

1. Napište výraz lambda kalkulu implementující funkci  $f$ . Pro přehlednost používejte symbolická jména standardních kombinátorů (např. 0) a jejich definice rozepište vedle. Náповěda: nejprve zkonstruujte generátor funkce  $f$ .

$$f(1, x, y) = x(1, y(0)) \tag{1a}$$

$$f(n, x, y) = x(n, f(n-1, x, y)) \quad \text{pro } n > 1 \tag{1b}$$

---

2. V jazyce Featherweight Java formálně odvoďte, že  $\text{new } X().g(\text{new } X().f()) \longrightarrow^* \text{new } \text{Object}()$ . Při každém redukčním kroku napište nad šipku název použitého redukčního pravidla.

```
class X extends Object {
  X() { super(); }
  X f() { return this.f(); }
  Object g(X x) { return new Object(); }
}
```

---

3. V jazyce Featherweight Java formálně odvoďte, že třída  $X$  z předchozího příkladu je správně otypovaná. Vedle každé “zlomkové” čáry napište název použitého odvozovacího pravidla.

---

4. Relace  $\longrightarrow$  ve Featherweight Javě definuje sémantiku malého kroku. Předpokládejte, že relace  $\Longrightarrow$  definuje ekvivalentní sémantiku velkého kroku a dopište následující pravidlo:

$$\frac{}{e_0.m(e_1, \dots, e_n) \Longrightarrow} \tag{2}$$

---

5. Vyhodnocování výrazů ve Featherweight Javě se může za určitých okolností zaseknout. Rozšířte FJ o hodnotu **error** tak, aby k zaseknutí nedocházelo. Neměňte existující pravidla typového systému a sémantiky, pouze přidávejte nová.