

LLLang reference

Jean CASPAR

3 février 2023

Table des matières

1	Grammar	2
2	Typing	2
2.1	Subtyping	2
2.2	Pattern typing	3
2.3	Irrefutable patterns	4
2.4	Pattern matching	4

1 Grammar

x is a identifier and n an integer.

$A, B, C, \dots ::= \text{int}$	$P, Q, \dots ::= _$	$e, f, g, \dots ::= x$
$ x$	$ x$	$ n$
$ x\langle A, B, \dots \rangle$	$ n$	$ ()$
$ ()$	$ ()$	$ e\langle A, B, \dots \rangle$
$ \text{!}$	$ P, Q, \dots$	$ \text{inj } A \ n \ e$
$ \text{!}A$	$ \text{inj } n \ P$	$ \text{unroll } e$
$ A * B * \dots$		$ \text{roll } A \ e$
$ A + B + \dots$		$ e \ f$
$ A \multimap B$		$ \text{let } P = e \text{ in } f$
$ \mu x. A$		$ - e$
$ \forall x. A$		$ e + f$
		$ e - f$
		$ e * f$
		$ e / f$
		$ e \% f$
		$ e = f$
		$ e < f$
		$ e, f, \dots$
		$ \text{match } e \{P \Rightarrow f, Q \Rightarrow g, \dots\}$
		$ \text{fun } \langle x, y, \dots \rangle (P : A, Q : B, \dots) \multimap C\{e\}$
		$ \text{rec fun } \langle x, y, \dots \rangle (P : A, Q : B, \dots) \multimap C\{e\}$

Currently, in the syntaxe $x\langle A, B, \dots \rangle$, x should be a named type, and not a type variable. Furthermore, x should be the name of a type of the form $\forall y_1. \dots \forall y_n. T$ in order for $x\langle A_1, \dots, A_n \rangle$ to be a type.

2 Typing

2.1 Subtyping

Subtyping is the relation $A < B$, ich means that A can be used wherever B is needed. It is the smallest preorder that satisfies the following relations :

$$\overline{\text{!} < A}$$

$$\begin{array}{c}
\overline{!A \prec A} \\
\\
\frac{A' \prec A \quad B \prec B'}{A \multimap B \prec A' \multimap B'} \\
\\
\frac{\forall 1 \leq i \leq n \quad A_i \prec B_i}{A_1 * \dots * A_n \prec B_1 * \dots * B_n} \\
\\
\frac{\forall 1 \leq i \leq n \quad A_i \prec B_i}{A_1 + \dots + A_n \prec B_1 + \dots + B_n} \\
\\
\frac{A \prec B}{\mu x. A \prec \mu x. B} \\
\\
\frac{A \prec B}{\forall x. A \prec \forall x. B}
\end{array}$$

2.2 Pattern typing

We say that a pattern P can match a type T and bind variables $x_1 : T_1, \dots, x_n : T_n$ if one can derive $x_1 : T_1, \dots, x_n : T_n \vdash P \prec T$ from the following relations :

$$\begin{array}{c}
\overline{\vdash _ \prec T} \\
\\
\overline{x : T \vdash x \prec T} \\
\\
\overline{\vdash n \prec \mathbf{int}} \\
\\
\overline{\vdash () \prec ()} \\
\\
\frac{x_1 : T_1, \dots, x_n : T_n \vdash P \prec T}{x_1 : !T_1, \dots, x_n : !T_n \vdash P \prec !T} \\
\\
\frac{\forall 1 \leq i \leq n \quad x_{i,1}, \dots, x_{i,n_i} \vdash P_i \prec T_i \quad \forall 1 \leq i < j \leq n \quad \{x_{i,k} \mid 1 \leq k \leq n_i\} \cap \{x_{j,k} \mid 1 \leq k \leq n_j\} = \emptyset}{x_{1,1} : T_{1,1}, \dots, x_{1,n_1} : T_{1,n_1}, \dots, x_{k,n_k} : T_{k,n_k} \vdash P_1, \dots, P_k \prec T_1 * \dots * T_k} \\
\\
\frac{x_1 : T_1, \dots, x_k : T_n \vdash P \prec A_i \quad 1 \leq i \leq n}{x_1, \dots, x_k \vdash \mathbf{inj} \ i \ P \prec A_1 + \dots + A_n}
\end{array}$$

2.3 Irrefutable patterns

A pattern is said to be irrefutable if it cannot fail to match. Such patterns are :

- Discarding
- Binding to a variable
- A tuple of irrefutable patterns (the empty tuple is such a tuple)
- An injection of a sum type of size 1 and an irrefutable pattern inside (this should probably not happen)

In let bindings or in function arguments, patterns that may appear should be irrefutable.

2.4 Pattern matching