Some expanded proofs from the book [1]
Feedback is welcome

Andrey Rybalchenko
rybal@in.tum.de

TUM

Definition 1. Let X and Y be sets.

$\mathcal{L}(X) = \{nil\} \cup (X \times \mathcal{L}(X))$	lists of elements from X
$\lvert \_ \rvert : \mathcal{L}(X) \to \mathbb{N}$	length
$\lvert nil \rvert = 0$	Eq. $\lvert \_ \rvert_1$
$\lvert x :: xr \rvert = 1 + \lvert xr \rvert$	Eq. $\lvert \_ \rvert_2$
$@ : \mathcal{L}(X) \times \mathcal{L}(X) \to \mathcal{L}(X)$	concatenation
$nil@ys = ys$	Eq. $@_1$
$(x :: xr)@ys = x :: (xr@ys)$	Eq. $@_2$
$rev : \mathcal{L}(X) \to \mathcal{L}(X)$	reversal
$rev\ nil = nil$	Eq. rev_1
$rev\ (x :: xr) = (rev\ xr)@[x]$	Eq. rev_2
$foldl : (X \times Y \to Y) \times Y \times \mathcal{L}(X) \to Y$	folding left
$foldl(f, y, nil) = y$	Eq. $foldl_1$
$foldl(f, y, x :: xr) = foldl(f, f(x, y), xr)$	Eq. $foldl_2$

□

Proposition 1. *Concatenation is associative, i.e.,*

$$\forall xs \in \mathcal{L}(X)\ \forall ys \in \mathcal{L}(X)\ \forall zs \in \mathcal{L}(X) : (xs@ys)@zs = xs@(ys@zs).$$

Proof. Induction over the structure of the first list above. Induction hypothesis:

$$H(xs) = \forall ys \in \mathcal{L}(X)\ \forall zs \in \mathcal{L}(X) : (xs@ys)@zs = xs@(ys@zs).$$

Base case We prove $H(nil)$, i.e.,

$$\forall ys \in \mathcal{L}(X)\ \forall zs \in \mathcal{L}(X) : (nil@ys)@zs = nil@(ys@zs).$$

Let ys and zs be elements of $\mathcal{L}(X)$. We prove:

$$(nil@ys)@zs = nil@(ys@zs)$$

as follows:

$$
\begin{array}{llll}
 & (nil@ys)@zs & & \\
= & & nil@ys \to ys & \text{by Eq. } @_1 \\
 & (ys)@zs & & \\
= & & & \text{parenthesis} \\
 & (ys@zs) & & \\
= & & nil@(\dots) \leftarrow (\dots) & \text{by Eq. } @_1 \\
 & nil@(ys@zs). & &
\end{array}
$$

Induction step Let x be an element of X, and let xr be an element of $\mathcal{L}(X)$. Assume $H(xr)$, i.e.,

$$\forall ys \in \mathcal{L}(X)\ \forall zs \in \mathcal{L}(X) : (xr@ys)@zs = xr@(ys@zs).$$

We prove $H(x :: xr)$, i.e.,

$$\forall ys \in \mathcal{L}(X)\ \forall zs \in \mathcal{L}(X) : ((x :: xr)@ys)@zs = (x :: xr)@(ys@zs).$$

Let ys and zs be elements of $\mathcal{L}(X)$. We prove

$$((x :: xr)@ys)@zs = (x :: xr)@(ys@zs)$$

as follows:

$$
\begin{array}{llll}
 & ((x :: xr)@ys)@zs & & \\
= & & (x :: xr)@ys \to x :: (xr@ys) & \text{by Eq. } @_2 \\
 & (x :: (xr@ys))@zs & & \\
= & & (x :: (\dots))@zs \to x :: (\dots)@zs & \text{by Eq. } @_2 \\
 & x :: (xr@ys)@zs & & \\
= & & & \text{parenthesis, see Appendix A [1]} \\
 & x :: ((xr@ys)@zs) & & \\
= & & (xr@ys)@zs \to xr@(ys@zs) & \text{by instantiation of } H(xr) \\
 & x :: (xr@(ys@zs)). & &
\end{array}
$$

