L o g i q u e

Résumé des épisodes précédents

La logique des prédicats

La notion de théorie

La notion de modèle

Les théorèmes de correction et de complétude

A démontrable dans $\mathcal{T}$
si et seulement si
A valide dans tous les modèles de $\mathcal{T}$

Les théorèmes d'indécidabilité et d'incomplétude

La question de la décidabilité

Deux manières de répondre à des questions

Est-ce que 4 est pair ?

- Trouver une démonstration dans l'arithmétique de

$$\exists x \ (4 = 2 \times x)$$

- Effectuer un calcul : appliquer le programme $Rec^1(\circ_1^0(S, Z^0), Rec^2(\circ_1^1(S, Z^1), Z^3))$ à l'entier 4

Parfois les deux méthodes en compétition

$$\exists x \ (x^2 - 5x + 6 = 0)$$

$$450 + 99 = 549$$

La question de la décidabilité

Jusqu'où peut-on aller dans le remplacement du raisonnement par le calcul ?

Un algorithme qui décide si une proposition est démontrable dans l'arithmétique ?

Et la théorie des ensembles ?
Et la logique des prédicats (sans axiomes) ?

Des raisons d'espérer

Le théorème de Presburger

Le théorème de Skolem

Le théorème de Church

Il n'existe pas d'algorithme qui décide si une proposition est démontrable dans l'arithmétique

Il n'existe pas d'algorithme qui décide si une proposition est démontrable dans la logique des prédicats sans axiomes

I. L'idée

Par l'absurde : un tel algorithme existait...
il permettrait de décider la démontrabilité des propositions de la forme

« Le programme f termine en n »

Contradiction

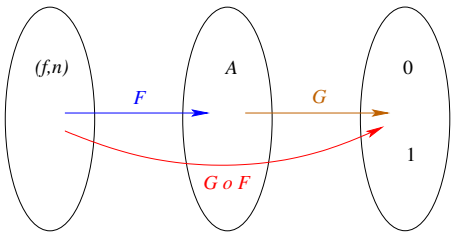
Qu'est-ce que la proposition Le programme f termine en n ?

Une proposition A qui exprime que le programme f termine en n

Une proposition A qui est démontrable ssi le programme f termine en n

La fonction qui associe A à (f, n) doit être calculable

La notion de réduction



On construit F calculable telle que $G \circ F$ ne soit pas calculable
Si G était calculable... donc G n'est pas calculable
Construire un algorithme pour montrer qu'une fonction n'est pas calculable

II. Le théorème de Church

Les fonctions calculables

Ensemble inductivement défini par

- les fonctions $x_1, ..., x_n \mapsto x_i$
- les fonctions $x_1, ..., x_n \mapsto 0$
- la fonction $x \mapsto x + 1$
- l'addition
- la multiplication
- la fonction caractéristique de $\leq$
- la composition
- ~~la définition par récurrence~~
- la minimisation

Programmes

Dérivations étiquetées par les noms des règles des fonctions calculables

$$\frac{\vdots\ S \quad \vdots\ S}{\cdot}\ \circ_1^1$$

En notation linéaire : $\circ_1^1(S, S)$

Un petit langage de programmation fonctionnel sans variables

La représentation des programmes

f programme : on construit une proposition A, dont les variables libres sont parmi $x_1, ..., x_n, y$, telle que
$q = f(p_1, ..., p_n)$ ssi $(\underline{p_1}/x_1, ..., \underline{p_n}/x_n, \underline{q}/y)A$ démontrable dans PA

$$\underline{p} = \underbrace{S(...S}_{p}(0))$$

Notation $A[\underline{p_1}, ..., \underline{p_n}, \underline{q}]$ pour $(\underline{p_1}/x_1, ..., \underline{p_n}/x_n, \underline{q}/y)A$

Sept d'un coup

$f = Z^n$?

$f = S$?

$f = \pi_i^n$?

$f = +$?

$f = \times$?

$f = \chi_{\leq}$?

Composition ?

Sept d'un coup

$f = Z^n : y = 0$

$f = S : y = S(x_1)$

$f = \pi_i^n : y = x_i$

$f = + : y = x_1 + x_2$

$f = \times : y = x_1 \times x_2$

$f = \chi_{\leq} : (x_1 \leq x_2 \wedge y = 1) \vee (x_2 < x_1 \wedge y = 0)$
où $x \leq y$ abréviation de $\exists z\ (x + z = y)$ et $x < y$ de $S(x) \leq y$

Composition : $\exists w_1 ... \exists w_m\ (B_1[x_1, ..., x_n, w_1] \wedge ... \wedge B_m[x_1, ..., x_n, w_m] \wedge C[w_1, ..., w_m, y])$

