Compiling from F_i^+ to JavaScript
(simplified scheme with call-by-value)

Yaozhu Sun

March 4, 2023

Syntax of F_i^+

$$
\begin{array}{lrl}
\text{Types} & A, B, C ::= & \top \mid \bot \mid \mathbb{B} \mid X \mid A \to B \mid \forall X{*}A.\ B \mid \{\ell : A\} \mid A \,\&\, B \\
\text{Type indices} & T ::= & \mathbb{B} \mid \overrightarrow{T} \mid T^{\forall} \mid \{\ell : T\} \mid T_1 \,\&\, T_2 \\
\text{Expressions} & e ::= & \{\} \mid b \mid x \mid \textbf{fix}\, x{:}A.\ e \mid \lambda x{:}A.\ e{:}B \mid e_1\ e_2 \mid \Lambda X{*}A.\ e{:}B \mid e\ A \mid \{\ell = e\} \mid e.\ell \\
& & \mid e_1 \,,,\, e_2 \mid e : A \\
\text{Values} & v ::= & \{\} \mid b \mid \lambda x{:}A.\ e{:}B \mid \Lambda X{*}A.\ e{:}B \mid \{\ell = v\} \mid v_1 \,,,\, v_2
\end{array}
$$

$\boxed{|A| = T}$ *(Type translation)*

$$
|\mathbb{B}| = \mathbb{B} \qquad |X| = \textbf{atoi}(X) \qquad |A \to B| = \overrightarrow{|B|} \qquad |\forall X{*}A.\ B| = |B|^{\forall} \qquad |\{\ell : A\}| = \{\ell : |A|\}
$$

$$
\frac{A_{k_1} < A_{k_2} < \cdots < A_{k_m} \qquad \neg\rceil A_k \lceil}{|A_1 \& A_2 \& \cdots \& A_n| = |A_{k_1}| \& |A_{k_2}| \& \cdots \& |A_{k_m}|}
$$

$\boxed{\rceil A \lceil}$ *(Top-like types)*

$$
\textsc{TL-Top}\ \frac{}{\rceil \top \lceil} \qquad
\textsc{TL-And}\ \frac{\rceil A \lceil \qquad \rceil B \lceil}{\rceil A \,\&\, B \lceil} \qquad
\textsc{TL-Arrow}\ \frac{\rceil B \lceil}{\rceil A \to B \lceil} \qquad
\textsc{TL-All}\ \frac{\rceil B \lceil}{\rceil \forall X{*}A.\ B \lceil} \qquad
\textsc{TL-Rcd}\ \frac{\rceil A \lceil}{\rceil \{\ell : A\} \lceil}
$$

$\boxed{A^{\circ}}$ *(Ordinary types)*

$$
\textsc{O-Top}\ \frac{}{\top^{\circ}} \qquad
\textsc{O-Bot}\ \frac{}{\bot^{\circ}} \qquad
\textsc{O-Base}\ \frac{}{\mathbb{B}^{\circ}} \qquad
\textsc{O-Var}\ \frac{}{X^{\circ}} \qquad
\textsc{O-Arrow}\ \frac{B^{\circ}}{(A \to B)^{\circ}} \qquad
\textsc{O-All}\ \frac{B^{\circ}}{(\forall X{*}A.\ B)^{\circ}} \qquad
\textsc{O-Rcd}\ \frac{A^{\circ}}{\{\ell : A\}^{\circ}}
$$

$\boxed{\Gamma \vdash e \Leftrightarrow A \leadsto J \mid z^{\pm}}$ *(Type-directed compilation)*

$$\textsc{J-Gen}\quad \frac{\Gamma \vdash e \Leftrightarrow A \leadsto J \mid z^-}{\Gamma \vdash e \Leftrightarrow A \leadsto \textsf{code} \mid z^+}$$

$$\textsc{J-Top}\quad \frac{}{\Gamma \vdash \{\} \Rightarrow \top \leadsto \varnothing \mid z^-}$$

$$\textsc{J-TopAbs}\quad \frac{\rceil B \lceil}{\Gamma \vdash \lambda x : A.\ e : B \Rightarrow A \to B \leadsto \varnothing \mid z^-}$$

$$\textsc{J-TopTAbs}\quad \frac{\rceil B \lceil}{\Gamma \vdash \Lambda X * A.\ e : B \Rightarrow \forall X * A.\ B \leadsto \varnothing \mid z^-}$$

$$\textsc{J-TopRcd}\quad \frac{\Gamma \vdash e \Rightarrow A \qquad \rceil A \lceil}{\Gamma \vdash \{\ell = e\} \Rightarrow \{\ell : A\} \leadsto \varnothing \mid z^-}$$

