



Matemáticas Avanzadas para Computación

Tema 1. Introducción

Maestría en Sistemas Computacionales

Dra. Mildreth Alcaraz, mildreth@iteso.mx

Tel. 3669-3434 xt 3975, Oficina T - 316





Objetivos de la clase

- Conocer los propósitos del curso
- Consejos para el éxito del curso
- Identificar cuáles son las Matemáticas para el área de la Computación
- Conocer algunas aplicaciones relacionadas
- Estudiar los conceptos básicos de:
 - Principios del Conteo
 - Permutaciones



1.1. Introducción



Principales propósitos de MAC

- Emplear métodos de conteo para el análisis combinatorio e identificar la importancia del estudio de estos en el contexto de la implementación de sistemas computacionales.
- Identificar a los grafos como una estructura discreta, conocer sus aplicaciones, y comprender su relevancia en el contexto del desarrollo e implementación de sistemas computacionales.
- Identificar a los árboles como una estructura discreta, conocer sus aplicaciones, y comprender su relevancia en el contexto del desarrollo e implementación de sistemas computacionales.
- Identificar a los autómatas como una estructura discreta, conocer sus aplicaciones, y comprender su relevancia en el contexto del desarrollo e implementación de sistemas computacionales.
- Conocer la existencia de otras estructuras discretas basadas en grafos, y sus aplicaciones en el contexto de los sistemas computacionales.



Comentarios Adicionales

- Para aprender las matemáticas, hay que practicarlas... es como los deportes... No sólo saber las reglas y tener habilidad:
 - Paciencia...
 - Constancia...
 - Perseverancia...
 - Práctica
 - Disposición
 - Determinación a lograr el objetivo



Consejos para el Éxito del Curso

- Resolver dudas del tema.
- Repasar las diapositivas de la clase y sobre todo resolver **POR USTEDES MISMOS** los ejercicios de la clase, además de las tareas → suficiente para los exámenes.
- Complementar sus notas con los comentarios en clase.
- Las matemáticas son divertidas, sobre todo si trabajas para comprender lo poderosa que son...

- Leer la Guía de Aprendizaje.

Cuáles son las Matemáticas para Computación?

MAESTRÍA EN
SISTEMAS
COMPUTACIONALES



ITESO

Universidad Jesuita
de Guadalajara



Cuáles son las Matemáticas para Computación?

- Matemáticas discretas
 - Lógica, teoría de conjuntos, conteo, funciones, relaciones, árboles, grafos, autómatas, redes de Petri,...
- Álgebra abstracta
 - Grupos, subgrupos, anillos, cuerpos o espacios vectoriales,...
- Combinatoria y Probabilidad Discreta
 - Permutaciones, Combinaciones, con y sin repeticiones,...



1.2. Aplicaciones relacionadas a la Computación



Motivación

- Simplemente, **no existe** sistema computacional sin el uso de las matemáticas...



Aplicaciones

- Inteligencia Artificial (AI):
 - FOL (first order logic):
 - Propositiones con cuantificadores, funciones, operadores, etc.
 - Representación de conocimiento que puede ser evaluada.
 - Se dice que es el más general y potente método para describir y analizar información.
 - Si Muy detallado \rightarrow se reduce a problema de computabilidad (computable, se refiere a que una computadora puede resolverlo)
 - Cats are mammals [cat¹, mammal¹]
 - $\forall x, \text{cat}(x) \rightarrow \text{mammal}(x)$

[Chowdhary, 2011]



Aplicaciones

- Sistemas Expertos (Reglas tipo IF THEN), $SE \subset FOL$
 - Expert System = Knowledge Base + Inference Engine
- Un SE simple para reconocimiento de vehículos:
 - R1: If ?x has wings
 - then ?x is a plane
 - R2: If ?x flies
 - then ?x is a plane
 - R3: If ?x runs on tracks
 - then ?x is a train-or-tram
 - R4: If ?x is a plane
 - ?x can take off vertically
 - ?x has rotors
 - then ?x is a helicopter
 - R5: If ?x is a train-or-tram
 - ?x stays underground
 - then ?x is a subway car
 - R6: If ?x is a helicopter
 - ?x made in South Africa
 - then ?x is a Rooivalk

* Enseguida, se requiere de la generación de conocimiento (deducción y razonamiento)



Aplicaciones

- Ingeniería de Software:
 - Knowledge-Based Software Assistant:
 - Autómatas y Lenguajes Formales
- Validación de “correctitud” de especificaciones de software.
- Transformación de especificaciones en código eficiente en diversas plataformas, y probar la equivalencia entre la implementación y especificación: weapons systems, security systems, real time financial systems, donde las fallas implican costos humanos y/o financieros altos.
- para chips de CPUs u otros componentes digitales, pues no aceptan modificaciones posteriores, como en VLSI ([Very Large Scale Integrated](#)) Design.



