Introduction à la sémantique des langages de programmation

Adrien Guatto

Compilation M1

Ce cours et les deux suivants

Quoi ?

- Mathématiser des concepts essentiels des langages de programmation.
- Implémenter ces concepts en OCaml, le cas échéant.

Pourquoi ?

- Acquérir les notions et la terminologie de base du domaine.
- Comprendre les sujets des prochains jalons du projet.
- Se préparer à vos futurs cours de sémantique (par exemple, au S2).

Quel rôle pour Marthe ?

- Beaucoup plus simple qu'OCaml, Java, Python, ou le λ-calcul.
- Exhibe pourtant certaines difficultés caractéristiques.
- Peut être progressivement étendu en un langage plus riche.

Cours 1

Syntaxe et sémantique de Marthe

Contexte : les compilateurs

Un compilateur traduit un langage source vers un langage cible.

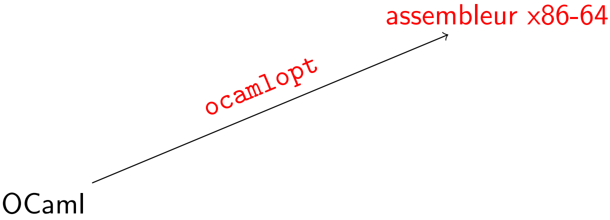
Contexte : les compilateurs

Un compilateur traduit un langage source vers un langage cible.

OCaml

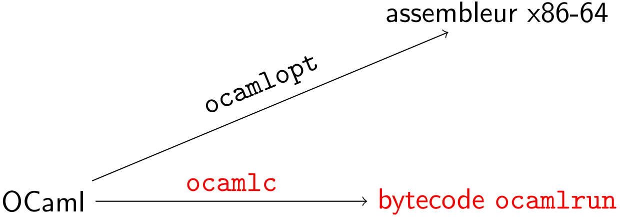
Contexte : les compilateurs

Un compilateur traduit un langage source vers un langage cible.



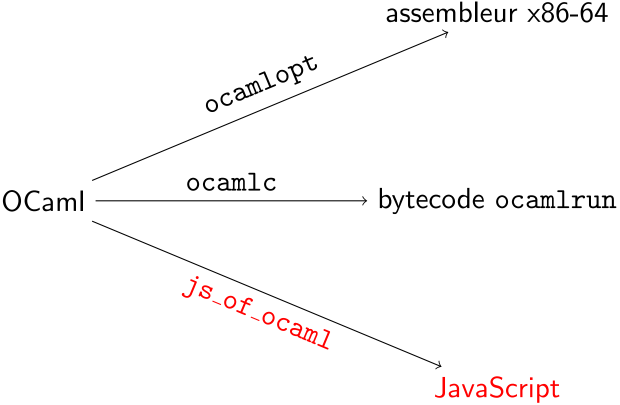
Contexte : les compilateurs

Un compilateur traduit un langage source vers un langage cible.



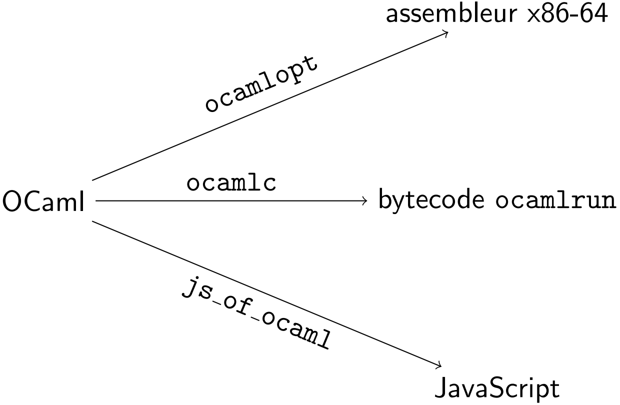
Contexte : les compilateurs

Un compilateur traduit un langage source vers un langage cible.



Contexte : les compilateurs

Un compilateur traduit un langage source vers un langage cible.



Un compilateur doit avant tout être correct. Qu'est-ce que cela signifie ?

Correction des compilateurs, dans l'abstrait

Quelques notations :

- $|\mathcal{S}|$ désigne l'ensemble des *termes* du langage source,
- $|\mathcal{T}|$ désigne l'ensemble des termes du langage cible,
- $C : |\mathcal{S}| \to |\mathcal{T}|$ désigne le compilateur, fonction de $|\mathcal{S}|$ dans $|\mathcal{T}|$.

Intuitivement, on aimerait adopter une définition ressemblant à :

$$C \text{ est correct} \overset{\text{def}}{=} \forall M \in |\mathcal{S}|, M \text{ et } C(M) \text{ ``font la même chose''}.$$

Correction des compilateurs, dans l'abstrait

Quelques notations :

- $|\mathcal{S}|$ désigne l'ensemble des *termes* du langage source,
- $|\mathcal{T}|$ désigne l'ensemble des termes du langage cible,
- $C : |\mathcal{S}| \to |\mathcal{T}|$ désigne le compilateur, fonction de $|\mathcal{S}|$ dans $|\mathcal{T}|$.

Intuitivement, on aimerait adopter une définition ressemblant à :

$$C \text{ est correct} \stackrel{\text{def}}{=} \forall M \in |\mathcal{S}|, M \text{ et } C(M) \text{ “font la même chose”}.$$

C'est un peu vague. Essayons :

$$C \text{ est correct} \stackrel{\text{def}}{=} \forall M \in |\mathcal{S}|, M \text{ et } C(M) \text{ ont le } \textcolor{red}{\text{même résultat}}.$$

Correction des compilateurs, dans l'abstrait

Quelques notations :

- $|\mathcal{S}|$ désigne l'ensemble des *termes* du langage source,
- $|\mathcal{T}|$ désigne l'ensemble des termes du langage cible,
- $C : |\mathcal{S}| \to |\mathcal{T}|$ désigne le compilateur, fonction de $|\mathcal{S}|$ dans $|\mathcal{T}|$.

Intuitivement, on aimerait adopter une définition ressemblant à :

$$C \text{ est correct} \stackrel{\text{def}}{=} \forall M \in |\mathcal{S}|, M \text{ et } C(M) \text{ “font la même chose”}.$$

C'est un peu vague. Essayons :

$$C \text{ est correct} \stackrel{\text{def}}{=} \forall M \in |\mathcal{S}|, M \text{ et } C(M) \text{ ont le même résultat.}$$

Chaque langage $L \in \{\mathcal{S}, \mathcal{T}\}$ doit donc définir le résultat $R_L(M)$ de tout terme $M \in |L|$. On obtient donc, formellement :

$$C \text{ est correct} \stackrel{\text{def}}{=} \forall M \in |\mathcal{S}|, R_{\mathcal{S}}(M) = R_{\mathcal{T}}(C(M)).$$

La sémantique de Marthe (1/3)

Comment définir mathématiquement le langage Marthe, noté $\mathcal{M}$?

La sémantique de Marthe (1/3)

Comment définir mathématiquement le langage Marthe, noté $\mathcal{M}$?

- $|\mathcal{M}|$ est l'ensemble des arbres de syntaxe abstraite de Marthe, décrits sous la forme d'une grammaire BNF au premier cours.

$$|\mathcal{M}| \ni M, N, P ::= x \mid \underline{n} \mid M \mathbin{\underline{+}} N \mid M \mathbin{\underline{*}} N \mid \underline{\textstyle\sum}_{x=M}^{N} P$$

La sémantique de Marthe (1/3)

Comment définir mathématiquement le langage Marthe, noté $\mathcal{M}$?

- $|\mathcal{M}|$ est l'ensemble des arbres de syntaxe abstraite de Marthe, décrits sous la forme d'une grammaire BNF au premier cours.

$$|\mathcal{M}| \ni M, N, P ::= x \mid \underline{n} \mid M \underline{+} N \mid M \underline{*} N \mid \underline{\sum}_{x=M}^{N} P$$

- Pour définir $R_{\mathcal{M}}$, on va reformuler notre interprète écrit en OCaml sous la forme d'un ensemble de triplets appelé *Eval*.

$$(M, \sigma, n) \in \mathit{Eval} \Leftrightarrow \text{“}M \text{ s'évalue en } n \text{ dans l'environnement } \sigma\text{”}$$

La sémantique de Marthe (1/3)

Comment définir mathématiquement le langage Marthe, noté $\mathcal{M}$?

- $|\mathcal{M}|$ est l'ensemble des arbres de syntaxe abstraite de Marthe, décrits sous la forme d'une grammaire BNF au premier cours.

$$|\mathcal{M}| \ni M, N, P ::= x \mid \underline{n} \mid M \mathbin{\underline{+}} N \mid M \mathbin{\underline{*}} N \mid \underline{\Sigma}_{x=M}^{N} P$$

- Pour définir $R_{\mathcal{M}}$, on va reformuler notre interprète écrit en OCaml sous la forme d'un ensemble de triplets appelé *Eval*.

$$(M, \sigma, n) \in \mathit{Eval} \Leftrightarrow \text{“}M \text{ s'évalue en } n \text{ dans l'environnement } \sigma\text{”}$$

- En OCaml, un environnement est une liste de couples variable/entier. Et en sémantique ? Un choix commode : les fonctions partielles finies.

$$\mathit{Env} \stackrel{\text{def}}{=} \{\sigma : \mathit{Var} \rightarrow \mathbb{N} \mid \qquad\qquad\qquad \}$$

La sémantique de Marthe (1/3)

Comment définir mathématiquement le langage Marthe, noté $\mathcal{M}$?

- $|\mathcal{M}|$ est l'ensemble des arbres de syntaxe abstraite de Marthe, décrits sous la forme d'une grammaire BNF au premier cours.

$$|\mathcal{M}| \ni M, N, P ::= x \mid \underline{n} \mid M \mathbin{\underline{+}} N \mid M \mathbin{\underline{*}} N \mid \underline{\sum}_{x=M}^{N} P$$

- Pour définir $R_{\mathcal{M}}$, on va reformuler notre interprète écrit en OCaml sous la forme d'un ensemble de triplets appelé *Eval*.

$$(M, \sigma, n) \in \mathit{Eval} \Leftrightarrow \text{“}M \text{ s'évalue en } n \text{ dans l'environnement } \sigma\text{”}$$

- En OCaml, un environnement est une liste de couples variable/entier. Et en sémantique ? Un choix commode : les fonctions partielles finies.

$$\mathit{Env} \stackrel{\text{def}}{=} \{\sigma : \mathit{Var} \rightharpoonup \mathbb{N} \mid \qquad\qquad\qquad \}$$

La sémantique de Marthe (1/3)

Comment définir mathématiquement le langage Marthe, noté $\mathcal{M}$?

- $|\mathcal{M}|$ est l'ensemble des arbres de syntaxe abstraite de Marthe, décrits sous la forme d'une grammaire BNF au premier cours.

$$|\mathcal{M}| \ni M, N, P ::= x \mid \underline{n} \mid M \mathbin{\underline{+}} N \mid M \mathbin{\underline{*}} N \mid \underline{\sum}_{x=M}^{N} P$$

- Pour définir $R_{\mathcal{M}}$, on va reformuler notre interprète écrit en OCaml sous la forme d'un ensemble de triplets appelé *Eval*.

$$(M, \sigma, n) \in \mathit{Eval} \Leftrightarrow \text{“}M \text{ s'évalue en } n \text{ dans l'environnement } \sigma\text{”}$$

- En OCaml, un environnement est une liste de couples variable/entier. Et en sémantique ? Un choix commode : les fonctions partielles finies.

$$\mathit{Env} \stackrel{\text{def}}{=} \{\sigma : \mathit{Var} \rightharpoonup \mathbb{N} \mid \sigma^{-1}(\mathbb{N}) \text{ est fini}\}$$

La sémantique de Marthe (1/3)

Comment définir mathématiquement le langage Marthe, noté $\mathcal{M}$?

- $|\mathcal{M}|$ est l'ensemble des arbres de syntaxe abstraite de Marthe, décrits sous la forme d'une grammaire BNF au premier cours.

$$|\mathcal{M}| \ni M, N, P ::= x \mid \underline{n} \mid M \mathbin{\underline{+}} N \mid M \mathbin{\underline{*}} N \mid \underline{\Sigma}_{x=M}^{N} P$$

- Pour définir $R_{\mathcal{M}}$, on va reformuler notre interprète écrit en OCaml sous la forme d'un ensemble de triplets appelé *Eval*.

$$(M, \sigma, n) \in \mathit{Eval} \Leftrightarrow \text{“}M \text{ s'évalue en } n \text{ dans l'environnement } \sigma\text{”}$$

- En OCaml, un environnement est une liste de couples variable/entier. Et en sémantique ? Un choix commode : les fonctions partielles finies.

$$\mathit{Env} \stackrel{\text{def}}{=} \{\sigma : \mathit{Var} \rightharpoonup \mathbb{N} \mid \sigma^{-1}(\mathbb{N}) \text{ est fini}\}$$

- Le résultat d'un terme Marthe est l'ensemble des entiers vers lesquels il s'évalue dans l'environnement vide :

$$R_{\mathcal{M}}(M) \stackrel{\text{def}}{=} \{n \in \mathbb{N} \mid (M, \emptyset, n) \in \mathit{Eval}\}.$$

La sémantique de Marthe (2/3)

Que doit contenir notre ensemble de triplets $\mathit{Eval} \subseteq |\mathcal{M}| \times \mathit{Env} \times \mathbb{N}$?

La sémantique de Marthe (2/3)

Que doit contenir notre ensemble de triplets $\mathit{Eval} \subseteq |\mathcal{M}| \times \mathit{Env} \times \mathbb{N}$?

- Le terme $\underline{n}$ doit s'évaluer en n dans tout σ.

La sémantique de Marthe (2/3)

Que doit contenir notre ensemble de triplets $\mathit{Eval} \subseteq |\mathcal{M}| \times \mathit{Env} \times \mathbb{N}$?

- Le terme $\underline{n}$ doit s'évaluer en n dans tout σ.

$$\{(\underline{n}, \sigma, n) \mid n \in \mathbb{N}, \sigma \in \mathit{Env}\} \subseteq \mathit{Eval}$$

La sémantique de Marthe (2/3)

Que doit contenir notre ensemble de triplets $\mathit{Eval} \subseteq |\mathcal{M}| \times \mathit{Env} \times \mathbb{N}$?

- Le terme $\underline{n}$ doit s'évaluer en n dans tout σ.

$$\{(\underline{n}, \sigma, n) \mid n \in \mathbb{N}, \sigma \in \mathit{Env}\} \subseteq \mathit{Eval}$$

- Le terme x doit s'évaluer en v dans σ **si** $\sigma(x) = v$.

La sémantique de Marthe (2/3)

Que doit contenir notre ensemble de triplets $\mathit{Eval} \subseteq |\mathcal{M}| \times \mathit{Env} \times \mathbb{N}$?

- Le terme $\underline{n}$ doit s'évaluer en n dans tout σ.

$$\{(\underline{n}, \sigma, n) \mid n \in \mathbb{N}, \sigma \in \mathit{Env}\} \subseteq \mathit{Eval}$$

- Le terme x doit s'évaluer en v dans σ **si** $\sigma(x) = v$.

$$\{(x, \sigma, \sigma(x)) \mid x \in \mathit{Var}, \sigma \in \mathit{Env}\} \subseteq \mathit{Eval}$$

La sémantique de Marthe (2/3)

Que doit contenir notre ensemble de triplets $\mathit{Eval} \subseteq |\mathcal{M}| \times \mathit{Env} \times \mathbb{N}$?

- Le terme $\underline{n}$ doit s'évaluer en n dans tout σ.

$$\{(\underline{n}, \sigma, n) \mid n \in \mathbb{N}, \sigma \in \mathit{Env}\} \subseteq \mathit{Eval}$$

- Le terme x doit s'évaluer en v dans σ **si** $\sigma(x) = v$.

$$\{(x, \sigma, \sigma(x)) \mid x \in \mathit{Var}, \sigma \in \mathit{Env}\} \subseteq \mathit{Eval}$$

- Le terme $M \underline{+} N$ doit s'évaluer en $m + n$ dans σ **si** M s'évalue en m dans σ et N s'évalue en n dans σ. De même pour $\underline{*}$.

La sémantique de Marthe (2/3)

Que doit contenir notre ensemble de triplets $Eval \subseteq |\mathcal{M}| \times Env \times \mathbb{N}$?

- Le terme $\underline{n}$ doit s'évaluer en n dans tout σ.

$$\{(\underline{n}, \sigma, n) \mid n \in \mathbb{N}, \sigma \in Env\} \subseteq Eval$$

- Le terme x doit s'évaluer en v dans σ **si** $\sigma(x) = v$.

$$\{(x, \sigma, \sigma(x)) \mid x \in Var, \sigma \in Env\} \subseteq Eval$$

- Le terme $M \,\underline{+}\, N$ doit s'évaluer en $m + n$ dans σ **si** M s'évalue en m dans σ et N s'évalue en n dans σ. De même pour $\underline{*}$.

$$\{(M \,\underline{+}\, N, \sigma, m + n) \mid (M, \sigma, m), (N, \sigma, n) \in Eval\} \subseteq Eval$$
$$\{(M \,\underline{*}\, N, \sigma, mn) \mid (M, \sigma, m), (N, \sigma, n) \in Eval\} \subseteq Eval$$

La sémantique de Marthe (2/3)

Que doit contenir notre ensemble de triplets $\mathit{Eval} \subseteq |\mathcal{M}| \times \mathit{Env} \times \mathbb{N}$?

- Le terme $\underline{n}$ doit s'évaluer en n dans tout σ.

$$\{(\underline{n}, \sigma, n) \mid n \in \mathbb{N}, \sigma \in \mathit{Env}\} \subseteq \mathit{Eval}$$

- Le terme x doit s'évaluer en v dans σ **si** $\sigma(x) = v$.

$$\{(x, \sigma, \sigma(x)) \mid x \in \mathit{Var}, \sigma \in \mathit{Env}\} \subseteq \mathit{Eval}$$

- Le terme $M \mathbin{\underline{+}} N$ doit s'évaluer en $m + n$ dans σ **si** M s'évalue en m dans σ et N s'évalue en n dans σ. De même pour $\underline{*}$.

$$\{(M \mathbin{\underline{+}} N, \sigma, m + n) \mid (M, \sigma, m), (N, \sigma, n) \in \mathit{Eval}\} \subseteq \mathit{Eval}$$
$$\{(M \mathbin{\underline{*}} N, \sigma, mn) \mid (M, \sigma, m), (N, \sigma, n) \in \mathit{Eval}\} \subseteq \mathit{Eval}$$

- **[Exercice]** Déterminer les contraintes pour $\underline{\sum}_{x=M}^{N} P$.

La sémantique de Marthe (3/3)

Au moins deux approches possibles pour $\sum_{x=M}^{N} P$.

Au moins deux approches possibles pour $\underline{\sum}_{x=M}^{N} P$.

1. Un seul cas : évaluer P sur la plage complète en une seule fois.

$$\left\{ \begin{pmatrix} \underline{\sum}_{x=M}^{N} P, \\ \sigma, \sum_{m \leq i \leq n} p_i \end{pmatrix} \middle| \begin{array}{r} (M, \sigma, m), (N, \sigma, n) \in \mathit{Eval}, \\ \forall m \leq i \leq n, (P, \sigma[x \mapsto i], p_i) \in \mathit{Eval} \end{array} \right\} \subseteq \mathit{Eval}$$

La sémantique de Marthe (3/3)

Au moins deux approches possibles pour $\underline{\sum}_{x=M}^{N} P$.

1. Un seul cas : évaluer P sur la plage complète en une seule fois.

$$\left\{ \begin{pmatrix} \underline{\sum}_{x=M}^{N} P, \\ \sigma, \sum_{m \leq i \leq n} p_i \end{pmatrix} \middle| \begin{array}{r} (M, \sigma, m), (N, \sigma, n) \in \mathit{Eval}, \\ \forall m \leq i \leq n, (P, \sigma[x \mapsto i], p_i) \in \mathit{Eval} \end{array} \right\} \subseteq \mathit{Eval}$$

2. Deux cas : évaluer P itération par itération.

$$\left\{ \left(\underline{\sum}_{x=M}^{N} P, \sigma, 0\right) \middle| (M, \sigma, m), (N, \sigma, n) \in \mathit{Eval}, n < m \right\} \subseteq \mathit{Eval}$$

$$\left\{ \left(\underline{\sum}_{x=M}^{N} P, \sigma, p + r\right) \middle| \begin{array}{r} (M, \sigma, m), (N, \sigma, n), (P, \sigma[x \mapsto m], p), \\ (\underline{\sum}_{x=M \underline{+} 1}^{N} P, \sigma, r) \in \mathit{Eval}, m \leq n \end{array} \right\} \subseteq \mathit{Eval}$$

La sémantique de Marthe (3/3)

Au moins deux approches possibles pour $\underline{\sum}_{x=M}^{N} P$.

1. Un seul cas : évaluer P sur la plage complète en une seule fois.

$$\left\{ \begin{pmatrix} \underline{\sum}_{x=M}^{N} P, \\ \sigma, \sum_{m \leq i \leq n} p_i \end{pmatrix} \middle| \begin{array}{r} (M, \sigma, m), (N, \sigma, n) \in \mathit{Eval}, \\ \forall m \leq i \leq n, (P, \sigma[x \mapsto i], p_i) \in \mathit{Eval} \end{array} \right\} \subseteq \mathit{Eval}$$

2. Deux cas : évaluer P itération par itération.

$$\left\{ \left(\underline{\sum}_{x=M}^{N} P, \sigma, 0\right) \middle| (M, \sigma, m), (N, \sigma, n) \in \mathit{Eval}, n < m \right\} \subseteq \mathit{Eval}$$

$$\left\{ \left(\underline{\sum}_{x=M}^{N} P, \sigma, p + r\right) \middle| \begin{array}{r} (M, \sigma, m), (N, \sigma, n), (P, \sigma[x \mapsto m], p), \\ (\underline{\sum}_{x=M\underline{+}\underline{1}}^{N} P, \sigma, r) \in \mathit{Eval}, m \leq n \end{array} \right\} \subseteq \mathit{Eval}$$

Les deux approches sont *mathématiquement* équivalentes.