$$\textsc{J-Base}\quad \frac{T = |\mathbb{B}|}{\Gamma \vdash b \Rightarrow \mathbb{B} \leadsto \textsf{code} \mid z^-}$$

$$\textsc{J-Var}\quad \frac{x : A \in \Gamma}{\Gamma \vdash x \Rightarrow A \leadsto \textsf{code} \mid z^-}$$

$$\textsc{J-VarGen}\quad \frac{x : A \in \Gamma}{\Gamma \vdash x \Rightarrow A \leadsto \varnothing \mid x^+}$$

$$\textsc{J-Fix}\quad \frac{\Gamma, x : A \vdash e \Leftarrow A \leadsto J \mid z^-}{\Gamma \vdash \mathbf{fix}\, x : A.\ e \Rightarrow A \leadsto \textsf{code} \mid z^-}$$

$$\textsc{J-Abs}\quad \frac{\begin{array}{c} T = \overrightarrow{|B|} \\ \Gamma, x : A \vdash e \Leftarrow B \leadsto J \mid y^- \end{array}}{\Gamma \vdash \lambda x : A.\ e : B \Rightarrow A \to B \leadsto \textsf{code} \mid z^-}$$

$$\textsc{J-App}\quad \frac{\begin{array}{c} \Gamma \vdash e_1 \Rightarrow A \leadsto J_1 \mid x^+ \\ A \rhd B \to C \\ \Gamma \vdash e_2 \Leftarrow B \leadsto J_2 \mid y^+ \\ x : A \bullet y \leadsto J_3 \mid z \end{array}}{\Gamma \vdash e_1\, e_2 \Rightarrow C \leadsto J_1; J_2; J_3 \mid z^-}$$

$$\textsc{J-TAbs}\quad \frac{\begin{array}{c} T = |B|^{\forall} \\ \Gamma, X * A \vdash e \Leftarrow B \leadsto J \mid y^- \end{array}}{\Gamma \vdash \Lambda X * A.\ e : B \Rightarrow \forall X * A.\ B \leadsto \textsf{code} \mid z^-}$$

$$\textsc{J-TApp}\quad \frac{\begin{array}{c} \Gamma \vdash e \Rightarrow B \leadsto J_1 \mid y^+ \\ B \rhd \forall X * C_1.\ C_2 \\ \Gamma \vdash A * C_1 \qquad Ts = \mathbf{itoa}\, |A| \\ y : B \bullet Ts \leadsto J_2 \mid z \end{array}}{\Gamma \vdash e\, A \Rightarrow C_2[X \mapsto A] \leadsto J_1; J_2 \mid z^-}$$

$$\textsc{J-Rcd}\quad \frac{\begin{array}{c} T = \{\ell : |A|\} \\ \Gamma \vdash e \Rightarrow A \leadsto J \mid y^+ \end{array}}{\Gamma \vdash \{\ell = e\} \Rightarrow \{\ell : A\} \leadsto \textsf{code} \mid z^-}$$

$$\textsc{J-Proj}\quad \frac{\begin{array}{c} \Gamma \vdash e \Rightarrow A \leadsto J_1 \mid y^+ \\ A \rhd \{\ell : B\} \\ y : A \bullet \ell \leadsto J_2 \mid z \end{array}}{\Gamma \vdash e.\ell \Rightarrow B \leadsto J_1; J_2 \mid z^-}$$

$$\textsc{J-Merge}\quad \frac{\begin{array}{c} \Gamma \vdash e_1 \Rightarrow A \leadsto J_1 \mid z^- \\ \Gamma \vdash e_2 \Rightarrow B \leadsto J_2 \mid z^- \\ \Gamma \vdash A * B \end{array}}{\Gamma \vdash e_1 ,, e_2 \Rightarrow A \,\&\, B \leadsto J_1; J_2 \mid z^-}$$

$$\textsc{J-Anno}\quad \frac{\Gamma \vdash e \Leftarrow A \leadsto J \mid z^-}{\Gamma \vdash e : A \Rightarrow A \leadsto J \mid z^-}$$

$$\textsc{J-AnnoGen}\quad \frac{\Gamma \vdash e \Leftarrow A \leadsto J \mid z^+}{\Gamma \vdash e : A \Rightarrow A \leadsto J \mid z^+}$$

