¿Qué es la lógica?

- La definición del Diccionario General de la Lengua Española dice:

  > Disciplina que estudia los principios formales del conocimiento humano, es decir, las formas y las leyes más generales del pensamiento humano considerado puramente en sí mismo, sin referencia a los objetos. Los problemas principales de la lógica son las doctrinas del concepto, del juicio, del silogismo y del método.

- Esfuerzos por modelar estas "leyes del pensamiento humano". Han existido desde la antigüedad.

- Como ejemplo podemos tomar los silogismos:

  ➡Todos los perros son mamíferos.
  ➡Todos los mamíferos son animales
  Podemos concluir que:
  ➡Todos los perros son animales.

  ➡Ninguna gaviota es un traductor.
  ➡Algunas arañas son gaviotas.

Podemos concluir que:
➦Algunas arañas no son traductores

- Los silogismos datan de la época de Aristóteles (350 AC aprox).

- Nosotros nos preocuparemos de la lógica matemática.

- La lógica es una *disciplina matemática* relativamente nueva (100 años aprox.)

- ¿Es el razonamiento humano lógico?

  Es común que mucha gente realice razonamientos incorrectos como el siguiente:

  ➦Si Daniela tiene prueba, estudia toda la tarde
  ➦Daniela ha estudiado toda la tarde
  Entonces:
  ➦Daniela tiene prueba.

Por qué es bueno saber lógica

- Porque parte esencial del razonamiento matemático.

- Muchas otras disciplinas usan lógica: Psicología (ej: *Wason's selection task*), Filosofía, Física, Linguística.

- La lógica es esencial en ciencia de la computación.Algunos usos:
  - Programación en general.
  - Modelación Formal de algoritmos, verificación de propiedades.
  - Modelación Formal de máquinas.
  - Representación formal del conocimiento y razonamiento.
  - Bases de Datos.

- Además, algunas lógicas son implementables (demostradores mecánicos de teoremas).

- Existen lenguajes de programación basados en lógica (PROLOG).

- Procesamiento de Lenguaje Natural.

Algunas Lógicas

Hay muchas...

Lógicas para razonamiento matemático : proposicional, primer orden, segundo orden.

Lógicas Descriptivas , usadas en representación de conocimiento y Web Semántico.

Ejemplo: Para el dominio que representa a las familias,

$\mathtt{Padres} \sqcap \mathtt{Hombre} \sqcap \forall \mathtt{Hijo}.\mathtt{Mujer}$

Puede representar a la clase de padres varones que sólo tienen hijas mujeres.

Lógicas para razonamiento con sentido común . Ej: *default logics*.

Lógica Difusa . Se pierde la noción de lo verdadero y lo falso.Aparecen nociones intermedias. Ha tenido gran éxito en la programación de controladores automáticos.

Lógicas Modales . Estas lógicas tienen por objetivo expresar nociones de necesidad y posibilidad. Por ejemplo:

$\Box p$ significa "necesariamente p".
$\Diamond p$ significa "posiblemente p". Observemos que

$\Diamond p$ es equivalente a $\neg\Box\neg p$.

Lógicas multivaluadas : Se usan más de dos valores de verdad, para describir conceptos más allá de lo verdadero y lo falso.

Qué haremos en este curso

El objetivo es que estudiemos algunas lógicas y veamos cómo éstas se relacionan con la Ciencia de la Computación.

Temas:

1. Lógica Proposicional.
2. Demostración Mecánica de Teoremas.
3. Lógica de Primer Orden.
4. Computabilidad y Complejidad Computacional.
5. Teorías.
6. Otras Lógicas.

Aspectos de Evaluación

- Tres interrogaciones, un examen.

- Tareas (alrededor de 7). La nota final se calcula como

$$NF = 0.8PE + 0.2PT$$

, donde PT es el promedio de tareas y PE se calcula como:

$$PE = \frac{I_1 + I_2 + I_3 + 2EX - \min(I_1, I_2, I_3, EX)}{4},$$

en caso que $PE \geq 3.5$ y $PT \geq 3.5$. En caso contrario, la nota final corresponderá al mínimo entre PE y PT:

➡Una tarea computacional podrá valer como dos tareas escritas.

