Chapter 1

Being Discrete

In order to transmit information, one must first represent that information in some physical form. This applies not only to computer networks, but to all forms of information transmission. When we talk, our thoughts are represented by the physical changes the sounds we produce cause in the air between ourselves and our audience. Written forms of communication depend on physically placing ink on paper in the shapes of letters or other symbols. Communications, of course, is not limited to language. The cliche that a picture is worth a thousand words certainly applies here. In fact, the range of physical techniques we have developed for representing and then communicating information is enormous.

The communication and computer technologies that have swept such enormous changes into our lives in the last few decades also depend on physical representations of information. A wide range of techniques for representing information has been developed for such devices including holes in antique punched "IBM" cards, pulses of light on fiber networks, magnetic polarization on floppy disks, and microscopic indentations on CDs and DVDs. While these techniques are certainly varied, they share an important common feature. They are all based on digital representations of information.

In some cases the "digital" nature of these technologies is very obvious. We can all distinguish a digital clock from an analog clock or a digital thermometer from its analog counterpart. There are also, however, many examples where the digital nature of the technologies we use might not be so obvious. What makes a digital camcorder different from a non-digital camcorder or a digital portable phone different from a non-digital phone?

The Internet and all the computing devices connected to it are digital and being digital is fundamental to their nature. Accordingly, we will start our exploration of the technology underlying the Internet by trying to understand exactly what it means to be digital and what advantages being digital has over the alternative analog techniques for representing information.

1.1 Discrete vs. Continuous

dig • i • tal, adj.

1. Of, relating to, or resembling a digit, especially a finger.
2. Operated or done with the fingers.
3. Having digits.

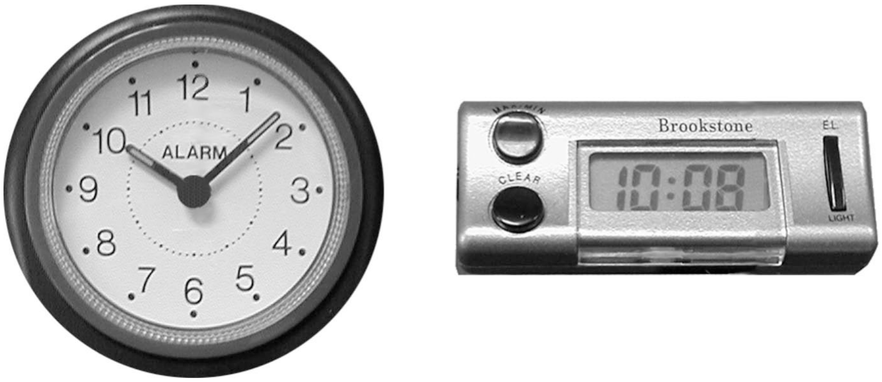
4. Expressed in digits, especially for use by a computer.
5. Using or giving a reading in digits.

(From the American Heritage Dictionary, 1992 (digital edition, of course))

You can probably guess that the first two meanings given for “digital” in this dictionary definition have little to do with our interest in the digital representation of information. The remaining definitions, however, probably seem very relevant. After all, the two “obvious” examples of digital devices just mentioned, digital clocks and digital thermometers, certainly suggest that being digital has something to do with using the symbols we call digits, namely 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9.

If this brings out any latent Math-phobia in your psyche, don’t worry. The real significance of being digital has almost nothing to do with digits or numbers. This may be a surprise, but notice that having digits certainly isn’t enough to make something digital. An analog thermometer has digits labeling its scale. In fact, there are typically more digits on the scale of an analog thermometer than there are in the display of a digital thermometer. Most analog clock faces are also adorned with their share of digits. If the use of digits made something digital, then we would have to consider both analog clocks and analog thermometers digital. We must think a bit more carefully about the differences between analog and digital clocks to get a glimpse of what being digital is really about.

Imagine that I had a digital clock and an analog clock that had been very precisely synchronized. If I showed them both to you at 10:08, they would look something like:



If I asked you to look at them thirty seconds later, the digital clock would not have changed at all, while the minute hand of the analog clock would have moved half of the way from 10:08 to 10:09.

