$v_1 = \lambda\_:() \to \text{else}\ (\text{send}\ ()\ w)$

$s_2 = \text{let}\ (x, a) = \text{receive}\ a\ \text{in}\ \text{Wait}\ a$

$s = \text{let}\ (w, a) = \text{new}\ !().?().\text{end}!\ \text{in}\ s_2; (\text{fork}\ v_1); x$

Type Derivations

$$\frac{\dagger\text{-Const}}{\vdash \text{send} : () \to (!().\text{end}!) \to \text{end}!} \qquad \frac{}{() : ()}\ \dagger\text{-Const}$$

$$\frac{\vdash \text{send} : () \to (!().\text{end}!) \to \text{end}! \qquad () : ()}{\vdash \text{send}\ () : (!().\text{end}!) \to \text{end}!}\ \dagger\text{-APP}$$

$$\frac{}{w : !().\text{end}! \vdash w : !().\text{end}!}\ \dagger\text{-VAR}$$

$$\frac{\vdash \text{send}\ () : (!().\text{end}!) \to \text{end}! \qquad w : !().\text{end}! \vdash w : !().\text{end}!}{w : !().\text{end}! \vdash \text{send}\ ()\ w : \text{end}!}\ \dagger\text{-APP}$$

$$\frac{\dagger\text{-Const}}{\vdash \text{else} : \text{end}! \to ()} \qquad \frac{w : !().\text{end}! \vdash \text{send}\ ()\ w : \text{end}!}{w : !().\text{end}! \vdash \text{else}\ (\text{send}\ ()\ w) : ()}\ \dagger\text{-APP}$$

$$\frac{w : !().\text{end}! \vdash \text{else}\ (\text{send}\ ()\ w) : ()}{** \; w : !().\text{end}! \vdash \lambda\_:() \to \text{else}\ (\text{send}\ ()\ w) : () \to ()}\ \dagger\text{-ABS}$$

$$\frac{}{x : () \vdash x : ()}\ \dagger\text{-Var} \qquad \frac{}{\vdash \text{fork} : (() \to ()) \to ()}\ \dagger\text{-Const} \qquad **$$

$$\frac{\vdash (\lambda\_:().x) : () \to () \qquad w : !().\text{end}! \vdash \text{fork}\ (\lambda\_:() \to \text{else}\ (\text{send}\ ()\ w)) : ()}{w : !().\text{end}! \vdash (\lambda\_:().x)\ \text{fork}\ (\lambda\_:() \to \text{else}\ (\text{send}\ ()\ w)) : ()}\ \dagger\text{-APP}$$

$$\frac{w : !().\text{end}! \vdash (\lambda\_:().x)\ \text{fork}\ (\lambda\_:() \to \text{else}\ (\text{send}\ ()\ w)) : ()}{w : !().\text{end}! \vdash (\lambda\_:().(\lambda\_:().x)\ \text{fork}\ (\lambda\_:() \to \text{else}\ (\text{send}\ ()\ w))) : () \to ()}\ \dagger\text{-ABS}$$

$$\frac{}{* \; a : ?().\text{end}? \vdash a : ?().\text{end}?}\ \dagger\text{-VAR}$$

$$\frac{\dagger\text{-CONST}}{\vdash \text{receive} : ?().\text{end}? \to () \times \text{end}?} \qquad * \qquad \frac{\dagger\text{-Const}}{\vdash \text{Wait} : \text{end}? \to ()} \qquad \frac{\dagger\text{-VAR}}{a : \text{end}? \vdash a : \text{end}?}$$

$$\frac{\vdash \text{receive}\ a : () \times \text{end}? \qquad x : () , a : \text{end}? \vdash \text{Wait}\ a : ()}{a : ?().\text{end}? \vdash \text{receive}\ a : () \times \text{end}? \qquad x : () , a : \text{end}? \vdash \text{Wait}\ a : ()}\ \dagger\text{-APP}$$

$$\frac{a : ?().\text{end}? \vdash \text{receive}\ a : () \times \text{end}? \qquad x : () , a : \text{end}? \vdash \text{Wait}\ a : ()}{a : ?().\text{end}? \vdash \text{let}\ (x, a) = \text{receive}\ a\ \text{in}\ \text{Wait}\ a : ()}\ \dagger\text{-LET}$$

