Elaborating F_i^+ to record calculus λ^{rec}

Snow

November 30, 2022

1 Notes

Properties (to prove) 0) Type-safety in the target calculus 1) Any well-typed source expression is translated into a well-typed target expression 2) For type index generated from (source) types, define an equivalence relations on types (`A =B`). `|A|=|B| iff A≅B`

Questions 0) Why $\top$ and $\perp$ corresponds to no type index? 1) Duplicated labels in records are unsafe in the target calculus. How to compile $1 : Int\&Int$?

Changes in λ^{rec} 0) Simplify the system, drop polymorphism, and use simple labels

2 Syntax of $\lambda^{\mathbf{rec}}$

Target expressions	$t ::= \ell \mid b \mid x \mid \mathbf{fix}\, x.t \mid \lambda x.t \mid t_1\; t_2 \mid \{t_1 \Rightarrow t'_1, .., t_n \Rightarrow t'_n\} \mid t_1.t_2 \mid t_1; t_2$
Target values	$tv, tu ::= b \mid \lambda x.t \mid \{\ell_1 \Rightarrow tv_1, .., \ell_n \Rightarrow tv_n\}$

Notes 0) Multi-field records are nullable. $\lambda\_.t$ denotes $\lambda x.t$ where x does not appear in t.
~~1) We need functions that take a type index for big lambda expressions and type applications.~~
2) The whole program has a record form. 3) l stands for a string.

E	$::=$		evaluation context
	$\mid$	$[-]$	
	$\mid$	$E.t$	projection
	$\mid$	$tv.E$	projection
	$\mid$	$E\; t$	
	$\mid$	$tv\; E$	
	$\mid$	$E; t$	concatenation
	$\mid$	$tv; E$	concatenation
	$\mid$	$\{\ell_1 \Rightarrow tv_1, .., \ell_m \Rightarrow tv_m, E \Rightarrow t, t_1 \Rightarrow t'_1, .., t_n \Rightarrow t'_n\}$	eval a label
	$\mid$	$\{\ell_1 \Rightarrow tv_1, .., \ell_m \Rightarrow tv_m, \ell \Rightarrow E, t_1 \Rightarrow t'_1, .., t_n \Rightarrow t'_n\}$	eval a field

$\boxed{t \to t'}$ *(Record Calculus Structrual Reduction)*

TS-CTX
$$\frac{t \to t'}{E[t] \to E[t']}$$

TS-PROJ
$$\frac{\{\ell_1, .., \ell_m\} \cap \{\ell\} = \varnothing \qquad \{\ell'_1, .., \ell'_n\} \cap \{\ell\} = \varnothing}{\{\ell_1 \Rightarrow tv_1, .., \ell_m \Rightarrow tv_m, \ell \Rightarrow tv, \ell'_1 \Rightarrow tv'_1, .., \ell'_n \Rightarrow tv'_n\}.\ell \to tv}$$

TS-CONCAT
$$\frac{\{\ell_1, .., \ell_m\} \cap \{\ell'_1, .., \ell'_n\} = \varnothing}{\{\ell_1 \Rightarrow tv_1, .., \ell_m \Rightarrow tv_m\}; \{\ell'_1 \Rightarrow t'_1, .., \ell'_n \Rightarrow t'_n\} \to \{\ell_1 \Rightarrow tv_1, .., \ell_m \Rightarrow tv_m, \ell'_1 \Rightarrow t'_1, .., \ell'_n \Rightarrow t'_n\}}$$

TS-BETA
$$\frac{}{\lambda x.t\; tv \to t[x \mapsto tv]}$$

TS-FIX
$$\frac{}{\mathbf{fix}\, x.t \to t[x \mapsto \mathbf{fix}\, x.t]}$$

Problems ~~0) To achieve a safe language, we need to check the projected label is unique in TS-proj.~~
1) The side-condition in rule TS-concat does not guarantee the original types are disjoint. E.g. $\{\mathsf{Int} \Rightarrow b_1, \mathsf{Bool} \Rightarrow b_2\}; \{\mathsf{Int} \,\&\, \mathsf{Bool} \Rightarrow b_3\}$
2) Regarding efficiency, the time cost of concatenation is high.