Actually, depending on how the gears that drive the minute hand work, the minute hand might not be exactly one half of the way from 10:08 to 10:09 thirty seconds after 10:08. At some point between the times 10:08 and 10:09, however, it will be exactly one half of the way between the positions for 10:08 and 10:09 on the clock face. It can’t physically get from the 10:08 position to the 10:09 position without going past the point one half of the way between them. For the same reason, of course, the minute hand of the analog clock must also pass through the point one third of the way between 10:08 and 10:09 and through all the other points between 10:08 and 10:09.

For the digital clock, however, there is no “between”. It simply goes from 10:08 to 10:09. Of course, we could build a digital clock with an extra digit to represent tenths of seconds. Then, at thirty seconds after 10:08 it could display: 10:08:5. Unfortunately, this extra digit won’t enable the digital clock to display the correct time at fifteen seconds after 10:08. That would require yet another added digit. No matter how many digits we add there will always be times that would require additional digits to represent them. In particular, at twenty seconds past 10:08, the digital clock would have to read

10:08:33333333333333333333333333333333333.......

since 1/3 can only be exactly represented by an infinitely repeating decimal representation. The analog clock's hands, however, somehow mysteriously must even pass through the point exactly one third of the way between 10:08 and 10:09.

This difference between analog and digital clocks is an example of the difference between what are called continuous and discrete phenomena. Between any two states of a continuous phenomena there are an infinite number of "in-between" states. The states of a discrete phenomena, on the other hand, are separate. There is nothing between them.

Given this distinction, we can now take a fresh look at the difference between the roles of the digits used in an analog clock and the digits that are used on the faces of most analog clocks. The digits on an analog clock label a few selected positions to which the hands of the clock may point, but they themselves are not used to represent the current time. Instead, the continuously varying positions of the clock hands represent the time. Given that the hands can be in infinitely many different positions, they can (at least in theory) represent infinitely many different times.

In a digital clock it is the digits themselves that represent the time. Given the fact that there are only ten distinct digits and that a typical digital clock display shows four digits, the number of different times that can be represented is distinctly finite. There are at most 10,000 different numbers that can be displayed in 4 digits, and some of the combinations that could be displayed are not considered meaningful as times (i.e. 83:47).

Digital and analog clocks exemplify two very distinct approaches to representing information. An analog device represents information using some continuous physical property of the device. In an analog clock, the positions of the hands represents the time. In an analog thermometer, the length of the mercury represents the temperature. In a digital device, information is represented using a discrete set of symbols, like the digits 0 through 9, rather than using some continuously varying physical property. We will see that this difference has profound consequences.

1.2 Now I Know My A, B, C's

While we have claimed that the properties that make digital devices digital have nothing to do with numbers, we haven't yet encountered an example of a digital device that doesn't use digits. It would probably help the case quite a bit to see one.

Imagine a clock that displayed the time textually, as in:

EIGHT AFTER TEN

In our view, such a clock would be just as digital as a clock that used digits. Like the digital clock based on numeric digits, such a clock could not display all the times between 10:08 and 10:09. We could increase the number of different times it could express by increasing the number of characters that could appear on its display. With a big enough display the clock might say things like:

THIRTY FIVE SECONDS AFTER EIGHT MINUTES AFTER TEN

However, once the display size is fixed (as it must be in a physical clock) there would be a finite set of discrete times the clock could express.

You may feel somewhat cheated by this example because, although we are using text now, most of the words in the text represent numbers. To recognize that the numerical nature of the information represented by these words is irrelevant, look at the letters instead of the words.

Notice that the letters (just like the digits) are discrete. There is nothing halfway in between an A and a B (unless your handwriting is quite poor). Furthermore, there is no clear relationship between the individual letters and the numbers they may be used to represent as words. In particular, there is no inherent relationship between any single letter and any of the "number words" it may be used to form. Even when the letters are used to form words that have nothing to do with numbers, they still have the "nothing-in-between" property that we claim distinguishes a digital representation scheme. In our view, therefore, the text in all the books in any library makes an excellent example of digital technology. The credit, then, for the notion of representing information digitally doesn't belong to some computer pioneer like John von Neumann, but to the ancient Phoenicians or whichever ancient culture rightly deserves recognition for introducing the use of an alphabet as the basis for written language.