Sintaxis versus Semántica

- La *sintaxis* se refiere a la forma en que escribimos el "lenguaje objeto".
- Por ejemplo, si el lenguaje objeto es el lenguaje de programación C, entonces la sintaxis del lenguaje indica que el siguiente programa es correcto:

```
i=0;
while (i < 10) {
   printf("%d",i);
   i++;
}
```

- Por otro lado, la *semántica* tiene por objetivo identificar qué está expresando el lenguaje objeto. Usualmente esto se realiza en un *metalenguaje*.
- En el ejemplo la semántica en castellano del programa es, en términos resumidos, una iteración que imprime los números del 0 al 9.
- En el caso de la lógica también haremos esta distinción.

- Por ejemplo, seremos capaces de decir que la fórmula de lógica proposicional

$$(p \wedge q)$$

tiene una sintaxis adecuada y que es verdadera cuando p y q lo son en forma simultánea.

Lógica Proposicional (LP)

- Tal como su nombre lo indica, ésta es una lógica para representar proposiciones.
- Una proposición en el castellano es, por ejemplo,

  "*El cielo es azul*"

  esta es una proposición porque es un **hecho**. Además este hecho es verdadero.
- Las proposiciones en LP se representan con letras. Usualmente se usan las letras p, q, r y s, posiblemente con subíndices. Éstas se denominan *variables proposicionales*.

Sintaxis para lógica Proposicional

- En LP se distinguen los siguientes elementos:

  1. Constantes: $\top$, $\bot$.
  2. Conectivos unarios: $\neg$.
  3. Conectivos binarios: $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$
  4. Símbolos de puntuación: (,).
  5. Un conjunto P, posiblemente infinito, de variables proposicionales.

- Mediante una combinación de estos elementos es posible definir cualquier lenguaje de la lógica proposicional.

- Dado un conjunto fijo P de variables, es posible definir un lenguaje proposicional $L(P)$, que contiene todas las fórmulas posibles a través de la siguiente definición inductiva:

Definición 1. *El lenguaje $L(P)$ está formado por fórmulas. Una fórmula es:*

- *una constante o un elemento de P (también llamadas fórmulas atómicas).*
- *Si φ es una fórmula, entonces $\neg\varphi$ también es una fórmula.*
- *Si φ y ψ son ambos fórmulas, entonces*

$$(\varphi \wedge \psi)$$
$$(\varphi \vee \psi)$$
$$(\varphi \rightarrow \psi)$$
$$(\varphi \leftrightarrow \psi)$$

También son formulas. Normalmente esto se anota como

$$(\varphi * \psi)$$

. Y se hace explícito que $*$ representa a cualquier conectivo lógico binario.

- Dada la naturaleza de la definición, toda propiedad que queramos demostrar de las fórmulas, deberá ser hecha de manera inductiva.

- Ejercicio: Demuestre que $((p \wedge q) \rightarrow q)$ es una fórmula.

Convenciones

- Usualmente querremos evitar el uso de paréntesis cuando éstos no sean necesarios. Es así como preferiremos escribir $p \wedge q \wedge r$ en vez de $((p \wedge q) \wedge r)$.

- Supondremos desde ahora que si una fórmula carece de paréntesis se interpretará usando la siguiente convención: Se asocia por la izquierda, tomando en cuenta al conectivo $\neg$ en primera prioridad, al conectivo $\wedge$ en segunda prioridad, al conectivo $\vee$ con tercera, y finalmente a los conectivos $\rightarrow$ y $\leftrightarrow$.

- De esta manera, la fórmula

$$p \wedge q \vee s \rightarrow p \vee s$$

corresponde a la siguiente fórmula:

$$(((p \wedge q) \vee s) \rightarrow (p \vee s))$$

Definiciones y demostraciones que involucran a fórmulas

- El principio de inducción es usado con dos fines:
  - Definir conceptos/funciones asociados a fórmulas.
  - Demostrar propiedades generales del lenguaje.

