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 mayusculas (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
$\rightarrow$	Condicional	$P \rightarrow Q$	Si P, entonces Q
$\leftrightarrow$	Bicondicional	$P \leftrightarrow Q$	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

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

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

$$(P \rightarrow Q) \leftrightarrow (P \wedge Q)$$

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:

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

$$(P \to Q) \leftrightarrow (P \wedge Q) \text{ con P = 1 y Q = 0}$$
$$(1 \to 0) \leftrightarrow (1 \wedge 0)$$
$$0 \leftrightarrow 0$$
$$1$$

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
$\to$	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:

$$\alpha(p, q, r) := p \wedge (q \to r)$$
$$\beta(p, q) := (p \wedge \neg q) \vee (\neg p \wedge 1)$$

4 Equivalencia Lógica

4.1 Definición

Si tenemos dos formulas proposicionales con las mismas variables proposicionales

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

Entonces, α y β serán logicamente equivalentes

$$\alpha \equiv \beta$$

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

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

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:

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

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
$$\bigvee_{i=1}^{n} p_i \equiv p_1 \vee p_2 \vee \ldots \vee p_n$$
$$\bigwedge_{i=1}^{n} p_i \equiv p_1 \wedge p_2 \wedge \ldots \wedge p_n$$
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.
$$\alpha = \beta_1 \vee \beta_2 \vee \ldots \vee \beta_k$$
donde $\beta_i = (l_{i_1} \wedge \ldots \wedge l_{i_{k_i}})$ y $l_{i_1}, \ldots, l_{i_{k_i}}$ 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.

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

donde $\beta_i = (l_{i_1} \vee \ldots \vee l_{i_{k_i}})$ y $l_{i_1}, \ldots, l_{i_{k_i}}$ 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 $\Sigma = \{\alpha_1, \ldots, \alpha_m\}$ un conjunto de formulas con variables $p_1, \ldots, p_n$. Diremos que α es *consecuencia lógica* de Σ si, y solo si, para toda valuación $v_1, \ldots, v_n$ se tiene que

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

Esto se denota como $\Sigma \models \alpha$ (leido 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	$\cdots$	v_n	α_1	α_2	$\cdots$	α_m	α
$\cdots$	$\cdots$	$\cdots$	1	1	$\cdots$	0	1
$\cdots$	$\cdots$	$\cdots$	1	1	$\cdots$	1	1
$\cdots$	$\cdots$	$\cdots$	0	0	$\cdots$	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	$\cdots$	v_n	α_1	α_2	$\cdots$	α_m	α
$\cdots$	$\cdots$	$\cdots$	1	1	$\cdots$	0	1
$\cdots$	$\cdots$	$\cdots$	1	1	$\cdots$	1	0
$\cdots$	$\cdots$	$\cdots$	0	0	$\cdots$	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 sabiduria para demostraciones!

7.1 Consecuencias lógicas clásicas

7.1.1 Modus ponens

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

p	q	p	$p \to q$	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

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

p	q	p	$p \to q$	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: $\{p, p \to q\} \models q$
2. Modus tollens: $\{p \vee q, \neg q \vee r\} \models p \vee r$
3. Silogismo: $\{p \to q, q \to r\} \models p \to r$
4. Silogismo disyuntivo: $\{p \vee q, \neg p\} \models q$

5. Conjunción: $\{p, q\} \models p \land q$
6. Simplificación Conjuntiva: $\{p \land q\} \models p$
7. Amplificación Disyuntiva: $\{p\} \models p \lor q$
8. Demostración Condicional: $\{p \land q, p \to (q \to r)\} \models r$
9. Demostración por casos: $\{p \to r, q \to r\} \models (p \lor q) \to r$

7.4 Composición y Consecuencia lógica

7.4.1 Definición

Considerando un conjunto $\Sigma = \{\alpha_1(p_1, \ldots, p_n), \ldots, \alpha_m(p_1, \ldots, p_n)\}$ y $\beta_1, \ldots, \beta_n$ como formulas proposicionales.
Una composición $\Sigma(\beta_1, \ldots, \beta_n)$, consiste en el conjunto resultante de valuar cada formula de Σ con $\beta_1, \ldots, \beta_n$. En otras palabras:

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

7.4.2 Teorema

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

8 Satisfacibilidad

8.1 Satisfacción de un conjunto de formulas

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

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

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

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

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

8.2 Consecuencia lógica vs satisfacibilidad

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