Product Rule, Bijection Rule

DISCRETE STRUCTURES II

DARRYL HILL

BASED ON THE TEXTBOOK:

DISCRETE STRUCTURES FOR COMPUTER SCIENCE: COUNTING, RECURSION, AND PROBABILITY

BY MICHIEL SMID

How Effective is My Password?

What makes a "good" password?

- Hard to guess
- Algorithm to guess a password:
  - Try every combination
- We want lots of possible combinations

How do we count all possible combinations?

Product Rule

Simple example

Strict password rules:

- 3 characters – Upper case, digit, lower case
- How many combinations is that?

Examples: A7d, G4s, P0x, etc.

One Upper Case Letter	One Digit (0-9)	One Lower Case Letter
$26 \times$	$10 \times$	26

Product Rule

We have to do 3 tasks:

1. Choose an upper case letter (26 ways to do this)
2. Choose a digit (10 ways to do this)
3. Choose a lower case letter (26 ways to do this)

One Upper Case Letter	One Digit (0-9)	One Lower Case Letter
$26 \times$	$10 \times$	26

$= 6760$

Important! How we do Task 1 does not affect the number of ways to do Task 2!

Likewise, once we have finished Task 1 and Task 2, these do not affect the number of ways we can do Task 3

These Tasks are *Independent*

Product Rule

Product Rule in general:

Procedure: We have a sequence of m tasks to be done in *order*.

For $i = 1, 2, 3, \dots, m$ let N_i be the number of ways to do task i, and N_i **does not depend** on how the previous tasks were done.

Then the number of ways to do the Procedure is:

$$N_1 \times N_2 \times N_3 \times \cdots \times N_m$$

This rule is used *extensively* throughout the course.

Product Rule Example

$n \geq 1$ we want bitstrings of length n. How many are there?

A bitstring is a string of two symbols, 0's and 1's.

Example: if $n = 5$ then possible bitstrings are

01001
00000
11110
11111
etc.

Does this fit the product rule?

Product Rule:

Procedure: We have a sequence of m tasks to be done in *order*.

For $i = 1, 2, 3, \ldots, m$ let N_i be the number of ways to do task i, and N_i does not depend on how the previous tasks were done.

Then the number of ways to do the Procedure is:

$$N_1 \times N_2 \times N_3 \times \cdots \times N_m$$

Product Rule Example

How many bitstrings of length n, $n \geq 1$ are there?

What is the procedure?

Procedure: Write a bitstring one bit at a time from left to right.

Tasks: For $i = 1, 2, \ldots, n$ task i = write 0 or 1.

$N_i = 2$ since there are two ways to do task i

Notice if we have already done tasks $1 \,..\, i - 1$, then task i still has two ways to do it

1110001☐ - still two choices, 0 or 1

Since we have a sequence of n tasks and there is no dependency, we can apply the product rule

Product Rule Example

How many bitstrings of length n, $n \geq 1$ are there?

What is the procedure?

Procedure: Write a bitstring one bit at a time from left to right.

Tasks: For $i = 1, 2, \ldots, n$ task i = write 0 or 1.

$N_i = 2$ since there are two ways to do task i

Total number of ways to write a bitstring of length 5 is

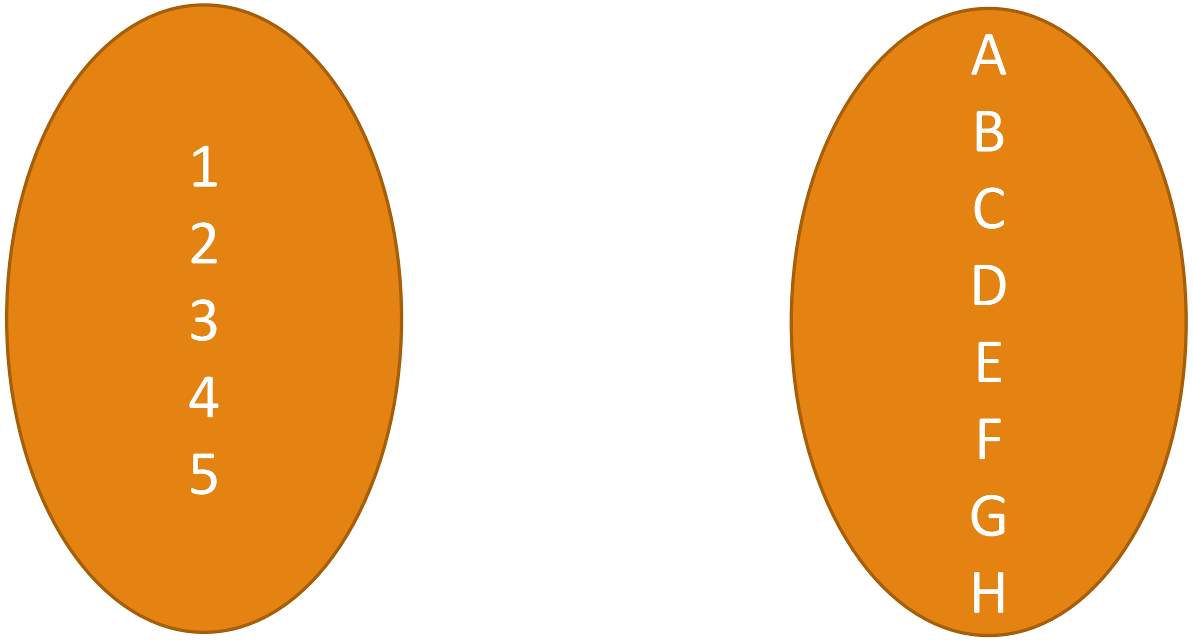
$= N_1 \cdot N_2 \cdot N_3 \cdot N_4 \cdot N_5$
$= 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2$
$= 32$

Product Rule Example

Sets A and B, $|A| = m, |B| = n.$ How many functions $f: A \to B$?

(Recall the definition of a function)

Can we define all functions in a way that uses the product rule?

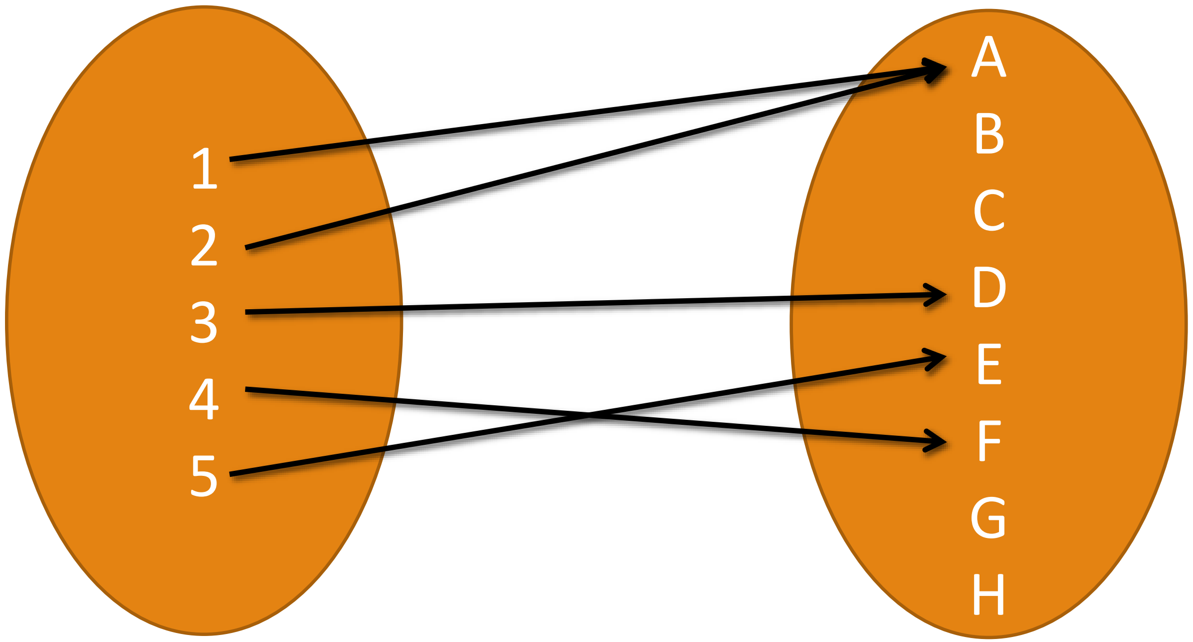


Product Rule Example

Sets A and B, $|A| = m, |B| = n$. How many functions $f: A \to B$?

Each element in A is mapped to exactly one element in B.

Example function:



Product Rule Example

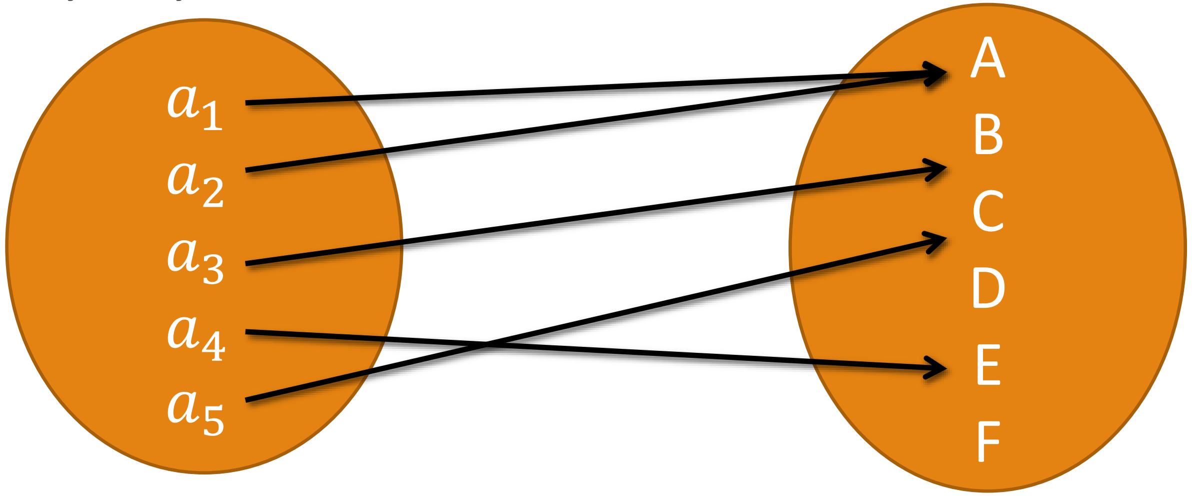
Sets A and B, $|A| = m, |B| = n.$ How many functions $f: A \to B$?

(Let's generalize the elements.)

What procedure did we use to create this? What are the tasks?

Procedure: $\forall x \in A$, specify $f(x)$ (assign an element from B).

Tasks: For each element $a_i \in A, i = 1 \ldots m$, task i = choose an element $b \in B$ such that $f(a_i) = b$. How many ways can we do this?



Product Rule Example

Sets A and B, $|A| = m, |B| = n$. How many functions $f: A \to B$?

