

Siteswap FAQ

[Allen Knutson](#) / version 2.0 (November 10, 1993)

The most Frequently-Asked siteswap Question on rec.juggling:

"What are all those numbers??!"

Contents:

- 1. [What could you write down about a juggling pattern?](#)
 - The two ways out of the notational morass
- 2. [Space-time diagrams](#)
- 3. [Vanilla siteswap](#)
 - The general idea and the vanilla restriction
 - The "0" and "2" kludges
- 4. [Some examples](#)
- 5. [Working backwards](#)
 - The averaging rule (and misconceptions about)
 - Orbits
- 6. [Some more examples](#)
- 7. [A deeper level](#): landing schedules, or "juggling states"
 - Ground state vs. excited state
 - Finding transitions
- 8. [More general notations](#) (rocky road siteswap)
 - Synchronous siteswap
 - Multiplexing
- 9. [What's this good for?](#)
- 10. [A little history](#)
 - The origin of the ugly name!

1. What could you write down about a juggling pattern?

To REALLY describe a juggling pattern, ideally we would like to describe the position of every muscle in the juggler's body at every microsecond, including the gag line being uttered. That being too much to write down, we have to adopt a mix of two interconnected ways to simplify matters:

- only describe some patterns, not all
- gloss over some aspects of a pattern.

The first route is an unfortunate one, and luckily we'll only have to do it in vanilla siteswap. The second is not so bad; if we feel that some important aspect of a pattern has been neglected by the description, we'll just include a little dressing up, on the side.

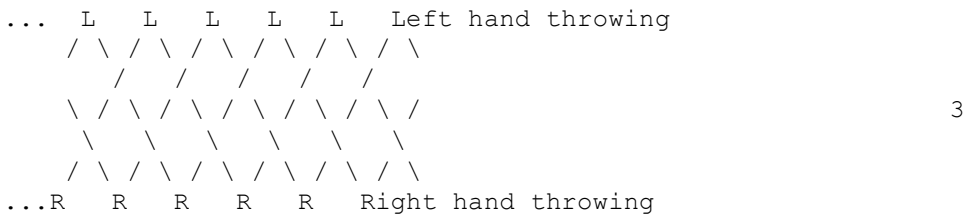
The idea behind siteswap is to keep track of the order that balls are thrown and caught, and only that. Since no matter how you juggle, you're catching and throwing in some order, there will always be some siteswap

describing it. On the other hand, if you're most interested in where the balls are thrown from and to (as in Mills' Mess), then this won't be a very satisfying partial description.

The above was about taking a juggling pattern and writing a description. The other end of things involves taking a description and trying to juggle "that way". Since we already decided it was impossible to keep track of everything about a juggling pattern, there probably won't only one way to put this description into practice. In siteswap, as it turns out, there is a most standard way. Unfortunately, people have taken to only calling patterns juggled in that standard way siteswaps and saying "These are siteswaps - other things aren't, because they aren't being juggled in the standard way." Much better to say "The siteswap for that isn't such a useful description, because this throw is kinda funky..."

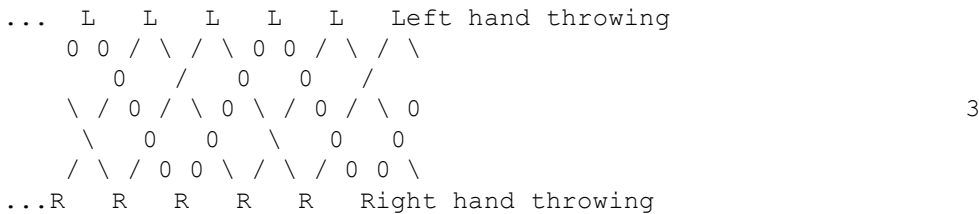
2.Space-time diagrams

Let's take the standard 3-ball cascade. There are many things to notice about this pattern, like every throw being the same height, but for our purposes, we'll only note that the hands alternate, every throw goes from one hand to the other, and most importantly, while each ball is in its arc exactly two other throws occur (the two other balls get thrown).

