Division Rule, Counting Subsets

Michael L. Littman

CS 22 2020

March 11, 2020
Overview

The Division Rule (15.4)
   Another Chess Problem (15.4.1)
   Knights of the Round Table (15.4.2)

Counting Subsets (15.5)
   The Subset Rule (15.5.1)
   Bit Sequences (15.5.2)
Division rule

**Definition**: A $k$-to-1 function maps exactly $k$ elements of the domain to every element of the codomain.
Division rule

**Definition:** A $k$-to-1 function maps exactly $k$ elements of the domain to every element of the codomain.

In the “4-legged zoo” song, the function mapping each leg to its owner is 4-to-1.
Division rule

**Definition:** A $k$-to-1 function maps exactly $k$ elements of the domain to every element of the codomain.

In the “4-legged zoo” song, the function mapping each leg to its owner is 4-to-1.

- $A$: legs
Division rule

**Definition:** A \( k \)-to-1 function maps exactly \( k \) elements of the domain to every element of the codomain.

In the “4-legged zoo” song, the function mapping each leg to its owner is 4-to-1.

- \( A \): legs
- \( B \): animals
### Division rule

**Definition:** A \( k \)-to-1 function maps exactly \( k \) elements of the domain to every element of the codomain.

In the “4-legged zoo” song, the function mapping each leg to its owner is 4-to-1.

- **A:** legs
- **B:** animals
- **\( f \):** leg \( a \in A \) belongs to animal \( b \in B \)

Example: Highway counter counts axles. At the end of the day, it counted 2742 axles. How many buses? Mapping from axles to buses is 2-to-1, so axles is twice the buses. So, \( 2742 / 2 = 1371 \) buses.
Division rule

**Definition:** A \(k\)-to-1 function maps exactly \(k\) elements of the domain to every element of the codomain.

In the “4-legged zoo” song, the function mapping each leg to its owner is 4-to-1.

- \(A\): legs
- \(B\): animals
- \(f\): leg \(a \in A\) belongs to animal \(b \in B\)
- Every element in \(B\) is pointed to by 4 elements in \(A\)
Division rule

**Definition:** A \( k \)-to-1 function maps exactly \( k \) elements of the domain to every element of the codomain.

In the “4-legged zoo” song, the function mapping each leg to its owner is 4-to-1.

- \( A \): legs
- \( B \): animals
- \( f \): leg \( a \in A \) belongs to animal \( b \in B \)
- Every element in \( B \) is pointed to by 4 elements in \( A \)

**Rule:** If \( f : A \to B \) is \( k \)-to-1, then \( |A| = k|B| \).
Division rule

**Definition:** A $k$-to-1 function maps exactly $k$ elements of the domain to every element of the codomain.

In the “4-legged zoo” song, the function mapping each leg to its owner is 4-to-1.

- $A$: legs
- $B$: animals
- $f$: leg $a \in A$ belongs to animal $b \in B$
- Every element in $B$ is pointed to by 4 elements in $A$

**Rule:** If $f : A \rightarrow B$ is $k$-to-1, then $|A| = k|B|$.

**Example:** Highway counter counts axles.
Division rule

**Definition**: A $k$-to-1 function maps exactly $k$ elements of the domain to every element of the codomain.

In the “4-legged zoo” song, the function mapping each leg to its owner is 4-to-1.

- $A$: legs
- $B$: animals
- $f$: leg $a \in A$ belongs to animal $b \in B$
- Every element in $B$ is pointed to by 4 elements in $A$

**Rule**: If $f : A \to B$ is $k$-to-1, then $|A| = k|B|$.

Example: Highway counter counts axles. At the end of the day, it counted 2742 axles.
Division rule

**Definition**: A \(k\)-to-1 function maps exactly \(k\) elements of the domain to every element of the codomain.

In the “4-legged zoo” song, the function mapping each leg to its owner is 4-to-1.

