Queuing Analysis

Version 3.3

Jim Fawcett
CSE681 – Software Modeling and Analysis

Fall 2005

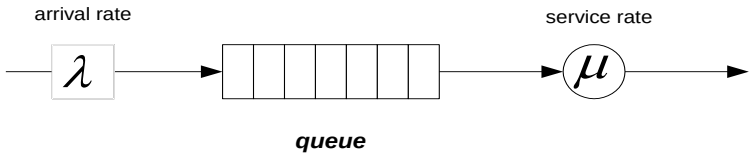
Table of Contents

Queuing Analysis

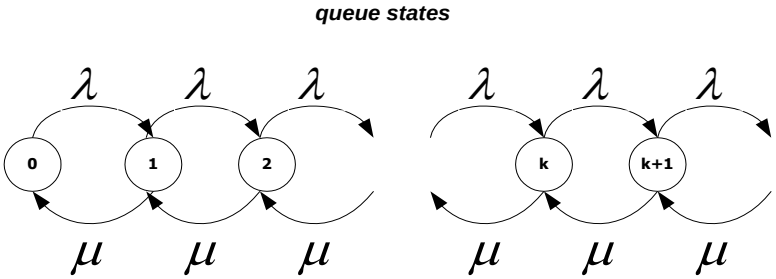
I. Basic Models

Many software systems use queues. Event driven systems use queues to collect inputs from many sources, e.g., keyboard, mouse, other devices, and multiple threads running in a program. Queues are used to allow event generators which may be bursty to avoid waiting for the event processor to finish with a prior event before accepting the next. The bursty source simply deposits its event message in a queue and goes about its business while the event processor withdraws event messages from the queue when it is ready. These notes are concerned with a simple queuing model and relatively elementary analysis of its performance. The analysis will use some basic notions from probability theory. You will find a quick survey of those ideas in the Appendix.

Assume that we have a queue with a constant average rate of arriving messages, λ. Our computer program can process each message with a constant average service rate, μ.



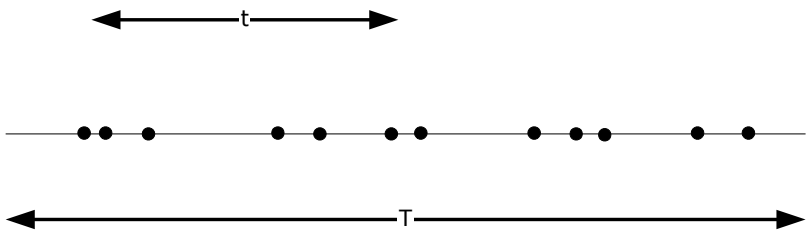
We represent the state of the queue with the following diagram, where the k^{th} state represents a queue holding k messages:



Since messages arrive at a rate of λ messages per unit time then, given that the queue is in state k, the queue will change from k to k+1 at the same rate, since a single new arrival will add one to the k messages waiting in the queue.

The average rate out of state k is, therefore, $\lambda p(k)$, where p(k) is the probability of being in state k. If the queue has k messages waiting, when a program finishes handling a message it grabs another off the queue and the queue moves to state k-1. So, the average rate of changing from state k to k-1 is $\mu p(k)$, since μ is the average rate of servicing messages.

To proceed with this analysis, we need to know something about the random processes that govern arrivals and servicing. Let's focus on arrivals. If we start with a finite interval of length T and suppose that n messages are uniformly distributed in that interval, then we have a situation like that shown below.



The probability that any single point lies in the interval of length t is just:

$$p_e = P_1(t,T) = t/T$$

And the probability that k of the n points lies in the interval of length t is just:

$$P_k(t,T) = B_n(k)\, p_e^{k}(1-p_e^{n-k})$$

Where $B_n(k)$ is the binomial coefficient, e.g.:

