École polytechnique de Louvain

LINFO1123

CalculEAU

Formulaire

Année académique :
EPL BAC3

Auteurs :
Félix Gaudin

16 juin 2021

VSTUPLENIYE

Privet komrade, sintez formul i sovmestnaya demonstratsiya. (ce qui veut dire : *Salut komrade, synthèse des formules et démonstration collaboratif.*)

1 CONCEPT

1.1 Fonction

Soit $f : A \to B$

- **Domaine** de f : $dom(f) = \{a \in A \mid f(a) \neq \perp\}$
- **Image** de f : $image(f) = \{b \in B \mid \exists\, a \in A : b=f(a)\}$
- f est fonction **totale** ssi $dom(f) = A$ ($\nexists\, a \in A$ t.q. $f(a) = \perp$)
- f est fontcion **partielle** ssi $dom(f) \subseteq A$
(f totale est partielle mais f partielle n'est pas nécessairement totale)
- f est **surjective** ssi $image(f) = B$
- f est **injective** ssi $\forall a, a' \in A : a \neq a' \to f(a) \neq f(a')$ (ex : $f(x) = x^2$, $f(-1) = f(1)$ donc pas injective)
- f est **bijective** ssi f est totale + injective + surjective

1.2 Enumérable

Un ensemble est énumérable si soit il est fini ou si on peut le mettre en bijection avec $\mathbb{N}$.

Techniques pour prouver

- Faire un tableau
- Un sous-ensemble d'un ensemble énumérable est énumérable
- L'ensemble des sous ensemble d'un ensemble énumérable (infini) n'est pas énumérable
- **Cantor** (pour dire que non énumérable)
Soit $E = \{ x \in \mathbb{R} \mid 0 < x \leqslant 1 \}$
E est non énumérable.
Preuve :

1. Supposons E énumérable. Il existe donc une énumération des éléments de E : $x_0, x_1, \ldots, x_k, \ldots$

	1 digit	2 digit	3 digit	$\ldots$	$k+1$ digit	$\ldots$
x_0	x_{00}	x_{01}	x_{02}	$\cdots$	x_{0k}	$\cdots$
x_1	x_{10}	x_{11}	x_{12}	$\cdots$	x_{1k}	$\cdots$
x_0	x_{20}	x_{21}	x_{22}	$\cdots$	x_{2k}	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
x_k	x_{k0}	x_{k1}	x_{k2}	$\cdots$	x_{kk}	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

TABLE 1 – On va construire une table t.q. le nombre $x_k = 0, x_{k0}x_{k1}x_{k2}\ldots x_{kk}\ldots$

2. On va prendre la diagonale ($d = 0, x_{00}x_{11}x_{22}\ldots x_{kk}\ldots$)

3. Modifier la diagonale tel que
$$x'_{ii} = \begin{cases} 5 & \text{si } x_{ii} \neq 5 \\ 6 & \text{sinon} \end{cases}$$
On a donc d'= $0, x'_{00}x'_{11}x'_{22}\ldots x'_{kk}\ldots$ (d' $\in$ E)
4. **Contradiction**
Vu que E énumérable et que d'$\in$ E, alors d' doit être dans l'énumération. **Or** si d'= x_p alors
$$\begin{aligned} d' &= 0, x_{p0}\ x_{p1}\ x_{p2}\ \ldots x_{pp}\ \ldots \\ &= 0, x'_{p0}x'_{p1}x'_{p2}\ldots x'_{pp}\ldots \end{aligned}$$
5. **Conclusion**
E n'est pas énumérable

2 RÉSULTATS FONDAMENTAUX

2.1 Fonction calculable

Une fonction $f : \mathbb{N} \to \mathbb{N}$ est **calculable** ssi il existe un programme Oz qui, recevant comme données n'importe quel nombre naturel x, fourni comme résultat $f(x)$ si il est défini. Sinon $\perp$ (si ne se termine pas ou erreur)

