Siteswap Ben's

Guide to Juggling Patterns

Benjamin Beever
(benbeever@hotmail.com)

AD2000

Writings on juggling, for:

1) Jugglers

2) Mathematicians

3) Other curious people

ABOUT THE BOOK

The scientific understanding of ‘air-juggling’ has improved dramatically over the last 2 decades. This book aims to bring the reader right up to the forefront of current knowledge (or pretty close anyway).

There are 3 kinds of people who might be reading this (see cover). As far as possible, I wanted to cater for everyone in the same book. Maybe this would help jugglers to appreciate maths, mathematicians to get into juggling, or even non-juggling, non-mathematicians to develop a favourable perception of the juggling game.

In order to fulfil this ambition, I have indicated which parts of the text are aimed at a specific kind of reader. Different fonts have been used to indicate the main intended audience. Everyone (with many exceptions) should be able to understand most of the writings for the curious. However, non-jugglers may not be able to envisage the patterns discussed in the jugglers text, and non-mathematicians may find some of the concepts in the technical sections hard to get their grey matter around. I must point out however, that juggling is a fairly complicated business. It can be very hard to understand what’s going on in a juggling pattern when it’s being juggled in front of you, and it’s not always easy to understand on paper.

This book introduces ‘Generalised siteswap’ (GS) notation, and shows how most air-juggling patterns can be formalised within the GS framework. The reader is not assumed to have any knowledge of even basic ‘siteswap’ notation however. Siteswap (in all it’s forms) is introduced slowly and with much explanation.

For those who come to feel comfortable with the notation, appendices of solo and passing patterns can be found at the end of the book..

THE AUTHOR

My name is Benjamin Beever, aka ‘Siteswap Ben’. I am double-Gemini and a Christian (they happened in that order). I learnt to juggle 4 balls about age 14 (1990), then lost interest for 3 years. After seeing a certain 5 ball pattern on ‘Tomorrow’s World’ (UK TV program), my interest in juggling was rejuvenated. I am (was) 24, have a maths degree, have spent countless hours ‘juggling on paper’, have flashed 12 balls (the world record as far as I know), and am very curious (in both senses). Hopefully then, I am qualified to write this book.

As you might guess, I like juggling. I am writing this book, because if I’d had access to it when I was just starting out, I would have very much enjoyed reading it.

MOTIVATION

(The Curious)

Gravity is mysterious, but crucial. Why on Earth (or otherwise) would large objects with nothing in-between them, start moving towards each other? Scientists can’t tell us. They can tell us however, that if gravity didn’t exist, then neither would we.

In a sense then, we all ‘enjoy’ gravity. Although juggling is possible in ‘zero gravity’, most jugglers do not get a chance to experience this (it is also an entirely different ball-game). In what follows, I am therefore assuming the juggled objects are being acted on by gravity, which tries to make them move down, towards the ground. The general principle is thus: you throw something up, and after a short time (in the absence of intervening entities), it comes back down. If you are reading this sentence, you should already be familiar with this kind of event.

So why bother to keep throwing things up, if they are only going to come back down again?

Good question. I can only answer it superficially: because it is fun. Here are some lines of reasoning:

If I throw something up, then gravity will make it come back down. Gravity is magic. Magic is fun. Therefore juggling is fun. A fairly water-tight argument I hope you’ll agree. Here’s another:

Repeatedly throwing things up, and catching them, makes pretty patterns in the air that stay there for a while (until you drop them). Pretty patterns are fun to watch. Therefore juggling is fun. And another:

Keeping more than 2 objects off the ground, without holding more than 1 in each hand at any time, makes you feel clever. Feeling clever is fun. Therefore juggling is fun. And the reasons go on.... Lets face facts – juggling is fun.

My aim, through this book, is that the randomly curious will come to appreciate the different 'types of game' which can be played with gravity, by throwing and catching things.

(Jugglers)

Now we all know how much fun can be had throwing and catching things. But what can we do to make it even more fun? If we examine the arguments above, several ways suggest themselves:

1) Juggle under different gravity-situations. For example, juggle on the moon, on another planet, in space, under water, during 'freefall' etc. Not all of these are realistic options I know.
2) Juggle different patterns (see rest of book).
3) Juggle more objects to feel cleverer (see rest of book).

Another possibility is juggling whilst performing other actions. There are infinitely many things one could do at the same time as juggling, so I will list none of them – this book will take long enough to write as it is.

My objective (in the 'juggler-friendly' sections of this book) is to show you how to understand, record, communicate, and design juggling patterns, from the simple, to the ludicrously complex. In fact, jugglers who can stomach some of the mathsy bits, should, by the end of the book, be able to design (if not juggle) a 2 person, synchronous, multiplex, 9 ball 'Mill's Penguin', whatever that is.

(Mathematicians)

Juggling involves mathematics in at least 2 ways. Firstly, there are variable forces (hence accelerations) acting on juggled objects, exerted by the juggler (who can produce forces in any direction). Normally there is also the force of gravity (producing a 9.81 m/s^2 acceleration vertically downwards), and air resistance (which acts in the opposite direction to the motion of the object). We can pretty much ignore air-resistance, as the speeds attained by juggled objects are usually not sufficient for it to have a significant effect.

Thus, after an object is released (and before contact with another object), it has displacement s, approximately given by the famous formula:

$s = ut + \frac{1}{2} gt^2$,   where u is the initial velocity of the object downwards, t is the time elapsed, and g is the acceleration due to gravity.

If we say that gravity is about 10 m/s^2, the formula becomes: $s = ut + 5t^2$. If we wanted to calculate how long an object would spend in the air if we threw it a certain height, we need to get rid of the 'u' in the formula. To do this, split its journey into 2 halves: one where it goes up, the other where is comes down. These will take the same amount of time, so just consider the object on it's way down. In this case, u = 0, so the formula reduces to: $s = 5t^2$. We can then get: $t = \sqrt{(s/5)}$. As the object spends twice as long in the air as this (counting the time when it's going up), the total time in the air is given by: $T = 2\sqrt{(s/5)}$. This formula can be used to work out just how fast certain patterns need to be juggled, given that objects are thrown to a certain height (see page 23).

There is however a much more interesting mathematical side to juggling, which is concerned with the sequence in which balls are thrown....

<u>WARNING</u>

As much as I hate to admit it, this book may not be accessible to everyone. Those who intensely dislike symbols, or find it very difficult to translate them into meaning, will probably struggle with the notation. I make a fair effort to sell it to you, but you have been warned. For those who want a nice fluffy book, with lots of pictures, Charlie Dancey's superb (though scientifically, slightly naive) 'Encyclopaedia of Ball Juggling' is recommended. Those who want the hard facts about the nature of juggling patterns, read on....

CONTENTS

(A) = Suitable for all, (J) = Aimed at jugglers, (M) = Mathematical content

1) WHAT MAKES A PATTERN?

(In this book, I will refer to ‘juggled objects’ as ‘balls’, because I have a preference for this type of object.)

This chapter is designed to introduce juggling notation – that is, the juggler’s equivalent of sheet music.

I should point out in advance, that reading the following section (on the 3 ball cascade) is a bit like climbing a mountain: it may be hard work, but once you’ve made it, you get a good view of the surrounding area.

Here goes ....

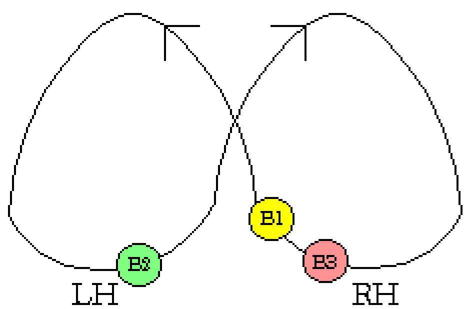
THE 3 BALL CASCADE

(A)

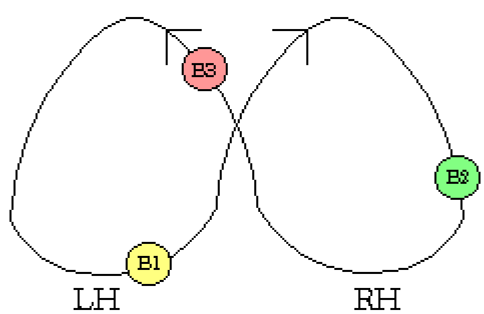
The easiest ‘proper’ juggling pattern is called the ‘3 ball cascade’. What is a ‘proper’ juggling pattern? For the moment, it means a pattern where there are more balls than hands, and no hand holds more than 1 ball at any time during the pattern.

It takes an ‘average’ (please don’t shout at me for using words like ‘proper’ and ‘average’ – they serve a useful purpose here) person about half-an-hour to learn to control this pattern well enough to make 6 consecutive catches.

What happens in this pattern? Well, you start with 2 balls in one hand (let’s say the right hand) and 1 in the other:



Your right hand throws a ball (B1) from just in front of your stomach, over to your left hand, which has to throw it’s ball (B2), before it can catch B1. By now, B2 is heading towards the right hand, which throws the last ball (B3) before catching B2. We should now be in the situation shown below:



– hopefully, you can see how the pattern continues.

It is all well and good to describe the 3 ball cascade using words and pictures like this, but how can we describe it in a totally precise way? You may ask, ‘Why would we want to do this?’ The reasons are at least 2-fold:

(a) So that we can more easily design new patterns.

(b) So that we can type the descriptions into a computer (which requires precise information) and watch the patterns being juggled (perfectly, and for as long as we want). This can help us learn to juggle them.

If you can't yet see why reason (a) is valid, be assured that you probably will, by the end of this chapter.

Now that you are convinced (or not) of the usefulness of the task, we shall proceed to build up a description of the 3 ball cascade:

<u>Throw time</u>

Throws are made at regular intervals. Let's say we throw the first ball at time 0. Then the second ball at time 1. Then the third ball at time 2. Then another (the first ball again) at time 3, and so on. We can formalise this as follows:

THR(Time) **{** 0 1 2 3 4 5 ... **}**

THR(Time) stands for 'throw-time', or 'time-of-throwing'.

The curly brackets ' **{** ' and ' **}** ' indicate that the values inside refer to a juggling pattern, as well as making the whole thing look more formal (as if we know exactly what we're doing).

<u>Throw site</u>

It is also important to know what part of our body we are throwing (or dropping) with. Most commonly it is the *hands* which are employed for this task. The 3 ball cascade is no exception. The right and left hands take it in turn to throw. We will use the letters 'R' and 'L' to denote the right and left hands respectively. We then have:

THR(Site) **{** R L R L R L ... **}**

Other possibilities for throw-sites are arms, elbows, head, neck, feet, chin, mouth, knees, armpits, shoulders, crotch, back – just about anywhere. These will also need letters or symbols to denote them (which will be given in the next chapter).

<u>Throw position</u>

Where, in relation to our body, are we going to launch these balls from? Well, for the sake of argument, we'll say that every ball is thrown from somewhere just in front of the middle of our body, so we write:

THR(Pos) **{** m m m m m m ... **}**

Note that I'm using lower-case letters for positions, and capitals for sites.

<u>Catch position</u>

Whereabouts the ball is going to be *caught* is equally important. In the cascade, the balls are caught at the sides of the pattern – that is, balls thrown by the right hand are caught on the left side, and vice-versa:

CAT(Pos) **{** l r l r l r ... **}**

i.e. left and right sides alternately.

Catch site

We can also record what site the ball will be caught by:

CAT(Site)	{	L	R	L	R	L	R	...	}

i.e. the left and right hands alternately.

Airtime minimum

I think most jugglers would agree, that in a proper 3 ball cascade, you are not allowed to hold more than 1 ball in either hand at any time. We could rephrase this by saying that every ball should stay in the air for at least 1 'beat' of time – that is, at least until the next ball has been thrown. So we can write:

AIR(Min)	{	1	1	1	1	1	1	...	}

– This says, after each throw, the ball should spend a minimum of 1 beat in the air.
In summary, so far, our 3 ball cascade description is:

THR(Time)	{	0	1	2	3	4	5	...	}
THR(Site)	{	R	L	R	L	R	L	...	}
THR(Pos)	{	m	m	m	m	m	m	...	}
CAT(Pos)	{	l	r	l	r	l	r	...	}
CAT(Site)	{	L	R	L	R	L	R	...	}
AIR(Min)	{	1	1	1	1	1	1	...	}

The above collection of rows/columns is called a 'matrix'. As a quick example of how to read this notation, let's take the throw at time 4: it says (reading down the column), '*4* beats after the start of the pattern, the *right hand* throws a ball from the *middle* of the pattern, to the *left* of the pattern, where it will be caught by the *left hand*, after spending at least *1* beat of time in the air'. By now it should be obvious why describing patterns in this formal way can be beneficial to pattern designers: change a letter or number in the matrix, and create a new pattern.

The above formal description isn't bad, but it still doesn't quite uniquely describe the 3 ball cascade. For example, a 5 ball cascade also satisfies these requirements. What we need is a way to 'link' the columns in the matrix, to tell us (for example) that the ball thrown at time 0 is the same ball as the one thrown at time 3. If you don't see why it is the same ball, think about it (or juggle it) – it should soon become clear.

Siteswap base

Don't worry about the name of this feature. It was invented by mathematicians and is therefore not surprisingly, technical-sounding. The reason for it's name will become clearer later.

In any air-juggling pattern, balls are thrown in a sequence. In the 3 ball cascade (and many other patterns), only 1 ball is thrown at a time. Now let's call these 3 balls 'a', 'b', and 'c'. First of all, 'a' is thrown by the right hand, then 'b' by the left, then 'c' by the right, then 'a' by the left, then 'b' by the right, then 'c' by the left, and so on. Writing down the sequence in which the balls are thrown, we have: a b c a b c a b c a b c .... It should be no surprise that each ball is thrown 3 throws later than it was last thrown – there are 3 balls, we treat them all the same way, so we have to deal with any given ball every third throw. We could represent this information formally as:

BALL(Identity)	{	a	b	c	a	b	c	...	}

However, a better way to express this feature, is by:

SS(Base)	**{**	3	3	3	3	3	3	...	**}**	'SS' = 'Siteswap'

This may require explanation. First, I should say that this row no longer tells us whether we are throwing ball 'a', 'b' or 'c'. If we wanted this information, then we could keep the BALLS(Identity) row. What the SS(Base) does tell us, is when a ball will next be thrown. In this case it says, 'Each ball that is thrown, will next be thrown *3 throws later*' – in other words, it will next be thrown in the manner described in the column 3 places to the right (in the feature-matrix).

The underlying concept behind the SS(Base) feature was only discovered around 1985, but as we shall see, it is a crucial attribute of all juggling patterns. (I shall sometimes call the SS(Base) values simply 'SS values' for short.)

We now have all the defining features of the 3 ball cascade:

THR(Time)	**{**	0	1	2	3	4	5	...	**}**
SS(Base)	**{**	<u>3</u>	3	3	3	3	3	...	**}**
THR(Site)	**{**	R	L	R	<u>L</u>	R	L	...	**}**
THR(Pos)	**{**	m	m	m	m	m	m	...	**}**
CAT(Pos)	**{**	l	r	l	r	l	r	...	**}**
CAT(Site)	***{***	*<u>L</u>*	*R*	*L*	*R*	*L*	*R*	*...*	***}***
AIR(Min)	**{**	1	1	1	1	1	1	...	**}**

To finish off this section, we will tidy this matrix up a bit. First, we can do away with the CAT(Site) row, as the values of this row can be calculated using the THR(Site) row and the SS(Base) row. For example, we know that the first value in the CAT(Site) row has to be 'L', because the SS(Base) row tells us that the first ball is next thrown 3 throws later, and the THR(Site) row tells us that this throw (occurring at time 3) is made by the left hand (L). This ball therefore must be caught by the left hand (note that the relevant matrix values have been underlined).

The other thing we can do, is to get rid of the repetition in the matrix. We don't want to have to specify what happens at every individual time (6, 7, 8, and so on), when these throws are exactly the same as those which occur at times 0 and 1. Let's just write out what happens until the throws start repeating:

THR(Time)	**{**	0	1	**}**
SS(Base)	**{**	3	3	**}**
THR(Site)	**{**	R	L	**}**
THR(Pos)	**{**	m	m	**}**
CAT(Pos)	**{**	l	r	**}**
AIR(Min)	**{**	1	1	**}**

Much nicer. So all we have to bear in mind is that the matrix 'loops around' – that is, when we reach the last column, we go back to the first again.

There is one last point to clear up here. It is rather technical, but I don't want to go into maths-mode and scare everybody off, because it is fairly important. So try to bear with me:

In order to make this condensed matrix description work, time has to go: 0 ... 1 ... 0 ... 1 ... 0 ... 1 ... – so that we can keep making the throws which happen at times 0 and 1. We therefore need to know when to 'reset' the time to 0. A bit of thought tells us that this should happen at time 2. We will therefore replace the '0' in the THR(Time) row, with a '2', to indicate this. Note that the '0' was actually redundant anyway (if we assume the pattern starts at time 0).

So, our final description is:

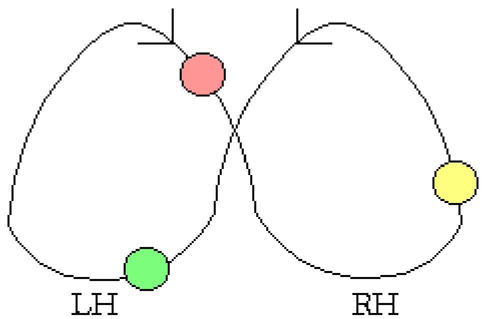
The 3 ball cascade

THR(Time)	{	2	1	}
SS(Base)	{	3	3	}
THR(Site)	{	R	L	}
THR(Pos)	{	m	m	}
CAT(Pos)	{	l	r	}
AIR(Min)	{	1	1	}

This is the GS (generalised siteswap) description of the 3 ball cascade. For the non-mathematicians who are still here: well done – that's the hardest part over with.

THE REVERSE CASCADE

The reverse cascade differs from the standard cascade, in that balls are thrown from (outside) right and left positions, into the middle of the pattern:



In GS notation, the 3 ball reverse cascade is:

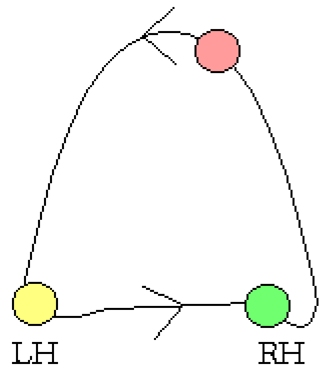
THR(Time)	{	2	1	}
SS(Base)	{	3	3	}
THR(Site)	{	R	L	}
THR(Pos)	{	r	l	}
CAT(Pos)	{	m	m	}
AIR(Min)	{	1	1	}

Throwing from the sides and catching in the middle is a technique rarely used by most jugglers, although being able to hold a fairly stable reverse cascade is agreed to be a prerequisite for the famous 'Mills Mess', and many other patterns besides.

THE SHOWER

This is the pattern of the masses! In some countries it is the only known juggling pattern, and even in well-developed (in juggling terms) countries, it is portrayed as *the* way to juggle, by cartoons, logos, and other medias. Persuade a 'non-juggler' to juggle, and they will attempt (often quite successfully) to perform a 2 ball shower. Of course, there is a distinct lack of urgency about this pattern, due to the numerical equality between hands and balls. Adding an extra ball increases the hands' workload considerably – the 3 ball shower is regarded by most, to be harder than the 3 ball cascade. The shower is an asymmetric pattern, so I shall take this opportunity to apologise to left-handers and ambidextrons – you may switch the words (or symbols for) 'right' and 'left' anywhere you like.

In the pattern, the right hand throws, whilst the left hand passes balls across to the right, as shown below:



Now there are 2 ways to juggle the shower: using alternating-handed throws (R, L, R, L, ...), or synchronously (both hands throwing simultaneously). Most jugglers learn the alternating version first, and then progress to the synchronous version (which is considered to be easier) when the pattern 'clicks'. In GS notation, the alternating (3 ball) version is:

THR(Time)	**{**	2	1	**}**
SS(Base)	**{**	5	1	**}**
THR(Site)	**{**	R	L	**}**
CAT(Site)	***{***	*L*	*R*	***}***
AIR(Min)	***{***	*3*	*0*	***}***

(often referred to simply as '5 1')

I have added the redundant CAT(Site) row for readability, and under normal conditions (no more than 1 ball in a hand) the AIR(Min) row is also redundant, as it can be calculated from THR(Time) and SS(Base) – details later. Notice also that I have not specified what positions throws and catches occur; I don't really think that they are defining features of this pattern.

Compare the synchronous version:

THR(Time)	**{**	2	0	**}**
SS(Base)	**{**	5	1	**}**
THR(Site)	**{**	R	L	**}**
CAT(Site)	***{***	*L*	*R*	***}***
AIR(Min)	***{***	*2*	*0*	***}***

Notice that both hands throw together, at times 0, 2, 4, 6 etc. I could have defined it so that both hands throw together, at times 0, 1, 2, 3, 4 etc., but it's more practical to work on the principle that the same hand does not have to throw more than once every 2 beats.

There are at least 2 reasons why the synch version is easier. Firstly, as can be seen from the AIR(Min) row, balls don't have to be in the air for as long. Secondly, it is impossible to juggle the alternating version with the 'showering hand' (the one throwing the '5's) being empty for less than half the time (see the discussion of the SS value '1' on page 22 to understand why this is so).

For completeness, I should say that in 'extended siteswap' ('ES') notation, the synchronous version is written (4x,2x). An explanation of synchronous patterns can be found on page 17.

More patterns (in GS notation) can be found in the appendices, but now we will take a closer look at the range of possibilities for each feature of a juggling pattern, and how we can get a written description of patterns which use them.

2) PATTERN FEATURES

THROW & CATCH SITES

(A)

Most jugglers concentrate on juggling using just 2 sites: their right and left hands (denoted ‘R’ and ‘L’ respectively). This is, of course, because our hands are by far the most dextrous parts of the body. There are many other potential sites one can use for juggling though. Below, I have listed all the ones I could think of, and the approximate maximum SS value I think could be accurately thrown from that site (as a 1-off throw in an otherwise alternating 2 handed pattern). Suggested GS notation for each site is also shown.

Site	Max throw height	GS notation
right / left hand	20	R / L
right / left foot	15	FR / FL
on right / left arm	15	AR / AL
back of neck	10	N
right / left elbow	10	ER / EL
right / left knee	10	KR / KL
top of head	5	H
on chin	5	OC
on nose	5	Ns
on forehead	5	Fh
in mouth	5	M
right / left shoulder	5	SR / SL
under chin	3	C
under right / left arm	3	UR / UL
between legs	1	BL
on the ground (, table etc)	1	G
juggler number 7’s knee, mouth, ...	--	7-- (eg ‘7H’ = ‘juggler 7’s head’)

To illustrate the usefulness of using symbols to denote sites, below is a pattern which would be difficult to describe otherwise, as it uses sites other than R and L; namely UR and UL:

Orang-utan Fountain (5 balls)

SS(Base)	**{**	8	4	8	3	8	4	8	0	3	4	.	8	4	8	3	8	4	8	0	3	4	**}**
THR(Site)	**{**	R	UR	L	UL	R	UR	L	UL	R	UR		L	UL	R	UR	L	UL	R	UR	L	UL	**}**

- Firstly, I must state that this pattern is purely to illustrate the point about using different THR(Sites), so don’t worry if you can’t work out what’s going on. If you really want to try it now, then the following might help (if you don’t, then skip the rest of this paragraph). Don't be put off by the 8's; there are four throw-sites so the 8's can be thrown as if they were 4's in a two-handed pattern. Notice also that the 4's can be held in this pattern because the same throw-site is used every fourth throw. The effect is of four throws of a standard four-ball fountain, then the next hand to throw puts a ball under the opposite arm, after which the cycle repeats starting from the other hand. Every so-often a ball drops from under an arm into that arm's hand. (Note the dot between the ‘4’ and the ‘8’ in the middle of the pattern – this is where the pattern starts repeating on the other side.)

We will return to this pattern later, where an easier way to ‘read it’ will be given.

The sites used in a pattern, are recorded (in GS notation) by the THR(Site) row (and the optional, though redundant CAT(Site) row).

THROW & CATCH POSITIONS

With our hands (and a few other parts of our body), we can throw and catch objects at different positions, such as under an arm, behind our back, under a leg, above our head etc. This is a crucial element of many patterns, such as ‘Mills Mess’, ‘The Machine’, ‘Trickledown’ etc. What follows is a list of possible throwing and catching positions, along with the position-axis they modify (up-down, left-right or forward-back). I have also given my opinion of the approximate ‘awkwardness’ (difficulty) rating (out of 10: 0 = natural, 10 = very difficult) for throwing/catching at this position, assuming the throw/catch is ‘normal’ in every other way. Again, the suggested GS denotation of each position is also shown. Note that this table is just to give some idea of the possible throwing and catching positions to those who want to design their own patterns. You will not be tested on it later.

Position(modifier)	Axis modified	Throw difficulty	Catch difficulty	GS notation
right / left	left-right	0, 3*[1]	0, 2*[1]	r / l
middle	left-right	0	1	m
inside	left-right	0	1	i
outside	left-right	1	0	o
low down (stomach height)	up-down	0	0	d
central (chest height)	up-down	1	1	c
high up (face height)	up-down	3	2	u
over head	up-down	4	2	h
below right / left hand	up-down	0	0	bR / bL
above right / left hand	up-down	0	0	aR / aL
below right / left knee (leg)	up-down	4	4	bKR / bKL
in front of body	forward-back	0	0	f
in same plane as body	forward-back	0	0	p
behind back	forward-back	6	7	bb
Albert position	all	10, 8*[2]	10	alb
trebla position	all	8, 6*[2]	8	treb
degree modifier*[3]	any	--	--	1 / 2 / 3 /...

*[1] Depending on whether a hand is throwing/catching on it’s natural side of the body or not (respectively).

*[2] Depending on whether balls or clubs are being used (respectively). For those who don’t know, the Albert position is under the crotch (with both feet on the floor), throwing or catching behind you. An actual ‘Albert’, is understood by most jugglers to be a *throw* (usually of a club) from the Albert position. The trebla position is the same, except your arm reaches from the back to the front (the ball thrown or caught in front, between your legs).

*[3] To use a degree modifier in a GS description, simply place the number in front of the position modifier, eg 1r (1 degree right), 2u3l (2 degrees up, 3 degrees left), etc. Note that these are not *absolute* positions – they are only relative to other positions.

Note: the throw/catch difficulty scores here (and on the next page) are only a bit of fun. They are not ‘scientific’. You *can* though use them as a guide for designing patterns of a practical difficulty.

Positions at which throws and catches are made, can be recorded by the THR(Pos) and CAT(Pos) rows in a GS matrix.

THROW & CATCH TYPES

Normally, when using their hands, jugglers throw and catch with their fingers pointing forward and their palms upward. There are other hand-orientations though, with varying movement ranges and comfort:

The ‘claw’ – Here the ball is held in the hand, palm downwards. This is mostly used for rapidly snatching balls downward out of the air; this downward motion is usually continued for a short time, in order to quickly move the ball to a different part of the pattern, ready for its next throw. See the patterns ‘Rubenstein’s Revenge’, ‘The Factory’, and ‘Shuffle’, for instances of the claw.

The ‘backhand’ – As the name suggests, this is where the ball is balanced on the back of the hand. This is rarely used for either throwing or catching, as it is highly difficult, without adding much to the pattern. And I have never seen anyone attempt a backhand (throw or catch) with a club or a ring.

The ‘fork’ – The ball is balanced between the first 2 fingers (in a vee shape), with your palm downwards. This leaves 2 fingers and a thumb to hold another ball (in a ‘claw’) if desired.

The ‘inverse’ – Here the palm is upward, the fingers point at least slightly backward, and the arm is in the same sort of position as if you are about to throw a dart. This hand-orientation makes it the obvious choice for throwing and catching high up (above head height), and most useful for saving balls which would otherwise land behind you.

The ‘penguin’ – The ball is held similar to normal, but the hand (palm upwards) is turned inward (about ¾ of the way round), so that the fingers point outward, even slightly forward. This is such a strange-looking orientation, that it is the basis for its own, quite popular (but difficult) 3-ball pattern.

Below I have listed these, along with my opinion of how difficult it is to throw/catch in this way; this score relates specifically to using balls. Also indicated is the most natural position for executing these throws and catches.

Type	Throw difficulty	Catch difficulty	Natural position	GS notation
normal	0	0	df	n
claw	2	4	cf	c
backhand	3	6	cf	b
fork	4	8	cf	f
inverse	5	2	hp	i
penguin	8	6	odf	p

There are many other ways to hold balls – see if you can think of some.

The types of throws and catches used in a pattern can be recorded by the THR(Type) and CAT(Type) rows in a GS matrix.