$$\textsc{J-Sub}\quad \frac{\begin{array}{c} \Gamma \vdash e \Rightarrow A \leadsto J_1 \mid x^+ \\ x : A <: y : B \leadsto J_2 \end{array}}{\Gamma \vdash e \Leftarrow B \leadsto J_1; J_2 \mid y^-}$$

$$\textsc{J-SubEqGen}\quad \frac{\Gamma \vdash e \Rightarrow A \leadsto J \mid z^+}{\Gamma \vdash e \Leftarrow A \leadsto J \mid z^+}$$

$$\textsc{J-Def}\quad \frac{\Gamma \vdash e_1 \Rightarrow A \;\boxed{\rightsquigarrow J_1 \mid x^-} \qquad \Gamma, x : A \vdash e_2 \Rightarrow B \;\boxed{\rightsquigarrow J_2 \mid z^-}}{\Gamma \vdash x = e1; e_2 \Rightarrow B \;\boxed{\rightsquigarrow \mathsf{code} \mid z^-}}$$

```
/* J-Gen */
var z = {};  J;

/* J-Base */
z[T] = b;

/* J-Var */
Object.assign(z, x);

/* J-Fix */
var x = z;  J;

/* J-Abs */
z[T] = (x, y) => { J };

/* J-TAbs */
z[T] = (X, y) => { J };

/* J-Rcd */
J;
z[T] = y;

/* J-Def */
export var x = {};
J1;
J2;
```

$\boxed{x : A \bullet arg \;\rightsquigarrow\; J \mid z}$ *(Distributive application)*

$$\textsc{A-Top}\quad \frac{\rceil A \lceil}{x : A \bullet arg \;\rightsquigarrow\; \varnothing \mid z} \qquad \textsc{A-Arrow}\quad \frac{T = \overrightarrow{|B|}}{x : A \to B \bullet y \;\rightsquigarrow\; \mathsf{code} \mid z} \qquad \textsc{A-All}\quad \frac{T = |B|^{\forall}}{x : \forall X {*} A.\, B \bullet Ts \;\rightsquigarrow\; \mathsf{code} \mid z}$$

$$\textsc{A-Rcd}\quad \frac{T = \{\ell : |A|\}}{x : \{\ell : A\} \bullet \ell \;\rightsquigarrow\; \mathsf{code} \mid z} \qquad \textsc{A-And}\quad \frac{x : A \bullet arg \;\rightsquigarrow\; J_1 \mid z \qquad x : B \bullet arg \;\rightsquigarrow\; J_2 \mid z}{x : A \,\&\, B \bullet arg \;\rightsquigarrow\; J_1; J_2 \mid z}$$

```
/* A-Arrow */
x[T](y, z);

/* A-All */
x[T](Ts, z);

/* A-Rcd */
Object.assign(z, x[T]);
```

$\boxed{x : A <: y : B \rightsquigarrow J}$ *(Coercive subtyping)*

$$\frac{B^\circ \qquad \rceil B \lceil}{x : A <: y : B \rightsquigarrow \varnothing}\;\textsc{S-Top} \qquad \frac{T = |A| \qquad A^\circ}{x : \bot <: y : A \rightsquigarrow \mathsf{code}}\;\textsc{S-Bot} \qquad \frac{}{x : A <: y : A \rightsquigarrow \mathsf{code}}\;\textsc{S-Eq}$$

$$\frac{T = |\mathbb{B}|}{x : \mathbb{B} <: y : \mathbb{B} \rightsquigarrow \mathsf{code}}\;\textsc{S-Base} \qquad \frac{}{x : X <: y : X \rightsquigarrow \mathsf{code}}\;\textsc{S-Var}$$

$$\frac{\begin{array}{c} T_1 = \overrightarrow{|A_2|} \\ T_2 = \overrightarrow{|B_2|} \qquad B_2{}^\circ \\ x_1 : B_1 <: y_1 : A_1 \rightsquigarrow J_1 \\ x_2 : A_2 <: y_2 : B_2 \rightsquigarrow J_2 \end{array}}{x : A_1 \to A_2 <: y : B_1 \to B_2 \rightsquigarrow \mathsf{code}}\;\textsc{S-Arrow}$$

$$\frac{\begin{array}{c} T_1 = |A_2|^\forall \qquad T_2 = |B_2|^\forall \\ B_2{}^\circ \qquad B_1 <: A_1 \\ x_0 : A_2 <: y_0 : B_2 \rightsquigarrow J \end{array}}{x : \forall X * A_1.\ A_2 <: y : \forall X * B_1.\ B_2 \rightsquigarrow \mathsf{code}}\;\textsc{S-All} \qquad \frac{\begin{array}{c} T_1 = \{\ell : |A|\} \\ T_2 = \{\ell : |B|\} \\ B^\circ \qquad x_0 : A <: y_0 : B \rightsquigarrow J \end{array}}{x : \{\ell : A\} <: y : \{\ell : B\} \rightsquigarrow \mathsf{code}}\;\textsc{S-Rcd}$$