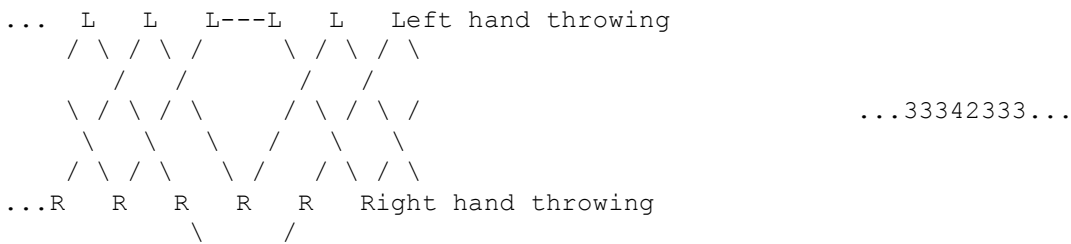


(On a first reading, ignore the numbers on the right side, like the 3 above.)

The picture here is what you'd get, watching from above somebody juggling three glowing balls while walking forward, on time-lapse film. Alternately you can think of the horizontal axis as time, and the vertical axis as space (the space between the hands), so this is a "space-time diagram" of the 3-ball cascade. Since ASCII isn't the greatest medium for drawing these, I'll draw it again with one of the balls singled out.

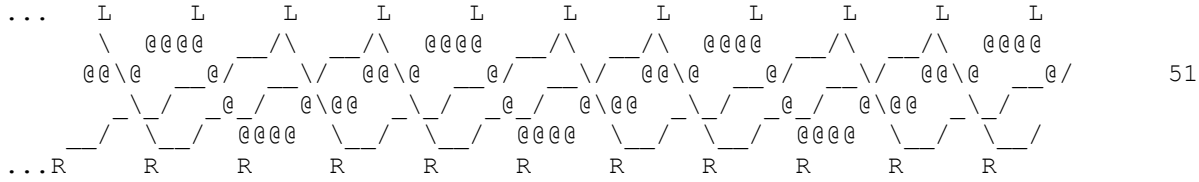


A rather simple trick you could do in the middle of this is throw a single 2-in-one-hand throw from the right hand.



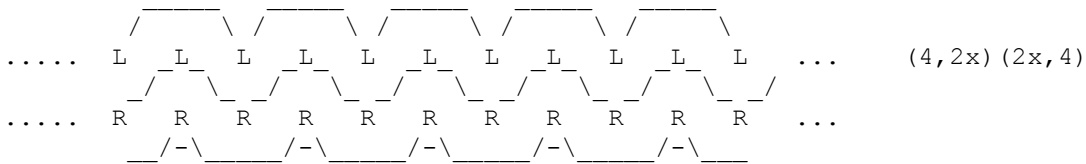
Notice that the throw from the right hand isn't the only new thing in this diagram - the left hand holds once. This looks more complicated than it has to be; why not just write L once up there? But this makes things more complicated in another way, though, as will be explained the "Vanilla siteswap" section.

Here are some more diagrams, for your amusement. The three-ball shower done asynchronously (with one ball singled out, hopefully improving clarity):

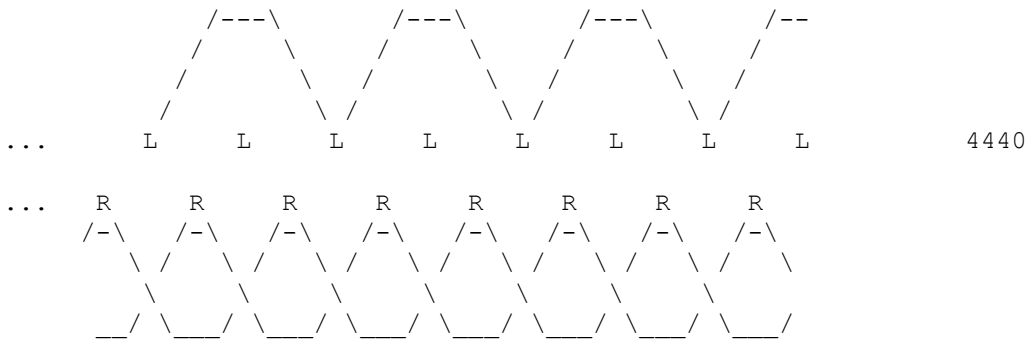


So every ball goes up from the right hand for a long time (long enough for the other two balls two visit each hand underneath) to the right hand, then is shuttled instantly back to the right.

