

Compiling from F_i^+ to JavaScript

Yaozhu Sun

January 30, 2023

Syntax of F_i^+

Types	$A, B, C ::= \top \mid \perp \mid \mathbb{B} \mid X \mid A \rightarrow B \mid \forall X * A. B \mid \{\ell : A\} \mid A \& B$
Type indices	$T ::= \mathbb{B} \mid \vec{T} \mid T^\forall \mid \{\ell : T\} \mid T_1 \& T_2$
Expressions	$e ::= \{\} \mid b \mid x \mid \mathbf{fix} \ x : A. e \mid \lambda x : A. e : B \mid e_1 \ e_2 \mid \Lambda X * A. e : B \mid e \ A \mid \{\ell = e\} \mid e.\ell$ $\mid e_1 \ , \ e_2 \mid e : A$
Values	$v ::= \{\} \mid b \mid \lambda x : A. e : B \mid \Lambda X * A. e : B \mid \{\ell = v\} \mid v_1 \ , \ v_2$

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

$$|\mathbb{B}| = \mathbb{B} \quad |X| = \mathbf{atoi}(X) \quad |A \rightarrow B| = |\vec{B}| \quad |\forall X * A. B| = |B|^\forall \quad |\{\ell : A\}| = \{\ell : |A|\}$$

$$\frac{A_{k_1} < A_{k_2} < \dots < A_{k_m} \quad \neg |A_k|}{|A_1 \& A_2 \& \dots \& A_n| = |A_{k_1}| \& |A_{k_2}| \& \dots \& |A_{k_m}|}$$

$$\boxed{|A|} \quad (Top-like\ types)$$

TL-TOP	TL-AND	TL-ARROW	TL-ALL	TL-RCD
$\frac{}{ \top }$	$\frac{ A \quad B }{ A \& B }$	$\frac{ B }{ A \rightarrow B }$	$\frac{ B }{ \forall X * A. B }$	$\frac{ A }{ \{\ell : A\} }$

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

O-TOP	O-BOT	O-BASE	O-VAR	O-ARROW	O-ALL	O-RCD
$\overline{\top^\circ}$	$\overline{\perp^\circ}$	$\overline{\mathbb{B}^\circ}$	$\overline{X^\circ}$	$\overline{B^\circ}$	$\overline{B^\circ}$	$\overline{A^\circ}$
				$\overline{(A \rightarrow B)^\circ}$	$\overline{(\forall X * A. B)^\circ}$	$\overline{\{\ell : A\}^\circ}$

$$\boxed{\Gamma \vdash e \Leftrightarrow A \rightsquigarrow J \mid z^\pm}$$

(Type-directed compilation)