CLUBS

Unfortunately, not all juggling objects are quite as soft, friendly and lusciously spherical as balls. In fact, the juggling club can be your worst enemy. One minute it’s happily jumping from one hand to the other; the next, it’s dive-bombing your head. When irritated by constant harassment, they have also been known to inflict severe hand-damage. Unpleasant experiences have thus made me unwilling to entertain these objects on a regular basis.

Although clubs show little respect for our delicate hands, they do have some redeeming features, the main one being that you get to play an extra game whilst juggling them – having to get them to do a whole number of spins in the air before catching them. (This does make them considerably more difficult to juggle than balls.)

Another nice feature of clubs, is that throwing alberts and treblas is far easier than with any handle-less prop. Also, I have to admit that there is something particularly pleasant about passing clubs – even if it's because you get to randomly throw uncountable-spinning buzzsaws (instead of single spinners) at your unimpressed friend.

Anyway, there are 2 extra features I imagine club jugglers might want to record about club patterns: the number of times the club spins in the air, and twirly bits you do with the club just after you catch it, such as 'flourishes'.

The GS row AIR(Spin) is used to record the first of these. Possible values in this row include 0, 1, 2, 3, ... for 'flats', single-spins, doubles, triples, etc (or ¼, 1¼, 2¼, for passed flats, singles, doubles etc). Also possible, are –1, –2, –3, ... for reverse spins, or 1½, 2½ etc for wrong-end catches. Here, for example, is 5 clubs (in the SS '5'), juggled on double-spins.

THR(Time)	**{**	2	1	**}**
SS(Base)	**{**	5	5	**}**
THR(Site)	**{**	R	L	**}**
AIR(Spin)	**{**	2	2	**}**

The other aspect (the twirly bits), I am not so sure about. I think I should leave it up to an expert club juggler (or individuals) to decide how these should be recorded.

RINGS

Another topologically challenged piece of apparatus, is the infamous 'ring'. Juggling rings is an effective method of finger amputation. They can also cause calluses, and other unpleasant things. Of course, I am not trying to put you off them; it *is* possible to have fun juggling rings (I've heard). Actually there is considerable artistic scope afforded by this prop.

Normally it's difficult to tell how fast they are spinning, because they are thrown with flat spin (so spin doesn't really matter, unlike with clubs). They can be thrown either with the edge forwards, or with the ring in a left-right plane. They can also be thrown so that they flip over (like pancakes being tossed). To record the type of flight a ring performs, we can use the AIR(Ring) row in the GS matrix. The main values you might want to use in this row are:

e (edge on, flat spin), s (side-on, flat spin), p (pancake flips). If you were particularly interested in the number of flips in the pancake throws, you could use p½, p1, p1½ etc (pn = a pancake throw with n whole flips).

Some of the more interesting ring manipulation involves techniques closer to those of contact juggling than air-juggling; this is not really catered for by GS notation.

BOUNCING BALLS

An 'AIR(Bnce)' row can be used to record how many times a ball bounces before being caught. See page 66 for suggestions of siteswaps which work particularly well as bounce-patterns.

TV REMOTE CONTROLS

Jugglers who enjoy experimenting with throwing various kinds of household item will no doubt have discovered the fun to be had with remote controls. The best ones have a fair weight to them. For those who still don't know what I'm on about; with a remote control (or any other, fairly thin, block-like object), you get one dimension more than with clubs – you get to play the whole-number-of-twists game as well as the whole-number-of-somersaults game. I should point out that it is extremely irresponsible to attempt these feats with a remote control over a hard floor – in case it slips through your fingers and becomes a multiplex.

Anyway, to record the number of twists, we can use 'AIR(Twst)', which takes values of ½, 1, 1½, 2 etc.

3) SITESWAP NOTATION

‘VANILLA’ SITESWAP (VSS)

(A)

Siteswap (SS) notation is a way of writing down a key feature of juggling patterns: the order in which the balls are thrown. It was discovered around 1985 by several groups of people who found it independently. It is quite incredible how it took so long to find such a simple yet powerful tool. In fact, it’s as easy as ‘1 2 3’, which is a valid, jugglable, SS sequence with 2 balls. To illustrate it’s usefulness, it’s discovery made the number of known (practical) juggling patterns increase about ten-fold overnight. I’ll now tell you how to ‘read’ the simplest form of SS notation.

Let’s assume (for the moment) that you are juggling 3 balls, and you only throw 1 ball at a time, with your right and left hands throwing alternately. We’ll call the balls ‘a’, ‘b’, and ‘c’. Suppose you (somehow) throw the balls in the order: a b c c a b b c a a b c c .... SS notation records this sequence as ‘4 4 1 4 4 1 4 4 1 ...’ Why? Because the first ball (a) is next thrown 4 throws later (ie with 3 intervening throws – b c c). The second ball (b) is next thrown 4 throws later (after c c a). The next ball (c) is next thrown 1 throw later, and so on. – This is the key definition of SS: “Throwing a (SS value of) X, means that *this same ball* will be *next thrown, X throws later*” (ie after X – 1 intervening throws). This definition suggests that higher SS values need to be thrown higher into the air.

Normally, when juggling the SS sequence ‘V_1 V_2 V_3 V_4 V_5 ...’ (using 2 hands), your RH throws the ‘V_1’, then your LH throws the ‘V_2’, then your RH throws the ‘V_3’, and so on.

Now the sequence 4 4 1 4 4 1 4 4 1 ... repeats, so to save time (and paper), we just write down the smallest repeating sequence, ie 4 4 1. Note that the average of this sequence is ((4 + 4 + 1) ÷ 3 =) 3, which is the number of balls in the pattern. In general, the number of balls in a SS, is equal to the sum of the SS values, divided by the number of throws in the pattern.

As we have seen in the first section of this book, a 3 ball cascade has SS notation ‘3’. Now a 4 ball fountain has SS notation ‘4’, a 5 ball cascade ‘5’, and so on. These are the ‘simplest’ (in SS terms) patterns, because all the throws are ‘the same’. The ‘period’ (length, cycle-time) of these patterns is 1, because each throw is like the last.

A commonly juggled 4 ball pattern is ‘5 3’ (period 2). The right hand throws ‘5’s, and the left hand throws ‘3’s. Suppose you decided to try to juggle this pattern. How would you go about it? Well, in patterns where the 2 hands throw alternately, even-valued SS throws land back in the same hand (I’ll call these ‘straight’ throws), whilst odd-valued throws travel to the other hand (‘crossing’ throws). To see why this is, look at the following:

RH	**?**		2		4		6		8		...
LH		1		3		5		7		9	...

– Suppose we throw a ball with the right hand (this throw is denoted by the ‘**?**’). If we were to next throw this ball 2 or 4 or 6 ... throws later, we will be throwing it with the right hand again, so the ball must have landed back in the right hand. If on the other hand (no pun intended), the ball is next thrown 1 or 3 or 5 ... throws later, we will then be throwing with our left hand, and so the ball must have *landed* in the left hand.

To return to trying to juggle ‘5 3’; we know therefore, that all of the throws must travel to the other hand. We also know that we are going to have to throw the ‘5’s higher than the ‘3’s – in fact, the ‘5’s should be thrown so that they spend about 2 beats longer in the air.

The SS value ‘0’ is slightly odd (no lie intended). The definition of SS given above breaks down for ‘0’s; lets try it: a SS ‘0’ means ‘you next throw this ball 0 throws later’. Let’s humour this for a second. This would mean that you have to throw the same ball again instantly, and again, and again, an infinite number of times, all in the space of no time at all. This is clearly either meaningless, or impossible to achieve.

Back in reality, we can use the averaging rule to suggest a meaning for ‘0’s: If we ‘juggle’ the pattern ‘0’ (ie perform a string of ‘0’s), we would need (0 ÷ 1 =) 0 balls, and hence no balls are involved when we ‘throw’ ‘0’s. In other words, a ‘0’ is not a throw at all – rather it is a ‘gap’ in the pattern – an opportunity to re-adjust your hair.

A convention has arisen amongst siteswappers, to write or speak SSs with the highest value first (and where this value occurs more than once, with the highest possible next value, and so on). This is useful because it means that we all refer to SSs by the same name.

POPULAR MISCONCEPTIONS

By now, most jugglers have heard something about siteswap (SS) notation, and grudgingly accepted that sequences of numbers might serve a useful function in juggling. There are however, still several myths about SS floating around. 10 of them will now be explored.

1) "The SS values tell you how high to throw."

This is not quite true as it stands. However, under 'normal conditions' (not allowing balls to bounce etc), provided we know what throw-rate and holdtime is being used, the quotation becomes true.

2) "The time a ball spends in the air, is always 1 beat less than it's SS value (ie '3's spend 2 beats in the air etc)."

This is not a part of the definition of SS. As an example, '3's (in a 2-handed pattern) can spend anywhere between 1 and 3 beats of time in the air. More generally SS values of X (= 2 or higher) have between X – 2 and X beats of airtime, '1's have between 0 and 1 beats of airtime, and '0's have no airtime (see 'Hold Time', page 21).

3) "A '2' is a tiny hop to the same hand."

When you do a '2', you are promising to throw that same ball, 2 throws later – ie the very next throw from the same hand. This means it is quite 'safe' to hold the '2', as no other ball needs to be dealt with by that hand in the meantime. This is by far the easiest way to deal with '2's in practice. (If we particularly want a '2' to be thrown into the air, rather than held, we can denote it by '2_T'.)

4) "If the average value of a SS sequence is a whole number, then it is a valid pattern."

Try juggling '5 4 3'. The first 3 throws require their balls to be next thrown at the same time as each other. This means that 3 balls will have to be thrown at the same time (on the 6^{th} throw), which is denied by the pattern (as only a single ball (a '3') is supposed to be thrown at this time). There *are* tests for validating SSs, as we shall see later.

5) "SS notation assumes 2 hands are being used."

The notation does not impose any restrictions on the number of hands, feet, noses, or tables being used. It *is* however a convention, that unless specifically stated otherwise, people who talk about SS sequences are assuming that they are being juggled using 2 hands and nothing else. I should also point out, that when a different number of throw-sites are being used, the rule about even SS values landing in the same hand (or site) and odd values in another, no longer works. For example, with 3 hands, '3's, '6's, '9's etc. land back in the hand that threw them.

6) "If I throw a '5', followed by a '4', I have to catch 2 balls at the same time."

Suppose you throw a '5' at time 0, then a '4' at time 1 (in an alternating 2-handed pattern). The definition of SS states that you will have to *throw* both of these balls at time 5 (ie do a multiplex). Under normal conditions, you will also have to *catch* both of these balls in the same hand, between times 3 and 5 (more realistically, between times 3 and 4, as you need a bit of time to prepare the throw), which will be quite difficult. Note however, that if you hold the '2', then juggling the sub-sequence '3 2' does not pose this second problem.

7) " '3' is an accurate description of the 3 ball cascade."

We have already seen from the first chapter that this is not so. Many other details are needed to describe the cascade properly, such as where balls are thrown and caught. (Several other patterns also have SS description '3'.)

8)   " '3' is the only SS description of the 3 ball cascade."

There are infinite SS sequences which the 3 ball cascade can be juggled on: 5 2 2, 7 2 0, 9 0 0, 7 2 2 2 2, ... These other descriptions contain information on how much hold time is allowed. The three period 3 patterns above (5 2 2, 7 2 0 and 9 0 0), divide the pattern '3' into three subpatterns: 5 2 2 (the laziest), 9 0 0 (the most energetic), and 7 2 0 (mid-effort). (This is assuming that the '2's are held.)

9)   "SS is no use for describing patterns like Burke's Barrage, and Rubenstein's Revenge."

Just as humans have a spiritual dimension, juggling patterns have a SS dimension. Trying to describe any juggling pattern without indicating it's underlying SS sequence, is like trying to describe your new sports car without referring to it's colour. Burke's Barrage has the underlying SS '4 2 3'. Many people struggle with learning this pattern because, not knowing it is '4 2 3', they don't realise it involves a 2-in-one-hand throw. Similarly, Rubenstein's Revenge is best understood as '5 2 2 3 3' (with '2's held); you can call it a shape distortion of a 3 ball cascade ('3'), but this obscures the use for a higher underarm throw before the twirly bit (the 'orbit').

10) "I finish my routine with a '7'."

The last few throws you ever make in your life have no siteswap values; there is no 'next throw' of these balls. By the same token, a '7' is not a *real* '7', unless you throw the ball again *7 throws later*.

SYNCHRONOUS PATTERNS

Synchronous patterns are where 2 or more sites throw at the same time. Most commonly, it is where the right and left hands throw together. So instead of throwing R L R L R L ..., we throw (R&L) (R&L) (R&L) ....

First, let's assume that there are 2 beats of time between successive (dual) throws; so both hands throw at time 0, then at time 2, then 4, then 6, and so on. We could have said both hands throw *every* beat (ie at times 0, 1, 2, 3, ...), but this would afford less comparison with alternating patterns, where each hand throws only once every 2 beats. Let's illustrate what I mean:

	Time:	0	1	2	3	4	5	6	7	8	9	10	11	...
Alternating patterns:		R	-	R	-	R	-	R	-	R	-	R	-	...
		-	L	-	L	-	L	-	L	-	L	-	L	
Synchronous patterns:		*R*	-	R	-	R	-	R	-	R	-	R	-	...
		L	-	L	-	L	-	L	-	L	-	L	-	

You can think of synchronous patterns as alternating patterns where the left hand has been 'slid back' 1 timebeat. More on site-sliding later.

So what SS values are possible in synchronous patterns? Let's suppose we are throwing the right hand throw indicated by '*R*'. Now this ball must be next thrown an *even* number of beats later (at time 2 or 4 or 6 etc) – because no throw occurs at an odd time in the pattern's future; therefore odd values are impossible. We are however, allowed to throw even-valued throws to the other hand. Normally, even values are straights and odd values are crossing; when this is not the case, ES ('extended siteswap') notation uses a cross-symbol ('x'), or a straight-symbol ('s') to indicate otherwise (eg '4x', '5s'). We also put synchronous throws in round brackets, to indicate that they occur simultaneously. As an example, the ES notation for the (synch) 3 ball shower is (4x,2x) – the right hand throws '4x's, the left hand '2x's.

Note the averaging rule still works; eg (4x,2x): (4 + 2) ÷ 2 = 3 balls. Note also the theoretical possibility (but practical impossibility) for a ‘0x’.

MULTIPLEX PATTERNS

These are patterns which contain a multiplex throw – ie where a site throws more than 1 ball at the same time. Putting it in the same sort of format as above, an alternating multiplex could look something like:

Time:	0	1	2	3	4	5	6	7	8	9	10	11	...
	R	-	R	-	R	-	R	-	R	-	R	-	
	R				R		R						...
		L	-	L	-	L	-	L	-	L	-	L	
				L						L			

We can even have synchronous multiplex patterns:

Time:	0	1	2	3	4	5	6	7	8	9	10	11	...
	R	-	R	-	R	-	R	-	R	-	R	-	
	R				R		R						...
	L	-	L	-	L	-	L	-	L	-	L	-	
					L						L		

In ES, multiplex throws are indicated by square brackets, eg [7,6] means throw a ‘7’ and a ‘6’ with the same hand at the same time. Multiplex patterns are generally quite difficult. The main problem with most of them, is that 2 or more balls have to be caught by the same hand within a short space of time (about 1 beat’s worth). This problem can be avoided however, if all-but-one of the multiplexed balls are held (on ‘2’s) immediately prior to being multiplexed; eg [5,4] 2 4 (see pages 33-34 to find out how to design multiplex patterns).

The averaging rule for multiplex patterns works by adding together the values inside a multiplex, but counting it as a single throw, so for [5 4] 2 4, we get ((5+4) + 2 + 4) ÷ 3 = 5 balls.

SYMMETRIC PATTERNS

Assuming we are using only our right and left hands, alternating SSs with odd length (period), such as 5 0 4 and 8 1 4 7 5 (or are reducible to odd period, such as 6 4 5 6 4 5) are (mirror) symmetric. This is because whichever hand begins the sequence, also ends the sequence, meaning that the other hand then restarts the sequence, so the pattern repeats on the other side. Patterns which are not reducible to odd period are asymmetric. Some synchronous patterns are also symmetric, such as (6x,4)(4,6x). As a convention (for concision’s sake), I suggest we use ‘*’ to indicate a synch pattern which repeats on the other side, so (6x,4)(4,6x) can be written (6x,4)*. See page 34 for details of how to generate symmetric synchronous patterns efficiently.

GENERALISED SITESWAP (GS)

SS is the last piece of the puzzle, as far as describing juggling patterns is concerned; I invented GS notation in order to collect the pieces together.

A pattern in GS notation has the form of a matrix (a 2-dimensional list of symbols). Each column specifies a complete object-journey from throw to catch. Each row describes a particular feature of the pattern, such as the catch-positions. Each of these feature-rows are optional; if you don't care where balls are caught, you don't need to specify this information. You just take the 'tools' you need for the job. A fairly adequate collection of these tools (feature-rows) is given below:

Primary kit:

	Main	(Detail)	Meaning of **{**... X ...**}**
1.	SS	(Base)	The object's next journey is described X columns to the right
2.	THR	(Time)	The *throw* occurs at *time* X (ie X beats after the start of the pattern)
3.	THR	(Site)	The *throw* is made from *site* X (eg left hand, on the ground etc)
4.	THR	(Pos)	The *throw* is made from spatial *position* X (eg inner right, low down, left etc)
5.	THR	(Type)	The *throw* is of *type* X (eg normal, backhand, penguin etc)
6.	AIR	(Min)	The object spends a *minimum* of X beats of time in the *air* on this journey
7.	AIR	(Rec)	*Recommended* (approximate) *airtime*
8.	AIR	(Max)	The object spends a *maximum* of X beats of time in the *air* on this journey
*9.	AIR	(Spin)	The club (or other spinning object) performs X *spins* in the *air*
*10.	AIR	(Ring)	The *ring* spends it's *airtime* in fashion X (eg edge-on, pancakes etc)
*11.	AIR	(Bnce)	The ball *bounces* X times during this journey
*12.	AIR	(Twst)	The object performs X *twists* in the *air*
13.	CAT	(Pos)	The *catch* is made at spatial *position* X (examples as for 'throw-pos')
14.	CAT	(Type)	The *catch* is of *type* X (examples as for 'throw-type')

Secondary kit: (These are derived features – useful for visualising the pattern.)

15.	SS	(Real)	The ball is next thrown X beats later (calculated from tools 1 & 2)
16.	SS	(As)	Perform action X (details given later) (calc'd from 1, 2, 3 & 6-8)
17.	THR	(Dir)	The ball is *thrown* in *direction* X (calc'd from 4 & 13)
18.	CAT	(Time)	The ball is *caught* at *time* X (calc'd from 2 & 6-8)
19.	CAT	(Site)	The ball is *caught* by *site* X (calc'd from 1 & 3)

* 9, 10, 11, and 12 are mainly useful with clubs, rings, bouncing balls, and twisty things respectively.

Notice the difference in definition between ES, and the GS SS(Base) row. In GS, The SS(Base) will always have the form of a VSS, but the THR(Time) row is also crucial in determining how many beats later a ball will next be thrown. Recall the 3 ball shower: in ES the alternating and synchronous versions are '5 1' and '(4x,2x)' respectively. In GS, the SS(Base) is **{** 5 1 **}** for both; it is the THR(Time) which changes from **{** 2 1 **}** to **{** 2 0 **}**, and hence 'converts' the SS **{** 5 1 **}** into a synchronous pattern.

The primary features have either already been described, or are self-explanatory. The CAT(Time) and CAT(Site) rows are also straightforward. The THR(Dir) indicates the direction of a throw, eg r = right, dl = down and left, s = straight up, l1 = left a bit, r5 = right a lot etc. Descriptions of the SS(Real) and SS(As) rows follow.

SS(Real)

(M)

In patterns where 1 ball is thrown every beat, we can read the SS(Base) row like a VSS. However, for patterns where this is not the case (eg synchronous patterns), it is less obvious how high the throws should be. It is for this purpose that we generate the SS(Real) row (which will mirror the ES description). To calculate the SS(Real) value of a throw, simply subtract its THR(Time), from the THR(Time) of the next throw made with the same ball (adding on the cycle-time if you have to loop).

For example, the 5 ball synchronous SS, (6x,4)(4,6x) (or (6x,4)*) has GS description:

THR(Time)	**{**	4	0	2	2	**}**
SS(Base)	**{**	7	4	4	5	**}**
THR(Site)	**{**	R	L	R	L	**}**

To calculate the first SS(Real) value, we note that the ‘7’ is next thrown as the ‘5’ (counting right 7 columns). This ‘5’ occurs 2 – 0 + 4 beats later (the THR(Time) of the ‘5’ (=2), minus the THR(Time) of the ‘7’ (=0), plus the cycle time of the pattern (=4) because we had to loop back to the start once). Anyway, this equals 6, so that’s the SS(Real) value. Continuing in this way, we end up with SS(Real) row:**{**6 4 4 6 **}**. To further increase its readability, we could note which throws go to the other hand, and put ‘x’s in appropriately (we could also bracket the synch throws) to get **{** (6x 4) (4 6x) **}**.

The reason why SS(Base) and SS(Real) values sometimes differ, is because SS(Base) is ‘ordinal’, and SS(Real) is ‘cardinal’. In other words, GS matrices put the throws (columns) in order (even ordering throws which happen at the same time), and then the SS(Base) indicates the number of *throws* later at which a ball is next thrown. SS(Real) (and ES notation) on the other hand, measures the number of *timebeats* later at which the ball is next thrown.

SS(As)

This one is seriously technical. It’s also rather imprecise. It has the same purpose as the SS(Real) row (readability), but it is used for the more unusual patterns – those for which even the SS(Real) row fails to indicate the necessary throws. We have already met an example of this kind: the Orang-utan fountain. I’ve included a standard THR(Time) row and the SS(Real) row (which you will notice is identical to the SS(Base).

THR(Time)	**{**	20	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	**}**
SS(Base)	**{**	8	4	8	3	8	4	8	0	3	4 .	8	4	8	3	8	4	8	0	3	4	**}**
THR(Site)	**{**	R	U_R	L	U_L	R	U_R	L	U_L	R	U_R	L	U_L	R	U_R	L	U_L	R	U_R	L	U_L	**}**
SS(Real)	***{***	*8*	*4*	*8*	*3*	*8*	*4*	*8*	*0*	*3*	*4*	*8*	*4*	*8*	*3*	*8*	*4*	*8*	*0*	*3*	*4*	***}***

What we need is a SS(As) row:

SS(As)	***{***	*4*	*-*	*4*	*dr*	*4*	*-*	*4*	*-*	$1U_L$	*-*	*4*	*-*	*4*	*dr*	*4*	*-*	*4*	*-*	$1U_R$	*-*	***}***

Notation: dr = drop into hand, $1U_L$ = put under left arm, $1U_R$ = put under right arm.

To make this row, ‘8's become ‘4's (because there are 4 sites rather than 2); ‘3's going from U_R/U_L positions become ‘dr's, ‘3's from R/L positions become $1U_L$/$1U_R$ (as they are placed under an arm), and ‘4's & ‘0's are ignored, as they require no action (‘4’s require no action in 4-site patterns, just like ‘2’s in 2-site patterns). Incidentally, to juggle this pattern as written, start with a ball under each arm, two in the right hand and one in the left.

Another reason you might want to make a SS(As) row, is if the THR[Time] is not **{** P 1 2 3 4 ... **}**, or if ‘R’ and ‘L’ do not occur alternately in the THR(Site) row. For example:

THR(Time)	**{**	5	0	2	3	**}**
SS(Base)	**{**	5	3	4	4	**}**
THR(Site)	**{**	L	R	R	L	**}**
SS(Real)	***{***	*5*	*5*	*5s*	*5s*	***}***
SS(As)	***{***	*(4x*	*5)*	*5s*	*4*	***}***

If you want to try this pattern, think of it as $(5_R,4x)\ 5s_R\ 4$, the right hand throws having subscript ‘R’.

Here is the pattern (using the SS(Real) values), with the hands separated:

Time:	0	1	2	3	4
R:	5	--	5s	--	--
L:	5	--	--	5s	--

Notice how the ‘5-straight’ has reared its ugly head in this pattern – this is to be expected, since the pattern changes from synchronous to alternating and back again. But how did we get the SS(As) row?

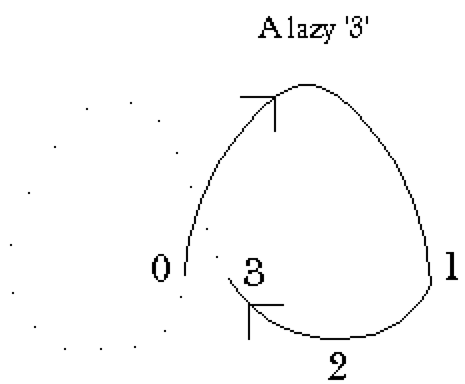
Normally ‘5’s have to be in the air for at least 3 beats. But the ‘5’ at time 0 from the left hand (going to the right), can land anytime after time 2, as this is when the right hand is cleared. This means that it only has to be airborne for 2 beats (and can subsequently be held for the remaining 3 beats). This is the minimum-airtime requirement for a normal ‘4’, so we can throw it to the same height as we would normally throw a ‘4’; hence its SS(As) value is ‘4x’. I’ll leave you to check the other SS(As) values.

In conclusion then, the SS(As) row is intended to store values which most closely resemble, or are most informative of the required throw-action.

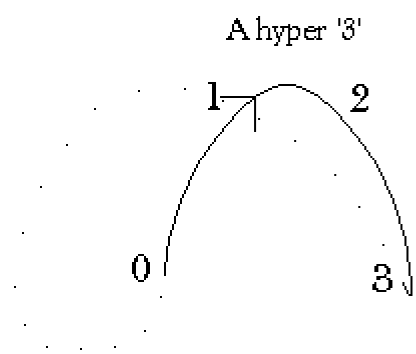
HOLD TIME

(A)

A slightly technical, though important issue, is the question of holdtime. What are the restrictions on hold-time in a 2-handed pattern? I’ll illustrate what’s going on with a picture:



(2 beat holdtime)



(0.1 beat holdtime)

When you throw a ‘3’ (in an alternating pattern), the ball instantly leaves your hand. The reason it is (or will have been) a ‘3’, is because that same ball will next be thrown 3 throws (therefore 3 beats) later. Now those 3 beats include a) the time spent in the air, and b) the time spent in the catching hand.

Let’s suppose you are juggling the 3 ball cascade. If you throw a ‘3’ from (say) your left hand at time 0, and you were feeling lazy, you could wait until the ball was about to land in your right hand, before you throw the ball currently residing in your right hand. At this moment the time is 1 (because it is 1 throw later than time 0). At time 2 you will throw the last ball (from your left hand), and finally, at time 3 you will throw the first ball again (this time from your right). So backtracking, the first ball spent 1 beat in the air, followed by 2 beats of hold time (in your right hand), making up the ‘3’. Now you can reduce the time the ball is held, by throwing the 2nd ball (at time 1) earlier (so there will be less *real* time between time 0 and time 1). You could even throw the 3rd ball (at time 2) before catching the 1st. At time 3 though, we must throw the 1st ball again. We could feasibly (although with some difficulty) slap this ball back to the left hand, so that the hold time was almost zero. To formalise this slightly, we have the simple equation:

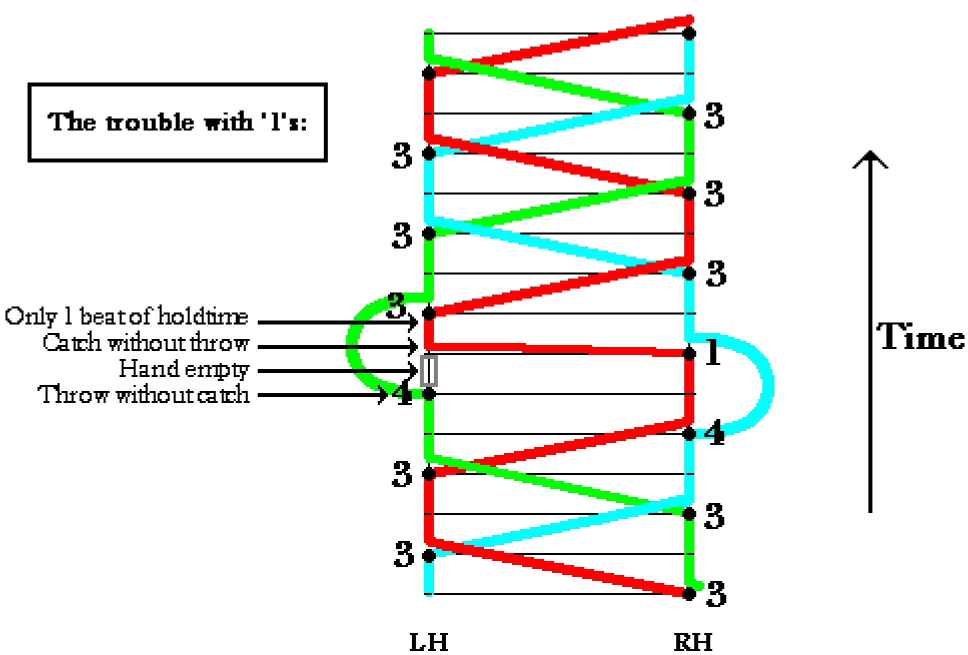
SS value = airtime + subsequent holdtime.

