Counting by Mapping

Lecture 15: Nov 4
Plan

We will study how to define mappings to count.

There will be many examples shown.

- Bijection rule (this set of slides)
- Division rule (next set of slides)
- More mapping (next set of slides)
Counting Rule: Bijection

If \( f \) is a bijection from \( A \) to \( B \), then \(|A| = |B|\)

To compute \(|A|\), one strategy is to define a bijection of \( A \) and \( B \), where \( B \) is easier to count and we can compute \(|B|\) directly.
Power Set

How many subsets of a set $S$?

$P(S) =$ the power set of $S$

$= \text{the set of all subsets of } S$

for $S = \{a, b, c\}$,

$P(S) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a,b\}, \{a,c\}, \{b,c\}, \{a,b,c\} \}$

Suppose $S$ has $n$ elements.

How large is the power set of $S$?
Bijection: Power Set and Binary Strings

\[ S: \{s_1, s_2, s_3, s_4, s_5, \ldots, s_n\} \]

We define a bijection \( f \) between subsets and binary strings

\[ A: \text{the set of all subsets of } S \]

\[ B: \text{the set of all } n\text{-bit strings} \]

The mapping \( f: A \rightarrow B \) is defined in the following way:

Given a subset \( T \), we define \( f(T) \) as an \( n \)-bit string with the \( i \)-th bit equal to 1 if and only if \( s_i \) is in \( T \).
Bijection: Power Set and Binary Strings

\[ S : \{a, b, c\} \]

\[ P(S) \quad \text{3-bit strings} \]

- \( \emptyset \) maps to 000
- \{a\} maps to 001
- \{b\} maps to 010
- \{c\} maps to 011
- \{a, b\} maps to 100
- \{a, c\} maps to 101
- \{b, c\} maps to 110
- \{a, b, c\} maps to 111
Bijection: Power Set and Binary Strings

The mapping is defined in the following way:

subset: \{s_1, s_3, s_4, \ldots, s_n\}

string: 1 0 1 1 0 \ldots 1

This mapping is a bijection, because

- two different subsets are mapped to two different strings (injection)
- each binary string is mapped by some subset (surjection).

Therefore, \(|A| = |B|\), and \(|B|\) can be computed directly.

So, \(|P(S)| = |n\text{-bit binary strings}| = 2^n.\)
There is a bijection between Full Houses and sequences specifying:

1. The value of the triple, which can be chosen in 13 ways.
2. The suits of the triple, which can be selected in \((4 \ 3)\) ways.
3. The value of the pair, which can be chosen in 12 ways.
4. The suits of the pair, which can be selected in \((4 \ 2)\) ways.

\[
(2, \{\spadesuit, \heartsuit, \diamondsuit\}, J, \{\clubsuit, \diamondsuit\}) \leftrightarrow \{ 2\spadesuit, 2\heartsuit, 2\diamondsuit, J\clubsuit, J\diamondsuit \}
\]
\[
(5, \{\diamondsuit, \clubsuit, \heartsuit\}, 7, \{\heartsuit, \clubsuit\}) \leftrightarrow \{ 5\diamondsuit, 5\clubsuit, 5\heartsuit, 7\heartsuit, 7\clubsuit \}
\]

\(A\): the set of full houses

\(B\): the set of sequences which satisfy (1)-(4).
A Chess Problem

In how many different ways can we place a pawn (p), a knight (k), and a bishop (b) on a chessboard so that no two pieces share a row or a column?
We define a mapping $f$ between configurations to sequences $(r(p), c(p), r(k), c(k), r(b), c(b))$, where $r(p), r(k),$ and $r(b)$ are distinct rows, and $c(p), c(k),$ and $c(b)$ are distinct columns.

If we can define a bijection between $A$ and $B$, and also calculate $|B|$, then we can determine $|A|$.
We define a mapping $f$ between configurations to sequences $(r(p), c(p), r(k), c(k), r(b), c(b))$, where $r(p)$, $r(k)$, and $r(b)$ are distinct rows, and $c(p)$, $c(k)$, and $c(b)$ are distinct columns.