Here's the shower-box, which has synchronous throws; correspondingly, the space-time diagram has the R throws directly below the L throws. Note that you can follow the ball in the middle and see that it never goes up on the sides. (Admittedly we didn't need these diagrams to notice that!)

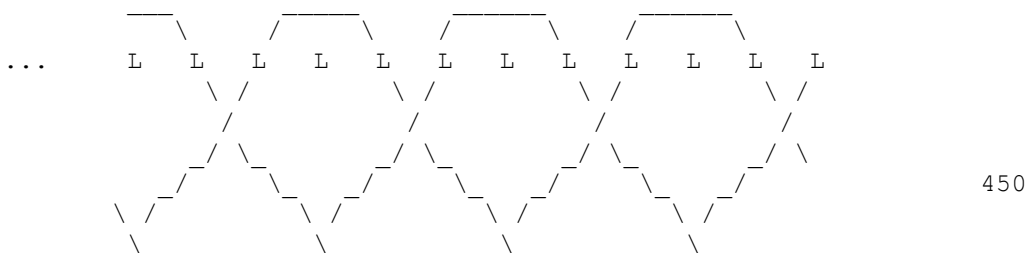


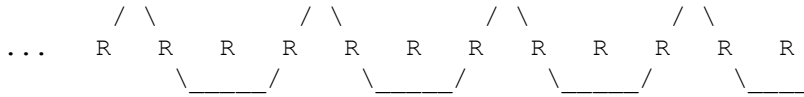
Here's 3 balls out of a 4-fountain.



(Once again, I've written in the hand even when it's not doing anything. This too shall be justified in the next section.)

Here's a goofier one, much less well-known (in fact, I've only seen one guy do it that I didn't teach it to!). It has been christened "A complete waste of a 5-ball juggler".





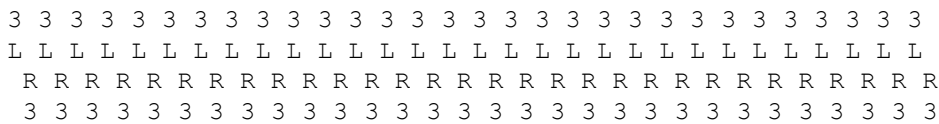
What do these diagrams have in common? The main thing is that at every vertex (here labeled L or R), there are the same number of lines coming in as going out. (Usually that number is 1; in the last two it's sometimes zero.) Drawing multiplex patterns we'd have several balls coming in and the same number going out. On the other hand, given a picture like the above, it's easy to see how to juggle it, at least in theory - throw the balls so that they land in the right order. (Of course, you can draw something that requires you to throw balls the height of the Empire State Building, but that's covered in a different section of the rec.juggling FAQ. : -8)

3. Vanilla siteswap

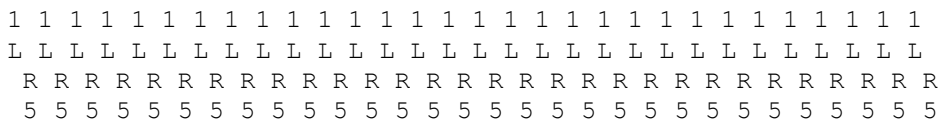
Drawing pictures is nice (except in ASCII), but plainly the pictures above contain lots of redundant information. How can we write them more compactly?

The Siteswap Idea: for each throw, write down the number of throws that occur before that ball is thrown again. In other words, call a throw an N if while that ball is in the air, N-1 other throws occur.

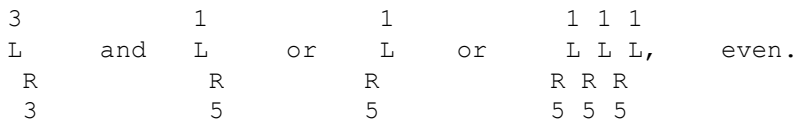
For instance, the 3-ball cascade becomes



and the 3-ball shower becomes



Of course, you don't have to write all that, since these patterns repeat - just write one period, like



We can simplify this a bit further with the Vanilla Siteswap restrictions:

The hands always alternate - in particular, only one hand at a time.

Only one throw is done at a time: no multiplexing.

This rules out lots of patterns, including the shower-box drawn in the last section, but it does include the cascade and fountain. With this assumption in place, it's time to actually write down some patterns like you'll see them on rec.juggling. The patterns written above (the 3-cascade and 3-shower) get written simply as "3" and "51".

