Inicio de la Lógica

Originalmente, la Lógica trataba con argumentos en el lenguage natural.

Ejemplo: ¿Es el siguiente argumento válido?

Todos los hombres son mortales.

Sócrates es hombre.

Por lo tanto, Sócrates es mortal.

La lógica debería poder usarse para demostrar que sí.

Inicio de la Lógica

Otro ejemplo: ¿Qué pasa con el siguiente caso?

Algunas personas son mujeres.

Sócrates es una persona.

Por lo tanto, Sócrates es mujer.

En este caso deberíamos decir que el argumento no es válido.

Inicio de la Lógica

Pero los argumentos pueden ser más complejos ...

Creo que todos los hombres son mortales.

Creo que Sócrates es hombre.

Por lo tanto, creo que Sócrates es mortal.

¿Es este argumento válido? ¿Por qué?

¿Qué significa creo? ¿Qué pasaría si reemplazamos creo que por no se si?

Paradojas en el lenguage natural

Un día de la próxima semana les voy a hacer una interrogación, y les aseguro que el día que se las haga van a estar sorprendidos.

¿Qué día voy a hacer la interrogación?

Matemática en el lenguage natural: Paradoja de Berry

Podemos representar los números naturales usando oraciones del lenguage natural: “Mil quinientos veinte”, “el primer número”, ...

El número de palabras en el Diccionario de la Real Academia es finito.

El número de oraciones con a los más 50 palabras también es finito.

Matemática en el lenguage natural: Paradoja de Berry

Sea B el siguiente número natural:

> El primer número natural que no puede ser definido por una oración con a lo más cincuenta palabras tomadas del Diccionario de la Real Academia.

B está bien definido, pero con sólo 25 palabras. **¡Tenemos una contradicción!**

¿Qué pasó?

Más paradojas: Russell (1902)

También pueden aparecer paradojas usando lenguage matemático.

Sea $A = \{1,\ 2,\ 3\}$

¿$A \in A$? No.

Sea $B = \{\{1,\ 2,\ 3\},\ \{4, 5\}\}$

¿$A \in B$? Sí.

¿$B \in B$? No.

Más paradojas: Russell (1902)

Sea C el conjunto de todos los conjuntos que tienen a lo menos dos elementos: $C = \{A, B, \ldots\}$

¿$C \in C$? Sí.

Entonces podemos definir el siguiente conjunto: $U = \{X \mid X \notin X\}$.

Tenemos: $A \in U$, $B \in U$, $C \notin U$.

¿$U \in U$? Por definición, $U \in U$ si y sólo si $U \notin U$. ¡Tenemos una contradicción!

¿Cómo definimos la noción de conjunto?

¿Por qué necesitamos la Lógica?

Necesitamos un lenguage con una sintaxis precisa y una semántica bien definida.

Queremos usar este lenguage en matemáticas.

- Definición de objetos matemáticos: conjunto, números naturales, números reales.
- Definición de teorías matemáticas: teoría de conjuntos, teoría de los número naturales.
- Definición del concepto de demostración.

También queremos usar este lenguage en computación. ¿Por qué?

¿Por qué necesitamos la Lógica en computación?

Algunas aplicaciones:

- Bases de datos: Lenguages de consulta, lenguages para restricciones de integridad.
- Inteligencia artificial: Representación de conocimiento, razonamiento con sentido común.
- Ingeniería de software: Especificación de sistemas (lenguage Z), verificación de propiedades.
- Teoría de la computación: complejidad descriptiva, algoritmos de aproximación.
- Criptografía: verificación de protocolos criptográficos.
- Procesamiento de lenguage natural.
- ...

Lógica Proposicional: Sintaxis

Tenemos los siguientes elementos:

- Variables proposicionales (P): p, q, r, $\ldots$
- Conectivos lógicos: $\neg$, $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$
- Símbolos de puntuación: (,)

Cada variable proposicional representa una proposición **completa** e **indivisible**, que puede ser **verdadera** o **falsa**.

Ejemplo:

$$P = \{socrates\_es\_hombre,\ socrates\_es\_mortal\}.$$

Lógica Proposicional: Sintaxis

Conectivos lógicos son usados para construir expresiones que también pueden ser verdaderas o falsas.

Ejemplos:

$$socrates\_es\_hombre \quad \rightarrow \quad socrates\_es\_mortal$$

$$socrates\_es\_hombre \quad \rightarrow \quad (\neg\, socrates\_es\_mortal)$$

Símbolos de puntuación son usados para evitar ambigüedades.