2.1.1 *Ensembles récursif*

On dit que l'ensemble A est récursif ssi il existe un programme qui prend en input x et qui renvoi
$\begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$
Le programme calcule donc une fonction totale
Exemple : $\{x \in \mathbb{N} \mid x \text{ pair}\}$

2.1.2 *Ensemble récursivement énumérable*

On dit que l'ensemble A est récursif ssi il existe un programme qui prend en input x et qui renvoi
$\begin{cases} 1 & \text{si } x \in A \\ 0 \text{ ou boucle} & \text{si } x \notin A \end{cases}$

2.1.3 *Propriétés*

— A récursif $\implies$ A récursivement énumérable
— A **récursif** $\iff$ $\bar{A}$ **récursif**
— A récursif énum et $\bar{A}$ récursif énum $\iff$ A récursif
— A fini ou $\bar{A}$ fini $\implies$ A et $\bar{A}$ récursif

2.2 Calculabilité

Soit φ_k la fonction *avec le numéro* k ($\varphi_k : \mathbb{N} \to \mathbb{N}$) qui est calculée par le programme P_k qui est le programme *avec le numéro* k dans l'ensemble P.

2.2.1 *Problème de l'arrêt*

Soit $\mathtt{halt} : \text{PxN} \to \text{N}$ telle que $\mathtt{halt}(n,x) = \begin{cases} 1 & \text{Si } \varphi_n(x) \neq \perp \\ 0 & \text{Sinon} \end{cases}$

Preuve :
Supposons halt calculable

1. Construction de la table

	0	1	2	...	k	...
P_0	$\mathtt{halt}(0,0)$	$\mathtt{halt}(0,1)$	$\mathtt{halt}(0,2)$	...	$\mathtt{halt}(0,k)$	...
P_1	$\mathtt{halt}(1,0)$	$\mathtt{halt}(1,1)$	$\mathtt{halt}(1,2)$	...	$\mathtt{halt}(1,k)$	...
P_2	$\mathtt{halt}(2,0)$	$\mathtt{halt}(2,1)$	$\mathtt{halt}(2,2)$	...	$\mathtt{halt}(2,k)$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
P_k	$\mathtt{halt}(k,0)$	$\mathtt{halt}(k,1)$	$\mathtt{halt}(k,2)$	...	$\mathtt{halt}(k,k)$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

TABLE 2 – Table des valeurs de la fonction halt

2. Sélectionner la diagonale : $\mathtt{diag}(n) = \mathtt{halt}(n,n)$
3. Modifier la diag
   $\mathtt{diag\_mod}(n) = \begin{cases} 1 & \text{si } \mathtt{halt}(n,n) = 0 \\ \perp & \text{si } \mathtt{halt}(n,n) = 1 \end{cases}$
   Si $\mathtt{halt}$ caculable alors $\mathtt{diag\_mod}$ calculable. On va noter P_d le programme qui calcule cette fonction.
4. Contradiction (ça tourne mal explications)
   Quelle est la valeur de $\mathtt{diag\_mod}(d)$?
   - *Si* $\mathtt{diag\_mod}(d) = 1$
     $\mathtt{diag\_mod}(d) = 1 \to \mathtt{halt}(d,d) = 0 \to P_d$ se termine pas **OR** on a dit que $\mathtt{diag\_mod}(d)$ valait 1
   - *Si* $\mathtt{diag\_mod}(d) = \perp$
     $\mathtt{diag\_mod}(d) = \perp \to \mathtt{halt}(d,d) = 1 \to P_d$ se termine **OR** on a dit qu'il vallait $\perp$
5. Conclusion : $\mathtt{diag\_mod}$ n'est pas calculable or notre seule hypothèse était que $\mathtt{halt}$ est calculable. Donc $\mathtt{halt}$ n'est pas calculable.

