Tutorial #9

Problem 1 (Counting tree edges) Use structural induction to prove that $e(T)$, the number of edges of a full binary tree T, can be computed via formula

$$e(T) \quad = \quad 2(n(T) - \ell(T))$$

where $n(T)$ is the number of nodes in T and $\ell(T)$ is the number of *leaves*. Note that a *leaf* is a tree node that does not have descendants (children nodes). You can use the following recursive definition for the set of leaves:

Basis step: If a tree has a single node, then it is a *leaf* (as well as a *root*).
Recursive step: The set of leaves of the tree $T = T_1 \cdot T_2$ is the union of the set of leaves of T_1 and T_2.

You can also use the fact that the full binary tree $T = T_1 \cdot T_2$ adds two new edges when connecting T_1 and T_2 to the new common root. Provide detailed justification.

Solution 1 We use structural induction

Basis step: If the full binary tree T consists of a single root vertext r then we have $n(T) = 1$, $\ell(T)) = 1$ and $e(T) = 0$. These values satisfy the relation:

$$e(T) \quad = \quad 2(n(T) - \ell(T))$$

Recursive step: Let $T = T_1 \cdot T_2$ be a full binary tree built from two full binary trees T_1, T_2 and a root vertext r connected to the root vertices r_1 of T_1 and r_2 of T_2. Thus, the set edges(T) of edges of T is given by:

$$\mathrm{edges}(T) = \mathrm{edges}(T_1) \cup \mathrm{edges}(T_2) \cup \{\{r, r_1\}, \{r, r_2\}\}.$$

Hence we have:

$$e(T) = e(T_1) + e(T_2) + 2.$$

Let us assume that the formula to be proved holds for T_1 and T_2. Thus we have:

$$e(T_1) \;=\; 2(n(T_1) - \ell(T_1)) \text{ and } e(T_2) \;=\; 2(n(T_2) - \ell(T_2)).$$

Note that we have (from the lectures and Tutorial 8):

$$\ell(T) = \ell(T_1) + \ell(T_2) \text{ and } n(T) = n(T_1) + n(T_2) + 1.$$

Combining the above equations, we deduce:

$$\begin{aligned} e(T) &= e(T_1) + e(T_2) + 2 \\ &= 2(n(T_1) - \ell(T_1)) + 2(n(T_2) - \ell(T_2)) + 2 \\ &= 2(n(T_1) + n(T_2) + 1 - (\ell(T_1) + \ell(T_2))) \\ &= 2(n(T) - \ell(T)). \end{aligned}$$

Therefore, we have proved by induction that for all full binary tree T:

$$e(T) \;=\; 2(n(T) - \ell(T))$$

Problem 2 Consider all *genes* (strings with $\Sigma = \{A, T, C, G\}$) of length 10.

1. How many genes begin with AGT?
2. How many genes begin with AG and end with TT?
3. How many genes begin with AG or end with TT?
4. How many genes have exactly four A's?
5. How many genes have exactly four A's non-adjacent to each other?

Provide detailed justification for your answers.

Solution 2

1. Each of the 7 remaining characters need to be chosen from 4, leading to 4^7 genes.
2. Each of the 6 remaining characters need to be chosen from 4, leading to 4^6 genes.
3. We apply the subtraction rule: $4^8 + 4^8 - 4^6$.
4. We apply the product rule:
    - choose where to place the A's: $\binom{10}{4}$,
    - choose the 6 remaining characters from $\{T, C, G\}$: 3^6

   So the answer is: $\binom{10}{4} \times 3^6$

Problem 3 (Counting binary strings) Consider all bit strings of length 15.

1. How many begin with 00?
2. How many begin with 00 and end with 11?

3. How many begin with 00 or end with 10?
4. How many have exactly ten 1's?
5. How many have exactly ten 1's such as none of these 1's are adjacent to each other?

Provide detailed justifications for your answers.