Aplicaciones

- Clasificadores Automáticos

- las definiciones se mapean directamente a teoría de conjuntos y cálculo de predicados
- Se analiza la información, y a partir de esto, se genera nueva.

```
@prefix : <http://infomesh.net/2001/proofexample/#> .
@prefix p: <http://www.w3.org/2001/07/imaginary-proof-ontology#> .
@prefix log: <http://www.w3.org/2000/10/swap/log#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

p:ProvenTruth rdfs:subClassOf log:Truth .

# Proof

{ { <a.n3> p:checksum <md5:blargh>;
    log:resolvesTo [ log:includes { :Joe :loves :MJS } ] }
log:implies
{ :Step1 a p:Success } } a log:Truth .

{ { <b.n3> p:checksum <md5:test>;
    log:resolvesTo [ log:includes { :MJS = :Mary } ] }
log:implies
{ :Step2 a p:Success } } a log:Truth .

{ { :Step1 a p:Success . :Step2 a p:Success }
log:implies
{ { :Joe :loves :Mary } a p:ProvenTruth } } a log:Truth .
```

[Palmer,2001]



Aplicaciones

- Bases de Datos:
 - Utilizan la mayoría de la matemática computacional denominada discreta, principalmente, conjuntos y relaciones.
 - La verificación de las consultas, se realiza a través de autómatas y lenguajes formales.
 - *Recomendación de lectura:* DISCRETE MATHEMATICS VIA RELATIONAL DATABASES, Raymond Turner, February 26, 2011.
- Una **Tabla** es un conjunto finito de tuplas.

Name	Age	Salary
Jim	34	1200
Peter	23	14000

[Turner,2011]



Aplicaciones

Name	Age	Salary
Jim	34	1200
Peter	23	14000

- Una **relación** define a una tabla, sus atributos y tipos de datos.

- Set([name:String, age:N, salary:N]) =
{ [name:Jim, age:34, salary:1200],
[name:Peter, age:23, salary:14000]}.

A set of relations R:

<Jim, 34, 1200>,
<Peter, 23, 14000>

- $R(\text{Name, Age, Salary}) \subseteq \text{Name} \times \text{Age} \times \text{Salary}$

Set Name:

Jim
Peter

Set Age:

34
23

Set Salary:

1200
14000



Aplicaciones

Name	Age	Salary
Jim	34	1200
Peter	23	14000

- Las **operaciones** toman relaciones como argumentos, y regresan una relación como resultado.

A set of relations R:

<Jim, 34, 1200>,
<Peter, 23, 14000>

Name	Title	Year
C1	AA	3
C2	BB	2
C3	CC	1
C8	GG	2
C9	HH	3

R1

U

Name	Title	Year
C5	NNN	4
C6	NM	2
C7	PP	1
C2	BB	2
C8	GG	2

R2

=



Aplicaciones

Name	Age	Salary
Jim	34	1200
Peter	23	14000

- Las **operaciones** toman relaciones como argumentos, y regresan una relación como resultado.

A set of relations R:

<Jim, 34, 1200>,
<Peter, 23, 14000>

Name	Title	Year
C1	AA	3
C2	BB	2
C3	CC	1
C8	GG	2
C9	HH	3

R1

U

Name	Title	Year
C5	NNN	4
C6	NM	2
C7	PP	1
C2	BB	2
C8	GG	2

R2

=

Name	Title	Year
C1	AA	3
C2	BB	2
C3	CC	1
C8	GG	2
C9	HH	3
C5	NNN	4
C6	NM	2
C7	PP	1



Aplicaciones

- Productos Cartesianos

<i>Name</i>	<i>Title</i>	<i>Year</i>
<i>C1</i>	<i>AA</i>	<i>3</i>
<i>C2</i>	<i>BB</i>	<i>2</i>
<i>C3</i>	<i>CC</i>	<i>1</i>
<i>C8</i>	<i>GG</i>	<i>2</i>
<i>C9</i>	<i>HH</i>	<i>3</i>

R1



<i>Name</i>	<i>Title</i>	<i>Year</i>
<i>C5</i>	<i>NNN</i>	<i>4</i>
<i>C6</i>	<i>NM</i>	<i>2</i>

