

FOUINE

Benjamin Duhamel & Ulysse Durand

La fouine est un interpréteur OCaml qui a pour but de faire fouiner ses créateurs dans l'implémentation d'un langage fonctionnel.

Exemple de code

```
let rec f x n = if n = 0 then x else f (match x with  
| (y,z) -> (z, y+z)) (n-1) in f (0,1) 10
```

est un code OCaml qui permettrait de calculer le dixième terme
de la suite de fibonacci

C'est aussi un code fouine

Les expressions

On traduit le code en un arbre d'expression

Exemple avec `if true then (3+4)*7 else false`

Implémentation des références

On a une mémoire qui est un long array, l'adressage mémoire est naïf.

Implémentation des fonctions

Pour une fonction représentée par $\text{VFun}(x, e, \text{env})$,
appliquée à une expression $e2$, on évalue $e2$ en ajoutant à
 env que x vaut e

Pour les fonctions récursive

La représentation est différente, `VFunRec (f, e, env)`

On fait de même mais en "déroulant" l'environnement `env`, c'est à dire en remettant ce même `env` dans les environnements des `VFunRec` de `env`.

Exemple avec

```
let rec f n = if n = 0 then 1 else n * (f (n-1)) in
f 10
```

Implémentation des boucles (While et For)

Plutôt simple

Implémentation des couples et listes (avec leur matching)

Plutôt compliqué