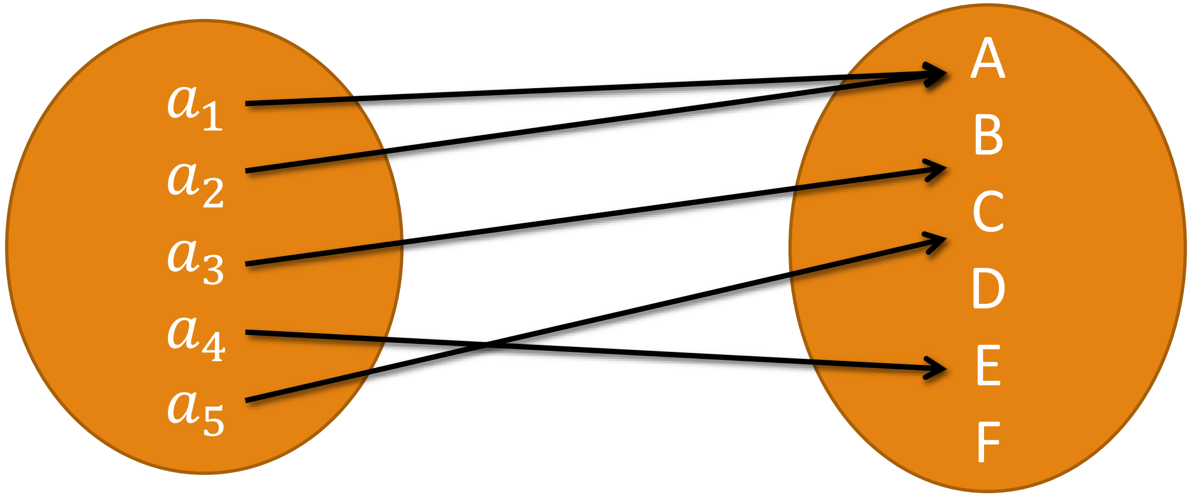
Procedure: $\forall x \in A$, specify $f(x)$ (assign an element from B).

Tasks: For each element $a_i \in A, i = 1 \dots m$, task i = choose an element $b \in B$ such that $f(a_i) = b$.

$N_1 = n, N_2 = n, N_3 = n, \dots, N_m = n$

$N_1 \cdot N_2 \cdot N_3 \cdot \dots \cdot N_m = n \cdot n \cdot \dots \cdot n = n^m$

In general: $|B|^{|A|}$

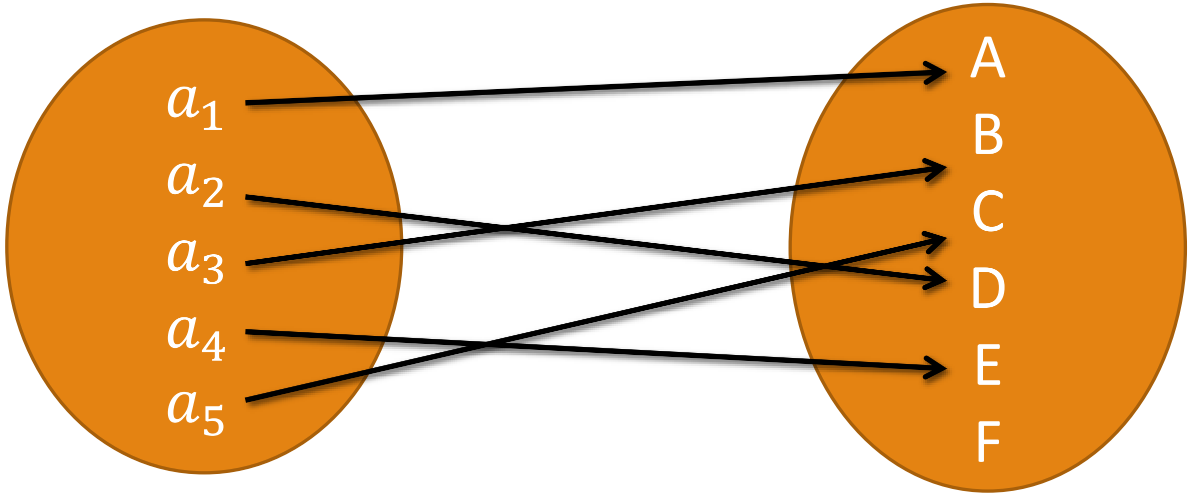


Product Rule Example

Sets A and B, $|A| = m, |B| = n$. How many **injective** functions $f: A \rightarrow B$?

Procedure: $\forall x \in A$, specify $f(x)$ (assign an element from B).

Tasks: For each element $a_i \in A, i = 1 \dots m$, task i = choose an element $b \in B$ such that $f(a_i) = b$ and b has not been selected yet. How many ways can we do this?



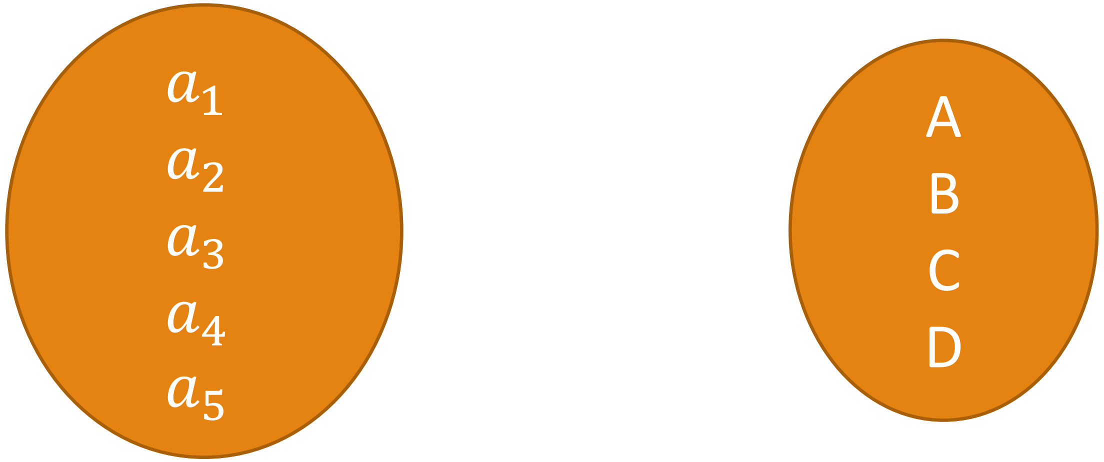
Product Rule Example

Sets A and B, $|A| = m, |B| = n$. How many **injective** functions $f: A \to B$?

Procedure: $\forall x \in A$, specify $f(x)$ (assign an element from B).

Tasks: For each element $a_i \in A, i = 1 \dots m$, task i = choose an element $b \in B$ such that $f(a_i) = b$ and b has not been selected yet. How many ways can we do this?

What if $|B| < |A|$? How many 1-to-1 functions are there?

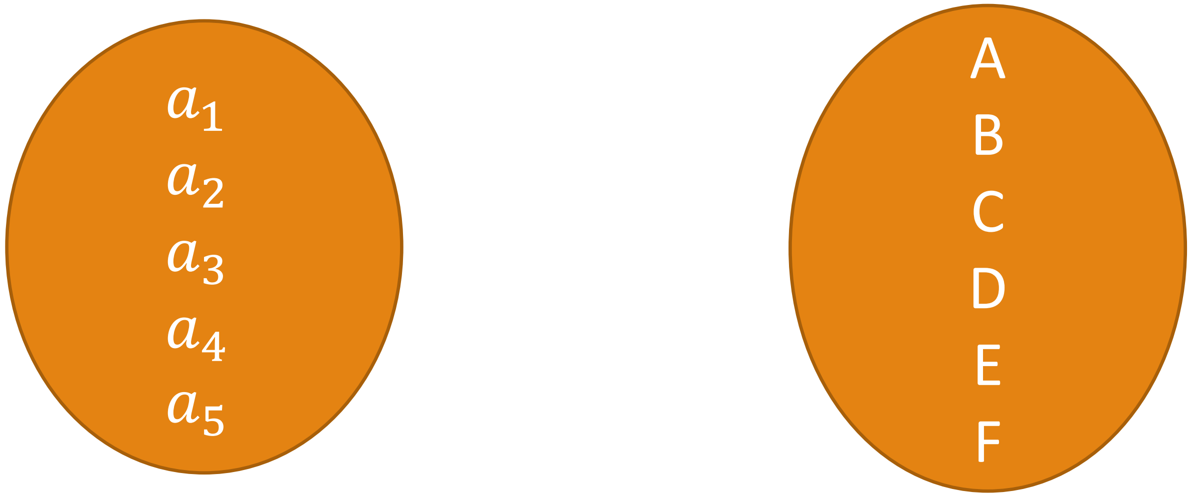


Product Rule Example

Sets A and B, $|A| = m, |B| = n$. How many **injective** functions $f: A \to B$? $|B| \geq |A|$

Procedure: $\forall x \in A$, specify $f(x)$ (assign an element from B).

Tasks: For each element $a_i \in A, i = 1 \dots m$, task i = choose an element $b \in B$ such that $f(a_i) = b$ and b has not been selected yet. How many ways can we do this?



Product Rule Example

Sets A and B, $|A| = m, |B| = n$. How many **injective** functions $f: A \to B$? $|B| \geq |A|$

Procedure: $\forall x \in A$, specify $f(x)$ (assign an element from B).

Tasks: For each element $a_i \in A, i = 1 \dots m$, task i = choose an element $b \in B$ such that $f(a_i) = b$ and b has not been selected yet.

$N_1 = n$
$N_2 = n - 1$
$N_3 = n - 2$
…
$N_{m-1} = n - (m-2) = n - m + 2$
$N_m = n - m + 1$

Product Rule Example

Sets A and B, $|A| = m, |B| = n$. How many **injective** functions $f: A \to B$? $|B| \geq |A|$

$$N_1 \cdot N_2 \cdot N_3 \cdot \ldots \cdot N_m = n \cdot (n-1) \cdot \ldots \cdot (n-m+2) \cdot (n-m+1)$$

How can we express this? Recall factorials:

$0! \;=\; 1$

if $k \geq 1$, then $k! = 1 \cdot 2 \cdot \ldots \cdot k$

$$5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5$$
$$3! = 1 \cdot 2 \cdot 3$$

Can we manipulate factorials into the expression $(n-m+1) \cdot (n-m+2) \cdot \ldots \cdot (n-1) \cdot n$?

