Logique et Calculabilité
INF551

$$\exists \Rightarrow \forall$$

Dr. Stéphane Lengrand,

Stephane.Lengrand@Polytechnique.edu

Cours 5
Les théorèmes de Church et Gödel

I. Résumé des épisodes précédents

Logique des prédicats

– Syntaxe, Notion de démonstration

Logique des prédicats

- Syntaxe, Notion de démonstration
- Sémantique, Notion de modèle

Logique des prédicats

– Syntaxe, Notion de démonstration
– Sémantique, Notion de modèle

Rappel :

Logique des prédicats

- Syntaxe, Notion de démonstration
- Sémantique, Notion de modèle

Rappel :

- Ce n'est pas parce qu'on a $\mathcal{T} \vdash A \vee \neg A$ (cf tiers exclu) que l'on a soit $\mathcal{T} \vdash A$ soit $\mathcal{T} \vdash \neg A$

Logique des prédicats

– Syntaxe, Notion de démonstration
– Sémantique, Notion de modèle

Rappel :

– Ce n'est pas parce qu'on a $\mathcal{T} \vdash A \vee \neg A$ (cf tiers exclu)
que l'on a soit $\mathcal{T} \vdash A$ soit $\mathcal{T} \vdash \neg A$

– Mais dans tout modèle (bivalué) de $\mathcal{T}$ on a soit $[\![A]\!] = 1$ soit $[\![\neg A]\!] = 1$

Logique des prédicats

- Syntaxe, Notion de démonstration
- Sémantique, Notion de modèle

Rappel :

- Ce n'est pas parce qu'on a $\mathcal{T} \vdash A \vee \neg A$ (cf tiers exclu) que l'on a soit $\mathcal{T} \vdash A$ soit $\mathcal{T} \vdash \neg A$
- Mais dans tout modèle (bivalué) de $\mathcal{T}$ on a soit $[\![A]\!] = 1$ soit $[\![\neg A]\!] = 1$
- Une proposition (close) A est indéterminée dans une théorie $\mathcal{T}$ si ni A ni $\neg A$ ne sont prouvables dans $\mathcal{T}$

- Syntaxe, Notion de démonstration
- Sémantique, Notion de modèle

Rappel :

- Ce n'est pas parce qu'on a $\mathcal{T} \vdash A \vee \neg A$ (cf tiers exclu) que l'on a soit $\mathcal{T} \vdash A$ soit $\mathcal{T} \vdash \neg A$
- Mais dans tout modèle (bivalué) de $\mathcal{T}$ on a soit $[\![A]\!] = 1$ soit $[\![\neg A]\!] = 1$
- Une proposition (close) A est indéterminée dans une théorie $\mathcal{T}$ si ni A ni $\neg A$ ne sont prouvables dans $\mathcal{T}$ (s'il existe à la fois des modèles de $\mathcal{T}$ où $[\![A]\!] = 1$ et d'autres où $[\![\neg A]\!] = 1$)

- Syntaxe, Notion de démonstration
- Sémantique, Notion de modèle

Rappel :

- Ce n'est pas parce qu'on a $\mathcal{T} \vdash A \vee \neg A$ (cf tiers exclu) que l'on a soit $\mathcal{T} \vdash A$ soit $\mathcal{T} \vdash \neg A$
- Mais dans tout modèle (bivalué) de $\mathcal{T}$ on a soit $[\![A]\!] = 1$ soit $[\![\neg A]\!] = 1$
- Une proposition (close) A est indéterminée dans une théorie $\mathcal{T}$ si ni A ni $\neg A$ ne sont prouvables dans $\mathcal{T}$ (s'il existe à la fois des modèles de $\mathcal{T}$ où $[\![A]\!] = 1$ et d'autres où $[\![\neg A]\!] = 1$)
- Théorème de complétion : Toute théorie cohérente peut être complétée en une théorie cohérente où toute proposition close est déterminée

Questions de décidabilité

Def : Un sous-ensemble de $\mathbb{N}^n$ (ou d'un ensemble $\mathcal{C}$ équipé d'une injection vers $\mathbb{N}$) est décidable si sa fonction caractéristique est calculable.

Questions de décidabilité

Def : Un sous-ensemble de $\mathbb{N}^n$ (ou d'un ensemble $\mathcal{C}$ équipé d'une injection vers $\mathbb{N}$) est décidable si sa fonction caractéristique est calculable.

Def : La dérivation qui justifie qu'une fonction de $\mathbb{N}^n$ dans $\mathbb{N}$ est calculable est appelé programme. Ce programme calcule ladite fonction.

Questions de décidabilité

Def : Un sous-ensemble de $\mathbb{N}^n$ (ou d'un ensemble $\mathcal{C}$ équipé d'une injection vers $\mathbb{N}$) est décidable si sa fonction caractéristique est calculable.

Def : La dérivation qui justifie qu'une fonction de $\mathbb{N}^n$ dans $\mathbb{N}$ est calculable est appelé programme. Ce programme calcule ladite fonction.

Il peut être représentée par un arbre (2-articulé) étiqueté par les symboles $\pi_i^n, Z^n, Succ, \circ_m^n, \mu^n$ and Rec^n or $+, \times, \chi_\leq$

Questions de décidabilité

Def : Un sous-ensemble de $\mathbb{N}^n$ (ou d'un ensemble $\mathcal{C}$ équipé d'une injection vers $\mathbb{N}$) est décidable si sa fonction caractéristique est calculable.

Def : La dérivation qui justifie qu'une fonction de $\mathbb{N}^n$ dans $\mathbb{N}$ est calculable est appelé programme. Ce programme calcule ladite fonction.

Il peut être représentée par un arbre (2-articulé) étiqueté par les symboles
$\pi_i^n, Z^n, Succ, \circ_m^n, \mu^n$ and Rec^n or $+, \times, \chi_\leq$

L'ensemble des programmes vient donc avec une notion de calculabilité

Questions de décidabilité

Def : Un sous-ensemble de $\mathbb{N}^n$ (ou d'un ensemble $\mathcal{C}$ équipé d'une injection vers $\mathbb{N}$) est décidable si sa fonction caractéristique est calculable.

Def : La dérivation qui justifie qu'une fonction de $\mathbb{N}^n$ dans $\mathbb{N}$ est calculable est appelé programme. Ce programme calcule ladite fonction.

