

Annexe

4.1 Guide d'utilisation

La syntaxe du langage LCI est très simple, elle se base sur la syntaxe du λ -calcul et les expressions arithmétiques et logiques (« \ » est utilisé au lieu du « λ » de l'abstraction). En plus de tout ce qu'on peut programmer avec les expressions du λ -calcul pure, on a ajouté les extensions qu'on va décrire dans ce qui suit.

Exemple

$\backslash x y.x+y$ représente une fonction qui prend en entrée deux entiers et retourne la leurs somme.

Remarque

Il faut choisir la méthode de réduction « lambda calcul » (LC) pour bénéficier de ces extensions.

4.1.1 Le typage explicite

On peut typer explicitement les variables non liées grâce à la syntaxe suivante :

nom_Variable : type

Exemple : $\backslash x. \text{abs:int} \rightarrow \text{int } x$

Le type est générer par la grammaire suivante :

type : nom_type | type -> type | (type)

4.1.2 Les types

Type	Définition	Opération	Exemple
int	1 2 3 ...	+ - * / < >	2+2 ; 13/6
Entier de Church	c1 c2 c3 ...	add mul sub dev succ pred	add c2 c4 ; mul c3 c2
real	1.5 2. 5.02 ...	+. -. *. /. sqrt < >	3. *. 2.5 ; 4.2 -. 1.6
bool	true false	&& not	not false ; true && true

4.1.3 Les déclarations

- Globale : *let nom_fonction = expression ;*
- Locale : *let nom_fonction = expression in expression ;*
- Récursive : *let rec nom_fonction = expression ;*
- Locale récursive : *let rec nom_fonction = expression in expression ;*

4.1.4 Les structures de contrôle (Sauts conditionnels)

Il existe deux types de structures selon la nature de la condition :

- Une condition logique hors du λ -calcul pure :

if condition then expression else expression

Exemple

let abs = \x. if x<0 then -x else x

let ouExclusif = \a b. if (a && b || not a && not b) then false else true

let rec factorielle = \n. if n==0 then 1 else n(factorielle (n-1))*

- Une condition logique appartenant au λ -calcul pure :

IF condition THEN expression ELSE expression

Exemple

let estNul = \x. IF iszero x THEN (\x y.x) ELSE (\x y.y)

let rec factorielle = \n. IF iszero n THEN c1 ELSE mul n (factorielle (pred n))