□

Proposition 2. *Let $f = \lambda(x, a) \in \mathcal{L}(X) \times \mathbb{N}.a + 1$. The length of a list xs can be computed as $foldl(f, 0, xs)$, i.e.,*

$$\forall xs \in \mathcal{L}(X) : |xs| = foldl(f, 0, xs).$$

Proof. (failed) Induction over the structure of the list. Induction hypothesis:

$$H(xs) = (|xs| = foldl(f, 0, xs)).$$

Base case We prove $H(nil)$, i.e.,

$$|nil| = foldl(f, 0, nil),$$

as follows:

$$
\begin{array}{llll}
 & |nil| & & \\
= & & |nil| \to 0 & \text{by Eq. } |\_|_1 \\
 & 0 & & \\
= & & foldl(\dots, 0, nil) \leftarrow 0 & \text{by Eq. } foldl_1 \\
 & foldl(f, 0, nil). & &
\end{array}
$$

Induction step Let x be an element of X, and let xr be an element of $\mathcal{L}(X)$. Assume $H(xr)$, i.e.,

$$|xr| = \mathit{foldl}(f, 0, xr).$$

We attempt to prove $H(x :: xr)$, i.e.,

$$|x :: xr| = \mathit{foldl}(f, 0, x :: xr)$$

as follows:

$$\begin{array}{lll} & |x :: xr| & \\ = & & |x :: xr| \to 1 + |xr| \quad \text{by Eq. } |\_|_2 \\ & 1 + |xr| & \\ = & & |xr| \to \mathit{foldl}(f, 0, xr) \quad \text{by } H(xr) \\ & 1 + \mathit{foldl}(f, 0, xr). & \end{array}$$

Our sequence of proof steps did not reach the goal, while no further useful steps can be applied. We attept to prove

$$1 + \mathit{foldl}(f, 0, xr) = \mathit{foldl}(f, 0, x :: xr)$$

by applying proof steps on $\mathit{foldl}(f, 0, x :: xr)$ as follows:

$$\begin{array}{lll} & \mathit{foldl}(f, 0, x :: xr) & \\ = & & \mathit{foldl}(f, 0, x :: \dots) \to \mathit{foldl}(f, f(x, 0), \dots) \quad \text{by Eq. } \mathit{foldl}_2 \\ & \mathit{foldl}(f, f(x, 0), xr) & \\ = & & f(x, 0) \to 1 \quad \text{by definition of } f \\ & \mathit{foldl}(f, 1, xr). & \end{array}$$

Our sequence of proof steps did not reach the goal, while no further useful steps can be applied. Our proof attempt is stuck at proving

$$1 + \mathit{foldl}(f, 0, xr) = \mathit{foldl}(f, 1, xr).$$

■

Proposition 3. *Let* $f = \lambda(x, a) \in \mathcal{L}(X) \times \mathbb{N}.a + 1$. *The length of a list* xs *and a natural number* n *are related to* $\mathit{foldl}(f, n, xs)$ *as follows.*

$$\forall xs \in \mathcal{L}(X)\ \forall n \in \mathbb{N} : |xs| + n = \mathit{foldl}(f, n, xs).$$

Proof. Induction over the structure of the list. Induction hypothesis:

$$H(xs) = (\forall n \in \mathbb{N} : |xs| + n = \mathit{foldl}(f, n, xs)).$$

Base case We prove $H(\mathit{nil})$, i.e.,

$$\forall n \in \mathbb{N} : |\mathit{nil}| + n = \mathit{foldl}(f, n, \mathit{nil}).$$

Let n be a natural number. We apply the following steps.

$$\begin{array}{lll} & |\mathit{nil}| + n & \\ = & & |\mathit{nil}| \to 0 \quad \text{by Eq. } |\_|_1 \\ & n & \\ = & & \mathit{foldl}(\dots, n, \mathit{nil}) \leftarrow n \quad \text{by Eq. } \mathit{foldl}_1 \\ & \mathit{foldl}(f, n, \mathit{nil}). & \end{array}$$