- En términos simples, para definir un concepto (o función) sobre las fórmulas, hay que definir todos los casos (base e inductivo).

- Formalmente, hay que hacer lo siguiente para definir el valor de una función f para todas las fórmulas de $L(P)$.

  Caso base: se define el valor de f para las fórmulas atómicas.

  Pasos inductivos:
  - Se define el valor de $f(\neg\varphi)$ en términos del valor de $f(\varphi)$
  - El valor de $f((\varphi * \psi))$ es especificado en términos $f(\varphi)$ y $f(\psi)$, donde $*$ es un conectivo binario.

- Esta forma de definir se conoce como principio de recursión estructural.

Ejemplo de Inducción

- Ejemplo: la función $variables(\varphi)$ que cuenta el número de variables proposicionales en φ se puede definir de la siguiente manera:

  Caso base:

  $$variables(\top) = 0$$
  $$variables(\bot) = 0$$
  $$variables(p) = 1 \qquad \text{(con } p \in P\text{)}$$

  Pasos inductivos:

  $$variables(\neg\varphi) = variables(\varphi)$$
  $$variables(\varphi * \psi) = variables(\varphi) + variables(\psi)$$

Semántica de la LP

- En la lógica proposicional existen dos valores posibles para fórmulas: *verdadero* y *falso*

- La semántica nos debe proveer tres cosas:

  1. Significado de las fórmulas.
  2. Noción de verdad.
  3. Noción de consecuencia lógica.

- El punto 1 pasa por una especificación adecuada de parte del modelador de la teoría.

  Así, podemos decir que p significa "hoy sale el sol a las 7:12 am". Por otro lado, esto puede quedar claro en el mismo nombre de la variable.

  Por ejemplo $saleSol712am$ puede ser una variable proposicional que claramente representa el mismo hecho.

- Para definir la noción de verdad debemos, antes, definir el concepto de *valuación* o *asignación de verdad*.

Definición 2. *Una valuación es una función $\sigma : P \to \{0, 1\}$, que sirve para asignar un valor de verdad a una variable.*

Nota: Aquí estamos suponiendo que 0 representa al valor de verdad $falso$ y 1 a $verdadero$.

Ejemplo: si $P = \{p, q\}$, entonces la función σ_1 es una valuación definida por:

$$\sigma_1(p) = 1 \tag{1}$$

$$\sigma_1(q) = 0 \tag{2}$$

- Por simplicidad usaremos el entero 1 para referirnos a verdadero y 0 para falso.
- Nótese que para un conjunto P hay $2^{|P|}$ funciones de valuación distintas.
- Esta definición aún no es suficiente, pues no tenemos una definición clara acerca del valor de verdad de las fórmulas del lenguaje.

Extendiendo la Semántica a Fórmulas

- Dada una asignación $\sigma : P \to \{0,1\}$, extenderemos la función a

$$\hat{\sigma} : L(P) \to \{0,1\}.$$

  Si φ es una fórmula proposicional, entonces:

  - Si $\varphi \in P$ entonces $\hat{\sigma}(\varphi) = \sigma(\varphi)$.
  - Si $\varphi = \top$ entonces $\hat{\sigma}(\varphi) = 1$.
  - Si $\varphi = \bot$ entonces $\hat{\sigma}(\varphi) = 0$.
  - Si $\varphi = \neg\psi$ entonces $\hat{\sigma}(\varphi) = 1 - \hat{\sigma}(\psi)$.
  - Si $\varphi = \psi \wedge \chi$ entonces $\hat{\sigma}(\varphi) = \min(\hat{\sigma}(\psi), \hat{\sigma}(\chi))$.
  - Si $\varphi = \psi \vee \chi$ entonces $\hat{\sigma}(\varphi) = \max(\hat{\sigma}(\psi), \hat{\sigma}(\chi))$.
  - Si $\varphi = \psi \to \chi$ entonces, si $\hat{\sigma}(\psi) = 0$ entonces $\hat{\sigma}(\varphi) = 1$, en caso contrario $\hat{\sigma}(\varphi) = \sigma(\chi)$.
  - Si $\varphi = \psi \leftrightarrow \chi$ entonces $\hat{\sigma}(\varphi) = 1$ si $\hat{\sigma}(\psi) = \hat{\sigma}(\chi)$ y $\hat{\sigma}(\varphi) = 0$ en caso contrario.