<p>J-GEN</p> $\frac{\Gamma \vdash e \Leftrightarrow A \rightsquigarrow J \mid z^-}{\Gamma \vdash e \Leftrightarrow A \rightsquigarrow \text{code} \mid z^+}$	<p>J-TOP</p> $\frac{}{\Gamma \vdash \{\} \Rightarrow \top \rightsquigarrow \emptyset \mid z^-}$	<p>J-TOPABS</p> $\frac{\lceil B \rceil}{\Gamma \vdash \lambda x:A. e:B \Rightarrow A \rightarrow B \rightsquigarrow \emptyset \mid z^-}$	
<p>J-TOPTABS</p> $\frac{\lceil B \rceil}{\Gamma \vdash \Lambda X * A. e:B \Rightarrow \forall X * A. B \rightsquigarrow \emptyset \mid z^-}$	<p>J-TOPRCD</p> $\frac{\Gamma \vdash e \Rightarrow A \quad \lceil A \rceil}{\Gamma \vdash \{\ell = e\} \Rightarrow \{\ell : A\} \rightsquigarrow \emptyset \mid z^-}$		
<p>J-BASE</p> $\frac{T = \mathbb{B} }{\Gamma \vdash b \Rightarrow \mathbb{B} \rightsquigarrow \text{code} \mid z^-}$	<p>J-VAR</p> $\frac{x : A \in \Gamma}{\Gamma \vdash x \Rightarrow A \rightsquigarrow \text{code} \mid z^-}$	<p>J-VARGEN</p> $\frac{x : A \in \Gamma}{\Gamma \vdash x \Rightarrow A \rightsquigarrow \text{code} \mid z^+}$	
<p>J-FIX</p> $\frac{\Gamma, x : A \vdash e \Leftarrow A \rightsquigarrow J \mid z^-}{\Gamma \vdash \mathbf{fix} x:A. e \Rightarrow A \rightsquigarrow \text{code} \mid z^-}$			<p>J-ABS</p> $\frac{T = \overrightarrow{ B }}{\Gamma, x : A \vdash e \Leftarrow B \rightsquigarrow J \mid y^-}$ $\frac{}{\Gamma \vdash \lambda x:A. e:B \Rightarrow A \rightarrow B \rightsquigarrow \text{code} \mid z^-}$
<p>J-APP</p> $\frac{\Gamma \vdash e_1 \Rightarrow A \rightsquigarrow J_1 \mid x^+ \quad A \triangleright B \rightarrow C \quad \Gamma \vdash e_2 \Leftarrow B \rightsquigarrow J_2 \mid y^+ \quad x : A \bullet y_0 \rightsquigarrow J_3 \mid z}{\Gamma \vdash e_1 e_2 \Rightarrow C \rightsquigarrow \text{code} \mid z^-}$	<p>J-TABS</p> $\frac{T = B ^\forall \quad \Gamma, X * A \vdash e \Leftarrow B \rightsquigarrow J \mid y^-}{\Gamma \vdash \Lambda X * A. e:B \Rightarrow \forall X * A. B \rightsquigarrow \text{code} \mid z^-}$		
<p>J-TAPP</p> $\frac{\Gamma \vdash e \Rightarrow B \rightsquigarrow J_1 \mid y^+ \quad B \triangleright \forall X * C_1. C_2 \quad \Gamma \vdash A * C_1 \quad Ts = \mathbf{itoa} \mid A \mid \quad y : B \bullet Ts \rightsquigarrow J_2 \mid z}{\Gamma \vdash e A \Rightarrow C_2[X \mapsto A] \rightsquigarrow J_1; J_2 \mid z^-}$	<p>J-RCD</p> $\frac{T = \{\ell : A \} \quad \Gamma \vdash e \Rightarrow A \rightsquigarrow J \mid y^+}{\Gamma \vdash \{\ell = e\} \Rightarrow \{\ell : A\} \rightsquigarrow \text{code} \mid z^-}$		
<p>J-PROJ</p> $\frac{\Gamma \vdash e \Rightarrow A \rightsquigarrow J_1 \mid y^+ \quad A \triangleright \{\ell : B\} \quad y : A \bullet \ell \rightsquigarrow J_2 \mid z}{\Gamma \vdash e.\ell \Rightarrow B \rightsquigarrow J_1; J_2 \mid z^-}$	<p>J-MERGE</p> $\frac{\Gamma \vdash e_1 \Rightarrow A \rightsquigarrow J_1 \mid z^- \quad \Gamma \vdash e_2 \Rightarrow B \rightsquigarrow J_2 \mid z^-}{\Gamma \vdash A * B}$ $\frac{}{\Gamma \vdash e_1, e_2 \Rightarrow A \& B \rightsquigarrow J_1; J_2 \mid z^-}$		
<p>J-ANNO</p> $\frac{}{\Gamma \vdash e \Leftarrow A \rightsquigarrow J \mid z^-}$	<p>J-ANNO</p> $\frac{}{\Gamma \vdash e : A \Rightarrow A \rightsquigarrow J \mid z^-}$		
<p>J-ANNOGEN</p> $\frac{\Gamma \vdash e \Leftarrow A \rightsquigarrow J \mid z^+}{\Gamma \vdash e : A \Rightarrow A \rightsquigarrow J \mid z^+}$	<p>J-SUB</p> $\frac{\Gamma \vdash e \Rightarrow A \rightsquigarrow J_1 \mid x^+ \quad x : A <: y : B \rightsquigarrow J_2}{\Gamma \vdash e \Leftarrow B \rightsquigarrow J_1; J_2 \mid y^-}$	<p>J-SUBEQGEN</p> $\frac{\Gamma \vdash e \Rightarrow A \rightsquigarrow J \mid z^+}{\Gamma \vdash e \Leftarrow A \rightsquigarrow J \mid z^+}$	

```

/* 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;
  }
};

