Il y a plusieurs manières de parler de fonctions calculables et d'ensembles récursifs ou récursivement énumérables. J'ai choisi ici la présentation la plus mathématique possible en passant sous silence les machines de Turing. Cependant il faut avoir conscience que les fonctions calculables sont exactement celles calculables par un ordinateur (la thèse de Church affirme que tous les ordinateurs peuvent calculer les mêmes fonctions et que donc les machines de Turing sont un modèle universel).

1 Fonctions récursives

On s'intéresse ici aux fonctions de $\mathbb{N}^k$ dans $\mathbb{N}$ pour tous les entiers k.

Fonctions élémentaires.

1. Les fonctions constantes ;
2. Les projections de $\mathbb{N}^k$ dans $\mathbb{N}$;

Opérations algébriques. La somme, le produit, etc. (en fait, on peut supprimer cette opération en ajoutant la fonction successeur dans les fonctions élémentaires.)

Composée. Si $f : \mathbb{N}^k \to \mathbb{N}, \quad h_1 : \mathbb{N}^{r_1} \to \mathbb{N}, \ldots, h_k : \mathbb{N}^{r_k} \to \mathbb{N}$ sont des fonctions, on considère la composée $f(h_1, \ldots, h_k)$.

Récurrence primitive. Si $f : \mathbb{N}^{k+2} \to \mathbb{N}$ et $g : \mathbb{N}^k \to \mathbb{N}$ sont des fonctions, on considère la fonction définie par récurrence $h : \mathbb{N}^{k+1} \to \mathbb{N}$ telle que

$$h(\bar{x}, 0) = g(\bar{x})$$

$$h(\bar{x}, n+1) = f(\bar{x}, n, h(\bar{x}, n)).$$

Définition 1.1 *Les **fonctions récursives primitives** forment le plus petit ensemble de fonctions contenant les fonctions récursives élémentaires et clos par les opérations algébriques, la composition et la récurrence primitive.*

Exercice 1. Montrer que toutes les fonctions que vous connaissez sont récursive primitives. Vous pourrez commencer par $n \mapsto 2^n$, $(m, n) \mapsto m^n$, $n \mapsto p_n$ où p_n est le n^{e} nombre premier, $\mathbf{1}_{\text{premier}} : \mathbb{N} \to \{0, 1\}$ la fonction caractéristique des nombres premiers.

Il existe cependant des fonctions, comme la fonction d'Ackermann, qui sont calculables et qui ne sont pas récursives primitives. On introduit donc le schéma de minimisation. La difficulté est qu'il ne définit que des foncitons partielles et en particulier rien ne permet de décider si pour une valeur donnée la fonction est définie ou pas : il faut laisser tourner l'algorithme.

Minimisation. Si $f : \mathbb{N}^{k+1} \to \mathbb{N}$ est une fonction, on considère la fonciton partielle $g : \mathbb{N}^k \to \mathbb{N}$ définie par $g(\bar{x}) = \min\{y \mid f(\bar{x}, y) = 0\}$.

Définition 1.2 *Les **fonctions récursives** forment le plus petit ensemble de fonctions contenant les fonctions élémentaires et clos par les opérations algébriques, la composition, la récurrence primitive et la minimisation.*

Proposition 1.3 *Il existe des fonctions non récursives.*

Démonstration : Il y a un nombre dénombrable de fonctions récursive primitives alors qu'il y a un nombre non-dénombrables de fonctions $\mathbb{N} \to \mathbb{N}$. $\square$

Pour terminer une curiosité. Une des plus grandes fonctions connues est la **fonction d'Ackermann**. Vous pourrez vérifier qu'elle est récursive, que pour tout entier m, $A(m, .)$ est récursive primitive mais que A n'est pas récursive primitive. En fait on peut montrer que la fonction d'Ackermann majore toutes les fonctions récursives primitives. Essayez de calculer les plus petites valeurs de A (attention $\mathrm{Ack}(4, 4)$ est plus grand que le nombre d'atomes dans l'univers).

La fonction d'Ackermann est définie par :

$$\mathrm{Ack}(0, n) = n + 1$$

$$\mathrm{Ack}(m + 1, 0) = \mathrm{Ack}(m, 1)$$
$$\mathrm{Ack}(m + 1, n + 1) = \mathrm{Ack}(m, \mathrm{Ack}(m + 1, n)).$$

2 Ensembles récursifs et récursivement énumérables

Définition 2.1 *Une partie A de $\mathbb{N}^k$ est **récursive** si sa fonction caractéristique est récursive.*

*Si f est récursive et totalement définie alors on dit que $A = Imf$ est **récursivement énumérable**.*

(Notons qu'on dit qu'une fonction $f : \mathbb{N}^k \mapsto \mathbb{N}^l$ est récursive si les l fonctions coordonnées le sont. La définition ci-dessus s'applique donc aux parties de $\mathbb{N}^k$.)

Il faut penser aux parties récursives comme celles pour lesquelles il existe un algorithme qui décide si un élément est dans cette partie ou non.

Un ensemble est récursivement énumérable si il existe un algorithme qui, si on lui donne un élément, termine en temps fini si l'élément est dans la partie et sinon boucle indéfiniment (on parle d'algorithme de semi-décision). Cette idée de semi-décision se formalise :

Proposition 2.2 *Une partie A est récursive si et seulement si A et son complémentaire A^C sont récursivement énumérables.*

Démonstration : L'idée est que si on a deux algorithmes de semi-décision pour A et A^C alors en les exécutant en parallèle pour tout élément on aura une réponse positive ou négative, en temps fini, par l'un ou l'autre des deux algorithmes. $\square$

Bien sûr, toutes les parties de $\mathbb{N}$ auxquelles vous pouvez penser (nombres premiers, etc.) sont récursives. Pourtant comme pour les fonctions récursive primitives, un argument de cardinal montre qu'il y a des parties de $\mathbb{N}$ non récursivement énumérables.

3 Et le groupe libre ?

Toutes les notions que nous avons définies ci-dessus l'ont été dans le cadre des entiers (ou des uplets d'entiers). Les mots d'un groupe libre F peuvent être rangés par ordre alphabétique ce qui définit une bijection entre F et $\mathbb{N}$. On parle de **codage**. On peut alors transporter les notions de fonctions récursives ou de parties récursives ou récursivement énumérable dans F. On se convaincra que ces notions ne dépendent pas des codages choisis pour peu que ceux-ci soient raisonnables.