- \(A\): legs
- \(B\): animals
- \(f\): leg \(a \in A\) belongs to animal \(b \in B\)
- Every element in \(B\) is pointed to by 4 elements in \(A\)

**Rule**: If \(f : A \rightarrow B\) is \(k\)-to-1, then \(|A| = k|B|\).

Example: Highway counter counts axles. At the end of the day, it counted 2742 axles. How many buses?
Division rule

**Definition:** A $k$-to-1 function maps exactly $k$ elements of the domain to every element of the codomain.

In the “4-legged zoo” song, the function mapping each leg to its owner is 4-to-1.

- $A$: legs
- $B$: animals
- $f$: leg $a \in A$ belongs to animal $b \in B$
- Every element in $B$ is pointed to by 4 elements in $A$

**Rule:** If $f : A \rightarrow B$ is $k$-to-1, then $|A| = k|B|$.

Example: Highway counter counts axles. At the end of the day, it counted 2742 axles. How many buses? Mapping from axles to buses is 2-to-1, so axles is twice the buses.
Division rule

**Definition:** A \( k \)-to-1 function maps exactly \( k \) elements of the domain to every element of the codomain.

In the “4-legged zoo” song, the function mapping each leg to its owner is 4-to-1.

- **A:** legs
- **B:** animals
- **\( f \):** leg \( a \in A \) belongs to animal \( b \in B \)
- Every element in \( B \) is pointed to by 4 elements in \( A \)

**Rule:** If \( f : A \to B \) is \( k \)-to-1, then \( |A| = k|B| \).

Example: Highway counter counts axles. At the end of the day, it counted 2742 axles. How many buses? Mapping from axles to buses is 2-to-1, so axles is twice the buses. So, \( 2742 / 2 = 1371 \) buses.
Rook placements

We counted placements of a pawn, knight, and bishop, so no two occupied the same row or column via the generalized product rule.
Rook placements

We counted placements of a pawn, knight, and bishop, so no two occupied the same row or column via the generalized product rule.

What about two identical rooks?
Rook placements

We counted placements of a pawn, knight, and bishop, so no two occupied the same row or column via the generalized product rule.

What about two identical rooks?

▶ A: all sequences \((r_1, c_1, r_2, c_2)\) where \(r_1\) and \(r_2\) are distinct rows and \(c_1\) and \(c_2\) are distinct columns.
Rook placements

We counted placements of a pawn, knight, and bishop, so no two occupied the same row or column via the generalized product rule.

What about two identical rooks?

- $A$: all sequences $(r_1, c_1, r_2, c_2)$ where $r_1$ and $r_2$ are distinct rows and $c_1$ and $c_2$ are distinct columns.

- $B$: all valid rook configurations.
Rook placements

We counted placements of a pawn, knight, and bishop, so no two occupied the same row or column via the generalized product rule.

What about two identical rooks?

- **A**: all sequences \((r_1, c_1, r_2, c_2)\) where \(r_1\) and \(r_2\) are distinct rows and \(c_1\) and \(c_2\) are distinct columns.

- **B**: all valid rook configurations.

- **f**: map the sequence \((r_1, c_1, r_2, c_2)\) to a configuration with one rook at row \(r_1\), column \(c_1\) and the other at row \(r_2\), column \(c_2\).

Note: \((6, 2, 4, 1)\) and \((4, 1, 6, 2)\) are actually the same because the rooks are identical.
Rook placements

We counted placements of a pawn, knight, and bishop, so no two occupied the same row or column via the generalized product rule. What about two identical rooks?

- **A**: all sequences \((r_1, c_1, r_2, c_2)\) where \(r_1\) and \(r_2\) are distinct rows and \(c_1\) and \(c_2\) are distinct columns.
- **B**: all valid rook configurations.
- **f**: map the sequence \((r_1, c_1, r_2, c_2)\) to a configuration with one rook at row \(r_1\), column \(c_1\) and the other at row \(r_2\), column \(c_2\).

