CS 4110 – 9/5/25
Lab # 5 – Induction

I. ML Semantics

Recall our "Mini List" language from last time.

$$
\begin{aligned}
e ::= \ & n \\
| \ & [\,] \\
| \ & e_1 :: e_2 \\
| \ & e_1 @ e_2 \\
| \ & \text{rev } e \\
| \ & \text{len } e
\end{aligned}
\qquad
\begin{aligned}
\ell ::= \ & [\,] \\
| \ & n :: \ell \\
\\
v ::= \ & n \\
| \ & \ell
\end{aligned}
$$

We can define a "big-step" semantics as a relation on expressions and values. We write $e \Downarrow v$

$\boxed{e \Downarrow v}$

$$\frac{}{n \Downarrow n}\ \text{Int} \qquad \frac{}{[\,] \Downarrow [\,]}\ \text{Nil} \qquad \frac{e_1 \Downarrow n_1 \quad e_2 \Downarrow \ell_2}{e_1 :: e_2 \Downarrow n_1 :: \ell_2}\ \text{Cons}$$

$$\frac{e_1 \Downarrow [\,] \quad e_2 \Downarrow l_2}{e_1 @ e_2 \Downarrow l_2}\text{ App Nil}$$

$$\frac{e_1 \Downarrow n_1 :: l_1 \quad e_2 \Downarrow l_2 \quad l_1 @ l_2 \Downarrow l'}{e_1 @ e_2 \Downarrow n_1 :: l'}\text{ AppCons}$$

$$\frac{e \Downarrow [\,]}{rev\ e \Downarrow [\,]}\text{ Rev Nil}$$

$$\frac{e \Downarrow n :: l \quad rev\ l \Downarrow l_r \quad l_r @ (n :: [\,]) \Downarrow l'}{rev\ e \Downarrow l'}\text{ Rev Cons}$$

$$\frac{e \Downarrow [\,]}{len\ e \Downarrow 0}\text{ LenNil}$$

$$\frac{e \Downarrow n :: l \quad len\ l \Downarrow n_l \quad n' = n_l + 1}{len\ e \Downarrow n'}\text{ LenCons}$$

Proposition: If $len\ l_0 \Downarrow n_0$ and $rev\ l_0 \Downarrow l_0'$ then $len\ l_0' \Downarrow n_0$.

Proof: By induction on l_0.

Case $[\,]$:

Assume $len\ [\,] \Downarrow n_0$ and $rev\ [\,] \Downarrow l_0'$.

By Inversion*, $n_0 = 0$ and $l_0' = [\,]$

The result follows by LenNil and Nil, as shown in the following derivation:

$$\frac{\dfrac{}{[\,] \Downarrow [\,]}\text{ Nil}}{len\ [\,] \Downarrow 0}\text{ LenNil}$$

✓✓

means the case is finished

*

means a deferred Lemma.

Case $m::l$:

Assume $\text{len } (m::l) \Downarrow n_o$ and $\text{rev } (m::l) \Downarrow l_o$.

By Inversion*, $\text{len } l \Downarrow n$ and $n_o = n+1$.

By Inversion*, $\text{rev } l \Downarrow l_r$ and $l_r @ (m::[]) \Downarrow l_o$.

By Total*, $\text{len } l_r \Downarrow n_r$.

We can build the following derivation, using Total*

$$\dfrac{\dfrac{}{m::[] \Downarrow m::[]}\,\text{Total} \qquad \dfrac{\dfrac{}{[] \Downarrow []}\,\text{Nil}}{\text{len } [] \Downarrow 0}\,\text{LenNil} \qquad 1=0+1}{\text{len } (m::[]) \Downarrow 1}\,\text{LenCons}$$

By AppLength*, $\text{len } l_o \Downarrow n_o'$ and $n_o' = n_r + 1$

By IH, $n_r = n$.

So, $n_o' = n_r + 1$
$= n + 1$
$= n_o$ ✓

Supporting Lemmas

Lemma [Inversion]

- if $e \Downarrow []$ and $\text{len } e \Downarrow n'$ then $n' = 0$.
- if $e \Downarrow []$ and $\text{rev } e \Downarrow l'$ then $l' = []$.
- if $e \Downarrow m::l$ and $\text{len } e \Downarrow n'$ then $\exists n.\ \text{len } l \Downarrow n$ and $n' = n+1$.
- if $e \Downarrow m::l$ and $\text{rev } e \Downarrow l'$ then $\exists l_r.\ \text{rev } l \Downarrow l_r$ and $l_r @ (m::[]) \Downarrow l'$.
- if $e_1 \Downarrow []$ and $e_1 @ e_2 \Downarrow l'$ then $e_2 \Downarrow l_2$ and $l' = l_2$.
- if $e_1 \Downarrow m::l_1$ and $e_1 @ e_2 \Downarrow l'$ then $e_2 \Downarrow l_2$ and $l_1 @ l_2 \Downarrow l''$ and $l' = m::l''$.

Lemma [Total]

- $\forall v.\ v \Downarrow v$
- $\forall l.\ \exists l'.\ \text{rev } l \Downarrow l'$
- $\forall l.\ \exists n'.\ \text{len } l \Downarrow n'$.

Lemma [App Length]

If $l_1 \Downarrow n_1$ and $l_2 \Downarrow n_2$ and $l_1 @ l_2 \Downarrow l'$,
Then $\text{len } l' \Downarrow n$ where $n = n_1 + n_2$

Proof. By induction on l_1.

Case []:

By Inversion*, $n_1 = 0$ and $l' = l_2$.
We have $\text{len } l_2 \Downarrow n_2$ by assumption,
and $n = 0 + n_2$. ✓
✓✓

Case $m_1 :: l_1'$:

By Inversion*, $\text{len } l_1' \Downarrow n_1'$ and $n_1 = n_1' + 1$.
Also by Inversion*, $(m_1 :: l_1') @ l_2 \Downarrow m_1 :: l'$ where $l_1' @ l_2 \Downarrow l'$.
By IH, $\text{len } l' \Downarrow n'$ and $n' = n_1' + n_2$.
Note that $n' + 1 = n_1' + 1 + n_2 = n_1 + n_2 = n$.
We can construct the following derivation

$$\dfrac{\dfrac{}{m_1 :: l' \Downarrow m_1 :: l'}\,\text{Total*} \qquad \dfrac{}{\text{len } l' \Downarrow n}\,\text{IH} \qquad n = n' + 1}{\text{len } (m_1 :: l') \Downarrow n}\,\text{Len Cons}$$

✓✓ □