

A Lollibot Primer

Lollibot is a bottom-up linear logic programming language that implements saturation for persistent facts and quiescence for ephemeral facts, the two termination strategies from lecture. Its syntax is similar to Twelf's, with a few important caveats:

- Rules are not named, and terms and predicates need not be declared
- Rules are written only in the left-to-right direction
- Instead of `->`, Lollibot uses the linear logic lollipop, rendered in ASCII as `-o`
- Rules may have multiple premises and multiple conclusions, using comma `,` to join facts and `()` to represent none at all
- Predicates are treated as ephemeral unless marked with a bang `!`

To run a set of rules, you use one of the following declarations:

```
%exec <steps> <initial facts>.
%trace <steps> <initial facts>.
```

where `<steps>` is the number of steps to run for and `<initial facts>` is the initial database of facts, separated by commas. To run until saturation or quiescence, use a `<steps>` value of `*`. The `%exec` declaration displays only the final database, while the `%trace` declaration additionally displays each step of execution.¹

As a simple example illustrating the new syntax, recall the linear logic program from lecture that decomposes a list and then reassembles it into a permutation of a sublist, possibly dropping some elements.

```
% list Xs --> perm Ys, where Ys is a permutation of a sublist of Xs.

list (cons X Xs) -o elem X, list Xs.
list nil -o perm nil.
elem X -o ().
elem X, perm Xs -o perm (cons X Xs).

% test case: permute [1, 2, 3], showing all steps
%trace * list (cons 1 (cons 2 (cons 3 nil))).
```

Lollibot also has built-in primitive (non-saturating) predicates for testing equality and disequality of terms. To test whether `X` and `Y` are equal, use the predicate `X == Y`. To test whether `X` and `Y` are unequal, use the predicate `X <> Y`. For instance, the following code generates all unequal pairs of items:

```
!item A, !item B, A <> B -o !pair A B.
%exec * !item(1), !item(2), !item(3).
```

Many more [Lollibot examples](#) are available from the course's [Software](#) page.

¹... but saturation is regarded as a single step.