- Por simplicidad, desde ahora en adelante, utilizaremos σ en vez de $\hat{\sigma}$. La semántica estará dada por el caso.

- Esta misma definición se puede resumir a través de una tabla de verdad:

φ	ψ	$\neg\varphi$	$\varphi \wedge \psi$	$\varphi \vee \psi$	$\varphi \rightarrow \psi$	$\varphi \leftrightarrow \psi$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

- **Definición 3.** *Una formula φ es equivalente a otra fórmula ψ si para toda valuación σ, $\sigma(\varphi) = \sigma(\psi)$*

- Ejercicios:

  Demuestre que las siguientes fórmulas son equivalentes:

  - $(\varphi \rightarrow \psi)$ y $(\neg\varphi \vee \psi)$.
  - $\neg(\varphi \wedge \psi)$ y $(\neg\varphi \vee \neg\psi)$.
  - $(\varphi \leftrightarrow \psi)$ y $((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))$.

Otros Conectivos

- Usualmente, en LP se utilizan otros conectivos lógicos que están descritos en función de los que ya definimos.

- En la siguiente tabla se muestran los más comunes.

Símbolo	**Uso**	**Equivalencia**	**Descripción**
$\leftarrow$	$a \leftarrow b$	$b \rightarrow a$	condicional reverso
$\downarrow$	$a \downarrow b$	$\neg(a \vee b)$	conocido como NOR, "ni a ni b"
$\mid$	$a \mid b$	$\neg(a \wedge b)$	conocido como NAND, "a y b no son simultáneamente verdaderos"
$\otimes$	$a \otimes b$	$(a \vee b) \wedge \neg(a \wedge b)$	conocido como XOR, "o bien a o bien b, pero no ambos"

- **Definición 4.** *Un conjunto de conectivos C es **funcionalmente completo** si es posible definir a los conectivos estándar, en función los otros.*

- **Teorema 1.** *El conjunto $\{\neg, \wedge\}$ es funcionalmente completo.*

Demostración: Sabemos que $p \leftrightarrow q$ *es equivalente a* $(p \rightarrow q) \wedge (q \rightarrow p)$ *y que* $(p \rightarrow q)$ *es equivalente a* $(\neg p \vee q)$*, luego sólo nos falta expresar* $\vee$ *en términos de* $\neg$ *y* $\wedge$*.*

En efecto $p \vee q$ *es equivalente a* $\neg(\neg p \wedge \neg q)$*.*

- Ejercicio: Demuestre que $\{\downarrow\}$ es un conjunto funcionalmente completo.

Formas Normales

- Las formas normales son formas sintácticas estándares que pueden cumplir las fórmulas.

- Estudiaremos dos: la $forma\ normal\ conjuntiva$ y la $forma\ normal\ disyuntiva$.

- Veamos, antes, un par de definiciones.

Definición 5. *Un literal es una variable proposicional o una variable proposicional negada, o una constante ($\top$ o $\bot$).*

Definición 6. *Una cláusula es una disyunción de literales, es decir, es de la forma $\bigvee_{i=0}^{n} l_i$, donde cada l_i es un literal.*

Una cláusula dual es una conjunción de literales, es decir, es de la forma $\bigwedge_{i=0}^{n} l_i$, donde cada l_i es un literal.

- Ahora, veamos qué son las formas normales.

Definición 7. *Una fórmula en forma normal conjuntiva (FNC) es una conjunción de cláusulas.*

Ejemplo:

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

- **Definición 8.** *Una fórmula en forma normal disyuntiva (FND) es una disyunción de cláusulas duales.*

  Ejemplo:

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

- ¿Dada una fórmula arbitraria ϕ, podremos construir, en forma mecánica, otra fórmula χ equivalente en alguna de las formas normales? La respuesta es Sí!

Traducción a FND

- Veamos el caso de llevar una fórmula φ cualquiera a forma normal disyuntiva.

