Para demostrar que el bytecode generado mediante la optimización de taillcall es efectivamente menor queremos mostrar que

$$\text{length}\left(\mathsf{bcc}\ t\right) \ \leq\ \text{length}\left(\mathsf{bcc}'\ t\right). \qquad \texttt{t::TTerm}$$

donde $\mathsf{bcc}\ t$ es el bytecode optimizado y $\mathsf{bcc}'\ t$ el original, es decir, previa optimización con taillcall

Antes de demostrar eso demostraremos el siguiente lema:

Lema 1. *Para todo* t : TTerm *se verifica*

$$\text{length}\ \left(\mathsf{bctc}\ t\right)\ \leq\ \text{length}\ \left(\mathsf{bcc}\ t\right) + 1.$$

Demostración. Inducción estructural sobre t.

Caso base. t construido con `V` o `Const`:

$$\text{length}\left(\mathsf{bctc}\ t\right) \overset{\text{def.bctc}}{=} \text{length}\ \left(\mathsf{bcc}\ t \mathbin{+\!\!+} [\texttt{RETURN}]\right) \overset{\text{length.(++)}}{=} \text{length}\left(\mathsf{bcc}\ t\right) + 1.$$

Caso Inductivo. Sea t un término cualquiera y supongamos la hipótesis inductiva (H.I) length $\mathsf{bctc}\ t' \leq$ length $\mathsf{bcc}\ t' + 1$ para todo sub-término $t' \preccurlyeq t$. Analizamos cada constructor de `TTerm`:

(a) Aplicación ($t = \texttt{App}\ \_\ t_1\, t_2$):

$$\begin{aligned}
\text{length}\ \mathsf{bctc}\ t &\overset{\text{def.bctc}}{=} \text{length}\ (\mathsf{bcc}\ t_1 \mathbin{+\!\!+} \mathsf{bcc}\ t_2 \mathbin{+\!\!+} [\texttt{RETURN}]) \\
&\overset{\text{length.(++)}}{=} \text{length}\ (\mathsf{bcc}\ t_1 \mathbin{+\!\!+} \mathsf{bcc}\ t_2) + 1 \\
&\overset{\text{def.bcc}}{=} \text{length}\ (\mathsf{bcc}\ t) + 1.
\end{aligned}$$

(b) Condicional cero ($t = \texttt{IfZ}\ \_\ c\ t_1\ t_2$). Sean $\ell_1 = \text{length}\ (\mathsf{bctc}\ t_1) + 2$, $\ell_1' = \text{length}\ (\mathsf{bcc}\ t_1) + 2$, $\ell_2 = \text{length}\ (\mathsf{bctc}\ t_2)$ y $\ell_2' = \text{length}\ (\mathsf{bcc}\ t_2)$:

$$\begin{aligned}
\text{length}\ (\mathsf{bctc}\ t) &\overset{\text{def.bctc}}{=} \text{length}\left(\mathsf{bcc}\ c \mathbin{+\!\!+} [\texttt{CJUMP}, \ell_1] \mathbin{+\!\!+} \text{longJump}\ (\mathsf{bctc}\ t_1)\ \ell_2 \mathbin{+\!\!+} [\texttt{JUMP}, \ell_2] \mathbin{+\!\!+} \mathsf{bctc}\ t_2\right) \\
&\overset{\text{length.(++)}}{=} \text{length}\ (\mathsf{bcc}\ c) + 2 + \text{length}\ (\mathsf{bctc}\ t_1) + 2 + \text{length}\ (\mathsf{bctc}\ t_2) \\
&\overset{\text{H.I.}}{\leq} \text{length}\ (\mathsf{bcc}\ c) + 2 + \left(\text{length}\ (\mathsf{bcc}\ t_1) + 1\right) + 2 + \left(\text{length}\ (\mathsf{bcc}\ t_2) + 1\right) \\
&\overset{\text{length.(++)}}{=} \text{length}\left(\mathsf{bcc}\ c \mathbin{+\!\!+} [\texttt{CJUMP}, \ell_1'] \mathbin{+\!\!+} \text{longJump}\ (\mathsf{bcc}\ t_1)\ \ell_2' \mathbin{+\!\!+} [\texttt{JUMP}, \ell_2'] \mathbin{+\!\!+} \mathsf{bcc}\ t_2\right) \\
&\overset{\text{def.bcc}}{=} \text{length}\ \left(\mathsf{bcc}\ t\right) + 1
\end{aligned}$$

