Langages formels
« Ad Litteram »

Fonctions récursives primitives, Argument diagonal, Fonction d'Ackermann-Péter, Fonctions récursives, partielles et totales.

Révolution Galiléenne, Langage, nature et pensée.

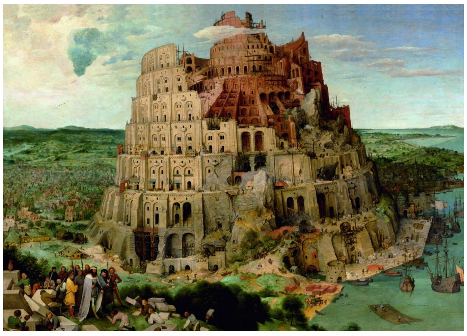
Modèle des machines à registres, Notations, Séquentialité, Théorème d'équivalence.

Sensibilit

Olivier Raynaud

1

Tour de Babel



La Tour de Babel vue par Pieter Brueghel l'Ancien au XVI^{e} siècle.

2

Civilisation mésopotamienne

- *Scribes de Babylone – Ecriture cunéiforme*

https://photo.geo.fr/irak-menace-sur-les-joyaux-du-patrimoine-mesopotamien-32059#cite-sumerienne-de-mesopotamie-571542

La XI^{e} tablette de la version de Ninive de l'*Épopée de Gilgamesh*, relatant le Déluge. British Museum.

3

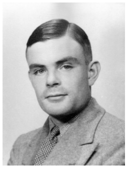
Thèse de Church-Turing

- *Modèle de calcul*

Thèse (Calculabilité - Church-Turing) :

Les règles formelles de calcul décrivant les machines de Turing, les fonctions $\lambda-$définissables, ou les fonctions récursives formalisent correctement la notion de méthode effective de calcul ou de calculabilité.

Alonzo Church (1903-1995)



Alan Turing (1912 -1954)

Wikipedia: https://fr.wikipedia.org/wiki/Th%C3%A8se_de_Church

4

Thèse de Church-Turing

- *Méthode effective de calcul*

Méthode effective de calcul :
On considère généralement que la notion intuitive de méthode effective de calcul correspond aux caractéristiques suivantes :

1) l'algorithme consiste en un ensemble fini d'instructions simples et précises qui sont décrites avec un nombre limité de symboles ;
2) l'algorithme doit toujours produire le résultat en un nombre fini d'étapes ;
3) l'algorithme peut en principe être suivi par un humain avec seulement du papier et un crayon ;
4) l'exécution de l'algorithme ne requiert pas d'intelligence de l'humain sauf celle qui est nécessaire pour comprendre et exécuter les instructions.

Wikipedia: https://fr.wikipedia.org/wiki/Th%C3%A8se_de_Church

5

Notations

- *Notation issues du lambda calcul*

Notations (Fonctions et arité) :

On désigne par $\mathcal{F}_*$ l'ensemble de toutes les fonctions (partielle ou totale) de la forme $f : \mathbb{N}^k \rightarrow \mathbb{N}$ pour un entier k quelconque. On utilise aussi la notation $\lambda x_1 ... x_k [f(x_1 ... x_k)]$ pour désigner la fonction $f : \mathbb{N}^k \rightarrow \mathbb{N}$. L'entier k s'appelle l'arité de la fonction f. On note $\mathcal{F}_k$ l'ensemble des fonctions de $\mathcal{F}_*$ qui sont d'arité k. Ainsi $\mathcal{F}_* = \bigcup_{k \in \mathbb{N}} \mathcal{F}_k$.

6

Fonctions de base

Fonction nulle $\text{zero} : \mathbb{N}^0 \to \mathbb{N}$
$zero() = 0$

Fonction successeur $\text{succ} : \mathbb{N} \to \mathbb{N}$
$succ(n) = n + 1$

Fonctions de projection $\text{projection } \Pi_i^k : \mathbb{N}^k \to \mathbb{N}$
$\Pi_i^k(x_1, ..., x_k) = x_i$

7

Opérateur de composition

- *Composition :*

Définition (Composition) **:**

Etant donnés k, j dans $\mathbb{N}$, et des fonctions $g : \mathbb{N}^k \to \mathbb{N}$ et $h_1, h_2, ..., h_k : \mathbb{N}^j \to \mathbb{N}$, nous définissons la composition $f : N^j \to \mathbb{N}$ de g avec $h_1, h_2, ..., h_k$

$$f(\overrightarrow{n}) = g(h_1(\overrightarrow{n}), h_2(\overrightarrow{n}), ..., h_k(\overrightarrow{n}))$$

$\overrightarrow{n} = \{x_1, x_2, ..., x_k\}$

8

Opérateurs de récursion

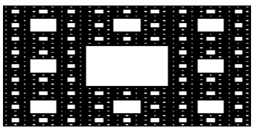
- *Récursion primitive :*

Intuitivement : permet d'implémenter la structure de contrôle de boucle « pour ».

