Compiling from F_i^+ to JavaScript

Yaozhu Sun

January 30, 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)*

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

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

J-Gen
$$\frac{\Gamma \vdash e \Leftrightarrow A \;\rightsquigarrow J \mid z^{-}}{\Gamma \vdash e \Leftrightarrow A \;\rightsquigarrow \mathsf{code} \mid z^{+}}$$

J-Top
$$\frac{}{\Gamma \vdash \{\} \Rightarrow \top \;\rightsquigarrow \varnothing \mid z^{-}}$$

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

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

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

J-Base
$$\frac{T = |\mathbb{B}|}{\Gamma \vdash b \Rightarrow \mathbb{B} \;\rightsquigarrow \mathsf{code} \mid z^{-}}$$

J-Var
$$\frac{x : A \in \Gamma}{\Gamma \vdash x \Rightarrow A \;\rightsquigarrow \mathsf{code} \mid z^{-}}$$

J-VarGen
$$\frac{x : A \in \Gamma}{\Gamma \vdash x \Rightarrow A \;\rightsquigarrow \mathsf{code} \mid z^{+}}$$

J-Fix
$$\frac{\Gamma, x : A \vdash e \Leftarrow A \;\rightsquigarrow J \mid z^{-}}{\Gamma \vdash \mathbf{fix}\, x : A.\ e \Rightarrow A \;\rightsquigarrow \mathsf{code} \mid z^{-}}$$

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

J-App
$$\frac{\begin{array}{c} \Gamma \vdash e_1 \Rightarrow A \;\rightsquigarrow J_1 \mid x^{+} \\ A \rhd B \to C \\ \Gamma \vdash e_2 \Leftarrow B \;\rightsquigarrow J_2 \mid y^{+} \\ x : A \bullet y_0 \;\rightsquigarrow J_3 \mid z \end{array}}{\Gamma \vdash e_1\, e_2 \Rightarrow C \;\rightsquigarrow \mathsf{code} \mid z^{-}}$$

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

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

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

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

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

J-Anno
$$\frac{\Gamma \vdash e \Leftarrow A \;\rightsquigarrow J \mid z^{-}}{\Gamma \vdash e : A \Rightarrow A \;\rightsquigarrow J \mid z^{-}}$$

J-AnnoGen
$$\frac{\Gamma \vdash e \Leftarrow A \;\rightsquigarrow J \mid z^{+}}{\Gamma \vdash e : A \Rightarrow A \;\rightsquigarrow J \mid z^{+}}$$

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

J-SubEq
$$\frac{\Gamma \vdash e \Rightarrow A \;\rightsquigarrow J \mid z^{-}}{\Gamma \vdash e \Leftarrow A \;\rightsquigarrow J \mid z^{-}}$$

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

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

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

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

/* J-VarGen */
var z = x.get;

/* J-Fix */
var x = { get: z };
J;
```

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

/* J-App */
J1;
var y0 = {
  get get() {
    J2;
    Object.defineProperty(
      this, "get",
      {value: y}
    );  return y;
  }
};  J3;
```