=

R2



Aplicaciones

- Productos Cartesianos

<i>Name</i>	<i>Title</i>	<i>Year</i>
<i>C1</i>	<i>AA</i>	<i>3</i>
<i>C2</i>	<i>BB</i>	<i>2</i>
<i>C3</i>	<i>CC</i>	<i>1</i>
<i>C8</i>	<i>GG</i>	<i>2</i>
<i>C9</i>	<i>HH</i>	<i>3</i>

R1



<i>Name</i>	<i>Title</i>	<i>Year</i>
<i>C5</i>	<i>NNN</i>	<i>4</i>
<i>C6</i>	<i>NM</i>	<i>2</i>

=

R2

$$Set([l_1 : T_1, \dots, l_n : T_n]) \otimes Set([k_1 : S_1, \dots, k_m : S_m])$$

$$\Rightarrow$$

$$Set([l_1 : T_1, \dots, l_n : T_n, k_1 : S_1, \dots, k_m : S_m])$$



Aplicaciones

- Intersection
- Natural Join
- Outer Join
- Multisets
- Lógica proposicional y cálculo de predicados

\Rightarrow

- Queries...

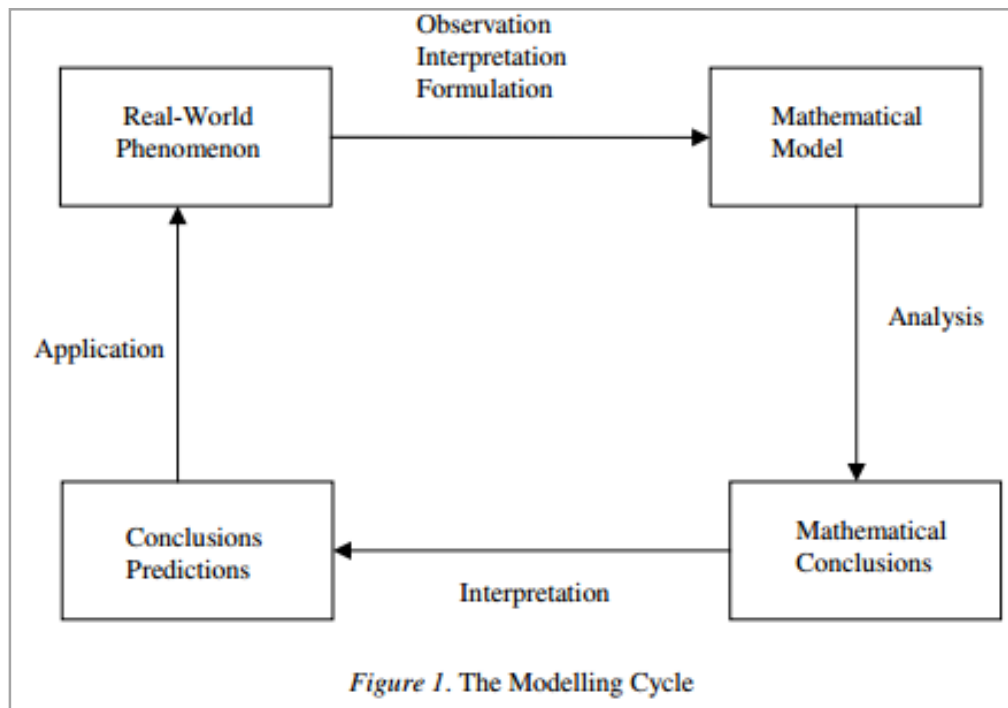
$$\{z \in Teacher \cdot \exists x \in Student \cdot z \circ CourseNo = x \circ CourseNo\}$$

- Todas las tuplas donde los profesores cuentan con un curso donde este curso tiene al menos un estudiante.

