Assignment 1 Report of DSL

Faiz Ilham Muhammad (5231981) and Xinliang Lu (0822760)

May 24, 2023

1 Implemented Functionalities

We have implemented the following functionalities:

- Type inference for the underlying type system (including polymorphism, arrays, pairs and lists).
- Support confidentiality labels to the type system.
- Report errors for violations of constant time.
- Support mutable arrays, pairs, lists to the type system.
- (Bonus) Subtyping extension to subeffecting.
- (Bonus) Allow type annotation without labels.

The examples required by the assignment are attached in Section 3.

2 Typing Rules

2.1 Syntax-driven Rules

$$\frac{\ell \sqsubseteq \mathrm{L}}{\hat{\Gamma} \vdash_{\mathrm{CTC}} n : \mathsf{Nat}^{\ell}} \text{ [CT-Nat]}$$

$$\frac{\ell \sqsubseteq \mathrm{L}}{\hat{\Gamma} \vdash_{\mathrm{CTC}} \textbf{true} : \mathsf{Bool}^{\ell}} \text{ [CT-True]}$$

$$\frac{\ell \sqsubseteq \mathrm{L}}{\hat{\Gamma} \vdash_{\mathrm{CTC}} \textbf{false} : \mathsf{Bool}^{\ell}} \text{ [CT-False]}$$

$$\frac{\hat{\Gamma}[x] = \hat{\tau}^{\ell}}{\hat{\Gamma} \vdash_{\mathrm{CTC}} x : \hat{\tau}^{\ell}} \text{ [CT-Var]}$$

$$\frac{\hat{\Gamma}[x \mapsto \hat{\tau}_1{}^{\ell_1}] \vdash_{\mathrm{CTC}} e : \hat{\tau}_2{}^{\ell_2}}{\hat{\Gamma} \vdash_{\mathrm{CTC}} \textbf{fn}\ (x \to e) : (\hat{\tau}_1{}^{\ell_1} \to \hat{\tau}_2{}^{\ell_2})^{\ell_3}} \text{ [CT-Fn]}$$

$$\frac{\hat{\Gamma}[f \mapsto (\hat{\tau}_1{}^{\ell_1} \to \hat{\tau}_2{}^{\ell_2})^{\ell_3}][x \mapsto \hat{\tau}_1{}^{\ell_1}] \vdash_{\mathrm{CTC}} e : \hat{\tau}_2{}^{\ell_2}}{\hat{\Gamma} \vdash_{\mathrm{CTC}} \textbf{fun}\ (fx \to e) : (\hat{\tau}_1{}^{\ell_1} \to \hat{\tau}_2{}^{\ell_2})^{\ell_3}} \text{ [CT-Fun]}$$

$$\frac{\hat{\Gamma} \vdash_{\mathrm{CTC}} e_1 : \hat{\tau}_1{}^{\ell_1} \quad \hat{\Gamma}[x \mapsto \hat{\tau}_1{}^{\ell_1}] \vdash_{\mathrm{CTC}} e_2 : \hat{\tau}^{\ell}}{\hat{\Gamma} \vdash_{\mathrm{CTC}} \textbf{let}\ x = e_1\ \textbf{in}\ e_2 : \hat{\tau}^{\ell}} \text{ [CT-Let]}$$

$$\frac{\hat{\Gamma} \vdash_{\mathrm{CTC}} e_1 : (\hat{\tau}_1{}^{\ell_1} \to \hat{\tau}^{\ell_2})^{\ell_3} \quad \hat{\Gamma} \vdash_{\mathrm{CTC}} e_2 : \hat{\tau}_1{}^{\ell_1} \quad \ell_2, \ell_3 \sqsubseteq \ell}{\hat{\Gamma} \vdash_{\mathrm{CTC}} e_1 e_2 : \hat{\tau}^{\ell}} \text{ [CT-App]}$$

