Millet type inference

March 31, 2024

1 primitive values

$$\frac{}{\Gamma \vdash_v true : \mathbf{bool}}$$

$$\frac{}{\Gamma \vdash_v false : \mathbf{bool}}$$

$$\frac{(n \in \mathbb{N})}{\Gamma \vdash_v n : \mathbf{int}}$$

$$\frac{(fl \in \mathbb{R})}{\Gamma \vdash_v fl : \mathbf{float}}$$

$$\frac{(x : A) \in \Gamma}{\Gamma \vdash_v x : \mathbf{A}}$$

$$\frac{(utf8(ch))}{\Gamma \vdash_v \text{"}ch*\text{"} : \mathbf{string}}$$

where $utf8()$ means that a character that can be encoded with the **utf**-8 standard and $ch*$ stands for a possibly empty sequence of characters.

$$\frac{}{\Gamma \vdash_v [\,] : \mathbf{listA}}$$

$$\frac{\Gamma \vdash_v x : \mathbf{A} \quad \Gamma \vdash_v xs : \mathbf{listA}}{\Gamma \vdash_c x :: xs : \mathbf{listA}}$$

$$\frac{\Gamma \vdash_v a : \mathbf{A} \quad \Gamma \vdash_v b : \mathbf{B}}{\Gamma \vdash_v (a, b) : \mathbf{A} * \mathbf{B}}$$

this is the type constructor for pair types, which are a special case of the product type. It should be noted that Millet also supports n-tuples.

$$\frac{\Gamma \vdash_v M : \mathbf{A}}{\Gamma \vdash_v \mathbf{inl}\, M : \mathbf{A} + \mathbf{B}}$$

$$\frac{\Gamma \vdash_v M : \mathbf{B}}{\Gamma \vdash_v \mathbf{inr}\, M : \mathbf{A} + \mathbf{B}}$$

where **inr** and **inl** are aliases for the names defined with the **type** command. As with product types Millet supports variant types for an arbitrary number of variants.

2 primitive operations

2.1 int operations

$$\frac{\Gamma \vdash_v n_1 : \textbf{int} \quad \Gamma \vdash_v n_2 : \textbf{int}}{\Gamma \vdash_c n_1 + n_2 : \textbf{int}}$$

$$\frac{\Gamma \vdash_v n_1 : \textbf{int} \quad \Gamma \vdash_v n_2 : \textbf{int}}{\Gamma \vdash_c n_1 - n_2 : \textbf{int}}$$

$$\frac{\Gamma \vdash_v n_1 : \textbf{int} \quad \Gamma \vdash_v n_2 : \textbf{int}}{\Gamma \vdash_c n_1 * n_2 : \textbf{int}}$$

$$\frac{\Gamma \vdash_v n_1 : \textbf{int} \quad \Gamma \vdash_v n_2 : \textbf{int}}{\Gamma \vdash_c n_1 / n_2 : \textbf{int}}$$

$$\frac{\Gamma \vdash_v n_1 : \textbf{int} \quad \Gamma \vdash_v n_2 : \textbf{int}}{\Gamma \vdash_c n_1 \% n_2 : \textbf{int}}$$

$$\frac{\Gamma \vdash_v n : \textbf{int}}{\Gamma \vdash_c -n : \textbf{int}}$$

2.2 float operations

$$\frac{\Gamma \vdash_v n_1 : \textbf{float} \quad \Gamma \vdash_v n_2 : \textbf{float}}{\Gamma \vdash_c n_1 + .n_2 : \textbf{float}}$$

$$\frac{\Gamma \vdash_v n_1 : \textbf{float} \quad \Gamma \vdash_v n_2 : \textbf{float}}{\Gamma \vdash_c n_1 - .n_2 : \textbf{float}}$$

$$\frac{\Gamma \vdash_v n_1 : \textbf{float} \quad \Gamma \vdash_v n_2 : \textbf{float}}{\Gamma \vdash_c n_1 * .n_2 : \textbf{float}}$$

$$\frac{\Gamma \vdash_v n_1 : \textbf{float} \quad \Gamma \vdash_v n_2 : \textbf{float}}{\Gamma \vdash_c n_1 / .n_2 : \textbf{float}}$$

$$\frac{\Gamma \vdash_v n_1 : \textbf{float} \quad \Gamma \vdash_v n_2 : \textbf{float}}{\Gamma \vdash_c n_1 \% .n_2 : \textbf{float}}$$

$$\frac{\Gamma \vdash_v n_1 : \textbf{float} \quad \Gamma \vdash_v n_2 : \textbf{float}}{\Gamma \vdash_c n_1 * * n_2 : \textbf{float}}$$

$$\frac{\Gamma \vdash_v n : \textbf{float}}{\Gamma \vdash_c -.n : \textbf{float}}$$