Il peut être représentée par un arbre (2-articulé) étiqueté par les symboles $\pi_i^n, Z^n, Succ, \circ_m^n, \mu^n$ and Rec^n or $+, \times, \chi_{\leq}$

L'ensemble des programmes vient donc avec une notion de calculabilité

Def : Un programme f termine en $q \in \mathbb{N}$ si q est dans le domaine de la fonction calculée par f

Questions de décidabilité

Def : Un sous-ensemble de $\mathbb{N}^n$ (ou d'un ensemble $\mathcal{C}$ équipé d'une injection vers $\mathbb{N}$) est décidable si sa fonction caractéristique est calculable.

Def : La dérivation qui justifie qu'une fonction de $\mathbb{N}^n$ dans $\mathbb{N}$ est calculable est appelé programme. Ce programme calcule ladite fonction.

Il peut être représentée par un arbre (2-articulé) étiqueté par les symboles $\pi_i^n, Z^n, Succ, \circ_m^n, \mu^n$ and Rec^n or $+, \times, \chi_\leq$

L'ensemble des programmes vient donc avec une notion de calculabilité

Def : Un programme f termine en $q \in \mathbb{N}$ si q est dans le domaine de la fonction calculée par f

Théorème de l'arrêt : l'ensemble des couples (f, q), où f est un programme et q un entier, tels que f termine en q, est indécidable.

II. Théorème de Church

Deux manières de résoudre des problèmes

Est-ce que 4 est pair ?

Deux manières de résoudre des problèmes

Est-ce que 4 est pair ?

– Trouver une démonstration dans l'arithmétique de la proposition close

$$A = \quad \exists x\,(4 = 2 \times x)$$

Deux manières de résoudre des problèmes

Est-ce que 4 est pair ?

– Trouver une démonstration dans l'arithmétique de la proposition close

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

– 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.

Deux manières de résoudre des problèmes

Est-ce que 4 est pair ?

– Trouver une démonstration dans l'arithmétique de la proposition close

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

– 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.
  C'est-à-dire calculer $g(4)$ avec

$$\begin{array}{lll} g(0) & := 1 & \\ g(n+1) & := 1 & \text{si } g(n) = 0 \\ g(n+1) & := 0 & \text{si } g(n) = 1 \end{array}$$

Deux manières de résoudre des problèmes

Est-ce que 4 est pair ?

– Trouver une démonstration dans l'arithmétique de la proposition close

$$A = \quad \exists x \; (4 = 2 \times x)$$

– 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.
C'est-à-dire calculer $g(4)$ avec

$$\begin{array}{lll} g(0) & := 1 & \\ g(n+1) & := 1 & \text{si } g(n) = 0 \\ g(n+1) & := 0 & \text{si } g(n) = 1 \end{array}$$

On a un algorithme qui décide si A est prouvable dans l'arithmétique.

Jusqu'où peut-on aller ?

Existe-t-il un algorithme générique qui décide si une proposition est prouvable dans l'arithmétique ?

Jusqu'où peut-on aller ?

Existe-t-il un algorithme générique qui décide si une proposition est prouvable dans l'arithmétique ?

Formalisons la question.

Les propositions de l'arithmétique (comme tous les arbres articulés) peuvent se numéroter ($\ulcorner A \urcorner \in \mathbb{N}$)

Jusqu'où peut-on aller ?

Existe-t-il un algorithme générique qui décide si une proposition est prouvable dans l'arithmétique ?

Formalisons la question.

Les propositions de l'arithmétique (comme tous les arbres articulés) peuvent se numéroter ($\ulcorner A \urcorner \in \mathbb{N}$)

La fonction qui à toute proposition close A associe 1 si $\mathcal{PA} \vdash A$ et associe 0 si $\mathcal{PA} \not\vdash A$ est-elle calculable ?

Jusqu'où peut-on aller ?

Existe-t-il un algorithme générique qui décide si une proposition est prouvable dans l'arithmétique ?

Formalisons la question.

Les propositions de l'arithmétique (comme tous les arbres articulés) peuvent se numéroter ($\ulcorner A \urcorner \in \mathbb{N}$)

La fonction qui à toute proposition close A associe 1 si $\mathcal{PA} \vdash A$ et associe 0 si $\mathcal{PA} \not\vdash A$ est-elle calculable ?

Et la théorie des ensembles (même chose avec $ZF \vdash A$) ?

Jusqu'où peut-on aller ?

Existe-t-il un algorithme générique qui décide si une proposition est prouvable dans l'arithmétique ?

Formalisons la question.

Les propositions de l'arithmétique (comme tous les arbres articulés) peuvent se numéroter ($\ulcorner A \urcorner \in \mathbb{N}$)

La fonction qui à toute proposition close A associe 1 si $\mathcal{PA} \vdash A$ et associe 0 si $\mathcal{PA} \not\vdash A$ est-elle calculable ?

Et la théorie des ensembles (même chose avec $ZF \vdash A$) ?

Et la logique des prédicats (même chose avec $\vdash A$) ?

Des raisons d'espérer ?

Le théorème de Presburger :

Des raisons d'espérer ?

Le théorème de Presburger :

La fonction qui à toute proposition close A associe 1 si $\mathcal{P}res \vdash A$ et associe 0 si $\mathcal{P}res \not\vdash A$ est calculable.

Des raisons d'espérer ?...ou pas

Le théorème de Presburger :

La fonction qui à toute proposition close A associe 1 si $\mathcal{P}res \vdash A$ et associe 0 si $\mathcal{P}res \nvdash A$ est calculable.

Le théorème de Church

Il n'existe pas d'algorithme qui décide si une proposition est prouvable dans l'arithmétique
(La fonction qui à toute proposition close A associe 1 si $\mathcal{PA} \vdash A$ et associe 0 si $\mathcal{PA} \nvdash A$ n'est pas calculable.)

Des raisons d'espérer ?. . .ou pas

Le théorème de Presburger :

La fonction qui à toute proposition close A associe 1 si $\mathcal{P}res \vdash A$ et associe 0 si $\mathcal{P}res \not\vdash A$ est calculable.

Le théorème de Church

Il n'existe pas d'algorithme qui décide si une proposition est prouvable dans l'arithmétique
(La fonction qui à toute proposition close A associe 1 si $\mathcal{PA} \vdash A$ et associe 0 si $\mathcal{PA} \not\vdash A$ n'est pas calculable.)

