

adventures in cubing

A journey down the rabbit hole of Rubik's-style puzzles.

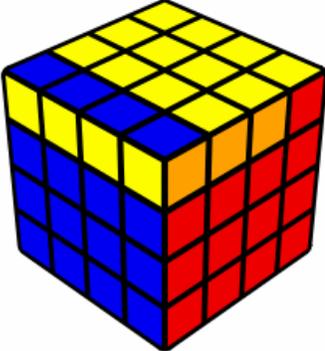
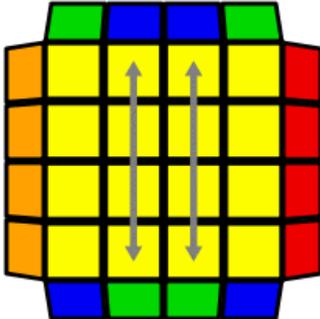
PARITY OR: HOW I LEARNED TO STOP WORRYING AND LOVE THE 4x4

After painstakingly modding my Shengshou v4s a couple months ago (mod post coming soon), my interest in 4x4s has waxed and waned. On the one hand, I find it a welcome and more challenging distraction from 3x3s; on the other hand, those damn parity algorithms!?! I have finally committed them to muscle memory and am now trotting along at a 2:20 average using the [Yau variation](#) of the [reduction method](#). (For more on Yau, and especially the cross-on-right variation that I find easier, check out Cyoubx' really good [Yau intro video](#).)

For months, parity was my albatross. I finally conquered it, so to speak, not only by finding and learning algorithms that worked for me, but also my learning about the root causes of parity. This was by far my deepest dive into puzzle theory and its associated patois. Keep reading past the jump for much more on the types of cube parity, what causes each, and, most importantly, what we really mean when we sloppily say that a cube has "parity."

PARITY ALGORITHMS

If you're just here for the algorithms, look no further:

OLL Parity	PLL Parity
 <p data-bbox="235 1430 682 1482">[1] $Rw\ U2\ x\ Rw\ U2\ Rw\ U2\ Rw'\ U2\ Lw\ U2$ $Rw'\ U2\ Rw\ U2\ Rw'\ U2\ Rw'$</p> <p data-bbox="102 1528 815 1591">[2] $(Rw\ Lw)\ U2\ Lw'\ U2\ Rw'\ U2'\ x'\ U2\ Rw'\ U2'\ Rw\ U2\ Rw'\ U2'\ Rw2$ $U2$</p>	 <p data-bbox="1040 1430 1364 1455">$Uw2\ Rw2\ U2\ r2\ U2\ Rw2\ Uw2$</p>

Here's a (now muted due to copyright claims) video showing each:

4x4 Rubix Cube Parity - Algorithms, Explanation, Theory



(*music: Bob Dylan, "Subterranean Homesick Blues"; cubes: Konsta+Florian modded Shengshou v4s with Cube Specialists Bright+ modded stickers*)

WHAT IS PARITY?

There are technical and colloquial uses of parity. Colloquially, "parity" is used to describe certain positions in higher-order cubes (when reduced to a 3x3 equivalent with center blocks and dedged/tredged/...edge blocks) that cannot occur on a 3x3. Cubers often describe those states as "parity" or a "parity errors."

The source is math/puzzle theory, which uses "parity" simply to mean "the fact of being even or odd." So understood, all states in a puzzle have parity, just like a light switch always has a position. The question is whether that parity state (for the particular aspect being measured) is odd or even.

Since hyper-correctness often feels clumsy and confusing ("the sort of pedantry up with which I shall not put") much of this post uses "parity" informally to mean the states in a reduced 4x4 not naturally attainable in a 3x3.

AN ILLUMINATING EXAMPLE

The light switch example is actually really helpful: Let's treat off (down) as the initial state — analogous to "solved" on a cube. If you flip the switch up once to turn on the light, you've made 1 (odd) state change; if you flip the switch back down, you've made a total of 2 (even) state changes and restored the light to its initial state of off. If you start with the switch off (solved), flip it 5 times quickly (scramble), and then flip it 5 more times (solve), you would have made 10 total switchings, resulting in an even parity state. The light would be off, its initial state. But had the switch been flipped only four times during the "solve," the switch would have an odd parity state after the 5 subsequent switchings. The light would be on (opposite its initial state). In colloquial terms, this would be called "parity" or a "parity error." In technical terms, this would mean that the light system has an odd parity state.

Now let's extrapolate a bit further. Instead of a single switch for a single light, assume two "three-way" switches controlling one light. Three-way switches are the ones often found at opposite ends of a hallway or a staircase, such that either switch can be used to turn a light on or off. Assume that in their initial state both switches are down and the light is off. Flipping switch 1 up (leaving switch 2 down) turns on the light. Switch 1 has odd parity, switch 2 has even parity, and the system has odd parity (odd + even = odd). Now flip switch 2. The light turns off, back to its initial state. That is because both switches have odd parity and the system has even parity (odd + odd = even). *But here's the rub: Both switches are now up, not down.* The light may be in its initial state, but the system is not; the return to the initial state is illusory.