La minimisation

f construit par minimisation à partir de g
B représente g

$$A = (\forall z\ (z < y \Rightarrow \exists w\ (\neg w = 0 \wedge B[x_1, ..., x_n, z, w]))) \wedge B[x_1, ..., x_n, y, 0]$$

Le théorème de représentation

$q = f(p_1, ..., p_n)$
$A[\underline{p_1}, ..., \underline{p_n}, \underline{q}]$ démontrable dans l'arithmétique
$A[\underline{p_1}, ..., \underline{p_n}, \underline{q}]$ valide dans $\mathbb{N}$

(i) $\Rightarrow$ (ii) : récurrence sur la structure de f
Si $(\underline{0}/x)A$, $(\underline{1}/x)A$, ..., $(\underline{10}/x)A$ démontrables alors $\forall x\ (x \leq 10 \Rightarrow A)$ démontrable
(ii) $\Rightarrow$ (iii) : trivial
(iii) $\Rightarrow$ (i) : si valide dans $\mathbb{N}$ alors il existe des entiers qui...

La proposition le programme f termine en $p_1, ..., p_n$

$$\exists y \ (A[\underline{p_1}, ..., \underline{p_n}, y])$$

f termine en $p_1, ..., p_n$
$\exists y \ A[\underline{p_1}, ..., \underline{p_n}, y]$ démontrable dans l'arithmétique
$\exists y \ A[\underline{p_1}, ..., \underline{p_n}, y]$ valide dans $\mathbb{N}$

Corollaire du théorème de représentation

La fonction F qui associe le numéro de la proposition Le programme f termine en $p_1, ..., p_n$ au numéro de f et à p_1, ..., p_n est calculable

L'ensemble des propositions démontrables dans l'arithmétique n'est pas décidable (CQFD)

L'ensemble des propositions closes démontrables dans l'arithmétique n'est pas décidable

Les extensions de l'arithmétique

Même démonstration : $\mathcal{T}$ extension de l'arithmétique qui a $\mathbb{N}$ comme modèle : démontrabilité dans $\mathcal{T}$ indécidable

Généralisation : $\mathcal{T}$ extension de l'arithmétique cohérente (= qui a un modèle) : démontrabilité dans $\mathcal{T}$ indécidable

Et les extensions incohérentes : par exemple, on ajoute l'axiome $\perp$?

Théories riches dans langages pauvres

On traduit π_2^3 en $y = x_2$, Z^3 en $y = 0$, S en $y = S(x)$

Demande que le langage ait des symboles 0, $=$, S...
Et la théorie des ensembles ?
Une proposition

$$\forall z \ (z \in y \Leftrightarrow (z \in x \vee z = x))$$

mais pas de symbole S
On traduit S, non en $y = S(x_1)$, mais en $Succ[x_1, y]$

Théories riches dans langages pauvres

Langage dans lequel on peut construire des propositions N, $Null$, $Succ$, $Plus$, $Mult$ et Eq
Théorie dans laquelle on peut démontrer ($\mathcal{T}_0$)

$$\forall x \forall y \forall x' \forall y' \ ((N[x] \wedge N[y] \wedge Succ[x, x'] \wedge Succ[y, y'] \wedge Eq[x', y']) \Rightarrow Eq[x, y])$$

...

et qui a un $\mathbb{N}$-modèle

Démontrabilité indécidable

Théories riches (?) dans langages pauvres

Langage dans lequel on peut construire des propositions N, $Null$, $Succ$, $Plus$, $Mult$ et Eq
Qui a un $\mathbb{N}$-modèle
Démontrabilité indécidable

H conjonction des axiomes de $\mathcal{T}_0$
A démontrable dans $\mathcal{T}_0$ ssi $H \Rightarrow A$ démontrable dans la théorie vide

Logique des prédicats sans axiomes

Un symbole de prédicat binaire R
On peut définir N, $Null$, $Succ$, $Plus$, $Mult$ et Eq et construire un $\mathbb{N}$-modèle

Démontrabilité dans la logique des prédicats indécidable

Un prédicat à plusieurs arguments : indécidable
Un prédicat unaire et un symbole de fonction à plusieurs arguments : indécidable

Des théories décidables

La logique des prédicats sans axiomes est indécidable

Mais avec certains axiomes : décidable

- Presburger, Skolem
- $\forall x \forall y \ R(x, y)$
- la géométrie d'Euclide

Une application surprenante

Équation polynomiale
Exemple : $X^7 + X^5 - 2 = 0$ ou $X^2 - 2 = 0$
Peut-on décider si une telle équation a une solution dans $\mathbb{N}$?

$$a_n X^n + a_{n-1} X^{n-1} + ... + a_0 = 0$$

$$1 + (a_{n-1}/a_n)1/X + ... + (a_0/a_n)1/X^n = 0$$

Pour X assez grand, chaque terme $< 1/n$ en valeur absolue : somme non nulle
On énumère et teste tous les entiers inférieurs

