Slick Lang

1 Acknowledgements

This type system is heavily adapted from the one presented in *Complete and Easy Bidirectional Typing for Higher-Rank Polymorphism* by Dunfield and Krishnaswami.

Other references include *Types and Programming Langauges* by Benjamin Pierce and Chapter 10 of *Advanced Types and Programming Languages*, "The Essence of ML Type Inference," by François Pottier and Didier Rémy.

2 Declarative Typing

2.1 Terms

labels (includes capital labels)	ℓ		
capital labels	L		
patterns	p		
variables	v		
expressions	e	::=	
	functions	|	$\backslash v \to e$
	applications	|	$e_1 e_2$
	sequences	|	$e_1; e_2$
	assignments	|	$v := e$
	annotations	|	$e : A$
	records	|	$\{\ell_1 = e_1, \ell_2 = e_2, \dots\}$
	variants	|	$L\ e$
	case	|	$\texttt{case}\ e : \lvert p_1 \to e_1 \rvert p_2 \to e_2 \dots$

2.2 Types and type machinery

labels	ℓ		
type variables	α, β		
row type variables	ρ		
contexts	Ψ	$::=$	$\cdot$
		$\mid$	$v = A, \Psi$
monomorphic types	τ, σ	$::=$	$\{\rho\}$
		$\mid$	$[\![\rho]\!]$
		$\mid$	$\mu\alpha.\tau$
		$\mid$	$\tau \to \sigma$
types	A, B	$::=$	$\forall\alpha.A$
		$\mid$	τ
rows	r	$::=$	$\cdot$
		$\mid$	$\ell : A, r$
monomorphic rows	ρ	$::=$	$\cdot$
		$\mid$	$\ell : \tau, \rho$

2.3 Typing rules

This section is a little incomplete since the Algorithmic Typing is what matters more for the implementation.

2.3.1 Functions

$$\frac{\Psi, v = A, \Psi' \qquad v \notin \Psi}{\Psi, v = A, \Psi' \vdash v : A}\ \text{(Var)} \qquad \frac{v = A, \Psi \vdash e : B}{\Psi \vdash \backslash v \to e : A \to B}\ \text{(Function)}$$

$$\frac{\Psi \vdash e_1 : A \to B \qquad \Psi \vdash e_2 : A}{\Psi \vdash e_1 e_2 : B}\ \text{(App)}$$

2.3.2 Records

$$\frac{}{\Psi \vdash \{\} : \cdot}\ \text{(Empty record)} \qquad \frac{\Psi \vdash \{rcd\} : \{r\}}{\Psi \vdash \{\ell = e, rcd\} : \{\ell : A, r\}}\ \text{(Record)}$$

3 Algorithmic typing

3.1 Terms

Same as those from the Declarative typing section.

3.2 Types and type machinery

labels	ℓ		
existential variables	$\hat{\alpha}$		
existential row variables	α_ρ		
type variables	α, β		
row type variables	ρ		
base types	$\mathcal{B}$	$::=$	$\texttt{Int}\vert\texttt{Bool}\vert\ldots$
contexts	Γ, Δ, Θ	$::=$	$\cdot$
	vars	$\vert$	$v : A, \Gamma$
	solved evars	$\vert$	$\hat{\alpha} : A, \Gamma$
	solved row evars	$\vert$	$\alpha_\rho : r, \Gamma$
	evars	$\vert$	$\hat{\alpha}, \Gamma$
	type vars	$\vert$	α, Γ
	markers	$\vert$	$\blacktriangleright_{\hat{\alpha}}, \Gamma$
monomorphic types	τ, σ	$::=$	$\{\rho\}$
		$\vert$	$[\![\rho]\!]$
		$\vert$	$\mu\alpha.\tau$
		$\vert$	$\tau \to \sigma$
		$\vert$	$\mathcal{B}$
types	A, B	$::=$	$\forall\alpha.A$
		$\vert$	τ
rows	r	$::=$	$\cdot$
		$\vert$	$\ell : A, r$
monomorphic rows	ρ	$::=$	$\cdot$
		$\vert$	$\ell : \tau, \rho$

3.3 A note on quantifiers

You'll note that there are two quantifiers in Slick: $\forall\alpha.A$ and $\aleph\alpha_\rho.A$. This is because row tails don't strictly contain types, so it makes sense to have a quantifier over just the universe of rows, which is what the latter is.

