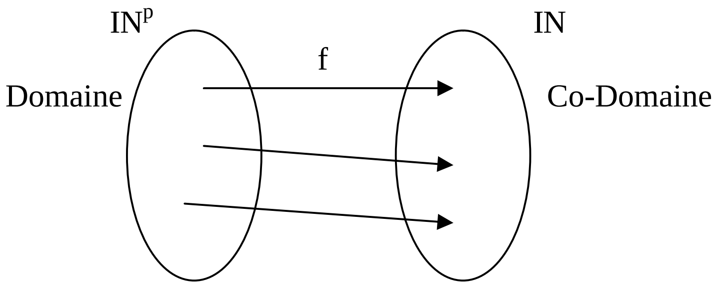
CALCULABILITE
1ère Partie : RECURSIVITE

I) – <u>INTRODUCTION</u> :

La notion de <u>décidabilité</u> d'un problème et les notions qui lui sont liées comme la <u>semi-décidabilité</u> , la <u>calculabilité</u> et la <u>récursivité</u> sont fondamentale pour l'informatique en général. Car elles caractérisent les choses qui sont à sa portée :

- Un problème qu'un ordinateur est susceptible de résoudre est un problème dit, décidable càd un problème correspondant à un prédicat décidable. Ce dernier est un terme ayant un sens mathématique rigoureux et indépendant de tout langage de programmation.
- Une fonction qu'un ordinateur peut calculer est une fonction dite, récursive càd une fonction à laquelle correspond un processus de calcul (un algorithme qui la calcule). La récursivité elle aussi peut être définie mathématiquement et formellement

Dans ce chapitre nous nous intéresserons à la calculabilité des fonctions de IN^p ---> IN :



p est appelé l'arité de la fonction f :
p =1 , f est d'arité 1 : f est dite monaire
p =2, f est d'arité 2 : f est dite binaire
p =3, f est d'arité 3 : f est dite tertiaire
…
p =n, f est d'arité n : f est dite n-aire

Nous allons étudier deux (02) aspects de la calculabilité des fonctions :

- La notion de RECURSIVITE
- La notion de MACHINE DE TURING

II) – LES FONCTIONS RECURSIVES :

Notation : Nous utiliserons par la suite la notation dite λ - Notation pour définir une fonction.
Exemple la fonction ⊕ sera notée : $\oplus = \lambda\ x\ y\ .\ x+y$

Les Variables Le corps de la fonction

Nous nous intéressons à présent à la question suivante :
Etant donnée une fonction f quelconque de IN^p ---> IN , f est elle calculable ?
Réponse : f est calculable si elle vérifie certains critères.

Pour cela, nous nous donnons des outils qui nous permettent de construire cette fonction. Si nous réussissons, cela veut dire que la fonction est calculable. Mais toute construction à besoin de matière première et de règles de construction. Dans notre cas la matière première sont des fonctions de bases supposées calculables, alors que nos règles de construction sont des règles qui nous permettent de combiner des fonctions déjà calculables pour en construire d'autres.

II .1) – LES FONCTIONS DE BASE :

- La Fonction Z d'arité 1 : $Z = \lambda\ n\ .\ 0$
- La Fonction S d'arité 1 : $S = \lambda\ n\ .\ n+1$
- Les Fonctions Projections d'arité n (n>=1) : $P^n_i = \lambda\ x_1 \ldots x_n\ .\ x_i$ (1<= i <=n)

Définies par : $Z(n) = 0 \quad \forall n \in IN$
$S(n) = n+1 \quad \forall n \in IN$
$P^n_i\,(x_1, \ldots, x_n) = x_i \quad \forall\ (x_1, \ldots, x_n) \in IN^n$

Cas particulier : si i = n = 1 nous obtenons la fonction Identité $P^1_1 = \lambda\ x\ .\ x$

II.2) – LES REGLES DE CONSTRUCTIONS DES FONCTIONS :

Se sont des règles qui permettent de construire des fonctions à l'aides d'autres fonctions calculables. Les fonctions ainsi construites deviennent à leur tour calculable.
Il s'agit des trois (03) règles suivantes :
- Règle de COMPOSITION
- Règle de RECURSION
- Règle de MINIMISATION

II.2.1) - Règle de COMPOSITION :

