

# Type Inferencing in MPL

March 10, 2017

## 1 Type Inferencing

### 1.1 Introduction

This chapter deals with the type inferencing of MPL programs, namely functions, processes, terms and process commands when they are not type annotated and type checking when they are. Type inference process involves two main steps.

- Generating Type Equations
- Solving Type Equations

In this chapter, the type equations generation for the Sequential MPL (terms and functions) is dicussed first followed by that of the Concurrent MPL (process commands and processes).

Lets deconstruct the type Inferencing algorithm to gain a better understanding. It has two main parts.

- **Generation of Type Equations** - Type Equations captures the constraint relationship between the different constructs of a program.
- **Solving Type Equation** - The type equations are solved to find out the type of a program construct.

### 1.2 Generating Type Equations

Type Equations captures the constraint relationship between the different constructs of a program.

#### 1.2.1 Generating Type Equations for Sequential Terms

$\frac{\Gamma \vdash t_1 : T_1 \quad \langle E_1 \rangle \quad \dots \quad \Gamma \vdash t_n : T_n \quad \langle E_n \rangle}{\Gamma \vdash \text{cons} ([t_1, \dots, t_n]) : T} \text{cons}$		$\left\langle \exists \begin{array}{l} T_1, \dots, T_n, \\ A_1, \dots, A_n. \end{array} \begin{array}{l} T = O_{new} \\ T_1 = I_{1,new}, \dots, T_n = I_{n,new} \\ E_1, E_2, \dots, E_n \end{array} \right\rangle$
$\frac{\Gamma \vdash t_1 : T_1 \quad \langle E_1 \rangle \quad \dots \quad \Gamma \vdash t_n : T_n \quad \langle E_n \rangle}{\Gamma \vdash \text{dest} ([t_1, \dots, t_n]) : T} \text{dest}$		$\left\langle \exists \begin{array}{l} T_1, \dots, T_n, \\ A'_1, \dots, A'_n. \end{array} \begin{array}{l} T = O_{new} \\ T_1 = I_{1,new}, \dots, T_n = I_{n,new} \\ E_1, E_2, \dots, E_n \end{array} \right\rangle$

Table 1: Rules for type inference

$t : T_0 \quad \langle E_0 \rangle, \quad X_{11} : T_{11}, \dots, X_{1a} : T_{1a} \vdash t_1 : T_1 \quad \langle E_1 \rangle, \quad \dots, \quad X_{m1} : T_{m1}, \dots, X_{mn} : T_{mn} \vdash t_m : T_m \quad \langle E_m \rangle$		
$\Gamma \vdash \begin{array}{l} \text{fold } t \\ \text{of} \end{array} \left  \begin{array}{l} C_1 : X_{11}, \dots, X_{1a} \rightarrow t_1 \\ \vdots \\ C_m : X_{m1}, \dots, X_{mn} \rightarrow t_n \end{array} \right. : T$	$\left\langle \exists \begin{array}{l} T_0, T_1, \dots, T_m, \\ T_{11}, \dots, T_{1a}, \\ \vdots \\ T_{m1}, \dots, T_{mn}, \\ A'_1, \dots, A'_k \end{array} \begin{array}{l} T_0 = D_{new} \\ T_1 = T, \dots, T_m = T \\ T_{11} = F_{11,new}, \dots, T_{1a} = F_{1a,new} \\ \vdots \\ T_{m1} = F_{m1,new}, \dots, T_{mn} = F_{mn,new} \\ E_1, E_2, \dots, E_m \end{array} \right\rangle$	

Table 2: Typing rules for fold

$t : T_0 \quad \langle E_0 \rangle, \quad X_{11} : T_{11}, \dots, X_{1a} : T_{1a} \vdash t_1 : T_1 \quad \langle E_1 \rangle, \quad \dots, \quad X_{m1} : T_{m1}, \dots, X_{mn} : T_{mn} \vdash t_m : T_m \quad \langle E_m \rangle$		
$\Gamma \vdash \begin{array}{l} \text{case } t \\ \text{of} \end{array} \left  \begin{array}{l} C_1 : X_{11}, \dots, X_{1a} \rightarrow t_1 \\ \vdots \\ C_m : X_{m1}, \dots, X_{mn} \rightarrow t_n \end{array} \right. : T$	$\left\langle \exists \begin{array}{l} T_0, T_1, \dots, T_m, \\ T_{11}, \dots, T_{1a}, \\ \vdots \\ T_{m1}, \dots, T_{mn}, \\ A'_1, \dots, A'_k \end{array} \begin{array}{l} T_0 = D_{new} \\ T_1 = T, \dots, T_m = T \\ T_{11} = F_{11,new}, \dots, T_{1a} = F_{1a,new} \\ \vdots \\ T_{m1} = F_{m1,new}, \dots, T_{mn} = F_{mn,new} \\ E_1, E_2, \dots, E_m \end{array} \right\rangle$	

Table 3: Typing rules for Case