In practice, the two are checked equivalently. All rules pertaining to $\forall\alpha.A$, unless stated otherwise, have mirrored counterparts for $\aleph\alpha_\rho.A$.

3.4 Typing rules

We define algorithmic typing with the following judgments:

$$\Gamma \vdash e \Leftarrow A \dashv \Delta \qquad \Gamma \vdash e \Rightarrow A \dashv \Delta$$

which respectively represent type checking (inputs: Γ, e, A; output: Δ) and type synthesis (inputs: Γ, e; outputs: A, Δ).

We also define a binary algorithmic judgement:

$$\Gamma \vdash X \square Y \Rrightarrow Z \dashv \Delta$$

which represents a binary judgement $\square$ under the context Γ on values X and Y that synthesizes Z with output context Δ. For example, the syntax that Dunfield and Krishnaswami use for function application synthesis judgements would be

$$\Gamma \vdash A \bullet e \Rrightarrow C \dashv \Delta$$

which means that under context Γ, A applied to the term e synthesizes output type C and context Δ.

$$\frac{(x : A) \in \Gamma}{\Gamma \vdash x \Rightarrow A \dashv \Gamma}\ (\text{Var}) \qquad \frac{\Gamma \vdash e \Rightarrow A \dashv \Theta \qquad \Theta \vdash [\Theta]\, A <: [\Theta]\, B \dashv \Delta}{\Gamma \vdash e \Leftarrow B \dashv \Delta}\ (\text{Sub})$$

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash e \Leftarrow A \dashv \Delta}{\Gamma \vdash (e : A) \Rightarrow A \dashv \Delta}\ (\text{Annotation}) \qquad \frac{\Gamma, \alpha \vdash e \Leftarrow A \dashv \Delta, \alpha, \Theta}{\Gamma \vdash e \Leftarrow \forall \alpha . A \dashv \Delta}\ (\forall\ \text{I})$$

$$\frac{\Gamma, v : A \vdash e \Leftarrow B \dashv \Delta, v : A, \Theta}{\Gamma \vdash \backslash v \to e \Leftarrow A \to B \dashv \Delta}\ (\to \text{I})$$

$$\frac{\Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha}, \hat{\beta}, v : \hat{\alpha} \vdash e \Leftarrow \hat{\beta} \dashv \Delta, \blacktriangleright_{\hat{\alpha}}, \Theta \qquad \tau = [\Delta]\,(\hat{\alpha} \to \hat{\beta}) \qquad \overline{\hat{\alpha}} = \text{unsolved}(\tau)}{\Gamma \vdash \backslash v \to e \Rightarrow \forall \overline{\alpha} . \tau[\overline{\hat{\alpha}} \mapsto \overline{\alpha}] \dashv \Delta}\ (\to \text{I} \Longrightarrow)$$

$$\frac{\Gamma \vdash e_1 \Rightarrow A \dashv \Theta \qquad \Theta \vdash [\Theta]\, A \bullet e_2 \Rrightarrow C \dashv \Delta}{\Gamma \vdash e_1\ e_2 \Rightarrow C \dashv \Delta}\ (\to \text{E}) \qquad \frac{\Gamma, \hat{\alpha} \vdash A[\alpha := \hat{\alpha}] \bullet e \Rrightarrow C \dashv \Delta}{\Gamma \vdash \forall \alpha . A \bullet e \Rrightarrow C \dashv \Delta}\ (\forall\ \text{App})$$

$$\frac{\Gamma \vdash e \Leftarrow A \dashv \Delta}{\Gamma \vdash A \to C \bullet e \Rrightarrow C \dashv \Delta}\ (\to \text{App}) \qquad \frac{\Gamma \vdash A[\alpha \mapsto \mu \alpha . A] \bullet e \Rrightarrow C \dashv \Delta}{\Gamma \vdash \mu \alpha . A \bullet e \Rrightarrow C \dashv \Delta}\ (\mu \text{App})$$