In conclusion then, We have a range of possibilities for hold times: we can choose to hold a ‘3’ (after it has been caught) for anywhere between 0 and 2 beats. In fact, we can hold any SS value ‘2’ or higher, for up to 2 beats. In general, with H hands, after catching a throw of SS value Y, we can hold the ball for up to H or Y beats (whichever is the smallest), before throwing it again, the exact amount of holdtime depending on how high we throw and the rate we juggle at. A bit of thought should convince you of this generalisation.

There are several good reasons for a juggler wanting to maximise their holdtime in a pattern:

1) It allows more time to prepare the throws.
2) It reduces the airtime – thereby reducing the time when a ball is ‘out of control’.
3) It reduces the number of balls in the air – so that there is less to concentrate on.
4) It means the pattern can be juggled at a slower rate.
5) The downward impulse from catching a ball in a hand that has just thrown, helps you to reverse the upward motion of your arm more quickly, thereby increasing the ‘smoothness’ of the pattern.

To illustrate; if you juggle a 3 ball cascade with low holdtime (say 0.5 beats), it is almost as difficult (as fast and as high), as a 5 ball cascade juggled with maximum (2 beats) holdtime; there will be nearly the same number of balls in the air, in both patterns (2.5 and 3 respectively). Many books suggest (or even state) that you should use 1 beat holdtimes. I suggest that (close to) 2 beat holdtimes will make your juggling considerably easier. There are however, 2 main disadvantages with using near-2 beat holdtimes. Firstly, because there is so little time between a throw and its following catch, ‘body moves’ (eg behind-the-back throws) are very difficult. The other problem, is throwing ‘1’s; you can’t possibly have 2 beats of holdtime after catching the ‘1’, because there simply isn’t enough time available before the ball needs to be thrown again, as can be seen from the following ‘ladder diagram’ of 3 3 3 3 4 4 1 3 3.... This means that there is an awkward kludge in the rhythm. (For an introduction to ladder diagrams, see page 28.)



You can see the problems. First, the LH ‘4’ is thrown without an incoming ball. Then there is a whole timebeat of wasted handtime (when the hand is empty). Then there is a catch (of the ‘1’) without an immediately-preceding throw (from this hand). Finally, there is the reduced (by half) amount of time for the ball to be prepared for it’s next journey (on a ‘3’); ie just 1 beat’s worth.

If you were to use 1 beat holdtimes (exclusively) instead of 2, these problems would not go away; on the contrary, it would mean that the hands are empty for one beat *after every single throw*, thereby making the pattern *even more* difficult. The advantage though, is that the rhythm of the pattern wouldn’t change when a ‘1’ occurs.

HEIGHT AND SPEED

(M)

When you're juggling a pattern, how do you know how high to throw? For most jugglers, it is simply a matter of experience. Of course, you can work it out, based on how many throws you can accurately make per second, and how much hold time you use. So, suppose you are juggling a standard B-ball pattern (cascade or fountain), at the rate of R throws per second, with constant holdtime H (lying between 0 and 2). Now there will be B – H balls in the air, so we must make B – H throws in the (total) time it takes for a ball to go up and come down again. Each throw takes us 1/R seconds (before we can throw again), so each ball will have to be in the air for (B – H) /R seconds to enable us to make the required number of intervening throws.

From the mathematician's introduction, we saw that $T = 2\sqrt{(s/5)}$ where T is the time spent in the air, and s is the maximum height reached. If we had used the more accurate figure of 9.81 m/s^2 for gravity, then this formula would read $T = 2\sqrt{(s/4.905)}$ Making s the subject, we get $s = (4.905/4)\ T^2$. Substituting the (B – H) /R in for T, gives the height to which we need to throw: $s = 1.226((B – H)\ /R)^2$.

Assuming we juggle with maximum holdtime (so H=2), here are the heights we would have to juggle at (in metres), for various throw-rates (again: R = number of throws per second, B = number of balls):

R: \B:	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
1	***1.23***	4.90	11.04	19.62	30.66	44.14	60.07	78.48	99.33
2	0.31	***1.23***	2.76	4.90	7.66	11.04	15.02	19.62	24.83
3	0.14	0.54	***1.23***	2.18	3.41	4.90	6.67	8.72	11.03
4	0.08	0.31	0.69	***1.23***	1.92	2.76	3.75	4.90	6.21
5	0.05	0.20	0.44	0.78	***1.23***	1.77	2.40	3.14	3.97
6	0.03	0.14	0.31	0.54	0.85	***1.23***	1.67	2.18	2.76
7	0.03	0.10	0.23	0.40	0.63	0.90	***1.23***	1.60	2.03
8	0.02	0.08	0.17	0.31	0.48	0.69	0.94	***1.23***	1.55

Having noticed that some 5 ball jugglers do their 5 ball cascade about head height, we can use this table in reverse, to deduce that they are throwing about 4 balls per second (as there is about 69cm between the stomach and the top of the head).

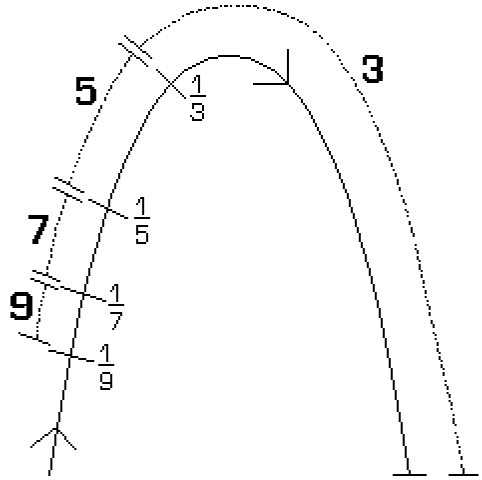
You might also think, from looking at this table, that if such a juggler had designs on an 11 ball cascade, they would probably be wise to increase their throw rate a bit, rather than trying to juggle at over 6 metres high. Note that if you want to juggle at 1.23 m, you have to increase your throw rate by 1 throw/sec for every extra ball in your pattern.

We can also use this table to work out the approximate height ratios required for juggling SSs containing several values. For example, when you want to throw a '7' out of a 5 ball cascade, how much higher does the '7' have to be, than the '5's? Well, divide the height required for a 7 ball throw, by that for a '5' (at any throw rate), and the answer is nearly 3 – so throw the '7' almost 3 times as high. In general, you can use the formula,

$$\text{Ratio} = (\ (V_1 – H) \div (V_2 – H)\ )^2$$

– This tells you how many times higher to throw the larger SS value.
(where V_1 is this higher SS value, and V_2 is the lower; H is the holdtime you are using)

Another way to look at the speeds needed to get different sizes of cascade running, is with the following picture:



This shows the required time-windows for the second throw, in order to juggle the 3, 5, 7 and 9 ball cascades at a sustainable rate. With the 3 ball cascade, you can throw the second ball anytime after the first ball has gone 1/3 of the way (in time) along its trajectory (but before it lands). With the 5 ball cascade, the window is much smaller – between 1/5 and 1/3. And so on, the windows getting ever smaller with increasing balls.

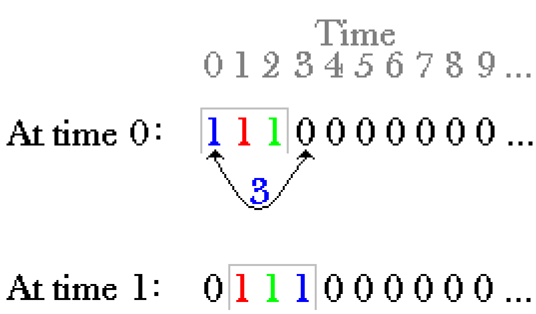
STATE NOTATION

(A)

This was discovered even later than SS notation, but is closely related. State notation describes not a whole pattern, but rather an 'instant in time', showing the requirements (present and future) for keeping the pattern going. It can be used for generating patterns or calculating transitions between patterns.

A 'state', like SS, has the form of a sequence of digits (usually '1's and '0's, alternatively 'X's and '–'s). The SS '3' remains in the same state for as long as it is juggled – the state being 11100000000.... So how can we 'read' this? Well, the digit on the far left of a state always represents the immediate requirement. In this case, this digit is a '1'. A '1' can be thought of, as a ball that needs to be thrown. So when we are in a state starting with a '1', as we are in this case, we have to throw a ball immediately. Now notice that the second digit is also a '1'. This means that after we have thrown the first ball, we will have to throw another ball (at the next available opportunity). There is also another ball to be thrown just after this second ball. However, as things stand, there are no balls to be thrown at any time after this third ball; the further to the right we look along the state-sequence (or 'state-string'), the further into the future we are looking.

So what happens when we throw the first ball as a '3'? Well the '1' simply moves 3 places along the state-string:

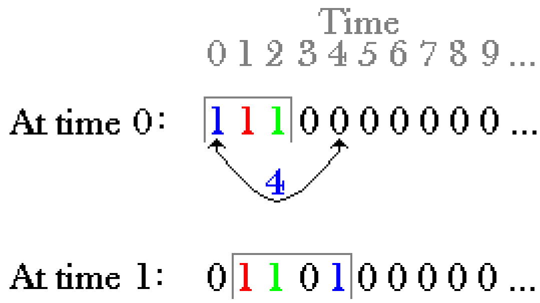


– When we throw the blue ball (as a '3'), we are promising to throw it again 3 throws later – ie at time 3. So the '1', which represents the blue ball, moves to time 3.

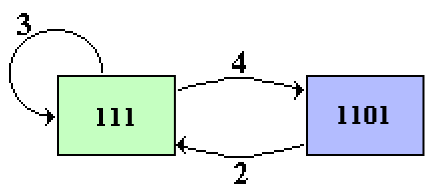
Time then moves on, so we find ourselves at time 1 – we now need to throw the red ball....

To make states easier to read, we write them starting from the present, and ending with the final ‘1’, so the states at both times 0 and 1 (above), are written ‘111’. We can put this process into the more concise form: 111 (3)→ 111.

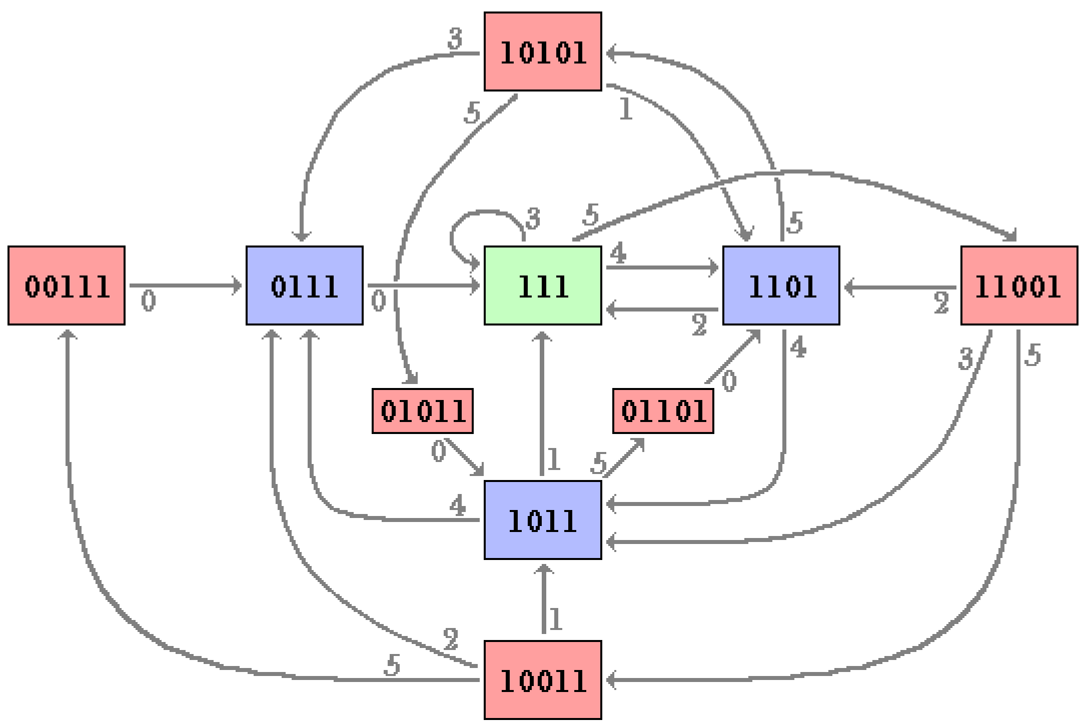
So what happens if we were to throw a ‘4’?



– In other words, 111 (4)→ 1101. It is also true, that 1101 (2)→ 111, and using this information, we can draw part of the 3 ball state-map:



This diagram shows that we could do alternate ‘4’s and ‘2’s indefinitely, and so ‘4 2’ is a valid SS sequence. The full 3 ball state-map, (maximum SS value 5) is:



Similar maps can be drawn for any number of balls, although with more balls, the maps get very complex.

Now consider what would happen if we were in state 111, and we threw a ‘2’? Well this ball must next be thrown 2 throws later, but there is already a ball that must be thrown at this time – so now there are 2 balls to be thrown then. This is what happens during multiplex patterns. I will say no more on multiplexes here, but for the maths people, this discussion will be continued in the next-but-one section (on excitation levels).

For synch patterns, states can be written in the form (A,B)(C,D)(E,F).... The synch 4 ball fountain on (4,4) for example, remains in the state (1,1)(1,1). The 4 ball shower on (6x,2x), also uses just one state: (1,1)(0,1)(0,1).

PRIME & COMPOSITE PATTERNS

(M)

Some SSs, such as 44404413 can be split up into shorter patterns, which can be played individually as many times as you want, before continuing with the rest of the pattern. For example, 44404413 can be split into 4440, 441, and 3 (each of these is a valid SS in its own right). Thus, we call 44404413 a ‘composite’ SS. 4440, 441, and 3 are all ‘prime’ SSs, as they cannot be split up in this way. But being mathematicians, we want a nice rigorous definition of these concepts, so here it is:

A prime SS, is one which passes through no state more than once per cycle.

SSs which are not prime, are called composite. Illustrating this definition, 44404413 is not prime, because it passes through the state 111 three times (at the moments in-between the 3 sub-patterns). Similarly, 74135 is also composite (splitting into 741 and 35), as it passes through 11101 twice.

Some patterns look like they’re composite, but are not, such as 315 (aka 531). You might think that this can be split up into 3 and 15, but if you try to go into either of these, from 315, you will have problems – for example if you try 3 1 5 3 1 5 3 3 3 3 ***3***..., or 3 1 5 3 1 5 1 ***5*** 1 5 ..., the two underlined throws ‘clash’, as these balls both have to be next thrown at the same time, indicated in bold italic.

There are mathematical games to be played with this definition, such as, ‘find the longest prime VSS with B balls, and maximum height M’ (see puzzle 4, page 40).

The potential length of prime SSs is clearly limited by the number of different states, which (using a bit of combinatorics) is M! ÷ (B! × (M – B)!), where ‘!’ means ‘factorial’: M! = M × (M–1) × (M–2) × ... × 1 (and 0! = 1). There are thus 5! ÷ (3!×2!) = 10 possible states with 3 balls and max height 5, as shown on page 25.

EXCITATION LEVELS

Still on the topic of state notation, the concept of excitation levels serves to provide a way of measuring how ‘energetic’ or ‘excited’ a pattern is, at each moment in time. The ‘basic’ state (ie 111...) is also called the ‘ground’ state. This state has excitation level 0. Let’s suppose we are juggling 5 balls in the ground state. If we throw a ‘6’, we will ‘go up’ 1 level (to 111101). If we next throw a ‘4’, we go back down again. If we were to throw a ‘7’ (from the ground state), we would go up 2 levels (to 1111001). The general rule, is that when you make a throw which has SS value V, and you are juggling B balls, the excitation level goes up by V – B (this may be negative, in which case the level decreases).

To calculate the level (L), directly from the state (S), we can use the formula:

$$L(S) = \sum(i \times S_i) - \sum(1, 2, ..., B),$$

- where $\sum$ is ‘the sum of’, S_i is the ith digit of the state-string (ordered from the left), and B is the number of balls. (A similar formula can be devised for synchronous SSs – details left as an exercise.)

As an example, let’s take the ground state for 3 balls: 111. Here, $S_1 = 1$, $S_2 = 1$, $S_3 = 1$, and $S_i = 0$ for all other i. So L(111) = (1×1) + (2×1) + (3×1) – (1+2+3) = 6 – 6 = 0.

Another example: L(0111) = (1×0) + (2×1) + (3×1) + (4×1) – (1+2+3) = 9 – 6 = 3.

This formula can even be used for the multiplex states which I briefly alluded to a minute ago. In multiplexes, you throw 2 or more balls at the same time. Correspondingly, the state-string contains a digit larger than 1. For example, when you juggle [5,4] 2 4, you pass throughout the states 1211 and 2111. When you reach 2111, you have to throw 2 balls at once (a ‘5’ and a ‘4’), after which, you end up in 11111.

Calculating the level of the state just before the multiplex throw, L(2111) = (1×2) + (2×1) + (3×1) + (4×1) – (1+2+3+4+5) = 11 – 15 = - 4. So state 2111 is actually *below* ground state. If you think about it, this is quite reasonable, as there are less balls in the air (in this state), than during a normal 5 ball cascade.

Finally, we can define the excitation level of a SS sequence (we like defining things!), to be the average of the levels of the states it passes through, so, eg L(53) = (L(11101) + L(1111)) ÷ 2 = (1 + 0) ÷ 2 = ½.

DIFFICULTY

If you are thinking of learning to juggle a particular SS sequence, it might be fairly useful to have some idea of how difficult the pattern is, in advance. To this end, below is the simplest, credible SS difficulty formula I have come across. It gives difficulty ratings for SS sequences which seem reasonably consistent with my own experience for most patterns:

$$D_1 = 2 + \sqrt{(\sum (V_i - 2)^2) \div P}$$

√ means 'take the square root'; ∑ means 'sum over all the values V_i in the SS sequence'; P is the period of the SS. As a quick example, lets take '4 4 1': $D_1 = 2 + \sqrt{(\ ((4\text{-}2)^2 + (4\text{-}2)^2 + (1\text{-}2)^2)\div 3\ )} = 2 + \sqrt{(\ (4+4+1)\div 3\ )} = 2 + \sqrt{3} = 2 + 1.7 = 3.7$. The 2 is added to make the score indicate the number of balls you would have to juggle (in a standard cascade or fountain) in order to encounter similar actual juggling difficulty.
A useful amendment to this formula, is to ignore '0's (rather than adding on $(0 – 2)^2 = 4$ for them).

Another difficulty measure, D_2, involves the (average) excitation level of a SS, which can be halved (say), and added to the number of balls in the pattern to give similar scores to D_1. Using these systems, I've worked out the difficulty rating of several SS sequences (to 1 decimal place). See what you think:

Siteswap	No. balls	D_1	D_2
0	0	2.0	0.0
2	2	2.0	2.0
3 1 2	2	2.8	2.2
1	1	3.0	1.0
3 1	2	3.0	2.3
3	3	3.0	3.0
4 2 3	3	3.3	3.2
4 1 1	2	3.4	2.5
(4x,2x]	3	3.4	n/a
4 4 1	3	3.7	3.5
5 0 1	2	3.8	3.2
5 3 1	3	3.9	3.7
4	4	4.0	4.0

Siteswap	No. balls	D_1	D_2
5 1	3	4.2	3.7
5 3	4	4.2	4.3
5 5 5 5 0	4	4.7	5.0
6 1 5	4	4.9	5.2
7 1 1	3	5.0	5.0
5	5	5.0	5.0
7 4 1	4	5.3	5.5
[6x,4]*	5	5.3	n/a
8 0 1	3	5.5	6.2
9 7 5 3 1	5	6.1	7.0
9 1	5	7.0	9.0
11 0 1	4	7.2	8.7
999999990	8	8.6	10.0

Of course, throwing or catching using different positions or 'types' will increase the difficulty of any pattern somewhat. To incorporate these aspects, the formula:

$$X = D \times (1 + (\sum R \div 10P))$$

seems to work reasonably well, where X is the difficulty of the pattern, D is the SS difficulty (eg D_1 or D_2), ΣR is the sum of the difficulty scores for throw/catch positions/types, and P is the period. As an example, for the 3 ball Mills Mess (see appendix), the THR(Pos) row scores 12, the CAT(Pos) 6, so ∑R = 18. 10P = 60, so ∑R ÷ 10P (= 18 ÷ 60) = 0.3, and so the 3 ball Mills Mess has difficulty rating: 3 × (1+ 0.3) = 3 × 1.3 = 3.9.

Note that the practice-time it takes, to be able to do patterns of a certain difficulty score, increases (highly) exponentially. Finally, I'll state the obvious: the difficulty of a pattern is quite subjective.

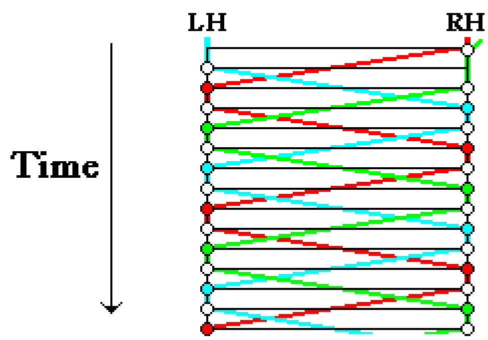
4) DESIGNING SITESWAPS

There are many methods for generating SSs. The one you use depends partly on taste, and partly on the type and complexity of the kind of pattern you're looking for.

1) LADDER DIAGRAMS

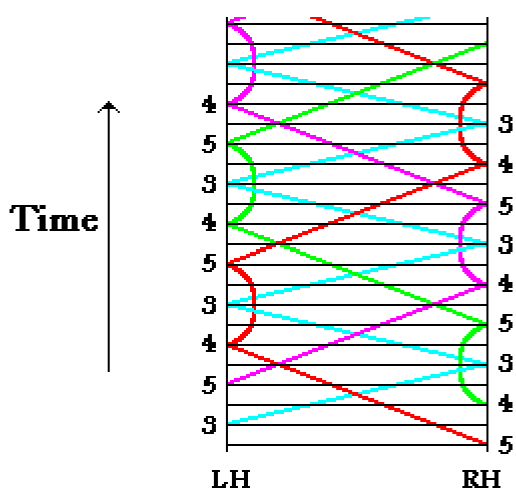
(A)

Devised slightly before SS, this is one of the more visual notations. The 'rungs' of the 'ladder' represent points in time, and the 2 sides represent the 2 hands. Lines drawn on the ladder represent the paths of the balls. Ladder notation is good for sorting out issues of holdtime and airtime (as on page 22), as they can be explicitly shown on the ladder. Conventionally, ladder diagrams are written with the ladder vertical, and time running down, although you can draw them any way you like. Personally, I prefer time to run upwards, as this means that the movement of the ball-lines more closely corresponds to the direction you throw the ball in. Anyway, here is a ladder diagram of the SS '3' (time running downwards):



Key: An empty circle is a throw (the emptying of a hand). A filled circle is a catch. Each of the 3 balls has its path drawn with a different colour.

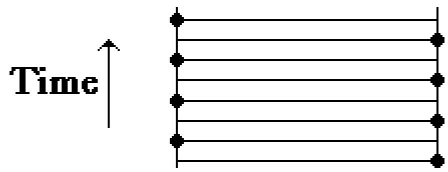
Notice how each hand is alternatively full for a beat, then empty for a beat – the pattern represented by this ladder diagram has 1 beat holdtimes. This makes the diagram fairly simple to draw and understand. Anyhow, if you are not bothered about holdtimes, you may as well simplify the diagram further, by ignoring holdtimes altogether (equivalently, using 0 beat holdtimes). Here is 534 in this format (time running upwards):



It doesn't really matter whether you draw even-valued throws inside the ladder (as above), or outside (as on page 22).

Ladders can be used to design SSs, as follows:

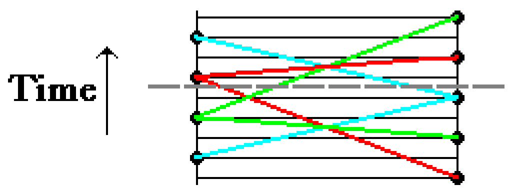
1) Draw a ladder (highlighting the throw-points), and decide which direction time goes, eg:



2) Now join up each throw-point to one in the future, bearing in mind that the average number of rungs between joined-up throw points, will be the number of balls in the pattern. Make sure every throw-point has exactly 1 line going in (except the first few), and 1 coming out.

3) The SS values can be 'read off' the diagram, by counting how many rungs up the ladder each ball is next thrown.

To check how many balls are in a finished ladder diagram, take a slice across the ladder, and count how many ball-lines pass through it. So if we ended up with:

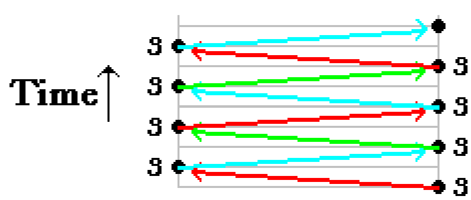


We can see that *3* balls are used, as exactly 3 ball-lines (red, blue, and green) cross the dotted grey line. Of course, using different colours for each ball makes it obvious that there are 3 balls, but it isn't always clear if all the ball-lines are the same colour.

There are disadvantages of ladder notation. Firstly, if you are trying to design a repeatable pattern, you need to either carry on forever, drawing the sequence of throws over and over; or work out whether the sequence can loop back to the start (by checking that the sequence of balls which are available to be thrown, match those at the start of the ladder). Secondly, there are problems of communicating ladder diagrams to other jugglers, which is cumbersome and time-consuming; if you design a pattern on a ladder, it would probably be useful to calculate its siteswap description, before attempting to convey it to an enquirer.

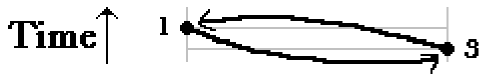
2) CAUSAL DIAGRAMS

These are very similar to ladder diagrams. The angle to be taken when thinking about causal diagrams is, "If I throw a ball, what problem does this cause?". To illustrate what I mean, let's look at the SS '3' in this notation:



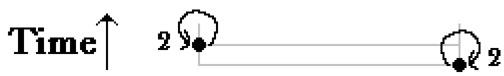
The red arrow says, “Throwing (the red ball as) a ‘3’ from my right hand, will cause me to have to throw the (blue) ball out of my left hand at the next available opportunity (so that it is free to catch the red ball)”.

Now let’s look at 2 throws of ‘3 1’:



Again, the ‘3’ from the RH causes me to have to empty the LH at the next opportunity. However, the arrow representing the LH ‘1’ (pointing backwards in time) says, “For my LH to be able to throw the ‘1’, I must *already* have emptied my RH (on it’s last throw)”.

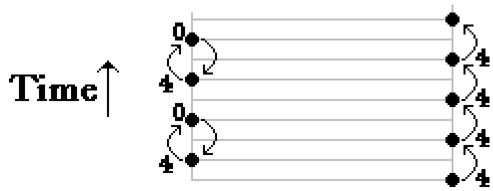
Now what about the SS ‘2’? In this ‘pattern’, you have no ‘juggling problem’ – you don’t have to empty a hand to allow you to catch an incoming ball. Here is the causal diagram of two ‘2’s:



The looping arrows mean, “If I throw this ball, then I will have to throw this (the same) ball”. This may seem a bit tautological (or obvious), but basically, it indicates that we don’t have to throw anything if we don’t want to.

As you may have worked out by now, the ‘causal-line’ of a SS value N points to the throw-point which is N–H rungs further up (where H is the number of hands in the pattern), so ‘3’s point to the throw 1 rung up, ‘4’s point up 2 rungs, ‘5’s point up 3 rungs and so on (with 2 hands). In fact, for solo patterns involving only SS values of ‘3’ or higher, causal diagrams look almost identical to ladder diagrams using 2 beat holdtimes.

