

Resumen general de Matemáticas Discretas

Andrés Cabezas y Sebastián Poblete

March 29, 2022

1 ¿Qué son las matemáticas discretas?

“El lenguaje necesario para entender y modelar la computación”. Las matemáticas discretas usan conjuntos finitos e infinitos al momento de estudio. Modelan los objetos y conceptos abstractos de las matemáticas que pueden ser representados dentro de un computador.

1.1 Lógica

La lógica consiste en el uso y estudio del razonamiento válido. Para esto, es necesario un lenguaje, pero esto también supone un problema: Los lenguajes que los humanos hablan tiene ciertas subjetividades y diferencias entre si, lo que conduce a errores al momento de usar la lógica. Para resolver esto, es necesario usar un lenguaje formal.

Durante el curso, se estudiarán dos lógicas (o lenguajes), sin embargo, existen muchos más

- Lógica Proposicional
- Lógica de Predicados

¿Para qué son necesarias estas lógicas? Recordemos nuestro objetivo. Queremos usar esto para realizar nuestro razonamiento matemático. De esta forma, podemos definir correctamente objetos matemáticos, teorías matemáticas y realizar demostraciones más formales

2 Lógica Proposicional (LP)

2.1 Proposición

Una proposición consiste en una afirmación, la cual puede ser *verdadera* (1) o *falsa* (0). Para denotar proposiciones básicas, usaremos letras mayúsculas (Ej.: P, Q, R...)

2.2 Conectivos Lógicos

La LP usa conectivos sencillos para conseguir formar proposiciones más complejas.

Conectivos	Nombre	Uso	Significado
\wedge	Conjunción	$P \wedge Q$	P y Q
\vee	Disyunción	$P \vee Q$	P o Q
\neg	Negación	$\neg P$	No P
\rightarrow	Condición	$P \rightarrow Q$	Si P, entonces Q
\leftrightarrow	Bicondición	$P \leftrightarrow Q$	P, si y solo si, Q

2.2.1 Conjunción (\wedge)

El valor de verdad de una conjunción es *verdadero* si ambas proposiciones (a cada lado del signo) son verdaderas. En cualquier otro caso, es *falso*.

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

2.2.2 Disyunción (\vee)

El valor de verdad de una disyunción es *verdadero* si al menos una de las proposiciones (a cada lado del signo), es verdadera.

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

2.2.3 Negación (\neg)

El valor de verdad corresponde al opuesto del valor entregado (a la derecha del signo)

P	$\neg P$
1	0
0	1

2.2.4 Condicional (\rightarrow)

El valor de verdad de una condicional del tipo $P \rightarrow Q$ es *falso* si P es verdadero, pero Q es falso. En cualquier otro caso, es *verdadero*.

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

Hint: “Si P es verdadero, entonces necesariamente Q es verdadero”. Si P es verdadero, entonces Q deberá ser verdadero para tener un valor de verdad *verdadero*. Si P es falso, entonces de forma automática el valor de verdad es *verdadero*.

2.2.5 Bicondicional (\leftrightarrow)

El valor de verdad de una bicondicional es verdadero si ambas proposiciones (a ambos lados del signo) son iguales (en otras palabras, ambas verdaderas o ambas falsas).

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

2.3 Proposición Compuesta

Una proposición es compuesta si corresponde a la negación (\neg), conjunción (\wedge), disyunción (\vee), condicional (\rightarrow) o bicondicional (\leftrightarrow) de proposiciones compuestas.

Como por ejemplo

$$\begin{aligned} &P \wedge (Q \vee R) \\ &\neg(P \vee (\neg R \wedge Q)) \\ &(P \rightarrow Q) \leftrightarrow (P \wedge Q) \end{aligned}$$

Si se desea obtener el valor de verdad de alguna proposición compuesta, se debe evaluar de forma recursiva cada uno de los conectivos lógicos presentes.

