Visualization Plan

Monday, November 25, 2019 5:20 PM

For the visualizer, we are going to be recursively moving backwards through the sequence to define the resulting mailing list.

What this means is that it will go about doing the following:

1) printing the resulting list
2) printing its representation
3) for each reference, we will go back to the sequence that it was referenced in last.
4) We then recurse if there are more references

In order to provide more spec, we can go through examples w/ each operator.

ex1: normal mailing list

in: a@mit.edu

out: a@mit.edu

this works for lists with no references

ex2: single reference

in: a = b@mit.edu

out: b@mit.edu
↓
a = b@mit.edu

it simply is branching down

ex3: multiple references in a row

in: a = b@mit, c = d@mit

out: b@mit, d@mit
↙ ↘
a = b@mit c = d@mit.

ex4: sequences

in: a = b@mit.edu ; a

out: b@mit.edu
↓
a = b@mit.edu

ex5: redefined

in: a = b@mit ; a = c@mit ; a

out: c@mit.edu
↓
a = c@mit.edu

ex6: recursive defs

in: a = a, b@mit ; a = a, c@mit ; a

out: b@mit, c@mit
↓
a = a ∪ b@mit
↓
a = a ∪ c@mit
↓
a = None

ex7: defined in a union

in: a = b@mit, c = j@mit, f = c@mit ; c

out: j@mit
↓
c = j@mit

Hopefully this clears up the idea of making the visualization of a tree branching out of the resulting mailing list to its definitions.