$$\frac{\Gamma[\hat{\alpha_2}, \hat{\alpha_1}, \hat{\alpha} = \hat{\alpha_1} \to \hat{\alpha_2}] \vdash e \Leftarrow \hat{\alpha_1} \dashv \Delta}{\Gamma[\hat{\alpha}] \vdash \hat{\alpha} \bullet e \Rrightarrow \hat{\alpha}_2 \dashv \Delta}\ (\hat{\alpha} \text{App})$$

$$\frac{}{\Gamma \vdash \{\} \Leftarrow \{\} \dashv \Gamma}\ (\{\}\text{I}) \qquad \frac{}{\Gamma \vdash \{\} \Rightarrow \{\} \dashv \Gamma}\ (\{\}\text{I} \Longrightarrow)$$

$$\frac{\Gamma \vdash e \Leftarrow A \dashv \Theta \qquad \Theta \vdash [\Theta]\, \{rcd\} \Leftarrow [\Theta]\, r \dashv \Delta}{\Gamma \vdash \{\ell : e, rcd\} \Leftarrow \{\ell : A, r\} \dashv \Delta}\ (\text{RcdI}) \qquad \frac{\Gamma \vdash e \Rightarrow A \dashv \Theta \qquad \Theta \vdash [\Theta]\, \{rcd\} \Rightarrow [\Theta]\, r \dashv \Delta}{\Gamma \vdash \{\ell : e, rcd\} \Rightarrow \{\ell : A, r\} \dashv \Delta}\ (\text{RcdI}$$

$$\frac{\Gamma \vdash e \Rightarrow A \dashv \Theta \qquad \Theta \vdash [\Theta]\, A \,.\, \ell \Rrightarrow C \dashv \Delta}{\Gamma \vdash e \# \ell \Rightarrow C \dashv \Delta}\ (\text{Prj}) \qquad \frac{\Gamma, \hat{\alpha} \vdash A[\alpha := \hat{\alpha}] \,.\, \ell \Rrightarrow C \dashv \Delta}{\Gamma \vdash \forall \alpha . A \,.\, \ell \Rrightarrow C \dashv \Delta}\ (\forall\ \text{Prj})$$

$$\frac{\Gamma \vdash R\#l \longrightarrow C \dashv \Delta}{\Gamma \vdash R\,.\,\ell \Rrightarrow C \dashv \Delta}\ (\text{RcdPrjR}) \qquad \frac{\Gamma \vdash A[\alpha \mapsto \mu\alpha.A]\,.\,\ell \Rrightarrow C \dashv \Delta}{\Gamma \vdash \mu\alpha.A\,.\,\ell \Rrightarrow C \dashv \Delta}\ (\mu\text{Prj})$$

We define record lookup $\Gamma \vdash \rho\#\ell \longrightarrow A \dashv \Delta$ as follows (inputs: Γ, ρ, l; outputs: A, Δ):

$$\frac{}{\Gamma \vdash \{\ell : A, R\}\#\ell \longrightarrow A \dashv \Gamma}\ (\text{lookupYes}) \qquad \frac{\ell \neq \ell' \qquad \Gamma \vdash \{R\}\#\ell \longrightarrow A \dashv \Delta}{\Gamma \vdash \{\ell' : A', R\}\#\ell \longrightarrow A \dashv \Delta}\ (\text{lookupNo})$$

$$\frac{}{\Gamma[\hat{\alpha}] \vdash \hat{\alpha}\#\ell \longrightarrow \hat{\alpha}_0 \dashv \Gamma[\hat{\alpha}_0, \hat{\alpha}_1, \hat{\alpha} = \{\ell : \hat{\alpha}_0, \hat{\alpha}_1\}]}\ (\text{Lookup } \hat{\alpha})$$

$$\frac{}{\Gamma[\hat{\alpha}] \vdash \{\hat{\alpha}\}\#\ell \longrightarrow \hat{\alpha}_0 \dashv \Gamma[\hat{\alpha}_0, \hat{\alpha}_1, \hat{\alpha} = (\ell : \hat{\alpha}_0, \hat{\alpha}_1)]}\ (\text{Lookup RowTail})$$

