# Clam semantics

**Maxime Mulder**

## Introduction

This document is the work-in-progress specification of the Clam programming language semantics. It is mostly based on the semantics of System $F^{\omega}_{<:}$ and captures most of the behaviours of Clam. However, it is not yet complete and lacks proofs. It could also benefit from more elegant evaluation semantics.

This document is written using Typst.

## Contents

# Grammar

The abstract syntax of Clam is given by the following grammar:

$$u ::= \texttt{unit} \qquad b ::= \texttt{true} \mid \texttt{false} \qquad n ::= 0 \mid 1 \mid \ldots \qquad s ::= \texttt{"}\ldots\texttt{"}$$

| | | |
|---|---|---|
| Expression $e$ ::= | $u$ | unit |
| | $\mid b$ | boolean |
| | $\mid n$ | integer |
| | $\mid s$ | string |
| | $\mid \langle e_1, \ldots, e_n \rangle$ | tuple |
| | $\mid e.n$ | tuple projection |
| | $\mid \langle l_1 = e_1, \ldots, l_n = e_n \rangle$ | record |
| | $\mid e.l$ | record projection |
| | $\mid \lambda x : \tau.\ e$ | abstraction |
| | $\mid e(e)$ | application |
| | $\mid \lambda T <: \tau.\ e$ | universal abstraction |
| | $\mid e[\tau]$ | universal application |
| | $\mid e : \tau$ | type ascription |

| | | |
|---|---|---|
| Type $\tau$ ::= | $\top$ | top |
| | $\mid \bot$ | bottom |
| | $\mid \tau \cup \tau$ | union |
| | $\mid \tau \cap \tau$ | intersection |
| | $\mid \texttt{Unit}$ | unit |
| | $\mid \texttt{Bool}$ | boolean |
| | $\mid \texttt{Int}$ | integer |
| | $\mid \texttt{String}$ | string |
| | $\mid \langle \tau_1, \ldots, \tau_n \rangle$ | tuple |
| | $\mid \langle l_1 : \tau_1, \ldots, l_n : \tau_n \rangle$ | record |
| | $\mid \tau \to \tau$ | abstraction |
| | $\mid \forall T <: \tau.\ \tau$ | universal abstraction |
| | $\mid \Lambda T <: \tau.\ \tau$ | type abstraction |
| | $\mid \tau[\tau]$ | type application |

# Kinding

The syntax of kinds is given by the following grammar:

$$\begin{array}{rcll} \text{Kind } k & ::= & * & \text{type} \\ & | & k \to k & \text{arrow} \end{array}$$

The kinding judgement is of the form $\Delta \vdash \tau :: k$. The hypotheses of $\Delta$ are of the form $T <: \tau$.

$$\frac{}{\texttt{Unit} :: *}\ \textsc{K-Unit} \qquad \frac{}{\texttt{Bool} :: *}\ \textsc{K-Bool} \qquad \frac{}{\texttt{Int} :: *}\ \textsc{K-Int} \qquad \frac{}{\texttt{String} :: *}\ \textsc{K-String}$$

$$\frac{}{\top :: *}\ \textsc{K-Top} \qquad \frac{}{\bot :: *}\ \textsc{K-Bot} \qquad \frac{T <: \tau \in \Delta \qquad \Delta \vdash \tau :: K}{\Delta \vdash T :: K}\ \textsc{K-Var}$$

$$\frac{\Delta \vdash \tau_1 :: * \quad ... \quad \Delta \vdash \tau_n :: *}{\Delta \vdash \langle \tau_1, ..., \tau_n \rangle :: *}\ \textsc{K-Tuple} \qquad \frac{\Delta \vdash \tau_1 :: * \quad ... \quad \Delta \vdash \tau_n :: *}{\Delta \vdash \langle l_1 : \tau_1, ..., l_n : \tau_n \rangle :: *}\ \textsc{K-Record}$$

