# Solutions

**1** The easy solution is to introduce a COND instruction that assumes that the values of all three expressions are on the stack. Then

$$\mathcal{C}[\![e?e_1 : e_2]\!] = \mathcal{C}[\![e_2]\!] : \mathcal{C}[\![e_1]\!] : \mathcal{C}[\![e]\!] : COND$$

If just want to evaluate one of the expressions we have store the code for both of them on the stack using a instruction $COND(c_1, c_2)$ and choose to evaluate the correct one using CHOOSE

$$\mathcal{C}[\![e?e_1 : e_2]\!] = COND(\mathcal{C}[\![e_1]\!], \mathcal{C}[\![e_2]\!]) : \mathcal{C}[\![e]\!] : CHOOSE$$

**2**

$$< n : n_1 : n_2 : s, env, COND : c, d > \triangleright < r : s, env, c, d >$$

where $r = n_1$ if $n = 0$ and $r = n_1$ otherwise.

With the other solution we have add a pair of instructions to possible values on the stack

$$Value = \mathbb{Z} \cup (Var \times Code \times Env) \cup (Code \times Code)$$

$$< s, env, COND(c_1, c_2) : c, d > \triangleright < (c_1, c_2) : s, env, c, d >$$
$$< n : (c_1, c_2) : s, env, CHOOSE : c, d > \triangleright < s, env, c_0 : c, d >$$

where $c_0$ is $c_1$ if $n = 0$ and $c_2$ otherwise.

**3** $\mathcal{S}_{ds}[\![\texttt{read x}]\!](\sigma, i) = (\sigma[x \mapsto head\ i], tail\ i)$

**4** Let $g_0 = \perp$ and $g_{i+1} = F\ g_i$.

$$g_1 = \lambda\ \sigma.(\mathcal{B}[\![\texttt{b}]\!]\sigma ? \mathcal{S}_{ds}[\![S]\!](g_0\ \sigma) : \sigma) = \begin{cases} \lambda\sigma.\sigma & \text{if } \mathcal{B}[\![\texttt{b}]\!]\sigma = \mathbf{ff} \\ \text{undefined} & \text{otherwise} \end{cases}$$

and $g_i = g_1$ for all $i > 1$. Thus $\bigsqcup\{g_i \mid \in \mathbb{N}\} = g_1$

**5** $\mathcal{A}[\![\text{let } x = a_0 \text{ in } a_1]\!]\sigma = \mathcal{A}[\![a_1]\!](\sigma[x \mapsto \mathcal{A}[\![a_0]\!])$

**6** **a.** $\emptyset$, $\{\emptyset\}$,$\{\emptyset, \{\emptyset\}\}$, $\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$, $\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}\}$

**b.** We denote element number $i$ by $c_i$. We observe that $c_i \subseteq c_j$ when $i \leq j$. Since $F(c_i) = c_{i+1}$ it follows that $F(c_i) \subseteq F(c_j)$ when $c_i \subseteq c_j$.

**c.** It is not a ccpo since $\bigcup N = N$ does not belong to N. However $D = n \cup \{N\}$, is a ccpo.

**d.** $F$ is not a function in $D \to D$ since $F(N) = D$ which does not belong to $D$, so the answer is no.

**e.** N is a representation for $\mathbb{N}$, F is the successor function.