Recall that we want to define

$$S_n(e,\overline{x},S(z)) = \begin{cases} h_1(e,\overline{x},z,S_n(e,\overline{x},z)) & \text{if conf. encoded by } S_n(e,\overline{x},z) \text{ requires increment} \\ h_2(e,\overline{x},z,S_n(e,\overline{x},z)) & \text{if conf. encoded by } S_n(e,\overline{x},z) \text{ requires decrement} \\ h_3(e,\overline{x},z,S_n(e,\overline{x},z)) & \text{if conf. encoded by } S_n(e,\overline{x},z) \text{ requires jump} \\ 0 & \text{otherwise.} \end{cases}$$

The scheme of case distinction is primitive recursive, if the condition guards are primitive recursive and the functions h_i are primitive recursive.

We will need to determine the code of the next instruction to be executed, which is given by

$$\ell = \langle e \rangle_{\langle S_n(e,\overline{x},z)\rangle_0}$$

The configuration encoded by $S_n(e,\overline{x},z)$ requires..

- .. an increment, if $[\ell]_1 = 0$ and $[\ell]_2 > 0$. In this case, register $X_{[\ell]_2}$ and the line number have to be incremented.
- .. a decrement, if $[\ell]_1 > 0$ and $[\ell]_2 = 0$. In this case, register $X_{[\ell]_1}$ has to be decremented and the line number has to be incremented.
- .. a jump, if $[\ell]_1 > 0$ and $[\ell]_2 > 0$. In this case, the line number has to be set to $[\ell]_2$, if the current value of register $X_{[\ell]_1}$ is greater than zero, and the line number is incremented otherwise (the registers stay unchanged).

Thus, the guards are primitive recursive, as $[\cdot]_1$ and $[\cdot]_2$ are primitive recursive.

Now, we use the following functions to manipulate sequences encoded by $\langle\cdot\rangle$:

- $\mathrm{inc}\colon \mathbb{N}^2 \to \mathbb{N}$ with $\mathrm{inc}(0,i) = 0$ and

  $$\mathrm{inc}(\langle x_0,\dots,x_k\rangle, i) = \begin{cases} \langle x_0,\dots,x_{i-1},x_i+1,x_{i+1},\dots,x_k\rangle & \text{if } i < k+1, \\ 0 & \text{otherwise.} \end{cases}$$

  We have

  $$\mathrm{inc}(x,i) = \chi_<(0,x)\cdot\chi_<(i,\mathrm{len}(x))\cdot((x+1)\cdot p(i) - 1)$$

  where p returns the i-th prime number on input i.

- $\mathrm{dec}\colon \mathbb{N}^2 \to \mathbb{N}$ with $\mathrm{inc}(0,i) = 0$ and

  $$\mathrm{dec}(\langle x_0,\dots,x_k\rangle, i) = \begin{cases} \langle x_0,\dots,x_{i-1},x_i \dot{-} 1,x_{i+1},\dots,x_k\rangle & \text{if } i < k+1, \\ 0 & \text{otherwise.} \end{cases}$$

  Note that we use $\dot{-}$, which means we cannot just divide $x+1$ by $p(i)$ to decrement the entry in coordinate i, we first have to check that it is greater than zero. Also we have to pay attention to the fact that the exponent corresponding to the last entry of the sequence is incremented, hence, we have to treat it specially (in this case, $\chi_=(i, \mathrm{len}(x)-1)$ evaluates to 1). Thus, we have

$$\begin{aligned}\mathrm{dec}(x,i) = \chi_<(0,x)\cdot\chi_<(i,\mathrm{len}(x))\cdot \\ \Big(\chi_<(\chi_=(i,\mathrm{len}(x)-1),(x+1)_i)\cdot\big(\mathrm{div}(x+1,p(i))-1\big) + \\ \chi_=(\chi_=(i,\mathrm{len}(x)-1),(x+1)_i)\cdot x\Big)\end{aligned}$$

where $(x)_y$ is the power of the i-th prime number in the prime factorization of x (see problem set 4), and where div is the division function (see problem set 3).

- upd: $\mathbb{N}^3 \to \mathbb{N}$ with $\mathrm{inc}(0,i,v) = 0$ and

$$\mathrm{upd}(\langle x_0,\ldots,x_k\rangle,i,v) = \begin{cases}\langle x_0,\ldots,x_{i-1},v,x_{i+1},\ldots,x_k\rangle & \text{if } i < k+1,\\ 0 & \text{otherwise.}\end{cases}$$

We have

$$\mathrm{upd}(x,i,v) = \chi_<(0,x)\cdot\chi_<(i,\mathrm{len}(x))\cdot[(\mathrm{div}(x+1,p(i)^{(x+1)_i+\chi_=(i,\mathrm{len}(x)-1)}))\cdot p(i)^v - 1].$$

Hence, inc, dec, and upd are primitive recursive.

Now, we can define

$$h_1(e,\overline{x},z,S_n(e,\overline{x},z)) = \mathrm{inc}(\mathrm{inc}(S_n(e,\overline{x},z),[\ell]_2),0)$$

$$h_1(e,\overline{x},z,S_n(e,\overline{x},z)) = \mathrm{inc}(\mathrm{dec}(S_n(e,\overline{x},z),[\ell]_1),0)$$

$$\begin{aligned}h_3(e,\overline{x},z,S_n(e,\overline{x},z)) = {}& \chi_<(0,\langle S_n(e,\overline{x},z)\rangle_{[\ell]_1})\cdot\mathrm{upd}(S_n(e,\overline{x},z),0,[\ell]_2) + \\ & \chi_=(0,\langle S_n(e,\overline{x},z)\rangle_{[\ell]_1})\cdot\mathrm{inc}(S_n(e,\overline{x},z),0)\end{aligned}$$

where $\ell = \langle e\rangle_{\langle S_n(e,\overline{x},z)\rangle_0}$ is still the code of the next instruction that has to be executed.