Logique

François Schwarzentruber

Préparation à l'option informatique de l'agrégation de mathématiques
ÉNS Rennes

Table des matières

Première partie

Logique propositionnelle

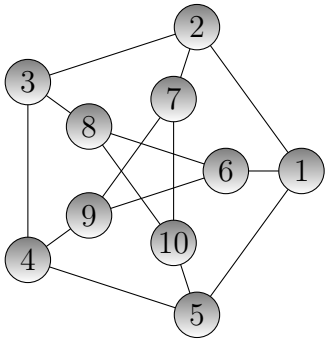
Chapitre 1

Logique propositionnelle et problème SAT

Points du programme de l'agrégation
Calcul propositionnel : syntaxe et sémantique. Tables de vérité. Tautologies.

1.1 Motivation : problème de coloration

On souhaite trouver un 3-coloriage pour un graphe G non orienté. L'idée est d'exprimer les contraintes de coloriage avec une formule de la logique propositionnelle.



1.2 Un modèle : une valuation

Soit AP un ensemble dénombrable de propositions atomiques. Un modèle de la logique propositionnelle est une valuation. Formellement :

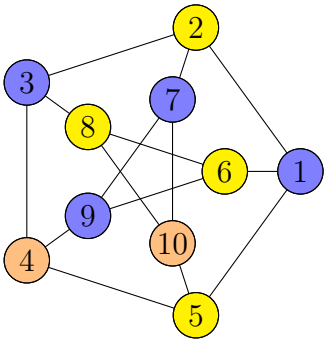
Définition 1 (valuation)
Une **valuation** V est une fonction de AP dans $\{0, 1\}$.

Exemple 1 *Une valuation représente une 3-coloriation. On modélise les faits du monde par des propositions atomiques. Pour toute sommet s et toute couleur c, on introduit la proposition atomique*

$$p_{s,c}$$

qui signifie intuitivement que

la couleur du sommet s est c.



Ainsi, par exemple la 3-coloration donnée par :
est représentée par la valuation V définie par :

- $V(p_{1,●}) = V(p_{2,●}) = \ldots = 1$;
- $V(p_{1,●}) = V(p_{1,●}) = \ldots = 0$.

1.3 Langage

1.3.1 Syntaxe

Définition 2 (syntaxe du langage de la logique propositionnelle)
Le **langage de la logique propositionnelle** $\mathcal{L}$ est défini par la grammaire suivante :

$$\varphi \ ::= \ \bot \ \mid \ p \ \mid \ \neg\varphi \ \mid \ (\varphi \vee \varphi) \ \mid \ (\varphi \wedge \varphi)$$

où p désigne une proposition atomique dans AP.

On introduit les abréviations suivantes :
— $(\varphi \to \psi)$ pour $(\neg\varphi \vee \psi)$;
— $(\varphi \leftrightarrow \psi)$ pour $((\varphi \to \psi) \wedge (\psi \to \varphi))$.
On omet les parenthèses quand elles sont évidentes.

Exemple 2 *On écrit* $p \vee q \vee r$ *au lieu de* $(p \vee (q \vee r))$.

1.3.2 Sémantique

Définition 3 (condition de vérité)
$V \models \varphi$ est défini par induction structurelle sur φ :
— $V \not\models \bot$;
— $V \models p$ si $V(p) = 1$;
— $V \models \neg\varphi$ si not $V \models \varphi$;
— $V \models (\varphi \vee \psi)$ si $V \models \varphi$ ou $V \models \psi$;
— $V \models (\varphi \wedge \psi)$ si $V \models \varphi$ et $V \models \psi$.

Tables de vérité

Dans une table de vérité, chaque ligne correspond à une valuation V, et on inscrit 1 dans la colonne pour φ si $V \models \varphi$ et 0 si $V \not\models \varphi$.

p	q	$\neg q$	$p \vee \neg q$
0	0	1	1
0	1	0	0
1	0	1	1
1	1	0	1

Définition 4 (satisfiable)
φ est **satisfiable**
s'il existe une valuation V telle que $V \models \varphi$.

		φ
		.
		1
		.
		.

Définition 5 (valide)
φ est **valide** [1]
si pour toute valuation V, on a $V \models \varphi$.

		φ
		1
		1
		⋮
		1

1. On dit aussi φ est une tautologie

1.4 Théories

Définition 6 (théorie)
Une **théorie** T est un ensemble de formules.

Définition 7 (notation $V \models T$)
On écrit $V \models T$ pour dire que pour toute formule ψ de T, on a $V \models \psi$.

Définition 8 (conséquence sémantique)
φ est **conséquence sémantique** d'une théorie T, noté $T \models \varphi$, si pour toute valuation V, on a $V \models T$ implique $V \models \varphi$.

1.5 Systèmes de connecteurs complets

Définition 9 (formules équivalentes)
φ et ψ sont **équivalentes** si pour toute valuation V, ($V \models \varphi$ si et seulement $V \models \psi$).

Proposition 1 *Le système de connecteurs $\{\neg, \wedge\}$ est **complet**, c'est-à-dire que toute formule est équivalente à une formule qui ne contient que les connecteurs $\neg$ et $\wedge$. De même pour les systèmes $\{\neg, \rightarrow\}$, $\{\neg, \vee\}$.*

1.6 Problèmes de décision

Définition 10 (problème SAT)
Le **problème SAT** est le problème de décision suivant :
- entrée : φ ;
- sortie : oui si φ est satisfiable ; non, sinon.

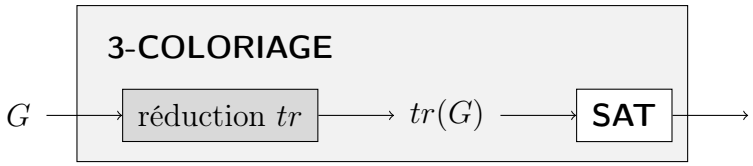
Théorème 1 *Le problème SAT est NP-complet.*

Définition 11 (problème VALID)
Le **problème VALID** est le problème de décision suivant :
- entrée : φ ;
- sortie : oui si φ est valide ; non, sinon.

Théorème 2 *Le problème VALID est coNP-complet.*

1.7 Application pratique

On réduit le problème de 3-coloriage au problème SAT :



Chapitre 2

Formes normales

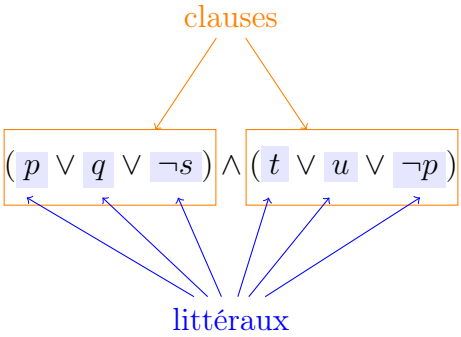
Points du programme de l'agrégation
Formes normales, forme clausale.

2.1 Forme normale conjonctive

Définition 12 (forme normale conjonctive [1])

'un et de ou' $\bigwedge_{i=1}^{n} \bigvee_{j=1}^{m_i} \ell_{i,j}$ peut aussi se réécrire $\bigwedge_{i=1}^{n} \left(\bigwedge_{j=1}^{k_i} p_{i,j} \to \bigvee_{j=1}^{k'_i} p'_{i,j} \right)$

Définition 13 (vocabulaire)



Théorème 3 *Toute formule est équivalente à une FNC* *...potentiellement exponentiellement plus longue.*

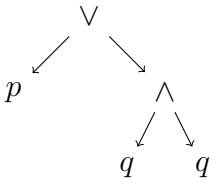
Exemple 3

$$\begin{aligned}(p \wedge q) \vee (r \wedge s) &\equiv ((p \wedge q) \vee r) \wedge ((p \wedge q) \vee s) \\ &\equiv (p \vee r) \wedge (q \vee r) \wedge (p \vee s) \wedge (q \vee s)\end{aligned}$$

2.2 Transformation de Tseitin ([BA12], p. 91)

Théorème 4 *Toute formule φ est équisatisfiable à une FNC $tr(\varphi)$* *...de taille $O(\varphi)$.*

Exemple 4



$p \vee (q \wedge r)$ *est satisfiable ssi*

$\alpha_{(p\vee(qr))} \quad \wedge(\alpha_{(p\vee(qr))} \leftrightarrow (p \vee \alpha_{(q\wedge r)}))$
$\wedge(\alpha_{(q\wedge r)} \leftrightarrow (q \wedge r))$ *satisfiable ssi*

$\alpha_{(p\vee(qr))} \wedge \quad (\alpha_{(p\vee(qr))} \rightarrow (p \vee \alpha_{(q\wedge r)}))$
$\wedge(p \rightarrow \alpha_{(p\vee(qr))}) \wedge (\alpha_{(q\wedge r)} \rightarrow \alpha_{(p\vee(qr))})$
$\wedge(\alpha_{(q\wedge r)} \rightarrow q)$ *satisfiable ssi*
$\wedge(\alpha_{(q\wedge r)} \rightarrow r)$
$\wedge((q \wedge r) \rightarrow \alpha_{(q\wedge r)})$
$\alpha_{(p\vee(qr))} \wedge \quad (\alpha_{(p\vee(qr))} \rightarrow (\alpha_p \vee \alpha_{(q\wedge r)}))$
$\wedge(\alpha_p \rightarrow \alpha_{(p\vee(qr))}) \wedge (\alpha_{(q\wedge r)} \rightarrow \alpha_{(p\vee(qr))})$
$\wedge(\alpha_{(q\wedge r)} \rightarrow \alpha_q)$ *satisfiable.*
$\wedge(\alpha_{(q\wedge r)} \rightarrow \alpha_r)$
$\wedge((\alpha_q \wedge \alpha_r) \rightarrow \alpha_{(q\wedge r)})$

où $\alpha_{(p\vee(qr))}$, $\alpha_{(q\wedge r)}$, α_p, α_q et α_r sont des propositions atomiques fraîches dont la signification intuitive est respectivement '$p \vee (q \wedge r)$ est vraie', '$(q \wedge r)$ est vraie', 'p est vraie', 'q est vraie' et 'r est vraie'.

On introduit des propositions atomiques α_ψ pour toute les sous-formules ψ. La signification intuitive de α_ψ est 'la sous-formule ψ est vraie'. On définit :

$$tr(\varphi) = \bigwedge_{\psi \in SF(\varphi)\setminus AP} r(\psi) \wedge \alpha_\varphi$$

où $SF(\varphi)$ est l'ensemble des sous-formules de φ et $r(\psi)$ est définie par :
— $r(\neg\psi) = (\neg\alpha_{\neg\psi} \vee \neg\alpha_\psi) \wedge (\alpha_{\neg\psi} \vee \alpha_\psi)$;
— $r(\psi_1 \vee \psi_2) = (\alpha_{\psi_1\vee\psi_2} \rightarrow (\alpha_{\psi_1} \vee \alpha_{\psi_2})) \wedge (\alpha_{\psi_1} \rightarrow \alpha_{\psi_1\vee\psi_2}) \wedge (\alpha_{\psi_2} \rightarrow \alpha_{\psi_1\vee\psi_2})$;
— $r(\psi_1 \wedge \psi_2) = (\alpha_{\psi_1\wedge\psi_2} \rightarrow \alpha_{\psi_1}) \wedge (\alpha_{\psi_1\wedge\psi_2} \rightarrow \alpha_{\psi_2}) \wedge ((\alpha_{\psi_1} \wedge \alpha_{\psi_2}) \rightarrow \alpha_{\psi_1\wedge\psi_2})$.
La formule $r(\psi)$ exprime les conditions de vérité de ψ.

Proposition 2 *La taille de $tr(\varphi)$ est $O(\varphi)$.*

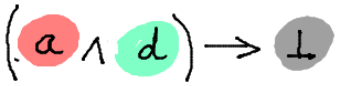
Proposition 3 *φ satisfiable ssi $tr(\varphi)$ satisfiable.*

2.3 Fragments syntaxiques basés sur les FNC

2.3.1 Formules de Horn ([DPV16], p. 157)

Définition 14 (clause de Horn)
Une **clause de Horn** est une clause de la forme $(p_1 \wedge \cdots \wedge p_n) \to p$ ou $(p_1 \wedge \cdots \wedge p_n) \to \bot$.

Définition 15 (formule de Horn)
Une **formule de Horn** est une conjonction de clauses de Horn.



Définition 16 (problème HORN-SAT)
Le **problème HORN-SAT** est le problème décision :
- Entrée : une FNC où les clauses sont de Horn ;
- Sortie : oui si elle est satisfiable ; non sinon.