  Supongamos que las variables que aparecen en φ son $p_1, p_2, \ldots, p_n$.

  Una posibilidad es hacer lo siguiente:

  1. Hacer una tabla de verdad.
  2. Por cada fila en la cual la fórmula es verdadera generar la conjunción

     $$\bigwedge_{i=0}^{n} l_i,$$

     donde $l_i = p_i$ si en esa fila le corresponde valor 1, y $l_i = \neg p_i$ si en esa fila le corresponde valor 0.
  3. La formula final se arma con la disyunción de las conjunciones generadas en el punto anterior.

- ¿Qué problema tiene este método?

Traducción a FNC

- El método que veremos a continuación tiene especial relevancia en demostración mecánica de teoremas.

- Nuestro objetivo es transformar una fórmula φ en una lista de disyunciones de literales $\langle D_1, D_2, \ldots, D_n \rangle$ tales que

$$\varphi \text{ equivalente a } \bigwedge_{i=1}^{n} D_i.$$

- El algoritmo es el siguiente:

  1. Se comienza con $X := \langle \varphi \rangle$.
  2. Se repite la siguiente iteración: Suponemos que después del paso n, X es una conjunción de disyunciones representada por

$$\langle D_1, \ldots, D_n \rangle$$

.

Si X no está en FNC, se selecciona un D_i que no sea una disyunción de literales y se escoge un miembro N de la fórmula que no sea un literal. Reemplazar N usando las siguientes reglas:

(a) Si N es $\neg\top$, reemplazarlo por $\bot$.
(b) Si N es $\neg\bot$, reemplazarlo por $\top$.
(c) Si N es $\neg\neg Z$, reemplazarlo por Z.
(d) Si N es $(X \vee Y)$, reemplazarlo por $X \vee Y$
(e) Si N es $(X \rightarrow Y)$, reemplazarlo por $\neg X \vee Y$
(f) Si N es $\neg(X \vee Y)$, reemplazarlo por $(\neg X \wedge \neg Y)$
(g) Si N es $\neg(X \wedge Y)$, reemplazarlo por $\neg X \vee \neg Y$
(h) Si N es $\neg(X \rightarrow Y)$, reemplazarlo por $(X \wedge \neg Y)$
(i) Si N es $(X \wedge Y)$, reemplazar la disyunción D_i por otras dos en disyunciones D_i' y D_i'' en las cuales se reemplaza a N por X en D_i' y por Y en D_i''.

- Ejemplo:

  Llevar a FNC la siguiente fórmula

$$(\neg p \wedge (q \wedge r)) \rightarrow (q \wedge \neg r)$$

  Partimos con la lista

$\langle(\neg p \wedge (q \wedge r)) \rightarrow (q \wedge \neg r)\rangle$

Podemos seguir los siguientes pasos:

$$\begin{aligned}
\langle\neg(\neg p \wedge (q \wedge r)) \vee (q \wedge \neg r)\rangle \\
\langle p \vee \neg(q \wedge r) \vee (q \wedge \neg r)\rangle \\
\langle p \vee \neg(q \wedge r) \vee (q \wedge \neg r)\rangle \\
\langle p \vee \neg q \vee \neg r \vee (q \wedge \neg r)\rangle \\
\langle p \vee \neg q \vee \neg r \vee q, p \vee \neg q \vee \neg r \vee \neg r\rangle
\end{aligned}$$

Por lo tanto, la fórmula equivalente es:

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

Fórmulas válidas y satisfacibles

Una fórmula válida es aquélla que es satisfacible por toda valuación σ.

Ejemplo:

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

y

$$\neg q \wedge (p \rightarrow q) \rightarrow \neg p$$

son fórmulas válidas.

A este tipo de fórmulas también se les llama **tautologías**.

