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

\[
e(T) = 2(n(T) - \ell(T))
\]

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 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

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

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: $7 \times 6!$.
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 triominos 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 of 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.