$\Theta(n)$ List Reverse
Justin Raymond
December 1, 2015

Source Language

```
datatype list = Nil of unit | Cons of int × list
```

The following function reverses a list on $\Theta(n)$ time.

```
rev xs = λxs.rec(xs, Nil ↦ λa.a,
                 Cons ↦ ⟨x,⟨xs,r⟩⟩.λa.force(r) Cons⟨x,a⟩) Nil
```

If we write out the match explicitly using splits:

```
rev xs = λxs.rec(xs,
               Nil ↦ λa.a,
               Cons↦b.split(b,x.c.split(c,xs'.r.
                      λa.force(r) Cons⟨x,a⟩))) Nil
```

The specification of `rev` is `rev` $[x_0, \ldots, x_{n-1}]$ = $[x_{n-1}, \ldots, x_0]$. The specification of the auxilary function `rec(xs,...)` is `rec(`$[x_0, \ldots, x_{n-1}]$`,...)` $[y_0, \ldots, y_{m-1}]$ = $[x_{n-1}, \ldots, x_0, y_0, \ldots, y_{m-1}]$.

Complexity Language

Translation of `rev` using matching

The translation into the complexity language proceeds as follows. First we apply the rule $\|\lambda \mathtt{x.e}\|$ = $\langle 0, \lambda \mathtt{x}.\|\mathtt{e}\|\rangle$

```
‖rev‖ = ⟨0,λxs.‖rec(xs, Nil ↦ λa.a
                  , Cons ↦ ⟨x,⟨xs,r⟩⟩.λa.force(r) Cons⟨x,a⟩) Nil‖⟩
```

The we apply the rule for function application, $\|\mathtt{e}_0\ \mathtt{e}_1\| = 1 + \|\mathtt{e}_0\|_c + \|\mathtt{e}_1\|_c +_c \|\mathtt{e}_0\|_p \|\mathtt{e}_1\|_p$.

$\|\texttt{rev}\|$ = $\langle \texttt{0}, \lambda \texttt{xs}.(\texttt{1}+\|\texttt{xs}\|_c + \|\texttt{rec(...)}\|_c + \|\texttt{Nil}\|_c) +_c \|\texttt{rec(...)}\|_p \|\texttt{Nil}\|_p\rangle$

We will focus on the translation of the `rec` construct. We apply the rule $\|\texttt{rec(xs,}\overline{C \mapsto x.e_C})\|$ = $\|xs\|_c +_c$ $\texttt{rec}(\|\texttt{xs}\|_p, \overline{C \mapsto x.1 +_c \|e_C\|})$

$$
\begin{aligned}
\|\texttt{rec(xs,...)}\| &= \|\texttt{xs}\|_c +_c \texttt{ rec}(\|\texttt{xs}\|_p,\\
&\qquad \texttt{Nil} \mapsto \texttt{1} +_c \|\lambda\texttt{a.a}\|,\\
&\qquad \texttt{Cons} \mapsto \langle \texttt{x}, \langle \texttt{xs}, \texttt{r}\rangle\rangle.\texttt{1} +_c \|\lambda\texttt{a.force(r) Cons}\langle \texttt{x},\texttt{a}\rangle\|)\\
&= \langle \texttt{0}, \texttt{xs}\rangle_c +_c \texttt{ rec}(\langle \texttt{0}, \texttt{xs}\rangle_p,\\
&\qquad \texttt{Nil} \mapsto \texttt{1} +_c \|\lambda\texttt{a.a}\|,\\
&\qquad \texttt{Cons} \mapsto \langle \texttt{x}, \langle \texttt{xs}, \texttt{r}\rangle\rangle.\texttt{1} +_c \|\lambda\texttt{a.force(r) Cons}\langle \texttt{x},\texttt{a}\rangle\|)\\
&= \texttt{rec(xs},\\
&\qquad \texttt{Nil} \mapsto \texttt{1} +_c \|\lambda\texttt{a.a}\|,\\
&\qquad \texttt{Cons} \mapsto \langle \texttt{x}, \langle \texttt{xs}, \texttt{r}\rangle\rangle.\texttt{1} +_c \|\lambda\texttt{a.force(r) Cons}\langle \texttt{x},\texttt{a}\rangle\|)
\end{aligned}
$$

The translation of the `Nil` branch is simple.

$$
\begin{aligned}
&= \texttt{1} +_c \|\lambda\texttt{a.a}\|\\
&= \texttt{1} +_c \langle \texttt{0}, \lambda\texttt{a.}\|\texttt{a}\|\rangle\\
&= \texttt{1} +_c \langle \texttt{0}, \lambda\texttt{a.}\langle \texttt{0},\texttt{a}\rangle\rangle\\
&= \langle \texttt{1}, \lambda\texttt{a.}\langle \texttt{0},\texttt{a}\rangle\rangle
\end{aligned}
$$

The translation of the `Cons` branch is a slightly more involved.

$$= \texttt{1} +_c \|\lambda\texttt{a.force(r) Cons}\langle \texttt{x},\texttt{a}\rangle\|)$$

$$= \texttt{1} +_c \langle 0, \|\lambda\texttt{a.force(r) Cons}\langle \texttt{x},\texttt{a}\rangle\|\rangle$$

$$= \langle 1, \lambda\texttt{a.}\|\texttt{force(r) Cons}\langle \texttt{x},\texttt{a}\rangle\|\rangle$$

$$= \langle 1, \lambda\texttt{a.}(1 + \|\texttt{force(r)}\|_c + \|\texttt{Cons}\langle \texttt{x},\texttt{a}\rangle\|_c) +_c \|\texttt{force(r)}\|_p \ \|\texttt{Cons}\langle \texttt{x},\texttt{a}\rangle\|_p\rangle$$

$$= \langle 1, \lambda\texttt{a.}(1 + (\|\texttt{r}\|_c +_c \|\texttt{r}\|_p)_c + \|\texttt{Cons}\langle \texttt{x},\texttt{a}\rangle\|_c) +_c (\|\texttt{r}\|_c +_c \|\texttt{r}\|_p)_p \ \|\texttt{Cons}\langle \texttt{x},\texttt{a}\rangle\|_p\rangle$$

$$= \langle 1, \lambda\texttt{a.}(1+\texttt{r}_c + \|\texttt{Cons}\langle \texttt{x},\texttt{a}\rangle\|_c)+_c \ \texttt{r}_p \ \|\texttt{Cons}\langle \texttt{x},\texttt{a}\rangle\|_p\rangle$$

$$= \langle 1, \lambda\texttt{a.}(1+\texttt{r}_c + (\langle\|\langle \texttt{x},\texttt{a}\rangle\|_c, \texttt{Cons}\|\langle \texttt{x},\texttt{a}\rangle\|_p\rangle)_c)+_c \ \texttt{r}_p(\langle\|\langle \texttt{x},\texttt{a}\rangle\|_c, \texttt{Cons}\|\langle \texttt{x},\texttt{a}\rangle\|_p\rangle)_p\rangle$$

$$= \langle 1, \lambda\texttt{a.}(1+\texttt{r}_c + \|\langle \texttt{x},\texttt{a}\rangle\|_c)+_c \ \texttt{r}_p\texttt{Cons}\|\langle \texttt{x},\texttt{a}\rangle\|_p\rangle$$