In saying this, I want to make sure that you appreciate the distinction between the development of written language and the introduction of an alphabet. The first written languages used a different pictorial symbol for each word. Egyptian hieroglyphics are probably the best known example. In such a language it is difficult to say exactly how many symbols are used to represent information. Each word requires its own symbol. In a living language, new words are invented as needed. So, the number of symbols used in writing also grows as needed if a distinct pictorial symbol is used for each word.

Languages based on alphabets are very different in this regard. If you ask how many symbols are used to represent words in English, you can give a definite answer. It will be greater than 26 (to account for apostrophes, commas, etc.), but it doesn't grow each time a word is added to the language. The vocabulary of the English language has certainly grown since you were born, but the alphabet hasn't.

This fact is key to the relationship between the "digits" in the obviously digital representation of time in a digital clock and the "letters" in what I claim to be the equally digital representation of written language. In both cases, a relatively small set of symbols are used together to form larger units that can represent vast amounts of information. In both cases, the symbols used are quite distinct from one another with no notion of "in-between" symbols.

We will call any such set of symbols an alphabet. In addition to applying this term to the English alphabet, the Russian alphabet and alphabets used by other human languages, we will apply it to any sets of discrete symbols used to represent information. Thus, we will think of the set of digits 0 through 9 as an alphabet. The term alphabet will also apply to non-written physical systems for representing information. The 12 tones used to signal which button is pressed when you "dial" a touch-tone phone for example form an alphabet.

1.3 Digital Data, Approximation and Distortion

To help you appreciate the difference between digital and analog representations of information, consider the impact that choosing one of these encoding techniques over the other has on the accuracy of the information ultimately recorded or transmitted.

We have already hinted at the fact that choosing a digital representation may result in some loss of information due to the discrete nature of the encoding. For example, when using a digital clock, there is no way to set the clock exactly to the time 20 seconds past 10:09. It would require an infinite number of digits in the clocks display (10:09:333333333...). In practice, most digital clocks force us to approximate the time to the nearest minute (or at least the nearest second). By contrast, an analog clock can (in theory) be set precisely to any time we desire. (In reality, the limits of our vision and our ability to control the placement of the clock hands with the little knob on the clock make setting the clock to anything more precise than the nearest 15 seconds unlikely. This inaccuracy, however, is not inherent in the scheme chosen to represent the time.)

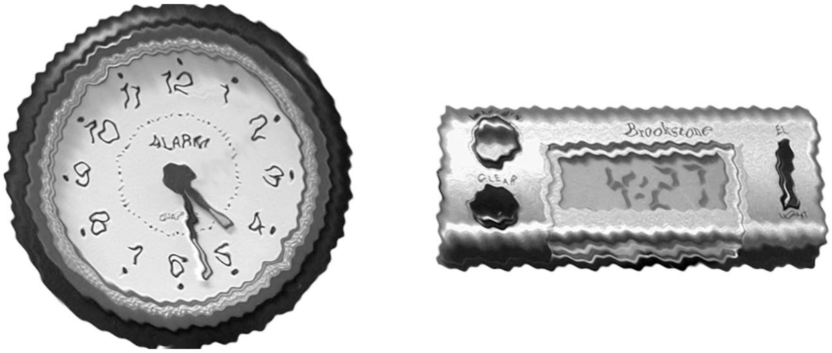
Both digital and analog schemes for representing data involve manipulating physical objects to force them into states that encode information. These physical objects are always subject to the "ravages of nature" so that they will frequently be moved out of the carefully chosen configurations that represented the information we had wished to encode. For example, we might carefully write a message on a piece of paper only to have the ink smear when the paper is carried outside on a rainy day. Such "damage" to encoded information is what we will call distortion. Distortion is a particularly significant threat when encoded information is being transmitted from one location to another. Static on you radio, for example, is a very common example of the distortion of an analog signal during its transmission.

The fact that digital representations use a finite set of symbols makes them far more resilient to distortion. When we try to interpret a distorted digital message, we may not see symbols that look exactly like we expected, but because there are a relatively small number of possibilities, we can identify the symbol closest in appearance to a distorted symbol and assume with reasonable accuracy that this closest symbol was the intended symbol. In an analog representation, however, there are an infinite number of possibilities. A distorted signal is very likely to correspond to one of the valid "in-between" states. We may get close to the intended interpretation, but we are almost certain not to retrieve the exact information that had been encoded.