Por ejemplo:

$$\begin{aligned} &\neg(P \vee (\neg R \wedge Q)) \text{ con } P = 0, Q = 1 \text{ y } R = 0 \\ &\neg(0 \vee (\neg 0 \wedge 1)) \\ &\neg(0 \vee (1 \wedge 1)) \\ &\neg(0 \vee 1) \\ &\neg 1 \\ &0 \end{aligned}$$

$$\begin{aligned} &(P \rightarrow Q) \leftrightarrow (P \wedge Q) \text{ con } P = 1 \text{ y } Q = 0 \\ &(1 \rightarrow 0) \leftrightarrow (1 \wedge 0) \\ &0 \leftrightarrow 0 \\ &1 \end{aligned}$$

2.3.1 Paréntesis y prioridad

El orden de prioridad entre conectivos lógicos, al momento de evaluar proposiciones compuestas, será el siguiente:

Conectivo	Precedencia
\neg	1
\wedge	2
\vee	3
\rightarrow	4
\leftrightarrow	5

3 Formulas y Valuaciones

3.1 Variables Proposicionales

Una variable proposicional es una variable que puede ser reemplazada con los valores 1 o 0. Generalmente son representadas con una letra minúscula (Amiga eri boolean)

3.2 Formulas Proposicionales

Una formula proposicional es una formula que puede ser

- Una variable proposicional
- Los valores 1 o 0
- Una combinación con conectivos lógicos

Generalmente son representadas con letras griegas (Ej.: α)

Ejemplos:

$$\begin{aligned}\alpha(p, q, r) &:= p \wedge (q \rightarrow r) \\ \beta(p, q) &:= (p \wedge \neg q) \vee (\neg p \wedge 1)\end{aligned}$$

4 Equivalencia Lógica

4.1 Definición

Si tenemos dos formulas proposicionales con las mismas variables proposicionales

$$\alpha(p_1, \dots, p_n) \text{ y } \beta(p_1, \dots, p_n)$$

Entonces, α y β serán logicamente equivalentes

$$\alpha \equiv \beta$$

si para toda valuación posible (v_1, \dots, v_n) se cumple que:

$$\alpha(v_1, \dots, v_n) = \beta(v_1, \dots, v_n)$$

Ejemplo: Para las fórmulas $p \wedge (q \vee r)$ y $(p \wedge q) \vee (p \wedge r)$ se tiene la siguiente tabla de verdad:

p	q	r	$p \wedge (q \vee r)$	$(p \wedge q) \vee (p \wedge r)$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Como ambas formulas son equivalentes para toda valuación, entonces:

$$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$$

4.2 Equivalencias útiles

1. Conmutatividad:

$$p \wedge q \equiv q \wedge p$$

$$p \vee q \equiv q \vee p$$

2. Asociatividad:

$$p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$$

$$p \vee (q \vee r) \equiv (p \vee q) \vee r$$

3. Idempotente:

$$p \wedge p \equiv p$$

$$p \vee p \equiv p$$

4. Doble negación:

$$\neg \neg p \equiv p$$

5. Distributividad:

$$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$$

$$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$$

6. De Morgan:

$$\neg(p \wedge q) \equiv \neg p \vee \neg q$$

7. Implicación:

$$p \rightarrow q \equiv \neg p \vee q$$

$$p \rightarrow q \equiv \neg q \rightarrow \neg p$$

$$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$$

8. Absorción:

$$p \vee (p \wedge q) \equiv p$$

$$p \wedge (p \vee q) \equiv p$$

9. Identidad:

$$p \vee 0 \equiv p$$

$$p \wedge 1 \equiv p$$

10. Dominación:

$$p \wedge 0 \equiv 0$$

$$p \vee 1 \equiv 1$$

5 Operadores Generalizados

Debido a que \vee y \wedge son operadores asociativos, podemos escribir las siguientes generalizaciones

$$\bigvee_{i=1}^n p_i \equiv p_1 \vee p_2 \vee \dots \vee p_n$$

$$\bigwedge_{i=1}^n p_i \equiv p_1 \wedge p_2 \wedge \dots \wedge p_n$$

Además, podemos saltarnos los parentesis al momento de escribir estas operaciones. Por ejemplo:

$$(p_1 \vee p_2) \vee p_3 \equiv p_1 \vee (p_2 \vee p_3) \equiv p_1 \vee p_2 \vee p_3$$

$$(p_1 \wedge p_2) \wedge p_3 \equiv p_1 \wedge (p_2 \wedge p_3) \equiv p_1 \wedge p_2 \wedge p_3$$

6 Formas normales

Primero, consideremos que un *literal* es una variable proposicional o la negación de una variable.