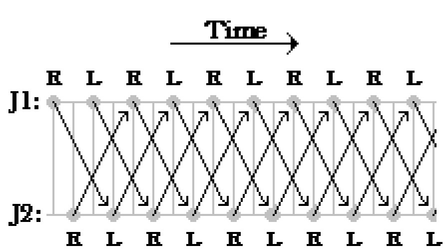
Going the other way, ‘2’s point to themselves, ‘1’s point backwards 1 rung, and ‘0’s? According to the rule, these should point backwards 2 rungs. In fact, this does make real sense. Let’s use 4440 to understand why:



The arrows from a ‘0’ back to a ‘4’ can be read, “In order for me to do a ‘0’ (ie have an empty hand), I must have emptied this hand on it’s last throw”.

There is also a rule for calculating the number of balls in a causal diagram, similar to that of ladder diagrams: take a slice across the ladder, count the number of ball-lines which intersect the slice (subtracting 1 for each line which goes backwards in time), then add on the number of hands in the pattern.

Because causal diagrams effectively ignore held balls, they are quite a bit easier to read than ladder diagrams, and are especially good for understanding *passing* patterns. Here is ‘7 ball ultimates’, as an example:



- Effectively, causal diagrams only show the balls which need to be kept in the air, so only 3 ‘ball-points’ per ‘time-slice’ need to be drawn here, compared to the 7 which the corresponding ladder diagram needs (see page 47).

Designing SSs using causal diagrams is very similar to using ladder diagrams – except now you have the option of lines travelling backward 1 or 2 beats, in addition to the forward-travelling lines.

Finally, I'll leave you with a suggestion to draw the causal diagram of 501, and witness the mysterious 'ever-backward-time-travelling problem'. Spooky.

3) PERMUTATIONS

This is one of the simplest methods, but the resulting (vanilla) SS sequences are quite 'random' – it is hard to 'choose' the values. Here is the algorithm:

a) Choose the period (=P) of pattern you want. Let's suppose it's 5.

b) Choose a permutation (ordering) of the integers 1 to P, eg: 4 1 3 5 2

c) Subtract 1 to P respectively from these: -1 -2 -3 -4 -5
= 3 -1 0 1 -3

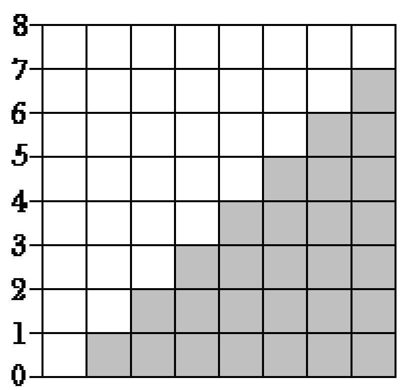
d) Add the number of balls you want to use, eg 4: +4 +4 +4 +4 +4
= 7 3 4 5 1

This method will always produce a 'valid' SS, although sometimes some of the values will be negative. Provided all the resulting values are 0 or above however, the sequence can be juggled, as is the case here with 73451.

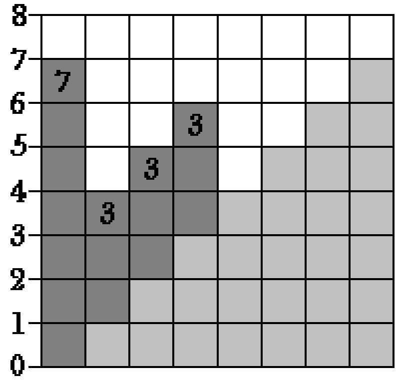
Note that this method doesn't allow synch or multiplex SSs to be generated. In fact, even some VSSs cannot (eg 801).

4) STAIRS

This is a nice visual method. You don't have to decide (in advance) how many balls you want, or even the period of the pattern. First, we need a 'staircase':

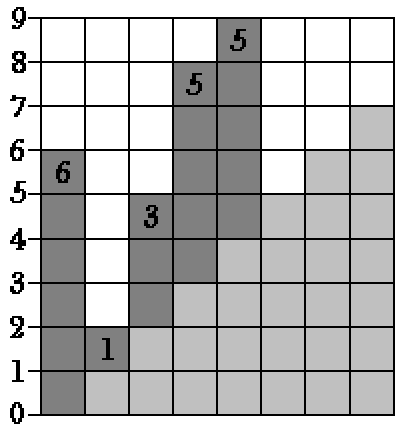


Then we put columns of blocks (representing the throws) on the steps. The blocks should be the same height as the steps. We'll jump straight in and have a look at a couple of patterns in this framework. First, 7333; put a stack of 7 blocks in the first column (on the 'floor'), 3 blocks in the second (on the first step), 3 in the third, and 3 in the fourth, like so:

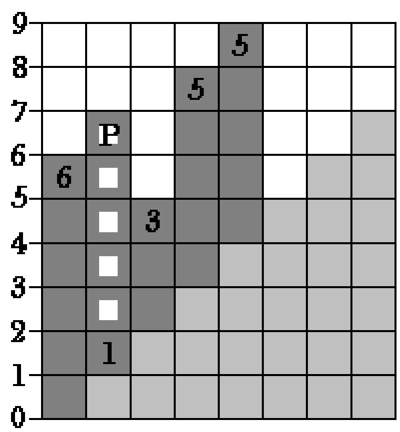


Notice that the tops of the columns are all on different, but consecutive levels (4, 5, 6 and 7). This is how we know that the pattern 'works' – if two columns had their tops on the same level, then the throws (represented by the columns) would clash. We can also immediately read off how many balls are in the pattern by looking at the top of the lowest-rising column, which reaches level 4, so it uses 4 balls.

Here's another; 61355:



Now when the column-tops are not on consecutive levels, we add multiples of the period (in this case 5) onto the lowest, and recheck:



The column-tops now *are* on consecutive levels, so the SS is valid (note that the above picture also shows the 5 ball SS 66355 to be valid). We can again work out the number of balls in it, using the height of the lowest top: in this case it rises to level 5, but as we had to add 1 multiple of the period onto a column, we subtract 1 off the 5, to get 4. In general, however many stacks of the period are added to columns for checking purposes, this number should be subtracted from the lowest column-top height, to get the number of balls in the pattern. Anyway, if you want to design a pattern using this method, there are 2 approaches:

1) Start with a known SS, and modify it, by moving blocks from one column to another.
2) Build the columns from scratch. Remember that a B ball, period P pattern, will use B×P blocks.

The best thing about this method, is that you can use Lego bricks (remember them?) for the stairs and blocks, making it easy to alter the height of the columns. In fact, this method is so simple, even non-jugglers can invent complex vanilla SSs with it!

5) EVENTS

When I discovered this method, I had no idea that it would turn out to establish a link between SSs, and every other process in the known universe. Expanding this a little, think back to the brief appearances the letters ‘a’, ‘b’, and ‘c’ made, to represent balls in a 3 ball pattern. Remember how we could write down the sequence in which the balls were thrown, eg a b c c a b b c a a ... (for the SS 441). Now what stops us viewing *events* as *thrown-balls*, and translating any sequence of events into a SS?

Take your tropical fish for example. To get them to help you design a SS, mark a line down the middle of the fish-tank. Whenever your upside-down catfish swims past the line, write down ‘U’. When the Chinese fighting fish does it, write down C. Angel fish: A. Watch them for several minutes to produce a sequence such as U, C, A, A, U, C, A, C, U, U, ... and convert this into a SS, by counting how many events later, each fish swam past the middle. So the first ‘U’ produces a ‘4’ (as the next ‘U’ occurs, 4 events later), the ‘C’ produces a ‘4’, the ‘A’ a ‘1’ etc, giving ‘4 4 1 3 4 2 ....’ If you want a repeating pattern of period P, just use the first P observations, and say that it loops, so if you want a 3 ball, period 5 SS, use 3 fish, and record the first 5 results – eg, if you got U C C A A, you convert to get 51414. Notice however that with this method, you can never generate ‘0’s or any values greater than the period.

6) PLATFORMS

(M)

Platforms are probably the most versatile SS design method of all. We have already seen the concept behind them, in the context of synch and multiplex patterns in the last chapter. In fact, you can create the most hideously complex synch, multiplexing, passing SSs imaginable using platforms. However, VSSs are the simplest to design, so we’ll start there.

Suppose we want a period 4 pattern. We draw 4 platforms next to each other, like so: _ _ _ _ .

Now suppose, we want the highest throw in the pattern to be a ‘6’. We may as well start with this throw, so write ‘6’ on the first platform: 6 _ _ _ . Now, so that we don’t end up having to throw more than 1 ball at a time, we need to keep track of when this ‘6’ will next be thrown. Well, if we count along 6 places (cycling back to the start when necessary), we find that the ‘6’ will next be thrown here (where the brackets are): 6 _ (_) _ . So all we have to do, is make sure that no other balls have to be thrown at this time. Let’s continue.

Now we have to choose our second throw. Notice that we can’t have a ‘1’, because this will violate the principle just mentioned – it will have to be thrown again at the same time as the ‘6’. So let’s have a ‘3’. Again we bracket the place where it will next be thrown. So we get: (6) 3 (_) _ .

For the third throw, let’s have another ‘3’. After bracketing, we have: (6) (3) (3) _ . We are not left with many options for the final throw – it could be a ‘0’ (an empty hand), or a ‘4’ (an ‘8’, ‘12’, ‘16’ etc). We’ll have a ‘4’; and so end up with: (6) (3) (3) (4). This is guaranteed to be a valid SS, and indeed it’s quite a satisfying 4 ball pattern (the average of 6, 3, 3, 4 being 4). Still with me? Then let’s go on to multiplex patterns.

With multiplex SSs, as well as the period we need to decide how many balls are to be thrown at each time. Suppose we want a period 5 pattern, with one duplex (2 ball multiplex throw), which we may as well have as our first throw. Drawing the platforms, we have:

[_ _] _ _ _ _

I should point out that using brackets to indicate 'used-up' throw-sites should not be confused with the square brackets indicating a multiplex throw above – this is just an unfortunate clash of notation. It is probably better to put dots underneath 'used-up' throw sites, rather than to bracket them (I'm unable to use dots here for technical reasons).

Anyway, to avoid having to catch 2 balls with the same hand within 2 beats of time, we need to make 1 of the multiplexed balls be *held*, prior to being multiplexed. In other words, we need a '2' on the last-but-one platform. Now bracket (or dot) one of the multiplexed platforms (where the '2' is next thrown):

```
[ (_) _ ] _ _ 2 _
```

(alternatively '[_ (_)] _ _ 2 _' ; if you're not bothered about keeping track of balls, it doesn't matter which)

We can now fill in the rest of the platforms. Make sure the multiplex throw is practical though – throwing the duplex as [7,3] for example, is an undesirable task. [6,5] is much easier – this gives:

```
                                         [(6)(5)] (_)  _   2   _
And to finish the example, we could have:
                                         [(6)(5)] (3) (6) (2) (3)     - aka [6,5]3623.
```

Next, as alluded to earlier, here is how to design synchronous symmetric SSs efficiently. First, recall that these patterns have the form (in ES): (A,B)(C,D)...(BA)(DC).... Now when designing such patterns, we don't want to have to draw all of the platforms, when half of them will contain the mirror image throws of the other half – so we just draw the first half of them. Suppose we want a pattern of the form (A,B)(C,D)* – ie (A,B)(C,D)(B,A)(D,C):

```
A  .  C  .*
B  .  D  .
```

Let's suppose we want A to be a '6x'. Where would this ball be thrown next? Well if we weren't intending to repeat the pattern on the other side, the answer would be 'D'. As we are, the answer is C. We calculate this as follows: count along in the usual way, but each time we have to loop back to the start, 'switch' to the other hand. If the throw is crossing then we switch hands one final time. So in our example, if A = 6x, to find where it is next thrown, we go: '2 to C, 4 to A, switch to B (because we've just looped), 6 to D, and cross (because of the 'x') to C. So we bracket the C. Now for B (= 4x say), we go: 2 to D, 4 to B, switch to A, cross to B – so bracket B. We could end up with (for example):

```
(6x).   (6) .*
(4x).  (4x).        -aka (6x,4x)(6,4x)*
```

Finally, let's design a weird and wonderful pattern that no-one would ever want to juggle, just to illustrate the power of this method (skip this bit if you want). I've put in the useful (pre-multiplex) '2's already:

Time:	0	1	2	3	4	5	6	7
R:	[(_) _]	.	2	.	[(_) _]	.	2	.
L:	_	.	.	_	.	.	_	.

- we could end up with:

R:	[(6x)(4)]	.	(2)	.	[(4x)(4)]	.	(2)	.
L:	(2x)	.	.	(3)	.	.	(5s)	.

It's a good idea to design the multiplex throws first, so that you have some choice in how high to throw them.

Again, the values on the platforms are the SS(Real) values, but it would be useful to calculate the SS(As) values, to indicate how to juggle the pattern most efficiently (with minimum airtime).

7) STATES

In the section on state notation, we saw how SSs pass through different states. In order to create a valid (repeating) SS using states, we just have to make sure that we return to a previous state. For example, with 4 balls, we could start in state 11101, and go: 11101 (7)→ 1101001 (4)→ 101101 (1)→ 11101. This state is the same as the first, and so we could repeat this sequence of throws (ie 741) over and over.

Synch patterns can also be made using states, eg:

(1,1)(0,1)(1,0) , (6x,2x)→ (1,1)(1,0)(0,1) , (2x,8x)→ (1,1)(0,1)(0,0)(1,0) , (4x,2x)→ (1,1)(0,1)(1,0).

- aka (6x,2x)(2x,8x)(4x,2x). Alternatively, (8x,2x)(2x,4x)(2x,6x), to start with the 8x.

In fact any kind of SS can be made using states. In order to facilitate the making of VSSs by this method, the 3 ball state-map given earlier can be used. Serving the same purpose with 4 or 5 balls, are the state-charts below. These allow the generation of all (4 and 5 ball) SSs involving (only) heights between 1 and 8. For concision I've assigned an integer label to each state. Each entry in the table is the label of the new state entered, after a certain throw has been made from a certain state.

4 Ball State Transitions

Label	(State) Throw:	1	2	3	4	5	6	7	8
1	(1111)	.	.	.	1	2	5	11	21
2	(11101)	.	.	1	.	3	6	12	22
3	(11011)	.	1	.	.	4	8	15	26
4	(10111)	1	.	.	.	.	.	.	.
5	(111001)	.	.	2	3	.	7	13	23
6	(110101)	.	2	.	4	.	9	16	27
7	(110011)	.	3	4	.	.	10	18	30
8	(101101)	2	.	.	.	.	.	.	.
9	(101011)	3	.	.	.	.	.	.	.
10	(100111)	4	.	.	.	.	.	.	.
11	(1110001)	.	.	5	6	7	.	14	24
12	(1101001)	.	5	.	8	9	.	17	28
13	(1100101)	.	6	8	.	10	.	19	31
14	(1100011)	.	7	9	10	.	.	20	33
15	(1011001)	5	.	.	.	.	.	.	.
16	(1010101)	6	.	.	.	.	.	.	.
17	(1010011)	7	.	.	.	.	.	.	.
18	(1001101)	8	.	.	.	.	.	.	.
19	(1001011)	9	.	.	.	.	.	.	.
20	(1000111)	10	.	.	.	.	.	.	.
21	(11100001)	.	.	11	12	13	14	.	25
22	(11010001)	.	11	.	15	16	17	.	29
23	(11001001)	.	12	15	.	18	19	.	32
24	(11000101)	.	13	16	18	.	20	.	34
25	(11000011)	.	14	17	19	20	.	.	35
26	(10110001)	11	.	.	.	.	.	.	.
27	(10101001)	12	.	.	.	.	.	.	.
28	(10100101)	13	.	.	.	.	.	.	.
29	(10100011)	14	.	.	.	.	.	.	.
30	(10011001)	15	.	.	.	.	.	.	.
31	(10010101)	16	.	.	.	.	.	.	.
32	(10010011)	17	.	.	.	.	.	.	.
33	(10001101)	18	.	.	.	.	.	.	.
34	(10001011)	19	.	.	.	.	.	.	.
35	(10000111)	20	.	.	.	.	.	.	.

5 Ball State Transitions

Label	(State) Throw	1	2	3	4	5	6	7	8
1	(11111)	.	.	.	.	1	2	6	16
2	(111101) .	.	.	1	.	3	7	17	
3	(111011) .	.	1	.	.	4	9	20	
4	(110111) .	1	.	.	.	5	12	26	
5	(101111) 1	.	.	.	.	.	.	.	
6	(1111001)	.	.	.	2	3	.	8	18
7	(1110101)	.	.	2	.	4	.	10	21
8	(1110011)	.	.	3	4	.	.	11	23
9	(1101101)	.	2	.	.	5	.	13	27
10	(1101011)	.	3	.	5	.	.	14	29
11	(1100111)	.	4	5	.	.	.	15	32
12	(1011101)	2	.	.	.	.	.	.	.
13	(1011011)	3	.	.	.	.	.	.	.
14	(1010111)	4	.	.	.	.	.	.	.
15	(1001111)	5	.	.	.	.	.	.	.
16	(11110001)	.	.	.	6	7	8	.	19
17	(11101001)	.	.	6	.	9	10	.	22
18	(11100101)	.	.	7	9	.	11	.	24
19	(11100011)	.	.	8	10	11	.	.	25
20	(11011001)	.	6	.	.	12	13	.	28
21	(11010101)	.	7	.	12	.	14	.	30
22	(11010011)	.	8	.	13	14	.	.	31
23	(11001101)	.	9	12	.	.	15	.	33
24	(11001011)	.	10	13	.	15	.	.	34
25	(11000111)	.	11	14	15	.	.	.	35
26	(10111001)	6	.	.	.	.	.	.	.
27	(10110101)	7	.	.	.	.	.	.	.
28	(10110011)	8	.	.	.	.	.	.	.
29	(10101101)	9	.	.	.	.	.	.	.
30	(10101011)	10	.	.	.	.	.	.	.
31	(10100111)	11	.	.	.	.	.	.	.
32	(10011101)	12	.	.	.	.	.	.	.
33	(10011011)	13	.	.	.	.	.	.	.
34	(10010111)	14	.	.	.	.	.	.	.
35	(10001111)	15	.	.	.	.	.	.	.

To use these tables, simply choose the number of balls (eg 4), a start state (eg '7'), and a valid throw (eg 2), and read off the label of the next state reached (in this case 3). Then choose a valid throw from this state, and so on, until you return to the start state. When this happens, you will have created a valid SS sequence. Finally, any SS which passes through state 1 (111…1), also known as the ground state, is called a 'ground SS'.

8) AXIOMS

The axiomatic method serves 3 purposes. Firstly, mathematicians can get immense amounts of pleasure playing games with axioms (which are 'the rules' of the game). Secondly, it serves to illustrate the relationship between different SSs. Thirdly, it can be used to design new SSs. Here I will describe an axiom system for VSS generation. Analogous systems can be formulated for other kinds of SSs.

This axiom set is both 'sound' and 'complete' – that is, it can generate only and all valid VSSs.

SS Axiom Set

A1:	(Basic)	SS(c c c ... c)	(c is any integer (whole number))
A2:	(Siteswap)	SS($\underline{a}$ b c) → SS($\underline{a}$ c+1 b–1)	($\underline{a}$ is a sequence, b & c are integers)
A3:	(Cyclicity)	SS($\underline{a}$ b) → SS(b $\underline{a}$)	($\underline{a}$ sequence, b integer)
A4:	(Periodicity)	SS($\underline{a}$ b $\underline{c}$) → SS($\underline{a}$ b+P $\underline{c}$)	($\underline{a}$, $\underline{c}$ sequences, b integer)

SS(X) means 'X is a valid SS'. '→' means 'logically implies' (ie 'F → G' means 'if F, then G').
Underlined letters represent SS sequences of any length (possibly zero length). P is the period of the SS.

Here are some examples of these axioms in action:

A1	SS(333)	(c = 3)
A2	SS(28) → SS(91)	($\underline{a}$ = '', b = 2, c = 8)
A3	SS(5551) → SS(1555)	($\underline{a}$ = 555, b = 1)
A4	SS(711) → SS(741)	($\underline{a}$ = 7, b = 1, $\underline{c}$ = 1)

In fact, not all of these axioms are needed; we don't need A3 *and* A4 – just 1 of them will do. Note that A3 doesn't actually generate new SSs – it just enables them to be rewritten starting from a different place in the sequence. Also note that A1 is the only axiom with no preconditions, so when designing a SS, we always have to start with an instance of A1. Also, the only requirement for A1, is that c is an integer; in rare cases it may be desirable for c to be negative, although clearly this pattern would be unjugglable. A final note, is that A4 is the only axiom which can generate SSs containing *more* balls. We finish with 2 different 'proofs' that 71 is a valid SS:

Proof without using A4:

1) Using A1: SS(44)
2) Using A2 and line 1: SS(53)
3) Using A3 and line 2: SS(35)
4) Using A2 and line 3: SS(62)
5) Using A3 and line 4: SS(26)
6) Using A2 and line 5: SS(71)

Proof using only A1 and A4:

1) Using A1: SS(11)
2) Using A4 and line 1: SS(31)
3) Using A4 and line 2: SS(51)
4) Using A4 and line 3: SS(71)

SITE SLIDING

(A)

This is a quick way to generate synch patterns from alternating ones, and vice versa. The idea is to 'slide' one hand back or forward 1 beat, to make it throw in (or out of) time with the other hand :

R:	A	C	E	G	...		R:	A'	C'	E'	G'	...
L: ←slide back:	B	D	F	H		to get:	L:	B'	D'	F'	H'	

Each new value Y' is calculated from Y by the following: If Y is even, then Y' = Y. If Y is an odd, right hand throw, then Y' = (Y – 1)x. If Y is an odd, left hand throw, then Y' = (Y + 1)x. Let's do this on 7445:

R:	7		4	...		R:	6x	4
L:		4		5	becomes:	L:	4	6x.

In other words, 7445 becomes (6x,4)(4,6x).

If we slide the left hand *forward*, instead of *back* (equivalently, slide the *right hand back*), we would have:

R:		7		4 ...		R:	8x	4
L:	5		4		becomes:	L:	4x	4.

In general. whichever hand is slid backwards (relative to the other), has its odd values increased by 1; the other hand has its odd values decreased by 1. All of these ex-odd values still cross, indicated by appending a 'x'.

Similar operations can be performed, to change synch SSs into alternating ones.

VANILLA CONVERSION

(M)

We can generalise the site-sliding technique to enable any type of SS to be generated from a vanilla sequence. Suppose for example, that we want to make a 5 ball, period 3, multiplex pattern of the form [a,b]cd. Rather than design it from scratch, we can convert a VSS, as follows:

1) Let P equal the number of throws in the desired pattern (counting each throw within any multiplex or synch throw separately). In our example, there are 4 'throws': 'a', 'b', 'c' and 'd', so P = 4 (even though 'a' and 'b' are thrown together). Also, let B = the number of balls in your desired pattern (=5 in this case).

2) Take a B-ball, period P, VSS. (If you are after a multiplex, and you don't want to have any frantic catching, then make sure this VSS has a '2' or a '3' in it.) Let's use 7 5 3 5.

3) Write down the platform-structure of your desired pattern (including the multiplex/synch brackets), with the VSS above it; each value of the VSS having exactly 1 platform directly below it (and with the '2' or '3' in the penultimate position):

```
 7   5   3   5
[ _ , _ ] _   _
```

4) Now work out where each vanilla value is next thrown. Eg, the '7' is next thrown at the place indicated by '( )':

```
 7   5   3  (5)
[ _ , _ ] _   _
```

5) Then work out what value would have to go on the platform below the vanilla value, so as to 'hit' the same 'place'. In our example, a '5' on the first platform of our multiplex pattern, would hit the same position as the '7' in the vanilla sequence.

```
 7   5   3  (5)
[ 5 , _ ] _  (_)
```

Repeat steps 4 and 5 until the new pattern is complete. In our example, we end up with:

```
 (7)  (5)  (3)  (5)
[(5), (4)] (2)  (4)      - ie [5,4] 2 4 (aka 'Gatto's multiplex').
```

You should now be able to see why we insisted on having a ‘2’ or ‘3’ in the penultimate position: because it is converted into a ‘2’ in the multiplex – which can be held, thereby eliminating the problem of having to catch both of the balls (the ones to be multiplexed) in the same hand at about the same time. You may be disappointed however, to find that several different VSSs get converted into the same multiplex pattern; for example, 6635, 6662 (in the form 6626), and 7526 also produce [5,4]24. There are also practical problems with many multiplex SSs: multiplex throws containing heights which are far apart are notoriously difficult; as are those combining a ‘1’ with a higher throw. Having said this, there are definitely a few gems to be found (see SS appendices), so good hunting....

Admittedly, carrying out this algorithm by hand is rather cumbersome; if you want to generate lots of patterns quickly, I recommend writing a computer program to do it.

In the case of designing multiplexes containing a single duplex, ie of the form $[M_1,M_2]\ M_3\ M_4\ M_5 \ldots M_P$, the following formula may be used, to convert a VSS. Let’s call its values $V_1\ V_2\ V_3 \ldots V_P$.

$M_i = V_i - ((V_i + i - 2) \backslash P)$ (Except when i=1: subtract 1 from the value calculated for M_1)

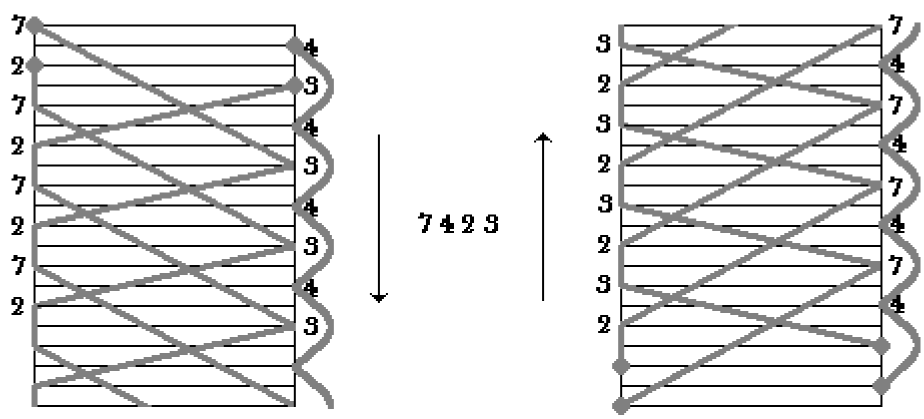
(‘\’ is division, with the result rounded down to the nearest whole number, eg 21\2 = 10)

TIME REVERSALS

(A)

How do you work out what the time-reversal of a SS is? Well one of the most instructive ways to see the answer, is using ladder notation. To illustrate, lets have a look at 7423, with time first running down, then up (with the indicated values calculated downwards, in both diagrams). Hold times are ignored.

Time & values running down: Time running up, values down:



From the above, we can see that the time reversal of 7423 is 7342. With a bit of thought, we can deduce that the following algorithm will always produce the time reversal:

1) Write down each throw value underneath where it is next thrown, eg

7	4	2	3
2	4	3	7

2) Then, read this new sequence backwards, eg 2 4 3 7 backwards, is 7 3 4 2 – this is the time reversal.

Some questions may spring to mind, such as: what SSs are equal to their own time-reversal (ie are ‘time-symmetric’)? Well, all of the period 1 and 2 SSs are, as well as most of the period 3 and 4’s. In fact, low-period SSs which are not time-symmetric, are few and far between. The simplest one is 603 (whose reversal is 630).

So anyway, if you have a favourite SS, work out its reversal to see if it is time-symmetric. If it isn’t, try juggling its reversal.

BALL FOLLOWING

(M)

One of the nice features of SS notation, is that you can easily work out what each ball is going to do next. For example, with the SS 7441, you can see that 2 balls cycle round on ‘7’s and ‘1’s, whilst 1 ball stays in the left hand on ‘4’s, and the other, also on ‘4’s, stays in the right hand. How do we know this?

Well let’s look at where the ‘7’ goes. Counting on 7 places, we see that the ‘7’s are next thrown as ‘1’s. We then notice that the ‘1’s are next thrown as ‘7’s again. Adding these values together, and dividing by the period of the pattern, gives the result that 2 balls continuously patrol this path.

Similarly, we notice that each ‘4’ ‘lands’ on itself. As even numbered values stay in the same hand (in 2-handed patterns), each ‘4’ has its own hand, and remains there, leaving its hand only to hop over the horizontally moving ‘1’s.

The cycle of throws which a ball follows, is technically termed its ‘orbit’ (not to be confused with the swirly juggling move of the same name). Some SSs have just a single orbit (such as 441 and 6312), whereas some have several. A subtle point, is that when dealing with odd-period alternating SSs, it may look like there is just 1 orbit, when in fact, there is (in a very real sense) 2. An example of this kind is 741. The ‘7’ becomes the ‘4’, the ‘4’ ‘hits’ the ‘1’, and the ‘1’ hits the ‘7’. In fact, in 741, 2 (out of the 4) balls are never thrown as a left-hand ‘7’. This can be clearly seen, if we expand the SS into its full sequence (with the hands taken into account) 741741, and follow the ‘7’ again. It ‘hits’ the throws in brackets as indicated: (7) (4) 1 7 4 (1), and returns to the first ‘7’ before hitting the other 3 throws.