Returning to the example of distortion in the form of ink smeared by rain, imagine that the original message written on the paper had included the time of a meeting. Normally, this time would have been encoded digitally in a hand written message. It could, however, also be encoded in analog form by drawing a picture of a clock face. To appreciate the difference between the two, the images below are intended to approximate the appearance of "smeared" examples of each of the analog and digital representations of a particular time. The amounts of distortion applied to the two representations were equal.[1] Which one has the distortion made most difficult to interpret exactly?

The digital clock could never be expected to provide the time more accurately than to the nearest minute. That information has been preserved despite the distortion. In its undistorted form, the analog clock provided the time to at least the nearest minute. Can you confidently determine whether the analog clock in the distorted picture reads 4:27 or 4:28?

[1] The smeared versions were produced by taking a perfectly clear image of an analog and digital clock and applying an Adobe Photoshop filter to the entire image.

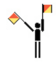


In general, with digital representations, we often have to introduce some approximation when initially encoding information. Once this is done, however, we can generally interpret the message without any further loss of information even if the encoding is distorted. Analog representations, on the other hand, can encode information exactly. Any distortion, however will lead to some error when the information is interpreted.

1.4 Now I know my a,b,c's

The next step in understanding the importance of the use of an alphabet is to recognize that the characters used are unimportant. If I got tired of the letter "t" and decided to replace all the "t"s in my writing by some alternate symbol like "@", i@ would no@ change @he informa@ion represen@ed by @he @ex@ I wro@e. One could simply go @hrough my @ex@ and replace all @he "@"s with plain old "t"s and my meaning would again become clear.

In fact, I could pick a new symbol for each letter from A to Z and use my new symbols in place of the old symbols and still not lose any information. Having been patiently taught to read and write the symbols of our own alphabet from childhood, we would find such writing much more difficult to understand than the original. The "information content", however, would be the same. If one patiently replaced every letter in my personal alphabet with the corresponding letter in the standard alphabet the meaning of the text would again be apparent.

While making up your own alphabet would be a bit eccentric, equivalent translations between alphabets have been used practically in many situations. For example, to signal between ships, sailors have developed systems for using flags to send messages. In one of these systems, the person sending the signals holds two identical flags in each hand. For each letter of the alphabet, there is a specified position to hold the flags. A chart of these positions is shown in Figure 1.1. Essentially, just as I suggested using the symbol "@" to replace "t"s, the signal flag system replaces "t"s with  's.

Clearly, translating a message from the standard alphabet to the signal code alphabet preserves all the information in the original message. This is the important test. Our choice of alphabet may make it easier or harder to understand some information we want to communicate, but if it is possible to translate exactly between one alphabet and another and back again there will be no information loss.

To appreciate this better, we might consider a change in alphabet that would lead to information loss. Suppose I was so tired of "t"s that I decided to leave them out completely. Again, his would

