

# Sémantique des langages de programmation

## TD n° 3

Semestre 5, 2024-25

### 1 MiniML et let in

On souhaite ajouter une construction de définitions locales à notre langage MiniML :  
`let x = e in e'`

1. Étendre les sémantiques de MiniML à grands et petits pas, par nom et par valeur.
2. A-t-on l'équivalence suivante : `let x = t1 in t2`  $\equiv$  `(fun x -> t2) t1` ?

### 2 MiniM et couples

On étend la syntaxe du langage MiniML vu en cours avec le possibilité de créer des couples et de récupérer les membres d'un couple (projections) :

$$e ::= n \mid x \mid \text{fun } x \rightarrow e \mid e e \mid e + e \mid (e, e) \mid \text{fst}(e) \mid \text{snd}(e)$$

1. Étendre la sémantique à grand pas par valeur de MiniML pour donner une sémantique aux couples et projections. Informellement, un couple d'expressions est évalué en le couple des valeurs des expressions. La première projection (`fst(e)`) d'un couple s'évalue comme le membre gauche du couple, la deuxième (`snd(e)`) comme le membre droit.
2. Comment la définition des valeurs doit-elle être étendue ?
3. Donner la sémantique à grand pas par valeur des expressions suivantes :

$$\begin{aligned} & (1 + 2, (\text{fun } x \rightarrow 3) 4) \\ & (\text{fun } f \rightarrow (f 1, f (\text{fun } x \rightarrow x))) (\text{fun } y \rightarrow y) \\ & \text{snd}((\text{fun } x \rightarrow (x, x)) 2) \\ & (\text{fun } c \rightarrow (c, \text{fst}(c))) (3 + 1, 8) \end{aligned}$$

4. Quelle condition (syntaxique) sur l'expression  $e$  implique l'équivalence des programmes  $e$  et  $(\text{fst}(e), \text{snd}(e))$  ? Détaillez votre réponse.
5. Étendre la sémantique à petit pas par valeur de MiniML pour donner une sémantique aux couples et projections.
6. Démontrer l'équivalence des sémantiques opérationnelles à grands pas et à petit pas. On pourra se contenter d'exposer les nouveaux cas.