6.1 Forma Normal Disyuntiva (DNF)

Una formula α está en DNF si es una disyunción de conjunciones de literales.

$$\alpha = \beta_1 \vee \beta_2 \vee \dots \vee \beta_k$$

donde $\beta_i = (l_{i_1} \wedge \dots \wedge l_{i_{k_i}})$ y $l_{i_1}, \dots, l_{i_{k_i}}$ son literales.

Por si esta forma de anotarlo es muy complicada de entender, en palabras más simples, nos referimos a que α está en DNF si es que es una formula proposicional tal que este compuesta por disyunciones de otras formulas, las cuales son conjunciones de variables proposicionales.

Ejemplo:

$$(p \wedge \neg q) \vee (\neg p \wedge p \wedge s) \vee (r \wedge \neg s)$$

6.2 Forma Normal Conjuntiva (CNF)

Una formula α está en CNF si es una conjuncion de disyunciones de literales.

$$\alpha = \beta_1 \wedge \beta_2 \wedge \dots \wedge \beta_k$$

donde $\beta_i = (l_{i_1} \vee \dots \vee l_{i_{k_i}})$ y $l_{i_1}, \dots, l_{i_{k_i}}$ son literales.

Al final, la idea de la CNF es algo así como una “forma inversa” de la DNF, ya que se invierte que es lo que se encuentra en disyunción y lo que se encuentra en conjunción.

Ejemplo:

$$(p \vee \neg q) \wedge (\neg p \vee p \vee s) \wedge (r \vee \neg s)$$

6.3 Formas normales y Equivalencia lógica

Tenemos el siguiente teorema:

1. Toda formula α es lógicamente equivalente a una formula en DNF.
2. Toda formula α es lógicamente equivalente a una formula en CNF.

Debido a las limitadas herramientas de demostración que tenemos por el momento, la demostración será escrita a futuro en esta parte del resumen. En caso de estar estudiando con este resumen, no considere la demostración hasta tener las herramientas suficientes como para demostrar.

7 Consecuencia Lógica

Sea $\Sigma = \{\alpha_1, \dots, \alpha_m\}$ un conjunto de formulas con variables p_1, \dots, p_n . Diremos que α es *consecuencia lógica* de Σ si, y solo si, para toda valuación v_1, \dots, v_n se tiene que

$$\text{si } \left[\bigwedge_{i=1}^m \alpha_i \right](v_1, \dots, v_n) = 1, \text{ entonces } \alpha(v_1, \dots, v_n) = 1$$

Esto se denota como $\Sigma \models \alpha$ (leído como α es consecuencia lógica de Σ)

Posiblemente no quede del todo claro el significado de esta formula, pero en palabras, lo que queremos decir es que si tenemos una valuación, tal que al aplicarla a toda formula presente en el conjunto retorne 1, entonces si una formula es consecuencia lógica del conjunto, esta debe también retornar 1.

Si lo intentamos ver con una tabla de verdad, a modo de ejemplo, una consecuencia lógica se vería de la siguiente forma:

v_1	\dots	v_n	α_1	α_2	\dots	α_m	α
\dots	\dots	\dots	1	1	\dots	0	1
\dots	\dots	\dots	1	1	\dots	1	1
\dots	\dots	\dots	0	0	\dots	1	0

En donde tenemos una fila marcada en gris, en donde podemos ver que todas las formulas del conjunto Σ son iguales a 1 y tambien que α es 1. Si esto se cumple y no sucede que tenemos todas las formulas de la izquierda con unos, y el alpha de la derecha con un 0, entonces tenemos consecuencia lógica.