Sintaxis de la Lógica Proposicional: Definición

Dado: Conjunto P de variables proposicionales.

Definición: $L(P)$ es el menor conjunto que satisface las siguientes reglas:

1. $P \subseteq L(P)$.
2. Si $\varphi \in L(P)$, entonces $(\neg\varphi) \in L(P)$.
3. Si $\varphi, \psi \in L(P)$, entonces $(\varphi \vee \psi) \in L(P)$, $(\varphi \wedge \psi) \in L(P)$, $(\varphi \rightarrow \psi) \in L(P)$ y $(\varphi \leftrightarrow \psi) \in L(P)$.

Ejercicio: Verifique que $((\neg p) \rightarrow (q \vee r))$ es una fórmula.

Sintaxis de la Lógica Proposicional: Definición

La naturaleza de la definición es inductiva.

- Permite construir programas recursivos para chequear si una fórmula está bien construida.
- Permite definir inductivamente conceptos asociados a las fórmulas.
- Permite demostrar inductivamente propiedades de las fórmulas.

Definiciones inductivas

Queremos definir una función la que indica cuantos símbolos tiene una fórmula: $la((p \wedge q)) = 5$.

Caso base: Para cada $p \in P$, $la(p) = 1$.

Caso inductivo: $la((\neg\varphi)) = 3 + la(\varphi)$ y $la((\varphi \star \psi)) = 3 + la(\varphi) + la(\psi)$, donde $\star$ corresponde a $\vee$, $\wedge$, $\rightarrow$ o $\leftrightarrow$.

En el ejemplo: $la((p \wedge q)) = 3 + la(p) + la(q) = 3 + 1 + 1 = 5$.

Ejercicio: Defina las funciones pi y pd que indican cuáles son los números de paréntesis izquierdos y derechos en una fórmula, respectivamente.

Demostraciones inductivas

Lo siguiente parece ser cierto: Cada fórmula contiene el mismo número de paréntesis izquierdos y derechos.

$$pi(\varphi) = pd(\varphi), \text{ para cada fórmula } \varphi.$$

¿Cómo podemos demostrar esto?

Podemos usar inducción ...

Inducción en los números naturales

Principio de inducción: para cada $A \subseteq \mathbb{N}$ tal que

Caso base: $0 \in A$,

Caso inductivo: si $n \in A$, entonces $n + 1 \in A$,

se tiene que $A = \mathbb{N}$.

Este principio se usa para demostrar que los naturales tienen alguna propiedad. ¿Por qué funciona?

Ejercicio: Dar un principio de inducción para las fórmulas de un lenguage proposicional $L(P)$.

Inducción en la lógica proposicional

Principio de inducción: Para cada $A \subseteq L(P)$ tal que

Caso base: $p \in A$, para cada $p \in P$,

Caso inductivo: si $\varphi, \psi \in A$, entonces $(\neg\varphi) \in A$ y $(\varphi \star \psi) \in A$, donde $\star \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$,

se tiene que $A = L(P)$.

¿Por qué funciona?

Ejercicio: Demuestre que cada fórmula contiene el mismo número de paréntesis izquierdos y derechos.

Inducción en la lógica proposicional: Ejercicios

1. Defina $v(\varphi)$ como el número de ocurrencias de variables proposicionales en φ.

2. Demuestre que para cada fórmula proposicional φ que no contiene el símbolo $\neg$ se tiene que $la(\varphi) \leq 3 \cdot v(\varphi)^2$.

   ¿Qué sucede si φ contiene el símbolo $\neg$?

   ¿Qué sucede si las fórmulas de la forma $(\neg(\neg\varphi))$ no son permitidas?

3. Demuestre que un prefijo propio de una fórmula no es una fórmula.

Lógica proposicional: Lectura única

Una fórmula φ es atómica si $\varphi = p$, donde $p \in P$.

Una fórmula φ es compuesta si no es atómica.

- Si $\varphi = (\neg \alpha)$, entonces $\neg$ es un conectivo primario de φ y α es una subfórmula inmediata de φ.

- Si $\varphi = (\alpha \star \beta)$, entonces $\star$ es un conectivo primario de φ y α, β son subfórmulas inmediatas de φ.

Teorema (Lectura única): Cada fórmula compuesta tiene un único conectivo primario y únicas subfórmulas inmediatas.