$$\frac{C^\circ \qquad x : A <: y : C \rightsquigarrow J}{x : A \,\&\, B <: y : C \rightsquigarrow J}\;\textsc{S-AndL} \qquad \frac{C^\circ \qquad x : B <: y : C \rightsquigarrow J}{x : A \,\&\, B <: y : C \rightsquigarrow J}\;\textsc{S-AndR}$$

$$\frac{\begin{array}{c} B_1 \lhd B \rhd B_2 \\ y_1 : B_1 \rhd z : B \lhd y_2 : B_2 \rightsquigarrow J_3 \\ x : A <: y_1 : B_1 \rightsquigarrow J_1 \\ x : A <: y_2 : B_2 \rightsquigarrow J_2 \end{array}}{x : A <: z : B \rightsquigarrow \mathsf{code}}\;\textsc{S-Split}$$

```
/* S-Bot */
y[T] = null;

/* S-Eq */
Object.assign(y, x);

/* S-Base */
y[T] = x[T];

/* S-Var */
for (var T of X) {
  y[T] = x[T];
}

/* S-Arrow */
y[T2] = (x1, y2) => {
  var y1 = {};  J1;
  var x2 = {};
  x[T1](y1, x2);
  J2;
};

/* S-All */
y[T2] = (X, y0) => {
  var x0 = {};
  x[T1](X, x0);
  J;
};

/* S-Rcd */
var x0 = x[T1];
var y0 = {};  J;
y[T2] = y0;

/* S-Split */
var y1 = {}; // if y1 != z
var y2 = {}; // if y2 != z
J1;  J2;  J3;
```

$\boxed{x : A \ \rhd\ z : C \ \lhd\ y : B \ \rightsquigarrow\ J}$ *(Coercive merging)*

$$\frac{}{z : A \ \rhd\ z : A \,\&\, B \ \lhd\ z : B \ \rightsquigarrow\ \varnothing}\ \textsc{M-And}$$

$$\frac{T = |\overrightarrow{B}| \qquad T_1 = |\overrightarrow{B_1}| \qquad T_2 = |\overrightarrow{B_2}| \qquad y_1 : B_1 \ \rhd\ y : B \ \lhd\ y_2 : B_2 \ \rightsquigarrow\ J}{x_1 : A \to B_1 \ \rhd\ z : A \to B \ \lhd\ x_2 : A \to B_2 \ \rightsquigarrow\ \texttt{code}}\ \textsc{M-Arrow}$$

$$\frac{T = |B|^{\forall} \qquad T_1 = |B_1|^{\forall} \qquad T_2 = |B_2|^{\forall} \qquad y_1 : B_1 \ \rhd\ y : B \ \lhd\ y_2 : B_2 \ \rightsquigarrow\ J}{x_1 : \forall X * A.\ B_1 \ \rhd\ z : \forall X * A.\ B \ \lhd\ x_2 : \forall X * A.\ B_2 \ \rightsquigarrow\ \texttt{code}}\ \textsc{M-All}$$

$$\frac{T = \{\ell : |A|\} \qquad T_1 = \{\ell : |A_1|\} \qquad T_2 = \{\ell : |A_2|\} \qquad y_1 : A_1 \ \rhd\ y : A \ \lhd\ y_2 : A_2 \ \rightsquigarrow\ J}{x_1 : \{\ell : A_1\} \ \rhd\ z : \{\ell : A\} \ \lhd\ x_2 : \{\ell : A_2\} \ \rightsquigarrow\ \texttt{code}}\ \textsc{M-Rcd}$$

```
/* M-Arrow */
z[T] = (p, y) => {
  var y1 = {}; // if y1 != y
  var y2 = {}; // if y2 != y
  x1[T1](p, y1);
  x2[T2](p, y2);
  J;
};
```

```
/* M-All */
z[T] = (X, y) => {
  var y1 = {}; // if y1 != y
  var y2 = {}; // if y2 != y
  x1[T1](X, y1);
  x2[T2](X, y2);
  J;
};
```

```
/* M-Rcd */
var y = {};
var y1 = {}; // if y1 != y
var y2 = {}; // if y2 != y
Object.assign(y1, x1[T1]);
Object.assign(y2, x2[T2]);
J;
z[T] = y;
```