Product Rule Example

Sets A and B, $|A| = m, |B| = n$. How many **injective** functions $f: A \to B$? $|B| \geq |A|$

How do we get from:

$$n! = 1 \cdot 2 \cdot \ldots \cdot (n-m) \cdot (n-m+1) \cdot (n-m+2) \cdot \ldots \cdot (n-1) \cdot n$$

to

$$(n-m+1) \cdot (n-m+2) \cdot \ldots \cdot (n-1) \cdot n$$

Divide by the first part:

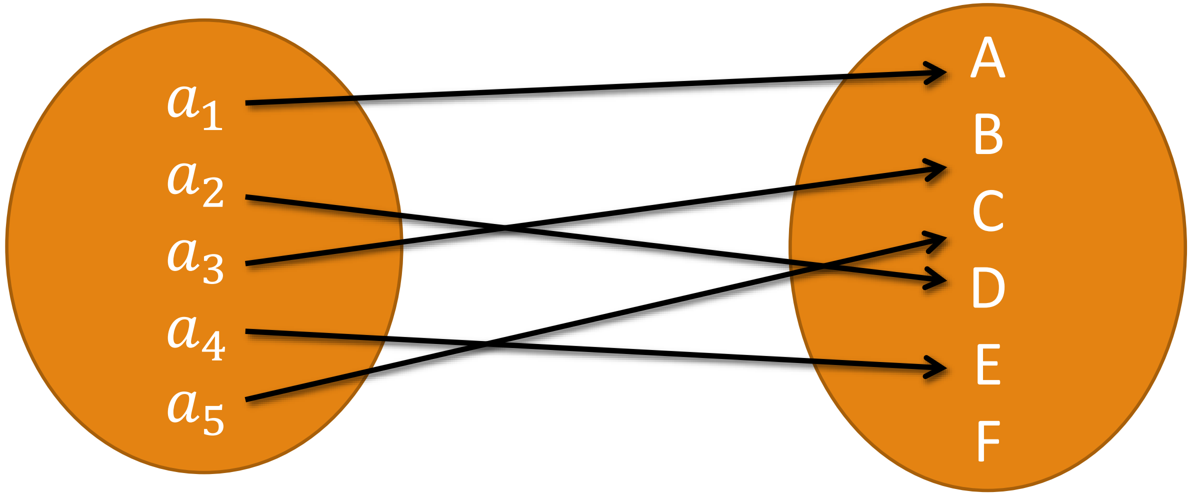
$$\frac{1 \cdot 2 \cdot \ldots \cdot (n-m) \cdot (n-m+1) \cdot (n-m+2) \cdot \ldots \cdot (n-1) \cdot n}{1 \cdot 2 \cdot \ldots \cdot (n-m)}$$

$$= \frac{n!}{(n-m)!}$$

Product Rule Example

Sets A and B, $|A| = m, |B| = n.$ How many **injective** functions $f: A \to B$? $|B| \geq |A|$

There are $\frac{n!}{(n-m)!}$ injective functions $f: A \to B$, where $|A| = m, |B| = n$.



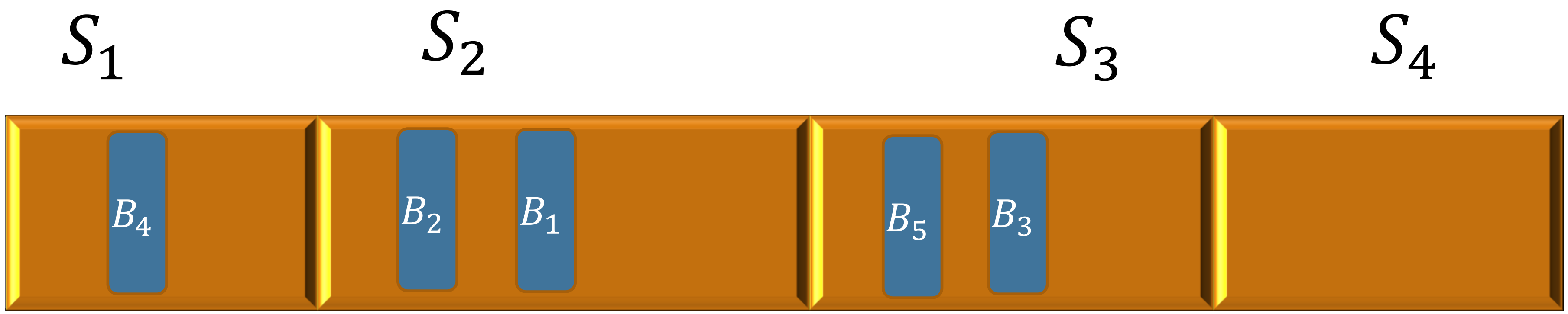
Review of Product Rule

Express as a procedure

Where a procedure is a **sequence of tasks**

The choice we make for Task *i* does not affect the number of ways to do Task *i+1*

Then we can apply the product rule



Books on Shelves

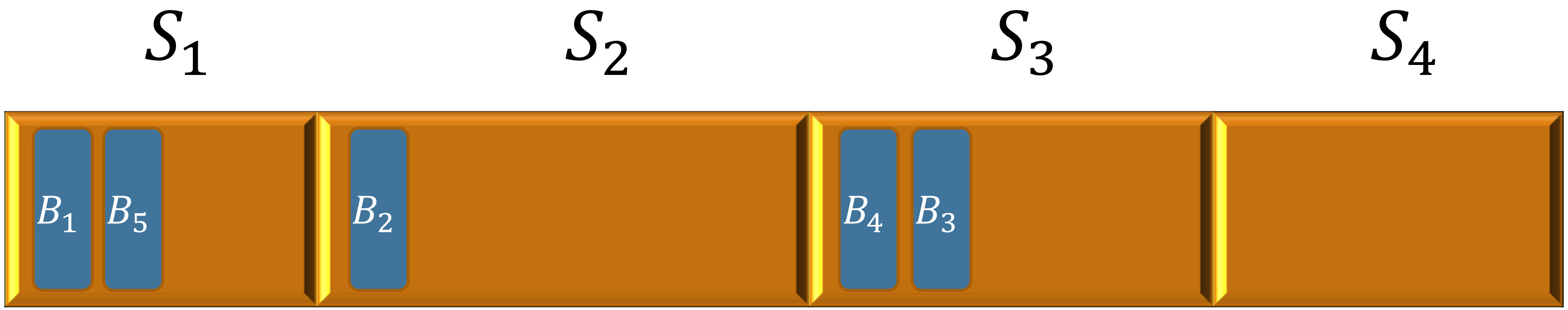
m books $B_1, B_2, B_3, \ldots, B_m$

n shelves $S_1, S_2, S_3, \ldots, S_n$

Procedure: Place books on shelves – two things to keep track of

1. For each book, which shelf
2. For each shelf: what is the left-to-right ordering of books on this shelf

Observe that this order is different from



Books on Shelves

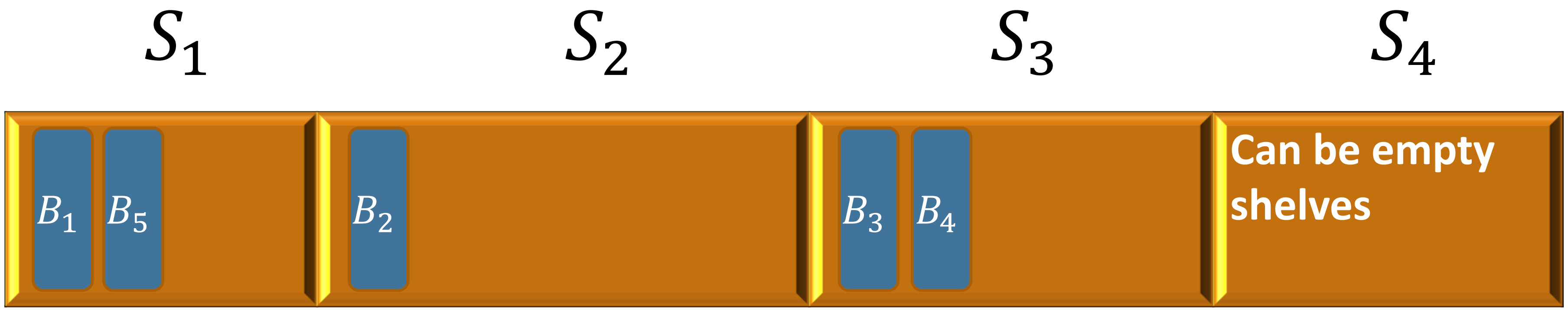
m books $B_1, B_2, B_3, \ldots, B_m$

n shelves $S_1, S_2, S_3, \ldots, S_n$

Procedure: Place books on shelves – two things to keep track of

1. For each book, which shelf
2. For each shelf: what is the left-to-right ordering of books on this shelf

Observe that this order is different from this order (B_3 and B_4 are switched)



Books on Shelves

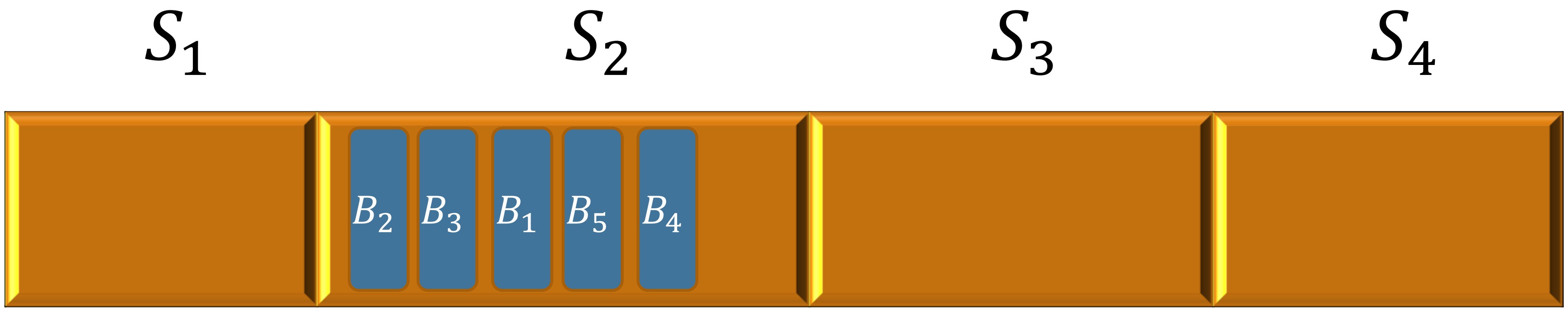
m books $B_1, B_2, B_3, \ldots, B_m$

n shelves $S_1, S_2, S_3, \ldots, S_n$

Procedure: Place books on shelves – two things to keep track of

1. For each book, which shelf
2. For each shelf: what is the left-to-right ordering of books on this shelf

Can have all books on one shelf (order still matters)



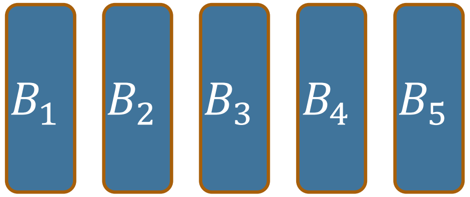
Books on Shelves

$m = 5$ books $B_1, B_2, B_3, \dots, B_m$
$n = 4$ shelves $S_1, S_2, S_3, \dots, S_n$