Définition :
Soient H une fonction d'arité m et G1, …, Gm m fonctions d'arité n. La fonction F d'arité n est dite construite par composition à partir des fonctions H et G1, …, Gm si F est définie de la manière suivante :

$$\mathbf{F = \lambda\ x_1 \ldots x_n\ .\ H\ (\ G1\ (x_1 \ldots x_n), \ldots, Gm\ (x_1 \ldots x_n\ )\ )}$$

Notation :
H (G1 ($x_1 \ldots x_n$), …, Gm ($x_1 \ldots x_n$)) est noté H o (G1, …, Gm) ($x_1 \ldots x_n$)
C'est une composition généralisée.
Cas particulier : m = 1
F = $\lambda\, x_1 \ldots x_n$. H (G ($x_1 \ldots x_n$)) , nous retrouvons la composition simple H o G

II.2.2) - Règle de RECURSION :

Définition :
Soient G une fonction d'arité n et H une fonction d'arité n+2. La fonction F d'arité (n+1) est dite construite par récursion à partir des fonctions G et H si $\forall$ ($x_1, \ldots, x_n$) $\in$ IN^n

$$\mathbf{F(x_1, \ldots, x_n, 0) = G(x_1, \ldots, x_n)}$$
$$\mathbf{F(x_1, \ldots, x_n, y+1) = H(x_1, \ldots, x_n, F(x_1, \ldots, x_n, y), y)}$$

Cas particuliers :
n=1 : F(x, 0) = G (x)
F(x, y+1) = H (x, F(x, y), y)

n=0 : F(0) = Cst
F(y+1) = H (F(y), y)
où Cst est une constante entière considérée comme une fonction d'arité 0 appelée fonction constante.

Convention :
Les fonctions constantes sont des fonctions calculables.

II.2.3) - Règle de MINIMISATION :

Définition :
Soit G une fonction d'arité n+1 telle que pour tout ($x_1, \ldots, x_n$)$\in$ IN^n il existe au moins un y $\in$ IN tel que : G ($x_1, \ldots, x_n$, y) = 0 .
La fonction F d'arité n est dite construite par minimisation à partir de la fonction G si F est définie da la manière suivante :

$$\mathbf{F = \lambda\, x_1 \ldots x_n . [\ Minus\ y\ /\ G(x_1, \ldots, x_n, y) = 0\ ]}$$

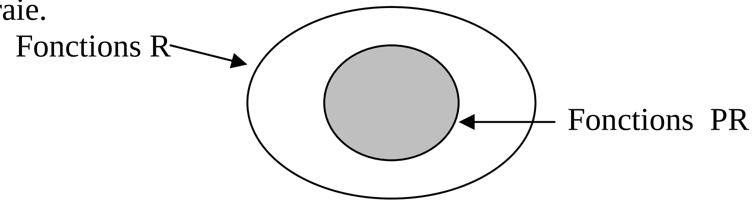
où
Minus y / G ($x_1, \ldots, x_n$, y) = 0 représente le plus petit entier y tel que G ($x_1, \ldots, x_n$, y) = 0 .

II .3) – LES FONCTIONS PRIMITIVES RECURSIVES ET LES FONCTIONS RECURSIVES :

Définition 1 :
Une fonction est dite PRIMITIVE RECURSIVE (PR) si elle est :
- soit une fonction de base,
- soit une fonction construite à partir d'autres fonctions PR au moyen des règles de Composition et/ou de Récursion.

Définition 2 :
Une fonction est dite RECURSIVE (R) si elle est :
- soit une fonction de base,
- soit une fonction construite à partir d'autres fonctions R au moyen des règles de Composition et/ou de Récursion et/ou de Minimisation.

Remarque 1 :
Toute fonction PRIMITIVE RECURSIVE est RECURSIVE mais la réciproque n'est pas vraie.



Remarque 2 :
Plusieurs constructions différentes peuvent être associées à une même fonction.

Remarque 3 :
Une fonction est PRIMITIVE RECURSIVE si elle est définie sans utiliser la règle de minimisation. Mais dire qu'une fonction RECURSIVE n'est pas PRIMITIVE RECURSIVE signifie que toutes tentative de construction de cette fonction fait intervenir la règle de minimisation.

EXEMPLE :
Montrons que la fonction $\oplus$ d'arité 2 définie par : $\oplus = \lambda\, x\, y\, .\, x+y$ est PR.