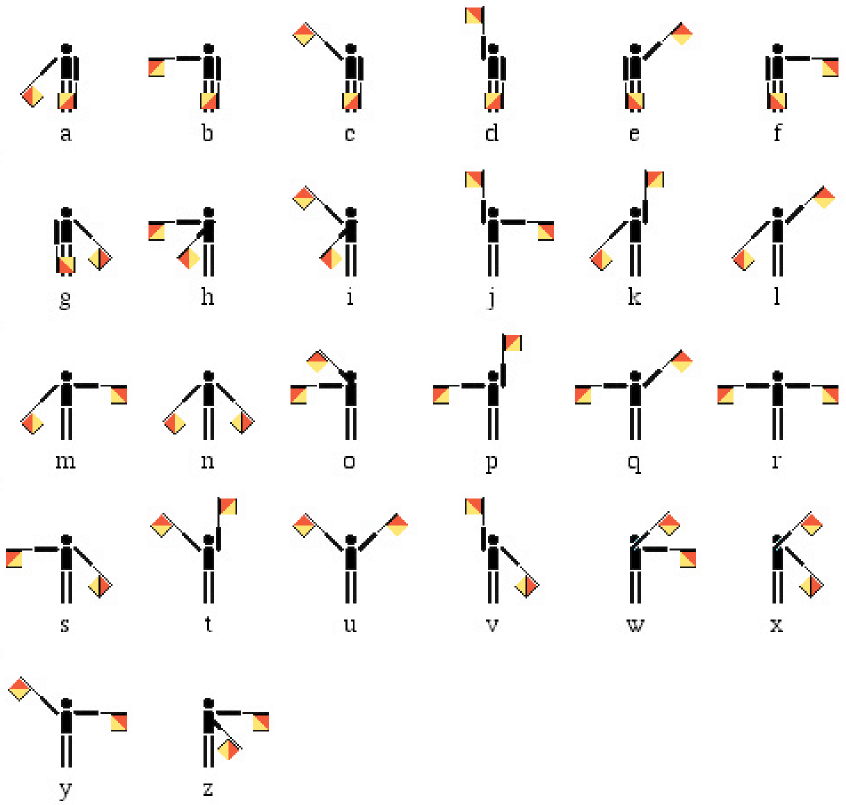


Figure 1.1: Semaphore Signal Flag Interpretations.

immediaely make my ex more difficul o comprehend. Wih a bi of effor, one would sill be able o resore mos of he missing leers.

In some cases, however, it might be impossible to precisely restore the original text given only a copy of the text with all the “t”s removed. A very simple example is the phrase “his ha is red”. The original phrase could have been “this hat is red” or “his hat is red.” Unless additional context made the intent clear, the deletion of the “t” would result in information loss.

This should not be at all surprising. One would expect to be able to represent more information using 26 letters than one could represent using 25 letters. The surprise is that this expectation is false. With a bit more care we can get rid of all the “t”s without losing any information. The trick is to replace the “t”s with other symbols from the remainder of the standard alphabet rather than just deleting them.

A simple way to make “t”s unnecessary is to represent each “t” with some combination of other letters. For example, I could replace all the “t”s in my text with something unusual like a pair of “z”s. This will work fine most of the time. The only problem will be words which turn into other “real” words when we make such a substitution. For example, “but” would become “buzz”. This means that if we run into the word “buzz” when trying to read some “t”-less text we may have problems deciding whether the intended word is “but” or “buzz”.

We can solve this problem by doing something to all the real “z”s in the original text to make them distinguishable from the pairs of “z”s created to replace “t”s. For example, we could replace any “z” in the original text by the pair “za”.

As an example, the phrase:

try a buzz cut

would be rewritten as

zzry a buzaza cuzz

If we make both of these substitutions, “zz” for every “t” and “za” for every “z”, the resulting text will use an alphabet that is one symbol smaller than the original, but no information will have been lost. We can restore the original message quite simply. Just read through the “t”-less text looking for “z”s. Each “z” will either be followed by another “z” or an “a”. In the first case, turn the “zz” into a “t”. In the second case, just delete the “a”.

1.5 Can You Count to 2?

Getting rid of t’s may not seem like an earth shaking accomplishment. Clearly, no one would be willing to read your writing if it looked like “zzry a buzaza cuzz”. What then is the point of even discussing such a scheme?

First, there is a very important idea about digital communications that our scheme for replacing t’s illustrates. There are generally many distinct ways of representing a given kind of information. Each way may have its own advantages and disadvantages. For example, one way may require a smaller alphabet while another may be easier to interpret. Despite these differences, the encoding schemes may be considered equivalent in the sense that a given unit of information can be represented precisely in any of the schemes.

This fact isn’t too relevant when the information being represented begins as simple English text. There really isn’t any good reason to do something silly like eliminate all the t’s in a document. As soon as the information gets just a bit more complicated, however, the fact that there are many potential ways to represent information (and no clear “right” way) becomes apparent and can have significant consequences.

As an example, note that word processors have to represent more than “simple text” when you save a document. Additional information about font sizes, margins, use of bold and italics, etc. all has to be encoded somehow. There is no one, right way to do this. As a result, different word processors use different schemes for encoding this information. You may have noticed that an application like Microsoft Word can’t read documents produced by WordPerfect (or even sometimes different versions of Microsoft Word) without the help of a special conversion program. The conversion program required to let one word processor work with a document created with another word processor does a translation from one representation to another similar to (but more complicated than) our example’s rule that replaces pairs of z’s with t’s and “za”s with z’s.