Since we assume you're going left, right, left, right, you can always tell whether a throw goes across or back

to the same hand - it's whether the number corresponding to it is odd or even. Similarly, if a pattern has an odd number of numbers in it, when it repeats it'll be starting at the other hand, and so will be symmetrical.

In the standard cascade or fountain with N balls, while each ball is in the air one throws each of the other balls exactly once, so N-1 other throws are occur - so the throws are all N's.

For instance, let's take "4413", and try and work backwards to juggling. Remember, this is a shorthand for ..4413441344134413... Let's follow the first ball thrown, the initial 4. It goes up (to the same hand, 4 being even) and comes down 4 throws later, which is again a 4, in fact the next copy of the same 4. Then it goes up as a 4 again, and again, forever. Similarly for the second 4. Now follow the 1, which crosses quickly enough that it's thrown the very next throw - it's a handoff. That next throw is the 3, which also crosses, landing on the next copy of the 1. So that ball is handed off, then tossed over, then handed off, then tossed over, and so on. And it's the only ball that crosses. So the pattern looks like an anemic shower-box - one of the hands isn't fast enough to do the handoff, so both hands rest as that hand makes a little toss every time.

It might seem like the vanilla restrictions are really fierce - they rule out most of the patterns presented in the last section, like 3-out-of-a-4-fountain, which has two right-hand throws in a row. We will fix this by two clever kludges. The first is to say that when a hand holds a ball while the other hand does two throws in a row, it is in fact doing a tiny, tiny little throw. Such a throw is thus even, since it's going back to the same hand; it's a 2. Look back at the pattern labeled ...33342333... above, and you'll see where the funny 2-throw occurred.

We also have to assign a name to when a hand is empty while the other hand does two in a row, and this name will be 0. (Look back on the patterns labeled 4440 and 450 above.) I can mainly say that this turns out to be a nice idea mathematically; however, in its defense I will also point out that since N's are the throws one uses in the standard pattern with N (cascade or fountain), 0's should be the "throws" when one's hands are empty!

4. Some examples

20 Holding one ball in one hand, none in the other. (You wouldn't have complained had it turned out our notation didn't encompass this case!)

31 The 2-ball shower.

42 Two in one hand, holding one in the other.

423 The "half-shower-box"? One ball hops back and forth (the 3), while another one on each side just hops up and down.

441 A great pattern, still not known as well as it deserves to be. It is sometimes done as a club trick, with the handoff (the 1) done behind the back.

4413 Discussed above.

4440 3 out of a 4-fountain.

51 The 3-shower.

501 My proudest invention. Two balls, each going 1,5,1,5 as if they were in a 3-shower, except in opposite directions. Start it with both in the same hand. Very useful for bringing people up to 5-speed (but not accuracy). (I lay claim to the invention of this one only because the people I knew searching for patterns by computer forbade 0s.)

52512 The baby-juggling pattern. Two balls doing 5s, the baby going 1,2,2,1,2,2,1,2,2 forever.

53 The 4-ball half-shower.

531 A somewhat tricky three ball pattern, filling in the hole in 501.

5313 An asymmetric version of the above.

55500 The "flash", 3 out of 5.

50505 The "snake", 3 out of 5.
 5551 A sneaky 4-ball pattern, where one hand doesn't know it isn't doing 5.
 55514 A much prettier pattern, where three of the balls are always doing 5's, and the other one is sneaking underneath going 1,4,1,4 forever.
 450 A very goofy 3-ball pattern; "A complete waste of a 5-ball juggler."
 453 A very pretty 4-ball pattern, particularly with the 4's outside.
 55550 The active way to do 4 out of 5.
 552 The lazy way to do 4 out of 5.

5. Working backwards

By this point, I hope you're excited by the possibility that you could come up with new patterns just by correctly choosing numbers - even ones that would require a much more competent numbers juggler! The question is then:

When is a list of numbers actually a genuine siteswap?

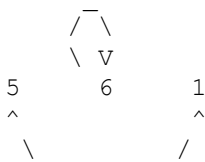
There are two things to check: at each time, we should be catching at most one ball (none if we're doing a 0 then). Also we should actually catch one, to have one available to throw (unless we're doing a 0 then.) (It is simpler to think of a 0-throw as one where there is actually a ball borrowed from the Astral Plane only long enough to throw and catch at the exact same time - then at each time, we're always throwing, and catching, exactly one ball. This is the main reason why it is mathematically nice to call these throws 0s.)

