Notas de Matemáticas Discretas

Luis Eduardo Gamboa Guzmán [1]

Universidad Michoacana de San Nicolás de Hidalgo
Facultad de Ingeniería Eléctrica

08 de julio de 2008

[1] http://lc.fie.umich.mx/~legg/

Índice general

Capítulo 1

Sobre este documento

Este documento es una recopilación de conceptos y ejercicios obtenidos mayormente de [Alagar 1989], [Doerr 1985], [Enderton 2000], [Hopcroft 1979], [Knuth 1989], [Rosen 1999] y [Tourlakis 1984]. Debido a la cantidad de traducciones de estas fuentes, las referencias a cada concepto en específico han sido removidas. Los cambios realizados incluyen traducción, notación, reordenamiento, expansión y corrección de errores (aunque pocos) del material.

Los capítulos 2, 3, 4, 5 y 6 están basados en [Alagar 1989] y [Doerr 1985]. En el tema de inducción se utiliza material de [Aho 1995]. La sección de funciones primitivas recursivas fue adecuada del material presentado por [Tourlakis 1984].

Lo referente grafos y árboles (capítulos 9 y 10) contiene material de [Doerr 1985], [Alagar 1989] y [Rosen 1999].

Muchos ejercicios han sido desarrollados enteramente por el autor de esta recopilación. Otros tantos han sido modificados de los expuestos en la bibliografía para hacerlos más claros y utilizando los conceptos que abarca el curso.

El documento ha sido desarrollado utilizando LaTeX, una excelente herramienta para la edición profesional de textos.

Capítulo 2

Métodos de demostración

La resolución de problemas, diseño de algoritmos y programación requieren un razonamiento lógico completo. La *lógica* trata los métodos y el arte del razonamiento sistemático.

2.1. Lógica proposicional

Una proposición es una sentencia declarativa que es verdadera o falsa pero no ambas. Por ejemplo, "la mañana es fría".

2.1.1. Proposiciones compuestas

Una proposición que es indivisible se conoce como proposición primitiva. Las sentencias derivadas de las primitivas y de varios conectores lógicos como *no*, *y*, *o*, *si...entonces* y *sí y sólo sí* se conocen como proposiciones compuestas.

Ejemplo

- Un girasol es amarillo.
- El Sahara es un desierto.
- 17 es un número primo y 25 no es un cuadrado perfecto.
- Existe una infinidad de números perfectos.
- ¿Estás durmiendo?

2.1.2. Tablas de verdad

Las tablas de verdad son una forma conveniente de mostrar los valores de una proposición compuesta. En su construcción, usamos *1* para verdadero y *0* para falso, aunque también es común utilizar *T* y *F*.

NO

Una sentencia que es modificada con el conectivo *no* es llamada la negación de la sentencia original. Simbólicamente, sí P es una proposición entonces $\neg P$ (no P), denota la negación de P. En el cuadro 2.1 se muestra la tabla de verdad de NO.

P	$\neg P$
1	0
0	1

Cuadro 2.1: Tabla de verdad de NO

Y

La conjunción de P,Q es denotada por $P \wedge Q$. La conjunción es verdadera sólo si P y Q son verdaderos. En el cuadro 2.2 se muestra la tabla de verdad de Y.

P	Q	$P \wedge Q$
0	0	0
0	1	0
1	0	0
1	1	1

Cuadro 2.2: Tabla de verdad de Y

O

La disyunción de P,Q es denotada por $P \vee Q$. La disyunción es verdadera si al menos uno de sus elementos es verdad P, Q es verdadero. En el cuadro 2.3 se muestra la tabla de verdad de O.

P	Q	$P \vee Q$
0	0	0
0	1	1
1	0	1
1	1	1

Cuadro 2.3: Tabla de verdad de O

O EXCLUSIVO

El símbolo $\oplus$ representa el O EXCLUSIVO (XOR), que es incluido en muchos lenguajes de programación. Una proposición $P \oplus Q$ se lee como "P o Q pero no ambos". En el cuadro 2.4 se muestra la tabla de verdad de XOR.

P	Q	$P \oplus Q$
0	0	0
0	1	1
1	0	1
1	1	0

Cuadro 2.4: Tabla de verdad de XOR

IMPLICACION

Para dos declaraciones P,Q, decimos "P implica Q" y se escribe $P \rightarrow Q$ para denotar la implicación de Q por P. La proposición P es llamada la hipótesis o antecedente de la implicación; Q es llamada la conclusión o consecuente de la implicación. En el cuadro 2.5 se muestra la tabla de verdad de la IMPLICACION.

P	Q	$P \rightarrow Q$
0	0	1
0	1	1
1	0	0
1	1	1

Cuadro 2.5: Tabla de verdad de IMPLICACION

Como ejemplo, consideremos que el profesor dice a sus alumnos: "si obtienes 9 o más en el examen, aprobaras el curso". Entonces:

- P: Obtienes 9 o más en el examen.
- Q: Apruebas el curso.

Una vez que se termina el curso, existen 4 posibles situaciones:

1. La calificación del examen ha sido menor que 9 y no se aprobó el curso. La promesa no ha sido rota, pues no se cumplió con P.
2. La calificación del examen ha sido menor que 9 y se aprobó el curso. La promesa no ha sido rota, es posible que por otras razones se haya aprobado.
3. La calificación del examen ha sido mayor o igual que 9 y no se aprobó el curso. La promesa ha sido rota, pues se ha cumplido con P y no se ha aprobado el curso.
4. La calificación del examen ha sido mayor o igual que 9 y se aprobó el curso. La promesa ha sido cumplida.

SI Y SOLO SI

Otra declaración común en matemáticas es "P si y sólo si Q", o simbólicamente $P \leftrightarrow Q$. Esto es llamado la equivalencia de dos proposiciones, P, Q. Formulaciones alternativas son:

- si P entonces Q, y si Q entonces P
- Q es una condición necesaria y suficiente para P

La tabla de verdad de SII se muestra en el cuadro 2.6.

P	Q	$P \leftrightarrow Q$
0	0	1
0	1	0
1	0	0
1	1	1

Cuadro 2.6: Tabla de verdad de SII

2.1.3. Fórmulas, Tautologías y Contradicciones

Una fórmula o forma lógica $f(x, y, z, \ldots)$ es una expresión lógica en la que $x, y, z, \ldots$ son proposiciones o variables lógicas. Por ejemplo $(x \rightarrow y) \rightarrow z$ y $(x \wedge \neg y) \vee z$ son fórmulas.

Por convención los conectores en una fórmula sin paréntesis son aplicados en el siguiente orden de prioridad:

- $\neg$ (prioridad más alta)
- $\wedge$
- $\vee$
- $\rightarrow$, $\leftrightarrow$ (prioridad más baja)

los paréntesis refuerzan la prioridad para subexpresiones encerradas. Los conectores con la misma precedencia son aplicados de izquierda a derecha. Entonces la fórmula $(x \wedge \neg y) \vee z$ puede ser escrita sin paréntesis como $x \wedge \neg y \vee z$; las fórmulas $(x \rightarrow \neg y) \rightarrow z$, $x \rightarrow \neg(y \rightarrow z)$ y $x \rightarrow (\neg y \rightarrow z)$ son diferentes.

Tautología

Una fórmula que siempre es verdad se conoce como tautología. Entonces $x \vee \neg x$ es una tautología. Si dos fórmulas f y g tienen valores idénticos en sus tablas de verdad, entonces $f \leftrightarrow g$ es una tautología. Esto es, si dos fórmulas f y g son lógicamente equivalentes, denotado por $f \equiv g$, si y sólo si $f \leftrightarrow g$ es una tautología.

Por ejemplo, se quiere comprobar si $P \rightarrow Q \equiv \neg P \vee Q$ son lógicamente equivalentes, la tabla de verdad se muestra en el cuadro 2.7.

Contradicción

Se dice que una fórmula es una contradicción si siempre es falsa. $x \wedge \neg x$ es una contradicción.

P	Q	$P \to Q$	$\neg P \vee Q$	$(P \to Q) \leftrightarrow (\neg P \vee Q)$
0	0	1	1	1
0	1	1	1	1
1	0	0	0	1
1	1	1	1	1

Cuadro 2.7: Tautología para demostrar $P \to Q \equiv \neg P \vee Q$

Conjunto de equivalencias lógicas

Leyes de idempotencia

$$P \equiv P \vee P$$
$$P \equiv P \wedge P$$

Leyes conmutativas

$$P \vee Q \equiv Q \vee P$$
$$P \wedge Q \equiv Q \wedge P$$

Leyes asociativas

$$(P \vee Q) \vee R \equiv P \vee (Q \vee R)$$
$$(P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R)$$

Leyes distributivas

$$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$$
$$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$$

Leyes de absorción

$$P \vee 0 \equiv P$$
$$P \vee 1 \equiv 1$$
$$P \wedge 0 \equiv 0$$
$$P \wedge 1 \equiv P$$
$$P \wedge (P \vee Q) \equiv P$$
$$P \vee (P \wedge Q) \equiv P$$

Leyes de De Morgan

$$\neg(P \vee Q) \equiv \neg P \wedge \neg Q$$
$$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$$

Leyes de complemento

$$\neg 1 \equiv 0$$
$$\neg 0 \equiv 1$$
$$P \vee \neg P \equiv 1$$
$$P \wedge \neg P \equiv 0$$
$$\neg(\neg P) \equiv P$$

Ley de implicación

$$P \to Q \equiv \neg P \vee Q$$
$$P \to Q \equiv \neg Q \to \neg P$$

Ley de Equivalencia

$$(P \equiv Q) \equiv (P \to Q) \wedge (Q \to P)$$

2.1.4. Simplificación

Se dice que una fórmula g es una simplificación de f si g es una equivalencia lógica de f y tiene menos conectores. Ejemplo de simplificación:

$$\begin{array}{lcll}
(P \to Q) \wedge \neg Q & \equiv & (\neg P \vee Q) \wedge \neg Q & \text{Implicación} \\
& \equiv & \neg Q \wedge (\neg P \vee Q) & \text{Conmutativa} \\
& \equiv & (\neg Q \wedge \neg P) \vee (\neg Q \wedge Q) & \text{Distributiva} \\
& \equiv & (\neg Q \wedge \neg P) \vee (Q \wedge \neg Q) & \text{Conmutativa} \\
& \equiv & (\neg Q \wedge \neg P) \vee 0 & \text{Complemento} \\
& \equiv & \neg Q \wedge \neg P & \text{Absorción} \\
& \equiv & \neg(Q \vee P) & \text{De Morgan}
\end{array}$$

2.1.5. Obtención de fórmulas

Supongamos una función $f(P, Q)$ cuya tabla de verdad es conocida y se desea encontrar la expresión equivalente. Para encontrar f se pueden utilizar dos técnicas:

1. Forme expresiones lógicas utilizando el operador conjunción para generar un valor *verdadero* en los casos en los que la función regresa un valor verdadero. Finalmente, forme una disyunción con las expresiones encontradas.

2. Forme expresiones lógicas utilizando el operador disyunción para generar un valor *falso* en los casos en los que la función regresa un valor falso. Finalmente, forme una conjunción con las expresiones encontradas.

Ejemplo: Encuentre una función f que genere los valores de la tabla de verdad en el cuadro 2.8.

P	Q	R	$f(P, Q, R)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Cuadro 2.8: Tabla de verdad para $f(P, Q, R)$

A continuación se ilustran las expresiones obtenidas para valores verdaderos y para valores falsos.

P	Q	R	$f(P,Q,R)$	Técnica 1	Técnica 2
0	0	0	0		$P \vee Q \vee R$
0	0	1	0		$P \vee Q \vee \neg R$
0	1	0	0		$P \vee \neg Q \vee R$
0	1	1	1	$\neg P \wedge Q \wedge R$	
1	0	0	0		$\neg P \vee Q \vee R$
1	0	1	0		$\neg P \vee Q \vee \neg R$
1	1	0	1	$P \wedge Q \wedge \neg R$	
1	1	1	1	$P \wedge Q \wedge R$	

Dadas las expresiones que generan valores verdaderos, podemos deducir que:

$$f(P,Q,R) \equiv (\neg P \wedge Q \wedge R) \vee (P \wedge Q \wedge \neg R) \vee (P \wedge Q \wedge R)$$

y dadas las expresiones que generan valores falsos podemos deducir:

$$f(P,Q,R) \equiv (P \vee Q \vee R) \wedge (P \vee Q \vee \neg R) \wedge (P \vee \neg Q \vee R) \wedge (\neg P \vee Q \vee R) \wedge (\neg P \vee Q \vee \neg R)$$

2.1.6. Forma Normal Conjuntiva

Una forma lógica está en forma normal conjuntiva (CNF) si cumple con alguno de los siguientes criterios:

1. Es una sola variable.
2. Es la negación de un sólo símbolo.
3. Es la disyunción de varios términos en donde cada término es una variable o una negación de una variable.
4. Es la conjunción de dos o más conjuntos del tipo de los tres tipos anteriores.

P, $\neg Q$, $(\neg P \vee Q \vee R) \wedge P$, $\neg P \vee Q \vee R$ y $(\neg P \vee Q) \wedge (\neg Q \vee R) \wedge (\neg R \vee P)$ están en forma normal conjuntiva. $(P \wedge Q) \vee R$ y $\neg(P \wedge Q) \vee (P \rightarrow Q)$ no están en forma normal conjuntiva.

Para llevar una forma lógica a una forma normal conjuntiva equivalente, las siguientes operaciones deben realizarse:

1. Quitar $\rightarrow$ $\leftrightarrow$, reemplazándolos con expresiones equivalentes que usen únicamente $\neg$,$\wedge$,$\vee$.
2. Usar leyes distributivas y asociativas.
3. Aplicar repetidamente leyes de De Morgan si es necesario, para obtener una conjunción de disyunciones.

Cuando cada conjunción contiene una variable y su negación, la forma normal conjuntiva es una tautología. De manera análoga, cada conjunción de la forma normal equivalente de una tautología debe contener una variable y su negación.