Définition (Récursion primitive) **:**

Etant donnés k dans $\mathbb{N}$, des fonctions $g : \mathbb{N}^k \to \mathbb{N}$ et $h : \mathbb{N}^{k+2} \to \mathbb{N}$, nous définissons la récursion primitive $f : N^{k+1} \to \mathbb{N}$ de g avec h

$$f(0, \overrightarrow{n}) = g(\overrightarrow{n})$$
$$f(m+1, \overrightarrow{n}) = h(f(m, \overrightarrow{n}), \overrightarrow{n}, m)$$



$\overrightarrow{n} = \{x_1, x_2, ..., x_k\}$

$$add(0, x) = \Pi_1^1(x)$$
$$add(y+1, x) = succ(\Pi_1^3(add(y, x), x, y))$$

9

Exemples

Fonction prédecesseur $pred : \mathbb{N} \to \mathbb{N}$

$$pred(x) = \begin{cases} x - 1 & \text{si} \quad x > 0 \\ 0 & \text{sinon} \end{cases}$$

$pred(0) = zero()$
$pred(x+1) = \Pi_2^2(Pred(x), x)$

Fonction différence $moins : \mathbb{N}^2 \to \mathbb{N}$

$$moins(y, x) = \begin{cases} x - y & \text{si} \quad x > y \\ 0 & \text{sinon} \end{cases}$$

$moins(0, x) = \Pi_1^1(x)$
$moins(y+1, x) = pred(\Pi_1^3(moins(y, x), x, y))$

10

Fonctions récursives primitives

- *Fonctions récursives primitives:*

Définition (Fonctions primitives récursives) **:**
L'ensemble des fonctions récursives primitives, noté $PRIM$, est composé des fonctions suivantes :
- La fonction constante égale à 0 : $cst() = 0$
- La fonction successeur : $succ(n) = n + 1$
- Les fonctions de projection : $\pi_i^k(\vec{n}) = n_i$, $\forall 1 \leq i \leq k$.

et est le plus petit ensemble clos par les opérateurs de composition et de récursion primitive.

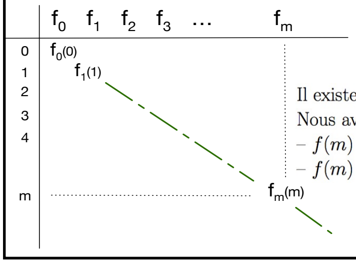
Exemple : Les fonctions constantes $cst_k() = k$
La fonction de $\mathbb{N}^2 \to \mathbb{N}$: $somme(x, y) = x + y$

11

Fonction non récursive primitive

Théorème : Il existe une fonction calculable qui n'est pas dans PRIM.

Démonstration par argument diagonale
L'ensemble des fonctions de PRIM peut être ordonné.



Soit $f : \mathbb{N} \to \mathbb{N}$
$n \to f_n(n) + 1$

Il existe m tel que $f = f_m$.
Nous avons alors :
- $f(m) = f_m(m)$ par sa place dans la diagonale
- $f(m) = f_m(m) + 1$ par définition de f

12

Fonction d'Ackermann-Péter

- *Fonction récursive mais non récursive primitive.*

La fonction d'**Ackermann-Péter** est définie récursivement comme suit :

$$A(0,n) = n+1$$
$$A(m+1,0) = A(m,1)$$
$$A(m+1,n+1) = A(m, A(m+1,n))$$

$$\begin{aligned}
\underline{A(2,1)} &\rightarrow_{r3} A(1,\underline{A(2,0)}) \\
&\rightarrow_{r2} A(1,\underline{A(1,1)}) \\
&\rightarrow_{r3} A(1,A(0,\underline{A(1,0)})) \\
&\rightarrow_{r2} A(1,A(0,\underline{A(0,1)})) \\
&\rightarrow_{r1} A(1,\underline{A(0,2)}) \\
&\rightarrow_{r1} \underline{A(1,3)} \\
&\rightarrow_{r3} A(0,\underline{A(1,2)}) \\
&\rightarrow_{r3} A(0,A(0,\underline{A(1,1)})) \\
&\rightarrow_{r3} A(0,A(0,A(0,\underline{A(1,0)}))) \\
&\rightarrow_{r2} A(0,A(0,A(0,\underline{A(0,1)}))) \\
&\rightarrow_{r1} A(0,A(0,\underline{A(0,2)})) \\
&\rightarrow_{r1} A(0,\underline{A(0,3)}) \\
&\rightarrow_{r1} \underline{A(0,4)} \\
&\rightarrow_{r1} 5
\end{aligned}$$