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

Réduire le problème à celui de l'arrêt

L'idée.

Par l'absurde : si un tel algorithme existait. . .

Réduire le problème à celui de l'arrêt

L'idée.

Par l'absurde : si un tel algorithme existait...

... il permettrait de décider la prouvabilité des prop. de la forme

“Le programme f termine en n”

Réduire le problème à celui de l'arrêt

L'idée.

Par l'absurde : si un tel algorithme existait. . .

. . . il permettrait de décider la prouvabilité des prop. de la forme

"Le programme f termine en n"

Or, ceci contredirait le théorème de l'arrêt.

Réduire le problème à celui de l'arrêt

L'idée.

Par l'absurde : si un tel algorithme existait...

... il permettrait de décider la prouvabilité des prop. de la forme

"Le programme f termine en n"

Or, ceci contredirait le théorème de l'arrêt.

Il faut donc exprimer le proposition "Le programme f termine en n" sous la forme d'une proposition arithmétique.

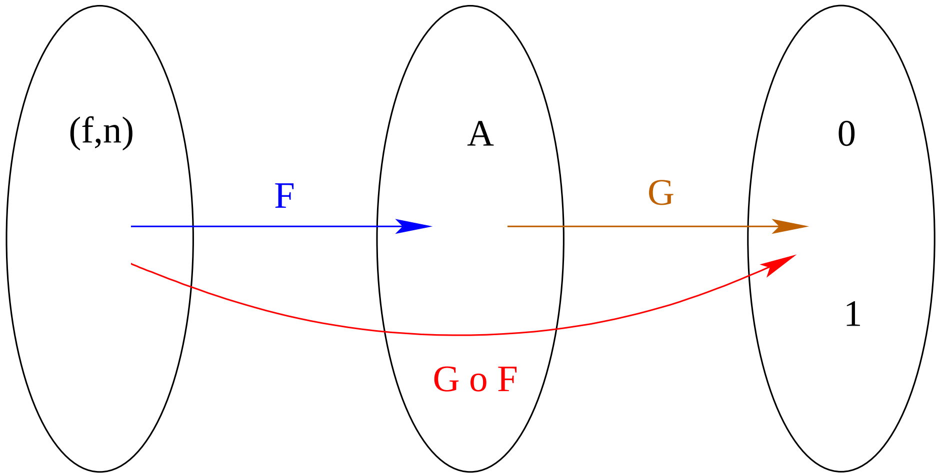
Réduire le problème à celui de l'arrêt

Soit G la fonction qui à toute proposition close A associe 1 si $\mathcal{PA} \vdash A$ et associe 0 si $\mathcal{PA} \not\vdash A$

Réduire le problème à celui de l'arrêt

Soit G la fonction qui à toute proposition close A associe 1 si $\mathcal{PA} \vdash A$ et associe 0 si $\mathcal{PA} \nvdash A$

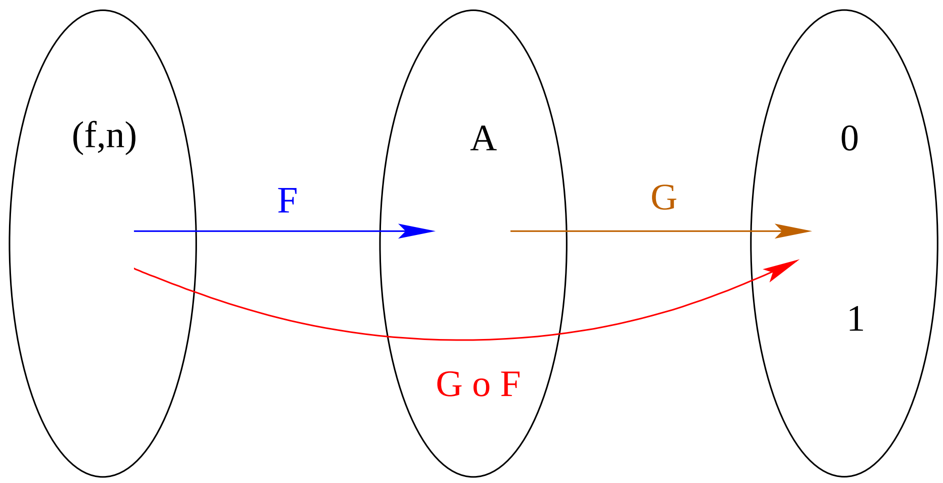
Soit F une fonction calculable qui à tout (f, n) associe une proposition A telle que $\mathcal{PA} \vdash A$ ssi le programme f termine en n



Réduire le problème à celui de l'arrêt

Soit G la fonction qui à toute proposition close A associe 1 si $\mathcal{PA} \vdash A$ et associe 0 si $\mathcal{PA} \nvdash A$

Soit F une fonction calculable qui à tout (f, n) associe une proposition A telle que $\mathcal{PA} \vdash A$ ssi le programme f termine en n



Si G était calculable, $G \circ F$ le serait aussi et déciderait du problème de l'arrêt.

Construisons une telle fonction calculable F

Pour tout programme f de $\mathbb{N}^n$ dans $\mathbb{N}$, on construit

une proposition A, dont les variables libres sont parmi $x_1, \ldots, x_n, y$

telle que

$$f(p_1, ..., p_n) = q$$

ssi

$$\mathcal{PA} \vdash (\underline{p_1}/x_1, ..., \underline{p_n}/x_n, \underline{q}/y)A$$