Note: \((6, 2, 4, 1)\) and \((4, 1, 6, 2)\) are actually the same because the rooks are *identical*. 
Double counted

Our function $f$ maps exactly two sequences to every board configuration,
Double counted

Our function \( f \) maps \textit{exactly} two sequences to every board configuration, one assigning the rooks in either order.
Double counted

Our function $f$ maps *exactly* two sequences to every board configuration, one assigning the rooks in either order. Thus, $f$ is a 2-to-1 function.
Double counted

Our function $f$ maps exactly two sequences to every board configuration, one assigning the rooks in either order.

Thus, $f$ is a 2-to-1 function. By the division rule, $|A| = 2|B|$. 
Double counted

Our function $f$ maps \textit{exactly} two sequences to every board configuration, one assigning the rooks in either order.

Thus, $f$ is a 2-to-1 function. By the division rule, $|A| = 2|B|$. By the generalized product rule, $|A| = 8^2 \cdot 7^2 = 3136$. 
Double counted

Our function $f$ maps exactly two sequences to every board configuration, one assigning the rooks in either order.

Thus, $f$ is a 2-to-1 function. By the division rule, $|A| = 2|B|$. By the generalized product rule, $|A| = 8^2 \cdot 7^2 = 3136$. Since we double counted, the number of configurations is half that, 1568.
Circular ordering

If we’re going to seat 4 people around a table, we can do it in $4! = 24$ ways.
Circular ordering

If we’re going to seat 4 people around a table, we can do it in $4! = 24$ ways. For example, dinner with my spouse and kids could be: (Max, Michael, Lisa, Molly) or (Lisa, Molly, Max, Michael).
Circular ordering

If we’re going to seat 4 people around a table, we can do it in $4! = 24$ ways. For example, dinner with my spouse and kids could be: (Max, Michael, Lisa, Molly) or (Lisa, Molly, Max, Michael).

Note, however, these two orderings induce the same ordered-next-to-relations. Michael has Max to his left and Lisa to his right in both.
Circular ordering

If we’re going to seat 4 people around a table, we can do it in $4! = 24$ ways. For example, dinner with my spouse and kids could be: (Max, Michael, Lisa, Molly) or (Lisa, Molly, Max, Michael).

Note, however, these two orderings induce the same ordered-next-to-relations. Michael has Max to his left and Lisa to his right in both.

If we only care about ordered-next-to-relations, how many orderings are there?
Formal seating

- **A**: A *seating* is determined by the sequence of diners going clockwise around the table starting at the top seat.
Formal seating

▶ A: A *seating* is determined by the sequence of diners going clockwise around the table starting at the top seat. Seatings are formally equivalent permutations of the $n$ diners.
Formal seating

- **A**: A *seating* is determined by the sequence of diners going clockwise around the table starting at the top seat. Seatings are formally equivalent permutations of the $n$ diners.

- **B**: An *arrangement* is determined by the sequence of diners going clockwise around the table starting after diner number 1, so it is formally the same as the permutations of diners 2 through $n$. 
Formal seating

- **A**: A *seating* is determined by the sequence of diners going clockwise around the table starting at the top seat. Seatings are formally equivalent permutations of the $n$ diners.

- **B**: An *arrangement* is determined by the sequence of diners going clockwise around the table starting after diner number 1, so it is formally the same as the permutations of diners 2 through $n$.

- **$f$**: We can map each permutation $a \in A$ to an arrangement $b \in B$ by seating the first diner in the permutation at the top of the table, putting the second diner to his left, the third diner to the left of the second, and so forth all the way around the table.
Formal seating

A: A *seating* is determined by the sequence of diners going clockwise around the table starting at the top seat. Seatings are formally equivalent permutations of the $n$ diners.

B: An *arrangement* is determined by the sequence of diners going clockwise around the table starting after diner number 1, so it is formally the same as the permutations of diners 2 through $n$.

$f$: We can map each permutation $a \in A$ to an arrangement $b \in B$ by seating the first diner in the permutation at the top of the table, putting the second diner to his left, the third diner to the left of the second, and so forth all the way around the table.