It is possible to prove mathematically that these two conditions are equivalent - so you only need to check one of them, and it doesn't matter which. (It seems that everybody checks that no two throws land at the same time, though!) It is also possible to prove that in this case, the average of the numbers must be the number of balls involved, which is convenient.

MISCONCEPTION: word has gotten out that the average of the numbers is the number of balls, and many people believe this to be the property to be checked. (If the average isn't an integer, than obviously the list is not a real juggling pattern.) But this isn't good enough - for example, if you tried to juggle the "pattern" 543, you'd be throwing three balls that would land at the same time.

When analyzing 4413 above, I talked about following one ball, and figuring out its entire destiny. You can do this with any siteswap (and it's the first thing I do when trying to learn them, though other people learn them differently). This procedure breaks up the numbers into bunches called "orbits", where each ball stays within a particular orbit. So you can remove all the balls in an orbit by setting all those numbers to zero. (For example, from 4413 we can make the patterns 0413, 4013, 4400, and 4000.) So the averaging theorem implies the more precise fact that the number of balls in a given orbit is the sum of the numbers in that orbit, divided by the length of the whole pattern.

For example, here are the orbits of the pattern 561:



So in a sense, this pattern is a combination of the patterns 501 and 060. The total height in the 51 orbit is 6:

since the pattern is length 3, this indicates that $6/3=2$ balls are trapped going 1,5,1,5,1,5 forever. Similarly, the other two balls just go 6,6,6,6 forever, hopping up and down in the same hand. And since the delay is 3 between one 6 and the next, they are in opposite hands. So in this pattern, there are two balls hopping up and down exactly out of phase from one another, surrounding a 501 in the middle.

In 72312, the orbits are (7,3) and (2,1,2). There are two balls in the (7,3) loop (going in opposite directions), and the baby in the (2,1,2) loop.

In 5241, the orbits are (5,2,1), with two balls, and (4), with just one.

In 5551, there is just one orbit (5,5,5,1). This pattern mixes up the balls completely.

In 55514, each orbit holds just one ball. The orbits are (5), (5), (5), and (1,4).

When I try to learn these, I fixate on a ball, or a couple of balls, that are doing something I understand, like one ball going 5,5,5,5 or two balls going 501.

6. Some more examples

Actually, here are just some of the highlights, now that you know how to generate your own.

633	61616	642	6424	66661	56414	645	661515	71
771	777171	723	7272712	723	75751	7562		7161616
70701	7531	741	714	7571	67561	17170170		801

7. A deeper level

Imagine someone shows you a videotape of a juggler doing the cascade, freeze-frames it, and asks "What might this juggler do after this point?" You can answer (assuming this someone is also hip to the lingo) "Since they've just been doing ...333, they can't now do a 0, 1, or 2, since if they were to do so two balls would land at the same time. But any other throw is fair game." Your friend tells you that you've gotten too theoretical - in fact this juggler is only good enough to do 5-throws. "Okay, 3, 4, or 5."

They show you a couple more frames. It turns out the juggler in fact does a 4. You now say "Okay, ...334, they still can't do a 0 or 1, and now they can't do a 3, but they can now do a 2 - which would take them back to the cascade. Or a 4 or 5, which would go to something qualitatively new."

The idea being used is this: you know when the balls are scheduled to land. In the first case, there's one landing now, next throw, the throw after, and none after that. In the second, there's one landing now, next throw, NOT the throw after that, but again the throw after that. These are the "landing schedules" or "juggling states" that you are watching this juggler go through, and I'll denote them by ...xxx|xxx-----..... andxxx|xx-x-----....., where the | means "now", the x's mean a ball landing, and the -'s mean no ball landing.

To see how one works with these, look at the following example of a transition from the 4-fountain to the 4-shower. We start at the state "xxxx" and throw the ball that is landing right now as a 5. Then move the indicator of the present forward one.

... x x x|x x x x - - - - - ...

```

... x x x x|x x x - x - - - - ... Okay, now throw a 6.
      /      \
... x x x x x|x x - x - x - - - ... Ready for the shower; throw a 7.
      /      \
... x x x x x x|x - x - x - x - ... Still in the shower, throw a 1.
      /      \
... x x x x x x x|x x - x - x - ...