$= \langle 1, \lambda\texttt{a}\,.\,(1{+}\texttt{r}_c + \langle \|\texttt{x}\|_c + \|\texttt{a}\|_c\,, \langle \|\texttt{x}\|_p\,, \|\texttt{a}\|_p \rangle\rangle_c) {+}_c\ \texttt{r}_p \texttt{Cons}\langle \|\texttt{x}\|_c + \|\texttt{a}\|_c\,, \langle \|\texttt{x}\|_p\,, \|\texttt{a}\|_p \rangle\rangle_p \rangle$

$= \langle 1, \lambda\texttt{a}\,.\,(1{+}\texttt{r}_c + \|\texttt{x}\|_c + \|\texttt{a}\|_c) {+}_c\ \texttt{r}_p \texttt{Cons}\langle \|\texttt{x}\|_p\,, \|\texttt{a}\|_p \rangle\rangle$

$= \langle 1, \lambda\texttt{a}\,.\,(1{+}\texttt{r}_c + \langle \texttt{0}\,, \texttt{x}\rangle_c + \langle \texttt{0}\,, \texttt{a}\rangle_c) {+}_c\ \texttt{r}_p \texttt{Cons}\langle \langle \texttt{0}\,, \texttt{x}\rangle_p\,, \langle \texttt{0}\,, \texttt{a}\rangle_p \rangle\rangle$

$= \langle 1, \lambda\texttt{a}\,.\,(1{+}\texttt{r}_c + 0 + 0) {+}_c\ \texttt{r}_p \texttt{Cons}\langle \texttt{x}\,, \texttt{a}\rangle\rangle$

$= \langle 1, \lambda\texttt{a}\,.\,(1{+}\texttt{r}_c) {+}_c\ \texttt{r}_p \texttt{Cons}\langle \texttt{x}\,, \texttt{a}\rangle\rangle$

So the translation of the whole `rec` is:

$\texttt{rec(xs, Nil}\mapsto \langle \texttt{1}, \lambda\texttt{a}.\langle \texttt{0}, \texttt{a}\rangle\rangle\texttt{, Cons}\mapsto \langle \texttt{x}, \langle \texttt{xs'}, \texttt{r}\rangle\rangle.\langle 1, \lambda\texttt{a}\,.\,(1{+}\texttt{r}_c) {+}_c\ \texttt{r}_p \texttt{Cons}\langle \texttt{x}, \texttt{a}\rangle\rangle\texttt{)}$

We observe that in both cases, the cost of `rec` is 1, so we can simplify $\texttt{r}_c$ to 1.

$\texttt{rec(xs, Nil}\mapsto \langle \texttt{1}, \lambda\texttt{a}.\langle \texttt{0}, \texttt{a}\rangle\rangle\texttt{, Cons}\mapsto \langle \texttt{x}, \langle \texttt{xs'}, \texttt{r}\rangle\rangle.\langle 1, \lambda\texttt{a}\,.\,2 {+}_c\ \texttt{r}_p \texttt{Cons}\langle \texttt{x}, \texttt{a}\rangle\rangle\texttt{)}$

We will pick up where we left off with out translation of `rev`.

$\|\texttt{rev}\| = \langle \texttt{0}, \lambda\texttt{xs}\,.\,(\texttt{1}{+}\|\texttt{xs}\|_c + \|\texttt{rec(...)}\|_c + \|\texttt{Nil}\|_c) +_c \|\texttt{rec(...)}\|_p \|\texttt{Nil}\|_p\rangle$

First we will translate the variables.

$$\begin{aligned}\|\texttt{rev}\| &= \langle \texttt{0}, \lambda\texttt{xs}\,.\,(1 + \langle \texttt{0}, \texttt{xs}\rangle_c + \|\texttt{rec(...)}\|_c + \langle \texttt{0}, \texttt{Nil}\rangle_c) +_c \|\texttt{rec(...)}\|_p \langle \texttt{0}, \texttt{Nil}\rangle_p\rangle \\ &= \langle \texttt{0}, \lambda\texttt{xs}\,.\,(1 + 0 + \|\texttt{rec(...)}\|_c + 0) +_c \|\texttt{rec(...)}\|_p \texttt{Nil}\rangle\end{aligned}$$

We use our translation of `rec(xs,...)` and the fact that the cost of every call to `rec` is 1 to get:

$$\begin{aligned}\|\texttt{rev}\| &= \langle \texttt{0}, \lambda\texttt{xs}\,.\,(1 + 0 + 1 + 0) +_c \|\texttt{rec(...)}\|_p \texttt{Nil}\rangle \\ &= \langle \texttt{0}, \lambda\texttt{xs}\,.\,2 {+}_c \texttt{rec(xs, Nil}\mapsto \langle \texttt{1}, \lambda\texttt{a}.\langle \texttt{0}, \texttt{a}\rangle\rangle, \\ &\qquad \texttt{Cons}\mapsto \langle \texttt{x}, \langle \texttt{xs'}, \texttt{r}\rangle\rangle.\langle 1, \lambda\texttt{a}\,.\,2 {+}_c\ \texttt{r}_p \texttt{Cons}\langle \texttt{x}, \texttt{a}\rangle\rangle\texttt{)}_p \texttt{Nil}\rangle\end{aligned}$$

So our complete translation of the linear time reversal function is

$$\boxed{\begin{aligned}\|\texttt{rev}\| &= \langle \texttt{0}, \lambda\texttt{xs}\,.\,(1 + 0 + 1 + 0) +_c \|\texttt{rec(...)}\|_p \texttt{Nil}\rangle \\ &= \langle \texttt{0}, \lambda\texttt{xs}\,.\,2 {+}_c \texttt{rec(xs, Nil}\mapsto \langle \texttt{1}, \lambda\texttt{a}.\langle \texttt{0}, \texttt{a}\rangle\rangle, \\ &\qquad \texttt{Cons}\mapsto \langle \texttt{x}, \langle \texttt{xs'}, \texttt{r}\rangle\rangle.\langle 1, \lambda\texttt{a}\,.\,2 {+}_c\ \texttt{r}_p \texttt{Cons}\langle \texttt{x}, \texttt{a}\rangle\rangle\texttt{)}_p \texttt{Nil}\rangle\end{aligned}}$$

The interpretation of `rev` is rather dull as the cost of `rev` is always null. Instead of interpreting `rev`, we will interpret `rev xs`. In preparation we will translate `rev xs`.