3 Syntax of F_i^+

Types	$A, B, C ::= \top \mid \bot \mid \mathbb{B} \mid X \mid A \to B \mid \forall X{*}A.\ B \mid \{\ell : A\} \mid A \,\&\, B$
Type indices	$T ::= \mathbb{B} \mid \overrightarrow{T} \mid T^{\forall} \mid \{\ell : T\} \mid T_1 \,\&\, T_2$
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	$v ::= \{\,\} \mid b \mid \lambda x{:}A.\ e{:}B \mid \Lambda X{*}A.\ e{:}B \mid \{\ell = v\} \mid v_1 \,,,\, v_2$

3.1 Mapping between source types and target values

$A \,\&\, B$	$\{\lvert A\rvert \Rightarrow t_1, \lvert B\rvert \Rightarrow t_2\}$
$\top$	$\{\,\}$
$\bot$	
$\mathbb{B}$	$\{\mathbb{B} \Rightarrow b\}$
X	$\{X \Rightarrow t\}$ (not a value)
$A \to B$	$\{\overrightarrow{\lvert B\rvert} \Rightarrow \lambda x.t\}$
$\forall X{*}A.\ B$	$\{\lvert B\rvert^{\forall} \Rightarrow \lambda X.t\}$
$\{\ell : A\}$	$\{\{\ell : \lvert A\rvert\} \Rightarrow \lambda\,\_.t\}$

3.2 Elaboration rules

1) All top-like values are translated into empty lists.

$$\textsc{Ela-Top}\quad \frac{}{\Gamma \vdash \mathbf{top} \ \Rightarrow\ \top \rightsquigarrow \{\,\}}$$

$$\textsc{Ela-TopAbs}\quad \frac{\rceil B \lceil}{\Gamma \vdash \lambda x{:}A.\ e{:}B \ \Rightarrow\ A \to B \rightsquigarrow \{\,\}}$$

$$\textsc{Ela-TopTAbs}\quad \frac{\rceil B \lceil}{\Gamma \vdash \Lambda X{*}A.\ e{:}B \ \Rightarrow\ \forall X{*}A.\ B \rightsquigarrow \{\,\}}$$

$$\textsc{Ela-TopRcd}\quad \frac{\Gamma \vdash e \ \Rightarrow\ A \rightsquigarrow t \qquad \rceil A \lceil}{\Gamma \vdash \{\ell = e\} \ \Rightarrow\ \{\ell : A\} \rightsquigarrow \{\,\}}$$

2) All basic values (that are not merges) are translated into single-field records.

$$\textsc{Ela-Base}\quad \frac{}{\Gamma \vdash b \ \Rightarrow\ \mathbb{B} \rightsquigarrow \{\lvert\mathbb{B}\rvert \Rightarrow b\}}$$

$$\textsc{Ela-Abs}\quad \frac{\Gamma, x : A \vdash e \ \Leftarrow\ B \rightsquigarrow t}{\Gamma \vdash \lambda x{:}A.\ e{:}B \ \Rightarrow\ A \to B \rightsquigarrow \{\overrightarrow{\lvert B\rvert} \Rightarrow \lambda x.t\}}$$

$$\textsc{Ela-TAbs}\quad \frac{\Gamma, X * A \vdash e \ \Leftarrow\ B \rightsquigarrow t}{\Gamma \vdash \Lambda X{*}A.\ e{:}B \ \Rightarrow\ \forall X{*}A.\ B \rightsquigarrow \{\lvert B\rvert^{\forall} \Rightarrow \lambda X.t\}}$$

$$\textsc{Ela-Rcd}\quad \frac{\Gamma \vdash e \ \Rightarrow\ A \rightsquigarrow t}{\Gamma \vdash \{\ell = e\} \ \Rightarrow\ \{\ell : A\} \rightsquigarrow \{\{\ell : \lvert A\rvert\} \Rightarrow \lambda\,\_.t\}}$$

