INFORMATION, CONCEPTS ET ENSEMBLES

A Les ensembles et la logique

Comme mentionné dans l'introduction de l'informatique commune, l'informatique est la construction de l'information par le calcul. Mais comment peut-on calculer l'information, au-delà de l'information purement numérique ? Calculer sur les entiers est une chose, calculer un information en général en est une autre.

Le deux pierres angulaires aux fondements de l'informatique sont la théorie des ensembles et la logique. La fin du XIX^{e} siècle a marqué un tournant dans l'histoire des sciences et en particulier pour les mathématiques : la théorie des ensembles et ses paradoxes ont forcé les mathématiciens à mieux formaliser les raisonnements et les démonstrations, c'est-à-dire la logique.

Lorsque l'homme analyse l'information, il fait des regroupements : regrouper les informations d'un même type permet de trouver des propriétés à l'ensemble, de les manipuler par lot, de leur appliquer un même traitement : c'est une abstraction très efficace pour le calcul.

■ **Exemple 1 — Les ingrédients d'une recette de cuisine.** Les ingrédients d'une recette de cuisine forment un concept intéressant : on peut les classer (par texture, goût, couleur), les cuire, les mesurer, les combiner. Et même si un ingrédient est un concept qui n'a pas de réalité [a], les ingrédients forment un ensemble intéressant car on peut le construire : avec de la moutarde, un jaune d'œuf, du sel, du poivre et un mélange adapté, on construit une mayonnaise. La mayonnaise elle-même est un ingrédient : on peut l'utiliser pour élaborer d'autres ingérdients comme une salade de légumes par exemple. Cette salade pourra être incorporée dans un wrap... On en déduit qu'un ingrédient est lui-même un ingrédient, c'est un type récursif ! Ce type d'ensemble est nommé **ensemble inductif** et, si on prend la peine de réfléchir au monde qui nous entoure, il est très présent dans notre réalité.

a. un œuf n'existe pas, mais l'œuf qu'une poule a pondu sous vos yeux existe... De la même manière, une voiture est une abstraction : il n'existe que des modèles construits comme la 206 GTI ou la corolla break 2021 hybride.

B Ensembles inductifs

Les **ensembles inductifs** sont à la base du développement de l'informatique. C'est pourquoi ils irriguent toutes les parties du programme officiel de l'option informatique et donc tous les développement de ce cours. Ils ont été formalisés dans la théorie des types par Russel au début du XX^{e} siècle, ouvrant ainsi la voie aux systèmes formels. Pour ces ensembles inductifs, on dispose de **types de base** et de **constructeurs** qui permettent de créer des types à partir

d'autres types. Tout objet d'un ensemble inductif est d'un certain type et il existe un ordre lié à la construction de l'ensemble. C'est ce qui différencie cette approche de la théorie des ensembles. Le programme de l'option informatique aborde des structures de données qui sont des ensembles inductifs en première et deuxième année dont les listes chaînées, les arbres, formules logiques, les expressions régulières.

■ **Exemple 2 — Définir l'ensemble des ingrédients de manière inductive.** Pour nos ingrédients, les termes de base pourraient être le sel, le poivre, l'eau, la farine, le beurre, le sucre, le jaune et le blanc d'œuf. Les règles de construction : prendre, mélanger, étaler, broyer, fondre, cuire.

Formellement, on pourrait définir l'ensemble des ingrédients $\mathcal{I}$ de manière inductive de la façon suivante :

Termes de base $\mathcal{B} = \{\text{sel, poivre, eau, farine, beurre, sucre, jaune d'œuf, blanc d'œuf}\}$

Constructeur (FONDRE) $\forall i \in \mathcal{I}$, FONDRE $(i) \in \mathcal{I}$

Constructeur (MÉLANGER) $\forall i_1, i_2 \in \mathcal{I}$, MÉLANGER $(i_1, i_2) \in \mathcal{I}$

En appliquant le constructeur FONDRE à BEURRE, on obtient FONDRE(BEURRE), du beurre fondu. En appliquant les constructeurs :
MÉLANGER(SUCRE, FONDRE(BEURRE)),
on obtient un nouvel ingrédient qui nous permettra de créer un Kouign-amann.

R Les éléments d'un ensemble inductif sont appelés **termes**.

R Un constructeur possède une certaine **arité** : il permet de construire un terme à partir de un ou plusieurs termes. Dans l'exemple des ingrédients, FONDRE est un constructeur d'arité 1 car il ne prend qu'un seul paramètre. MÉLANGER en prend deux, il est d'arité 2. On ne peut donc pas mélanger du beurre, mais on peut mélanger du beurre et du sucre.