$$
\begin{aligned}
\|\texttt{rev xs}\| &= (1+\|\texttt{rev}\|_c + \|\texttt{xs}\|_c) +_c \|\texttt{rev}\|_p \|\texttt{xs}\|_p \\
&= (1+0+\langle \texttt{0,xs}\rangle_c) +_c \|\texttt{rev}\|_p \langle \texttt{0,xs}\rangle_p \\
&= (1+0+0) +_c \|\texttt{rev}\|_p \texttt{xs} \\
&= 1 +_c \|\texttt{rev}\|_p \texttt{xs} \\
&= 1 +_c (\lambda \texttt{xs.rec(xs, Nil}\mapsto \langle \texttt{1},\lambda \texttt{a.}\langle \texttt{0,a}\rangle\rangle, \\
&\qquad\qquad \texttt{Cons}\mapsto \langle \texttt{x},\langle \texttt{xs',r}\rangle\rangle.\langle 1,\lambda \texttt{a}.2 +_c \texttt{r}_p \texttt{Cons}\langle \texttt{x,a}\rangle\rangle)_p \texttt{Nil) xs} \\
&= 1 +_c \texttt{rec(xs, Nil}\mapsto \langle \texttt{1},\lambda \texttt{a.}\langle \texttt{0,a}\rangle\rangle, \\
&\qquad\qquad \texttt{Cons}\mapsto \langle \texttt{x},\langle \texttt{xs',r}\rangle\rangle.\langle 1,\lambda \texttt{a}.2 +_c \texttt{r}_p \texttt{Cons}\langle \texttt{x,a}\rangle\rangle)_p \texttt{Nil)} \\
&= 1 +_c \texttt{rec(xs, Nil}\mapsto \langle \texttt{1},\lambda \texttt{a.}\langle \texttt{0,a}\rangle\rangle, \\
&\qquad\qquad \texttt{Cons}\mapsto \langle \texttt{x},\langle \texttt{xs',r}\rangle\rangle.\langle 1,\lambda \texttt{a}.2 +_c \texttt{r}_p \texttt{Cons}\langle \texttt{x,a}\rangle\rangle)_p \texttt{Nil}
\end{aligned}
$$

$$
\begin{aligned}
\|\texttt{rev xs}\| &= 1 +_c \texttt{rec(xs, Nil}\mapsto \langle \texttt{1},\lambda \texttt{a.}\langle \texttt{0,a}\rangle\rangle, \\
&\qquad\qquad \texttt{Cons}\mapsto \langle \texttt{x},\langle \texttt{xs',r}\rangle\rangle.\langle 1,\lambda \texttt{a}.2 +_c \texttt{r}_p \texttt{Cons}\langle \texttt{x,a}\rangle\rangle)_p \texttt{Nil}
\end{aligned}
$$

Interpretation

We intepret the size of an `list` to be the number of list constructors.

$[\![\ \texttt{list}\ ]\!] = \mathbb{N}^\infty$
$D^{list} = \{*\} + \{1\} \times \mathbb{N}^\infty$
$size_{list}(\texttt{Nil}) = 1$
$size_{list}(\texttt{Cons(1,n)}) = 1 + n$

The interpretation of `rev xs` proceeds as follows.

$$
\begin{aligned}
&[\![\|\texttt{rev xs}\|]\!] = [\![1 +_c \texttt{rec(xs,...)}_p\ \texttt{Nil}]\!] \\
&= [\![\langle 1 + (\texttt{rec(xs,...)}_p\ \texttt{Nil})_c, (\texttt{rec(xs,...)}_p\ \texttt{Nil})_p\rangle]\!] \\
&= \langle 1 + [\![(\texttt{rec(xs,...)}_p\ \texttt{Nil})_c]\!], [\![(\texttt{rec(xs,...)}_p\ \texttt{Nil})_p]\!]\rangle
\end{aligned}
$$

We will focus on the interpretation of the auxilary function `rec(xs,...)`.

Let $g(n) = [\![\texttt{rec(xs,...)}]\!]\{xs \mapsto n\}$

$$g(n) = \bigvee_{size\ ys \le n} case(ys, Nil \mapsto \langle 1, \lambda a.\langle 0, a\rangle\rangle, Cons \mapsto \langle 1, m\rangle.\langle 1, \lambda a.2{+_c}\pi_1 g(m)(a{+}1)\rangle)$$

For $n = 0$, $g(0) = \langle 1, \lambda a.\langle 0, a\rangle\rangle$.

For $n > 0$,

$$g(n{+}1) = \bigvee_{size\ ys \le n+1} case(ys, Nil \mapsto \langle 1, \lambda a.\langle 0, a\rangle\rangle, Cons \mapsto \langle 1, m\rangle.\langle 1, \lambda a.2{+_c}\pi_1 g(m)(a{+}1)\rangle)$$

$$g(n{+}1) = \bigvee_{size\ ys \le n} case(ys, Nil \mapsto \langle 1, \lambda a.\langle 0, a\rangle\rangle, Cons \mapsto \langle 1, m\rangle.\langle 1, \lambda a.2{+_c}\pi_1 g(m)(a{+}1)\rangle)$$

$$\vee \bigvee_{size\ ys = n+1} case(ys, Nil \mapsto \langle 1, \lambda a.\langle 0, a\rangle\rangle, Cons \mapsto \langle 1, m\rangle.\langle 1, \lambda a.2 +_c \pi_1 g(m)(a+1)\rangle)$$

$$g(n{+}1) = g(n) \vee \bigvee_{size\ ys = n+1} case(ys, Nil \mapsto \langle 1, \lambda a.\langle 0, a\rangle\rangle, Cons \mapsto \langle 1, m\rangle.\langle 1, \lambda a.2{+_c}\pi_1 g(m)(a{+}1)\rangle)$$

$$g(n+1) = g(n) \vee \langle 1, \lambda a.2 +_c \pi_1 g(n)(a+1)\rangle\rangle$$

We want to show that g is monotonically increasing; $\forall n.g(n) \le g(n{+}1)$. By definition of $\le$, $g(n) \le g(n{+}1) \Leftrightarrow \pi_0 g(n) \le \pi_0 g(n{+}1) \wedge \pi_1 g(n) \le \pi_1 g(n{+}1)$. First we will show $\forall n.\pi_0 g(n) = 1$, the immediate corollary of which is $\forall n.\pi_0 g(n) \le \pi_0 g(n+1)$.

Proof. We prove this by induction on n.

Base case: $n = 0$

By definition, $\pi_0 g(0) = 1$.

Induction step: $n > 0$

By definition $\pi_0 g(n+1) = \pi_0(g(n) \vee \langle 1, \lambda a.2 +_c \pi_1 g(n)(a+1)\rangle)$. We distribute the projection over the max: $\pi_0 g(n+1) = \pi_0 g(n) \vee 1$. By the induction hypothesis, $\pi_0 g(n) = 1$, so $\pi_0 g(n+1) = 1$.

