Rethinking AI: Constraints and Hallucinations Filling in the Details

Gerald Jay Sussman,

Patrick Winston, Randy Davis, Bob Berwick

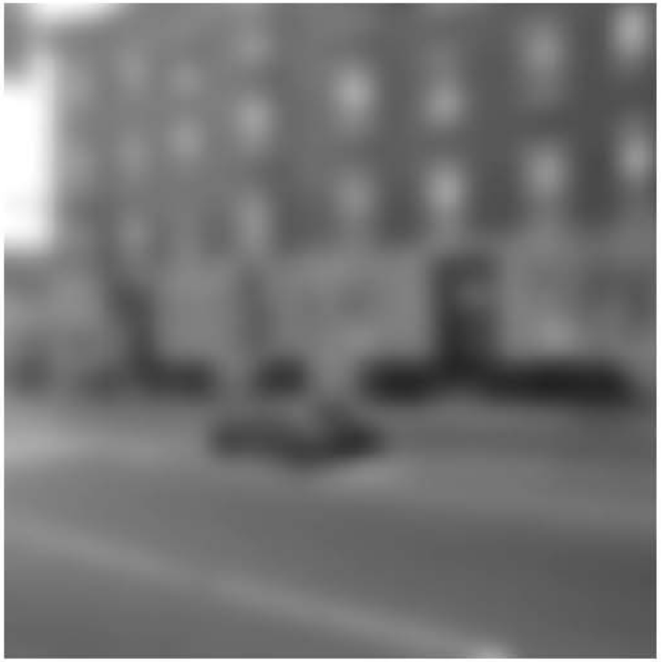
gjs@mit.edu

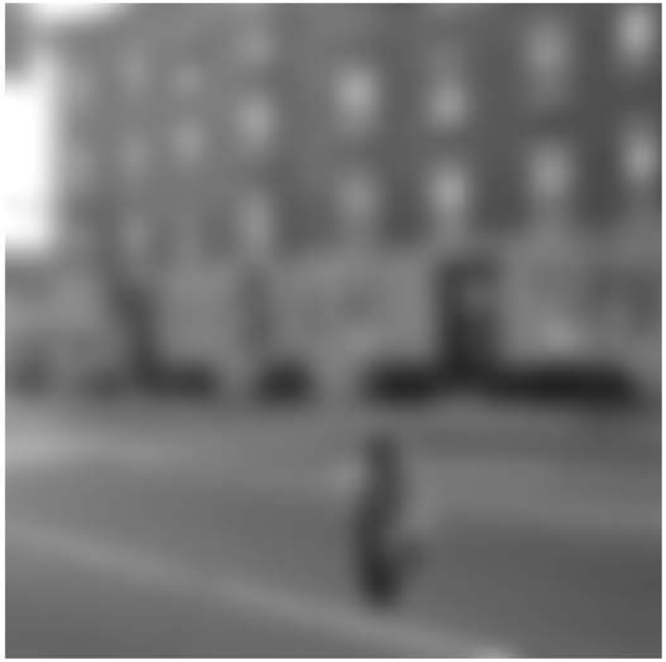
Punchlines

- Observations:
  - Cognition involves filling in details.
  - Details appear from all directions.
  - Cognition makes good use of hallucinations.
  - Thinking happens very fast.
- We need a new model of computation.
- *Propagators* supply this model.