Fabriquer la sémantique (1/2)

On a obtenu un ensemble de contraintes sur notre ensemble *Eval*.

$$\{(\underline{n}, \sigma, n) \mid n \in \mathbb{N}, \sigma \in Env\} \subseteq Eval \tag{1}$$

$$\{(x, \sigma, \sigma(x)) \mid x \in Var, \sigma \in Env\} \subseteq Eval \tag{2}$$

$$\{(M \underline{+} N, \sigma, m + n) \mid (M, \sigma, m), (N, \sigma, n) \in Eval\} \subseteq Eval \tag{3}$$

$$\{(M \underline{*} N, \sigma, mn) \mid (M, \sigma, m), (N, \sigma, n) \in Eval\} \subseteq Eval \tag{4}$$

$$\left\{ \left(\underline{\textstyle\sum}_{x=M}^{N} P, \sigma, 0 \right) \middle| (M, \sigma, m), (N, \sigma, n) \in Eval, n < m \right\} \subseteq Eval \tag{5}$$

$$\left\{ \begin{pmatrix} \underline{\textstyle\sum}_{x=M}^{N} P, \\ \sigma, p + r \end{pmatrix} \middle| \begin{array}{r} (M, \sigma, m), (N, \sigma, n), (P, \sigma[x \mapsto m], p), \\ (\underline{\textstyle\sum}_{x=M \underline{+} \underline{1}}^{N} P, \sigma, r) \in Eval, m \leq n \end{array} \right\} \subseteq Eval \tag{6}$$

Comment construire un ensemble *Eval* les respectant *exactement* ?

Fabriquer la sémantique (1/2)

On a obtenu un ensemble de contraintes sur notre ensemble *Eval*.

$$\{(\underline{n}, \sigma, n) \mid n \in \mathbb{N}, \sigma \in Env\} \subseteq Eval \tag{1}$$
$$\{(x, \sigma, \sigma(x)) \mid x \in Var, \sigma \in Env\} \subseteq Eval \tag{2}$$
$$\{(M \mathbin{\underline{+}} N, \sigma, m + n) \mid (M, \sigma, m), (N, \sigma, n) \in Eval\} \subseteq Eval \tag{3}$$
$$\{(M \mathbin{\underline{*}} N, \sigma, mn) \mid (M, \sigma, m), (N, \sigma, n) \in Eval\} \subseteq Eval \tag{4}$$
$$\left\{\left(\underline{\sum}_{x=M}^{N} P, \sigma, 0\right) \middle| (M, \sigma, m), (N, \sigma, n) \in Eval, n < m\right\} \subseteq Eval \tag{5}$$
$$\left\{\begin{pmatrix}\underline{\sum}_{x=M}^{N} P, \\ \sigma, p + r\end{pmatrix} \middle| \begin{matrix}(M, \sigma, m), (N, \sigma, n), (P, \sigma[x \mapsto m], p), \\ (\underline{\sum}_{x=M \underline{+} \underline{1}}^{N} P, \sigma, r) \in Eval, m \leq n\end{matrix}\right\} \subseteq Eval \tag{6}$$

Comment construire un ensemble *Eval* les respectant *exactement* ?

- En appliquant le théorème de Knaster-Tarski.

Fabriquer la sémantique (1/2)

On a obtenu un ensemble de contraintes sur notre ensemble *Eval*.

$$\{(\underline{n}, \sigma, n) \mid n \in \mathbb{N}, \sigma \in Env\} \subseteq Eval \qquad (1)$$

$$\{(x, \sigma, \sigma(x)) \mid x \in Var, \sigma \in Env\} \subseteq Eval \qquad (2)$$

$$\{(M \underline{+} N, \sigma, m + n) \mid (M, \sigma, m), (N, \sigma, n) \in Eval\} \subseteq Eval \qquad (3)$$

$$\{(M \underline{*} N, \sigma, mn) \mid (M, \sigma, m), (N, \sigma, n) \in Eval\} \subseteq Eval \qquad (4)$$

$$\left\{ \left(\underline{\sum}_{x=M}^{N} P, \sigma, 0 \right) \middle| (M, \sigma, m), (N, \sigma, n) \in Eval, n < m \right\} \subseteq Eval \qquad (5)$$

$$\left\{ \begin{pmatrix} \underline{\sum}_{x=M}^{N} P, \\ \sigma, p + r \end{pmatrix} \middle| \begin{matrix} (M, \sigma, m), (N, \sigma, n), (P, \sigma[x \mapsto m], p), \\ (\underline{\sum}_{x=M \underline{+} \underline{1}}^{N} P, \sigma, r) \in Eval, m \leq n \end{matrix} \right\} \subseteq Eval \qquad (6)$$

Comment construire un ensemble *Eval* les respectant *exactement* ?

- En appliquant le théorème de Knaster-Tarski.
- En le construisant à la main, cf. transparent suivant.

Fabriquer la sémantique (2/2)

L'idée : construire une séquence croissante d'ensembles $Eval_k$ pour $k \in \mathbb{N}$.

Fabriquer la sémantique (2/2)

L'idée : construire une séquence croissante d'ensembles $Eval_k$ pour $k \in \mathbb{N}$.

$Eval_0$

Fabriquer la sémantique (2/2)

L'idée : construire une séquence croissante d'ensembles Eval_k pour $k \in \mathbb{N}$.

$$\mathit{Eval}_0 = \{(\underline{n}, \sigma, n) \mid n \in \mathbb{N}, \sigma \in \mathit{Env}\} \cup \{(x, \sigma, \sigma(x)) \mid x \in \mathit{Var}, \sigma \in \mathit{Env}\}$$

Fabriquer la sémantique (2/2)

L'idée : construire une séquence croissante d'ensembles Eval_k pour $k \in \mathbb{N}$.

$$\mathit{Eval}_0 = \{(\underline{n}, \sigma, n) \mid n \in \mathbb{N}, \sigma \in \mathit{Env}\} \cup \{(x, \sigma, \sigma(x)) \mid x \in \mathit{Var}, \sigma \in \mathit{Env}\}$$

$$\mathit{Eval}_1 =$$

Fabriquer la sémantique (2/2)

L'idée : construire une séquence croissante d'ensembles $Eval_k$ pour $k \in \mathbb{N}$.

$$Eval_0 = \{(\underline{n}, \sigma, n) \mid n \in \mathbb{N}, \sigma \in Env\} \cup \{(x, \sigma, \sigma(x)) \mid x \in Var, \sigma \in Env\}$$
$$Eval_1 = Eval_0$$

Fabriquer la sémantique (2/2)

L'idée : construire une séquence croissante d'ensembles Eval_k pour $k \in \mathbb{N}$.

$$\mathit{Eval}_0 = \{(\underline{n}, \sigma, n) \mid n \in \mathbb{N}, \sigma \in \mathit{Env}\} \cup \{(x, \sigma, \sigma(x)) \mid x \in \mathit{Var}, \sigma \in \mathit{Env}\}$$
$$\mathit{Eval}_1 = \mathit{Eval}_0 \cup \{(M \mathbin{\underline{+}} N, \sigma, m + n) \mid (M, \sigma, m), (N, \sigma, n) \in \mathit{Eval}_0\}$$

Fabriquer la sémantique (2/2)

L'idée : construire une séquence croissante d'ensembles $Eval_k$ pour $k \in \mathbb{N}$.

$$
\begin{aligned}
Eval_0 &= \{(\underline{n}, \sigma, n) \mid n \in \mathbb{N}, \sigma \in Env\} \cup \{(x, \sigma, \sigma(x)) \mid x \in Var, \sigma \in Env\} \\
\color{red}{Eval_1} &= \color{red}{Eval_0} \cup \{(M \mathbin{\underline{+}} N, \sigma, m + n) \mid (M, \sigma, m), (N, \sigma, n) \in \color{red}{Eval_0}\} \\
&\quad \cup \{(M \mathbin{\underline{*}} N, \sigma, mn) \mid (M, \sigma, m), (N, \sigma, n) \in \color{red}{Eval_0}\} \\
&\quad \cup \left\{ \left(\underline{\sum}_{x=M}^{N} P, \sigma, 0 \right) \middle| (M, \sigma, m), (N, \sigma, n) \in \color{red}{Eval_0}, n < m \right\} \\
&\quad \cup \left\{ \begin{pmatrix} \underline{\sum}_{x=M}^{N} P, \\ \sigma, p + r \end{pmatrix} \middle| \begin{array}{r} (M, \sigma, m), (N, \sigma, n), (P, \sigma[x \mapsto m], p), \\ (\underline{\sum}_{x=M \mathbin{\underline{+}} \underline{1}}^{N} P, \sigma, r) \in \color{red}{Eval_0}, m \leq n \end{array} \right\}
\end{aligned}
$$

Fabriquer la sémantique (2/2)

L'idée : construire une séquence croissante d'ensembles $Eval_k$ pour $k \in \mathbb{N}$.

$$
\begin{aligned}
Eval_0 &= \{(\underline{n}, \sigma, n) \mid n \in \mathbb{N}, \sigma \in Env\} \cup \{(x, \sigma, \sigma(x)) \mid x \in Var, \sigma \in Env\} \\
Eval_1 &= Eval_0 \cup \{(M \mathbin{\underline{+}} N, \sigma, m + n) \mid (M, \sigma, m), (N, \sigma, n) \in Eval_0\} \\
&\quad \cup \{(M \mathbin{\underline{*}} N, \sigma, mn) \mid (M, \sigma, m), (N, \sigma, n) \in Eval_0\} \\
&\quad \cup \left\{ \left(\underline{\textstyle\sum}_{x=M}^{N} P, \sigma, 0 \right) \middle| (M, \sigma, m), (N, \sigma, n) \in Eval_0, n < m \right\} \\
&\quad \cup \left\{ \begin{pmatrix} \underline{\sum}_{x=M}^{N} P, \\ \sigma, p + r \end{pmatrix} \middle| \begin{matrix} (M, \sigma, m), (N, \sigma, n), (P, \sigma[x \mapsto m], p), \\ (\underline{\sum}_{x=M \mathbin{\underline{+}} \underline{1}}^{N} P, \sigma, r) \in Eval_0, m \leq n \end{matrix} \right\}
\end{aligned}
$$

$$
{\color{red}Eval_{k+1}} = {\color{red}Eval_k} \cup \{(M \mathbin{\underline{+}} N, \sigma, m + n) \mid (M, \sigma, m), (N, \sigma, n) \in {\color{red}Eval_k}\} \cup \ldots
$$

Fabriquer la sémantique (2/2)

L'idée : construire une séquence croissante d'ensembles $Eval_k$ pour $k \in \mathbb{N}$.

$$
\begin{aligned}
Eval_0 &= \{(\underline{n}, \sigma, n) \mid n \in \mathbb{N}, \sigma \in Env\} \cup \{(x, \sigma, \sigma(x)) \mid x \in Var, \sigma \in Env\} \\
Eval_1 &= Eval_0 \cup \{(M \mathbin{\underline{+}} N, \sigma, m + n) \mid (M, \sigma, m), (N, \sigma, n) \in Eval_0\} \\
&\quad \cup \{(M \mathbin{\underline{*}} N, \sigma, mn) \mid (M, \sigma, m), (N, \sigma, n) \in Eval_0\} \\
&\quad \cup \left\{ \left(\underline{\textstyle\sum}_{x=M}^{N} P, \sigma, 0 \right) \middle| (M, \sigma, m), (N, \sigma, n) \in Eval_0, n < m \right\} \\
&\quad \cup \left\{ \begin{pmatrix} \underline{\textstyle\sum}_{x=M}^{N} P, \\ \sigma, p + r \end{pmatrix} \middle| \begin{array}{c} (M, \sigma, m), (N, \sigma, n), (P, \sigma[x \mapsto m], p), \\ (\underline{\textstyle\sum}_{x=M \mathbin{\underline{+}} \underline{1}}^{N} P, \sigma, r) \in Eval_0, m \leq n \end{array} \right\} \\
Eval_{k+1} &= Eval_k \cup \{(M \mathbin{\underline{+}} N, \sigma, m + n) \mid (M, \sigma, m), (N, \sigma, n) \in Eval_k\} \cup \ldots
\end{aligned}
$$

La séquence $(Eval_k)_{k \in \mathbb{N}}$ converge vers le $Eval$ qu'on cherche à construire. Il suffit donc de poser comme définition :

$$Eval \stackrel{\text{def}}{=} \bigcup_{k \in \mathbb{N}} Eval_k.$$

Présentation alternative de la sémantique

Il est commode d'adopter une notation infixe pour *Eval*.

$$\boxed{M; \sigma \Downarrow n} \stackrel{\text{def}}{=} (M, \sigma, n) \in \mathit{Eval} \Leftrightarrow \exists k \in \mathbb{N}, (M, \sigma, n) \in \mathit{Eval}_k$$

Ensuite, on peut montrer que $M; \sigma \Downarrow n$ est vrai si et seulement c'est la racine d'un arbre dont les noeuds sont étiquettés par les règles ci-dessous.

Présentation alternative de la sémantique

Il est commode d'adopter une notation infixe pour *Eval*.

$$\boxed{M; \sigma \Downarrow n} \stackrel{\text{def}}{=} (M, \sigma, n) \in \mathit{Eval} \Leftrightarrow \exists k \in \mathbb{N}, (M, \sigma, n) \in \mathit{Eval}_k$$

Ensuite, on peut montrer que $M; \sigma \Downarrow n$ est vrai si et seulement c'est la racine d'un arbre dont les noeuds sont étiquettés par les règles ci-dessous.

$$\frac{}{\underline{n}; \sigma \Downarrow n}$$

Présentation alternative de la sémantique

Il est commode d'adopter une notation infixe pour *Eval*.

$$\boxed{M; \sigma \Downarrow n} \stackrel{\text{def}}{=} (M, \sigma, n) \in \mathit{Eval} \Leftrightarrow \exists k \in \mathbb{N}, (M, \sigma, n) \in \mathit{Eval}_k$$

Ensuite, on peut montrer que $M; \sigma \Downarrow n$ est vrai si et seulement c'est la racine d'un arbre dont les noeuds sont étiquettés par les règles ci-dessous.

$$\frac{}{\underline{n}; \sigma \Downarrow n} \qquad \frac{}{x; \sigma \Downarrow \sigma(x)}$$

Présentation alternative de la sémantique

Il est commode d'adopter une notation infixe pour *Eval*.

$$\boxed{M; \sigma \Downarrow n} \stackrel{\text{def}}{=} (M, \sigma, n) \in \mathit{Eval} \Leftrightarrow \exists k \in \mathbb{N}, (M, \sigma, n) \in \mathit{Eval}_k$$

Ensuite, on peut montrer que $M; \sigma \Downarrow n$ est vrai si et seulement c'est la racine d'un arbre dont les noeuds sont étiquettés par les règles ci-dessous.

$$\frac{}{\underline{n}; \sigma \Downarrow n} \qquad \frac{}{x; \sigma \Downarrow \sigma(x)} \qquad \frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n}{M \mathbin{\underline{+}} N; \sigma \Downarrow m + n}$$

Présentation alternative de la sémantique

Il est commode d'adopter une notation infixe pour *Eval*.

$$\boxed{M; \sigma \Downarrow n} \stackrel{\text{def}}{=} (M, \sigma, n) \in Eval \Leftrightarrow \exists k \in \mathbb{N}, (M, \sigma, n) \in Eval_k$$

Ensuite, on peut montrer que $M; \sigma \Downarrow n$ est vrai si et seulement c'est la racine d'un arbre dont les noeuds sont étiquettés par les règles ci-dessous.

$$\frac{}{\underline{n}; \sigma \Downarrow n} \qquad \frac{}{x; \sigma \Downarrow \sigma(x)} \qquad \frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n}{M \mathbin{\underline{+}} N; \sigma \Downarrow m + n}$$

$$\frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n}{M \mathbin{\underline{*}} N; \sigma \Downarrow mn}$$

Présentation alternative de la sémantique

Il est commode d'adopter une notation infixe pour *Eval*.

$$\boxed{M; \sigma \Downarrow n} \stackrel{\text{def}}{=} (M, \sigma, n) \in \mathit{Eval} \Leftrightarrow \exists k \in \mathbb{N}, (M, \sigma, n) \in \mathit{Eval}_k$$

Ensuite, on peut montrer que $M; \sigma \Downarrow n$ est vrai si et seulement c'est la racine d'un arbre dont les noeuds sont étiquettés par les règles ci-dessous.

$$\frac{}{\underline{n}; \sigma \Downarrow n} \qquad \frac{}{x; \sigma \Downarrow \sigma(x)} \qquad \frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n}{M \mathbin{\underline{+}} N; \sigma \Downarrow m + n}$$

$$\frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n}{M \mathbin{\underline{*}} N; \sigma \Downarrow mn} \qquad \frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n}{\underline{\sum}_{x=M}^{N} P; \sigma \Downarrow 0} \ n < m$$

Présentation alternative de la sémantique

Il est commode d'adopter une notation infixe pour *Eval*.

$$\boxed{M;\sigma \Downarrow n} \stackrel{\text{def}}{=} (M,\sigma,n) \in \mathit{Eval} \Leftrightarrow \exists k \in \mathbb{N}, (M,\sigma,n) \in \mathit{Eval}_k$$

Ensuite, on peut montrer que $M;\sigma \Downarrow n$ est vrai si et seulement c'est la racine d'un arbre dont les noeuds sont étiquettés par les règles ci-dessous.

$$\frac{}{\underline{n};\sigma \Downarrow n} \qquad \frac{}{x;\sigma \Downarrow \sigma(x)} \qquad \frac{M;\sigma \Downarrow m \qquad N;\sigma \Downarrow n}{M \,\underline{+}\, N;\sigma \Downarrow m+n}$$

$$\frac{M;\sigma \Downarrow m \qquad N;\sigma \Downarrow n}{M \,\underline{*}\, N;\sigma \Downarrow mn} \qquad \frac{M;\sigma \Downarrow m \qquad N;\sigma \Downarrow n}{\underline{\sum}_{x=M}^{N} P;\sigma \Downarrow 0}\ n < m$$

$$\frac{M;\sigma \Downarrow m \qquad N;\sigma \Downarrow n \qquad P;\sigma[x \mapsto m] \Downarrow p \qquad \underline{\sum}_{x=M \,\underline{+}\, \underline{1}}^{N} P;\sigma \Downarrow r}{\underline{\sum}_{x=M}^{N} P;\sigma \Downarrow p+r}\ m \leq n$$

Présentation alternative de la sémantique

Il est commode d'adopter une notation infixe pour *Eval*.

$$\boxed{M; \sigma \Downarrow n} \stackrel{\text{def}}{=} (M, \sigma, n) \in \mathit{Eval} \Leftrightarrow \exists k \in \mathbb{N}, (M, \sigma, n) \in \mathit{Eval}_k$$

Ensuite, on peut montrer que $M; \sigma \Downarrow n$ est vrai si et seulement c'est la racine d'un arbre dont les noeuds sont étiquettés par les règles ci-dessous.

$$\frac{}{\underline{n}; \sigma \Downarrow n} \qquad \frac{}{x; \sigma \Downarrow \sigma(x)} \qquad \frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n}{M \mathbin{\underline{+}} N; \sigma \Downarrow m + n}$$

$$\frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n}{M \mathbin{\underline{*}} N; \sigma \Downarrow mn} \qquad \frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n}{\underline{\Sigma}_{x=M}^{N} P; \sigma \Downarrow 0}\ n < m$$

$$\frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n \qquad P; \sigma[x \mapsto m] \Downarrow p \qquad \underline{\Sigma}_{x=M \mathbin{\underline{+}} \underline{1}}^{N} P; \sigma \Downarrow r}{\underline{\Sigma}_{x=M}^{N} P; \sigma \Downarrow p + r}\ m \leq n$$

[Exercice] Construire un arbre de racine $\underline{\Sigma}_{x=\underline{1}}^{2}\, x \mathbin{\underline{*}} \underline{3}; \emptyset \Downarrow n$ (n au choix).

Jugements inductifs et sémantiques à grands pas

En pratique

Plutôt que de construire $(Eval_k)_{k\in\mathbb{N}}$ et *Eval* sous forme ensembliste, on préfère définir directement la relation $M; \sigma \Downarrow n$ par des règles.

- On appelle une telle relation un *jugement inductif* et un arbre de racine $M; \sigma \Downarrow N$ une *dérivation* de *conclusion* $M; \sigma \Downarrow N$.
- Les deux définitions sont équivalentes car $(M, \sigma, n) \in Eval_k$ si et seulement si $M; \sigma \Downarrow n$ admet une dérivation de hauteur k.

Jugements inductifs et sémantiques à grands pas

En pratique

Plutôt que de construire $(Eval_k)_{k \in \mathbb{N}}$ et *Eval* sous forme ensembliste, on préfère définir directement la relation $M; \sigma \Downarrow n$ par des règles.

- On appelle une telle relation un *jugement inductif* et un arbre de racine $M; \sigma \Downarrow N$ une *dérivation* de *conclusion* $M; \sigma \Downarrow N$.
- Les deux définitions sont équivalentes car $(M, \sigma, n) \in Eval_k$ si et seulement si $M; \sigma \Downarrow n$ admet une dérivation de hauteur k.

- Une sémantique qui associe à un terme son résultat final est dite **"à grands pas"**. On en verra d'autres exemples.

Jugements inductifs et sémantiques à grands pas

En pratique

Plutôt que de construire $(\mathit{Eval}_k)_{k \in \mathbb{N}}$ et Eval sous forme ensembliste, on préfère définir directement la relation $M; \sigma \Downarrow n$ par des règles.

- On appelle une telle relation un *jugement inductif* et un arbre de racine $M; \sigma \Downarrow N$ une *dérivation* de *conclusion* $M; \sigma \Downarrow N$.
- Les deux définitions sont équivalentes car $(M, \sigma, n) \in \mathit{Eval}_k$ si et seulement si $M; \sigma \Downarrow n$ admet une dérivation de hauteur k.

- Une sémantique qui associe à un terme son résultat final est dite **"à grands pas"**. On en verra d'autres exemples.
- Celle que nous venons de définir est **déterministe**.

$$\text{Si } M; \sigma \Downarrow n \text{ et } M; \sigma \Downarrow m \text{ alors } m = n.$$

Jugements inductifs et sémantiques à grands pas

En pratique

Plutôt que de construire $(Eval_k)_{k \in \mathbb{N}}$ et *Eval* sous forme ensembliste, on préfère définir directement la relation $M; \sigma \Downarrow n$ par des règles.