Pour ce faire nous allons procéder à une construction en utilisant la règle de récursion :

$\oplus\ (x , 0) = x + 0 = x = P^1_1\ (x)$

$\oplus\ (x , y+1) = x + (y+1) = (x+y) + 1 = \oplus\ (x , y) + 1 = S\ (\oplus\ (x , y))$
$= S\ (\ P^3_2\ (x , \oplus\ (x , y) , y\ )\ )$
$= (\ S \circ P^3_2\ )\ (x , \oplus\ (x , y) , y\ )$

Nous avons donc : $G = P^1_1$ est PR car c'est fonction de base.
$H = (\ S \circ P^3_2\ )$ est PR car c'est une composition de 2 fonctions de base.
La fonction $\oplus$ est construite par récursioin à partir de 2 fonctions PR , à savoir G et H, donc $\oplus$ est PR.

Définition 3 (Equivalente) :
Une fonction F est PRIMITIVE RECURSIVE (Respectivement : RECURSIVE) s'il existe une suite finie de fonctions $F_1 , \dots , F_n$ telles que :
- $F_n = F$
- Et pour chaque $i \in \{1 , \dots, n\}$

- soit F_i est une fonction de base,
- soit il a été déjà prouvé que F_i est PRIMITIVE RECURSIVE (Respectivement : RECURSIVE)

- soit F_i est définie à partir de certaines des fonctions F_1 , … , F_{i-1} au moyen de la règle de composition ou de récursion (<u>Respectivement</u> : composition ou de récursion ou de minimisation).

La suite des fonctions F_1 , … , F_n est applée une dérivation PRIMITIVE RECURSIVE (<u>Respectivement</u> : RECURSIVE) de F

<u>EXEMPLE</u> :

Montrons que la fonction $\otimes$ d'arité 2 définie par : $\otimes = \lambda\, x\, y\, .\, x+y$ est PR.
Puis donnons une dérivation PRIMITIVE RECURSIVE de $\otimes$.

Pour ce faire nous allons procéder à une construction en utilisant la règle de récursion :

$\otimes (x , 0) = x * 0 = 0 = Z (x)$

$$\begin{aligned} \otimes (x , y+1) &= x * (y+1) = (x*y) + x = \otimes (x , y) + x = \oplus (\otimes (x , y) , x) \\ &= \oplus (P^3_2 (x , \otimes (x , y) , y) , P^3_1 (x , \otimes (x , y) , y)) \\ &= \oplus \text{ o } (P^3_2 , P^3_1) (x , \otimes (x , y) , y) \end{aligned}$$

Nous avons donc : $G = Z$ est PR car c'est fonction de base.
$H = \oplus \text{ o } (P^3_2 , P^3_1)$ est PR car c'est une composition de fonctions PR , P^3_2 et P^3_1 de base et $\oplus$ déjà montrée.

La fonction $\otimes$ est construite par récursioin à partir de 2 fonctions PR , à savoir G et H, donc $\otimes$ est PR.

A partir de cette construction, nous pouvons définir pour la fonction $\otimes$, la dérivation PR suivante :

$F_1 = Z$ fonction de base (arité 1)
$F_2 = \oplus$ PR déjà montrée (arité 2)
$F_3 = P^3_2$ fonction de base (arité 3)
$F_4 = P^3_1$ fonction de base (arité 3)
$F_5 = F_2 \text{ o } (F_3 , F_4)$ composition appliquée à F_2 ,F_3 et F_4 (arité 3)
$F_6 = \otimes$ récursion appliquée à F_1 et F_5 (arité 2)

En conclusion la suite de fonctions F_1 , F_2 , F_3 , F_4 , F_5 , F_6 est une dérivation PR de la fonction $\otimes$.

<u>Remarque</u> :

Nous pouvons étendre les notions de fonctions PRIMITIVES RECURSIVES et de fonctions RECURSIVES aux fonctions de IN^p ---> IN^q avec q>=1.

Pour cela, nous utiliserons le fait qu'une fonction F de IN^p ---> IN^q peut être décomposée en q fonctions F_i de IN^p ---> IN comme le montre le diagramme suivant :

$$\begin{tikzcd} IN^p \arrow[r, "F"] & IN^q \arrow[r, "P^q_i"] & IN \\ (x_1, \dots , x_p) \arrow[r, mapsto] \arrow[rr, bend right, "F_i = P^q_i \text{ o } F"'] & (y_1, \dots , y_q) \arrow[r, mapsto] & y_i \end{tikzcd}$$

Alors F est dite PRIMITIVE RECURSIVE (<u>Respectivement</u> : RECURSIVE) si et seulement si toutes les fonctions de projection $F_i = P^q{}_i \text{ o } F$ ($1 <= i <= q$) sont PRIMITIVES RECURSIVES (<u>Respectivement</u> : RECURSIVES).

