# LinML

### Ada Vienot

# 1 Typing

The types in LinML have the following syntax:

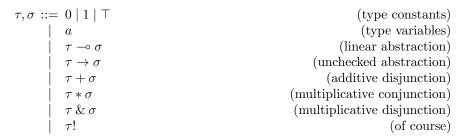


Figure 1: LinML types

## 1.1 Typing rules

We define the inductive predicate exp over types as follows:

$$\exp(\tau) = \begin{cases} \text{true} & \text{if } \tau = * \text{ or } \tau = !\sigma \\ \text{false} & \text{elsewise} \end{cases}$$

In all the following, typing contexts  $\Gamma$ ;  $\Delta$  are to be seen as a pair with:

- $\Gamma$  a multiset of pairs  $(x:\tau)$ , with x a variable and  $\tau$  a type such that  $\exp(\tau) = \text{false}$
- $\Delta$  a set of pairs  $(x:\tau)$ , with x a variable and  $\tau$  a type such that  $\exp(\tau)$  = true

Intuitively,  $\Gamma$  contains the linear bindings, and  $\Delta$  contains the exponential bindings.

Since  $\forall \tau, \exp(\tau) = true$  implies the existence of context weakening and contraction rules for  $\tau$ , we treat the context  $\Delta$  as the kind of contexts we manipulate in intuitionistic sequent calculus, that is: they can be duplicated, erased, and one binding is allowed to erase another. This is not the case of the context  $\Gamma$  containing linear bindings.

### 1.1.1 Terms

Let t, u be terms,  $\tau, \sigma$  types, and  $(\Gamma; \Delta), (\Gamma'; \Delta')$  typing contexts. The typing judgements for terms have the following shape:

$$\Gamma; \Delta \vdash t : \tau \Rightarrow \Gamma'; \Delta'$$

 $\Gamma'; \Delta'$  is the typing context after consuming the necessary bindings to type  $t : \tau$ .

The typing rules are the following:

Figure 2: LinML terms typing rules

#### 1.1.2 Patterns