(c) Let anidado ($t = \texttt{Let}\ \_\ \_\ t_1\, (\texttt{Sc1}\ \ t_2)$) con subcasos:

(i) $t_2 = \texttt{Print}\ \ \_\ s\ (\texttt{V}\ \ \_\ (\texttt{Bound}\ \ 0))$

$$\begin{aligned}
\text{length}\ (\mathsf{bctc}\ t) &\overset{\text{def.bctc}}{=} \text{length}\ \left(\mathsf{bctc}\ (\texttt{Print}\ \ _\ s\ t_1)\right) \\
&\overset{\text{H.I.}}{\leq} \text{length}\ \left(\mathsf{bcc}\ (\texttt{Print}\ \ _\ s\ t_1)\right) + 1 \\
&\overset{\text{def.bcc}}{=} \text{length}\ (\mathsf{bcc}\ t) + 1
\end{aligned}$$

(II) $t_2 = \texttt{Let}\ i\ \_\ \_\ (\texttt{Print}\ \ \_\ s\ (\texttt{V}\ \ \_\ (\texttt{Bound}\ \ 0)))\ (\texttt{Sc1}\ \ t_3)$

$$\begin{aligned}
\text{length}\ (\mathsf{bctc}\ t) &\overset{\text{def.bctc}}{=} \text{length}\big(\mathsf{bcc}\ (\texttt{Print}\ \ i\, s\, t_1) +\!\!+ [\texttt{SHIFT}] +\!\!+ \mathsf{bctc}\ (\texttt{letSimp}\ \ t_3)\big)\\
&\overset{\text{length.++}}{=} \text{length}\big(\mathsf{bcc}\ (\texttt{Print}\ \ i\, s\, t_1)\big) + 1 + \text{length}\big(\mathsf{bctc}\ (\texttt{letSimp}\ \ t_3)\big)\\
&\overset{\text{H.I.}}{\leq} \text{length}\big(\mathsf{bcc}\ (\texttt{Print}\ \ i\, s\, t_1)\big) + 1 + \big(\text{length}\ (\mathsf{bcc}\ (\texttt{letSimp}\ \ t_3)) + 1\big)\\
&\overset{\text{length.++}}{=} \text{length}\big(\mathsf{bcc}\ (\texttt{Print}\ \ i\, s\, t_1) +\!\!+ [\texttt{SHIFT}] +\!\!+ \mathsf{bcc}\ (\texttt{letSimp}\ \ t_3)\big) + 1\\
&\overset{\text{def.bcc}}{=} \text{length}\ (\mathsf{bcc}\ t) + 1
\end{aligned}$$

(III) $t_2 = \texttt{V}\ \ \_\ (\texttt{Bound}\ \ 0)$

$$\begin{aligned}
\text{length}\ (\mathsf{bctc}\ t) &\overset{\text{def.bctc}}{=} \text{length}\ (\mathsf{bctc}\ t_1)\\
&\overset{\text{H.I.}}{\leq} \text{length}\ (\mathsf{bcc}\ t_1) + 1\\
&\overset{\text{def.bcc}}{=} \text{length}\ (\mathsf{bcc}\ t) + 1
\end{aligned}$$

(IV) *Resto de subcasos*

$$\begin{aligned}
\text{length}\ (\mathsf{bctc}\ t) &\overset{\text{def.bctc}}{=} \text{length}\ (\mathsf{bcc}\ t)\\
&\leq \text{length}\ (\mathsf{bcc}\ t) + 1
\end{aligned}$$

□

Observación. Se usa que $\text{length}\ \big(\text{longJump}\ xs\, j\big) = \text{length}\ (xs)$, demostrable por inducción en la lista xs.

Teorema 1. *Para todo* $t :: \text{TTerm}$ *se cumple*

$$\text{length}\big(\mathsf{bcc}\ t\big)\ \leq\ \text{length}\big(\mathsf{bcc}'\ t\big).$$

Demostración. Por inducción estructural sobre `TTerm`.

Caso base. Cuando t es construido con `V` o `Const`, ambas implementaciones son idénticas, así que la desigualdad es trivial.

