

MENG INDIVIDUAL PROJECT

DEPARTMENT OF COMPUTING

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

Ochre: A Dependently Typed Systems Programming Language

Author:
Charlie Lidbury

Supervisor(s):
Steffen van Bakel
Nicolas Wu

June 24, 2024

```
1 Bool = 'true | 'false;
2 Pair = (Bool, Bool);
3
4 overwrite = (p: &mut Pair) {
5     *p.0 = 'true;
6     *p.1 = 'true;
7 };
8
9 pair = ('false, 'false);
10 overwrite(&mut pair);
```

```
1 type Pair = (bool, bool);
2
3 fn overwrite(p: &mut Pair) {
4     *p = ('true, 'true);
5 }
6
7 fn main() {
8     let mut pair = (false, false);
9     overwrite(&mut pair);
10 }
```

```
1 Bool = 'true | 'false;
2 Same = (Bool, L -> L);
3
4 overwrite = (p: &mut Same) {
5     *p = ('true, 'true);
6 };
7
8 pair = ('false, 'false);
9 overwrite(&mut pair);
```

```
1 type Same = | MkSame: b:bool -> b2:bool{b == b2} -> Same
2
3 val overwrite: p:ptr Same -> Stack unit
4 let overwrite p =
5     p := Same true true
6
7 let main () : Stack unit =
8     let pair = alloc (MkSame false false) in
9     overwrite pair;
10     ()
```

$$\frac{\langle \text{def.} \Rightarrow \text{for } x \rangle \quad \Omega' = \Omega \left[\frac{x \mapsto \top}{x \mapsto v} \right]}{\Omega \vdash x \Rightarrow m \dashv \Omega'}$$

Figure 1: Reading removes a value from the environment, whereas writing adds a value.

```

1                                     //  $\Omega_0 = \emptyset$ 
2 Same = ('a | 'b, L -> L); //  $\Omega_1 = \emptyset, \text{Same} \mapsto (\{ 'a, 'b \}, L \rightarrow L)$ 
3
4 overwrite = (p: &mut Same) -> 'unit {
5     //  $\Omega_{10} = \Omega_1, p \mapsto \text{borrow}^m l(\{ 'a, 'b \}, L \rightarrow L), l \mapsto (\{ 'a, 'b \}, L \rightarrow L)$ 
6     (*p).0 = 'a; //  $\Omega_{11} = \Omega_1, p \mapsto \text{borrow}^m l('a, \_ \rightarrow \{ 'a, 'b \}), l \mapsto (\{ 'a, 'b \}, L \rightarrow L)$ 
7     (*p).1 = 'a; //  $\Omega_{12} = \Omega_1, p \mapsto \text{borrow}^m l('a, \_ \rightarrow 'a), l \mapsto (\{ 'a, 'b \}, L \rightarrow L)$ 
8     'unit //  $\Omega_{12} \vdash \text{drop}$ 
9 } //  $\Omega_2 = \Omega_1, \text{overwrite} \mapsto (p: \&\text{mut Same}) \rightarrow 'unit$ 
10
11 pair = ('b, 'b); //  $\Omega_3 = \Omega_2, \text{pair} \mapsto ('b, \_ \rightarrow 'b)$ 
12 overwrite(&mut pair); //  $\Omega_4 = \Omega_2, \text{pair} \mapsto (\{ 'a, 'b \}, L \rightarrow L)$ 
13
14 pair //  $(\{ 'a, 'b \}, L \rightarrow L)$ 

```

```

1 Bool = 'true | 'false;
2 Same = (Bool, L -> L);
3
4 overwrite = (p: &mut Same) {
5
6
7     *p.0 = 'true;
8     *p.1 = 'true;
9
10 };

```

$$\{x \mapsto \{ 'five \}\} \vdash x \Rightarrow \{ 'five \} \dashv \{x \mapsto \top\}$$

$$\Omega \vdash M \diamond t \dashv \Omega'$$

$$\left\{ \begin{array}{l} \Rightarrow, \rightarrow, \rightsquigarrow, \\ \Leftarrow, \leftarrow, \leftrightsquigarrow \end{array} \right\}$$