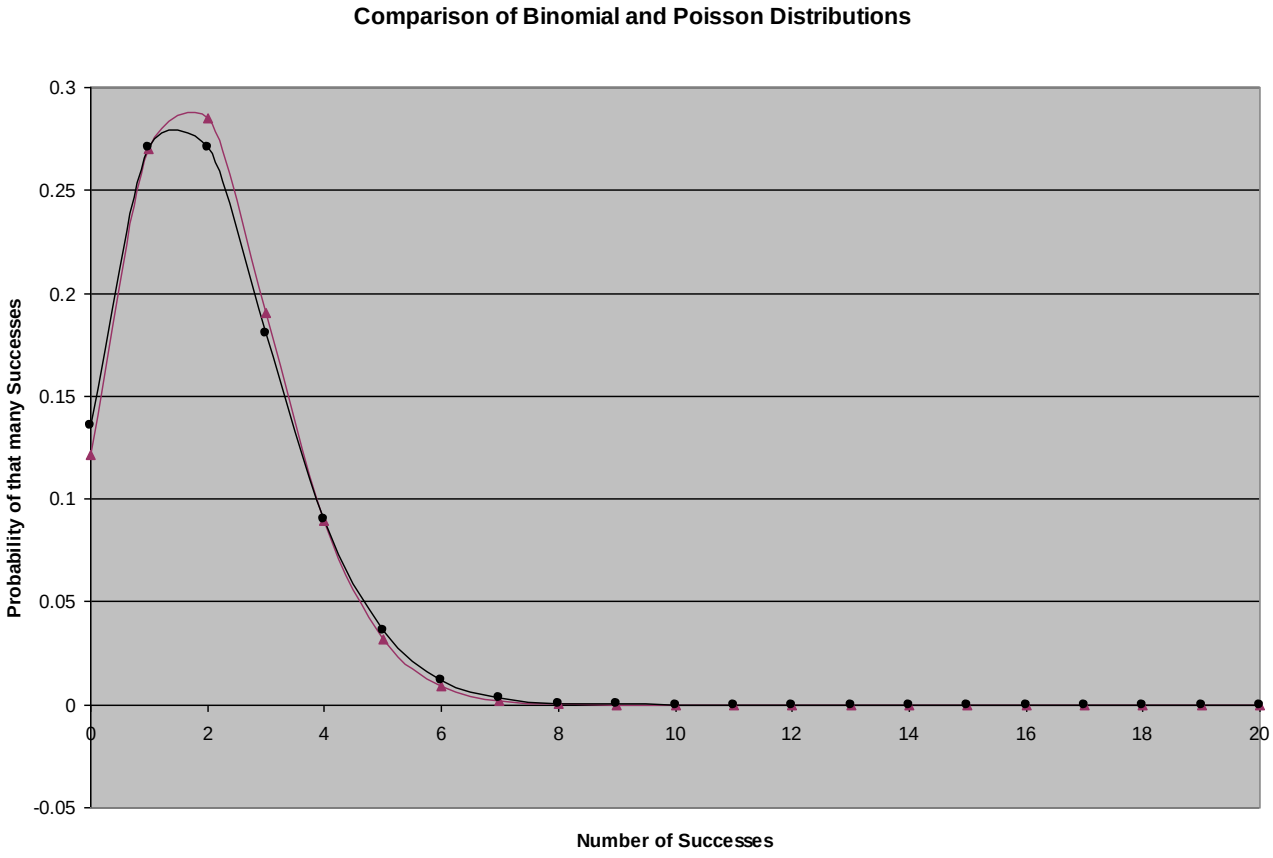
$$B_n(k) = n!/(k!(n-k)!)$$

Now, let's let the length of the interval go from T to infinity and increase the number n of points so that $\lambda = N/T$, the density of points, stays fixed. It is fairly easy to show[1] that, for constant λ, this binomial distribution tends, as T goes to infinity, to the Poisson distribution:

$$P(k, \lambda t) = (\lambda t)^k e^{-\lambda t}/k!$$

[1] "An Introduction to Probability Theory and its Applications", 2nd Edition, William Feller, Wiley, 1957

This is the probability that k points are found in an interval of length t, when λ is the density of points on the line.



II. Infinite Queue Analysis

We shall model the arrival process by $p(k, \lambda t)$ and the service process by $p(k,\mu t)$. Now, consider first an empty queue with a message arrival rate of λ. Then, the probability that the queue stays empty for t units of time is just:

$$p_0(t) = p(0, \lambda t) = e^{-\lambda t}$$

The rate of change of this probability is:

$$d\, p_0(t) \;/\; dt = -\lambda\, e^{-\lambda t} = -\lambda\, p_0(t)$$

Now, the probability that our queue is in state 0 at time $t + \Delta t$ is:

$$p_0(t+\Delta t) = p_0(t)\, e^{-\lambda \Delta t} + p_1(t)\, \mu \Delta t\, e^{-\mu \Delta t} + \text{negligible probabilities}$$

The first term on the right is the probability that we have no messages waiting at time t times the probability that we get no new messages in time Δt[2]. The second term is the probability that we have one message waiting at time t times the probability that it gets serviced in time Δt^2, sending the state back to zero waiting messages. We are going to let Δt become infinitesimal, so the probabilities that we might have more than one waiting message at time t and they all got serviced is neglibible.

