Satisfactibilidad y consecuencia lógica

Clase 5

IIC 1253

Prof. Pedro Bahamondes

Outline

Conectivos y fórmulas

Tenemos una fórmula $\varphi \in L(P)$, con $P = \{p, q, r\}$, y lo único que conocemos de ella es que cumple la siguiente tabla de verdad:

p	q	r	φ
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

¿Podemos construir una fórmula equivalente a φ?

Conectivos y fórmulas

Ejemplo

	p	q	r	φ
σ_1	0	0	0	1
σ_2	0	0	1	0
σ_3	0	1	0	0
σ_4	0	1	1	1

	p	q	r	φ
σ_5	1	0	0	1
σ_6	1	0	1	0
σ_7	1	1	0	0
σ_8	1	1	1	1

Estrategia:

- Codificar cada valuación σ tal que $\sigma(\varphi) = 1$
- Combinar dichas codificaciones mediante disyunción

Con esto, proponemos la siguiente fórmula equivalente a φ

$$\underbrace{((\neg p) \wedge (\neg q) \wedge (\neg r))}_{\sigma_1} \vee \underbrace{((\neg p) \wedge q \wedge r)}_{\sigma_4} \vee \underbrace{(p \wedge (\neg q) \wedge (\neg r))}_{\sigma_5} \vee \underbrace{(p \wedge q \wedge r)}_{\sigma_8}$$

Podemos generalizar esta idea para n variables

Conectivos y fórmulas

Consideremos el conectivo n-ario siguiente

	p_1	p_2	$\dots$	p_{n-1}	p_n	φ
σ_1	0	0	$\dots$	0	0	$\sigma_1(\varphi)$
σ_2	0	0	$\dots$	0	1	$\sigma_2(\varphi)$
$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$	$\vdots$	$\vdots$
σ_{2^n}	1	1	$\dots$	1	1	$\sigma_{2^n}(\varphi)$

Para cada σ_j con $j \in \{1, \dots, 2^n\}$ consideremos la siguiente fórmula:

$$\varphi_j = \left(\bigwedge_{\substack{i=1\dots n \\ \sigma_j(p_i)=1}} p_i \right) \wedge \left(\bigwedge_{\substack{i=1\dots n \\ \sigma_j(p_i)=0}} (\neg p_i) \right)$$

La fórmula φ_j codifica la j-ésima valuación

Conectivos y fórmulas

	p_1	p_2	$\dots$	p_{n-1}	p_n	φ
σ_1	0	0	$\dots$	0	0	$\sigma_1(\varphi)$
σ_2	0	0	$\dots$	0	1	$\sigma_2(\varphi)$
$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$	$\vdots$	$\vdots$
σ_{2^n}	1	1	$\dots$	1	1	$\sigma_{2^n}(\varphi)$

Ahora tomamos la disyunción de las valuaciones que satisfacen a φ

$$\bigvee_{\substack{j=1\dots 2^n \\ \sigma_j(\varphi)=1}} \varphi_j \quad = \quad \bigvee_{\substack{j=1\dots 2^n \\ \sigma_j(\varphi)=1}} \left(\left(\bigwedge_{\substack{i=1\dots n \\ \sigma_j(p_i)=1}} p_i \right) \wedge \left(\bigwedge_{\substack{i=1\dots n \\ \sigma_j(p_i)=0}} (\neg p_i) \right) \right)$$

La fórmula resultante es equivalente a φ

Conectivos y fórmulas

Teorema

Para toda tabla de verdad existe una fórmula equivalente.

Corolario

Para toda tabla de verdad existe una fórmula equivalente que solo usa símbolos $\neg, \wedge$ y $\vee$.

Conectivos funcionalmente completos

Definición

Un conjunto de conectivos lógicos se dice **funcionalmente completo** si toda fórmula en $\mathcal{L}(P)$ es lógicamente equivalente a una fórmula que sólo usa esos conectivos.