$$\frac{\Delta \vdash \tau_1 :: * \qquad \Delta \vdash \tau_2 :: *}{\Delta \vdash \tau_1 \to \tau_2 :: *}\ \textsc{K-Abs} \qquad \frac{\Delta, T <: \tau_1 \vdash \tau_2 :: *}{\Delta \vdash \forall T <: \tau_1.\ \tau_2 :: *}\ \textsc{K-Univ}$$

$$\frac{\Delta \vdash \tau_1 :: k_1 \qquad \Delta, T <: \tau_1 \vdash \tau_2 :: k_2}{\Delta \vdash \Lambda T <: \tau_1.\ \tau_2 :: k_1 \to k_2}\ \textsc{K-TAbs}$$

$$\frac{\Delta \vdash \tau_1 <: \Lambda T <: \tau_2. \to \tau_3 \qquad \Delta \vdash \tau_3 :: k \qquad \Delta \vdash \tau_4 <: \tau_2}{\Delta \vdash \tau_1[\tau_4] :: k}\ \textsc{K-TAbsApp}$$

$$\frac{\Delta \vdash \tau_1 :: k \qquad \Delta \vdash \tau_2 :: k}{\Delta \vdash \tau_1 \cup \tau_2 :: k}\ \textsc{K-Union} \qquad \frac{\Delta \vdash \tau_1 :: k \qquad \Delta \vdash \tau_2 :: k}{\Delta \vdash \tau_1 \cap \tau_2 :: k}\ \textsc{K-Inter}$$

## Type equivalence

The type equivalence judgement is of the form $\Delta \vdash \tau \equiv \tau'$.

$$\frac{}{\tau \equiv \tau}\ \textsc{E-Refl} \qquad \frac{\Delta \vdash \tau_1 \equiv \tau_2}{\Delta \vdash \tau_2 \equiv \tau_1}\ \textsc{E-Symm} \qquad \frac{\Delta \vdash \tau_1 \equiv \tau_2 \qquad \Delta \vdash \tau_2 \equiv \tau_3}{\Delta \vdash \tau_1 \equiv \tau_3}\ \textsc{E-Trans}$$

$$\frac{\Delta \vdash \tau_1 \equiv \tau_1' \quad \ldots \quad \Delta \vdash \tau_n \equiv \tau_n'}{\Delta \vdash \langle \tau_1, \ldots, \tau_n \rangle \equiv \langle \tau_1', \ldots, \tau_n' \rangle}\ \textsc{E-Tuple}$$

$$\frac{\Delta \vdash \tau_1 \equiv \tau_1' \quad \ldots \quad \Delta \vdash \tau_n \equiv \tau_n'}{\Delta \vdash \langle l_1 : \tau_1, \ldots, l_n : \tau_n \rangle \equiv \langle l_1 : \tau_1', \ldots, l_n : \tau_n' \rangle}\ \textsc{E-Record}$$

$$\frac{\Delta \vdash \tau_1 \equiv \tau_1' \qquad \Delta \vdash \tau_2 \equiv \tau_2'}{\Delta \vdash \tau_1 \to \tau_2 \equiv \tau_1' \to \tau_2'}\ \textsc{E-Abs} \qquad \frac{\Delta \vdash \tau_1 \equiv \tau_1' \qquad \Delta \vdash \tau_2 \equiv \tau_2'}{\Delta \vdash \forall X <: \tau_1.\ \tau_2 \equiv \forall X <: \tau_1'.\ \tau_2'}\ \textsc{E-Univ}$$

$$\frac{\Delta \vdash \tau_1 \equiv \tau_1' \qquad \Delta \vdash \tau_2 \equiv \tau_2'}{\Delta \vdash \Lambda T <: \tau_1.\ \tau_2 \equiv \Lambda T <: \tau_1'.\ \tau_2'}\ \textsc{E-TAbs} \qquad \frac{\Delta \vdash \tau_1 \equiv \tau_1' \qquad \Delta \vdash \tau_2 \equiv \tau_2'}{\Delta \vdash \tau_1[\tau_2] \equiv \tau_1'[\tau_2']}\ \textsc{E-TApp}$$