Construisons une telle fonction calculable F

Pour tout programme f de $\mathbb{N}^n$ dans $\mathbb{N}$, on construit

une proposition A, dont les variables libres sont parmi $x_1, \ldots, x_n, y$

telle que

$$f(p_1, ..., p_n) = q$$

ssi

$$\mathcal{PA} \vdash (\underline{p_1}/x_1, ..., \underline{p_n}/x_n, \underline{q}/y)A$$

$$\text{où } \underline{p} = \overbrace{S(\ldots S}^{p \text{ fois}}(0)\ldots)$$

Construisons une telle fonction calculable F

Pour tout programme f de $\mathbb{N}^n$ dans $\mathbb{N}$, on construit

une proposition A, dont les variables libres sont parmi $x_1, \ldots, x_n, y$

telle que

$$f(p_1, ..., p_n) = q$$

ssi

$$\mathcal{PA} \vdash (\underline{p_1}/x_1, ..., \underline{p_n}/x_n, \underline{q}/y)A$$

où $\underline{p} = \overbrace{S(\ldots S}^{p \text{ fois}}(0)\ldots)$

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

Construisons une telle fonction calculable F

Pour tout programme f de $\mathbb{N}^n$ dans $\mathbb{N}$, on construit

une proposition A, dont les variables libres sont parmi $x_1, \ldots, x_n, y$

telle que

$$f(p_1, ..., p_n) = q$$

ssi

$$\mathcal{PA} \vdash (\underline{p_1}/x_1, ..., \underline{p_n}/x_n, \underline{q}/y)A$$

où $\underline{p} = \overbrace{S(\ldots S}^{p \text{ fois}}(0)\ldots)$

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

On dit que A représente f

Sept d’un coup

$f = Z^n$

$f = S$

$f = \pi_i^n$

$f = +$

$f = \times$

$f = \chi_{\leq}$

$f = \circ_m^n(g, g_1, \ldots, g_m)$

Sept d'un coup

$f = Z^n$ $\qquad y = 0$

$f = S$

$f = \pi_i^n$

$f = +$

$f = \times$

$f = \chi_\leq$

$f = \circ_m^n(g, g_1, \ldots, g_m)$

Sept d'un coup

$f = Z^n$ $\qquad$ $y = 0$

$f = S$ $\qquad$ $y = S(x_1)$

$f = \pi_i^n$

$f = +$

$f = \times$

$f = \chi_\leq$

$f = \circ_m^n(g, g_1, \ldots, g_m)$

Sept d'un coup

$f = Z^n$ $\qquad y = 0$

$f = S$ $\qquad y = S(x_1)$

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

$f = +$

$f = \times$

$f = \chi_{\leq}$

$f = \circ_m^n(g, g_1, \ldots, g_m)$

Sept d'un coup

$f = Z^n$ $\qquad$ $y = 0$

$f = S$ $\qquad$ $y = S(x_1)$

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

$f = +$ $\qquad$ $y = x_1 + x_2$

$f = \times$

$f = \chi_{\leq}$

$f = \circ_m^n(g, g_1, \ldots, g_m)$

Sept d'un coup

$f = Z^n$ $\qquad y = 0$

$f = S$ $\qquad y = S(x_1)$

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

$f = +$ $\qquad y = x_1 + x_2$

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

$f = \chi_{\leq}$

$f = \circ_m^n(g, g_1, \ldots, g_m)$

Sept d'un coup

$f = Z^n$ $\qquad y = 0$

$f = S$ $\qquad y = S(x_1)$

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

$f = +$ $\qquad y = x_1 + x_2$

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

$f = \chi_{\leq}$ $\qquad (y = 1 \wedge \exists z\,(x_2 = x_1 + z)) \vee (y = 0 \wedge \neg\exists z\,(x_2 = x_1 + z))$

$f = \circ_m^n(g, g_1, \ldots, g_m)$

Sept d'un coup

$$f = Z^n \qquad\qquad y = 0$$

$$f = S \qquad\qquad y = S(x_1)$$

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

$$f = + \qquad\qquad y = x_1 + x_2$$

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

$$f = \chi_{\leq} \qquad\qquad (y = 1 \land \exists z\ (x_2 = x_1 + z)) \lor (y = 0 \land \neg\exists z\ (x_2 = x_1 + z))$$

$$f = \circ_m^n(g, g_1, \ldots, g_m)$$

$\exists y_1 \ldots \exists y_m\ B[y_1, \ldots, y_m, y] \land B_1[x_1, \ldots, x_n, y_1] \land \ldots \land B_m[x_1, \ldots, x_n, y_m]$
où $B, B_1, \ldots, B_m$ représentent resp. $g, g_1, \ldots, g_m$

f construite par minimisation de g

Soit B une formule qui représente g

La minimisation

f construite par minimisation de g

Soit B une formule qui représente g

On définit

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

où $x < y$ est $\exists z\ y = x + S(z)$

Le théorème de représentation

Les trois assertions suivantes sont équivalentes :

$$f(p_1, ..., p_n) = q$$

$$\mathcal{PA} \vdash A[\underline{p_1}, ..., \underline{p_n}, \underline{q}]$$

$$A[\underline{p_1}, ..., \underline{p_n}, \underline{q}] \text{ valide dans } \mathbb{N}$$

Le théorème de représentation

Les trois assertions suivantes sont équivalentes :

$$f(p_1, ..., p_n) = q$$

$$\mathcal{PA} \vdash A[\underline{p_1}, ..., \underline{p_n}, \underline{q}]$$

$$A[\underline{p_1}, ..., \underline{p_n}, \underline{q}] \text{ valide dans } \mathbb{N}$$

(i) $\Rightarrow$ (ii) : récurrence sur la construction de f

Nécessite de montrer e.g.

Si $(\underline{p}/x)A$ prouvable alors $\exists x\ A$ prouvable

Si $(\underline{0}/x)A$, ..., $(\underline{p}/x)A$ prouvables alors $\forall x\ (x \leq \underline{p} \Rightarrow A)$ prouvable

Le théorème de représentation

Les trois assertions suivantes sont équivalentes :

$$f(p_1, ..., p_n) = q$$
$$\mathcal{PA} \vdash A[\underline{p_1}, ..., \underline{p_n}, \underline{q}]$$
$$A[\underline{p_1}, ..., \underline{p_n}, \underline{q}] \text{ valide dans } \mathbb{N}$$

(i) $\Rightarrow$ (ii) : récurrence sur la construction de f

Nécessite de montrer e.g.

Si $(\underline{p}/x)A$ prouvable alors $\exists x\ A$ prouvable

Si $(\underline{0}/x)A$, …, $(\underline{p}/x)A$ prouvables alors $\forall x\ (x \leq \underline{p} \Rightarrow A)$ prouvable

(ii) $\Rightarrow$ (iii) : correction du système de preuve

Le théorème de représentation

Les trois assertions suivantes sont équivalentes :

$$f(p_1, ..., p_n) = q$$
$$\mathcal{PA} \vdash A[\underline{p_1}, ..., \underline{p_n}, \underline{q}]$$
$$A[\underline{p_1}, ..., \underline{p_n}, \underline{q}] \text{ valide dans } \mathbb{N}$$

(i) $\Rightarrow$ (ii) : récurrence sur la construction de f

Nécessite de montrer e.g.