Algorithme

fonction satHorn(φ)
  $V :=$ valuation où toutes les propositions atomiques sont fausses
  tant que il existe une clause $\Gamma \to p$ avec $V \not\models \Gamma \to p$ **do**
    $V := V[p := \top]$
  si une clause $\Gamma \to \bot$ avec $V \models \Gamma$ **alors**
    retourner insatisfiable
  sinon
    retourner V

Théorème 5 *Le problème HORN-SAT est dans P.*

Applications

Calcul de premier, suivant en analyse syntaxique LL(1), des non-terminaux productifs d'une grammaire algébrique, des non-terminaux qui engendre ϵ [LS15].

2.3.2 2-formes normales

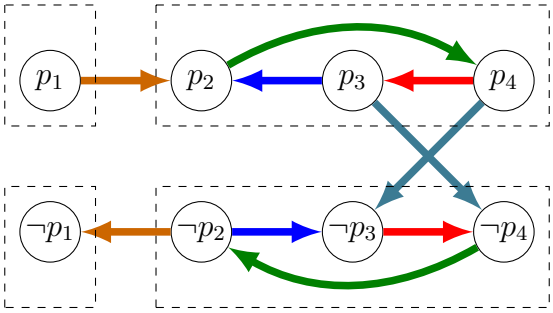
Définition 17 (problème 2SAT)
Le **problème 2SAT** est le problème décision :
- Entrée : une FNC avec 2 littéraux par clause ;
- Sortie : oui si elle est satisfiable ; non sinon.

Théorème 6 *Le problème 2-SAT est dans P.*

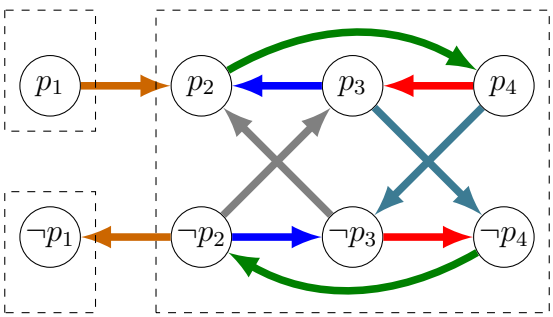
Algorithme

fonction sat2sat(φ)
  Construire le graphe G_φ où :
    - les sommets sont les propositions atomiques de φ et leurs négations ;
    - il y a arcs (α, β) si $\alpha \to \beta$ est équivalente à une clause de φ
  Calculer les composantes fortement connexes de G_φ
  si il existe p tel que p et $\neg p$ sont dans la même composante fortement connexe **alors**
    retourner insatisfiable
  sinon
    retourner satisfiable

Exemple 5
$(p_3 \vee \neg p_4) \wedge (p_2 \vee \neg p_3) \wedge (p_4 \vee \neg p_2) \wedge (\neg p_1 \vee p_2) \wedge (\neg p_4 \vee \neg p_3)$ *est satisfiable.*



Exemple 6
$(p_3 \vee \neg p_4) \wedge (p_2 \vee \neg p_3) \wedge (p_4 \vee \neg p_2) \wedge (\neg p_1 \vee p_2) \wedge (\neg p_4 \vee \neg p_3) \wedge (p_3 \vee p_2)$ *n'est pas satisfiable.*



Proposition 4 *sat2sat(φ) retourne* satisfiable *ssi φ est satisfiable.*

Idée de la démonstration.

($\Leftarrow$) Soit V une valuation telle que $V \models \varphi$. Par l'absurde, supposons qu'il existe p et $\neg p$ dans la même composante fortement connexe (cfc). Sans perte de généralité, supposons que $V \models p$. Comme il y a un chemin de $p = \ell_0, \ell_1, \ldots, \ell_n = \neg p$ dans G_φ, et comme $V \models \ell_i \to \ell_{i+1}$, on montre par récurrence sur i que $V \models \ell_i$ pour tout i. Donc $V \models \neg p$. Contradiction.

($\Rightarrow$) Supposons que sat2sat(φ) retourne satisfiable. Considérons l'algorithme suivant :

fonction construireValuation(G)
  si le graphe G est vide **alors**
    retourner valuation partielle vide
  sinon
    Soit C une cfc finale de G
    $V :=$ construireValuation($G - (C \cup \hat{C})$)
      où $\hat{C} = \{\neg \ell \mid \ell \in C\}$ et $G - (C \cup \hat{C})$ est le graphe G restreint aux sommets hors de $C \cup \hat{C}$.
    Mettre tous les littéraux de C à vrai dans V
    retourner V

On considère la valuation $V :=$ construireValuation(G_φ).

Fait 1 *L'opération 'Mettre tous les littéraux de C à vrai dans V' est bien définie.*

Idée de la démonstration.

Comme p et $\neg p$ ne sont pas dans la même cfc, on donne une valeur unique à p. ■

Fait 2 *V est une valuation totale sur les propositions apparaissant dans φ.*

Idée de la démonstration.

Car on considère tous les sommets de G_φ. ■

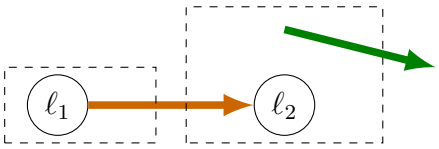
Fait 3 *$V \models \varphi$.*

Idée de la démonstration.

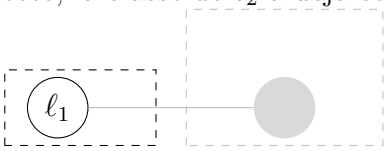
On considère une clause $\ell_1 \to \ell_2$ de φ. Montrons $V \models \ell_1 \to \ell_2$. Supposons que $V \models \ell_1$. Ainsi, ℓ_1 a été mis à vrai dans l'algorithme : on avait $\ell_1 \in C_1$ où C_1 est finale dans un certain graphe G_1.

— Si ℓ_1 et ℓ_2 dans une même cfc de G_φ, alors ℓ_2 est aussi mis à vraie, comme ℓ_1.

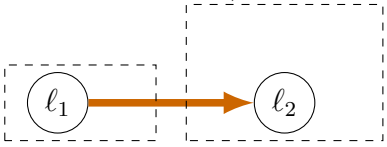
— Sinon, dans G_φ :



Mais au moment, où ℓ_1 est affecté, la classe de ℓ_2 a déjà été supprimée car C_1 est finale :



Mais lorsque ℓ_2 a été supprimé, il n'était pas dans une classe initiale $\bar{C}$ car $\ell_1 \to \ell_2$ était un arc entrant dans dans la classe de ℓ_2. Ainsi, ℓ_2 était dans une classe finale C_2



et il a été affecté à vraie. Ainsi, $V \models \ell_2$.

■

■

Application

Soit $\mathcal{C}$ un ensemble fini de créneaux horaires, soit $\mathcal{P}$ un ensemble fini de professeurs et G un ensemble fini de groupes d'élèves. Chaque professeur $i \in \mathcal{P}$ dispose d'un ensemble de créneaux disponibles $\mathcal{C}_i \subseteq \mathcal{C}$, chaque groupe $j \in \mathcal{G}$ possède un ensemble de créneaux disponibles $\mathcal{D}_i \subseteq \mathcal{C}$ et d'un ensemble d'heures $\mathcal{R}_{ij}$ à enseigner à un groupe d'élèves $j \in \mathcal{G}$. On souhaite créer un emploi du temps qui satisfait les contraintes.

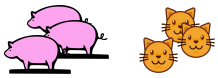
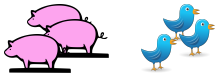
Dans le cas général, le problème est NP-complet [EIS75] mais si pour tout professeur $i \in \mathcal{P}$, $\mathbf{card}(\mathcal{C}_i) \leq 2$, alors le problème est réductible à 2SAT en temps polynomial et est donc dans P. Dans ce cas, si $\mathcal{R}_{ij} \leq 2$. Si $\mathcal{R}_{ij} = 2$, alors on peut supprimer le professeur i et considérer le groupe j comme disponible sur ces deux créneaux. Si $\mathbf{card}(\mathcal{C}_i) = 1$, alors on supprime le professeur i et on affectue un créneau à l'unique groupe j demandé et on considère le groupe j comme non disponible sur ce créneau.

On considère alors que $\mathcal{R}_{ij} \leq 1$. Soit $\mathcal{COMP}$ l'ensemble des couples (i, j) où $i \in \mathcal{P}$, $j \in \mathcal{G}$ et $\mathcal{R}_{ij} = 1$. Pour tout $(i, j) \in \mathcal{COMP}$, on introduit la variable propositionnelle v_{ij} qui signifie 'i fait cours à j sur son premier créneau disponible'.

On note c_i^1 et c_i^2 le créneau n° 1 et n° 2 du professeur i. L'instance de 2-SAT est alors l'ensemble des clauses suivantes :

- v_{ij} si $(i, j) \in \mathcal{COMP}$ si $c_i^2 \notin \mathcal{D}_j$;
- $\neg v_{ij}$ si $(i, j) \in \mathcal{COMP}$ si $c_i^1 \notin \mathcal{D}_j$;
- pour tout (i, i', j) tel que $(i, j), (i', j) \in \mathcal{COMP}$ et $i \neq i'$:
  - $(v_{ij} \wedge v_{i'j}) \rightarrow \bot$ si $c_i^1 = c_{i'}^1$;
  - $(\neg v_{ij} \wedge v_{i'j}) \rightarrow \bot$ si $c_i^2 = c_{i'}^1$;
  - $(v_{ij} \wedge \neg v_{i'j}) \rightarrow \bot$ si $c_i^1 = c_{i'}^2$;
  - $(\neg v_{ij} \wedge v_{i'j}) \rightarrow \bot$ si $c_i^2 = c_{i'}^2$;
- pour tout (i, j, j') tel que $(i, j), (i', j) \in \mathcal{COMP}$ et $j \neq j'$:
  - $(v_{ij} \wedge v_{ij'}) \rightarrow \bot$;
  - $(\neg v_{ij} \wedge \neg v_{ij'}) \rightarrow \bot$.

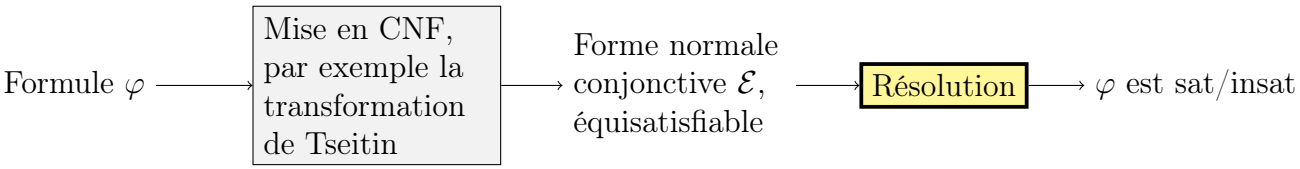
	lundi	mardi	mercredi	jeudi	vendredi
8h-10h					
10h-12h					
14h-16h					
16h-18h					

Professeur	Groupes
	
	
	

Chapitre 3

Résolution en logique propositionnelle

Points du programme de l'agrégation
Théorème de complétude du calcul propositionnel



3.1 Règle de résolution

Définition 18 (règle de résolution)
([BA12], p. 82) La **règle de résolution** est la règle

$$\frac{(\,p \vee \ell_1 \vee \dots \ell_n\,) \qquad (\,\neg p \vee \ell'_1 \vee \dots \vee \ell'_k\,)}{(\,\ell_1 \vee \dots \ell_n \vee \ell'_1 \vee \dots \vee \ell'_k\,)}$$

à permutation près des littéraux et en supprimant les répétitions de littéraux [1].
La nouvelle clause obtenue ($\ell_1 \vee \dots \ell_n \vee \ell'_1 \vee \dots \vee \ell'_k$) s'appelle le **résolvant**.

Exemple 7 *Le modus ponens* $\frac{p \qquad (p \to q)}{q}$ *en est un cas particulier.*

3.2 Arbre de preuve

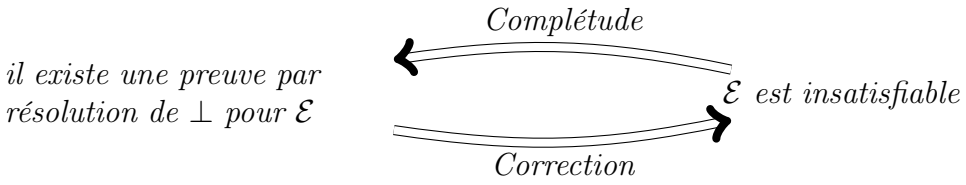
Définition 19 (preuve par résolution)
Soit $\mathcal{E}$ une forme normale conjonctive [2]. Une **preuve par résolution** de φ pour $\mathcal{E}$ est un arbre fini étiqueté par des clauses tel que :

- Les feuilles sont des clauses de $\mathcal{E}$;
- La racine est φ ;
- Chaque nœud interne correspond à l'application de la règle de résolution.

1. Une présentation plus bas niveau les clauses sont représentées des ensembles évite ce problème. Par exemple $(p \vee \neg q)$ est représentée par l'ensemble $\{p, \neg q\}$. Nous préférons ici une présentation haut niveau.
2. Certains auteurs [BA12] parlent d'ensemble de clauses.