Induction step Let x be an element of X, and let xr be an element of $\mathcal{L}(X)$. Assume $H(xr)$, i.e.,

$$\forall n \in \mathbb{N} : |xr| + n = \mathit{foldl}(f, n, xr).$$

We prove $H(x :: xr)$, i.e.,

$$\forall n \in \mathbb{N} : |x :: xr| + n = \mathit{foldl}(f, n, x :: xr).$$

Let n be a natural number. We apply the following steps.

$$\begin{array}{llll} & |x :: xr| + n & & \\ = & & |x :: xr| \to 1 + |xr| & \text{by Eq. } |\_|_2 \\ & 1 + |xr| + n & & \\ = & & & \text{by associativity of } + \\ & |xr| + (n+1) & & \\ = & & |xr| + (n+1) \to \mathit{foldl}(f, n+1, xr) & \text{by } H(xr) \\ & \mathit{foldl}(f, n+1, xr) & & \\ = & & f(x,n) \leftarrow n+1 & \text{by definition of } f \\ & \mathit{foldl}(f, f(x,n), xr) & & \\ = & & \mathit{foldl}(f, n, x :: xr) \leftarrow \mathit{foldl}(f, f(x,n), xr) & \text{by Eq. } \mathit{foldl}_2 \\ & \mathit{foldl}(f, n, x :: xr). & & \end{array}$$

□

Proposition 4. *Let $f = \lambda(x, xs) \in X \times \mathcal{L}(X).x :: xs$. A list xs can be reversed as $\mathit{foldl}(f, \mathit{nil}, xs)$, i.e.,*

$$\forall xs \in \mathcal{L}(X) : \mathit{rev}\ xs = \mathit{foldl}(f, \mathit{nil}, xs).$$

Proof. (failed) Induction over the structure of the list. Induction hypothesis:

$$H(xs) = (\mathit{rev}\ xs = \mathit{foldl}(f, \mathit{nil}, xs)).$$

Base case We prove $H(\mathit{nil})$, i.e.,

$$\mathit{rev}\ \mathit{nil} = \mathit{foldl}(f, \mathit{nil}, \mathit{nil}),$$

as follows:

$$\begin{array}{llll} & \mathit{rev}\ \mathit{nil} & & \\ = & & \mathit{rev}\ \mathit{nil} \to \mathit{nil} & \text{by Eq. } \mathit{rev}_1 \\ & \mathit{nil} & & \\ = & & \mathit{foldl}(\ldots, \mathit{nil}, \mathit{nil}) \leftarrow \mathit{nil} & \text{by Eq. } \mathit{foldl}_1 \\ & \mathit{foldl}(f, \mathit{nil}, \mathit{nil}). & & \end{array}$$

Induction step Let x be an element of X, and let xr be an element of $\mathcal{L}(X)$. Assume $H(xr)$, i.e.,

$$\mathit{rev}\ xr = \mathit{foldl}(f, \mathit{nil}, xr).$$

We attempt to prove $H(x :: xr)$, i.e.,

$$\mathit{rev}\ x :: xr = \mathit{foldl}(f, \mathit{nil}, x :: xr)$$

as follows:

$$\begin{array}{llll} & \mathit{rev}\ x :: xr & & \\ = & & \mathit{rev}\ x :: xr \to (\mathit{rev}\ xr)@[x] & \text{by Eq. } \mathit{rev}_2 \\ & (\mathit{rev}\ xr)@[x] & & \\ = & & \mathit{rev}\ xr \to \mathit{foldl}(f, \mathit{nil}, xr) & \text{by } H(xr) \\ & \mathit{foldl}(f, \mathit{nil}, xr)@[x]. & & \end{array}$$