That, in an abstract nutshell, is cube "parity." Such parity arises when the reduction of a 4x4 appears to have created a 3x3 analogue (e.g., the light returning to the off position when switches 1 and 2 each are switched once), but the system being in a state other than the initial (both switches are now up).

PROBABILITIES

There are two noticeable* parities that can occur on a higher-order cube — OLL and PLL parity. Each has a 50% probability of occurring, since each is the result

of a parity state being either even or odd. And, since the causes of each parity state are independent, so are their probabilities of occurring. Any given solve can feature (1) OLL parity but not PLL parity, (2) PLL parity but not OLL parity, (3) no parity, or (4) both parities. There is an equal 25% chance of any of these four combinations. Half of all solves will have one parity but not the other, 25% will have neither, and 25% will have both.

** Excluding the parity state of the combined centers. On a standard cube with standard stickers, such parity states are not apparent; on a [super cube](#), the center parity state is apparent.*

OLL PARITY

OLL parity becomes evident when trying to orient the cubies of the last layer so that each has the top layer color facing up. On a 3x3, zero, two, or four edges will face up after completing the first two layers; there cannot be an odd number (one or three) facing up. On a 4x4, one or three combined edges (dedges) can face up — with either case, colloquially, OLL parity.

OLL parity is measured by the number (QTM) of inner-layer turns that occur when forming the reduced centers. If the number of quarter inner-layer turns (of the scramble + the solution to solved centers) is odd, there will be OLL parity. Note that the parity state is locked in once the centers are formed — essentially, at the moment you execute the last inner layer slice to form the last centers.

Subsequent inner-layer turns related to pairing the edge pieces into dedges will not affect the parity state vis-à-vis OLL parity. If odd before edge pairing, it will stay odd; if even before edge pairing it will stay even. After center formation, the only way to alter the OLL parity state is to perform an OLL parity algorithm. Edge pairing does not affect the OLL parity state because each pairing is just a set of two (even) inner slice moves: inner slice > outer slice > inner slice. An even number of inner-slice turns will not affect the parity state of the inner slices.

PLL PARITY

PLL parity becomes evident when the bottom 3 layers (bottom layer plus combined middle layers) and top face are solved. What is left are the corners and dedges — all facing “out.” When those corners and dedges are not in a configuration solvable using any of the 21 3x3 PLL permutations, the cube is said to have parity.

PLL parity is the result of there being a difference (one being odd and the other being even) between (a) the corner parity state and (b) the dedge parity state. Corner and dedge parity is measured by the outer-slice quarter turns required to place all corners and all dedges in their correct positions relative to the combined centers. When it would take an odd number of moves to place the corners but an even number of moves to place the dedges, there is PLL parity. When the opposite is true (even for the corners and odd for the dedges), there is PLL parity. When both are even or both are odd, there is no PLL parity; the dedges and corners are in a configuration attainable on a 3x3 and therefore solvable with 3x3 PLL perms.

The PLL parity state is locked in during the last stage of pairing (reducing) the 24 edges into 12 dedges — at the moment of pairing the last two edge groups. How those would-be pairs are solved (which ends up on the left and which on the right), will put the dedges into either an odd or even parity state. [Chris Hardwick's example](#) is helpful:

You have one group at FL and one group at FR. At uFL you have yellow/red and at uFR you have the other yellow/red. At dFL you have yellow/orange and at dFR you have the other yellow/orange. Based on how you pair these last two edge groups you can either end up with a yellow/orange edge group at FL and a yellow/red edge group at FR; or you could end up with a yellow/red edge group at FL and a yellow/orange edge group at FR. This effectively swaps two edge groups, which changes the parity of the edge permutation. One of the above cases for pairing edges will create even edge group permutation parity, the other odd permutation parity



PLL parity occurs when that pairing creates a state (odd or even) opposite the pre-existing corner parity.

Once there is a state difference between the parities of the corners and dedges, no combination of F2L or 3x3 PLL perms will resolve the difference. Only a PLL parity algorithm will “disrupt” the difference by switching the state of either the corner or dedge parity. (Standard 3x3 permutations affect both corner and edge parities simultaneously — either leaving both as they were before or flipping both.) To better understand this think about a quarter turn on the outer slice. That turn changes the parity state of each of all four edges and each of all four corners on that layer. So, if the initial parity state is corners-odd/edges-even, a quarter turn changes each corner to even and each edge to odd. The corner and dedge parities each get flipped, but the difference as between the two

remains — now corners-even/dedges-odd.