3.3 Correction et complétude

Théorème 7 *([BA12], p. 82) Soit $\mathcal{E}$ un forme normale conjonctive.*



Idée de la démonstration.

$\boxed{\Rightarrow}$ Par contraposée, supposons $\mathcal{E}$ satisfiable, i.e. il existe une valuation V telle que $V \models \mathcal{E}$.

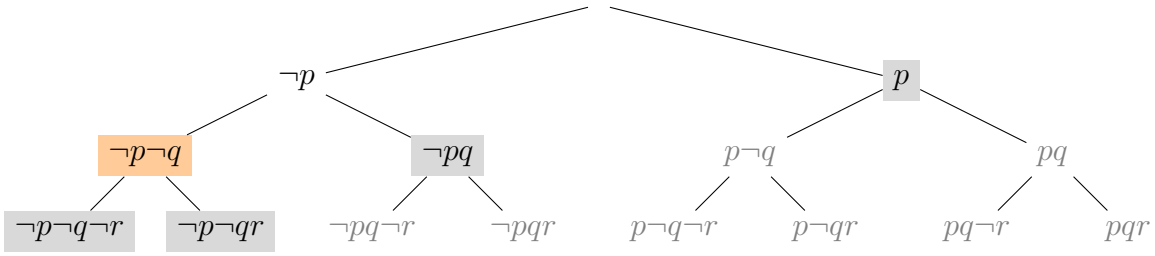
Lemme 1 *Si* $\dfrac{C_1 \qquad C_2}{C}$, *$V \models C_1$ et $V \models C_2$ alors $V \models C$.*

On montre par induction structurelle sur les preuves $\dfrac{\vdots}{\varphi}$ pour $\mathcal{E}$ la propriété

$$\mathcal{P}(\frac{\vdots}{\varphi}) : V \models \varphi$$

Ainsi, il n'existe pas de preuve par résolution de $\bot$ pour $\mathcal{E}$.

$\boxed{\Leftarrow}$ Supposons $\mathcal{E}$ insatisfiable. Considérons l'arbre de décision sur les propositions $p_1, \ldots, p_n$ apparaissant dans $\mathcal{E}$. Un nœud interne correspond à une valuation partielle et une feuille à une valuation totale sur $p_1, \ldots, p_n$. Un **nœud d'échec** est un nœud le plus proche de la racine, dont la valuation partielle rend une des clauses de $\mathcal{E}$ fausse. Un **nœud d'inférence** est un nœud dont les fils sont des nœuds d'échec.



Lemme 2 *Si $\mathcal{E}$ est insatisfiable, alors il existe un nœud d'inférence.*

Idée de la démonstration.

Par l'absurde, supposons qu'il n'y a pas de nœud d'inférence. Alors on considère un nœud d'échec n_0. Son frère n'est pas un nœud d'échec mais son sous-arbre en contient un strictement plus profond. On construit une suite infinie n_0, n_1, etc. de nœuds d'échec de plus en plus profond. Contradiction. ■

Lemme 3 *Soit n un nœud d'inférence de fils n_1 et n_2. Si C_1 est falsifié par n_1 et C_2 est falsifié par n_2, alors on peut appliquer la règle de résolution* $\dfrac{C_1 \qquad C_2}{C}$ *et n falsifie C.*

L'algorithme suivant construit une preuve par résolution de $\bot$ pour $\mathcal{E}$:

tant que $\bot$ n'est pas produite **do**
| Appliquer la règle de résolution sur un nœud d'inférence.

L'algorithme termine car le nombre de nœuds d'échec décroit strictement : avec $\mathcal{E} \wedge C$ à la place de $\mathcal{E}$, un ancêtre du nœud d'inférence considéré devient nœud d'échec. ■

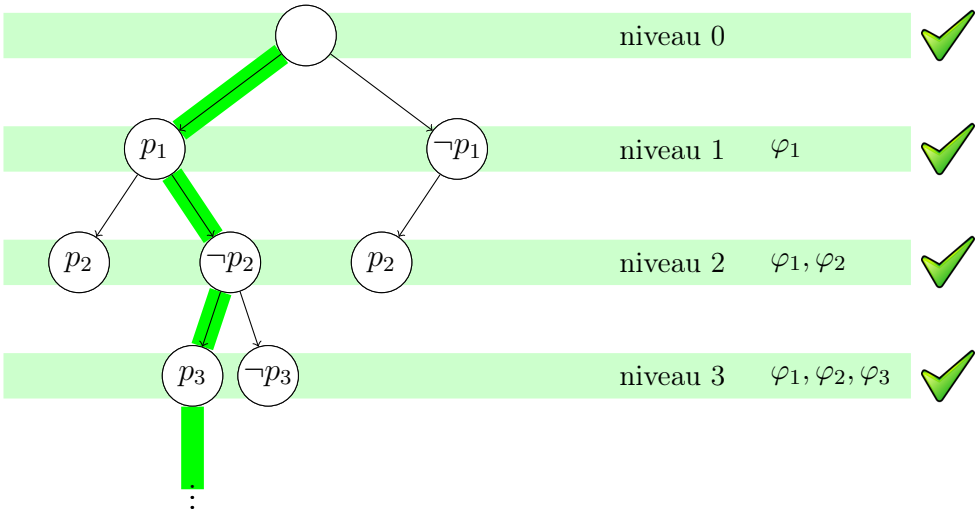
Chapitre 4

Théorème de compacité

Théorème 8 *Soit E un ensemble de formules de la logique propositionnelle.*
Si tout sous-ensemble fini de E est satisfiable, alors E l'est aussi.

Idée de la démonstration.
On considère :
— $\varphi_1, \varphi_2, \ldots$ une énumération des formules de E ;
— et $p_1, p_2, \ldots$ une énumération des propositions atomiques.

On construit l'arbre [1] suivant par récurrence.
— La racine est le nœud de niveau 0.
— Supposons que tous les nœuds de niveau $n-1$ soient construits. On attache :
  — on attache un nœud p_n à une branche $\pm p_1, \ldots, \pm p_{n-1}$
    si $\{\pm p_1, \ldots, \pm p_{n-1}, p_n, \varphi_1, \ldots, \varphi_n\}$ est satisfiable ;
  — on attache un nœud $\neg p_n$ à une branche $\pm p_1, \ldots, \pm p_{n-1}$
    si $\{\pm p_1, \ldots, \pm p_{n-1}, \neg p_n, \varphi_1, \ldots, \varphi_n\}$ est satisfiable ;



Comme pour tout $n \in \mathbb{N}$, $\{\varphi_1, \ldots, \varphi_n\}$ est satisfiable. On peut toujours continuer à attacher des nœuds. Ainsi, l'arbre est infini.

Comme l'arbre est infini et à branchement fini, d'après le **lemme de König**, il a une branche infinie. Cette branche définit une valuation qui satisfait T. ■

1. C'est un sous-arbre de l'arbre de décision.

Corollaire 1 (Théorème de De Bruijn-Erdős) *Un graphe G est 3-coloriable si et seulement si tout sous-graphe fini de G est 3-coloriable.*

IDÉE DE LA DÉMONSTRATION.
$\boxed{\Rightarrow}$ Immédiat.
$\boxed{\Leftarrow}$ Supposons que tout sous-graphe fini de G est 3-coloriable.
Soit S' un sous-ensemble de sommets de $G = (S, A)$. On note :

$$\begin{aligned} E_{S'} := \quad & \{\Big(\bigvee_{c\in\{\bullet,\bullet,\bullet\}} p_{s,c}\Big) \wedge \bigwedge_{c,c'\in\{\bullet,\bullet,\bullet\}c\neq c'} (p_{s,c} \to \neg p_{s,c'}) \mid s \in S'\} \\ & \{\bigwedge_{c\in\{\bullet,\bullet,\bullet\}} (p_{s,c} \to \neg p_{t,c}) \mid s,t \in S' \text{ et } (s,t) \in A\}. \end{aligned}$$

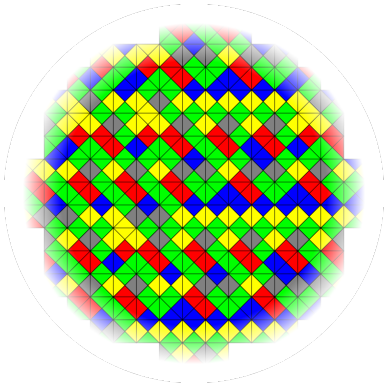
On a :

Lemme 4 *$G_{|S'}$ est 3-coloriable ssi $E_{S'}$ est satisfiable.*

Soit E' une partie finie de E_S. Il existe $S' \subseteq S$, S' fini, tel que $E' \subseteq E_{S'}$. Comme $G_{S'}$ est un graphe fini, il est 3-coloriable, et donc par le lemme 4, $E_{S'}$ est satisfiable. Comme $E' \subseteq E_{S'}$, E' est satisfiable.

D'après le **théorème de compacité de la logique propositionnelle**, E_S est satisfiable. D'après le lemme 4, $G = G_{|S}$ est 3-coloriable. ■

Corollaire 2 *Le plan est pavable par un ensemble de type de tuiles T si et seulement si tout carré fini est pavable par T.*



Deuxième partie

Logique des prédicats du premier ordre

Chapitre 5

Syntaxe et sémantique

Points du programme de l'agrégation
Logique du premier ordre : aspects syntaxiques. Langages, termes, formules. Variables libres et variables liées.

Logique du premier ordre : aspects sémantiques. Interprétation d'une formule dans un modèle. Validité, satisfiabilité.

But : écrire des propriétés et raisonner sur des structures
(base de données, groupes, arithmétiques, etc.).

5.1 Modèles

Définition 20 (signature)
Une **signature** Σ est un ensemble de symboles de fonctions et un ensemble de prédicats.

Exemple 8

Définition 21 (structure)
Une Σ-**structure** (ou Σ-modèle) est la donnée $\mathcal{M} = \langle D, .^{\mathcal{M}} \rangle$ où :
- D est un ensemble non-vide appelé **domaine** ;
- $.^{\mathcal{M}}$ est une fonction appelée **interprétation** qui :
  - à tout symbole de fonction f d'arité n dans Σ associe une fonction $f^{\mathcal{M}} : D^n \to D$;
  - à tout symbole de prédicat p d'arité n dans Σ associe une fonction $p^{\mathcal{M}} : D^n \to \{0, 1\}$.

Exemple 9

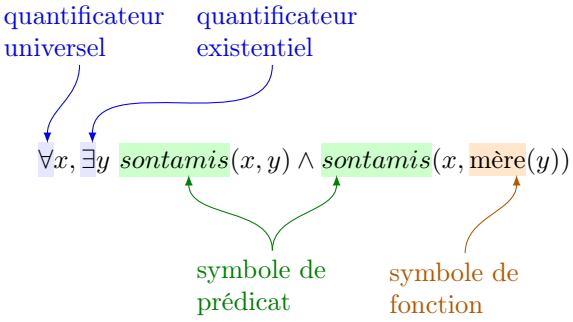
5.2 Langage

5.2.1 Syntaxe

Soit Σ une signature. Soit $\mathcal{V}$ un ensemble dénombrable de variables.

Définition 22 (terme)
L'ensemble des Σ-termes est défini par induction :
- Une variable x est un Σ-terme ;
- Pour tout symbole de fonction f d'arité n de Σ, et pour tous Σ-termes $t_1, \ldots, t_n$,
  - $f(t_1, \ldots, t_n)$ est un Σ-terme.

Définition 23 (formule)
L'ensemble des Σ-formules est défini par induction :
- Pour tout symbole de prédicat p d'arité n de Σ, et pour tous Σ-termes $t_1, \ldots, t_n$,
  - $p(t_1, \ldots, t_n)$ est une Σ-formule ;
- Pour toute Σ-formule φ,
  - $\neg\varphi$ est une Σ-formule ;
- Pour toutes formules φ et ψ,
  - $(\varphi \vee \psi)$ est une Σ-formule ;
  - $(\varphi \wedge \psi)$ est une Σ-formule ;
- Pour toute variable x, toute Σ-formule φ,
  - $\forall x\varphi$ est une Σ-formule ;
  - $\exists x\varphi$ est une Σ-formule.



Définition 24 (formule atomique)
Une **formule atomique** est une formule de la forme $p(t_1, \ldots, t_n)$.

5.2.2 Sémantique

Etant donné un Σ-modèle $\mathcal{M} = (D, .^{\mathcal{M}})$, une assignation des variables v est une fonction de $\mathcal{V}$ dans D. On va définir $\mathcal{M}, v \models \varphi$ qui signifie la Σ-formule φ est vraie dans $\mathcal{M}$ avec l'assignation des variables v.