Anyway, the reason this section is related to the designing of SSs, is because sometimes, you want certain balls to stay in certain orbits. The pattern(s) known as ‘Tennis’, will serve as an example:

You can juggle 3 ball tennis on the SS 52233. (You can also use plain old ‘3’, but this causes awkward timings.) In the pattern, one of the balls (the ‘tennis ball’) is thrown higher than the others, over the top of the pattern, into the other hand. You will notice, that in 52233, the ‘5’ is always the same ball, and this ball only travels on ‘5’s. These are 2 of the features that ‘make’ the pattern. So what SS could we use for a 4 ball version? Well, to maintain these crucial features, we need the SS to be the same period as the value of the tennis throws. We would also like all of the other throws to be lower than the tennis ball throw. The SS that springs to mind is 53444. Another, more exotic SS that we could use, is 7333444, and there are many others. For 5 ball tennis, we could use 7445555. For 6: 7566666. And so on.

TRANSITIONS

How can you get from one SS to another? The way most jugglers do it, is to go back to the standard cascade or fountain, and into the new pattern from there. This is fair enough, but it’s not very elegant. Also, you might not even know how to get to/from some SSs from/to the standard pattern. Here are a couple of solutions to this problem.

Suppose we are juggling SS A, and we want to go into SS B. Now if both A and B are ground SSs (ie pass through ground state), or we know which states both of these pass through, and there is a state which both of them use, then we can change over from one to the other (and back again) when we hit this state. Even if there is no such state, we might still be able to find a throw which takes us into a state used by the target pattern (B). A more reliable approach though, is the following:

1) Decide what throw of SS A you want to end on.
2) Decide what throw of SS B you would ideally like to start with.
3) After the throw decided in 1), try to start SS B (on the throw decided in 2), using the following rules:

    a) If you can make the intended throw (ie if it doesn’t cause a clash), then make it, and go to d).
    b) If not, then make the nearest-valued, lower throw possible, and go to d).
    c) If there is no valid lower throw, then make the lowest throw possible.
    d) Move onto the next value in the SS B, and repeat from a).

Of course, if patterns A and B don’t use the same number of balls, then we need magic.

Eventually, following these rules, we will be juggling SS B. Let’s use this algorithm to generate a transition from the fountain to the shower (with 4 balls), so A = 4, and B = 71. We have to end with a 4, but let’s start the shower with the ‘7’. Ideally then, we would like to juggle: 4444717171.... This is not possible however, as we’ll see by tracking the states. We know that the SS ‘4’ uses the single state 1111. So starting here, let’s go: 1111 (7)$\rightarrow$ 1110001 (1?). We cannot throw this ‘1’, as it would clash with another ball. Neither can we throw lower than a ‘1’. We therefore have to make the lowest throw possible, which is a 3. So we continue.... 1110001 (3)$\rightarrow$ 111001 (7)$\rightarrow$ 1100101 (1?). Again, we cannot do a ‘1’ or lower, so we do a ‘2’, and continue: 1100101 (2)$\rightarrow$ 110101 (7)$\rightarrow$ 1010101 (1)$\rightarrow$ 110101. We now recognise this state (we passed through it 2 throws previous), and so the transition is complete, being ‘7372’ (so 4 (7372)$\rightarrow$ 71). An easier, though less precise method, involves platforms, as follows:

1) Write out at least M or P (whichever is biggest) throws of the initial SS (M = maximum value of the pattern, P = period), then a few empty platforms (at least B; (B = the number of balls) – you may just need more), then a few throws of the target SS (again, at least M or P (whichever is biggest) of this pattern).

2) Mark where all the known throws go (eg by bracketing, circling or dotting the platforms).

3) Then choose values to go on the empty platforms, making sure they won’t cause a clash.

Let’s use this method to find a transition from 71 to 831:

1) M(71) = 7, P(71) = 2, so we need at least 7 throws of 71 to begin with (let’s have 8). B = 4, so we’ll try 4 empty platforms in the middle. M(831) = 8, P(831) = 3, so put at least 8 throws of 831 at the end.

7 1 7 1 7 1 7 1 _ _ _ _ 8 3 1 8 3 1 8 3 1

2) 7 1 (7) 1 (7) 1 (7) (1) (_) (_) _ (_) 8 (3) 1 (8) (3) 1 (8) (3) (1)

3) Note that (after the end of 71) there are 4 platforms without a throw, and 4 platforms without an available ball. It is always a good sign when these numbers are equal. If they are not, then something has gone wrong. Filling in the remaining platforms with the lowest possible throws, we end up with:

7 1 (7) 1 (7) 1 (7) (1) (2) (3) (4) (6) (8) (3) (1) (8) (3) (1) (8) (3) (1)

So our transition is 2346. Of course, we don’t have to choose the lowest possible throws – we could have designed the transition to be 4146, 6126, 6171 etc.

PUZZLES

For the particularly keen, here are five SS puzzles (answers on page 74):

1) Find all VSSs which use only ‘1’s and ‘7’s (there are only a finite number of these).

2) Find all 7 ball, period 3 VSSs (containing no values higher than 12), in which 3 balls visit both hands, 2 balls never visit the left hand, and 2 never visit the right.

3) Find a prime, non-ground, 3 ball VSS with no height over 7, containing no ‘2’s, in which each ball has its own orbit, and no 2 orbits have a SS value in common. (Then juggle 5 rounds of it without a drop.)

4) Using the 4 ball state transition table on page 35, find (using only throw-heights 1 to 8):
a) a ground, prime SS, period $\geq$ 19; b) a non-ground, prime SS, period $\geq$ 17.

5) Using the axioms on page 36, prove that 501 is a valid SS, without using a) A4; b) A3 (but using A4).

5) DESIGNING PATTERNS

INVENTING PATTERNS

(J)

If we want to design a new pattern, where do we start? The obvious place is with the most basic of all patterns – the 3 ball cascade, juggled with normal throws and catches:

THR(Time)	**{**	2	1	**}**
SS(Base)	**{**	3	3	**}**
THR(Site)	**{**	R	L	**}**
THR(Pos)	**{**	m	m	**}**
THR(Type)	**{**	n	n	**}**
CAT(Pos)	**{**	l	r	**}**
CAT(Type)	**{**	n	n	**}**
AIR(Min)	**{**	1	1	**}**

(If we wanted a pattern with longer period, we could just write out more columns of the matrix.)

Now we decide how we want our new pattern to differ from the 3 ball cascade. To do this, we can use the following checklist:

1) What type of object do we intend to use in the pattern? If we want to use something other than balls, then we may want to add some of the extra rows AIR(Spin), AIR(Ring), AIR(Twst) or AIR(Bnce). (We don't need to worry about what values to put in them yet.)

2) Do we want to use only our hands in the new pattern? If we want to use other parts of our body (or any other sites) then how often do we want to use them? Once per cycle? Just after throws from each hand? The THR(Site) row should be set up accordingly.

3) Do we want a synch pattern or an alternating one? If synch (with 2 sites), then change the THR(Time) row to **{** P 0 2 2 4 4 .. **}** (where P is the cycle time of the pattern). What cycle time do we want the pattern to have?

4) How many objects do we want in our pattern? Do we want all the throws to have the same SS value?

5) Do we want our hands to move about, or stay put (in the normal positions)?

6) Do we want to use any unusual throw or catch types, such as claws or penguins?

7) Do we want to distort the airtime of any throws, or even hold them (if possible)?

8) What values do we want in any extra rows chosen in 1).

After all these questions have been answered, and the appropriate GS rows have been altered, we can attempt to juggle our creation. If we don't like any aspects of it, we can easily change them. Hopefully, we'll end up with a totally new pattern. If it's funky enough, we can name it, polish up our performance of it, and let people enjoy watching it at a juggling convention.

As an alternative to using the 3 ball cascade as your template, try modifying other patterns (see the Pattern Appendix for examples (pages 70-73)).

Finally, note that the vast majority of currently known patterns (or something very close to them) can be arrived at by the above procedure, and there are bound to be plenty of beautiful but unknown patterns still out there.

COMBINING PATTERNS

As mentioned previously, there are several ‘dimensions’ to a juggling pattern. Different patterns utilise different features. To illustrate this point, below is a list of some (mostly) well-known patterns, categorised by the variant feature (‘the dimension’) which most characterises the pattern:

Throw/Catch Site	Throw/Catch Pos’n	Throw/Catch Type	Siteswap	Airtime
Eating the apple	Mills Mess	Overhead Cascade	Shower	Martin
Chin balance	Boston Mess	Penguin	Half-shower	Tennis
Helicopter	Shuffle		Sprung cascade	4 ball cascade
Orang-utan	Alberts			

Some patterns, such as Rubenstein’s Revenge, Burke's Barrage, and The Factory operate in more than 1 dimension – Rubenstein’s Revenge for example uses non-standard positions as well as claw catches; Burke’s Barrage uses varying positions with a 423 SS sequence, and The Factory (at least the version defined in Charlie D’s EBJ) uses positions, types, and SS (424233).

Of course such patterns are likely to be harder than those which concentrate on only 1 aspect, but they are often very satisfying to juggle and stimulating to watch. Actually, combining the essential features of patterns which play on different dimensions is often fairly straightforward, at least in theory. A few suggestions to get you started are: Mills Mess on 441 (3 versions, dependent on which throw is the underarm throw), Boston Mess on backhands, Mills Martin (5 balls), and Eating-the-apple on 6252535 (4 balls; bite the first ‘2’).

ADDING BALLS TO 3 BALL PATTERNS

Many jugglers, having mastered the 3 ball Mills Mess, have ambitions to juggle the 4 ball version. Although this may take years to achieve, it is actually very easy to write down the GS notation for it, as the only difference from the 3 ball version, is the SS(Base) row: it is **{** 4 4 4 4 4 4 **}** instead of **{** 3 3 3 3 3 3 **}**. Unsurprisingly, the legendary 5 ball MM is likewise identical, except the throws are ‘5’s. However, not all patterns can have a ball added in such an obvious way.

Take for example Burke's Barrage, which uses SS 423. The ‘2’ (the hold) is crucial in this pattern – without it, the ‘swirly bit’ would be impossible due to lack of an opportunity to hold a ball for long enough. It is clear then, that the 4 ball version, must also contain a ‘2’ (as well as having period 3). The obvious choice then, is 552 (or even 642) . And for 5 balls, 726 or 825 could be used.

Similarly, Rubenstein’s Revenge can be smoothly juggled as 52233 – the consecutive ‘2’s allowing enough time for the ‘double-orbit’. A suitable SS candidate for the 4 ball RR would ideally have the underarm throw (the throw just before the ‘2’s) at least as high as the other throws, to maintain as much resemblance to the 3 ball version as possible. Using platforms (or some other method), a couple of possibilities emerge: 66224, and 72236. For those brave enough to attempt the 5 ball RR, 77722 is the most likely option, the next most plausible being 10,2247. Good luck with these.

CHOOSING BACKGROUND MUSIC

Like most things in life, juggling is even more fun when done to music. Audiences also greatly appreciate musical accompaniments to routines. The most appropriate type of music for practising and performing to, depends on what style of juggling you are doing. Jazz and classical music (0 to 100 beats-per-minute) is often suitable for smooth, graceful 3 ball routines; jungle (120+ bpm) for fast, swirly 3 or 4 ball stuff; house and techno (120 to 140 bpm) for slower, 5 or 6 ball routines; and happy hardcore or hi-speed drum & bass (140 to 180 bpm) is ideal for quick 5 ball siteswaps or higher numbers.

BEAT ACCENTUATION

(M)

Once you've become fairly confident with a few different patterns, you can make the game more interesting (and harder) by juggling to music, and trying to 'mirror' the rhythm of it, with your pattern. With 3 balls, this could mean throwing balls higher, on say, the 1st beat of the 'bar' (= a short musical phrase). If your music has a constant beat to it however, you can try to juggle in time to it, and vary the SS values. For example, suppose the music is '4/4 time' (like most pop music); then any SS – period 2, 4, 8, 16 etc will fit into the timing of the music. Now certain beats (in the music) will be emphasised, so you could make higher-valued throws on these beats, eg try 3 3 4 4 1 4 4 1 or 4 2 5 2 4 1 3 3. Mirroring complex drum patterns or crisp, clean melodies with SSs is particularly nice, especially with 4 or more balls. Period 8 SSs beginning with an '8' look good, as when they are repeated, the same ball initiates each bar. Also, beat-accentuation works much better if you use exactly 1 or 2 beat holdtimes, so that catches are also 'on the beat'.

The more balls you are juggling, the more (theoretical) scope there is to accentuate. For SSing with more than 3 balls, music of 120+ bpm is best – slower than this, and your '6's will probably need to be too high for most ceilings.

ASSEMBLING ROUTINES

(J)

Still on the music theme, think of a juggling pattern as a 'broken chord' on a guitar or piano, each throw being one of the notes. Now just as it is pleasing to hear nice chord sequences, it is very enjoyable to watch a good juggling routine (a sequence of patterns). Unfortunately, designing a routine can feel like a chore. It can be difficult to know where to start. To help you with this, here are some general guidelines you may find helpful:

1) Decide what audience you intend the routine to be aimed at. There are essentially 3 main possibilities. Most non-jugglers can't tell the difference between a Burke's Barrage and a Rubenstein's Revenge – so keep it reasonably simple for them, eg throw a ball high, then go back into a 3 ball cascade. It isn't hard, but it impresses them, so don't try too difficult tricks – *not dropping* counts for more. For a juggling audience on the other hand, you have to try harder stuff (drops don't matter so much). If it's just as a warm up exercise for yourself, put whatever you like in.

2) Try to order your tricks/patterns with most of the easier ones earlier on. This way you can hope to build up your confidence (which will help you with the harder tricks) by not dropping too much at the start. Also, (in a good routine) the audience's anticipation will increase during the routine, so they will be wanting to see the best stuff last (of course 'better' doesn't necessarily have to mean 'harder, but unimpressive hard tricks are best avoided).

3) Audiences like to be able to understand some (but probably not all) of what you are doing. To achieve this, it is a good idea to make consecutive patterns have something in common, if possible; for example, going from one synchronous pattern to another, or doing several Mills Mess variations in succession. It is probably a good idea to break your routine down into several subroutines, each containing a different kind of trick from the others. This is not to say you should never put 2 totally different tricks next to each other – sometimes the contrast can give a nice effect; but I would advise against doing this too much.

4) Like chord-changes on a musical instrument, the transitions between patterns are more difficult than the patterns themselves, and should be practised in their own right, as they will commonly be your 'drop-points'.

5) With the possible exception of your finale, only include tricks that you can normally pull-off 3 times in a row (you should normally pull-off finale tricks at least once every 3 attempts – most audiences will allow you this many chances). You should also aim to drop fewer than 10 times (per 5 minutes). If you are regularly dropping more than this, it would probably be wise to change or remove some of the harder material before performing it to an audience.

See also 'Practising' (page 57).

6) PASSING PATTERNS

Another way to increase the number of throw-sites in your juggling pattern (if your other body parts are not very skilled) is to borrow some off someone else. Most other people also have 2 hands that you can use. There are a couple of drawbacks with this idea though: firstly, you will need to ask the owner of the hands, and secondly, you have no direct control over the kind of throws which they make. It is a good idea to choose hands which have already been trained to about the same level as your own (or slightly higher); if you use low-grade hands, they don't throw or catch very well, and the pattern probably won't last very long, but if they're too much better than yours, the owner might lose interest and take them home.

Once you've got a partner (or partners) you still have to choose which air-sculpture you want to form. There are many possibilities – even more than for a solo juggler. Unfortunately, due to control of the hands being divided between multiple brains, some complex solo patterns are not easy to translate into passing patterns. Common problems are caused by differences in juggling styles between the participants; compatible heights, rates, and positions have to be agreed on beforehand, and maintained during the pattern.

This chapter begins by describing 3 of the main types of passing patterns: 4 counts, 2 counts, and 1 counts; the numbers indicate how often passes occur. A 4 count is one where every 4th throw is thrown to a partner. In 1 counts, every throw is a pass. 3 counts and other passing sequences are also possible, but the general principles are similar. (See also 'Passing Patterns' (pages 67-69) for more ideas.)

In this chapter, assume that each pattern involves 2 jugglers who face each other and throw 'tramline passes', unless stated otherwise. Note: in the context of passing patterns, a 'self' is any throw which is not a pass.

EVERY OTHERS (4 COUNT)

(J)

6 Balls

Let's start with the most basic 6 ball 4 count. Both jugglers repeatedly juggle (what feels like) 3p 3 3 3, where the 'p' means 'passed to partner'. In GS notation, the pattern is:

THR(Time)	**{**	4	0	1	1	2	2	3	3	**}**
THR(Site)	**{**	1R	2R	1L	2L	1R	2R	1L	2L	**}**
SS(Base)	**{**	7	5	6	6	6	6	6	6	**}**

Admittedly, this is not very readable- it is better to write the 2 juggler's 'parts' separately, and implicitly assume the R and L hands (of both jugglers) throw alternately, at times 0, 1, 2, ..., to get the more informal, but instructive description (using SS(As) values):

J1:	**{**	3p	3	3	3	**}**
J2:	**{**	3p	3	3	3	**}**

J1 = juggler 1, J2 = juggler 2. The above notation format is used for the rest of this chapter. Note that patterns in this form still admit a simple platform-based procedure for verifying its validity as a siteswap.

Both jugglers start with a right hand pass, at the same time. The passes are all 'tramlines' (ie travel down the side of the pattern). 'Crossing' passes (ie right to right or left to left passes) are best avoided in patterns where both people pass at the same time, due to their high potential for collisions.

Tricks later. First, how can higher numbers of balls be juggled in a 4 count?

7 Balls

For 7 balls, the simplest pattern is:

J1:	**{**	5p	3	3	3	**}**
J2:	**{**	3	3	5p	3	**}**

In this pattern, J1 starts with 4 balls (2 in each hand). J2 has 3 (2 in right, 1 in left). J2 can either start with a ‘3’ at the same time as J1’s ‘5p’, or wait until J1’s pass is about halfway over before starting with their own ‘5p’ (from their right hand). As the passes don’t occur at the same time, they can be thrown as crossing passes, providing both partners know that this is the idea, and one of them performs the ‘5p’s with their left hand. Remember to make the ‘5’s considerably higher than the ‘3’s – perhaps 5 or 6 times as high.

<u>8 Balls</u>

2 possibilities are:

J1: { 5p 3 4 4 }   J1: { 7p 3 3 3 }
J2: { 5p 3 4 4 }   J2: { 7p 3 3 3 }

However, the following synchronous pattern is probably the easiest way to handle 8:

J1: { (4p,4x) (4 , 4) }
J2: { (4p,4x) (4 , 4) }

<u>9 Balls</u>

J1: { 5p 4 4 5 }   J1: { 6xp 4 4 4 }
J2: { 4 5 5p 4 }   J2: { 4 4 6xp 4 }

In the pattern on the right, the ‘6’s can be thrown as tramlines if one of the jugglers does them with their left hand. Note that this would mean that all the passes occur on the same side of the pattern.

Again though, there is a nice synchronous pattern:

J1: { (6p,4x) (4 , 4) }
J2: { (4 , 4) (6p,4x) }

Patterns with 10 balls are (in one sense) quite like those with 6 – just add 2 to all values in a 6 ball pattern. Do similarly to turn 7 ball patterns into 11 ball ones; 8 into 12, 9 into 13, and so on.

<u>SOLIDS (2 COUNT)</u>

<u>6 Balls</u>

Another favourite format with ‘one-handed’ jugglers is the 2-count. Pass with your right, and self with your left is the basic idea.

The beginner’s pattern is:

J1: { 3p 3 }
J2: { 3p 3 }

- both jugglers throw a right hand pass at the same time, followed by a left hand crossing self.

An alternative base pattern, is using the ‘shower’ technique:

J1: { (4p,2x) - }   J1: { 5p 1 }
J2: { (4p,2x) - }   or   J2: { 5p 1 }

<u>7 Balls</u>

The main options are (again both jugglers pass with their right hands):

J1: { 4p 3 }   J1: { (5p,2x) - }   J1: { 6p 1 }
J2: { 3 4p }   ,   J2: { - (5p,2x) }   ,   J2: { 1 6p }

8 Balls

J1: **{** 5p 3 **}** , J1: **{** (6p,2x) - **}** , J1: **{** 7p 1 **}**
J2: **{** 5p 3 **}** , J2: **{** (6p,2x) - **}** , J2: **{** 7p 1 **}**

Another feasible pattern with 8 (and probably the easiest) is:

J1: **{** (4p,4x) - **}**
J2: **{** (4p,4x) - **}**

9 Balls

The nicest has to be:

J1: **{** (5p,4x) - **}**
J2: **{** - (5p,4x) **}**

- although there is also the trickier:

J1: **{** 6p 3 **}**
J2: **{** 3 6p **}**

10 Balls

You might think that the easiest pattern for 10 balls would be:

J1: **{** 5p 5 **}**
J2: **{** 5p 5 **}**

- but this is actually rather difficult, due to your right hand wanting to throw the passes higher than your left hand selfs. A more natural-feeling pattern is thus:

J1: **{** (6p,4x) - **}**
J2: **{** (6p,4x) - **}**

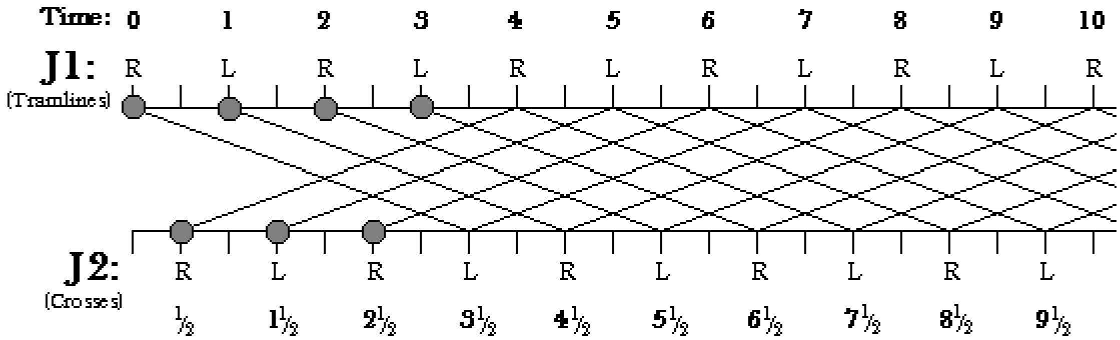
You may find that starting with 2 balls in your right and 3 in your left works best; if you throw a ($6p_R$,$4x_L$) (hands subscripted) with 3 balls in your right, the '4x' will land in your right before it gets rid of all its balls.

For 11, 12, 13 etc balls, add 2 onto each value in the patterns for 7, 8, 9 etc.

ULTIMATES (1 COUNT)

Every throw is a pass. Usually, all throws are to the same height. 2 person ultimates works best with odd numbers of balls; with even numbers, if all throws are to the same height, then both jugglers have to either throw tramlines (in which case they will be juggling 2 separate cascades, 1 on each side) or crossing passes, thrown at the same time on the same side (with high collision potential); again the pattern consists of 2 independent subpatterns.

To achieve full 'ball circulation', odd numbers of balls are used, with the jugglers throwing ½ a beat out of time to each other. One person throws tramline passes, the other throws crossing passes. The following ladder diagram (which ignores holdtimes) shows where each ball goes, in the 7 ball version (compare with the causal diagram of the same pattern, at the bottom of page 30):



This principle can be applied to any odd number of balls.
In GS notation, we could record this as:

THR(Time)	{	2	½	1	1½	}
THR(Site)	{	1R	2R	1L	2L	}
SS(Base)	{	7	7	7	7	}
SS(As)	{	3½tp	3½xp	3½tp	3½xp	}

('t' stands for tramline pass)
For the rest of this section though, I'll just use the shorthand passing notation, so the above becomes:

J1:	{	3½tp	.	}
J2:	{	.	3½xp	}

For simplicity, assume that J1 always starts (and hence begins with 1 more ball than J2), and both jugglers make their first throw with their right hand.

<u>5 Balls</u>
This is slow. You can almost doze off and wake up again without dropping in this pattern. This makes it ideal for the uninitiated.

J1:	{	2½xp	.	}
J2:	{	.	2½tp	}

<u>7 Balls</u>
As already stated:

J1:	{	3½tp	.	}
J2:	{	.	3½xp	}

<u>9 Balls</u>
This is similar to 5 balls (in that the one who starts, throws crossing passes).

J1:	{	4½xp	.	}
J2:	{	.	4½tp	}

In general, just add 2 onto the values to change 'B ball ultimates' into 'B+4 ball ultimates'.

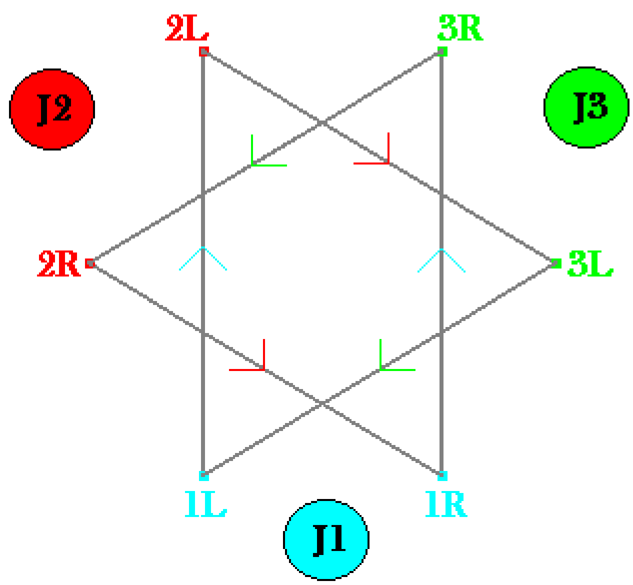
With a bit of ingenuity (or luck), it is possible for the partners to swap roles mid-pattern, so that the tramliner starts doing crosses, and vice-versa. See if you can work out how this can be done (in real-time or on paper).

Mills Mess

The 2 person Mills Mess is a highly satisfying pattern. You can use any number of balls, and surprisingly it's not actually that difficult. The only thing you need to remember, is that you should aim all throws at the middle of your partners body. There are 6 versions with any given number of balls (differing by how you and your partner's throws interlock); some are easier/more aesthetically appealing than others. For the more adventurous: try one person passing a 4 ball MM, whilst the other does 3.

Star of David

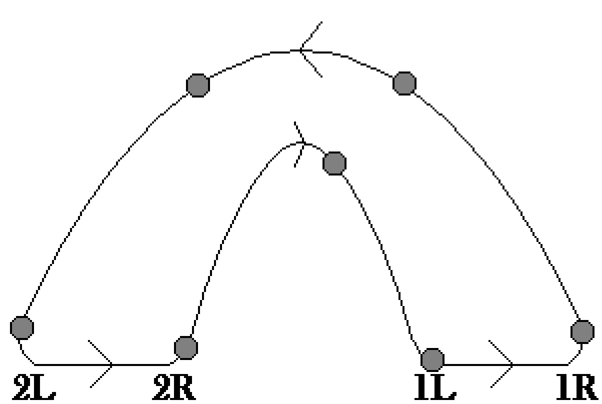
This is a pattern for 3. For it to be interesting, all 3 jugglers need to be at least comfortable with the 4 ball fountain. Here is the plan view of the positions of the jugglers, hands, and ball-routes:



Everyone starts at the same time, with a right hand throw. The right and left hands alternate. Notice that there are 2 independent triangles, the 3 right hands being the vertices of one triangle, and the left hands forming the other. This means that half the pattern can be practised at a time, so if anyone isn't pulling their weight, the offending hands can be quickly identified and tutored. Any multiple of 6 balls can be juggled in this pattern; the 6 ball version is a challenge for young children, the 12 for competent jugglers, and the 18 for experts.

DOUBLE SHOWER

This is another pattern for any number of balls, which is highly pleasant to watch. Here is a front-on view:



The idea is as follows: J1 and J2 stand side by side. J1 launches each ball (with their right hand) high over the top of the pattern, to be caught by J2's left hand, passed across to their right hand, showered to J1's left, and finally passed back to J1's right. It doesn't really matter about the exact height or timing of the throws, but one way of keeping 'in time' (using 7 balls) would be:

J1: **{** (6p,2x) - **}**
J2: **{** (4p,2x) - **}**

UNEVEN PATTERNS

It is possible for jugglers of vastly different skill levels to do passing patterns (together) which provide a suitable level of challenge for all. There are several ways to make one partner's task more difficult – by altering the types or positions of throws or catches, having them do twice as many throws in the same amount of time, or increasing the SS values.