Si $(\underline{p}/x)A$ prouvable alors $\exists x\ A$ prouvable

Si $(\underline{0}/x)A$, ..., $(\underline{p}/x)A$ prouvables alors $\forall x\ (x \leq \underline{p} \Rightarrow A)$ prouvable

(ii) $\Rightarrow$ (iii) : correction du système de preuve

(iii) $\Rightarrow$ (i) : si valide dans $\mathbb{N}$ alors il existe des entiers qui...

Corollaires et Théorème de Church

Les trois assertions suivantes sont équivalentes :

$$f \text{ termine en } p_1, ..., p_n$$

$$\mathcal{PA} \vdash \exists y A[\underline{p_1}, ..., \underline{p_n}, y]$$

$$\exists y A[\underline{p_1}, ..., \underline{p_n}, y] \text{ valide dans } \mathbb{N}$$

Corollaires et Théorème de Church

Les trois assertions suivantes sont équivalentes :

$$f \text{ termine en } p_1, ..., p_n$$

$$\mathcal{PA} \vdash \exists y A[\underline{p_1}, ..., \underline{p_n}, y]$$

$$\exists y A[\underline{p_1}, ..., \underline{p_n}, y] \text{ valide dans } \mathbb{N}$$

Notez que la fonction F qui au programme f associe A est calculable

Corollaires et Théorème de Church

Les trois assertions suivantes sont équivalentes :

$$f \text{ termine en } p_1, ..., p_n$$
$$\mathcal{PA} \vdash \exists y A[\underline{p_1}, ..., \underline{p_n}, y]$$
$$\exists y A[\underline{p_1}, ..., \underline{p_n}, y] \text{ valide dans } \mathbb{N}$$

Notez que la fonction F qui au programme f associe A est calculable

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

(CQFD)

III. Les extensions du Théorème de Church

Langages pauvres

Rappel : On a traduit π_2^3 en $y{=}x_2$, Z^3 en $y = 0$, S en $y = S(x)$

Langages pauvres

Rappel : On a traduit π_2^3 en $y{=}x_2$, Z^3 en $y = 0$, S en $y = S(x)$

Ceci nécessite que le langage ait des symboles 0, $=$, S,...

Langages pauvres

Rappel : On a traduit π_2^3 en $y{=}x_2$, Z^3 en $y = 0$, S en $y = S(x)$

Ceci nécessite que le langage ait des symboles 0, $=$, S,...

Et la théorie des ensembles ZF ?

Une proposition $Succ[x, y]$ $\forall z \ (z \in y \Leftrightarrow (z \in x \vee z = x))$

mais pas de symbole S

Langages pauvres

Rappel : On a traduit π^3_2 en $y{=}x_2$, Z^3 en $y = 0$, S en $y = S(x)$

Ceci nécessite que le langage ait des symboles 0, $=$, S,...

Et la théorie des ensembles ZF ?

Une proposition $Succ[x, y]$ $\qquad \forall z\ (z \in y \Leftrightarrow (z \in x \vee z = x))$

mais pas de symbole S

Plus généralement :

Soit $\mathcal{L}_0$ un langage dans lequel on peut construire des propositions
- N, “être un entier”
- $Null$, “être zéro”
- $Succ$, “être le successeur de...”,
- $Plus$, “être la somme de ... et ...”,
- $Mult$, “être la multiplication de ... et ...”
- Eq, “être deux entiers égaux”

Langages pauvres

Par ailleurs, a-t-on besoin de tout $\mathcal{PA}$ pour le théorème de Church ?

(nombre d'axiome infini à cause du schéma de récurrence)

Langages pauvres

Par ailleurs, a-t-on besoin de tout $\mathcal{PA}$ pour le théorème de Church ?

(nombre d'axiome infini à cause du schéma de récurrence)

Def : Soit $\mathcal{T}_0$ la théorie qui (grosso modo) exprime avec N, $Null$, $Succ$, $Plus$, $Mult$, Eq les axiomes de $\mathcal{PA}$ $(+, \times, =)$ mais sans la récurrence (voir poly).

Langages pauvres

Par ailleurs, a-t-on besoin de tout $\mathcal{PA}$ pour le théorème de Church ?

(nombre d'axiome infini à cause du schéma de récurrence)

Def : Soit $\mathcal{T}_0$ la théorie qui (grosso modo) exprime avec N, $Null$, $Succ$, $Plus$, $Mult$, Eq les axiomes de $\mathcal{PA}$ $(+, \times, =)$ mais sans la récurrence (voir poly).

Example :

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

Langages pauvres

Par ailleurs, a-t-on besoin de tout $\mathcal{PA}$ pour le théorème de Church ?

(nombre d'axiome infini à cause du schéma de récurrence)

Def : Soit $\mathcal{T}_0$ la théorie qui (grosso modo) exprime avec N, $Null$, $Succ$, $Plus$, $Mult$, Eq les axiomes de $\mathcal{PA}$ $(+, \times, =)$ mais sans la récurrence (voir poly).

Example :

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

Idée :

$\mathcal{T}_0$ suffisante pour les constructions nécessaires au th. de représentation

Langages pauvres

Par ailleurs, a-t-on besoin de tout $\mathcal{PA}$ pour le théorème de Church ?

(nombre d'axiome infini à cause du schéma de récurrence)

Def : Soit $\mathcal{T}_0$ la théorie qui (grosso modo) exprime avec N, $Null$, $Succ$, $Plus$, $Mult$, Eq les axiomes de $\mathcal{PA}$ $(+, \times, =)$ mais sans la récurrence (voir poly).