2.3 bool operations

$$\frac{\Gamma \vdash_v n_1 : \mathbf{int} \quad \Gamma \vdash_v n_2 : \mathbf{int}}{\Gamma \vdash_c n_1 < n_2 : \mathbf{bool}}$$

$$\frac{\Gamma \vdash_v n_1 : \mathbf{int} \quad \Gamma \vdash_v n_2 : \mathbf{int}}{\Gamma \vdash_c n_1 > n_2 : \mathbf{bool}}$$

$$\frac{\Gamma \vdash_v n_1 : \mathbf{int} \quad \Gamma \vdash_v n_2 : \mathbf{int}}{\Gamma \vdash_c n_1 <= n_2 : \mathbf{bool}}$$

$$\frac{\Gamma \vdash_v n_1 : \mathbf{int} \quad \Gamma \vdash_v n_2 : \mathbf{int}}{\Gamma \vdash_c n_1 >= n_2 : \mathbf{bool}}$$

$$\frac{\Gamma \vdash_v n_1 : \mathbf{float} \quad \Gamma \vdash_v n_2 : \mathbf{float}}{\Gamma \vdash_c n_1 < n_2 : \mathbf{bool}}$$

$$\frac{\Gamma \vdash_v n_1 : \mathbf{float} \quad \Gamma \vdash_v n_2 : \mathbf{float}}{\Gamma \vdash_c n_1 > n_2 : \mathbf{bool}}$$

$$\frac{\Gamma \vdash_v n_1 : \mathbf{float} \quad \Gamma \vdash_v n_2 : \mathbf{float}}{\Gamma \vdash_c n_1 <= n_2 : \mathbf{bool}}$$

$$\frac{\Gamma \vdash_v n_1 : \mathbf{float} \quad \Gamma \vdash_v n_2 : \mathbf{float}}{\Gamma \vdash_c n_1 >= n_2 : \mathbf{bool}}$$

$$\frac{\Gamma \vdash_v n_1 : \mathbf{A} \quad \Gamma \vdash_v n_2 : \mathbf{A}}{\Gamma \vdash_c n_1 = n_2 : \mathbf{bool}}$$

$$\frac{\Gamma \vdash_v n_1 : \mathbf{A} \quad \Gamma \vdash_v n_2 : \mathbf{A}}{\Gamma \vdash_c n_1 <> n_2 : \mathbf{bool}}$$

3 Functions and computations

$$\frac{\Gamma, x : \mathbf{A} \vdash_c T : \mathbf{B}}{\Gamma \vdash_v \mathbf{fun}\, x \to T : \mathbf{A} \to \mathbf{B}}$$

Functions can be defined with an arbitrary number of arguments.

$$\frac{\Gamma, x : \mathbf{A} \vdash_c T : \mathbf{B}}{\Gamma \vdash_v \mathbf{let}\, fx = T : \mathbf{A} \to \mathbf{B}}$$

Alternatively, functions can be defined with the $\mathbf{let} f = \mathbf{function} \ldots$ syntax just like in Ocaml and is equivalent to writing $\mathbf{fun} fx = \mathbf{match}\, x\, \mathbf{with} \ldots$

$$\frac{\Gamma, f : A \to B, x : \mathbf{A} \vdash_c T : \mathbf{B}}{\Gamma \vdash_v \mathbf{let\, rec}\, fx = T : \mathbf{A} \to \mathbf{B}}$$

$$\frac{\Gamma \vdash_v V_1 : \mathbf{A} \to \mathbf{B} \quad \Gamma \vdash_v V_2 : \mathbf{A}}{\Gamma \vdash_c V_1 V_2 : \mathbf{B}}$$

$$\frac{\Gamma \vdash_v A * B}{\Gamma \vdash_c \mathbf{fst} V : \mathbf{A}}$$

$$\frac{\Gamma \vdash_v A * B}{\Gamma \vdash_c \mathbf{snd} V : \mathbf{B}}$$

$$\frac{\Gamma \vdash_v xs : \mathbf{list A} \quad \Gamma \vdash_v xss : \mathbf{list A}}{\Gamma \vdash_c xs @ xss : \mathbf{list A}}$$

$$\frac{\Gamma \vdash_c T_1 : \mathbf{A} \quad \Gamma, x : \mathbf{A} \vdash_c T_2 : \mathbf{B}}{\Gamma \vdash_c \mathbf{let} x = T_1 \mathbf{in} T_2 : \mathbf{B}}$$

$$\frac{\Gamma \vdash_v V : \mathbf{list A} \quad \Gamma \vdash_c T_1 : \mathbf{B} \quad \Gamma, x : \mathbf{A}, xs : \mathbf{list A} \vdash_c T_2 : \mathbf{B}}{\Gamma \vdash_c \mathbf{match} V \mathbf{with} [\,] \to T_1 | x :: xs \to T_2 : \mathbf{B}}$$

$$\frac{\Gamma \vdash_v V : \mathbf{A_1} + \mathbf{A_2} \quad \Gamma, x_1 : \mathbf{A_1} \vdash_c T_1 : \mathbf{B} \quad \Gamma, x_2 : \mathbf{A_2} \vdash_c T_2 : \mathbf{B}}{\Gamma \vdash_c \mathbf{match} V \mathbf{with\ inl} x_1 \to T_1 | \mathbf{inr} x_2 \to T_2 : \mathbf{B}}$$

This is a special case of the match statement for variant types where the given type only has two variants, however Millet supports variant types with an arbitrary number of variants.

$$\frac{\Gamma \vdash_v V : \mathbf{bool} \quad \Gamma \vdash_c T_1 : \mathbf{B} \quad \Gamma \vdash_c T_2 : \mathbf{B}}{\Gamma \vdash_c \mathbf{if} V \mathbf{then} T_1 \mathbf{else} T_2 : \mathbf{B}}$$

$$\frac{\Gamma \vdash_v V : \mathbf{A}}{\Gamma \vdash_c \mathbf{return} V : \mathbf{A}}$$