1 Inductive Linear Types

1.1 Arity

For some type A and sort s, the judgment $\mathit{arity}(A, s)$ is inductively defined over the structure of A as follows.

$$\frac{}{\mathit{arity}(s, s)}\,\text{AR-SORT} \qquad \frac{\mathit{arity}(A, s)}{\mathit{arity}((x :_U M) \to A, s)}\,\text{AR-ARROW}$$

For some arity type A and sort s', the term $A\langle s' \rangle$ is inductively defined over the structure of A as follows.

$$((x :_U M) \to A)\langle s' \rangle = (x :_U M) \to A\langle s' \rangle$$
$$(s)\langle s' \rangle = s'$$

For some arity type A, term I and sort s', the term $A\{I, s'\}$ is inductively defined over the structure of A as follows.

$$((x :_U M) \to A)\langle I, s' \rangle = (x :_U M) \to A\langle (I\ x), s' \rangle$$
$$(s)\langle I, s' \rangle = (\_ :_U I) \to s'$$

1.2 Strict Positivity

For some type P and type variable X, the judgment $\mathit{postive}(P, X)$ is inductively defined over the structure of P as follows.

$$\frac{(\forall i = 1...n) \qquad X \notin m_i}{\mathit{postive}((X\ m_1...m_n), X)}\,\text{POS-X} \qquad \frac{X \notin M \qquad \mathit{postive}(P, X)}{\mathit{postive}((x :_s M) \to P, X)}\,\text{POS-ARROW} \qquad \frac{X \notin M \qquad \mathit{postive}(P, X)}{\mathit{postive}((x :_s M) \multimap P, X)}\,\text{POS-LOLLI}$$

We say that X occurs strictly postive in P if the judgment $\mathit{postive}(P, X)$ is valid.

1.3 Non-Linear Constructor

For some type C and type variable X, the judgment $\mathit{constructor}_U(C, X)$ is inductively defined over the structure of C as follows.

$$\frac{(\forall i = 1...n) \qquad X \notin m_i}{\mathit{constructor}_U((X\ m_i...m_n), X)}\,\text{U-CONSTR-X} \qquad \frac{\mathit{postive}(P, X) \qquad \mathit{constructor}_U(C, X)}{\mathit{constructor}_U((\_ :_U P) \to C, X)}\,\text{U-CONSTR-POS}$$

$$\frac{X \notin M \qquad \mathit{constructor}_U(C, X)}{\mathit{constructor}_U((x :_U M) \to C, X)}\,\text{U-CONSTR-ARROW}$$

Non-linear constructors are used for constructing non-linear objects that may be freely used. These constructors may not applied to linear terms as this may cause duplication of linear variables.

1.4 Linear Constructor

For some type C and type variable X, the judgment $\mathit{constructor}_L(C, X)$ is inductively defined over the structure of C as follows.

$$\frac{(\forall i = 1...n) \qquad X \notin m_i}{\mathit{constructor}_L((X\ m_i...m_n), X)}\,\text{L-CONSTR-X}$$

$$\frac{\mathit{postive}(P, X) \qquad \mathit{constructor}_L(C, X)}{\mathit{constructor}_L((\_ :_U P) \to C, X)}\,\text{L-CONSTR-POS-1} \qquad \frac{X \notin M \qquad \mathit{constructor}_L(C, X)}{\mathit{constructor}_L((x :_U M) \to C, X)}\,\text{L-CONSTR-ARROW-1}$$

$$\frac{\mathit{postive}(P, X) \qquad \mathit{activation}(C, X)}{\mathit{constructor}_L((\_ :_L P) \to C, X)}\,\text{L-CONSTR-POS-2} \qquad \frac{X \notin M \qquad \mathit{activation}(C, X)}{\mathit{constructor}_L((x :_L M) \to C, X)}\,\text{L-CONSTR-ARROW-2}$$

For some type C and type variable X, the judgment $\mathit{activation}(C, X)$ is inductively defined over the structure of C as follows.

$$\frac{(\forall i = 1...n) \qquad X \notin m_i}{\mathit{activation}((X\ m_i...m_n), X)}\,\text{ACT-X} \qquad \frac{\mathit{postive}(P, X) \qquad \mathit{activation}(C, X)}{\mathit{activation}((\_ :_s P) \multimap C, X)}\,\text{ACT-POS}$$

$$\frac{X \notin M \qquad \mathit{activation}(C, X)}{\mathit{activation}((x :_s M) \multimap C, X)}\,\text{ACT-LOLLI}$$

Linear constructors are used for constructing linear objects that must be used once. These constructors can be applied to linear terms as restricted usage prevent duplication of linear variables.

An unapplied linear constructor is a non-linear entity as they can be created freely "out of thin air". However, once a linear constructor has been partially applied to a linear term, its linearity is "activated" as seen in rules L-CONSTR-POS-2 and L-CONSTR-ARROW-2, essentially forcing the rest of its arguments to be fully applied. This is done to prevent partially applied constructors from duplicating linear variables.