13

Fonction d'Ackermann-Péter

- *Fonction récursive mais non récursive primitive.*

$$A(0,n) = n+1$$
$$A(m+1,0) = A(m,1)$$
$$A(m+1,n+1) = A(m, A(m+1,n))$$

Valeurs de *A*(*m*, *n*)

m\n	0	1	2	3	4	n
0	1	2	3	4	5	$n+1$
1	2	3	4	5	6	$n+2$
2	3	5	7	9	11	$2n+3$
3	5	13	29	61	125	$2^{n+3}-3$
4	13	65533	$2^{65536}-3$	$A(3, 2^{65536}-3)$	$A(3, A(4, 3))$	$2^{2^{\cdot^{\cdot^{2}}}}-3$ ($n+3$ termes)
5	65533	$A(4, 65533)$	$A(4, A(5, 1))$	$A(4, A(5, 2))$	$A(4, A(5, 3))$	
6	$A(5, 1)$	$A(5, A(5, 1))$	$A(5, A(6, 1))$	$A(5, A(6, 2))$	$A(5, A(6, 3))$	

14

Minimisation et fonction récursive

- *Minimisation :*

Intuitivement : permet d'implémenter la structure de contrôle de boucle « tant que ».

Définition (Minimisation) :
Etant donnés une fonction totale $f : \mathbb{N}^{k+1} \to \mathbb{N}$,
on définit l'opérateur de minimisation noté μ

$$\mu(f)(\overrightarrow{n}) = m \text{ si } f(i,\overrightarrow{n}) > 0 \text{ pour } i \in [1, m-1] \text{ et } f(m,\overrightarrow{n}) = 0$$

$$\begin{aligned} div_2 : \mathbb{N} &\to \mathbb{N} \\ x &\to \left\lfloor \frac{x}{2} \right\rfloor \end{aligned}$$

$$div_2(x) = \mu_y[2 * (y+1) > x]$$

$$racine : \mathbb{N} \to \mathbb{N}$$
$$racine(x) = \begin{cases} \sqrt{x} & \text{si} \quad x \; est \; carre \\ \uparrow & \text{sinon} \end{cases}$$

$$\sqrt{x} = \mu_y[y * y = x]$$

15

Fonctions récursives

❑ *Fonctions récursives :*

Définition (Fonctions Récursives) **:**
L'ensemble des fonctions μ−récursives, noté REC, est composé des fonctions suivantes

- La fonction constante égale à 0 : $cst() = 0$
- La fonction successeur : $succ(n) = n + 1$
- Les fonctions de projection : $\pi_i^k(\overrightarrow{n}) = n_i$, $\forall 1 \leq i \leq k$.

et est le plus petit ensemble clos par les opérateurs de composition, de récursion primitive et de minimisation.

16

Fonctions totales ou partielles

❑ *Fonctions totales ou partielles :*

Notation :
Soit $f : \mathbb{N}^k \to \mathbb{N}$ une fonction récursive
- Si f est définie en $\overrightarrow{n}$ nous notons $f(\overrightarrow{n}) \downarrow$
- Sinon nous notons $f(\overrightarrow{n}) \uparrow$

Définition (Fonction totale et fonction partielle) **:**
Soit $f : \mathbb{N}^k \to \mathbb{N}$ une fonction récursive
- Si pour tout $\overrightarrow{n}$ dans $\mathbb{N}^k$ on a $f(\overrightarrow{n}) \downarrow$, on dit que f est totale ;
- Sinon on dit que f est partielle.

17

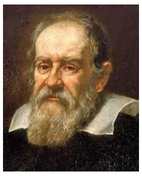
Fonctions récursives

❑ *Fonctions récursives :*

Question (Fonctions Récursives) :
Existe-t-il des fonction non récursives?

18

Méthode scientifique



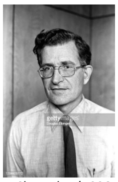
Révolution Galiléenne :
Ce sont les systèmes abstraits que l'on construit qui correspondent vraiment à la vérité.
L'ensemble des phénomènes est une distorsion de la vérité car trop de facteurs de toutes sortes y sont impliqués.

