \vee \Phi$

  $\Leftrightarrow \neg(\Phi_1 \wedge$ é $\wedge\, \Phi_{n-1}) \vee \neg\Phi_n \vee \Phi$

  $\Leftrightarrow \neg(\Phi_1 \wedge$ é $\wedge\, \Phi_{n-1}) \vee (\Phi_n \Rightarrow \Phi)$

  $\Leftrightarrow \Phi_1 \wedge$ é $\wedge\, \Phi_{n-1} \Rightarrow (\Phi_n \Rightarrow \Phi)$ ssi Φ_1, é , $\Phi_{n-1} \models \Phi_n \Rightarrow \Phi$

Raisonnement → Conséquence

Å Raisonnement = un ensemble dôhypothèses dont dérive une conclusion = conséquence logique

Ex: « Sôil pleut je prends mon parapluie; Il ne pleut pas si le chien du voisin est dehors; Lorsquôil est dehors, ce chien aboie. Donc, lorsque le chien du voisin aboie, je ne prends pas mon parapluie.»

est valide ssi

$$p \Rightarrow u, \quad d \Rightarrow \neg p, \quad d \Rightarrow a \quad \models \quad a \Rightarrow \neg u$$

Déduction logique

- **Règle de déduction** $\mathbf{r}$ = fonction qui à Φ_1, é , Φ_n associe Φ. On note $\mathbf{\Phi_1}$**, é ,** $\mathbf{\Phi_n \vdash_r \Phi}$

- **Système déductif** (SD) = système formel ($\mathcal{A}$,$\mathcal{R}$) où
  - $\mathcal{A}$ = ensemble de formules admises, les **axiomes**
  - $\mathcal{R}$ = ensemble de règles de déduction

- Φ est **déductible** de Φ_1, é , Φ_n par un système déductif $\mathcal{S}$ (noté $\mathbf{\Phi_1}$**, é ,** $\mathbf{\Phi_n \vdash_S \Phi}$) si et seulement si $\exists$ Ψ_1, é , Ψ_m telles que $\Psi_m = \Phi$ et $\forall i \in [1,m]$:
  - Soit $\Psi_i \in \mathcal{A} \cup \{\Phi_1$, é , $\Phi_n\}$
  - Soit $\exists j1$, é , $jk \in [1,i]$ et $\exists r \in \mathcal{R}$ tel que Ψ_{j1}, é $\Psi_{jk} \vdash_r \Psi_i$

Déduction logique

- Déduction Í Conséquence
  - Syntaxique <u>**VS**</u> Sémantique
  - Dépendant dôun SD <u>**VS**</u> indépendant
- <u>Théorème:</u> En logique propositionnelle, il existe des SD dont les déductions correspondent aux conséquences logiques

  i.e., $\exists \mathcal{S}$ tq Φ_1, é , $\Phi_n \vdash_{\mathcal{S}} \Phi$ ssi Φ_1, é , $\Phi_n \models \Phi$
- La déduction naturelle (N) est un SD sans axiome, avec 9 règles : une pour lôajout et le retrait des connecteurs $\neg$, $\vee$, $\wedge$ et $\Rightarrow$, et une pour lôintroduction dôhypothèses de raisonnement
- <u>Théorème:</u> Φ_1, é , $\Phi_n \vdash_{\mathrm{DN}} \Phi$ ssi Φ_1, é , $\Phi_n \models \Phi$

Déduction naturelle

- Règles de la DN :
  - Hypothèse (axiome) : $\Phi, \Gamma \vdash_N \Phi$ règle *ax*
  - Conjonction :
    - Introduction :

      $\Gamma \vdash_N A, \quad \Delta \vdash_N B$ alors $\Gamma\,\Delta \vdash_N A \wedge B$ règle $\wedge^+$
    - Retrait :

      $\Gamma \vdash_N (A \wedge B)$ alors $\Gamma \vdash_N A$ règles $\wedge^-$

      $\Gamma \vdash_N (A \wedge B)$ alors $\Gamma \vdash_N B$
  - DISJONCTION :
    - Introduction : $\Gamma \vdash_N A$ alors $\Gamma \vdash_N A \vee B$ règles $\vee^+$

      $\Gamma \vdash_N B$ alors $\Gamma \vdash_N A \vee B$
    - Retrait : $\Gamma \vdash_N A \vee B, \quad [A]\,\Delta \vdash_N C, \quad [B]\,\Sigma \vdash_N C$

      alors $\Gamma\,\Delta\,\Sigma \vdash_N C$ règle $\vee^-$

Déduction naturelle

- Implication :
  - Introduction :

    $[A]\ \Gamma \vdash_N B$ alors $\Gamma \vdash_N A \Rightarrow B$ règle $\Rightarrow^+$

  - Retrait :

    $\bar{x}\Gamma \vdash_N A,\quad \Delta \vdash_N (A \Rightarrow B)$ alors $\Gamma\ \Delta \vdash_N B$ règle $\Rightarrow^-$

- Négation :
  - Introduction :

    $[A]\ \Gamma \vdash_N B,\ [A]\ \Delta \vdash_N \neg B$ alors $\Gamma\ \Delta \vdash_N \neg A$ règle $\neg^+$

  - Retrait :

    $\Gamma \vdash_N \neg\neg A$ alors $\Gamma \vdash_N A,$ règle $\neg^-$

Déduction naturelle

Å Chaque règle de DN peut être prouvée

Ex: $[\Psi]\ \Phi_1, \Phi_2 \vdash_{DN} [\Psi]\ \Phi_1 \wedge \Phi_2$

ssi $\Phi_1, \Phi_2 \models \Phi_1 \wedge \Phi_2$

ssi $\Phi_1 \wedge \Phi_2 \Rightarrow \Phi_1 \wedge \Phi_2$ est une tautologie

Φ_1	Φ_2	$\Phi_1 \wedge \Phi_2$	$\Phi_1 \wedge \Phi_2 \Rightarrow \Phi_1 \wedge \Phi_2$
F	F	F	V
F	V	F	V
V	F	F	V
V	V	v	v

Déduction naturelle

- Par déduction naturelle on peut montrer la validité ou la contradiction dõun raisonnement :
  - Validité : Φ_1, é , $\Phi_n \vdash_{DN} \Phi$
  - Contradiction : Φ_1, é , $\Phi_n \vdash_{DN} \neg\Phi$

<u>Attention :</u> la déduction naturelle ne peut prouver ni la validité des raisonnements non tautologiques, ni la contradiction des raisonnements non antilogiques

=> Il faut avoir une intuition de la validité avant de la prouver

Déduction naturelle

Exemple :

Å Prouvons la validité de la conséquence logique:

$$a \Rightarrow (b \Rightarrow c) \models a \wedge b \Rightarrow c$$

Å Preuve directe par DN :

