【Discrete Math】Day 1(2)

【Ch2】Let us count(2)

2.3 The number of subsets

What is the number of all subsets of a set with n elements?

Consider to list all the subsets of a set {a, b, c} with 3 elements:

$$\emptyset, \{a\}, \{b\}, \{c\}, \{a,b\}, \{b,c\}, \{a,c\}, \{a,b,c\}.$$

We can make a little table from these data:

No. of elements	0	1	2	3
No. of subsets	1	2	4	8

Suppose we have to select a subset of a set *A* with *n* elements; let us call these elements $a_1, a_2, ..., a_n$.

Then, we may or may not want to include a_1, in other words, we can make two possible decisions at this point.

No matter how we decide about a_1, we may or may not want to include a_2 in the subset; this means two possible decisions, and so the number of ways we can decide about a_1 and a_2 is $2 \times 2 = 4$.

We can go on similarly: no matter how we decide about the first k elements, we have two possible decisions about the next, and so the number of possibilities doubles whenever we take a new element.

Thus, we have proved the following theorem:

Theorem 2.1 *A set with n elements has 2^n subsets.*

We can illustrate the argument in the proof by the picture in Figure 1:

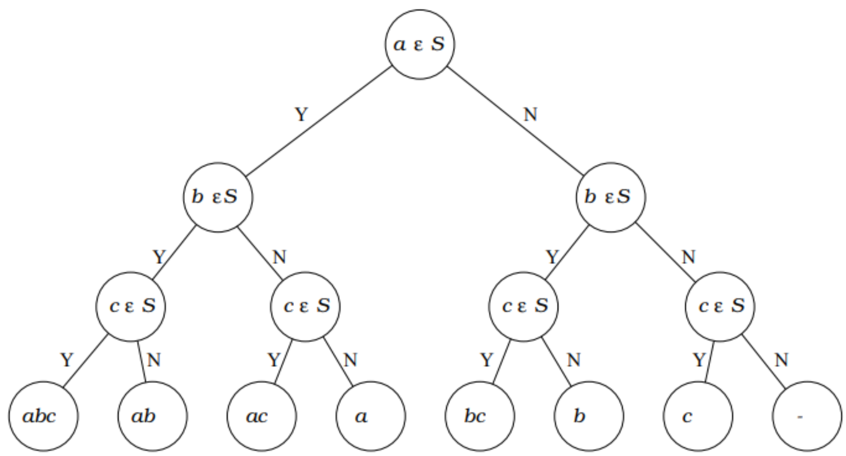

Figure 1: A decision tree for selecting a subset of $\{a, b, c\}$.

We want to select a subset called *S*. We start from the node on the top.

The node contains a question: *is a_1 an element of S?* The two arrows going out of this node are labelled with the two possible answers(Yes or No).

We make a decision and follow the appropriate arrow(also called an edge) to the node at the other end.

Since the number of nodes doubles from level to level as we go down, the last level contains $2^3 = 8$ nodes(and if we had an n-element set, it would contain 2^n nodes)

Binary Representation of Subsets

Let us start with another way of denoting subsets(another encoding in the mathematical jargon).

We look at the elements one by one, and write down a 1 if the element occurs in the subset an a 0 if it does not.

Thus, for the subset {a, c}, we write down 101, since a is in the subset, b is not, and c is in it again.

Now such strings consisting of 0's and 1's remind us of the binary representation of integers.

Thus, we see the subsets of {a, b, c} correspond to numbers 0, 1, ..., 7:

$$\begin{array}{ccccccc}
0 & \Leftrightarrow & 0_2 & \Leftrightarrow & 000 & \Leftrightarrow & \emptyset \\
1 & \Leftrightarrow & 1_2 & \Leftrightarrow & 001 & \Leftrightarrow & \{c\} \\
2 & \Leftrightarrow & 10_2 & \Leftrightarrow & 010 & \Leftrightarrow & \{b\} \\
3 & \Leftrightarrow & 11_2 & \Leftrightarrow & 011 & \Leftrightarrow & \{b,c\} \\
4 & \Leftrightarrow & 100_2 & \Leftrightarrow & 100 & \Leftrightarrow & \{a\} \\
5 & \Leftrightarrow & 101_2 & \Leftrightarrow & 101 & \Leftrightarrow & \{a,c\} \\
6 & \Leftrightarrow & 110_2 & \Leftrightarrow & 110 & \Leftrightarrow & \{a,b\} \\
7 & \Leftrightarrow & 111_2 & \Leftrightarrow & 111 & \Leftrightarrow & \{a,b,c\}
\end{array}$$

What happens if we consider subsets of n elements?

The subsets correspond to numbers 0, 1, ..., $2^n - 1$.

It is clear that the number of subsets is 2^n

Why do we learn two proofs of a single statement?

The first proof given above introduced the idea of breaking down the selection of a subset into independent decisions, and the representation of this idea by a tree.

The second proof introduced the idea of enumerating these subsets(labelling them with integers 0, 1, 2...).

We also saw that we established a correspondence between the objects we wanted to count(the subsets) and some other kinds of objects that we count easily(the numbers 0, 1, ..., $2^n - 1$). In this correspondence:

- For every subset, we had exactly one corresponding number, and
- For every number, we had exactly one corresponding subset.

A correspondence with these properties is called a one-to-one correspondence. If we can make a one-to-one between the elements of two sets, then they have the same number of elements.

So we know that the number of subsets of a 100-element set is 2^{100}. This is a large number, but how large?

We know that $2^3 = 8 < 10$, and hence $2^{99} < 10^{33}$. Therefore, $2^{100} < 2 \times 10^{33}$. Now 2×10^{33} is a 2 followed by 33 zeroes; it has 34 digits, and therefore 2^{100} has at most 34 digits.

We also know that $2^{10} = 1024 > 1000 = 10^3$, and hence$2^{100} > 10^{30}$, which means that 2^{100}has at least 30 digits.

If a number has k digits, we know that

$$10^{k-1} \leq 2^{100} < 10^k$$

Now, we can write 2^{100} in the form 10^x, only x will not be an integer: the appropriate value of x is $x = lg2^{100} = 100lg2$. we thus have:

$$k - 1 \leq x < k$$

which means that k-1 is the largest integer not exceeding x. It is the integer part or floor of x, and it is denoted by $\lfloor x \rfloor$. We can also say that we obtain k by round x down to the next integer.

There is also a name for the number obtained by rounding x up to the next integer: it is called the ceiling of x, and denoted by $\lceil x \rceil$.

We know that $lg2 = 0.30103$, thus $100lg2 = 30.103$, and rounding this down we get that $k - 1 = 30$. Thus, 2^{100} has 31 digits.