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

/* J-Rcd */
z[T] = {
  get get() {
    J;
    Object.defineProperty(
      this, "get",
      {value: y}
    );  return y;
  }
};
```

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

$$\textsc{A-Top}\quad \frac{\rceil A \lceil}{x : A \bullet \mathit{arg} \;\rightsquigarrow\; \varnothing \mid z}$$

$$\textsc{A-Arrow}\quad \frac{T = |\overrightarrow{B}|}{x : A \to B \bullet y \;\rightsquigarrow\; \mathsf{code} \mid z}$$

$$\textsc{A-All}\quad \frac{T = |B|^{\forall}}{x : \forall X * A.\, B \bullet \mathit{Ts} \;\rightsquigarrow\; \mathsf{code} \mid z}$$

$$\textsc{A-Rcd}\quad \frac{T = \{\ell : |A|\}}{x : \{\ell : A\} \bullet \ell \;\rightsquigarrow\; \mathsf{code} \mid z}$$

$$\textsc{A-And}\quad \frac{x : A \bullet \mathit{arg} \;\rightsquigarrow\; J_1 \mid z \qquad x : B \bullet \mathit{arg} \;\rightsquigarrow\; J_2 \mid z}{x : A \,\&\, B \bullet \mathit{arg} \;\rightsquigarrow\; J_1; J_2 \mid z}$$

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

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

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

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

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

$$\text{S-Base}\ \frac{T = |\mathbb{B}|}{x : \mathbb{B} <: y : \mathbb{B} \ \leadsto \mathsf{code}} \qquad \text{S-Var}\ \frac{}{x : X <: y : X \ \leadsto \mathsf{code}} \qquad \text{S-Arrow}\ \frac{\begin{array}{c} T_1 = \overrightarrow{|A_2|} \\ T_2 = \overrightarrow{|B_2|} \qquad B_2{}^\circ \\ x_1 : B_1 <: y_1 : A_1 \ \leadsto J_1 \\ x_2 : A_2 <: y_2 : B_2 \ \leadsto J_2 \end{array}}{x : A_1 \to A_2 <: y : B_1 \to B_2 \ \leadsto \mathsf{code}}$$

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

$$\text{S-AndL}\ \frac{C^\circ \qquad x : A <: y : C \ \leadsto J}{x : A \,\&\, B <: y : C \ \leadsto J} \qquad \text{S-AndR}\ \frac{C^\circ \qquad x : B <: y : C \ \leadsto J}{x : A \,\&\, B <: y : C \ \leadsto J} \qquad \text{S-Split}\ \frac{\begin{array}{c} B_1 \lhd B \rhd B_2 \\ y_1 : B_1 \ \rhd\ z : B \ \lhd\ y_2 : B_2 \ \leadsto J_3 \\ x : A <: y_1 : B_1 \ \leadsto J_1 \\ x : A <: y_2 : B_2 \ \leadsto J_2 \end{array}}{x : A <: z : B \ \leadsto \mathsf{code}}$$

```
/* 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] = (p, y2) => {
  var x2 = {};
  x[T1]({
      get get() {
        var x1 = p.get;
        var y1 = {};  J1;
        Object.defineProperty(
          this, "get",
          {value: y1}
        );  return y1;
      }
    }, x2);
    J2;
};

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

/* S-Rcd */
y[T2] = {
  get get() {
    var x0 = x[T1].get;
    var y0 = {};  J;
    Object.defineProperty(
      this, "get",
      {value: y0}
    );  return y0;
  }
}

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

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

$$\textsc{M-And}\quad \frac{}{z : A \ \triangleright\ z : A \,\&\, B \ \triangleleft\ z : B \ \rightsquigarrow\ \varnothing}$$

$$\textsc{M-Arrow}\quad \frac{\begin{array}{c} T = |\overrightarrow{B}| \\ T_1 = |\overrightarrow{B_1}| \qquad T_2 = |\overrightarrow{B_2}| \\ y_1 : B_1 \ \triangleright\ y : B \ \triangleleft\ y_2 : B_2 \ \rightsquigarrow\ J \end{array}}{x_1 : A \to B_1 \ \triangleright\ z : A \to B \ \triangleleft\ x_2 : A \to B_2 \ \rightsquigarrow\ \texttt{code}}$$

$$\textsc{M-All}\quad \frac{\begin{array}{c} T = |B|^{\forall} \\ T_1 = |B_1|^{\forall} \qquad T_2 = |B_2|^{\forall} \\ y_1 : B_1 \ \triangleright\ y : B \ \triangleleft\ y_2 : B_2 \ \rightsquigarrow\ J \end{array}}{x_1 : \forall X * A.\ B_1 \ \triangleright\ z : \forall X * A.\ B \ \triangleleft\ x_2 : \forall X * A.\ B_2 \ \rightsquigarrow\ \texttt{code}}$$

$$\textsc{M-Rcd}\quad \frac{\begin{array}{c} T = \{\ell : |A|\} \\ T_1 = \{\ell : |A_1|\} \\ T_2 = \{\ell : |A_2|\} \\ y_1 : A_1 \ \triangleright\ y : A \ \triangleleft\ y_2 : A_2 \ \rightsquigarrow\ J \end{array}}{x_1 : \{\ell : A_1\} \ \triangleright\ z : \{\ell : A\} \ \triangleleft\ x_2 : \{\ell : A_2\} \ \rightsquigarrow\ \texttt{code}}$$

```
/* 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 */
z[T] = {
  get get() {
    var y = {};
    var y1 = {}; // if y1 != y
    var y2 = {}; // if y2 != y
    Object.assign(y1, x1[T1].get);
    Object.assign(y2, x2[T2].get);
    J;
    Object.defineProperty(
      this, "get",
      {value: y}
    );  return y;
  }
};
```