1.	$[\varnothing]$	$a \Rightarrow (b \Rightarrow c)$	(prémisse)
2.	$[a \wedge b]$	$a \wedge b$	(hypothèse)
3.	$[a \wedge b]$	a	$(\wedge^{-}\ 2)$
4.	$[a \wedge b]$	b	$(\wedge^{-}\ 2)$
5.	$[a \wedge b]$	$b \Rightarrow c$	$(\Rightarrow^{-}\ 1,3)$
6.	$[a \wedge b]$	c	$(\Rightarrow^{-}\ 4,5)$
7.	$[\varnothing]$	$a \wedge b \Rightarrow c$	($\Rightarrow^{+}$ 2,6 ï fin hypothèse 2)

Déduction naturelle

Å $a \Rightarrow (b \Rightarrow c)$ $[a \wedge b]^1$ $[a \wedge b]^1$

axiome ax

$a \wedge b$ $a \wedge b$

$\wedge^-$ $\wedge^-$

a b

$\Rightarrow^-$

$b \Rightarrow c$ $\Rightarrow^-$

c

1 $\Rightarrow^+$

$a \wedge b \Rightarrow c$

Déduction naturelle

Exemple :

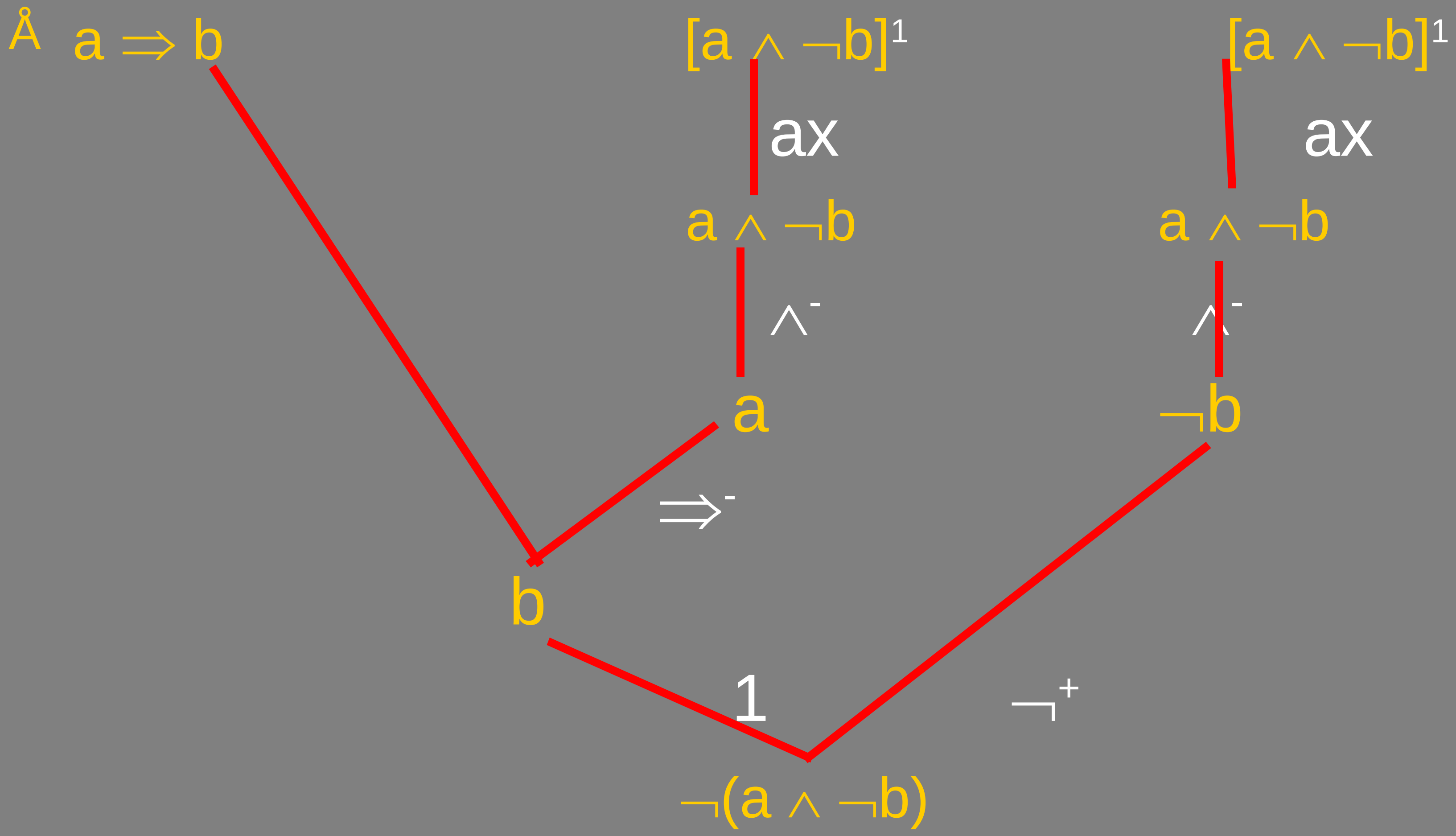
Å Prouvons lînvalidité de la conséquence logique:

$$a \Rightarrow b \not\models a \wedge \neg b$$

Å Preuve de contradiction par DN :

1.	$[\varnothing]$	$a \Rightarrow b$	(prémisse)
2.	$[a \wedge \neg b]$	$a \wedge \neg b$	(hypothèse)
3.	$[a \wedge \neg b]$	a	$(\wedge^{-}\ 2)$
4.	$[a \wedge \neg b]$	b	$(\Rightarrow^{-}\ 1{,}3)$
5.	$[\varnothing]$	$a \wedge \neg b \Rightarrow b$	$(\Rightarrow^{+}$ 2,4 ï fin hypothèse 2)
6.	$[a \wedge \neg b]$	$a \wedge \neg b$	(hypothèse)
7.	$[a \wedge \neg b]$	$\neg b$	$(\wedge^{-}\ 6)$
8.	$[\varnothing]$	$a \wedge \neg b \Rightarrow \neg b$	$(\Rightarrow^{+}$ 6,7 ï fin hypothèse 6)
9.	$[\varnothing]$	$\neg (a \wedge \neg b)$	$(\neg^{+}\ 5{,}8)$