3) Variables are directly used (unwrap the thunk) and merged terms are simply concatenated, with the expectation that $\cdot \vdash e \ \Rightarrow\ A \rightsquigarrow t$ implies $t \to^{*} \{\lvert A\rvert \Rightarrow t'\}$ (or a multi-field record) (i.e. the variable will be substituted by a term of this exact type).

$$\textsc{Ela-Var}\quad \frac{x : A \in \Gamma}{\Gamma \vdash x \Rightarrow A \leadsto x\,\{\,\}} \qquad \textsc{Ela-Merge}\quad \frac{\Gamma \vdash e_1 \Rightarrow A \leadsto t_1 \qquad \Gamma \vdash e_2 \Rightarrow B \leadsto t_2 \qquad \Gamma \vdash A * B}{\Gamma \vdash e_1 \,,,\, e_2 \Rightarrow A \,\&\, B \leadsto t_1 ; t_2}$$

4) Projection, application, and type application make use of the property that every translated subterms are records. Application is lazy.

$$\textsc{Ela-Proj}\quad \frac{\Gamma \vdash e \Rightarrow A \leadsto t_1 \qquad A \rhd \{\ell : B\} \qquad t_1 : A ; \{\,\} \leadsto t_2}{\Gamma \vdash e.\ell \Rightarrow B \leadsto t_1}$$

$$\textsc{Ela-App}\quad \frac{\Gamma \vdash e_1 \Rightarrow A \leadsto t_1 \qquad A \rhd B \to C \qquad \Gamma \vdash e_2 \Leftarrow B \leadsto t_2 \qquad t_1 : A ; t_2 \leadsto t_3}{\Gamma \vdash e_1\, e_2 \Rightarrow C \leadsto t_3}$$

$$\textsc{Ela-TApp}\quad \frac{\Gamma \vdash e \Rightarrow B \leadsto t_1 \qquad B \rhd \forall X * C_1.\, C_2 \qquad \Gamma \vdash A * C_1 \qquad t_1 : B ; \mathbf{itoa}(A) \leadsto t_3}{\Gamma \vdash e\, A \Rightarrow C_2[X \mapsto A] \leadsto t_3}$$

5) Annotated expressions, like fixpoints, rely on the subsumption rule to insert coercions. The coerced expression always evaluates to a record.

$$\textsc{Ela-Anno}\quad \frac{\Gamma \vdash e \Leftarrow A \leadsto t}{\Gamma \vdash e : A \Rightarrow A \leadsto t} \qquad \textsc{Ela-Fix}\quad \frac{\Gamma, x : A \vdash e \Leftarrow A \leadsto t}{\Gamma \vdash \mathbf{fix}\, x{:}A.\; e \Rightarrow A \leadsto \mathbf{fix}\, x.t} \qquad \textsc{Ela-Sub}\quad \frac{\Gamma \vdash e \Rightarrow A \leadsto t_1 \qquad t_1 : A <: B \leadsto t_2}{\Gamma \vdash e \Leftarrow B \leadsto t_2}$$

3.3 Coercions

$$x : \mathsf{Int} \,\&\, \mathsf{Bool} <: \mathsf{Int} \leadsto \{\mathsf{Int} \Rightarrow x.\mathsf{Int}\}$$
$$x : \mathsf{Int} \,\&\, \mathsf{Bool} <: \mathsf{Bool} \,\&\, \mathsf{Int} \leadsto \{\mathsf{Bool} \Rightarrow x.\mathsf{Bool}\} ; \{\mathsf{Int} \Rightarrow x.\mathsf{Int}\}$$
$$x : \mathbb{B} \to \mathsf{Int} \,\&\, \mathsf{Bool} <: \mathbb{B} \to \mathsf{Int} \,\&\, \mathsf{Bool} \leadsto \{\overrightarrow{(\mathsf{Int} \,\&\, \mathsf{Bool})} \Rightarrow \lambda y.(x.\overrightarrow{\mathsf{Int}})\; y ; (x.\overrightarrow{\mathsf{Bool}})\; y\}$$

Problems 1) Derivation of subtyping is not unique. No type well-formedness, no coherence.
2) The algorithm is against the effort we put in typing application. As types are split whenever possible, functions are duplicated (and merged afterwards).
~~3) It is strange that the domain of arrow types has no effect to coercion generation.~~