Definición 9. *Una fórmula es satisfacible si existe al menos una valuación que la hace verdadera.*

Esta definición se extiende para un conjunto de fórmulas:

Definición 10. *Una conjunto de fórmulas* Σ *es satisfacible si existe al menos una valuación* σ *que hace verdadera a todas las fórmulas del conjunto.*

En este caso diremos que $\sigma \models \Sigma$

Un conjunto de fórmulas que no se puede satisfacer (insatisfacible) se le conoce como $inconsistente$

Consecuencia Lógica

- La consecuencia lógica es el elemento que nos provee la semántica para identificar cuándo, a partir de un conjunto de fórmulas (axiomas), que suponemos son verdaderos, es posible concluir otras fórmulas que no están en esta "base de conocimiento".

- **Definición 11.** *Si Σ es un conjunto de fórmulas en $L(P)$ y φ es una fórmula particular en $L(P)$ entonces decimos que φ es consecuencia lógica de Σ ($\Sigma \models \varphi$) si y sólo si,*

  para cada valuación σ tal que $\sigma \models \Sigma$, entonces $\sigma(\varphi) = 1$.

  Ejemplos:

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

$$\{q, p \to q\} \not\models \{p\}$$

Resultados Acerca de la Consecuencia Lógica

- A partir de la noción de consecuencia lógica se pueden establecer resultados resultados interesantes.

- El primero establece que de una base de conocimiento inconsistente, es posible deducir cualquier fórmula de $L(P)$.

  En efecto, si $\Sigma \subseteq L(P)$ es inconsistente no existe ninguna valuación que haga verdadera, por lo que el antecedente de la "implicación en metalenguaje":

  si, para cada valuación σ tal que $\sigma \models \Sigma$, entonces $\sigma(\varphi) = 1$.

  nunca se cumple, por lo que asumimos este argumento como **verdadero para todo** $\varphi \in L(P)$

Consecuencia Lógica de un Conjunto Vacío de Fórmulas

- Supongamos que se cumple que

$$\{\} \models \varphi$$

- ¿Cuándo se puede dar esto? ¿Cuál es la intuición detrás de esto?

- Serán consecuencia lógica de un conjunto vacío de fórmulas todas aquellas fórmulas que son siempre verdaderas (las tautologías).

LP es una lógica monótona

- En términos resumidos, este resultado significa que, a medida que se agregan fórmulas a una base de conocimiento, los hechos que se concluían a partir de la base original siguen siendo válidos.

- En términos formales:
  - Sean Σ_1 y Σ_2 dos conjuntos de fórmulas tales que $\Sigma_1 \subseteq \Sigma_2$, entonces se cumple que:

$$\text{si } \Sigma_1 \models \varphi, \text{ entonces } \Sigma_2 \models \varphi$$

- Esta propiedad se conoce como $monoton\acute{\imath}a$ de la lógica proposicional.

- ¿Qué consecuencias tiene este teorema?

- ¿En que casos no es deseable esta propiedad?

Demostrando la Monotonía

- Sean Σ_1 y Σ_2 dos conjuntos de fórmulas tales que $\Sigma_1 \subseteq \Sigma_2 \subset L(P)$. Además, sea φ una fórmula

- Supongamos que tenemos una valuación cualquiera, σ, tal que $\sigma \models \Sigma_2$. Como $\Sigma_1 \subseteq \Sigma_2$ también tenemos que $\sigma \models \Sigma_1$. Por definición de consecuencia lógica, $\sigma \models \varphi$. Obtenemos de inmediato que

$$\Sigma_2 \models \varphi.$$

Teorema de Deducción

- El teorema de deducción es fundamental y de uso diario por los seres inteligentes.
- Formalmente dice lo siguiente:

  Sea $\Sigma \subseteq L(P)$ entonces

$$\Sigma \models (\varphi \rightarrow \psi) \text{ si y sólo si } \Sigma \cup \{\varphi\} \models \psi$$

- Ejercicio: demuéstrelo.

Relación entre Consistencia y Consecuencia Lógica

- Esta relación indica lo siguiente:

$$\Sigma \models \varphi \text{ si y sólo si } \Sigma \cup \{\neg\varphi\} \text{ es inconsistente}$$

- Este teorema tiene gran relevancia en demostración mecánica de teoremas.