Déduction naturelle



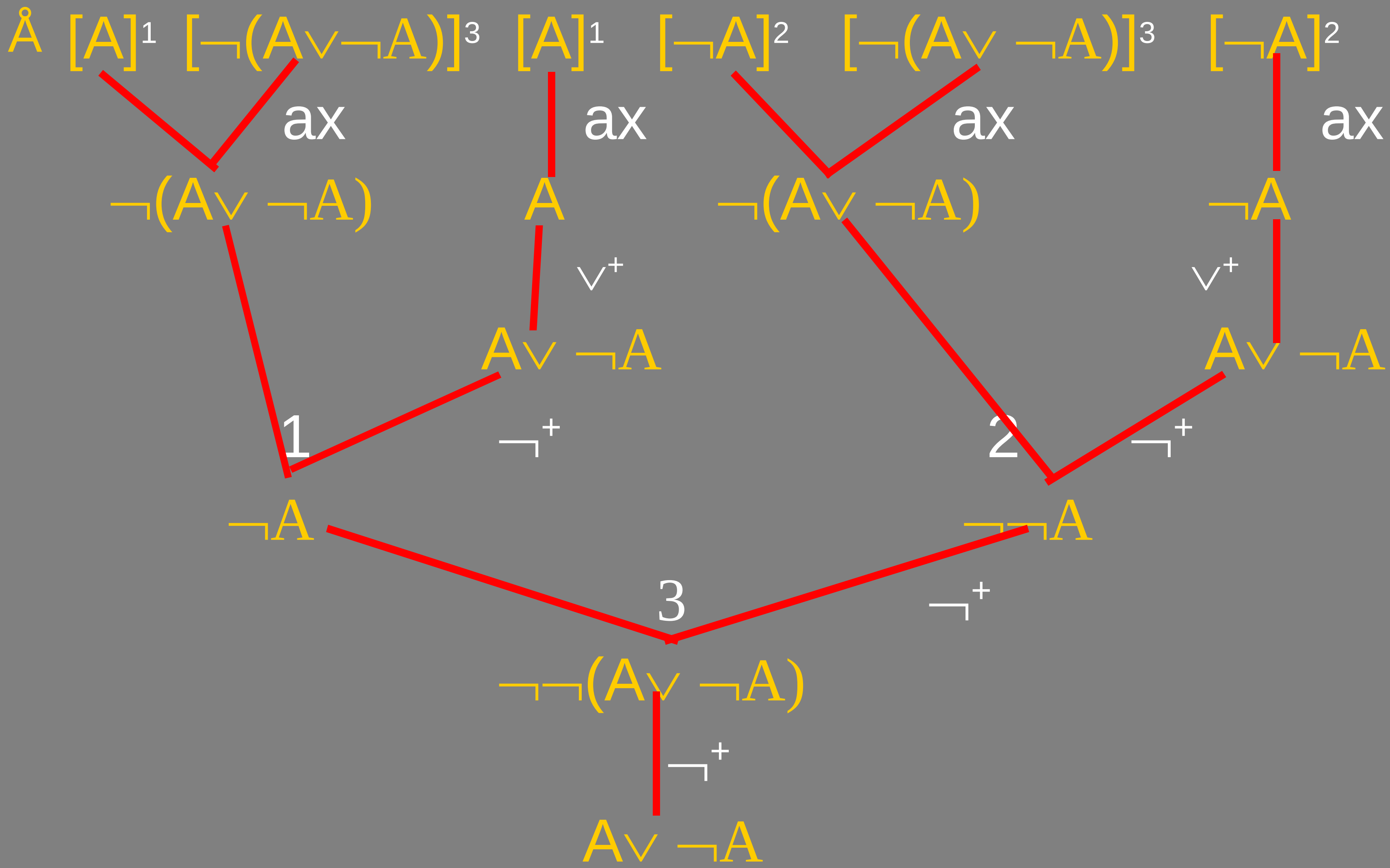
Déduction naturelle

- On peut ajouter des règles dérivées :

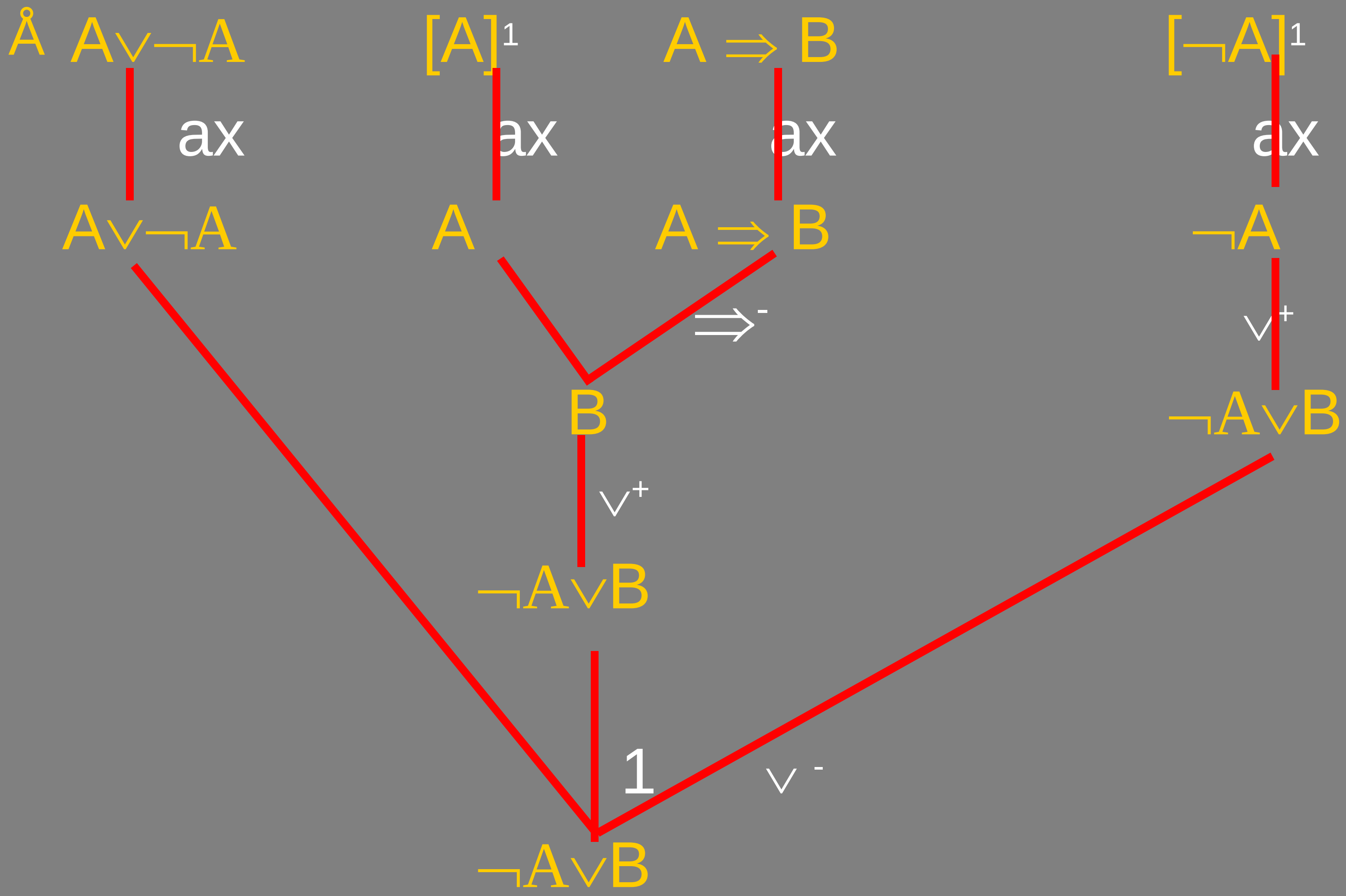
  - Φ_1, é , $\Phi_n \vdash_r \Phi$ est dérivée de DN ssi Φ_1, é , $\Phi_n \vdash_{DN} \Phi$
  - i.e., une règle dérivée est lâabréviation dâune démonstration annexe