- On appelle une telle relation un *jugement inductif* et un arbre de racine $M; \sigma \Downarrow N$ une *dérivation* de *conclusion* $M; \sigma \Downarrow N$.
- Les deux définitions sont équivalentes car $(M, \sigma, n) \in Eval_k$ si et seulement si $M; \sigma \Downarrow n$ admet une dérivation de hauteur k.

- Une sémantique qui associe à un terme son résultat final est dite **"à grands pas"**. On en verra d'autres exemples.
- Celle que nous venons de définir est **déterministe**.

$$\text{Si } M; \sigma \Downarrow n \text{ et } M; \sigma \Downarrow m \text{ alors } m = n.$$

- **[Exercice]** Est-elle **totale** ?

$$\forall M \in |\mathcal{M}|, \forall \sigma \in Env, \exists n \in \mathbb{N}, M; \sigma \Downarrow n$$

Quand l'évaluation est-elle définie ?

- Il n'existe pas d'entier n tel que que $\texttt{x}; \emptyset \Downarrow n$. Donc $R_{\mathcal{M}}(x) = \emptyset$.
- Notre sémantique définit donc une fonction partielle de $|\mathcal{M}| \times \mathit{Env}$ dans $\mathbb{N}$, que notre fonction OCaml `eval` implémente.
- **[Exercice]** Quels sont les termes M tels que $R_{\mathcal{M}}(M) = \emptyset$?

Quand l'évaluation est-elle définie ?

- Il n'existe pas d'entier n tel que que $\mathtt{x}; \emptyset \Downarrow n$. Donc $R_{\mathcal{M}}(x) = \emptyset$.
- Notre sémantique définit donc une fonction partielle de $|\mathcal{M}| \times \mathit{Env}$ dans $\mathbb{N}$, que notre fonction OCaml `eval` implémente.
- **[Exercice]** Quels sont les termes M tels que $R_{\mathcal{M}}(M) = \emptyset$?
- Pour que l'évaluation d'un terme soit définie, il faut que la valeur de chacune de ses variables x soit définie. Donc, x doit :
  - soit appartenir à $\sigma^{-1}(\mathbb{N})$,
  - soit apparaître sous une construction $\underline{\Sigma}_{x=M}^{N}(-)$ (qui la "lie").

  Comment rendre ces intuitions précises ?

Variables libres et termes clos

- L'ensemble $FV(M)$ des *variables libres* d'un terme M est défini par récursion sur M.

$$
\begin{aligned}
FV(\underline{n}) &= \\
FV(x) &= \\
FV(M \mathbin{\underline{+}} N) &= \\
FV(M \mathbin{\underline{*}} N) &= \\
FV(\underline{\Sigma}_{x=M}^{N} P) &=
\end{aligned}
$$

Variables libres et termes clos

- L'ensemble $FV(M)$ des *variables libres* d'un terme M est défini par récursion sur M.

$$
\begin{aligned}
FV(\underline{n}) &= \emptyset \\
FV(x) &= \\
FV(M \mathbin{\underline{+}} N) &= \\
FV(M \mathbin{\underline{*}} N) &= \\
FV(\underline{\Sigma}_{x=M}^{N} P) &=
\end{aligned}
$$

Variables libres et termes clos

- L'ensemble $FV(M)$ des *variables libres* d'un terme M est défini par récursion sur M.

$$
\begin{aligned}
FV(\underline{n}) &= \emptyset \\
FV(x) &= \{x\} \\
FV(M \mathbin{\underline{+}} N) &= \\
FV(M \mathbin{\underline{*}} N) &= \\
FV(\underline{\Sigma}_{x=M}^{N} P) &=
\end{aligned}
$$

Variables libres et termes clos

- L'ensemble $FV(M)$ des *variables libres* d'un terme M est défini par récursion sur M.

$$
\begin{aligned}
FV(\underline{n}) &= \emptyset \\
FV(x) &= \{x\} \\
FV(M \mathbin{\underline{+}} N) &= FV(M) \cup FV(N) \\
FV(M \mathbin{\underline{*}} N) &= \\
FV(\underline{\sum}_{x=M}^{N} P) &=
\end{aligned}
$$

Variables libres et termes clos

- L'ensemble $FV(M)$ des *variables libres* d'un terme M est défini par récursion sur M.

$$
\begin{aligned}
FV(\underline{n}) &= \emptyset \\
FV(x) &= \{x\} \\
FV(M \mathbin{\underline{+}} N) &= FV(M) \cup FV(N) \\
FV(M \mathbin{\underline{*}} N) &= FV(M) \cup FV(N) \\
FV(\underline{\Sigma}_{x=M}^{N} P) &=
\end{aligned}
$$

Variables libres et termes clos

- L'ensemble $FV(M)$ des *variables libres* d'un terme M est défini par récursion sur M.

$$
\begin{aligned}
FV(\underline{n}) &= \emptyset \\
FV(x) &= \{x\} \\
FV(M \underline{+} N) &= FV(M) \cup FV(N) \\
FV(M \underline{*} N) &= FV(M) \cup FV(N) \\
FV(\underline{\sum}_{x=M}^{N} P) &= FV(M) \cup FV(N) \cup (FV(P) \setminus \{x\})
\end{aligned}
$$

Variables libres et termes clos

- L'ensemble $FV(M)$ des *variables libres* d'un terme M est défini par récursion sur M.

$$
\begin{aligned}
FV(\underline{n}) &= \emptyset \\
FV(x) &= \{x\} \\
FV(M \mathbin{\underline{+}} N) &= FV(M) \cup FV(N) \\
FV(M \mathbin{\underline{*}} N) &= FV(M) \cup FV(N) \\
FV(\underline{\sum}_{x=M}^{N} P) &= FV(M) \cup FV(N) \cup (FV(P) \setminus \{x\})
\end{aligned}
$$

- Un terme M est dit *clos* si $FV(M) = \emptyset$ et *ouvert* sinon.
- **[Exercice]** $\underline{\sum}_{\mathtt{x}=\underline{1}}^{\underline{4}} \mathtt{x}$, $\underline{\sum}_{\mathtt{x}=\underline{1}}^{\underline{4}} \mathtt{y}$ et $(\underline{\sum}_{\mathtt{y}=\underline{1}}^{\underline{4}} \mathtt{y}) \mathbin{\underline{+}} \mathtt{y}$ sont-ils clos ou ouverts ?

Variables libres et termes clos

- L'ensemble $FV(M)$ des *variables libres* d'un terme M est défini par récursion sur M.

$$\begin{aligned}
FV(\underline{n}) &= \emptyset \\
FV(x) &= \{x\} \\
FV(M \mathbin{\underline{+}} N) &= FV(M) \cup FV(N) \\
FV(M \mathbin{\underline{*}} N) &= FV(M) \cup FV(N) \\
FV(\underline{\Sigma}_{x=M}^{N} P) &= FV(M) \cup FV(N) \cup (FV(P) \setminus \{x\})
\end{aligned}$$

- Un terme M est dit *clos* si $FV(M) = \emptyset$ et *ouvert* sinon.
- **[Exercice]** $\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\,\mathtt{x}$, $\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\,\mathtt{y}$ et $(\underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{4}}\,\mathtt{y}) \mathbin{\underline{+}} \mathtt{y}$ sont-ils clos ou ouverts ?
- **[Exercice]** Programmer FV en OCaml.

Variables libres et termes clos

- L'ensemble $FV(M)$ des *variables libres* d'un terme M est défini par récursion sur M.

$$
\begin{aligned}
FV(\underline{n}) &= \emptyset \\
FV(x) &= \{x\} \\
FV(M \mathbin{\underline{+}} N) &= FV(M) \cup FV(N) \\
FV(M \mathbin{\underline{*}} N) &= FV(M) \cup FV(N) \\
FV(\underline{\sum}_{x=M}^{N} P) &= FV(M) \cup FV(N) \cup (FV(P) \setminus \{x\})
\end{aligned}
$$

- Un terme M est dit *clos* si $FV(M) = \emptyset$ et *ouvert* sinon.
- **[Exercice]** $\underline{\sum}_{\mathtt{x}=\underline{1}}^{\underline{4}} \mathtt{x}$, $\underline{\sum}_{\mathtt{x}=\underline{1}}^{\underline{4}} \mathtt{y}$ et $(\underline{\sum}_{\mathtt{y}=\underline{1}}^{\underline{4}} \mathtt{y}) \mathbin{\underline{+}} \mathtt{y}$ sont-ils clos ou ouverts ?
- **[Exercice]** Programmer FV en OCaml.

Propriété

Il existe $n \in \mathbb{N}$ tel que $M; \sigma \Downarrow n$ si et seulement si $FV(M) \subseteq \sigma^{-1}(\mathbb{N})$.

Équivalence observationnelle de termes Marthe

Il semble raisonnable de considérer que deux termes Marthe sont équivalents lorsqu'ils calculent le même entier dans tout environnement.

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in Env, \forall n \in \mathbb{N},\ M; \sigma \Downarrow n \Leftrightarrow N; \sigma \Downarrow n$$

Que peut-on dire sur cette relation d'*équivalence observationnelle* ?

Équivalence observationnelle de termes Marthe

Il semble raisonnable de considérer que deux termes Marthe sont équivalents lorsqu'ils calculent le même entier dans tout environnement.

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in Env, \forall n \in \mathbb{N},\ M; \sigma \Downarrow n \Leftrightarrow N; \sigma \Downarrow n$$

Que peut-on dire sur cette relation d'*équivalence observationnelle* ?

- Elle est clairement réflexive, symétrique et transitive.

Équivalence observationnelle de termes Marthe

Il semble raisonnable de considérer que deux termes Marthe sont équivalents lorsqu'ils calculent le même entier dans tout environnement.

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in Env, \forall n \in \mathbb{N},\ M; \sigma \Downarrow n \Leftrightarrow N; \sigma \Downarrow n$$

Que peut-on dire sur cette relation d'*équivalence observationnelle* ?

- Elle est clairement réflexive, symétrique et transitive.
- Elle valide les identités arithmétiques habituelles.

$$(M \mathbin{\underline{+}} N) \mathbin{\underline{+}} P \equiv M \mathbin{\underline{+}} (N \mathbin{\underline{+}} P) \qquad M \mathbin{\underline{+}} N \equiv N \mathbin{\underline{+}} M \qquad \ldots$$

Équivalence observationnelle de termes Marthe

Il semble raisonnable de considérer que deux termes Marthe sont équivalents lorsqu'ils calculent le même entier dans tout environnement.

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in \mathit{Env}, \forall n \in \mathbb{N},\ M; \sigma \Downarrow n \Leftrightarrow N; \sigma \Downarrow n$$

Que peut-on dire sur cette relation d'*équivalence observationnelle* ?

- Elle est clairement réflexive, symétrique et transitive.
- Elle valide les identités arithmétiques habituelles.

$$(M \mathbin{\underline{+}} N) \mathbin{\underline{+}} P \equiv M \mathbin{\underline{+}} (N \mathbin{\underline{+}} P) \qquad M \mathbin{\underline{+}} N \equiv N \mathbin{\underline{+}} M \qquad \ldots$$

- Elle valide certains renommages de variables.

$$\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} (\mathtt{x} \mathbin{\underline{*}} \mathtt{z}) \stackrel{?}{\equiv} \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} (\mathtt{y} \mathbin{\underline{*}} \mathtt{k}) \qquad \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} (\mathtt{x} \mathbin{\underline{*}} \mathtt{z}) \stackrel{?}{\equiv} \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} (\mathtt{y} \mathbin{\underline{*}} \mathtt{z})$$

$$\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} (\mathtt{x} \mathbin{\underline{+}} \mathtt{y}) \stackrel{?}{\equiv} \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} (\mathtt{y} \mathbin{\underline{+}} \mathtt{x}) \qquad \mathtt{x} \mathbin{\underline{+}} \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}}\, \mathtt{x} \stackrel{?}{\equiv} \mathtt{y} \mathbin{\underline{+}} \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}}\, \mathtt{y}$$

Équivalence observationnelle de termes Marthe

Il semble raisonnable de considérer que deux termes Marthe sont équivalents lorsqu'ils calculent le même entier dans tout environnement.

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in \mathit{Env}, \forall n \in \mathbb{N},\ M; \sigma \Downarrow n \Leftrightarrow N; \sigma \Downarrow n$$

Que peut-on dire sur cette relation d'*équivalence observationnelle* ?

- Elle est clairement réflexive, symétrique et transitive.
- Elle valide les identités arithmétiques habituelles.

$$(M \mathbin{\underline{+}} N) \mathbin{\underline{+}} P \equiv M \mathbin{\underline{+}} (N \mathbin{\underline{+}} P) \qquad M \mathbin{\underline{+}} N \equiv N \mathbin{\underline{+}} M \qquad \ldots$$

- Elle valide certains renommages de variables.

$$\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} (\mathtt{x} \mathbin{\underline{*}} \mathtt{z}) \not\equiv \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} (\mathtt{y} \mathbin{\underline{*}} \mathtt{k}) \qquad \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} (\mathtt{x} \mathbin{\underline{*}} \mathtt{z}) \stackrel{?}{\equiv} \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} (\mathtt{y} \mathbin{\underline{*}} \mathtt{z})$$

$$\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} (\mathtt{x} \mathbin{\underline{+}} \mathtt{y}) \stackrel{?}{\equiv} \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} (\mathtt{y} \mathbin{\underline{+}} \mathtt{x}) \qquad \mathtt{x} \mathbin{\underline{+}} \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{x} \stackrel{?}{\equiv} \mathtt{y} \mathbin{\underline{+}} \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} \mathtt{y}$$

Équivalence observationnelle de termes Marthe

Il semble raisonnable de considérer que deux termes Marthe sont équivalents lorsqu'ils calculent le même entier dans tout environnement.

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in \mathit{Env}, \forall n \in \mathbb{N},\ M; \sigma \Downarrow n \Leftrightarrow N; \sigma \Downarrow n$$

Que peut-on dire sur cette relation d'*équivalence observationnelle* ?

- Elle est clairement réflexive, symétrique et transitive.
- Elle valide les identités arithmétiques habituelles.

$$(M \mathbin{\underline{+}} N) \mathbin{\underline{+}} P \equiv M \mathbin{\underline{+}} (N \mathbin{\underline{+}} P) \qquad M \mathbin{\underline{+}} N \equiv N \mathbin{\underline{+}} M \qquad \ldots$$

- Elle valide certains renommages de variables.

$$\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}}\, (\mathtt{x} \mathbin{\underline{*}} \mathtt{z}) \not\equiv \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}}\, (\mathtt{y} \mathbin{\underline{*}} \mathtt{k}) \qquad \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}}\, (\mathtt{x} \mathbin{\underline{*}} \mathtt{z}) \equiv \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}}\, (\mathtt{y} \mathbin{\underline{*}} \mathtt{z})$$

$$\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}}\, \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}}\, (\mathtt{x} \mathbin{\underline{+}} \mathtt{y}) \stackrel{?}{\equiv} \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}}\, \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}}\, (\mathtt{y} \mathbin{\underline{+}} \mathtt{x}) \qquad \mathtt{x} \mathbin{\underline{+}} \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}}\, \mathtt{x} \stackrel{?}{\equiv} \mathtt{y} \mathbin{\underline{+}} \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}}\, \mathtt{y}$$

Équivalence observationnelle de termes Marthe

Il semble raisonnable de considérer que deux termes Marthe sont équivalents lorsqu'ils calculent le même entier dans tout environnement.

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in \mathit{Env}, \forall n \in \mathbb{N},\ M; \sigma \Downarrow n \Leftrightarrow N; \sigma \Downarrow n$$

Que peut-on dire sur cette relation d'*équivalence observationnelle* ?

- Elle est clairement réflexive, symétrique et transitive.
- Elle valide les identités arithmétiques habituelles.

$$(M \underline{+} N) \underline{+} P \equiv M \underline{+} (N \underline{+} P) \qquad M \underline{+} N \equiv N \underline{+} M \qquad \ldots$$

- Elle valide certains renommages de variables.

$$\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} (\mathtt{x} \mathbin{\underline{*}} \mathtt{z}) \not\equiv \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} (\mathtt{y} \mathbin{\underline{*}} \mathtt{k}) \qquad \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} (\mathtt{x} \mathbin{\underline{*}} \mathtt{z}) \equiv \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} (\mathtt{y} \mathbin{\underline{*}} \mathtt{z})$$

$$\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} (\mathtt{x} \mathbin{\underline{+}} \mathtt{y}) \equiv \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} (\mathtt{y} \mathbin{\underline{+}} \mathtt{x}) \qquad \mathtt{x} \mathbin{\underline{+}} \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{x} \stackrel{?}{\equiv} \mathtt{y} \mathbin{\underline{+}} \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} \mathtt{y}$$

Équivalence observationnelle de termes Marthe

Il semble raisonnable de considérer que deux termes Marthe sont équivalents lorsqu'ils calculent le même entier dans tout environnement.

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in \mathit{Env}, \forall n \in \mathbb{N},\ M; \sigma \Downarrow n \Leftrightarrow N; \sigma \Downarrow n$$

Que peut-on dire sur cette relation d'*équivalence observationnelle* ?

- Elle est clairement réflexive, symétrique et transitive.
- Elle valide les identités arithmétiques habituelles.

$$(M \mathbin{\underline{+}} N) \mathbin{\underline{+}} P \equiv M \mathbin{\underline{+}} (N \mathbin{\underline{+}} P) \qquad M \mathbin{\underline{+}} N \equiv N \mathbin{\underline{+}} M \qquad \ldots$$

- Elle valide certains renommages de variables.

$$\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} (\mathtt{x} \mathbin{\underline{*}} \mathtt{z}) \not\equiv \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} (\mathtt{y} \mathbin{\underline{*}} \mathtt{k}) \qquad \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} (\mathtt{x} \mathbin{\underline{*}} \mathtt{z}) \equiv \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} (\mathtt{y} \mathbin{\underline{*}} \mathtt{z})$$

$$\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} (\mathtt{x} \mathbin{\underline{+}} \mathtt{y}) \equiv \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} (\mathtt{y} \mathbin{\underline{+}} \mathtt{x}) \qquad \mathtt{x} \mathbin{\underline{+}} \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{x} \not\equiv \mathtt{y} \mathbin{\underline{+}} \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} \mathtt{y}$$

Équivalence observationnelle de termes Marthe

Il semble raisonnable de considérer que deux termes Marthe sont équivalents lorsqu'ils calculent le même entier dans tout environnement.

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in \mathit{Env}, \forall n \in \mathbb{N},\ M; \sigma \Downarrow n \Leftrightarrow N; \sigma \Downarrow n$$

Que peut-on dire sur cette relation d'*équivalence observationnelle* ?

- Elle est clairement réflexive, symétrique et transitive.
- Elle valide les identités arithmétiques habituelles.

$$(M \underline{+} N) \underline{+} P \equiv M \underline{+} (N \underline{+} P) \qquad M \underline{+} N \equiv N \underline{+} M \qquad \ldots$$

- Elle valide certains renommages de variables.

$$\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} (\mathtt{x} \,\underline{*}\, \mathtt{z}) \not\equiv \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} (\mathtt{y} \,\underline{*}\, \mathtt{k}) \qquad \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} (\mathtt{x} \,\underline{*}\, \mathtt{z}) \equiv \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} (\mathtt{y} \,\underline{*}\, \mathtt{z})$$

$$\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} (\mathtt{x} \,\underline{+}\, \mathtt{y}) \equiv \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} (\mathtt{y} \,\underline{+}\, \mathtt{x}) \qquad \mathtt{x} \,\underline{+}\, \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{x} \not\equiv \mathtt{y} \,\underline{+}\, \underline{\Sigma}_{\mathtt{y}=\underline{1}}^{\underline{9}} \mathtt{y}$$

L'égalité à un renommage des variables liées près est une notion universelle dans les langages de programmation : l'α-conversion.

L'α-conversion : intuitions

Une *occurence* liée n'a pas d'identité : elle ne fait que référence à un lieur.

$$\underline{\sum}_{\mathtt{x}=\underline{0}}^{\underline{9}} \left(\mathtt{x} \mathbin{\underline{+}} \underline{\sum}_{\mathtt{x}=\underline{0}}^{\underline{9}} \mathtt{x}\right) \equiv_\alpha \underline{\sum}_{\mathtt{x}=\underline{0}}^{\underline{9}} \left(\mathtt{x} \mathbin{\underline{+}} \underline{\sum}_{\mathtt{y}=\underline{0}}^{\underline{9}} \mathtt{y}\right) \not\equiv_\alpha \underline{\sum}_{\mathtt{x}=\underline{0}}^{\underline{9}} \left(\mathtt{x} \mathbin{\underline{+}} \underline{\sum}_{\mathtt{y}=\underline{0}}^{\underline{9}} \mathtt{x}\right)$$

L'α-conversion : intuitions

Une *occurence* liée n'a pas d'identité : elle ne fait que référence à un lieur.

$$\underline{\sum}_{\mathtt{x}=\underline{0}}^{\underline{9}} \left(\mathtt{x} \mathbin{\underline{+}} \underline{\sum}_{\mathtt{x}=\underline{0}}^{\underline{9}} \mathtt{x}\right) \equiv_\alpha \underline{\sum}_{\mathtt{x}=\underline{0}}^{\underline{9}} \left(\mathtt{x} \mathbin{\underline{+}} \underline{\sum}_{\mathtt{y}=\underline{0}}^{\underline{9}} \mathtt{y}\right) \not\equiv_\alpha \underline{\sum}_{\mathtt{x}=\underline{0}}^{\underline{9}} \left(\mathtt{x} \mathbin{\underline{+}} \underline{\sum}_{\mathtt{y}=\underline{0}}^{\underline{9}} \mathtt{x}\right)$$

On peut donc voir les termes comme des graphes de liaison.

$$\underline{\sum}_{\square=\underline{0}}^{\underline{9}} \left(\square \mathbin{\underline{+}} \underline{\sum}_{\square=\underline{0}}^{\underline{9}} \square\right) \neq \underline{\sum}_{\square=\underline{0}}^{\underline{9}} \left(\square \mathbin{\underline{+}} \underline{\sum}_{\square=\underline{0}}^{\underline{9}} \square\right)$$

L'α-conversion : intuitions

Une *occurence* liée n'a pas d'identité : elle ne fait que référence à un lieur.

$$\underline{\Sigma}_{\mathtt{x}\underline{=}\underline{0}}^{\underline{9}}\left(\mathtt{x}\,\underline{+}\,\underline{\Sigma}_{\mathtt{x}\underline{=}\underline{0}}^{\underline{9}}\,\mathtt{x}\right) \equiv_\alpha \underline{\Sigma}_{\mathtt{x}\underline{=}\underline{0}}^{\underline{9}}\left(\mathtt{x}\,\underline{+}\,\underline{\Sigma}_{\mathtt{y}\underline{=}\underline{0}}^{\underline{9}}\,\mathtt{y}\right) \not\equiv_\alpha \underline{\Sigma}_{\mathtt{x}\underline{=}\underline{0}}^{\underline{9}}\left(\mathtt{x}\,\underline{+}\,\underline{\Sigma}_{\mathtt{y}\underline{=}\underline{0}}^{\underline{9}}\,\mathtt{x}\right)$$

On peut donc voir les termes comme des graphes de liaison.

$$\underline{\Sigma}_{\square\underline{=}\underline{0}}^{\underline{9}}\left(\square\,\underline{+}\,\underline{\Sigma}_{\square\underline{=}\underline{0}}^{\underline{9}}\,\square\right) \neq \underline{\Sigma}_{\square\underline{=}\underline{0}}^{\underline{9}}\left(\square\,\underline{+}\,\underline{\Sigma}_{\square\underline{=}\underline{0}}^{\underline{9}}\,\square\right)$$

Les graphes de liaison :

- rendent l'α-conversion triviale ($M \equiv_\alpha N$ ssi $\mathrm{Gr}(M) = \mathrm{Gr}(N)$),
- assignent une identité uniquement aux occurences libres,

$$\mathrm{Gr}(\underline{\Sigma}_{\mathtt{x}\underline{=}\underline{0}}^{\underline{9}}\,(\mathtt{x}\,\underline{+}\,\mathtt{y})) = \underline{\Sigma}_{\square\underline{=}\underline{0}}^{\underline{9}}\,(\square\,\underline{+}\,\mathtt{y})$$

- sont utilisés en pratique dans les implémentations efficaces.