This mapping is $n$-to-1, since all $n$ cyclic shifts of any seating $a$ maps to the same arrangement $b$. 
Seatings to arrangements

number of arrangements

- $|B| \quad \text{defn of } B$
- $|A|/n \quad \text{division rule}$
- $n!/n \quad \text{generalized product rule}$
- $(n-1)! \quad \text{factorial defn}$
Seatings to arrangements

number of arrangements

\[ = |B| \quad \text{defn of } B \]
\[ = |A|/n \quad \text{division rule} \]
\[ = n!/n \quad \text{generalized product rule} \]
\[ = (n - 1)! \quad \text{factorial defn} \]

I also think of it as not having a choice where the first person sits (because that decision establishes anchor of the arrangement).
Choose

How many subsets of an \( n \)-element set?
Choose

How many subsets of an \( n \)-element set? \( 2^n \).
Choose

How many subsets of an $n$-element set? $2^n$.
What if we insist that the subsets have $k$ elements?
Choose

How many subsets of an $n$-element set? $2^n$.

What if we insist that the subsets have $k$ elements?

▶ Room in luggage for 4 of my 20 shirts.
Choose

How many subsets of an \( n \)-element set? \( 2^n \).

What if we insist that the subsets have \( k \) elements?

- Room in luggage for 4 of my 20 shirts.
- Deal a 5-card hand from 52 cards.
Choose

How many subsets of an \( n \)-element set? \( 2^n \).

What if we insist that the subsets have \( k \) elements?

- Room in luggage for 4 of my 20 shirts.
- Deal a 5-card hand from 52 cards.
- Sundae with 3 ice cream choices from 32 flavors.
Choose

How many subsets of an \( n \)-element set? \( 2^n \).

What if we insist that the subsets have \( k \) elements?

- Room in luggage for 4 of my 20 shirts.
- Deal a 5-card hand from 52 cards.
- Sundae with 3 ice cream choices from 32 flavors.

\[
\binom{n}{k}
\]

The number of \( k \)-element subsets of an \( n \)-item set.
Choose

How many subsets of an \( n \)-element set? \( 2^n \).

What if we insist that the subsets have \( k \) elements?

- Room in luggage for 4 of my 20 shirts.
- Deal a 5-card hand from 52 cards.
- Sundae with 3 ice cream choices from 32 flavors.

\[
\binom{n}{k}
\]

The number of \( k \)-element subsets of an \( n \)-item set. “\( n \) choose \( k \)."
Choose

How many subsets of an \( n \)-element set? \( 2^n \).

What if we insist that the subsets have \( k \) elements?

- Room in luggage for 4 of my 20 shirts.
- Deal a 5-card hand from 52 cards.
- Sundae with 3 ice cream choices from 32 flavors.

\[
\binom{n}{k}
\]

The number of \( k \)-element subsets of an \( n \)-item set. “\( n \) choose \( k \).”

Answers: \( \binom{20}{4}, \binom{52}{5}, \binom{32}{3} \).
Special cases

3 laser pointers, choose one for lecture.
Special cases

3 laser pointers, choose one for lecture. How many possibilities?
Special cases

Special cases

In general, $n$ items choose 1?
Special cases

In general, \( n \) items choose 1? \( \binom{n}{1} = n \).
Special cases

In general, $n$ items choose 1? $\binom{n}{1} = n$.
3 HTAs, choose 3 for a staff meeting.
Special cases

In general, $n$ items choose 1? $\binom{n}{1} = n$.

3 HTAs, choose 3 for a staff meeting. How many possibilities?
Special cases

In general, $n$ items choose 1? \( \binom{n}{1} = n \).

3 HTAs, choose 3 for a staff meeting. How many possibilities? 1.
Special cases

In general, $n$ items choose 1? $\binom{n}{1} = n$.

3 HTAs, choose 3 for a staff meeting. How many possibilities? 1.
In general, $n$ items choose $n$?
Special cases

In general, $n$ items choose 1? $\binom{n}{1} = n$.