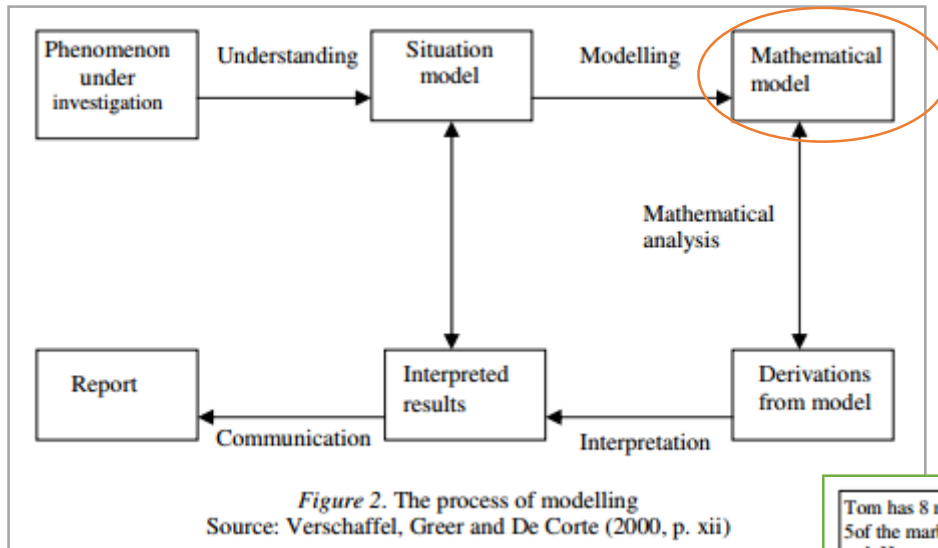


Aplicaciones

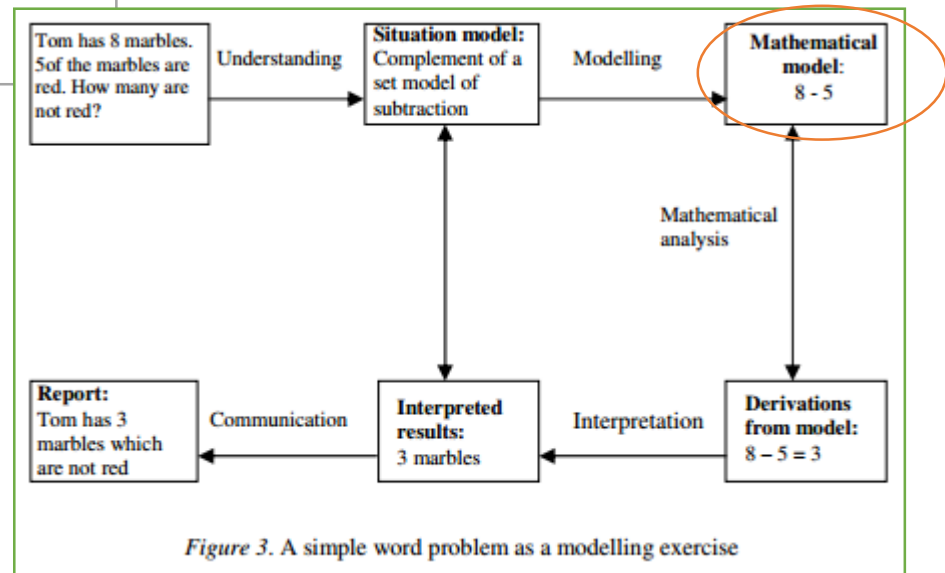
- Resolución de Problemas



Aplicaciones



- Resolución de Problemas



[Dindyal, 2010]



Referencias

1. Fundamentals of Discrete Mathematical Structures, K. R. Chowdhary, 2da. Edición, PHI Learning Pvt. Ltd., 2011.
2. Expert Systems Notes, Wayne Goddard, <http://people.cs.clemson.edu/~goddard/texts/cpsc810/chapA7.pdf>, Otoño 2014.
3. Yannibelli, Virginia, and Analía Amandi. "A knowledge-based evolutionary assistant to software development project scheduling." *Expert Systems with Applications* 38.7 (2011): 8403-8413.
4. Sean B. Palmer, The Semantic Web: An Introduction, <http://infomesh.net/2001/swintro/>
5. Discrete Mathematics Via Relational Databases, Raymond Turner, Class Notes, February 2011, http://www.academia.edu/1739363/DISCRETE_MATHEMATICS_VIA_RELATIONAL_DATABASES
6. Mathematical Applications and Modelling : Yearbook 2010, Dindyal, Jaguthsing, Kaur, Berinderjeet, Association of Mathematics Educators. Singapore: World Scientific Publishing; 2010.



1.3. Conceptos básicos: Combinatoria y probabilidad discreta



Qué es Combinatoria y Probabilidad Discreta?

- Combinatoria → arte de contar
- Responde a preguntas del tipo:
 - De cuántas maneras “esto” puede realizarse?
 - No cuáles, sólo cuántas



Qué es Combinatoria y Probabilidad Discreta?

- Combinatoria → arte de contar
- Responde a preguntas del tipo:
 - De cuántas maneras “esto” puede realizarse?
 - No cuáles, sólo cuántas
- Ejemplo:
 - El tipo de placa en Jalisco para automóviles consiste en 3 letras y 4 números. Supongamos que no hay restricciones sobre el uso de los elementos:
 - Cuántas placas es posible tener con esta combinación de letras y números?
 - Una forma de encontrar la respuesta en definitiva NO ES enumerando todas las opciones.