```

Notice that we've come back to the state two above, so we can do 71 forever.

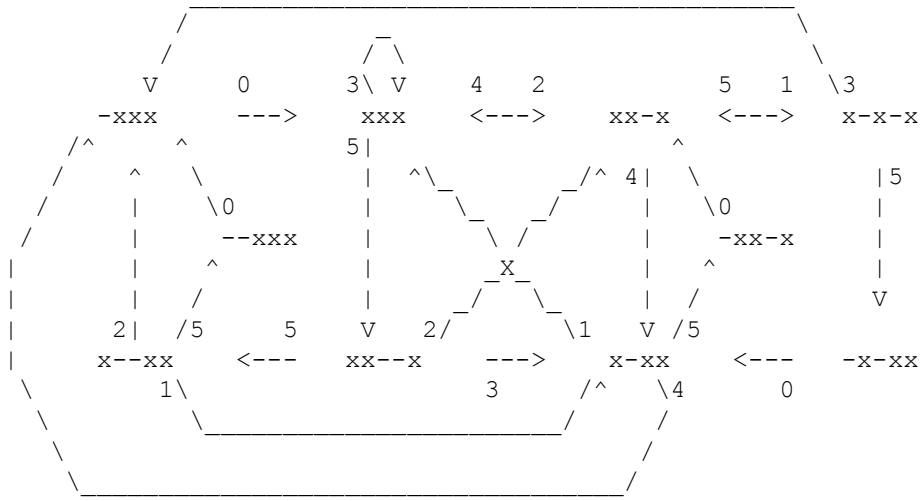
The writing here is a little redundant, of course - we don't need to know that in the past (before the "|") one ball landed every time. Once we don't write the past, we don't need the present indicated. And we only need to write the future out to the last ball. So we typically just write them like so:

```

      5          6          7 >
xxxx  ----->  xxx-x  ----->  xx-x-x  -----  x-x-x-x
                        <-----<

```

For example, here's the complete picture for juggling 3 balls with at most 5's.



So, a closed loop on this "directed graph" is exactly a siteswap.

One common division drawn between siteswaps is whether they are "ground state" or "excited state", which is whether one can begin juggling them directly from the cascade/fountain. For example, the trick 441 is ground-state -33333441441... is a valid history whereas 51 isn't.333351515151... has the last 3 and the first 1 land simultaneously. For the same reason, 414 is not ground-state, so it's not just a property of the pattern, but where one tries to jump into it.

The state diagram makes this division clear - patterns are ground state if they start at the state "xxxx" (or however many x's). Another thing that this clarifies is finding startup and finishing transitions for excited tricks like 771. What state does one start at to do 771?

```

771771771771771...
 1 177177177...

```


The numbers underneath indicate where you know a ball to be landing just because the juggler is doing 771. For instance, each 1 thrown in the first row will land 1 throw later, giving a 1 in the second row. There are five places where no ball is known to be landing - and these are the five balls one has to set up to go into this pattern. So the initial state is xxx-xx.

There is still art involved in deciding how you'd like to get from xxxxx to xxx-xx and back. For a more down-home example, let's say we want to settle back down from a 4-shower (71, xx-x-x) to a 4-fountain (4, xxxx). The first thing you might think of is to plug the holes:

xx-x-x --2--> xxx-x --3--> xxxxx

This is a rather painful way to get out of a shower, though. For one thing, it has a boring-looking 2. For the other, it's hard to handle a dinky 3-throw underneath a couple of 7's coming down (being hard to see both). So let's look for something else.

xx-x-x --4--> x-xxx --1--> xxxxx

This is better, but the 4 is still annoyingly low compared to the 7. Nothing to do for it but go to 6's.

xx-x-x --6--> x-x-xx --1--> xx-xx

We could go for the boring 2 at this point, or the more exciting

xx-xx --5--> x-xxx --1--> xxxxx

So we've found the pleasant transition ...717171 6151 44444.... Check it out!

8. More general notations

So far we've been doing things with the vanilla restriction that only one hand is throwing at a time - and in fact that the hands are alternating, saying the other hand is "throwing" a 0 or 2 if it seems to be doing nothing. How does the notation complicate if we remove this?