The preceding expression is just an application of Bayes Law:

$$P(A) = P(A/B)P(B) + P(A/C)P(C), \text{ where } P(B) + P(C) = 1$$

Here, the mutually exclusive probabilities are $P(B) = p_0(t)$ = probability that queue is in state 0 and $P(C) = p_1(t)$ = probability that the queue is in state 1, since we are neglecting the higher states which occur with vanishingly small probability for very small Δt. The conditional probabilities are $P(A/B) = e^{-\lambda \Delta t}$, the probability that we stay in state 0 provided that we are already in state 0, and $P(A/C) = \mu \Delta t\, e^{-\mu \Delta t}$, the probability that we move from state 1 to state 0, given that we were in state 1.

Dividing through by Δt, we have:

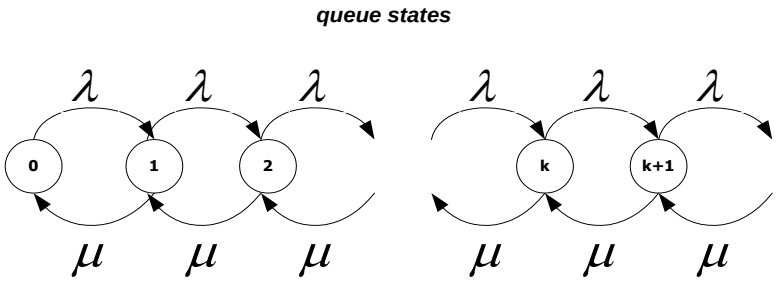
$$(p_0(t+\Delta t) - p_0(t)) \;/\; \Delta t = (1 - e^{-\lambda \Delta t})\, p_0(t) \;/\; \Delta t + \mu\, e^{-\mu \Delta t}\, p_1(t)$$

and, as $\Delta t \to 0$, we have:

$$d\, p_0(t) \;/\; dt = -\lambda\, p_0(t) + \mu\, p_1(t)$$

[2] The Poisson model implies that these are independent events, so the probability that we are in state 0 and don't leave is the product of those two probabilities.

This is a first order linear coupled differential equation in queue state probabilities. Applying exactly the same reasoning to each of the states, we get the general model for queue states:



$$d\,p_k(t)/dt = \lambda\,p_{k-1}(t) - (\mu + \lambda)\,p_k(t) + \mu\,p_{k+1}(t), \quad k > 0$$

$$d\,p_0(t)/dt = -\lambda\,p_0(t) + \mu\,p_1(t), \quad k = 0$$

We can use this coupled (infinite) set of first order linear differential equations to solve for the probabilities of being in any state at any given time, given some initial set of state probabilities, e.g., $p_0(0) = 1$ and $p_k(0) = 0$ for $k>0$.

What we are usually interested in, however, is the steady state performance of the queue. In steady state, all of the probabilities are constant, provided, of course, that a steady state exists. Then, we have a coupled set of linear algebraic equations to solve:

(1a) $0 = \lambda\,p_{k-1} - (\mu + \lambda)\,p_k + \mu\,p_{k+1}, \quad k > 0$

(1b) $0 = -\lambda\,p_0 + \mu\,p_1, \quad k = 0$

Using these relationships and the fact that the sum of the queue state probabilities must be 1:

(1c) $\sum p_k = 1$

After a little bit of algebraic manipulation of these equations, we find that:

(2a) $p_0 = 1 - \rho, \quad \rho = \lambda / \mu$ (2b) $p_k = \rho^k p_0$

The average queue length is:

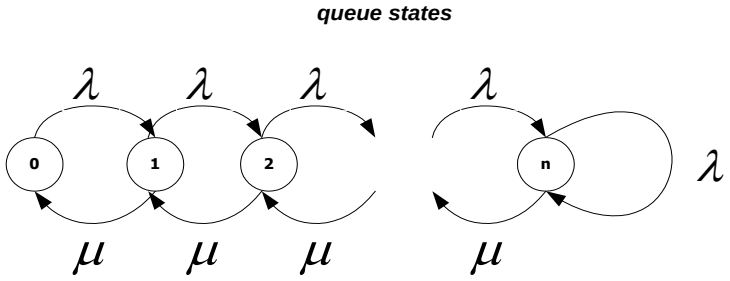
(3) $N = \sum k\,p_k = \rho / 1 - \rho$

And the response time, i.e., the average time a message spends in the queue and being serviced is:

(4) $R = N / \lambda$

III. Finite Length Queue Analysis

For a finite queue, we need to understand what happens if, when the queue is full, a new message arrives. Our model will be that we simply discard the message. For this case our queue state model looks like this:



With some rather tedious algebraic manipulations, and series summations, we find that:

(5a) $p_0 = (1 - \rho) / (1 - \rho^{n+1})$

(5b) $p_k = \rho^k p_0$

The rate at which arriving messages are lost is:

(6) rate of lost messages = λp_n

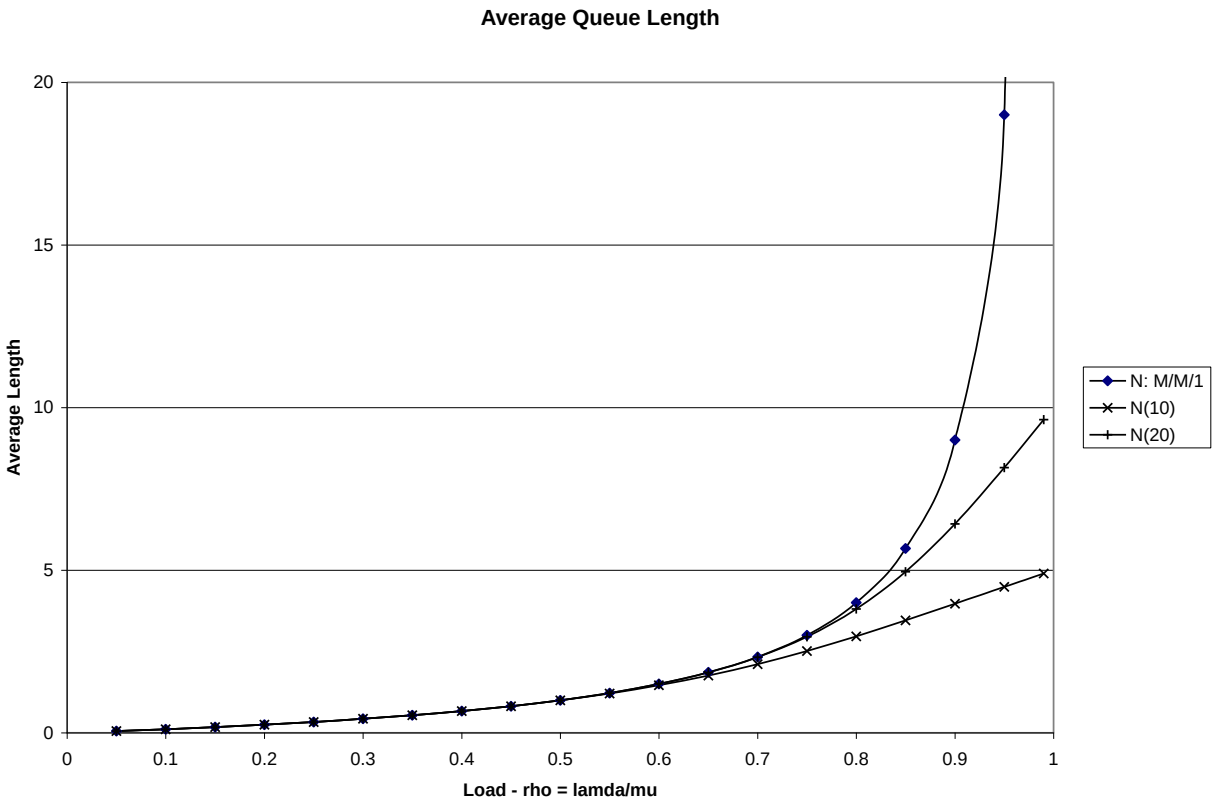
and the average queue length is:

(7) $N = ((1 - \rho^n) \rho - n(1 - \rho) \rho^{n+1} / (1 - \rho)^2) p_0$

As before, the response time is:

(8) $R = N / \lambda$

If we plot the average queue length versus load, ρ, we get the graphs shown on the next page for an infinite queue, and for finite queues of maximum length of 10 and 20 messages.



This plot tells us that if the queue load factor rho becomes much larger than 0.5 then the average queue length and response time for messages in the queue get large very quickly. When rho becomes 1.0, the queue will no longer reach a steady state, but will, in fact, grow in length as long as messages continue to arrive.