Définition 25 (assignation étendue aux termes)
D'abord, on étend v en $v^{\mathcal{M}}$ aux termes avec :
- $v^{\mathcal{M}}(x) = v(x)$;
- $v^{\mathcal{M}}(f(t_1, \ldots, t_n)) = f^{\mathcal{M}}(v^{\mathcal{M}}(t_1), \ldots, v^{\mathcal{M}}(t_n))$.

Définition 26 (Conditions de vérité)

- $\mathcal{M}, v \models p(t_1, \ldots, t_n)$ si $p^{\mathcal{M}}(v^{\mathcal{M}}(t_1), \ldots, v^{\mathcal{M}}(t_n)) = 1$;
- $\mathcal{M}, v \models \neg\varphi$ si $\mathcal{M}, v \not\models \varphi$;
- $\mathcal{M}, v \models (\varphi \vee \psi)$ si $\mathcal{M}, v \models \varphi$ ou $\mathcal{M}, v \models \psi$;
- $\mathcal{M}, v \models (\varphi \wedge \psi)$ si $\mathcal{M}, v \models \varphi$ et $\mathcal{M}, v \models \psi$;
- $\mathcal{M}, v \models \exists x\varphi$ s'il existe $d \in D$ tel que $\mathcal{M}, v[x := d] \models \varphi$.
- $\mathcal{M}, v \models \forall x\varphi$ si pour tout $d \in D$ on a $\mathcal{M}, v[x := d] \models \varphi$.

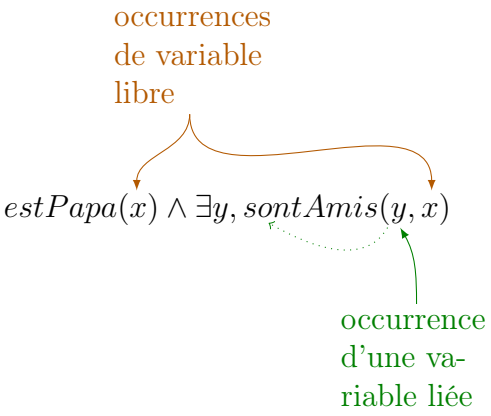
Remarque 1 *Dans le contexte de la logique du premier ordre égalitaire, le symbole de prédicats $=$ est interprété par l'égalité :*
- $\mathcal{M}, v \models t = t'$ *si* $v(t) = v(t')$.

5.3 Variables libres/liées

Définition 27 (occurrence libre)
Une occurrence d'une variable x est libre si elle n'est pas sous la portée d'un quantificateur.

Définition 28 (occurrence liée)
Une occurrence est liée si elle n'est pas libre.

Définition 29 (variable libre)
x est une variable libre dans φ s'il existe une occurrence libre de x dans φ.



Notation 1 *On note $\varphi(x_1, \ldots, x_n)$ pour dire que les variables libres de φ sont dans $\{x_1, \ldots, x_n\}$.*

Exemple 10

Définition 30 (terme clos)
Un terme est **clos** s'il est sans variables libres.

Définition 31 (formule close)
Une formule est **close** si elle est sans variables libres.

Définition 32 (clôture universelle)
La **clôture universelle** de $\varphi(x_1, \ldots, x_n)$ est la formule $\forall x_1 \ldots \forall x_n \varphi$.

Exemple 11

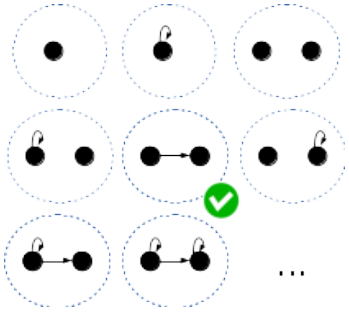
Notation 2 *Si φ est close, $\mathcal{M}, v \models \varphi$ ne dépend pas de v. On note alors $\mathcal{M} \models \varphi$.*

5.4 Satisfiable, valide, conséquence sémantique

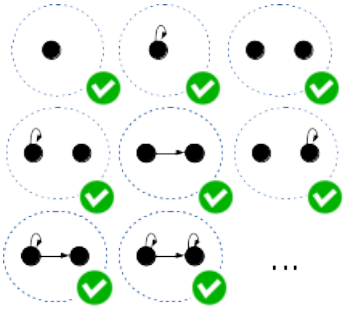
On ne s'intéresse qu'aux formules closes ([BA12], p. 138).

Définition 33 (satisfiable)
Une formule φ close est **satisfiable** s'il existe un modèle $\mathcal{M}$ telle que $\mathcal{M} \models \varphi$.

Définition 34 (valide)
Une formule φ close est **valide** (ou est une tautologie) si pour tout modèle $\mathcal{M}$, on a $\mathcal{M} \models \varphi$.



φ est satisfiable



φ est valide

5.5 Model checking

Définition 35 (problème de model checking en logique du premier ordre)
Le problème de model checking est :
— Entrée : un modèle fini $\mathcal{M}$, une formule close φ ;
— Sortie : oui si $\mathcal{M} \models \varphi$, non sinon.

Proposition 5 *Le problème du model checking en logique du premier ordre est PSPACE-complet.*

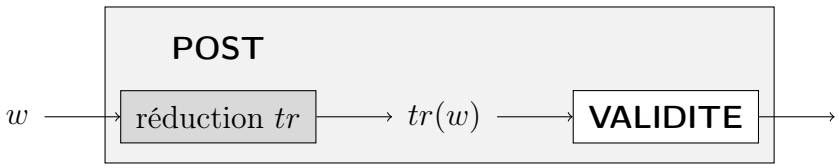
5.6 Problème de la satisfiabilité et problème de la validité

Définition 36 (problème de la satisfiabilité)
Le **problème de la satisfiabilité** est :
— Entrée : une formule close φ ;
— Sortie : oui si φ est satisfiable, non sinon.

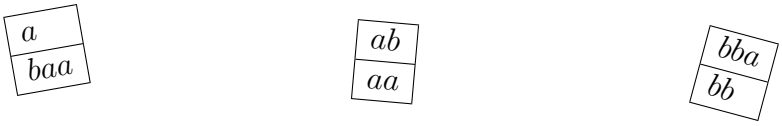
Définition 37 (problème de la validité)
Le **problème de la validité** est :
— Entrée : une formule close φ ;
— Sortie : oui si φ est valide, non sinon.

Proposition 6 *([GLM01], p. 205) Le problème de la satisfiabilité et le problème de la validité en logique du premier ordre sont indécidables.*

Idée de la démonstration.



Considérons une instance du problème de correspondance de Post qui ne contient pas de tuile $\boxed{\begin{matrix}\epsilon\\ \epsilon\end{matrix}}$:



On construit la formule $tr(w) = (\varphi \to \psi)$ où :
— φ est la conjonction de
  — $p(\epsilon, \epsilon)$ où ϵ est un symbole de constante ;
  — $\forall x, y, p(x, y) \to p(u_i(x), v_i(y))$ pour toute tuile $\boxed{\begin{matrix}u_i\\ v_i\end{matrix}}$ où, si $m = a_1 \dots a_n$ est un mot, on note $m(.) = a_n(\dots a_1(t) \dots)$;
— $\psi := \exists x, p(a(x), a(x)) \vee p(b(x), b(x))$.
On a :
— $tr(w)$ est calculable à partir de w ;
— w est une instance positive de **POST** ssi $tr(w)$ est valide.

■

Chapitre 6

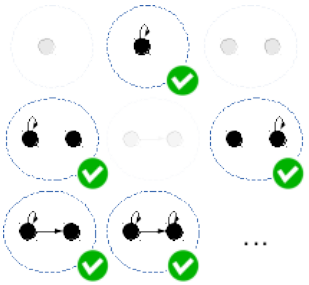
Théories

Points du programme de l'agrégation
Théories cohérentes, théories complètes. Théories décidables, indécidables. Exemples de théories : égalité, arithmétique de Peano.

6.1 Définitions

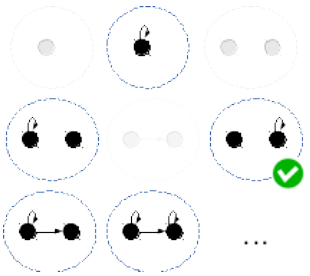
Définition 38 (Théorie)
Une **théorie** T est un ensemble de formules closes.

Définition 39 (T-valide [1])
φ close est **T-valide**, noté $T \models \varphi$, si pour tout modèle $\mathcal{M}$, on a $\mathcal{M} \models T$ implique $\mathcal{M} \models \varphi$.



Le **problème de la T-validité** est :
- Entrée : une formule close φ ;
- Sortie : oui si φ est T-valide, non sinon.

Définition 40 (T-satisfiable)
Une formule φ est **T-satisfiable** s'il existe un modèle $\mathcal{M}$ tel que $\mathcal{M} \models T$ et $\mathcal{M} \models \varphi$.



Le **problème de la T-satisfiabilité** est :
- Entrée : une formule close φ ;
- Sortie : oui si φ est T-satisfiable, non sinon.

Définition 41 (théorie décidable)
Une théorie T est **décidable** si le problème de T-validité est décidable.
ou de manière équivalente le problème de T-satisfiabilité est décidable.

Définition 42 (théorie cohérente)
Une théorie T est **cohérente** si elle est satisfiable.

Définition 43 (théorie (sémantiquement) complète)
Une théorie T est **(sémantiquement) complète** si pour toute formule φ, on a $T \models \varphi$ ou $T \models \neg\varphi$.

1. Ou conséquence sémantique de T.

6.2 Exemples de théories

6.2.1 Théorie de l'égalité

$$T_{eq}\begin{cases}\forall x, x = x\\ \forall x, y, (x = y) \to (y = x)\\ \forall x, y, z, (x = y \wedge y = z) \to x = z\\ \forall x_1, \ldots, x_n, y_1, \ldots, y_n, (\bigwedge_i (x_i = y_i)) \to f(\vec{x}) = f(\vec{y}) \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{pour tout symbole de fonction } f \text{ d'arité } n\\ \forall x_1, \ldots, x_n, y_1, \ldots, y_n, (\bigwedge_i (x_i = y_i)) \to (p(\vec{x}) \leftrightarrow p(\vec{y}))\\ \qquad\qquad\qquad\qquad\qquad\qquad \text{pour tout symbole de prédicat } p \text{ d'arité } n\end{cases}$$

Proposition 7 *Si φ est T_{eq}-satisfiable ssi φ est satisfiable dans un modèle où le symbole $=$ est interprété par l'égalité.*

Idée de la démonstration.

($\Leftarrow$) Soit $\mathcal{M}$ un modèle où le symbole $=$ est interprété par l'égalité tel que $\mathcal{M} \models \varphi$. Par ailleurs, on peut montrer que $\mathcal{M} \models T_{eq}$, donc φ est T_{eq}-satisfiable.

($\Rightarrow$) Soit $\mathcal{M} = (D, .^{\mathcal{M}})$ un modèle tel que $\mathcal{M} \models T_{eq}$ et $\mathcal{M} \models \varphi$. Comme $\mathcal{M} \models \forall x, x = x$, $\mathcal{M} \models \forall x, y, (x = y) \to (y = x)$ et $\mathcal{M} \models \forall x, y, z, (x = y \wedge y = z) \to x = z$, la relation $=^{\mathcal{M}}$ est une relation d'équivalence.

Soit $\mathcal{M}' = (D', .^{\mathcal{M}'})$ tel que

1. $D' = D/_{=^{\mathcal{M}}}$;
2. Pour tout symbole de fonction f d'arité n,

$$\begin{array}{rccl} f^{\mathcal{M}'} : & D'^n & \to & D' \\ & ([t_1], \ldots, [t_n]) & \mapsto & [f^{\mathcal{M}}(t_1, \ldots, t_n)] \end{array}$$

3. Pour tout symbole de prédicat p d'arité n,

$$\begin{array}{rccl} p^{\mathcal{M}'} : & D'^n & \to & \{0, 1\} \\ & ([t_1], \ldots, [t_n]) & \mapsto & p^{\mathcal{M}}(t_1, \ldots, t_n) \end{array}$$

$f^{\mathcal{M}'}$ est bien définie car $\mathcal{M} \models \forall x_1, \ldots, x_n, y_1, \ldots, y_n, (\bigwedge_i (x_i = y_i)) \to f(\vec{x}) = f(\vec{y})$.
$p^{\mathcal{M}'}$ est bien définie car $\mathcal{M} \models \forall x_1, \ldots, x_n, y_1, \ldots, y_n, (\bigwedge_i (x_i = y_i)) \to (p(\vec{x}) \leftrightarrow p(\vec{y}))$.
Enfin, on démontre que pour toute formule ψ,

$$\mathcal{P}(\psi) : \mathcal{M}, v \models \psi \text{ ssi } \mathcal{M}', v' \models \psi$$

où $v'(x) = [v(x)]$.