Ejemplo: Convertir $(P \rightarrow Q) \wedge [Q \vee (P \wedge R)]$ a forma normal conjuntiva:

$$
\begin{array}{lcll}
(P \to Q) \wedge [Q \vee (P \wedge R)] & \equiv & (\neg P \vee Q) \wedge [Q \vee (P \wedge R)] & \text{Implicación} \\
& \equiv & (\neg P \vee Q) \wedge [(Q \vee P) \wedge (Q \vee R)] & \text{Distributiva} \\
& \equiv & (\neg P \vee Q) \wedge (Q \vee P) \wedge (Q \vee R) & \text{Asociativa}
\end{array}
$$

2.1.7. Forma Normal Disyuntiva

Una forma lógica está en forma normal disyuntiva (DNF) si cumple con alguno de los siguientes criterios:

1. Es una sola variable.
2. Es la negación de un sólo símbolo.
3. Es la conjunción de varios términos en donde cada término es una variable o una negación de una variable.
4. Es la disyunción de dos o más conjuntos de los tres tipos anteriores.

P, $\neg Q$, $(\neg P \wedge Q \wedge R) \vee P$, $\neg P \wedge Q \wedge R$ y $(\neg P \wedge Q) \vee (\neg Q \wedge R) \vee (\neg R \wedge P)$ están en forma normal disyuntiva. $(P \vee Q) \wedge R$ y $\neg(P \wedge Q) \vee (P \to Q)$ no están en forma normal conjuntiva.

Para llevar una forma lógica a una forma normal disyuntiva equivalente, las siguientes operaciones deben realizarse:

1. Quitar $\to \leftrightarrow$, reemplazándolos con expresiones equivalentes que usen únicamente $\neg$,$\wedge$,$\vee$.
2. Usar leyes distributivas y asociativas.
3. Aplicar repetidamente leyes de De Morgan si es necesario, para obtener una disyunción de conjunciones.

Ejemplo: Convertir $(P \to Q) \wedge [Q \vee (P \wedge R)]$ a forma normal disyuntiva:

$$
\begin{array}{lcll}
(P \to Q) \wedge [Q \vee (P \wedge R)] & \equiv & (\neg P \vee Q) \wedge [Q \vee (P \wedge R)] & \text{Implicación} \\
& \equiv & [(\neg P \vee Q) \wedge Q] \vee [(\neg P \vee Q) \wedge (P \wedge R)] & \text{Distributiva} \\
& \equiv & [(\neg P \wedge Q) \vee (Q \wedge Q)] \vee [(\neg P \vee Q) \wedge (P \wedge R)] & \text{Distributiva} \\
& \equiv & [(\neg P \wedge Q) \vee Q] \vee [(\neg P \vee Q) \wedge (P \wedge R)] & \text{Idempotencia} \\
& \equiv & [(\neg P \wedge Q) \vee Q] \vee [((\neg P \vee Q) \wedge P) \wedge R] & \text{Asociativa} \\
& \equiv & [(\neg P \wedge Q) \vee Q] \vee [((\neg P \wedge P) \vee (Q \wedge P)) \wedge R] & \text{Distributiva} \\
& \equiv & [(\neg P \wedge Q) \vee Q] \vee [(0 \vee (Q \wedge P)) \wedge R] & \text{Complemento} \\
& \equiv & (\neg P \wedge Q) \vee Q \vee (Q \wedge P \wedge R) &
\end{array}
$$

2.2. Inferencia Lógica

En lógica proposicional, utilizamos reglas de inferencia para deducir proposiciones verdaderas de aquellas que se saben son verdad. Utilizamos $A \Rightarrow B$ para indicar que B es verdadero siempre y cuando A sea verdadero.

Modus Ponens

$$P \wedge (P \to Q) \quad \Rightarrow \quad Q$$

Modus Tollens

$$\neg Q \wedge (P \to Q) \quad \Rightarrow \quad \neg P$$

Adición disyuntiva

$$P \quad \Rightarrow \quad P \vee Q$$

Simplificación conjuntiva

$$P \wedge Q \quad \Rightarrow \quad P$$

$$P \wedge Q \quad \Rightarrow \quad Q$$

Simplificación disyuntiva

$$(P \vee Q) \wedge \neg Q \quad \Rightarrow \quad P$$

$$(P \vee Q) \wedge \neg P \quad \Rightarrow \quad Q$$

Regla de la cadena

$$(P \to Q) \wedge (Q \to R) \quad \Rightarrow \quad P \to R$$

Tautologías

$$P \to (Q \to P)$$

$$P \to (Q \to R) \to ((P \to Q) \to (P \to R))$$

$$(\neg Q \to \neg P) \to (P \to Q)$$

Estas reglas no son equivalencias, meramente son proposiciones que se cumplen bajo ciertas circunstancias. En la siguiente tabla analizamos el Modus Ponens.

P	Q	$P \to Q$	$P \wedge (P \to Q)$
0	0	1	0
0	1	1	0
1	0	0	0
1	**1**	1	**1**

Noté que cuando $P \wedge (P \to Q)$ es verdad Q también es verdadero. Cabe señalar que en un caso $P \wedge (P \to Q)$ es falso y Q es verdadero, este caso no es de nuestro interés, pues no se trata de equivalencias lógicas, meramente de poder inferir valores de verdad.

Ahora analicemos la regla de la cadena:

P	Q	R	$P \to Q$	$Q \to R$	$(P \to Q) \wedge (Q \to R)$	$P \to R$
0	0	0	1	1	**1**	**1**
0	0	1	1	1	**1**	**1**
0	1	0	1	0	0	1
0	1	1	1	1	**1**	**1**
1	0	0	0	1	0	0
1	0	1	0	1	0	1
1	1	0	1	0	0	0
1	1	1	1	1	**1**	**1**

Nuevamente, se puede observar como siempre que $(P \to Q) \wedge (Q \to R)$ es verdadero, $P \to R$ es verdadero también.

2.3. Argumentos válidos

Un patrón general de inferencia o argumento es usualmente presentado como una serie de declaraciones $P_1, P_2, \ldots, P_n$ seguidos de una conclusión Q. Las proposiciones $P_1, P_2, \ldots, P_n$ son llamadas *premisas* y Q es llamado *consecuencia*. El argumento $P_1 \wedge P_2 \wedge \cdots \wedge P_n \Rightarrow Q$ es válido si y sólo si $P_1 \wedge P_2 \wedge \cdots \wedge P_n \rightarrow Q$ es una tautología. Un argumento que no es válido se conoce como *falacia*.

En otras palabras, para que un argumento sea válido es necesario que cuando todas las premisas sean verdaderas, la consecuencia también lo sea.

2.4. Prueba directa

Para probar si un argumento $P \Rightarrow Q$ es válido:

1. Se sustituye P por una secuencia de declaraciones $P_1, P_2, \ldots, P_n$, donde cada P_i está en P o es una tautología,

2. o puede ser derivado de declaraciones P_j, P_k anteriores $(j, k < i)$ por medio de reglas de inferencia.

Ejemplo 1. Probar la declaración $[P \rightarrow (Q \rightarrow R)] \rightarrow [Q \rightarrow (P \rightarrow R)]$:

1.	$P \rightarrow (Q \rightarrow R)$	Premisa
2.	$[P \rightarrow (Q \rightarrow R)] \rightarrow [(P \rightarrow Q) \rightarrow (P \rightarrow R)]$	Tautología
3.	$(P \rightarrow Q) \rightarrow (P \rightarrow R)$	Modus Ponen 1,2
4.	$Q \rightarrow (P \rightarrow Q)$	Tautología
5.	$Q \rightarrow (P \rightarrow R)$	Regla de la cadena 4,3

Ejemplo 2. Probar la declaración $P \rightarrow P$:

1.	P	Premisa
2.	$P \rightarrow (P \rightarrow P)$	Tautología
3.	$P \rightarrow P$	Modus Ponens 1,2

Ejemplo 3. *Estoy cansado o estoy enfermo. Si estoy enfermo me voy a mi casa. No me voy a mi casa. Entonces estoy cansado.* Suponemos que los primeras tres declaraciones son verdaderas, queremos comprobar la verdad de la última declaración, que es la consecuencia. Denotemos "estoy cansado" con P, "estoy enfermo" con Q, y "me voy a mi casa" con R. La secuencia de declaraciones se convierte en:

$$\begin{array}{c} P \vee Q \\ Q \rightarrow R \\ \neg R \\ P \end{array}$$

y se quiere probar $[(P \vee Q) \wedge (Q \rightarrow R) \wedge \neg R] \Rightarrow P$.

1.	$P \vee Q$	Premisa
2.	$Q \rightarrow R$	Premisa
3.	$\neg R$	Premisa
4.	$(Q \rightarrow R) \rightarrow (\neg R \rightarrow \neg Q)$	Tautología
5.	$\neg R \rightarrow \neg Q$	Modus Ponens 2,4
6.	$\neg Q$	Modus Ponens 3,5
7.	P	Simplificación disyuntiva 1,6

2.5. Prueba indirecta

Para probar si un argumento $P \Rightarrow Q$ es válido, se puede optar por utilizar una equivalencia lógica y en su lugar probar $\neg Q \Rightarrow \neg P$, esta prueba se conoce como prueba indirecta. El proceso para probar $\neg Q \Rightarrow \neg P$ puede hacerse mediante prueba directa.

Ejemplo 1. Probar $[(P \vee Q) \wedge (Q \rightarrow R) \wedge \neg R] \Rightarrow P$, equivale a demostrar que: $\neg P \Rightarrow \neg[(P \vee Q) \wedge (Q \rightarrow R) \wedge \neg R]$, lo cual puede ser poco visible, ya que la consecuencia es más compleja que la premisa, sin embargo, se puede trabajar en la consecuencia para hacerla más simple:

	$\neg[(P \vee Q) \wedge (Q \rightarrow R) \wedge \neg R]$	Consecuencia
$\equiv$	$\neg(P \vee Q) \vee \neg(Q \rightarrow R) \vee R]$	De Morgan
$\equiv$	$\neg P \wedge \neg Q \vee Q \wedge \neg R \vee R$	De Morgan
$\equiv$	$\neg P \wedge \neg Q \vee (Q \vee R) \wedge (\neg R \vee R)$	Distributiva
$\equiv$	$\neg P \wedge \neg Q \vee (Q \vee R) \wedge 1$	Complemento
$\equiv$	$\neg P \wedge \neg Q \vee Q \vee R$	Absorción
$\equiv$	$(\neg P \vee Q) \wedge (\neg Q \vee Q) \vee R$	Distributiva
$\equiv$	$(\neg P \vee Q) \wedge 1 \vee R$	Complemento
$\equiv$	$\neg P \vee Q \vee R$	Absorción

Una vez que la consecuencia está en una forma manejable, procedemos a probar que $\neg P \Rightarrow \neg P \vee Q \vee R$:

1.	$\neg P$	Premisa
2.	$\neg P \vee Q$	Adición disyuntiva 1
2.	$\neg P \vee Q \vee R$	Adición disyuntiva 2

con lo cual queda demostrado que $\neg P \Rightarrow \neg P \vee Q \vee R$ y por prueba indirecta se demuestra $[(P \vee Q) \wedge (Q \rightarrow R) \wedge \neg R] \Rightarrow P$

2.6. Pruebas vacuas

Toda implicación es verdadera cuando la premisa es falsa, por lo tanto, si es posible demostrar que P es falso en $P \Rightarrow Q$, el argumento es válido.

Ejemplo 1. *Si* $0 > 1$ *entonces* $0^2 > 0$, usemos P para denotar la proposición $0 > 1$ y Q para denotar $0^2 > 0$. Entonces la prueba consiste en demostrar que $P \Rightarrow Q$.

Puesto que P es evidentemente falso y siempre que la premisa es falsa la implicación es verdadera, se demuestra por prueba vacua que $P \Rightarrow Q$, es decir *Si* $0 > 1$ *entonces* $0^2 > 0$, es un argumento válido.

2.7. Pruebas triviales

Si se tiene una implicación y se conoce que la consecuencia es verdadera, entonces la implicación es verdadera. La prueba trivial consiste en demostrar que en $P \Rightarrow Q$, Q es verdadero.

Ejemplo 1. *Si* $a \geq b$ *entonces* $a^0 \geq b^0$. Puesto que $a^0 = b^0 = 1$, se tiene que la consecuencia es verdad y por lo tanto queda demostrado que *Si* $a \geq b$ *entonces* $a^0 \geq b^0$ es un argumento válido.

2.8. Prueba por contradicción

Considere un teorema $P \Rightarrow Q$, donde P representa las premisas $P_1 \wedge P_2 \wedge \cdots \wedge P_n$. Este método de prueba está basado en la equivalencia $P \rightarrow Q \equiv \neg(P \wedge \neg Q)$. Lo que indica que si $P \Rightarrow Q$, entonces $P \wedge \neg Q$ es siempre falso. Esto indica que un método de prueba válido es negar la consecuencia del teorema e incluir esta negación a las premisas. Si una contradicción puede ser implicada desde este conjunto de proposiciones, la prueba está completa.

Ejemplo 1. Probar la declaración $P \rightarrow R, Q \rightarrow S, P \vee Q \Rightarrow S \vee R$ por contradicción:

1.	$\neg(S \vee R)$	Consecuencia negada
2.	$\neg S \wedge \neg R$	De Morgan 1
3.	$\neg S$	Simplificación conjuntiva 2
4.	$Q \rightarrow S$	Premisa
5.	$\neg Q$	Modus Tollens 3,4
6.	$\neg R$	Simplificación conjuntiva 2
7.	$P \rightarrow R$	Premisa
8.	$\neg P$	Modus Tollens 6,7
9.	$\neg P \wedge \neg Q$	Conjunción 5,8
10.	$\neg(P \vee Q)$	De Morgan 9
11.	$P \vee Q$	Premisa
12.	0	Conjunción 10,11

Ejemplo 2. Probar $[(P \vee Q) \wedge (Q \rightarrow R) \wedge \neg R] \Rightarrow P$ por contradicción:

1.	$P \vee Q$	Premisa
2.	$Q \rightarrow R$	Premisa
3.	$\neg R$	Premisa
4.	$\neg P$	Consecuencia negada
5.	Q	Simplificación disyuntiva 1,4
6.	R	Modus Ponens 5,2
7.	$R \wedge \neg R$	Conjunción 3,6

2.9. Prueba por casos

Si se tuviera un argumento de la forma $[P_1 \vee P_2 \vee P_3 \vee \ldots \vee P_n] \Rightarrow Q$, es decir como una disyunción de premisas, es posible utilizar una tautología y expresar:

$$[(P_1 \vee P_2 \vee P_3 \vee \ldots \vee P_n) \rightarrow Q] \leftrightarrow [(P_1 \rightarrow Q) \wedge (P_2 \rightarrow Q) \wedge (P_3 \rightarrow Q) \wedge \ldots \wedge (P_n \rightarrow Q)]$$

Por lo que es posible demostrar que el argumento es válido probando que todos los casos $P_i \rightarrow Q$ son argumentos válidos.