Example :

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

Idée :

$\mathcal{T}_0$ suffisante pour les constructions nécessaires au th. de représentation

Def : $\mathbb{N}$-modèle : toute extension de $(\mathbb{N}, 0, (n \mapsto n+1), +, \times, =)$ où $\mathbb{N}$ interprète N, 0 interprète $Null$, $n \mapsto n+1$ interprète $Succ$, $+$ interprète $Plus$, $\times$ interprète $Mult$, $=$ interprète Eq

Théories riches dans langages pauvres

Théorème : Soit $\mathcal{T}$ une théorie dans $\mathcal{L}_0$,

qui a un $\mathbb{N}$-modèle et dans laquelle on peut prouver $\mathcal{T}_0$

La prouvabilité dans cette théorie est indécidable

Théories riches dans langages pauvres

Théorème : Soit $\mathcal{T}$ une théorie dans $\mathcal{L}_0$,

qui a un $\mathbb{N}$-modèle et dans laquelle on peut prouver $\mathcal{T}_0$

La prouvabilité dans cette théorie est indécidable

Preuve :

on adapte la représentation des fonctions calculables en remplaçant
- $(S(t)/x)A$ par $\exists x\ Succ[x, t] \wedge A$
- $(0/x)A$ par $\exists x\ Null[x] \wedge A$
- $t = u$ par $Eq[t, u]$
- $\ldots$

On adapte le théorème de représentation avec $\mathcal{T}$ et son $\mathbb{N}$-modèle à la place de $\mathcal{PA}$ et $\mathbb{N}$. Pour le prouver on utilise le fait que $\mathcal{T}$ prouve $\mathcal{T}_0$.

Théories riches dans langages pauvres

Théorème : Soit $\mathcal{T}$ une théorie dans $\mathcal{L}_0$,

qui a un $\mathbb{N}$-modèle et dans laquelle on peut prouver $\mathcal{T}_0$

La prouvabilité dans cette théorie est indécidable

Preuve :

on adapte la représentation des fonctions calculables en remplaçant

– $(S(t)/x)A$ par $\exists x\ Succ[x,t] \wedge A$
– $(0/x)A$ par $\exists x\ Null[x] \wedge A$
– $t = u$ par $Eq[t,u]$
– $\ldots$

On adapte le théorème de représentation avec $\mathcal{T}$ et son $\mathbb{N}$-modèle à la place de $\mathcal{PA}$ et $\mathbb{N}$. Pour le prouver on utilise le fait que $\mathcal{T}$ prouve $\mathcal{T}_0$.

Application : La prouvabilité dans ZF est indécidable.

Théories riches dans langages pauvres

Théorème : Soit $\mathcal{T}$ une théorie dans $\mathcal{L}_0$,

qui a un $\mathbb{N}$-modèle et dans laquelle on peut prouver $\mathcal{T}_0$

La prouvabilité dans cette théorie est indécidable

Preuve :

on adapte la représentation des fonctions calculables en remplaçant
- $(S(t)/x)A$ par $\exists x\ Succ[x,t] \wedge A$
- $(0/x)A$ par $\exists x\ Null[x] \wedge A$
- $t = u$ par $Eq[t,u]$
- $\ldots$

On adapte le théorème de représentation avec $\mathcal{T}$ et son $\mathbb{N}$-modèle à la place de $\mathcal{PA}$ et $\mathbb{N}$. Pour le prouver on utilise le fait que $\mathcal{T}$ prouve $\mathcal{T}_0$.

Application : La prouvabilité dans ZF est indécidable.

Et les extensions incohérentes ? *e.g.* on ajoute l'axiome $\perp$ à $\mathcal{T}_0$?

Théories pauvres dans langages pauvres

Théorème :

La prouvabilité dans la théorie vide (dans le langage $\mathcal{L}_0$) est indécidable

Théories pauvres dans langages pauvres

Théorème :

La prouvabilité dans la théorie vide (dans le langage $\mathcal{L}_0$) est indécidable

Preuve :

Soit H la conjonction des axiomes de $\mathcal{T}_0$ (le nombre d'axiomes est **fini !**)

A prouvable dans $\mathcal{T}_0$ ssi $H \Rightarrow A$ prouvable dans la théorie vide

Théories pauvres dans langages pauvres

Théorème :

La prouvabilité dans la théorie vide (dans le langage $\mathcal{L}_0$) est indécidable

Preuve :

Soit H la conjonction des axiomes de $\mathcal{T}_0$ (le nombre d'axiomes est **fini !**)

A prouvable dans $\mathcal{T}_0$ ssi $H \Rightarrow A$ prouvable dans la théorie vide

Exemples :

– Langage avec un symbole de prédicat binaire R

Théories pauvres dans langages pauvres

Théorème :

La prouvabilité dans la théorie vide (dans le langage $\mathcal{L}_0$) est indécidable

Preuve :

Soit H la conjonction des axiomes de $\mathcal{T}_0$ (le nombre d'axiomes est **fini !**)

A prouvable dans $\mathcal{T}_0$ ssi $H \Rightarrow A$ prouvable dans la théorie vide

Exemples :

– Langage avec un symbole de prédicat binaire R indécidable

Théories pauvres dans langages pauvres

Théorème :

La prouvabilité dans la théorie vide (dans le langage $\mathcal{L}_0$) est indécidable

Preuve :

Soit H la conjonction des axiomes de $\mathcal{T}_0$ (le nombre d'axiomes est **fini !**)

A prouvable dans $\mathcal{T}_0$ ssi $H \Rightarrow A$ prouvable dans la théorie vide

Exemples :

- Langage avec un symbole de prédicat binaire R indécidable
- Langage avec un symbole de prédicat à plusieurs arguments

Théories pauvres dans langages pauvres

Théorème :

La prouvabilité dans la théorie vide (dans le langage $\mathcal{L}_0$) est indécidable

Preuve :

Soit H la conjonction des axiomes de $\mathcal{T}_0$ (le nombre d'axiomes est **fini !**)

A prouvable dans $\mathcal{T}_0$ ssi $H \Rightarrow A$ prouvable dans la théorie vide

Exemples :

– Langage avec un symbole de prédicat binaire R indécidable

– Langage avec un symbole de prédicat à plusieurs arguments indécidable

Théories pauvres dans langages pauvres

Théorème :

La prouvabilité dans la théorie vide (dans le langage $\mathcal{L}_0$) est indécidable

Preuve :

Soit H la conjonction des axiomes de $\mathcal{T}_0$ (le nombre d'axiomes est **fini !**)

A prouvable dans $\mathcal{T}_0$ ssi $H \Rightarrow A$ prouvable dans la théorie vide

Exemples :

- Langage avec un symbole de prédicat binaire R indécidable
- Langage avec un symbole de prédicat à plusieurs arguments indécidable
- Langage avec un symbole de prédicat unaire et un symbole de fonction à plusieurs arguments

Théories pauvres dans langages pauvres

Théorème :

La prouvabilité dans la théorie vide (dans le langage $\mathcal{L}_0$) est indécidable

Preuve :

Soit H la conjonction des axiomes de $\mathcal{T}_0$ (le nombre d'axiomes est **fini !**)

A prouvable dans $\mathcal{T}_0$ ssi $H \Rightarrow A$ prouvable dans la théorie vide

Exemples :

- Langage avec un symbole de prédicat binaire R indécidable
- Langage avec un symbole de prédicat à plusieurs arguments indécidable
- Langage avec un symbole de prédicat unaire et un symbole de fonction à plusieurs arguments indécidable

Des théories décidables

Le calcul des prédicats sans axiomes est indécidable

Mais si les symboles sont régis par certains axiomes :

On peut récupérer la décidabilité

Exemple : Presburger (arithmétique avec $+$ seulement)

Exemple : la géométrie d'Euclide

Une application surprenante : le 10ème problème de Hilbert

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 $\mathbb{N}$?