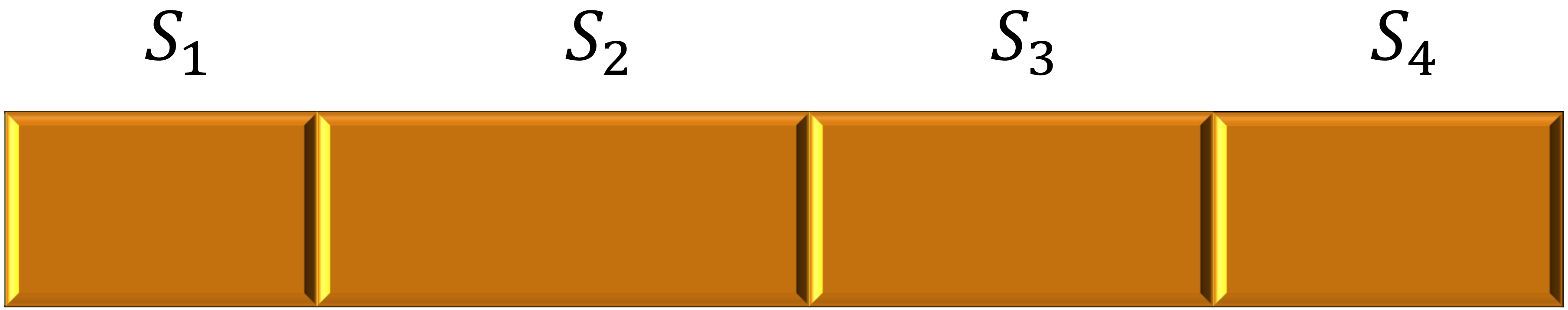
Can use the Product rule

Procedure: Start with empty shelves, place books one by one

Tasks: for $i = 1,2,\dots,m$ place B_i on a shelf ($B_1, B_2, \dots, B_{i-1}$ are already placed)



$N_1 = 4$



Books on Shelves

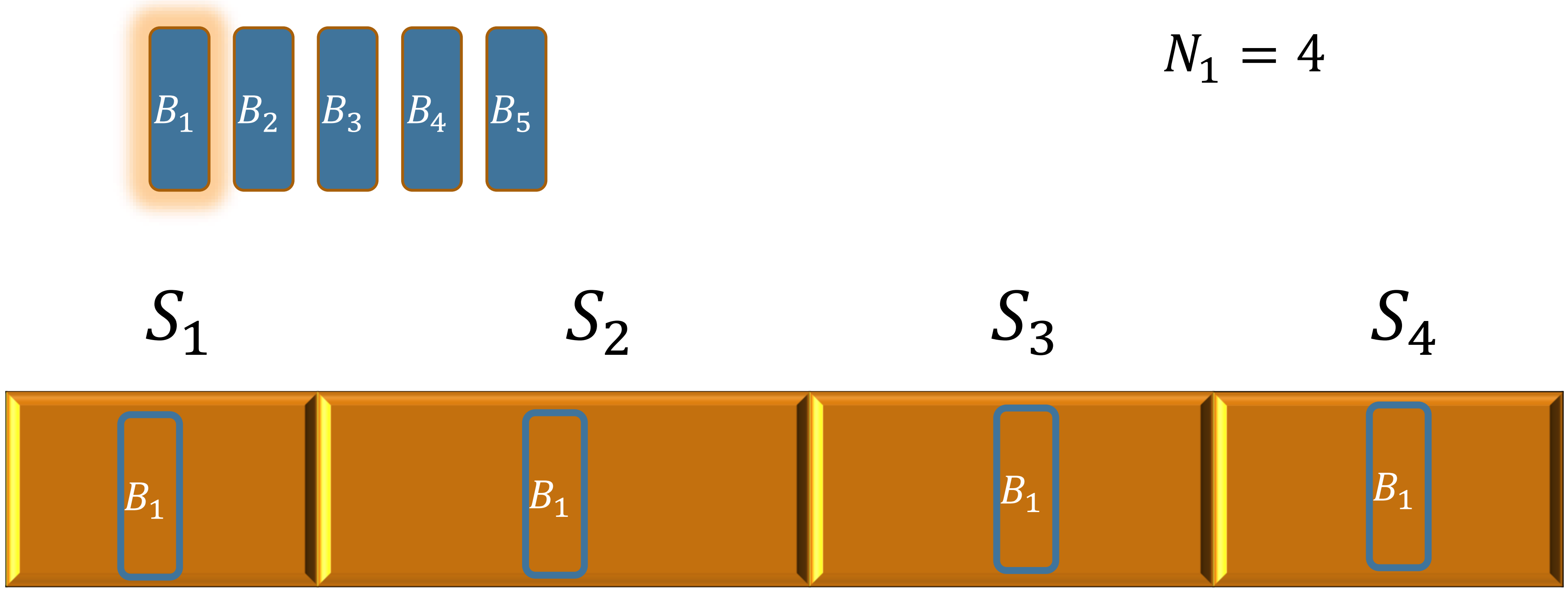
$m = 5$ books $B_1, B_2, B_3, \dots, B_m$
$n = 4$ shelves $S_1, S_2, S_3, \dots, S_n$

Can use the Product rule

Procedure: Start with empty shelves, place books one by one

Tasks: for $i = 1,2,\dots,m$ place B_i on a shelf ($B_1, B_2, \dots, B_{i-1}$ are already placed)

$N_1 = 4$



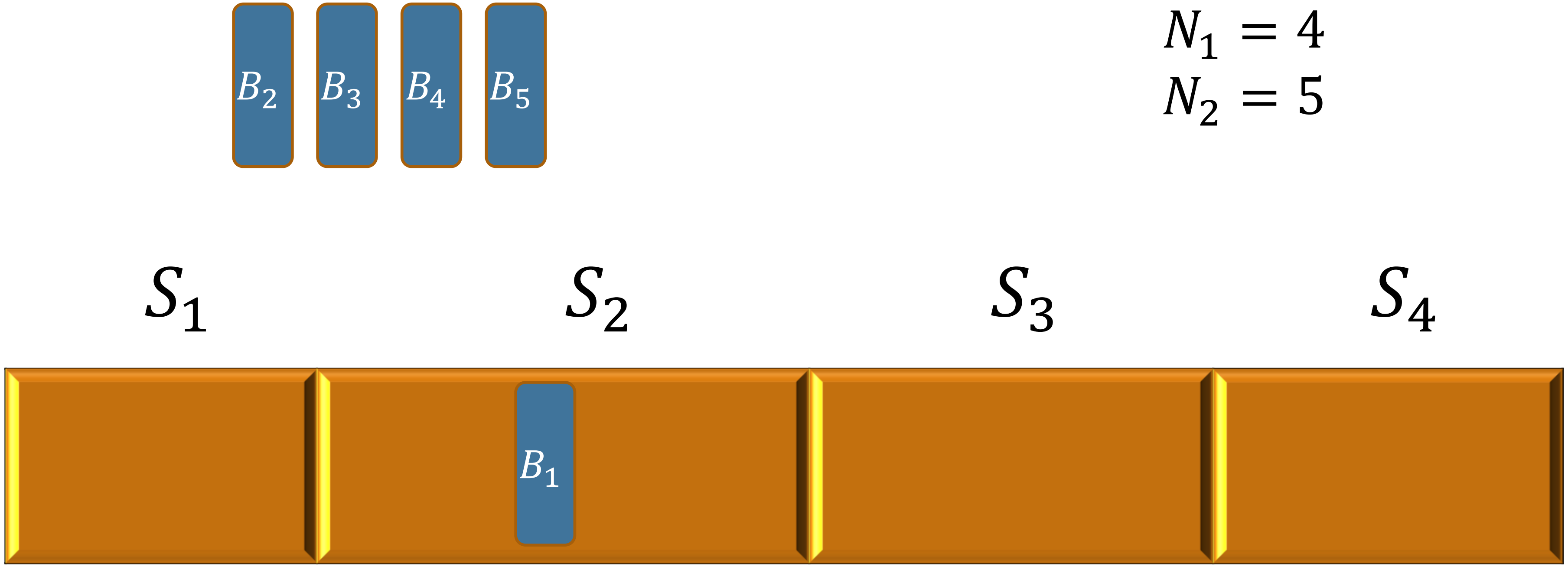
Books on Shelves

$m = 5$ books $B_1, B_2, B_3, \dots, B_m$
$n = 4$ shelves $S_1, S_2, S_3, \dots, S_n$

Can use the Product rule

Procedure: Start with empty shelves, place books one by one

Tasks: for $i = 1,2, \dots, m$ place B_i on a shelf ($B_1, B_2, \dots, B_{i-1}$ are already placed)



$N_1 = 4$
$N_2 = 5$

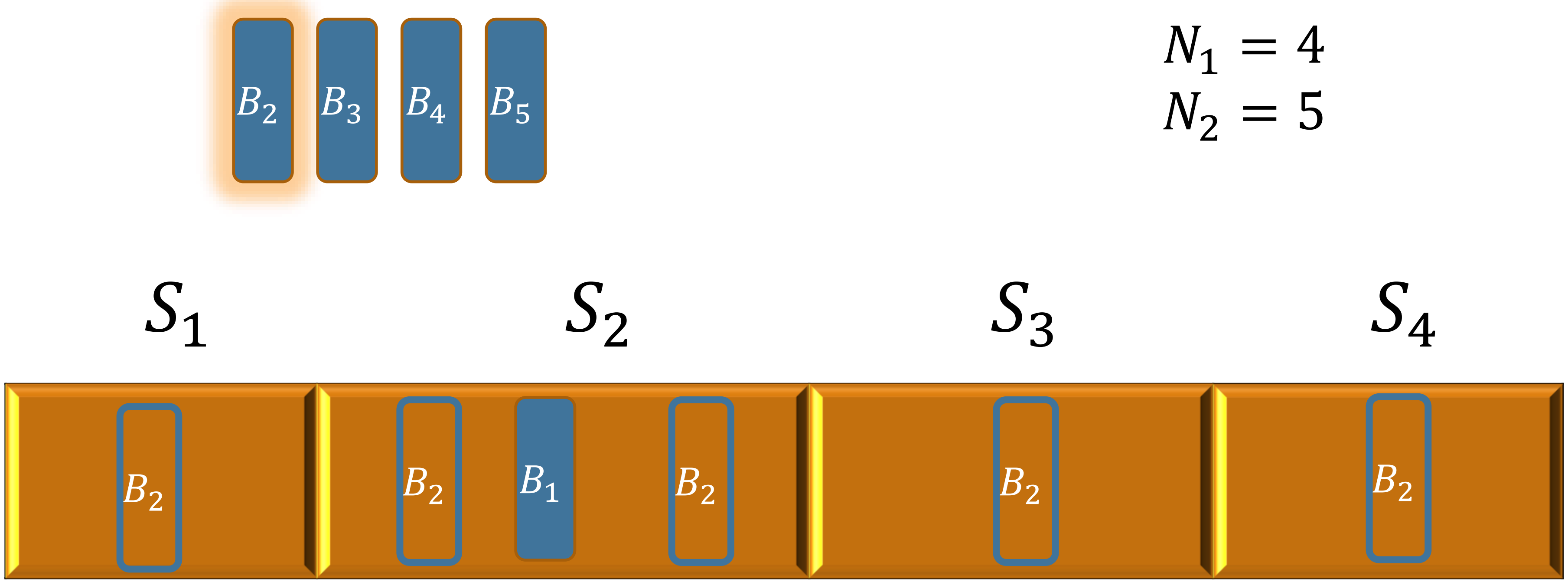
Books on Shelves

$m = 5$ books $B_1, B_2, B_3, \dots, B_m$
$n = 4$ shelves $S_1, S_2, S_3, \dots, S_n$