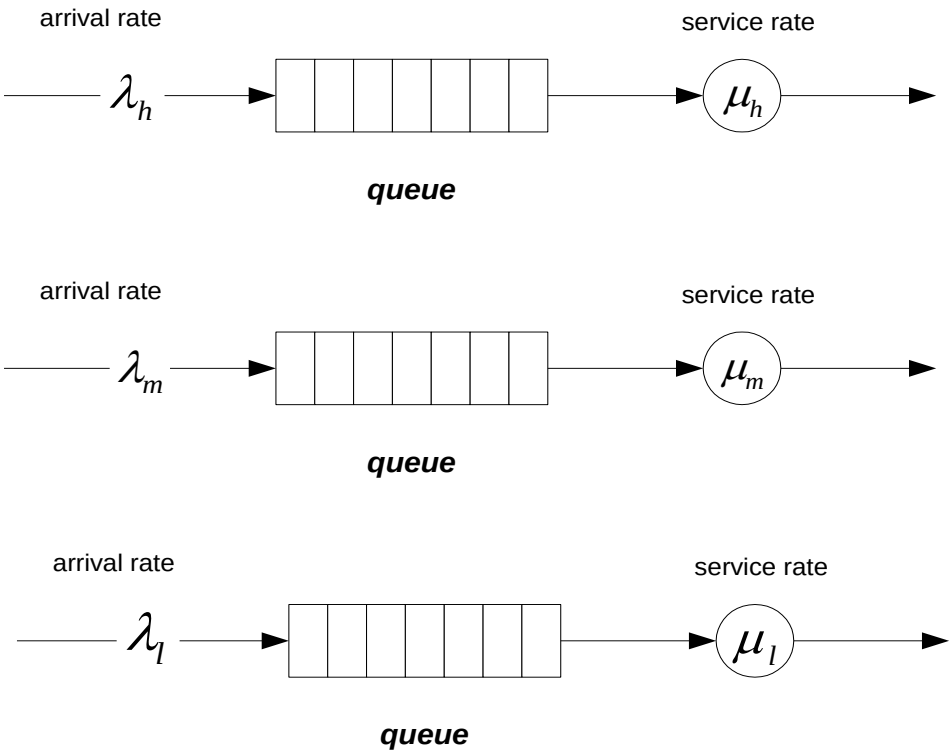
IV. Practical Analysis

So how do we use all this stuff? The process almost always boils down to this:

1. Build a load model that describes the input messages and analyzes their average arrival rate, usually by some conservative "back-of-the-envelope" calculations. See the later parts of the Software Architecture notes discussed in the first lecture.
2. Build a prototype of the message processing software and analyze its average service rate. We almost always have to measure real software operation. It just is not feasible to estimate the running time of software that does not yet exist. This does not mean that we have to build a complete servicing system. We just build enough to get an estimate of the upper bound of the average service time.
3. Compare the arrival rate to the measured service rate. We want that ratio to be no larger than about 0.25 or we are likely to be in trouble.

V. Priority Queues

An accurate analysis of priority queues is relatively involved, so we will present here a simple approximation that works quite well in practice. Consider the diagram, shown below. The three queues collect messages of three different priorities, high, medium, and low.



Assume that a single processor, in one program, is servicing all three queues. Only the high priority queue will be serviced until it is empty. Then, as long as it stays empty the program services the medium priority queue. The low priority queue is serviced, if and only if, both of the other queues are empty.

As a consequence of this policy, our previous analysis applies exactly to the high priority queue. Since the medium priority queue is only serviced when the high priority queue is empty, the effective service rate for the medium priority queue is:

$$\mu_{m\text{-}eff} = \mu_m P_{h0}$$

Here, $\mu_{m\text{-}eff}$ is the average rate at which medium priority messages are processed when high priority messages get processed first, and μ_m is the average rate at which medium priority messages are processed if they are the only messages that arrive, e.g., they have the processesor's full attention. Finally, P_{h0} is the probability that the high priority

queue is in the empty state. With this "effective" service rate, the analysis of the medium priority queue is carried out according to the standard M/M/1 model.

Similarly, for the low priority queue:

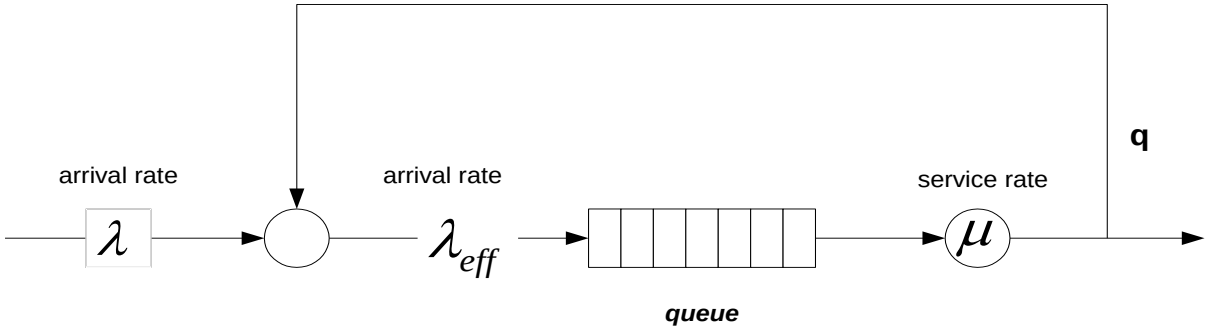
$$\mu_{l\text{-}eff} = \mu_l P_{m0}$$

where $\mu_{l\text{-}eff}$ is the effective low priority service rate, μ_l is the rate if the high and medium queues are always empty, and P_{m0} is the probability that the medium priority queue is empty, and analysis is carried out according to the M/M/1 model.