Motivación

- Se utiliza casi en todos los casos de análisis de algoritmos → relacionado con la eficiencia en tiempo:
 - Para grafos, recursividad,...
 - En cómputo paralelo y distribuido
- La encontramos en las **bases de datos** → útil para analizar la eficiencia en espacio de las consultas a bd.
- Big Data
- Muchas otras aplicaciones...



Principios Fundamentales del Conteo

- Estos principios encuentran su base en:

<u>Set operations</u>	<u>Logic operations</u>
$A \cap B$	$A \text{ AND } B$
$A \cup B$	$A \text{ OR } B$
A'	$\text{COMP}(A)$
$A \oplus B$	$A \text{ XOR } B$
$A - B$	$A \text{ AND } (\text{COMP}(B))$

- Principios fundamentales:
 - 1) Adición,
 - 2) Inclusión-Exclusión y
 - 3) Multiplicación.



Tarea -> Evento
o Elemento

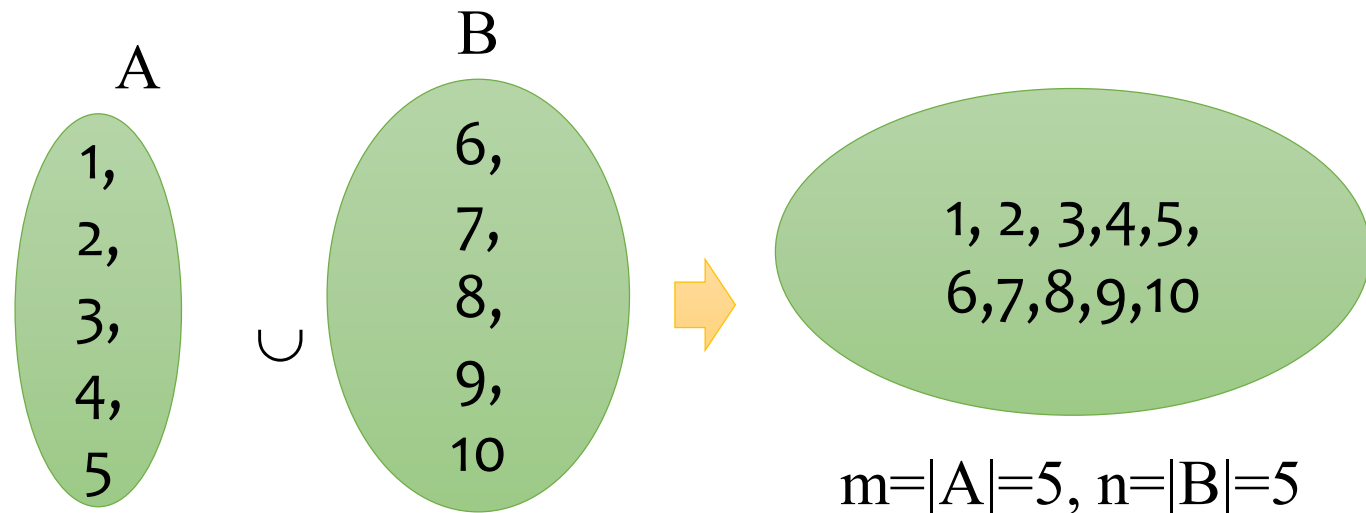
Principio de Adición

- Sea A y B dos tareas mutuamente excluyentes.
- Supongamos que la tarea A puede hacerse en m maneras y la tarea B en n maneras.
- Entonces, A y B pueden hacerse en $m + n$ maneras.



Principio de Adición

- Sea A y B dos tareas mutuamente excluyentes.
- Supongamos que la tarea A puede hacerse en m maneras y la tarea B en n maneras.
- Entonces, A y B pueden hacerse en $m + n$ maneras.





Principio de Adición

- EJEMPLO:
 - Un estudiante ha elegido 4 cursos y necesita uno más para terminar. Hay 15 cursos en inglés, 10 en francés y 6 en alemán. Cuántas maneras de escoger el quinto curso tiene el estudiante?