This is a bijection, because:
- no two configs map to same sequence (injection)
- every such sequence is mapped (surjection)

So, to count the number of chess configurations, it is equivalent to count the number of such sequences.
A Chess Problem

We define a mapping between configurations to sequences (r(p), c(p), r(k), c(k), r(b), c(b)), where r(p), r(k), and r(b) are distinct rows, and c(p), c(k), and c(b) are distinct columns.

Using the generalized product rule, there are 8 choices of r(p) and c(p), there are 7 choices of r(k) and c(k), there are 6 choices of r(b) and c(b).

Thus, total number of configurations = \((8 \times 7 \times 6)^2 = 112896\).
Counting Rule: Bijection

If \( f \) is a bijection from \( A \) to \( B \), then \( |A| = |B| \)

Steps:
1) Come up with \( B \).
2) Come up with \( f \).
3) Show \( f \) is a bijection.
4) Compute \( |B| \).

Usually the first two steps are more difficult.

Now we see some more interesting examples.
Counting Doughnut Selections

There are five kinds of doughnuts.

How many different ways to select a dozen doughnuts?

\[ A ::= \text{all selections of a dozen doughnuts} \]

Hint: define a bijection to some bit strings!
**Counting Doughnut Selections**

\[ A ::= \text{all selections of a dozen doughnuts} \]

\[ B ::= \text{all 16-bit binary strings with exactly four 1's.} \]

Define a bijection \( f \) between \( A \) and \( B \).

<table>
<thead>
<tr>
<th></th>
<th>Chocolate</th>
<th>Lemon</th>
<th>Sugar</th>
<th>Glazed</th>
<th>Plain</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>(none)</td>
<td>000000</td>
<td>00</td>
<td>000000</td>
<td>00</td>
</tr>
<tr>
<td>0011000000100100</td>
<td>1</td>
<td>1</td>
<td>00</td>
<td>1</td>
<td>00</td>
</tr>
</tbody>
</table>

Each doughnut is represented by a 0, and four 1’s are used to separate five types of doughnuts.
Counting Doughnut Selections

\[ A ::= \text{doughnuts selections} \]

\[ B ::= \text{all 16-bit strings with four 1's.} \]

\[
\begin{array}{cccccc}
12 & 0 & 0 & 0 & 0 & 0 \\
\text{Chocolate} & \text{Lemon} & \text{Sugar} & \text{Glazed} & \text{Plain} & \rightarrow \ 000000000000001111 \\
\end{array}
\]

\[
\begin{array}{cccccc}
2 & 3 & 0 & 3 & 4 & \\
\text{Chocolate} & \text{Lemon} & \text{Sugar} & \text{Glazed} & \text{Plain} & \rightarrow \ 0010001100010000 \\
\end{array}
\]

\[
\begin{array}{cccccc}
2 & 0 & 6 & 2 & 2 & \\
\text{Chocolate} & \text{Lemon} & \text{Sugar} & \text{Glazed} & \text{Plain} & \rightarrow \ 00110000000100100 \\
\end{array}
\]

\[
\begin{array}{cccccc}
0 & 0 & 0 & 0 & 12 & \\
\text{Chocolate} & \text{Lemon} & \text{Sugar} & \text{Glazed} & \text{Plain} & \rightarrow \ 11110000000000000 \\
\end{array}
\]
Counting Doughnut Selections

c chocolate, l lemon, s sugar, g glazed, p plain maps to

\[0^{c}10^{l}10^{s}10^{g}10^{p}\]

\[A ::= \text{all selections of a dozen doughnuts}\]

\[B ::= \text{all 16-bit binary strings with exactly four 1’s.}\]

This is a bijection because

• Two doughnut selections map to two different strings. (injection)
  (think of a doughnut selection as a sequence of 5 numbers \((c,l,s,g,p)\),
  consider the first number which is different in the two sequences).

