1 Lecture

1.1 Monads

Warning: the following is not proper math, it serves as an illustration only!

$$
\begin{aligned}
f &: A \to B && (1)\\
sf &: (A \times S) \to (B \times S) && (2)\\
sf &: A \to S \to (B \times S) && (3)\\
sf(param) &: S \to (B \times S) && (4)\\
sf(param) &: Action\langle S, B\rangle && (5)
\end{aligned}
$$

$$
\begin{aligned}
\circ &: \forall A, B, C.(A \to B) \to (B \to C) \to (A \to C) && (6)\\
\circ(g, f) &= \lambda x.f(g(x)) && (7)
\end{aligned}
$$

$$
\begin{aligned}
\bigcirc &: \forall A, B, C, S.(A \to Action\langle S, B\rangle) \to (B \to Action\langle S, C\rangle) \to (A \to Action\langle S, C\rangle) && (8)\\
\bigcirc(g, f) &= \lambda x.g(x) \odot f && (9)
\end{aligned}
$$

$$
\begin{aligned}
\odot &: \forall B, C, S.Action\langle S, B\rangle \to (B \to Action\langle S, C\rangle) \to Action\langle S, C\rangle && (10)\\
x \odot y &= \lambda s.y(b)(s') \quad \text{where } (b, s') = x(s) && (11)
\end{aligned}
$$

$$
\begin{aligned}
u &: A \to Action\langle S, B\rangle && (12)\\
v &: B \to Action\langle S, C\rangle && (13)\\
w &: A \to Action\langle S, C\rangle && (14)\\
(u \bigcirc v \bigcirc w)(x) &= u(x) \odot \lambda x'.v(x') \odot \lambda x''.w(x'') && (15)\\
(u \bigcirc v \bigcirc w)(x) &= \text{do } x' \leftarrow u(x)\\
&\quad\ x'' \leftarrow v(x') && (16)\\
&\quad\ w(x'')
\end{aligned}
$$

1.2 Dynamic Locking

$$
\begin{aligned}
lock(key, value) &= \lambda k.\text{if } k = key \text{ then } value && (17)\\
unlock(key, lockedValue) &= lockedValue(key) && (18)
\end{aligned}
$$

1.3 Existential Type

$$
\begin{aligned}
\exists Point.Record(create &: Num \times Num \to Point,\\
shift &: Point \times Num \times Num \to Point, && (19)\\
equal &: Point \times Point \to Bool)
\end{aligned}
$$

The above-defined type will be from now on referred to as PI.

$$
\frac{\Gamma \cup \{X\} \vdash A}{\Gamma \vdash \exists X.A} \qquad (20)
$$

$$PM = \text{pack}_{PI}\ Point = Num \times Num$$
$$\text{with } record(create = \lambda x : Num, y : Num.(x, y),$$
$$shift = \lambda p : Point, x : Num, y : Num.(\pi_1(p) + x, \pi_2(p) + y),$$
$$equal = \lambda p_1 : Point, p_2 : Point : p_1 = p_2) \tag{21}$$

The above-defined module will be from now on referred to as PM.

$$\frac{\Gamma \vdash M[X \mapsto B] : A[X \mapsto B]}{\Gamma \vdash \text{pack}_{\exists X.A} X = B \text{ with } M : \exists X.A} \tag{22}$$

$$equals(create(1, 2), shift(create(0, 0), 1, 2)) : Bool \tag{23}$$

$$\text{open}_{Bool} PM \text{ as } Point, ops : Record(create : \ldots, shift : \ldots, equals : \ldots)$$
$$\text{in } ops.equals(ops.create(1, 2), ops.shift(ops.create(0, 0), 1, 2)) \tag{24}$$

$$\frac{\Gamma \vdash M : \exists X.A \quad \Gamma \cup \{X, x : A\} \vdash N : B \quad \Gamma \vdash B}{\Gamma \vdash \text{open}_B\ M \text{ as } X, x : A \text{ in } N : B} \tag{25}$$

2 Seminar

1. Look at the parser code and extend it according to your taste.