

Compiling from F_i^+ to JavaScript

Yaozhu Sun

April 5, 2023

Syntax of F_i^+

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

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

| | | |
|--|--|--|
| J-GEN $\frac{\Gamma \vdash e \Leftrightarrow A \rightsquigarrow J \mid z^-}{\Gamma \vdash e \Leftrightarrow A \rightsquigarrow \mathbf{code} \mid z^+}$ | J-TOP $\frac{}{\Gamma \vdash \{\} \Rightarrow \top \rightsquigarrow \emptyset \mid z^-}$ | J-TOPABS $\frac{}{\Gamma \vdash \lambda x : A. e : B \Rightarrow A \rightarrow B \rightsquigarrow \emptyset \mid z^-}$ |
| J-TOPTABS $\frac{}{\Gamma \vdash \Lambda X * A. e : B \Rightarrow \forall X * A. B \rightsquigarrow \emptyset \mid z^-}$ | J-TOPRCD $\frac{\Gamma \vdash e \Rightarrow A \quad \lceil A \rceil}{\Gamma \vdash \{\ell = e\} \Rightarrow \{\ell : A\} \rightsquigarrow \emptyset \mid z^-}$ | |
| J-INT $\frac{T = \mathbb{Z} }{\Gamma \vdash n \Rightarrow \mathbb{Z} \rightsquigarrow \mathbf{code} \mid z^-}$ | J-VAR $\frac{x : A \in \Gamma}{\Gamma \vdash x \Rightarrow A \rightsquigarrow \mathbf{code} \mid z^-}$ | J-VARGEN $\frac{x : A \in \Gamma}{\Gamma \vdash x \Rightarrow A \rightsquigarrow \mathbf{code} \mid z^+}$ |
| J-FIX $\frac{\Gamma, x : A \vdash e \Leftarrow A \rightsquigarrow J \mid z^-}{\Gamma \vdash \mathbf{fix} \, x : A. e \Rightarrow A \rightsquigarrow \mathbf{code} \mid z^-}$ | J-ABS $\frac{T = \vec{B} \quad \Gamma, x : A \vdash e \Leftarrow B \rightsquigarrow J \mid y^-}{\Gamma \vdash \lambda x : A. e : B \Rightarrow A \rightarrow B \rightsquigarrow \mathbf{code} \mid z^-}$ | |
| J-APP $\frac{\Gamma \vdash e_1 \Rightarrow A \rightsquigarrow J_1 \mid x^+ \quad \Gamma \vdash e_2 \Rightarrow B \rightsquigarrow J_2 \mid y^+ \quad \Gamma \vdash x : A \bullet y_0 : B \rightsquigarrow J_3 \mid z : C}{\Gamma \vdash e_1 \, e_2 \Rightarrow C \rightsquigarrow \mathbf{code} \mid z^-}$ | J-TABS $\frac{T = B ^\forall \quad \Gamma, X * A \vdash e \Leftarrow B \rightsquigarrow J \mid y^-}{\Gamma \vdash \Lambda X * A. e : B \Rightarrow \forall X * A. B \rightsquigarrow \mathbf{code} \mid z^-}$ | |

J-TAPP

$$\frac{\Gamma \vdash e \Rightarrow B \rightsquigarrow J_1 \mid y^+ \quad \Gamma \vdash y : B \bullet A \rightsquigarrow J_2 \mid z : C}{\Gamma \vdash e A \Rightarrow C \rightsquigarrow J_1; J_2 \mid z^-}$$

J-RCD

$$\frac{T = \{\ell : |A|\} \quad \Gamma \vdash e \Rightarrow A \rightsquigarrow J \mid y^+}{\Gamma \vdash \{\ell = e\} \Rightarrow \{\ell : A\} \rightsquigarrow \text{code} \mid z^-}$$

J-PROJ

$$\frac{\Gamma \vdash e \Rightarrow A \rightsquigarrow J_1 \mid y^+ \quad y : A \bullet \{\ell\} \rightsquigarrow J_2 \mid z : B}{\Gamma \vdash e.\ell \Rightarrow B \rightsquigarrow J_1; J_2 \mid z^-}$$

J-MERGE

$$\frac{\Gamma \vdash e_1 \Rightarrow A \rightsquigarrow J_1 \mid z^- \quad \Gamma \vdash e_2 \Rightarrow B \rightsquigarrow J_2 \mid z^- \quad \Gamma \vdash A * B}{\Gamma \vdash e_1, e_2 \Rightarrow A \& B \rightsquigarrow J_1; J_2 \mid z^-}$$

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

$$\frac{\Gamma \vdash e \Rightarrow A \rightsquigarrow J_1 \mid x^+ \quad x : A <:^+ y : B \rightsquigarrow J_2}{\Gamma \vdash e \Leftarrow B \rightsquigarrow J_1; J_2 \mid y^-}$$

J-SUBEQGEN

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

J-DEF

$$\frac{\Gamma \vdash e_1 \Rightarrow A \rightsquigarrow J_1 \mid y^+ \quad \Gamma, x : A \vdash e_2 \Rightarrow B \rightsquigarrow J_2 \mid z^-}{\Gamma \vdash x = e_1; e_2 \Rightarrow B \rightsquigarrow \text{code} \mid z^-}$$

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

```
/* J-Int */
z[T] = n;
```

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

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

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

```
/* J-App */
```

```
J1;
var y0 = {
  get get() {
    J2;
    delete this.get;
    return this.get = y;
  }
}; J3;
```

```
/* J-Abs */
```