□

Now we argue that $\pi_1 g(n) \leq \pi_1 g(n+1)$. First we prove the lemma $\forall n. \pi_1 g(n) a \leq \pi_1 g(n)(a+1)$.

Proof. We prove this by induction on n.

$n = 0$

$\pi_1 g(0) a = \langle 0, a \rangle \leq \pi_1 g(0)(a+1) = \langle 0, a+1 \rangle$.

$n > 0$

We assume $\pi_1 g(n) a \leq \pi_1 g(n)(a+1)$.

$$\pi_1 g(n) a \leq \pi_1 g(n)(a+1)$$

$$\pi_1 g(n) a \vee 2 +_c g(n) a \leq \pi_1 g(n)(a+1) \vee 2 +_c g(n)(a+1)$$
$$\pi_1 g(n+1) a \leq \pi_1 g(n+1)(a+1)$$

$\square$

Now we show $\pi_1 g(n) \leq \pi_1 g(n+1)$.

Proof. By reflexivity, $\pi_1 g(n) \leq \pi_1 g(n)$. By the lemma we just proved:

$$\pi_1 g(n) a \leq \pi_1 g(n)(a+1)$$

$$\pi_1 g(n) a \leq 2 +_c \pi_1 g(n)(a+1)$$
$$\lambda a. \pi_1 g(n) a \leq \lambda a. 2 +_c \pi_1 g(n)(a+1)$$

$\square$

So since for all n, $\pi_0 g(n) = 1$ and $\pi_1 g(n) \leq \lambda a. 2 +_c \pi_1 g(n)(a+1)$, we can say

$$g(n) \leq \langle 1, \lambda a. 2 +_c \pi_1 g(n)(a+1) \rangle)$$

So

$$g(n+1) = \langle 1, \lambda a. 2 +_c \pi_1 g(n)(a+1) \rangle$$

To extract a recurrence from g, we apply g to the interpretation of a list a.

Let $h(n, a) = \pi_1 g(n) a$

For $n = 0$

$$\begin{aligned} h(0, a) &= \pi_1 g(0) a \\ &= (\lambda a. \langle 0, a\rangle) a \\ &= \langle 0, a\rangle \end{aligned}$$

For $n > 0$

$$\begin{aligned} h(n, a) &= \pi_1 g(n) a \\ &= (\lambda a. 2 +_c \pi_1 g(n-1)(a+1)) a \\ &= 2 +_c \pi_1 g(n-1)(a+1)) \\ &= 2 +_c h(n-1, a+1) \\ &= \langle 2 + \pi_0 h(n-1, a+1), \pi_1 h(n-1, a+1)\rangle \end{aligned}$$

From this recurrence, we can extract a recurrence for the cost. Let $h_c = \pi_0 \circ h$.

For $n = 0$

$$\begin{aligned} h_c(0, a) &= \pi_0 h(0, a) \\ &= \pi_0 \langle 0, a\rangle \\ &= 0 \end{aligned}$$

For $n > 0$

$$\begin{aligned} h_c(n, a) &= \pi_0 \langle 2 + \pi_0 h(n-1, a+1), \pi_1 h(n-1, a+1)\rangle \\ &= 2 + \pi_0 h(n-1, a+1) \\ &= 2 + h_c(n-1, a+1) \end{aligned}$$

We now have a recurrence for the cost of the auxilary function `rec(xs,...)`:

$$\boxed{h_c(n, a) = \begin{cases} 0 & n = 0 \\ 2 + h_c(n-1, a+1) & n > 0 \end{cases} \qquad (1)}$$

Theroem: $h_c(n, a) = 2n$

Proof. We prove this by induction on n.

Base case: $n = 0$

$$h_c(0, a) = 0 = 2 \cdot 0$$

Induction case:

We assume $h_c(n, a+1) = 2n$.

$$h_c(n+1, a) = 2 + h_c(n, a+1) = 2 + 2n = 2(n+1)$$

$\square$

The solution to the recurrence for the cost of the auxilary function `rec(xs,...)` is:

$$\boxed{h_c(n, a) = 2n}$$

We can also extract a recurrence for the potential. Let $h_p = \pi_1 \circ h$.

For $n = 0$

$$\begin{aligned} h_p(0, a) &= \pi_1 h(0, a) \\ &= \pi_1 \langle 0, a \rangle \\ &= a \end{aligned}$$

For $n > 0$

$$\begin{aligned} h_p(n, a) &= \pi_1 \langle 2 + \pi_0 h(n-1, a+1), \pi_1 h(n-1, a+1) \rangle \\ &= \pi_1 h(n-1, a+1) \\ &= h_p(n-1, a+1) \end{aligned}$$

We now have a recurrence for the potential of the auxilary function in `rev xs`:

$$\boxed{h_p(n, a) = \begin{cases} a & n = 0 \\ h_p(n-1, a+1) & n > 0 \end{cases} \qquad (2)}$$

Theroem: $h_p(n, a) = n + a$

Proof. We prove this by induction on n.

Base case: $n = 0$

$$h_p(0, a) = a$$

Induction case:

$$h_p(n, a) = h_p(n-1, a+1) = n - 1 + a + 1 = n + a$$

□

So the solution to the recurrence for the potential of the auxilary function.

$$\boxed{h_p(n, a) = n + a}$$

We return to our interpretation of `rev xs`.

$[\![\|\texttt{rev xs}\|]\!] = \langle 1 + [\![(\texttt{rec(xs,...)}_p\ \texttt{Nil})_c]\!], [\![(\texttt{rec(xs,...)}_p\ \texttt{Nil})_p]\!]\rangle$
$= \langle 1 + \pi_0([\![(\texttt{rec(xs,...)}_p]\!]0), \pi_1([\![\texttt{rec(xs,...)}_p]\!]0)\rangle$
$= \langle 1 + \pi_0(\pi_1 g(n)\ \ 0), \pi_1(\pi_1 g(n)\ \ 0)\rangle$ `where` n `is the length of xs`
$= \langle 1 + \pi_0 h(n, 0), \pi_1 h(n, 0)\rangle$
$= \langle 1 + h_c(n, 0), h_p(n, 0)\rangle$
$= \langle 1 + 2n, n\rangle$

This result tells us the cost of applying `rev` to a list `xs` of length n is $1 + 2n$, and the resulting list has size n. So `rev` $= \Theta(n)$.