3.5 Subsumption

We define the algorithmic subsumption:

$$\Gamma \vdash A_0 <: A_1 \dashv \Delta$$

which represents A_0 subsumes A_1 with input context Γ and output context Δ. Subsumption is like subtyping, but only applies to quantifiers. Everything else must be strict equality (for now, this also means records, so you can't use $\{\ell_1 : \texttt{Bool}, \ell_2 : \texttt{Num}\}$ in place of $\{\ell_1 : \texttt{Bool}\}$ even though you really *should* be able to).

$$\frac{}{\Gamma[\hat{\alpha}] \vdash \hat{\alpha} <: \hat{\alpha} \dashv \Gamma[\hat{\alpha}]}\ (\text{EVar}) \qquad \frac{}{\Gamma[\alpha] \vdash \alpha <: \alpha \dashv \Gamma[\alpha]}\ (\text{Var}) \qquad \frac{}{\Gamma \vdash \mathcal{B} <: \mathcal{B} \dashv \Gamma}\ (\text{Const})$$

$$\frac{\Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha} \vdash A[\alpha := \hat{\alpha}] <: B \dashv \Delta, \blacktriangleright_{\hat{\alpha}}, \Theta}{\Gamma \vdash \forall\alpha.A <: B \dashv \Delta}\ (\forall\text{L}) \qquad \frac{\Gamma, \alpha \vdash A <: B \dashv \Delta, \alpha, \Theta}{\Gamma \vdash A <: \forall\alpha.B \dashv \Delta}\ (\forall\text{R})$$

$$\frac{\Gamma \vdash B_1 <: A_1 \dashv \Theta \qquad \Theta \vdash [\Theta]\,A_2 <: [\Theta]\,B_2 \dashv \Delta}{\Gamma \vdash A_1 \to A_2 <: B_1 \to B_2 \dashv \Delta}\ (\to)$$

$$\frac{\hat{\alpha} \notin FV(A) \qquad \Gamma[\hat{\alpha}] \vdash \hat{\alpha} \mathrel{:\!\leq} A \dashv \Delta}{\Gamma[\hat{\alpha}] \vdash \hat{\alpha} <: A \dashv \Delta}\ \text{InstantiateL} \qquad \frac{\hat{\alpha} \notin FV(A) \qquad \Gamma[\hat{\alpha}] \vdash A \mathrel{\leq\!:} \hat{\alpha} \dashv \Delta}{\Gamma[\hat{\alpha}] \vdash A <: \hat{\alpha} \dashv \Delta}\ \text{InstantiateR}$$

$$\frac{\hat{\alpha} \in FV(A) \qquad \Gamma[\hat{\alpha}] \vdash \hat{\alpha} \mathrel{:\!\leq} \mu\alpha.A[\hat{\alpha} \mapsto \alpha] \dashv \Delta}{\Gamma[\hat{\alpha}] \vdash \hat{\alpha} <: A \dashv \Delta}\ \text{InstantiateL}\mu$$

$$\frac{\hat{\alpha} \in FV(A) \qquad \Gamma[\hat{\alpha}] \vdash \mu\alpha.A[\hat{\alpha} \mapsto \alpha] \mathrel{\leq\!:} \hat{\alpha} \dashv \Delta}{\Gamma[\hat{\alpha}] \vdash A <: \hat{\alpha} \dashv \Delta}\ \text{InstantiateR}\mu$$

The above rules for recursive types will likely be changed to be more restrictive: probably only when A is a record or variant (or perhaps contains one).

$$\frac{\Gamma \vdash R_0 <: R_1 \dashv \Delta}{\Gamma \vdash \{R_0\} <: \{R_1\} \dashv \Delta}\ \text{(Record)} \qquad \frac{\Gamma \vdash R_0 <: R_1 \dashv \Delta}{\Gamma \vdash [\![R_0]\!] <: [\![R_1]\!] \dashv \Delta}\ \text{(Variant)}$$

For this rule, we treat the rows as sets and assume they are reordered so that the matching labels are at the front of the row. An algorithmic implementation would want to deal with the recursive and base cases by looking at the set intersection and difference of the rows.