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

```
/* J-TAbs */
```

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

```
/* J-Rcd */
```

```
z[T] = {
  get get() {
    J;
    delete this.get;
    return this.get = y;
  }
};
```

```
/* J-Def */
```

```
export var x = {
  get get() {
    J1;
    delete this.get;
    return this.get = y;
  }
}; J2;
```

$$\boxed{\Gamma \vdash x : A \bullet p \rightsquigarrow J \mid z : B}$$

(Distributive application)

$$\frac{\text{A-Top} \quad \lceil A \rceil}{\Gamma \vdash x : A \bullet p \rightsquigarrow \emptyset \mid z : \top}$$

$$\frac{\text{A-ARROW} \quad T = \overrightarrow{|B|} \quad y : C <^+ y_0 : A \rightsquigarrow J}{\Gamma \vdash x : A \rightarrow B \bullet y : C \rightsquigarrow \text{code} \mid z : B}$$

$$\frac{\text{A-ALL} \quad \Gamma \vdash A * C \quad T = |B|^\forall \quad Ts = \mathbf{itoa} | C |}{\Gamma \vdash x : \forall X * A. B \bullet C \rightsquigarrow \text{code} \mid z : B[X \mapsto C]}$$

$$\frac{\text{A-AND} \quad \Gamma \vdash x : A \bullet p \rightsquigarrow J_1 \mid z : A' \quad \Gamma \vdash x : B \bullet p \rightsquigarrow J_2 \mid z : B'}{\Gamma \vdash x : A \& B \bullet p \rightsquigarrow J_1; J_2 \mid z : A' \& B'}$$

```
/* A-Arrow */
var y0 = {}; J;
x[T](y0, z);
```

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

$$\boxed{x : A \bullet \{\ell\} \rightsquigarrow J \mid z : B}$$

(Distributive projection)

$$\frac{\text{P-Top} \quad \lceil A \rceil}{x : A \bullet \{\ell\} \rightsquigarrow \emptyset \mid z : \top}$$

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

$$\frac{\text{P-RcdNeq} \quad \ell_1 \neq \ell_2 \quad T = \{\ell : |A|\}}{x : \{\ell_1 : A\} \bullet \{\ell_2\} \rightsquigarrow \emptyset \mid z : \top}$$

$$\frac{\text{P-AND} \quad x : A \bullet \{\ell\} \rightsquigarrow J_1 \mid z : A' \quad x : B \bullet \{\ell\} \rightsquigarrow J_2 \mid z : B'}{x : A \& B \bullet \{\ell\} \rightsquigarrow J_1; J_2 \mid z : A' \& B'}$$

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

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

(Coercive subtyping)

$$\begin{array}{c}
\text{S-Top} \quad \frac{B^{\circ} \quad]B[}{x : A <:^{\pm} y : B \rightsquigarrow \emptyset} \quad \text{S-Bot} \quad \frac{T = |A| \quad A^{\circ}}{x : \perp <:^{\pm} y : A \rightsquigarrow \text{code}} \quad \text{S-Eq} \quad \frac{}{x : A <:^+ y : A \rightsquigarrow \text{code}} \\
\\
\text{S-Int} \quad \frac{T = |\mathbb{Z}|}{x : \mathbb{Z} <:^{\pm} y : \mathbb{Z} \rightsquigarrow \text{code}} \quad \text{S-Var} \quad \frac{}{x : X <:^{\pm} y : X \rightsquigarrow \text{code}} \quad \text{S-Arrow} \quad \frac{\begin{array}{c} T_1 = \overrightarrow{|A_2|} \\ T_2 = \overrightarrow{|B_2|} \quad B_2^{\circ} \\ x_1 : B_1 <:^+ y_1 : A_1 \rightsquigarrow J_1 \\ x_2 : A_2 <:^+ y_2 : B_2 \rightsquigarrow J_2 \end{array}}{x : A_1 \rightarrow A_2 <:^{\pm} y : B_1 \rightarrow B_2 \rightsquigarrow \text{code}} \\
\\
\text{S-All} \quad \frac{\begin{array}{c} T_1 = |A_2|^{\vee} \quad T_2 = |B_2|^{\vee} \\ B_2^{\circ} \quad B_1 <: A_1 \\ x_0 : A_2 <:^+ y_0 : B_2 \rightsquigarrow J \end{array}}{x : \forall X * A_1. A_2 <:^{\pm} y : \forall X * B_1. B_2 \rightsquigarrow \text{code}} \quad \text{S-Rcd} \quad \frac{\begin{array}{c} T_1 = \{\ell : |A|\} \\ T_2 = \{\ell : |B|\} \\ B^{\circ} \quad x_0 : A <:^+ y_0 : B \rightsquigarrow J \end{array}}{x : \{\ell : A\} <:^{\pm} y : \{\ell : B\} \rightsquigarrow \text{code}} \\
\\
\text{S-AndL} \quad \frac{C^{\circ} \quad x : A <:^- y : C \rightsquigarrow J}{x : A \& B <:^{\pm} y : C \rightsquigarrow J} \quad \text{S-AndR} \quad \frac{C^{\circ} \quad x : B <:^- y : C \rightsquigarrow J}{x : A \& B <:^{\pm} y : C \rightsquigarrow J} \quad \text{S-Split} \quad \frac{\begin{array}{c} B_1 \triangleleft B \triangleright B_2 \\ y_1 : B_1 \triangleright z : B \triangleleft y_2 : B_2 \rightsquigarrow J_3 \\ x : A <:^{\pm} y_1 : B_1 \rightsquigarrow J_1 \\ x : A <:^{\pm} y_2 : B_2 \rightsquigarrow J_2 \end{array}}{x : A <:^{\pm} z : B \rightsquigarrow \text{code}}
\end{array}$$

```

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

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

/* S-Int */
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;
    delete this.get;
    return this.get = 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;
    delete this.get;
    return this.get = 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

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

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;
    delete this.get
    return this.get = y;
  }
};
```