1.5 Introduction Rules

The formation of new inductive types is defined as follows.

$$\frac{(\forall i = 1...n) \qquad \mathit{arity}(A, s) \qquad \mathit{constructor}_s(C_i, X) \qquad |\Gamma| \qquad \Gamma \vdash A : U_k \qquad \Gamma, X :_U A \vdash C_i : U_k}{\Gamma \vdash \mathrm{Ind}_s(X : A)\{C_1|...|C_n\} : A}\text{IND-INTRO}$$

The formation of constructors for defined inductive types is defined as follows.

$$\frac{|\Gamma| \qquad I := \mathrm{Ind}_s(X : A)\{C_1|...|C_n\} \qquad \Gamma \vdash I : A \qquad 1 \leq i \leq n}{\Gamma \vdash \mathrm{Constr}(i, I) : C_i[I/X]}\text{CONSTR-INTRO}$$

1.6 Non-Dependent Case

For some constructor type C and type variables X and Q, the term $C\{X, Q\}$ is inductively defined over the structure of C as follows.

$$\begin{aligned}
((_ :_s P) \to C)\{X, Q\} &= (_ :_s P) \multimap C\{X, Q\} \\
((_ :_s P) \multimap C)\{X, Q\} &= (_ :_s P) \multimap C\{X, Q\} \\
((x :_s M) \to C)\{X, Q\} &= (x :_s P) \multimap C\{X, Q\} \\
((x :_s M) \multimap C)\{X, Q\} &= (x :_s P) \multimap C\{X, Q\} \\
(X\ m_1...m_n)\{X, Q\} &= (Q\ m_1...m_n)
\end{aligned}$$

We define the notation $C[I, P] := C\{X, Q\}[I/X, P/Q]$.

Inductive objects can be eliminated by a case expression as follows.

$$\frac{(\forall i = 1...n) \qquad \Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma \qquad \mathit{arity}(A, s) \qquad I := \mathrm{Ind}_s(X : A)\{C_1|...|C_n\} \qquad \Gamma_1 \vdash m : (I\ a_1...a_n) \qquad \overline{\Gamma_2} \vdash Q : A\langle s' \rangle \qquad \Gamma_2 \vdash f_i : C_i[I, Q]}{\Gamma \vdash \mathrm{Case}(m, Q)\{f_1|...|f_n\} : (Q\ a_1...a_n)}\text{CASE}$$

1.7 Dependent Case

For some non-linear constructor type C and type variables X, Q and c, the term $C\{X, Q, c\}$ is inductively defined over the structure of C as follows.

$$\begin{aligned}
((_ :_U P) \to C)\{X, Q, c\} &= (p :_U P) \multimap C\{X, Q, (c\ p)\} \\
((x :_U M) \to C)\{X, Q, c\} &= (x :_U M) \multimap C\{X, Q, (c\ x)\} \\
(X\ m_1...m_n)\{X, Q, c\} &= (Q\ m_1...m_n\ c)
\end{aligned}$$

We define the notation $C[I, P, t] := C\{X, Q, c\}[I/X, P/Q, t/c]$.

Non-linear inductive objects can be eliminated by a dependent case expression as follows.

$$\frac{(\forall i = 1...n) \qquad \Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma \qquad \mathit{arity}(A, U_i) \qquad I := \mathrm{Ind}_U(X : A)\{C_1|...|C_n\} \qquad |\Gamma_1| \qquad \Gamma_1 \vdash m : (I\ a_1...a_n) \qquad \overline{\Gamma_2} \vdash Q : A\langle I, s' \rangle \qquad \Gamma_2 \vdash f_i : C_i[I, Q, \mathit{Constr}(i, I)]}{\Gamma \vdash \mathrm{DCase}(m, Q)\{f_1|...|f_n\} : (Q\ a_1...a_n\ m)}\text{DCASE}$$

1.8 Fixpoint

$$\frac{|\Gamma| \qquad \Gamma \vdash T : U_i \qquad \Gamma, f :_U T \vdash m : T \qquad \text{guard condition}}{\Gamma \vdash \mathrm{Fix}\ f.m : T}\text{FIX}$$

1.9 Conversion

$$\begin{aligned}
\text{Case}((\text{Constr}(i, I)\ a_1 ... a_n), Q)\{f_1|...f_n\} &\rightsquigarrow_\iota (f_i\ a_1 ... a_n) \\
\text{DCase}((\text{Constr}(i, I)\ a_1 ... a_n), Q)\{f_1|...f_n\} &\rightsquigarrow_\iota (f_i\ a_1 ... a_n) \\
\text{Fix}\ f.m &\rightsquigarrow_\iota m[(\text{Fix}\ f.m)/f]
\end{aligned}$$