Ejemplo

El conjunto $C = \{\neg, \wedge, \vee\}$ es funcionalmente completo, pues para toda fórmula φ, se tiene que

$$\varphi \equiv \bigvee_{\substack{j=1\ldots 2^n \\ \sigma_j(\varphi)=1}} \left(\left(\bigwedge_{\substack{i=1\ldots n \\ \sigma_j(p_i)=1}} p_i \right) \wedge \left(\bigwedge_{\substack{i=1\ldots n \\ \sigma_j(p_i)=0}} (\neg p_i) \right) \right)$$

Conectivos funcionalmente completos

Ejercicios

1. Demuestre que $\{\neg, \wedge\}$, $\{\neg, \vee\}$ y $\{\neg, \rightarrow\}$ son funcionalmente completos.
2. Demuestre que $\{\neg\}$ no es funcionalmente completo.
3. ¿Es $\{\wedge, \vee, \rightarrow, \leftrightarrow\}$ funcionalmente completo?

Demostraremos 1. $\{\neg, \rightarrow\}$ y 2.
El resto quedan propuestos ★!

Conectivos funcionalmente completos

Ejercicio 1.

Demostraremos que $\{\neg, \rightarrow\}$ es funcionalmente completo.

Como sabemos que $C = \{\neg, \wedge, \vee\}$ es funcionalmente completo, demostraremos por inducción que toda fórmula construida usando sólo los conectivos anteriores es lógicamente equivalente a otra fórmula que solo usa $\neg$ y $\rightarrow$. Con esto, queda demostrado que $C' = \{\neg, \rightarrow\}$ es funcionalmente completo.

- **BI:** Si $\varphi = p$, con $p \in P$, la propiedad se cumple trivialmente.
- **HI:** Supongamos que $\varphi, \psi \in L(P)$, que sólo usan conectivos en C, son tales que $\varphi \equiv \varphi'$ y $\psi \equiv \psi'$, donde φ', ψ' sólo usan conectivos en C'.

Es crucial la dirección. Queremos probar que $C' = \{\neg, \rightarrow\}$ es F.C. por lo que debemos usar los símbolos de C' para expresar los de C

Conectivos funcionalmente completos

Ejercicio 1.

- **HI:** Supongamos que $\varphi, \psi \in L(P)$, que sólo usan conectivos en C, son tales que $\varphi \equiv \varphi'$ y $\psi \equiv \psi'$, donde φ', ψ' sólo usan conectivos en C'.
- **TI:** Consideraremos una fórmula θ construida con los pasos inductivos para los operadores en C. Tenemos tres casos que analizar:
  - $\theta = (\neg\varphi)$
  - $\theta = \varphi \wedge \psi$
  - $\theta = \varphi \vee \psi$

Conectivos funcionalmente completos

Ejercicio 1.

- **TI:** Consideraremos una fórmula θ construida con los pasos inductivos para los operadores en C.
  - $\theta = (\neg\varphi) \overset{HI}{\equiv} (\neg\varphi')$, y como φ' sólo usa conectivos en C', θ es equivalente a una fórmula que sólo usa conectivos en C'.
  - $\theta = \varphi \wedge \psi \overset{HI}{\equiv} \varphi' \wedge \psi'$

    Usando las leyes de doble negación, De Morgan y de implicancia:

    $\theta \equiv \neg(\neg(\varphi' \wedge \psi')) \equiv \neg((\neg\varphi') \vee (\neg\psi')) \equiv \neg(\varphi' \rightarrow (\neg\psi'))$

    Y como φ', ψ' sólo usan conectivos en C', θ es equivalente a una fórmula que sólo usa conectivos en C'.

Conectivos funcionalmente completos

Ejercicio 1.

- **TI:** Consideraremos una fórmula θ construida con los pasos inductivos para los operadores en C.
  - $\theta = \varphi \vee \psi \overset{HI}{\equiv} \varphi' \vee \psi'$

    Usando la ley de implicancia:

    $\theta \equiv (\neg\varphi') \to \psi'$

    Y como φ', ψ' sólo usan conectivos en C', θ es equivalente a una fórmula que sólo usa conectivos en C'.

Luego, por inducción estructural se concluye que para toda fórmula con símbolos solo de C, existe una fórmula equivalente con símbolos en C'. $\square$

Conectivos funcionalmente completos

Ejercicio 2.