Our scheme for eliminating t’s from text also sheds light on an interesting question about alphabets. There clearly are alphabets of different sizes. We use 26 letters. The Romans only used 23 (no J, V or W). Given these differences, one might ask how many letters you really need. By exploring the trick we used to eliminate t’s further, we can see that one really only needs two letters in an alphabet.

To see this note that a key property of the letter reducing trick is that it can be repeated. After getting rid of t’s you could pick some other letter you don’t like and use the same basic scheme to eliminate it from your text.

First, think carefully about how we got rid of t's. To get rid of one letter, "t", we picked a second letter, "z", with the intent of using pairs of z's to represent t's. To make it work, we needed to choose a third letter, "a", to pair with "z" when we needed to unambiguously represent a "z" found in the original text.

So, suppose having already eliminated t's you now wanted to get rid of u's. First, you would need to pick a second letter, "r" for example, so that the pair "rr" could be used to replace all u's. Then, you would also have to pick a third letter, "k" for example, and designate the pair "rk" as the replacement for all r's found in the original text.

Recall our example:

> try a buzz cut

which became

> zzry a buzaza cuzz

after t's were eliminated. Eliminating u's from this phrase would produce:

> zzrky a brrzaza crazz

Pretty unreadable? Yes! By applying our substitutions in reverse, however, we can still get back to the original phrase. The transformed string may be longer and harder to read, but it encodes all the information found in the original.

Of course, we could do it again! We could get rid of another letter, and then another, and so on.

Could we get rid of all the letters? If you think about our scheme carefully, you will notice that it requires three letters. The one you want to get rid of, the one used in pairs to replace the first letter and the one used with the second letter to replace instances of the second letter found in the original text. Therefore, you can apply our trick over and over again until there are only two letters left. Then you are stuck.

This is a remarkable fact. Any information represented using any finite alphabet can be rewritten using an alphabet containing only two letters without losing any information!

This fact leads immediately to another even more remarkable fact. Any information represented using any finite alphabet can be rewritten using any other alphabet containing at least two letters. Just rewrite any information represented using the original alphabet into a two letter version. Then pick any two letters from the target alphabet and replace the two letters used in the two letter version. The result is a message in the target alphabet that only uses two of that alphabet's symbols.

You have probably heard that computers work in "binary", a system for writing numbers using only 0's and 1's. What is really going on here is that computer designers have chosen to be as efficient as possible when selecting an alphabet to represent information within the computer. Since a two letter alphabet is sufficient, computers only use two symbols, 0 and 1, internally. This makes designing computers simpler since the electronics required only have to know about two symbols. Because of the property of alphabets we have just discussed, however, it doesn't limit the computer's ability to process digital information. Any information expressed in any finite alphabet can be represented in the binary alphabet used within the computer without any information loss.

1.6 Universal Information Transport

Now that we have an understanding of what it means for information to be represented digitally, we can begin to consider some of the advantages this approach offers.

Transporting information is what communications in general and computer networks in particular are all about. As a result, it is appropriate to first observe that the nature of digital representations offers a significant benefit when constructing systems for communicating information.

The fact that anything expressed in one alphabet can be encoded in any other alphabet makes systems for transporting information very flexible. Basically, if you build a system for transmitting messages expressed in any alphabet, even the trivial binary alphabet, it will be sufficiently powerful to transmit any message written using any alphabet.

By contrast, the suitability of mechanisms for transporting physical objects often depends critically on the type of objects being transported. A boat trailer is a great way to bring your boat to the lake, but it can not take the place of a pipeline if your goal is to deliver oil any more than a pipeline could be used to carry a boat. It is not the case that every different type of physical cargo requires a distinct form of transportation. A wide variety of objects can be shipped in a standard tractor trailer. There are, however, many examples of physical cargoes that require specialized shipping equipment. Boats, oil, perishable foods, livestock, and people are just a few examples (try putting a house on a boat trailer or through a pipeline). This is not true of mechanisms for transporting digital information. They are all potentially universal.

You see the impact of the flexibility of mechanisms for communicating digital information each time you use a web browser. As we will see, the Internet is basically only capable of transmitting messages composed of 0's and 1's. While visiting web sites, you expect to see much more than 0's and 1's. You expect to encounter text, color images, sounds and other forms of information. Somehow, however, every form of information transmitted to your web browser through the Internet made the trip encoded as a sequence of 0's and 1's.