Translation of `rev` using `split`

The linear time reversal function using splits instead of the matching syntactic sugar is written as follows:

```
rev xs = λxs.rec(xs,
                 Nil ↦ λa.a,
                 Cons↦b.split(b,x.c.split(c,xs'.r.
                            λa.force(r) Cons⟨x,a⟩))) Nil
```

Like before, we will begin by translating the `rec` construct.

$$
\begin{aligned}
\|\texttt{rec(...)}\| = \|&\texttt{rec(xs, Nil} \mapsto \lambda\texttt{a.a,}\\
&\quad\texttt{Cons}\mapsto\texttt{b.split(b,x.c.split(c,xs'.r.}\\
&\qquad\qquad\lambda\texttt{a.force(r) Cons}\langle\texttt{x,a}\rangle\texttt{)))}\|
\end{aligned}
$$

$$
\begin{aligned}
= \langle\texttt{0,xs}\rangle_c +_c \texttt{ rec(}\langle\texttt{0,xs}\rangle_p\texttt{, Nil} &\mapsto 1 +_c \|\lambda\texttt{a.a}\|,\\
\texttt{Cons}&\mapsto\texttt{b}.1 +_c \|\texttt{split(b,x.c.split(c,xs'.r.}\\
&\qquad\lambda\texttt{a.force(r) Cons}\langle\texttt{x,a}\rangle\texttt{))}\|\texttt{)}
\end{aligned}
$$

$$
\begin{aligned}
= \texttt{rec(xs, Nil} &\mapsto 1 +_c \|\lambda\texttt{a.a}\|,\\
\texttt{Cons}&\mapsto\texttt{b}.1 +_c \|\texttt{split(b,x.c.split(c,xs'.r.}\\
&\qquad\lambda\texttt{a.force(r) Cons}\langle\texttt{x,a}\rangle\texttt{))}\|\texttt{)}
\end{aligned}
$$

The translation of the `Nil` branch is simple.

$$= 1 +_c \|\lambda\texttt{a.a}\|$$

$$= 1 +_c \langle 0, \lambda\texttt{a.}\|\texttt{a}\|\rangle$$

$$= 1 +_c \langle 0, \lambda\texttt{a.}\langle 0, \texttt{a}\rangle\rangle$$

$$= \langle 1, \lambda\texttt{a.}\langle 0, \texttt{a}\rangle\rangle$$

The translation of the `Cons` branch is a slightly more involved.

$$
\begin{aligned}
= \texttt{Cons}\mapsto\texttt{b}.1 +_c &\|\texttt{split(b,x.c.split(c,xs'.r.}\\
&\lambda\texttt{a.force(r) Cons}\langle\texttt{x,a}\rangle\texttt{))}\|\texttt{)}
\end{aligned}
$$

$$
\begin{aligned}
= \texttt{Cons}\mapsto\texttt{b}.1 +_c &\|\texttt{b}\|_c +_c \|\texttt{split(c,xs'.r.}\\
&\lambda\texttt{a.force(r) Cons}\langle\texttt{x,a}\rangle\texttt{)}\|[\pi_0\|\texttt{b}\|_p/x, \pi_1\|\texttt{b}\|_p/c]
\end{aligned}
$$

The translation of the type of `b` will illuminate the translation of the `split`. The type of `b` is `b`::$\texttt{int}\times\langle\texttt{list}\times\langle\texttt{list}\to\texttt{list}\rangle\rangle$.
The type of $\|\texttt{b}\|$ is $\langle\texttt{C}\times\langle\texttt{int}\times\langle\texttt{list}\times\langle\texttt{C}\times\texttt{list}\to \langle\texttt{C}\times\texttt{list}\rangle\rangle\rangle\rangle\rangle$. We can say that $\pi_0\|\texttt{b}\|_p$ is the head of the list `xs`, $\pi_0\pi_1\|\texttt{b}\|_p$ is the tail of the list `xs`, and $\pi_1\pi_1\|\texttt{b}\|_p$ is the result of the recursive call.

$$
\begin{aligned}
= \texttt{Cons}\mapsto\texttt{b}.1 +_c &\langle 0, \|\texttt{b}\|_p\rangle_c +_c \|\texttt{split(c,xs'.r.}\\
&\lambda\texttt{a.force(r) Cons}\langle\texttt{x,a}\rangle\|[\pi_0\|\texttt{b}\|_p/x, \pi_1\|\texttt{b}\|_p/c]
\end{aligned}
$$

$$= \texttt{Cons}\mapsto\texttt{b}.1 +_c 0 +_c (\|\texttt{c}\|_c +_c$$

$$(\|\lambda \texttt{a}\,.\,\texttt{force(r)}\ \ \texttt{Cons}\langle \texttt{x},\texttt{a}\rangle\|)[\pi_0\|\texttt{c}\|_p/xs', \pi_1\|\texttt{c}\|_p/r])[\pi_0\|\texttt{b}\|_p/x, \pi_1\|\texttt{b}\|_p/c]$$

$$= \texttt{Cons}\mapsto\texttt{b}\,.\,1 +_c (\|\texttt{c}\|_c +_c \langle 0, \lambda\texttt{a}\,.\,\|\texttt{force(r)}\ \ \texttt{Cons}\langle\texttt{x},\texttt{a}\rangle\|\rangle[\pi_0\|\texttt{c}\|_p/xs', \pi_1\|\texttt{c}\|_p/r])[\pi_0\|\texttt{b}\|_p/x, \pi_1\|\texttt{b}\|_p/c]$$

Let us focus on the translation of $\|\texttt{force(r) Cons}\langle\texttt{x,a}\rangle\|$.

$$= (1 + \|\texttt{force(r)}\|_c + \|\texttt{Cons}\langle\texttt{x},\texttt{a}\|_c) +_c \|\texttt{force(r)}\|_p\|\texttt{Cons}\langle\texttt{x},\texttt{a}\rangle\|_p$$

$$= (1 + (\|\texttt{r}\|_c +_c \|\texttt{r}\|_p)_c + \|\texttt{Cons}\langle\texttt{x},\texttt{a}\rangle\|_c) +_c (\|\texttt{r}\|_c +_c \|\texttt{r}\|_p)_p\|\texttt{Cons}\langle\texttt{x},\texttt{a}\rangle\|_p$$

$$= (1 + (\|\texttt{r}\|_c +_c \|\texttt{r}\|_p)_c + \|\texttt{Cons}\langle\texttt{x},\texttt{a}\rangle\|_c) +_c (\|\texttt{r}\|_c +_c \|\texttt{r}\|_p)_p\|\texttt{Cons}\langle\texttt{x},\texttt{a}\rangle\|_p$$

$$= (1 + \|\texttt{r}\|_c + \|\texttt{Cons}\langle\texttt{x},\texttt{a}\rangle\|_c) +_c \|\texttt{r}\|_p\|\texttt{Cons}\langle\texttt{x},\texttt{a}\rangle\|_p$$

$$= (1 + \|\texttt{r}\|_c + (\langle\|\langle\texttt{x},\texttt{a}\rangle\|_c, \texttt{Cons}\|\langle\texttt{x},\texttt{a}\rangle\|_p\rangle)_c) +_c \|\texttt{r}\|_p\langle\|\langle\texttt{x},\texttt{a}\rangle\|_c, \texttt{Cons}\|\langle\texttt{x},\texttt{a}\rangle\|_p\rangle_p$$