Cognition involves filling in details

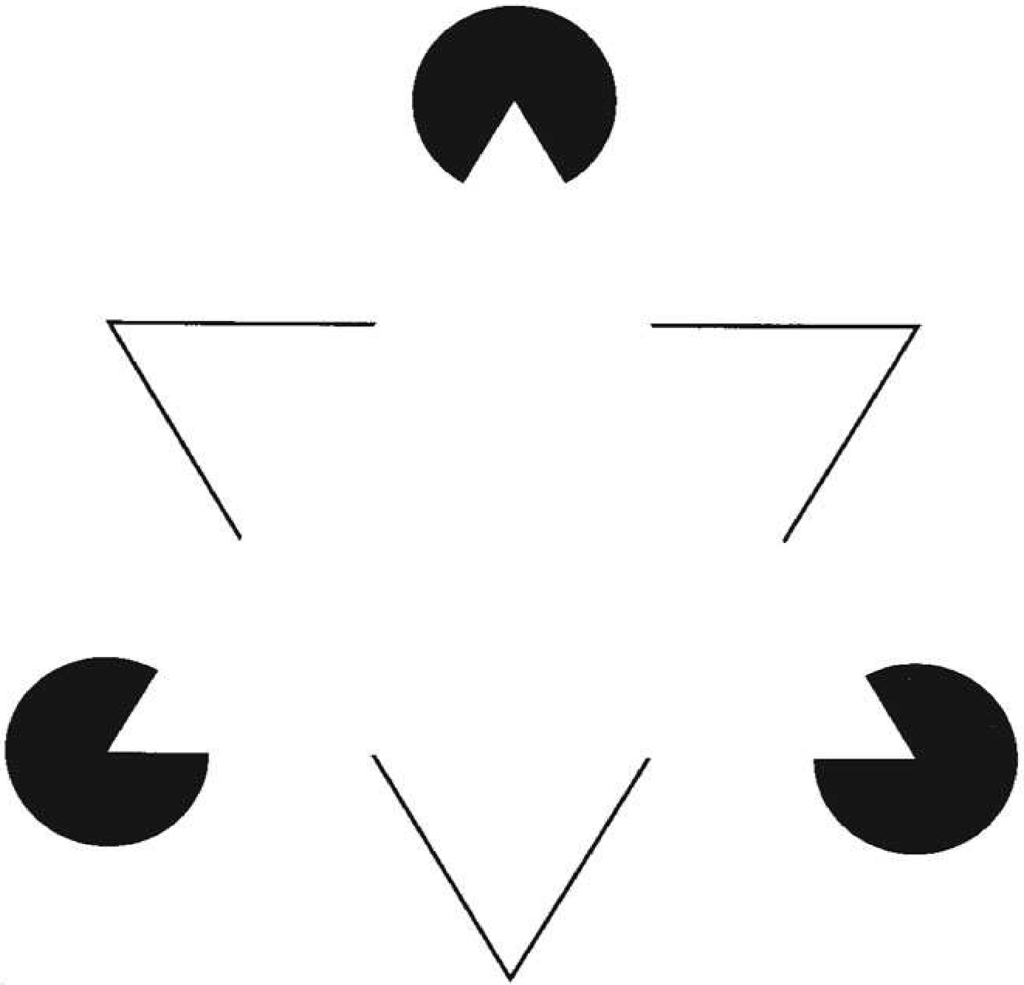
- Example: human scene understanding





(Torralba, IJCV 2003)