Demostraremos que $\{\neg\}$ no es funcionalmente completo.

Dado $P = \{p\}$, demostraremos por inducción que toda fórmula en $\mathcal{L}(P)$ construida usando sólo p y $\neg$ es lógicamente equivalente a p o a $\neg p$. Como ninguna de estas fórmulas es equivalente a $p \wedge \neg p$, se concluye que $\{\neg\}$ no puede ser funcionalmente completo.

Esta demostración es "*negativa*"... daremos un caso
en que **no se cumple** la definición de funcionalmente completo

Ejercicio 2.

Demostraremos la propiedad

$$P(\varphi) := \varphi \text{ es equivalente a } p \text{ o a } \neg p$$

- **BI:** Si $\varphi = p$, con $p \in P$, la propiedad se cumple trivialmente.
- **HI:** Supongamos que $\varphi \in L(P)$ construida usando sólo p y $\neg$ es equivalente a p o a $\neg p$.

Estamos acotando las tablas de verdad que son posibles en $\{\neg\}$

Conectivos funcionalmente completos

Ejercicio 2.

- **TI:** El único caso inductivo que tenemos que demostrar es $\psi = \neg\varphi$, pues sólo podemos usar el conectivo $\neg$.

  Por **HI**, sabemos que para toda valuación σ se cumple que $\sigma(\varphi) = \sigma(p)$ o $\sigma(\varphi) = \sigma(\neg p)$. Podemos hacer una tabla de verdad:

φ	$\psi = \neg\varphi$
p	$\neg p$
$\neg p$	p

Concluimos que ψ es equivalente a p o a $\neg p$.

Como la fórmula $\psi = p \wedge \neg p$ no es equivalente a ninguna fórmula que solo usa símbolos de $\{\neg\}$, tenemos que $\{\neg\}$ no es funcionalmente completo. $\square$

Objetivos de la clase

- Comprender el concepto de satisfactibilidad de fórmulas y conjuntos
- Aplicar lógica para modelar problemas
- Conocer las formas normales

Outline

Satisfactibilidad

Definición

Una fórmula φ es **satisfactible** si existe una valuación σ tal que $\sigma(\varphi) = 1$.

Ejemplo

Las siguientes fórmulas son satisfactibles:

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

Las siguientes fórmulas no son satisfactibles:

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

Una fórmula es satisfactible si hay algún "*mundo*" en el cual es verdadera

El problema de satisfactibilidad

Problema de satisfactibilidad (SAT)

Sea φ una fórmula proposicional. El **problema de satisfactibilidad** consiste en determinar si φ es satisfactible o no.

Este es un problema central en computación

- Permite resolver problemas fuera de la lógica...
- ... usando modelación en lógica proposicional

¿Es un problema difícil? ¿Cómo se resuelve?

Modelación en lógica proposicional

Ejercicio

Sea M un mapa conformado por n países. Decimos que M es 3-coloreable si se pueden pintar todos los países con 3 colores sin que ningún par de países adyacentes tenga el mismo color. En otras palabras, los países vecinos deben tener colores distintos.

Dado un mapa M, construya una fórmula $\varphi \in \mathcal{L}(P)$ tal que M es 3-coloreable si y sólo si φ es satisfactible.

La fórmula φ debe **codificar** los requisitos y estructura del problema

Ejercicio (mapa 3-coloreable)

Sea M el mapa. Consideremos la lista de países $\{1, 2, \ldots, n\}$ y una lista de pares de países adyacentes $A = \{(i, j), (k, m), \ldots\}$.

Seguiremos la siguiente estrategia para resolver el problema

1. Definición de variables proposicionales
   - Variables predefinidas por el problema
   - Variables que hay que asignar
2. Construcción de restricciones a través de fórmulas proposicionales
3. Demostración de que φ cumple lo pedido (si y solo si)

Satisfactibilidad

Ejercicio (mapa 3-coloreable)

Primero, definimos las variables proposicionales. Usamos dos tipos de variables:

- Para $1 \leq i, j \leq n$ definimos

  $$p_{ij} = \begin{cases} 1 & \text{si } i \text{ es adyacente con } j \\ 0 & \text{si no} \end{cases}$$

  Observamos que cada p_{ij} se conoce de antemano una vez que conocemos el mapa M. Debemos **inicializarlas**.