$$= (1 + \|\texttt{r}\|_c + \|\langle\texttt{x},\texttt{a}\rangle\|_c) +_c \|\texttt{r}\|_p\texttt{Cons}\|\langle\texttt{x},\texttt{a}\rangle\|_p$$

$$= (1 + \|\texttt{r}\|_c + \langle\|\texttt{x}\|_c + \|\texttt{a}\|_c, \langle\|\texttt{x}\|_p, \|\texttt{a}\|_p\rangle\rangle_c) +_c \|\texttt{r}\|_p\texttt{Cons}\langle\|\texttt{x}\|_c + \|\texttt{a}\|_c, \langle\|\texttt{x}\|_p, \|\texttt{a}\|_p\rangle\rangle_p$$

$$= (1 + \|\texttt{r}\|_c + (\|\texttt{x}\|_c + \|\texttt{a}\|_c)) +_c \|\texttt{r}\|_p\texttt{Cons}\langle\|\texttt{x}\|_p, \|\texttt{a}\|_p\rangle$$

$$= (1 + \texttt{r}_c + (\langle\texttt{0},\texttt{x}\rangle_c + \langle\texttt{0},\texttt{a}\rangle_c)) +_c \langle\texttt{0},\texttt{r}\rangle_p\texttt{Cons}\langle\langle\texttt{0},\texttt{x}\rangle_p, \langle\texttt{0},\texttt{a}\rangle_p\rangle$$

$$= (1 + \texttt{r}_c + 0 + 0) +_c \texttt{r}_p\texttt{Cons}\langle\texttt{x},\texttt{a}\rangle$$

$$= (1 + \texttt{r}_c) +_c \texttt{r}_p\texttt{Cons}\langle\texttt{x},\texttt{a}\rangle$$

We can now use this in our translation of the `Cons` case.

$$= \texttt{Cons}\mapsto\texttt{b}\,.\,1 +_c (\|\texttt{c}\|_c +_c \langle 0, \lambda\texttt{a}\,.\,(1 + \texttt{r}_c) +_c \texttt{r}_p\texttt{Cons}\langle\texttt{x},\texttt{a}\rangle\rangle[\pi_0\|\texttt{c}\|_p/xs', \pi_1\|\texttt{c}\|_p/r])[\pi_0\|\texttt{b}\|_p/x, \pi_1\|\texttt{b}\|_p/c]$$

$$= \texttt{Cons}\mapsto\texttt{b}\,.\,1 +_c (\|\texttt{c}\|_c +_c \langle 0, \lambda\texttt{a}\,.\,(1 + (\pi_1\|\texttt{c}\|_p)_c) +_c (\pi_1\|\texttt{c}\|_p)_p\texttt{Cons}\langle\texttt{x},\texttt{a}\rangle\rangle)[\pi_0\|\texttt{b}\|_p/x, \pi_1\|\texttt{b}\|_p/c]$$

$$= \texttt{Cons}\mapsto\texttt{b}\,.\,1 +_c (\|\pi_1\|\texttt{b}\|_p\|_c +_c \langle 0, \lambda\texttt{a}\,.\,(1 + (\pi_1\|\pi_1\|\texttt{b}\|_p\|_p)_c) +_c (\pi_1\|\pi_1\|\texttt{b}\|_p\|_p)_p\texttt{Cons}\langle\pi_1\|\texttt{b}\|_p, \texttt{a}\rangle\rangle)$$

$$= \texttt{Cons}\mapsto \texttt{b}\,.\,1 +_c (\|\pi_1\langle \texttt{0}\,,\texttt{b}\rangle_p\|_c +_c \\ \langle 0, \lambda\texttt{a}\,.\,(1 + (\pi_1\|\pi_1\langle \texttt{0}\,,\texttt{b}\rangle_p\|_p)_c) +_c (\pi_1\|\pi_1\langle \texttt{0}\,,\texttt{b}\rangle_p\|_p)_p \texttt{Cons}\langle \pi_1\langle \texttt{0}\,,\texttt{b}\rangle_p\,,\texttt{a}\rangle\rangle)$$

$$= \texttt{Cons}\mapsto \texttt{b}\,.\,1 +_c (\|\pi_1\texttt{b}\|_c +_c \\ \langle 0, \lambda\texttt{a}\,.\,(1 + (\pi_1\|\pi_1\texttt{b}\|_p)_c) +_c (\pi_1\|\pi_1\texttt{b}\|_p)_p \texttt{Cons}\langle \pi_1\texttt{b}\,,\texttt{a}\rangle\rangle)$$

$$= \texttt{Cons}\mapsto \texttt{b}\,.\,1 +_c (\langle \texttt{0}\,,\pi_1\texttt{b}\rangle_c +_c \\ \langle 0, \lambda\texttt{a}\,.\,(1 + (\pi_1\langle \texttt{0}\,,\pi_1\texttt{b}\rangle_p)_c) +_c (\pi_1\langle \texttt{0}\,,\pi_1\texttt{b}\rangle_p)_p \texttt{Cons}\langle \pi_1\texttt{b}\,,\texttt{a}\rangle\rangle)$$

$$= \texttt{Cons}\mapsto \texttt{b}\,.\,1 +_c \langle 0, \lambda\texttt{a}\,.\,(1 + (\pi_1\pi_1\texttt{b})_c) +_c (\pi_1\pi_1\texttt{b})_p \texttt{Cons}\langle \pi_1\texttt{b}\,,\texttt{a}\rangle\rangle$$

$$= \texttt{Cons}\mapsto \texttt{b}\,.\,\langle 1, \lambda\texttt{a}\,.\,(1 + (\pi_1\pi_1\texttt{b})_c) +_c (\pi_1\pi_1\texttt{b})_p \texttt{Cons}\langle \pi_1\texttt{b}\,,\texttt{a}\rangle\rangle$$

$$\|\texttt{rec}\,(\texttt{xs}\,,\ldots)\| \;=\; \texttt{rec}\,(\texttt{xs}\,,\texttt{Nil}\mapsto \langle \texttt{1}\,,\lambda\texttt{a}\,.\,\langle \texttt{0}\,,\texttt{a}\rangle\rangle\,, \\ \texttt{Cons}\mapsto \texttt{b}\,.\,\langle 1, \lambda\texttt{a}\,.\,(1 + (\pi_1\pi_1\texttt{b})_c) +_c (\pi_1\pi_1\texttt{b})_p \texttt{Cons}\langle \pi_1\texttt{b}\,,\texttt{a}\rangle\rangle)$$

