CS34110 Discrete Mathematics and Graph Theory

UNIT – II, Module – 2

Lecture 13: Counting

[Basic counting principles; Product rule; Applicability of product rule in practice; Product rule's applicability to set theory]

Dr. Suddhasil De

Counting

- Counting:
  - Need: counting of objects to solve different types of problems (like, probability problems, algorithm complexity analysis, phone number or IP address allocation, password management, DNA sequencing in mathematical biology etc.).
  - Foundation of Combinatorics: basic counting principles.
  - Property:: Job: main goal in given counting problem.
  - Property:: Task: component(s) of job; to be considered as sub-goal.
  - Property:: Procedure: technique to perform task.
    - Property:: Multiple ways to conduct procedure.

Basic counting principles

- Four basic counting principles —
  - Product rule: also called 'rule of product' or 'multiplication principle'.
  - Sum rule: also called 'rule of sum' or 'addition principle'.
  - Subtraction rule: also called 'subtraction principle'.
  - Division rule.

Basic counting principles

- Four basic counting principles —
  - Property:: Product rule: applicable when job composed of multiple separate tasks (where, all tasks MUST be completed in sequence), with single procedure per task, and number of ways to perform individual tasks (i.e., to complete procedures) independent from all others.
  - Property:: Sum rule: applicable when single task of job (with multiple procedures to complete it) to be completed by only one of such procedures, and number of ways to perform that task (i.e., to complete procedure) to be pairwise-independent of each other.

Basic counting principles

- Four basic counting principles —
  - Property:: Subtraction rule: applicable when single task of job (with multiple procedures to complete it) to be completed by only one of such procedures, and number of ways to perform that task (i.e., to complete procedure) to be NOT pairwise-independent (i.e., multiple common ways possible to complete that procedure).
  - Property:: Division rule: applicable when single task of job (with one procedure to complete it) to be performed, and number of ways to perform that task (i.e., to complete procedure) include same number of equivalent ways for every distinct way to perform it.

Product rule

- Product rule [generalized case]: If any job to be carried out by performing m constituent tasks $T_1, T_2, \ldots, T_m$ in sequence, and if any i-th task T_i $(i = 1, 2, \ldots, m)$ to be done in n_i ways, regardless of how previous $(i-1)$ tasks already done, then $(n_1 \cdot n_2 \cdot \cdots \cdot n_m)$ total number of ways to carry out given job.
  - Product rule [two-task case]: If any job be broken down into sequence of two tasks, and if n_1 ways possible to do first task, and for each of these ways of doing first task, n_2 ways possible to do second task, then $n_1 \cdot n_2$ total number of ways to do given job.
  - Property:: Resembling conjunction of procedures of tasks.

Product rule

- Product rule examples:
  - Example-1:: case of labeling of chairs of auditorium, with uppercase English letter followed by positive integer not exceeding 100.
    Job = labeling chair; constituents: two tasks.
    Task 1: assigning one of 26 uppercase English letters to each chair.
    Task 2: assigning one of 100 possible integers to each chair.
    Task 2 performed regardless of how Task 1 performed.
    Then, according to product rule, number of different ways to label any chair $= 26 \cdot 100 = 2600$.
    So, maximum number of chairs labeled differently $= 2600$.

Product rule

- Product rule examples:
  - Example-2:: different bit strings of length seven.
    Each bit = either 0 or 1.
    Job = choosing bit; constituents: 7 tasks.
    Task T_i $(i = 1,2, \dots, 7)$: choosing bit for i-th position in bit string.
    Single procedure to perform task T_i.
    Task T_i performing regardless of how $T_1, T_2, \dots, T_{i-1}$ performed.
    Then, according to product rule, number of different ways to choose 7 bits $= 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^7 = 128$.
    So, number of different bit strings of length seven $= 128$.

Product rule

- Product rule examples:
  - Example-3:: case of vehicle license plate, with sequence of two uppercase English letters followed by four digits, with no restrictions.
    Constituents of job: 6 tasks.
    Task T_i $(i = 1,2)$: assigning one of 26 uppercase English letters.
    Task T_j $(j = 3,4,5,6)$: assigning one of 10 digits.
    Later tasks performing regardless of how earlier tasks performed.
    Then, according to product rule, number of different ways to label any license plate $= 26 \cdot 26 \cdot 10 \cdot 10 \cdot 10 \cdot 10 = 6760000$.
    So, number of different license plates made $= 6760000$.

Product rule

- Product rule examples:
  - Example-4:: case of function $f: X \to Y$ from domain X (of m elements) to codomain Y (of n elements), with no restrictions.
    Job: counting functions for each of m elements of X.
    Constituents: m tasks.
    Task T_i $(i = 1,2, \dots, m)$: choice of one of n elements in Y for i-th element of X, performing regardless of how $T_1, T_2, \dots, T_{i-1}$ performed.
    Then, according to product rule, number of different ways to choose mapping of functions (from X to Y) $= n \cdot n \cdot \dots \cdot n = n^m$.
    So n^m functions present from X (of m elements) to Y (of n elements).

