Présentation finale projet 2
Où l'on parle de rongeurs

Valque Léo - Guillaume Coiffier

2017

ENS de Lyon

un interpréteur fouine $\bigcup$ une machine SECD

L'interpréteur fouine

L'interpréteur fouine

Fonctionnalités de l'interpréteur

- fouine pur

L'interpréteur fouine

Fonctionnalités de l'interpréteur

- fouine pur
- Exceptions

L'interpréteur fouine

Fonctionnalités de l'interpréteur

- fouine pur
- Exceptions
- Références sur des entiers

L'interpréteur fouine

Fonctionnalités de l'interpréteur

- fouine pur
- Exceptions
- Références sur des entiers
- Tableaux d'entiers

L'interpréteur fouine

Fonctionnalités de l'interpréteur

- fouine pur
- Exceptions
- Références sur des entiers
- Tableaux d'entiers

Environnement utilisé : Table de hachage

L'interpréteur fouine

Comment ça marche en pratique ?

L'interpréteur fouine

Comment ça marche en pratique ?

Les exceptions : On propage un booléen du `raise` au `try` juste au dessus

L'interpréteur fouine

Comment ça marche en pratique ?

Les exceptions : On propage un booléen du `raise` au `try` juste au dessus

Tableaux et références : Stockés dans l'environnement

Interprétation mixte

Interprétation mixte

Les fonctions et fonctions récursives sont pures.

Interprétation mixte

Les fonctions et fonctions récursives sont pures.

- Constructeur `Pure of programme`

Interprétation mixte

Les fonctions et fonctions récursives sont pures.

- Constructeur `Pure of programme`
- Fonction `label_pure_code` qui insère les constructeurs purs

Interprétation mixte

Les fonctions et fonctions récursives sont pures.

- Constructeur `Pure of programme`
- Fonction `label_pure_code` qui insère les constructeurs purs
- Interprétation quasiment normale

Interprétation mixte

Les fonctions et fonctions récursives sont pures.

- Constructeur `Pure of programme`
- Fonction `label_pure_code` qui insère les constructeurs purs
- Interprétation quasiment normale
- Recopie d'environnement de fouine vers la machine

La machine à pile SECD

Indices de Bruijn

- Dans la machine, on veut accèder aux élements par leurs position et non pas par leur nom

Indices de Bruijn

- Dans la machine, on veut accèder aux élements par leurs position et non pas par leur nom
- Lorsqu'on compile le code, on stocke le nom des variables défini dans une pile

Indices de Bruijn

- Dans la machine, on veut accèder aux élements par leurs position et non pas par leur nom
- Lorsqu'on compile le code, on stocke le nom des variables défini dans une pile
- Lorsqu'on compile l'access à un élement, on regarde quel est la position actuel de cette élement dans la pile. on remplace le nom par la valeur de cette position

Indices de Bruijn

- Dans la machine, on veut accèder aux élements par leurs position et non pas par leur nom
- Lorsqu'on compile le code, on stocke le nom des variables défini dans une pile
- Lorsqu'on compile l'access à un élement, on regarde quel est la position actuel de cette élement dans la pile. on remplace le nom par la valeur de cette position
- `let x = 2 in ( let y = 3 in x+y) + x`
  devient :
  `let 2 in (let 3 in access(1)+access(0) ) + access(0)`

Conversion d'environnement

- **Objectif :** Traduire l'environnement de l'interpreteur vers l'environnement de la machine.

Conversion d'environnement

- **Objectif :** Traduire l'environnement de l'interpreteur vers l'environnement de la machine.
- **Problème :** Ils sont très différents.

Conversion d'environnement

- **Objectif :** Traduire l'environnement de l'interpreteur vers l'environnement de la machine.
- **Problème :** Ils sont très différents.

Dans l'interpreteur, c'est une table de hachage avec pour clés des strings.

Conversion d'environnement

- **Objectif :** Traduire l'environnement de l'interpreteur vers l'environnement de la machine.
- **Problème :** Ils sont très différents.

Dans l'interpreteur, c'est une table de hachage avec pour clés des strings.

Dans la machine, c'est une simple liste avec pour "clé" des entiers (indices de bruijn).

Conversion d'environnement

$\Longrightarrow$ On transmet pas directement l'environnement.

Conversion d'environnement

$\Longrightarrow$ On transmet pas directement l'environnement.

- On transforme l'environnement de l'interpreteur en programme avec des let in qui met devant le code envoyer à la machine

Conversion d'environnement

$\Longrightarrow$ On transmet pas directement l'environnement.

- On transforme l'environnement de l'interpreteur en programme avec des let in qui met devant le code envoyer à la machine
- La machine reçoit donc de l'environnement par ce biais

Conversion d'environnement

$\Longrightarrow$ On transmet pas directement l'environnement.

- On transforme l'environnement de l'interpreteur en programme avec des let in qui met devant le code envoyer à la machine
- La machine reçoit donc de l'environnement par ce biais
- **Problème** : c'est lent, la traduction de l'environnement est en O(n) où n est la taille de l'environnement

Conclusion

Merci pour votre attention !