On peut donc définir un ensemble $\mathsf{HALT} = \{(n,x) \mid \mathtt{halt}(n,x) = 1\}$ et un ensemble $K = \{n \mid (n,n) \in \mathsf{HALT}\}$. Les deux ensembles sont

2.3 Hoare-Allison

Soit un langage Q qui a des programmes Q_k. La fonction φ'_k est calculée par le programme Q_k. Le langage Q calcule **que** des fonctions totales. On peut donc calculer la fonction $\mathtt{halt}$ de ce langage par ce langage (fonction constante qui vaut 1). Nonobstant, ce langage ne peut pas calculer son interpreteur appelé $\mathtt{interpret}(n,x)$.

Preuve

1. Construction de la table

	0	$\dots$	k	$\dots$
Q_0	$\mathtt{interpret}(0,0)$	$\dots$	$\mathtt{interpret}(0,k)$	$\dots$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
Q_k	$\mathtt{interpret}(k,0)$	$\dots$	$\mathtt{interpret}(k,k)$	$\dots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

2. On prend la diagonale qu'on appelle $\mathtt{diag}(n) = \mathtt{interpret}(n,n)$ qui est totale et donc dans Q
3. On modifie $\mathtt{diag}$ avec $\mathtt{diag\_mod} = \mathtt{interpret}(n,n)+1$ qui est totale et donc dans Q
4. On note d le numéro de $\mathtt{diag\_mod}$ dans Q. On a donc $\mathtt{diag\_mod}(d) = \mathtt{interpret}(d,d)+1$ **OR** $\mathtt{diag\_mod}(d) = \varphi'_d(d) = \mathtt{interpret}(d,d)$
5. Conclusion : l'interpreteur de Q n'est pas calculable dans Q

Conséquences

— La fonction $\mathtt{tot}(n)$ qui renvoit 1 si φ_n est totale sinon 0 n'existe pas

— L'ensemble $\{n \mid \varphi_n \text{ totale}\}$ est non récursif

2.4 Insufisance des fonctions totales

2.5 Rice

Y'a deux formes :

1. Soit $A \subseteq \mathbb{N}$, Si A récursif et $A \neq \emptyset$ et $A \neq \mathbb{N}$ alors $\exists i \in A$ et $j \in \bar{A}$ tel que $\varphi_i = \varphi_j$
2. Si $\forall i \in A$ et $\forall j \in \bar{A}$: $\varphi_i \neq \varphi_j$ alors A non récursif ou $A = \emptyset$ ou $A = \mathbb{N}$

Preuve

1. On suppose un ensemble A récursif
   On pose $P_k(x) \equiv$ while(True) : $(\varphi_k = \perp)$
   Si $k \in \bar{A}$ alors $A \neq \emptyset \implies \exists m \in A$ et par la définition, $\varphi_k \neq \varphi_m$
2. Construction de $\mathtt{halt}$

$$\mathtt{halt}(n,x) \equiv \left[\begin{array}{l} \text{Construire le programme (PAS L EXECUTER) } P(z) \equiv P_n(x); P_m(z) \\ \text{d = numéro du programe} P(z) \\ \text{If } d \in A \text{ then } \mathtt{print}(1) \text{ Si } P_n \text{ se termine } \varphi_d = \varphi_m \to d \in A \\ \text{Sinon } \mathtt{print}(0) \text{ Si } P_n \text{ se termine pas } \varphi_d = \varphi_k \to d \in \bar{A} \end{array}\right.$$

3. A non récursif (sinon on sait calculer $\mathtt{halt}$ ce qui est pas possible)

2.6 Paramétrisation

Forme S-1-1

Il existe une fonction totale calculable $S_1^1 : \mathbb{N}^2 \to \mathbb{N}$ t.q. $\forall$ k : $\varphi_k(x_1, x_2) = \varphi_{S_1^1(k,x_2)}(x_1)$