$$\frac{\Gamma \vdash A <: B \dashv \Theta \qquad \Theta \vdash [\Theta]R_1 <: [\Theta]R_2 \dashv \Delta}{\Gamma \vdash \ell : A, R_1 <: \ell : B, R_2 \dashv \Delta}\ \text{(Row)} \qquad \frac{}{\Gamma \vdash \cdot <: \cdot \dashv \Gamma}\ \text{(Row Nil)}$$

$$\frac{\Gamma[\hat{\alpha}] \vdash \hat{\alpha} \dot{\leq} r \dashv \Delta}{\Gamma[\hat{\alpha}], \cdot \vdash \hat{\alpha} <: r \dashv \Delta}\ \text{(RowMissingL)} \qquad \frac{\Gamma[\hat{\alpha}] \vdash r \dot{\leq} \hat{\alpha} \dashv \Delta}{\Gamma[\hat{\alpha}], \cdot \vdash r <: \hat{\alpha} \dashv \Delta}\ \text{(RowMissingR)}$$

This rule also treats the rows as sets and assumes that there are no equal labels between the two rows. Assume an analagous rule for a different order of EVars (the order doesn't matter).

$$\frac{\Gamma \vdash \hat{\alpha} <: m_1 : B_1, m_2 : B_2, \ldots \dashv \Theta \qquad \Theta \vdash \ell_1 : [\Theta]\, A_1, \ell_2 : [\Theta]\, A_2, \ldots <: \hat{\beta} \dashv \Delta}{\Gamma[\hat{\alpha}][\hat{\beta}] \vdash \ell_1 : A_1, \ell_2 : A_2, \ldots, \hat{\alpha} <: m_1 : B_1, m_2 : B_2, \ldots, \hat{\beta} \dashv \Delta}\ \text{(RowMissingLR)}$$

$$\frac{}{\Gamma[\hat{\alpha}][\hat{\beta}] \vdash \hat{\alpha} <: \hat{\beta} \dashv \Gamma[\hat{\alpha}][\hat{\beta} = \hat{\alpha}]}\ \text{(RowTailReach)}$$

3.6 Instantiation

Instantiation is a judgement that solves an EVar, either on the left or right side of the judgement. It is important to recurisvely assign "helper" EVars to match the shape of whatever the EVar is being instantiated to instead of blindly assigning it, as there may be EVars inside of the type it is being assigned to.

The EVar is instantiated so that it subsumes or is subsumed by the type, depending on the type of instantiation (left or right, respectively).

!!! Need an instantiation for rows that is like InstRcd !!!

3.6.1 Left Instantiation

$$\frac{\Gamma \vdash \mathcal{B}}{\Gamma[\hat{\alpha}] \vdash \hat{\alpha} \dot{\leq} \mathcal{B} \dashv \Gamma[\hat{\alpha} = \mathcal{B}]}\ \text{InstLSolve} \qquad \frac{}{\Gamma[\hat{\alpha}][\hat{\beta}] \vdash \hat{\alpha} \dot{\leq} \hat{\beta} \dashv \Gamma[\hat{\alpha}][\hat{\beta} = \hat{\alpha}]}\ \text{InstLReach}$$

$$\frac{\Gamma[\hat{\alpha}_2, \hat{\alpha}_1, \hat{\alpha} = \hat{\alpha}_1 \to \hat{\alpha}_2] \vdash A_1 \dot{\leq} \hat{\alpha}_1 \dashv \Theta \qquad \Theta \vdash \hat{\alpha}_2 \dot{\leq} [\Theta]\, A_2 \dashv \Delta}{\Gamma[\hat{\alpha}] \vdash \hat{\alpha} \dot{\leq} A_1 \to A_2 \dashv \Delta}\ \text{InstLArr}$$

