

# Travail Pratique : Examen 2023

Rémi Cozot, 15 octobre 2023

## Logique/algèbre des intervalles D'Allen (1983)

Le but est d'écrire un ensemble de fonctionnalité en python et en F#

Vous trouverez des informations sur la logique d'Allen sur :

[https://fr.wikipedia.org/wiki/Alg%C3%A8bre\\_des\\_intervalles\\_d%27Allen](https://fr.wikipedia.org/wiki/Alg%C3%A8bre_des_intervalles_d%27Allen)

### Relation directe entre deux intervalles X et Y

1. **X Before Y**
2. **X Meets Y**
3. **X Overlaps Y**
4. **X Starts Y**
5. **X During Y**
6. **X Finishes Y**
7. **X Equal Y**

## Travail demandé

Le code devra être fourni en 2 versions : **python** et **F#**

### Représentation des intervalles

Vous prendrez en compte 4 représentations d'un intervalle :

1. Nom (chaîne de caractères) et Couple start et end
2. Nom (chaîne de caractères) et Couple start et length
3. Nom (chaîne de caractères) et Couple end et length
4. Nom (chaîne de caractères) et Couple center et half length

### Fonctions logiques

Vous codez les 7 fonctions logiques (retourne un booléen) : isXXX avec XXX dans Before, Meets, Overlaps, Starts, During, Finishes, Equal. Exemple :

```
isDuring(("a", startEnd (3,4)) ,("b", startLength (2, 3))) → true
```

### Relations logiques

Vous codez un arbre binaire (qui aura une profondeur de 1 : opérateur de relation logique et 2 fils des intervalles) permettant d'encoder une relation entre 2 intervalles, par exemple (**Equal**, "x", "y") avec "x", "y" le nom de deux intervalles.

### Interrogation

Vous écrivez une fonction qui renvoie une liste de relations logiques pour un intervalle X par rapport à une liste d'intervalle donné en paramètre, par exemple :

```
Relation(("x", startEnd (5, 7)), [("a", startEnd (0,3)) ,("b", startEnd (2, 5)), ("c", startEnd (5,7))]) → [(Before, "a", "x"),(Meets, "b", "x") ,(Equal, "x", "c")]
```