■

6.2.2 Théorie des groupes ([Lal90], p. 139) ([DNRC01], p. 105)

$$T_{groupes}\begin{cases}\text{Théorie de l'égalité}\\ \forall x, (x \times e = x)\\ \forall, x \times x^{-1} = e\\ \forall x, y, z, ((x \times y) \times z) = (x \times (y \times z))\end{cases}$$

Proposition 8 *$T_{groupes} \models \varphi$ ssi pour tout groupe $(G, .)$, on a $(G, .) \models \varphi$.*

6.2.3 Théorie des ordres denses ([DNRC01], p. 130)

$$T_O \begin{cases} \text{Théorie de l'égalité} \\ \forall x, y, \neg(x < y \wedge y < x) \\ \forall x, y, z, ((x < y \wedge y < z) \rightarrow x < z) \\ \forall x, y, (x < y \vee x = y \vee y < x) \\ \forall x, y, \exists z, (x < y \rightarrow (x < z \wedge z < y)) \\ \forall x, \exists y, (x < y) \\ \forall x, \exists y, (y < x) \end{cases}$$

Proposition 9 *$T_O \models \varphi$ ssi $(\mathbb{Q}, =, <) \models \varphi$ ssi $(\mathbb{R}, =, <) \models \varphi$.*

6.2.4 Théorie des corps clos ([DNRC01], p. 133)

$$T_{CC} \begin{cases} \text{Théorie de l'égalité} \\ \forall x, y, z, (x + y) + z = x + (y + z) \\ \forall x, y, x + y = y + x \\ \forall x, x + 0 = x \\ \forall x, x + (-x) = 0 \\ \forall x, y, z, (x \times y) \times z = x \times (y \times z) \\ \forall x, y, x \times y = y \times x \\ \forall x, x \times 1 = x \\ \forall x, (x \neq 0) \rightarrow \exists y, x \times y = 1 \\ \forall x, y, z, x \times (y + z) = x \times y + x \times z \\ 0 \neq 1 \\ \forall x, y, (x \leq y \wedge y \leq x) \rightarrow x = y \\ \forall x, y, z, (x \leq y \wedge y \leq z) \rightarrow x \leq z \\ \forall x, y, (x \leq y) \vee (y \leq x) \\ \forall x, y, z, x \leq y \rightarrow (x + z \leq y + z) \\ \forall x, y, (x \leq 0 \wedge y \leq 0) \rightarrow x \times y \leq 0 \\ \forall x \exists y, (x = y \times y) \vee (x = -y \times y) \\ \forall x_0, \ldots, x_{n-1}, \exists y, y^n + x_{n-1}y^{n-1} + \ldots x_1 y + x_0 = 0 \text{ pour tout nombre impair } n \end{cases}$$

Proposition 10 *$T_{CC} \models \varphi$ ssi $(\mathbb{R}, =, 0, 1, +, \times) \models \varphi$.*

6.2.5 Arithmétique de Presburger ([DNRC01], p. 136)

$$T_{Pres} \begin{cases} \text{Théorie de l'égalité} \\ \forall x, x^{+1} \neq 0 \\ \forall x, (x = 0) \vee \exists y, x = y^{+1} \\ \forall x, y, x^{+1} = y^{+1} \rightarrow x = y \\ \forall x, x + 0 = x \\ \forall x, y, x + (y^{+1}) = ((x + y)^{+1}) \\ (\varphi(0) \wedge (\forall y, \varphi(y) \rightarrow \varphi(y^{+1}))) \rightarrow \forall x, \varphi(x) \text{ pour toute formule } \varphi(x) \end{cases}$$

Proposition 11 *$T_{Pres} \models \varphi$ ssi $(\mathbb{N}, =, 0, 1, +) \models \varphi$.*

6.2.6 Arithmétique de Peano ([DNRC01], p. 111)

$$PA \begin{cases} \text{Arithmétique de Presburger} \\ \forall x, x \times 0 = x \\ \forall x, y, x \times (y^{+1}) = (x \times y) + x \end{cases}$$

Proposition 12 *$PA \models \varphi$ <u>implique</u> $(\mathbb{N}, =, 0, 1, +, \times) \models \varphi$.*

6.2.7 Arithmétique vraie sur $\mathbb{N}$

Soit $\Sigma = \{0, 1, +, \times\}$.
$T_{\mathbb{N}} = \{\varphi \mid \varphi \text{ est une } \Sigma\text{-formule et } (N, =, 0, 1, +, \times) \models \varphi\}$.

Proposition 13 *$T_{\mathbb{N}} \models \varphi$ ssi $(\mathbb{N}, =, 0, 1, +, \times) \models \varphi$.*

6.3 Résultats

	Théorie complète ?	SAT	SAT d'une formule existentielle	SAT d'une conjonction existentielle
$\emptyset$	✗	indéc, dans co-RE	?	?
Théorie de l'égalité	✗	indéc, dans co-RE	?	?
Théorie de l'égalité (avec que des symboles de fonction)	✗	indéc [? ?]	NP-complet	dans P [KS08]
Théorie de l'égalité (sans autres symboles de fonction et symboles de prédicats) ([DNRC01], p. 132)	✗	PSPACE-complet [SM73]	NP-complet	dans P [KS08]
Théorie des ordres denses ([DNRC01], p. 130) ([Har09], p. 333)	✓	PSPACE-complet	NP-complet	dans P
Théorie des corps clos ([DNRC01], p. 133)	✓	dans EXPSPACE [BOKR86], PSPACE-dur	NP-dur dans PSPACE [Can88]	?
Arithmétique linéaire sur $\mathbb{R}$	✓		NP-complet	dans P
Arithmétique linéaire sur $\mathbb{N}$ (de Presburger) ([DNRC01], p. 136) ([Har09], p. 336) ([Car08])	✓	dans 3EXPTIME [Opp78], 2EXPTIME-dur [FFR74]	NP-complet	NP-complet
Arithmétique de Robinson ([CL93], p. 73)	✗	indéc, dans co-RE	?	?
Arithmétique de Peano ([DNRC01], p. 123)	✗	indéc, dans co-RE	indéc [?]	indéc [?]
Arithmétique sur $\mathbb{N}$ (théorie non récursive)	✓	indéc, ni dans RE, ni dans co-RE	indéc, dans RE	indéc[Mat03], dans RE

Chapitre 7

Déduction naturelle

Points du programme de l'agrégation
... substitutions, capture de variables.
Logique du premier ordre : systèmes formels de preuve. Déduction naturelle. Théorème de complétude du calcul des prédicats du premier ordre.

7.1 Substitution

Définition 44 (substitution ([BA12], p. 187))
Une **substitution** est un ensemble de la forme

$$[x_1 := t_1, \ldots, x_n := t_n]$$

où chaque x_i est une variable et chaque t_i est un terme.

Définition 45 (instantiation)
Étant donnée une substitution $[x_1 := t_1, \ldots, x_n := t_n]$, la formule $\varphi[x_1 := t_1, \ldots, x_n := t_n]$ est la formule φ dans laquelle on a remplacé simultanément les occurrences libres x_i par le terme t_i.

Exemple 12

Attention, on autorise uniquement les **substitutions licites**. Par exemple

$$\underbrace{(\forall x, x < y)}_{\varphi}[y := \overbrace{x - 1}^{t}] \quad = \quad \forall x, x < x - 1$$

n'est pas licite. On parle de **capture de variables**. Sans perte de généralité, on renomme les variables liées de φ par des variables fraîches afin qu'elles n'apparaissent pas dans t ([DNRC01], p. 20-22) :

$$(\forall x', x' < y)[y := x - 1] \quad = \quad \forall x', x' < x - 1$$

7.2 Règles de la déduction naturelle ([DNRC01], p. 25)

Définition 46 (séquent en déduction naturelle ([DNRC01], p. 24))
Un **séquent** est un couple (Γ, φ) où Γ est un ensemble fini de formules et φ une formule. Il se note $\Gamma \vdash \varphi$ et se lit « à partir des hypothèses Γ, j'ai prouvé la formule φ ».

Définition 47 (règles de la déduction naturelle)
Les **règles de la déduction naturelle** sont données ci-dessous :

$$\text{axiome}\frac{}{\Gamma, A \vdash A} \qquad \text{affaiblissement}\frac{\Gamma \vdash A}{\Gamma, B \vdash A} \qquad \text{absurde}\frac{\Gamma, \neg A \vdash \bot}{\Gamma \vdash A}$$

	Règles d'introduction	Règles d'élimination
$\rightarrow$	$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash (A \rightarrow B)}$	$\dfrac{\Gamma \vdash A \quad \Gamma \vdash (A \rightarrow B)}{\Gamma \vdash B}$
$\wedge$	$\dfrac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash (A \wedge B)}$	$\dfrac{\Gamma \vdash (A \wedge B)}{\Gamma \vdash A}$ $\dfrac{\Gamma \vdash (A \wedge B)}{\Gamma \vdash B}$
$\vee$	$\dfrac{\Gamma \vdash A}{\Gamma \vdash (A \vee B)}$ $\dfrac{\Gamma \vdash B}{\Gamma \vdash (A \vee B)}$	$\dfrac{\Gamma \vdash (A \vee B) \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}$
$\neg$	$\dfrac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}$	$\dfrac{\Gamma \vdash \neg A \quad \Gamma \vdash A}{\Gamma \vdash \bot}$
$\exists$	$\dfrac{\Gamma \vdash A[x := t]}{\Gamma \vdash \exists x A}$	$\dfrac{\Gamma \vdash \exists x A \quad \Gamma, A \vdash C}{\Gamma \vdash C}$ où x non libre dans Γ, C
$\forall$	$\dfrac{\Gamma \vdash A}{\Gamma \vdash \forall x A}$ où x non libre dans Γ	$\dfrac{\Gamma \vdash \forall x A(x)}{\Gamma \vdash A[x := t]}$
$=$	$\dfrac{}{\Gamma \vdash (t = t)}$	$\dfrac{\Gamma \vdash \varphi(x := t) \quad \Gamma \vdash (t = u)}{\Gamma \vdash \varphi(x := u)}$

Exemple 13 (règle d'introduction du $\forall$)

$$\dfrac{\dfrac{\cancel{p(x) \vdash p(x)}}{\cancel{p(x) \vdash \forall x, p(x)}}\ \text{!}}{\dfrac{\vdash p(x)}{\vdash \forall x, p(x)}}\ \checkmark$$

Exemple 14 (règle d'élimination du $\exists$)

$$\dfrac{\dfrac{p(x) \vdash \exists x, q(x) \quad \cancel{p(x), q(x) \vdash r(x)}}{\cancel{p(x) \vdash r(x)}}\ \text{!}}{\dfrac{\vdash \exists x, p(x) \quad p(x) \vdash q}{\vdash q}}\ \checkmark$$

7.3 Exemple d'arbre de preuve

Définition 48 (arbre de preuve en déduction naturelle)
Une **(arbre de) preuve (en déduction naturelle)** de $\Gamma \vdash \varphi$ est un arbre fini tel que :
- La racine est $\Gamma \vdash \varphi$;
- Chaque nœud correspond à l'application d'une des règles de la déduction naturelle.

Exemple 15

Définition 49 (existence d'un arbre de preuve ([DNRC01], p. 80))
Soit T une théorie. Soit φ une formule close. On note $T \vdash_{DN} \varphi$ pour dire[1] qu'**il existe un arbre de preuve** de $\Gamma \vdash \varphi$ avec Γ fini et $\Gamma \subseteq T$.

Définition 50 (théorie consistante)
Soit T une théorie. T est **consistante** si $T \nvdash_{DN} \bot$.

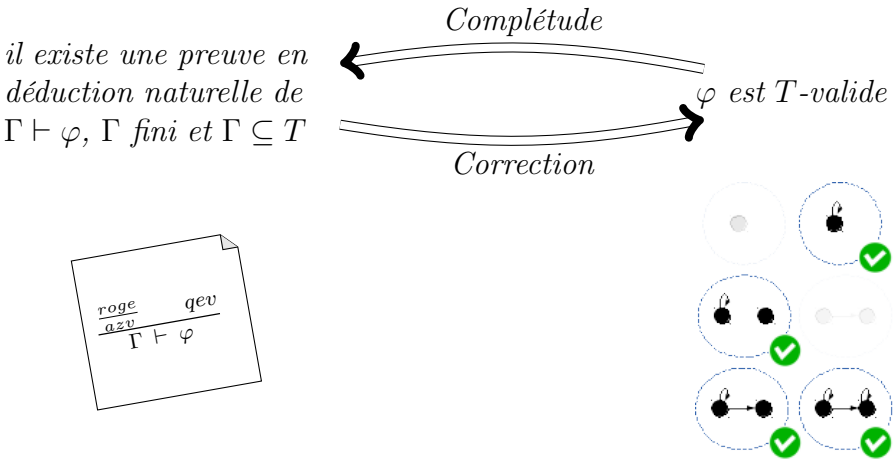
Définition 51 (théorie syntaxiquement complète)
Soit T une théorie. T est **syntaxiquement complète** si
pour toute formule close φ, $T \vdash_{DN} \varphi$ ou $T \vdash_{DN} \neg\varphi$.