- Quelques règles dérivées très utiles :
  - $\vdash_N A \vee \neg A$ *tiers exclus*
  - $\Gamma \vdash_N A$, $\Delta \vdash_N \neg A$ alors $\Gamma\ \Delta \vdash_N \neg B$ *contradiction*
  - $\Gamma \vdash_N A \Rightarrow B$ alors $\Gamma \vdash_N \neg A \vee B$
  - $\Gamma \vdash_N \neg A \vee B$ alors $\Gamma \vdash_N A \Rightarrow B$

Déduction naturelle



Déduction naturelle

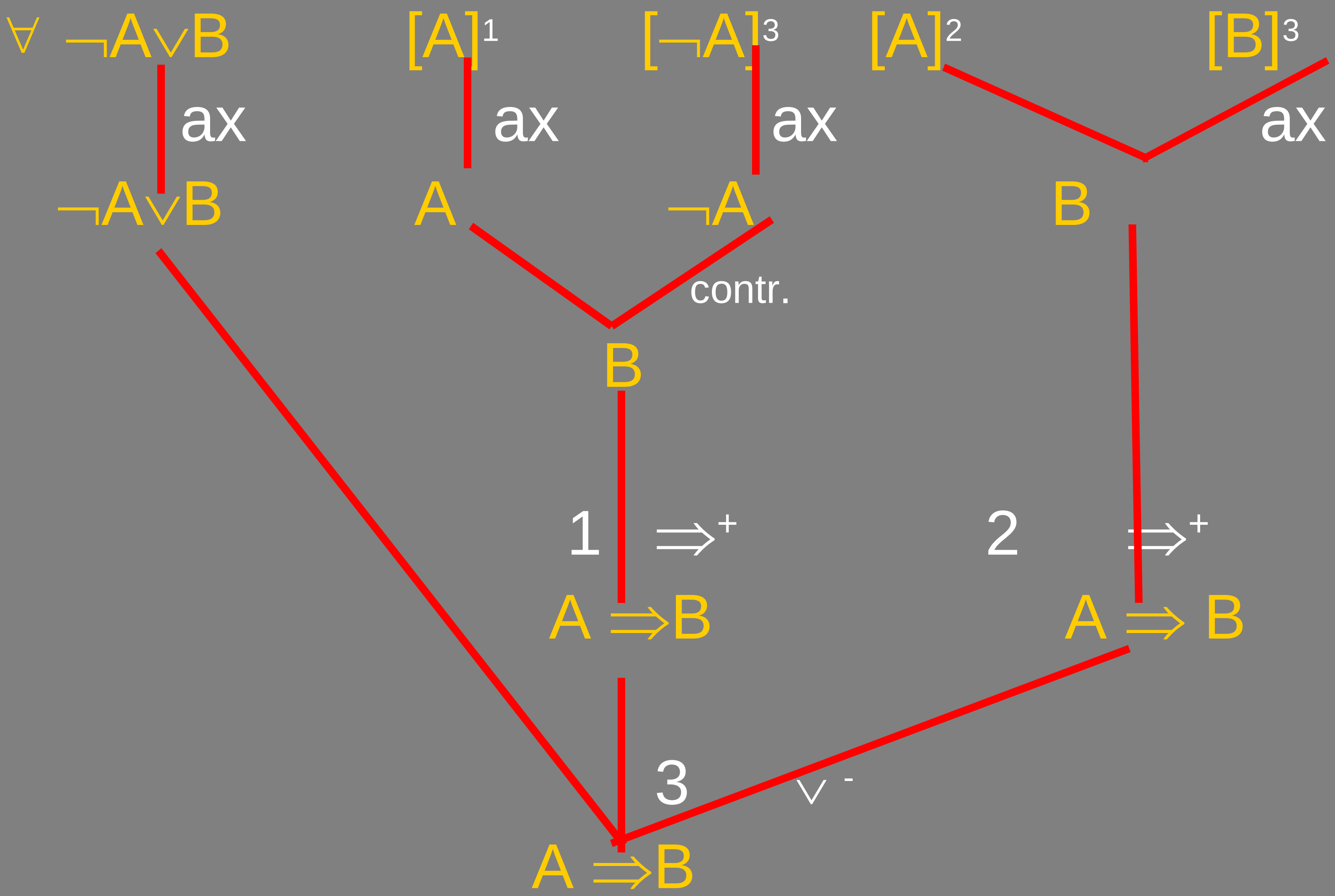


Déduction naturelle

Å

$$\dfrac{\dfrac{\dfrac{A \qquad [\neg B]^1}{A}\,ax \qquad\qquad \dfrac{\neg A \qquad [\neg A]^1}{\neg A}\,ax}{\neg\neg B}\;1\;\neg^+}{B}\,\neg^-$$

Déduction naturelle



En résumé

- Logique = outil formel pour analyser le raisonnement
- Logique = formules (syntaxe)
  + valeurs de vérité (sémantique)
- Les manipulations syntaxiques correspondent aux opérations sémantiques :
  - La simplification préserve la valeur de vérité
  - La comparaison (syntaxique) des FND canonique indique la synonymie (sémantique)
  - La déduction naturelle (syntaxique) correspond à la conséquence logique (sémantique)
- Programme = description dûune méthode pour obtenir un résultat ~ raisonnement

=> Logique pour la preuve de programme ?

Contenu

- Logique propositionnelle
  - Syntaxe, sémantique
  - Valuation, validité, satisfaisabilité
  - Tables de vérité
  - Simplification, comparaison, formes normales
  - Conséquence VS Déduction
- Logique du premier ordre
  - Syntaxe et sémantique de la logique du premier ordre
  - Principes généraux du premier ordre
  - Systèmes formels du premier ordre

Logique du premier ordre

La syntaxe et la sémantique :

1) Insuffisance de la logique propositionnelle :

Pour le montrer, regardons quelques exemples :

Exemple 1 : Syllogisme dô Aristote :

Prémisse : Toutes les baleines habitent la mer.

Prémisse : Toute baleine est un mammifère.