2.10. Prueba por equivalencia

2.11. Lógica de predicados

Frecuentemente nos encontramos con proposiciones que representan hechos sobre una colección de objetos. Por ejemplo:

- Algunos programadores son inteligentes.
- Todos los municipios tienen escuelas públicas.
- Todos los matemáticos son tenaces.
- Existe un número impar que no es primo.

Cada declaración conlleva una aserción común a algunos objetos que pertenecen a un universo. Puesto que las declaraciones *para todos* y *existe* (o *para algún*) no están disponibles en lógica proposicional, ninguna de estas declaraciones puede ser escrita en forma lógica. Cuando agregamos símbolos para estas declaraciones junto con las reglas de uso en lógica proposicional, obtenemos lógica de predicados. El lenguaje de lógica de primer orden es obtenido cuando símbolos de función son agregados a lógica de predicados.

Contrario a las constantes, las variables no tienen un significado por sí mismas. Una sentencia como $2 + 3 = 5$ es una aserción cuyo valor de verdad es conocido. Sin embargo sentencias como $x > 6$, "él es abogado" y "y es un entero" contienen variables: x, "él", y. Dichas sentencias no pueden ser comprobadas ni refutadas. Sin embargo, cuando asignamos valores a estas variables el valor de verdad puede ser conocido. Por lo tanto cuando $x = 3$, "$x > 6$" es falso; cuando "él" es reemplazado por Juan, la declaración "Juan es abogado" tendrá un valor de verdad; cuando $y = 2$ la declaración "y es un entero" es verdadera. Tales declaraciones cuyos valores de verdad dependen de los valores que tengan las variables se conocen como *predicados*.

Es importante observar que en lógica proposicional una variable toma valores (falso, verdadero) y en lógica de predicados una variable toma valores de un universo de discurso U. Las variables $x_1, x_2, \ldots, x_n$ en $P(x_1, x_2, \ldots, x_n)$ son llamadas variables libres del predicado. Por consecuencia, el valor de verdad de

$P(x_1, x_2, \ldots, x_n)$ varia conforme $x_1, x_2, \ldots, x_n$ asumen diferentes valores en U. Por lo que los predicados en lógica de predicados son las variables de lógica proposicional, y la *atadura* transforma un predicado en una proposición.

Por ejemplo, en $P(x) =$ "x es abogado" y $Q(y) =$ "y es hombre", podríamos formar un predicado $P(x) \wedge Q(y)$ (donde $P(x)$ y $Q(y)$ son variables de lógica proposicional) y por medio de atadura podríamos convertirlo a una proposición del tipo $P(Juan) \wedge Q(Pedro)$.

2.11.1. Cuantificador universal

El cuantificador universal $\forall$ es utilizado para crear una proposición $\forall x P(x)$, leída como "para todo x, $P(x)$ es verdadero". Esta proposición es verdad si y sólo si $P(a)$ es verdad para cada a en un universo U. Esto es:

$$\begin{aligned} \forall x P(x) &= P(x_1) \wedge P(x_2) \wedge P(x_3) \wedge \cdots \\ &= \bigwedge_{x_i \in U} P(x_i) \end{aligned}$$

2.11.2. Cuantificador existencial

El cuantificador existencial $\exists$ es usado para formar una proposición $\exists x P(x)$, leída como "existe un x tal que $P(x)$ es verdadero" o "para algún x, $P(x)$ es verdadero". Esta proposición es verdad si y sólo si $P(a)$ es verdad para al menos un a en U. Esto es:

$$\begin{aligned} \exists x P(x) &= P(x_1) \vee P(x_2) \vee P(x_3) \vee \cdots \\ &= \bigvee_{x_i \in U} P(x_i) \end{aligned}$$

2.11.3. Escritura de declaraciones

Una sentencia que afirma que todo bajo cierta categoría tiene una propiedad se traduce como:

$$\forall x(\_\_ \rightarrow \_\_)$$

donde el antecedente es una proposición verdadera únicamente si se cumple el criterio de la categoría. Si queremos expresar: "Todas las manzanas son malas" escribiríamos $\forall x(A(x) \rightarrow B(x))$ y para expresar "Todas las manzanas verdes son malas" escribiríamos: $\forall x(A(x) \wedge G(x) \rightarrow B(x))$.

Una sentencia que afirma que algún objeto u objetos bajo cierta categoría tienen una propiedad se traduce como:

$$\exists x(\_\_ \wedge \_\_)$$

por ejemplo, "Algunas manzanas son malas" se escribe como $\exists x(A(x) \wedge B(x))$.

Se debe tener cuidado de no confundir los dos patrones, por ejemplo: $\forall x(A(x) \wedge B(x))$ se traduce como "Todo es una manzana y es malo", que es una aserción

mucho más fuerte. De manera similar, $\exists x(A(x) \rightarrow B(x))$ se traduce como "Hay algo que es malo, si es una manzana".

Ejemplos:

1. Cada entero es un número racional: $\forall x(x \in \mathbb{Z} \rightarrow x \in \mathbb{Q})$
2. No hay un número racional x tal que $x^2 = 2$: $\forall x(x \in \mathbb{Q} \rightarrow x^2 \neq 2) \equiv \neg\exists x(x \in \mathbb{Q} \wedge x^2 = 2)$
3. Para números reales x y y, existe un número real z tal que $z^2 = x^2 + y^2$: $\forall x \forall y\{x \in \mathbb{R} \wedge y \in \mathbb{R} \rightarrow \exists z[\mathbb{R}(z) \wedge z^2 = x^2 + y^2]\}$
4. Cada entero par positivo mayor que 2 es la suma de dos primos: $\forall x\{x \in \mathbb{Z} \wedge (x/2 \in \mathbb{Z}) \wedge (x > 2) \rightarrow \exists y, z(y \in \mathbb{P} \wedge z \in \mathbb{P} \wedge x = y + z)\}$
5. Cada entero puede ser expresado como la suma de cuatro cuadrados: $\forall x\{x \in \mathbb{Z} \rightarrow \exists q, r, s, t(x = q^2 + r^2 + s^2 + t^2)\}$
6. Algunos enteros pueden ser expresados como la suma de tres cuadrados: $\exists x\{x \in \mathbb{Z} \wedge \exists q, r, s(x = q^2 + r^2 + s^2)\}$
7. Un entero $n > 1$ es primo si 1 y n son sus únicos divisores. Primero expresemos la declaración en otras palabras: "Si 1 y n son los únicos divisores de n, $n > 1$ y n es entero entonces n es primo". Que nuevamente puede ser expresado como: "Para todo n, si n es entero, $n > 1$ y no existe un x entero diferente de 1 y n tal que n/x sea entero entonces n es primo". $\forall n\{[n \in \mathbb{Z} \wedge (n > 1) \wedge \neg\exists x(x \in \mathbb{Z} \wedge x \neq 1 \wedge x \neq n \wedge n/x \in \mathbb{Z})] \rightarrow n \in \mathbb{P}\}$

2.11.4. Propiedades de los cuantificadores

Los cuantificadores del mismo tipo pueden ser intercambiados y combinados sin cambiar el valor de verdad de las declaraciones:

$$\forall x \forall y P(x, y) \equiv \forall y \forall x P(x, y) \equiv (\forall x, y)P(x, y)$$
$$\exists x \exists y P(x, y) \equiv \exists y \exists x P(x, y) \equiv (\exists x, y)P(x, y)$$

Pero esto no puede ser hecho con cuantificadores de diferentes tipos.

Ejemplo 1: Para números reales a,b,c es sabido que si $a < b$ y $b < c$ entonces $a \leq c$. Esta propiedad de transitividad está dada por:

$$\forall a \forall b\{MENOR(a, b) \rightarrow \forall c[MENOR(b, c) \rightarrow MENOR(a, c)]\}$$

puesto que el orden de los primeros dos cuantificadores no importa y la variable c no aparece en el predicado $MENOR(a, b)$, la fórmula puede ser reescrita combinando todos los cuantificadores existenciales al frente:

$$(\forall a, b, c)\{MENOR(a, b) \rightarrow [MENOR(b, c) \rightarrow MENOR(a, c)]\}$$

Ejemplo 2: Considere el predicado $P(n, m) : n > m^2$ sobre $\mathbb{N} \times \mathbb{N}$. La proposición

$$\forall m \exists n(n > m^2)$$

es equivalente a

$$\forall m[\exists n(n > m^2)]$$

Consideremos la expresión $\exists n(n > m^2)$, donde n está atada pero m es libre. Esta proposición dice que existe un $n \in \mathbb{N}$ con $n > m^2$. Esto es verdad si escogemos $n = m^2 + 1$, por lo que la proposición $\exists n(n > m^2)$ es verdad y, además, es verdad para cada $m \in \mathbb{N}$. Por consecuencia, la proposición $\forall m \exists n(n > m^2)$ es verdad. Ahora si se intercambian los cuantificadores:

$$\exists n \forall m(n > m^2)$$

que es equivalente a $\exists n[\forall m(n > m^2)]$. Una vez más consideremos la expresión interna $\forall m(n > m^2)$. Esta proposición es falsa para $m = n$ y por lo tanto $\exists n \forall m(n > m^2)$ es falso.

Ya hemos visto las leyes para intercambiar cuantificadores idénticos, sin embargo existen otras leyes:

1. $\neg \forall x P(x) \equiv \exists x \neg P(x)$ Decir "no todas las x son P" es equivalente a decir "existe un x que no es P".
2. $\neg \exists x P(x) \equiv \forall x \neg P(x)$ Decir "no existe una x que sea P" es equivalente a decir "toda x no es P".
3. $[\forall x P(x)] \wedge [\forall x Q(x)] \equiv \forall x[P(x) \wedge Q(x)]$ Decir "todo x es P y todo x es Q" equivale a decir "todo x es P y Q".
4. $[\forall x P(x)] \vee [\forall x Q(x)] \Rightarrow \forall x[P(x) \vee Q(x)]$ Decir "todo x es P o todo x es Q" implica "todo x es P o Q". Esta ley no puede aplicarse en el otro sentido.
5. $[\exists x P(x)] \vee [\exists x Q(x)] \equiv \exists x[P(x) \vee Q(x)]$ Decir "existe un x que es P o existe un x que es Q" equivale a decir "existe un x que es P o Q".
6. $\exists x[P(x) \wedge Q(x)] \Rightarrow [\exists x P(x)] \wedge [\exists x Q(x)]$ Decir "existe un x que es P y Q" implica "existe un x que es P y existe un x que es Q".
7. En un predicado de dos lugares $\exists x \forall y P(x, y) \rightarrow \forall y \exists x P(x, y)$ es verdad, sin embargo $\forall y \exists x P(x, y) \rightarrow \exists x \forall y P(x, y)$ no es verdad en lo general.

2.11.5. Instanciación e Interpretación

Una fórmula bien formada se dice *cerrada* si todas las variables en la fórmula están cuantificadas, en caso contrario se dice que está *abierta*. Un predicado con argumentos constantes es una proposición, también llamada formula *atómica cerrada*. Una fórmula atómica cerrada es un *hecho* si es verdad.

Existen cuatro reglas fundamentales en un esquema de inferencia de fórmulas cuantificadas:

1. Generalización Universal: Si escogemos un elemento arbitrario c del dominio U y probamos $P(c)$, entonces podemos inferir $\forall x P(x)$. Por ejemplo $(x + 2)^2 = x^2 + 4x + 4$ es verdad para cualquier real. No hay restricciones en la forma de escoger x, y por lo tanto podemos inferir que $\forall x[(x + 2)^2 = x^2 + 4x + 4]$.

2. Generalización Existencial: Si podemos probar que $P(c)$ es verdad para algún c en el universo U, entonces podemos inferir $\exists x P(x)$.

3. Especificación Universal: De la declaración $\forall x P(x)$ podemos inferir $P(c)$ para cada c en el universo. Por ejemplo, de "cada entero es un racional" podemos inferir "2 es un número racional".

4. Especificación Existencial: De la declaración $\exists x P(x)$ podemos inferir que es posible escoger c en el universo tal que $P(c)$ es verdadero.

Ejemplo 1. Considere las siguientes declaraciones:

1. Todas las personas inteligentes son nobles.

2. Todos son inteligentes o tontos.

3. Algunas personas no son tontas.

4. Por lo tanto, algunas personas son nobles.

definamos los predicados como:

- $I(x) : x$ es inteligente.
- $N(x) : x$ es noble.
- $T(x) : x$ es tonto.

En notación formal se desea probar: $\forall x[I(x) \rightarrow N(x)], \forall x[I(x) \vee T(x)], \exists x[\neg T(x)] \Rightarrow \exists x[N(x)]$, asumiendo que el universo de discurso es de personas.

1.	$\forall x[I(x) \rightarrow N(x)]$	Premisa
2.	$\forall x[I(x) \vee T(x)]$	Premisa
3.	$\exists x[\neg T(x)]$	Premisa
4.	$\neg T(c)$	Especificación existencial 3
5.	$I(c) \vee T(c)$	Especificación universal 2
6.	$I(c)$	Simplificación disyuntiva 4,5
7.	$I(c) \rightarrow N(c)$	Especificación universal 1
8.	$N(c)$	Modus Ponens 6,7
9.	$\exists x[N(x)]$	Generalización existencial

Ejemplo 2. Considere los siguientes hechos y reglas:

1. Jorge y Karla son miembros del Club ABC.

2. Jorge está casado con María.

3. Juan es un hermano de María y está casado con Karla.

4. Jorge y Juan se reúnen en casa de Jorge.

5. El cónyuge de cada persona en el club ABC es también miembro del club.

6. Las personas casadas viven juntas.

De esto queremos determinar la verdad de las siguientes declaraciones:

1. Jorge, Karla, Juan y María son miembros del club ABC.
2. Juan visita la casa de su hermana.

definamos los predicados como:

- $H(x)$: x es miembro del club ABC.
- $M(x, y)$: x está casado con y.
- $L(x, y)$: x y y viven juntos.
- $B(x, y)$: x es hermano de y.
- $V(x, y)$: x visita la casa de y.