Suppose J1 is an expert juggler, and J2 is a novice. J1 could do a 4 count, whilst J2 does a 2 count at half the speed. The most elegant kind of handicap patterns however, use different heights of throws. Examples of this kind include the delectable 2 height ultimates patterns, such as (with 7 balls):

J1: **{** 4xp **}** (Start: 2 in R, 1 in L, 1st throw: R)
J2: **{** 3tp **}** (Start: 2 in R, 2 in L, 1st throw: R)

or

J1: **{** 4tp **}** (Start: 2 in R, 1 in L, 1st throw: R)
J2: **{** 3xp **}** (Start: 2 in R, 2 in L, 1st throw: L)

Here are a few more handicap patterns. All passes are tramlines unless stated otherwise; the number of balls under the control of each juggler (= the average SS value they have to throw) is indicated in brackets, eg J1(4) means J1 juggles 4 balls on average. Just add the 2 numbers together to work out how many balls are used in the pattern.

J1(4): **{** 5p 3 4 4 **}** J1(4): **{** 5p 3 4 4 **}**
J2(3): **{** 5p 2 2 3 **}** J2(3): **{** 3 3 3p 3 **}**

J1(3½): **{** 5p 5p 2 2 **}** (Start: 2 in R, 2 in L, 1st throw: R)
J2(3½): **{** 3 3 4xp 4xp **}** (Start: 2 in R, 1 in L, 1st throw: R)

J1(3¾): **{** 4p 4p 4p 3 **}** (Start: 2 in R, 2 in L, 1st throw: R)
J2(3¼): **{** 4p 4p 1 4p **}** (Start: 1 in R, 2 in L, 1st throw: L)

J1(4¼): **{** 5p 4 5p 3 **}** (Start: 2 in R, 2 in L, 1st throw: R)
J2(2¾): **{** 4xp 1 2 4xp **}** (Start: 1 in R, 2 in L, 1st throw: R)

J1(4¾): **{** 6p 4 4 5 **}** (Start: 3 in R, 2 in L, 1st throw: R)
J2(3¼): **{** 3 3 3 4p **}** (Start: 1 in R, 2 in L, 1st throw: L)

J1(3½): **{** 6xp 2 5p 1 **}** alternatively (4½): **{** 6xp 3 5p 4 **}**
J2(3½): **{** 5p 3 4xp 2 **}** alternatively (3½): **{** 5p 4 4xp 1 **}**

J1(4): **{** 5p 3 4 **}**
J2(3): **{** 3 3 3p **}**

The *platform method* is recommended for designing complex passing patterns like these.

PARTNER THROW INVARIANCE (PT-INVARIANCE)

Once you and your partner have a nice pattern up and running, you can start to do variations, in terms of one-off tricks and pattern-changing. The easiest moves (for your partner) are those in which they don't have to change the throws they make; I will refer to such moves as 'partner throw-invariant' (PT-invariant, for short). These divide into 2 subgroups, depending on whether your partner has to catch differently weighted throws than normal. If your partner's catches *are* as normal, then the trick will also be called 'partner catch-invariant' (PC-invariant). To illustrate these concepts, let's use the standard 2 person 6 ball 4 count (with tramline passes):

J1: **{** 3p 3 3 3 **}**
J2: **{** 3p 3 3 3 **}**

Now either juggler could replace one (or more) repetition of **{**3p 3 3 3**}**, with **{**3p 3 4 2**}**, without their partner having to do anything different. However, if J1 (say) did **{**2 3 3 4xp**}** (beginning with the 4xp, an 'early double'), then J2 (whilst still being able to carry on doing **{**3p 3 3 3**}**) would have to catch a 4xp rather than the usual tramline 3p.

To get you started, here are some (PT-invariant) tricks you can do without upsetting your partner. These should be thrown from the indicated base pattern. In the moves which can't be initiated with the leftmost throw of the sequence, a '!' indicates the point at which the base pattern is deviated from, and subsequently returned to.

<u>6 ball 4 count</u>: Base: J1 & J2: **{** 3p 3 3 3 **}**

PC-invariant: **{** 3p 4 2 3 **}**, **{** 3p 3 4 2 **}**, **{** 3p 4 4 1 **}**, **{** 3p 5 2 2 **}**, **{** 3p 5 3 1 **}**, **{** 3p 3 6 2 3p 1 3 3 **}**...
PC-variant: **{** 2 3 3 !4xp **}**, **{** 4 1 3 !4xp **}**, **{** 2 3 !5p 2 **}**, **{** 4 1 !5p 2 **}**, **{** 1 3 !5p 3**}**, **{** 2 !6xp 2 2 **}**...

<u>7 ball 4 count</u>: Base: J1: **{** 5p 3 3 3 **}**
J2: **{** 3 3 5p 3 **}**
(For J1)
PC-invariant: **{** 5p 4 2 3 **}**, **{** 5p 3 4 2 **}**, **{** 5p 4 4 1 **}**, **{** 5p 5 2 2 **}**, **{** 5p 5 3 1 **}**, **{** 5p 3 3 5 5p 1 3 3 **}**...
PC-variant: **{** 4 4xp 3 3 **}**, **{** 4 4 3p 3 **}**, **{** 5 5 3p 1 **}**, **{** 4 1 3 !6xp **}**, **{** 5p 3 7p 3 1 3 3 3 **}**...

<u>8 ball 4 count</u>: Base: J1 & J2: **{** (4p,4x) (4,4) **}**

PC-invariant: **{** (4p,6x) (2,4) **}**, **{** (4p,6x) (4x,2x) **}**, **{** (4p,2) !(6,4x) **}**...
PC-variant: **{** (4,2) !(6p,4x) **}**, **{** (4,2x) !(6p,4) **}**, **{** (2,2) !(6p,6x) **}**...

<u>6 ball 2 count</u>: Base: J1 & J2: **{** 3p 3 **}**

PC-invariant: **{** 3p 5 3p 1 **}**, **{** 3p 7 3p 1 3p 1 **}**...
PC-variant: **{** 3p 4xp 2 3 **}**, **{** 3p 4xp 4 1 **}**, **{** 3p 4xp 5p 1 2 3 **}**, **{** 3p 6xp 3p 1 4 1 **}**, **{** 3p 4xp 5p 6xp 0 1 2 3 **}**...

<u>7 ball 2 count</u>: Base: J1: **{** 4p 3 **}**
J2: **{** 3 4p **}**
(For J1)
PC-invariant: **{** 4 3xp **}**, **{** 4p 5 4p 1 **}**, **{** 4p 7 4p 1 4p 1 **}**...
PC-variant: **{** 4p 5xp 2 3 **}**, **{** 4p 5xp 6p 3 0 3 **}**, **{** 4p 5xp 6p 7xp 0 1 2 3 **}**, **{** 4 3xp 4p 5xp 6p 7xp 0 1 4 1 **}**...

<u>8 ball 2 count</u>: Base: J1 & J2: **{** (4p,4x) **}**

PC-invariant: **{** (4p,6x) (4p,2x) **}**, **{** (4p,8x) (4p,2x) (4p,2x) **}**...
PC-variant: **{** (4,4xp) **}**, **{** (4p,6xp) (2,4x) **}**, **{** (4p,6xp) (4,2x) **}**, **{** (6p,4xp) (6p,2x) (6p,2x) ... (6p,2x) (4,2x) **}**...

There is a large number of potential variations for 4 and 2 count patterns (and 3 counts also). There does seem however, to be considerably less feasible PT-invariant possibilities for 1 counts, particularly with less than about 11 balls. In fact, it is clear that there are in fact *no* PC-invariant SS tricks at all for 1 counts. This is because, as every throw is a pass, any higher throws you make will have to be caught by your partner. Unfortunately, the PT-invariants that *do* exist are either collision prone or involve hideously difficult catches; note that passes of value ½, 1½ and 2½ are going to be extremely difficult in a pattern with 7 balls or more.

Below are some of the few possible PT-invariant 1 count tricks for 9 balls:

9 ball 1 count: Base: J1: **{** 4½xp . **}**
J2: **{** . 4½tp **}**

(Tricks for J1; switch 't's and 'x's below, if J2 is to do them)
PC-variant: **{** 5½tp 3½tp **}**, **{** 6½xp 3½tp 3½tp **}**, **{** 5½tp 5½tp 2½xp **}**, **{** 6½xp 4½xp 2½xp **}**

FORCED RESPONCES

If, whilst doing a fairly straightforward passing pattern, your partner begins to fall asleep, one of the best ways of reviving them (or at least collapsing their pattern), is by throwing a 'response-forcing' combination. There are several 'grades' of forced-response moves, based on how difficult they are for your partner to survive. If you don't want to cause any trouble, you can just throw a 'soft response-force' (SR-force) at them. This merely requires your opponent – i mean *partner*, to hold (or have a gap) when they would normally have thrown.

The next level up are 'medium response-forces' (MR-forces). These require your partner to either recognise the danger, and know an appropriate defence, or (if your partner is particularly skilled), they may just be able to solve the problem 'on the spot'; MR-forces give partner at least 2 beats of time to notice and rectify the incoming nastiness.

More difficult still, are 'hard response-forces' (HR-forces). These are similar to MR-forces, except partner has to respond immediately to the impending threat (ie on the very next throw). This is difficult even if they *do* know the defence, and virtually impossible otherwise! It is therefore a good idea to warn partner a few throws in advance.

Finally, if you're feeling particularly mischievous, you can throw a 'crash' at them. Unless your partner is psychic, this will inevitably cause their side of the pattern to meet its doom.

Note then, that unlike those of the previous section, all the tricks here are 'PT-variant'. Here are some examples, using the 7 ball 2 count <J1: **{** 4p 3 **}**, J2: **{** 3 4p **}** (J1 and J2 both pass with their right hands)> as the base pattern. In these examples, J1 is the 'deviator', and J2 is the 'responder'. The throws which differ from those of the base pattern are underlined.

SR-forcing:

a) J1: **{** 4 4p 4p 3 **}**
J2: **{** 3 4p 2 4p **}**

b) J1: **{** 5xp 3 4p 3 **}**
J2: **{** 3 4p 2 4p **}**

c) J1: **{** 5xp 5xp 2 3 **}**
J2: **{** 3 4p 2 4p **}**

d) J1: **{** 6p 4p 4 1 **}**
J2: **{** 3 4p 2 4p **}**

e) J1: **{** 4 4p 4 4p 4p 3 **}**
J2: **{** 3 4p 2 4p 2 4p **}**

f) J1: **{** 6p 6p 2 3 4p 3 **}**
J2: **{** 3 4p 3 4p 0 4p **}**

g) J1: **{** 4p 6p 6p 1 2 3 **}**
J2: **{** 3 4p 3 4p 2 4p **}**

h) J1: **{** 5xp 3 6p 4p 4 1 **}**
J2: **{** 3 4p 2 4p 2 4p **}**

MR-forcing: (the throw (by J1) which removes the possibility of a soft-response (from J2) is in italics)

i) J1: **{** 4p 5xp *5xp* 1 2 3 **}**
J2: **{** 3 4p 3 4p 4 4p **}**

j) J1: **{** 6p 6p 2 5xp *5xp* 1 4p 1 **}**
J2: **{** 3 4p 3 4p 0 4p 4p 4p **}**

k) J1: { 4p 6p *4p* 1 4 1 }
J2: { 3 4p 3 4p 4 4p }

l) J1: { 4p 6p *4p* 1 4p 1 }
J2: { 3 4p 3 4p 4p 4p }

m) J1: { 6p 6p 6p *6p* 0 1 2 3 }
J2: { 3 4p 3 4p 0 4p 4 4p }

(Note the difference between k) and l) – partners need to agree in advance which one to do, or else accept a 50% crash probability!)

HR-forcing:

n) J1: { 4p 4p 4p 1 }
J2: { 3 4p 4p 4p }

o) J1: { 4p 4p 5xp 1 4p 3 }
J2: { 3 4p 4p 4p 2 4p }

p) J1: { 4p 4p 5xp 1 4p 2 }
J2: { 3 4p 4p 5xp 2 4p }

Crashes:

q) Into (2-height) 1 count (crash avoidable): J1: { 3xp 3xp }
J2: { 4p 4p }

Incidentally, if you (J1) want to give a signal in advance, to tell your partner (J2) that you want to go into 1 count, use something like:

J1: { 4p 5xp 2 5xp 2 3 3xp 3xp ... }
J2: { 3 4p 3 4p 3 4p 4p 4p ... }

or

J1: { 4p 5xp 2 5xp 2 4p 4p ... }
J2: { 3 4p 3 4p 3 3xp 3xp ... }

Unavoidable crashes:

r) J1: { 4p 3xp }
J1: { - - }

s) J1: { 4p 5xp 4p 3 }
J2: { <Picks a ball up> }

t) J1: { 6xp 5p 4xp 3p }
J2: { "Yes, very funny..." }

r), s) and t) are only useful either as a practical joke, or to find out just how good your partner really is.

There are also many potential PT-variant moves for other base patterns. Here I have only dealt with the 2 person 7 ball 2 count to any extent. If you want SS tricks for any other passing pattern, then try to get to grips with the platform SS design method and invent your own, or else check out the various web-pages devoted to them.

SUICIDES

Instead of putting your partner's pattern at risk of collapse by throwing a response-forcing combo, you can risk your own pattern, by throwing a 'suicide'. To rescue you, your partner has to perform an appropriate 'saving sequence'. As an example, consider how J1 could go from the 7 ball 2 count, into a 5 ball cascade (with J2 holding the other 2 balls) and back:

J1: { 4p 3 4 5 ***5*** 5 ... 4p 4 4p 3 ... }
J2: { 3 4p 3 4p 2 ***2*** ... 2 2 3 4p ... }

- this is a suicide (on the part of J1), because if J2 throws the usual '4p' instead of (holding) the '***2***', then J1 will not be able to deal with it, as it will clash with J1's '***5***'.

In general, in any 2 count pattern (with RH passing), all self throws to the left hand (ie straight selfs from the left hand or crossing selfs from the right) are suicide throws. Note that (in the 7 ball 2 count) a left hand '4' guarantees a crash, as partner has no time to see it, having simultaneously thrown a '4p' to this hand.

Here are some more (for the 7 ball 2 count):

a) J1: { 4p 6 4p 1 }
J2: { 3 4p 3 3xp }

b) J1: { 4p 5xp 5 1 }
J2: { 3 4p 3 3xp }

c) J1: { 7 3xp 4p 1 2 3xp }
J2: { 3 4p 3 5xp 3 4p }

d) J1: { 4p 6 4p 4p 0 1 }
J2: { 3 4p 3 5xp 4 4p }

e) J1: { 6p 6 3xp 4p 0 1 }
J2: { 3 4p 2 5xp 4 4p }

Note: d) and e) put both ends of the pattern in danger.

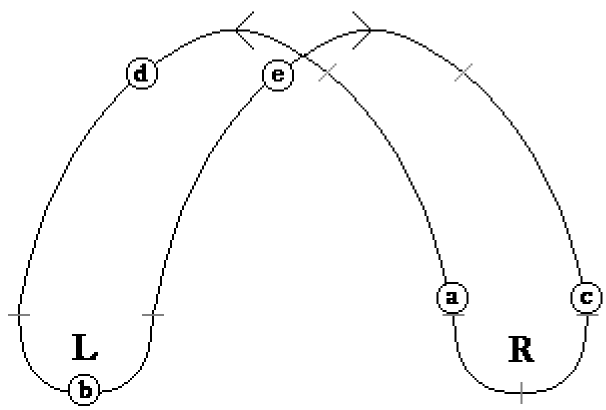
STEALING

(M)

This branch of passing should really be called ‘borrowing’, as very few jugglers who ‘steal’ a 5 ball cascade actually attempt to run off with it. Anyway, the idea is that one person juggles the pattern, and another person meddles with it in some way. There are many possible things one can do after removing a ball from a juggling pattern, so to simplify the situation, I will assume that we are only interested in when balls are taken out of the pattern, and when they are returned to it.

We will use siteswap and state notation to analyse the problem. Let’s assume J1 is juggling a 5 ball cascade (about to make a right hand throw), and J2 is the ‘thief’ (ideally, standing on a chair behind J1; alternatively J1 can kneel down). We’ll name the balls: ‘a’, ‘b‘, ‘c’, ‘d’ and ‘e’, and use these names instead of the usual ‘1’s (and ‘-‘s instead of ‘0’s) in the state string.

Obviously, putting balls back into a 5 ball cascade requires precise timing. We will therefore make the following assumptions. The pattern is juggled using 2 beat holdtimes, and when J2 steals a ball, they take it shortly after it reaches its highest point and starts to come down towards J1’s hand (note: initially, ‘d’ is in 1 of the 2 such positions). J2 also returns balls from these places. Here are the locations of the balls initially:



Let time = 0. We start the pattern: (note the ‘stealing/dropping position’ (position 4) is in bold)

Time:	State:					J1...
0	a(R)	b(L)	c(R)	**d(L)**	e(R)	throws ‘a’ from R to L
1	b(L)	c(R)	d(L)	**e(R)**	a(L)	throws ‘b’ from L to R
2	c(R)	d(L)	e(R)	**a(L)**	b(R)	throws ‘c’ from R to L

At time 2, ‘a’ is in the ‘stealable position’, so let’s suppose J2 swipes it at this time. We remove ‘a’ and continue:

Time:	State:					J1...
2	c(R)	d(L)	e(R)	**- L -**	b(R)	throws ‘c’ from R to L
3	d(L)	e(R)	- L -	**b(R)**	c(L)	throws ‘d’ from L to R
4	e(R)	- L -	b(R)	**c(L)**	d(R)	throws ‘e’ from R to L
5	- L -	b(R)	c(L)	**d(R)**	e(L)	Has a gap in L
6	b(R)	c(L)	d(R)	**e(L)**	- R -	throws ‘b’ from R to L
7	c(L)	d(R)	e(L)	**- R -**	b(L)	throws ‘c’ from L to R

At time 7, we see that there is a (right hand) gap at the dropping position, so J2 can safely drop ‘a’ *then* (into J1’s right hand). As ‘a’ was stolen at time 2 (as it headed for J1’s left hand), and dropped at time 7 (over J1’s right hand), we have the conclusion that stolen balls can be dropped 5 throws later, into a gap which occurs on the other side of the pattern.

In fact, the above argument can be rephrased as 55550 !

A minor point, is that the drop should be made exactly 5 beats after the steal, only when the ball is returned to the pattern with exactly the same vertical velocity as when it was removed; if the ball is stolen when moving downwards, but dropped back into the pattern with no initial downward velocity, then the ball should be dropped slightly earlier.

Examining 55550 (or continuing with our ‘state table’) also tells us that we could drop a ball 10 (or any multiple of 10) beats later on the same side of the pattern as we stole it from. Note also, that after stealing a ball, instead of simply waiting for 10 beats and dropping the ball back in, we could (say) wait for 8 beats, and then throw what feels like a ‘4’ (so it spends 2 beats in the air) – this should then land at about the right time.

The most interesting kinds of stealing involves the removal of more than 1 ball, and a bit of thought allows us to apply the above conclusions to this situation also. For example if two consecutive balls are stolen from J1’s right to left parabola, then as the 2^{nd} ball is stolen 2 beats after the 1^{st}, J2 can drop one ball 5 – 2 (= 3) beats after stealing the 2^{nd} (again on the opposite side), and then drop the other ball 2 beats later (from the same place). For those who have tried stealing 3 consecutive balls from the same side, it becomes obvious why there is so little time: you steal 1 ball; 2 beats later you steal the 2^{nd}; 2 beats later the 3^{rd}; but now you only have one beat left to move your stolen balls to the other side of the pattern and start dropping them as the 1^{st} gap appears. Incidentally, after taking 2 balls in this way, J1 is left juggling 50505. If 3 balls are stolen, the 5 ball carcass is 50500. In fact J2 can even take 4 or 5 balls in this manner, but clearly the dropping needs to start before the last balls are stolen, unless another gap is allowed to reach one of J1’s hands.

Another form of stealing is when two (or more) consecutive balls are taken (ie taking 1 which is soon to land in J1’s right hand, then 1 beat later, taking a ball aiming for J1’s left), leaving J1 juggling 55500. If these balls are taken by J2’s right and left hands respectively (assume J2 is standing behind J1), then if J2 wants to return the balls at the earliest opportunity, they need to wait 4 beats; then either drop from left hand, then (1 beat later) drop from right; or cross their hands and drop from right hand (on the left) and (1 beat later) drop from left hand (on the right). It should be fairly clear how to generalise this to returning more than 2 consecutively stolen balls.

‘Throwing the gap’ is not the only possibility when you have a ball swiped – you can ‘hold the gap’ instead; in other words go into 552 instead of 55550. Of course it helps if you tell the person doing the stealing that this is what you are going to do. If J1 goes into 552 after J2 steals a ball, then J2 must drop the ball 3 (instead of 5) beats later (again on the other side of the pattern); alternatively 9, 15, etc beats later, or 6, 12, 18 etc. beats later on the same side of the pattern as where the ball was stolen from.

Another alternative is for J1 to ‘spring’ the gap, ie go into 5551, in which case J2 should drop the ball 4 beats later on the same side of the pattern. There are many other SSs which you can go into after balls are stolen – if you have a patient J2, then experiment!

Finally, a bit of thought allows us to work out what happens when other ‘SS patterns’ are stolen from. If J1 is juggling 534 for example, and J2 steals a ‘5’ heading for J1’s left hand, then this ball can be returned 4 beats later to the left hand, 9 beats later to the right, 13 to the right, 18 to the left, and so on. These numbers are calculated by considering when (and where) the stolen ball would have next landed, had it not been stolen. To work them out, note that ‘5’s are next thrown as ‘4’s, and so a stolen ‘5’ should be returned 4 beats later (to make it land when the ‘4’ would have landed). The next gap would be when the ‘5’ (which the ‘4’ becomes) lands in the other hand (9 (= 4 + 5) beats later). And so on. All this is assuming similar holdtimes are used in-between each throw; and when dropped by J2, balls spend the same amount of time in the air (before being caught by J1) as the extra time it would have taken for the ball to land in J1’s hand (on its original journey) if it hadn’t been stolen. Phew!

7) MISCELLANEOUS

BALLS

(A)

The type of balls you juggle with can make a big difference. Different kinds have different positive and negative attributes. These include: grip, give, weight, size, and visual impact. How important some of these aspects are, depends on the type of juggling you want to do, as well as personal preference. However, with most of them, there is a balance to be found, as illustrated by the table below, which describes the problems encountered with balls at either end of the scales.

Aspect	If too little	If too much
Grip	Balls will often slip out of your hand when you throw or catch	Loss in throw accuracy, as the ball 'sticks' to your hand when you release it Balls get dirty or sandy easily.
Give	Bounces out of your hand on catch. Rolls away on hard surfaces. Mid-air collisions are often terminal.	Loss in throw accuracy, as the shape of the ball changes every time you catch it
Weight	More difficult to throw accurately. Bounces out of your hand on catch. Wind can blow balls off course.	Muscles get tired too quickly. High throws difficult to make. Hard to juggle fast enough.
Size	Inaccurate throws are more common. Balls can slip between fingers on catch.	Balls collide in the air due to lack of room. Cannot hold many balls in hands.
Visual Impact	Inappropriate for performance. Can't see balls against the background.	High risk of theft.

Grip: Cloth is nice material for grip purposes. Suede is particularly pleasant, although it does start to flake off after a couple of years of use. The plastic-type coatings commonly found on cheap balls are decidedly too sticky for high-skill work, whilst smooth solid balls can become very slippy when your hands start to sweat.

Give: Lentils, seed (preferably sterilised), rice, shot, and small plastic pellets are all good fillings for balls if you want a fair amount of give. How compactly they are filled is also crucial. I personally prefer balls which can be squashed quite a bit, but equally, some like very little give, such as with hard silicone.

Weight: If you want to build up your muscles, or improve your endurance, then 400+ gram balls are in order. For normal practising purposes though, most jugglers prefer balls weighing somewhere between 50 and 200 grams, depending on the number of balls in their pattern, and the stamina of their arms. Generally, balls weighing 120 grams or more are desirable when juggling 4 (or less). With 5, 80 to 150 grams seems to give the optimum balance between endurance and accuracy. With 7, lighter balls are definitely advantageous; 50 to 90 grams is recommended. With 9 or more, 35 to 80 gram balls will probably afford the best chance of achieving decent runs. If you get to this stage though, taking your watch off could make a difference.

Size: Again dependent on how big a pattern you want to juggle. Most jugglers prefer balls between 2½ and 3 inches (6 to 8 cm) in diameter, and use these same balls for all their patterns. In fact, when juggling 6 or more, smaller balls can help considerably, as they greatly reduce the likelihood of collisions. I personally find 4 to 5½ cm (about 1½ - 2 inch) balls preferable for high numbers.

Visual Impact: White, bright, or multicoloured balls are recommended, both for performance and general juggling. Many-a-time I've thrown some dark coloured balls up, only to lose them against the ceiling. For ultimate visual impact, glow-in-the-dark balls are a must, which are now available from several juggling companies.

MOTIVATION

This goes back to what I said in the introduction about juggling being fun. Well, to some people it is, and (bizarrely) to some people, it isn't – we all like different things. In general though, the more you enjoy something, the more motivated you are to do it. You will then practice more, and therefore get better at it. Juggling is no exception – if you love it, you'll probably become an expert.

Now there's nothing better for boosting your motivation, than seeing someone else perform a gorgeous trick or pattern. As I said earlier, it was the 'Tomorrow's World' TV program back in 1993 which made me decide to get seriously into juggling. In case you haven't seen it, a chap called Mike Day was juggling the synchronous pattern (6x,4)*, with white balls against a black background. At the time, I could only just manage a shaky 4 ball fountain, and I couldn't work out *what* was going on in this weird 5 ball pattern. I remember that it seemed to look like some kind of multiplex pattern; but most of all, I remember sitting there with my mouth open, thinking, 'This is the greatest thing I have ever seen anyone do'. I knew I had to learn to juggle that pattern, or die trying.

Even after you learn to juggle proficiently, motivation levels can still go up and down. Discovering siteswap notation for example, can be very satisfying for jugglers who enjoy mathematics, allowing them to understand and learn ever more complex patterns, or attempt ever higher throw values.

The more competitive-minded jugglers can be spurred into practice by watching someone pull off a feat of skill which they themselves might just be able to do.

Numbers jugglers get excited when they manage to flash one more ball, or pull off their longest run.

Other turn-ons include buying a new set of balls, finding the joys of passing patterns, joining a local juggling club, or going to a juggling convention.

There are also reasons why motivation levels can drop. For example, if you've been working on a pattern for months without any apparent progress, it can be very disheartening. There is good evidence however, that even though you seem to have reached a plateau, your subconscious *is* actually getting closer to 'solving' your pattern.

NUMBERS JUGGLING

Some jugglers relish the challenge of keeping ever higher numbers of balls in the air. Not surprisingly though, the more balls you are juggling, the harder it is to do 'tricks' and pattern variations. With 2 hands, 3 balls is the minimum number you can 'properly' air-juggle with (I expect some will disagree), and therefore allows the maximum amount of creative freedom; there are vastly more variations performed with 3 balls, than with all higher numbers put together.

Initially with 4 balls, you struggle just to keep the basic fountain going – it may take months of practice to get it comfortable. With time though (and luck), it is possible to do fairly complex tricks, such as behind the back throws, Mills Mess, Rubenstein's Revenge, and so on. However, the slow graceful elegance, which was possible with 3 balls, is no longer attainable. Also, being about twice as many balls in the air, much more effort is required just to keep the balls dancing, so there is little in reserve for smoothing out inaccuracies which (now more commonly) occur.

5 is considered by most jugglers to be very much more difficult than 4. It often takes years to get confident with the 5 ball cascade. The amount of freedom afforded whilst juggling 5, is greatly reduced (ie you can't just throw balls whenever you feel like it), and throws need to be quite a lot higher, faster, and more accurate than with 4. Again, the number of realistic, 'non-siteswap' patterns is slashed. A few jugglers learn the '5 ball Mills Mess', but not without a lot of dedication. In fact, getting the basic 5 ball cascade 'solid' is quite an achievement – there may well be less than a thousand 5 ball jugglers in Britain. If you *do* get to the stage of being able to hold a 5 ball cascade together, you will probably have a fair idea of how many more balls you have the potential to juggle, based on how much effort it took

to do 5. If you got there with less than 3 years of 5 hours-per-week practice, you may fancy your chances against the higher numbers....