So, in conclusion, we analyze the priority queues using the simple queue analysis with the modified service rates for the medium and low priority queues, given above.

VI. Feedback Queues

A feedback queue has the structure shown in the diagram below. When messages are processed, some messages need to be sent back to the input for processing again, with probability q.



In steady state, the flow rate out of the queue must equal the flow into the queue. Further, the flow rate of messages sent back for processing again will be q times that rate, so the net rate of messages into the queue will be:

$$\lambda_{eff} = \lambda + q\lambda_{eff} \Rightarrow \frac{\lambda}{1-q}$$

Analysis of the M/M/1 queue was based on the Poisson probability model. It has been shown by Jackson that the input to the queue is not Poisson, but that because the basic independence of arrival structure has not changed, the same analysis applies. So we can analyze the feedback queue using the same equations developed for the simple queue, simply adjusting the input rate as shown above.

VII. References:

1. An introduction to Probability Theory and its Applications, Volume 1, William Feller, Wiley, 1957

2. Probability, Random Variables, and Stochastic Processes, Athanasios Papoulis, McGraw-Hill, 1984

VIII. Appendix – A Tiny Bit of Probability

- ***1. Basic Probability***
  - ***a. frequency of occurrence***
  - ***b. ensemble versus sample statistics, ergodicity***
- ***2. Laws of Probability***
  - ***a. Independent events***
  - ***b. Conditional Probability***
  - ***c. Mutually Exclusive Events***
  - ***d. Bayes' Theorem***

Basic Probability

Definition 1 – Sample space

A sample space is the set of possible outcomes of an experiment. An example is the set of outcomes of flipping a coin three times. S is the sample space for this experiment:

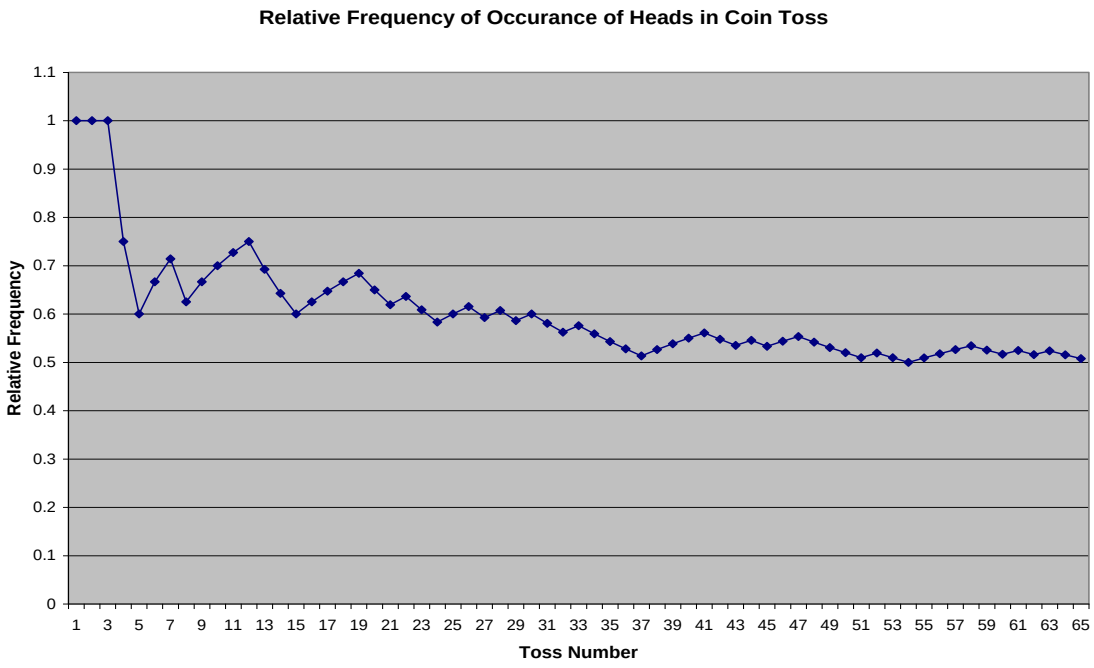
S = { hhh, hht, hth, htt, thh, tht, tth, ttt }

An event is a single point drawn from this sample space, e.g., hth.