• Every string in B corresponds to a valid doughnut selection. (surjection)
Counting Doughnut Selections

c chocolate, l lemon, s sugar, g glazed, p plain maps to

\[0^c 10^l 10^s 10^g 10^p\]

\(A := \text{all selections of a dozen doughnuts}\)

\(B := \text{all 16-bit binary strings with exactly four 1's}\).

\[|A| = |B| = \binom{16}{4}\]
Counting Loops

for i=1 to n do
  for j=1 to i do
    for k=1 to j do
      printf("hello world\n");

How many "hello world" will this program print?

\[ A ::= \text{all (i,j,k) triple} \]

\[ B ::= \text{all strings with n-1 zeroes and 3 ones.} \]
Counting Loops

for i=1 to n do
    for j=1 to i do
        for k=1 to j do
            printf("hello world\n");

There are n possible values for the i,j,k.

1  2  3  4  5  ...  n

Imagine there are n-1 separators for the n values.

If i=4, j=2, k=2, then there are two zeroes in 2 and one zero in 4.
Counting Loops

for i=1 to n do
   for j=1 to i do
      for k=1 to j do
         printf("hello world\n");

There are n possible values for the i,j,k.

1 2 3 4 5 … n

k j i ...

In general, the position of the first zero corresponds to the value of k. The position of the second zero corresponds to the value of j. The position of the third zero corresponds to the value of i.
Counting Loops

for i=1 to n do
    for j=1 to i do
        for k=1 to j do
            printf("hello world\n");

There are n possible values for the i,j,k.

1  2  3  4  5  ...  n

   |   |           |   |  ... |
   |   |           |   |  ... |
   |   |           |   |  ... |

This is a bijection (verify this!) between the values for i,j,k and the set of strings with n-1 ones and 3 zeros.

So, the program prints "hello world" exactly \( \binom{n+2}{3} \) times.
Counting Non-Negative Integer Solutions

How many solutions are there to the equation $x_1 + x_2 + x_3 + x_4 = 10$, where $x_1, x_2, x_3, x_4$ are nonnegative integers?

It is just like buying 10 doughnuts from 4 types.

$A ::= \text{all non-negative integer solutions } x_1 + x_2 + x_3 + x_4 = 10$

$B ::= \text{all 13-bit binary strings with exactly three 1's.}$

Think of there are 10 points to be distributed into 4 variables.

```
  x_1  x_2  x_3  x_4
  0    0    0    0
```

e.g. Suppose $x_1 = 3, x_2 = 5, x_3 = 2, x_4 = 0$,

the 3 ones are used for separations.
Counting Non-Negative Integer Solutions

How many solutions are there to the equation \(x_1 + x_2 + x_3 + x_4 = 10\), where \(x_1, x_2, x_3, x_4\) are nonnegative integers?

It is just like buying 10 doughnuts from 4 types.

\[A := \text{all non-negative integer solutions } x_1 + x_2 + x_3 + x_4 = 10\]

\[B := \text{all 13-bit binary strings with exactly three 1's.}\]

It is not difficult to verify that this is a bijection, by the same argument as in doughnut selections.

So, the are exactly \(\binom{13}{3}\) integer solutions.
How many integer solutions to $x_1 + x_2 + x_3 + x_4 = 14$ if each $x_i \geq 1$?

Set $x_i = y_i + 1$.

Consider the equation $y_1 + y_2 + y_3 + y_4 = 10$ where each $y_i \geq 0$.

There is a bijection between the solutions for $y$ and the solutions for $x$.

Therefore we can apply the previous result, and conclude that the answer is $\binom{13}{3}$.
How many integer solutions to \( x_1 + x_2 + x_3 + x_4 \leq 10 \) if each \( x_i \geq 0 \)?

Consider the equation \( y_1 + y_2 + y_3 + y_4 + y_5 = 10 \) where each \( y_i \geq 0 \).

There is a bijection between the solutions for \( y \) and the solutions for \( x \).