$$\frac{}{(\Lambda T <: \tau_1.\ \tau_2)[\tau_3] \equiv [T/\tau_3]\tau_2}\ \textsc{E-TAbsApp}$$

$$\frac{}{\tau_1 \cup \tau_2 \equiv \tau_2 \cup \tau_1}\ \textsc{E-UnionSymm} \qquad \frac{}{\tau_1 \cap \tau_2 \equiv \tau_2 \cap \tau_1}\ \textsc{E-InterSymm}$$

$$\frac{\Delta \vdash \tau_1 <: \tau_2}{\Delta \vdash \tau_1 \cup \tau_2 \equiv \tau_2}\ \textsc{E-UnionSub} \qquad \frac{\Delta \vdash \tau_1 <: \tau_2}{\Delta \vdash \tau_1 \equiv \tau_1 \cap \tau_2}\ \textsc{E-InterSub}$$

## Subtyping

The subtyping judgement is of the form $\Delta \vdash \tau <: \tau'$.

$$\frac{\Delta \vdash \tau_1 \equiv \tau_2}{\Delta \vdash \tau_1 <: \tau_2}\,\textsc{S-Eq} \qquad \frac{\Delta \vdash \tau_1 <: \tau_2 \qquad \Delta \vdash \tau_2 <: \tau_3}{\Delta \vdash \tau_1 <: \tau_3}\,\textsc{S-Trans} \qquad \frac{T <: \tau \in \Delta}{\Delta \vdash T <: \tau}\,\textsc{S-Var}$$

$$\frac{\Delta \vdash \tau :: *}{\Delta \vdash \tau <: \top}\,\textsc{S-Top} \qquad \frac{\Delta \vdash \tau :: *}{\Delta \vdash \bot <: \tau}\,\textsc{S-Bot}$$

$$\frac{\Delta \vdash \tau_1 <: \tau_1' \quad \ldots \quad \Delta \vdash \tau_n <: \tau_n'}{\Delta \vdash \langle \tau_1, \ldots, \tau_n \rangle <: \langle \tau_1', \ldots, \tau_n' \rangle}\,\textsc{S-Tuple}$$

$$\frac{\Delta \vdash \tau_1 <: \tau_1' \quad \ldots \quad \Delta \vdash \tau_n <: \tau_n' \qquad n \leqslant m}{\Delta \vdash \langle l_1 : \tau_1, \ldots, l_m : \tau_m \rangle <: \langle l_1 : \tau_1', \ldots, l_n : \tau_n' \rangle}\,\textsc{S-Record}$$

$$\frac{\Delta \vdash \tau_1' <: \tau_1 \qquad \Delta \vdash \tau_2 <: \tau_2'}{\Delta \vdash \tau_1 \to \tau_2 <: \tau_1' \to \tau_2'}\,\textsc{S-Abs} \qquad \frac{\Delta \vdash \tau_1 \equiv \tau_1' \qquad \Delta, T <: \tau_1 \vdash \tau_2 <: \tau_2'}{\Delta \vdash \forall T <: \tau_1.\ \tau_2 <: \forall T <: \tau_1'.\ \tau_2'}\,\textsc{S-Univ}$$

$$\frac{\Delta \vdash \tau_1' \equiv \tau_1 \qquad \Delta \vdash \tau_2 <: \tau_2'}{\Delta \vdash \Lambda T <: \tau_1.\ \tau_2 <: \Lambda T <: \tau_1'.\ \tau_2'}\,\textsc{S-TAbs} \qquad \frac{\Delta \vdash \tau_1' <: \tau_1 \qquad \Delta \vdash \tau_2 \equiv \tau_2'}{\Delta \vdash \tau_1[\tau_2] <: \tau_1'[\tau_2']}\,\textsc{S-TApp}$$

$$\frac{\Delta \vdash \tau <: \tau_1}{\Delta \vdash \tau <: \tau_1 \cup \tau_2}\,\textsc{S-Union1} \qquad \frac{\Delta \vdash \tau_1 <: \tau}{\Delta \vdash \tau_1 \cap \tau_2 <: \tau}\,\textsc{S-Inter2}$$

$$\frac{\Delta \vdash \tau_1 <: \tau \qquad \Delta \vdash \tau_2 <: \tau}{\Delta \vdash \tau_1 \cup \tau_2 <: \tau}\,\textsc{S-Union1} \qquad \frac{\Delta \vdash \tau <: \tau_1 \qquad \Delta \vdash \tau <: \tau_2}{\Delta \vdash \tau <: \tau_1 \cap \tau_2}\,\textsc{S-Inter2}$$

