Assignment 1 Solutions

COMP2804 Fall 2019

September 23, 2019

1 ID

Name: Lenny Learning Combinatorics
Student ID: 100000000

2 Non-Local Strings

1. We can create a block by choosing the first character, then the second character, and so on, until the 20th character. At each of the 20 steps, there are 26 options regardless of what was chosen in the previous steps so, by the Product Rule, the number of blocks is

$$26^{20} = 19\,928\,148\,895\,209\,409\,152\,340\,197\,376 \ .$$

2. We can also construct a squarefree block one character at a time. For the first character we have 26 options, but each of the 19 subsequent characters must be different from the character that came before it, so we have only 25 options. Therefore, by the Product Rule, then number of squarefree blocks is

$$26 \times 25^{19} = 9\,458\,744\,898\,438\,453\,674\,316\,406\,250 \ .$$

3. For a non-local block, we have 26 choices for the first character. Then we have 25 choices for the second character, since it must be different from the first. For each of the remaining 18 characters we have only $24 = 26 - 2$ choices since this character must be different from the two characters that precede it *and* the two characters that precede it are distinct. Therefore, by the Product Rule, then number of non-local blocks is

$$26 \times 25 \times 24^{18} = 4\,536\,445\,601\,729\,847\,895\,680\,614\,400 \ .$$

4. By now the pattern should become clear. For a k-non-local block $s_0, s_1, \ldots, s_{19}$ we have $26 - i$ choices for s_i for each $i \in \{0, \ldots, k-1\}$ and we have $26 - k$ choices for s_i for each $i \in \{k, \ldots, 19\}$. Therefore, by the Product Rule, then number of k-non-local blocks is

$$\left(\prod_{i=0}^{k-1}(26-i)\right) \times (26-k)^{20-k} = (26 \times 25 \times \cdots \times (26-k+1)) \times (26-k)^{20-k} \ .$$

3 Restricted Bitstrings

1. We've seen in class that the number of n-bit binary strings is 2^n, which is an easy application of the Product Rule.

2. To create such a string we have 3 choices, namely $\{01, 10, 11\}$, for $b_1 b_2$. Then we have 2 choices for each of $b_3, \ldots, b_n$. Therefore, by the Product Rule, the number of bitstrings that do not begin with 00 is

$$3 \times 2^{n-2} \ .$$

3. These are the four valid choices for $b_1b_2b_3$: $\{010, 101, 100, 110\}$. (The invalid choices are $\{000, 001, 011, 111\}$.) After choosing $b_1b_2b_3$ we have 2 choices for each of $b_4, \ldots, b_n$. Therefore, by the Product Rule, the number of strings we are interested in is

$$4 \times 2^{n-3} = 2^{n-1} \ .$$

4. These are the valid choices of $b_1b_2b_3$: $\{011, 010, 100, 110, 111\}$. (The invalid choices are $\{000, 001, 101\}$.) After choosing $b_1b_2b_3$ we have 2 choices for each of $b_4, \ldots, b_n$. Therefore, by the Product Rule, the number of strings we are interested in is

$$5 \times 2^{n-3} \ .$$

4 Bar Management

1. The bartender can first choose 5 Molson brands in one of $\binom{41}{5}$ ways and 5 Labatt brands in one of $\binom{17}{5}$ ways. Therefore, by the Product Rule the number of options is

$$\binom{41}{5} \times \binom{17}{5} = 4\,637\,274\,824$$

2. Since no two Molson products and no two Labatt products can be adjacent, the bottles must interleave like MLMLMLMLML or LMLMLMLMLM. Let's call this an *interleaving arrangement*.

   To define an interleaving arrangement we first choose whether the leftmost bottle will be a Molson or a Labatt product (2 options). Once we've done this, there are 5 choices for the first Molson product and 5 choices for the first Labatt product. Then 4 choices for the second Molson product and 4 for the second Labatt product. Continuing this way shows that the number of interleaving arrangements is

$$2 \times 5 \times 5 \times 4 \times 4 \times 3 \times 3 \times 2 \times 2 \times 1 \times 1 = 2 \times 5! \times 5! = 28\,800$$

3. Here it will be easier to count the number of interleaving arrangements that don't work and subtract this from the answer of the previous question.

   First choose whether the leftmost bottle will be a Molson or Labatt product (2 options). Next, choose one of the 9 locations $i \in \{1, \ldots, 9\}$ where the 50 or Export will go. The other (Export or 50, respectively) will go at location i+1. There are now 8 locations left to fill. We have 4 choices for the first Molson product in these unfilled locations, 3 choices for the second, and so on. The same is true for the four Labatt products in the unfilled locations. By the Product Rule, the number of interleaving arrangements in which the bottle of 50 *is* adjacent to the bottle of Export is

$$2 \times 9 \times 4! \times 4! = 10\,368.$$

   Since the total number of interleaving arrangements is 28 800, the number of interleaving arrangements in which the bottle of 50 *is not* adjacent to the bottle of Export is

$$28\,800 - 10\,368 = 18\,432 \ .$$

4. Using the Product Rule directly, we first decide whether the leftmost bottle is Labatt or Molson. Then we choose locations for the 50 and the Export among the first 5 bottles. Then we choose bottles that go into the remaining the 8 locations. The only tricky bit is counting the number of options in the second step. Without loss of generality, assume we are starting with a Labatt product, so we have a sequence like LMLMLMLMLM. Then there are three options for the bottle of 50:

   (a) 50 is the first of the 10 bottles (position 1). Then the Export cannot be placed at position 2 (it would be adjacent to the 50). It cannot be placed at positions 3 or 5 (since those are reserved for Labatt products). So in this case, the Export must be placed at position 4.