3 HTAs, choose 3 for a staff meeting. How many possibilities? 1.
In general, $n$ items choose $n$? $\binom{n}{n} = 1$. 
Deriving the subset rule

- $A$: all permutations of $n$ elements.
Deriving the subset rule

- **A**: all permutations of $n$ elements.
- **B**: separating $n$ elements into $k$ chosen and $n - k$ not chosen.
Deriving the subset rule

- **A**: all permutations of \( n \) elements.
- **B**: separating \( n \) elements into \( k \) chosen and \( n - k \) not chosen.
- **\( f \)**: Take permutation \( a \in A \) and map it to \( b \in B \) such that the first \( k \) elements in \( a \) are the chosen elements in \( b \).
Deriving the subset rule

- A: all permutations of $n$ elements.
- $B$: separating $n$ elements into $k$ chosen and $n - k$ not chosen.
- $f$: Take permutation $a \in A$ and map it to $b \in B$ such that the first $k$ elements in $a$ are the chosen elements in $b$.

The function $f$ is $k!(n - k)!$-to-1 because every rearrangement of the first $k$ and every rearrangement of the last $n - k$ go to the same $b \in B$. 
Deriving the subset rule

- **A**: all permutations of $n$ elements.
- **B**: separating $n$ elements into $k$ chosen and $n - k$ not chosen.
- **$f$**: Take permutation $a \in A$ and map it to $b \in B$ such that the first $k$ elements in $a$ are the chosen elements in $b$.

The function $f$ is $k!(n - k)!$-to-1 because every rearrangement of the first $k$ and every rearrangement of the last $n - k$ go to the same $b \in B$. 
Subset rule

By the division rule: \( n! = k!(n - k)! \binom{n}{k} \).
Subset rule

By the division rule: $n! = k!(n - k)! \binom{n}{k}$.

**Rule:**

$$\binom{n}{k} = \frac{n!}{k!(n - k)!}.$$
An identity

\[
\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}.
\]
An identity

\[
\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}.
\]

**Proof:** Algebraically, tricky. But, can reason it out in terms of the definitions!
An identity

\[ \binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}. \]

**Proof:** Algebraically, tricky. But, can reason it out in terms of the definitions!

\( \binom{n}{k} \): The number of different ways we can choose \( k \) items out of a set of \( n \).
An identity

\[
\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}.
\]

**Proof:** Algebraically, tricky. But, can reason it out in terms of the definitions!

\(\binom{n}{k}\): The number of different ways we can choose \(k\) items out of a set of \(n\).

Induction-style, we can choose our \(k\) items all from the first \(n - 1\) or
An identity

\[
{n \choose k} = {n-1 \choose k} + {n-1 \choose k-1}.
\]

**Proof:** Algebraically, tricky. But, can reason it out in terms of the definitions!

\( {n \choose k} \): The number of different ways we can choose \( k \) items out of a set of \( n \).

Induction-style, we can choose our \( k \) items all from the first \( n - 1 \) or we can choose the last item and \( k - 1 \) items from the first \( n - 1 \).
An identity

\[
\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}.
\]

**Proof:** Algebraically, tricky. But, can reason it out in terms of the definitions!

\(\binom{n}{k}\): The number of different ways we can choose \(k\) items out of a set of \(n\).

Induction-style, we can choose our \(k\) items all from the first \(n-1\) or we can choose the last item and \(k-1\) items from the first \(n-1\). That’s, \(\binom{n-1}{k} + \binom{n-1}{k-1}\). QED.
An identity

\[
\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}.
\]

**Proof:** Algebraically, tricky. But, can reason it out in terms of the definitions!

\(\binom{n}{k}\): The number of different ways we can choose \(k\) items out of a set of \(n\).