$$\frac{a : ?().\text{end}? \vdash \text{let}\ (x, a) = \text{receive}\ a\ \text{in}\ \text{Wait}\ a : ()}{w : (!().\text{end}!) , a : (?().\text{end}?) \vdash (\lambda\_:().(\lambda\_:().x)\ \text{fork}\ (\lambda\_:() \to \text{else}\ (\text{send}\ ()\ w)))\ \text{let}\ (x, a) = \text{receive}\ \text{in}\ \text{Wait}\ a : ()}\ \dagger\text{-APP}$$

$$\frac{w : (!().\text{end}!) , a : (?().\text{end}?) \vdash (\lambda\_:().(\lambda\_:().x)\ \text{fork}\ (\lambda\_:() \to \text{else}\ (\text{send}\ ()\ w)))\ \text{let}\ (x, a) = \text{receive}\ \text{in}\ \text{Wait}\ a : () \qquad w : (!().\text{end}!) , a : (?().\text{end}?) \vdash \text{let}\ (x, a) = \text{receive}\ a\ \text{in}\ \text{Wait}\ a ; \text{fork}\ (\lambda\_:() \to \text{else}\ (\text{send}\ ()\ w)) ; x : ()}{\vdash \text{let}\ (w, a) = \text{new}\ !().\text{end}!\ \text{in}\ s_2 ; (\text{fork}\ v_1) ; x : ()}\ \dagger\text{-LET}$$

$$\frac{\dagger\text{-NEW}}{\vdash \text{new}\ !\text{unit}.\text{end}! : (!().\text{end}!) \times (?().\text{end}?)} \qquad w : (!().\text{end}!) , a : (?().\text{end}?) \vdash s_2 ; (\text{fork}\ v_1) ; x : ()$$

Reduction Derivation Steps

$$\frac{}{\langle e \rangle \to (\nu xy)\langle \text{let } (w,r) = \text{new } !\text{unit}.\text{end}! \text{ in } e_2; (\text{fork } v_1); x} \ \text{R-NEW}$$

(e2) E-SPLIT (e1)

$$\cfrac{\cfrac{\cfrac{}{\text{let } (w,r) = (x,y) \text{ in } e_2; (\text{fork } v_1); x \to \text{let } (x,y) = \text{receive } y \text{ in wait } y; \text{fork}(\lambda\_:() \to \text{else } (\text{end } () \, x)); x}}{\langle \text{let } (w,r) = (x,y) \text{ in } e_2; (\text{fork } v_1); x\rangle \to \langle e_2; (\text{fork } v_1); x\rangle} \ \text{R-THREAD}}{(\nu xy)\langle \text{let } (w,r) = (x,y) \text{ in } e_2; (\text{fork } v_1); x \to (\nu xy)\langle e_2; (\text{fork } v_1); x\rangle} \ \text{R-RES}$$

$$\cfrac{\cfrac{}{\langle(\lambda\_:().(\lambda\_:().x)e_2)(\text{fork } v_1)\rangle \to \langle(\lambda\_:().(\lambda\_:().y)e_2)()\rangle \mid \langle v_1()\rangle} \ \text{R-FORK}}{(\nu xy)\langle \underbrace{(\lambda\_:().(\lambda\_:().x)e_2)(\text{fork } v_1)}_{\Updownarrow \ e_2; (\text{fork } v_1); x}\rangle \to (\nu xy)\langle(\lambda\_:().(\lambda\_:().x)e_2)()\rangle \mid \langle v_1()\rangle} \ \text{R-RES}$$

$$\cfrac{\cfrac{\cfrac{\cfrac{}{(\lambda\_:().(\lambda\_:().x)e_2)() \to (\lambda\_:().x)e_2} \ \text{E-APP}}{\langle(\lambda\_:().(\lambda\_:().x)e_2)()\rangle \to \langle(\lambda\_:().y)e_2\rangle} \ \text{R-THREAD}}{\langle(\lambda\_:().(\lambda\_:().x)e_2)()\rangle \mid \langle v_1()\rangle \to \langle(\lambda\_:().x)e_2\rangle \mid \langle v_1()\rangle} \ \text{R-PAR}}{(\nu xy)\langle(\lambda\_:().(\lambda\_:().x)e_2)()\rangle \mid \langle v_1()\rangle \to (\nu xy)\langle(\lambda\_:().x)e_2\rangle \mid \langle v_1()\rangle} \ \text{R-RES}$$

Repeating the last derivation tree, we have that:

$(\nu xy)\langle(\lambda\_:().x)e_2\rangle \mid \langle v_1()\rangle \to (\nu xy)\langle x\rangle \mid \langle v_1()\rangle$

We can conclude that, since the process <x> can never end up in a "wait x" or "else y" format.

With this, we can conclude that R-Close will never apply and we don't have to reduce any further to show this.