L'α-conversion : renommage des variabes libres

On définit $M[y/x]$, le renommage de x en y dans M, comme suit.

$$z[y/x] =$$

$$\underline{n}[y/x] =$$

$$(M \mathbin{\underline{+}} P)[y/x] =$$

$$(M \mathbin{\underline{*}} P)[y/x] =$$

$$(\underline{\Sigma}_{z=M}^{N} P)[y/x] =$$

L'α-conversion : renommage des variabes libres

On définit $M[y/x]$, le renommage de x en y dans M, comme suit.

$$
\begin{aligned}
z[y/x] &= \begin{cases} y \text{ si } z = x \\ z \text{ sinon} \end{cases} \\
\underline{n}[y/x] &= \\
(M \mathbin{\underline{+}} P)[y/x] &= \\
(M \mathbin{\underline{*}} P)[y/x] &= \\
\\
(\underline{\Sigma}_{z=M}^{N} P)[y/x] &=
\end{aligned}
$$

L'α-conversion : renommage des variabes libres

On définit $M[y/x]$, le renommage de x en y dans M, comme suit.

$$
\begin{aligned}
z[y/x] &= \begin{cases} y \text{ si } z = x \\ z \text{ sinon} \end{cases} \\
\underline{n}[y/x] &= \underline{n} \\
(M \mathbin{\underline{+}} P)[y/x] &= \\
(M \mathbin{\underline{*}} P)[y/x] &= \\
(\underline{\Sigma}_{z=M}^{N} P)[y/x] &=
\end{aligned}
$$

L'α-conversion : renommage des variabes libres

On définit $M[y/x]$, le renommage de x en y dans M, comme suit.

$$
\begin{aligned}
z[y/x] &= \begin{cases} y \text{ si } z = x \\ z \text{ sinon} \end{cases} \\
\underline{n}[y/x] &= \underline{n} \\
(M \mathbin{\underline{+}} P)[y/x] &= M[y/x] \mathbin{\underline{+}} P[y/x] \\
(M \mathbin{\underline{*}} P)[y/x] &= \\
\\
(\underline{\Sigma}_{z=M}^{N} P)[y/x] &=
\end{aligned}
$$

L'α-conversion : renommage des variabes libres

On définit $M[y/x]$, le renommage de x en y dans M, comme suit.

$$
\begin{aligned}
z[y/x] &= \begin{cases} y \text{ si } z = x \\ z \text{ sinon} \end{cases} \\
\underline{n}[y/x] &= \underline{n} \\
(M \mathbin{\underline{+}} P)[y/x] &= M[y/x] \mathbin{\underline{+}} P[y/x] \\
(M \mathbin{\underline{*}} P)[y/x] &= M[y/x] \mathbin{\underline{*}} P[y/x] \\
\\
(\underline{\Sigma}_{z=M}^{N} P)[y/x] &=
\end{aligned}
$$

L'α-conversion : renommage des variabes libres

On définit $M[y/x]$, le renommage de x en y dans M, comme suit.

$$
\begin{aligned}
z[y/x] &= \begin{cases} y \text{ si } z = x \\ z \text{ sinon} \end{cases} \\
\underline{n}[y/x] &= \underline{n} \\
(M \mathbin{\underline{+}} P)[y/x] &= M[y/x] \mathbin{\underline{+}} P[y/x] \\
(M \mathbin{\underline{*}} P)[y/x] &= M[y/x] \mathbin{\underline{*}} P[y/x] \\
(\underline{\sum}_{z=M}^{N} P)[y/x] &\overset{?}{=} \begin{cases} \underline{\sum}_{z=M[y/x]}^{N[y/x]} P \text{ si } z = x \\ \underline{\sum}_{z=M[y/x]}^{N[y/x]} P[y/x] \text{ sinon} \end{cases}
\end{aligned}
$$

L'α-conversion : renommage des variabes libres

On définit $M[y/x]$, le renommage de x en y dans M, comme suit.

$$z[y/x] = \begin{cases} y \text{ si } z = x \\ z \text{ sinon} \end{cases}$$

$$\underline{n}[y/x] = \underline{n}$$

$$(M \underline{+} P)[y/x] = M[y/x] \underline{+} P[y/x]$$

$$(M \underline{*} P)[y/x] = M[y/x] \underline{*} P[y/x]$$

$$(\underline{\sum}_{z=M}^{N} P)[y/x] \overset{?}{=} \begin{cases} \underline{\sum}_{z=M[y/x]}^{N[y/x]} P \text{ si } z = x \\ \underline{\sum}_{z=M[y/x]}^{N[y/x]} P[y/x] \text{ sinon} \end{cases}$$

Cette définition est-elle raisonnable ?

[Exercice] Calculer :

$$(\underline{\sum}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{y})[\mathtt{z}/\mathtt{y}] =$$

$$(\underline{\sum}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{x})[\mathtt{z}/\mathtt{x}] =$$

$$(\underline{\sum}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{y})[\mathtt{x}/\mathtt{y}] =$$

L'α-conversion : renommage des variabes libres

On définit $M[y/x]$, le renommage de x en y dans M, comme suit.

$$z[y/x] = \begin{cases} y \text{ si } z = x \\ z \text{ sinon} \end{cases}$$

$$\underline{n}[y/x] = \underline{n}$$

$$(M \mathbin{\underline{+}} P)[y/x] = M[y/x] \mathbin{\underline{+}} P[y/x]$$

$$(M \mathbin{\underline{*}} P)[y/x] = M[y/x] \mathbin{\underline{*}} P[y/x]$$

$$(\underline{\Sigma}_{z=M}^{N} P)[y/x] \stackrel{?}{=} \begin{cases} \underline{\Sigma}_{z=M[y/x]}^{N[y/x]} P \text{ si } z = x \\ \underline{\Sigma}_{z=M[y/x]}^{N[y/x]} P[y/x] \text{ sinon} \end{cases}$$

Cette définition est-elle raisonnable ?

$$(\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{y})[\mathtt{z}/\mathtt{y}] = \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{z}$$

[Exercice] Calculer :

$$(\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{x})[\mathtt{z}/\mathtt{x}] =$$

$$(\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{y})[\mathtt{x}/\mathtt{y}] =$$

L'α-conversion : renommage des variabes libres

On définit $M[y/x]$, le renommage de x en y dans M, comme suit.

$$z[y/x] = \begin{cases} y \text{ si } z = x \\ z \text{ sinon} \end{cases}$$

$$\underline{n}[y/x] = \underline{n}$$

$$(M \mathbin{\underline{+}} P)[y/x] = M[y/x] \mathbin{\underline{+}} P[y/x]$$

$$(M \mathbin{\underline{*}} P)[y/x] = M[y/x] \mathbin{\underline{*}} P[y/x]$$

$$(\underline{\sum}_{z=M}^{N} P)[y/x] \overset{?}{=} \begin{cases} \underline{\sum}_{z=M[y/x]}^{N[y/x]} P \text{ si } z = x \\ \underline{\sum}_{z=M[y/x]}^{N[y/x]} P[y/x] \text{ sinon} \end{cases}$$

Cette définition est-elle raisonnable ?

$$(\underline{\sum}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{y})[\mathtt{z}/\mathtt{y}] = \underline{\sum}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{z}$$

[Exercice] Calculer :

$$(\underline{\sum}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{x})[\mathtt{z}/\mathtt{x}] = \underline{\sum}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{x}$$

$$(\underline{\sum}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{y})[\mathtt{x}/\mathtt{y}] =$$

L'α-conversion : renommage des variabes libres

On définit $M[y/x]$, le renommage de x en y dans M, comme suit.

$$z[y/x] = \begin{cases} y \text{ si } z = x \\ z \text{ sinon} \end{cases}$$

$$\underline{n}[y/x] = \underline{n}$$

$$(M \mathbin{\underline{+}} P)[y/x] = M[y/x] \mathbin{\underline{+}} P[y/x]$$

$$(M \mathbin{\underline{*}} P)[y/x] = M[y/x] \mathbin{\underline{*}} P[y/x]$$

$$(\underline{\Sigma}_{z=M}^{N} P)[y/x] \overset{?}{=} \begin{cases} \underline{\Sigma}_{z=M[y/x]}^{N[y/x]} P \text{ si } z = x \\ \underline{\Sigma}_{z=M[y/x]}^{N[y/x]} P[y/x] \text{ sinon} \end{cases}$$

Cette définition est-elle raisonnable ?

[Exercice] Calculer :

$$(\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{9}\, \mathtt{y})[\mathtt{z}/\mathtt{y}] = \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{9}\, \mathtt{z}$$

$$(\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{9}\, \mathtt{x})[\mathtt{z}/\mathtt{x}] = \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{9}\, \mathtt{x}$$

$$(\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{9}\, \mathtt{y})[\mathtt{x}/\mathtt{y}] = \underline{\Sigma}_{\mathtt{x}=\underline{1}}^{9}\, \mathtt{x}$$

L'α-conversion : renommage des variabes libres

On définit $M[y/x]$, le renommage de x en y dans M, comme suit.

$$z[y/x] = \begin{cases} y \text{ si } z = x \\ z \text{ sinon} \end{cases}$$

$$\underline{n}[y/x] = \underline{n}$$

$$(M \mathbin{\underline{+}} P)[y/x] = M[y/x] \mathbin{\underline{+}} P[y/x]$$

$$(M \mathbin{\underline{*}} P)[y/x] = M[y/x] \mathbin{\underline{*}} P[y/x]$$

$$(\underline{\sum}_{z=M}^{N} P)[y/x] \color{red}{\neq} \begin{cases} \underline{\sum}_{z=M[y/x]}^{N[y/x]} P \text{ si } z = x \\ \underline{\sum}_{z=M[y/x]}^{N[y/x]} P[y/x] \text{ sinon} \end{cases}$$

Cette définition est-elle raisonnable ? Non. **[Exercice]** Calculer :

$$(\underline{\sum}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{y})[\mathtt{z}/\mathtt{y}] = \underline{\sum}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{z} \qquad (\underline{\sum}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{x})[\mathtt{z}/\mathtt{x}] = \underline{\sum}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{x}$$

$$(\underline{\sum}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{y})[\mathtt{x}/\mathtt{y}] \color{red}{\neq} \underline{\sum}_{\mathtt{x}=\underline{1}}^{\underline{9}} \mathtt{x}$$

L'α-conversion : définition

Il ne faut pas *capturer* de variable libre lors d'un renommage :

$$(\underline{\sum}_{z=M}^{N} P)[y/x] = \begin{cases} \underline{\sum}_{z=M[y/x]}^{N[y/x]} P \text{ si } z = x \\ \underline{\sum}_{k=M[y/x]}^{N[y/x]} P[k/z][y/x] \text{ sinon} \end{cases}$$

où k est une variable *fraîche*, au sens où $k \notin FV(P) \cup \{y\}$.

L'α-conversion : définition

Il ne faut pas *capturer* de variable libre lors d'un renommage :

$$(\sum_{z=M}^{N} P)[y/x] = \begin{cases} \sum_{z=M[y/x]}^{N[y/x]} P \text{ si } z = x \\ \sum_{k=M[y/x]}^{N[y/x]} P[k/z][y/x] \text{ sinon} \end{cases}$$

où k est une variable *fraîche*, au sens où $k \notin FV(P) \cup \{y\}$.
L'α-conversion, notée $\equiv_\alpha$, est définie comme un jugement inductif.

L'α-conversion : définition

Il ne faut pas *capturer* de variable libre lors d'un renommage :

$$(\textstyle\sum_{z=M}^{N} P)[y/x] = \begin{cases} \sum_{z=M[y/x]}^{N[y/x]} P \text{ si } z = x \\ \sum_{k=M[y/x]}^{N[y/x]} P[k/z][y/x] \text{ sinon} \end{cases}$$

où k est une variable *fraîche*, au sens où $k \notin FV(P) \cup \{y\}$.
L'α-conversion, notée $\equiv_\alpha$, est définie comme un jugement inductif.

$$\frac{}{x \equiv_\alpha x}$$

L'α-conversion : définition

Il ne faut pas *capturer* de variable libre lors d'un renommage :

$$(\underline{\sum}_{z=M}^{N} P)[y/x] = \begin{cases} \underline{\sum}_{z=M[y/x]}^{N[y/x]} P \text{ si } z = x \\ \underline{\sum}_{k=M[y/x]}^{N[y/x]} P[k/z][y/x] \text{ sinon} \end{cases}$$

où k est une variable *fraîche*, au sens où $k \notin FV(P) \cup \{y\}$.
L'α-conversion, notée $\equiv_\alpha$, est définie comme un jugement inductif.

$$\frac{}{x \equiv_\alpha x} \qquad \frac{}{\underline{n} \equiv_\alpha \underline{n}}$$

L'α-conversion : définition

Il ne faut pas *capturer* de variable libre lors d'un renommage :

$$(\underline{\sum}_{z=M}^{N} P)[y/x] = \begin{cases} \underline{\sum}_{z=M[y/x]}^{N[y/x]} P \text{ si } z = x \\ \underline{\sum}_{k=M[y/x]}^{N[y/x]} P[k/z][y/x] \text{ sinon} \end{cases}$$

où k est une variable *fraîche*, au sens où $k \notin FV(P) \cup \{y\}$.
L'α-conversion, notée $\equiv_\alpha$, est définie comme un jugement inductif.

$$\frac{}{x \equiv_\alpha x} \qquad \frac{}{\underline{n} \equiv_\alpha \underline{n}} \qquad \frac{M \equiv_\alpha M' \qquad N \equiv_\alpha N'}{M \underline{+} N \equiv_\alpha M' \underline{+} N'}$$

L'α-conversion : définition

Il ne faut pas *capturer* de variable libre lors d'un renommage :

$$(\underline{\sum}_{z=M}^{N} P)[y/x] = \begin{cases} \underline{\sum}_{z=M[y/x]}^{N[y/x]} P \text{ si } z = x \\ \underline{\sum}_{k=M[y/x]}^{N[y/x]} P[k/z][y/x] \text{ sinon} \end{cases}$$

où k est une variable *fraîche*, au sens où $k \notin FV(P) \cup \{y\}$.
L'α-conversion, notée $\equiv_\alpha$, est définie comme un jugement inductif.

$$\frac{}{x \equiv_\alpha x} \qquad \frac{}{\underline{n} \equiv_\alpha \underline{n}} \qquad \frac{M \equiv_\alpha M' \qquad N \equiv_\alpha N'}{M \mathbin{\underline{+}} N \equiv_\alpha M' \mathbin{\underline{+}} N'}$$

$$\frac{M \equiv_\alpha M' \qquad N \equiv_\alpha N'}{M \mathbin{\underline{*}} N \equiv_\alpha M' \mathbin{\underline{*}} N'}$$

L'α-conversion : définition

Il ne faut pas *capturer* de variable libre lors d'un renommage :

$$(\underline{\Sigma}_{z=M}^{N} P)[y/x] = \begin{cases} \underline{\Sigma}_{z=M[y/x]}^{N[y/x]} P \text{ si } z = x \\ \underline{\Sigma}_{k=M[y/x]}^{N[y/x]} P[k/z][y/x] \text{ sinon} \end{cases}$$

où k est une variable *fraîche*, au sens où $k \notin FV(P) \cup \{y\}$.
L'α-conversion, notée $\equiv_\alpha$, est définie comme un jugement inductif.

$$\frac{}{x \equiv_\alpha x} \qquad \frac{}{\underline{n} \equiv_\alpha \underline{n}} \qquad \frac{M \equiv_\alpha M' \qquad N \equiv_\alpha N'}{M \mathbin{\underline{+}} N \equiv_\alpha M' \mathbin{\underline{+}} N'}$$

$$\frac{M \equiv_\alpha M' \qquad N \equiv_\alpha N'}{M \mathbin{\underline{*}} N \equiv_\alpha M' \mathbin{\underline{*}} N'} \qquad \frac{}{\underline{\Sigma}_{x=M}^{N} P \equiv_\alpha \underline{\Sigma}_{y=M'}^{N'} P'}$$

L'α-conversion : définition

Il ne faut pas *capturer* de variable libre lors d'un renommage :

$$(\underline{\Sigma}_{z=M}^{N} P)[y/x] = \begin{cases} \underline{\Sigma}_{z=M[y/x]}^{N[y/x]} P \text{ si } z = x \\ \underline{\Sigma}_{k=M[y/x]}^{N[y/x]} P[k/z][y/x] \text{ sinon} \end{cases}$$

où k est une variable *fraîche*, au sens où $k \notin FV(P) \cup \{y\}$.
L'α-conversion, notée $\equiv_\alpha$, est définie comme un jugement inductif.

$$\frac{}{x \equiv_\alpha x} \qquad \frac{}{\underline{n} \equiv_\alpha \underline{n}} \qquad \frac{M \equiv_\alpha M' \qquad N \equiv_\alpha N'}{M \underline{+} N \equiv_\alpha M' \underline{+} N'}$$

$$\frac{M \equiv_\alpha M' \qquad N \equiv_\alpha N'}{M \underline{*} N \equiv_\alpha M' \underline{*} N'} \qquad \frac{M \equiv_\alpha M'}{\underline{\Sigma}_{x=M}^{N} P \equiv_\alpha \underline{\Sigma}_{y=M'}^{N'} P'}$$

L'α-conversion : définition

Il ne faut pas *capturer* de variable libre lors d'un renommage :

$$(\underline{\Sigma}_{z=M}^{N} P)[y/x] = \begin{cases} \underline{\Sigma}_{z=M[y/x]}^{N[y/x]} P \text{ si } z = x \\ \underline{\Sigma}_{k=M[y/x]}^{N[y/x]} P[k/z][y/x] \text{ sinon} \end{cases}$$

où k est une variable *fraîche*, au sens où $k \notin FV(P) \cup \{y\}$.
L'α-conversion, notée $\equiv_\alpha$, est définie comme un jugement inductif.

$$\frac{}{x \equiv_\alpha x} \qquad \frac{}{\underline{n} \equiv_\alpha \underline{n}} \qquad \frac{M \equiv_\alpha M' \qquad N \equiv_\alpha N'}{M \mathbin{\underline{+}} N \equiv_\alpha M' \mathbin{\underline{+}} N'}$$

$$\frac{M \equiv_\alpha M' \qquad N \equiv_\alpha N'}{M \mathbin{\underline{*}} N \equiv_\alpha M' \mathbin{\underline{*}} N'} \qquad \frac{M \equiv_\alpha M' \qquad {\color{red} N \equiv_\alpha N'}}{\underline{\Sigma}_{x=M}^{N} P \equiv_\alpha \underline{\Sigma}_{y=M'}^{N'} P'}$$

L'α-conversion : définition

Il ne faut pas *capturer* de variable libre lors d'un renommage :

$$(\underline{\Sigma}_{z=M}^{N} P)[y/x] = \begin{cases} \underline{\Sigma}_{z=M[y/x]}^{N[y/x]} P \text{ si } z = x \\ \underline{\Sigma}_{k=M[y/x]}^{N[y/x]} P[k/z][y/x] \text{ sinon} \end{cases}$$

où k est une variable *fraîche*, au sens où $k \notin FV(P) \cup \{y\}$.
L'α-conversion, notée $\equiv_\alpha$, est définie comme un jugement inductif.

$$\frac{}{x \equiv_\alpha x} \qquad \frac{}{\underline{n} \equiv_\alpha \underline{n}} \qquad \frac{M \equiv_\alpha M' \qquad N \equiv_\alpha N'}{M \mathbin{\underline{+}} N \equiv_\alpha M' \mathbin{\underline{+}} N'}$$

$$\frac{M \equiv_\alpha M' \qquad N \equiv_\alpha N'}{M \mathbin{\underline{*}} N \equiv_\alpha M' \mathbin{\underline{*}} N'} \qquad \frac{\begin{array}{c} M \equiv_\alpha M' \qquad N \equiv_\alpha N' \\ P[z/x] \equiv_\alpha P'[z/y] \end{array}}{\underline{\Sigma}_{x=M}^{N} P \equiv_\alpha \underline{\Sigma}_{y=M'}^{N'} P'}$$

L'α-conversion : définition

Il ne faut pas *capturer* de variable libre lors d'un renommage :

$$(\underline{\Sigma}_{z=M}^{N} P)[y/x] = \begin{cases} \underline{\Sigma}_{z=M[y/x]}^{N[y/x]} P \text{ si } z = x \\ \underline{\Sigma}_{k=M[y/x]}^{N[y/x]} P[k/z][y/x] \text{ sinon} \end{cases}$$

où k est une variable *fraîche*, au sens où $k \notin FV(P) \cup \{y\}$.
L'α-conversion, notée $\equiv_\alpha$, est définie comme un jugement inductif.

$$\frac{}{x \equiv_\alpha x} \qquad \frac{}{\underline{n} \equiv_\alpha \underline{n}} \qquad \frac{M \equiv_\alpha M' \qquad N \equiv_\alpha N'}{M \mathbin{\underline{+}} N \equiv_\alpha M' \mathbin{\underline{+}} N'}$$

$$\frac{M \equiv_\alpha M' \qquad N \equiv_\alpha N'}{M \mathbin{\underline{*}} N \equiv_\alpha M' \mathbin{\underline{*}} N'} \qquad \frac{M \equiv_\alpha M' \qquad N \equiv_\alpha N' \qquad P[z/x] \equiv_\alpha P'[z/y] \qquad z \notin FV(P) \cup FV(P')}{\underline{\Sigma}_{x=M}^{N} P \equiv_\alpha \underline{\Sigma}_{y=M'}^{N'} P'}$$

L'α-conversion : définition

Il ne faut pas *capturer* de variable libre lors d'un renommage :

$$(\underline{\Sigma}_{z=M}^{N} P)[y/x] = \begin{cases} \underline{\Sigma}_{z=M[y/x]}^{N[y/x]} P \text{ si } z = x \\ \underline{\Sigma}_{k=M[y/x]}^{N[y/x]} P[k/z][y/x] \text{ sinon} \end{cases}$$

où k est une variable *fraîche*, au sens où $k \notin FV(P) \cup \{y\}$.
L'α-conversion, notée $\equiv_\alpha$, est définie comme un jugement inductif.

$$\frac{}{x \equiv_\alpha x} \qquad \frac{}{\underline{n} \equiv_\alpha \underline{n}} \qquad \frac{M \equiv_\alpha M' \qquad N \equiv_\alpha N'}{M \mathbin{\underline{+}} N \equiv_\alpha M' \mathbin{\underline{+}} N'}$$

$$\frac{M \equiv_\alpha M' \qquad N \equiv_\alpha N'}{M \mathbin{\underline{*}} N \equiv_\alpha M' \mathbin{\underline{*}} N'} \qquad \frac{\begin{array}{c} M \equiv_\alpha M' \qquad N \equiv_\alpha N' \\ P[z/x] \equiv_\alpha P'[z/y] \qquad z \notin FV(P) \cup FV(P') \end{array}}{\underline{\Sigma}_{x=M}^{N} P \equiv_\alpha \underline{\Sigma}_{y=M'}^{N'} P'}$$

[Exercice] Programmer renommage et test d'α-conversion en OCaml.