## Typing

The typing judgement is of the form $\Delta\ \Gamma \vdash e : \tau$. The hypotheses of $\Gamma$ are of the form $x : \tau$.

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau}\text{T-VAR} \qquad \frac{\Delta\ \Gamma \vdash e : \tau \qquad \Delta \vdash \tau <: \tau'}{\Delta\ \Gamma \vdash e : \tau'}\text{T-SUB}$$

$$\frac{}{u : \texttt{Unit}}\text{T-UNIT} \qquad \frac{}{b : \texttt{Bool}}\text{T-BOOL} \qquad \frac{}{i : \texttt{Int}}\text{T-INT} \qquad \frac{}{s : \texttt{String}}\text{T-STRING}$$

$$\frac{\Delta\ \Gamma \vdash e_1 : \tau_1 \quad \dots \quad \Delta\ \Gamma \vdash e_n : \tau_n}{\Delta\ \Gamma \vdash \langle e_1, \dots e_n \rangle : \langle \tau_1, \dots, \tau_n \rangle}\text{T-TUPLE} \qquad \frac{\Delta\ \Gamma \vdash e : \langle \tau_1, \dots, \tau_n \rangle}{\Delta\ \Gamma \vdash e.i : \tau_i}\text{T-TUPLEPROJ}$$

$$\frac{\Delta\ \Gamma \vdash e_1 : \tau_1 \quad \dots \quad \Delta\ \Gamma \vdash e_n : \tau_n}{\Delta\ \Gamma \vdash \langle l_1 = e_1, \dots, l_n = e_n \rangle : \langle l_1 : \tau_1, \dots, l_n : \tau_n \rangle}\text{T-RECORD}$$

$$\frac{\Delta\ \Gamma \vdash e : \langle l_1 : \tau_1, \dots, l_n : \tau_n \rangle}{\Delta\ \Gamma \vdash e.l_i : \tau_i}\text{T-RECORDPROJ}$$

$$\frac{\Delta \vdash \tau_1 :: * \qquad \Delta\ \Gamma, x : \tau_1 \vdash e : \tau_2}{\Delta\ \Gamma \vdash \lambda x : \tau_1.\ e : \tau_1 \to \tau_2}\text{T-ABS} \qquad \frac{\Delta\ \Gamma \vdash e_1 : \tau_1 \to \tau_2 \qquad \Delta\ \Gamma \vdash e_2 : \tau_1}{\Delta\ \Gamma \vdash e_1(e_2) : \tau_2}\text{T-ABSAPP}$$

$$\frac{\Delta, T <: \tau_1\ \Gamma \vdash e : \tau_2}{\Delta\ \Gamma \vdash \lambda T <: \tau_1.\ e : \forall T <: \tau_1.\ \tau_2}\text{T-UNIV}$$

$$\frac{\Delta\ \Gamma \vdash e : \forall T <: \tau_1.\ \tau_2 \qquad \Delta \vdash \tau_3 <: \tau_1}{\Delta\ \Gamma \vdash e[\tau_3] : [T/\tau_3]\tau_2}\text{T-UNIVAPP}$$

$$\frac{\Delta \vdash \tau :: *}{\Delta\ \Gamma \vdash e : \tau : \tau}\text{T-ASCR}$$