Ejercicio: Demuestre el teorema de Lectura única.

Semántica de la lógica proposicional

¿Cómo podemos determinar si una fórmula es verdadera o falsa?

Este valor de verdad depende de los valores de verdad asignados a las variables proposicionales y de los conectivos utilizados.

Valuación (asignación): $\sigma : P \to \{0, 1\}$.

Ejemplo: $\sigma(socrates\_es\_hombre) = 1$ y $\sigma(socrates\_es\_mortal) = 0$.

Semántica: Definición

Dado $\sigma : P \to \{0,1\}$, queremos extender σ:

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

Definición: Dado $\varphi \in L(P)$,

- Si $\varphi = p$, entonces $\hat{\sigma}(\varphi) := \sigma(p)$.
- Si $\varphi = (\neg\alpha)$, entonces

$$\hat{\sigma}(\varphi) = \begin{cases} 1 & \text{si } \hat{\sigma}(\alpha) = 0 \\ 0 & \text{si } \hat{\sigma}(\alpha) = 1 \end{cases}$$

- Si $\varphi = (\alpha \vee \beta)$, entonces

$$\hat{\sigma}(\varphi) = \begin{cases} 1 & \text{si } \hat{\sigma}(\alpha) = 1 \text{ o } \hat{\sigma}(\beta) = 1 \\ 0 & \text{si } \hat{\sigma}(\alpha) = 0 \text{ y } \hat{\sigma}(\beta) = 0 \end{cases}$$

Semántica: Definición (continuación)

- Si $\varphi = (\alpha \wedge \beta)$, entonces

$$\hat{\sigma}(\varphi) = \begin{cases} 1 & \text{si } \hat{\sigma}(\alpha) = 1 \text{ y } \hat{\sigma}(\beta) = 1 \\ 0 & \text{si } \hat{\sigma}(\alpha) = 0 \text{ o } \hat{\sigma}(\beta) = 0 \end{cases}$$

- Si $\varphi = (\alpha \rightarrow \beta)$, entonces

$$\hat{\sigma}(\varphi) = \begin{cases} 1 & \text{si } \hat{\sigma}(\alpha) = 0 \text{ o } \hat{\sigma}(\beta) = 1 \\ 0 & \text{si } \hat{\sigma}(\alpha) = 1 \text{ y } \hat{\sigma}(\beta) = 0 \end{cases}$$

- Si $\varphi = (\alpha \leftrightarrow \beta)$, entonces

$$\hat{\sigma}(\varphi) = \begin{cases} 1 & \text{si } \hat{\sigma}(\alpha) = \hat{\sigma}(\beta) \\ 0 & \text{si } \hat{\sigma}(\alpha) \neq \hat{\sigma}(\beta) \end{cases}$$

Por simplicidad vamos a usar σ en lugar de $\hat{\sigma}$.

Semántica: Ejemplos

Supongamos que $\sigma(socrates\_es\_hombre) = 1$ y $\sigma(socrates\_es\_mortal) = 0$.

Entonces:

$$\sigma((socrates\_es\_hombre \rightarrow socrates\_es\_mortal)) = 0$$

$$\sigma((((socrates\_es\_hombre \rightarrow socrates\_es\_mortal) \wedge socrates\_es\_hombre) \rightarrow socrates\_es\_mortal)) = 1$$

Equivalencia de fórmulas

Definición: Dos fórmulas φ, ψ son equivalentes si para toda valuación σ se tiene que $\sigma(\varphi) = \sigma(\psi)$.

Notación: $\varphi \equiv \psi$.

Algunas equivalencias útiles:

$$
\begin{array}{rclrcl}
(\neg(\varphi \wedge \psi)) & \equiv & ((\neg\varphi) \vee (\neg\psi)) & (\varphi \rightarrow \psi) & \equiv & ((\neg\varphi) \vee \psi) \\
(\neg(\varphi \vee \psi)) & \equiv & ((\neg\varphi) \wedge (\neg\psi)) & (\varphi \leftrightarrow \psi) & \equiv & ((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)) \\
(\varphi \wedge (\psi \wedge \theta)) & \equiv & ((\varphi \wedge \psi) \wedge \theta) & (\neg(\neg\varphi)) & \equiv & \varphi \\
(\varphi \vee (\psi \vee \theta)) & \equiv & ((\varphi \vee \psi) \vee \theta) & & &
\end{array}
$$

Equivalencia de fórmulas

Notación: Desde ahora en adelante

- vamos a omitir los paréntesis externos,
- vamos a escribir $\varphi \wedge \psi \wedge \theta$ en lugar de $(\varphi \wedge \psi) \wedge \theta$ (lo mismo para $\vee$).