$$\frac{\hat{\Gamma} \vdash_{\mathrm{CTC}} e_1 : \mathsf{Bool}^{\ell_1} \quad \hat{\Gamma} \vdash_{\mathrm{CTC}} e_2 : \hat{\tau}^{\ell} \quad \hat{\Gamma} \vdash_{\mathrm{CTC}} e_3 : \hat{\tau}^{\ell} \quad \ell_1 \sqsubseteq \mathrm{L}}{\hat{\Gamma} \vdash_{\mathrm{CTC}} \textbf{if}\ e_1\ \textbf{then}\ e_2\ \textbf{else}\ e_3 : \hat{\tau}^{\ell}} \text{ [CT-If]}$$

$$\frac{\hat{\Gamma} \vdash_{\text{CTC}} e_1 : \hat{\tau}_{\oplus}^{1\,\ell_1} \quad \hat{\Gamma} \vdash_{\text{CTC}} e_2 : \hat{\tau}_{\oplus}^{2\,\ell_2} \quad \ell_1, \ell_2 \sqsubseteq \ell}{\hat{\Gamma} \vdash_{\text{CTC}} e_1 \oplus e_2 : \hat{\tau}_{\oplus}{}^{\ell}} \text{ [CT-Op]}$$

$$\frac{\hat{\Gamma} \vdash_{\text{CTC}} e_1 : \text{Nat}^{\ell} \quad \hat{\Gamma} \vdash_{\text{CTC}} e_2 : \text{Nat}^{\ell} \quad \ell \sqsubseteq \text{L}}{\hat{\Gamma} \vdash_{\text{CTC}} e_1 / e_2 : \text{Nat}^{\ell}} \text{ [CT-Div]}$$

$$\frac{\hat{\Gamma} \vdash_{\text{CTC}} e : \hat{\tau}_1{}^{\ell_1} \quad \hat{\tau}_1{}^{\ell_1} \leqslant \hat{\tau}^{\ell}}{\hat{\Gamma} \vdash_{\text{CTC}} e :: \hat{\tau}^{\ell} : \hat{\tau}^{\ell}} \text{ [CT-Ann]}$$

$$\frac{\hat{\Gamma} \vdash_{\text{CTC}} e_1 : \hat{\tau}_1{}^{\ell_1} \quad \hat{\Gamma} \vdash_{\text{CTC}} e_2 : \hat{\tau}^{\ell}}{\hat{\Gamma} \vdash_{\text{CTC}} e_1 ; e_2 : \hat{\tau}^{\ell}} \text{ [CT-Seq]}$$

$$\frac{\hat{\Gamma} \vdash_{\text{CTC}} e_1 : \text{Nat}^{\ell_1} \quad \hat{\Gamma} \vdash_{\text{CTC}} e_2 : \hat{\tau}^{\ell_2}}{\hat{\Gamma} \vdash_{\text{CTC}} \textbf{array } e_1\ e_2 : (\text{Array } \hat{\tau}^{\ell_2})^{\ell}} \text{ [CT-ArrAlo]}$$

$$\frac{\hat{\Gamma} \vdash_{\text{CTC}} e_1 : (\text{Array } \hat{\tau}^{\ell_1})^{\ell_2} \quad \hat{\Gamma} \vdash_{\text{CTC}} e_2 : \text{Nat}^{\ell_3} \quad \ell_1, \ell_2 \sqsubseteq \ell \quad \ell_3 \sqsubseteq \text{L}}{\hat{\Gamma} \vdash_{\text{CTC}} e_1[e_2] : \hat{\tau}^{\ell}} \text{ [CT-ArrRead]}$$

$$\frac{\hat{\Gamma} \vdash_{\text{CTC}} e_1 : (\text{Array } \hat{\tau}^{\ell_1})^{\ell} \quad \hat{\Gamma} \vdash_{\text{CTC}} e_2 : \text{Nat}^{\ell_2} \quad \hat{\Gamma} \vdash_{\text{CTC}} e_3 : \hat{\tau}^{\ell_3} \quad \ell_3 \sqsubseteq \ell_1 \quad \ell_2 \sqsubseteq \text{L}}{\hat{\Gamma} \vdash_{\text{CTC}} e_1[e_2] = e_3 : (\text{Array } \hat{\tau}^{\ell_1})^{\ell}} \text{ [CT-ArrWrite]}$$