So our translation of `rev` is

$$\|\texttt{rev}\| \;=\; \langle \texttt{0}\,,\lambda\texttt{xs}\,.\,\texttt{rec}\,(\texttt{xs}\,,\texttt{Nil}\mapsto \langle \texttt{1}\,,\lambda\texttt{a}\,.\,\langle \texttt{0}\,,\texttt{a}\rangle\rangle\,, \\ \texttt{Cons}\mapsto \texttt{b}\,.\,\langle 1, \lambda\texttt{a}\,.\,(1 + (\pi_1\pi_1\texttt{b})_c) +_c (\pi_1\pi_1\texttt{b})_p \texttt{Cons}\langle \pi_1\texttt{b}\,,\texttt{a}\rangle\rangle)\rangle$$

We observe that in both cases of the `rec`, the cost of the recursive call is 1, so we can replace $\pi_1\pi_1\texttt{b}_c$ with 1.

$$\|\texttt{rev}\| \;=\; \langle \texttt{0}\,,\lambda\texttt{xs}\,.\,\texttt{rec}\,(\texttt{xs}\,,\texttt{Nil}\mapsto \langle \texttt{1}\,,\lambda\texttt{a}\,.\,\langle \texttt{0}\,,\texttt{a}\rangle\rangle\,, \\ \texttt{Cons}\mapsto \texttt{b}\,.\,\langle 1, \lambda\texttt{a}\,.\,(2 +_c (\pi_1\pi_1\texttt{b})_p \texttt{Cons}\langle \pi_1\texttt{b}\,,\texttt{a}\rangle\rangle)\rangle$$

We are interested in the interpretation of `rev xs`.

$$\|\texttt{rev xs}\| \;=\; (1 + \|\texttt{rev}\|_c\|\texttt{xs}\|_c) +_c \|\texttt{rev}\|_p\|\texttt{xs}\|_p$$

$$= (1 + \langle \texttt{0}\,,\lambda\texttt{xs}\,.\,\texttt{rec}\,(\ldots)\rangle_c + \langle \texttt{0}\,,\texttt{xs}\rangle_c) +_c \|\texttt{rev}\|_p\langle \texttt{0}\,,\texttt{xs}\rangle_p$$

$$= (1 + 0 + 0) +_c (\lambda\texttt{xs}\,.\,\texttt{rec}\,(\texttt{xs}\,,\texttt{Nil}\mapsto \langle \texttt{1}\,,\lambda\texttt{a}\,.\,\langle \texttt{0}\,,\texttt{a}\rangle\rangle\,, \\ \texttt{Cons}\mapsto \texttt{b}\,.\,\langle 1, \lambda\texttt{a}\,.\,(2 +_c (\pi_1\pi_1\texttt{b})_p \texttt{Cons}\langle \pi_1\texttt{b}\,,\texttt{a}\rangle\rangle))_p\texttt{xs}$$

$$= 1 +_c \texttt{rec}\,(\texttt{xs}\,,\texttt{Nil}\mapsto \langle \texttt{1}\,,\lambda\texttt{a}\,.\,\langle \texttt{0}\,,\texttt{a}\rangle\rangle\,, \\ \texttt{Cons}\mapsto \texttt{b}\,.\,\langle 1, \lambda\texttt{a}\,.\,(2 +_c (\pi_1\pi_1\texttt{b})_p \texttt{Cons}\langle \pi_1\texttt{b}\,,\texttt{a}\rangle\rangle)$$

Interpretation

We intepret the size of an `list` to be the number of list constructors.

$[\![\ \texttt{list}\ ]\!] = \mathbb{N}^\infty$
$D^{list} = \{*\} + \{1\} \times \mathbb{N}^\infty$
$size_{list}(\texttt{Nil}) = 1$
$size_{list}(\texttt{Cons(1,n)}) = 1 + n$

The interpretation of `rev xs` proceeds as follows.

$[\![\|\texttt{rev xs}\|]\!] = [\![1 +_c \texttt{rec(xs,...)}_p\ \texttt{Nil}]\!]$
$= [\![\langle 1 + (\texttt{rec(xs,...)}_p\ \texttt{Nil})_c, (\texttt{rec(xs,...)}_p\ \texttt{Nil})_p\rangle]\!]$
$= \langle 1 + [\![(\texttt{rec(xs,...)}_p\ \texttt{Nil})_c]\!], [\![(\texttt{rec(xs,...)}_p\ \texttt{Nil})_p]\!]\rangle$

We will focus on the interpretation of the auxilary function `rec(xs,...)`.

Let $g(n) = [\![\texttt{rec(xs,...)}]\!]\{xs \mapsto n\}$

$$g(n) = \bigvee_{size\ ys \leq n} case(ys, Nil \mapsto \langle 1, \lambda a.\langle 0, a\rangle\rangle, Cons \mapsto \langle 1, m\rangle.\langle 1, \lambda a.2 +_c \pi_1 g(m)(a+1)\rangle)$$

For $n = 0$, $g(0) = \langle 1, \lambda a.\langle 0, a\rangle\rangle$.

For $n > 0$,

$$g(n+1) = \bigvee_{size\ ys \leq n+1} case(ys, Nil \mapsto \langle 1, \lambda a.\langle 0, a\rangle\rangle, Cons \mapsto \langle 1, m\rangle.\langle 1, \lambda a.2 +_c \pi_1 g(m)(a+1)\rangle)$$

$$g(n+1) = \bigvee_{size\ ys \leq n} case(ys, Nil \mapsto \langle 1, \lambda a.\langle 0, a\rangle\rangle, Cons \mapsto \langle 1, m\rangle.\langle 1, \lambda a.2 +_c \pi_1 g(m)(a+1)\rangle)$$

$$\vee \bigvee_{size\ ys = n+1} case(ys, Nil \mapsto \langle 1, \lambda a.\langle 0, a\rangle\rangle, Cons \mapsto \langle 1, m\rangle.\langle 1, \lambda a.2 +_c \pi_1 g(m)(a+1)\rangle)$$

$$g(n+1) = g(n) \vee \bigvee_{size\ ys = n+1} case(ys, Nil \mapsto \langle 1, \lambda a.\langle 0, a\rangle\rangle, Cons \mapsto \langle 1, m\rangle.\langle 1, \lambda a.2 +_c \pi_1 g(m)(a+1)\rangle)$$

$$g(n+1) = g(n) \vee \langle 1, \lambda a.2 +_c \pi_1 g(n)(a+1) \rangle)$$

We want to show that g is monotonically increasing; $\forall n.g(n) \leq g(n{+}1)$. By definition of $\leq$, $g(n) \leq g(n{+}1) \Leftrightarrow \pi_0 g(n) \leq \pi_0 g(n{+}1) \wedge \pi_1 g(n) \leq \pi_1 g(n{+}1)$. First we will show $\forall n.\pi_0 g(n) = 1$, the immediate corollary of which is $\forall n.\pi_0 g(n) \leq \pi_0 g(n+1)$.

Proof. We prove this by induction on n.