Conclusion : Il existe des mammifères qui habitent la mer.

Dans cet exemple, le problème est que la conclusion de ce syllogisme nô est pas impliquée par les prémisses 1 et 2. Il faut que la sémantique soit suffisamment précise pour le voir.

Logique du premier ordre

<u>Exemple 2 :</u>

Prémisse : A qui est ce portrait ?.

Prémisse : Cêest le fils de mon père et je nêai ni frère ni sî ur.

Conclusion : Cêest mon portrait.

Dans cet exemple, il nous manque plusieurs choses :

- Il sêavère que les valeurs booléennes des propositions telle que être frère dépendent des paramètres portant sur les domaines des objets non booléens (des personnes). Il faut savoir exprimer ces propositions paramétrées
- Il faut exprimer les opérateurs quelque soit x et il existe x. Le premier aurait le sens que la proposition paramétrée à laquelle il porte est vraie indépendamment des valeurs des paramètres. Le second aurait le sens que cette proposition est vraie au moins pour une valeur des paramètres.

Logique du premier ordre

Exemple 3 : Soit la phrase suivante : « Paul lui a posé un lapin ».

En logique propositionnelle, on ne peut que poser la questions vrai ou faux? Et sôil faut demander à qui? Ce nôest pas possible.

Ces exemples montrent que pour les exprimer il faut analyser, cette fois, les clauses elles-mêmes.

A ce niveau de lôanalyse, la logique convenable est la logique du premier ordre.

Logique du premier ordre

Å Dans toutes les langues, on trouve :

1- les verbes dont les arguments ont les groupes nominaux. Cette structure argumentale est représentée en logique par les prédicats :

Sortir(x, y) : « x sort avec y pour se distraire »

Poser-lapin(x, y) : « x ne vient pas au rendez-vous accordé par y ».

2- Les moyens dêexpression de la référence dans les groupes nominaux : les articles, les prénoms,é .

Le langage du premier ordre (syntaxe)

Å **Symboles :**

Symboles de la signature :

Constantes : 0,1, Adam, Eve, é

Foncteurs : +, *.

Prédicats: aimer, <, =. Notation p/n (prédicat p à n arguments ou dâarité n): aimer/2, </2, =/2

Connecteurs : $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$, $\neg$, $\exists$ (quantificateur existentiel), $\forall$ (quantificateur universel).

Variables : X, Y, é .

Å **Termes :**

Termes primitifs : les variables et les constantes.

Termes composés : si t_1, é t_k sont des termes alors $f(t_1$, é $t_k)$ est aussi un terme pour tout foncteur f/k.

Exemple : Adam, X, X+3.

Le langage du premier ordre (syntaxe)

Å **Formules :**

Formules atomiques, atomes : p(α_1, é , α_n), où p/n et α_i sont des termes.

Exemple : connaître(adam, ève), (3-x)/y>2, rouge(X), porche(Y).

Formules composées avec les connecteurs propositionnels : $\neg\Phi$, $(\Phi_1 \wedge \Phi_2)$, $(\Phi_1 \vee \Phi_2)$, $(\Phi_1 \Rightarrow \Phi_2)$, $(\Phi_1 \Leftrightarrow \Phi_2)$ sont des formules de premier ordre pour toute formule Φ,Φ_1, Φ_2. .

Formules quantifiées : si Φ est une formule et X une variable alors $\exists$X Φ, $\forall$X Φ sont des formules de premier ordre (X est liée par le quantificateur et Φ est sa portée). Une variable. Une variable Y non liée est libre) Une formule sans variable est dite proposition.

Exemples :

Φ = (x>0 $\wedge$ $\exists$ z(x= y+z)); x, y sont libres et (x= y+z) est la portée de $\exists$ z dans Φ.

Φ =$\forall$x (x= x) est une proposition.

Le langage du premier ordre (syntaxe)

- Définition : Une variable x est libre pour un terme t dans une formule Φ si aucune occurrence libre de x dans Φ ne se trouve dans la portée d'un quantificateur (Q y), où y est une des variables de t.

- Exemple :

La variable z dans $\exists$ z(x= y+z) $\wedge$ (z>x+y) n'est pas libre dans z(x= y+z) mais libre (z>x+y).

Le langage du premier ordre (sémantique)

- Pour interpréter les formules de premier ordre, il faut choisir un contexte c'est-à-dire :

- Fixer une structure I avec un domaine $D \neq \emptyset$ sur lequel porte les variables, et où à tout foncteur f/k (une constante) correspond une fonction à k arguments dur D (un objet dans D) et à tout prédicat p/n correspond une relation à n places sur D.
- Choisir une affectation σ des valeurs des variables libres.

<u>Exemple :</u> Soit l'énoncé $\exists x\ (x^2 - y = 0)$ qui affirme l'existence d'une solution de l'équation $x^2 - y = 0$.

Cette formule peut être interprétée de plusieurs façons :

Interprétation I_e : D : les entiers, alors pour y = 2, il n'y a pas de solution. Pour y=0, il en existe.

Interprétation I_r: D : les réels, alors pour y = 2, il y a deux solutions.

Le langage du premier ordre (sémantique)

Å <u>Vérité des atomes dans un contexte :</u>

$I,\sigma \models p(\alpha_1$, é , $\alpha_n)$, $(p(\alpha_1$, é , $\alpha_n)$ est vrai dans le contexte $I,\sigma)$:

$I,\sigma \models p(\alpha_1$, é , $\alpha_n)$ si $p(a_1$, é , $a_n)^I = 1$,

Où a_i sont les valeurs des termes α_i dans lôinterprétation I quand les variables obtiennent les valeurs à travers lôaffectation σ.

Å <u>Vérité dans un contexte de formules complexes :</u>

- $\Phi = \neg\Psi$:$\bar{x}$ $I,\sigma \models \Phi$ ssi $I, \sigma \models \neg\Psi$
- $\Phi = \Psi_1 \wedge \Psi_2$:$\bar{x}$ $I, \sigma \models \Phi$ ssi $I, \sigma \models \Psi_1$ et $I, \sigma \models \Psi_2$
- $\Phi = \exists X\ \Psi$: $I, \sigma \models \Phi$ ssi $I, \sigma \models \Psi$ pour un objet $a \in D$
- $\Phi = \forall X\ \Psi$: $I, \sigma \models \Phi$ ssi $I, (\sigma, X=a) \models \Psi$ pour tout objet $a \in D$.

Le langage du premier ordre (sémantique)

Å <u>Exemple :</u> Interprétation arithmétique I_e :

(il ya un entier Y pour tout entier X)

$I_e \models \neg\exists X\ \forall Y\ (Y<X \vee X=Y)$ (il nôy a pas dôentier maximal).