## Primitives

The types of the primitive values of Clam are given by the following table:

| Symbol | Type |
|---|---|
| `+` $(unary)$ | `Int` $\to$ `Int` |
| `-` $(unary)$ | `Int` $\to$ `Int` |
| `!` | `Bool` $\to$ `Bool` |
| `+` $(binary)$ | `Int` $\to$ `Int` $\to$ `Int` |
| `-` $(binary)$ | `Int` $\to$ `Int` $\to$ `Int` |
| `*` | `Int` $\to$ `Int` $\to$ `Int` |
| `/` | `Int` $\to$ `Int` $\to$ `Int` |
| `%` | `Int` $\to$ `Int` $\to$ `Int` |
| `++` | `String` $\to$ `String` $\to$ `String` |
| `==` | $\top \to \top \to$ `Bool` |
| `!=` | $\top \to \top \to$ `Bool` |
| `<` | `Int` $\to$ `Int` $\to$ `Bool` |
| `>` | `Int` $\to$ `Int` $\to$ `Bool` |
| `<=` | `Int` $\to$ `Int` $\to$ `Bool` |
| `>=` | `Int` $\to$ `Int` $\to$ `Bool` |
| `\|` | `Bool` $\to$ `Bool` $\to$ `Bool` |
| `&` | `Bool` $\to$ `Bool` $\to$ `Bool` |
| `if` | $\forall T_1.\ \forall T_2.$ `Bool` $\to T_1 \to T_2 \to T_1 \cup T_2$ |

Note: `if` is not implemented as a primitive value yet.

# Evaluation

The syntax of values is given by the following grammar:

$$
\begin{array}{rll}
\text{Value } v ::= & u & \text{unit} \\
| & b & \text{boolean} \\
| & i & \text{integer} \\
| & s & \text{string} \\
| & \langle v_1, ..., v_n \rangle & \text{tuple} \\
| & \langle l_1 = v_1, ..., l_n = v_n \rangle & \text{record} \\
| & \lambda x.\ e & \text{abstraction} \\
| & \lambda T.\ e & \text{universal abstraction}
\end{array}
$$

The evaluation judgement is of the form $e \Downarrow v$.

$$
\frac{}{u \Downarrow u}\ \textsc{V-Unit} \qquad \frac{}{b \Downarrow b}\ \textsc{V-Bool} \qquad \frac{n = i}{n \Downarrow i}\ \textsc{V-Int} \qquad \frac{}{s \Downarrow s}\ \textsc{V-String}
$$

$$
\frac{e_1 \Downarrow v_1 \quad ... \quad e_n \Downarrow v_n}{\langle e_1, ...e_n \rangle \Downarrow \langle v_1, ..., v_n \rangle}\ \textsc{V-Tuple} \qquad \frac{e \Downarrow \langle v_1, ..., v_n \rangle}{e.i \Downarrow v_i}\ \textsc{V-TupleProj}
$$

$$
\frac{e_1 \Downarrow v_1 \quad ... \quad e_n \Downarrow v_n}{\langle l_1 = e_1, ..., l_n = e_n \rangle \Downarrow \langle l_1 = v_1, ..., l_n = v_n \rangle}\ \textsc{V-Record}
$$

$$
\frac{e \Downarrow \langle l_1 = v_1, ..., l_n = v_n \rangle}{e.l_i \Downarrow v_i}\ \textsc{V-RecordProj}
$$

$$
\frac{}{\lambda x : \tau.\ e \Downarrow \lambda x.\ e}\ \textsc{V-Abs} \qquad \frac{e_1 \Downarrow \lambda x.\ e_3 \qquad [e_2/x]e_3 \Downarrow v}{e_1(e_2) \Downarrow v}\ \textsc{V-AbsApp}
$$

$$
\frac{}{\lambda T <: \tau.\ e \Downarrow \lambda T.\ e}\ \textsc{V-Univ} \qquad \frac{e_1 \Downarrow \lambda T.\ e_2}{e_1[\tau] \Downarrow e_2}\ \textsc{V-UnivApp}
$$

$$
\frac{e \Downarrow v}{e : \tau \Downarrow v}\ \textsc{V-Ascr}
$$