Base case: $n = 0$

By definition, $\pi_0 g(0) = 1$.

Induction step: $n > 0$

By definition $\pi_0 g(n+1) = \pi_0(g(n) \vee \langle 1, \lambda a.2 +_c \pi_1 g(n)(a+1)\rangle)$. We distribute the projection over the max: $\pi_0 g(n+1) = \pi_0 g(n) \vee 1$. By the induction hypothesis, $\pi_0 g(n) = 1$, so $\pi_0 g(n+1) = 1$.

$\square$

Now we argue that $\pi_1 g(n) \leq \pi_1 g(n+1)$. First we prove the lemma $\forall n.\pi_1 g(n) a \leq \pi_1 g(n)(a+1)$.

Proof. We prove this by induction on n.

$n = 0$

$\pi_1 g(0) a = \langle 0, a \rangle \leq \pi_1 g(0)(a+1) = \langle 0, a+1 \rangle$.

$n > 0$

We assume $\pi_1 g(n) a \leq \pi_1 g(n)(a+1)$.

$$\pi_1 g(n) a \leq \pi_1 g(n)(a+1)$$

$$\pi_1 g(n) a \vee 2 +_c g(n) a \leq \pi_1 g(n)(a+1) \vee 2 +_c g(n)(a+1)$$
$$\pi_1 g(n+1) a \leq \pi_1 g(n+1)(a+1)$$

$\square$

Now we show $\pi_1 g(n) \leq \pi_1 g(n+1)$.

Proof. By reflexivity, $\pi_1 g(n) \leq \pi_1 g(n)$. By the lemma we just proved:

$$\pi_1 g(n)a \leq \pi_1 g(n)(a+1)$$

$$\pi_1 g(n)a \leq 2 +_c \pi_1 g(n)(a+1)$$

$$\lambda a.\pi_1 g(n)a \leq \lambda a.2 +_c \pi_1 g(n)(a+1)$$

$\square$

So since for all n, $\pi_0 g(n) = 1$ and $\pi_1 g(n) \leq \lambda a.2 +_c \pi_1 g(n)(a+1)$, we can say

$$g(n) \leq \langle 1, \lambda a.2 +_c \pi_1 g(n)(a+1) \rangle)$$

So

$$g(n+1) = \langle 1, \lambda a.2 +_c \pi_1 g(n)(a+1) \rangle$$

To extract a recurrence from g, we apply g to the interpretation of a list a.

Let $h(n,a) = \pi_1 g(n)a$

For $n = 0$

$$\begin{aligned} h(0,a) &= \pi_1 g(0)a \\ &= (\lambda a.\langle 0, a\rangle)a \\ &= \langle 0, a \rangle \end{aligned}$$

For $n > 0$

$$\begin{aligned} h(n,a) &= \pi_1 g(n)a \\ &= (\lambda a.2 +_c \pi_1 g(n-1)(a+1))a \\ &= 2 +_c \pi_1 g(n-1)(a+1)) \\ &= 2 +_c h(n-1, a+1) \\ &= \langle 2 + \pi_0 h(n-1, a+1), \pi_1 h(n-1, a+1) \rangle \end{aligned}$$

From this recurrence, we can extract a recurrence for the cost. Let $h_c = \pi_0 \circ h$.

For $n = 0$

$$\begin{aligned} h_c(0,a) &= \pi_0 h(0,a) \\ &= \pi_0 \langle 0, a \rangle \\ &= 0 \end{aligned}$$

For $n > 0$

$$\begin{aligned} h_c(n, a) &= \pi_0\langle 2 + \pi_0 h(n-1, a+1), \pi_1 h(n-1, a+1)\rangle \\ &= 2 + \pi_0 h(n-1, a+1) \\ &= 2 + h_c(n-1, a+1) \end{aligned}$$

We now have a recurrence for the cost of the auxilary function `rec(xs,...)`:

$$h_c(n, a) = \begin{cases} 0 & n = 0 \\ 2 + h_c(n-1, a+1) & n > 0 \end{cases} \tag{3}$$

Theroem: $h_c(n, a) = 2n$

Proof. We prove this by induction on n.

Base case: $n = 0$

$$h_c(0, a) = 0 = 2 \cdot 0$$

Induction case:

We assume $h_c(n, a+1) = 2n$.

$$h_c(n+1, a) = 2 + h_c(n, a+1) = 2 + 2n = 2(n+1)$$

□

The solution to the recurrence for the cost of the auxilary function `rec(xs,...)` is:

$$h_c(n, a) = 2n$$

We can also extract a recurrence for the potential. Let $h_p = \pi_1 \circ h$.

For $n = 0$

$$\begin{aligned} h_p(0, a) &= \pi_1 h(0, a) \\ &= \pi_1\langle 0, a\rangle \\ &= a \end{aligned}$$

For $n > 0$

$$\begin{aligned} h_p(n, a) &= \pi_1 \langle 2 + \pi_0 h(n-1, a+1), \pi_1 h(n-1, a+1) \rangle \\ &= \pi_1 h(n-1, a+1) \\ &= h_p(n-1, a+1) \end{aligned}$$

We now have a recurrence for the potential of the auxilary function in `rev xs`:

$$h_p(n, a) = \begin{cases} a & n = 0 \\ h_p(n-1, a+1) & n > 0 \end{cases} \tag{4}$$

Theroem: $h_p(n, a) = n + a$

Proof. We prove this by induction on n.

Base case: $n = 0$

$$h_p(0, a) = a$$

Induction case:

$$h_p(n, a) = h_p(n-1, a+1) = n - 1 + a + 1 = n + a$$

$\square$

So the solution to the recurrence for the potential of the auxilary function.

$$h_p(n, a) = n + a$$

We return to our interpretation of `rev xs`.

$$\begin{aligned} &[\![\| \texttt{rev xs} \|]\!] = \langle 1 + [\![(\texttt{rec(xs,...)}_p \ \texttt{Nil})_c]\!], [\![(\texttt{rec(xs,...)}_p \ \texttt{Nil})_p]\!] \rangle \\ &= \langle 1 + \pi_0([\![(\texttt{rec(xs,...)}_p]\!] 0), \pi_1([\![\texttt{rec(xs,...)}_p]\!] 0) \rangle \\ &= \langle 1 + \pi_0(\pi_1 g(n) \ \ 0), \pi_1(\pi_1 g(n) \ \ 0) \rangle \ \texttt{where } n \texttt{ is the length of xs} \\ &= \langle 1 + \pi_0 h(n, 0), \pi_1 h(n, 0) \rangle \\ &= \langle 1 + h_c(n, 0), h_p(n, 0) \rangle \\ &= \langle 1 + 2n, n \rangle \end{aligned}$$

This result tells us the cost of applying `rev` to a list `xs` of length n is $1 + 2n$, and the resulting list has size n. So $\texttt{rev} = \Theta(n)$.