Solution 3 For every bit string $b_1 b_2 \cdots b_{15}$ each of the bits $b_1, b_2, \ldots, b_{15}$ can take two values, namely 0 or 1. Applying the product rule, the sum rule and the subtraction rule,

1. the number of bit strings $b_1 b_2 \cdots b_{15}$ beginning with 00 is 2^{13},
2. the number of bit strings $b_1 b_2 \cdots b_{15}$ beginning with 00 and ending with 11 is 2^{11},
3. the number of bit strings $b_1 b_2 \cdots b_{15}$ beginning with 00 or ending with 10 is $2^{13} + 2^{13} - 2^{11}$,
4. the number of bit strings $b_1 b_2 \cdots b_{15}$ with exactly ten 1's is $\binom{15}{10}$, that is, the number of ways of choosing 10 bits among $b_1, b_2, \ldots, b_{15}$,
5. the number of bit strings $b_1 b_2 \cdots b_{15}$ having exactly ten 1's such as none of these 1's are adjacent to each other is zero. Indeed, in order to separate each of these ten 1's from the others, we would need (at least) nine 0's.

Problem 4 (Counting permutations) Solve the following counting problems:

1. How many permutations of the eight letters A, B, C, D, E, F, G, H have A in the second position?
2. How many permutations of the eight letters A, B, C, D, E, F, G, H have A in one of the first two positions?
3. How many permutations of the eight letters A, B, C, D, E, F, G, H have the two vowels after the six consonants?
4. How many permutations of the eight letters A, B, C, D, E, F, G, H neither begin nor end with D?
5. How many permutations of the eight letters A, B, C, D, E, F, G, H do not have the vowels next to each other?

Provide detailed justifications for your answer.

Solution 4

1. Choose a letter to be the first one and then choose a permutation of the remaining six: $7 \times 6! = 7!$.
2. Choose where to place A, then choose a permutation of the remaining seven: $2 \times 7!$.

3. Choose a permutation of the consonants, then a choose a permutation of the vowels: $6! \times 2!$.
4. Choose a place for D, then choose a permutation of the remaining seven: $6 \times 7!$.
5. $7 \times 2! \times 6!$ do have the vowels next to each other, so $8! - 7 \times 2! \times 6!$ do not have the vowels next to each other.

Problem 5 (Counting triominos) We saw in class that every $2^n \times 2^n$ board, with one square removed, could be covered with triominos. Determine a formula counting the number of triominos covering such a truncated $2^n \times 2^n$ board. Prove this formula by induction.

Solution 5

Basis step: if $n = 1$, then $2^n \times 2^n - 1 = 3$ and a single triomino suffices

Recursive step: Let $t(n)$ be the number of triomino needed to cover a truncated $2^n \times 2^n$ board. We want to express $t(n+1)$ as a function of $t(n)$. So, consider a truncated $2^{n+1} \times 2^{n+1}$ board. Removing one square from one the four quadrants and removing three squares forming a triomino from the other three yields:

$$t(n+1) = 4t(n) + 1.$$

This suggests:

$$t(n) = \frac{4^n - 1}{3},$$

which is easy to verify by induction.

Problem 6 (Pigeonhole principle and combinatorial proofs)

1. Let S be a subset of $\mathbb{N}$ (where $\mathbb{N}$ is the set of non-negative integers) such that S has at least 3 elements. Prove that there exist at least two elements s, y of S so that $x + y$ is even.

2. Let S be a subset of $\mathbb{N} \times \mathbb{N}$ such that S has at least 5 elements. Prove that there exist at least two elements (x_1, x_2) and (y_1, y_2) in S so that $x_1 + y_1$ and $x_2 + y_2$ are both even.

3. Prove that, in the previous question, one can not replace that 5 by 4 while preserving the same conclusion.

Solution 6

1. The sum of two integers of the same parity gives an even integer. We classify the elements of S into two classes even and odd integers. Since there are at least three elements in S one of the two classes contains at least two elements. There are therefore in S two integers of the same parity.

2. We reproduce the same reasoning but this time we make 4 classes.

   (a) PP couples whose two components are even

   (b) PI the couples whose first component is even and the second component is odd

   (c) IP the couples whose first component is odd and the second component is even

   (d) II couples whose two components are odd.

   As soon as we have 5 elements, we are sure that there is at least one class that contains two elements. Now the sum of two elements of a class gives a couple whose two components are even. This confirms that from 5 elements we are sure of the existence of the two desired couples.

3. considering the part S = (0, 0), (0, 1), (1, 0), (1, 1) we see that 4 couples do not allow to ensure existence.