Otro ejemplo:

v_1	\dots	v_n	α_1	α_2	\dots	α_m	α
\dots	\dots	\dots	1	1	\dots	0	1
\dots	\dots	\dots	1	1	\dots	1	0
\dots	\dots	\dots	0	0	\dots	1	0

En este caso, ya no hay consecuencia lógica, debido a que no se cumple la condición establecida antes. La fila marcada en gris, tiene un 0 en la formula final, lo que nos dice que no es consecuencia lógica.

NOTA: Si tenemos un conjunto Σ tal que es imposible obtener una fila con solo unos, entonces **cualquier cosa** puede ser consecuencia lógica de Σ . ¡Usar con sabiduría para demostraciones!

7.1 Consecuencias lógicas clásicas

7.1.1 Modus ponens

$$\{p, p \rightarrow q\} \models q$$

p	q	p	$p \rightarrow q$	q
0	0	0	1	0
0	1	0	1	0
1	0	1	0	0
1	1	1	1	1

7.1.2 Modus tollens

$$\{\neg q, p \rightarrow q\} \models \neg p$$

p	q	p	$p \rightarrow q$	q
0	0	1	1	1
0	1	0	1	1
1	0	1	0	0
1	1	0	1	0

7.1.3 Resolución

$$\{p \vee q, \neg q \vee r\} \models p \vee r$$

p	q	r	$p \vee q$	$\neg q \vee r$	$p \vee r$
0	0	0	0	1	0
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	1	1

7.2 Trucos de consecuencia lógica

1. $\{1\} \models \alpha$, entonces α es una tautología
2. Si α es una contradicción, entonces $\{\alpha\} \models \beta$
3. Si $\Sigma \models \alpha$, entonces $\Sigma \cup \{\beta\} \models \alpha$ para todo β
4. Si $\Sigma \cup \{\alpha\} \models \beta$ y $\Sigma \models \alpha$, entonces $\Sigma \models \beta$

7.3 Más consecuencias lógicas clásicas

A continuación, vamos a mostrar más consecuencias lógicas, sumando a la lista en 7.1

1. Modus ponens: $\{p, p \rightarrow q\} \models q$
2. Modus tollens: $\{p \vee q, \neg q \vee r\} \models p \vee r$
3. Silogismo: $\{p \rightarrow q, q \rightarrow r\} \models p \rightarrow r$
4. Silogismo disyuntivo: $\{p \vee q, \neg p\} \models q$
5. Conjunción: $\{p, q\} \models p \wedge q$
6. Simplificación Conjuntiva: $\{p \wedge q\} \models p$
7. Amplificación Disyuntiva: $\{p\} \models p \vee q$
8. Demostración Condicional: $\{p \wedge q, p \rightarrow (q \rightarrow r)\} \models r$
9. Demostración por casos: $\{p \rightarrow r, q \rightarrow r\} \models (p \vee q) \rightarrow r$

7.4 Composición y Consecuencia lógica

7.4.1 Definición

Considerando un conjunto $\Sigma = \{\alpha_1(p_1, \dots, p_n), \dots, \alpha_m(p_1, \dots, p_n)\}$ y β_1, \dots, β_n como formulas proposicionales.

Una composición $\Sigma(\beta_1, \dots, \beta_n)$, consiste en el conjunto resultante de evaluar cada formula de Σ con β_1, \dots, β_n . En otras palabras:

$$\Sigma(\beta_1, \dots, \beta_n) = \{\alpha_1(\beta_1, \dots, \beta_n), \dots, \alpha_m(\beta_1, \dots, \beta_n)\}$$

7.4.2 Teorema

Sea Σ un conjunto de formulas (similar al de antes), y $\alpha, \beta_1, \dots, \beta_n$, formulas proposicionales. Si $\Sigma \models \alpha$, entonces $\Sigma(\beta_1, \dots, \beta_n) \models \alpha(\beta_1, \dots, \beta_n)$.