- Demostración: $(\Rightarrow)$

  - Caso 1: Sea σ tal que $\sigma \models \Sigma$.
    Por definición de consecuencia lógica, $\sigma \models \varphi$, luego $\sigma \not\models \neg\varphi$. Por lo que

$$\sigma \not\models \Sigma \cup \{\neg\varphi\}$$

  - Caso 2(trivial): Sea σ tal que $\sigma \not\models \Sigma$, entonces de inmediato tenemos que

$$\sigma \not\models \Sigma \cup \{\neg\varphi\}$$

$(\Leftarrow)$ Sea σ una valuación cualquiera. Tenemos 2 casos que cumplen con la hipótesis.

- Caso 1: $\sigma \not\models \Sigma$, trivial, puesto que de inmediato tenemos que

$$\Sigma \models \varphi$$

.

- Caso 2: $\sigma \models \Sigma$.
Por hipótesis, $\sigma \not\models \neg\varphi$, con lo cual se obtiene que $\sigma \models \varphi$, y por lo tanto:

$$\Sigma \models \varphi$$

.

Un ejemplo de Consecuencia Lógica

- Suponga la siguiente situación:

  En una cierta isla hay individuos de dos clases: aquéllos que siempre dicen la verdad, y aqu ellos que siempre mienten. Usted llega a esta isla y se encuentra con tres habitantes A, B y C. Le pregunta a A "¿Usted dice la verdad o miente?" A balbucea en un idioma desconocido para usted.
  Luego le pregunta a B "¿que es lo que A dijo?".
  B responde, "A dijo que él es un mentiroso".
  C agrega, "No le creas a B, porque miente!".
  ¿Qué se puede decir sobre A, B y C?

- Solución: Ejercicio.

Teorema de Compacidad (o finitud)

- Usaremos este teorema para demostrar otras propiedades interesantes en el futuro.

- El teorema de compacidad dice lo siguiente:

  Un conjunto Σ de fórmulas de LP es contradictorio ssi tiene un subconjunto finito que es contradictorio.

- Una forma alternativa de ver esto es:

  Un conjunto Σ de fórmulas de LP es satisfacible ssi todo subconjunto finito de éste que es satisfacible.

- **Demostración:** La parte $\Rightarrow$ es clara. ($\Leftarrow$) Es relevante sólo el caso en que Σ es un conjunto infinito. Sea Σ un conjunto satisfacible finitamente.

  y una enumeración de las fórmulas de $L(P)$[1] $\alpha_1, \alpha_2, \ldots,$

  Construimos una extensión de Σ de la siguiente manera:

[1] Una enumeración está compuesta por el conjunto de **todas** las fórmulas proposicionales numeradas con un natural

$$\Delta_0 = \Sigma$$

$$\Delta_n = \begin{cases} \Delta_{n-1} \cup \{\alpha_n\} & \text{si este conjunto es satisfacible finitamente} \\ \Delta_{n-1} \cup \{\neg\alpha_n\} & \text{en otro caso} \end{cases}$$

Sea Δ la unión de todos estos conjuntos. Es decir,

$$\Delta = \bigcup_n \Delta_n$$

De esta manera, para cualquier fórmula $\varphi \in L(P)$, $\varphi \in \Delta$ o $\neg\varphi \in \Delta$.

Ahora basta que que construyamos una asignación de verdad tal que para toda fórmula del tipo p ($p \in P$) en Δ:

$$\sigma(p) = 1$$

y para toda fórmula del tipo $\neg p$ ($p \in P$) en Δ:

$$\sigma(p) = 0$$

Nótese que todas las variables proposicionales aparecen directamente o negadas en Δ.

No es difícil demostrar por inducción que, para toda fórmula φ:

$\sigma(\varphi) = 1$ ssi $\varphi \in \Delta$

Como $\Sigma \subseteq \Delta$, tenemos que $\sigma \models \Sigma$.

- El teorema de compacidad tiene varias consecuencias.

- Por ejemplo, si Σ es un conjunto infinito de fórmulas, se cumple que

$$\Sigma \models \varphi$$

ssi hay un conjunto finito Σ_0 tal que

$$\Sigma_0 \models \varphi$$

.