```

$$\boxed{x : A \bullet \text{arg} \rightsquigarrow J \mid z}$$

(Distributive application)

$$\frac{\text{A-TOP} \quad \boxed{\mid A \mid}}{x : A \bullet \text{arg} \rightsquigarrow \emptyset \mid z}$$

$$\frac{\text{A-ARROW} \quad T = \overrightarrow{\mid B \mid}}{x : A \rightarrow B \bullet y \rightsquigarrow \text{code} \mid z}$$

$$\frac{\text{A-ALL} \quad T = \mid B \mid^{\forall}}{x : \forall X * A. B \bullet Ts \rightsquigarrow \text{code} \mid z}$$

$$\frac{\text{A-RCD} \quad T = \{\ell : \mid A \mid\}}{x : \{\ell : A\} \bullet \ell \rightsquigarrow \text{code} \mid z}$$

$$\frac{\text{A-AND} \quad \begin{array}{l} x : A \bullet \text{arg} \rightsquigarrow J_1 \mid z \\ x : B \bullet \text{arg} \rightsquigarrow J_2 \mid z \end{array}}{x : A \& B \bullet \text{arg} \rightsquigarrow J_1; J_2 \mid z}$$

```

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

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

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

```

$$\boxed{x : A <: y : B \rightsquigarrow J}$$

(Coercive subtyping)

$$\begin{array}{c}
\text{S-TOP} \\
\frac{B^\circ \quad \lceil B \rceil}{x : A <: y : B \rightsquigarrow \emptyset} \\
\\
\text{S-BOT} \\
\frac{T = |A| \quad A^\circ}{x : \perp <: y : A \rightsquigarrow \text{code}} \\
\\
\text{S-EQ} \\
\frac{}{x : A <: y : A \rightsquigarrow \text{code}} \\
\\
\text{S-BASE} \\
\frac{T = |\mathbb{B}|}{x : \mathbb{B} <: y : \mathbb{B} \rightsquigarrow \text{code}} \\
\\
\text{S-VAR} \\
\frac{}{x : X <: y : X \rightsquigarrow \text{code}} \\
\\
\text{S-ARROW} \\
\frac{\begin{array}{c} T_1 = \overrightarrow{|A_2|} \\ T_2 = \overrightarrow{|B_2|} \quad B_2^\circ \\ x_1 : B_1 <: y_1 : A_1 \rightsquigarrow J_1 \\ x_2 : A_2 <: y_2 : B_2 \rightsquigarrow J_2 \end{array}}{x : A_1 \rightarrow A_2 <: y : B_1 \rightarrow B_2 \rightsquigarrow \text{code}} \\
\\
\text{S-ALL} \\
\frac{\begin{array}{c} T_1 = |A_2|^\forall \quad T_2 = |B_2|^\forall \\ B_2^\circ \quad B_1 <: A_1 \\ x_0 : A_2 <: y_0 : B_2 \rightsquigarrow J \end{array}}{x : \forall X * A_1. A_2 <: y : \forall X * B_1. B_2 \rightsquigarrow \text{code}} \\
\\
\text{S-RCD} \\
\frac{\begin{array}{c} T_1 = \{\ell : |A|\} \\ T_2 = \{\ell : |B|\} \\ B^\circ \quad x_0 : A <: y_0 : B \rightsquigarrow J \end{array}}{x : \{\ell : A\} <: y : \{\ell : B\} \rightsquigarrow \text{code}} \\
\\
\text{S-SPLIT} \\
\frac{\begin{array}{c} B_1 \triangleleft B \triangleright B_2 \\ y_1 : B_1 \triangleright z : B \triangleleft y_2 : B_2 \rightsquigarrow J_3 \\ x : A <: y_1 : B_1 \rightsquigarrow J_1 \\ x : A <: y_2 : B_2 \rightsquigarrow J_2 \end{array}}{x : A <: z : B \rightsquigarrow \text{code}} \\
\\
\text{S-ANDL} \\
\frac{C^\circ \quad x : A <: y : C \rightsquigarrow J}{x : A \& B <: y : C \rightsquigarrow J} \\
\\
\text{S-ANDR} \\
\frac{C^\circ \quad x : B <: y : C \rightsquigarrow J}{x : A \& B <: y : C \rightsquigarrow J}
\end{array}$$

```

/* 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)

M-ARROW

$$\frac{\text{M-AND} \quad \overline{z : A \triangleright z : A \& B \triangleleft z : B \rightsquigarrow \emptyset}}{\quad} \quad \frac{\begin{array}{c} T = \overrightarrow{|B|} \\ T_1 = \overrightarrow{|B_1|} \quad T_2 = \overrightarrow{|B_2|} \\ y_1 : B_1 \triangleright y : B \triangleleft y_2 : B_2 \rightsquigarrow J \end{array}}{x_1 : A \rightarrow B_1 \triangleright z : A \rightarrow B \triangleleft x_2 : A \rightarrow B_2 \rightsquigarrow \text{code}}$$

M-ALL

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

M-RCD

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

```
/* 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;
  }
};
```