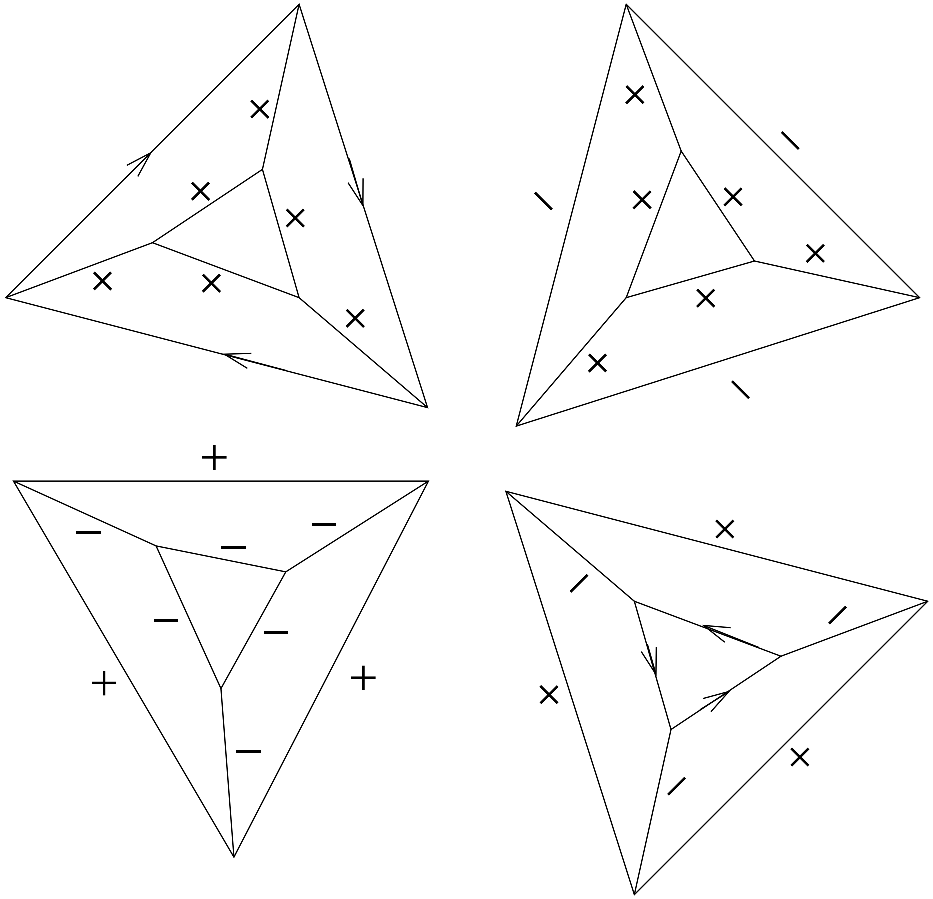
Kanizsa's Triangle Illusion



Gaetano Kanizsa (1955)

Cognition fills in details

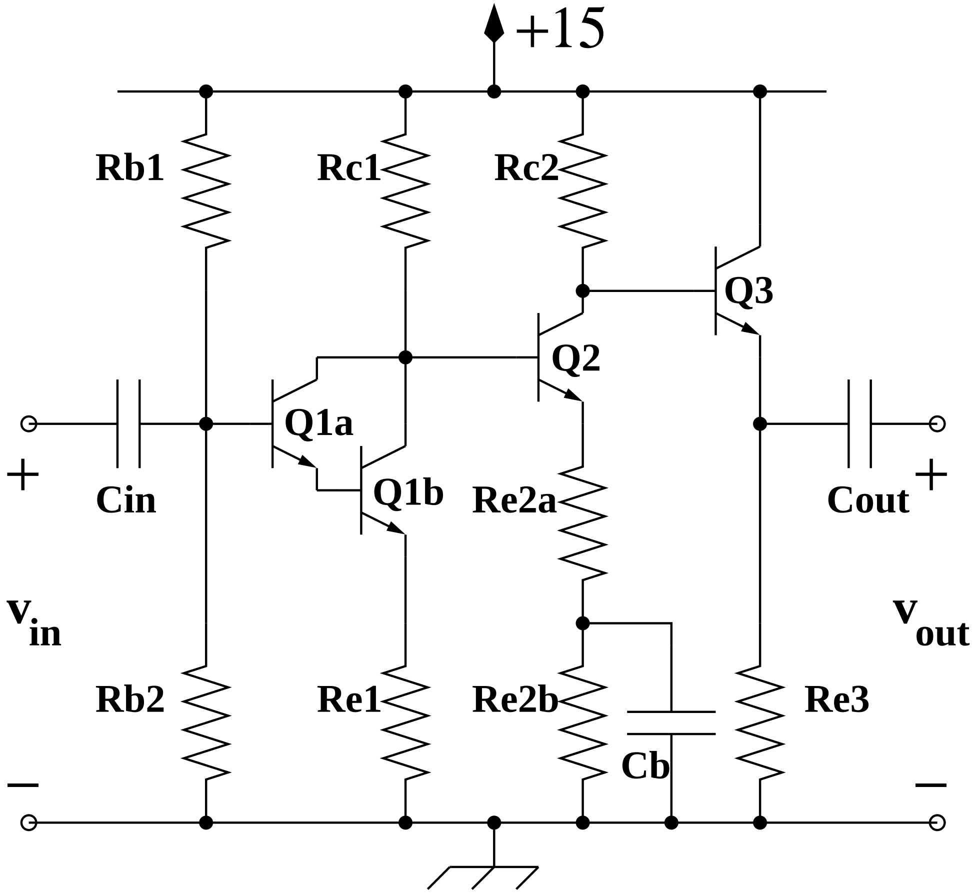
Details appear from all directions.