Ejercicio: ¿Es $\rightarrow$ asociativo? Vale decir, ¿Es cierto que $(\alpha \rightarrow \beta) \rightarrow \gamma \equiv \alpha \rightarrow (\beta \rightarrow \gamma)$?

Tablas de verdad

Cada fórmula se puede representar y analizar en una tabla de verdad.

p	q	$\neg p$	$p \vee q$	$p \wedge q$	$p \rightarrow q$	$p \leftrightarrow q$
0	0	1	0	0	1	1
0	1	1	1	0	1	0
1	0	0	1	0	0	0
1	1	0	1	1	1	1

Observación: Dos fórmulas son equivalentes si tienen la misma tabla de verdad.

Ejercicio: Asuma que $P = \{p, q\}$. ¿Cuántas fórmulas contiene $L(P)$? ¿Cuántas fórmulas no equivalentes contiene este conjunto?

Conectivos ternarios

Queremos definir el conectivo lógico: si p entonces q si no r.

p	q	r	si p entonces q si no r
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

¿Cómo se puede representar este conectivo usando $\neg$, $\wedge$ y $\rightarrow$?

Conectivos ternarios (continuación)

Solución: $(p \rightarrow q) \wedge ((\neg p) \rightarrow r)$.

p	q	r	si p entonces q si no r	$(p \rightarrow q) \wedge ((\neg p) \rightarrow r)$
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	1	1
1	1	1	1	1

¿Por qué el conectivo es equivalente a la fórmula? Porque tienen la misma tabla de verdad.

Conectivos n-arios

Usando tablas de verdad podemos definir conectivos n-arios: $C(p_1, \ldots, p_n)$.

p_1	p_2	$\cdots$	p_{n-1}	p_n	$C(p_1, p_2, \ldots, p_{n-1}, p_n)$
0	0	$\cdots$	0	0	b_1
0	0	$\cdots$	0	1	b_2
$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$
1	1	$\cdots$	1	1	b_{2^n}

¿Es posible representar $C(p_1, \ldots, p_n)$ usando $\neg$, $\vee$, $\wedge$, $\rightarrow$ y $\leftrightarrow$?

Conectivos n-arios

Veamos un ejemplo: $C_1(p, q, r, s)$.

p	q	r	s	$C_1(p, q, r, s)$	p	q	r	s	$C_1(p, q, r, s)$
0	0	0	0	0	1	0	0	0	1
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	1	0	1	0	1	1	0
0	1	0	0	0	1	1	0	0	0
0	1	0	1	0	1	1	0	1	0
0	1	1	0	1	1	1	1	0	1
0	1	1	1	0	1	1	1	1	0

¿Cómo definimos $C_1(p, q, r, s)$ usando $\neg$, $\vee$, $\wedge$, $\rightarrow$ y $\leftrightarrow$?

Conectivos n-arios

Solución: $C_1(p,q,r,s)$ es equivalente a la siguiente fórmula

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

Notación: Desde ahora en adelante $\neg$ tiene mayor precedencia que los conectivos binarios. Así por ejemplo, $(\neg p) \rightarrow q$ es lo mismo que $\neg p \rightarrow q$ y la fórmula anterior es lo mismo que:

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

Conectivos n-arios

Solución a nuestro problema original:

Asumiendo que σ_i es la valuación correspondiente a la fila i de la tabla de verdad de $C(p_1, \ldots, p_n)$, este conectivo es equivalente a:

$$\bigvee_{i\,:\,b_i=1} \left(\left(\bigwedge_{j\,:\,\sigma_i(p_j)=1} p_j \right) \wedge \left(\bigwedge_{k\,:\,\sigma_i(p_k)=0} \neg p_k \right) \right).$$

Conclusión: Basta con los conectivos lógicos $\neg, \vee, \wedge$ para representar cualquier tabla de verdad.

Conectivos funcionalmente completos

Definición: Un conjunto de conectivos es funcionalmente completo si es posible definir cada fórmula usando sólo estos conectivos.