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

Une application surprenante : le 10ème problème de Hilbert

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 $\mathbb{N}$?

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

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

Pour X assez grand, chaque terme (sauf 1) est $< 1/n$ en valeur absolue.

Donc la somme est non nulle.

Une application surprenante : le 10ème problème de Hilbert

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 $\mathbb{N}$?

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

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

Pour X assez grand, chaque terme (sauf 1) est $< 1/n$ en valeur absolue.

Donc la somme est non nulle.

On énumère et teste tous les entiers inférieurs à ce X.

Une application surprenante : le 10ème problème de Hilbert

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 $\mathbb{N}$?

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

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

Pour X assez grand, chaque terme (sauf 1) est $< 1/n$ en valeur absolue.

Donc la somme est non nulle.

On énumère et teste tous les entiers inférieurs à ce X.

Le dixième problème de Hilbert :

Peut-on généraliser cet algorithme aux équations polynomiales multivariées ?

Une application surprenante : le 10ème problème de Hilbert

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

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

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

(Théorème de Matiyasevich, 1970)

Une application surprenante : le 10ème problème de Hilbert

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

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

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

(Théorème de Matiyasevich, 1970)

Remarque : t et u sont des polynômes en $x_1, \ldots, x_n$!

$\exists x_1 \ldots \exists x_n \ (t = u)$ est prouvable ssi

$t - u$ est un polynôme multivarié qui admet une racine entière.

Une application surprenante : le 10ème problème de Hilbert

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

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

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

(Théorème de Matiyasevich, 1970)

Remarque : t et u sont des polynômes en $x_1, \ldots, x_n$!

$\exists x_1 \ldots \exists x_n\ (t = u)$ est prouvable ssi

$t - u$ est un polynôme multivarié qui admet une racine entière.

Conséquence : Pas d'algorithme pour les équations polynomiales multivariées !

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

Décidabilité d'une fait d'être une dérivation

Soit E un ensemble (qui s'injecte dans $\mathbb{N}$)

Def : Une famille $f_1, f_2, \ldots$ de règles sur E est dite effective si l'ensemble $\mathcal{R}$ des listes $b, a_1, \ldots, a_n$
t.q. $b = f_i(a_1, \ldots, a_n)$ (pour un certain f_i)
est décidable

Décidabilité d'une fait d'être une dérivation

Soit E un ensemble (qui s'injecte dans $\mathbb{N}$)

Def : Une famille $f_1, f_2, ...$ de règles sur E est dite effective si l'ensemble $\mathcal{R}$ des listes $b, a_1, ..., a_n$
t.q. $b = f_i(a_1, \ldots, a_n)$ (pour un certain f_i)
est décidable

Théorème : Si la famille de règles $f_1, f_2, \ldots$ est effective,
l'ensemble des dérivations selon $f_1, f_2, \ldots$ est décidable

Décidabilité d'une fait d'être une dérivation

Soit E un ensemble (qui s'injecte dans $\mathbb{N}$)

Def : Une famille $f_1, f_2, \ldots$ de règles sur E est dite effective si l'ensemble $\mathcal{R}$ des listes $b, a_1, \ldots, a_n$ t.q. $b = f_i(a_1, \ldots, a_n)$ (pour un certain f_i) est décidable

Théorème : Si la famille de règles $f_1, f_2, \ldots$ est effective, l'ensemble des dérivations selon $f_1, f_2, \ldots$ est décidable

Preuve :

Algo. analyse récursivement l'arbre donné en argument.

Nœud étiqueté par b et enfants étiquetés par $a_1, \ldots, a_n$

on vérifie $a_1, \ldots, a_n, b$ est dans l'ensemble R

Vérification à chaque nœud

Semi-décidabilité du fait d'être dérivable

F ensemble des éléments de E dérivables par $f_1, f_2, \ldots$

Semi-décidabilité du fait d'être dérivable

F ensemble des éléments de E dérivables par $f_1, f_2, \ldots$

Théorème : Si la famille de règles $f_1, f_2, \ldots$ est effective,

F est semi-décidable

Semi-décidabilité du fait d'être dérivable

F ensemble des éléments de E dérivables par $f_1, f_2, \ldots$

Théorème : Si la famille de règles $f_1, f_2, \ldots$ est effective,

F est semi-décidable

Preuve :

Soit $f(x, y) = 1$ si x est le numéro d'une dérivation dont la racine est y,

et $f(x, y) = 0$ sinon.

Selon théorème précédent, f est calculable.

Soit $g(y)$ le plus petit entier x tel que $!f(x, y) = 0$

Soit g' la composée de g avec la fonction constante égale à 1

Si y appartient à F, alors $g'(y) = 1$, sinon g' n'est pas définie en y

Semi-décidabilité de la prouvabilité

Les règles de la logique des prédicats forment une famille effective.

Semi-décidabilité de la prouvabilité

Les règles de la logique des prédicats forment une famille effective.

Théorème : Soit $\mathcal{T}$ une théorie dont les axiomes forment un sous-ensemble décidable des propositions

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

Semi-décidabilité de la prouvabilité

Les règles de la logique des prédicats forment une famille effective.

Théorème : Soit $\mathcal{T}$ une théorie dont les axiomes forment un sous-ensemble décidable des propositions

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

Preuve : Soit $ax(p) = 1$ si $p = \ulcorner A_1 \wedge \ldots \wedge A_n \urcorner$ avec $A_1, \ldots, A_n$ axiomes de $\mathcal{T}$, et $ax(p) = 0$ sinon. ax est calculable.