Å <u>Classement des formules :</u>

Définition :

Φ satisfaisable : $(\Phi)^{I}_{\sigma} = 1$ dans un contexte I, σ.

Φ est une contradiction : Φ nôest pas satisfaisable.

Φ est une tautologie $(\Phi)^{I}_{\sigma} = 1$ dans tout contexte I, σ.

Le langage du premier ordre (sémantique)

Å Exemple :

- La formule $\exists x$ (x^2 ï y =0) est satisfaisable; elle nœst pas une tautologie.
- La formule $\neg\forall x\ Q \Leftrightarrow \exists x\ \neg Q$ est une tautologie.
- La formule $p(x) \wedge \neg p(x)$ est une contradiction.

Å <u>Principe de quantification :</u>

- $\forall x\ \forall y\ Q(x, y, Z)$ est équivalente à $\forall y\ \forall x\ Q(x, y, Z)$

– $\exists X\ \exists Y\ Q(x, y, Z)$ est équivalente à $\exists y\ \exists x\ Q(x, y, Z)$

(Z étant une liste de variables libres.)

Le langage du premier ordre (sémantique)

- Principe de quantificateurs duaux :
- Théorème :
  - $\neg\forall x\ Q(x, Z)$ est équivalente à $\exists x\ \neg Q(x, Z)$
  - $\neg\exists x\ Q(x, Z)$ est équivalente à $\forall x\ \neg Q(x, Z)$

- Principe de la portée :
  - Théorème : Pour toute formule $Q(x, Z)$ et toute formule $P(Y)$ qui nâa pas dâoccurrence de x libre, on a :
  - $P(Y) \wedge \forall x\ Q(x, Z)$ est équivalente à $\forall x\ (P(Y) \wedge Q(x, Z))$
  - $P(Y) \vee \forall x\ Q(x, Z)$ est équivalente à $\forall x\ (P(Y) \vee Q(x, Z))$
  - $P(Y) \wedge \exists x\ Q(x, Z)$ est équivalente à $\exists x\ (P(Y) \wedge Q(x, Z))$
  - $P(Y) \vee \exists\ x\ Q(x, Z)$ est équivalente à $\exists\ x\ (P(Y) \vee Q(x, Z))$

Pour toute formule P, Q on a :

- $\forall x\ (P \wedge Q)$ est équivalente à $\forall y\ P \wedge \forall z\ Q$
- $\exists x\ (P \vee Q)$ est équivalente à $\exists y\ P \vee \exists\ z\ Q$

Le langage du premier ordre (sémantique)

Å <u>Exemple :</u>

$\exists x \neg(q(x) \Rightarrow (\forall z\, p(z) \wedge \neg q(x)))$: remplacer $\neg(P \Rightarrow Q)$ par $P \wedge \neg Q$

$\exists x\, (q(x) \wedge \neg(\forall z\, p(z) \wedge \neg q(x)))$: remplacer $\neg(P \wedge Q)$ par $\neg P \vee \neg Q$

$\exists x\, (q(x) \wedge (\neg\forall z\, p(z) \vee \neg\neg q(x)))$: remplacer $\neg\neg P$ par P

$\exists x\, (q(x) \wedge (\neg\forall z\, p(z) \vee q(x)))$: remplacer $\neg\forall z\, p(z)$ par $\exists z \neg p(z)$

$\exists x\, (q(x) \wedge (\exists z \neg p(z) \vee q(x)))$: par la portée

$\exists x\, \exists z\, (q(x) \wedge (\neg p(z) \vee q(x)))$.

Forme normale prénexe, Forme Skolem

Å <u>Définition :</u> Une formule est prénexe si elle est de la forme Q_1x_1, Q_2x_2, é , Q_nx_n G, où les Q_i sont des quantificateurs ∀, ∃ et G est une formule sans quantificateur.

Å <u>Définition :</u> Forme de Skolem

Une formule du premier ordre est sous forme de Skolem si et seulement si :

1 ï elle est en forme prénexe;

2 ï les quantificateurs ∃ qui venaient en tête, dans le préfixe, ont été remplacés par des constantes a_1, a_2, é (constantes de Skolem);

3 ï les quantificateurs ∃ qui étaient précédés, dans le préfixe, par des quantificateurs ∀, ont été supprimés et les variables quÕils liaient ont été remplacées par des symboles de fonctions qui ne figurent pas déjà dans la formule (foncteurs de Skolem)

Forme normale prénexe, Forme Skolem

Å Exemple :

La formule est sous la forme prénexe :

$\exists x \, \exists y \, \forall z \, \forall u \, \exists v \, ((R(x, y) \Rightarrow (S(z, u, v) \vee K(w, a))) \wedge G(v))$

. Pour la mettre sous la forme de Skolem, il faut supprimer les quantificateurs existentiels.

On pose $x = a_1$, $y=a_2$, On obtient :

$\forall z \, \forall u \, \exists v \, ((R(a_1, a_2) \Rightarrow (S(z, u, v) \vee K(w, a))) \wedge G(v))$

On pose $v = f(z, u)$ foncteur de Skolem, on obtient :

$\forall z \, \forall u \, ((R(a_1, a_2) \Rightarrow (S(z, u, f(z, u)) \vee K(w, a))) \wedge G(f(z, u)))$

Forme normale de Skolem

Å Définition :

Une formule du premier ordre est en forme normale de Skolem si et seulement si :

1 ï elle est en forme de Skolem;

2 ï sa matrice G est en forme normale conjonctive (FNC).

Exemple :

$\forall z\ \forall u\ ((R(a_1, a_2) \Rightarrow (S(z, u, f(z, u)) \vee K(w, a))) \wedge G(f(z, u))),$

on remplace $(P \Rightarrow Q)$ par $\neg P \vee Q$, on obtient :

$\forall z\ \forall u\ ((\neg R(a_1, a_2) \vee S(z, u, f(z,u)) \vee K(w, a)) \wedge G(f(z,u))),$

Forme normale de Skolem

Exemple :

$\neg(\forall x\ \forall y\ (\ R(x, y) \Leftrightarrow S(x)) \Rightarrow \forall y\ (R(x, y) \Rightarrow S(x)\ ))$

1e Phase : Renommage des variables liée :

$\neg(\forall u\ \forall v\ (\ R(u, v) \Leftrightarrow S(u)\ ) \Rightarrow \forall y\ (R(x, y) \Rightarrow S(x))\ )$

2e Phase : Mise en forme prénexe :

on remplace $(P \Rightarrow Q)$ par $\neg\ P \vee Q$, on obtient :

$\forall u\ \forall v\ (\ R(u, v) \Leftrightarrow S(u)) \wedge \neg\ \forall y\ (R(x, y) \Rightarrow S(x))$

remplacer $\neg\forall z\ p(z)$ par $\exists z\ \neg p(z)$, on obtient :