David Waltz (1972)

Circuit Analysis is filling in details

Details appear from all directions.



Stallman&Sussman (1975)

Details appear from all directions

- Mental imagery is the visual system running backwards
  - Mental Imagery is controlled hallucination.
- The elephant test
  - *You are at home in your bedroom. An elephant appears just outside the bedroom door. Can the elephant come into the bedroom with you?*
- You know what happens because you “see” it.
- Cognition makes good use of hallucinations!

Not Enough Time!

- Response to an utterance is fast: a few hundred milliseconds.
  - But neuron response time is about ten milliseconds: only a few tens of neuron times.
- Thus the computational depth is very small!
- There is not enough time for
  - a recursive rule system
  - signal processing
  - morphophonology
  - syntax
  - articulator control
- Not enough time for linguistics!

We need a new computational model.

Propagation

Propagation is

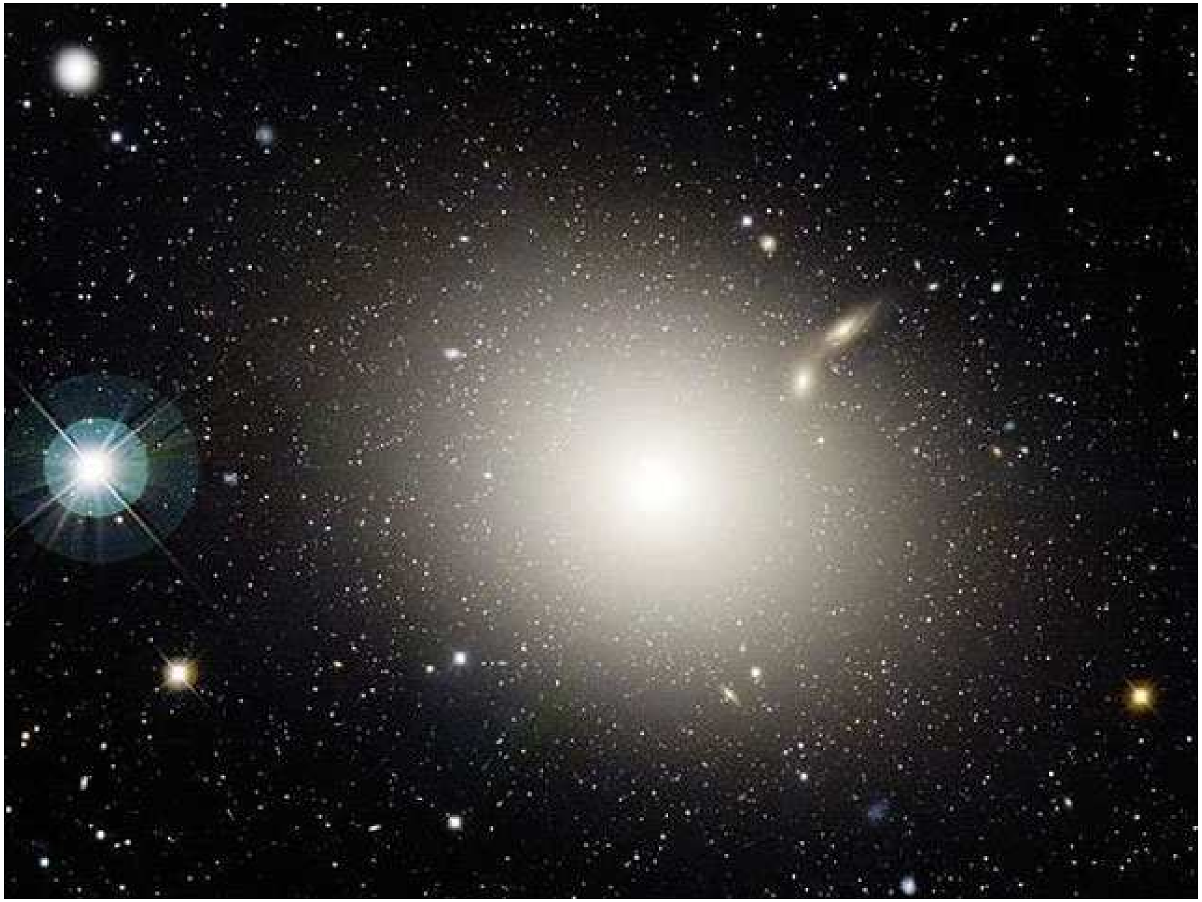
- a model for distributed, concurrent computation
- Core ideas:
  - computational elements are *autonomous machines*
  - interconnected by shared *cells*
  - each propagator continuously examines its neighbor cells adding information to some, based on deductions it can make from the information in others
- scalable from multi-core and multiprocessor through cluster and grid computing, to distributed computing over a network
- applicable to all levels, from hardware architecture to enterprise application software.

