This is the LAST PROBLEM SET! It is not required for 6.905 students. Everybody: Please work on your projects.

Readings:

SDF Chapter 7: Propagation

Radul & Sussman, "The Art of the Propagator," http://dspace.mit.edu/handle/1721.1/44215. This is a preliminary paper about the ideas in the propagator system. It is NOT about the system we are using, which is entirely new. However, this paper accurately captures the philosophy of the system and explains a simple implementation.

Alexey Radul’s PhD thesis dissertation: "Propagation Networks: A Flexible and Expressive Substrate for Computation" http://web.mit.edu/~axch/www/phd-thesis.pdf This is more detail and more worked out than the "Art" paper above, but it is much longer. It is also not about the system we are using.

Technical note:

Load the propagator system in the usual way, with (load "load"). But you also need to tell the system about which primitive propagators and which merge procedure is to be used. This requires a call to install-core-propagators!. You will also need to execute (initialize-scheduler) for each experiment, to clear out the history of the previous experiment.
To Do

Exercise 7.1: An electrical design problem

For this problem the system initialization you need after loading is:

(define text-arith
  (extend-arithmetic
    value-set-extender
    (extend-arithmetic
      layered-extender
      (extend-arithmetic interval-extender
        numeric-arithmetic))))

(install-arithmetic! text-arith)

(install-core-propagators!
  merge-value-sets
  text-arith
  (value-set-propagator-projector
    layered-propagator-projector))

Exercise 7.2: Local consistency

For this problem you will be making an arithmetic that is different from any we have, so you need to make it, its merge mechanism, and install its core propagators.

Exercise 7.4: Yacht Name Puzzle

For this problem you need:

(install-arithmetic! tms-arith)
(install-core-propagators! merge-value-sets
  tms-arith
  tms-projector)

tms-arith, merge-value-sets, and tms-projector are defined and loaded for you in the load.scm. They are defined in example-support.scm.

Exercise 7.7: Type inference

For this problem you will be making an arithmetic that is different from any we have, so you need to make it, its merge mechanism, and install its core propagators.