The first 6 ball pattern I attempted, was a half-shower, involving 5 sellotaped-up plastic-coated monstrosities and a battered lime. The idea came from trying to do 5 balls, with my right hand throwing higher than my left (over the top of the pattern), and my hands throwing alternately (this was before I knew about SS notation). It soon dawned on me that there was a bit of a gap in the pattern. Immediately, I ran downstairs to hunt for an approximate sphere, and returned with the fruit item. I launched the objects as before, and to my delight, the pattern worked; I found myself juggling 6 balls for the first time (in the SS 75 half-shower). Many jugglers agree that this pattern is the easiest way to handle 6, as the fountain requires much more accurate throws, to prevent the 3 balls (in each hand) from hitting each other. Indeed, not many jugglers spend a lot of time on the 6 ball fountain before trying 7.

The 7 ball cascade can take forever to learn, and very few jugglers get the pattern solid. The sheer number of balls in the pattern makes it a bewildering sight when you first attempt it, even though you know (in theory) exactly what to do. It can be weeks before you even 'flash' the pattern (throw and catch all 7 balls); starting with 4 balls in one hand is half of the problem. Having said all this, if you really want to learn 7, then you will probably get there eventually.

7 ball patterns are almost exclusively 'SS patterns' – that is, they play with the height (whilst keeping the rhythm constant), and sometimes use synchronous or multiplex throws, but rarely do the other pattern-features (such as throw site/position/type) vary from the basic. If and when you manage to get somewhere with 7, you might even think of attempting more. There are a handful of jugglers who can maintain a 9 ball cascade for a few seconds, but as yet, no-one has 'qualified' 11 (made 22 consecutive catches). If you are that way inclined, why not see how close you can get?

Finally, for those who like to have some idea of how remarkable their numbers juggling skill is, below are some probable 'ball-park' figures of the number of people in the world who have flashed different numbers of balls. These figures are based partly on my own beliefs, and partly on the figures put forward by someone on the rec.juggling newsgroup, who I believe wants to remain anonymous (although I think he was fairly accurate):

Balls:	3	4	5	6	7	8	9	10	11	12
People:	20M	1M	100K	20K	5K	1K	400	80	20	5

(M = 1,000,000 ; K = 1,000).

In 2004 I changed these figures somewhat from my original estimates – certainly more people have flashed higher numbers than in 2000, but I also feel my estimates of the quantity of people having flashed the lower numbers were rather inflated originally. I will not make any claims as to the accuracy of the above figures however.

Having worked on 13 myself for about 3 years now, I am convinced that it *is* humanly possible to flash 13 balls, but even using 50g beanbags (as I do), the demands placed on the arms, in terms of speed, power and accuracy to do this, are phenomenal.

PRACTISING

Practising requires a combination of enthusiasm and patience. If you have both in abundance, you will have no problems putting in the hours needed to reach the higher levels of skill. If you are not so fortunate, then you may have to practice sometimes when you don't really feel like it. Anyway, in this section I will try to give some tips for making the most of your practice time:

Firstly, I've found that juggling with different types of balls (even at the same time) can be very useful in a few ways:

1) It enables you to get a good idea of which type you most prefer.

2) It allows you to improve different skills – for example, juggling larger balls will improve your endurance, and your skill in dealing with collisions, whilst small balls will allow you to carry on for longer (with bigger patterns) and improve both your confidence and accuracy.

3) It gives you a slightly better understanding of the available juggling-space in front of you, and how to use it most effectively.

Secondly, it is important to remember that your subconscious mind tries to 'solve' your juggling problems long after you stop practising. This means that spending 10 minutes on a trick or pattern, resting, then trying again later that day or week (say for another 10 minutes), can be as beneficial as an hour's practice all in one go. The conclusion is, that it doesn't matter too much if you have several days of rest in-between practice sessions. A post I saw on the 'rec.juggling' newsgroup (due to Iain Duncan, June 5th 2000) addresses this issue, and is worth including here (though it is rather technical):

"Why does our skill seem to improve in-between practice sessions?"

"Neurologists call this the reminiscence effect, and it has been studied quite a bit in sport science in many other sports. It is definitely real, though of course no one knows exactly how it works. The prevailing theory is that new training of a motor activity sends neurons down new synapse pathways in your brain. As you do the motion more and more, these pathways become more deeply ingrained, and are called engrams. It is when they are much easier for your brain to fire than other related pathways that a trick becomes automatic. Hence the difficulty of breaking bad habits in any training. One is trying to ingrain a different but similar synapse pathway while it is a already easier for your neurons to fire down the old engram. Also, you can never erase learnt engrams (nothing to do with the scientologists misuse of the same word); you can only make the newer ones stronger. As to the reminiscence effect, the prevailing theory is that after a certain density of training is reached, the development of engrams cannot keep up with the training. However, when training is stopped, the brain continues to mull over these new engrams. A week, two days, whatever later, when training is resumed, the engrams are stronger as they were still being developed during rest. This is thought to be the same process that goes on during sports visualization, and there have been a number of double blind studies showing that neural pathways actually get further ingrained in the same way during visualization as during actual practice. At any rate, at the elite level in almost every sport, coaches have their athletes use visualization a lot, and deliberately schedule total rest as part of the training cycle, often as much as 6 weeks of scheduled rest (not injury time) in the training year. Personally, I've noticed the effect most when I have been practising something more than I'm used to for a while, and then take a lay off for whatever reason."

Here's one from Anthony Gatto: balance things on your chin whilst juggling. I can't really emphasise this, as I am hopeless at balancing, but he says it allows him to see where the middle of the pattern is, thereby helping him to keep his pattern in the same place, and make more accurate throws.

A tip for the numbers juggler: if you are trying to flash 7 (say), for the first time, then start with 2 balls, and throw them as you would the first 2 throws in a 7 ball cascade (ie do 7700000), and catch them. Try it with your right hand throwing first, then the other way round. If you don't drop, then get 3 balls and do the first 3 throws (ie 7770000), and so on, until you are attempting to flash the 7. By doing this, your brain can work on dealing with 1 extra ball at a time, and will be able to build up a model of where each ball goes, piece by piece. Similarly, if you are learning to juggle a SS, try increasing the length of your run by 1 throw on each successive attempt, until you can throw the whole sequence twice round.

If you want to improve your 4 ball fountain, it will almost certainly help to attempt 5. Even though you may catch none of the balls, have about 10 tries at flashing a 5 ball cascade. After this, have a rest for a minute, then try your 4 ball fountain again, and you should find that it feels more relaxed than before. In simple terms, this is because *any* experience with more difficult patterns 'stretches' your 'learning muscle'. This technique works for any number of balls.

Sometimes, when you watch someone practising a trick, they make the same kind of mistake again and again (eg their 2nd throw from the left hand goes too far forward etc). If this happens, I strongly advise you to approach them, and ask whether they are aware of it, because there is a fair chance that they aren't, and it could save them hours of practice. Obviously, you can't rely on others to just come up to you and tell you (especially if they can't do the trick themselves), so if you're getting stuck on a particular pattern, it is a good idea to ask someone (preferably a juggler) to watch and point out any specific problems. Failing this, use a video camera, and analyse the pattern yourself. If you can consciously see what the problem is, your subconscious doesn't have to spend hours working it out by trial and error.

A more general tip: think positive and talk to your body. A positive attitude and positive words are very important in juggling (and life in general) – if during your pattern you start to think, "There are at least 5 balls in the air – I can't possibly deal with that many", then your 7 ball cascade will collapse. Ideally, you should (make yourself) think, 'I can do this'. If you find it impossible to be so optimistic, then just try to not think about anything. Also, I am pretty certain that when you say something, what you say alters or reinforces what your brain thinks, and your body responds appropriately. This means that if you pull-off your longest ever run of the 4 ball Mills Mess, it might be a good idea to verbally congratulate your brain and hands for doing so well. This may encourage them to remember how they did it.

From time to time, people post tips on practising technique to the rec.juggling newsgroup. These can be read at the Internet Juggling Database (IJDb): www.jugglingdb.com, which is by far the best website for juggling-related stuff.

WORLD RECORDS

As you may have noticed, the Guinness Book of Records now seems to be more interested in tracking the longest cow pat throw, than the most number of balls, rings or clubs juggled. Here then, are the (unofficial) records (as far as I am aware) for the number of objects flashed and qualified, in solo and 2-person passing patterns (usually ultimates), with balls, rings and clubs:

	Objects	Flashed	Qualified
Solo:			
	Balls	12	10
	Rings	14	10
	Clubs	9	7
Passing:			
	Balls	18	15
	Rings	16	14
	Clubs	13	11

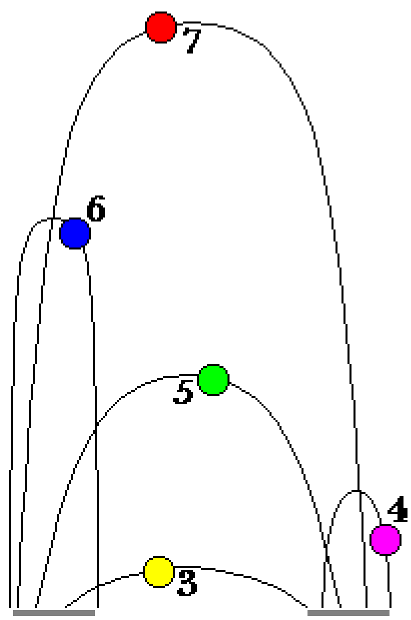
A 'flash' is when each object is thrown and caught at least once. 'Qualifying' is when each object is thrown and caught at least twice (at least, these are the definitions as I understand them).

There is every chance that one or more of the above records has been surpassed, as we seem to be in a time when juggling records are being improved upon every year or so.

8) SITESWAP APPENDIX

This chapter contains some of the more interesting siteswaps. The lists are by no means exhaustive, but they do provide a wide range of different-feeling patterns. For concision, throw-values will be written next to each other (without spaces or commas), so the SS 5 3 1 for example, will be written 531; any 2-digit throw values will be separated off by a comma, eg 10,47531. I have also underlined the throws which occur whilst the pattern is in ‘ground state’ – these are the places where the pattern can be entered/exited from/to the standard cascade or fountain. For example, with 51414, the last ‘4’ is underlined because this is where the SS ‘3’ can be inserted (as many rounds of it as you like)– ie the sequence 51414 51414... 5141 333...3 4 51414 51414... is jugglable.

As a visual-aid, below are the relative heights to which the SS values 3 to 7 should be thrown, when juggled with 1.6 beat holdtimes – which is about what most jugglers use:



Remember that ‘2’s do not have to be thrown at all, ‘1’s are ‘zipped’ across to the other hand (in any pattern containing ‘3’s or higher), and ‘0’s should involve an empty hand.

2 BALLS

(J)

Siteswapping with 1 ball is rather trivial, so we’ll start with 2. Although the possibilities are still very limited, 2 ball siteswaps do provide a useful introduction for those who are new to the concept of juggling a string of numbers.

Period 2: 31 40 (31 = Shower; 40 = 2 in 1 hand)

Period 3: 312 330 411 501

Period 4: 3302 4013 4112 4130 5111 6011

Period 5: 40123

3 BALLS

3 balls is just enough for some reasonably complicated SSs, of which ‘1’s are ‘bread and butter’. ‘2’s can also be used to great artistic effect, as they allow a bit of time to wave a ball around randomly, or take a bite out of an apple. ‘0’s (unless immediately followed by a ‘1’) tend to feel rather wasteful, so try putting the free hand on your hip to keep it active.

Period 2:	51	60	(51 = Shower, 60 = 3 in 1 hand)					
Period 3:	423	441	531	504	612	711	801	
Period 4:	4413	5124	5304	5340	5511	6015	6051	6231
	6312	6330	6411	7041	7131	7401	8013	8040
	8130	9111						
Period 5:	51234	51414	52413	52440	52512	53034	53403	55014
	61251	61305	61314	61350	63051	63141	63303	63501
	64005	64014	64050	64140	64500	66300	70161	70251
	70305	70314	70350	70701	72330	73131	73302	73401
	74130	74400	75300	81312	81330	83031	84012	84030
	90141	90303	90501	91401				
Others:	525141	615600	713151	5505051	6050505	6131451	6161601	6316131
	8123601	71701701	530534034	7330730370330	714014714700	741701740041		

Multiplex: [4,3]0521 [5,4]0141 [4,3]041 [5,4]01521 [5,4]60021

Synchronous (Remember, ‘*’ means, ‘repeat on the other side.’):

(4x,2x) (4x,2x)(2,4) (4,2x)* (4x,2x)(4,2x)* (8,2x)(4,2x)(2x,0)*

Synch-multiplex:

([4x,4],0)(6,0)(2,2x) ([6x,6],0)(2x,0)(0,6x)(2,2x)

([4x,4],0)(6x,0)(2x,2)* ([6x,6],0)(2x,0)(0,6)(2x,2)*

4 BALLS

Siteswapping with 4 balls affords a vastly increased amount of creative scope. From the immensely versatile 534, to the ridiculous but strangely addictive 9313, the possibilities are endless. If you are new to 4 ball SSs, I recommend starting with 5344, 6334, or 6424.

The most common mistake jugglers make with 4+ ball SSs, is to throw their '3's too high – remember, '3's should be as low as you can possibly make them (preferably less than 10cm high).

Period 2: 53 71 80 (53 = Half-shower, 71 = Shower, 80 = 4 in 1 hand)

Period 3: 534 552 615 633 642 660 714 723
741 831 912 10,11 11,01

Period 4: 5524 5551 6055 6235 6415 6451 6631 7045
7063 7126 7135 7333 7405 7441 8134 8170
8233 8413 9124 9151 9241 9304 9313 9601

Period 5: 53444 55514 61355 62345 62525 62561 63353 63524
63551 63623 64055 64145 64163 64253 64505 64613
66125 66161 66305 66314 66350 70166 70256 70355
70364 70616 70625 70661 70706 72335 72461 73136
73406 73424 72416 72425 73451 73631 74135 74162
74234 74405 74450 74612 74630 75161 75251 75305
75314 75350 75620 75701 77231 77312 77330 80345
80516 80525 80561 80723 80741 81236 81317 81335
81416 81425 81461 81731 81812 83333 84017 84035
84440 84512 85016 85061 85241 86420 90146 90173
90506 90551 90641 91334 91424 91451 91631 91901
92333 94034 94133 95141 95501 96131 96401 10,1612

Others: 661515 663504 731571 737313 747141 751515 915171 11,17131
6155155 6262525 6461641 6605155 6615163 7123456 7142635 7161616
7362514 7415263 7427242 7471414 71334455 71615156 11,131517191 8441841481441

Multiplex: [43]1 [43]23 [43]14 [53]22 [54]21 [65]01 [43]522 [53]323
[53]521 [54]124 [54]241 [43]5323 [43]5521 [43]6421 [53]3423 [54]1424
[54]5123 [54]6122 [64]1324 [64]2323 [64]4123 [65]3123 [74]2421 [54]24522
[54]51423 [54]51441 [74]41323

Synchronous: (4,4) (4x,4x) (6x,2x) (6x,2x)(6,2) (6x,2x)(2,6) (6,4)(2,4)
(6,4)(4x,2x) (6,4x)(4x,2) (6,4x)(2x,4) (6x,4)(4,2x) (6x,4)(2,4x) (6x,4x)(4,2)
(6x,4x)(2x,4x) (8x,2x)(2,4) (8,2x)(2x,4) (8x,2x)(4x,2x) (8x,2x)(2x,4x)(2x,6x)
(6x,2x)* (6,2x)(6,2x)* (6,4)(4x,2x)* (6,4)(2,4)* (6,4x)(4x,2)* (6,4x)(2x,4)*
(6x,2)(6,2x)* (6x,2)(2x,6)* (6x,4)(4,2x)* (6x,4)(2,4x)* (6x,4x)(4,2)* (6x,4x)(2x,4x)*
(8,2x)(4,2x)* (8,2x)(2,4x)* (8x,2x)(2x,4x)* (8,2x)(2x,4)(6x,2x)*

Synch-multiplex: ([4x,4],2)(4,2x) ([4x,4],2)(2,4x) ([4x,4],2x)(4x,2x) ([4x,4],2x)(2,4)
([6x,6],0)(2,2x) ([4x,4],2)(6,4)(2,2x) ([4x,4],2)(4x,6x)(2,2x)
([4x,4],2)(4x,2)* ([4x,4],2)(2x,4)* ([4x,4],2x)(4,2)* ([4x,4],2x)(2x,4x)*
([6x,6],0)(2x,2)* ([4x,4],2)(4,6)(2x,2)* ([4x,4],2)(6x,4x)(2x,2)*

5 BALLS

5 balls is perhaps the most satisfying number for siteswapping with. When juggled well, a complex 5 ball SS is a glorious fusion of art and mathematics. Throwing a 744 from a running 5 ball cascade is one of the best ways to get started. For the more advanced jugglers, try throwing balls high out of the pattern, then doing a 3 or 4 ball trick underneath, before going back into a 5 ball cascade as the high-throws land.

Period 2: 64 73 91

Period 3: 645 663 726 744 753 771 825 834
861 915 933 942 12,12

Period 4: 7166 7346 7445 7463 7526 7535 7562 7571
7733 8156 8174 8246 8273 8417 8516 8633
8642 8813 9128 9155 9164 9245 9281 9344
9353 9515 9524 9551 9641 9713 11,171 11,441

Period 5: 66661 72466 73456 73636 74635 74734 75364 75616
75625 75661 75751 77416 77425 77461 77731 81277
81466 81475 81727 81772 81817 83446 83833 84517
84733 84742 85516 85561 85741 86416 86425 86461
86731 88441 90808 91456 91474 91627 91672 91681
92491 92527 92581 92923 94444 94552 94642 95191
95551 95641 96181 96451 96901 97531 99133 10,1617

Others: 746625 757173 773355 824466 847182 935373 11,1,11,151 13,19151
7266716 11,633633 74467561 77461564 82445566 83571637 85145566 85716814
912345678 10,4448334 13,1517191,11,1 11,4,10,33333 11,444,11,333444 9552952592552

Multiplex: $[3,2_T]$ [5,4]1 [4,2]27 [4,3]26 [5,3]25 [5,4][2,2]2 [5,4]24 [6,2]25
[6,4]23 [6,5]22 [7,2]24 [7,5]21 [4,3]526 [5,4]623 [6,3]425 [6,5]324
[6,5]621 [7,4]126 [7,6]421 [6,5]6125 [7,4]4424 [7,6]4125 [7,6]3671424 [9,6]3344424
[5,4,3]24[2,2]3 [6,5,4]224[2,2]3 [7,6,4]4124[2,2]3 [7,6,5]8171671424[2,2]3 [9,8,7]3131324[2,2]3

Synchronous: (6,4) (6x,4x) (8x,2x) (6x,2x)(6,6) (6x,4)(6,4x) (6x,4x)(4,6)
(8,2x)(4x,6) (8,2x)(6x,4) (8,6)(2x,4x) (8,6x)(2x,4) (8x,2)(8,2x) (8x,2)(8,2x)
(8x,2x)(2,8) (8x,2x)(6,4) (8x,2x)(4x,6x) (8x,4x)(4,4) (8x,6)(4,2x) (8x,6x)(4x,2x)
(10x,2x)(6x,2x) (10x,2x)(6x,2x)(8x,2x) (10x,2x)(10x,2x)(4x,2x) (12x,2x)(8x,2x)(4x,2x)
(6,4x)* (6x,4)* (8,2x)* (6,4x)(6,4)* (6,6)(6,2x)* (6x,4)(4,6)*
(6x,4)(6x,4x)* (6x,4x)(6,4x)* (8,2x)(4,6)* (8,2x)(6x,4x)* (8,2)(8,2x)* (8,4x)(4,4)*
(8,6)(4,2x)* (8,6x)(4x,2x)* (8x,2)(8,2)* (8x,2x)(2x,8)* (8x,2x)(4,6x)* (8x,2x)(8x,2)*
(8x,2x)(6,4x)* (8x,4)(4,4)* (8x,6)(2x,4x)* (8x,6x)(2x,4)*

Synch-multiplex: ([4x,4],6)(2,4x) ([4x,4],6x)(2,4) ([6,4],6)(2,2) ([6,4x],4x)(2,4) ([6,4x],6)(2,2x)
([6x,4],4)(2,4x) ([6x,4],6x)(2,2) ([6x,4x],2x)(2,6x) ([6x,6],4)(2,2x) ([6x,6],4x)(2,2)
([8x,6x],2x)(2,2x) ([6x,6],2x)(6x,2x)(2,6) ([6x,6],4)(6,2x)(2,4) ([8x,6]),4x)(2x,4)(2,4x)
([8x,8],4)(4,2x)(2x,6)(2,4x) ([8x,6x,4x],2x)(2,2x)([2,2],2x)
([4x,4],6)(4,2)* ([4x,4],6x)(4x,2)* ([6,4],4)(4x,2)* ([6,4x]),6)(2,2)* ([6x,4]),4)(4,2)*
([6x,4]),6)(2x,2)* ([6x,6],2)(2,[2,2])* ([6x,6]),4)(2,2)* ([6x,6],4x)(2x,2)* ([8,6]),2x)(2,2)*
([6x,6]),4)(6,2x)(4x,2)* ([6x,6]),4)(6x,2x)(4,2)* ([8,6]),4x)(2x,2)(4x,2)*

6 BALLS

With 6 balls or more, the effort required (just to keep the balls in the air) can quickly lead to muscle tiredness; which in turn leads to loss in throw-accuracy, making the necessarily high SS throws (‘7’s, ‘8’s, ‘9’s etc) very difficult to catch. The easiest ‘real’ 6 ball SS is probably 756, although throwing a 9555 out of a half-shower (75) is also quite feasible. With this many balls, patterns containing ‘3’s are hideously difficult, due to having to make them much lower than most of the other throws, as well as extremely fast and accurate. Also, some jugglers find multiplex patterns easier than ‘uniplex’ ones, as they can involve lower maximum heights.

Period 2: 75 84 93

Period 3: 756 774 837 855 864 882 918 936
945 963 972 990 10,44 11,61

Period 4: 7746 7773 8277 8457 8556 8574 8637 8646
8673 8682 8817 8844 8853 8880 9168 9267
9285 9348 9357 9384 9528 9555 9564 9591
9627 9645 9663 9681 9708 9717 9744 9753
9780 9915 9924 9951 9960 10,455 11,445 11,571

Period 5: 75666 75756 77475 77772 81777 84567 84747 85575
85746 85845 86277 86475 86727 86781 86817 86862
88446 88527 91677 92577 92928 94188 94584 94692
94944 95646 96456 96474 96627 96672 96681 96852
97581 99192 99444 99552 10,5555 10,8642 11,4555

Others: 884466 975375 959445 11,55555 11,97531 96256677 96617586 96827925

Multiplex: [5,4]27 [6,4]26 [6,5]25 [7,2]27 [7,5]24 [7,6]23 [4,3]827 [5,4]627
[6,4]626 [6,5]625 [6,5]724 [7,4]427 [7,4]526 [7,5]525 [7,6]425 [7,6]722
[8,7]423 [8,7]522 [5,4]6627 [6,5]6625 [7,5]5625 [7,6]4625 [8,7]4425 [7,4]56626
[7,6]46724 [5,4,3]625[2,2]7 [5,4,3]724[2,2]7 [6,5,4]625[2,2]4 [7,6,4]425[2,2]4
[7,6,5]225[2,2]5 [7,6,5]325[2,2]4

Synchronous: (6,6) (6x,6x) (8,4) (8x,4x) (10x,2x) (8,6)(6x,4x)
(8,6x)(2x,8) (8,6x)(4x,6) (8,6x)(6x,4) (8,8)(4,4) (8x,2x)(6,8) (8x,4x)(4,8)
(8x,4x)(8,4) (8x,6)(4,6x) (8x,6)(6,4x) (8x,6)(8,2x) (8x,6x)(4x,6x) (8x,6x)(6,4)
(8x,6x)(8x,2x) (8x,8x)(4x,4x) (10x,2x)(6,6) (12x,2x)(8x,2x) (12x,2x)(8x,2x)(10x,2x)
(8,4)* (8x,4x)* (10x,2x)* (8,4x)(8,4x)* (8,6)(6x,4x)* (8,6x)(4x,6)*
(8,6x)(6x,4)* (8,6x)(8,2x)* (8x,2x)(6,8)* (8x,4)(4x,8)* (8x,4)(8,4x)* (8x,4)(8x,4)*
(8x,6)(2x,8)* (8x,6)(4,6x)* (8x,6)(6,4x)* (8x,6x)(2x,8x)* (8x,6x)(4x,6x)* (8x,6x)(6,4)*

Synch-multiplex: ([6,4],6)(2,6) ([6x,4],4)(2,8x) ([6x,6],4)(2,6x) ([6x,6],4x)(2,6) ([8,4x],6)(2,4x)
([8,6],4)(2,4) ([8,6x],4)(2,4x) ([8x,6],4x)(2,4) ([8x,8],2)(2,4x) ([8x,8],2x)(2,4)
([6,4],[6,4])(2,2) ([6x,6,4],4)(2,6x)([2,2],4)
([6,4],6)(6,2)* ([6x,4],4)(8x,2)* ([6x,6],4)(6x,2)* ([6x,6],4x)(6,2)* ([8x,4x],6)(4,2)*
([8,6],4x)(4x,2)* ([8,6x],4x)(4,2)* ([8x,8],2)(4x,2)* ([8x,8],2x)(4,2)*
([6x,6,4],4)(6x,2)(4,[2,2])*

7 BALLS

Another problem which becomes particularly prevalent in SSs with this many balls, is having to get the different heights accurate enough so that the throw-rhythm remains sufficiently constant; make a throw slightly too high or too low (in relation to the other values), and you will find yourself with 2 balls landing too close together (in time) in the same hand, making them impossible to deal with. There are no easy 7 ball SSs, but the multiplex pattern [4,3] – juggled as ([6x,6],2)*, is amongst the easiest. If you have any success with this, then the 7 ball version of ‘Gatto’s Multiplex’: [7,6]26 is also worth a try. Whilst on the subject, the SS for Gatto’s ‘high throw’ (out of a 7 ball cascade, as seen on his ‘To be the best’ video), is 11,6666. Having the power in reserve to throw the 11, necessitates either big muscles or light balls, the latter being the easier option.

Period 2: 86 95 10,4 11,3 13,1

Period 3: 867 885 948 966 10,29 10,47 10,56

Period 4: 8677 8857 8884 9388 9568 9667 9685 9748
9757 9784 9793 9928 9955 9964 9991 10,666
10,864 11,449 11,566 11,575

Others: 11,6666 10,45689 10,69584 11,57595 12,77772 13,666666

Duplex: [6,5]28 [7,5]27 [7,6]26 [8,6]25 [8,7]24 [9,2]28 [9,8]22 [6,5]728
[7,6]726 [8,7]526 [8,6]626 [9,6]625 [9,8]524 [6,5][7,6]22 [7,6][7,4]22
[8,6][6,4]22 [8,7][5,4]22 [8,6][8,6]322 [9,8]59425

Triplex: [6,5,4][2,2]2 [7,6,4]726[2,2]6 [7,6,5]726[2,2]5 [7,6,5]727[2,2]4 [8,7,6]426[2,2]5
[9,8,7]4426[2,2]5 [5,4][7,6,5]22[2,2]

Quadruplex: [7,6,5,4]26[2,2]5[2,2,2]4 [7,6,5,4]27[2,2]4[2,2,2]4 [9,8,7,6]3327[2,2]4[2,2,2]4

Synchronous: (8,6) (8x,6x) (10,4) (10x,4x) (12x,2x) (8,8)(8,4)
(8x,4x)(8,8) (8x,6)(8,6x) (8x,6x)(6,8) (10,4)(6,8) (10,4x)(6x,8) (10,4x)(8x,6)
(10,6)(6,6) (10,8)(6,4) (10,6)(10x,2x) (10x,4x)(6x,8x) (10x,6x)(6,6)
(8,6x)* (8x,6)* (10,4x)* (10x,4)* (8,6x)(8,6)* (8,8)(8,4x)*
(8x,4)(8,8)* (8,6)(8x,6)* (8x,6)(8x,6x)* (8x,6x)(8,6x)* (10,4)(8x,6)* (10x,6)(6,6)*
(10x,8x)(4x,6)* (10x,8x)(6x,4)*

Synch-multiplex: ([6x,6],8x)(2,6) ([8,6],6)(2,6) ([8x,6],6)(2,6x) ([8x,8],6)(2,4x) ([8x,8],6x)(2,4)
([10x,8],4)(2,4x) ([8x,8],6)(8x,2x)(2,8x) ([8x,8],6)(4,8x)(2,6) ([8x,8],6x)(8x,2x)(2,8)
([8x,8,6],4)(2,4)([2,2],6x) ([8x,8,6],4x)(2,4)([2,2],6) ([8x,8,6],[4x,4])(2,4)([2,2],2)
([8x,8,6],[6,4])(2,2x)([2,2],2) ([8x,8,6],[6,4x])(2,2)([2,2],2)
([6x,6],2)* ([6x,6],8x)(6x,2)* ([8,6],6)(6x,2)* ([8x,6],6)(6,2)* ([8x,8],6)(4,2)*
([8x,8],6x)(4x,2)* ([10x,8],4x)(4x,2)* ([8x,8],6)(8,2x)(8x,2)* ([8x,8],6)(4,8x)(6x,2)*
([8x,8],6x)(8,2x)(8,2)* ([8x,8,6],4x)(4x,2)(6,[2,2])* ([8x,8,6],[4x,4])(4x,2)(2,[2,2])*
([8x,8,6],[6,4])(2,2)(2,[2,2])* ([8x,8,6],[6,4x])(2x,2)(2,[2,2])*

BOUNCE PATTERNS

(M)

If you can find a hard flat surface (such as in an airport), bouncing balls can be lots of fun to juggle with. Many SSs work well as bounce patterns, and are quite a bit easier than when juggled in the air, as SS values do not have to be thrown as high. But how high should throws be? Well, there are general rules-of-thumb which can help you to estimate: if you let the ball bounce once, then a SS value V can be thrown to the same height as if it were a V/2 in an air-pattern. So for example, 6b 33 feels like: 3sb 3 3 ('b' or 'b_1' means 'bounces once'). If you are going to let the ball bounce twice, then throw it to about the same height as a standard V/3 (eg in $12b_2$,$12b_2$, 3333333, the '12's can be thrown to '4' height).