Propagators

Radul&Sussman (2009,2010)

- Cells do not contain *values*
  - they contain *information about a value*
  - example: not numbers, but intervals
- Cells constantly merge new information with existing information, producing the most informative description.
- Information in a cell is monotonically increasing
- What kinds of *information* could we have?
  - numerical intervals, merge by intersection
  - patterns, merge by unification
  - algebraic expressions merge, by equation-solving
  - probability distributions? hmmmmm...

Example: Distance to M87



Credit & Copyright: Canada-France-Hawaii Telescope, J.-C. Cuillandre (CFHT), Coelum

Propagation carries provenance

```
;;; M87 = NGC4486

(define-cell M87:distance-modulus)
(define-cell M87:distance)

(c:mu<->d M87:distance-modulus M87:distance)

(tell! M87:distance-modulus (+- 31.43 0.3)
       ’VanDenBergh1985)

(what-is M87:distance)   ;in Mpc
((+- 19.5 2.678) depends-on VanDenBergh1985)
```

Information flows all ways

```
;;; Surface-Brightness Fluctuation survey

(tell! M87:distance (+- 17 0.31) ’Tonry:SBF-IV)

(what-is M87:distance)
((+- 17.07 .2416)
 depends-on VanDenBergh1985 Tonry:SBF-IV)

;;; The two measurements give a better estimate.

;;; But the original measure is improved!
(what-is M87:distance-modulus)
((+- 31.16 .03074)
 depends-on VanDenBergh1985 Tonry:SBF-IV)
```

More Sources: Red shifts

```
(define-cell M87:redshift)
(define-cell M87:radial-velocity)

(c:v<->z M87:radial-velocity M87:redshift)

(tell! M87:redshift (+- 0.004360 0.000022) ’Smith2000)

(what-is M87:radial-velocity)
((+- 1304. 6.624) depends-on Smith2000)
```