Product rule

- Product rule examples:
  - Example-5:: case of one-to-one function $g: X' \to Y'$ from domain X' (of m' elements) to codomain Y' (of n' elements), where $m' \leq n'$.
    Job: counting functions for each of m' elements of X'.
    Constituents: m' tasks.
    Task T_j' $(j = 1,2, \dots, m')$: choice of one of $(n' - j + 1)$ elements in Y' for j-th element of X', performing regardless of how $T_1', T_2', \dots, T_{j-1}'$ performed, except that elements of Y' already chosen for previous tasks not to be used again, due to one-to-one nature of function.

(contd. to next slide)

Product rule

- Product rule examples:
  - Example-5 (contd.)::
    Then, according to product rule, number of different ways to choose mapping of functions (from X' to Y') $= n' \cdot (n' - 1) \cdot \dots \cdot (n' - m' + 1)$
    $= \frac{n'!}{(n'-m')!}$.
    So, $(n'!/(n' - m')!)$ functions present from X' (of m' elements) to Y' (of n' elements).

Product rule

- Product rule examples:
  - Example-6:: case of number of ordered pairs $(\mathbf{A}, \mathbf{B})$, such that $|\mathbf{A} \cap \mathbf{B}| = 1$, where $\mathbf{A}, \mathbf{B} \subseteq \boldsymbol{\mathcal{U}} = \{1,2,3,4,5\}$.
    Job: choosing elements from $\boldsymbol{\mathcal{U}}$, where $|\boldsymbol{\mathcal{U}}| = n = 5$.
    Constituents: 5 tasks.
    Task T_1: assigning one of $n = 5$ elements of $\boldsymbol{\mathcal{U}}$ to both $\mathbf{A}$ and $\mathbf{B}$, such that $|\mathbf{A} \cap \mathbf{B}| = 1$.
    Task T_i $(i = 2, \dots, 5)$: taking one element from rest of $(n - i + 1)$ elements of $\boldsymbol{\mathcal{U}}$ to add it in either $\mathbf{A}$ or $\mathbf{B}$ or none.
    Single procedure to perform every constituent task.

(contd. to next slide)

Product rule

- Product rule examples:
  - Example-6 (contd.)::
    Number of ways to perform T_1: 5.
    Number of ways to perform each of T_i $(i = 2, \dots, 5)$: deciding for each element to carry out one of 3 actions, i.e., 3 ways.
    Later tasks performed in manners independent of how earlier tasks performed.
    Then, according to product rule, number of different ways to form ordered pairs $(\mathbf{A}, \mathbf{B})$: $5 \cdot 3 \cdot 3 \cdot 3 \cdot 3 = 405$.
    So, number of different ordered pairs $(\mathbf{A}, \mathbf{B})$ formed $= 405$.

Product rule

- Product rule applied to set theory: for m finite sets $\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_m$, choosing an element in Cartesian product $(\mathbf{A}_1 \times \mathbf{A}_2 \times \cdots \times \mathbf{A}_m)$ done by choosing an element in $\mathbf{A}_1$ (done in $|\mathbf{A}_1|$ ways), an element in $\mathbf{A}_2$ (done in $|\mathbf{A}_2|$ ways),…, and an element in $\mathbf{A}_m$ (done in $|\mathbf{A}_m|$ ways), resulting in (as per product rule):
  $$|\mathbf{A}_1 \times \mathbf{A}_2 \times \cdots \times \mathbf{A}_m| = |\mathbf{A}_1| \cdot |\mathbf{A}_2| \cdot \ldots \cdot |\mathbf{A}_m| = \prod_{i=1}^{m} |\mathbf{A}_i|,$$
  i.e., number of elements in Cartesian product of $\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_m =$ product of number of elements in each of m sets.

Summary

- Focus: Basic counting principles.
- Basic counting principles.
- Product rule and sum rule as two basic counting principles.
- Product rule definition for generalized case.
- Product rule definition in two-task counting.
- Practical examples to demonstrate applicability of product rule.
- Applicability of product rule to set-theoretic problems.

References

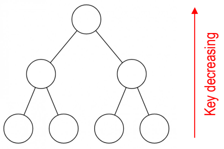
1. [Ros21] Kenneth H. Rosen, Kamala Krithivasan, *Discrete Mathematics and its Applications*, Eighth edition, McGraw-Hill Education, 2021.
2. [Ross12] Kenneth A. Ross, Charles R. B. Wright, *Discrete Mathematics*, Fifth edition, Pearson Education, 2012.
3. [Mot15] Joe L. Mott, Abraham Kandel, Theodore P. Baker, *Discrete Mathematics for Computer Scientists and Mathematicians*, Second edition, Pearson Education, 2015.
4. [Lip17] Seymour Lipschutz, Marc L. Lipson, Varsha H. Patil, *Discrete Mathematics (Schaum's Outlines)*, Revised Third edition, McGraw-Hill Education, 2017.
5. https://brilliant.org/wiki/rule-of-sum-and-rule-of-product-problem-solving/.

Further Reading

- Basic counting principles:: [Ros21]:405.
- Product rule definition for generalized case:: [Ros21]:406.
- Product rule definition in two-task counting:: [Ros21]:406.
- Practical examples to demonstrate applicability of product rule:: [Ros21]:406-409,411-412.
- Applicability of product rule to set-theoretic problems:: [Ros21]:408.

Lecture Exercises: Problem 1 [Ref: Gate 2018, Q.46, p.16 (Set-3)]

The number of possible min-heaps containing each value from $\{1,2,3,4,5,6,7\}$ exactly once is _______.



Min-heap of 7 keys