Substitutivité de l'équivalence observationnelle

Quand je programme, je peux toujours remplacer un fragment de code par un autre qui lui est équivalent. Comment rendre cette idée précise ?

Substitutivité de l'équivalence observationnelle

Quand je programme, je peux toujours remplacer un fragment de code par un autre qui lui est équivalent. Comment rendre cette idée précise ?

$$(\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\ \underline{12})\ \underline{+}\ (\underline{3}\ \underline{*}\ \underline{2}) \equiv (\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\ \underline{12})\ \underline{+}\ (\underline{1}\ \underline{+}\ \underline{5})$$

Comment séparer la partie commune des deux termes équivalents ?

Substitutivité de l'équivalence observationnelle

Quand je programme, je peux toujours remplacer un fragment de code par un autre qui lui est équivalent. Comment rendre cette idée précise ?

$$(\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\, \underline{12})\, \underline{+}\, (\underline{3}\, \underline{*}\, \underline{2}) \equiv (\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\, \underline{12})\, \underline{+}\, (\underline{1}\, \underline{+}\, \underline{5})$$

Comment séparer la partie commune des deux termes équivalents ?

Substitutivité de l'équivalence observationnelle

Quand je programme, je peux toujours remplacer un fragment de code par un autre qui lui est équivalent. Comment rendre cette idée précise ?

$$(\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\,\underline{12})\,\underline{+}\,(\underline{3}\,\underline{*}\,\underline{2}) \equiv (\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\,\underline{12})\,\underline{+}\,(\underline{1}\,\underline{+}\,\underline{5})$$

Comment séparer la partie commune des deux termes équivalents ?

Substitutivité de l'équivalence observationnelle

Quand je programme, je peux toujours remplacer un fragment de code par un autre qui lui est équivalent. Comment rendre cette idée précise ?

$$(\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\ \underline{12})\ \underline{+}\ (\underline{3}\ \underline{*}\ \underline{2}) \equiv (\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\ \underline{12})\ \underline{+}\ (\underline{1}\ \underline{+}\ \underline{5})$$

Comment séparer la partie commune des deux termes équivalents ?

$$((\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\ \underline{12})\ \underline{+}\ \mathtt{y})[\underline{3}\ \underline{*}\ \underline{2}/\mathtt{y}] \equiv ((\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\ \underline{12})\ \underline{+}\ \mathtt{y})[\underline{1}\ \underline{+}\ \underline{5}/\mathtt{y}]$$

Ce $N[M/x]$ désigne N où M a été *substitué* aux occurences libres de x.

Substitutivité de l'équivalence observationnelle

Quand je programme, je peux toujours remplacer un fragment de code par un autre qui lui est équivalent. Comment rendre cette idée précise ?

$$(\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\, \underline{12}) \,\underline{+}\, (\underline{3} \,\underline{*}\, \underline{2}) \equiv (\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\, \underline{12}) \,\underline{+}\, (\underline{1} \,\underline{+}\, \underline{5})$$

Comment séparer la partie commune des deux termes équivalents ?

$$((\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\, \underline{12}) \,\underline{+}\, \mathtt{y})[\underline{3} \,\underline{*}\, \underline{2}/\mathtt{y}] \equiv ((\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\, \underline{12}) \,\underline{+}\, \mathtt{y})[\underline{1} \,\underline{+}\, \underline{5}/\mathtt{y}]$$

Ce $N[M/x]$ désigne N où M a été *substitué* aux occurences libres de x.

Substitutivité de l'équivalence observationnelle

Quand je programme, je peux toujours remplacer un fragment de code par un autre qui lui est équivalent. Comment rendre cette idée précise ?

$$(\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\ \underline{12}) \mathbin{\underline{+}} (\underline{3} \mathbin{\underline{*}} \underline{2}) \equiv (\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\ \underline{12}) \mathbin{\underline{+}} (\underline{1} \mathbin{\underline{+}} \underline{5})$$

Comment séparer la partie commune des deux termes équivalents ?

$$((\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\ \underline{12}) \mathbin{\underline{+}} \mathtt{y})[\underline{3} \mathbin{\underline{*}} \underline{2}/\mathtt{y}] \equiv ((\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\ \underline{12}) \mathbin{\underline{+}} \mathtt{y})[\underline{1} \mathbin{\underline{+}} \underline{5}/\mathtt{y}]$$

Ce $N[M/x]$ désigne N où M a été *substitué* aux occurences libres de x.

Substitutivité de l'équivalence observationnelle

Quand je programme, je peux toujours remplacer un fragment de code par un autre qui lui est équivalent. Comment rendre cette idée précise ?

$$(\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\, \underline{12})\, \underline{+}\, (\underline{3}\, \underline{*}\, \underline{2}) \equiv (\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\, \underline{12})\, \underline{+}\, (\underline{1}\, \underline{+}\, \underline{5})$$

Comment séparer la partie commune des deux termes équivalents ?

$$((\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\, \underline{12})\, \underline{+}\, \mathtt{y})[\underline{3}\, \underline{*}\, \underline{2}/\mathtt{y}] \equiv ((\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\, \underline{12})\, \underline{+}\, \mathtt{y})[\underline{1}\, \underline{+}\, \underline{5}/\mathtt{y}]$$

Ce $N[M/x]$ désigne N où M a été *substitué* aux occurences libres de x.

Substitutivité de l'équivalence

Pour tout M, N_1, N_2, x, si $N_1 \equiv N_2$ alors $M[N_1/x] \equiv M[N_2/x]$.

Substitutivité de l'équivalence observationnelle

Quand je programme, je peux toujours remplacer un fragment de code par un autre qui lui est équivalent. Comment rendre cette idée précise ?

$$(\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\, \underline{12}) \mathbin{\underline{+}} (\underline{3} \mathbin{\underline{*}} \underline{2}) \equiv (\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\, \underline{12}) \mathbin{\underline{+}} (\underline{1} \mathbin{\underline{+}} \underline{5})$$

Comment séparer la partie commune des deux termes équivalents ?

$$((\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\, \underline{12}) \mathbin{\underline{+}} \mathtt{y})[\underline{3} \mathbin{\underline{*}} \underline{2}/\mathtt{y}] \equiv ((\underline{\Sigma}_{\mathtt{x}=\underline{1}}^{\underline{4}}\, \underline{12}) \mathbin{\underline{+}} \mathtt{y})[\underline{1} \mathbin{\underline{+}} \underline{5}/\mathtt{y}]$$

Ce $N[M/x]$ désigne N où M a été *substitué* aux occurences libres de x.

Substitutivité de l'équivalence

Pour tout M, N_1, N_2, x, si $N_1 \equiv N_2$ alors $M[N_1/x] \equiv M[N_2/x]$.

Il nous reste à définir formellement l'opération de substitution.

Substitution

La substitution généralisant le renommage, on imite sa définition.

$$
\begin{aligned}
y[M/x] &= \begin{cases} M \text{ si } y = x \\ y \text{ sinon} \end{cases} \\
\underline{n}[M/x] &= \underline{n} \\
(N \mathbin{\underline{+}} P)[M/x] &= N[M/x] \mathbin{\underline{+}} P[M/x] \\
(N \mathbin{\underline{*}} P)[M/x] &= N[M/x] \mathbin{\underline{*}} P[M/x] \\
\left(\underline{\sum}_{y=N}^{P} O\right)[M/x] &= \begin{cases} \underline{\sum}_{z=N[M/x]}^{P[M/x]} O \text{ si } y = x \\ \underline{\sum}_{z=N[M/x]}^{P[M/x]} O[z/y][M/x] \text{ sinon} \end{cases}
\end{aligned}
$$

où, dans la dernière clause, z est fraîche, i.e., $z \notin FV(O) \cup FV(M)$.

[Exercice] Programmer la substitution en OCaml.

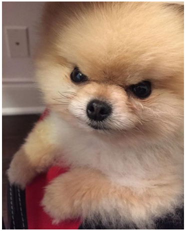
Bilan de cette séance

Un langage de programmation est défini par :

- sa syntaxe,
  - qui comprend des variables *libres* et des variables *liées*,
  - à laquelle on peut appliquer *renommage* et *substitution*,
  - sur laquelle on raisonne "à renommage des variables liées près",
- sa sémantique,
  - une relation d'*évaluation* associant programmes et résultats (ici),
  - qu'on peut implémenter comme un interprète écrit en OCaml,
  - à partir de laquelle on peut définir l'*équivalence observationnelle*.

La prochaine séance adaptera ces concepts à une extension de Marthe.

Pour la prochaine fois



[Exercice] Formuler une sémantique équivalente de Marthe utilisant un jugement auxiliaire $P; \sigma; x; m; n \Downarrow_{\Sigma} p$ pour l'évaluation des sommes.

$$\frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n \qquad P; \sigma; x; m; n \Downarrow_{\Sigma} p}{\underline{\sum}_{x=M}^{N} P; \sigma \Downarrow p}$$

Cette sémantique devra être équivalente aux deux autres.

(*Indice* : il s'agit de reformuler le code OCaml.)

Cours 2

Sémantique de Marthe^{++}

Au delà de Marthe

La semaine dernière, on a étudié la syntaxe, la sémantique et l'équivalence observationnelle de Marthe.

$$|\mathcal{M}| \ni M, N, P ::= x \mid \underline{n} \mid M \mathbin{\underline{+}} N \mid M \mathbin{\underline{*}} N \mid \underline{\sum}_{x=M}^{N} P$$

$$\frac{}{\underline{n}; \sigma \Downarrow n} \qquad \frac{}{x; \sigma \Downarrow \sigma(x)} \qquad \frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n}{M \mathbin{\underline{+}} N; \sigma \Downarrow m + n} \qquad \ldots$$

$$M \equiv N \overset{\text{def}}{=} \forall \sigma \in Env, \forall n \in \mathbb{N},\ M; \sigma \Downarrow n \Leftrightarrow N; \sigma \Downarrow n$$

Au delà de Marthe

La semaine dernière, on a étudié la syntaxe, la sémantique et l'équivalence observationnelle de Marthe.

$$|\mathcal{M}| \ni M, N, P ::= x \mid \underline{n} \mid M \mathbin{\underline{+}} N \mid M \mathbin{\underline{*}} N \mid \underline{\Sigma}_{x=M}^{N} P$$

$$\frac{}{\underline{n}; \sigma \Downarrow n} \qquad \frac{}{x; \sigma \Downarrow \sigma(x)} \qquad \frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n}{M \mathbin{\underline{+}} N; \sigma \Downarrow m + n} \qquad \cdots$$

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in \mathit{Env}, \forall n \in \mathbb{N},\ M; \sigma \Downarrow n \Leftrightarrow N; \sigma \Downarrow n$$

Au delà de Marthe

La semaine dernière, on a étudié la syntaxe, la sémantique et l'équivalence observationnelle de Marthe.

$$|\mathcal{M}| \ni M, N, P ::= x \mid \underline{n} \mid M \mathbin{\underline{+}} N \mid M \mathbin{\underline{*}} N \mid \underline{\Sigma}_{x=M}^{N} P$$

$$\frac{}{\underline{n}; \sigma \Downarrow n} \qquad \frac{}{x; \sigma \Downarrow \sigma(x)} \qquad \frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n}{M \mathbin{\underline{+}} N; \sigma \Downarrow m + n} \qquad \cdots$$

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in Env, \forall n \in \mathbb{N},\ M; \sigma \Downarrow n \Leftrightarrow N; \sigma \Downarrow n$$

Au delà de Marthe

La semaine dernière, on a étudié la syntaxe, la sémantique et l'équivalence observationnelle de Marthe.

$$|\mathcal{M}| \ni M, N, P ::= x \mid \underline{n} \mid M \mathbin{\underline{+}} N \mid M \mathbin{\underline{*}} N \mid \underline{\Sigma}_{x=M}^{N} P$$

$$\frac{}{\underline{n}; \sigma \Downarrow n} \qquad \frac{}{x; \sigma \Downarrow \sigma(x)} \qquad \frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n}{M \mathbin{\underline{+}} N; \sigma \Downarrow m + n} \qquad \cdots$$

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in \mathit{Env}, \forall n \in \mathbb{N},\ M; \sigma \Downarrow n \Leftrightarrow N; \sigma \Downarrow n$$

Au delà de Marthe

La semaine dernière, on a étudié la syntaxe, la sémantique et l'équivalence observationnelle de Marthe.

$$|\mathcal{M}| \ni M, N, P ::= x \mid \underline{n} \mid M \underline{+} N \mid M \underline{*} N \mid \underline{\Sigma}_{x=M}^{N} P$$

$$\frac{}{\underline{n}; \sigma \Downarrow n} \qquad \frac{}{x; \sigma \Downarrow \sigma(x)} \qquad \frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n}{M \underline{+} N; \sigma \Downarrow m + n} \qquad \ldots$$

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in Env, \forall n \in \mathbb{N},\ M; \sigma \Downarrow n \Leftrightarrow N; \sigma \Downarrow n$$

Que manque-t-il à Marthe pour être un vrai langage de programmation ?

Au delà de Marthe

La semaine dernière, on a étudié la syntaxe, la sémantique et l'équivalence observationnelle de Marthe.

$$|\mathcal{M}| \ni M, N, P ::= x \mid \underline{n} \mid M \mathbin{\underline{+}} N \mid M \mathbin{\underline{*}} N \mid \underline{\Sigma}_{x=M}^{N} P$$

$$\frac{}{\underline{n}; \sigma \Downarrow n} \qquad \frac{}{x; \sigma \Downarrow \sigma(x)} \qquad \frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n}{M \mathbin{\underline{+}} N; \sigma \Downarrow m + n} \qquad \cdots$$

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in Env, \forall n \in \mathbb{N},\ M; \sigma \Downarrow n \Leftrightarrow N; \sigma \Downarrow n$$

Que manque-t-il à Marthe pour être un vrai langage de programmation ?

- Des *types de données* plus riches : booléens, paires, listes, etc.

Au delà de Marthe

La semaine dernière, on a étudié la syntaxe, la sémantique et l'équivalence observationnelle de Marthe.

$$|\mathcal{M}| \ni M, N, P ::= x \mid \underline{n} \mid M \underline{+} N \mid M \underline{*} N \mid \underline{\Sigma}_{x=M}^{N} P$$

$$\frac{}{\underline{n}; \sigma \Downarrow n} \qquad \frac{}{x; \sigma \Downarrow \sigma(x)} \qquad \frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n}{M \underline{+} N; \sigma \Downarrow m + n} \qquad \ldots$$

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in Env, \forall n \in \mathbb{N},\ M; \sigma \Downarrow n \Leftrightarrow N; \sigma \Downarrow n$$

Que manque-t-il à Marthe pour être un vrai langage de programmation ?

- Des *types de données* plus riches : booléens, paires, listes, etc.
- Des constructions de contrôle, par exemple les *conditionnelles*.

Au delà de Marthe

La semaine dernière, on a étudié la syntaxe, la sémantique et l'équivalence observationnelle de Marthe.

$$|\mathcal{M}| \ni M, N, P ::= x \mid \underline{n} \mid M \mathbin{\underline{+}} N \mid M \mathbin{\underline{*}} N \mid \underline{\Sigma}_{x=M}^{N} P$$

$$\frac{}{\underline{n}; \sigma \Downarrow n} \qquad \frac{}{x; \sigma \Downarrow \sigma(x)} \qquad \frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n}{M \mathbin{\underline{+}} N; \sigma \Downarrow m + n} \qquad \cdots$$

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in Env, \forall n \in \mathbb{N},\ M; \sigma \Downarrow n \Leftrightarrow N; \sigma \Downarrow n$$

Que manque-t-il à Marthe pour être un vrai langage de programmation ?

- Des *types de données* plus riches : booléens, paires, listes, etc.
- Des constructions de contrôle, par exemple les *conditionnelles*.
- Une capacité d'abstraction, par exemple des *fonctions*.

Au delà de Marthe

La semaine dernière, on a étudié la syntaxe, la sémantique et l'équivalence observationnelle de Marthe.

$$|\mathcal{M}| \ni M, N, P ::= x \mid \underline{n} \mid M \mathbin{\underline{+}} N \mid M \mathbin{\underline{*}} N \mid \underline{\Sigma}_{x=M}^{N} P$$

$$\frac{}{\underline{n}; \sigma \Downarrow n} \qquad \frac{}{x; \sigma \Downarrow \sigma(x)} \qquad \frac{M; \sigma \Downarrow m \qquad N; \sigma \Downarrow n}{M \mathbin{\underline{+}} N; \sigma \Downarrow m + n} \qquad \ldots$$

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in Env, \forall n \in \mathbb{N},\ M; \sigma \Downarrow n \Leftrightarrow N; \sigma \Downarrow n$$

Que manque-t-il à Marthe pour être un vrai langage de programmation ?

- Des *types de données* plus riches : booléens, paires, listes, etc.
- Des constructions de contrôle, par exemple les *conditionnelles*.
- Une capacité d'abstraction, par exemple des *fonctions*.
- ...

Construire Marthe^{++}

Durant cette séance, on va progressivement construire Marthe^{++} :

1. ajout d'une construction à la syntaxe, puis
2. description de la sémantique de cette construction.

On verra chemin faisant que certaines constructions justifient des modifications de notre panoplie mathématique, par exemple une généralisation de la définition de l'équivalence observationnelle.

Partir sur des bases simples

Pour généraliser Marthe, on va commencer par :

- ne pas se limiter aux entiers comme seul type de données,
- supprimer la construction de somme formelle, trop spécifique.

Partir sur des bases simples

Pour généraliser Marthe, on va commencer par :

- ne pas se limiter aux entiers comme seul type de données,
- supprimer la construction de somme formelle, trop spécifique.

Cela nous mène à la syntaxe ci-dessous, qui inclue une catégorie de *valeurs*.

$$M, N, P ::= \underline{n}_{\mathtt{i}} \mid \underline{b}_{\mathtt{b}} \mid M\ bop\ N \mid \mathtt{let}\ x = M\ \mathtt{in}\ N \qquad (n \in \mathbb{N}, b \in \mathbb{B})$$

$$V, W ::= \underline{n}_{\mathtt{i}} \mid \underline{b}_{\mathtt{b}} \qquad bop ::= \underline{+} \mid \underline{*} \mid \underline{\wedge} \mid \underline{\vee} \mid \ldots$$

Partir sur des bases simples

Pour généraliser Marthe, on va commencer par :

- ne pas se limiter aux entiers comme seul type de données,
- supprimer la construction de somme formelle, trop spécifique.

Cela nous mène à la syntaxe ci-dessous, qui inclue une catégorie de *valeurs*.

$$M, N, P ::= \underline{n}_{\mathtt{i}} \mid \underline{b}_{\mathtt{b}} \mid M\ bop\ N \mid \mathtt{let}\ x = M\ \mathtt{in}\ N \qquad (n \in \mathbb{N}, b \in \mathbb{B})$$

$$V, W ::= \underline{n}_{\mathtt{i}} \mid \underline{b}_{\mathtt{b}} \qquad bop ::= \underline{+} \mid \underline{*} \mid \underline{\wedge} \mid \underline{\vee} \mid \ldots$$

La sémantique devient $M; \sigma \Downarrow V$, où σ associe des valeurs aux variables.

$$\frac{}{\underline{n}_{\mathtt{i}}; \sigma \Downarrow \underline{n}_{\mathtt{i}}} \qquad \frac{}{\underline{b}_{\mathtt{b}}; \sigma \Downarrow \underline{b}_{\mathtt{b}}}$$

Partir sur des bases simples

Pour généraliser Marthe, on va commencer par :

- ne pas se limiter aux entiers comme seul type de données,
- supprimer la construction de somme formelle, trop spécifique.