Définition : Le *langage* est un ensemble de moyens fini pour exprimer un éventail illimité de pensées. *(Galilée 1564-1642)*

19

Langage et nature…

Qu'est ce que le langage et en quoi est-ce important ?
Nous devons identifier ce qu'est le langage pour mieux en étudier les formes et les modèles.



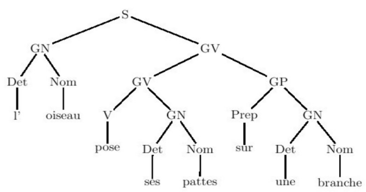
Noam Chomsky (1928 – …)

Propriété fondamentale du langage : Chaque langue fournit une palette infinie d'expressions structurées hiérarchiquement et qui peuvent être intérprétées au niveau de deux interfaces : l'interface conceptuelle intentionnelle et l'interface sensori motrice.

20

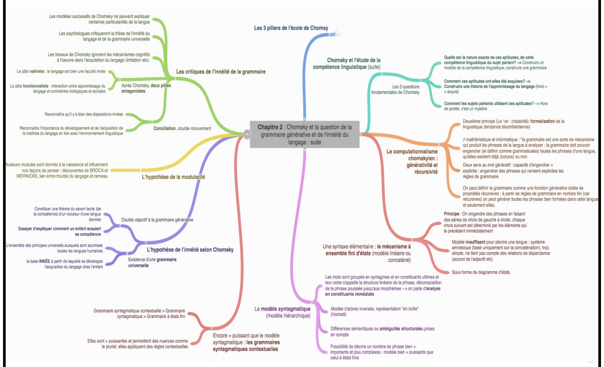
Grammaire générative

Grammaire générative :
On peut définir une langue non pas en énumérant exhaustivement les structures qu'elle autorise, mais de manière générative, parce qu'on peut engendrer à partir des règles de la grammaire toutes les phrases bien formées dans cette langue et seulement elles.



21

Grammaire générative et innéité du langage



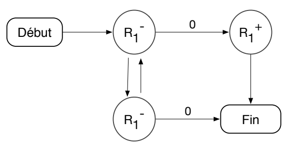
https://coggle.it/diagram/Wwwi7ufglzsI77P_/t/chapitre-2-chomsky-et-la-question-de-la-du-langage-suite

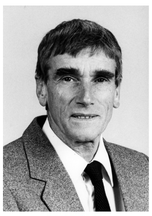
22

Machine à Registres

Il existe plusieurs modèles de machine à registres, introduits au cours des années 50 et 60. Nous utiliserons dans ce chapitre un modèle simplifié de ces machines.

- *Le modèle retenu s'inspire largement de celui introduit par* J.C. Shepherdson et H.E. Sturgis en 1963.





JOHN CEDRIC SHEPHERDSON

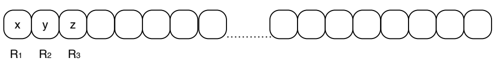
23

Machine à Registres

Les machines à registres utilisent un nombre quelconque de registres adressés de façon unique et contenant un entier positif ou nul.

Ainsi :

- La mémoire d'une machine à registres est composé d'un nombre virtuellement infini de *registres* (R_1, R_2,...). Un registre peut contenir un entier naturel *n* quelconque aussi grand que shouaité.
  A un instant donné, il n'y a qu'un nombre fini de registres non vides.
- Les opérations possibles sur un registre donné sont l'*incrémentation* et la *décrémentation*.



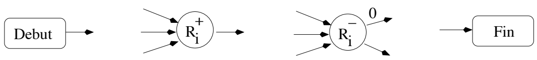
24

Machine à Registres

❑ Programme sous forme de graphe.

Programme :

- Le programme est représenté par un graphe orienté fini **G=(V,E)** étiqueté qui contient deux sommets particuliers:
  - Sommet ***Début*** aussi appelé état initial de la machine;
  - Sommet ***Fin*** aussi appelé état final de la machine.
- Les autres sommets sont de 2 sortes:
  - Les sommets incrément étiquetés par un entier ***n***;
  - Les sommets de décrément étiqueté par un entier ***n***.



25

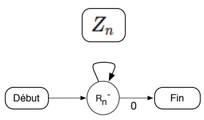
Calculabilité

Définition (Calculabilité) **:**
Une fonction $f : \mathbb{N}^k \to \mathbb{N}$ est dite calculable par une machine à registres, s'il existe une machine M telle que $\forall(x_1, ..., x_k) \in \mathbb{N}^k$, nous avons $f(x_1, ..., x_k) = M(x_1, ..., x_k)$.