Ya demostramos que $\{\neg, \vee, \wedge\}$ es funcionalmente completo. ¿Son $\{\neg, \vee\}$ y $\{\neg, \wedge\}$ funcionalmente completos?

Ejercicio: Demuestre que $\{\neg, \rightarrow\}$ es funcionalmente completo.

Ejercicio: ¿Es $\{\wedge, \vee, \rightarrow, \leftrightarrow\}$ funcionalmente completo?

***Ejercicio:** ¿Es $\{\neg, \leftrightarrow\}$ funcionalmente completo?

Decimos que una fórmula φ está en forma normal disjuntiva (DNF) si φ es de la forma:

$$\bigvee_{i=1}^{m} \Big(\bigwedge_{j=1}^{n_i} l_{i,j} \Big),$$

donde cada $l_{i,j}$ es un literal, es decir, una letra proposicional o la negación de una letra proposicional.

Ejemplo: $(p \wedge q) \vee (\neg p \wedge r)$.

Teorema: Toda fórmula es equivalente a una fórmula en DNF.

Ya demostramos este teorema, ¿Cierto?

Formas normales

Decimos que una fórmula φ está en forma normal conjuntiva (CNF) si φ es de la forma:

$$\bigwedge_{i=1}^{m} \left(\bigvee_{j=1}^{n_i} l_{i,j} \right),$$

donde cada $l_{i,j}$ es un literal.

Ejemplo: $(p \vee \neg q) \wedge (\neg p \vee \neg r \vee s) \wedge (\neg r \vee s)$.

Teorema: Toda fórmula es equivalente a una fórmula en CNF.

Ejercicio: Demuestre el teorema.

La noción de consecuencia lógica

Una valuación σ satisface un conjunto de fórmulas Σ si para cada $\varphi \in \Sigma$, se tiene que $\sigma(\varphi) = 1$.

Notación: $\sigma(\Sigma) = 1$.

¿Cuándo decimos que una fórmula ψ se deduce desde Σ?

Definición: ψ es consecuencia lógica de Σ si para cada valuación σ tal que $\sigma(\Sigma) = 1$, se tiene que $\sigma(\psi) = 1$.

Notación: $\Sigma \models \psi$.

La noción de consecuencia lógica: Ejemplos

Modus ponens:

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

Demostración por partes:

$$\{p \vee q \vee r,\ p \to s,\ q \to s,\ r \to s\} \models s$$

Ejercicio: Demuestre que si $\Sigma \models \alpha \wedge \beta$, entonces $\Sigma \models \alpha$ y $\Sigma \models \beta$.

Ejercicio: ¿Es cierto que si $\Sigma \models \alpha \vee \beta$, entonces $\Sigma \models \alpha$ o $\Sigma \models \beta$?

Teorema de monotonía

Teorema (Monotonía): Si $\Sigma \models \psi$, entonces para cada fórmula θ se tiene que $\Sigma \cup \{\theta\} \models \psi$.

Sabemos que $\{p,\ p \rightarrow q\} \models q$. Usando el teorema de monotonía deducimos que $\{p,\ p \rightarrow q,\ \neg q\} \models q$. ¿Cómo es esto posible?

Ejercicio: Demuestre el teorema de monotonía.

¿Puede usarse la lógica proposicional para modelar razonamiento con sentido común?

Un paréntesis: Revisión de conocimiento

Teorema de monotonía: Agregar conocimiento no nos permite retractarnos.

- No *actualizamos* nuestro conocimento de acuerdo a la nueva información.

Dado Σ y φ: queremos generar una fórmula que refleje la actualización de Σ dado φ.

Notación: $\Sigma \circ \varphi$.

¿Cómo podemos hacer esto? ¿Qué debería ser $\{p, p \rightarrow q\} \circ \neg q$?

Un paréntesis: Revisión de conocimiento

Una primera alternativa: $\Sigma \circ \varphi = \varphi$.

Vamos a mostrar una mejor alternativa: Belief Revision.

Notación: Dado un conjunto de variables proposicionales P

- *modelos*(Σ): Conjunto de las valuaciones de P que satisfacen Σ.

- $\Delta(\sigma_1, \sigma_2)$: Conjunto de las variables proposicionales $p \in P$ tales que $\sigma_1(p) \neq \sigma_2(p)$.

  Ejemplo: Si $P = \{p, q\}$, $\sigma_1(p) = 1$, $\sigma_1(q) = 1$, $\sigma_2(p) = 1$ $\sigma_2(q) = 0$, entonces $\Delta(\sigma_1, \sigma_2) = \{q\}$.