Can use the Product rule

Procedure: Start with empty shelves, place books one by one

Tasks: for $i = 1,2,\dots,m$ place B_i on a shelf ($B_1, B_2, \dots, B_{i-1}$ are already placed)



$N_1 = 4$
$N_2 = 5$

Books on Shelves

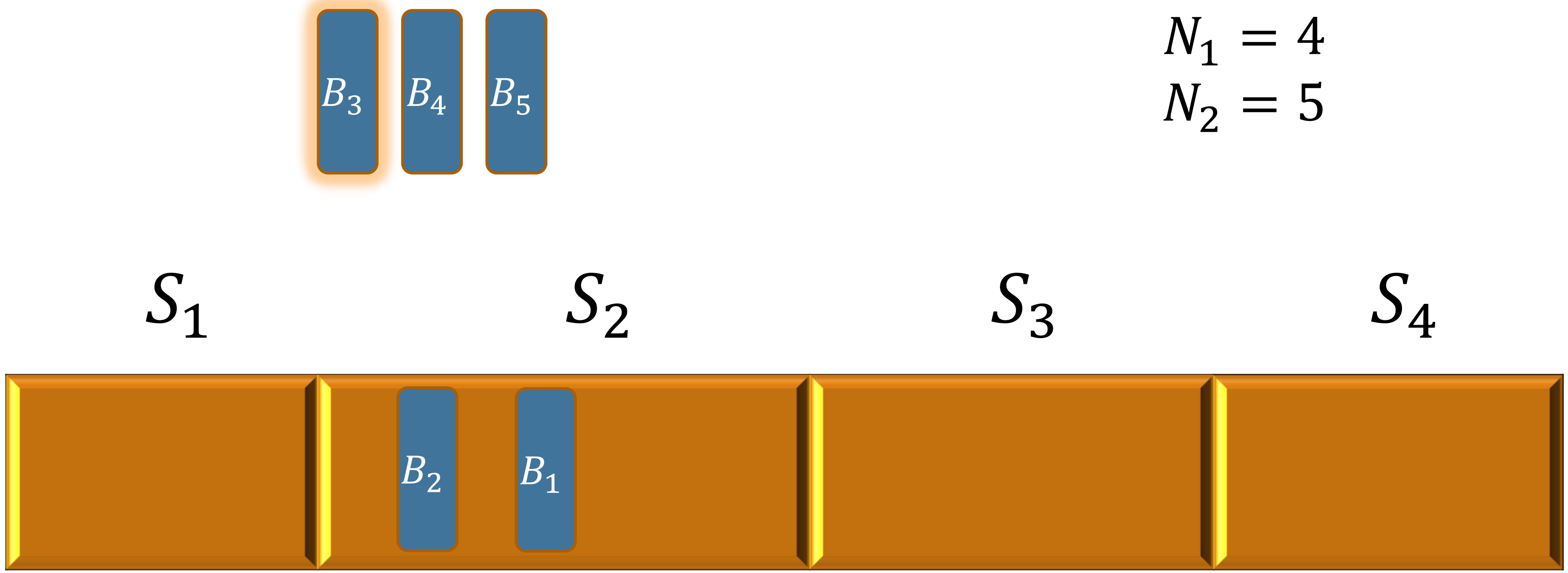
$m = 5$ books $B_1, B_2, B_3, \dots, B_m$
$n = 4$ shelves $S_1, S_2, S_3, \dots, S_n$

Can use the Product rule

Procedure: Start with empty shelves, place books one by one

Tasks: for $i = 1,2, \dots, m$ place B_i on a shelf ($B_1, B_2, \dots, B_{i-1}$ are already placed)

$N_1 = 4$
$N_2 = 5$



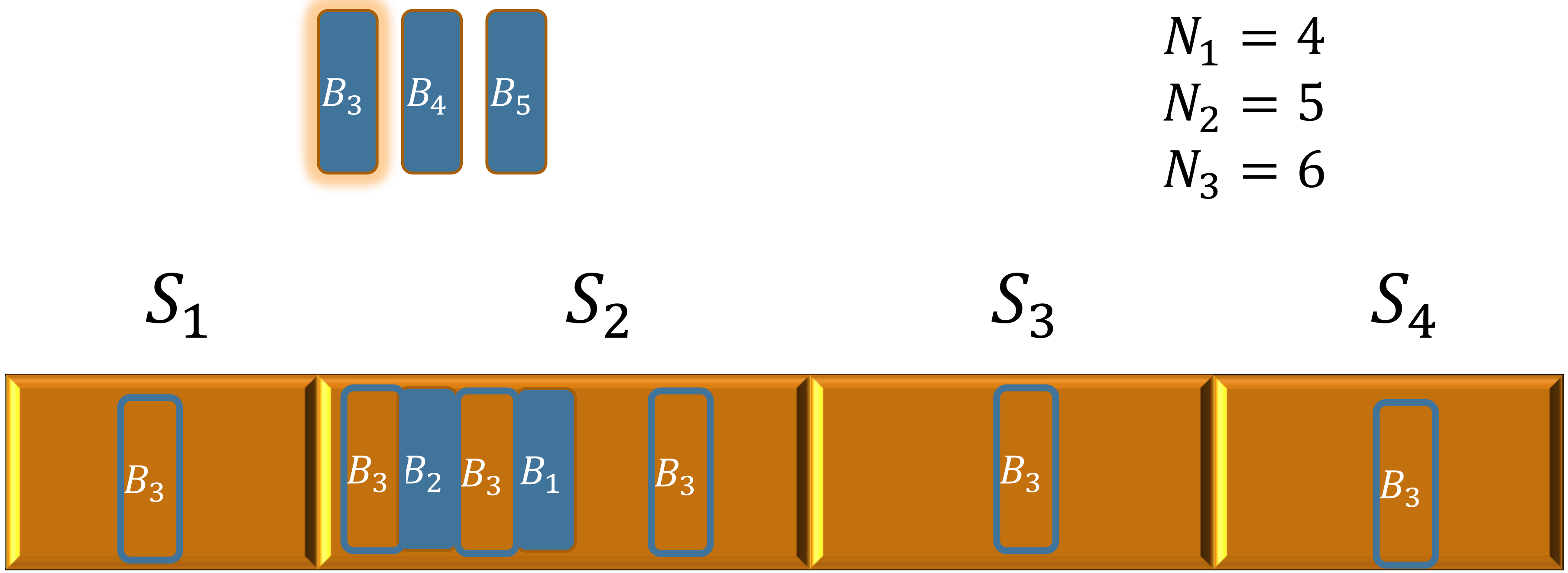
Books on Shelves

$m = 5$ books $B_1, B_2, B_3, \dots, B_m$
$n = 4$ shelves $S_1, S_2, S_3, \dots, S_n$

Can use the Product rule

Procedure: Start with empty shelves, place books one by one

Tasks: for $i = 1,2,\dots,m$ place B_i on a shelf ($B_1, B_2, \dots, B_{i-1}$ are already placed)



$N_1 = 4$
$N_2 = 5$
$N_3 = 6$

Books on Shelves

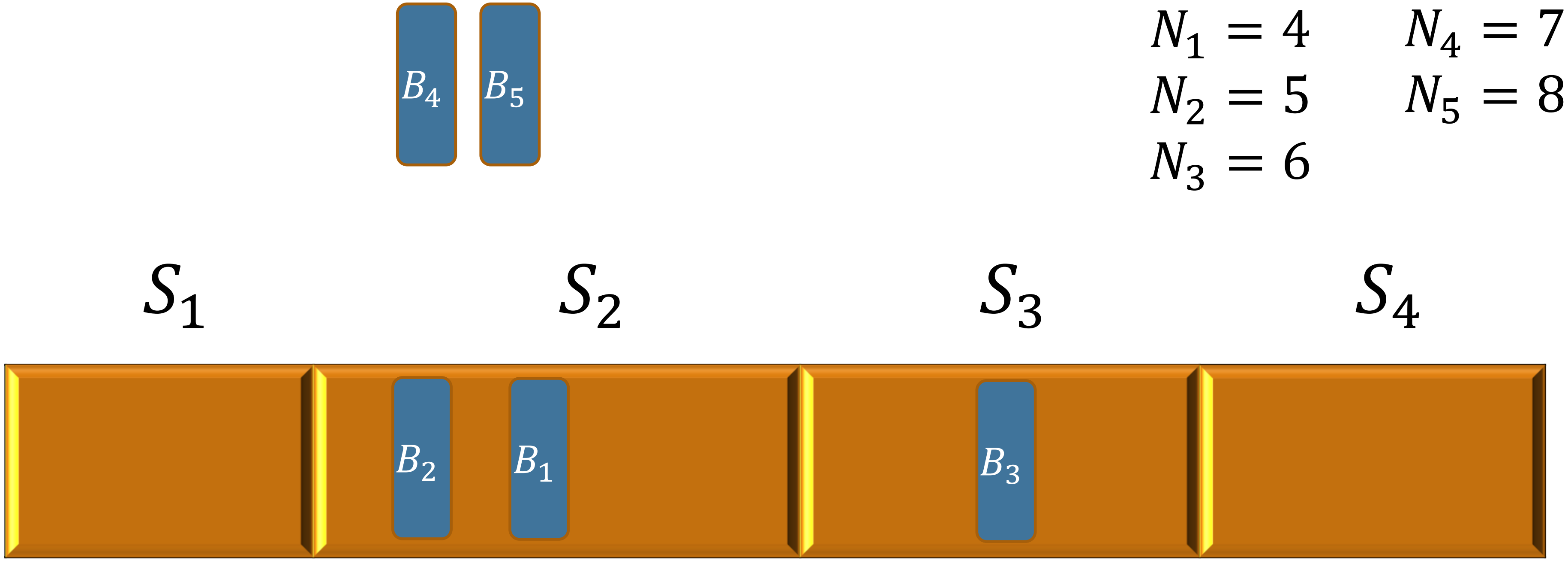
$m = 5$ books $B_1, B_2, B_3, \ldots, B_m$
$n = 4$ shelves $S_1, S_2, S_3, \ldots, S_n$