Le dixième problème de Hilbert

Généraliser cet algorithme aux équations polynomiales multivariées

Les propositions existentielles

La proposition de l'arithmétique Le programme f termine en n

On peut lui donner la forme $\exists x_1 ... \exists x_n \ (t = u)$

La démontrabilité dans l'arithmétique des propositions de cette forme est indécidable

Termes t et u polynômes en $x_1, ..., x_n$
$\exists x_1 \ ... \ \exists x_n \ (t = u)$ démontrable ssi $t = u$ a une solution

Pas d'algorithme pour les équations diophantiennes multivariées
Théorème de Matiyasevich (1970)

III. Après la pluie, le beau temps : la semi-décidabilité

Les définitions inductives effectives

A ensemble d'arbres

Règles $f_1, f_2...$: fonctions de A dans A

Famille $f_1, f_2...$ effective si l'ensemble R des listes $b, a_1, ..., a_n$ telles qu'il existe f_i telle que $b = f_i(a_1, ..., a_n)$ décidable

Décidabilité de l'ensemble des dérivations

A ensemble d'arbres

$f_1, f_2...$ règles effectives
B sous-ensemble de A inductivement défini par $f_1, f_2...$

L'ensemble des dérivations d'un élément de A est décidable

Nœud étiqueté par b et enfants étiquetés par $a_1, ..., a_n$ on vérifie $a_1, ..., a_n, b$ dans l'ensemble R
Vérification à chaque nœud

Décidabilité de l'ensemble des dérivations

$\mathcal{T}$ ensemble décidable d'axiome

π démonstration de A dans $\mathcal{T}$ si π démonstration de $\Gamma \vdash A$ et $\Gamma \subseteq \mathcal{T}$

π démonstration de A dans $\mathcal{T}$ décidable

Systèmes de vérification de démonstrations

Semi-décidabilité de la démontrabilité

L'ensemble des propositions démontrable dans $\mathcal{T}$ est semi-décidable

$f(x, y) = 1$ si x numéro d'une démonstration dans $\mathcal{T}$ d'une proposition de numéro y et $f(x, y) = 0$ sinon
$g(y)$ plus petit entier x tel que $1 \dot{-} f(x, y) = 0$ composée avec la fonction constante égale à 1

Si y démontrable dans $\mathcal{T}$, alors $g(y) = 1$, sinon g n'est pas définie en y

Énumérer et tester

Plus petit entier x tel que $1 \dot{-} f(x, \ulcorner A \urcorner) = 0$

On énumère tous les entiers, jusqu'à trouver un entier qui soit le numéro d'une démonstration de A

Si A est démontrable, ce numéro finira bien par sortir
sinon la recherche se poursuit à l'infini

Utile pour montrer la semi-décidabilité, sans utilité pratique
Mais l'idée elle-même d'énumération et de test peut être utilisée moins naïvement : démonstration automatique

IV. Chercher simultanément une démonstration de A et de $\neg A$

Plus petit entier x tel que $f(x, \ulcorner A \urcorner) = 1$

Plus petit entier x tel que $f(x, \ulcorner A \urcorner) = 1$ ou $f(x, \ulcorner \neg \urcorner; \ulcorner A \urcorner) = 1$

Les 4 possibilités

1. Si A est démontrable et $\neg A$ n'est pas démontrable

2. Si $\neg A$ est démontrable et A n'est pas démontrable

3. Si ni A ni $\neg A$ ne sont démontrables

4. Si A et $\neg A$ sont toutes les deux démontrables

Les 4 possibilités

1. Si A est démontrable et $\neg A$ n'est pas démontrable
g termine et retourne une démonstration de A

2. Si $\neg A$ est démontrable et A n'est pas démontrable
g termine et retourne une démonstration de $\neg A$

3. Si ni A ni $\neg A$ ne sont démontrables
g ne termine pas

4. Si A et $\neg A$ sont toutes les deux démontrables
g termine et retourne une démonstration de A ou une démonstration de $\neg A$

Les 4 possibilités

1. Si A est démontrable et $\neg A$ n'est pas démontrable
g termine et retourne une démonstration de A

2. Si $\neg A$ est démontrable et A n'est pas démontrable
g termine et retourne une démonstration de $\neg A$

3. Si ni A ni $\neg A$ ne sont démontrables
g ne termine pas

Les 4 possibilités

1. Si A est démontrable et $\neg A$ n'est pas démontrable
g termine et retourne une démonstration de A

2. Si $\neg A$ est démontrable et A n'est pas démontrable
g termine et retourne une démonstration de $\neg A$

Le théorème de Gödel

Il existe une proposition close A telle que ni A ni $\neg A$ ne soit démontrable dans l'arithmétique

La prochaine fois

La démonstration automatique