8 Satisfacibilidad

8.1 Satisfacción de un conjunto de formulas

Se dice que una formula proposicional $\alpha(p_1, \dots, p_n)$ es satisfacible si existe una valuación v_1, \dots, v_n tal que

$$\alpha(v_1, \dots, v_n) = 1$$

Un conjunto $\Sigma = \{\alpha_1, \dots, \alpha_m\}$ con variables p_1, \dots, p_n se dice que es satisfacible si existe una valuación v_1, \dots, v_n tal que

$$[\bigwedge_{i=1}^m \alpha_i](v_1, \dots, v_n) = 1$$

Si un conjunto o formula no es satisfacible, entonces se dice que es inconsistente.

8.2 Consecuencia lógica vs satisfacibilidad

Teorema: $\{\alpha_1, \dots, \alpha_m\} \models \alpha$ si y solo si $\{\alpha_1, \dots, \alpha_m, \neg\alpha\}$ es inconsistente.

8.3 Satisfacibilidad y representación de problemas

Problema: Dada una formula α , verificar si es o no satisfacible.

¿Como podemos resolver este problema? Si bien es posible ir probando todas las posibles valuaciones para verificar, es un proceso largo y poco eficiente. Tristemente, la respuesta a este problema, es que no es posible. No existe ningun otro método al momento de verificar si una formula es o no es consistente.

9 Lógica de Predicados

Hasta ahora, se ha trabajado unicamente con Lógica Proposicional. Esta funciona bien, pero tiene algunas limitaciones que vuelven imposible modelar algunas situaciones.

- No tiene objetos. Solo se pueden usar proposiciones.
- No tiene predicados.
- No tiene cuantificadores.

9.1 ¿Y qué tiene la Lógica de Predicados?

La lógica de predicados es una parte de la lógica de primer orden. La lógica de predicados nos va a permitir expresar ciertas estructuras las cuales no eramos capaces de expresar usando la lógica proposicional.

9.2 Predicados

Un predicado consiste en una proposicion abierta. El valor de verdad de un predicado dependerá del valor usado en la valuación. Generalmente, estos se simbolizan usando letras mayúsculas (Ej.: $P(x)$)

Ejemplos de predicados:

- $P(x) := x$ es par
- $R(x) := x$ es primo
- $M(x) := x$ es mortal

9.2.1 Predicados n-arios

Los predicados n-arios consisten en predicados los cuales usan más de una variable para verificar su valor de verdad.

Ejemplo: $O(x, y) := x \leq y$. Si $x = 2$ e $y = 3$, entonces $O(2, 3) = 1$

9.2.2 Dominio de predicado

Todo predicado está restringido a un cierto dominio de evaluación. Esto significa que sus valores de verdad solo se pueden evaluar cuando las variables que se usan en el predicado están dentro del dominio designado. Ej.: $O(x, y) := x \leq y$ sobre \mathbb{N}

9.2.3 Predicado 0-ario / Predicado degenerado

Corresponde a un predicado que no tiene ninguna variable libre. Tiene un valor de verdad el cual es totalmente independiente de su valuación.

9.2.4 Predicados compuestos

Un predicado compuesto corresponde a la combinación de diversos predicados básicos, usando distintos operadores para mezclarlos en la sentencia, o la cuantificación universal o existencial de algún predicado (Véase sección 9.3). Todos los predicados deben tener el mismo dominio.

9.3 Cuantificadores

9.3.1 Cuantificador Universal

Consideremos $P(x, y_1, \dots, y_n)$ un predicado compuesto de dominio D . El cuantificador universal corresponde a

$$P'(y_1, \dots, y_n) := \forall x. P(x, y_1, \dots, y_n)$$

donde x es la variable cuantificada e y_1, \dots, y_n son las variables libres.

Si para cierta valuación se cumple que $P'(\dots) = 1$ significa que para toda variable libre se cumple lo especificado anteriormente.

Ejemplo

$$O(x, y) := x \leq y \text{ sobre } \mathbb{N}$$

$$O'(y) := \forall x. O(x, y)$$

$$O'(2) = \forall x. O(x, 2) = 0$$

Podemos notar que para $y = 2$, no se cumple para todo valor de x lo dictado en el predicado $O(x, y)$

$$O''(x) := \forall y. O(x, y)$$

$$O''(0) := \forall y. O(0, y) = 1$$

Si $x = 0$, debido a que estamos considerando a los naturales, entonces para todo valor de y se cumple siempre el predicado.