Cela nous mène à la syntaxe ci-dessous, qui inclue une catégorie de *valeurs*.

$$M, N, P ::= \underline{n}_{\mathtt{i}} \mid \underline{b}_{\mathtt{b}} \mid M\ bop\ N \mid \mathtt{let}\ x = M\ \mathtt{in}\ N \qquad (n \in \mathbb{N}, b \in \mathbb{B})$$

$$V, W ::= \underline{n}_{\mathtt{i}} \mid \underline{b}_{\mathtt{b}} \qquad bop ::= \underline{+} \mid \underline{*} \mid \underline{\wedge} \mid \underline{\vee} \mid \ldots$$

La sémantique devient $M; \sigma \Downarrow V$, où σ associe des valeurs aux variables.

$$\frac{}{\underline{n}_{\mathtt{i}}; \sigma \Downarrow \underline{n}_{\mathtt{i}}} \qquad \frac{}{\underline{b}_{\mathtt{b}}; \sigma \Downarrow \underline{b}_{\mathtt{b}}} \qquad \frac{M; \sigma \Downarrow \underline{n_1}_{\mathtt{i}} \qquad N; \sigma \Downarrow \underline{n_2}_{\mathtt{i}}}{M \mathbin{\underline{+}} N; \sigma \Downarrow \underline{n_1 + n_2}_{\mathtt{i}}}$$

$$\frac{M; \sigma \Downarrow \underline{b_1}_{\mathtt{b}} \qquad N; \sigma \Downarrow \underline{b_2}_{\mathtt{b}}}{M \mathbin{\underline{\wedge}} N; \sigma \Downarrow \underline{b_1 \wedge b_2}_{\mathtt{b}}} \qquad \ldots$$

Partir sur des bases simples

Pour généraliser Marthe, on va commencer par :

- ne pas se limiter aux entiers comme seul type de données,
- supprimer la construction de somme formelle, trop spécifique.

Cela nous mène à la syntaxe ci-dessous, qui inclue une catégorie de *valeurs*.

$$M, N, P ::= \underline{n}_{\mathtt{i}} \mid \underline{b}_{\mathtt{b}} \mid M\ bop\ N \mid \mathtt{let}\ x = M\ \mathtt{in}\ N \qquad (n \in \mathbb{N}, b \in \mathbb{B})$$

$$V, W ::= \underline{n}_{\mathtt{i}} \mid \underline{b}_{\mathtt{b}} \qquad bop ::= \underline{+} \mid \underline{*} \mid \underline{\wedge} \mid \underline{\vee} \mid \dots$$

La sémantique devient $M; \sigma \Downarrow V$, où σ associe des valeurs aux variables.

$$\frac{}{\underline{n}_{\mathtt{i}}; \sigma \Downarrow \underline{n}_{\mathtt{i}}} \qquad \frac{}{\underline{b}_{\mathtt{b}}; \sigma \Downarrow \underline{b}_{\mathtt{b}}} \qquad \frac{M; \sigma \Downarrow \underline{n_1}_{\mathtt{i}} \qquad N; \sigma \Downarrow \underline{n_2}_{\mathtt{i}}}{M \mathbin{\underline{+}} N; \sigma \Downarrow \underline{n_1 + n_2}_{\mathtt{i}}}$$

$$\frac{M; \sigma \Downarrow \underline{b_1}_{\mathtt{b}} \qquad N; \sigma \Downarrow \underline{b_2}_{\mathtt{b}}}{M \mathbin{\underline{\wedge}} N; \sigma \Downarrow \underline{b_1 \wedge b_2}_{\mathtt{b}}} \qquad \dots \qquad \frac{M; \sigma \Downarrow V \qquad N; \sigma[x \mapsto V] \Downarrow V'}{\mathtt{let}\ x = M\ \mathtt{in}\ N; \sigma \Downarrow V'}$$

Partir sur des bases simples

Pour généraliser Marthe, on va commencer par :

- ne pas se limiter aux entiers comme seul type de données,
- supprimer la construction de somme formelle, trop spécifique.

Cela nous mène à la syntaxe ci-dessous, qui inclue une catégorie de *valeurs*.

$$M, N, P ::= \underline{n}_{\mathtt{i}} \mid \underline{b}_{\mathtt{b}} \mid M \; bop \; N \mid \mathtt{let}\ x = M \ \mathtt{in}\ N \qquad (n \in \mathbb{N}, b \in \mathbb{B})$$

$$V, W ::= \underline{n}_{\mathtt{i}} \mid \underline{b}_{\mathtt{b}} \qquad\qquad bop ::= \underline{+} \mid \underline{*} \mid \underline{\wedge} \mid \underline{\vee} \mid \dots$$

La sémantique devient $M; \sigma \Downarrow V$, où σ associe des valeurs aux variables.

$$\frac{}{\underline{n}_{\mathtt{i}}; \sigma \Downarrow \underline{n}_{\mathtt{i}}} \qquad \frac{}{\underline{b}_{\mathtt{b}}; \sigma \Downarrow \underline{b}_{\mathtt{b}}} \qquad \frac{M; \sigma \Downarrow \underline{n_1}_{\mathtt{i}} \qquad N; \sigma \Downarrow \underline{n_2}_{\mathtt{i}}}{M \underline{+} N; \sigma \Downarrow \underline{n_1 + n_2}_{\mathtt{i}}}$$

$$\frac{M; \sigma \Downarrow \underline{b_1}_{\mathtt{b}} \qquad N; \sigma \Downarrow \underline{b_2}_{\mathtt{b}}}{M \underline{\wedge} N; \sigma \Downarrow \underline{b_1 \wedge b_2}_{\mathtt{b}}} \qquad \dots \qquad \frac{M; \sigma \Downarrow V \qquad N; \sigma[x \mapsto V] \Downarrow V'}{\mathtt{let}\ x = M \ \mathtt{in}\ N; \sigma \Downarrow V'}$$

[Exercice] Ajouter négation booléenne $\underline{\neg}$ et comparaison d'entiers $\underline{\leq}$.

Les paires

On voudrait manipuler des paires de valeurs. Qu'ajouter à la syntaxe ?

Les paires

On voudrait manipuler des paires de valeurs. Qu'ajouter à la syntaxe ?

$$M, N, P ::= \cdots \mid (M, N) \mid \texttt{fst}\ M \mid \texttt{snd}\ M \qquad V, W ::= \underline{n}_{\texttt{i}} \mid \underline{b}_{\texttt{b}} \mid (V, W)$$

Quelle sémantique ?

Les paires

On voudrait manipuler des paires de valeurs. Qu'ajouter à la syntaxe ?

$$M, N, P ::= \cdots \mid (M, N) \mid \texttt{fst}\ M \mid \texttt{snd}\ M \qquad V, W ::= \underline{n}_{\mathtt{i}} \mid \underline{b}_{\mathtt{b}} \mid (V, W)$$

Quelle sémantique ?

$$\frac{M; \sigma \Downarrow V \qquad N; \sigma \Downarrow W}{(M, N); \sigma \Downarrow (V, W)} \qquad \frac{M; \sigma \Downarrow (V, W)}{\texttt{fst}\ M; \sigma \Downarrow V} \qquad \frac{M; \sigma \Downarrow (V, W)}{\texttt{snd}\ M; \sigma \Downarrow W}$$

Les paires

On voudrait manipuler des paires de valeurs. Qu'ajouter à la syntaxe ?

$$M, N, P ::= \cdots \mid (M, N) \mid \texttt{fst}\ M \mid \texttt{snd}\ M \qquad V, W ::= \underline{n}_{\texttt{i}} \mid \underline{b}_{\texttt{b}} \mid (V, W)$$

Quelle sémantique ?

$$\frac{M; \sigma \Downarrow V \qquad N; \sigma \Downarrow W}{(M, N); \sigma \Downarrow (V, W)} \qquad \frac{M; \sigma \Downarrow (V, W)}{\texttt{fst}\ M; \sigma \Downarrow V} \qquad \frac{M; \sigma \Downarrow (V, W)}{\texttt{snd}\ M; \sigma \Downarrow W}$$

[Exercice] Donner la valeur V telle que le jugement

$$\texttt{let x} = (\underline{3}_{\texttt{i}} \mathbin{\underline{+}} \underline{5}_{\texttt{i}}, \underline{\mathit{true}}_{\texttt{b}})\ \texttt{in}\ (\texttt{snd x}, \texttt{fst x}); \emptyset \Downarrow V$$

soit dérivable, ainsi que la dérivation correspondante.

Les conditionnelles

On voudrait ajouter une instruction conditionnelle à la OCaml au langage.

$$M ::= \cdots \mid \texttt{if}\ M\ \texttt{then}\ N\ \texttt{else}\ P$$

Quelle serait sa sémantique ? “Si l’évaluation de M renvoie vrai, le résultat est celui de N, sinon c’est celui de P”.

Les conditionnelles

On voudrait ajouter une instruction conditionnelle à la OCaml au langage.

$$M ::= \cdots \mid \texttt{if}\ M\ \texttt{then}\ N\ \texttt{else}\ P$$

Quelle serait sa sémantique ? “Si l’évaluation de M renvoie vrai, le résultat est celui de N, sinon c’est celui de P”.

$$\frac{M; \sigma \Downarrow \underline{\mathit{true}}_{\mathtt{b}} \qquad N; \sigma \Downarrow V}{\texttt{if}\ M\ \texttt{then}\ N\ \texttt{else}\ P; \sigma \Downarrow V} \qquad \frac{M; \sigma \Downarrow \underline{\mathit{false}}_{\mathtt{b}} \qquad P; \sigma \Downarrow V}{\texttt{if}\ M\ \texttt{then}\ N\ \texttt{else}\ P; \sigma \Downarrow V}$$

Les conditionnelles

On voudrait ajouter une instruction conditionnelle à la OCaml au langage.

$$M ::= \cdots \mid \texttt{if } M \texttt{ then } N \texttt{ else } P$$

Quelle serait sa sémantique ? “Si l'évaluation de M renvoie vrai, le résultat est celui de N, sinon c'est celui de P”.

$$\frac{M; \sigma \Downarrow \underline{\mathit{true}}_{\mathtt{b}} \qquad N; \sigma \Downarrow V}{\texttt{if } M \texttt{ then } N \texttt{ else } P; \sigma \Downarrow V} \qquad \frac{M; \sigma \Downarrow \underline{\mathit{false}}_{\mathtt{b}} \qquad P; \sigma \Downarrow V}{\texttt{if } M \texttt{ then } N \texttt{ else } P; \sigma \Downarrow V}$$

[Exercice] L'équation donnée ci-dessous vous semble-t-elle valide dans le langage décrit jusqu'ici, pour M, N, P clos et s'évaluant sans erreurs ?

$$\texttt{if } M \texttt{ then } N \texttt{ else } P \equiv \texttt{let x} = (N, P) \texttt{ in if } M \texttt{ then fst x else snd x}.$$

Et son analogue en OCaml ?

De nouvelles erreurs ?

Lors du cours précédent, on a vu que Marthe vérifiait la propriété suivante.

Propriété

Il existe V tel que $M; \sigma \Downarrow V$ si et seulement si $FV(M) \subseteq \sigma^{-1}(\mathbb{N})$.

Est-ce encore le cas pour le langage du transparent précédent ?

De nouvelles erreurs ?

Lors du cours précédent, on a vu que Marthe vérifiait la propriété suivante.

Propriété

Il existe V tel que $M; \sigma \Downarrow V$ si et seulement si $FV(M) \subseteq \sigma^{-1}(\mathbb{N})$.

Est-ce encore le cas pour le langage du transparent précédent ?

- Est-ce que si $M; \sigma \Downarrow V$ alors $FV(M) \subseteq \sigma^{-1}(\mathbb{N})$? Non :

$$\texttt{if}\ \underline{\mathit{true}}_{\texttt{b}}\ \texttt{then}\ \underline{42}_{\texttt{i}}\ \texttt{else}\ \texttt{x}; \emptyset \Downarrow \underline{\mathit{true}}_{\texttt{b}}.$$

De nouvelles erreurs ?

Lors du cours précédent, on a vu que Marthe vérifiait la propriété suivante.

Propriété

Il existe V tel que $M; \sigma \Downarrow V$ si et seulement si $FV(M) \subseteq \sigma^{-1}(\mathbb{N})$.

Est-ce encore le cas pour le langage du transparent précédent ?

- Est-ce que si $M; \sigma \Downarrow V$ alors $FV(M) \subseteq \sigma^{-1}(\mathbb{N})$? Non :

$$\texttt{if}\ \underline{\mathit{true}}_{\texttt{b}}\ \texttt{then}\ \underline{42}_{\texttt{i}}\ \texttt{else}\ \texttt{x}; \emptyset \Downarrow \underline{\mathit{true}}_{\texttt{b}}.$$

- Est-ce que si $FV(M) \subseteq \sigma^{-1}(\mathbb{N})$, il existe V tel que $M; \sigma \Downarrow V$?

De nouvelles erreurs ?

Lors du cours précédent, on a vu que Marthe vérifiait la propriété suivante.

Propriété (Marthe uniquement)

Il existe V tel que $M; \sigma \Downarrow V$ si et seulement si $FV(M) \subseteq \sigma^{-1}(\mathbb{N})$.

Est-ce encore le cas pour le langage du transparent précédent ?

- Est-ce que si $M; \sigma \Downarrow V$ alors $FV(M) \subseteq \sigma^{-1}(\mathbb{N})$? Non :

$$\texttt{if } \underline{\mathit{true}}_{\texttt{b}} \texttt{ then } \underline{42}_{\texttt{i}} \texttt{ else } \texttt{x}; \emptyset \Downarrow \underline{\mathit{true}}_{\texttt{b}}.$$

- Est-ce que si $FV(M) \subseteq \sigma^{-1}(\mathbb{N})$, il existe V tel que $M; \sigma \Downarrow V$? Non, à cause de potentielles erreurs, comme dans le terme

$$\texttt{if } (\underline{1}_{\texttt{i}}, \underline{2}_{\texttt{i}}) \texttt{ then } \underline{\mathit{true}}_{\texttt{b}} \texttt{ else } \underline{\mathit{false}}_{\texttt{b}}; \emptyset \Downarrow ??.$$

On verra à la prochaine séance comment s'en prémunir.

Les fonctions de seconde classe (1/2)

On ajoute des constructions permettant d'abstraire un sous-terme pour pouvoir le réutiliser plusieurs fois : définition de fonction et application.

$$M, N, P ::= \cdots \mid \texttt{def}\ f\ x = M\ \texttt{in}\ N \mid f\ M$$

Les fonctions de seconde classe (1/2)

On ajoute des constructions permettant d'abstraire un sous-terme pour pouvoir le réutiliser plusieurs fois : définition de fonction et application.

$$M, N, P ::= \cdots \mid \texttt{def}\ f\ x = M\ \texttt{in}\ N \mid f\ M$$

Les fonctions de seconde classe (1/2)

On ajoute des constructions permettant d'abstraire un sous-terme pour pouvoir le réutiliser plusieurs fois : définition de fonction et application.

$$M, N, P ::= \cdots \mid \texttt{def}\ f\ x = M\ \texttt{in}\ N \mid f\ M$$

Les fonctions de seconde classe (1/2)

On ajoute des constructions permettant d'abstraire un sous-terme pour pouvoir le réutiliser plusieurs fois : définition de fonction et application.

$$M, N, P ::= \cdots \mid \texttt{def}\ f\ x = M\ \texttt{in}\ N \mid f\ M$$

On désigne par f, g, h des variables distinctes de x, y, z. On dit que ces deux familles de variables constituent des espaces de noms différents.

Les fonctions de seconde classe (1/2)

On ajoute des constructions permettant d'abstraire un sous-terme pour pouvoir le réutiliser plusieurs fois : définition de fonction et application.

$$M, N, P ::= \cdots \mid \texttt{def}\ f\ x = M\ \texttt{in}\ N \mid f\ M$$

On désigne par f, g, h des variables distinctes de x, y, z. On dit que ces deux familles de variables constituent des espaces de noms différents. Par conséquent, on ne peut pas "mélanger" variables ordinaires et noms de fonction, et donc, pour des raisons purement syntaxiques :

Les fonctions de seconde classe (1/2)

On ajoute des constructions permettant d'abstraire un sous-terme pour pouvoir le réutiliser plusieurs fois : définition de fonction et application.

$$M, N, P ::= \cdots \mid \texttt{def}\ f\ x = M\ \texttt{in}\ N \mid f\ M$$

On désigne par f, g, h des variables distinctes de x, y, z. On dit que ces deux familles de variables constituent des espaces de noms différents. Par conséquent, on ne peut pas "mélanger" variables ordinaires et noms de fonction, et donc, pour des raisons purement syntaxiques :

- un terme ne peut pas s'évaluer vers une fonction,

$$\texttt{def f x} = \texttt{x in f} \text{ n'étant pas un terme,}$$

Les fonctions de seconde classe (1/2)

On ajoute des constructions permettant d'abstraire un sous-terme pour pouvoir le réutiliser plusieurs fois : définition de fonction et application.

$$M, N, P ::= \cdots \mid \texttt{def}\ f\ x = M\ \texttt{in}\ N \mid f\ M$$

On désigne par f, g, h des variables distinctes de x, y, z. On dit que ces deux familles de variables constituent des espaces de noms différents. Par conséquent, on ne peut pas "mélanger" variables ordinaires et noms de fonction, et donc, pour des raisons purement syntaxiques :

- un terme ne peut pas s'évaluer vers une fonction,

  $\texttt{def f x} = \texttt{x in f}$ n'étant pas un terme,

- une fonction ne peut pas prendre en argument une autre fonction,

  $\texttt{def f g} = \texttt{g}\ \underline{1}_{\texttt{i}}\ \texttt{in}\ M$ non plus.

Les fonctions de seconde classe (1/2)

On ajoute des constructions permettant d'abstraire un sous-terme pour pouvoir le réutiliser plusieurs fois : définition de fonction et application.

$$M, N, P ::= \cdots \mid \texttt{def}\ f\ x = M\ \texttt{in}\ N \mid f\ M$$

On désigne par f, g, h des variables distinctes de x, y, z. On dit que ces deux familles de variables constituent des espaces de noms différents. Par conséquent, on ne peut pas "mélanger" variables ordinaires et noms de fonction, et donc, pour des raisons purement syntaxiques :

- un terme ne peut pas s'évaluer vers une fonction,

  $\texttt{def f x} = \texttt{x in f}$ n'étant pas un terme,

- une fonction ne peut pas prendre en argument une autre fonction,

  $\texttt{def f g} = \texttt{g}\ \underline{1}_{\texttt{i}}\ \texttt{in}\ M$ non plus.

Les objets soumis à de telles restrictions sont dits de seconde classe.

Les fonctions de seconde classe (2/2)

La sémantique doit maintenant être paramétrée par deux environnements, un par espace de nom :

Les fonctions de seconde classe (2/2)

La sémantique doit maintenant être paramétrée par deux environnements, un par espace de nom :

- l'environnement σ usuel, associant aux variables $\mathtt{x}, \mathtt{y}, \mathtt{z}$ des valeurs,

Les fonctions de seconde classe (2/2)

La sémantique doit maintenant être paramétrée par deux environnements, un par espace de nom :

- l'environnement σ usuel, associant aux variables $\mathtt{x}, \mathtt{y}, \mathtt{z}$ des valeurs,
- un nouvel environnement ϕ, associant aux variables $\mathtt{f}, \mathtt{g}, \mathtt{h}$ des. . . ?

Les fonctions de seconde classe (2/2)

La sémantique doit maintenant être paramétrée par deux environnements, un par espace de nom :

- l'environnement σ usuel, associant aux variables $\mathtt{x}, \mathtt{y}, \mathtt{z}$ des valeurs,
- un nouvel environnement ϕ, associant aux variables $\mathtt{f}, \mathtt{g}, \mathtt{h}$ des... ?

Essayons d'écrire les règles de notre jugement $M; \sigma; \phi \Downarrow V$.

Les fonctions de seconde classe (2/2)

La sémantique doit maintenant être paramétrée par deux environnements, un par espace de nom :

- l'environnement σ usuel, associant aux variables $\mathtt{x}, \mathtt{y}, \mathtt{z}$ des valeurs,
- un nouvel environnement ϕ, associant aux variables $\mathtt{f}, \mathtt{g}, \mathtt{h}$ des. . . ?

Essayons d'écrire les règles de notre jugement $M; \sigma; \phi \Downarrow V$.

$$\frac{}{x; \sigma; \phi \Downarrow \sigma(x)} \qquad \dots \qquad \frac{M; \sigma; \phi \Downarrow \underline{m}_{\mathtt{i}} \qquad N; \sigma; \phi \Downarrow \underline{n}_{\mathtt{i}}}{M \underline{+} N; \sigma; \phi \Downarrow \underline{m + n}_{\mathtt{i}}} \qquad \dots$$

Les fonctions de seconde classe (2/2)

La sémantique doit maintenant être paramétrée par deux environnements, un par espace de nom :

- l'environnement σ usuel, associant aux variables $\mathtt{x}, \mathtt{y}, \mathtt{z}$ des valeurs,
- un nouvel environnement ϕ, associant aux variables $\mathtt{f}, \mathtt{g}, \mathtt{h}$ des. . . ?

Essayons d'écrire les règles de notre jugement $M; \sigma; \phi \Downarrow V$.

$$\frac{}{x; \sigma; \phi \Downarrow \sigma(x)} \qquad \dots \qquad \frac{M; \sigma; \phi \Downarrow \underline{m}_{\mathtt{i}} \qquad N; \sigma; \phi \Downarrow \underline{n}_{\mathtt{i}}}{M \mathbin{\underline{+}} N; \sigma; \phi \Downarrow \underline{m + n}_{\mathtt{i}}} \qquad \dots$$

$$\frac{N; \sigma; \phi[f \mapsto ?] \Downarrow V}{\mathtt{def}\ f\ x = M\ \mathtt{in}\ N; \sigma; \phi \Downarrow V}$$

Les fonctions de seconde classe (2/2)

La sémantique doit maintenant être paramétrée par deux environnements, un par espace de nom :

- l'environnement σ usuel, associant aux variables $\mathtt{x}, \mathtt{y}, \mathtt{z}$ des valeurs,
- un nouvel environnement ϕ, associant aux variables $\mathtt{f}, \mathtt{g}, \mathtt{h}$ des...?