<u>EXEMPLE</u> :
Montrons que la fonction F définie comme suit est PR :
F : IN^2 ---------> IN^2
(x , y)--- ---> (x+y , x*y)

soit F_1 la fonction définie par :

$$\begin{tikzcd}
IN^2 \arrow[r, "F"] & IN^2 \arrow[r, "P^2{}_1"] & IN \\
(x , y) \arrow[r, mapsto] \arrow[rr, bend right, "F_1 = P^2{}_1 \text{ o } F"'] & (x+y , x*y) \arrow[r, mapsto] & (x+y) = y1
\end{tikzcd}$$

$F_1 = \oplus$ donc PR.

De même soit F_2 la fonction définie par :

$$\begin{tikzcd}
IN^2 \arrow[r, "F"] & IN^2 \arrow[r, "P^2{}_2"] & IN \\
(x , y) \arrow[r, mapsto] \arrow[rr, bend right, "F_2 = P^2{}_2 \text{ o } F"'] & (x+y , x*y) \arrow[r, mapsto] & (x*y) = y2
\end{tikzcd}$$

$F_2 = \otimes$ donc PR.

Conclusion F_1 et F_2 sont PR donc F est PR

II .4) – <u>RECURSIVITE POUR LES ENSEMBLES</u>

<u>Définition 1</u> :
Soit A un sous ensemble de IN^p, nous appelons fonction CARACTERISTIQUE de A (par rapport à IN^p) , notée Car_A , la fonction :

$$Car_A \quad : \quad IN^p \longrightarrow \{ 0 , 1 \}$$

$$X \longmapsto 1 \quad \text{Si} \quad X \in A$$

$$X \longmapsto 0 \quad \text{Si} \quad X \notin A$$

<u>Définition 2</u> :
Un ensemble A est dit RECURSIF (<u>Respectivement</u> : PREMITIF RECURSIF) si Car_A est RECURSIVE (<u>Respectivement</u> : PREMITIVE RECURSIVE).

Définition 3 (Intuitive):
Un ensemble A de IN^p est dit DECIDABLE si nous pouvons décider quelque soit $x \in IN^p$ si $x \in A$ ou si $x \notin A$.

Proposition :
Si A est un ensemble RECURSIF alors A est un ensemble DECIDABLE.

Indications sur la preuve :
En effet, si un ensemble A est récursif cela signifie que la fonction caractéristique de A, soit Car_A, est elle même récursive. Donc nous savons calculer la valeur $Car_A(x)$ et cela quelque soit $x \in IN^p$. De plus, comme l'ensemble des images de Car_A étant restreint à 2 valeurs :
0 ou 1, nous sommes devant 2 cas :
- ou bien $Car_A(x) = 1$ au quel cas $x \in A$.
- ou bien $Car_A(x) = 0$ au quel cas $x \notin A$.

En conclusion, quelque soit $x \in IN^p$ nous pouvons toujours décider si $x \in A$ ou si $x \notin A$.

Définition 4 :
Un ensemble A est dit EFFECTIVEMENT ENUMERABLE si et seulement s'il existe une fonction F récursive telle que l'ensemble des images de F est l'ensemble A lui même.

Remarque :
D'une manière analogue, nous pouvons étendre la notion de RECURSIVITE aux relations.

Définition 5 :
Soit R une relation définie sur l'ensemble IN^p, nous appelons fonction CARACTERISTIQUE de R (par rapport à IN^p) , notée Car_R , la fonction :

$$Car_R : \quad IN^p \longrightarrow \{0 , 1\}$$

$$X \longmapsto 1 \text{ Si } X \text{ vérifie la relation R}$$

$$X \longmapsto 0 \text{ Si } X \text{ ne vérifie pas la relation R}$$

Définition 6 :
Une relation R est dite RECURSIVE (Respectivement : PREMITIVE RECURSIVE) si Car_R est RECURSIVE (Respectivement : PREMITIVE RECURSIVE).