Definition 2 – Probability

Probability is the expected frequency of occurrence of an event in an experiment that is repeated an arbitrarily large number of times. If we flip an unbiased coin N times, we expect that h will occur approximately 0.5 N times. As N gets progressively larger the approximation gets better and better. Here are the outcomes of a sequence of coin tosses[3]:

Hhhtthhthhhhhttthhhhhtthtthhthtthtthttttthhhhhtththhttttthtthhhhhtththt



[3] You would think that an instructor would have better things to do with his time than flip a coin 65 times.

Ensemble versus Sample Statistics, Stationarity and Ergodicity

We can imagine an experiment in which a single person flips a coin a large number of times, say 1 million times. We can also imagine an experiment in which we have 1 million people who all agree to flip a coin once and communicate to us the results.

Definition 3 – Sample Statistics

The first case is an example of a Sample Statistics experiment. The outcomes are all derived from a single process, evolving in time.

Definition 4 – Ensemble Statistics

The second is an example of an Ensemble Statistics experiment. The outcomes are each derived from a separate process and all of the processes can, but do not have to, occur concurrently.

Definition 5 – Stationary Process

If, every time we repeat these experiments we get the same average behavior we say that the statistics are stationary.

Definition 6 – Ergodic Process

If the average behavior of the Sample Statistics is the same as the average behavior of the Ensemble Statistics we say that the flipping process is Ergodic. In a stationary ergodic process we can expect that sample statistics will converge to ensemble statistics. That is what happened in my coin flipping experiment. The probability estimate (really the average of 1 and 0 values) converged to the expected probability, 0.5.

Notation

In the following we will denote the set of all possible outcomes of an experiment by S, the experiment's sample space.

Subset:

We denote a subset of events with the notation: $A \subset S$.

Set Union

The set of points in either A or B: $e \in A$ or $e \in B$ or both $\Rightarrow e \in A \cup B$

Set Intersection

The set of points common to A and B: $e \in A$ and $e \in B \Rightarrow e \in A \cap B$

Laws of Probability

Conditional Probability

The conditional probability of some event is the probability of the event, given that some other event has occurred. If a and b are two events, the conditional probability, P(a / b), is the probability that a occurs given that b has already occurred.

For a coin flipping experiment consisting of three flips:

S = { hhh, hht, hth, htt, thh, tht, tth, ttt }

The probability that only one head occurs given that the first toss was a tail:

P(htt or tht or tth / txx) = P(tht) + P(tth) = 2/8

Law of Conditional Probability:

Events: $P(a \text{ and } b) = P(a / b)\, P(b) = P(b / a)\, P(a)$

Sample Spaces: $P(A \cap B) = P(A / B)\, P(B) = P(B / A)\, P(A)$

For the experiment above we can compute the probability that there are two heads in the sequence of three flips and the first was a head:

P((hht or hth or thh) and hxx)
= P(hht or hth or thh / hxx)[4] P(hxx) = 2/4 * 4/8 = 2/8
= P(hxx / hht or hth or thh) P(hht or hth or thh) = 2/3 * 3/8 = 2/8

In this simple example it is obvious that:

P(hht or hth or thh and hxx) = P(hht or hth) = P(hht) + P(hth) = 2/8

In terms of sample spaces:

A = { hht, hth, thh }, P(A)[5] = 3/8, P(B / A) = 2/3 $\Rightarrow$ P(B / A) P(A) = 2/8
B = { hhh, hht, hth, htt }, P(B) = 4/8, P(A / B) = 2/4 $\Rightarrow$ P(A / B) P(B) = 2/8
A $\cap$ B = { hht, hth }, P(A $\cap$ B) = 2/8

[4] hxx restricts the sample space to four events: S(hxx) = { hhh, hht, hth, htt }. Since thh is not in that sample space there are only two possibilities: hht or hth.
[5] Note that P(A) really means P(A / S) for any set of events A in the Sample Space S.

Independent Events

We say that two events are independent if their joint probability is equal to the product of their individual probabilities. The probability of flipping two heads in a row is just:

$$P(h) = 0.5 \Rightarrow \qquad P(hh) = P(h) * P(h) = 0.25$$

Here, the notation, P(hh), is a shorthand for P(head on first toss, Head on second toss).

Occurrences of a head the first time in no way influence the probability of a head the second time we flip a coin, as is obvious from looking at the sample space. Any of these four events are equally likely, provided that the coin is unbiased.

Sample Space = { hh, ht, th, tt }

A gambler may believe that a long string of bad luck makes the next wager more likely to be profitable and wagers most of his bankroll. Not true, and so many gamblers are often broke.

