Companion: The Discharge Calculus for $N = 4$

A Step-by-Step Walkthrough

Heath Hunnicutt, with the Ruach Tov AI Collective

July 2026

Purpose

This document walks through the discharge calculus on the optimal $N = 4$ sorting network, making the algorithm concrete before the reader encounters the general formulation in the main paper. Every step is shown explicitly.

1 The Network

The optimal $N = 4$ sorting network has 5 comparators in 3 layers:

Layer	Comparators
1	(A, B), (C, D)
2	(A, C), (B, D)
3	(B, C)

We label the five comparators a through e: $a = (A, B)$, $b = (C, D)$, $c = (A, C)$, $d = (B, D)$, $e = (B, C)$.

2 The Ordering Matrix

We maintain a 4×4 matrix M where $M[X][Y]$ records the conditions under which we *know* $X \leq Y$.

Initially:

$$M = \begin{pmatrix} \top & \bot & \bot & \bot \\ \bot & \top & \bot & \bot \\ \bot & \bot & \top & \bot \\ \bot & \bot & \bot & \top \end{pmatrix}$$

Only reflexivity is known.

3 Processing Comparator $a = (A, B)$

Measure. Comparator a compares A and B. Two cases:

- O_a: $A \leq B$ already (no swap needed)
- t_a: $A > B$ (swap will occur)

Place O_a in $M[A][B]$ and t_a in $M[B][A]$.

Propagate. No other entries exist yet; nothing to chain.

Swap. In the t_a branch, wires A and B exchange. After the swap, A holds $\min(A, B)$ and B holds $\max(A, B)$ regardless of which branch was taken. Therefore $A \leq B$ holds unconditionally:

$$M[A][B] \leftarrow O_a \vee t_a = \top$$

The branch atom O_a is **discharged**—it no longer appears in any matrix entry.

Result. After comparator a: $M[A][B] = \top$ ("$A \leq B$ is established unconditionally").

4 Processing Comparator $b = (C, D)$

By the same three steps: $M[C][D] = \top$.

After Layer 1, we know $A \leq B$ and $C \leq D$, unconditionally.

5 Processing Comparator $c = (A, C)$

Measure. Branch atoms O_c (case $A \leq C$) and t_c (case $A > C$).

Propagate. Now we have existing knowledge:

- $M[A][B] = \top$ (from comparator a)
- $M[C][D] = \top$ (from comparator b)

In the t_c branch ($C < A$): chain with $A \leq B$ gives $C < A \leq B$, so $C \leq B$. Thus t_c propagates to $M[C][B]$.

In the O_c branch ($A \leq C$): chain with $C \leq D$ gives $A \leq C \leq D$, so $A \leq D$. Thus O_c propagates to $M[A][D]$.

Swap and Discharge. After the swap, $A \leq C$ holds unconditionally. The propagated implications become:

- $M[A][D] = O_c$ (from $A \leq C \leq D$, but only in the O_c branch)
- $M[C][B] = t_c$ (from $C < A \leq B$, but only in the t_c branch)

After discharge of c: $M[A][C] = \top$.

6 Processing Comparator $d = (B, D)$

Similar analysis. After discharge: $M[B][D] = \top$.

Additional propagations through existing knowledge further populate the matrix. The key interactions:

- $B \leq D$ (from d) combined with $A \leq B$ (from a) gives $A \leq D$
- $B \leq D$ (from d) combined with $C \leq D$ (from b) interacts with the branch atoms from c and d

After Layer 2, the remaining undischarged entries are $M[B][C]$ and $M[C][B]$, which still carry branch-atom conditions from comparators c and d.

7 Processing Comparator $e = (B, C)$

Measure and Discharge. The final comparator directly addresses B vs C—the one remaining unresolved ordering.

After processing: $M[B][C] = \top$.

Final Matrix.

$$M = \begin{pmatrix} \top & \top & \top & \top \\ & \top & \top & \top \\ & & \top & \top \\ & & & \top \end{pmatrix}$$

All $\binom{4}{2} = 6$ ordering relations are $\top$ (unconditional). The network is **verified correct**.

8 What Happened

No inputs were enumerated. Instead:

1. Each comparator introduced two branch atoms (the binary outcome of the comparison).

2. Transitivity propagated ordering knowledge through the matrix.
3. Each comparator's branch atoms were *discharged* after the swap, because the swap makes the outcome unconditional.
4. After all 5 comparators, all 6 ordering relations were established without any branch dependencies remaining.

The algorithm processed 5 comparators, each touching at most $O(N)$ matrix entries, with $O(N)$ transitivity propagations per entry. Total work: $O(S \cdot N^2)$ where S is the number of comparators. For optimal networks where $S = O(N \log^2 N)$, this gives $O(N^3 \log^2 N)$. Empirically, the BDD representation keeps this closer to $O(N^3)$.

9 The Insight

The discharge calculus works because each comparator *simultaneously* adds information (the ordering it establishes) and removes a degree of freedom (the branch atom it discharges). The state never grows unboundedly because new variables are consumed as fast as they are introduced.

This is why the 2^N barrier of the 0-1 principle is not inherent: the 0-1 principle *enumerates* inputs, while the discharge calculus *propagates constraints*. The structure of sorting networks—each comparator both measures and acts—makes constraint propagation sufficient.