$$\frac{\hat{\Gamma} \vdash_{\text{CTC}} e_1 : \hat{\tau}_1{}^{\ell_1} \quad \hat{\Gamma} \vdash_{\text{CTC}} e_2 : \hat{\tau}_2{}^{\ell_2}}{\hat{\Gamma} \vdash_{\text{CTC}} (e_1, e_2) : (\hat{\tau}_1{}^{\ell_1}, \hat{\tau}_2{}^{\ell_2})^{\ell}} \text{ [CT-Pair]}$$

$$\frac{\hat{\Gamma} \vdash_{\text{CTC}} e_1 : (\hat{\tau}_1{}^{\ell_1}, \hat{\tau}_2{}^{\ell_2})^{\ell_3} \quad \hat{\Gamma}[x \mapsto \hat{\tau}_1{}^{\ell_1}][y \mapsto \hat{\tau}_2{}^{\ell_2}] \vdash_{\text{CTC}} e_2 : \hat{\tau}^{\ell} \quad \ell_1, \ell_2, \ell_3 \sqsubseteq \ell}{\hat{\Gamma} \vdash_{\text{CTC}} \textbf{case } e_1 \textbf{ of } (x, y) \rightarrow e_2 : \hat{\tau}^{\ell}} \text{ [CT-CasePair]}$$

$$\frac{}{\hat{\Gamma} \vdash_{\text{CTC}} [] : (\text{List } \hat{\tau}_1{}^{\ell_1})^{\ell}} \text{ [CT-EmpList]}$$

$$\frac{\hat{\Gamma} \vdash_{\text{CTC}} e_1 : \hat{\tau}_1{}^{\ell_1} \quad \hat{\Gamma} \vdash_{\text{CTC}} e_2 : (\text{List } \hat{\tau}_1{}^{\ell_1})^{\ell}}{\hat{\Gamma} \vdash_{\text{CTC}} e1 : e2 : (\text{List } \hat{\tau}_1{}^{\ell_1})^{\ell}} \text{ [CT-Cons]}$$

$$\frac{\hat{\Gamma} \vdash_{\text{CTC}} e_1 : (\text{List } \hat{\tau}_1{}^{\ell_1})^{\ell_2} \quad \hat{\Gamma} \vdash_{\text{CTC}} e_2 : \hat{\tau}^{\ell} \quad \hat{\Gamma}[x \mapsto \hat{\tau}_1{}^{\ell_1}][y \mapsto (\text{List } \hat{\tau}_1{}^{\ell_1})^{\ell_2}] \vdash_{\text{CTC}} e_3 : \hat{\tau}^{\ell} \quad \ell_1, \ell_2 \sqsubseteq \text{L}}{\hat{\Gamma} \vdash_{\text{CTC}} \textbf{case } e_1 \textbf{ of } [] \mapsto e_2, x : y \mapsto e_3 : \hat{\tau}^{\ell}}$$

[CT-CaseList]

2.2 Subtyping Rules

$$\frac{}{\text{L} \sqsubseteq \text{L}} \text{ [CT-ST-Low]}$$

$$\frac{}{\text{H} \sqsubseteq \text{H}} \text{ [CT-ST-High]}$$

$$\frac{}{\text{L} \sqsubseteq \text{H}} \text{ [CT-ST-LowHigh]}$$

$$\frac{\ell_1 \sqsubseteq \ell_2}{\text{Nat}^{\ell_1} \leqslant \text{Nat}^{\ell_2}} \text{ [CT-ST-Nat]}$$

$$\frac{\ell_1 \sqsubseteq \ell_2}{\text{Bool}^{\ell_1} \leqslant \text{Bool}^{\ell_2}} \text{ [CT-ST-Bool]}$$