  $\Delta(\sigma_1, \sigma_2)$ mide la *distancia* entre σ_1 y σ_2.

Un paréntesis: Revisión de conocimiento

Para actualizar Σ dado φ, vamos a actualizar los modelos de Σ con respecto a φ.

Dado σ tal que $\sigma(\Sigma) = 1$, queremos seleccionar los modelos σ_1 de φ que están a distancia mínima de σ.

Formalmente:

$$\begin{aligned} \textit{mínimo}(\sigma, \varphi) \ = \ & \{\sigma_1 \mid \sigma_1(\varphi) = 1 \text{ y no existe } \sigma_2 \text{ tal que} \\ & \qquad \sigma_2(\varphi) = 1 \text{ y } \Delta(\sigma, \sigma_2) \subsetneq \Delta(\sigma, \sigma_1)\}. \end{aligned}$$

Un paréntesis: Revisión de conocimiento

Definimos los modelos de $\Sigma \circ \varphi$ como los modelos de φ que están *más cerca* de los modelos de Σ:

$$modelos(\Sigma \circ \varphi) \quad = \bigcup_{\sigma \ : \ \sigma(\Sigma)=1} mínimo(\sigma, \varphi)$$

y definimos $\Sigma \circ \varphi$ como una fórmula ψ arbitraria tal que $modelos(\psi) = modelos(\Sigma \circ \varphi)$.

¿Siempre existe esta fórmula? ¿Es única?

Un paréntesis: Revisión de conocimiento

Ejemplo: $\Sigma = \{p, p \rightarrow q\}$ y $\varphi = \neg q$

Primero calculamos los modelos de Σ y φ:

$$\begin{array}{rcll} modelos(\Sigma) & = & \{\sigma\}, & \text{donde } \sigma(p) = \sigma(q) = 1 \\ modelos(\varphi) & = & \{\sigma_1, \sigma_2\}, & \text{donde } \sigma_1(p) = 1,\ \sigma_1(q) = 0 \text{ y} \\ & & & \sigma_2(p) = \sigma_2(q) = 0. \end{array}$$

Después calculamos los modelos mínimos:

$$\begin{array}{rcl} \Delta(\sigma, \sigma_1) & = & \{q\} \\ \Delta(\sigma, \sigma_2) & = & \{p, q\} \\ mínimo(\sigma, \varphi) & = & \{\sigma_1\} \\ modelos(\Sigma \circ \varphi) & = & \{\sigma_1\} \end{array}$$

Resultado: $\{p, p \rightarrow q\} \circ \neg q \quad = \quad p \wedge \neg q.$

El problema de satisfacción

Definición: Un conjunto de fórmulas Σ es satisfacible si existe una valuación σ tal que $\sigma(\Sigma) = 1$. En caso contrario, Σ es inconsistente.

Existe una estrecha relación entre las nociones de consecuencia lógica y satisfacibilidad.

Teorema: $\Sigma \models \varphi$ si y sólo si $\Sigma \cup \{\neg\varphi\}$ es inconsistente.

Ejercicio: Demuestre el teorema.

El problema de satisfacción

El teorema anterior nos permite *reducir* el problema de verificar si $\Sigma \models \varphi$ al problema de verificar si $\Sigma \cup \{\neg\varphi\}$ es inconsistente.

Ejercicio: Demuestre que la reducción inversa también es posible. Vale decir, encuentre una fórmula ψ tal que Σ es satisfacible si y sólo si $\Sigma \not\models \psi$.

Entonces, si tenemos un algoritmo para uno de los problemas, lo tenemos para el otro.

¿Cómo verificamos si una fórmula es satisfacible?

El problema de satisfacción

Un algoritmo simple: Dada una fórmula φ, construya la tabla de verdad para φ y verifique si en alguna fila la salida es 1.

¿Cuál es la complejidad de este algoritmo?

Si φ menciona n variables proposicionales, entonces el algoritmo toma tiempo 2^n (aproximadamente).

Este es un algoritmo de tiempo exponencial ¿Es posible usar este algoritmo en la práctica?

El problema de satisfacción: Complejidad

Número estimado de electrones en el universo $\leq 10^{130}$.