$\forall u\ \forall v\ (R(u, v) \Leftrightarrow S(u)) \wedge \exists y\ \neg(R(x, y) \Rightarrow S(x))$

on met les quantificateurs au début, on obtient :

$\forall u\ \forall v\ \exists y\ ((\ R(u, v) \Leftrightarrow S(u)\ ) \wedge \neg(R(x, y) \Rightarrow S(x)))$

Exemple (suite)

3e Phase : Mise en forme de Skolem :

on pose y = f(u, v), x= a on obtient :

$\forall u\ \forall u\ ((\ R(u, v) \Leftrightarrow S(u)\ ) \wedge \neg(R(x, f(u, v)) \Rightarrow S(a)))$

4e Phase : Mise en forme normale de Skolem :

$\forall u \forall u\ ((R(u, v) \Rightarrow S(u)) \wedge (S(u) \Rightarrow R(u, v))) \wedge \neg(R(a, f(u, v)) \Rightarrow S(a)))$

on remplace $(P \Rightarrow Q)$ par $\neg P \vee Q$, on obtient :

$\forall u \forall u\ ((\neg R(u, v) \vee S(u)) \wedge (\neg S(u) \vee R(u,v)) \wedge \neg(\neg R(a,f(u,v)) \vee S(a)))$

$\forall u \forall u\ ((\neg R(u, v) \vee S(u)) \wedge (\neg S(u) \vee R(u, v)) \wedge R(a, f(u, v)) \wedge \neg S(a))$

5e Phase : Mise en forme normale de Skolem clausale :

$\{\neg R(u, v) \vee S(u),\ \neg S(u) \vee R(u, v),\ R(a, f(u, v)),\ \neg S(a)\}$

Exercice : Mettre le formule suivante en forme clausale :

$\Phi = \forall y\ \forall z\ (R(z) \Rightarrow \exists x\ (P(x, y, z) \wedge Q(y)))$

Systèmes formels de premier ordre

Å <u>Modèles dôHerbrand :</u>

Ces modèles représentent les calculs symboliques de fonctions et de relations. On peut supposer sans perte de généralité que toute signature fonctionnelle Σ_f a un ensemble non vide de constantes $\Sigma_c \subseteq \Sigma_f$ (sinon on prend $\Sigma_c =_{df} \{c/0\}$, c étant une nouvelle constante). On fixe pour Σ_f le domaine dôHerbrand $H(\Sigma_f) =_{df} T(\Sigma_f)$ (lôensemble des termes fermés (clos)). Ces termes représentent les valeurs symboliques.

Å <u>Exemple :</u> Soit la signature $\Sigma_f = \{a/0, f/1\}$. Alors $H(\Sigma_f) =_{df} \{a, f(a), f(f(a)),$ é $\}$.

Soit la formule $\exists x\ (P(x) \Rightarrow Q(x, f(x)))$. Alors $H(\Sigma_f) =_{df} \{c, f(c), f(f(c)),$ é $\}$.

Systèmes formels de premier ordre

- Définition 1 : Soit une signature $\Sigma = (\Sigma_p, \Sigma_f)$. Son interprétation I_H est une interprétation d'Herbrand si $H(\Sigma_f)$ lui sert de domaine et si les foncteurs sont interprétés comme des fonctions symboliques :

$$f(t_1, t_2, \ldots, t_n)^{IH} =_{df} f(t_1^{IH}, t_2^{IH}, \ldots, t_n^{IH}).$$

Ainsi, pour tout atome clos $A \in H(\Sigma_f)$, $I_H \models A$ si et seulement si $A \in I_H$.

- Définition 2 : Une proposition du premier ordre est universelle si elle est en forme prénexe et si les quantificateurs qui apparaissent dans son préfixe sont tous universels.

- Théorème : Un ensemble de propositions universelles est cohérent si et seulement si il a un modèle d'Herbrand.

Systèmes formels de premier ordre

Å <u>Extension de la méthode des tableaux :</u>

Soit U un ensemble infini de nouvelles variables : U ž Var = ∅.

Extension de la classification de formules : $X \in U$ étant une nouvelle variable.

$\forall$-formules < $\forall x\ \Psi$>	Formules dérivées
$\forall x\ \Phi$	$\Phi(x \backslash X)$
$\neg\ \exists x\ \Phi$	$\neg\ \Phi(x \backslash X)$

Systèmes formels de premier ordre

X_1, X_2, é , $X_n \in U$ étant les variables introduites dans ∃ï formules avant son développement, f étant un nouveau foncteur (une constante si n=0)

∃ï formules < $\exists x\ \Psi$ >	Formules dérivées
$\exists x\ \Phi$	$\Phi(x\backslash f(X_1, X_2,$ é $, X_n))$
$\neg\forall x\ \Phi$	$\neg\Phi(x\backslash f(X_1, X_2,$ é $, X_n))$

Systèmes formels de premier ordre

Å Nouvelle règles dôextension dôune branche :

$\forall$-extension :

{T, [Bô <$\forall$x Φ>]} alors {T, [Bô <$\forall$x Φ>, <Φ(x\X)>]}

(X $\in$ U étant nouvelle variable par rapport à Bô)

$\exists$ï extension :

{T, [Bô <$\exists$x Φ>]} alors {T, [Bô <$\exists$x Φ>, <Φ(x\f(X_1,X_2,é ,X_n))>]}

Å Nouvelle règle de fermeture dôune branche :

Une branche B=[ϕ_1, ϕ_2,é ϕ_n] est fermée par unification σ:

(Var $\cup$ U)→T_Σ[Var $\cup$ U], $\phi_j\sigma = \neg\phi_i\sigma$ pour un atome et 1<=i,j <=n.

Un tableau est fermé si toutes ses branches sont fermées par des unificateurs.

La méthode des tableaux du premier ordre

Å Pour prouver que $\Gamma \vdash_t \Phi$:

1- créer un tableau initial : $T_0 = \{[\Gamma, \neg\Phi]\}$, **toute variable libre étant remplacée par une variable** $X_i \in U$.

2- Etablir une dérivation $T_0 \rightarrow_t \ldots \rightarrow_t T_n$, dont chaque pas $T_{n-1} \rightarrow_t T_n$ (n>=1) est :

- soit une extension dûune branche $B \in T_{n-1}$ en utilisant une $\wedge$-formule, ou $\forall$-formule, ou $\exists$-formule,

- soit une ramification dûune branche $B \in T_{n-1}$ par $\vee$-formule.

3- $\Gamma \vdash_t \Phi$: si le dernier tableau T_n est fermée.