2.6.1 Forme S-m-n

$\forall m, n \geqslant 0$ $\exists$ une fonction totale calculable $S_n^m : \mathbb{N}^{m+1} \to \mathbb{N}$ t.q. $\forall k$:
$\varphi_k(x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m}) = \varphi_{S_1^1(k, x_{n+1}, \dots, x_{n+m})}(x_1, \dots, x_n)$

2.7 Point fixe

Soient $n \geqslant 0$ et f une fonction totale calculable. Il existe k tel que $\varphi_k = \varphi_f(k)$
Preuve
Soit f une fonction totale calculable. Il existe k tel que $\varphi_k = \varphi_{f(k)}$.
On va poser :
Lapin 1

$$h(u,x) = \begin{cases} \varphi_{\varphi_u(u)}(x) & \text{Si } \varphi_u(u) \neq \perp \\ \perp & \text{Sinon} \end{cases} \tag{1}$$

(h est calculable)
Lapin 2

$$h(u,x) = \varphi_{S(u)}(x) \tag{2}$$

Application de la propiété $S-1-1$
Lapin 3

$$g(x) = f(S(x)) \tag{3}$$

g est totale calculable (car S et f le sont)
f est donné par $\exists k'$ tel que $\varphi_{k'}(x) = g(x) = f(S(x))$

On a que k' est une contant par le Lapin 2

$$h(k',x) = \varphi_{S(k')}(x)$$

Par le lapin 1 et vu que $g = \varphi_{k'}$ on a :

$$h(k',x) = \varphi_{\varphi_{k'}(k')}(x)$$

Par le lapin 3, on a que $\varphi_{k'}(x) = g(x) = f(S(x))$ donc :

$$h(k',v) = \varphi_{f(S(k'))}(x)$$

Si on pose que $S(k') = k$ on a :

$$\varphi_k(x) = \varphi_{f(k)}(x)$$

3 MODÈLES

3.1 ND-JAVA

Fonction $\mathtt{choose}()$ qui renvoie 0 ou 1

3.1.1 ND-Récursif

A est ND-Récursif si il existe un programme ND-JAVA t.q. s'il reçoit un input $\in \mathbb{N}$
— $x \in A$ alors il existe une exécution qui renvoie 1
— $x \notin A$ alors pour toute exécution, le résultat est 0

ND-Récursif énumérable

La même sauf que le cas $x \notin A$ ne se fini pas d'office

Propriétés

— Récursif $\implies$ ND-Récursif
— Récursif énumérable $\implies$ ND-Récursif énumérable

4 RÉDUCTIONS

4.1 Réduction algorithmique (calculabilité)

Un ensemble A *est* ***algorithmiquement réductible*** *à un ensemble* B *(*$A \leqslant_a B$*) si en supposant* B *récursif, alors* A *est récursif*

Exemple :
Soit $P = \{n \mid \varphi_n$ renvoit un nombre pair$\}$
HALT $\leqslant_a$ P Si P énumérable on a HALT énumérable
Propriétés

— Si $A \leqslant_a B$ et B récursif, alors A récursif
— Si $A \leqslant_a B$ et A non récursif, alors B non récursif
— $A \leqslant_a \bar{A}$
— $A \leqslant_a B \leftrightarrow \bar{A} \leqslant_a \bar{B}$
— Si A récursif, alors pour tout B, $A \leqslant_a B$
— Si $A \leqslant_a B$ et B récursivement énumérable, alors A n'est pas nécessairement énumérable

4.2 Réduction fonctionnelle (complexité)

Un ensemble A *est* ***fonctionnellement réductible*** *à un ensemble* B *(*$A \leqslant_f B$*) ssi il existe une fonction totale calculable* f *telle que*

$$a \in A \leftrightarrow f(a) \in B$$

Vu qu'on connait pas f ça donne 0 info sur la complexité de A

4.3 Réduction polynomiale

Un ensemble A *est* ***polynomialement réductible*** *à un ensemble* B *(*$A \leqslant_p B$*) ssi il existe une fonction totale calculable* f *de complexité temporelle polynomiale t.q. :*

$$a \in A \leftrightarrow f(a) \in B$$

Si $A \leqslant_p B$ et $B \in P$ Alors $A \in P$

4.4 Relations entre classes de complexité

4.4.1 *Déterministe VS non déterministe*

— Si $A \in NTIME(f)$ Alors $A \in DTIME(c^f)$
— Si $A \in NSPACE(f)$ Alors $A \in DSPACE(f^2)$
— $NSPACE(f) = DSPACE(f)$