$\boxed{t_1 : A <: B \leadsto t_2}$ *(Coercive subtyping)*

$$\textsc{S-Top}\quad \frac{B^\circ \qquad \rceil B \lceil}{t : A <: B \leadsto \{\,\}} \qquad \textsc{S-Base}\quad \frac{}{t : \mathbb{B} <: \mathbb{B} \leadsto \{\mathbb{B} \Rightarrow t.\mathbb{B}\}}$$

$$\textsc{S-Arrow}\quad \frac{{B_2}^\circ \qquad x\,\{\,\} : B_1 <: A_1 \leadsto t_1 \qquad (t.\overrightarrow{|A_2|})\,(\lambda\_.t_1) : A_2 <: B_2 \leadsto t_2}{t : A_1 \to A_2 <: B_1 \to B_2 \leadsto \{\overrightarrow{|B_2|} \Rightarrow \lambda x.t_2\}}$$

$$\textsc{S-All}\quad \frac{{B_2}^\circ \qquad B_1 <: A_1 \qquad (t.|A_2|^\forall)\, X : A_2 <: B_2 \leadsto t_2}{t : \forall X * A_1.\, A_2 <: \forall X * B_1.\, B_2 \leadsto \{|B_2|^\forall \Rightarrow \lambda X.t_2\}}$$

$$\textsc{S-Rcd}\quad \frac{B^\circ \qquad t.\{\ell : |A|\}\,\{\,\} : A <: B \leadsto t_2}{t : \{\ell : A\} <: \{\ell : B\} \leadsto \{\{\ell : |B|\} \Rightarrow t_2\}}$$

$$\textsc{S-AndL}\quad \frac{C^\circ \qquad t : A <: C \leadsto t'}{t : A \,\&\, B <: C \leadsto t'} \qquad \textsc{S-AndR}\quad \frac{C^\circ \qquad t : B <: C \leadsto t'}{t : A \,\&\, B <: C \leadsto t'}$$

$$\textsc{S-Split}\quad \frac{B_1 \lhd B \rhd B_2 \qquad t : A <: B_1 \leadsto t_1 \qquad t : A <: B_2 \leadsto t_2 \qquad t_1 : B_1 \rhd B \lhd t_2 : B_2 \leadsto t_3}{t : A <: B \leadsto t_3}$$

$\boxed{t_1 : A \rhd C \lhd t_2 : B \rightsquigarrow t}$ *(Coercive merging)*

$$\textsc{M-And}\quad \frac{}{t_1 : A \rhd A \,\&\, B \lhd t_2 : B \rightsquigarrow x; y}$$

$$\textsc{M-Arrow}\quad \frac{(t_1.\overrightarrow{|B_1|})\, x : B_1 \rhd B \lhd (t_2.\overrightarrow{|B_2|})\, x : B_2 \rightsquigarrow t}{t_1 : A \to B_1 \rhd A \to B \lhd t_2 : A \to B_2 \rightsquigarrow \{\overrightarrow{|B|} \Rightarrow \lambda x.t\}}$$

$$\textsc{M-All}\quad \frac{(t_1.|B_1|^{\forall})\, x : B_1 \rhd B \lhd (t_2.|B_2|^{\forall})\, x : B_2 \rightsquigarrow t}{t_1 : \forall X {*} A.\ B_1 \rhd \forall X {*} A.\ B \lhd t_2 : \forall X {*} A.\ B_2 \rightsquigarrow \{|B|^{\forall} \Rightarrow \lambda x.t\}}$$

$$\textsc{M-Rcd}\quad \frac{(t_1.\{\ell : |A_1|\})\, \{\,\} : A_1 \rhd A \lhd (t_2.\{\ell : |A_2|\})\, \{\,\} : A_2 \rightsquigarrow t}{t_1 : \{\ell : A_1\} \rhd \{\ell : A\} \lhd t_2 : \{\ell : A_2\} \rightsquigarrow \{\{\ell : |A|\} \Rightarrow \lambda x.t\}}$$

3.4 Auxiliary definitions

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

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

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

$\boxed{\rceil A \lceil}$ *(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)*

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