(b) 50 is the third of the 10 bottles (position 3). Then there is no place to place the Export since positions 1 and 5 are reserved for Labatt products and positions 2 and 4 are adjacent to the 50.

(c) 50 is the fifth of the 10 bottles (position 5). Symmetric to the first case, the only possible location for the Export is position 2.

Therefore there are exactly two choices in the second step: The 50 goes at positions 1 or 5 forcing the Export into positions 4 or 2, respectively. Therefore, the number of alternating arrangements in which the 50 and Export are not adjacent but both appear among the first 5 bottles is

$$2 \times 2 \times 4! \times 4! = 2304 \ .$$

5 ABC-Free Permutations

1. Choose an index $i \in \{1, \dots, 24\}$ where the letter a will appear (so that b and c appear at indices $i+1$ and $i+2$). This leaves 23 indices that we fill with a permutation of $\{d, e, f, \dots, z\}$. There are 24 ways to complete the first step and 23! ways to perform the second step so, by the Product Rule, the number of permutations in which a, b, c occur consecutively and in order is

$$24 \times 23! = 620\,448\,401\,733\,239\,439\,360\,000 \ .$$

2. Here are two possible solutions:

(a) First pick the three locations for a, b, and c among the 26 positions $\{1, \dots, 26\}$. Then fill the remaining 23 positions with a permutation of $\{d, e, f, \dots, z\}$. There are $\binom{26}{3}$ ways to perform the first step and 23! ways to perform the second step. Therefore, the number of permutations of Σ in which a appears before b and b appears before c is

$$\binom{26}{3} \times 23! = ((26 \times 25 \times 24)/6) \times 23! = 26!/6 = 67\,215\,243\,521\,100\,939\,264\,000\,000 \ .$$

(b) Imagine that we write out the 26! permutations of Σ and then we paint a black square over a, b, and c, so that a permutation like

$$s = olzvdhasjcrbxuengypkifwmqt$$

becomes

$$s' = olzvdh\blacksquare sj\blacksquare r\blacksquare xuengypkifwmqt \ .$$

Now notice that the string s' appears $3! = 6$ times in our list since it's generated by each string in

$$\begin{array}{c} olzvdhasjbrcxuengypkifwmqt \\ olzvdhasjcrbxuengypkifwmqt \\ olzvdhbsjarcxuengypkifwmqt \\ olzvdhbsjcraxuengypkifwmqt \\ olzvdhcsjarbxuengypkifwmqt \\ olzvdhcsjbraxuengypkifwmqt \end{array}$$

Exactly one of these 6 occurrences of s' is generated by a string in which a appears before b and b appears before c. This means that the 26! permutations of Σ can be grouped into 26!/6 groups and exactly one permutation in each group has a before b and b before c. Therefore, the number of permutations of Σ in which a appears before b and b appears before c is

$$26!/6 = 67\,215\,243\,521\,100\,939\,264\,000\,000 \ .$$

6 Imprecise Counting — Long Runs in Binary Strings

1. Since the $k+c$ values of $b_j, \ldots, b_{j+k+c-1}$ are already fixed to 0, there $n-k+c$ bits left to choose values for. There are
$$2^{n-k-c} = \frac{2^n}{2^{k+c}} = \frac{2^n}{2^k 2^c} = \frac{2^n}{n2^c}$$
ways to do this.

2. From the previous question, we know that $|X_j| = 2^n/(n2^c)$, so
$$\begin{aligned}
\sum_{j=1}^{n-k-c+1} |X_j| &= \sum_{j=1}^{n-k-c+1} \frac{2^n}{n2^c} \\
&= (n-k-c+1) \times \frac{2^n}{n2^c} \\
&\leq n \times \frac{2^n}{n2^c} && \text{(since } k \geq 1 \text{ and } c \geq 0 \text{, so } n-k-c+1 \leq n\text{)} \\
&= 2^n/2^c \ .
\end{aligned}$$

3. By definition, every string in X_j contains a substring of $k+c$ consecutive zeros starting at index j. On the other hand, if s is an n-bit binary string that contains a substring of $k+c$ consecutive zeros, then s belongs to at least one X_j for some $j \in \{1, \ldots, n-k-c+1\}$. Therefore, the number of n-bit binary strings that contain a string of $k+c$ consecutive zeros is
$$\left| \bigcup_{j=1}^{n-k-c+1} X_j \right| \leq \sum_{j=1}^{n-k-c+1} |X_j| \leq 2^n/2^c$$

   There's another way to interpret this: If we toss a coin $n = 2^k$ times, the probability that we will encounter a sequence $k+c$ consecutive tails is is at most $1/2^c$.