Refinement Types for Elm

Master Thesis Report

Lucas Payr

26 January 2020

Topics of this Talk

- Revisiting the Max Function
- The Inference Algorithm
- Demonstration

Revisiting the Max Function

```
max : a:{ v:Int|True } -> b:{ v:Int|True } -> { v:Int|k4 }
max =
  \a -> \b ->
    if
      (<) a b
    then
      b
    else
      a
```

We want to derive the refinement label as $k4$.

Revisiting the Max Function

We remained with the following problem:

Find refinements $\kappa_1, \kappa_2, \kappa_3$ and κ_4 such that:

$$\{\nu : Int | \nu = b\} <:_{\{(a,\{Int|True\}),(b,\{Int|True\})\},\{a<b\}} \{\nu : Int | \kappa_3\},$$
$$\{\nu : Int | \nu = a\} <:_{\{(a,\{Int|True\}),(b,\{Int|True\})\},\{\neg(a<b)\}} \{\nu : Int | \kappa_3\},$$

$$a : \{\nu : Int | \kappa_1\} \to b : \{\nu : Int | \kappa_2\} \to \{\nu : Int | \kappa_3\}$$
$$<:_{\{\},\{\}} a : \{\nu : Int | True\} \to b : \{\nu : Int | True\} \to \{\nu : Int | \kappa_4\}$$

The Inference Algorithm: Definitions

Subtyping Condition

We say c is a *Subtyping Condition* $:\Leftrightarrow$ c is of form $\hat{T}_1 <:_{\Theta,\Lambda} \hat{T}_2$ where $\hat{T}_1, \hat{T}_2$ are a liquid types or templates, Θ is a type variable context and $\Lambda \subset \mathcal{Q}$.

Template

We say $\hat{T}$ is a *template* $:\Leftrightarrow$ $\hat{T}$ is of form $\{\nu : \mathit{Int} \mid [k]_S\}$ where $k \in \mathcal{K}$ and $S : \mathcal{V} \nrightarrow \mathcal{Q}$

$\vee$ $\hat{T}$ is of form $a : \{\nu : \mathit{Int} \mid [k]_S\} \to \hat{T}$ where $k \in \mathcal{K}$, $\hat{T}$ is a template and $S : \mathcal{V} \nrightarrow \mathit{IntExp}$.

We define $\mathcal{K} := \{\kappa_i \mid i \in \mathbb{N}\}$.

The Inference Algorithm

$$
\begin{aligned}
&\mathsf{Infer} : \mathcal{P}(\mathcal{C}) \to \ (\mathcal{K} \nrightarrow \mathcal{Q}) \\
&\mathsf{Infer}(C) = \\
&\quad \mathsf{Let}\ V := \bigcup_{\hat{T}_1 <:_{\Theta,\Lambda} \hat{T}_2 \in C} \{a \mid (a, \_) \in \Theta\} \\
&\qquad Q_0 := \mathit{Init}(V), \\
&\qquad A_0 := \{(\kappa, Q_0) \mid \kappa \in \bigcup_{c \in C} \mathsf{Var}(c)\}, \\
&\qquad A := \mathsf{Solve}(\bigcup_{c \in C} \mathsf{Split}(c), A_0) \\
&\quad \mathsf{in}\ \{(\kappa, \bigwedge Q) \mid (\kappa, Q) \in A\}
\end{aligned}
$$

The Inference Algorithm: Step 1 (Split)

$$\text{Split} : \mathcal{C} \nrightarrow \mathcal{P}(\mathcal{C}^-)$$

$$\text{Split}(a : \{\nu : Int | q_1\} \to \hat{T}_2 <:_{\Theta,\Lambda} a : \{\nu : Int | q_3\} \to \hat{T}_4) =$$
$$\{\{\nu : Int | q_3\} <:_{\Theta,\Lambda} \{\nu : Int | q_1\}\} \cup \text{Split}(\hat{T}_2 <:_{\Theta \cup \{(a,q_3)\},\Lambda} \hat{T}_4\})$$