Can use the Product rule

Procedure: Start with empty shelves, place books one by one

Tasks: for $i = 1,2,\ldots,m$ place B_i on a shelf ($B_1, B_2, \ldots, B_{i-1}$ are already placed)

$N_1 = 4$ $\quad N_4 = 7$
$N_2 = 5$ $\quad N_5 = 8$
$N_3 = 6$



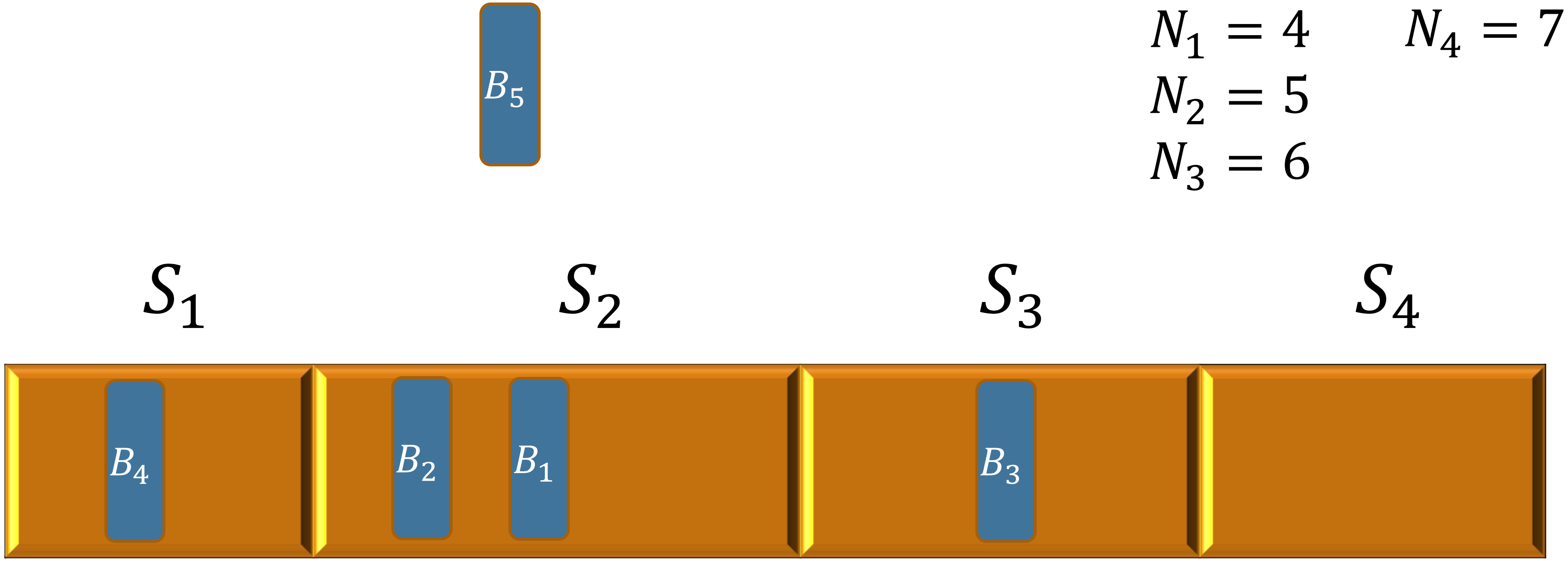
Books on Shelves

$m = 5$ books $B_1, B_2, B_3, \dots, B_m$
$n = 4$ shelves $S_1, S_2, S_3, \dots, S_n$

Can use the Product rule

Procedure: Start with empty shelves, place books one by one

Tasks: for $i = 1,2, \dots, m$ place B_i on a shelf ($B_1, B_2, \dots, B_{i-1}$ are already placed)



$N_1 = 4$
$N_2 = 5$
$N_3 = 6$
$N_4 = 7$

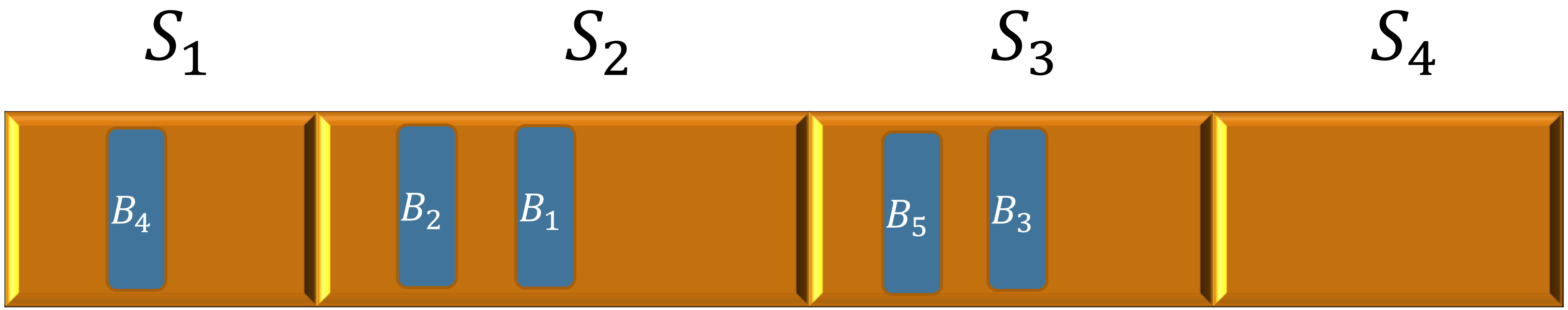
Books on Shelves

$m = 5$ books $B_1, B_2, B_3, \dots, B_m$
$n = 4$ shelves $S_1, S_2, S_3, \dots, S_n$

Using the product rule, the number of ways we can place 5 books on 4 shelves:

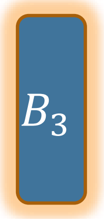
$$N_1 \cdot N_2 \cdot N_3 \cdot N_4 \cdot N_5$$

$N_1 = 4$
$N_2 = 5$
$N_3 = 6$
$N_4 = 7$
$N_5 = 8$



Books on Shelves

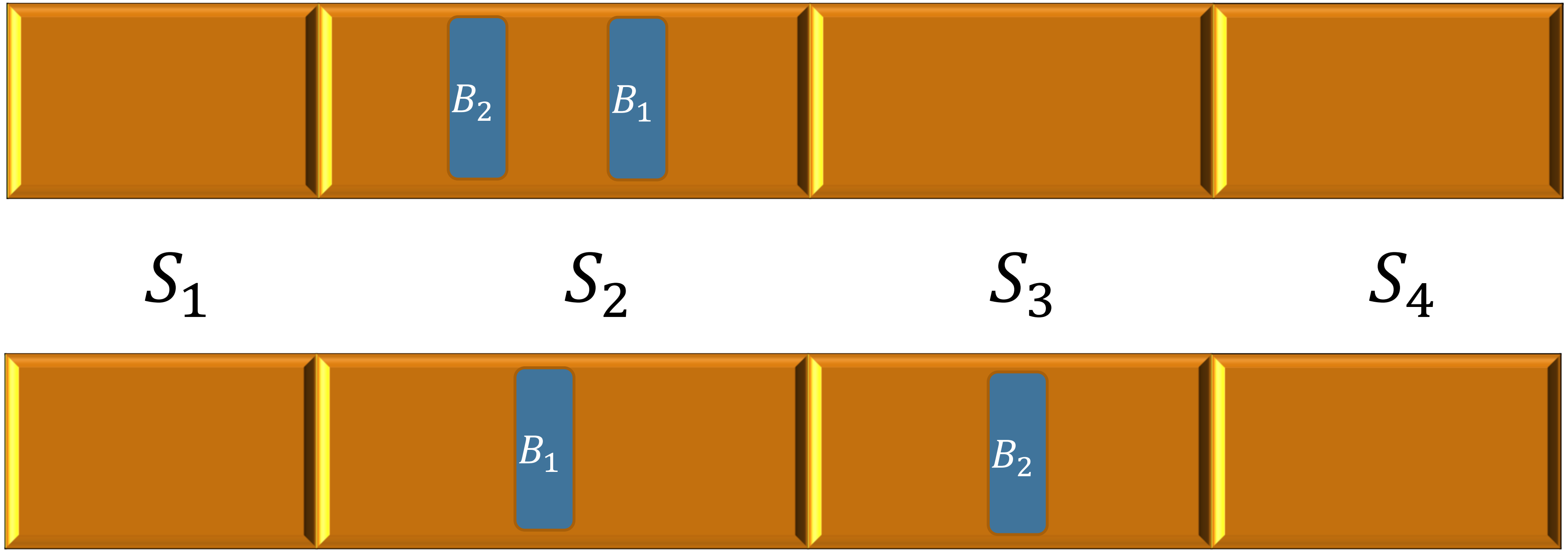
$m = 5$ books $B_1, B_2, B_3, \dots, B_m$
$n = 4$ shelves $S_1, S_2, S_3, \dots, S_n$



What if we placed them differently?

Where can B_3 go?