So how do you choose which throws to bounce? Well it seems reasonable to bounce the higher values rather than the lower ones. Bouncing a '3' in a pattern where a higher value is not bounced, will mean having to throw the higher value stupidly high. More generally, for any value you choose to bounce, all higher values should also be bounced, unless you want to make life very difficult. In what follows, patterns which flout this rule will be ignored.

Bounced SSs can be split into 2 sets: those which feel vaguely similar to their air-equivalent, and those which do not. For example, 6b 33 is in the first category, as the highest valued throw (the '6') is still thrown at least as high as the '3's, even though it is allowed to bounce. Compare this with 6b 15: this feels like (has SS(As) values) 3sb 1 5, so the '6' is no longer thrown as high as the '5'. The pattern is difficult because you almost have to look up (to follow the '5') *and* down (to follow the '6b') at the same time, but it also feels very strange throwing a '6' lower than a '5'! In the table below, bounce patterns with this height-order changing property are classed as 'strange'.

The easiest way to bounce juggle 615, is to bounce the '5's as well – ie juggle 6b 1 5b, which will feel like 3sb,1,2½xb. Because the highest throw is now only like a '3s', it can be juggled very slowly.

Below is a table of suggestions for bounce-siteswapping:

Balls	Pattern	SS(As)	Strange
3	3b	1½xb	No
3	4b 4b 1	2b 2b 1	No
3	5b 3 1	2½xb 3 1	Yes
3	8b 0 1	4b 0 1	No
4	5b 5b 2	2½xb 2½xb 2	No
4	5b 3 4	2½xb 3 4	Yes
4	5b 3 4b	2½xb 3 2b	Yes
4	6b 3 3	3sb 3 3	No
4	6b 4 2	3sb 4 2	Yes
4	6b 4b 2	3sb 2b 2	No
4	6b 1 5	3sb 1 5	Yes
4	6b 1 5b	3sb 1 2½xb	No
4	7b 1 4	3½xb 1 4	Yes
4	7b 1 4b	3½xb 1 2b	No
5	6b 4 5b	3sb 4 2½xb	Yes
5	6b 6b 3	3sb 3sb 3	No
5	7b 4 4	3½xb 4 4	Yes
5	7b 7b 1	3½xb 3½xb 1	No
5	8b 4 4 4	4b 4 4 4	No
5	8b 8b 3 3 3	4b 4b 3 3 3	No
5	10b 10b 3 3 3 3 3	5sb 5sb 3 3 3 3 3	No
5	$10b_2$ $10b_2$ 3 3 3 3 3	$3sb_2$ $3sb_2$ 3 3 3 3 3	No
6	10b 5 5 5 5	5sb 5 5 5 5	No

See Charlie Dancey's EBJ for more bounce-ideas (eg Dyer's Straights, Orbit Bounce, Robot Bounce).

PASSING PATTERNS

(J)

Here are some more ideas for passing patterns. Most of these can be found in Charlie D's EBJ on the pages indicated (where they are described in words rather than numbers).

2 Person Patterns (Jugglers face each other, passes are tramlines unless otherwise indicated)

6 Balls:

4 count: Basic pattern (p79):

J1: { 3p 3 3 3 }
J2: { 3p 3 3 3 }

Tricks: (For the sake of concision, J1 will be the one throwing the trick, although it could equally be J2 in practice.)

(end of p79)
J1: { 4xp 3 3 3 }
J2: { 3p 2 3 3 }

(bottom-left p80) J1: { 3p 3 5p 2 2 3 3 3 } (J2 as normal)

(top-middle p80) J1: { 3p 3 4 2 } (J2 as normal)

(centre p80) J1: { 3p 3 3 4xp 2 3 3 3 } (J2 as normal)

(bottom-middle p80) J1: { 3p 4 2 3 } (J2 as normal)

(top-right p80)
J1: { 3p 3 3 5p 2 3 3 3 }
J2: { 3p 3 3 3 3p 2 3 3 }

3 Count: Basic pattern (p170):

J1: { 3p 3 3 }
J2: { 3p 3 3 }

Tricks:

(right p170, 1st trick): J1: { 3p 3 4xp 2 3 3 } (J2 as normal)

(right p170, 2nd trick): J1: { 3p 4 2 } (J2 as normal)

(right p170, 3rd trick): J1: { 3p 5p 2 2 3 3 } (J2 as normal)

(p171, 1st trick):
J1: { 4xp 3 3 }
J2: { 3p 2 3 }

(p171, 2nd trick):
J1: { 3p 3 5p 2 3 3 }
J2: { 3p 3 3 3p 2 3 }

(p171, 3rd trick):
J1: { 3p 4 5 1 3p 3 }
J2: { 3p 3 3 3p 2 3 }

<u>2 Count</u>: Basic pattern: J1: **{** 3p 3 **}**
(p178) J2: **{** 3p 3 **}**

<u>Tricks</u>: (p178, 1st trick): J1: **{** 4xp 3 **}**
J2: **{** 3p 2 **}**

(p178, 2nd trick): J1: **{** 3p 4p 2 3 **}** (J2 as normal)

(p178, 3rd trick): J1: **{** 3p 4p 5p 1 2 3 **}** (J2 as normal)

<u>Other Counts</u>:

Three-Three-Ten (p173): J1 & J2: **{**3 3 3 3 3 3p**}**×3, **{**3 3 3 3p**}**×3, **{**3 3p**}**×10

Four-Four-Eight (p81): J1 & J2: **{**3 3 3 3 3p**}**×4, **{**3 3 3p**}**×4, **{**3p**}**×8.

Pass-Pass-Self (p126): J1 & J2: **{** 3p 3p 3 **}**

<u>7 Balls</u> (See also pages 50-51 of this book)

<u>4 count</u>: Basic pattern: J1: **{** 5p 3 3 3 **}**
(middle of p132) J2: **{** 3 3 5p 3 **}**

<u>2 Count</u>: Basic pattern: J1(R,L): **{** 4p 3 **}**
(p131) J2(L,R): **{** 3 4p **}**

<u>Tricks</u>:
(bottom-left p132): J1: **{** 5xp 3 4p 3 **}**
J2: **{** 3 4p 2 4p **}**

<u>Other Counts</u>:
'534 pattern': J1(start: R, 2 balls in each hand): **{** 5 3 4p 3 3 3 **}**
(p133) J2(start: L, 2 in L, 1 in R): **{** 3 3 3 5 3 4p **}**

<u>8 Balls</u> (These aren't in EBJ):

<u>2 count</u>: Pattern 1: J1: **{** 5p 4 5p 3 4xp 3 **}**
J2: **{** 4xp 3 5p 3 5p 4 **}**

Pattern 2: J1(4¼): **{** 5p 3 5p 4 **}**
J2(3¾): **{** 5p 3 4xp 3 **}**

Pattern 3: J1(4.3)(start: R): **{** 6p 3 6p 3 5xp 3 **}**
J2(3.7)(start: L): **{** 4 4p 3 4p 3 4p **}**

<u>3 count</u>: '5p34': J1: **{** 5p 3 4 **}**
J2: **{** 5p 3 4 **}**

3 Person Patterns (With jugglers in vee/triangle formation)

9 Balls:

4 Count Triangle:	J1:	{	$3p_2$	3	3	3	$3p_3$	3	3	3	}
(p175)	J2:	{	$3p_3$	3	3	3	$3p_1$	3	3	3	}
	J3:	{	$3p_1$	3	3	3	$3p_2$	3	3	3	}

9 Ball Feed:	J1:	{	$3p_2$	3	$3p_3$	3	}	(p_2 means ‘pass to juggler 2’)
(p46)	J2:	{	$3p_1$	3	3	3	}	
	J3:	{	3	3	$3p_1$	3	}	

Ogie’s Nightmare:	J1:	{	$3p_2$	$3p_3$	3	}
(top-right p108)	J2:	{	$3p_1$	3	3	}
	J3:	{	3	$3p_1$	3	}

Ogie’s Nightmare Part 2:	J1:	{	$3p_2$	$3p_3$	3	}
(bottom-right p108)	J2:	{	$3p_1$	3	$3p_3$	}
	J3:	{	3	$3p_1$	$3p_2$	}

10 Balls:

10 Ball Feed:	J1(start: R, 2 in each hand):	{	$4p_2$	3	$4p_3$	3	}
(p159)	J2(start: L, 2 in L, 1 in R):	{	3	$4p_1$	3	3	}
	J3(start: L, 2 in L, 1 in R):	{	3	3	3	$4p_1$	}

11 Balls (not in EBJ):

11 Ball Feed:	J1(start: R, 2 in each hand):	{	$5p_2$	3	$5p_3$	3	}
	J2(start: R, 2 in each hand):	{	$5p_1$	3	3	3	}
	J3(start: R, 2 in R, 1 in L):	{	3	3	$5p_1$	3	}

14 Balls (not in EBJ):

Synch Feed:	J1:	{	($6p_3$ 4x)	($6p_2$ 4x)	}
	J2:	{	($6p_1$ 4x)	(4 4)	}
	J2:	{	(4 4)	($6p_1$ 4x)	}

15 Balls (not in EBJ):

Synch Triangle:	J1:	{	($6p_2$ 4x)	}
	J2:	{	($6p_3$ 4x)	}
	J3:	{	($6p_1$ 4x)	}

4 Person Pattern (12 balls)

Speed Weave p149, all start: R):	J1:	{	$3p_2$	3	$3p_3$	3	$3p_4$	3	}
(see EBJ for movement of jugglers)	J2:	{	$3p_1$	3	3	3	3	3	}
	J3:	{	3	3	$3p_1$	3	3	3	}
	J4:	{	3	3	3	3	$3p_1$	3	}

9) PATTERN APPENDIX

What follows is an assortment of patterns for 3, 4 and 5 balls in GS notation. Unless indicated otherwise by an airtime feature-row, the standard restrictions on airtime are to be assumed – so that each hand holds no more than 1 ball at a time. In some cases for readability I have left off some of the details about throw and catch positions, such as when balls are thrown underneath the other arm. However, these details could be reasonably argued to not actually be a necessary requirement of the pattern – just the natural way to juggle it. Finally I should say that not everyone will agree with the names or descriptions that follow, but I can only give my understanding of the patterns (which are in no particular order).

3 BALLS

(J)

Windmill

THR(Time)	{	2	1	}
SS(Base)	{	3	3	}
THR(Site)	{	R	L	}
THR(Pos)	{	l	l	}
CAT(Pos)	{	r	r	}

Trickledown

THR(Time)	{	2	0	}
SS(Base)	{	5	1	}
THR(Site)	{	R	L	}
THR(Pos)	{	rd	lu	}
CAT(Pos)	{	lu	rd	}

Machine

THR(Time)	{	2	0	}
SS(Base)	{	5	1	}
THR(Site)	{	R	L	}
THR(Pos)	{	ld	ru	}
CAT(Pos)	{	lu	rd	}

Mills Mess

THR(Time)	{	6	1	2	3	4	5	}
SS(Base)	{	3	3	3	3	3	3	}
THR(Site)	{	R	L	R	L	R	L	}
THR(Pos)	{	l	l	l	r	r	r	}
CAT(Pos)	{	m	m	m	m	m	m	}

Boston Mess

THR(Time)	{	6	1	2	3	4	5	}
SS(Base)	{	3	3	3	3	3	3	}
THR(Site)	{	R	L	R	L	R	L	}
THR(Pos)	{	l	m	r	l	m	r	}
CAT(Pos)	{	l	m	r	l	m	r	}

Burke's Barrage

THR(Time)	{	6	1	2	3	4	5	}
SS(Base)	{	2	3	4	2	3	4	}
THR(Site)	{	R	L	R	L	R	L	}
THR(Pos)	{	l	l	l	r	r	r	}
CAT(Pos)	{	r	m	m	l	m	m	}

Mike's Mess

THR(Time)	{	6	1	2	3	4	5	}
SS(Base)	{	2	2	5	2	2	5	}
THR(Site)	{	R	L	R	L	R	L	}
THR(Pos)	{	l	l	lbL	r	r	rbR	}
CAT(Pos)	{	r	r	mbR	l	l	mbL	}

Rubenstein's Revenge

THR(Time)	{	10	1	2	3	4	5	6	7	8	9	}
SS(Base)	{	2	2	3	3	5	2	2	3	3	5	}
THR(Site)	{	R	L	R	L	R	L	R	L	R	L	}
THR(Pos)	{	l	l	l	l	l	r	r	r	r	r	}
CAT(Pos)	{	r	r	m	m	m	l	l	m	m	m	}
CAT(Type)	{	-	-	c	-	-	-	-	c	-	-	}

Shuffle

THR(Time)	{	2	0	}
SS(Base)	{	5	1	}
THR(Site)	{	R	L	}
THR(Pos)	{	rd	lu	}
THR(Type)	{	-	c	}
CAT(Pos)	{	ld	rd	}
CAT(Type)	{	c	-	}
SS(Real)	*{*	*(4x*	*2x)*	*}*

Fake

THR(Time)	{	4	2	2	3	}
SS(Base)	{	4	2	4	2	}
THR(Site)	{	R	L	R	L	}
THR(Pos)	{	md	ld	rd	lu	}
THR(Type)	{	-	c	-	c	}
AIR(Rec)	{	2	0	2	0	}
CAT(Pos)	{	md	lu	rd	ld	}

Factory

THR(Time)	{	6	2	2	4	4	5	}
SS(Base)	{	4	2	4	2	3	3	}
THR(Site)	{	R	L	R	L	R	L	}
THR(Pos)	{	md	ld	rd	lu	ld	ru	}
THR(Type)	{	-	c	-	c	-	c	}
AIR(Rec)	{	2	0	2	0	2	1	}
CAT(Pos)	{	md	lu	rd	ru	ld	rd	}
CAT(Type)	{	-	c	-	c	c	-	}

Box

THR(Time)	{	4	0	2	2	}
SS(Base)	{	4	1	3	4	}
THR(Site)	{	R	L	R	L	}
THR(Pos)	{	r	l	r	l	}
CAT(Pos)	{	r	r	l	l	}
SS(Real)	*{*	*(4 2x)*		*(2x 4)*		*}*

Extended Box

THR(Time)	{	8	0	2	2	4	4	6	6	}
SS(Base)	{	5	1	4	1	3	3	3	4	}
THR(Site)	{	R	L	R	L	R	L	R	L	}
THR(Pos)	{	m	l	r	m	r	m	m	l	}
CAT(Pos)	{	m	m	r	r	m	m	l	l	}
SS(Real)	*{*	*(4x 2x)*		*(4 2x)*		*(2x 4x)*		*(2x 4)*		*}*

Eye

THR(Time)	{	4	0	2	}
SS(Base)	{	3	3	3	}
THR(Site)	{	R	L	R	}
THR(Pos)	{	or	m	lbL	}
AIR(Rec)	{	2	0	2	}
CAT(Pos)	{	lbL	m	or	}
SS(As)	*{*	*4*	*2*	*4*	*}*

Penguin

THR(Time)	{	2	1	}
SS(Base)	{	3	3	}
THR(Site)	{	R	L	}
THR(Pos)	{	m	m	}
CAT(Pos)	{	l	r	}
CAT(Type)	{	p	p	}

Body Orbit

THR(Time)	{	2	1	}
SS(Base)	{	3	3	}
THR(Site)	{	R	L	}
THR(Pos)	{	rf	lbb	}
CAT(Pos)	{	lf	rbb	}

Eating The Apple

THR(Time)	{	5	1	2	2	3	4	4	}
SS(Base)	{	3	3	3	3	4	4	1	}
THR(Site)	{	R	L	R	L	R	L	M	}
SS(As)	*{*	*3*	*3*	*3*	*1*	*3*	*3*	*1*	*}*
CAT(Site)	*{*	*L*	*R*	*L*	*M*	*L*	*R*	*R*	*}*

Chops

THR(Time)	{	4	0	2	2	}
SS(Base)	{	5	2	2	3	}
THR(Site)	{	R	L	R	L	}
THR(Pos)	{	lbL	lu	ru	rbR	}
CAT(Pos)	{	l	rd	ld	r	}
SS(Real)	*{*	*(4x 2)*		*(2 4x)*		*}*

Luke's Shuffle

THR(Time)	{	4	0	2	2	}
SS(Base)	{	4	1	3	4	}
THR(Site)	{	R	L	R	L	}
THR(Pos)	{	rd	lu	ru	ld	}
THR(Type)	{	-	c	c	-	}
CAT(Pos)	{	rd	rd	ld	ld	}
SS(Real)	*{*	*(4 2x)*		*(2x 4)*		*}*

Ben's Barrage

THR(Time)	{	10	1	2	3	4	5	6	7	8	9	}
SS(Base)	{	4	2	3	3	3	4	2	3	3	3	}
THR(Site)	{	R	L	R	L	R	L	R	L	R	L	}
THR(Pos)	{	lbL	raR	r	rbR	lu	rbR	laL	l	lbL	ru	}
THR(Type)	{	-	-	-	-	c	-	-	-	-	c	}
CAT(Pos)	{	m	ld	mbR	ru	mbR	m	rd	mbL	lu	mbL	}
CAT(Type)	{	-	-	-	c	-	-	-	-	c	-	}

4 BALLS

Tennis

THR(Time)	{	10	1	2	3	4	5	6	7	8	9 }
SS(Base)	{	5	3	4	4	4	5	3	4	4	4 }
THR(Site)	{	R	L	R	L	R	L	R	L	R	L }
THR(Pos)	{	or	-	-	-	-	ol	-	-	-	- }
CAT(Pos)	{	ol	-	-	-	-	or	-	-	-	- }

Sprung-cascade

THR(Time)	{	4	0	2	2 }
SS(Base)	{	7	1	3	5 }
THR(Site)	{	R	L	R	L }
THR(Pos)	{	m	-	-	m }
CAT(Pos)	{	l	-	-	r }
SS(Real)	*{*	*(6x 2x)*		*(2x 6x)*	*}*

Mills Mess

THR(Time)	{	6	1	2	3	4	5 }
SS(Base)	{	4	4	4	4	4	4 }
THR(Site)	{	R	L	R	L	R	L }
THR(Pos)	{	l	l	l	r	r	r }
CAT(Pos)	{	m	m	m	m	m	m }

Burke's Barrage

THR(Time)	{	6	1	2	3	4	5 }
SS(Base)	{	2	5	5	2	5	5 }
THR(Site)	{	R	L	R	L	R	L }
THR(Pos)	{	l	l	l	r	r	r }
CAT(Pos)	{	r	m	m	l	m	m }

Machine

THR(Time)	{	6	1	2	3	4	5 }
SS(Base)	{	6	3	3	6	3	3 }
THR(Site)	{	R	L	R	L	R	L }
THR(Pos)	{	r	rbR	lu	l	lbL	ru }
CAT(Pos)	{	rd	ru	ld	ld	lu	rd }

Rubenstein's Revenge

THR(Time)	{	10	1	2	3	4	5	6	7	8	9 }
SS(Base)	{	2	2	4	6	6	2	2	4	6	6 }
THR(Site)	{	R	L	R	L	R	L	R	L	R	L }
THR(Pos)	{	l	l	l	l	l	r	r	r	r	r }
CAT(Pos)	{	r	r	m	m	m	l	l	m	m	m }

Bouncy Shuffle

THR(Time)	{	8	0	2	2	4	4	6	6 }
SS(Base)	{	7	2	6	1	2	5	3	6 }
THR(Site)	{	R	L	R	L	R	L	R	L }
THR(Pos)	{	r	ld	rd	lu	rd	l	ru	ld }
CAT(Pos)	{	l	lu	rd	rd	ru	r	ld	ld }
SS(Real)	*{*	*(6x 2)*		*(6 2x)*		*(2 6x)*		*(2x 6)*	*}*

Box

THR(Time)	{	8	0	2	2	4	4	6	6 }
SS(Base)	{	8	1	4	1	3	8	3	4 }
THR(Site)	{	R	L	R	L	R	L	R	L }
THR(Pos)	{	r	l	r	l	r	l	r	l }
CAT(Pos)	{	r	r	r	r	l	l	l	l }
SS(Real)	*{*	*(8 2x)*		*(4 2x)*		*(2x 8)*		*(2x 4)*	*}*

Mike's Mess

THR(Time)	{	6	1	2	3	4	5 }
SS(Base)	{	2	2	8	2	2	8 }
THR(Site)	{	R	L	R	L	R	L }
THR(Pos)	{	l	l	lbL	r	r	rbR }
CAT(Pos)	{	r	r	mbR	l	l	mbL }

4 Ball Drop

THR(Time)	{	2	1	1 }
SS(Base)	{	5	5	2 }
THR(Site)	{	R	L	G }
SS(As)	*{*	*3G*	*3R*	*1L }*

4 Ball Cascade

THR(Time)	{	2	1 }
SS(Base)	{	5	3 }
THR(Site)	{	R	L }
THR(Pos)	{	r	m }
AIR(Rec)	{	3	3 }
CAT(Pos)	{	l	r }
SS(As)	*{*	*5*	*5 }*

5 BALLS

Tennis

THR(Time)	{	14	1	2	3	4	5	6	7	8	9	10	11	12	13	}
SS(Base)	{	7	4	4	5	5	5	5	7	4	4	5	5	5	5	}
THR(Site)	{	R	L	R	L	R	L	R	L	R	L	R	L	R	L	}
THR(Pos)	{	or	-	-	-	-	-	-	ol	-	-	-	-	-	-	}
CAT(Pos)	{	ol	-	-	-	-	-	-	or	-	-	-	-	-	-	}

Box

THR(Time)	{	4	0	2	2	}
SS(Base)	{	8	1	3	8	}
THR(Site)	{	R	L	R	L	}
THR(Pos)	{	r	l	r	l	}
CAT(Pos)	{	r	r	l	l	}
SS(Real)	*{*	*(8*	*2x)*	*(2x*	*8)*	*}*

Mills Mess

THR(Time)	{	6	1	2	3	4	5	}
SS(Base)	{	5	5	5	5	5	5	}
THR(Site)	{	R	L	R	L	R	L	}
THR(Pos)	{	l	l	l	r	r	r	}
CAT(Pos)	{	m	m	m	m	m	m	}

Burke's Barrage

THR(Time)	{	6	1	2	3	4	5	}
SS(Base)	{	2	6	7	2	6	7	}
THR(Site)	{	R	L	R	L	R	L	}
THR(Pos)	{	l	l	l	r	r	r	}
CAT(Pos)	{	r	m	m	l	m	m	}

Machine

THR(Time)	{	4	1	2	3	}
SS(Base)	{	7	7	3	3	}
THR(Site)	{	R	L	R	L	}
THR(Pos)	{	-	-	lbL	raR	}
CAT(Pos)	{	-	-	l	r	}

Rubenstein's Revenge

THR(Time)	{	10	1	2	3	4	5	6	7	8	9	}
SS(Base)	{	2	2	7	7	7	2	2	7	7	7	}
THR(Site)	{	R	L	R	L	R	L	R	L	R	L	}
THR(Pos)	{	l	l	l	l	l	r	r	r	r	r	}
CAT(Pos)	{	r	r	m	m	m	l	l	m	m	m	}

Martin

THR(Time)	{	8	2	4	6	}
SS(Base)	{	4	6	4	6	}
THR(Site)	{	R	R	L	L	}
AIR(Rec)	{	4	4	4	4	}
SS(As)	*{*	*6*	*6x*	*6*	*6x*	*}*

Vesuvius

THR(Time)	{	20	0	2	2	4	10	10	12	12	14	}
SS(Base)	{	5	6	6	3	5	5	6	6	3	5	}
THR(Site)	{	R	R	L	L	R	L	L	R	R	L	}
AIR(Rec)	{	4	4	5	5	6	4	4	5	5	6	}
SS(As)	*{*	*(6x*	*6)*	*(7*	*7s)*	*8x*	*(6x*	*6)*	*(7*	*7s)*	*8x*	*}*

Mike's Mess

THR(Time)	{	6	1	2	3	4	5	}
SS(Base)	{	2	2	11	2	2	11	}
THR(Site)	{	R	L	R	L	R	L	}
THR(Pos)	{	l	l	lbL	r	r	rbR	}
CAT(Pos)	{	r	r	mbR	l	l	mbL	}

Boston Mess

THR(Time)	{	10	1	2	3	4	5	6	7	8	9	}
SS(Base)	{	5	5	5	5	5	5	5	5	5	5	}
THR(Site)	{	R	L	R	L	R	L	R	L	R	L	}
THR(Pos)	{	l2	l1	m	r1	r2	l2	l1	m	r1	r2	}
CAT(Pos)	{	l2	l1	m	r1	r2	l2	l1	m	r1	r2	}

ES QUICK REFERENCE

(A)

x	-	Crossing throw	5_R	-	Right hand 5	b	-	Bounced throw
s	-	Straight throw	3_L	-	Left hand 3	b_2	-	Double bounce
( )	-	Synchronous throw	2_T	-	Thrown 2 (eg in [3,2_T])	p	-	Pass
[ ]	-	Multiplex throw	4_H	-	Held 4 (eg in 64_H5)	p_3	-	Pass to juggler 3
*	-	Symmetric synchronous pattern				t	-	Tramline

ACKNOWLEDGEMENTS

Special thanks to the following:

Vincent Braslavschi – for giving me the final bit of encouragement I needed to start writing this book.
Steve and Nigel – for helpful comments and suggestions.
Hairy – for many technical discussions about juggling, helping me to learn to juggle in the first place, and providing much constructive criticism about the book.
Henry Segerman – for proof-reading and suggestions.
Colin E & everyone else at the IJDb – for putting this book on the net and maintaining a fantastic website.
Rupert Millard – for putting a pdf version of this book on the net.
My sister Bethany – for help with the word processing program.
My dad Stephen – for letting me use his computer to write the book, and his support.
Jugglers in general – for being lively but chilled, philosophical but fun-loving, and basically for being sweet and wonderful people. (Does juggling make you sweet, or do sweet people like to juggle?)
God – for the laws of gravity and maths, and for giving us hands. And thanks to my main man, Jesus C.

Best wishes,

Ben ><>

ANSWERS TO PUZZLES

(M)

1) There are 14 of them (if you include 7 and 1): 7, 1, 71, 711, 771, 711111, 771111, 717111, 777111, 771711, 771171, 777711, 777171, 777771.

2) There are 8 of them: 948 966 10,29 12,09 12,18 12,45 12,72 12,90.

3) Possible answer (there are probably others): 7031640. A quick way to tell whether a pattern is ground, is the following: it is ground if and only if the SS contains a sub-sequence $v_B\ v_{B-1}\ v_{B-2} \ldots v_1$, such that $v_k \geq k$ for each k (from B down to 1), where B is the number of balls in the pattern. So you can tell that 7031640 is non-ground because it is not possible to find a sub-sequence $v_3\ v_2\ v_1$, with $v_3 \geq 3$, $v_2 \geq 2$, and $v_1 \geq 1$.

4) Possible answers: a) 8531841881128186111. (Period 19)

b) 84178118151818411. (Period 17)

5) a) A1: 222, A2: 231, A3: 123, A2: 141, A3: 114, A2: 150, A3: 015, A3: 501.

b) A1: -1-1-1, A4: 2-1-1, A2: 20-2, A4: 201, A4: 501.