$$\frac{\begin{array}{l}\Gamma_0[(\hat{\alpha}_{k+1}), \hat{\alpha}_k, \ldots, \hat{\alpha}_1, \hat{\alpha} = \{\ell_1 : \hat{\alpha}_1, \ell_2 : \hat{\alpha}_2, \ldots, \ell_k : \hat{\alpha}_k, (\hat{\alpha}_{k+1})\}] \vdash \hat{\alpha}_1 \mathrel{:\leq} A_1 \dashv \Gamma_1 \\ \Gamma_1 \vdash \hat{\alpha}_2 \mathrel{:\leq} [\Gamma_1]\, A_2 \dashv \Gamma_2 \qquad \cdots \qquad (\Gamma_k \vdash \hat{\alpha}_{k+1} \mathrel{:\leq} [\Gamma_k]\, \hat{\beta} \dashv \Delta)\end{array}}{\Gamma_0[\hat{\alpha}] \vdash \hat{\alpha} \mathrel{:\leq} \{\ell_1 : A_1, \ell_2 : A_2, \ldots, \ell_{k-1} : A_{k-1}, (\hat{\beta})\} \dashv \Delta}\ \text{InstLRcd}$$

In the above rule, the parentheticals only come into play if there is a row tail in the record. If there isn't, assume that $\Delta = \Gamma_k$.

$$\frac{\Gamma[\hat{\alpha}], \beta \vdash \hat{\alpha} \mathrel{:\leq} B \dashv \Delta, \beta, \Delta'}{\Gamma[\hat{\alpha}] \vdash \hat{\alpha} \mathrel{:\leq} \forall \beta.B \dashv \Delta}\ \text{InstLAllR}$$

3.6.2 Right Instantiation

$$\frac{\Gamma \vdash \mathcal{B}}{\Gamma[\hat{\alpha}] \vdash \mathcal{B} \leq: \hat{\alpha} \dashv \Gamma[\hat{\alpha} = \mathcal{B}]}\ \text{InstRSolve} \qquad \frac{}{\Gamma[\hat{\alpha}][\hat{\beta}] \vdash \hat{\beta} \leq: \hat{\alpha} \dashv \Gamma[\hat{\alpha}][\hat{\beta} = \hat{\alpha}]}\ \text{InstRReach}$$

$$\frac{\Gamma[\hat{\alpha}_2, \hat{\alpha}_1, \hat{\alpha} = \hat{\alpha}_1 \to \hat{\alpha}_2] \vdash \hat{\alpha}_1 \mathrel{:\leq} A_1 \dashv \Theta \qquad \Theta \vdash [\Theta]\, A_2 \leq: \hat{\alpha}_2 \dashv \Delta}{\Gamma[\hat{\alpha}] \vdash A_1 \to A_2 \leq: \hat{\alpha} \dashv \Delta}\ \text{InstRArr}$$

$$\frac{\begin{array}{l}\Gamma_0[(\hat{\alpha}_{k+1}), \hat{\alpha}_k, \ldots, \hat{\alpha}_1, \hat{\alpha} = \{\ell_1 : \hat{\alpha}_1, \ell_2 : \hat{\alpha}_2, \ldots, \ell_k : \hat{\alpha}_k, (\hat{\alpha}_{k+1})\}] \vdash A_1 \leq: \hat{\alpha}_1 \dashv \Gamma_1 \\ \Gamma_1 \vdash [\Gamma_2]\, A_2 \leq: \hat{\alpha}_2 \dashv \Gamma_3 \qquad \cdots \qquad (\Gamma_k \vdash \hat{\beta} \leq: \hat{\alpha}_{k+1} \dashv \Delta)\end{array}}{\Gamma_0[\hat{\alpha}] \vdash \{\ell_1 : A_1, \ell_2 : A_2, \ldots, \ell_{k-1} : A_{k-1}, (\hat{\beta})\} \leq: \hat{\alpha} \dashv \Delta}\ \text{InstRRcd}$$

In the above rule, the parentheticals only come into play if there is a row tail in the record. If there isn't, assume that $\Delta = \Gamma_k$.

$$\frac{\Gamma[\hat{\alpha}], \blacktriangleright_{\hat{\alpha}}, \hat{\beta} \vdash A[\beta := \hat{\beta}] <: \hat{\alpha} \dashv \Delta, \blacktriangleright_{\hat{\beta}}, \Delta'}{\Gamma[\hat{\alpha}] \vdash \forall \beta.B <: \hat{\alpha} \dashv \Delta}\ \text{InstRAllL}$$