In two ways: before, we indicated the one throw that was going on at any time. Now, we have to indicate the throws that each hand is doing. (What if some hands aren't doing any throws? We've already figured out the trick for that; declare that they're doing the "hold throw" or the "empty hand throw", which in vanilla siteswap was 2 or 0, respectively.)

Take the example of the shower-box, whose space-time diagram was drawn back in section 2. The notation for this pattern is

(2x, 4) (4, 2x)

which means first that the right hand does a 4, while the left hand does a 2 that crosses. Then the left hand does a 4, while the right hand does a 2 that crosses.

This no longer really quite fits in with "The Siteswap Idea" that a throw be labeled by one plus the number of throws while it's in the air - shouldn't we just be calling those things 1's and 2's instead of 2's and 4's? Basically, mathematically, the answer is yes. But this doubling of all the numbers in "synchronous siteswap" is by now pretty standard, because it gives the proper indication of the relative heights of the throws. For

example, most people asynchronously fountain $2N$ balls at the same height as they synchronously fountain $2N$, so it would be a little weird calling the throws $2N$'s in vanilla siteswap and N 's in synchronous siteswap. Also, the averaging theorem is still true only if you double the numbers.

Here are some more examples of synchronous patterns to think about:

$(4x, 2x)$	the synchronous 3-ball shower
$(6x, 2x)$	the synchronous 4-ball shower
$(6, 6) (6, 2)$	The easy synchronous 3-in-one-hand, 2-in-the-other
$(6, 4)$	The much harder one
$(8, 2x) (4, 2x) (2x, 8) (2x, 4)$	Dan Bennett's amazing 4-ball box
$(6x, 2x) (2x, 6x)$	A much easier kind of 4-ball box
$(4x, 2x) (4, 2x) (2x, 4x) (2x, 4)$	the "double box"
$(6x, 2) (6, 2x) (2, 6x) (2x, 6)$	a really nice pattern - check it out!
$(6, 2x) (6, 2x) (2x, 6) (2x, 6)$	kind of a synchronous 61616 but with only two 6's on each side.

This notation extends in an evident way to more hands - put in more commas, and say which other hand you might be throwing to, instead of just "x". This gets called "Multi Hand Notation" - but there's nothing frightening in it beyond the step from 1 to 2 hands.

Basically all of the theory of vanilla siteswaps applies to these too, mutatis mutandis - you can follow a single ball to see what it does, divide the sum in that orbit by the total number of numbers to see how many balls are in the orbit, and so on. There are state diagrams for these too - they have two rows of x's and -'s instead of one. (And of course each arrow is labeled by a pair of throws instead of a single throw.)

Thus is defeated the first vanilla restriction. (Which of course doesn't mean that vanilla siteswap is now defunct - it is more convenient for describing those patterns that it can describe.)

The other restriction, No Multiplexing, is simpler to get around. Instead of saying what throw a hand is doing, we write down all of the throws it's doing, and put them in []'s to indicate that they happen at the same time. (Notice that this modification is completely independent of the ones we needed to institute for synchronous siteswap, and the examples below will be both synchronous and asynchronous multiplex patterns.)

For example, how do you juggle the asynchronous multiplex pattern ...444423[34]4444... in the middle of fountaining four? At some point hold with (say) the right hand, which is the 2, and so catch a second ball in the right hand. Then do a 3 from the left. Then do a multiplex from the right, throwing one across as a 3, the other up as a 4. Go back into 4.

N.B. The fact that siteswap doesn't tell you whether a 2 is a hold or a really dinky throw comes home to roost with these patterns. For example, the pattern [23] can either be interpreted as "juggle 3 while holding an extra one in each hand" or "do the easiest multiplex with 5", respectively.

Here are some asynchronous multiplex patterns:

24[45]	rather difficult, as the 4 tends to collide with the pattern.
25[65]2	a bit easier
25[75]51	a very fun variant of 75751
...717171 21[75]1 717171...	the multiplex in the middle of showering 4

And for my last trick, a synchronous multiplex!

([62], 2) (6x, [22]) (2, [62]) ([22], 6x)

This one's so full of 2's that it looks really elementary, like "Why on earth have I never seen that simple pattern before?" I saw it in Leeds, but didn't get the name of the juggler.