Paso inductivo. Supongamos como hipotesis inductiva (H.I.) que

$$\text{length}\ \big(\mathsf{bcc}\ t'\big) \leq \text{length}\ \big(\mathsf{bcc}'\ t'\big)$$

para todo sub-término $t' \preccurlyeq t$. Solo hay que tratar los constructores modificados; los restantes son idénticos en ambas versiones.

(a) Expresiones lambda ($t = \texttt{Lam}\ \_\ \_\ \_\ (\texttt{Sc1}\ t_1)$)

$$
\begin{aligned}
\text{length}\left(\mathsf{bcc}\ t\right) &\overset{\text{def.bcc}}{=} \text{length}\Big([\texttt{FUNCTION}, \text{length}\left(\mathsf{bctc}\ t_1\right)] +\!\!+ \mathsf{bctc}\ t_1\Big) \\
&\overset{\text{length.}(++)}{=} 2 + \text{length}\left(\mathsf{bctc}\ t_1\right) \\
&\overset{\text{Lema 1}}{\leq} 2 + \left(\text{length}\left(\mathsf{bcc}\ t_1\right) + 1\right) \\
&= 2 + 1 + \text{length}\left(\mathsf{bcc}\ t_1\right) \\
&\overset{\text{length.}(++)}{=} \text{length}\Big([\texttt{FUNCTION}, \text{length}\left(\mathsf{bcc}\ t_1\right)] +\!\!+ [\texttt{RETURN}]\Big) + \text{length}\left(\mathsf{bcc}\ t_1\right) \\
&\overset{\text{H.I.}}{\leq} \text{length}\Big([\texttt{FUNCTION}, \text{length}\left(\mathsf{bcc}\ t_1\right)] +\!\!+ [\texttt{RETURN}]\Big) + \text{length}\left(\mathsf{bcc}'\ t_1\right) \\
&= \text{length}\Big([\texttt{FUNCTION}, \text{length}\left(\mathsf{bcc}\ t_1\right)] +\!\!+ \mathsf{bcc}'\ t_1 +\!\!+ [\texttt{RETURN}]\Big) \\
&\overset{\text{def.}\mathsf{bcc}'}{=} \text{length}\left(\mathsf{bcc}'\ t\right).
\end{aligned}
$$

(b) Recursión mediante punto fijo ($t = \texttt{Fix}\ \_\ \_\ \_\ \_\ \_\ (\texttt{Sc2}\ t_1)$)

$$
\begin{aligned}
\text{length}\left(\mathsf{bcc}\ t\right) &\overset{\text{def.bcc}}{=} \text{length}\Big([\texttt{FUNCTION}, \text{length}\left(\mathsf{bctc}\ t_1\right)] +\!\!+ \mathsf{bctc}\ t_1 +\!\!+ [\texttt{FIX}]\Big) \\
&\overset{\text{length.}(++)}{=} 2 + \text{length}\left(\mathsf{bctc}\ t_1\right) \\
&\overset{\text{Lema 1}}{\leq} 2 + \left(\text{length}\left(\mathsf{bcc}\ t_1\right) + 1\right) \\
&\overset{\text{length.}(++)}{=} \text{length}\Big([\texttt{FUNCTION}, \text{length}\left(\mathsf{bcc}\ t_1\right)] +\!\!+ [\texttt{FIX}] +\!\!+ [\texttt{RETURN}]\Big) + \text{length}\left(\mathsf{bcc}\ t_1\right) \\
&\overset{\text{H.I.}}{\leq} \text{length}\Big([\texttt{FUNCTION}, \text{length}\left(\mathsf{bcc}\ t_1\right)] +\!\!+ [\texttt{FIX}] +\!\!+ [\texttt{RETURN}]\Big) + \text{length}\left(\mathsf{bcc}'\ t_1\right) \\
&= \text{length}\Big([\texttt{FUNCTION}, \text{length}\left(\mathsf{bcc}\ t_1\right)] +\!\!+ \mathsf{bcc}'\ t_1 +\!\!+ [\texttt{FIX}] +\!\!+ [\texttt{RETURN}]\Big) \\
&\overset{\text{def.}\mathsf{bcc}'}{=} \text{length}\left(\mathsf{bcc}'\ t\right).
\end{aligned}
$$

Como todos los casos (modificados o no) satisfacen la cota, la proposición es válida para todo t. $\square$