1. Certains livres surchargent la notation et écrivent $\vdash$.

7.4 Correction et complétude

7.4.1 Enoncé

Théorème 9 *Soit T un ensemble de formules closes et φ une formule close.*



Idée de la démonstration.

$\boxed{\Rightarrow}$ On démontre, par induction sur π, que pour tout arbre de preuve π, on a

$\mathcal{P}(\pi)$: Si π est une preuve de $\Gamma \vdash \varphi$, alors
pour tout modèle $\mathcal{M}$, pour tout assignation v, $\mathcal{M}, v \models \Gamma$ implique que $\mathcal{M}, v \models \varphi$.

$\boxed{\Leftarrow}$

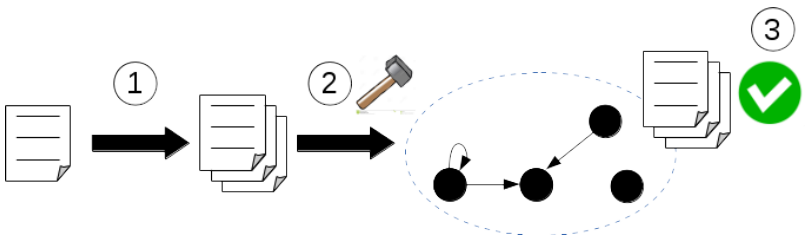
$T \models \varphi$	ssi	$T \cup \{\neg\varphi\}$ insatisfiable	
	implique	$T \cup \{\neg\varphi\}$ inconsistante	(par le lemme 5
	ssi	$T \vdash_{DN} \varphi$	

■

7.4.2 Idée de la démonstration

Lemme 5 *([DNRC01], p. 83) Δ est consistante alors Δ est satisfiable.*

Idée de la démonstration.



1. On étend Δ en Δ' tel que :
   — Pour toute formule $\psi(x)$, on introduit un symbole de constante c_ψ tel que

$$\Delta' \vdash_{DN} (\exists x \psi(x) \to \psi(c_\psi)).$$

   — Δ' soit une théorie syntaxiquement complète ;

2. On construit le modèle $\mathcal{M}$:
   — Le domaine D est l'ensemble des termes clos [2] ;
   — Si f est un symbole de fonction d'arité n,

$$\begin{array}{rlll} f^{\mathcal{M}} : & D^n & \to & D \\ & ([t_1], \ldots, [t_n]) & \mapsto & [f(t_1, \ldots, t_n)] \end{array}$$

   — Si p est un symbole de prédicats n-aire,

$$\begin{array}{rlll} p^{\mathcal{M}} : & D^n & \to & \{0,1\} \\ & ([t_1], \ldots, [t_n]) & \mapsto & 1 \text{ si } \Delta' \vdash_{DN} p(t_1, \ldots, t_n) \\ & & \mapsto & 0 \text{ sinon.} \end{array}$$

3. On montre :

Lemme 6 (lemme de la vérité)
Soit $\varphi(x_1, \ldots, x_n)$ une formule et $t_1, \ldots t_n$ des termes clos. On a :

$$\mathcal{M}[x_1 := [t_1], \ldots, x_n := [t_n]] \models \varphi \text{ ssi } \Delta' \vdash_{DN} \varphi(t_1, \ldots, t_n)$$

7.4.3 Détails

1. Témoins de Henkin. Soit Δ consistante. On considère $(\Sigma_n)_{n\in\mathbb{N}}$ la suite de signature et $(\Delta_n)_{n_{\mathbb{N}}}$ la suite de théories définie par :
   — $\Sigma_0 = \Sigma$;
   — $\Delta_0 = \Delta$;
   et
   — $\Sigma_{n+1} = \Sigma_n \cup \{c_{\psi(x)} \mid \psi(x) \text{ est une } \Sigma_n\text{-formule avec } x \text{ comme seule variable libre}\}$;
   — $\Delta_{n+1} := \Delta_n \cup \{\exists\psi(x) \to \psi(c_{\psi(x)}) \mid \psi(x) \text{ est une } \Sigma_n\text{-formule avec } x \text{ comme seule variable libre}\}$.
   Soit $\Delta'' := \bigcup_n \Delta_n$. Soit $\Sigma'' := \bigcup_n \Sigma_n$.

   Proposition 14 *Δ'' est une Σ''-théorie consistante.*

   Idée de la démonstration.
   On montre par récurrence sur n que Δ_n est consistante. ■

   Obtenir une théorie syntaxiquement complète. Soit $(\varphi_n)_{n\in\mathbb{N}}$ une énumération des Σ''-formules. On construit la suite de théories $(K_n)_{n\in\mathbb{N}}$ définie par :
   — $K_0 = \Delta''$;
   — $K_{n+1} = \begin{cases} K_n \cup \{\varphi_n\} & \text{si } K_n \nvdash_{DN} \varphi_n \text{ et } K_n \nvdash_{DN} \neg\varphi_n \\ K_n. \end{cases}$.
   On pose $\Delta' := \bigcup_{n\in\mathbb{N}} K_n$.

   Proposition 15 *Δ' est une Σ''-théorie :*
   — *consistante ;*
   — *syntaxiquement complète ;*
   — *telle que pour toute Σ''-formule $\psi(x)$, $\Delta' \vdash_{DN} (\exists x\psi(x) \to \psi(c_\psi))$.*

2. Le modèle $\mathcal{M}$ est bien défini.

2. Certaines personnes ([DNRC01], p. 83) quotientent l'ensemble des termes clos avec la relation d'équivalence $\sim$ défini par $t \sim t'$ ssi $\Delta' \vdash (t = t')$ car le symbole de prédicat $=$ s'interprète par l'égalité. Aussi ils ont rajouté les règles correspondantes à $=$.

3. On démontre le lemme 6 par induction sur φ. Soit la propriété

$\mathcal{P}(\varphi)$: Soit $x_1, \ldots, x_n$ les variables libres de φ et soit $t_1, \ldots t_n$ des termes clos. On a : $\mathcal{M}[x_1 := [t_1], \ldots, x_n := [t_n]] \models \varphi$ ssi $\Delta' \vdash_{DN} \varphi(t_1, \ldots, t_n)$

— On a :

$\mathcal{M}[x_1 := [t_1], \ldots, x_n := [t_n]] \models p(u_1, \ldots, u_k)$ ssi $p^{\mathcal{M}}([v_1], \ldots, [v_k]) = 1$
ssi $\Delta' \vdash_{DN} \varphi(v_1, \ldots, v_k)$
ssi $\Delta' \vdash_{DN} \varphi(u_1, \ldots, u_k)[x_1 := t_1, \ldots, x_n := t_n]$
où $v_i = u_i[x_1 := t_1, \ldots, x_n := t_n]$.

d'où $\mathcal{P}(p(v_1, \ldots, v_k))$.

— Supposons $\mathcal{P}(\varphi)$.

$\mathcal{M}[x_1 := [t_1], \ldots] \models \neg\varphi$ ssi $\mathcal{M}[x_1 := [t_1], \ldots] \not\models \varphi$
ssi $\Delta' \not\vdash_{DN} \varphi(\vec{t})$ par $\mathcal{P}(\varphi)$
ssi $\Delta' \vdash_{DN} \neg\varphi(\vec{t})$ car Δ' syntaxiquement complète

D'où $\mathcal{P}(\neg\varphi)$.

— $\vdots$

— Supposons $\mathcal{P}(\varphi)$.

$\mathcal{M}[x_1 := [t_1], \ldots] \models \exists x\varphi$ implique il existe $t \in D$ tq $\mathcal{M}[x_1 := [t_1], \ldots x := [t]] \models \varphi$
ssi $\Delta' \vdash_{DN} \varphi(\vec{t}, t)$ par $\mathcal{P}(\varphi)$
implique $\Delta' \vdash_{DN} \exists x, \varphi(\vec{t})$ car avec la règle $\exists_i$.

$\Delta' \vdash_{DN} \exists x, \varphi(\vec{t})$ implique $\Delta' \vdash_{DN} \varphi(\vec{t}, c_{\varphi(\vec{t},x)})$ car $\Delta' \vdash_{DN} \exists x, \varphi(\vec{t}) \to \varphi(\vec{t}, c_{\varphi(\vec{t},x)})$
$\mathcal{M}[x_1 := [t_1], \ldots, x = c_{\varphi(\vec{t},x)}] \models \varphi$ par $\mathcal{P}(\varphi)$
ssi $\Delta' \vdash_{DN} \varphi(\vec{t}, t)$ par $\mathcal{P}(\varphi)$
implique $\mathcal{M}[x_1 := [t_1], \ldots] \models \exists x\varphi$.

D'où $\mathcal{P}(\exists x, \varphi)$.

■

7.5 Conséquences

7.5.1 Problème de validité dans RE

Théorème 10 *Le problème de validité en logique du premier ordre est dans RE.*

Idée de la démonstration.
Voici une machine qui accepte le langage des formules valides :

```
procédure valide(φ)
    pour toute preuve π
        si π est une preuve de ⊢ φ alors
            accepter
```

■

Théorème 11 *Si T est dans RE, alors le problème de T-validité en logique du premier ordre est dans RE.*

Idée de la démonstration.
Soit $\psi_1, \psi_2, \ldots$ une énumération effective de T. Voici une machine qui accepte le langage des formules T-valides :

```
procédure Tvalide(φ)
    pour tout n = 0, 1, ...
        Énumérer les formules ψ_1, ..., ψ_n
        pour toute preuve π de longueur au plus n
            si π est une preuve de {ψ_1, ..., ψ_n} ⊢ φ alors
                accepter
```

■

7.5.2 Théorème de compacité

Théorème 12 (Compacité de la logique du premier ordre ([DNRC01], p. 86))
Soit T un ensemble de formules closes.

Si toute partie finie de T est satisfiable alors T satisfiable.

Idée de la démonstration.
Par contraposée.

T insatisfiable **implique** T inconsistante (par le lemme 5)
implique il existe une preuve de $\Gamma \vdash \bot$ avec Γ fini et $\Gamma \subseteq T$
implique il existe Γ fini et $\Gamma \subseteq T$ et Γ insatisfiable (par correction).

■

Corollaire 3 *Il n'existe pas de formule φ de la logique du premier ordre tel que*

$$\mathcal{M} \models \varphi \text{ ssi le domaine de } \mathcal{M} \text{ est fini.}$$

Idée de la démonstration.
Par l'absurde. Soit φ une telle formule et posons

$$T = \{\varphi\} \cup \bigcup_{n\in\mathbb{N}} \{\exists x_1 \dots x_n, \bigwedge_{i,j\in\{1,\dots n\}, i\neq j} x_i \neq x_j\}.$$

Toute partie finie de T est satisfiable mais T est insatisfiable.
Contradiction avec le théorème de compacité. ■

Corollaire 4 (modèle non-standard de l'arithmétique vraie)
$T_{\mathbb{N}} = \{\varphi \mid \varphi$ est une Σ-formule et $(\mathbb{N}, =, 0, 1, +, \times) \models \varphi\}$ admet un modèle non isomorphe[3] *à $(\mathbb{N}, =, 0, 1, +, \times)$.*

Idée de la démonstration.
Soit c un symbole frais de constante. Soit $T = T_{\mathbb{N}} \cup \{i < c \mid i \in \mathbb{N}\}$. Toute partie finie de T est satisfiable. D'après le théorème de compacité, T est satisfiable. ■

7.5.3 Théorème de Lowenheim-Skolem

Théorème 13 (Lowenheim-Skolem ([DNRC01], p. 99) ([BA12] , p. 228))
Si T est satisfiable dans un modèle infini alors T est satisfiable dans un modèle infini dénombrable.

Idée de la démonstration.
On pose $T' = T \cup \{c_i \neq c_j \mid i, j \in \mathbb{N}, i \neq j\}$ où $c_0, c_1, \dots$ sont des nouveaux symboles de constante.

T' satisfiable implique T' consistante
implique T' est satisfiable dans le modèle à l'étape 2 de la démonstration du lemme 5.

■

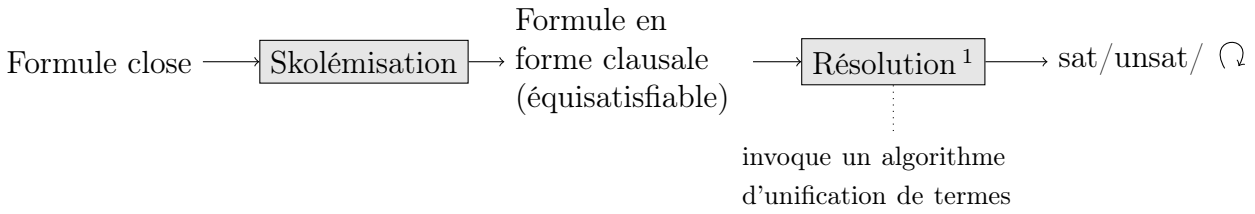
Corollaire 5 (modèle dénombrable pour la théorie des corps-clos)
La théorie des corps clos admet un modèle infini dénombrable.