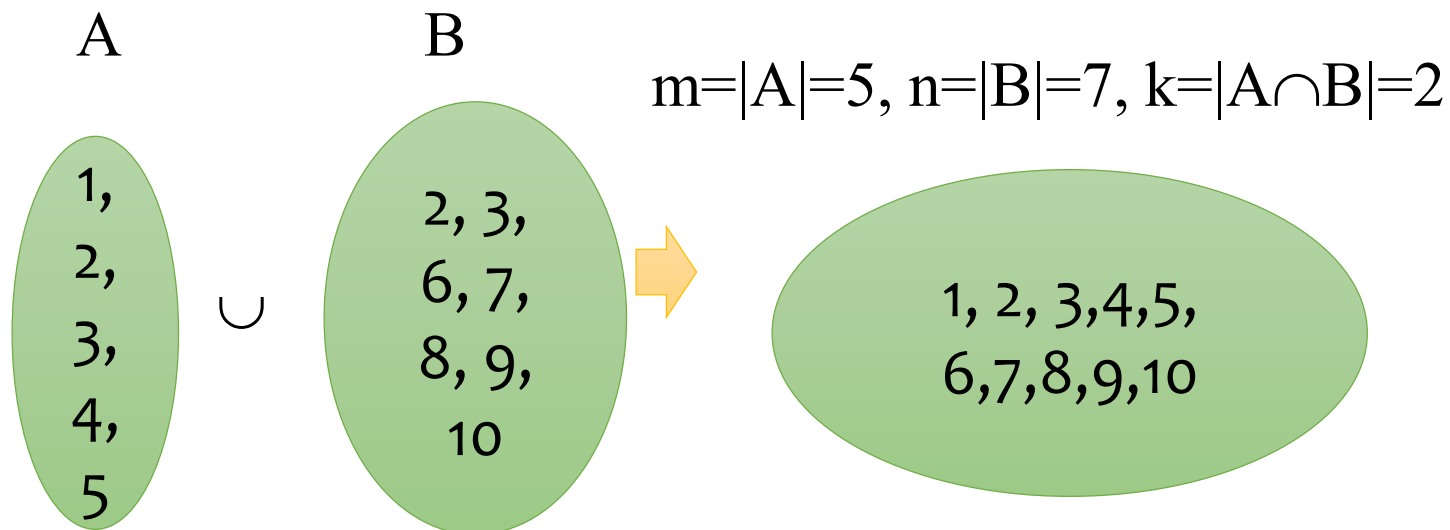
Principio de Adición

- EJEMPLO:
 - Un estudiante ha elegido 4 cursos y necesita uno más para terminar. Hay 15 cursos en inglés, 10 en francés y 6 en alemán. Cuántas maneras de escoger el quinto curso tiene el estudiante?
- Solución: $15 + 10 + 6 = 31$



Principio de Inclusión-Exclusión

- Supongamos que una tarea A puede realizarse en m maneras, la tarea B en n maneras, y ambas (i.e., simultáneamente o comunes) pueden realizarse en k diferentes maneras.
- Entonces la tarea A o la B pueden realizarse en $m + n - k$ maneras.





Principio de Inclusión-Exclusión

- Supongamos que una tarea A puede realizarse en m maneras, la tarea B en n maneras, y ambas (i.e., simultáneamente o comunes) pueden realizarse en k diferentes maneras.
- Entonces la tarea A o la B pueden realizarse en $m + n - k$ maneras.
 - EJEMPLO: En cuántas maneras puedes seleccionar un rey o una carta negra de un juego estándar de cartas? (26 son rojas y 26 negras).



Principio de Inclusión-Exclusión

- Supongamos que una tarea A puede realizarse en m maneras, la tarea B en n maneras, y ambas (i.e., simultáneamente o comunes) pueden realizarse en k diferentes maneras.
- Entonces la tarea A o la B pueden realizarse en $m + n - k$ maneras.
 - EJEMPLO: En cuántas maneras puedes seleccionar un rey o una carta negra de un juego estándar de cartas? (26 son rojas y 26 negras).
 - SOLUCIÓN: Cuántos reyes hay? 4, Cuántas cartas negras? 26, Cuántos reyes negros hay? 2...
 - Entonces, $4 + 26 - 2 = 28$ maneras.