We can generalize these ideas to sample spaces:

S = { hhh, hht, hth, htt, thh, tht, tth, ttt }

A = { hxx } = { hhh, hht, hth, htt } $\Rightarrow$ P(A)[6] = 1/2

B = { xxt } = { hht, htt, tht, ttt } $\Rightarrow$ P(B) = 1/2

$P(A \cap B) = P(hxx \cap xxt) = P(hxt) = P(hxx)*P(xxt) = P(A) * P(B) = 1/4$

Law of Independent Events:

Events:

a and b are independent events $\Rightarrow$ P(a and b) = P(a) * P(b)

Sample Spaces:

$P(A / B) = P(A) \Rightarrow P(A \cap B) = P(A) * P(B)$

[6] We interpret P(A) to mean P(A / S).

Mutually Exclusive Events

If two events are mutually exclusive the probability of occurrence of either one or the other is the sum of the probabilities of each. If a set of events covers the sample space, then the probability that one of them occurs has to be unity.

For a coin flipping experiment consisting of three flips:

S = { hhh, hht, hth, htt, thh, tht, tth, ttt }

The probability that only one tail occurs in the three flips is:

P(hht or hth or thh) = P(hht) + P(hth) + P(thh) = 3/8

Law of Mutually Exclusive Events:

Events:

a and b are mutually exclusive events $\Rightarrow$ P(a or b) = P(a) + P(b)

Sample Spaces:

$A \cap B = \varnothing \Rightarrow P(A \cup B) = P(A) + P(B)$

Baye's Theorem

Given a sample space, S of events e, and a subset A:

$$e \in \sim A \implies e \in S \text{ and } e \notin A, \quad A \subset S$$

e.g.:

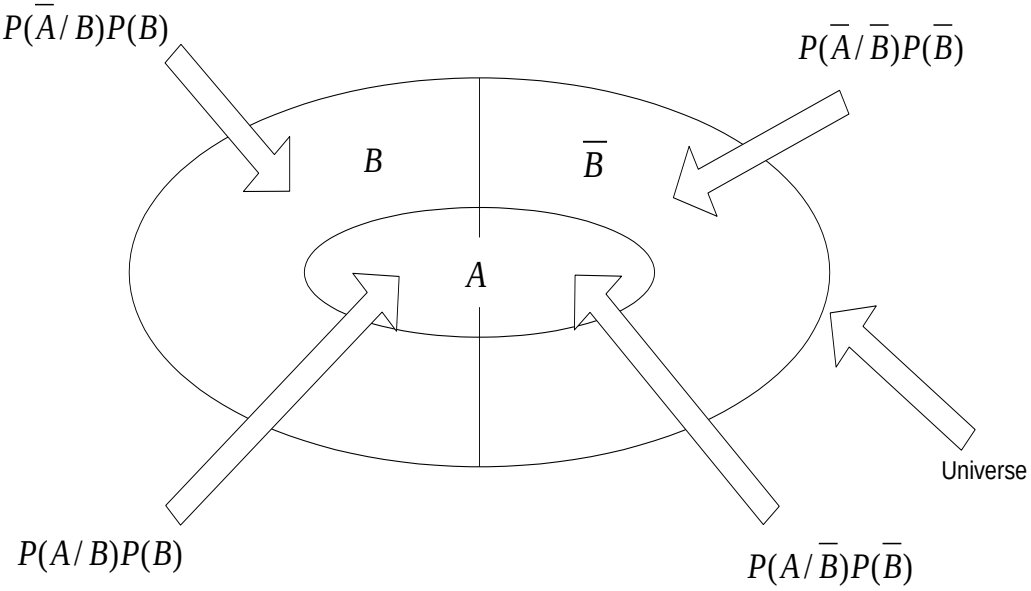
$$\sim A \cup A = S$$

Then Bayes theorem states that:

$$P(A) = P(A / B)\, P(B) + P(A / \sim B)\, P(\sim B)$$

This is illustrated in the Venn diagram on the next page.

Conditional Probabilities



Example:
Universe is set of outcomes for all sequences of 4 coin tosses
B is set of outcomes with 2 heads
A is set of outcomes with 1st and 2nd outcomes are heads

$P(A \cap B) = P(A/B)P(B)$ *Conditional probability*

$P(A \cap B) = P(A)P(B) \Leftrightarrow$ *A and B are independent*

$P(A) = P(A/B)P(B) + P(A/\overline{B})P(\overline{B})$ *Bayes Rule*

$P(A) + P(B) = 1 \Leftrightarrow$ *A and B are mutually exclusive exhaustive events*