Again, all the theory from the vanilla case extends to multiplex patterns too. The "averaging" theorem is a little different: you add up all the numbers, and divide by the number of throw times, not the number of balls thrown. For instance, in the 5-ball multiplex [23], we divide 2+3 by 1, not by 2. There are juggling state diagrams for these too - instead of just x's indicating whether or not a ball is landing, you need to put a number to say how many balls are landing. A pattern is valid if the number of balls landing at a given time is the same as the number of balls being thrown. And so on.

The substantial difference is that it becomes impossible to track a single ball - when you multiplex, it's up to you which ball does which throw. No doubt this freedom could be removed by adopting some convention regarding the order that the multiplex throws are listed - [23] being a slightly different pattern than [32] - but nobody has bothered to make the arbitrary choices to set up this system. (Not that you were worried about it anyway!)

The system as outlined above can be generalized in many ways, and the state diagram seems to be the easiest way to see how to go about it. For example, you might have one ball that's a different color than the others. Okay, call that ball y instead of x in the states, so that as you cascade three you go

xxxy -3-> xyxx -3-> yxxx -3-> xxxy and so on.

Or you may have some reason to call a right-handed pattern different from a left-handed pattern. Make two copies of each state, and label them R and L at the beginning, where every throw takes you from an R state to an L state and vice versa. Go wild!

9. What's this good for?

Basically four things: for communicating juggling patterns, for finding new ones, for learning things in pieces so as to not have to learn them all at once, and for seeing how patterns are related to each other. I've already said bunches about the first two. The latter two are intertwined, in that to learn a difficult pattern (like the 7-cascade) it can help to be able to do related, but easier, patterns.

For an example, take the pattern 51, the 3-shower. The large number 5 suggests that this pattern will not be easy - in particular, substantially more difficult than the 3-cascade. The more hopeful idea is that if you can do this pattern, you'll be able to do 5. This certainly isn't true - in particular, if you can only do this pattern right-handed, then it only says that your right hand can do 5. So to learn 5 you should learn to asynchronously shower 3 both directions.

(Caveats: there are many unfortunate people who have bad habits with regards to the 1, and so they find the left-hand shower impossible, including people who can actually do 5! Also, people trying this will often degenerate into the synchronous shower (4x,2x), in which case they aren't actually working up to 5, they're working up to (4x,4x). This latter problem can be cured by working on 501 instead, which cannot be made synchronous.)

Also, it's nice having your hands know many patterns with a given throw, so that if you drop, you can automatically fall from one into another. For instance, when I drop in 5 now, I always fall into 5551. This

comes in handy when passing 10, in that I can pick up back into 10 while never disturbing my partner. (Well, in theory!)

10.A little history

This notation (or the vanilla notation, at least) was invented around 1985 by three people independently: Bruce "Boppo" Tiemann at Caltech, Paul Klimek in Santa Cruz, and Mike Day in Cambridge. (Hence its British name "Cambridge Notation".) Jack Boyce (also at Caltech) came up with the juggling state model to explain the phenomenon of excited-state tricks. (So did I, but a couple of weeks later, damn it! :-8)

Bruce has two articles published on the notation: "The Physics Teacher" (November 1989) and "Juggler's World" (Summer 1991).

A few mathematicians (including rec.juggling's own Dr. Colin Wright) will soon too or may already have.

Now, about the name. You can take one of these patterns - say, 55555 - and pick two throws - say, the second and fifth. Now say "Make these balls land in the opposite order!" (You can only do this if the first ball lands after the second one is thrown, or else you'd have to throw backwards in time.) In this case, the mutated pattern, where the Sites that the balls were landing have been Swapped, is 585525. Here are some more examples, to give you the idea:

55 --> 64 ^^	555 --> 753 ^ ^	5555 --> 8552 ^ ^	55555 -----> 95551 ^ ^
534 --> 552 ^^	534 --> 444 ^^	534 --> 633 ^ ^	

Note that this doesn't change the average. An easy way to prove the Averaging Theorem is to show that if the numbers aren't all the same, then there is some way to swap sites and eliminate one of the largest numbers. Stir until done. (All right, nobody has ever asked this on rec.juggling. So sue me.)

This siteswap FAQ brought to you by [Allen Knutson](#), last updated November 10, 1993. Please don't remove this attribution.