Our sequence of proof steps did not reach the goal, while no further useful steps can be applied. We attept to prove

$$foldl(f, nil, xr)@[x] = foldl(f, nil, x :: xr)$$

by applying proof steps on $foldl(f, nil, x :: xr)$ as follows:

$$\begin{array}{lll} & foldl(f, nil, x :: xr) & \\ = & foldl(f, nil, x :: \dots) \to foldl(f, f(x, nil), \dots) & \text{by Eq. } foldl_2 \\ & foldl(f, f(x, nil), xr) & \\ = & f(x, nil) \to [x] & \text{by definition of } f \\ & foldl(f, [x], xr). & \end{array}$$

Our sequence of proof steps did not reach the goal, while no further useful steps can be applied. Our proof attempt is stuck at proving

$$foldl(f, nil, xr)@[x] = foldl(f, [x], xr).$$

■

Proposition 5. *Let* $f = \lambda(x, xs) \in X \times \mathcal{L}(X).x :: xs$*. The reversal of a list* xs *and a list* ys *are related to* $foldl(f, ys, xs)$ *as follows.*

$$\forall xs \in \mathcal{L}(X)\ \forall ys \in \mathcal{L}(X) : (rev\ xs)@ys = foldl(f, ys, xs).$$

Proof. Induction over the structure of the list. Induction hypothesis:

$$H(xs) = (\forall ys \in \mathcal{L}(X) : (rev\ xs)@ys = foldl(f, ys, xs)).$$

Base case We prove $H(nil)$, i.e.,

$$\forall ys \in \mathcal{L}(X) : (rev\ nil)@ys = foldl(f, ys, nil).$$

Let ys be an element from $\mathcal{L}(X)$. We apply the following steps.

$$\begin{array}{lll} & (rev\ nil)@ys & \\ = & rev\ nil \to nil & \text{by Eq. } rev_1 \\ & nil@ys & \\ = & nil@ys \to ys & \text{by Eq. } @_1 \\ & ys & \\ = & foldl(\dots, ys, nil) \leftarrow ys & \text{by Eq. } foldl_1 \\ & foldl(f, ys, nil). & \end{array}$$

Induction step Let x be an element of X, and let xr be an element of $\mathcal{L}(X)$. Assume $H(xr)$, i.e.,

$$\forall ys \in \mathcal{L}(X) : (rev\ xr)@ys = foldl(f, ys, xr).$$

We prove $H(x :: xr)$, i.e.,

$$\forall ys \in \mathcal{L}(X) : (rev\ (x :: xr))@ys = foldl(f, ys, x :: xr).$$

Let ys be an element of $\mathcal{L}(X)$. We apply the following steps.

	$(rev\ (x :: xr))@ys$		
$=$		$rev\ (x :: xr) \rightarrow (rev\ xr)@[x]$	by Eq. rev_2
	$((rev\ xr)@[x])@ys$		
$=$			by associativity of @
	$(rev\ xr)@([x]@ys)$		
$=$		$(rev\ xr)@\ldots \rightarrow foldl(f, \ldots, xr)$	by $H(xr)$
	$foldl(f, [x]@ys, xr)$		
$=$		$x :: nil \leftarrow [x]$	by definition of :: ?
	$foldl(f, (x :: nil)@ys, xr)$		
$=$		$(x :: nil)@ys \rightarrow x :: (nil@ys)$	by Eq. $@_2$
	$foldl(f, x :: (nil@ys), xr)$		
$=$		$nil@ys \rightarrow ys$	by Eq. $@_1$
	$foldl(f, x :: ys, xr)$		
$=$		$f(x, ys) \leftarrow x :: ys$	by definition of f
	$foldl(f, f(x, ys), xr)$		
$=$		$foldl(f, ys, x :: xr) \leftarrow foldl(f, f(x, ys), xr)$	by Eq. $foldl_2$
	$foldl(f, ys, x :: xr)$.		

□

References

1. Gert Smolka. *Programmierung - eine Einführung in die Informatik mit Standard ML.* Oldenbourg Wissenschaftsverlag, 2008.