Induction-style, we can choose our \(k\) items all from the first \(n-1\) or we can choose the last item and \(k-1\) items from the first \(n-1\). That’s, \(\binom{n-1}{k} + \binom{n-1}{k-1}\). QED.

\[
\{ o,o,o,o,\ldots,o,o,o \} \text{ vs. } \{ o,o,o,o,\ldots,o,o,o \}
\]
k bits out of n

How many \( n \)-bit sequences contain exactly \( k \) ones?
k bits out of n

How many $n$-bit sequences contain exactly $k$ ones? We talked about the bijection between subsets of an $n$-element set and $n$-bit sequences.
**k bits out of n**

How many \( n \)-bit sequences contain exactly \( k \) ones? We talked about the bijection between subsets of an \( n \)-element set and \( n \)-bit sequences.

Example: A 4-element subset of \( \{x_1, x_2, \ldots, x_8\} \) and the associated 8-bit sequence:

\[
\{ x_2, x_4, x_7, x_8 \} \\
(0 1 0 1 0 0 1 1).
\]
**k bits out of n**

How many \(n\)-bit sequences contain exactly \(k\) ones? We talked about the bijection between subsets of an \(n\)-element set and \(n\)-bit sequences.

Example: A 4-element subset of \(\{x_1, x_2, \ldots, x_8\}\) and the associated 8-bit sequence:

\[
\{x_2, x_4, x_7, x_8\} \\
\begin{pmatrix}
0 & 1 & 0 & 1 & 0 & 0 & 1 & 1
\end{pmatrix}.
\]

By the subset rule, the number of \(n\)-bit sequences with exactly \(k\) 1s is \(\binom{n}{k}\).
Flashback

Remember this slide from last month?
Time to make the donuts

\[ A = \text{all ways to select a half-dozen doughnuts when three varieties are available} \]

\[ B = \text{all 8-bit sequences with exactly two 1s} \]

\[
\begin{align*}
0 & 0 & 0 & 0 \\
\text{Boston cream} & \text{coconut} & \text{glazed}
\end{align*}
\]

Put our 6 donuts into the three bins. Note: Every way we can choose donuts becomes a pattern. And, every pattern (with 6 donuts) corresponds to a valid choice.

Use 1s to mark the gaps.

\[
\begin{align*}
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
\text{Boston cream} & \text{coconut} & \text{glazed}
\end{align*}
\]

Now, squeeze: 00001100.

Every 8-bit pattern with exactly 2 ones is a valid donut order. Every valid donut order can be encoded with an 8-bit pattern with exactly 2 ones. Same size!
Time to solve the donuts!

\[ A = \text{all ways to select a half-dozen doughnuts when three varieties are available} \]

\[ B = \text{all 8-bit sequences with exactly two 1s} \]
Time to solve the donuts!

\[ A = \text{all ways to select a half-dozen doughnuts when three varieties are available} \]

\[ B = \text{all 8-bit sequences with exactly two 1s} \]

Now, we know that \[ |B| = \binom{8}{2} = 28 \], so 28 ways to select a half-dozen donuts when three varieties available!
Time to solve the donuts!

\[ A = \text{all ways to select a half-dozen doughnuts when three varieties are available} \]

\[ B = \text{all 8-bit sequences with exactly two 1s} \]

Now, we know that \[ |B| = \binom{8}{2} = 28 \], so 28 ways to select a half-dozen donuts when three varieties available! Closure!
Donuts and separators rule

**Rule**: All ways to select $n$ donuts when $k$ varieties are available:
Donuts and separators rule

**Rule:** All ways to select $n$ donuts when $k$ varieties are available:

$$
\binom{n + k - 1}{k - 1}
$$
Donuts and separators rule

**Rule**: All ways to select \( n \) donuts when \( k \) varieties are available:

\[
\binom{n + k - 1}{k - 1}
\]

There’s \( k - 1 \) separators (1 bits) and \( n \) donuts (0 bits).
Donuts and separators rule

**Rule:** All ways to select $n$ donuts when $k$ varieties are available:

\[
\binom{n + k - 1}{k - 1}
\]

There’s $k - 1$ separators (1 bits) and $n$ donuts (0 bits). Need to put the $k - 1$ separators into the $n + k - 1$ slots.