$N_1 = 4$
$N_2 = 5$



Books on Shelves

$m = 5$ books $B_1, B_2, B_3, \dots, B_m$

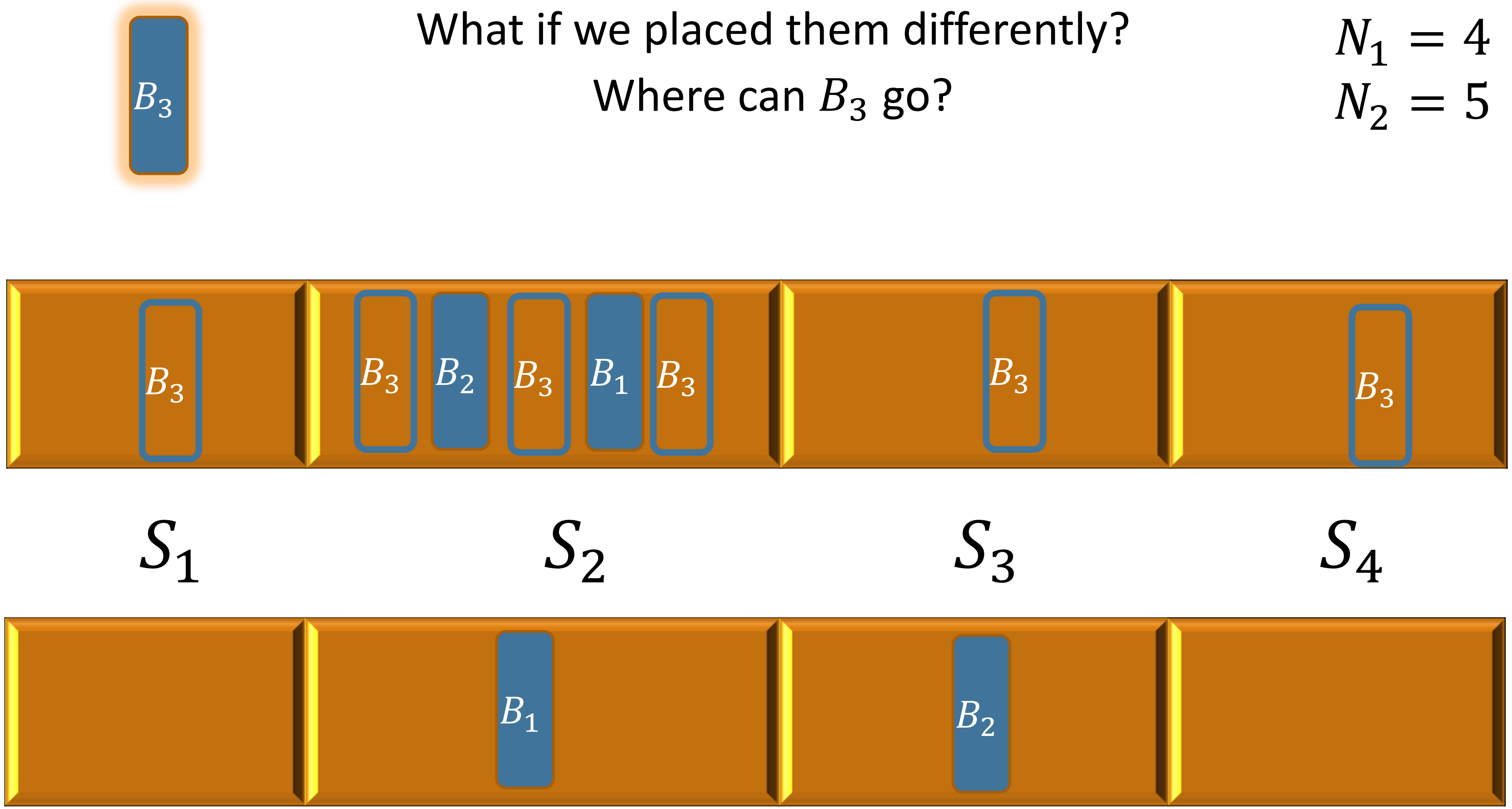
$n = 4$ shelves $S_1, S_2, S_3, \dots, S_n$

What if we placed them differently?

Where can B_3 go?

$N_1 = 4$

$N_2 = 5$



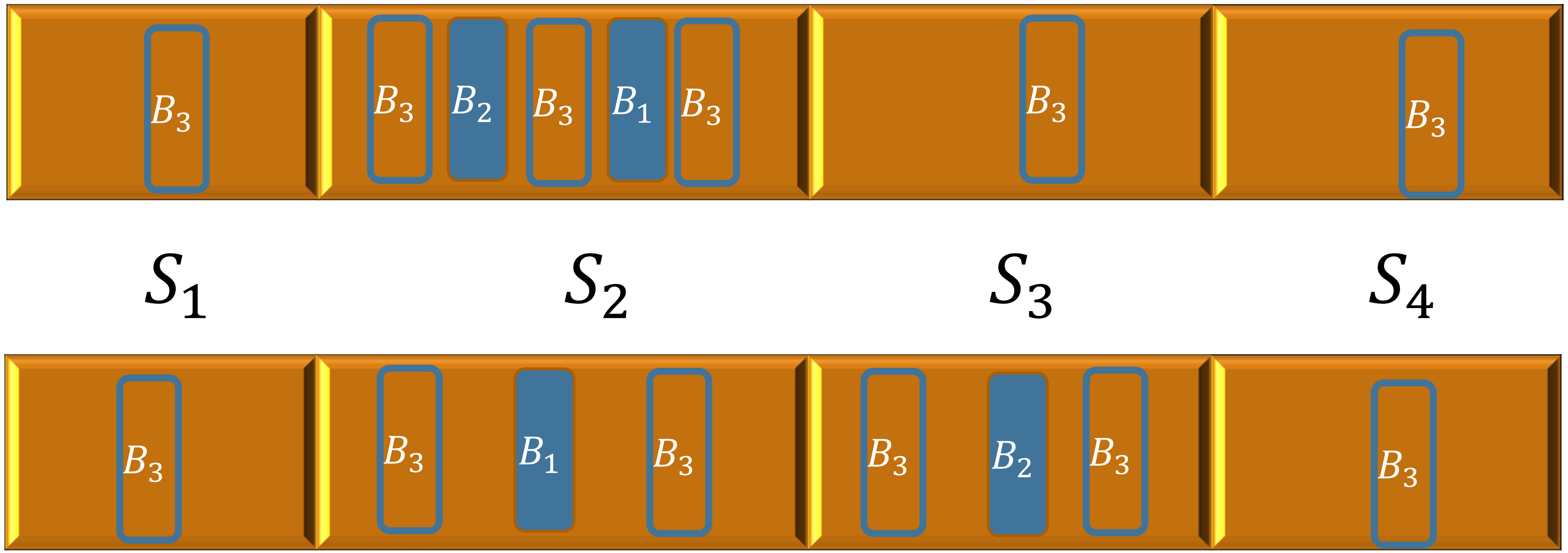
Books on Shelves

$m = 5$ books $B_1, B_2, B_3, \dots, B_m$
$n = 4$ shelves $S_1, S_2, S_3, \dots, S_n$

What if we placed them differently?

Where can B_3 go?

$N_1 = 4$
$N_2 = 5$

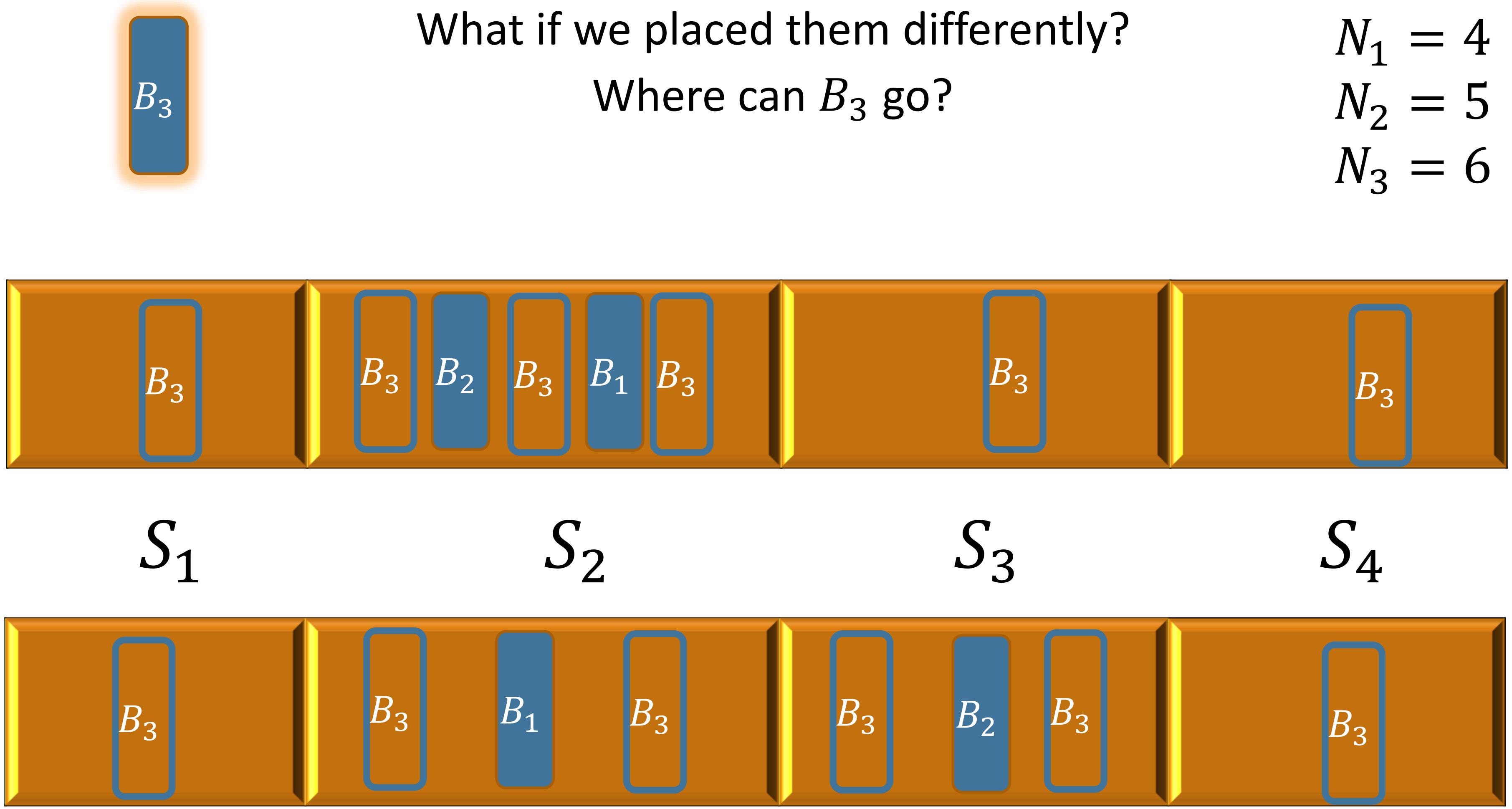


Books on Shelves

$m = 5$ books $B_1, B_2, B_3, \dots, B_m$
$n = 4$ shelves $S_1, S_2, S_3, \dots, S_n$

What if we placed them differently?
Where can B_3 go?

$N_1 = 4$
$N_2 = 5$
$N_3 = 6$



Books on Shelves

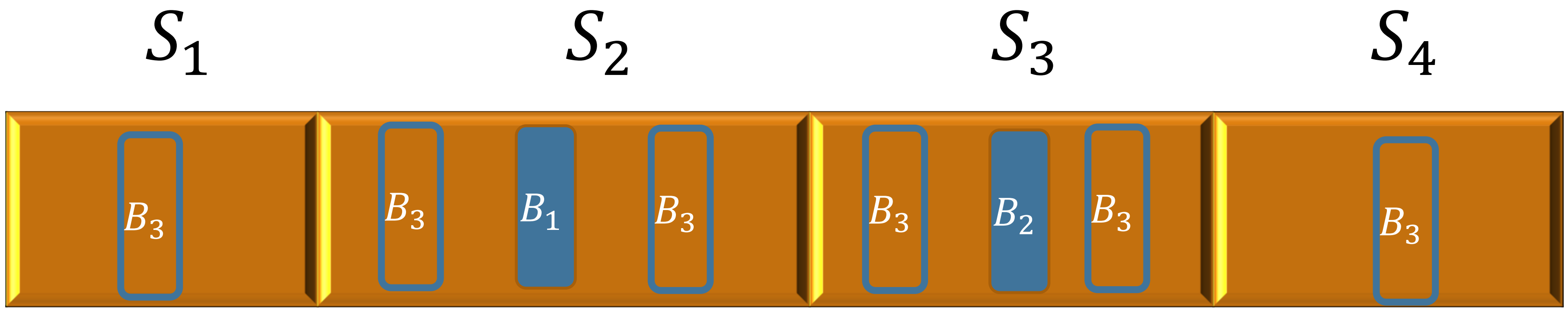
$m = 5$ books $B_1, B_2, B_3, \ldots, B_m$
$n = 4$ shelves $S_1, S_2, S_3, \ldots, S_n$

What if we placed them differently?