THE FICTION OF A REDUCED 4x4

In the end, parity errors reveal semantic differences between a 3x3 and a 4x4 reduced to a 3x3. A reduced 4x4 may look, smell, feel, and function (in almost all respects) like a 3x3 equivalent. But how you get from 24 center cubies to 6 combined centers (OLL parity), and from 24 edges to 12 combined dedges (PLL parity), determines whether the resulting 3x3 analogue is in a state that could be achieved on a 3x3. Analogous, though they may be, identical they are not.

When there is PLL parity, for example, you need to cut through the assumption that a dedge is just an oversized edge, and must temporarily go back to seeing each dedge as a combination of two single edge pieces. By breaking the reduction, splitting dedge pairs back into their component parts, and then re-reducing them into dedges of the opposite parity, the 4x4 enters a state that a 3x3 could have — then being solvable with 3x3 algorithms. **So understood, parity algorithms reveal the fiction that a reduced 4x4 is the same as a 3x3. The algorithms “break” up particular reductions and recast them in a way that more closely resembles states on a 3x3.**

SuperAntonioVivaldi's [very good parity video](#) addresses this exact point in the 17-20-minute range. It is by far the most accessible demonstration of parity I've seen, and the entire video is required viewing.

And perhaps the best written description of the fiction that a reduced 4x4 is *exactly* equivalent to a 3x3 is qqwref's [post on the speedsolving.com forum](#):

Reduction parity occurs when you try to reduce the puzzle so it can be solved by a constrained set of moves, putting it into some subset of the positions. However, you can often reach a position which seems like it is in your subset, but which is actually not, and to solve the puzzle you have to briefly go outside your constrained set of moves to bring the puzzle back into the subset you want.

In 25% of 4x4 solves, the constrained set of moves (3x3 algorithms) can bring the cube into a solved state. But in 75%, the reduction “seems like it is in [the 3x3] subset,” but is not. A parity algorithm is required “to briefly go outside [a 3x3's] constrained set of moves.”

EPILOGUE, CREDITS

Two weeks ago, I knew almost nothing about parity. I could blindly apply the two algorithms, but did not understand the source of the so-called “errors.” This post is just a [beginner's distillation of very complex concepts](#) that are described in more detail — and contextualized more deeply within math and puzzle theory — in a variety of other sources. The main source for my understanding was a [conversation I had with Chris Hardwick](#) (puzzle theory expert) on the speed-solving forum. Also helpful were a variety of other posts in that forum ([here](#), [here](#), and [here](#)) and some good articles/wiki entries on math and puzzle parity.

Many thanks to Chris for his clear explanations and patience.

Advertisements



Best Seller ★★★★★
9 Insanely Cool Gift Ideas
 Free Shipping By Christmas Guarantee

Worst Foods for Prostate

Prostate Revive Expert Doctor Reveals Natural Prostate Treatments Other MDs Won't Tell You.

This entry was posted in [favorites](#), [math](#), [not 3x3x3](#), [tutorials](#) and tagged [4x4](#), [Bob Dylan](#), [Chris Hardwick](#), [Cyoubx](#), [not 3x3x3](#), [parity](#), [puzzle theory](#), [SuperAntonioVivaldi](#), [tutorial](#) on [August 25, 2013](#) [<https://adventuresincubing.com/2013/08/25/4x4-parity/>] by [Adventures in Cubing](#).

7 thoughts on “Parity or: How I Learned to Stop Worrying and Love the 4×4”



Patrick M

August 25, 2013 at 9:16 pm

Very impressive! I remember learning parity ages ago. Such a beast of an alg.
You modded multiples with both florian konsta? Id so pay for one, i need motivation to 4×4 haha



Adventures in Cubing

Post author

August 25, 2013 at 9:22 pm

That OLL one is a total beast!

And, yeah, I modded them both one weekend. What a pain!?! The black one is great; I'm still breaking in the white one. I would pay decent money to buy already modded SS 4x4s, too!?!

Pingback: [Void Cube – Walk-Through, Re-Stickering](#) | adventures in cubing

Pingback: [So, What's this Blog All About?](#) | adventures in cubing



Tim Koop

January 8, 2016 at 5:51 am

I don't think I'll ever be able to memorize the 4×4 dedge parity algorithm. It's easier to just turn the cube in half (rR) then resolve the middles and edges. Too bad this messes up most of the cube. If you could test for edge parity in an earlier stage of solving, like when the 4×4 first resembles a 3×3 (just after solving the middles and edges), you could do the (rR) move without messing up so much of the cube. I have discovered such a test. Perhaps I'll make a YouTube video of it, explaining how to do it some day. Thanks for the page!



Adventures in Cubing

Post author

January 8, 2016 at 6:01 am

You CAN memorize it. I thought the same thing at first.

Like all algs, it's all about committing it to muscle memory so that you don't have to “think” the alg.



Tim Koop

January 11, 2016 at 7:11 am

Yeah, I suppose. But until that happens, I'll use this method I developed:

<http://timkoop.com/4x4parity>

☺