4.4.2 *Time VS Space*

- Si $A \in NTIME(f)$ Alors $A \in NSPACE(f)$
- Si $A \in DTIME(f)$ Alors $A \in DSPACE(f)$
- Si $A \in NSPACE(f)$ Alors $A \in NTIME(c^f)$
- Si $A \in DSPACE(f)$ Alors $A \in DTIME(c^f)$

4.5 Formalismes de calculabilité

Soit D un nouveau formalisme de calculabilité :
Caractéristiques d'un formalisme

- SD **(soundness des descriptions)**
  Toute fonction D-calculable est calculable
- CD **(Complétude des définitions)**
  Toute fonction calculable est D-calculable
- SA **(Soundness algorithmique)**
  Interpreteur de D est calculable
- CA **(Complétude algorithmique)**
  Si $p \in L$ (L est par exemple Oz), que $p' \in D$ et que $p \equiv p'$ (calculent la même fonction) Alors équivalence des formalismes
- U **(description universelle)**
  Interpreteur de D est D-calculable
- S **(S-m-n affaiblie)** $\forall d \in D \exists S : d(x, y) = [S(x)](y)$

Propriétés

- SA $\implies$ SD
- CA $\implies$ CD
- SD et U $\implies$ SA
- CD et S $\implies$ CA
- SA et CD $\implies$ U
- CA et SD $\implies$ S
- S et U $\implies$ S-m-n
- SA et CA $\iff$ SD et CD et U et S
- SA et CD et S $\iff$ CA et SD et U

PREUVES POUR LE BONUS

Soit F l'ensemble des fonction totales telle que $f : \mathbb{N} \to \mathbb{N}$ F est non énumérable.
Preuve :

1. Supposons F énumérable. Il existe donc une énumération des éléments de F : $f_0(0), f_1(0), \dots$

	1	2	3	$\dots$	k	$\dots$
f_0	$f_0(1)$	$f_0(2)$	$f_0(3)$	$\dots$	$f_0(k)$	$\dots$
f_1	$f_1(1)$	$f_1(2)$	$f_1(3)$	$\dots$	$f_1(k)$	$\dots$
f_2	$f_2(1)$	$f_2(2)$	$f_2(3)$	$\dots$	$f_2(k)$	$\dots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
f_k	$f_k(1)$	$f_k(2)$	$f_k(3)$	$\dots$	$f_k(k)$	$\dots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Table 3 – On va avoir pour la donnée $f_i(j)$ qui est le résultat de la fonction avec le numéro i pour la donnée j

2. On va prendre la diagonale (d) qui est aussi un fonction de $\mathbb{N} \to \mathbb{N}$ ($d \in F$)
3. Modifier la diagonale tel que
$$f'_i(i) = \begin{cases} 5 & \text{si } f_i(i) \neq 5 \\ 6 & \text{sinon} \end{cases}$$
On va noter cette diagonale modifiée d'
4. **Contradiction**
Vu que F énumérable et que $d' \in F$, alors d' doit être dans l'énumération. **Or** si d' a le numéro p alors
$$\begin{aligned} d' &= f_p(0), f_p(1), f_p(2), \dots, f_p(p), \dots \\ &= f'_p(0), f'_p(1), f'_p(2), \dots, f'_p(p), \dots \end{aligned}$$
Si $f_p(0)$ vaut 5 on aura $f_p(0) = 5 \neq f'_p(0) = 6$ donc on a une contradiction
5. **Conclusion**
F n'est pas énumérable