Los hechos y reglas están representados con predicados de la siguiente forma:

1. $H(Jorge) \wedge H(Karla)$
2. $M(Jorge, Maria)$
3. $B(Juan, Maria) \wedge M(Juan, Karla)$
4. $V(Juan, Jorge)$
5. $(\forall x, y)[H(x) \wedge M(x, y) \rightarrow H(y)]$, $\forall x, y[H(y) \wedge M(x, y) \rightarrow H(x)]$
6. $(\forall x, y)[M(x, y) \rightarrow L(x, y)]$

Para resolver “Jorge, Karla, Juan y María son miembros del club ABC.”:

1.	$M(Jorge, Maria)$	Hecho
2.	$H(Jorge)$	Hecho
3.	$(\forall x, y)[H(x) \wedge M(x, y) \rightarrow H(y)]$	Regla
4.	$H(Jorge) \wedge M(Jorge, Maria) \rightarrow H(Maria)$	S.3 $x = Jorge, y = Maria$
5.	$H(Maria)$	Modus Ponens 1,2 y 4
6.	$H(Karla)$	Hecho
7.	$M(Juan, Karla)$	Hecho
8.	$(\forall x, y)[H(y) \wedge M(x, y) \rightarrow H(x)]$	Regla
9.	$H(Karla) \wedge M(Juan, Karla) \rightarrow H(Juan)$	S.8 $x = Juan, y = Karla$
10.	$H(Juan)$	Modus Ponens 6,7 y 9
11.	$H(Jorge) \wedge H(Karla) \wedge H(Juan) \wedge H(Maria)$	Conjunción 2,6,10,5

Para resolver “Juan visita la casa de su hermana.”

1.	$(\forall x, y, z)[V(x, y) \wedge L(y, z) \rightarrow V(x, z)]$	Regla
2.	$M(Jorge, Maria)$	Hecho
3.	$V(Juan, Jorge)$	Hecho
4.	$B(Juan, Maria)$	Hecho
5.	$(\forall x, y)[M(x, y) \rightarrow L(x, y)]$	Regla
6.	$M(Jorge, Maria) \rightarrow L(Jorge, Maria)$	S.5 $x = Jorge, y = Maria$
7.	$L(Jorge, Maria)$	Modus Ponens 2,6
8.	$V(Juan, Jorge) \wedge L(Jorge, Maria) \rightarrow V(Juan, Maria)$	S.1 $x = Juan$, $y = Jorge, z = Maria$
9.	$V(Juan, Maria)$	Modus Ponens 3,7 y 8
10.	$B(Juan, Maria) \wedge V(Juan, Maria)$	Conjunción 4,9

2.11.6. Principio de resolución y procesamiento de interrogantes

Un algoritmo consiste esencialmente de dos partes, la *lógica* y el *control*. Una especificación del problema junto con la descripción de lo que se requiere resolver el la parte lógica. Una descripción paso por paso de como resolver el problema es la parte de control.

En un lenguaje procedural la lógica y el control están mezclados en el sentido que el control dicta la lógica del programa. En ejemplos que hemos visto los hechos y las reglas son declaradas y éstas de ninguna manera determinan la secuencia de acciones (control) en una inferencia. Esta es la principal observación que lleva al desarrollo de lenguajes de programación lógicos. PROLOG (Programming in Logic) es un lenguaje de este tipo.

Cláusulas de Horn

En lógica una cláusula de Horn es una fórmula bien formada de la forma $R(P_1 \wedge P_2 \wedge \cdots \wedge P_k \rightarrow Q)$ donde Q y P_i son fórmulas atómicas y R cuantifica universalmente todas las variables de P_i. PROLOG está basado en el subconjunto de cláusulas Horn de la lógica de primer orden. Por conveniencia se adoptan dos convenciones de PROLOG cuando se escriben cláusulas Horn:

1. La conjunción es reemplazada por ‘,’.

2. Los cuantificadores no están explícitos.

Por ejemplo la regla $(\forall x, y, z)[P(x, z) \wedge P(z, y) \rightarrow Q(x, y)]$ puede ser escrita simplemente como:

$$P(x, z), P(z, y) \rightarrow Q(x, y)$$

El predicado a la derecha de $\rightarrow$ es llamado la *cabeza* de la cláusula, y los predicados a la izquierda de $\rightarrow$ constituyen el *cuerpo* de la cláusula. Una regla debe tener un cuerpo no vacío. Puesto que un hecho describe información explícita, sólo tiene cabeza; y el símbolo $\rightarrow$ se omite.

Un programa lógico es un conjunto de cláusulas Horn que describen hechos (datos almacenados), reglas (para la manipulación de los datos) y una meta (interrogante). La cláusula de meta no tiene cuerpo y tiene una o más variables en la cabeza. Por ejemplo $MADRE(x, Maria)$ sería la interrogante "encuentra la madre de Maria".

En lógica el orden en que se escriben los hechos y reglas es irrelevante, en PROLOG el orden tiene significado.

Principio de resolución

Sean r_1 y r_2 dos reglas tales que el predicado P_i que aparece en el cuerpo de r_1 es la cabeza de r_2:

$$\begin{array}{ll} r_1: & P_1(x_1), P_2(x_2), \ldots, P_i(x_i), \ldots, P_n(x_n) \rightarrow Q(y) \\ r_2: & R_1(z_1), R_2(z_2), \ldots, R_m(z_m) \rightarrow P_i(x_i^1) \end{array}$$

donde $x_1, x_2, \ldots, x_n, y, z_1, z_2, \ldots, z_m, x_i^1$ son vectores de constantes y variables. Si los vectores x_i y x_i^1 pueden hacerse idénticos (unificados) substituyendo constantes o variables en x_i y x_i^1 entonces la regla r_2^1 puede ser derivada de r_1 y r_2:

$$\begin{array}{ll} r_2^1: & P_1(x_1'), P_2(x_2'), \ldots, P_{i-1}(x_{i-1}'), R_1(z_1'), R_2(z_2'), \ldots, R_m(z_m'), \\ & P_{i+1}(x_{i+1}'), \ldots, P_n(x_n') \rightarrow Q(y') \end{array}$$

Aquí los vectores x_j',z_j',y' son el resultado de la substitución de x_j, z_j, y.

Ejemplo: La regla para definir antepasados de individuos, "si x es el padre de y entonces x es el antepasado de y; si x es el padre de z y z es el antepasado de y, entonces x es el antepasado de y. Las reglas son:

$$\begin{array}{ll} r_1: & PADRE(x, y) \rightarrow ANTEPASADO(x, y) \\ r_2: & PADRE(x, z), ANTEPASADO(z, y) \rightarrow ANTEPASADO(x, y) \end{array}$$

Si cambiamos las variables en r_1 y renombramos x como z tenemos la regla:

$$r_1^1: \quad PADRE(z, y) \rightarrow ANTEPASADO(z, y)$$

Ahora la cabeza de r_1^1 concuerda con un predicado en el cuerpo de r_2. Por el principio de resolución tenemos la nueva regla:

$$r_2^1: \quad PADRE(x, z), PADRE(z, y) \rightarrow ANTEPASADO(x, y)$$

2.12. Errores en las demostraciones

Capítulo 3

Inducción Matemática

3.1. Inducción simple

Supongamos que $S(k)$ es una declaración válida para algún entero $k \geq n_0$ y queremos probar que $S(n)$ es verdadero para todos los enteros $n \geq n_0$. El principio de inducción simple nos dice que si (1) $S(n_0)$ es verdad y (2) $S(k) \rightarrow S(k+1)$, entonces $S(n)$ es verdad para todos los enteros $n \geq n_0$. Entonces para probar $S(n)$ para todos los enteros $n \geq n_0$, necesitamos probar únicamente:

1. Que $S(n_0)$ es verdad (caso base).
2. Que si $S(k)$ es verdad (hipótesis de inducción), entonces $S(k+1)$ es también verdad (paso de inducción).

Ejemplo 1

Dejemos que $S(n)$ denote la aserción

$$1 + 3 + 5 + \cdots + (2n - 1) = n^2$$

para todo $n \geq 1$. Ahora, $S(1)$ es $1 = 1^2$, que es verdad. Podemos verificar algunos más:

$$\begin{aligned} S(2) \text{ es } 1 + 3 &= 2^2 \\ S(3) \text{ es } 1 + 3 + 5 &= 3^2 \end{aligned}$$

que también se cumplen. Ahora asumamos que para algún $k \geq 1$, $S(k)$ es verdad, esto es, $S(k) : 1 + 3 + 5 + \cdots + (2k - 1) = k^2$. Considere:

$$1 + 3 + 5 + \cdots + (2k - 1) + (2k + 1)$$

y reagrupemos los términos de la siguiente forma $[1 + 3 + 5 + \cdots + (2k - 1)] + (2k + 1)$, y como $S(k)$ es verdad. reemplazamos la expresión entre corchetes por

k^2:

$$\begin{aligned} &= k^2 + (2k+1) \\ &= (k+1)^2 \end{aligned}$$

por lo que $1+3+5+\cdots+(2k-1)+(2k+1) = (k+1)^2$ y por lo tanto $S(k+1)$ es verdad. Entonces por inducción simple, $S(n)$ es verdad para todo $n \geq 1$.

Ejemplo 2

Probar que $1+2+3+\cdots+n = \frac{n(n+1)}{2}$ se cumple para todo $n \geq 1$. Denotemos por $S(n)$ esta aserción y probemos el caso base:

$$S(1) : 1 = \frac{1(1+1)}{2}$$

que es verdad. Ahora consideremos $1+2+3+\cdots+n+(n+1)$, reagrupando términos tenemos $[1+2+3+\cdots+n]+(n+1)$, como $S(n)$ es verdad, entonces reemplazamos la expresión entre corchetes por $\frac{n(n+1)}{2}$:

$$\begin{aligned} &= \frac{n(n+1)}{2} + (n+1) \\ &= \frac{n(n+1)}{2} + \frac{2(n+1)}{2} \\ &= \frac{n(n+1)+2(n+1)}{2} \\ &= \frac{(n+2)(n+1)}{2} \\ &= \frac{(n+1)(n+2)}{2} \end{aligned}$$

por lo que $S(n+1)$ es verdad y se deduce que $S(n)$ es verdad para todo $n \geq 1$.

Ejemplo 3

El número definido como $H_n = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n},\ n \geq 1$ es llamado número armónico. Pruebe que para todo $n \geq 1$,

$$\sum_{k=1}^{n} H_k = (n+1)H_n - n$$

Dejemos que $S(n)$ denote la declaración $H_1 + H_2 + \cdots + H_n = (n+1)H_n - n$. El caso base $S(1)$ es $H_1 = 2H_1 - 1$, puesto que H_1 es 1, $S(1)$ es verdad. Ahora consideremos $H_1 + H_2 + \cdots + H_n + H_{n+1}$, reagrupando términos tenemos $[H_1 + H_2 + \cdots + H_n] + H_{n+1}$ y puesto que $S(n)$ es verdad, reemplazamos la expresión

entre corchetes por $(n+1)H_n - n$:

$$\begin{aligned} &= (n+1)H_n - n + H_{n+1} \\ &= (n+1)\left(H_{n+1} - \frac{1}{n+1}\right) - n + H_{n+1} \\ &= (n+1)H_{n+1} - \frac{n+1}{n+1} - n + H_{n+1} \\ &= (n+2)H_{n+1} - 1 - n \\ &= (n+2)H_{n+1} - (n+1) \end{aligned}$$

por lo que $S(n+1)$ es verdad siempre que $S(n)$ es verdad. Entonces por inducción simple, $S(n)$ es verdad para todo $n \geq 1$.

Ejemplo 4

Pruebe que para $n \geq 1$,

$$\frac{1}{3} + \frac{1\cdot 2}{3\cdot 4} + \frac{1\cdot 2\cdot 3}{3\cdot 4\cdot 5} + \cdots + \frac{n!}{\frac{(n+2)!}{2}} = \frac{n}{n+2}$$

Sea $S(n)$ una declaración que denote dicha aserción. $S(1)$ es $\frac{1}{3} = \frac{1}{1+2}$. Consideremos:

$$\frac{1}{3} + \frac{1\cdot 2}{3\cdot 4} + \frac{1\cdot 2\cdot 3}{3\cdot 4\cdot 5} + \cdots + \frac{n!}{\frac{(n+2)!}{2}} + \frac{(n+1)!}{\frac{(n+3)!}{2}}$$

puesto que $S(n)$ es verdad, entonces:

$$\begin{aligned} \frac{1}{3} + \frac{1\cdot 2}{3\cdot 4} + \cdots + \frac{n!}{\frac{(n+2)!}{2}} + \frac{(n+1)!}{\frac{(n+3)!}{2}} &= \frac{n}{n+2} + \frac{2(n+1)!}{(n+3)!} \\ &= \frac{n}{n+2} + \frac{2(n+1)!}{(n+1)!(n+2)(n+3)} \\ &= \frac{n}{n+2} + \frac{2}{(n+2)(n+3)} \\ &= \frac{n(n+3)+2}{(n+2)(n+3)} \\ &= \frac{n^2+3n+2}{(n+2)(n+3)} \\ &= \frac{(n+2)(n+1)}{(n+2)(n+3)} \\ &= \frac{n+1}{n+3} \end{aligned}$$

por lo que $S(n) \Rightarrow S(n+1)$. Y por inducción simple $S(n)$ es verdad para todo $n \geq 1$.

Ejemplo 5

Pruebe que para todo $k \geq 4$, $2^k \geq k^2$.