$$\frac{\hat{\tau}_1'^{\ell_1'} \leqslant \hat{\tau}_1{}^{\ell_1} \quad \hat{\tau}_2{}^{\ell_2} \leqslant \hat{\tau}_2'^{\ell_2'} \quad \ell_3 \sqsubseteq \ell_3'}{(\hat{\tau}_1{}^{\ell_1} \rightarrow \hat{\tau}_2{}^{\ell_2})^{\ell_3} \leqslant (\hat{\tau}_1'^{\ell_1'} \rightarrow \hat{\tau}_2'^{\ell_2'})^{\ell_3'}} \text{ [CT-ST-Fun]}$$

$$\frac{\ell_2 \sqsubseteq \ell_2'}{(\text{Array } \hat{\tau}^{\ell_1})^{\ell_2} \leqslant (\text{Array } \hat{\tau}^{\ell_1})^{\ell_2'}} \text{ [CT-ST-Arr]}$$

$$\frac{\hat{\tau}_1{}^{\ell_1} \leqslant \hat{\tau}_1'{}^{\ell_1'} \quad \hat{\tau}_2{}^{\ell_2} \leqslant \hat{\tau}_2'{}^{\ell_2'} \quad \ell_3 \sqsubseteq \ell_3'}{(\hat{\tau}_1{}^{\ell_1}, \hat{\tau}_2{}^{\ell_2})^{\ell_3} \leqslant (\hat{\tau}_1'{}^{\ell_1'}, \hat{\tau}_2'{}^{\ell_2'})^{\ell_3'}} \text{ [CT-T-Pair]}$$

$$\frac{\hat{\tau}^{\ell} \leqslant \hat{\tau}'^{\ell'} \quad \ell_1 \sqsubseteq \ell_1'}{(\text{List } \hat{\tau}^{\ell})^{\ell_1} \leqslant (\text{List } \hat{\tau}'^{\ell'})^{\ell_1'}} \text{ [CT-ST-List]}$$

$$\frac{\ell \sqsubseteq \ell'}{\alpha^{\ell} \leqslant \alpha^{\ell'}} \text{ [CT-ST-TypVar]}$$

3 Examples

In this section, we provide code examples (including legal ones, which will be successfully compiled; and illegal ones, which will be reported as an error).

3.1 List

3.1.1 Pass

```
(1 : []) :: (List Nat^H)^H
```

```
let xs = (1 :: Nat^H) : 2 : [] in case xs of y : ys -> y, [] -> 1
```

3.1.2 Fail

```
let l = (1 : []) :: (List Nat^H)^H in case l of x : xs -> 1, [] -> 2
```

3.2 Array

3.3 Pass

```
let xs = (array 10 0 :: a0^H) in (xs, xs[0])
```

```
let xs = (array 10 0 :: a0^H) in xs[0] = xs[1]
```

```
let xs = (array 10 0 :: a0^H) in xs[0] = 1
```

3.3.1 Fail (violates constant time)

```
let len = 1 :: Nat^H in array len 0
```

```
let xs = (array 10 0 :: a0^H) in xs[0] / 3
```

```
let xs = (array 10 0 :: (Array Nat^L)^L) in xs[0] = (1 :: Nat^H)
```

3.4 Pair

3.4.1 Pass

```
(1, 2) :: (Nat^H, Nat^L)^H
```

```
let p = (1, 2) :: a0^H in case p of (x, y) -> x+1
```

3.4.2 Fail

```
let p = (1, 2) :: a0^H in case p of (x, y) -> x/3
```

```
let gx = (fn p -> case p of (x, y) -> x / 3) in gx ((1, 2) :: (Nat^H, Nat^H)^L)
```

3.5 Subeffecting & Subtyping

3.5.1 Subeffecting

```
let add = (fn x -> x + 1) :: (Nat^L -> Nat^H)^L in add 1
```

```
fn a -> if true then a else false :: Bool^H
```

```
fun f x -> if x < 1 then 0 :: Nat^H else 1 + f (x - 1)
```

3.5.2 Subtyping

```
(fn x -> fn y -> (x, y) :: a0^H) :: (Nat^b0 -> (Nat -> (Nat, Nat)^b0))
```

```
(3 :: Nat^b0) / 1
```

```
let p = (1, 2) :: a0^b0 in case p of (x, y) -> x/3
```