1. Napište co nejobecnější typ funkce f. Nápověda: pokud nedovedete určit přesný typ některé proměnné, použijte parametrický polymorfismus. Měli byste nějak reflektovat to, že některé proměnné se používají jako funkce a některé jako jejich parametry.

$$f(w, x, y, z) = \text{if } w(x) = y(z) \text{ then } w(x) \text{ else } f(w, w(z), y, y(x)) \tag{1}$$

---

2. Akce je funkce, která v parametru dostane stav a buď proběhne v pořádku a vrátí nový stav nebo selže a vrátí $\perp$.

$$Action = State \to (State \cup \{\perp\}), \quad \perp \notin State \tag{2}$$

Napište definici funkce $runInTransaction : Action^* \to (State \to State)$, která vrátí funkci, která buď zřetězí všechny akce a vrátí výsledný stav nebo, pokud některá z akcí v průběhu vykonávání selže, vrátí svůj počáteční stav. Nápověda: možná vám pomůže nadefinovat si pomocnou funkci.

---

3. Sémantika definovaná přepisovací relací $\to$ postupně bere instrukce ze zásobníku nalevo a manipuluje se zásobníkem hodnot napravo. Bohužel, pokud se programátor pokusí o operace typu součet dvou logických hodnot, negace neexistující hodnoty apod., sémantika se zasekne. Vaším úkolem je dodat do jazyka hodnotu $Error$ a upravit relaci $\to$ tak, aby místo zaseknutí doběhla do konfigurace s nulovým počtem nezpracovaných instrukcí a na vrcholu zásobníku vrátila $Error$.

$$Instruction ::= Num \mid Bool \mid \text{inc} \mid \text{if} \tag{3}$$

Množina hodnot $Value$ je definovaná jako $Value = Bool \cup Num$. Množina konfigurací je $Instruction^* \times Value^*$. Použité jmenné konvence: $n, n' \in Num$, $b, b' \in Bool$, $v_1, \cdots \in Value$ a $i_1, \cdots \in Instruction$.

$$\langle n, i_1, \ldots, i_a\rangle, \langle v_1, \ldots, v_b\rangle \to \langle i_1, \ldots, i_a\rangle, \langle n, v_1, \ldots, v_b\rangle \tag{4}$$

$$\langle b, i_1, \ldots, i_a\rangle, \langle v_1, \ldots, v_b\rangle \to \langle i_1, \ldots, i_a\rangle, \langle b, v_1, \ldots, v_b\rangle \tag{5}$$

$$\langle \text{inc}, i_1, \ldots, i_a\rangle, \langle n, v_1, \ldots, v_b\rangle \to \langle i_1, \ldots, i_a\rangle, \langle n+1, v_1, \ldots, v_b\rangle \tag{6}$$

$$\langle \text{if}, i_1, \ldots, i_a\rangle, \langle true, n, n', v_1, \ldots, v_b\rangle \to \langle i_1, \ldots, i_a\rangle, \langle n, v_1, \ldots, v_b\rangle \tag{7}$$

$$\langle \text{if}, i_1, \ldots, i_a\rangle, \langle false, n, n', v_1, \ldots, v_b\rangle \to \langle i_1, \ldots, i_a\rangle, \langle n', v_1, \ldots, v_b\rangle \tag{8}$$

4. Nakreslete graf znázorňující všechny způsoby přepsání konfigurace $\langle 3, 1, 2\rangle, \langle 0, 2, 5\rangle$ na normální formu.

$$\begin{aligned} &\langle a_1, \ldots, a_n\rangle, \langle b_1, \ldots, b_m\rangle \to \\ &\langle a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n\rangle, \langle b_1, \ldots, b_j, a_i, b_{j+1}, \ldots, b_m\rangle \\ &\text{kde } i \in \{1, \ldots, n\} \wedge b_j \le a_i \le b_{j+1} \end{aligned} \tag{9}$$

---

5. Svými slovy (ale přesně a srozumitelně) vysvětlete co konkrétně ve Featherweight Javě kontroluje typové pravidlo T-Method.