Primero el caso base $2^4 \geq 4^2$ es verdad. Ahora queremos probar que $2^{k+1} \geq (k+1)^2$, es claro que:

$$\begin{array}{rcll} (k+1)^2 & = & k^2+2k+1 & \\ k^2+2k+1 & \leq & k^2+2k+k & \text{Puesto que } k \geq 4 > 3 \\ k^2+2k+1 \leq k^2+3k & \leq & k^2+kk & \text{Puesto que } k \geq 4 > 3 \\ (k+1)^2 & \leq & 2k^2 & \\ 2k^2 & \geq & (k+1)^2 & \end{array}$$

Ahora, puesto que $2^k \geq k^2$ es verdad, entonces $(2)2^k = 2^{k+1} \geq 2k^2$. Entonces por inducción simple, $2^k \geq k^2$ es válido para todo $k \geq 4$.

3.2. Inducción completa

Sea $S(n)$ una declaración sobre cualquier entero $n \geq n_0$. Si $S(n_0)$ es verdad y si para cada $n_0 \leq m < n$, $S(m)$ es verdad, entonces $S(n)$ es verdad para todos los enteros $n \geq n_0$. Esta aserción es mucho más fuerte que la inducción simple. En algunos casos la prueba no puede ser efectuada por inducción simple, por lo que esta prueba es utilizada en algunos casos. Ambas pruebas son equivalentes.

Ejemplo 1

Cada número natural $n > 1$ puede factorizarse a números primos.

Sea $S(n)$ la declaración "n es el producto de números primos". Primero $S(2)$ es verdad, pues 2 es primo. Ahora asuma que $S(m)$ es verdad para todo $2 \leq m < n$. Si n es primo entonces $S(n)$ es verdad. Si n no es primo, entonces $n = ab$, donde $1 < a, b < n$. Entonces por la hipótesis de inducción $S(a)$ y $S(b)$ son verdad; esto es, a y b son productos de números primos. Lo que nos lleva decir que n es producto de primos, entonces $S(n)$ es verdad para todo $n \geq 2$.

Ejemplo 2

Para los números de Fibonacci definidos como:

$$\begin{array}{rcl} f_0 & = & 0 \\ f_1 & = & 1 \\ f_{n+1} & = & f_n + f_{n-1}, \quad n \geq 1 \end{array}$$

pruebe que si Φ es el número $\frac{1+\sqrt{5}}{2}$, entonces para todo $n \geq 1, \Phi^{n-2} \leq f_n \leq \Phi^{n-1}$

Primero probemos $\Phi^{n-2} \leq f_n$, y dejemos que $S_1(n)$ denote dicha aserción. $S_1(1)$ es $\Phi^{-1} \leq f_1 = 1$, que es verdad, y $S_1(2)$ es $\Phi^0 \leq f_2 = 1$, que también es verdad. Ahora asumimos que $S_1(m)$ es verdad para todo $m, 1 \leq m \leq n$. Demostraremos que $S_1(n+1)$ es verdad, esto es $\Phi^{n-1} \leq f_{n+1}$.

Por hipótesis de inducción $S_1(n)$ y $S_1(n-1)$ son verdad. Entonces $\Phi^{n-2} \leq f_n$ y $\Phi^{n-3} \leq f_{n-1}$. Por lo que:

$$\begin{aligned} f_{n+1} &= f_n + f_{n_1} \\ &\geq \Phi^{n-2} + \Phi^{n-3} \\ &= \Phi^{n-3}(\Phi + 1) \\ &= \Phi^{n-3}(\Phi^2) = \Phi^{n-1} \end{aligned}$$

por lo que $S_1(n+1)$ es verdad. Y por inducción completa, $S_1(n)$ es verdad para todo $n \geq 1$.

Ahora probemos $f_n \leq \Phi^{n-1}$ y dejemos que $S_2(n)$ denote dicha aserción. Primero $S_2(1)$ es $1 \leq \Phi^0$, que es verdad, y $S_2(2)$ es $1 \leq \Phi^1$, que también es verdad. Asumimos que $S_2(m)$ es verdad para todo $m, 1 \leq m \leq n$ y demostramos que $S_2(n+1)$ es verdad, esto es $f_{n+1} \leq \Phi^n$. Por hipótesis de inducción $S_2(n)$ y $S_2(n-1)$ son verdad. Entonces $f_{n-1} \leq \Phi^{n-2}$ y $f_n \leq \Phi^{n-1}$. Por lo que:

$$\begin{aligned} f_{n+1} &= f_{n-1} + f_n \\ &\leq \Phi^{n-2} + \Phi^{n-1} \\ &= \Phi^{n-2}(1 + \Phi) \\ &= \Phi^{n-2}(\Phi^2) = \Phi^n \end{aligned}$$

por lo que $S_2(n+1)$ es verdad. Y por inducción completa, $S_2(n)$ es verdad para todo $n \geq 1$.

Capítulo 4

Conjuntos

4.1. Definición y operaciones

Un conjunto es una colección finita o infinita de objetos en la que el orden no tiene importancia, y la multiplicidad también es ignorada. Miembros de un conjunto son comúnmente denominados elementos y la notación $a \in A$ es usada para denotar "a es un elemento del conjunto A". Es común utilizar letras mayúsculas para denotar conjuntos y letras minúsculas para denotar elementos.

Un conjunto debe ser descrito sin ambigüedades; esto es, dado un conjunto y un objeto, debe ser posible decidir si el objeto pertenece o no al conjunto. Un conjunto puede ser descrito enumerando sus miembros:

$$S = \{2, 3, 5, 7, 11, 13, 17, 19\}$$

o describiendo la propiedad que lo caracteriza:

$$S = \{n \mid n \text{ es un número primo menor que } 20\}$$

Dos conjuntos A y B son iguales, $A = B$, si contienen los mismos elementos. Por ejemplo, $\{2, 3, 5, 7\} = \{3, 5, 2, 7, 2\}$. El orden en que se listan los elementos es irrelevante, y un elemento puede estar listado más de una vez. Si A y B no son iguales escribimos $A \neq B$.

Un conjunto que no tiene elementos es un conjunto único llamado *conjunto vacío* o *conjunto nulo* y es denotado con el símbolo ϕ.

Subconjuntos especiales de $\mathbb{R}$ llamados intervalos son definidos como:

- Intervalo cerrado: $[a, b] = \{x \mid x \in \mathbb{R}, a \leq x \leq b\}$.
- Intervalo abierto: $(a, b) = \{x \mid x \in \mathbb{R}, a < x < b\}$
- Intervalos semi-cerrados (o semi-abierto):
  - $[a, b) = \{x \mid x \in \mathbb{R}, a \leq x < b\}$
  - $(a, b] = \{x \mid x \in \mathbb{R}, a < x \leq b\}$

4.1.1. Subconjuntos

Si A y B son conjuntos y si cada elemento de A es un elemento de B, entonces decimos que A es un subconjunto de B (o B contiene a A), y se denota por $A \subseteq B$.

Si $A \subseteq B$ y $A \neq B$ entonces A es un subconjunto propio, y escribimos $A \subset B$. Si $A \subseteq B$ y $B \subseteq A$, entonces $A = B$. Si $A \subseteq B$ y $B \subseteq C$ entonces $A \subseteq C$.

Del conocimiento de los números, tenemos $\mathbb{N} \subset \mathbb{Z}$,$\mathbb{Z} \subset \mathbb{Q}$, $\mathbb{Q} \subset \mathbb{R}$.

4.1.2. Definición Recursiva de Conjuntos

Muchos conjuntos son de carácter generativo. Esto es, contienen elementos fundamentales que son conocidos y reglas que permiten formar nuevos elementos basándose en los elementos que ya están en el conjunto. Por ejemplo, el conjunto $\mathbb{N}_0$ (todos los enteros no negativos) puede ser definido de la siguiente manera:

$$\begin{array}{rl} \text{Objetos fundamentales:} & 0, 1 \in \mathbb{N}_0 \\ \text{Regla de generación:} & a, b \in \mathbb{N}_0 \rightarrow a + b \in \mathbb{N}_0 \end{array}$$

Entonces el conjunto $\mathbb{N}_0$ puede ser visto como un conjunto que crece a partir de los elementos $0, 1$ hacia la colección de los enteros no negativos por medio de la inserción sucesiva de los números $2, 3, 4, 5, \ldots$ en $\mathbb{N}_0$ generados por la regla.

4.1.3. Conjunto potencia

El conjunto de todos los subconjuntos de un conjunto S es llamado *conjunto potencia*, y se denota por $\mathcal{P}(S)$.

- $\mathcal{P}(\phi) = \{\phi\}$.
- $\mathcal{P}(\{a\}) = \{\phi, \{a\}\}$.
- $\mathcal{P}(\{a, b\}) = \{\phi, \{a\}, \{b\}, \{a, b\}\}$

4.1.4. Algebra de Conjuntos

Unión

La *unión* de dos conjuntos A y B, denotada $A \cup B$, es el conjunto de todos los elementos que pertenecen a A o B o a ambos.

$$A \cup B = \{x \mid x \in A \vee x \in B\}$$

La unión satisface:

$$\begin{array}{rcl} A \cup \phi & = & A \\ A \cup B & = & B \cup A \\ A \cup (B \cup C) & = & (A \cup B) \cup C \\ A \cup A & = & A \\ A \subseteq B & \leftrightarrow & A \cup B = B \end{array}$$

Ejemplo: Si $A = \{x \mid x \in \mathbb{N}, x \text{ par}\}$ y $B = \{y \mid y \in \mathbb{N}, y \text{ múltiplo de } 3\}$. Entonces, $A \cup B = \{x \mid x \in \mathbb{N}, x \text{ par o múltiplo de } 3\}$.

Intersección

La *intersección* de conjuntos A y B, denotada $A \cap B$, es el conjunto de todos los elementos que pertenecen a A y B.

$$A \cap B = \{x \mid x \in A \wedge x \in B\}$$

La intersección satisface:

$$\begin{aligned} A \cap \phi &= \phi \\ A \cap B &= B \cap A \\ A \cap (B \cap C) &= (A \cap B) \cap C \\ A \cap A &= A \\ A \subseteq B &\leftrightarrow A \cap B = A \end{aligned}$$

Ejemplo 1: La intersección de los intervalos $[-\infty, 4]$ y $[-3, 10]$ es $[-3, 4]$.

Ejemplo 2: La intersección de los conjuntos $\{x \mid x \in \mathbb{R}, x^2 \geq 4\}$ y $\{x \mid x \in \mathbb{R}, x^2 - 3x = 0\}$ es $\{3\}$.

Dos identidades importantes que involucran uniones e intersecciones son las leyes distributivas:

$$\begin{aligned} A \cap (B \cup C) &= (A \cap B) \cup (A \cap C) \\ A \cup (B \cap C) &= (A \cup B) \cap (A \cup C) \end{aligned}$$

Complemento

El *complemento relativo* de un conjunto B con respecto a A denotado $A - B$ es el conjunto de los elementos que pertenecen a A pero no pertenecen a B.

$$A - B = \{x \mid x \in A, x \notin B\}$$

Cuando se asume un conjunto universal U, y un conjunto A, $A \subseteq U$, entonces el *complemento absoluto* o más comúnmente *complemento* de A es $U - A$, y es denotado $\overline{A}$.

El complemento satisface:

$$\begin{aligned} \overline{A} &\neq A \\ A \cup \overline{A} &= U \\ A \cap \overline{A} &= \phi \\ \overline{U} &= \phi \\ \overline{\phi} &= U \\ (\overline{A \cup B}) &= \overline{A} \cap \overline{B} \\ (\overline{A \cap B}) &= \overline{A} \cup \overline{B} \end{aligned}$$

4.2. Conjuntos contables e incontables

Es de importancia el tamaño de un conjunto y el tamaño de los elementos en un conjunto. Cuando se ignoran las características de los elementos de un conjunto y se mira a éste de manera abstracta, la única propiedad que gobierna es el número de elementos.

De manera abstracta uno puede asumir que los conjuntos con el mismo número de elementos son equivalentes, por ejemplo $A = \{1, 2, 3\}$ y $B = \{x, y, z\}$ son equivalentes, pero A no es equivalente a $C = \{a, b\}$.

La propiedad común de todos los conjuntos equivalentes a A es su número de elementos o *número cardinal* (*cardinalidad* o *tamaño*), denotado por $|A|$.

Para establecer si un conjunto A es finito, debemos demostrar que todos los elementos de A comenzando por un elemento arbitrario pueden ser etiquetados como primer elemento, segundo elemento,. . .,n-ésimo elemento para algún entero positivo n. Cuando esto puede ser efectuado decimos que A es finito y $|A| = n$.

Si este proceso de etiquetado no produce algún n pero el etiquetado con el conjunto de los números naturales es posible, el conjunto es *infinito* y decimos que A es contable. Si hay elementos de A que ningún proceso de etiquetado puede alcanzar el conjunto es *infinito* y decimos que A es *incontable*.

4.2.1. Producto

Consideremos los siguientes pares para los conjuntos $A = \{1, 2, 3\}$, $B = \{x, y, z\}$:

$$\begin{array}{rccc} R: & 1 & 2 & 3 \\ & \updownarrow & \updownarrow & \updownarrow \\ & x & y & z \end{array}$$

Una forma alternativa de escribir estos pares es:

$$R = \{\langle 1, x\rangle, \langle 2, y\rangle, \langle 3, z\rangle\}$$

Podemos observar que:

1. En cada par $\langle r, s\rangle$, r es un elemento de A y s es un elemento de B.
2. Los pares están ordenados en el sentido de que un elemento de A aparece primero y después aparece un elemento de B.
3. Muchos emparejamientos de A y B pueden existir.

Este concepto de emparejamiento se formaliza formando conjuntos producto.

Sean A y B dos conjuntos. El *conjunto producto cartesiano* $A \times B$ es definido como:

$$A \times B = \{\langle a, b\rangle \mid a \in A, b \in B\}$$

Los elementos de $A \times B$ son llamados *pares ordenados*. En general $A \times B \neq B \times A$. Podemos generalizar este concepto a n conjuntos:

$$A_1 \times A_2 \times \cdots \times A_n = \{\langle a_1, a_2, \ldots, a_n\rangle \mid a_1 \in A_1, a_2 \in A_2, \ldots, a_n \in A_n\}$$

Llamamos a $\langle a_1, a_2, \ldots, a_n\rangle$ una tupla-n ordenada.