R On observe qu'il y a un **ordre dans les termes** que l'on peut construire à partir de la définition inductive des ingrédients : par exemple, la mayonnaise est un ingrédient nécessaire à l'élaboration d'un sandwich au crabe et le sel est nécessaire à l'élaboration de la mayonnaise. On dit que le sel est un sous-terme de la mayonnaise et que la mayonnaise est un sous-terme du sandwich au crabe. D'une manière générale, $t = \text{CONSTRUCTEUR}(t_1, t_2, \ldots, t_n)$ signifie que l'on construit le terme t à partir des sous-termes $t_1, t_2, \ldots, t_n$. **Cet ordre est appelé ordre structurel.**

C Propriétés des ensembles inductifs

Les ensembles inductifs ont des propriétés qu'il est possible de démontrer grâce à l'**induction structurelle**.

■ **Définition 1 — Principe d'induction structurelle.** Soit $\mathcal{I}$ un ensemble inductif de termes de base $\mathcal{B}$ et de constructeurs $\mathcal{C}$. Soit $\mathcal{P}$ une propriété sur les termes de $\mathcal{I}$.

Si $\mathcal{P}$ est satisfaite pour chaque terme de base de $\mathcal{B}$ et pour chaque constructeur de $\mathcal{I}$, alors $\mathcal{P}$ est satisfaite pour tous les termes de $\mathcal{I}$.

Plus formellement,

$$\left.\begin{array}{ll} \forall b \in \mathcal{B}, & \mathcal{P}(b) \\ \forall c \in \mathcal{C}, \forall t_1, t_2, \dots t_n \in \mathcal{I}, & \mathcal{P}(c(t_1, t_2, \dots, t_n)) \end{array}\right\} \Longrightarrow \forall t \in \mathcal{I}, \mathcal{P}(t) \tag{1}$$

■ **Exemple 3 — Comestible.** Soit $\mathcal{I}$ l'ensemble inductif des ingrédients. Soit $\mathcal{P}$ la propriété *est comestible*. On cherche à montrer que tous les ingrédients sont comestibles par induction structurelle [a].

On sait que tous les ingrédients de base sont comestibles. Par ailleurs, faire fondre un ingrédient ne dégrade pas cette propriété, il est toujours comestible une fois fondu. Enfin, lorsqu'on mélange deux ingrédients comestibles, le résultat est comestible. C'est pourquoi, tous les ingrédients sont comestibles.

a. Attention, il ne s'agit que d'une modélisation des ingrédients limitée à des objets de base comestibles. Ne pas extrapoler le résultats ;-)

D Des fonctions pour calculer sur les termes d'un ensemble inductif

On peut facilement définir des fonctions sur les ensembles inductifs en s'appuyant sur leur définition. Cela permet de faire des calcul sur l'information qu'ils représentent.

■ **Exemple 4 — Masse d'un ingrédient.** On peut définir la masse d'un ingrédient en s'appuyant sur la définition inductive d'un ingrédient : la masse d'un ingrédient est la somme des masses des ingrédients qui la compose. Pour les ingrédients de base, on peut considérer que la masse d'un ingrédient $b \in \mathcal{B}$ est une constante connue m_b que l'on a mesurée.

On obtient alors une fonction formulée récursivement :

Cas de base $\forall b \in \mathcal{B}, \text{MASSE}(b) = m_b$

Construction $\forall i \in \mathcal{I}, \text{MASSE}(\text{FONDRE}(i)) = \text{MASSE}(i)$

Construction $\forall i_1, i_2 \in \mathcal{I}, \text{MASSE}(\text{MÉLANGER}(i_1, i_2)) = \text{MASSE}(i_1) + \text{MASSE}(i_2)$

En utilisant cette définition inductive, on peut calculer la masse de tous les ingrédients.

E Pourquoi OCaml ?

Comme on peut le voir ci-dessous et comme on le verra dans la section suivante, le langage OCaml est particulièrement adapté aux ensembles inductifs pour plusieurs raisons :

1. les types en OCaml sont nativement récursifs,
2. les types OCaml peuvent être des types somme | ou produit *. On les appelle des types algébriques.

3. OCaml procure la syntaxe dite du filtrage de motifs,
4. OCaml permet d'écrire des fonctions récursives.

Combiner ces quatre fonctionnalités permet de traduire directement la plupart des concepts mathématiques exprimés sous la forme d'un ensemble inductif. Les **fonctions** dont les paramètres peuvent être des types algébriques concrétisent les calculs sur les termes des ensembles inductifs.

Ces fonctionnalités de OCaml [1] expliquent en grande partie le choix du langage OCaml en CPGE pour l'option informatique.

Vocabulary 1 — Pattern matching ⇜⇝ Filtrage de motifs

```
type ingredient =
        | Sel | Poivre | Beurre | Sucre | Farine
        | Fondre of ingredient
        | Melanger of ingredient*ingredient;;

let rec masse = function
        | Sel -> 30
        | Poivre -> 10
        | Beurre -> 250
        | Sucre -> 250
        | Farine  -> 10
        | Fondre i ->  masse i
        | Melanger (i1,i2) -> (masse i1) + (masse i2);;

let kouignamann = Melanger(Beurre, Sucre);;
let m = masse(kouignamann);;
```

1. Tous les langages ne dispose pas de ces fonctionnalités : Python ne dispose pas de types récursifs et algébriques et son paradigme est impératif, pas fonctionnel.