La méthode des tableaux du premier ordre

Å <u>Exemple 1:</u> $\exists w\ \forall x\ R(x, w, g(x, w)) \vdash_t \exists w\ \forall x\ \exists y\ R(x, w, y)$

∀ $\exists w\ \forall x\ R(x, w, g(x, w))$ $[\ \Gamma\ ]$

∀ $\neg\exists w\ \forall x\ \exists y\ R(x, w, y)$ $[\neg\Phi]$

Å $\forall x\ R(x, c_0, g(x, c_0))$ $[\ \exists\text{: } 1/c_0\ ]$

∀ $\neg\forall x\ \exists y\ R(x, X_1, y)$ $[\ \forall\text{: } 2/\ X_1\ ]$

Å $R(X_2, c_0, g(X_2, c_0))$ $[\ \forall\text{: } 3/X_2\ ]$

∀ $\neg\exists y\ R(f(X_1), X_1, y)$ $[\ \exists\text{: } 4/\ f(X_1)\ ]$ $X_1 \in$ à la formule 4

∀ $\neg R(f(X_1), X_1, X_3)$ $[\ \forall\text{: } 6/X_3\ ]$

∀ ‘□□□□□[5, 7]

Cette branche est fermée par lôunificateur

$\sigma(X_1) = c_0$, $\sigma(X_2) = f(X_1)$, $\sigma(X_3) = g(f(X_1), c_0)$

De 5 et 7, ce qui ferme le tableau

La méthode des tableaux du premier ordre

Å <u>Exemple 2 :</u> $\exists x\ \forall y\ P(x, y) \vdash_t \forall x\ \exists y\ P(y, x)$

∀ $\exists x\ \forall y\ P(x, y)$ [Γ]

∀ $\neg\forall x\ \exists y\ P(y, x)$ [$\neg\Phi$]

∀ $\forall y\ P(c_0, y)$ [$\exists$: 1/c_0]

Å $P(c_0, X_1)$ [$\forall$: 3/ X_1]

∀ $\neg\exists y\ P(y, c_1)$ [$\exists$: 2/c_1] aucune U-variable dans 2

∀ $\neg P(X_2, c_1)$ [$\forall$: 5/ X_2]

∀ ‘□□□□□[4, 6]

La branche est le tableau sont fermés par lû unificateur

$\sigma(X_1) = c_1$, $\sigma(X_2) = c_0$,

La méthode des tableaux du premier ordre

Å <u>Théorème :</u>

Soit $T_1 \rightarrow T_2$ par lô application dô une règle r à une branche de T_1. Alors :

Å sô il existe un modèle $M_1 \models T_1$, il en existe aussi pour $M_2 \models T_2$,

Å T_1 est équi-cohérente à (équivalente) T_2

<u>Corollaire (correction) :</u>

Si $\Gamma \vdash_t \Phi$, alors $\Gamma \models \Phi$.

Systèmes formels de premier ordre

Å <u>Extension de la méthode de résolution :</u>

La résolution du premier ordre sõapplique aux ensembles de clauses du premier ordre. Un tel ensemble Γ_1 étant donné, la règle étendue de résolution quõon peut lui appliquer est la suivante :

Règle de résolution du premier ordre : $\Gamma_1 \rightarrow_r \Gamma_2$ si :

$\Gamma_1 = \Gamma \text{ U } \{Cl_1 \vee A_1 \vee é \quad \vee A_n\} \text{ U } \{Cl_2 \vee \neg B_1 \vee é \quad \vee \neg B_m\}$

$\Gamma_2 = \Gamma_1 \text{ U } \{Cl_1 \vee Cl_2\}\theta$,

A_1, é , A_n, B_1, é ,B_m étant des atomes (n, m >0),

et θ lõunificateur le plus général de {A_1, é , A_n, B_1, é ,B_m }

cõest à dire $A_1\theta =é = A_n\theta = B_1\theta =é = B_m\theta$

La méthode de résolution du premier ordre

Å Pour prouver que $\Gamma \vdash_r \Phi$:

1- Transformer Γ U $\{\neg\Phi\}$ en forme normale clausale $\Gamma_{0.}$

2– Etablir une dérivation $\Gamma_0 \rightarrow_r \Gamma_1 \rightarrow_r .... \rightarrow_r \Gamma_k$ où ‘ $\in \Gamma_{k.}$

<u>Exemple :</u>

Prouver : $\forall$x (P(x) $\Rightarrow$ $\exists$y Q(y)) $\vdash_r$ $\exists$x P(x) $\Rightarrow$ $\exists$y Q(y)

1- Skolemisation :

$\forall$x (P(x) $\Rightarrow$ $\exists$y Q(y))

$\forall$x $\exists$y ($\neg$P(x) $\vee$ Q(y))

{$\neg$ P(X) $\vee$Q(f(X))}

1- Skolemisation :

$\neg$($\exists$x P(x) $\Rightarrow$ $\exists$y Q(y))

$\neg$ ($\neg$ $\exists$x P(x) $\vee$ $\exists$y Q(y))

{P(a), $\neg$ Q(Y)}

2 ï Dérivation

∀ $\neg$ P(X) $\vee$ Q(f(X))

Å P(a)

∀ $\neg$Q(Y)

Å Q(f(a)) [1, 2] σ_1 ={X→a}

∀ ‘□□[1, 2] σ_2 ={Y→f(a)}

La méthode de résolution du premier ordre

Exemple : Prouver :

$\forall x\,(P(x) \Rightarrow (S(x)) \vee Q(g(x)))),\ \exists y\,\neg S(g(y)),\ \forall x\,P(x) \vdash_r \exists y\,Q(y)$

1- Skolemisation :

$\forall x\,(P(x) \Rightarrow (S(x)) \vee Q(g(x))))$ équivalent à : $\forall x\,(\neg P(x) \vee S(x)) \vee Q(g(x)))$:

$(\neg P(X) \vee S(X)) \vee Q(g(X)))$

$\exists y\,\neg S(g(y))$ équivalent à : $\neg S(g(c_0))$

$\forall x\,P(x)$ équivalent à : $P(X)$

$\neg\exists y\,Q(y)$ équivalent à : $\neg Q(Y)$

2- Dérivation :

∀ $\neg P(X) \vee S(X)) \vee Q(g(X))$

∀ $\neg S(g(c_0))$

Å $P(X)$

∀ $\neg Q(Y)$

Å $S(X)) \vee Q(g(X))$ [1, 3]

Å $Q(g(g(c_0)))$ [2, 5] $\sigma_1 = \{X \rightarrow g(c_0)\}$

∀ ‘□□□[4, 6] $\sigma_2 = \{Y \rightarrow g(g(c_0))\}$

La méthode de résolution du premier ordre

Å <u>Théorème :</u>

Soient deux ensembles de clauses Γ_1, Γ_2 . Si $\Gamma_1 \rightarrow_r \Gamma_2$,

alors $\Gamma_1 \models \Gamma_2$.

<u>Corollaire (correction) :</u>

Si $\Gamma \vdash_r \Phi$, alors $\Gamma \models \Phi$.