Soit f' la fonction calculable

$f'(n, \ulcorner A \urcorner) = ax(hd(n))\&\&f(tl(n), (\ulcorner \Rightarrow \urcorner; hd(n); \ulcorner A \urcorner))$

et $g(\ulcorner A \urcorner)$ plus petit entier n t.q. $!f'(n, \ulcorner A \urcorner) = 0$

Semi-décidabilité de la prouvabilité

Les règles de la logique des prédicats forment une famille effective.

Théorème : Soit $\mathcal{T}$ une théorie dont les axiomes forment un sous-ensemble décidable des propositions

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

Preuve : Soit $ax(p) = 1$ si $p = \ulcorner A_1 \wedge \ldots \wedge A_n \urcorner$ avec $A_1, \ldots, A_n$ axiomes de $\mathcal{T}$, et $ax(p) = 0$ sinon. ax est calculable.

Soit f' la fonction calculable

$f'(n, \ulcorner A \urcorner) = ax(hd(n))\&\&f(tl(n), (\ulcorner \Rightarrow \urcorner; hd(n); \ulcorner A \urcorner))$

et $g(\ulcorner A \urcorner)$ plus petit entier n t.q. $!f'(n, \ulcorner A \urcorner) = 0$

On énumère tous les entiers, jusqu'à en trouver un qui encode un certain nombre (fini) d'axiomes de $\mathcal{T}$ et une preuve de A utilisant ces axiomes.

Si A est prouvable dans $\mathcal{T}$, cet entier finira bien par sortir sinon la recherche se poursuit à l'infini

V. Le théorème de Gödel

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

$g(\ulcorner A \urcorner)$ = plus petit entier x t.q. $!f'(x, \ulcorner A \urcorner) = 0$

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

$g(\ulcorner A \urcorner)$ = plus petit entier x t.q. $!(f'(x, \ulcorner A \urcorner) || f'(x, \ulcorner \neg \urcorner ; \ulcorner A \urcorner)) = 0$

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

$g(\ulcorner A \urcorner)$ = plus petit entier x t.q. $!(f'(x, \ulcorner A \urcorner) || f'(x, \ulcorner \neg \urcorner ; \ulcorner A \urcorner)) = 0$

Les 4 possibilités

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

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

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

4. Si A et $\neg A$ sont tous les deux prouvables

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

$g(\ulcorner A \urcorner)$ = plus petit entier x t.q. $!(f'(x, \ulcorner A \urcorner) || f'(x, \ulcorner \neg \urcorner ; \ulcorner A \urcorner)) = 0$

Les 4 possibilités

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

   g termine et retourne une démonstration de A

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

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

4. Si A et $\neg A$ sont tous les deux prouvables

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

$g(\ulcorner A \urcorner)$ = plus petit entier x t.q. $!(f'(x, \ulcorner A \urcorner) || f'(x, \ulcorner \neg \urcorner ; \ulcorner A \urcorner)) = 0$

Les 4 possibilités

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

   g termine et retourne une démonstration de A

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

   g termine et retourne une démonstration de $\neg A$

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

4. Si A et $\neg A$ sont tous les deux prouvables

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

$g(\ulcorner A \urcorner)$ = plus petit entier x t.q. $!(f'(x, \ulcorner A \urcorner) || f'(x, \ulcorner \neg \urcorner ; \ulcorner A \urcorner)) = 0$

Les 4 possibilités

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

   g termine et retourne une démonstration de A

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

   g termine et retourne une démonstration de $\neg A$

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

   g ne termine pas

4. Si A et $\neg A$ sont tous les deux prouvables

Le théorème de Gödel

Théorème : Soit $\mathcal{T}$ une extension de $\mathcal{T}_o$

qui a un $\mathbb{N}$-modèle et où les axiomes sont décidables.

Il existe une proposition A telle que ni A ni $\neg A$ ne soit prouvable

Le théorème de Gödel

Théorème : Soit $\mathcal{T}$ une extension de $\mathcal{T}_o$

qui a un $\mathbb{N}$-modèle et où les axiomes sont décidables.

Il existe une proposition A telle que ni A ni $\neg A$ ne soit prouvable

Preuve : Sinon, la fonction g serait totale,

et la fonction (totale) calculable $A \mapsto f'(g(\ulcorner A \urcorner), \ulcorner A \urcorner)$ coinciderait avec la fonction caractéristique de l'ensemble des théorèmes de $\mathcal{T}$.

Théorème : Soit $\mathcal{T}$ une extension de $\mathcal{T}_o$

qui a un $\mathbb{N}$-modèle et où les axiomes sont décidables.

Il existe une proposition A telle que ni A ni $\neg A$ ne soit prouvable

Preuve : Sinon, la fonction g serait totale,

et la fonction (totale) calculable $A \mapsto f'(g(\ulcorner A \urcorner), \ulcorner A \urcorner)$ coinciderait avec la fonction

caractéristique de l'ensemble des théorèmes de $\mathcal{T}$.

A mettre en perspective avec le théorème de complétion.

Où est le problème ?

Le théorème de Gödel

Théorème : Soit $\mathcal{T}$ une extension de $\mathcal{T}_o$

qui a un $\mathbb{N}$-modèle et où les axiomes sont décidables.

Il existe une proposition A telle que ni A ni $\neg A$ ne soit prouvable

Preuve : Sinon, la fonction g serait totale,

et la fonction (totale) calculable $A \mapsto f'(g(\ulcorner A \urcorner), \ulcorner A \urcorner)$ coinciderait avec la fonction

caractéristique de l'ensemble des théorèmes de $\mathcal{T}$.

A mettre en perspective avec le théorème de complétion.

Où est le problème ?

En PC : Variations sur le théorème de Gödel

Le théorème de Gödel

Théorème : Soit $\mathcal{T}$ une extension de $\mathcal{T}_o$

qui a un $\mathbb{N}$-modèle et où les axiomes sont décidables.

Il existe une proposition A telle que ni A ni $\neg A$ ne soit prouvable

Preuve : Sinon, la fonction g serait totale,

et la fonction (totale) calculable $A \mapsto f'(g(\ulcorner A \urcorner), \ulcorner A \urcorner)$ coinciderait avec la fonction caractéristique de l'ensemble des théorèmes de $\mathcal{T}$.

A mettre en perspective avec le théorème de complétion.

Où est le problème ?

En PC : Variations sur le théorème de Gödel

La prochaine fois : le calcul comme une suite de petits pas

Questions?