Ejemplo.

$$\begin{aligned} A &= \{a \mid a \in \mathbb{N}, 1 \leq a \leq 5\} \\ B &= \{b \mid b \in \mathbb{Z}, 0 \leq b \leq 2\} \\ A \times B &= \{x \mid x = \langle a, b\rangle, a \in A, b \in B\} \\ &= \{\langle 1,0\rangle, \langle 1,1\rangle, \langle 1,2\rangle, \langle 2,0\rangle, \langle 2,1\rangle, \langle 2,2\rangle, \langle 3,0\rangle, \langle 3,1\rangle, \\ &\quad \langle 3,2\rangle, \langle 4,0\rangle, \langle 4,1\rangle, \langle 4,2\rangle, \langle 5,0\rangle, \langle 5,1\rangle, \langle 5,2\rangle\} \end{aligned}$$

El número de elementos del conjunto producto cartesiano, $|A \times B| = |B \times A| = |A||B|$. En general, si $A_1, A_2, \ldots, A_n$ son finitos entonces:

$$|A_1 \times A_2 \times \cdots \times A_n| = \prod_{i=1}^{n} |A_i|$$

En particular, si $A = A_1 = A_2 = \cdots = A_n$, entonces $A_1 \times A_2 \times \cdots \times A_n$ será denotado A^n y este conjunto consiste de todas las tuplas-n ordenadas $\langle a_1, \ldots, a_n\rangle$ con $a_i \in A$.

Si $A \times B$ y $C \times D$:

$$\begin{aligned} A \times (B \cup C) &= (A \times B) \cup (A \times C) \\ A \times (B \cap C) &= (A \times B) \cap (A \times C) \\ (A \cup B) \times C &= (A \times C) \cup (B \times C) \\ (A \cap B) \times C &= (A \times C) \cap (B \times C) \\ (A \cap B) \times (C \cap D) &= (A \times C) \cap (B \times D) \\ (A - B) \times C &= (A \times C) - (B \times C) \\ A \times B = \phi &\leftrightarrow A = \phi \vee B = \phi \\ A \subset C, B \subset D, A \times B \neq \phi &\rightarrow A \times B \subset C \times D \\ A \times B \neq \phi, A \times B \subset C \times D &\rightarrow A \subset C, B \subset D \end{aligned}$$

Capítulo 5

Relaciones

Una relación es un subconjunto de un conjunto producto. Una *relación n-aría* es un subconjunto de un conjunto producto de n conjuntos. Si $n = 2$ la relación es llamada *relación binaria*.

Si R es un subconjunto de $A \times B$, decimos que R es una relación de A hacia B. Para cualquier $\langle a, b\rangle \in R$ también se puede escribir aRb.

El conjunto $C = \{a \in A \mid \langle a, b\rangle \in R, b \in B\}$ es llamado dominio de R, y el conjunto $D = \{b \in B \mid \langle a, b\rangle \in R, a \in A\}$ es llamado el rango de R. Por consecuencia $C \subseteq A$ y $D \subseteq B$.

Ejemplo 1: Sean $A = \{0, 1, 2\}$ y $B = \{a, b\}$. Entonces, $\{\langle 0, a\rangle, \langle 0, b\rangle, \langle 1, a\rangle, \langle 2, b\rangle\}$ es una relación de A hacia B.

Las relaciones se pueden representar gráficamente utilizando flechas para indicar los pares ordenados. Otra forma de representarlas es usar una tablas.

Ejemplo 2: La relación aritmética $<$ en los enteros es un subconjunto de $\mathbb{Z} \times \mathbb{Z}$, que consiste de los pares $\langle a, b\rangle$:

$$<= \{\langle a, b\rangle \mid \langle a, b\rangle \in \mathbb{Z} \times \mathbb{Z}, a \text{ menor que } b\}$$

por lo que usamos $a < b$ en lugar de $\langle a, b\rangle \in <$. Otras relaciones en enteros como $>, \leq$ pueden ser definidas de modo similar, así como las comparaciones aritméticas con números reales.

Ejemplo 3: Sea $A = \{1, 2, 3, 4, 5, 9\}$, la relación mayor que (a es mayor que b) definida en A es:

$$\begin{aligned} M \quad = \quad & \{\langle 2, 1\rangle, \langle 3, 1\rangle, \langle 4, 1\rangle, \langle 5, 1\rangle, \langle 9, 1\rangle, \langle 3, 2\rangle, \langle 4, 2\rangle, \langle 5, 2\rangle, \langle 9, 2\rangle, \\ & \langle 4, 3\rangle, \langle 5, 3\rangle, \langle 9, 3\rangle, \langle 5, 4\rangle, \langle 9, 4\rangle\langle 9, 5\rangle\} \end{aligned}$$

De aquí podemos observar que $Dominio(M) = \{2, 3, 4, 5, 9\}$ y $Rango(M) = \{1, 2, 3, 4, 5\}$.

5.1. Relación Inversa

Para cualquier relación R de A a B podemos definir la relación inversa, denotada como R^{-1}, de B a A. Esta relación inversa meramente consiste de los pares ordenados de R al revés:

$$R^{-1} = \{\langle b, a\rangle \mid \langle a, b\rangle \in R\}$$

Ejemplo: Sea $A = \{1, 3, 5\}$, $B = \{a, b\}$, $R = \{\langle 1, b\rangle, \langle 3, a\rangle, \langle 5, b\rangle, \langle 3, b\rangle\}$, entonces la relación inversa es:

$$R^{-1} = \{\langle b, 1\rangle, \langle a, 3\rangle, \langle b, 5\rangle, \langle b, 3\rangle\}$$

5.2. Relaciones Reflexivas

Sea R una relación binaria definida en un conjunto A. Decimos que R es una relación reflexiva si aRa para cada $a \in A$.

Ejemplo: Si $A = \{a, b, c, d\}$. Una relación $R \subseteq A \times A$ es reflexiva si contiene $\{\langle a, a\rangle, \langle b, b\rangle, \langle c, c\rangle, \langle d, d\rangle\}$.

5.3. Relaciones Irreflexivas

Sea R una relación binaria definida en un conjunto A. Decimos que R es una relación irreflexiva si $\neg(aRa)$ para todo $a \in A$.

Ejemplo: Si $A = \{a, b, c, d\}$. Una relación $R \subseteq A \times A$ es irreflexiva si no contiene algún subconjunto de $\{\langle a, a\rangle, \langle b, b\rangle, \langle c, c\rangle, \langle d, d\rangle\}$.

5.4. Relaciones Simétricas

Sea R una relación binaria definida en un conjunto A. Decimos que R es una relación simétrica si aRb implica bRa para $a, b \in A$.

Ejemplo: Si $A = \{a, b, c, d\}$. Una relación $R \subseteq A \times A$ definida como:

$$R = \{\langle a, b\rangle, \langle c, a\rangle, \langle b, a\rangle, \langle a, c\rangle\}$$

es simétrica.

5.5. Relaciones Antisimétrica

Sea R una relación binaria definida en un conjunto A. Decimos que R es una relación antisimétrica si aRb y bRa implica $a = b$ para $a, b \in A$.

Ejemplo: Si $A = \{a, b, c, d\}$. Una relación $R \subseteq A \times A$ definida como:

$$R = \{\langle a, a\rangle, \langle c, a\rangle, \langle b, a\rangle, \langle a, d\rangle\}$$

es antisimétrica.

5.6. Relaciones Transitivas

Sea R una relación binaria definida en un conjunto A. Decimos que R es una relación transitiva si aRb y bRc implican aRc para $a, b, c \in A$.

Ejemplo: Si $A = \{a, b, c, d\}$. Una relación $R \subseteq A \times A$ definida como:

$$R = \{\langle a, b\rangle, \langle b, c\rangle, \langle a, c\rangle, \langle a, d\rangle\}$$

es transitiva.

5.7. Composición

Sea R una relación de A hacia B y S una relación de B hacia C. La composición de R y S, denotada $S \circ R$, es la relación:

$$S \circ R = \{\langle a, c\rangle \mid \langle a, b\rangle \in R, \langle b, c\rangle \in S\}$$

Ejemplo 1: Sea $A = \{1, 3, 5\}$, $B = \{a, b\}$, $C = \{\alpha, \beta, \gamma\}$, R una relación de A hacia B definida como $R = \{\langle 1, b\rangle, \langle 3, a\rangle, \langle 5, b\rangle, \langle 3, b\rangle\}$ y S una relación de B hacia C definida como $S = \{\langle a, \alpha\rangle, \langle a, \gamma\rangle, \langle b, \beta\rangle, \langle b, \gamma\rangle\}$. La composición de R y S es:

$$S \circ R = \{\langle 1, \beta\rangle, \langle 1, \gamma\rangle, \langle 3, \alpha\rangle, \langle 3, \gamma\rangle, \langle 5, \beta\rangle, \langle 5, \gamma\rangle, \langle 3, \beta\rangle, \langle 3, \gamma\rangle\}$$

5.8. Ordenes Parciales

Una relación R definida en un conjunto A es llamada *orden parcial* si es reflexiva, antisimétrica y transitiva.

Ejemplo: La relación $\leq$ sobre $\mathbb{Z} \times \mathbb{Z}$ es un orden parcial. La definición implica que para todo $a, b, c \in \mathbb{Z}$ tenemos:

$$\begin{aligned} a &\leq a \\ a \leq b, b \leq a &\rightarrow a = b \\ a \leq b, b \leq c &\rightarrow a \leq c \end{aligned}$$

5.9. Relaciones de Equivalencia

Una relación R definida en un conjunto A es llamada *relación de equivalencia* si es reflexiva, simétrica y transitiva.

Ejemplo: La relación R sobre $X = \{l \mid l \text{ es una línea recta}\}$, si x e y son paralelas entonces xRy, es una relación de equivalencia.

1. xRx es verdadero para cada $x \in X$.
2. xRy implica yRx, esto es, si x es paralela a y entonces y es paralela a x.
3. Para tres líneas x, y, z, si x es paralela a y e y es paralela a z, entonces x es paralela a z.

Capítulo 6

Funciones

Una función de un conjunto A a un conjunto B es una relación de A hacia B tal que cada elemento de A está relacionado únicamente con un elemento del conjunto B. El conjunto A es llamado el dominio y el conjunto B el codominio.

Formalmente, es una relación binaria no vacía $f \subseteq A \times B$ si cada elemento de A aparece exactamente una vez como el primer componente de un par ordenado en la relación f. Escribimos $f : A \longrightarrow B$ para denotar una función f de A a B y escribimos $f(a) = b$ cuando $\langle a, b\rangle \in f$.

La definición implica que para cada $\langle a, b\rangle \in f$, f asocia con $a \in A$ únicamente el elemento $b \in B$. Se dice que b es la imagen de a bajo f. El rango de f es el conjunto

$$f(A) = \{b \mid b = f(a), a \in A\}$$

Ejemplo

Sea $f : \mathbb{R} \longrightarrow \mathbb{R}$ dada por $f(x) = x^2$. ¿Cuál es su rango? y ¿cuál es la imagen de $\mathbb{Z}$ bajo f?

- El rango de f es $f(\mathbb{R}) = [0, +\infty)$.
- La imagen de $\mathbb{Z}$ bajo f es $f(\mathbb{Z}) = \{0, 1, 4, 9, ...\}$.

6.1. Propiedades

6.1.1. Funciones inyectivas o uno a uno

Una función $f : A \longrightarrow B$ es inyectiva si cada elemento en el rango de f es la imagen de exactamente un elemento del dominio. $a_1, a_2 \in A, f(a_1) = f(a_2)$ implica $a_1 = a_2$, o lo que es lo mismo $a_1, a_2 \in A, a_1 \neq a_2$ implica $f(a_1) \neq f(a_2)$.

Ejercicios

1. ¿Es $f : \mathbb{R} \longrightarrow \mathbb{R}$ dada por $f(x) = x^2$ una función inyectiva?

2. ¿Es $f : \mathbb{R} \longrightarrow \mathbb{R}$ dada por $f(x) = x^3$ una función inyectiva?

6.1.2. Funciones sobreyectivas

Una función $f : A \longrightarrow B$ es sobreyectiva si $f(A) = B$, esto es, para cada elemento $b \in B$ existe al menos un elemento $a \in A$ con $f(a) = b$.

$f : \mathbb{R} \longrightarrow \mathbb{R}$ dada por $f(x) = x^2$ no es sobreyectiva. $f : \mathbb{R} \longrightarrow \mathbb{R}$ dada por $f(x) = x^3$ si lo es.

6.1.3. Funciones biyectivas o de correspondencia uno a uno

Una función $f : A \longrightarrow B$ es biyectiva si f es inyectiva y sobreyectiva. De aquí que si f es una función biyectiva entonces $|A| = |B|$.

Ejercicio

Proponer una función biyectiva.

6.1.4. Composición

Si f es una función de A a B y g es una función de B a C, entonces la función composición $g \circ f$ es la función de A a C definida por

$$(g \circ f)(x) = g(f(x))$$

para cada $x \in A$.

Propiedades de la composición

1. Teorema: Si $f : A \longrightarrow B$, $g : B \longrightarrow C$, $h : C \longrightarrow D$, entonces $(h \circ g) \circ f = h \circ (g \circ f)$.

2. Teorema: Si f y g son inyectivas, entonces $g \circ f$ es inyectiva.

3. Teorema: Si f y g son sobreyectivas, entonces $g \circ f$ es sobreyectiva.

4. Corolario: Si $f : A \longrightarrow B$ y $g : B \longrightarrow C$ son biyectivas, entonces $g \circ f$ es biyectiva.

6.1.5. Funciones inversas

Una función $f^{-1} : B \longrightarrow A$ es inversa de $f : A \longrightarrow B$ si $f^{-1} \circ f = i_A$ y $f \circ f^{-1} = i_B$. Siendo i_A la función identidad definida como $i_A : A \longrightarrow A$ definida por $i_A(x) = x$ para todo $x \in A$.