Essayons d'écrire les règles de notre jugement $M; \sigma; \phi \Downarrow V$.

$$\frac{}{x; \sigma; \phi \Downarrow \sigma(x)} \qquad \cdots \qquad \frac{M; \sigma; \phi \Downarrow \underline{m}_{\mathtt{i}} \qquad N; \sigma; \phi \Downarrow \underline{n}_{\mathtt{i}}}{M \underline{+} N; \sigma; \phi \Downarrow \underline{m + n}_{\mathtt{i}}} \qquad \cdots$$

$$\frac{N; \sigma; \phi[f \mapsto ?] \Downarrow V}{\mathtt{def}\ f\ x = M\ \mathtt{in}\ N; \sigma; \phi \Downarrow V} \qquad \frac{N; \sigma; \phi \Downarrow V_a \qquad ?; ?[? \mapsto V_a]; ? \Downarrow V}{f\ N; \sigma; \phi \Downarrow V}$$

Les fonctions de seconde classe (2/2)

La sémantique doit maintenant être paramétrée par deux environnements, un par espace de nom :

- l'environnement σ usuel, associant aux variables $\mathtt{x}, \mathtt{y}, \mathtt{z}$ des valeurs,
- un nouvel environnement ϕ, associant aux variables $\mathtt{f}, \mathtt{g}, \mathtt{h}$ des. . . ?

Essayons d'écrire les règles de notre jugement $M; \sigma; \phi \Downarrow V$.

$$\frac{}{x; \sigma; \phi \Downarrow \sigma(x)} \qquad \dots \qquad \frac{M; \sigma; \phi \Downarrow \underline{m}_{\mathtt{i}} \qquad N; \sigma; \phi \Downarrow \underline{n}_{\mathtt{i}}}{M \underline{+} N; \sigma; \phi \Downarrow \underline{m + n}_{\mathtt{i}}} \qquad \dots$$

$$\frac{N; \sigma; \phi[f \mapsto \color{red}{?}] \Downarrow V}{\mathtt{def}\ f\ x = M\ \mathtt{in}\ N; \sigma; \phi \Downarrow V} \qquad \frac{N; \sigma; \phi \Downarrow V_a \qquad \color{red}{?}; \color{red}{?}[\color{red}{?} \mapsto V_a]; \color{red}{?} \Downarrow V}{f\ N; \sigma; \phi \Downarrow V}$$

Pour exécuter un appel à la fonction f, on doit :

Les fonctions de seconde classe (2/2)

La sémantique doit maintenant être paramétrée par deux environnements, un par espace de nom :

- l'environnement σ usuel, associant aux variables $\mathtt{x}, \mathtt{y}, \mathtt{z}$ des valeurs,
- un nouvel environnement ϕ, associant aux variables $\mathtt{f}, \mathtt{g}, \mathtt{h}$ des. . . ?

Essayons d'écrire les règles de notre jugement $M; \sigma; \phi \Downarrow V$.

$$\frac{}{x; \sigma; \phi \Downarrow \sigma(x)} \qquad \ldots \qquad \frac{M; \sigma; \phi \Downarrow \underline{m}_{\mathtt{i}} \qquad N; \sigma; \phi \Downarrow \underline{n}_{\mathtt{i}}}{M \mathbin{\underline{+}} N; \sigma; \phi \Downarrow \underline{m+n}_{\mathtt{i}}} \qquad \ldots$$

$$\frac{N; \sigma; \phi[f \mapsto {\color{red}?}] \Downarrow V}{\mathtt{def}\ f\ x = M\ \mathtt{in}\ N; \sigma; \phi \Downarrow V} \qquad \frac{N; \sigma; \phi \Downarrow V_a \qquad {\color{red}?}; {\color{red}?}[{\color{red}?} \mapsto V_a]; {\color{red}?} \Downarrow V}{f\ N; \sigma; \phi \Downarrow V}$$

Pour exécuter un appel à la fonction f, on doit :

- connaître au moins son corps et la variable qui lui sert d'argument,

Les fonctions de seconde classe (2/2)

La sémantique doit maintenant être paramétrée par deux environnements, un par espace de nom :

- l'environnement σ usuel, associant aux variables $\mathtt{x}, \mathtt{y}, \mathtt{z}$ des valeurs,
- un nouvel environnement ϕ, associant aux variables $\mathtt{f}, \mathtt{g}, \mathtt{h}$ des...?

Essayons d'écrire les règles de notre jugement $M; \sigma; \phi \Downarrow V$.

$$\frac{}{x; \sigma; \phi \Downarrow \sigma(x)} \qquad \ldots \qquad \frac{M; \sigma; \phi \Downarrow \underline{m}_{\mathtt{i}} \qquad N; \sigma; \phi \Downarrow \underline{n}_{\mathtt{i}}}{M \mathbin{\underline{+}} N; \sigma; \phi \Downarrow \underline{m + n}_{\mathtt{i}}} \qquad \ldots$$

$$\frac{N; \sigma; \phi[f \mapsto ?] \Downarrow V}{\mathtt{def}\ f\ x = M\ \mathtt{in}\ N; \sigma; \phi \Downarrow V} \qquad \frac{N; \sigma; \phi \Downarrow V_a \qquad ?; ?[? \mapsto V_a]; ? \Downarrow V}{f\ N; \sigma; \phi \Downarrow V}$$

Pour exécuter un appel à la fonction f, on doit :

- connaître au moins son corps et la variable qui lui sert d'argument,
- déterminer quels environnements utiliser (plusieurs choix possibles !).

Choix de portée (1/3)

On peut se restreindre aux fonctions dont le corps est clos.

$$\frac{N; \sigma; \phi[f \mapsto (x, M)] \Downarrow V}{\texttt{def } f\ x = M \texttt{ in } N; \sigma; \phi \Downarrow V} \qquad \frac{\phi(f) = (x, M) \quad N; \sigma; \phi \Downarrow V_a \quad M; \emptyset[x \mapsto V_a]; \emptyset \Downarrow V}{f\ N; \sigma; \phi \Downarrow V}$$

Choix de portée (1/3)

On peut se restreindre aux fonctions dont le corps est clos.

$$\frac{N;\sigma;\phi[f \mapsto (x,M)] \Downarrow V}{\texttt{def}\ f\ x = M\ \texttt{in}\ N;\sigma;\phi \Downarrow V} \qquad \frac{\phi(f) = (x,M) \qquad N;\sigma;\phi \Downarrow V_a \qquad M;\emptyset[x \mapsto V_a];\emptyset \Downarrow V}{f\ N;\sigma;\phi \Downarrow V}$$

[Exercice] Pour chaque M_i ci-dessous, déterminer s'il existe une valeur V_i telle qu'une dérivation $M_i;\emptyset;\emptyset \Downarrow V$ existe, et les donner le cas échéant.

$$M_1 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f x} = \texttt{x} \underline{+} \underline{1}_{\texttt{i}}\ \texttt{in f x} \qquad V_1 =$$
$$M_2 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y} \underline{+} \texttt{x in f x} \qquad V_2 =$$
$$M_3 \stackrel{\text{def}}{=} \texttt{def f x} = \texttt{x} \underline{+} \underline{1}_{\texttt{i}}\ \texttt{in def g x} = \texttt{f x in g}\ \underline{1}_{\texttt{i}} \qquad V_3 =$$
$$M_4 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y} \underline{+} \texttt{x in let x} = \underline{3}_{\texttt{i}}\ \texttt{in f x} \qquad V_4 =$$

Choix de portée (1/3)

On peut se restreindre aux fonctions dont le corps est clos.

$$\frac{N;\sigma;\phi[f \mapsto (x, M)] \Downarrow V}{\texttt{def}\ f\ x = M\ \texttt{in}\ N;\sigma;\phi \Downarrow V} \qquad \frac{\phi(f) = (x, M) \quad N;\sigma;\phi \Downarrow V_a \quad M;\emptyset[x \mapsto V_a];\emptyset \Downarrow V}{f\ N;\sigma;\phi \Downarrow V}$$

[Exercice] Pour chaque M_i ci-dessous, déterminer s'il existe une valeur V_i telle qu'une dérivation $M_i;\emptyset;\emptyset \Downarrow V$ existe, et les donner le cas échéant.

$$M_1 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f x} = \texttt{x} \underline{+} \underline{1}_{\texttt{i}}\ \texttt{in f x} \qquad V_1 = \underline{3}_{\texttt{i}}$$
$$M_2 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y} \underline{+} \texttt{x in f x} \qquad V_2 =$$
$$M_3 \stackrel{\text{def}}{=} \texttt{def f x} = \texttt{x} \underline{+} \underline{1}_{\texttt{i}}\ \texttt{in def g x} = \texttt{f x in g}\ \underline{1}_{\texttt{i}} \qquad V_3 =$$
$$M_4 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y} \underline{+} \texttt{x in let x} = \underline{3}_{\texttt{i}}\ \texttt{in f x} \qquad V_4 =$$

Choix de portée (1/3)

On peut se restreindre aux fonctions dont le corps est clos.

$$\frac{N;\sigma;\phi[f \mapsto (x, M)] \Downarrow V}{\texttt{def}\ f\ x = M\ \texttt{in}\ N;\sigma;\phi \Downarrow V} \qquad \frac{\phi(f) = (x, M) \quad N;\sigma;\phi \Downarrow V_a \quad M;\emptyset[x \mapsto V_a];\emptyset \Downarrow V}{f\ N;\sigma;\phi \Downarrow V}$$

[Exercice] Pour chaque M_i ci-dessous, déterminer s'il existe une valeur V_i telle qu'une dérivation $M_i;\emptyset;\emptyset \Downarrow V$ existe, et les donner le cas échéant.

$$M_1 \overset{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f x} = \texttt{x} \mathbin{\underline{+}} \underline{1}_{\texttt{i}}\ \texttt{in f x} \qquad V_1 = \underline{3}_{\texttt{i}}$$
$$M_2 \overset{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y} \mathbin{\underline{+}} \texttt{x in f x} \qquad V_2 = \times$$
$$M_3 \overset{\text{def}}{=} \texttt{def f x} = \texttt{x} \mathbin{\underline{+}} \underline{1}_{\texttt{i}}\ \texttt{in def g x} = \texttt{f x in g}\ \underline{1}_{\texttt{i}} \qquad V_3 =$$
$$M_4 \overset{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y} \mathbin{\underline{+}} \texttt{x in let x} = \underline{3}_{\texttt{i}}\ \texttt{in f x} \qquad V_4 =$$

Choix de portée (1/3)

On peut se restreindre aux fonctions dont le corps est clos.

$$\frac{N; \sigma; \phi[f \mapsto (x, M)] \Downarrow V}{\mathtt{def}\ f\ x = M\ \mathtt{in}\ N; \sigma; \phi \Downarrow V} \qquad \frac{\phi(f) = (x, M) \qquad N; \sigma; \phi \Downarrow V_a \qquad M; \emptyset[x \mapsto V_a]; \emptyset \Downarrow V}{f\ N; \sigma; \phi \Downarrow V}$$

[Exercice] Pour chaque M_i ci-dessous, déterminer s'il existe une valeur V_i telle qu'une dérivation $M_i; \emptyset; \emptyset \Downarrow V$ existe, et les donner le cas échéant.

$$\begin{array}{ll}
M_1 \stackrel{\text{def}}{=} \mathtt{let\ x} = \underline{2}_{\mathtt{i}}\ \mathtt{in\ def\ f\ x} = \mathtt{x} \mathbin{\underline{+}} \underline{1}_{\mathtt{i}}\ \mathtt{in\ f\ x} & V_1 = \underline{3}_{\mathtt{i}} \\
M_2 \stackrel{\text{def}}{=} \mathtt{let\ x} = \underline{2}_{\mathtt{i}}\ \mathtt{in\ def\ f\ y} = \mathtt{y} \mathbin{\underline{+}} \mathtt{x\ in\ f\ x} & V_2 = \times \\
M_3 \stackrel{\text{def}}{=} \mathtt{def\ f\ x} = \mathtt{x} \mathbin{\underline{+}} \underline{1}_{\mathtt{i}}\ \mathtt{in\ def\ g\ x} = \mathtt{f\ x\ in\ g}\ \underline{1}_{\mathtt{i}} & V_3 = \times \\
M_4 \stackrel{\text{def}}{=} \mathtt{let\ x} = \underline{2}_{\mathtt{i}}\ \mathtt{in\ def\ f\ y} = \mathtt{y} \mathbin{\underline{+}} \mathtt{x\ in\ let\ x} = \underline{3}_{\mathtt{i}}\ \mathtt{in\ f\ x} & V_4 =
\end{array}$$

Choix de portée (1/3)

On peut se restreindre aux fonctions dont le corps est clos.

$$\frac{N; \sigma; \phi[f \mapsto (x, M)] \Downarrow V}{\texttt{def}\ f\ x = M\ \texttt{in}\ N; \sigma; \phi \Downarrow V} \qquad \frac{\phi(f) = (x, M) \quad N; \sigma; \phi \Downarrow V_a \quad M; \emptyset[x \mapsto V_a]; \emptyset \Downarrow V}{f\ N; \sigma; \phi \Downarrow V}$$

[Exercice] Pour chaque M_i ci-dessous, déterminer s'il existe une valeur V_i telle qu'une dérivation $M_i; \emptyset; \emptyset \Downarrow V$ existe, et les donner le cas échéant.

$$\begin{array}{ll}
M_1 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f x} = \texttt{x} \underline{+} \underline{1}_{\texttt{i}}\ \texttt{in f x} & V_1 = \underline{3}_{\texttt{i}} \\
M_2 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y} \underline{+} \texttt{x in f x} & V_2 = \times \\
M_3 \stackrel{\text{def}}{=} \texttt{def f x} = \texttt{x} \underline{+} \underline{1}_{\texttt{i}}\ \texttt{in def g x} = \texttt{f x in g}\ \underline{1}_{\texttt{i}} & V_3 = \times \\
M_4 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y} \underline{+} \texttt{x in let x} = \underline{3}_{\texttt{i}}\ \texttt{in f x} & V_4 = \times
\end{array}$$

Choix de portée (2/3)

On peut choisir la portée dynamique.

$$\frac{N; \sigma; \phi[f \mapsto (x, M)] \Downarrow V}{\texttt{def } f\ x = M \texttt{ in } N; \sigma; \phi \Downarrow V} \qquad \frac{\phi(f) = (x, M) \quad N; \sigma; \phi \Downarrow V_a \quad M; \sigma[x \mapsto V_a]; \phi \Downarrow V}{f\ N; \sigma; \phi \Downarrow V}$$

Choix de portée (2/3)

On peut choisir la portée dynamique.

$$\frac{N;\sigma;\phi[f \mapsto (x,M)] \Downarrow V}{\texttt{def}\ f\ x = M\ \texttt{in}\ N;\sigma;\phi \Downarrow V} \qquad \frac{\phi(f) = (x,M) \qquad N;\sigma;\phi \Downarrow V_a \qquad M;\sigma[x \mapsto V_a];\phi \Downarrow V}{f\ N;\sigma;\phi \Downarrow V}$$

[Exercice] Pour chaque M_i ci-dessous, déterminer s'il existe une valeur V_i telle qu'une dérivation $M_i;\emptyset;\emptyset \Downarrow V$ existe, et les donner le cas échéant.

$$\begin{array}{ll}
M_1 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_\texttt{i}\ \texttt{in def f x} = \texttt{x} \underline{+} \underline{1}_\texttt{i}\ \texttt{in f x} & V_1 = \\
M_2 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_\texttt{i}\ \texttt{in def f y} = \texttt{y} \underline{+} \texttt{x in f x} & V_2 = \\
M_3 \stackrel{\text{def}}{=} \texttt{def f x} = \texttt{x} \underline{+} \underline{1}_\texttt{i}\ \texttt{in def g x} = \texttt{f x in g}\ \underline{1}_\texttt{i} & V_3 = \\
M_4 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_\texttt{i}\ \texttt{in def f y} = \texttt{y} \underline{+} \texttt{x in let x} = \underline{3}_\texttt{i}\ \texttt{in f x} & V_4 =
\end{array}$$

Choix de portée (2/3)

On peut choisir la portée dynamique.

$$\frac{N; \sigma; \phi[f \mapsto (x, M)] \Downarrow V}{\texttt{def}\ f\ x = M\ \texttt{in}\ N; \sigma; \phi \Downarrow V} \qquad \frac{\phi(f) = (x, M) \quad N; \sigma; \phi \Downarrow V_a \quad M; \sigma[x \mapsto V_a]; \phi \Downarrow V}{f\ N; \sigma; \phi \Downarrow V}$$

[Exercice] Pour chaque M_i ci-dessous, déterminer s'il existe une valeur V_i telle qu'une dérivation $M_i; \emptyset; \emptyset \Downarrow V$ existe, et les donner le cas échéant.

$$\begin{array}{ll}
M_1 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_\texttt{i}\ \texttt{in def f x} = \texttt{x} \,\underline{+}\, \underline{1}_\texttt{i}\ \texttt{in f x} & V_1 = \underline{3}_\texttt{i} \\
M_2 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_\texttt{i}\ \texttt{in def f y} = \texttt{y} \,\underline{+}\, \texttt{x in f x} & V_2 = \\
M_3 \stackrel{\text{def}}{=} \texttt{def f x} = \texttt{x} \,\underline{+}\, \underline{1}_\texttt{i}\ \texttt{in def g x} = \texttt{f x in g}\ \underline{1}_\texttt{i} & V_3 = \\
M_4 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_\texttt{i}\ \texttt{in def f y} = \texttt{y} \,\underline{+}\, \texttt{x in let x} = \underline{3}_\texttt{i}\ \texttt{in f x} & V_4 =
\end{array}$$

Choix de portée (2/3)

On peut choisir la portée dynamique.

$$\frac{N;\sigma;\phi[f \mapsto (x,M)] \Downarrow V}{\texttt{def}\ f\ x = M\ \texttt{in}\ N;\sigma;\phi \Downarrow V} \qquad \frac{\phi(f) = (x,M) \quad N;\sigma;\phi \Downarrow V_a \quad M;\sigma[x \mapsto V_a];\phi \Downarrow V}{f\ N;\sigma;\phi \Downarrow V}$$

[Exercice] Pour chaque M_i ci-dessous, déterminer s'il existe une valeur V_i telle qu'une dérivation $M_i;\emptyset;\emptyset \Downarrow V$ existe, et les donner le cas échéant.

$$\begin{array}{ll}
M_1 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f x} = \texttt{x} \mathbin{\underline{+}} \underline{1}_{\texttt{i}}\ \texttt{in f x} & V_1 = \underline{3}_{\texttt{i}} \\
M_2 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y} \mathbin{\underline{+}} \texttt{x in f x} & V_2 = \underline{4}_{\texttt{i}} \\
M_3 \stackrel{\text{def}}{=} \texttt{def f x} = \texttt{x} \mathbin{\underline{+}} \underline{1}_{\texttt{i}}\ \texttt{in def g x} = \texttt{f x in g}\ \underline{1}_{\texttt{i}} & V_3 = \\
M_4 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y} \mathbin{\underline{+}} \texttt{x in let x} = \underline{3}_{\texttt{i}}\ \texttt{in f x} & V_4 =
\end{array}$$

Choix de portée (2/3)

On peut choisir la portée dynamique.

$$\frac{N;\sigma;\phi[f \mapsto (x, M)] \Downarrow V}{\texttt{def}\ f\ x = M\ \texttt{in}\ N;\sigma;\phi \Downarrow V} \qquad \frac{\phi(f) = (x, M) \qquad N;\sigma;\phi \Downarrow V_a \qquad M;\sigma[x \mapsto V_a];\phi \Downarrow V}{f\ N;\sigma;\phi \Downarrow V}$$

[Exercice] Pour chaque M_i ci-dessous, déterminer s'il existe une valeur V_i telle qu'une dérivation $M_i;\emptyset;\emptyset \Downarrow V$ existe, et les donner le cas échéant.

$$M_1 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f x} = \texttt{x} \mathbin{\underline{+}} \underline{1}_{\texttt{i}}\ \texttt{in f x} \qquad V_1 = \underline{3}_{\texttt{i}}$$
$$M_2 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y} \mathbin{\underline{+}} \texttt{x in f x} \qquad V_2 = \underline{4}_{\texttt{i}}$$
$$M_3 \stackrel{\text{def}}{=} \texttt{def f x} = \texttt{x} \mathbin{\underline{+}} \underline{1}_{\texttt{i}}\ \texttt{in def g x} = \texttt{f x in g}\ \underline{1}_{\texttt{i}} \qquad V_3 = \underline{2}_{\texttt{i}}$$
$$M_4 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y} \mathbin{\underline{+}} \texttt{x in let x} = \underline{3}_{\texttt{i}}\ \texttt{in f x} \qquad V_4 =$$

Choix de portée (2/3)

On peut choisir la portée dynamique.

$$\frac{N; \sigma; \phi[f \mapsto (x, M)] \Downarrow V}{\texttt{def } f\ x = M \texttt{ in } N; \sigma; \phi \Downarrow V} \qquad \frac{\phi(f) = (x, M) \qquad N; \sigma; \phi \Downarrow V_a \qquad M; \sigma[x \mapsto V_a]; \phi \Downarrow V}{f\ N; \sigma; \phi \Downarrow V}$$

[Exercice] Pour chaque M_i ci-dessous, déterminer s'il existe une valeur V_i telle qu'une dérivation $M_i; \emptyset; \emptyset \Downarrow V$ existe, et les donner le cas échéant.

$$\begin{array}{ll}
M_1 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_\texttt{i} \texttt{ in def f x} = \texttt{x} \,\underline{+}\, \underline{1}_\texttt{i} \texttt{ in f x} & V_1 = \underline{3}_\texttt{i} \\
M_2 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_\texttt{i} \texttt{ in def f y} = \texttt{y} \,\underline{+}\, \texttt{x in f x} & V_2 = \underline{4}_\texttt{i} \\
M_3 \stackrel{\text{def}}{=} \texttt{def f x} = \texttt{x} \,\underline{+}\, \underline{1}_\texttt{i} \texttt{ in def g x} = \texttt{f x in g } \underline{1}_\texttt{i} & V_3 = \underline{2}_\texttt{i} \\
M_4 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_\texttt{i} \texttt{ in def f y} = \texttt{y} \,\underline{+}\, \texttt{x in let x} = \underline{3}_\texttt{i} \texttt{ in f x} & V_4 = \underline{6}_\texttt{i}
\end{array}$$

Choix de portée (3/3)

On peut choisir la portée lexicale grâce à l'usage de fermetures.