Where can B_3 go?

Every time we place a book, we split an existing "slot" into 2 "slots", regardless of where we place the book.

Each book becomes a new "divider". So where we place B_i during N_i does not affect the value of N_{i+1}



Books on Shelves

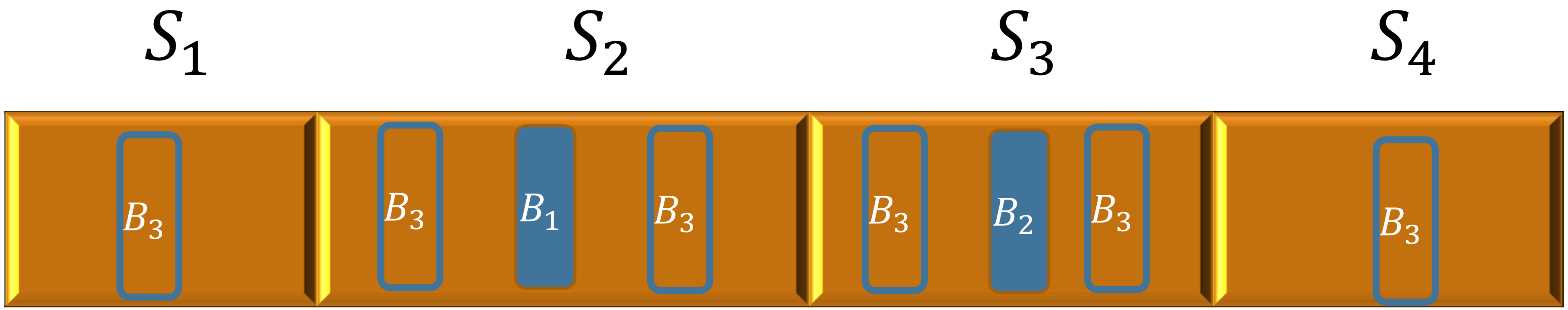
m books $B_1, B_2, B_3, \ldots, B_m$
n shelves $S_1, S_2, S_3, \ldots, S_n$

Task i: Place B_i

1. Place B_i on a shelf on the far left: n choices
2. Place B_i immediately to the right of a book $B_1, \ldots, B_{i-1}$ already on the shelf: $i-1$ choices.

Thus $N_i = n + i - 1$

Therefore the total number of ways to place books on shelves is $N_1 \cdot N_2 \cdot \ldots \cdot N_m$



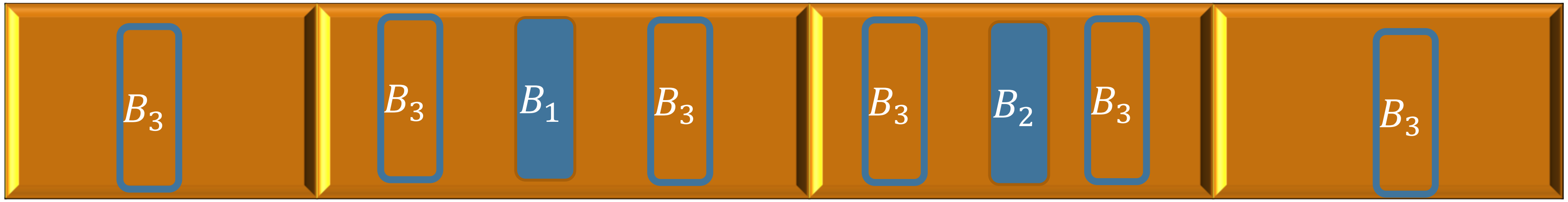
Books on Shelves

m books $B_1, B_2, B_3, \ldots, B_m$
n shelves $S_1, S_2, S_3, \ldots, S_n$

$$N_1 = n$$
$$N_2 = n + 1$$
$$N_3 = n + 2$$
$$\ldots$$
$$N_{m-1} = n + m - 2$$
$$N_m = n + m - 1$$

Therefore the total number of ways to place books on shelves is

$$N_1 \cdot N_2 \cdot \ldots \cdot N_m$$
$$= n \cdot (n+1) \cdot (n+2) \cdot \ldots \cdot (n+m-2) \cdot (n+m-1)$$



Books on Shelves

m books $B_1, B_2, B_3, \ldots, B_m$
n shelves $S_1, S_2, S_3, \ldots, S_n$

Again we want to express $n \cdot (n+1) \cdot (n+2) \cdot \ldots \cdot (n+m-2) \cdot (n+m-1)$ as a factorial

$$(n+m-1)! = 1 \cdot 2 \cdot \ldots \cdot (n-1) \cdot n \cdot (n+1) \cdot \ldots \cdot (n+m-2) \cdot (n+m-1)$$

to

$$n \cdot (n+1) \cdot \ldots \cdot (n+m-2) \cdot (n+m-1)$$

Divide by the first part:

$$\frac{1 \cdot 2 \cdot \ldots \cdot (n-1) \cdot n \cdot (n+1) \cdot \ldots \cdot (n+m-2) \cdot (n+m-1)}{1 \cdot 2 \cdot \ldots \cdot (n-1)}$$

$$= \frac{(n+m-1)!}{(n-1)!}$$

Books on Shelves

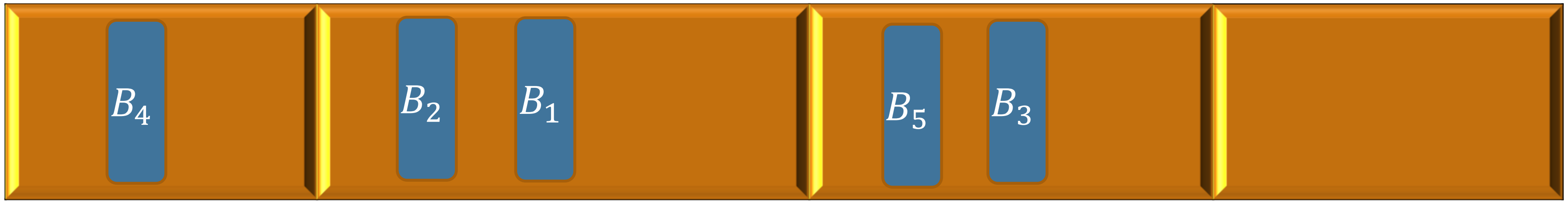
m books $B_1, B_2, B_3, \dots, B_m$
n shelves $S_1, S_2, S_3, \dots, S_n$

Therefore there are $\frac{(n+m-1)!}{(n-1)!}$ ways to place m books on n shelves

$$\frac{(n+m-1)!}{(n-1)!}$$

$$= \frac{(4+5-1)!}{(4-1)!}$$

$$= \frac{(8)!}{(3)!} = 6720$$



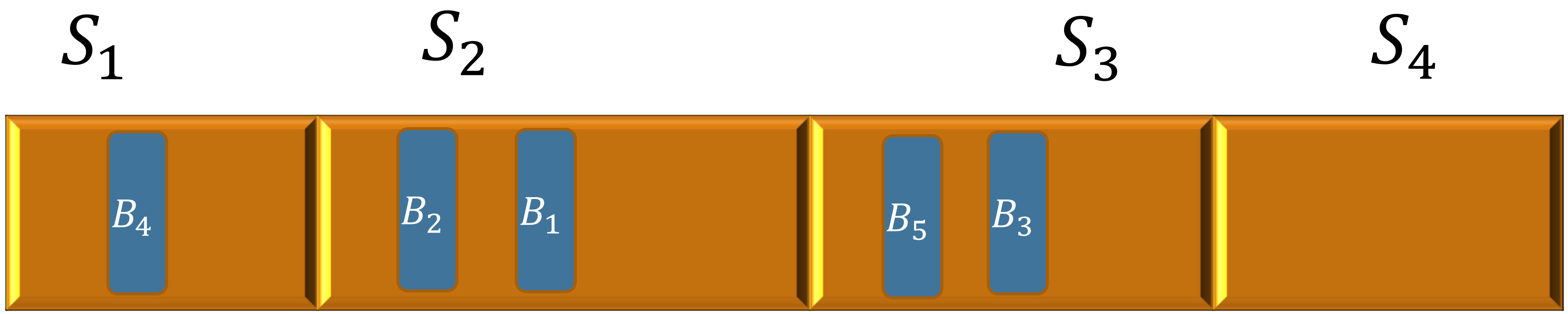
Review of Product Rule

Express as a procedure

Where a procedure is a **sequence of tasks**

The choice we make for Task *i* does not affect the number of ways to do Task *i+1*

Then we can apply the product rule



Product Rule

Set $S = \{a, b, c, d, e\}$

How many subsets?

$\{a\}, \{a, b, c\}, \{b, c\}, \{a, b, c, d, e\}, \emptyset$, etc.

$\emptyset \subseteq S$ if $\forall\, x \in \emptyset, x \in S$

$\neg\forall\, x \in \emptyset, x \in S$

$\exists x \in \emptyset, x \notin S$

$False$

Contradiction

Prove by contradiction, thus assume the negation

Product Rule

Set $S = \{a_1, a_2, a_3, \ldots, a_n\}$

How many subsets?

Can we use the Product rule?

Procedure: specify a subset

for $i = 1,2,3, \ldots, n$:

Task i: specify whether or not a_i is in the subset

Note a_i is either there or it is not – two choices

$N_i = 2$, and does not depend on first $i - 1$ tasks

number of subsets = 2^n (even if $n = 0$)

Product Rule

Set $S = \{a, b, c, d, e\}$

Very similar to bitstrings.

If a_i is in the subset, we write a 1
if a_i is not in the subset we write 0

Subsets of S $\leftrightarrow$ bitstrings of length $|S|$

$$\{a\} \leftrightarrow 10000$$
$$\{a, b, d, e\} \leftrightarrow 11011$$
$$\emptyset \leftrightarrow 00000$$
$$\{a, b, c, d, e\} \leftrightarrow 11111$$

Product Rule

Set $S = \{a, b, c, d, e\}$

Can also go from a bitstring to a subset

If we read a 1 then a_i is in the subset
If we read a 0 then a_i is not in the subset

bitstrings of length $\leftrightarrow |S|$

$10000 \leftrightarrow \{a\}$
$11011 \leftrightarrow \{a, b, d, e\}$
$00000 \leftrightarrow \emptyset$

Bijection Rule

Set $S = \{a, b, c, d, e\}$

Can also go from a bitstring to a subset

That means these are basically the same problem

(What it also means is we can apply the same solution)