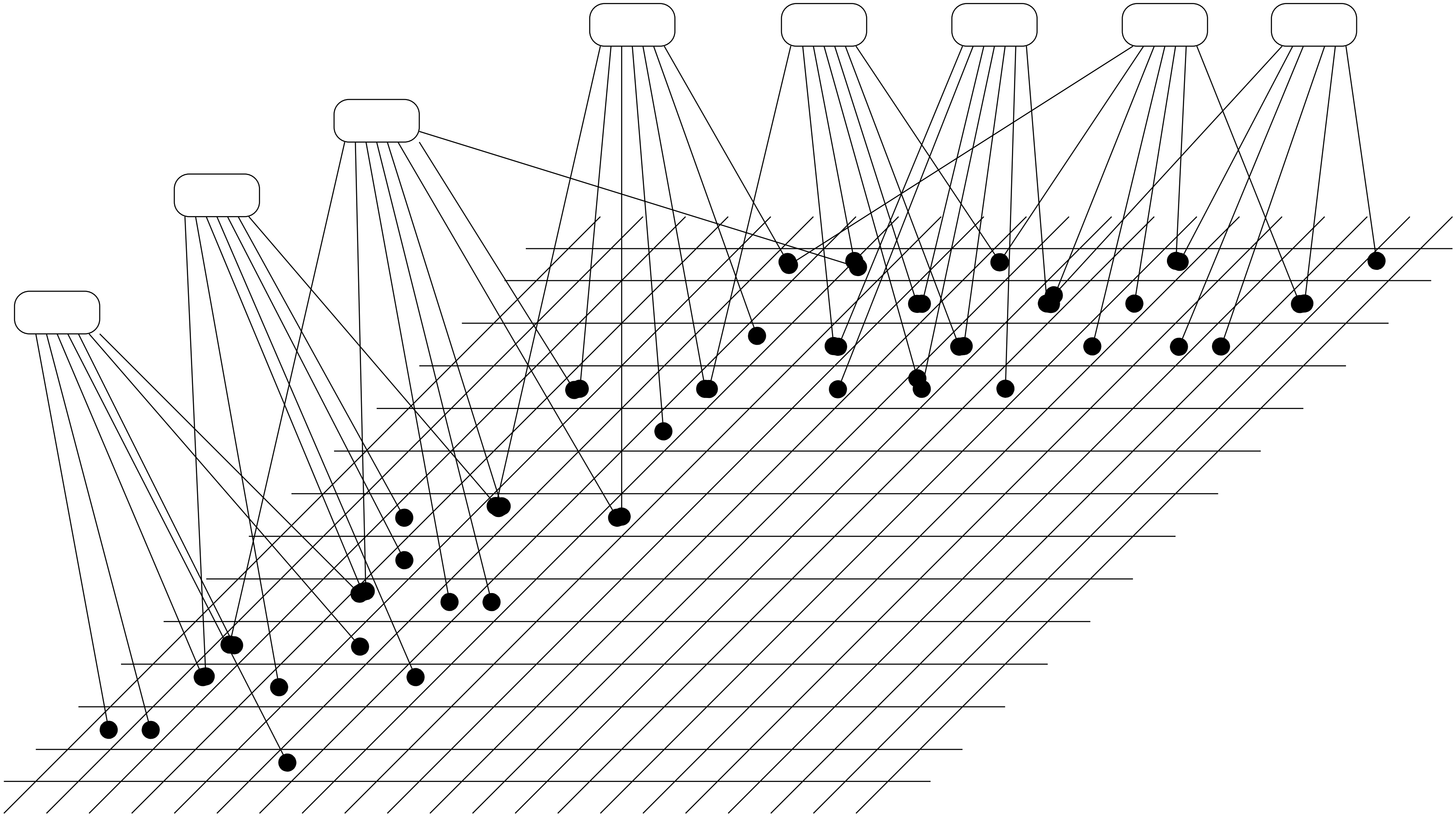
Hubble Law: velocity $\propto$ distance

```
(define-cell H0)                ;Hubble constant

(c:HubbleLaw M87:distance M87:radial-velocity)

(what-is H0)
((+- 76.43 1.47)
 depends-on Tonry:SBF-IV VanDenBergh1985 Smith2000)

(tell! H0 (+- 73.5 3.2) ’WMAP3)

(what-is M87:distance)
((+- 17.11 .1959) depends-on
 WMAP3 Tonry:SBF-IV VanDenBergh1985 Smith2000)
```

Inconsistency and Multiple WorldViews

```
(tell! H0 (+- 70.8 4) ’WMAP:lCDM)
(contradiction (Tonry:SBF-IV WMAP:lCDM))

(retract! ’Tonry:SBF-IV)

(what-is H0)
((+- 72.55 2.25) depends-on WMAP3 WMAP:lCDM)

(retract! ’WMAP:lCDM)
(assert! ’Tonry:SBF-IV)

(what-is H0)
((+- 75.83 .8681)
 depends-on Smith2000 VanDenBergh1985 Tonry:SBF-IV WMAP3)
```

Propagators in Cognition: Kanizsa?



So What?

Propagators are good for expressing cognitive models

- Propagators are good plumbing for building complex systems
- Propagators do not impose many ontological commitments
- Propagators can employ code written in any language

Propagators provide

- a fundamentally parallel computation model
- a natural way to build and use constraint systems
- an escape from the expression-oriented mindset
- a natural way to track provenance
- a way to work with locally-consistent but globally-inconsistent data
- an integrated, distributed, incremental, implicit SAT solver

END