3. que l'on appelle non-standard.

Chapitre 8

Résolution en logique du premier ordre

Points du programme de l'agrégation
Algorithme d'unification des termes. Preuves par résolution.



8.1 Skolémisation ([BA12], p. 174)

Définition 52 (formule prénexe)
Une **formule prénexe** est une formule close de la formule de la forme $Q_1 x_1 \dots Q_n x_n \psi$ où chaque Q_i est un quantificateur et ψ est sans quantificateur.

Définition 53 (forme normale prénexe universelle)
Une **forme prénexe universelle** est une formule close qui est de la forme $\forall_1 x_1 \dots \forall_n x_n \psi$ où ψ est sans quantificateur.

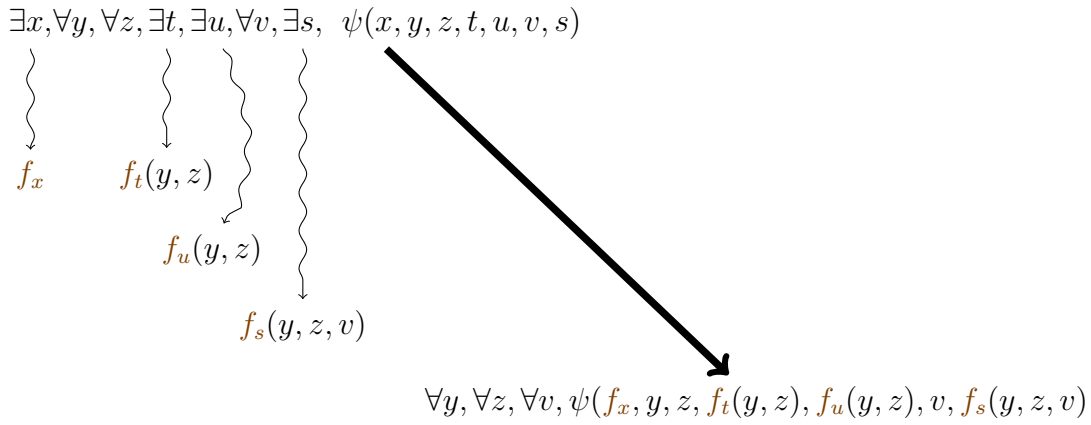
Définition 54 (littéral)
Un **littéral** est une formule atomique ou la négation d'une formule atomique.

Définition 55 (forme normale conjonctive)
Par abus, on dit qu'une formule est une **forme normale conjonctive** si elle est de la forme $\forall x_1 \dots \forall x_n \psi$ où ψ est une formule sans quantificateur qui est une forme normale conjonctive, c'est-à-dire une conjonction de disjonctions de littéraux.

1. Attention, ce n'est pas un algorithme.

Définition 56 (fonctions de Skolem)
Soit φ une formule prénexe.

On appelle **fonction de Skolem** un symbole de fonction f_x d'arité n pour un quantificateur $\exists x$ qui dépend des n variables quantifiées universellement qui précède $\exists x$ dans φ.



Théorème 14 (de Skolem ([BA12], p. 172)) *Pour toute formule φ, il existe une forme normale conjonctive φ' on a :*

$$\varphi \text{ satisfiable ssi } \varphi' \text{ satisfiable.}$$

Idée de la démonstration.

1. Renommer les variables liées afin que chaque quantification porte sur une variable différente.
2. Éliminer les opérateurs booléens autre que $\wedge$ et $\vee$
3. Mettre les négations à l'intérieur cf. loi de Morgan
4. Mettre en forme prénexe
5. Mettre la matrice en FNC
6. Introduire[2] des **fonctions de Skolem** pour chaque $\exists x$.

■

Exemple 16 ([BA12], p. 174))

$$[\forall x(p(x) \to q(x))] \to [(\forall x p(x)) \to (\forall x q(x))]$$

2. Étape qui fait que l'on obtient une formule équisatisfiable et pas forcément équivalente.

8.2 Résolution en logique du premier ordre

8.2.1 Unification ([LDR93], p. 86) ([BA12], p. 189)

Définition 57 (unificateur)
Soit $\{\psi_1, \ldots, \psi_i\}$ un ensemble de formules atomiques. Un **unificateur** de $\{\psi_1, \ldots, \psi_i\}$ est une substitution σ telle que $\psi_1\sigma = \ldots = \psi_k\sigma$.

Définition 58 (unificateur principal)
Soit $\{\psi_1, \ldots, \psi_i\}$ un ensemble de formules atomiques. Un **unificateur principal** [3]de $\{\psi_1, \ldots, \psi_i\}$ est un unificateur de cet ensemble tel que pour tout unificateur α, il existe une substitution β telle que $\alpha = \sigma\beta$.

Exemple 17
$\sigma := [x := f(a), z := y]$ est une unificateur principal de $\{p(f(x), g(y)), p(f(f(a)), g(z))\}$. $\alpha := [x := f(a), y := f(g(a)), z := f(g(a))]$ en est un unificateur et s'écrit $\sigma[y := f(g(a))]$.

8.2.2 Règle de résolution ([DNRC01], p. 265)

Définition 59 (règle de résolution)
La règle de résolution en logique du premier ordre est

$$\text{res}\ \frac{(\ \psi \vee \ell_1 \vee \ldots \ell_n\ ) \qquad (\ \neg\psi' \vee \ell'_1 \vee \cdots \vee \ell'_k\ )}{(\ \ell_1\sigma \vee \ldots \ell_n\sigma \vee \ell'_1\sigma \vee \cdots \vee \ell'_k\sigma\ )}$$

où :
— $\ell_1, \ldots, \ell_n, \ldots \ell'_1, \ldots \ell'_k$ sont des littéraux ;
— ψ, ψ' sont des formules atomiques ;
— les clauses $(\ \psi \vee \ell_1 \vee \ldots \ell_n\ )$ et $(\ \neg\psi' \vee \ell'_1 \vee \cdots \vee \ell'_k\ )$ n'ont pas de variables communes ;
— ψ et ψ' sont unifiables et σ en est un unificateur principal.

8.2.3 Contraction ([DNRC01], p. 265)

Définition 60 (règle de contraction)
La règle de contraction est

$$\text{contr}\ \frac{(\ \psi \vee \psi' \vee \ell_1 \vee \ldots \ell_n\ )}{(\ \psi\sigma \vee \ell_1\sigma \vee \ldots \ell_n\sigma\ )}$$

où :
— $L, L', \ell_1, \ldots, \ell_n$ sont des littéraux ;
— ψ, ψ' sont des formules atomiques ;
— ψ et ψ' sont unifiables et σ en est un unificateur principal.

Remarque 2 *Certains auteurs ([BA12], p. 196) ([LDR93], p. 93) présentent une seule règle de résolution qui fusionne la règle de résolution ci-dessus et la règle de contraction :*

$$\textit{super-res}\ \frac{(\ \psi_1 \vee \cdots \vee \psi_i \vee \ell_1 \vee \ldots \ell_n\ ) \qquad (\ \neg\psi'_1 \vee \cdots \vee \neg\psi'_j \vee \ell'_1 \vee \cdots \vee \ell'_k\ )}{(\ \ell_1\sigma \vee \ldots \ell_n\sigma \vee \ell'_1\sigma \vee \cdots \vee \ell'_k\sigma\ )}$$

si $\{\psi_1, \ldots, \psi_i, \psi'_1, \ldots \psi'_j\}$ sont unifiables et où σ en est un unificateur principal.

3. most general unifier

8.2.4 Exemples de preuve par résolution de $\bot$

Exemple 18 (Exemple de preuve de $\bot$ pour $(p(x) \vee p(y)) \wedge (\neg p(x) \vee \neg p(y))$)

$$res\dfrac{contr\ \dfrac{p(x) \vee p(y)}{p(x)} \qquad contr\ \dfrac{\neg p(x) \vee \neg p(y)}{\neg p(x')}}{\bot}$$

Exemple 19 (Exemple de preuve de $\bot$ pour $p(f(x)) \wedge \neg p(x)$)

$$res\dfrac{p(f(x)) \qquad \neg p(x')}{\bot}$$

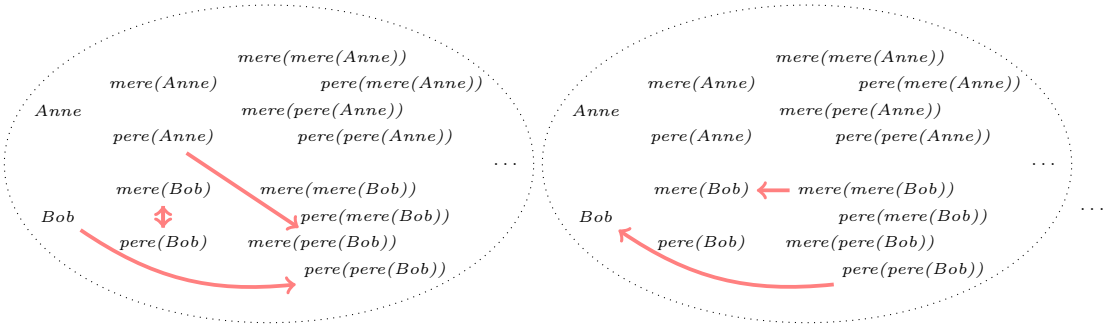
Exemple 20

8.3 Modèles de Herbrand

Définition 61 (modèles de Herbrand)
Un **modèle de Herbrand** est un modèle $\mathcal{H} = (D, .^{\mathcal{H}})$ tel que :
— D = ensemble des termes clos ; (on suppose que l'on a toujours au moins un symbole de constante)
— Pour tout symbole de fonction f d'arité n, on a :

$$\begin{array}{rccl} f^{\mathcal{H}} : & D^n & \to & D \\ & \vec{t} & \mapsto & f(\vec{t}). \end{array}$$

Exemple 21 (quelques modèles de Herbrand)



Proposition 16 ([LDR93], p. 99) *Soit φ une formule en forme prénexe universelle.*

φ satisfiable ssi φ satisfiable dans un modèle de Herbrand.

Idée de la démonstration.
($\Leftarrow$) Immédiat. ($\Rightarrow$) Soit $\mathcal{M}$ un modèle tel que $\mathcal{M} \models \varphi$. On construit $\mathcal{H}$ avec pour tout symbole de prédicat p d'arité n, on a :

$$\begin{array}{rccl} p^{\mathcal{H}} : & D^n & \to & \{0,1\} \\ & \vec{t} & \mapsto & \begin{cases} 1 & \text{si } \mathcal{M} \models p(\vec{t}) \\ 0 & \text{sinon} \end{cases} \end{array}$$

On démontre alors, par induction sur toute formule sans quantificateur ψ, que la propriété suivante est vraie :

$$\mathcal{P}(\psi) : \mathcal{H}, [\vec{x} := \vec{t}] \models \psi \text{ ssi } \mathcal{M} \models \psi[\vec{x} := \vec{t}]$$

■

Définition 62 (théorie universelle)
Une théorie T est **universelle** si toutes les formules de T sont en forme prénexe universelle.

Notation 3 *Soit T une théorie universelle. On note $Inst(T)$ l'ensemble des instances des formules de T obtenues par substitution de termes clos.*

Théorème 15 ([LDR93], p. 99) *Soit T une théorie universelle.*
T est insatisfiable ssi il existe $T' \subseteq Inst(T)$, T' fini tel que T' insatisfiable en logique propositionnelle.

Idée de la démonstration.
($\Leftarrow$) Immédiat.

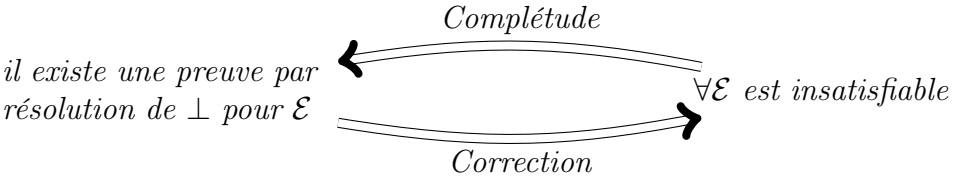
($\Rightarrow$)
T insatisfiable implique T n'a pas de modèle de Herbrand
implique $Inst(T)$ insatisfiable en logique prop.
implique qu'il existe $T' \subseteq Inst(T)$, T' fini, insatisfiable en logique prop.
d'après le **théorème de compacité de la logique prop.**

■

8.4 Correction et complétude

Notation 4 (clotûre universelle) *On note $\forall\varphi$ la clotûre universelle de φ.*

Théorème 16 *Soit $\mathcal{E}$ une forme normale conjonctive.*



Idée de la démonstration.

