Reversing a list in Prolog

Move one item from top of one list to the top of the other list.

```
(1::2::3::nil)     nil.
(2::3::nil)   (1::nil).
(3::nil)   (2::1::nil).
nil     (3::2::1::nil).
```

Unfortunately, this is not writable directly in Prolog. Instead, the following code is close (a third argument is needed).

```
reverse L K :- rev L K nil.

rev nil K K.
rev (X::L) K M :- rev L K (X::M).
```

Reversing a list in λProlog

We can write this with more directly and with more structure in lambda Prolog as:

```
reverse L K :-
  pi rv\  rv nil K =>
          (pi M\pi N\pi X\ rv (X::N) M :- rv N (X::M)) =>
          rv L nil.
```

$$\forall K, L[$$

$$(\forall rv \quad (rv\ nil\ K \supset (\forall M, N, X(rvN(X :: M) \supset rv\ (X :: N)\ M)) \supset rv\ L\ nil))$$

$$\supset reverse\ L\ K]$$

Reverse is symmetric

Thus, an attempt to prove

$$reverse\ (1 :: 2 :: 3 :: nil)\ (3 :: 2 :: 1 :: nil)$$

results in the introduction of a new predicate rv and the attempt to prove that from the two clauses

$$rv\ nil\ (3 :: 2 :: 1 :: nil).$$

$$\forall M, N, X (rv\ N\ (X :: M) \supset rv\ (X :: N)\ M)$$

it follows that

$$rv\ (1 :: 2 :: 3 :: nil)\ nil.$$

And this is the computation that we expected to take place.

Theorem. Reverse is symmetric; that is, if $\vdash reverse\ L\ K$ then $\vdash reverse\ K\ L$.

What is the informal proof?

A more formal proof

Proof. Assume that $\vdash_I reverse\ L\ K$. Thus, the body of $reverse$'s clause must be provable.

$$\vdash_I \forall rv\ (rv\ nil\ K \supset (\forall M, N, X(rvN(X :: M) \supset rv\ (X :: N)\ M)) \supset rv\ L\ nil)$$

Now, instantiate this higher-order quantifier with the term $\lambda x \lambda y. \neg rv\ y\ x$, and we have that

$$\vdash_I \neg rv\ K\ nil \supset (\forall M, N, X(\neg rv\ (X :: M)\ N \supset \neg rv\ M\ (X :: N))) \supset \neg rv\ nil\ L$$

Using the classically valid contrapositive rule: $\neg p \supset \neg q \equiv q \supset p$, we have

$$\vdash_C rv\ nil\ L \supset (\forall M, N, X(rv\ M\ (X :: N) \supset rv\ (X :: M)\ N)) \supset rv\ K\ nil$$

Since classical and intuitionistic Horn clause provability coincide, we can replace $\vdash_C$ with $\vdash_i$. By universal generalization over rv, we have

$$\vdash_I \forall rv\ (rv\ nil\ L \supset (\forall M, N, X(rv\ M\ (X :: N) \supset rv\ (X :: M)\ N)) \supset rv\ K\ nil)$$

Since this is the body of the reverse problem, we can conclude that $\vdash_I reverse\ K\ L$.

Comments

1. Challenge to hackers: Try this with C!
2. Challenge to us: Try this on more than a toy!
3. Notice that we didn't use induction. Almost certainly, this technique will not work in general without a notion for induction. For example, consider proving that the reverse of a list perserves its length.
4. This proof is even better when done using linear logic insights. In linear logic, the body of *reverse* can be written as

   $$\vdash_I \forall rv\ (rv\ nil\ K \multimap (\forall M, N, X(rvN(X :: M) \multimap rv\ (X :: N)\ M)) \supset rv\ L\ nil)$$

   Also, note that $\neg p \multimap \neg q \equiv q \multimap p$ is valid in linear logic. When using the stronger linear logic setting, we would not need to move from intuitionistic to classical to intuitionistic logic.