$$\text{Split}(\{\nu : Int | q_1\} <:_{\Theta,\Lambda} \{\nu : Int | q_2\}) =$$
$$\{\{\nu : Int | q_1\} <:_{\Theta,\Lambda} \{\nu : Int | q_2\}\}$$

$$\mathcal{C} := \{c \mid c \text{ is a subtyping condition}\}$$

$$\mathcal{C}^- := \{\ \{\nu : Int | q_1\} <:_{\Theta,\Lambda} \{\nu : Int | q_2\}$$
$$\mid (q_1 \in \mathcal{Q} \vee q_1 = [k_1]_{S_1} \text{ for } k_1 \in \mathcal{K}, S_1 \in \mathcal{V} \nrightarrow IntExp)$$
$$\wedge (q_2 \in \mathcal{Q} \vee q_2 = [k_2]_{S_2} \text{ for } k_2 \in \mathcal{K}, S_2 \in \mathcal{V} \nrightarrow IntExp)\}.$$

The Inference Algorithm: Step 1 (Split)

$$\Theta := \{(a, \{Int|\kappa_1\}), (b, \{Int|\kappa_2\})\}$$

$$\begin{aligned} C_0 := \{&\{\nu : Int|\nu = b\} <:_{\Theta,\{a<b\}} \{\nu : Int|\kappa_3\}, \\ &\{\nu : Int|\nu = a\} <:_{\Theta,\{\neg(a<b)\}} \{\nu : Int|\kappa_3\}, \\ &a : \{\nu : Int|\kappa_1\} \to b : \{\nu : Int|\kappa_2\} \to \{\nu : Int|\kappa_3\} \\ &\quad <:_{\{\},\{\}} a : \{\nu : Int|\, True\} \to b : \{\nu : Int|\, True\} \to \{\nu : Int|\kappa_4\} \end{aligned}$$

After Step 1:

$$\begin{aligned} C := \{&\{\nu : Int|\nu = b\} <:_{\Theta,\{a<b\}} \{\nu : Int|\kappa_3\}, \\ &\{\nu : Int|\nu = a\} <:_{\Theta,\{\neg(a<b)\}} \{\nu : Int|\kappa_3\}, \\ &\{\nu : Int|\, True\} <:_{\{\},\{\}} \{\nu : Int|\kappa_1\}, \\ &\{\nu : Int|\, True\} <:_{\{(a,\{\nu:Int|\, True\})\},\{\}} \{\nu : Int|\kappa_2\}, \\ &\{\nu : Int|\kappa_3\} <:_{\Theta,\{\}} \{\nu : Int|\kappa_4\}\} \end{aligned}$$

The Inference Algorithm: Step 2 (Solve)

$$
\begin{aligned}
\mathit{Init} : \mathcal{P}(\mathcal{V}) \to & \mathcal{P}(\mathcal{Q}) \\
\mathit{Init}(V) ::= & \{0 < \nu\} \\
& \cup \{a < \nu \mid a \in V\} \\
& \cup \{\nu < 0\} \\
& \cup \{\nu < a \mid a \in V\} \\
& \cup \{\nu = a \mid a \in V\} \\
& \cup \{\nu = 0\} \\
& \cup \{a < \nu \vee \nu = a \mid a \in V\} \\
& \cup \{\nu < a \vee \nu = a \mid a \in V\} \\
& \cup \{0 < \nu \vee \nu = 0\} \\
& \cup \{\nu < 0 \vee \nu = 0\} \\
& \cup \{\neg(\nu = a) \mid a \in V\} \\
& \cup \{\neg(\nu = 0)\}
\end{aligned}
$$

In our example $V := \{a, b\}$

The Inference Algorithm: Step 2 (Solve)

$\text{Solve} : \mathcal{P}(\mathcal{C}^-) \times (\mathcal{K} \nrightarrow \mathcal{P}(\mathcal{Q})) \to (\mathcal{K} \nrightarrow \mathcal{P}(\mathcal{Q}))$

$\text{Solve}(C, A) =$

Let $S := \{(k, \bigwedge Q) \mid (k, Q) \in A\}$.