$\boxed{\Rightarrow}$([CRK03], p. 270) ([BA12], p. 198) ([LDR93], p. 94-95)
Par contraposée, soit $\mathcal{M} \models \forall\mathcal{E}$. La démonstration est similaire au cas propositionnel :

Lemme 7 *si $\mathcal{M} \models \forall C_1$, $\mathcal{M} \models \forall C_2$ et* res$\dfrac{C_1 \quad C_2}{C}$ *alors $\mathcal{M} \models \forall C$.*

Lemme 8 *si $\mathcal{M} \models \forall C_1$ et* contr$\dfrac{C_1}{C}$ *alors $\mathcal{M} \models \forall C$.*

$\boxed{\Leftarrow}$([CRK03], p. 272) ([BA12], p. 199)
Supposons $\forall\mathcal{E}$ insatisfiable. D'après le théorème 15, il existe $\mathcal{E}' \subseteq Inst(\mathcal{E})$, $\mathcal{E}'$ ensemble fini de clauses, insatisfiable en logique propositionnelle. Par **complétude de la résolution en logique propositionnelle**, il existe une preuve par résolution de $\bot$ pour $\mathcal{E}'$ en logique propositionnelle.

Preuve de $\bot$ en résolution propositionnelle pour $\mathcal{E}'$ — Preuve de $\bot$ en résolution du premier ordre pour $\mathcal{E}$

On démontre la propriété suivante par induction sur une preuve par résolution propositionnelle π' :

$\mathcal{P}(\pi')$: « Si π' est une preuve par résolution propositionnelle de C' pour $\mathcal{E}'$ alors il existe une clause C, une preuve par résolution du premier ordre de C pour $\mathcal{E}$, une substitution σ tel que $C' = \sigma C$ ».

— Cas de base. Si $C' \in \mathcal{E}'$, alors par définition il existe $C \in \mathcal{E}$, σ tel que $C' = \sigma C$.
D'où $\mathcal{P}(C')$.

— Cas inductif. Supposons $\mathcal{P}(C'_1)$ et $\mathcal{P}(C'_2)$. On montre $\mathcal{P}(\text{res-prop}\dfrac{\vdots \quad \vdots}{C'})$ grâce à :
Lemme 9 *Soit C_1 et C_2 deux clauses sans variables communes.*
Soit σ_1 et σ_2 deux substitutions.
Si res-prop$\dfrac{\sigma_1 C_1 \quad \sigma_2 C_2}{C'}$ *alors il existe une substitution β et une clause C tel que* super-res$\dfrac{C_1 \quad C_2}{C}$ *et $C' = \beta C$.*

■

Chapitre 9

Calcul des séquents

Points du programme de l'agrégation
Calcul des séquents.

9.1 Motivation

— Règles du calcul des séquents plus symétriques que celles de la déduction naturelle ([DNRC01], p. 58).

éliminable ↓

— La créativité humaine est confinée dans les substitutions et la règle de coupure :
  — Propriété de la sous-formule. D'où la décidabilité de la logique propositionnelle ;
  — Calcul effectif d'un interpolant à partir d'un arbre de preuve ([DNRC01], p. 216).

9.2 Règles du calcul des séquents ([DNRC01], p. 187)

Définition 63 (séquent en calcul des séquents ([DNRC01], p. 186))
Un **séquent** est un couple (Γ, Δ) où Γ et Δ sont des multi-ensembles[1] finis de formules.
Il se note $\Gamma \vdash \Delta$ et se lit

« à partir de la conjonction des hypothèses Γ, j'ai prouvé la disjonction des formules de Δ ».

1. Il existe aussi des présentations avec des ensembles, cf. [BA12], p. 70.

Définition 64 (règles du calcul des séquents)
Les **règles du calcul des séquents** sont données ci-dessous :

$$\bot_g \frac{}{\bot \vdash} \qquad \text{axiome} \frac{}{A \vdash A}$$

	Règles d'introduction à gauche	Règles d'introduction à droite
affaiblissement	$\dfrac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta}$	$\dfrac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, A}$
contraction	$\dfrac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta}$	$\dfrac{\Gamma \vdash \Delta, A, A}{\Gamma \vdash \Delta, A}$
$\rightarrow$	$\dfrac{\Gamma \vdash A, \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, (A \rightarrow B) \vdash \Delta}$	$\dfrac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash (A \rightarrow B), \Delta}$
$\wedge$	$\dfrac{\Gamma, A, B \vdash \Delta}{\Gamma, (A \wedge B) \vdash \Delta}$	$\dfrac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash (A \wedge B), \Delta}$
$\vee$	$\dfrac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, (A \vee B) \vdash \Delta}$	$\dfrac{\Gamma, \vdash \Delta, A, B}{\Gamma \vdash \Delta, (A \vee B)}$
$\neg$	$\dfrac{\Gamma \vdash \Delta, A}{\Gamma, \neg A \vdash \Delta}$	$\dfrac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}$
$\exists$	$\dfrac{\Gamma, A \vdash \Delta}{\Gamma, \exists x A \vdash \Delta}$ où x non libre dans Γ, Δ	$\dfrac{\Gamma \vdash A[x := t], \Delta}{\Gamma \vdash \exists x A, \Delta}$
$\forall$	$\dfrac{\Gamma, A[x := t] \vdash \Delta}{\Gamma, \forall x A \vdash \Delta}$	$\dfrac{\Gamma \vdash A, \Delta}{\Gamma \vdash \forall x A, \Delta}$ où x non libre dans Γ, Δ

éliminable

$$\text{coupure} \frac{\Gamma \vdash A, \Delta \qquad \Gamma', A \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}$$

Exemple 22

9.3 Correction et complétude

Théorème 17

$\Gamma \vdash \varphi$ *est prouvable en déduction naturelle* *ssi* $\Gamma \vdash \varphi$ *est prouvable en calcul des séquents.*

Idée de la démonstration.

($\Rightarrow$) ([DNRC01], p. 195) On transforme un arbre de preuve en déduction naturelle

$$\dfrac{\dfrac{}{\neg\neg A, \neg A \vdash \neg\neg A}\,\text{ax} \qquad \dfrac{}{\neg\neg A, \neg A \vdash \neg A}\,\text{ax}}{\dfrac{\neg\neg A, \neg A \vdash \bot}{\neg\neg A \vdash A}\,\textit{absurde}}\,\neg_e$$

en un arbre de preuve en calcul des séquents :

$$\dfrac{\dfrac{}{\neg\neg A, \neg A \vdash \neg\neg A}\,\text{ax} \qquad \dfrac{\dfrac{}{\neg\neg A, \neg A \vdash \neg A}\,\text{ax}}{\neg\neg A \vdash}\,\neg_e}{\dfrac{\neg\neg A, \neg A \vdash \bot \qquad \dfrac{}{\bot \vdash}\,\bot_g}{\neg\neg A, \neg A \vdash}\,\textit{coupure} \qquad \dfrac{\dfrac{}{A \vdash A}\,\text{ax}}{\vdash \neg A, A}\,\neg_r}{\neg\neg A \vdash A}\,\textit{coupure}$$

($\Leftarrow$) Même idée, voir ([DNRC01], p. 197)

■

9.4 Élimination de la règle de la coupure

Théorème 18

$\Gamma \vdash \varphi$ *est prouvable en calcul des séquents* *ssi* $\Gamma \vdash \varphi$ *est prouvable en calcul des séquents sans utiliser la règle de la coupure.*

Idée de la démonstration.
($\Leftarrow$) Immédiat.
($\Rightarrow$) Voir ([DNRC01], p. 200-208)

■

Chapitre 10

Quizz

Est-ce que p est une formule satisfiable ?	
Est-ce que p est une formule valide ?	
Est-ce que $p \vee \neg p$ est une formule satisfiable ?	
Est-ce que $p \vee \neg p$ est une formule valide ?	
Complexité de SAT ?	
Complexité de la satisfiabilité d'une formule de Horn ?	
Complexité de 2-SAT ?	
Complexité de 3-SAT ?	
Complexité de la validité d'une formule de la logique propositionnelle ?	
Donner un nom d'un algorithme connu pour SAT.	
Donner un exemple d'une forme normale conjonctive.	
Donner un exemple d'une forme normale disjonctive.	
Donner un exemple d'application de la règle de résolution en logique propositionnelle.	

Est-ce que $\exists x, p(x)$ et $p(c)$ sont des formules équivalentes ?	
Est-ce que $\exists x, p(x)$ est satisfiable ?	
Est-ce que $\exists x, p(x)$ est satisfiable dans un modèle fini ?	
Est-ce que $\exists x, p(x)$ est valide ?	
Décidabilité du model checking d'une formule de la logique du premier ordre dans un modèle fini ?	
Classe de la validité d'une formule de la logique du premier ordre ? (RE, co-RE)	
Classe de la satisfiabilité d'une formule de la logique du premier ordre ? (RE, co-RE)	
Classe de la validité d'une formule de la logique du premier ordre sur la classe des modèles finis ? (RE, co-RE)	
Classe de la satisfiabilité d'une formule de la logique du premier ordre dans un modèle fini ? (RE, co-RE)	
Est-ce que le théorème de compacité est toujours vrai si on remplace 'satisfiable' par 'satisfiable dans un modèle fini' ?	
Existe-t-il une formule φ tel que $\mathcal{M} \models \varphi$ implique que le domaine de $\mathcal{M}$ est infini non dénombrable ?	
Donner un exemple de théorie indécidable.	
Donner un exemple de théorie décidable.	
Comment se lit le séquent $A, B \vdash C, D$ (en calcul des séquents) ?	

Bibliographie

[BA12] Mordechai Ben-Ari. *Mathematical logic for computer science*. Springer Science & Business Media, 2012.

[BOKR86] Michael Ben-Or, Dexter Kozen, and John Reif. The complexity of elementary algebra and geometry. *Journal of Computer and System Sciences*, 32(2) :251–264, 1986.

[Can88] John Canny. Some algebraic and geometric computations in pspace. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 460–467. ACM, 1988.

[Car08] Olivier Carton. *Langages formels, calculabilité et complexité*, volume 101. Vuibert, 2008.

[CL93] René Cori and Daniel Lascar. Logique mathématique ii. fonctions récursives, théorème de gödel, théorie des ensembles, théorie des modèles, 1993.

[CRK03] Lascar D CORI R and JL Krivine. Logique mathématique, tome 1 : Calcul propositionnel, algèbre de boole, calcul des prédicats, coll. *Sciences Sup, Dunod*, 2003.

[DNRC01] René David, Karim Nour, Christophe Raffalli, and Pierre-Louis Curien. *Introduction à la logique : théorie de la démonstration : cours et exercices corrigés*. Dunod, 2001.

[DPV16] Sanjoy Dasgupta, Christos H Papadimitriou, and Umesh Virkumar Vazirani. Algorithms. 2016.

[EIS75] Shimon Even, Alon Itai, and Adi Shamir. On the complexity of time table and multi-commodity flow problems. In *Foundations of Computer Science, 1975., 16th Annual Symposium on*, pages 184–193. IEEE, 1975.

[FFR74] Michael Jo Fischer, Michael J Fischer, and Michael O Rabin. Super-exponential complexity of presburger arithmetic. 1974.

[GLM01] Jean Goubault-Larrecq and Ian Mackie. *Proof theory and automated deduction*, volume 6. Springer Science & Business Media, 2001.

[Har09] John Harrison. *Handbook of practical logic and automated reasoning*. Cambridge University Press, 2009.

[KS08] Daniel Kroening and Ofer Strichman. *Decision procedures : an algorithmic point of view*. Springer Science & Business Media, 2008.

[Lal90] René Lalement. Logique, réduction, résolution, 1990.

[LDR93] Richard Lassaigne and Michel De Rougemont. Logique et fondements de l'informatique. *Hermes, Paris*, 1993.

[LS15] Romain Legendre and François Schwarzentruber. *Compilation : Analyse lexicale et syntaxique du texte à sa structure en informatique*. Reference Sciences. Ellipses, 2015.

[Mat03] Ju V Matijasevič. Enumerable sets are diophantine. In *Mathematical Logic In The 20th Century*, pages 269–273. 2003.

[Opp78] Derek C Oppen. A 2 22pn upper bound on the complexity of presburger arithmetic. *Journal of Computer and System Sciences*, 16(3) :323–332, 1978.

[SM73] Larry J Stockmeyer and Albert R Meyer. Word problems requiring exponential time (preliminary report). In *Proceedings of the fifth annual ACM symposium on Theory of computing*, pages 1–9. ACM, 1973.