6.1.6. Funciones características

Sea A un conjunto y S cualquier subconjunto de A. Sea $C_s : A \longrightarrow \{0,1\}$ definida por:

$$C_s(x) = \begin{cases} 1, & \text{si } x \in S; \\ 0, & \text{si } x \notin S. \end{cases} \tag{6.1}$$

La función C_s es llamada la función característica de S.

6.1.7. Funciones recursivas

- Función sucesor: $S(0) = 1, S(k) = 1 + S(k-1), k \geq 1$.
- Función factorial: Sea $f : \mathbb{N}_0 \longrightarrow \mathbb{N}$ definida como $f(0) = 1, f(n) = nf(n-1), n \geq 1$.
- Función de la serie de Fibonacci: Sea $g : \mathbb{N}_0 \longrightarrow \mathbb{N}_0$ definida como $g(0) = 0, g(1) = 1, g(n) = g(n-2) + g(n-1), n \geq 2$.

6.2. Funciones primitivas recursivas

La clase de *funciones primitivas recursivas*, $\mathcal{PR}$, es la cerradura de $I = \{s = \lambda x.x + 1, z = \lambda x.0, ((u_i^n = \lambda \vec{x}_n.x_i)_{1 \leq i \leq n})_{n \geq 1}\}$ bajo las operaciones de composición y recursión primitiva.

Se puede apreciar que $\mathcal{PR}$ contiene una gran cantidad de funciones; más aún, contiene funciones grandes. La teoría de la computabilidad no puede ser basada en tecnología (presente o futura). Tal teoría requeriría que $\lambda x.\underbrace{2^{2^{\cdots^2}}}_{x}$ fuera no algorítmica pues ninguna máquina, no importa que tan rápida sea, puede calcular $\underbrace{2^{2^{\cdots^2}}}_{x}$ para x grande en un tiempo razonable; sin embargo, teóricamente, existe un procedimiento sencillo mediante el cual, con suficiente tiempo, papel y lápiz, uno puede calcular $\underbrace{2^{2^{\cdots^2}}}_{x}$ para cualquier x.

6.2.1. Recursión primitiva

La recursión primitiva asigna una función f a un par de funciones g y h de acuerdo al siguiente esquema:

$$f(0, \vec{y}_m) = h(\vec{y}_m) \tag{6.2}$$
$$f(x+1, \vec{y}_m) = g(x, \vec{y}_m, f(x, \vec{y}_m)) \tag{6.3}$$

Predecesor

Demostrar que $p = \lambda x.x\dot{-}1 \in \mathcal{PR}$

$$\begin{aligned} 0\dot{-}1 &= 0 \\ (x+1)\dot{-}1 &= x \end{aligned}$$

$$\begin{aligned} p(0,y) &= z(y) \\ p(x+1,y) &= u_1^3(x,y,p(x,y)) \end{aligned}$$

Suma

Demostrar que $a = \lambda xy.x + y \in \mathcal{PR}$

$$\begin{aligned} 0+y &= y \\ x+1+y &= x+y+1 \end{aligned}$$

$$\begin{aligned} a(0,y) &= u(y) \\ a(x+1,y) &= s(u_3^3(x,y,a(x,y))) \end{aligned}$$

$$\begin{aligned} a(0,y) &= u(y) \\ a(x+1,y) &= g(x,y,a(x,y)) \end{aligned}$$

donde $g = s \circ u_3^3$.

Resta propia

Demostrar que $d = \lambda xy.x\dot{-}y \in \mathcal{PR}$

$$\begin{aligned} x\dot{-}0 &= x \\ x\dot{-}(y+1) &= x\dot{-}y\dot{-}1 \end{aligned}$$

este esquema se puede convertir al esquema primitivo recursivo:

$$\begin{aligned} \hat{d}(0,y) &= u(y) \\ \hat{d}(x+1,y) &= p(u_3^3(x,y,\hat{d}(x,y))) \end{aligned}$$

$$\begin{aligned} \hat{d}(0,y) &= u(y) \\ \hat{d}(x+1,y) &= h_1(x,y,\hat{d}(x,y)) \end{aligned}$$

donde $h_1 = p \circ u_3^3$. Es evidente que $\hat{d}(x,y) = d(y,x)$, por lo tanto, $d = \lambda xy.\hat{d}(u_2^2(x,y),u_1^2(x,y))$

Tarea

1. Demostrar que $m = \lambda xy.xy \in \mathcal{PR}$
2. Demostrar que $\lambda x.2^x \in \mathcal{PR}$
3. Demostrar que $\lambda xyz.$**if** $x = 0$ **then** y **else** $z \ \in \mathcal{PR}$

Capítulo 7

Técnicas de análisis

7.1. Conteo

La combinatoria, el estudio de arreglos de objetos, es una parte importante de las matemáticas discretas. Este tema fue estudiado en el siglo XVII, cuando preguntas de combinatorias surgieron a partir del estudio de juegos de azar. La enumeración, el conteo de objetos con ciertas características, es una parte importante de las combinatorias. Debemos contar objetos para resolver diferentes problemas. Por ejemplo: ¿cuantas posibles contraseñas existen si se usan 6 caracteres?, ¿cuál es la probabilidad de que una persona elija correctamente 6 números entre 48?

7.1.1. Principios Básicos del conteo

Regla de la suma

Si una tarea puede ser hecha en n_1 maneras y una segunda tarea puede ser hecha en n_2 maneras, y si estas dos no pueden ser hechas simultáneamente, entonces hay $n_1 + n_2$ maneras de hacer cualquier tarea.

Ejemplo: Un estudiante puede elegir un proyecto de entre tres diferentes listas. Cada una contiene 23, 15 y 19 posibles proyectos, respectivamente. ¿De entre cuantos proyectos puede escoger? *Solución:* El estudiante puede escoger un proyecto de la primera lista de 23 maneras, de la segunda de 15 maneras, y de la tercera de 19 maneras. Entonces, hay $23 + 15 + 19 = 57$ proyectos para escoger.

En términos de conjuntos, la regla de la suma puede ser expresada de la siguiente manera: si $A_1, A_2, \ldots, A_m$ son conjuntos disjuntos, entonces el número de elementos de la union de estos conjuntos es la suma de el número de elementos en ellos.

$$|A_1 \cup A_2 \cup \cdots \cup A_m| = |A_1| + |A_2| + \cdots + |A_m|$$

Regla del producto

Supóngase que un procedimiento puede ser dividido en dos tareas. Si existen n_1 maneras para hacer la primera tarea y n_2 maneras de hacer la segunda tarea después de que la primera haya sido completada, entonces existen $n_1 n_2$ maneras de hacer el procedimiento.

Ejemplo 1: ¿Cuantas posibles placas de automóvil existen si cada placa contiene una secuencia de 3 letras seguidas de 3 dígitos (y no existen secuencias prohibidas)?. *Solución:* Existen 27 opciones para cada una de las 3 letras y 10 opciones para cada uno de los 3 dígitos. Entonces, por medio de la regla del producto, existen un total de $27 \cdot 27 \cdot 27 \cdot 10 \cdot 10 \cdot 10 = 19683000$ posibles placas.

Ejemplo 2: ¿Cuantas diferentes funciones existen de un conjunto con m elementos a un conjunto con n elementos?. *Solución:* Una función corresponde a una elección de uno de los n elementos en el codominio para cada uno de los m elementos en el dominio. Entonces, por medio de la regla del producto, existen $n \cdot n \cdot n \cdot \cdots \cdot n = n^m$ funciones desde un conjunto de m elementos a uno con n elementos.

En términos de conjuntos, la regla del producto se describe de la siguiente forma: Si $A_1, A_2, \ldots, A_m$ son conjuntos finitos, entonces el número de elementos en el producto cartesiano de estos conjuntos es el producto del número de elementos en cada conjunto:

$$|A_1 \times A_2 \times \cdots \times A_m| = |A_1| \cdot |A_2| \cdot \cdots \cdot |A_m|$$

Principio de Inclusión-Exclusión

Cuando dos tareas pueden ser hechas simultáneamente, no podemos utilizar la regla de la suma para contar el número de formas de hacer una de las dos tareas. Sumar el número de formas de hacer cada una de las tareas nos lleva a un sobre-conteo, pues las formas de hacer ambas tareas se cuentan dos veces. Para contar correctamente el número de formas de hacer una de las dos tareas, se suma el número de formas de hacer cada una de las dos tareas y después se resta el número de formas de hacer ambas tareas.

Ejemplo: ¿Cuantas cadenas de 8 bits empiezan con un bit 1 o terminan con los dos bits 00?. *Solución:* La primera tarea, construir una cadena de longitud 8 empezando con un bit 1, puede ser hecha en $2^7 = 128$ maneras; por la regla del producto, pues el primer bit puede ser escogido sólo de una manera y cada uno de los 7 restantes en dos maneras. La segunda tarea, puede ser realizada en $2^6 = 64$ formas. Ambas tareas pueden ser hechas de $2^5 = 32$ formas. Por consecuencia, el número de cadenas de 8 bits que empiezan con 1 o terminan en 00, es $128 + 64 - 32 = 160$.

En términos de conjuntos, sean A_1 y A_2 dos conjuntos. Sea T_1 la tarea de escoger un elemento de A_1 y T_2 la tarea de escoger un elemento de A_2. Existen $|A_1|$ formas de hacer T_1 y $|A_2|$ formas de hacer T_2. El número de formas de hacer T_1 o T_2 es la suma del número de formas de hacer T_1 y el número de formas de hacer T_2 menos el número de formas de hacer T_1 y T_2.

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$$

7.1.2. Permutaciones y Combinaciones

Permutaciones

Una **permutación** de un conjunto de objetos distintos es un arreglo ordenado de estos objetos. Un arreglo ordenado de r elementos de un conjunto es llamado permutación-r.

Teorema: El número de permutaciones-r de un conjunto con n elementos distintos es: $P(n,r) = n(n-1)(n-2)\cdots(n-(r-1))$ o lo que es lo mismo, $n!/(n-r)!$

Ejemplo 1: ¿Cuantas maneras diferentes existen de seleccionar 4 diferentes jugadores de un conjunto de 10 para jugar 4 partidos? *Solución:* $P(10,4) = 10 \cdot 9 \cdot 8 \cdot 7 = 5040$

Ejemplo 2: Suponga que una mujer necesita visitar 8 diferentes ciudades. Debe empezar su viaje en una ciudad en específico, pero puede visitar las otras 7 ciudades en el orden que más le parezca. ¿Cuantas posibles rutas puede tomar la mujer visitando estas ciudades? *Solución:* Puesto que la primera ciudad está determinada, entonces el número de posibles rutas es el número de permutaciones de 7 elementos. Por consecuencia, existen $7! = 5040$ formas de realizar el recorrido.

Combinaciones

Una **combinación-r** de elementos de un conjunto, es una selección desordenada de r elementos de un conjunto. Entonces, una combinación-r es simplemente un subconjunto con r elementos de un conjunto.

Teorema: El número de combinaciones-r de un conjunto de n elementos, donde n es un entero positivo y r es un entero tal que $0 \leq r \leq n$, es $C(n,r) = \frac{n!}{r!(n-r)!}$. **Corolario:** Sean n y r dos enteros no negativos y $r \leq n$, entonces $C(n,r) = C(n,n-r)$

Ejemplo: ¿Cuántas formas de escoger 5 jugadores de un equipo de 10 existen? *Solución:* La respuesta está dada por el numero de combinaciones-5 de un conjunto de 10 elementos, $C(10,5) = 10!/(5!5!) = 252$.

7.1.3. El principio del palomar

Supóngase que una parvada de palomas vuela hacia un conjunto de palomares para descansar. El *principio del palomar* indica que si hay más palomas que palomares, entonces debe haber al menos un palomar con al menos dos palomas en él.

Ejemplo: ¿Cuántos estudiantes deben estar en una clase para garantizar que al menos dos estudiantes reciban la misma calificación en el examen final, si el examen es calificado en una escala de 0 a 100?. *Solución:* Existen 101 posibles

calificaciones. El principio del palomar indica que entre 102 estudiantes debe haber al menos dos estudiantes con la misma calificación.

Principio Generalizado del palomar

Si N objetos son colocados en k cajas, entonces hay una caja que contiene al menos $\lceil N/k \rceil$ objetos. Prueba: Suponga que ninguna de las cajas contiene más de $\lceil N/k \rceil - 1$ objetos, además se utiliza la propiedad de $\lceil N/k \rceil < (N/k) + 1$. Entonces, el número total de objetos es a lo mucho:

$$k(\lceil N/k \rceil - 1) < k(((N/k) + 1) - 1) = N$$

Esto es una contradicción pues existe un total de N objetos.

Ejemplo 1: De entre 100 personas existen al menos 9 que nacieron en el mismo mes. $\lceil 100/12 \rceil = 9$.

Ejemplo 2: ¿Cuál es el mínimo número de estudiantes requeridos para estar seguros que al menos 6 de ellos obtendrán la misma calificación?, asumiendo 5 posibles calificaciones. *Solución:* Se requiere encontrar el entero N más pequeño que satisfaga $\lceil N/5 \rceil = 6$, este entero es $N = 5 \cdot 5 + 1 = 26$, por lo tanto, se requieren 26 estudiantes para cumplir con la condición estipulada.

Capítulo 8

Estructuras algebraicas

Capítulo 9

Grafos

Los grafos son estructuras discretas que constan de vértices y de aristas que conectan entre sí los vértices.

9.1. Tipos de grafos

9.1.1. Grafo simple

Un grafo simple $G = (V, A)$ costa de un conjunto no vacío de vértices V y de un conjunto A de pares no ordenados de elementos distintos de V, a estos pares se les llama aristas. En otras palabras un grafo simple es un grafo en los que existe a lo más una arista que une dos vértices distintos.

9.1.2. Multigrafos

Un multigrafo $G = (V, A)$ consta de un conjunto V de vértices, un conjunto A de aristas y una función f de A hacia $\{\{u, v\}|u, v \in V, u \neq v\}$. Se dice que las aristas a_1 y a_2 son aristas múltiples o paralelas si $f(a_1) = f(a_2)$.