If there exists $(\{\nu : Int \mid q_1\} <:_{\Theta,\Lambda} \{\nu : Int \mid [k_2]_{S_2}\}) \in C$ such that

$\neg(\forall z \in \mathbb{Z}.\forall i_1 \in \text{value}_\Gamma(\{\nu : Int | r'_1\}). \ldots \forall i_n \in \text{value}_\Gamma(\{\nu : Int | r'_n\}).$

$[[r_1 \wedge p]]_{\{(\nu,z),(b_1,i_1),\ldots,(b_n,i_n)\}} \Rightarrow [[r_2]]_{\{(\nu,z),(b_1,i_1),\ldots,(b_n,i_n)\}})$

then $\text{Solve}(C, \text{Weaken}(c, A))$ else A

SMT statement:

$$((\bigwedge_{j=0}^{n} [r'_j]_{\{(\nu,b_j)\}}) \wedge r_1 \wedge p) \wedge \neg r_2$$

with free variables $\nu \in \mathbb{Z}$ and $b_i \in \mathbb{Z}$ for $i \in \mathbb{N}_1^n$.

The Inference Algorithm: Step 3 (Weaken)

$$\text{Weaken} : \mathcal{C}^- \times (\mathcal{K} \nrightarrow \mathcal{P}(\mathcal{Q})) \nrightarrow (\mathcal{K} \nrightarrow \mathcal{P}(\mathcal{Q}))$$

$$\text{Weaken}(\{\nu : Int | x\} <:_{\Theta,\wedge} \{\nu : Int | [k_2]_{S_2}\}, A) =$$

Let $S := \{(k, \bigwedge Q) \mid (k, Q) \in A\}$,

$$Q_2 := \{\ q$$
$$\mid q \in A(k_2)$$
$$\wedge(\forall z \in \mathbb{Z}.\forall i_1 \in \text{value}_\Gamma(\{\nu : Int | r_1'\}). \ldots \forall i_n \in \text{value}_\Gamma(\{\nu : Int | r_n'\}).$$
$$[[r_1 \wedge p]]_{\{(\nu,z),(b_1,i_1),\ldots,(b_n,i_n)\}} \Rightarrow [[[q]_{S_2}]]_{\{(\nu,z),(b_1,i_1),\ldots,(b_n,i_n)\}})\}$$

in $\{(k, Q) \mid (k, Q) \in A \wedge k \neq k_2\} \cup \{(k_2, Q_2)\}$

SMT statement:

$$\neg((\bigwedge_{j=0}^{n} [r_j']_{\{(\nu,b_j)\}}) \wedge r_1 \wedge p) \vee r_2$$

with free variables $\nu \in \mathbb{Z}$ and $b_i \in \mathbb{Z}$ for $i \in \mathbb{N}_1^n$ and $r_2 := [q]_{S_2}$.

Demonstration

$$\Theta := \{(a, \{Int|\kappa_1\}), (b, \{Int|\kappa_2\})\}$$

$$\begin{aligned} C_0 := \{&\{\nu : Int|\nu = b\} <:_{\Theta,\{a<b\}} \{\nu : Int|\kappa_3\}, \\ &\{\nu : Int|\nu = a\} <:_{\Theta,\{\neg(a<b)\}} \{\nu : Int|\kappa_3\}, \\ &a : \{\nu : Int|\kappa_1\} \to b : \{\nu : Int|\kappa_2\} \to \{\nu : Int|\kappa_3\} \\ &\quad <:_{\{\},\{\}} a : \{\nu : Int|\, True\} \to b : \{\nu : Int|\, True\} \to \{\nu : Int|\kappa_4\} \end{aligned}$$

Demonstration

Current State

1. Formal language similar to Elm **(DONE)**
2. Extension of the formal language with Liquid Types
   - 2.1 A formal syntax **(DONE)**
   - 2.2 A formal type system **(DONE)**
   - 2.3 Proof that the extension infers the correct types. **(DONE)**
   - 2.4 Implementation of the inference algorithm. **(DONE)**

Started thesis in July 2019

Expected finish in February 2021