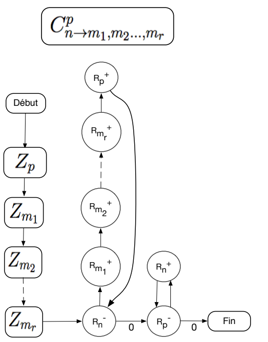
❑ Exemple de programme pour les *fonctions d'addition et de soustraction*

26

Quelques machines génériques



Remise à zéro



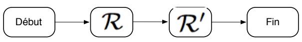
Copie du registre n dans les registres $m_1, ..., m_r$

27

Séquentialité

Notation (Enchainement de machines) **:**
Soient deux machines $\mathcal{R}$ et $\mathcal{R}'$, on note $\mathcal{R} \rhd \mathcal{R}'$ l'enchainement séquentiel des deux machines qui partagent le même ensemble de registres.

- Son graphe peut être représenté de cette façon :



28

Complexité

Notation (Complexité en mémoire)
Soit $\mathcal{R}$ une machine à registres, on désigne par $\rho(\mathcal{R})$ le plus grand numéro de registre figurant dans le graphe de $\mathcal{R}$.
Si tous les registres sont utilisés, $\rho(\mathcal{R})$ correspond au nombre de registres nécessaires à l'exécution de la machine $\mathcal{R}$.

Notation (Adresse d'éxecution)
Soit $\mathcal{R}$ une machine à registre, et n un entier, on note $\mathcal{R}_{\to n}$ la machine obtenue à partir de $\mathcal{R}$ en augmentant de n toutes les références aux registres dans les étiquettes des sommets du graphe.

29

Calculabilité

Théorème (Calculabilité)
Toute fonction μ-récursive est calculable par une machine à registres.

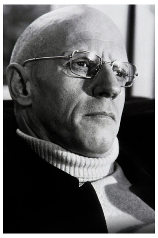
Démonstration
On pourra montrer que chacune des fonctions **zéro**, d'**incrément** et de **projection** sont calculables.

On montrera aussi que les opérateurs de **composition**, de **récursion** et de **minimisation** sont calculables par des machines à registres.

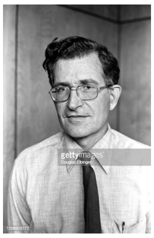
30

Langage, pensée et politique…

Débat (1971) : La nature humaine a-t-elle une base essentiellement inée ou s'acquiere-t-elle entièrement socialement?



Michel Foucault (1926 – 1984)



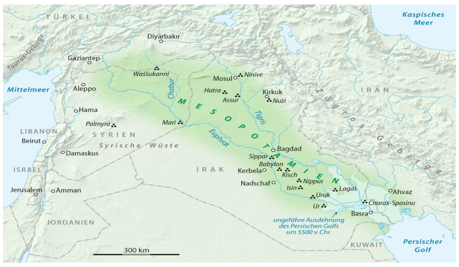
Noam Chomsky (1928 – …)

https://www.gettyimages.fr/photos/noam-chomsky

31

Le berceau mésopotamien et ses patriarches

Parole biblique "Abraham, quitte ton pays, ta parenté et la maison de ton père et va dans le pays que je te montrerai".



32

Pour résumer

- *Quelques points essentiels de l'exposé*
  - ✓ **Fonction primitive récursive (PRIM):** fonctions nulle, successeur et de projection, opérateur de composition et de récursion
  - ✓ **Argument diagonal** pour montrer qu'il existe des fonctions non primitives récursives.
  - ✓ **Fonction d'Ackermann-Péter**
  - ✓ **Fonction récursive (REC):** opérateur de minimisation, fonction totale et fonction partielle.
  - ✓ **Modèle de Machines à registres :** Mémoire, Graphe d'exécution, Unitarisation, exemples, Equivalence de calcul.

33

Bibliographie notoire

Ouvrages et liens de références exploités pour la réalisation de l'exposé.

- **Calculabilité, Complexité et Approximation,** édition Vuibert 2004, par J.F. Rey
- **Sur la nature et le langage,** Agone édition 2011, par Noam Chomsky, édité par Adriana Belliti & Luigi Rizzi (Préc. Édition Cambridge University Press 2002)
- **Sur la nature humaine,** édition Aden, par Noam Chomsky et Florent Foucault
- **Les mathématiciens de Babylone**, Edition presse del a rénaissance, par R. Caratini (2002)
- **Abraham où la 5ième alliance**, Gallimard 2021, Boualem Sansal
- **L'Intelligence Artificielle s'est-elle trompée** https://weekly-geekly-es.imtqy.com/articles/fr432846/index.html

34