9.1.3. Pseudografos

Un pseudografo $G = (V, A)$ consta de un conjunto V de vértices, un conjunto A de aristas y una función f de A hacia $\{\{u, v\}|u, v \in V\}$. Una arista a es un bucle o lazo, si $f(a) = \{u, u\} = \{u\}$ para algún $u \in V$.

9.1.4. Grafo dirigido

Un grafo dirigido (V, A) consta de un conjunto V de vértices y de un conjunto A de aristas, que son pares ordenados de elementos de V. Utilizamos el par ordenado $\langle u, v\rangle$ para indicar que es una arista dirigida del vértice u al vértice v.

Tipo	Aristas	Aristas múltiples	Bucles
Grafo simple	No dirigidas	No	No
Multigrafo	No dirigidas	Sí	No
Pseudografo	No dirigidas	Sí	Sí
Grafo dirigido	Dirigidas	No	Sí
Multigrafo dirigido	Dirigidas	Sí	Sí

Cuadro 9.1: Terminología en teoría de grafos

9.1.5. Multigrafos dirigidos

Un multigrafo dirigido $G = (V, A)$ consta de un conjunto V de vértices, un conjunto A de aristas y una función f de A hacia $\{\langle u, v\rangle | u, v \in V\}$. Se dice que las aristas a_1 y a_2 son aristas múltiples o paralelas si $f(a_1) = f(a_2)$.

9.1.6. Grado del vértice

El grado de un vértice u es el número de aristas incidentes a él.

9.1.7. Grafo completo

Un grafo completo es un grafo simple que tiene una arista entre cada par de vértices distintos.

9.2. Conexión

9.2.1. Caminos

Sea n un entero no negativo y sea G un grafo no dirigido. Un camino de longitud n de u a v en G es una secuencia de n aristas $a_1, a_2, \ldots, a_n$ de G tal que $f(a_1) = \{u, x_1\}$, $f(a_2) = \{x_1, x_2\}, \ldots, f(a_n) = \{x_{n-1}, v\}$. Si el grafo es simple podemos denotar el camino mediante los vértices, si es un multigrafo será necesario denotar el camino mediante las aristas, pues puede haber ambigüedades.

9.2.2. Circuitos

Un camino de longitud $n > 0$ es un circuito si comienza y termina en el mismo vértice.

9.2.3. Grafos conexos

Conexión en grafos no dirigidos

Se dice que un grafo no dirigido es conexo si hay un camino entre cada par de vértices distintos del grafo.

Conexión en grafos dirigidos

Se dice que un grafo es *fuertemente conexo* si hay un camino de a a b y un camino de b a a para cualesquiera dos vértices a y b del grafo.

Un grafo es *débilmente conexo* si hay un camino entre cada dos vértices del grafo no dirigido subyacente. El grafo no dirigido subyacente es el resultado de ignorar las direcciones de un grafo dirigido.

9.3. Caminos eulerianos y hamiltonianos

9.3.1. Caminos y circuitos eulerianos

Los siete puentes de Königsberg es un famoso problema matemático resuelto por el Leonhard Euler. Este problema tiene su origen en una situación real. La ciudad de Königsberg está situada en el Rio Pregel y se tenían dos grandes islas conectadas mediante siete puentes. El problema es simple, encontrar una ruta tal que se cruce cada puente exactamente una vez. En 1973 Leonhard Euler probó que no era posible utilizando teoría de grafos.

Un camino euleriano es un camino simple que contiene todas las aristas de G. Un circuito euleriano es un circuito que contiene a todas las aristas de G.

Teorema 1

Un multigrafo conexo contiene un camino euleriano, pero no un circuito euleriano, si y sólo si, tiene exactamente dos vértices de grado impar.

Teorema 2

Un multigrafo conexo contiene un circuito euleriano si y sólo si, cada uno de sus vértices tiene grado par.

9.3.2. Caminos y circuitos hamiltonianos

Se dice que un camino $v_0, v_1, \ldots, v_n$ del grafo $G = (V, A)$ es un camino hamiltoniano si $V = \{v_0, v_1, \ldots, v_{n-1}, v_n\}$ y $v_i \neq v_j$ para $0 \leq i < j \leq n$. En otras palabras, un camino hamiltoniano es un camino que visita todos los vértices una sola vez.

Se dice que un circuito $v_0, v_1, \ldots, v_n, v_0$ es un circuito hamiltoniano si $v_0, v_1, \ldots, v_n$ es un camino hamiltoniano.

9.4. Grafos ponderados

Llamamos grafos ponderados a los grafos en los que se asigna un número a cada una de las aristas. Este número representa un peso para el recorrido a través de la arista. Este peso podrá indicar, por ejemplo, la distancia, el costo monetario o el tiempo invertido, entre otros.

Definimos la longitud de un camino en un grafo ponderado como la suma de los pesos de las aristas de ese camino.

9.4.1. Caminos de longitud mínima

Uno de los problemas más comunes en grafos ponderados es determinar cuál es el camino más corto entre dos vértices dados. La solución a este problema tiene aplicaciones directas en muchas áreas, como transporte, manufactura y redes informáticas.

Otro problema importante que involucra grafos ponderados es el problema del agente viajero, que plantea la interrogante de cual es el orden en el que un viajante debe realizar un circuito visitando cada una de las ciudades de su ruta para que la distancia total recorrida sea mínima.

Algoritmo de Dijkstra

Se tiene un grafo G ponderado simple y conexo con todos los pesos positivos. Tiene vértices $v_0, v_1, \ldots, v_n$, siendo $a = v_0$ el vértice origen y $z = v_n$ el vértice destino. Además tenemos una función de pesos $w(v_i, v_j)$ que determina el peso de la arista que une los vértices v_i y v_j, si dicha arista no existe entonces $w(v_i, v_j) = \infty$.

El algoritmo incluye un conjunto auxiliar S de vértices y una función $L(v)$ que indica la longitud del camino más corto entre a y v.

- Desde $i = 1$ hasta n
  - $L(v_i) = \infty$ [Todos los elementos excepto a]
- $L(a) = 0$ [La longitud de a a a es 0]
- $S = \phi$
- Mientras $z \notin S$ hacer
  - $u =$ vértice no en S con $L(u)$ mínima.
  - $S = S \cup \{u\}$. [Agregamos u al conjunto]
  - Para todos los vértices v no en S
    - Si $L(u) + w(u, v) < L(v)$ entonces $L(v) = L(u) + w(u, v)$ [Actualizamos la longitud si fue menor]
- Al final $L(z)$ tiene la longitud del camino más corto entre a y z.

El algoritmo de Dijkstra realiza $O(n^2)$ operaciones para determinar la longitud del camino más corto entre dos vértices en un grafo ponderado simple con n vértices.

9.4.2. El problema del agente viajero

El problema del agente viajero pide determinar el circuito de peso total mínimo de un grafo ponderado, completo y no dirigido que visita cada vértice exactamente una vez y regresa al punto de partida. Esto es equivalente a encontrar un circuito hamiltoniano con peso total mínimo.

La complejidad de determinar una solución es muy grande. Una vez que se ha elegido el vértice inicial, se tienen $n-1$ vértices restantes, y una vez elegido el segundo vértice se tienen $n-2$ vértices restantes. Una búsqueda exhaustiva entonces tendrá que examinar $(n-1)!/2$ circuitos hamiltonianos distintos. Tratar de resolver el problema para unas cuantas decenas de vértices es prácticamente imposible. A la fecha no se conoce ningún algoritmo de complejidad polinómica que resuelva el peor caso.

Sin embargo existen algoritmos de aproximación, es decir, se garantiza que la solución propuesta esté cerca del óptimo. Lamentablemente para aplicar este tipo de algoritmos es necesario que el grafo tenga ciertas propiedades y el caso general sigue sin solución.

Una forma de solución que no garantiza estar cerca del óptimo pero que en ocasiones da buenos resultados es el algoritmo voraz.

9.5. Grafos planos

Se dice que un grafo es plano si puede dibujarse en el plano de manera que ningún par de sus aristas se corte. A ese dibujo se le llama representación plana del grafo.

9.6. Coloreado de grafos

Al colorear un mapa se suele asignar colores distintos a las regiones que tienen una frontera común. Una forma de garantizar que dos regiones adyacentes no tengan nunca el mismo color es emplear un color distinto para cada región del mapa. Esto no es eficiente y en los mapas con muchas regiones sería muy difícil distinguir colores parecidos. Por el contrario, debería utilizarse un número pequeño de colores siempre que sea posible.

Todo mapa en el plano puede representarse por medio de un grafo. Para establecer esta correspondencia, cada región del mapa se representa mediante un vértice. Una arista conecta dos vértices si las regiones representadas tienen una frontera común. Al grafo resultante se le llama *grafo dual* del mapa.

El problema de colorear las regiones de un mapa es equivalente al de colorear los vértices del grafo dual de tal manera que ningún par de vértices adyacentes del grafo tengan el mismo color.

Una *coloración* de un grafo simple consiste en asignarle un color a cada vértice del grafo de manera que a cada dos vértices adyacentes se les asignan colores distintos.

El *número cromático* de un grafo es el número mínimo de colores que se requieren para una coloración del grafo.

Teorema El teorema de los cuatro colores dice que el número cromático de un grafo plano es menor o igual que cuatro.

Para grafos no planos el número cromático puede ser muy grande. Los mejores algoritmos que se conocen para calcular el número cromático de un grafo tienen complejidad exponencial en el peor caso. Incluso hallar una aproximación del número cromático de un grafo es un problema difícil.

Capítulo 10

Árboles

10.1. Definiciones

Un *árbol* es un grafo no dirigido, conexo y sin ciclos. Un grafo no dirigido es un árbol si, y sólo si, hay un único camino entre cada pareja de vértices.

Un *árbol con raíz* es un árbol en el que uno de sus vértices ha sido designado como la raíz y todas las aristas están orientadas de modo que se alejan de la raíz.

Supongamos que T es un árbol con raíz. Si v es un vértice de T distinto de la raíz, el *padre* de v es el único vértice u tal que hay una arista dirigida de u a v. Cuando u es padre de v, se dice que v es *hijo* de u. Los vértices con el mismo padre se llaman *hermanos*. Los *antecesores* de un vértice diferente de la raíz son todos los vértices que aparecen en el camino desde la raíz hasta ese vértice. Los *descendientes* de un vértice v son aquellos vértices para los cuales v es un antecesor.

Un vértice de un árbol se llama *hoja* si no tiene hijos. Los vértices que tienen hijos se llaman *vértices internos*.

Si a es un vértice de un árbol, el *subárbol* con raíz en a es el subgrafo del árbol que contiene al vértice a, a todos sus descendientes y todas las aristas incidentes en dichos descendientes.

10.1.1. Árboles n-arios

Un árbol con raíz se llama árbol n-ario si todos los vértices internos tienen a lo sumo n hijos. El árbol se llama árbol n-ario completo si todo vértice interno tiene exactamente n hijos. Un árbol n-ario con $n = 2$ se llama árbol binario.

10.2. Aplicaciones de los árboles

10.2.1. Árboles binarios de búsqueda

Es un árbol binario en el que cada hijo de un vértice se designa como hijo izquierdo o hijo derecho, ningún vértice tiene más de un hijo izquierdo y un hijo derecho y cada vértice está etiquetado con una clave, que es uno de los objetos. Además, a los vértices se les asignan las claves de modo que la clave de un vértice es mayor que la de todos los vértices de su subárbol izquierdo y menor que la de todos los vértices de su subárbol derecho.

10.2.2. Árboles de decisión

Un árbol con raíz en el que cada vértice interno corresponde a una decisión, con un subárbol en dichos vértices para cada posible resultado de la decisión, se llama árbol de decisión.

Las posibles soluciones del problema corresponden a los caminos desde la raíz hasta las hojas.

10.2.3. Códigos instantáneos

10.3. Recorridos de árboles

10.3.1. Recorrido preorden

Sea T un árbol ordenado con raíz r. Si T consta sólo de r, entonces r es el recorrido en preorden de T. En otro caso, $T_1, T_2, \ldots, T_n$ son los subárboles de r listados de izquierda a derecha en T. El recorrido preorden comienza visitando r, continúa recorriendo T_1 en preorden, luego T_2 en preorden y así sucesivamente hasta recorrer T_n en preorden.

10.3.2. Recorrido inorden

Sea T un árbol ordenado con raíz r. Si T consta sólo de r, entonces r es el recorrido en inorden de T. En otro caso, supongamos que $T_1, T_2, \ldots, T_n$ son los subárboles de r listados de izquierda a derecha en T. El recorrido en inorden comienza recorriendo T_1 en inorden y continúa visitando r, a continuación recorre T_2 en inorden, después T_3 en inorden y así sucesivamente hasta recorrer T_n en inorden.

10.3.3. Recorrido postorden

Sea T un árbol ordenado con raíz r. Si T consta sólo de r, entonces r es el recorrido en postorden de T. En otro caso, supongamos que $T_1, T_2, \ldots, T_n$ son los subárboles de r listados de izquierda a derecha en T. El recorrido en postorden comienza recorriendo T_1 en postorden, luego recorre T_2 en postorden y así sucesivamente hasta recorrer T_n en postorden y finaliza visitando r.

Bibliografía

[Aho 1995] Alfred V. Aho, Jeffrey D. Ullman *Foundations of Computer Science: C Edition*, W.H. Freeman and Company, 1995

[Alagar 1989] Alagar, Vangalur S., *Fundamentals of Computing: Theory and Practice*, Prentice Hall, 1989

[Doerr 1985] Doerr, Alan, *Applied Discrete Structures for Computer Science*, Science Research Associates, Inc., 1985

[Enderton 2000] Herbert B. Enderton, *A Mathematical Introduction to Logic*, Academic Press, 2000

[Hopcroft 1979] Hopcroft, John E., *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, 1979

[Knuth 1989] Knuth, Donald E.,*Concrete Mathematics*, 2nd Edition, Addison-Wesley, 1989

[Rosen 1999] Rosen, Kenneth H., *Discrete Mathematics and Its Applications*, 4th Edition, McGraw-Hill, 1999

[Tourlakis 1984] Tourlakis, George J., *Computability*, Reston Publishing Company, Inc., 1984