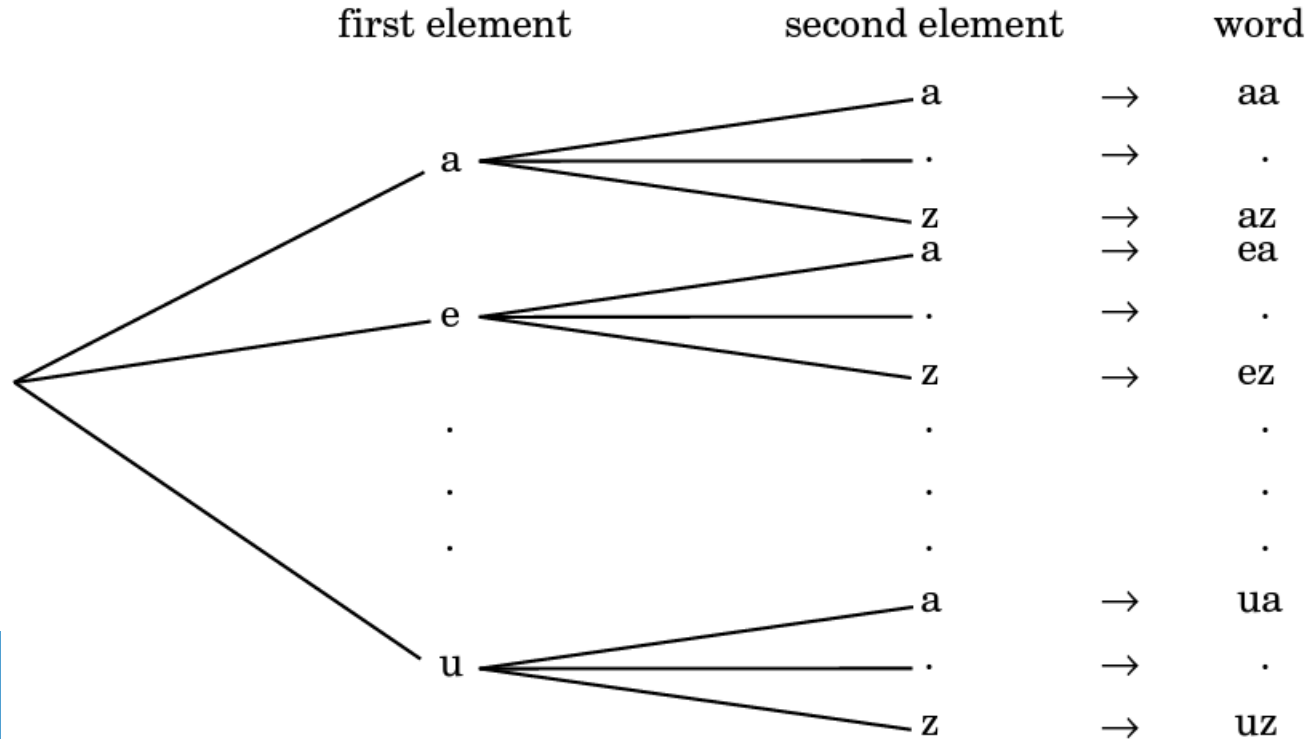


Principio de Multiplicación

- Supongamos que una tarea T se compone de dos subtareas: T_1 seguida de T_2 . Si la subtarea T_1 puede realizarse en m_1 maneras y la subtarea T_2 en m_2 maneras diferentes de hacerse por cada subtarea T_1 , entonces la tarea T puede hacerse en $m_1 m_2$ maneras.
- EJEMPLO: Encuentra el número de palabras de dos letras que comiencen con una vocal.