Si $n = 1000$ y en cada electrón del universo tuviéramos un supercomputador que ejecuta 10^{50} operaciones por segundo, entonces para verificar si φ es satisfacible necesitaríamos:

$$\frac{2^{1000}}{10^{50} \cdot 10^{130}} \approx 10^{121} \text{ segundos.}$$

Edad estimada del universo $< 10^{18}$ segundos! Y φ se puede almacenar en algunos kilobytes de memoria!

¿Existe un algoritmo eficiente para el problema de satisfacibilidad?

Otra noción útil

Definición: Una fórmula φ es una tautología si para cada valuación σ se tiene que $\sigma(\varphi) = 1$.

Ejemplo: $p \vee \neg p$.

Ejercicio: Sea Σ un conjunto finito de fórmulas. Demuestre que el problema de verificar si $\Sigma \models \varphi$ puede reducirse al problema de verificar si una fórmula es una tautología.

¿Puede ser Σ infinito? ¿Qué sucede en este caso?

Teorema de compacidad

Definición: Un conjunto de fórmulas Σ es finitamente satisfacible si cada subconjunto finito de Σ es satisfacible.

Ejemplo: El conjunto $\Sigma = \{p_0\} \cup \{p_i \rightarrow p_{i+1} \mid i \in \mathbb{N}\}$ es finitamente satisfacible.

¿Es Σ satisfacible?

Teorema (Compacidad): Un conjunto de fórmulas es satisfacible si y sólo si es finitamente satisfacible.

Teorema de compacidad: Demostración

Necesitamos el siguiente lema:

Lema: Sea $\Sigma \subseteq L(P)$ finitamente satisfacible y $p \in P$. Entonces $\Sigma \cup \{p\}$ es finitamente satisfacible o $\Sigma \cup \{\neg p\}$ es finitamente satisfacible.

¿Pueden ser ambos finitamente satisfacibles?

Ejercicio: Demuestre el lema.

Teorema de compacidad: Demostración

Ahora vamos a demostrar la dirección $\Leftarrow$ del teorema. La otra dirección es trivial.

($\Leftarrow$) Asuma que $P = \{p_i \mid i \geq 1\}$ y defina una sucesión $\{\Delta_i\}_{i\in\mathbb{N}}$ de conjuntos de literales:

Caso base: $\Delta_0 = \emptyset$.

Para $i \in \mathbb{N}$:

$$\Delta_{i+1} = \begin{cases} \Delta_i \cup \{p_{i+1}\} & \Sigma \cup \Delta_i \cup \{p_{i+1}\} \text{ es finitamente satisfacible,} \\ \Delta_i \cup \{\neg p_{i+1}\} & \text{en caso contrario.} \end{cases}$$

Finalmente: $\Delta = \bigcup_{i\in\mathbb{N}} \Delta_i$.

Teorema de compacidad: Demostración

Para cada $p_i \in P$: $p_i \in \Delta_i$ o $\neg p_i \in \Delta_i$, pero no ambas.

Por lo tanto: Existe una única valuación σ que satisface a Δ.

Vamos a demostrar que esta valuación satisface a Σ.

Por contradicción: Suponga que $\sigma(\Sigma) = 0$. Entonces existe $\varphi \in \Sigma$ tal que $\sigma(\varphi) = 0$.

Asuma que φ contiene n variables proposicionales y que p_k es la de mayor índice.

Teorema de compacidad: Demostración

Tenemos que considerar dos casos.

$\sigma(p_k) = 1$: entonces $\{\varphi\} \cup \Delta_{k-1} \cup \{p_k\}$ es inconsistente.

Entonces: $\Sigma \cup \Delta_{k-1} \cup \{p_k\}$ no es finitamente satisfacible y $\neg p_k \in \Delta_k$.

Contradicción: $\Delta_k \subseteq \Delta$ y $\sigma(p_k) = 1$.

$\sigma(p_k) = 0$: entonces $\{\varphi\} \cup \Delta_{k-1} \cup \{\neg p_k\}$ es inconsistente.

Entonces: $\Sigma \cup \Delta_{k-1} \cup \{p_k\}$ es finitamente satisfacible (por lema) y $p_k \in \Delta_k$.

Contradicción: $\Delta_k \subseteq \Delta$ y $\sigma(p_k) = 0$.

Teorema de compacidad y consecuencia lógica

Corolario: $\Sigma \models \varphi$ si y sólo si existe un subconjunto finito Σ' de Σ tal que $\Sigma' \models \varphi$.

Ejercicio: Demuestre el corolario.