$$\frac{N; \sigma; \phi[f \mapsto (x, M, \sigma, \phi)] \Downarrow V}{\texttt{def}\ f\ x = M\ \texttt{in}\ N; \sigma; \phi \Downarrow V} \qquad \frac{\phi(f) = (x, M, \sigma_f, \phi_f) \qquad N; \sigma; \phi \Downarrow V_a \qquad M; \sigma_f[x \mapsto V_a]; \phi_f \Downarrow V}{f\ N; \sigma; \phi \Downarrow V}$$

On peut choisir la portée lexicale grâce à l'usage de fermetures.

$$\frac{N;\sigma;\phi[f \mapsto (x, M, \sigma, \phi)] \Downarrow V}{\mathtt{def}\ f\ x = M\ \mathtt{in}\ N;\sigma;\phi \Downarrow V} \qquad \frac{\phi(f) = (x, M, \sigma_f, \phi_f) \quad N;\sigma;\phi \Downarrow V_a \qquad M;\sigma_f[x \mapsto V_a];\phi_f \Downarrow V}{f\ N;\sigma;\phi \Downarrow V}$$

[Exercice] Pour chaque M_i ci-dessous, déterminer s'il existe une valeur V_i telle qu'une dérivation $M_i;\emptyset;\emptyset \Downarrow V$ existe, et les donner le cas échéant.

$$\begin{array}{ll}
M_1 \stackrel{\text{def}}{=} \mathtt{let}\ \mathtt{x} = \underline{2}_{\mathtt{i}}\ \mathtt{in}\ \mathtt{def}\ \mathtt{f}\ \mathtt{x} = \mathtt{x} \mathbin{\underline{+}} \underline{1}_{\mathtt{i}}\ \mathtt{in}\ \mathtt{f}\ \mathtt{x} & V_1 = \\
M_2 \stackrel{\text{def}}{=} \mathtt{let}\ \mathtt{x} = \underline{2}_{\mathtt{i}}\ \mathtt{in}\ \mathtt{def}\ \mathtt{f}\ \mathtt{y} = \mathtt{y} \mathbin{\underline{+}} \mathtt{x}\ \mathtt{in}\ \mathtt{f}\ \mathtt{x} & V_2 = \\
M_3 \stackrel{\text{def}}{=} \mathtt{def}\ \mathtt{f}\ \mathtt{x} = \mathtt{x} \mathbin{\underline{+}} \underline{1}_{\mathtt{i}}\ \mathtt{in}\ \mathtt{def}\ \mathtt{g}\ \mathtt{x} = \mathtt{f}\ \mathtt{x}\ \mathtt{in}\ \mathtt{g}\ \underline{1}_{\mathtt{i}} & V_3 = \\
M_4 \stackrel{\text{def}}{=} \mathtt{let}\ \mathtt{x} = \underline{2}_{\mathtt{i}}\ \mathtt{in}\ \mathtt{def}\ \mathtt{f}\ \mathtt{y} = \mathtt{y} \mathbin{\underline{+}} \mathtt{x}\ \mathtt{in}\ \mathtt{let}\ \mathtt{x} = \underline{3}_{\mathtt{i}}\ \mathtt{in}\ \mathtt{f}\ \mathtt{x} & V_4 =
\end{array}$$

Choix de portée (3/3)

On peut choisir la portée lexicale grâce à l'usage de fermetures.

$$\frac{N;\sigma;\phi[f \mapsto (x, M, \sigma, \phi)] \Downarrow V}{\texttt{def}\ f\ x = M\ \texttt{in}\ N;\sigma;\phi \Downarrow V} \qquad \frac{\phi(f) = (x, M, \sigma_f, \phi_f) \quad N;\sigma;\phi \Downarrow V_a \quad M;\sigma_f[x \mapsto V_a];\phi_f \Downarrow V}{f\ N;\sigma;\phi \Downarrow V}$$

[Exercice] Pour chaque M_i ci-dessous, déterminer s'il existe une valeur V_i telle qu'une dérivation $M_i;\emptyset;\emptyset \Downarrow V$ existe, et les donner le cas échéant.

$$\begin{array}{ll}
M_1 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f x} = \texttt{x} \,\underline{+}\, \underline{1}_{\texttt{i}}\ \texttt{in f x} & V_1 = \underline{3}_{\texttt{i}} \\
M_2 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y} \,\underline{+}\, \texttt{x in f x} & V_2 = \\
M_3 \stackrel{\text{def}}{=} \texttt{def f x} = \texttt{x} \,\underline{+}\, \underline{1}_{\texttt{i}}\ \texttt{in def g x} = \texttt{f x in g}\ \underline{1}_{\texttt{i}} & V_3 = \\
M_4 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y} \,\underline{+}\, \texttt{x in let x} = \underline{3}_{\texttt{i}}\ \texttt{in f x} & V_4 =
\end{array}$$

Choix de portée (3/3)

On peut choisir la portée lexicale grâce à l'usage de fermetures.

$$\frac{N;\sigma;\phi[f \mapsto (x, M, \sigma, \phi)] \Downarrow V}{\texttt{def}\ f\ x = M\ \texttt{in}\ N;\sigma;\phi \Downarrow V} \qquad \frac{\phi(f) = (x, M, \sigma_f, \phi_f) \quad N;\sigma;\phi \Downarrow V_a \qquad M;\sigma_f[x \mapsto V_a];\phi_f \Downarrow V}{f\ N;\sigma;\phi \Downarrow V}$$

[Exercice] Pour chaque M_i ci-dessous, déterminer s'il existe une valeur V_i telle qu'une dérivation $M_i;\emptyset;\emptyset \Downarrow V$ existe, et les donner le cas échéant.

$$M_1 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f x} = \texttt{x} \underline{+} \underline{1}_{\texttt{i}}\ \texttt{in f x} \qquad V_1 = \underline{3}_{\texttt{i}}$$

$$M_2 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y} \underline{+} \texttt{x in f x} \qquad V_2 = \underline{4}_{\texttt{i}}$$

$$M_3 \stackrel{\text{def}}{=} \texttt{def f x} = \texttt{x} \underline{+} \underline{1}_{\texttt{i}}\ \texttt{in def g x} = \texttt{f x in g}\ \underline{1}_{\texttt{i}} \qquad V_3 =$$

$$M_4 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y} \underline{+} \texttt{x in let x} = \underline{3}_{\texttt{i}}\ \texttt{in f x} \qquad V_4 =$$

Choix de portée (3/3)

On peut choisir la portée lexicale grâce à l'usage de fermetures.

$$\frac{N;\sigma;\phi[f \mapsto (x, M, \sigma, \phi)] \Downarrow V}{\mathtt{def}\ f\ x = M\ \mathtt{in}\ N;\sigma;\phi \Downarrow V} \quad \frac{\phi(f) = (x, M, \sigma_f, \phi_f) \quad N;\sigma;\phi \Downarrow V_a \quad M;\sigma_f[x \mapsto V_a];\phi_f \Downarrow V}{f\ N;\sigma;\phi \Downarrow V}$$

[Exercice] Pour chaque M_i ci-dessous, déterminer s'il existe une valeur V_i telle qu'une dérivation $M_i;\emptyset;\emptyset \Downarrow V$ existe, et les donner le cas échéant.

$$\begin{array}{ll}
M_1 \stackrel{\text{def}}{=} \mathtt{let\ x} = \underline{2}_{\mathtt{i}}\ \mathtt{in\ def\ f\ x} = \mathtt{x} \mathbin{\underline{+}} \underline{1}_{\mathtt{i}}\ \mathtt{in\ f\ x} & V_1 = \underline{3}_{\mathtt{i}} \\
M_2 \stackrel{\text{def}}{=} \mathtt{let\ x} = \underline{2}_{\mathtt{i}}\ \mathtt{in\ def\ f\ y} = \mathtt{y} \mathbin{\underline{+}} \mathtt{x\ in\ f\ x} & V_2 = \underline{4}_{\mathtt{i}} \\
M_3 \stackrel{\text{def}}{=} \mathtt{def\ f\ x} = \mathtt{x} \mathbin{\underline{+}} \underline{1}_{\mathtt{i}}\ \mathtt{in\ def\ g\ x} = \mathtt{f\ x\ in\ g}\ \underline{1}_{\mathtt{i}} & V_3 = \underline{2}_{\mathtt{i}} \\
M_4 \stackrel{\text{def}}{=} \mathtt{let\ x} = \underline{2}_{\mathtt{i}}\ \mathtt{in\ def\ f\ y} = \mathtt{y} \mathbin{\underline{+}} \mathtt{x\ in\ let\ x} = \underline{3}_{\mathtt{i}}\ \mathtt{in\ f\ x} & V_4 =
\end{array}$$

Choix de portée (3/3)

On peut choisir la portée lexicale grâce à l'usage de fermetures.

$$\frac{N;\sigma;\phi[f \mapsto (x, M, \sigma, \phi)] \Downarrow V}{\texttt{def}\ f\ x = M\ \texttt{in}\ N;\sigma;\phi \Downarrow V} \quad \frac{\phi(f) = (x, M, \sigma_f, \phi_f) \qquad N;\sigma;\phi \Downarrow V_a \qquad M;\sigma_f[x \mapsto V_a];\phi_f \Downarrow V}{f\ N;\sigma;\phi \Downarrow V}$$

[Exercice] Pour chaque M_i ci-dessous, déterminer s'il existe une valeur V_i telle qu'une dérivation $M_i;\emptyset;\emptyset \Downarrow V$ existe, et les donner le cas échéant.

$$M_1 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f x} = \texttt{x}\,\underline{+}\,\underline{1}_{\texttt{i}}\ \texttt{in f x} \qquad V_1 = \underline{3}_{\texttt{i}}$$
$$M_2 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y}\,\underline{+}\,\texttt{x in f x} \qquad V_2 = \underline{4}_{\texttt{i}}$$
$$M_3 \stackrel{\text{def}}{=} \texttt{def f x} = \texttt{x}\,\underline{+}\,\underline{1}_{\texttt{i}}\ \texttt{in def g x} = \texttt{f x in g}\ \underline{1}_{\texttt{i}} \qquad V_3 = \underline{2}_{\texttt{i}}$$
$$M_4 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y}\,\underline{+}\,\texttt{x in let x} = \underline{3}_{\texttt{i}}\ \texttt{in f x} \qquad V_4 = \underline{5}_{\texttt{i}}$$

Choix de portée (3/3)

On peut choisir la portée lexicale grâce à l'usage de fermetures.

$$\frac{N; \sigma; \phi[f \mapsto (x, M, \sigma, \phi)] \Downarrow V}{\texttt{def}\ f\ x = M\ \texttt{in}\ N; \sigma; \phi \Downarrow V} \qquad \frac{\phi(f) = (x, M, \sigma_f, \phi_f) \quad N; \sigma; \phi \Downarrow V_a \qquad M; \sigma_f[x \mapsto V_a]; \phi_f \Downarrow V}{f\ N; \sigma; \phi \Downarrow V}$$

[Exercice] Pour chaque M_i ci-dessous, déterminer s'il existe une valeur V_i telle qu'une dérivation $M_i; \emptyset; \emptyset \Downarrow V$ existe, et les donner le cas échéant.

$$
\begin{array}{ll}
M_1 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f x} = \texttt{x} \mathbin{\underline{+}} \underline{1}_{\texttt{i}}\ \texttt{in f x} & V_1 = \underline{3}_{\texttt{i}} \\
M_2 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y} \mathbin{\underline{+}} \texttt{x in f x} & V_2 = \underline{4}_{\texttt{i}} \\
M_3 \stackrel{\text{def}}{=} \texttt{def f x} = \texttt{x} \mathbin{\underline{+}} \underline{1}_{\texttt{i}}\ \texttt{in def g x} = \texttt{f x in g}\ \underline{1}_{\texttt{i}} & V_3 = \underline{2}_{\texttt{i}} \\
M_4 \stackrel{\text{def}}{=} \texttt{let x} = \underline{2}_{\texttt{i}}\ \texttt{in def f y} = \texttt{y} \mathbin{\underline{+}} \texttt{x in let x} = \underline{3}_{\texttt{i}}\ \texttt{in f x} & V_4 = \underline{5}_{\texttt{i}}
\end{array}
$$

Pourquoi choisir la portée lexicale ?

- Plus simple (permet le raisonnement local), plus utile.
- Plus naturelle du point de vue de l'équivalence observationnelle.

Les fonctions de première classe

Autoriser les fonctions à passer/renvoyer des fonctions à leurs appelants nous rapproche des langages réels (p. ex. OCaml mais aussi Java, C. . .).

Les fonctions de première classe

Autoriser les fonctions à passer/renvoyer des fonctions à leurs appelants nous rapproche des langages réels (p. ex. OCaml mais aussi Java, C...).

Il suffit de remplacer la construction `def` par une construction créant une fonction anonyme, et d'ajouter les fermetures aux valeurs.

$$M, N, P ::= \cdots \mid \mathtt{fun}\, x.\, M \mid M\ N$$
$$V, W ::= \cdots \mid (x, M, \sigma)_{\mathtt{c}}$$

Les fonctions de première classe

Autoriser les fonctions à passer/renvoyer des fonctions à leurs appelants nous rapproche des langages réels (p. ex. OCaml mais aussi Java, C...).

Il suffit de remplacer la construction `def` par une construction créant une fonction anonyme, et d'ajouter les fermetures aux valeurs.

$$M, N, P ::= \cdots \mid \texttt{fun}\, x.\, M \mid M\ N$$
$$V, W ::= \cdots \mid (x, M, \sigma)_{\mathsf{c}}$$

La sémantique ressemble à celle des fonctions de seconde classe respectant la portée lexicale, mais sans environnement supplémentaire.

$$\frac{}{\texttt{fun}\, x.\, M; \sigma \Downarrow (x, M, \sigma)_{\mathsf{c}}} \qquad \frac{M; \sigma \Downarrow (x, M_f, \sigma_f)_{\mathsf{c}} \quad N; \sigma \Downarrow V_a \quad M_f; \sigma_f[x \mapsto V_a] \Downarrow V}{M\ N; \sigma \Downarrow V}$$

Bilan d'étape

La syntaxe de Marthe^{++} est la suivante.

$$
\begin{array}{rcl}
M, N, P & ::= & \underline{n}_{\mathtt{i}} \mid \underline{b}_{\mathtt{b}} \mid M\ \mathit{bop}\ N \mid \mathit{uop}\ M \mid \mathtt{let}\ x = M\ \mathtt{in}\ N \mid (M, N) \mid \\
& \mid & \mathtt{fst}\ M \mid \mathtt{snd}\ M \mid \mathtt{if}\ M\ \mathtt{then}\ N\ \mathtt{else}\ P \mid \mathtt{fun}\ x.\, M \mid M\ N \\
V, W & ::= & \underline{n}_{\mathtt{i}} \mid \underline{b}_{\mathtt{b}} \mid (V, W) \mid (x, M, \sigma)_{\mathtt{c}} \\
\mathit{bop} & ::= & \underline{+} \mid \underline{*} \mid \underline{\wedge} \mid \underline{\vee} \mid \underline{\leq} \\
\mathit{uop} & ::= & \underline{\neg}
\end{array}
$$

On pourra rajouter des opérateurs binaires ou unaires au besoin.

Bilan d'étape

La syntaxe de Marthe^{++} est la suivante.

$$
\begin{array}{rcl}
M, N, P & ::= & \underline{n}_{\mathtt{i}} \mid \underline{b}_{\mathtt{b}} \mid M \, bop \, N \mid uop \, M \mid \mathtt{let}\ x = M\ \mathtt{in}\ N \mid (M, N) \mid \\
& \mid & \mathtt{fst}\ M \mid \mathtt{snd}\ M \mid \mathtt{if}\ M\ \mathtt{then}\ N\ \mathtt{else}\ P \mid \mathtt{fun}\ x.\, M \mid M\ N \\
V, W & ::= & \underline{n}_{\mathtt{i}} \mid \underline{b}_{\mathtt{b}} \mid (V, W) \mid (x, M, \sigma)_{\mathtt{c}} \\
bop & ::= & \underline{+} \mid \underline{*} \mid \underline{\wedge} \mid \underline{\vee} \mid \underline{\leq} \\
uop & ::= & \underline{\neg}
\end{array}
$$

On pourra rajouter des opérateurs binaires ou unaires au besoin.
[Exercice] Implémenter l'évaluateur correspondant à la sémantique.

Équivalence observationnelle (1/2)

Adaptons naïvement l'équivalence observationnelle définie pour Marthe.

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in \mathit{Env}, \forall V \in \mathit{Val},\ M; \sigma \Downarrow V \Leftrightarrow N; \sigma \Downarrow V$$

Est-ce un choix raisonnable ?

Équivalence observationnelle (1/2)

Adaptons naïvement l'équivalence observationnelle définie pour Marthe.

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in Env, \forall V \in Val,\ M; \sigma \Downarrow V \Leftrightarrow N; \sigma \Downarrow V$$

Est-ce un choix raisonnable ?
[Exercice] Chercher deux termes M, N qui devraient intuitivement être équivalents mais tels que $M \not\equiv N$ pour la définition ci-dessus.

Équivalence observationnelle (1/2)

Adaptons naïvement l'équivalence observationnelle définie pour Marthe.

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in Env, \forall V \in Val,\ M; \sigma \Downarrow V \Leftrightarrow N; \sigma \Downarrow V$$

Est-ce un choix raisonnable ?
[Exercice] Chercher deux termes M, N qui devraient intuitivement être équivalents mais tels que $M \not\equiv N$ pour la définition ci-dessus.

On peut par exemple choisir $M = \texttt{fun x.} (\underline{1}_{\texttt{i}} \mathbin{\underline{+}} \underline{1}_{\texttt{i}})$ et $N = \texttt{fun x.} \underline{2}_{\texttt{i}}$.

Pourquoi M et N devraient-ils être équivalents ?

Dans tout terme P s'évaluant vers un booléen ou un entier et où M apparaît comme sous-terme, remplacer M par N n'affecte pas le résultat.

Équivalence observationnelle (1/2)

Adaptons naïvement l'équivalence observationnelle définie pour Marthe.

$$M \equiv N \stackrel{\text{def}}{=} \forall \sigma \in \mathit{Env}, \forall V \in \mathit{Val},\ M; \sigma \Downarrow V \Leftrightarrow N; \sigma \Downarrow V$$

Est-ce un choix raisonnable ?
[Exercice] Chercher deux termes M, N qui devraient intuitivement être équivalents mais tels que $M \not\equiv N$ pour la définition ci-dessus.

On peut par exemple choisir $M = \mathtt{fun}\,\mathtt{x}.\,(\underline{1}_{\mathtt{i}} \mathbin{\underline{+}} \underline{1}_{\mathtt{i}})$ et $N = \mathtt{fun}\,\mathtt{x}.\,\underline{2}_{\mathtt{i}}$.

Pourquoi M et N devraient-ils être équivalents ?

Dans tout terme P s'évaluant vers un booléen ou un entier et où M apparaît comme sous-terme, remplacer M par N n'affecte pas le résultat.

On veut élargir la définition de l'équivalence via une notion de *contexte*.

Équivalence observationnelle (2/2)

Un contexte est un terme K contenant *exactement une occurrence* du symbole spécial $\square$, qu'on appelle souvent le "trou" de K.

$$\begin{aligned} K ::= {} & \square \mid K\ \mathit{bop}\ M \mid M\ \mathit{bop}\ K \mid \mathit{uop}\ K \mid \texttt{let}\ x = K\ \texttt{in}\ M \\ \mid {} & \texttt{let}\ x = M\ \texttt{in}\ K \mid \texttt{fun}\ x.\, K \mid K\ M \mid M\ K \end{aligned}$$

Équivalence observationnelle (2/2)

Un contexte est un terme K contenant *exactement une occurrence* du symbole spécial $\square$, qu'on appelle souvent le “trou” de K.

$$\begin{aligned} K \;::=\; & \square \mid K \; bop \; M \mid M \; bop \; K \mid uop \; K \mid \texttt{let}\; x = K \;\texttt{in}\; M \\ \mid \;& \texttt{let}\; x = M \;\texttt{in}\; K \mid \texttt{fun}\; x. \, K \mid K \; M \mid M \; K \end{aligned}$$

On peut “boucher” le trou d'un contexte K avec un terme M pour obtenir un terme $K[M]$. Cette opération est définie récursivement sur K.

$$\begin{aligned} \square[M] &= M \\ (K \; bop \; N)[M] &= K[M] \; bop \; N \\ (N \; bop \; K)[M] &= N \; bop \; K[M] \\ & \ldots \end{aligned}$$

Équivalence observationnelle (2/2)

Un contexte est un terme K contenant *exactement une occurrence* du symbole spécial $\square$, qu'on appelle souvent le "trou" de K.

$$
\begin{aligned}
K ::= {} & \square \mid K \; bop \; M \mid M \; bop \; K \mid uop \; K \mid \texttt{let } x = K \texttt{ in } M \\
\mid {} & \texttt{let } x = M \texttt{ in } K \mid \texttt{fun } x. \, K \mid K \; M \mid M \; K
\end{aligned}
$$

On peut "boucher" le trou d'un contexte K avec un terme M pour obtenir un terme $K[M]$. Cette opération est définie récursivement sur K.

$$
\begin{aligned}
\square[M] &= M \\
(K \; bop \; N)[M] &= K[M] \; bop \; N \\
(N \; bop \; K)[M] &= N \; bop \; K[M] \\
&\ldots
\end{aligned}
$$

On utilise ces définitions pour raffiner l'équivalence observationnelle :

$$M \equiv N \stackrel{\text{def}}{=} \forall K \in Ctx, \forall b \in \mathbb{B}, K[M] \Downarrow \underline{b}_{\mathtt{b}} \Leftrightarrow K[N] \Downarrow \underline{b}_{\mathtt{b}}$$

où $M \Downarrow V$ est un raccourci pour $M; \emptyset \Downarrow V$.

Bilan de cette séance

La méthodologie vue dans cette séquence de cours s'étend :

- à un langage un peu plus réaliste, comme on l'a vu aujourd'hui,
  - Des types de données, des instructions de contrôle, des fonctions,
  - plus de comportements (*erreurs*...),
  - mais les mêmes concepts de base de la syntaxe.
- mais aussi, bien au delà, à des langages beaucoup plus riches !
  - L'équivalence observationnelle sera toujours *contextuelle*,
  - on implémentera très souvent la portée lexicale via les *fermetures*.

La prochaine séance traitera des systèmes de types.