9.3.2 Cuantificador Existencial

Consideremos $P(x, y_1, \dots, y_n)$ un predicado compuesto de dominio D . El cuantificador existencial corresponde a

$$P'(y_1, \dots, y_n) := \exists x. P(x, y_1, \dots, y_n)$$

donde x es la variable cuantificada e y_1, \dots, y_n son las variables libres.

Si para cierta valuación se cumple que $P'(\dots) = 1$ significa que para alguna combinación de variables libres se cumple lo especificado anteriormente.

Ejemplo

$$O(x, y) := x \leq y \text{ sobre } \mathbb{N}$$

$$O'(y) := \exists x. O(x, y)$$

$$O'(2) = \exists x. O(x, 2) = 1$$

9.4 Interpretaciones

En algunos casos, puede ocurrir que algún predicado o formula, dependiendo de ciertas condiciones, sea verdadero(a) o falso(a). Debido a esto, vamos a definir las *interpretaciones*.

Para comenzar, es importante destacar que desde ahora diremos que $P(x_1, \dots, x_n)$ es un símbolo de predicado.

Una interpretación \mathcal{I} para símbolo de predicado P_1, \dots, P_m se compone de

- Un dominio $\mathcal{I}(\text{dom})$
- Para cada símbolo P_i , un predicado $\mathcal{I}(P_i)$

Sea $\alpha(x_1, \dots, x_n)$ una formula y \mathcal{I} una interpretación de los símbolos en α . Se dice que la interpretación \mathcal{I} satisface α sobre a_1, \dots, a_n en $\mathcal{I}(\text{dom})$, expresado como

$$\mathcal{I} \models \alpha(a_1, \dots, a_n)$$

si $\alpha(a_1, \dots, a_n)$ es verdadero al evaluar cada símbolo en α según \mathcal{I} . En el caso que \mathcal{I} no logre satisfacer a α , se anotará como

$$\mathcal{I} \not\models \alpha(a_1, \dots, a_n)$$

Ejemplo

$$\mathcal{I}_1(\text{dom}) := \mathbb{N}$$

$$\mathcal{I}_1(P) := x \text{ es par}$$

$$\mathcal{I}_1(O) := x < y$$

$$\alpha(x) := \exists y. P(y) \wedge O(x, y)$$

$$\mathcal{I}_1 \models \alpha(1) := \exists y. y \text{ es par} \wedge 1 < y$$

10 Equivalencia lógica en Lógica de Predicados

Se tiene $\alpha(x_1, \dots, x_n)$ y $\beta(x_1, \dots, x_n)$ dos oraciones en lógica de predicados (no tienen variables libres). α y β serán lógicamente equivalentes, escrito como:

$$\alpha \equiv \beta$$

si para toda interpretación \mathcal{I} y para todo a_1, \dots, a_n se cumple que:

$$\mathcal{I} \models \alpha(a_1, \dots, a_n) \text{ si, y solo si, } \mathcal{I} \models \beta(a_1, \dots, a_n)$$

En otras palabras, funciona practicamente igual a la equivalencia lógica en lógica proposicional.

10.1 Equivalencias lógicas

Todas las equivalencias de lógica proposicional aplican aquí. Simplemente se tienen que cambiar las variables proposicionales por predicados de lógica de predicados. Aquí se muestra un ejemplo:

Conmutatividad (para operador \wedge)

En lógica proposicional	En lógica de predicados
$p \wedge q \equiv q \wedge p$	$\alpha \wedge \beta \equiv \beta \wedge \alpha$

Ahora, además de estas equivalencias, nos encontraremos con algunas nuevas

1. $\neg \forall x. \alpha \equiv \exists x. \neg \alpha$
2. $\neg \exists x. \alpha \equiv \forall x. \neg \alpha$
3. $\forall x. (\alpha \wedge \beta) \equiv (\forall x. \alpha) \wedge (\forall x. \beta)$
4. $\forall x. (\alpha \vee \beta) \equiv (\forall x. \alpha) \vee (\forall x. \beta)$