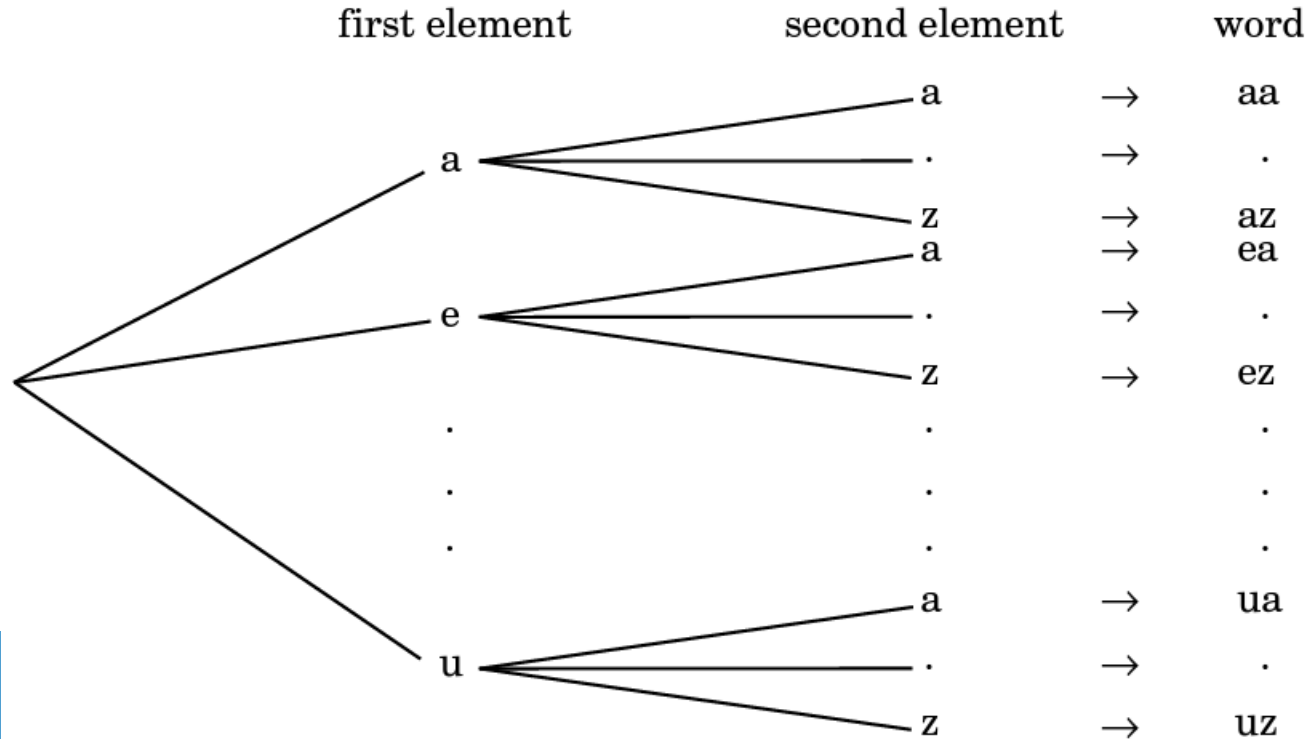


Principio de Multiplicación





Principio de Multiplicación



5

*

27

=

135



Principio de Multiplicación

- Supongamos que una tarea T se compone de dos subtareas: T_1 seguida de T_2 . Si la subtarea T_1 puede realizarse en m_1 maneras y la subtarea T_2 en m_2 maneras diferentes de hacerse por cada subtarea T_1 , entonces la tarea T puede hacerse en $m_1 m_2$ maneras.
- EJEMPLO: Encuentra el número de palabras de dos letras que comiencen con una vocal.
- SOLUCIÓN: La tarea T , consiste de dos subtareas: T_1 consiste en seleccionar la primera letra vocal, y T_2 consiste en seleccionar la segunda letra: $5 * 27 = 135$



Principio de Multiplicación

- Supongamos que una tarea T se compone de dos subtareas: T_1 seguida de T_2 . Si la subtaska T_1 puede realizarse en m_1 maneras y la subtaska T_2 en m_2 maneras diferentes de hacerse por cada subtaska T_1 , entonces la tarea T puede hacerse en $m_1 m_2$ maneras.
- Generalizando...
- Supongamos que una tarea T puede ser realizada por n subtareas sucesivas T_1, T_2, \dots, T_n . Si la subtaska T_i , $1 \leq i \leq n$, puede realizarse en m_i maneras después de que T_{i-1} se complete, entonces la tarea T puede hacerse en $m_1 m_2 \cdots m_n$ maneras.



Principio de Multiplicación

- EJERCICIO: El tipo de placa en Jalisco para automóviles (sin restricciones en el uso de letras) consiste en 3 letras (27 opciones) y 4 números (10 opciones).
 - Cuántas placas es posible tener con esta combinación de letras y números?

Z Z Z 9 9 9 9



Principio de Multiplicación

- EJERCICIO: El tipo de placa en Jalisco para automóviles (sin restricciones en el uso de letras) consiste en 3 letras (27 opciones) y 4 números (10 opciones).
 - Cuántas placas es posible tener con esta combinación de letras y números?

Z Z Z 9 9 9 9

$$196.83 \times 10^6 = 27 * 27 * 27 * 10 * 10 * 10 * 10$$

Actividad en Clase

MAESTRÍA EN
SISTEMAS
COMPUTACIONALES



ITESO

Universidad Jesuita
de Guadalajara



Qué hemos revisado al momento?

- Curso en Moodle, guía de aprendizaje
- Objetivos del Curso y de la clase
- Importancia de las matemáticas y sus aplicaciones
- Principios fundamentales del conteo:
 - 1) Adición,
 - 2) Inclusión-Exclusión y
 - 3) Multiplicación



Tarea

- Descripción en Moodle
- Fecha límite de entrega: Lunes, 21/08/2017, 11:55pm
- Incluir:
 - Nombre del estudiante, Fecha, Nombre del curso, Nombre del programa, Nombre del profesor.
- Subir al Moodle el documento en PDF.