Therefore we can apply the previous result, and conclude that the answer is \( \binom{14}{4} \) because we need 4 ones and 10 zeroes.
Choosing Non-Adjacent Books

There are 20 books arranged in a row on a shelf.

How many ways to choose 6 of these books so that no two adjacent books are selected?

\[ A ::= \text{all selections of 6 non-adjacent books from 20 books} \]

\[ B ::= \text{all 15-bit binary strings with exactly six 1's.} \]
Choosing Non-Adjacent Books

\[ A ::= \text{all selections of 6 non-adjacent books from 20 books} \]

\[ B ::= \text{all 15-bit binary strings with exactly six 1's.} \]

Map each zero to a non-chosen book, each of the first five 1's to a chosen book followed by a non-chosen book, and the last 1 to a chosen book.

This is a bijection, because:

- Two different selections are mapped to two different strings (injection)
  (consider the first position that is different in the two selections)
- Every string in B corresponds to a valid selection (surjection)
Choosing Non-Adjacent Books

\[ A := \text{all selections of 6 non-adjacent books from 20 books} \]

\[ B := \text{all 15-bit binary strings with exactly six 1's}. \]

\[ |A| = |B| = \binom{15}{6} \]
Choosing Non-Adjacent Books

\[ A ::= \text{all selections of 6 non-adjacent books from 20 books} \]

\[ B ::= \text{the set of integer solutions to } x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 20 \]
\[ \text{where } x_1 \geq 1 \text{ and } x_2, x_3, x_4, x_5, x_6 \geq 2. \]

Here, each \( x_i \) represents to pick the \( x_i \) book after the previous chosen book.

\[
\begin{array}{cccccccccccccccc}
\text{gray} & \text{white} & \text{gray} & \text{gray} & \text{white} & \text{white} & \text{gray} & \text{white} & \text{white} & \text{gray} & \text{white} & \text{white} & \text{gray} & \text{white} & \text{white} & \text{gray} \\
\end{array}
\]

E.g. this configuration corresponds to \( x_1 = 1, x_2 = 4, x_3 = 2, x_4 = 5, x_5 = 4, x_6 = 4 \).

It is not difficult to check that this is a bijection.
Choosing Non-Adjacent Books

\[ A ::= \text{all selections of 6 non-adjacent books from 20 books} \]

\[ B ::= \text{the set of integer solutions to } x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 20 \]
\[ \text{where } x_1 \geq 1 \text{ and } x_2, x_3, x_4, x_5, x_6 \geq 2. \]

From slide 24 we learned that \(|B| = |C|\) for

\[ C ::= \text{the set of integer solutions to } x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 9 \]
\[ \text{where } x_1, x_2, x_3, x_4, x_5, x_6 \geq 0. \]

From slide 26 we learned that \(|C| = |D|\) for

\[ D ::= \text{the set of integer solutions to } x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 = 9 \]
\[ \text{where } x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0. \]

From slide 24 we learned that \(|D| = \binom{15}{6}\)
Exercises

What is the number of non-negative integer solutions to $x_1 + x_2 + \ldots + x_m = n$?

What is the number of integer solutions to $x_1 + x_2 + \ldots + x_m = n$
   where $x_1 \geq a_1$, $x_2 \geq a_2$, ..., and $x_m \geq a_m$?

What is the number of integer solutions to $x_1 + x_2 + \ldots + x_m < n$
   where $x_1 \geq a_1$, $x_2 \geq a_2$, ..., and $x_m \geq a_m$?
**Exercises**

How many “hello world” will this program print?

```c
for i=1 to n do
    for j=1 to i-1 do
        for k=1 to j-1 do
            printf(“hello world\n”);
```

What is the number of non-negative integer solutions to $x_1 + x_2 = n$ where $x_1 < a_1$ and $x_2 < a_2$? (Go to tutorial!)

(Hint 1: count the complement.)

(Hint 2: inclusion-exclusion.)