Uvažujte jednoduchý embedded systém, ve kterém je procesor se třemi registry (dva obecné, jeden čítač aktuální adresy) a virtuálně nekonečný počet buněk paměti. V jedné buňce může být zapsána instrukce (komplet, včetně operandů) nebo číslo, v jednom registru může být zapsáno jen číslo (N označuje přirozená čísla):

$$\begin{aligned}
\mathit{Cell} \to\ & N \mid \\
& \text{HALT} \mid \\
& \text{CONST } \mathit{Reg}, N \mid \\
& \text{LOAD } \mathit{Reg}, N \mid \\
& \text{STORE } \mathit{Reg}, N \mid \\
& \text{ADD} \mid \\
& \text{SUBTR} \mid \\
& \text{COND } N \\
\mathit{Reg} \to\ & \text{A} \mid \text{B}
\end{aligned}$$

kde N reprezentuje přirozená čísla.

Konfigurace operační sémantiky tohoto stroje je čtveřice $N \times N \times N \times (N \to \mathit{Cell})$, kde první složka reprezentuje obsah obecného registru A, druhá obsah obecného registru B, třetí obsah čítače aktuální adresy a čtvrtá složka reprezentuje paměť jako funkci z indexu do obsahu buňky na daném indexu. Nad množinou všech konfigurací je nadefinována přepisovací relace $\leadsto$ $(a, b, i, m, n, n' \in N, \mathit{mem} \in N \to \mathit{Cell})$:

$$\frac{\mathit{mem}(i) = \text{CONST A}, n}{(a, b, i, \mathit{mem}) \leadsto (n, b, i+1, \mathit{mem})}$$

$$\frac{\mathit{mem}(i) = \text{CONST B}, n}{(a, b, i, \mathit{mem}) \leadsto (a, n, i+1, \mathit{mem})}$$

$$\frac{\mathit{mem}(i) = \text{LOAD A}, n \quad \mathit{mem}(n) = n'}{(a, b, i, \mathit{mem}) \leadsto (n', b, i+1, \mathit{mem})}$$

$$\frac{\mathit{mem}(i) = \text{LOAD B}, n \quad \mathit{mem}(n) = n'}{(a, b, i, \mathit{mem}) \leadsto (a, n', i+1, \mathit{mem})}$$

$$\frac{\mathit{mem}(i) = \text{STORE A}, n}{(a, b, i, \mathit{mem}) \leadsto (a, b, i+1, \lambda m.\text{if } m = n \text{ then } a \text{ else } \mathit{mem}(m) \text{ fi})}$$

$$\frac{\mathit{mem}(i) = \text{STORE B}, n}{(a, b, i, \mathit{mem}) \leadsto (a, b, i+1, \lambda m.\text{if } m = n \text{ then } b \text{ else } \mathit{mem}(m) \text{ fi})}$$

$$\frac{\mathit{mem}(i) = \text{ADD}}{(a, b, i, \mathit{mem}) \leadsto (a+b, 0, i+1, \mathit{mem})}$$

$$\frac{\mathit{mem}(i) = \text{SUBTR} \quad a > b}{(a, b, i, \mathit{mem}) \leadsto (a-b, 0, i+1, \mathit{mem})}$$

$$\frac{\mathit{mem}(i) = \text{SUBTR} \quad a \le b}{(a, b, i, \mathit{mem}) \leadsto (0, 0, i+1, \mathit{mem})}$$

$$\frac{mem(i) = \text{COND } n \quad a = b}{(a, b, i, mem) \rightsquigarrow (a, b, n, mem)}$$

$$\frac{mem(i) = \text{COND } n \quad a \neq b}{(a, b, i, mem) \rightsquigarrow (a, b, i + 1, mem)}$$

Množina finálních konfigurací je $\{(a, b, i, mem) \in N \times N \times N \times (N \to Cell) | mem(i) = \text{HALT}\}$, vstupní funkce $input$ je definována jako $input(a, b, mem) = (a, b, 0, mem)$, výstupní funkce $output$ je definována jako $output((a, b, i, mem)) = (a, b)$.

1. Napište příklad vstupu této sémantiky, který nepovede do finální konfigurace a vysvětlete proč.

   Např. $(0, 0, \lambda n.0)$ — abstraktní stroj se zasekne hned na své počáteční konfiguraci, protože neumí zpracovávat čísla jako instrukce.

2. Přidejte do sémantiky odvozovací pravidlo/a, které nadefinuje instrukci CLOAD n. Ta nahraje obsah buňky na indexu n do registru A, pokud je obsah registru A menší než obsah registru B a neudělá nic v opačném případě.

$$\frac{mem(i) = \text{CLOAD}, n \quad a < b \quad mem(n) = n'}{(a, b, i, mem) \rightsquigarrow (n', b, i + 1, mem)}$$

$$\frac{mem(i) = \text{CLOAD}, n \quad a \geq b}{(a, b, i, mem) \rightsquigarrow (a, n', i + 1, mem)}$$

3. Prvních šest pravidel (tzn. definice instrukcí CONST, LOAD, STORE) předělejte na sémantiku velkého kroku.

$$\frac{mem(i) = \text{HALT}}{(a, b, i, mem) \rightsquigarrow (a, b, i, mem)}$$

$$\frac{mem(i) = \text{CONST A}, n \quad (n, b, i + 1, mem) \rightsquigarrow (a', b', i', mem') \quad mem'(i') = \text{HALT}}{(a, b, i, mem) \rightsquigarrow (a', b', i', mem')}$$

$$\frac{mem(i) = \text{CONST B}, n \quad (a, n, i + 1, mem) \rightsquigarrow (a', b', i', mem') \quad mem'(i') = \text{HALT}}{(a, b, i, mem) \rightsquigarrow (a', b', i', mem')}$$

$$\frac{mem(i) = \text{LOAD A}, n \quad mem(n) = n' \quad (n', b, i + 1, mem) \rightsquigarrow (a', b', i', mem') \quad mem'(i') = \text{HALT}}{(a, b, i, mem) \rightsquigarrow (a', b', i', mem')}$$

$$\frac{mem(i) = \text{LOAD B}, n \quad mem(n) = n' \quad (a, n', i + 1, mem) \rightsquigarrow (a', b', i', mem') \quad mem'(i') = \text{HALT}}{(a, b, i, mem) \rightsquigarrow (a', b', i', mem')}$$

$$\frac{mem(i) = \text{STORE A}, n \quad (a, b, i + 1, \lambda m.\text{if } m = n \text{ then } a \text{ else } mem(m) \text{ fi}) \rightsquigarrow (a', b', i', mem') \quad mem'(i}{(a, b, i, mem) \rightsquigarrow (a', b', i', mem')}$$

$$\frac{mem(i) = \text{STORE B}, n \quad (a, b, i + 1, \lambda m.\text{if } m = n \text{ then } b \text{ else } mem(m) \text{ fi}) \rightsquigarrow (a', b', i', mem') \quad mem'(i'}{(a, b, i, mem) \rightsquigarrow (a', b', i', mem')}$$