- Análogamente, para $1 \leq i \leq n$ definimos

  $$r_i \quad b_i \quad g_i$$

  que valen 1 si el país i es pintado rojo, azul o verde respectivamente, y 0 en caso contrario. Estas variables deben ser **determinadas** para resolver el problema.

¿Qué restricciones son naturales para este problema?

Satisfactibilidad

Ejercicio (mapa 3-coloreable)

Para representar el problema vamos a definir φ como la conjunción de las siguientes fórmulas.

- "*Cada país tiene exactamente un color*"

$$\varphi_C = \bigwedge_{i=1}^{n}\Bigg(\Big(r_i \vee b_i \vee g_i\Big) \wedge \Big(r_i \rightarrow (\neg b_i \wedge \neg g_i)\Big) \wedge \Big(b_i \rightarrow (\neg r_i \wedge \neg g_i)\Big) \wedge \Big(g_i \rightarrow (\neg r_i \wedge \neg b_i)\Big)\Bigg)$$

- "*Países adyacentes tienen colores distintos*"

$$\varphi_D = \bigwedge_{i=1}^{n}\bigwedge_{j=1}^{n}\Bigg(p_{ij} \rightarrow \Big((r_i \rightarrow \neg r_j) \wedge (b_i \rightarrow \neg b_j) \wedge (g_i \rightarrow \neg g_j)\Big)\Bigg)$$

Ejercicio (mapa 3-coloreable)

- Inicializamos las variables conocidas por la instancia M del problema

$$\varphi_M = \bigwedge_{(i,j)\in A} p_{ij} \quad \wedge \bigwedge_{(i,j)\notin A} \neg p_{ij}$$

Entonces, nuestra fórmula será la conjunción de las fórmulas anteriores:

$$\varphi = \varphi_C \wedge \varphi_D \wedge \varphi_M$$

Ahora demostraremos que M es 3-coloreable si y sólo si φ es satisfactible.

Debemos demostrar dos direcciones

Satisfactibilidad

Ejercicio (mapa 3-coloreable)

($\Rightarrow$) **PDQ** Si M es 3-coloreable, entonces φ es satisfactible.

Supongamos que M es 3-coloreable. Luego, existe una coloración válida para M.Construimos una valuación σ según

$$\begin{aligned}
\sigma(p_{ij}) &= \begin{cases} 1 & \text{si } i,j \text{ son adyacentes en } M \\ 0 & \text{en otro caso} \end{cases} \\
\sigma(r_i) &= \begin{cases} 1 & \text{si } i \text{ es rojo en la coloración de } M \\ 0 & \text{en otro caso} \end{cases} \\
\sigma(b_i) &= \begin{cases} 1 & \text{si } i \text{ es azul en la coloración de } M \\ 0 & \text{en otro caso} \end{cases} \\
\sigma(g_i) &= \begin{cases} 1 & \text{si } i \text{ es verde en la coloración de } M \\ 0 & \text{en otro caso} \end{cases}
\end{aligned}$$

Esta dirección consiste en construir una valuación que satisface φ

Satisfactibilidad

Ejercicio (mapa 3-coloreable)

($\Rightarrow$) (continuación) Ahora verificamos que $\sigma(\varphi) = 1$:

- $\sigma(\varphi_C)$: para cada país i, se debe cumplir que $\sigma(r_i) = 1$, o que $\sigma(g_i) = 1$, o que $\sigma(b_i) = 1$, y solo una de estas, por construcción de σ. Luego, es claro que $\sigma(\varphi_C) = 1$.
- $\sigma(\varphi_D)$: para cada combinación de países i, j, sabemos que $\sigma(p_{ij}) = 1$ solo cuando los países son adyacentes. Entonces, como en φ_D tenemos una implicancia, solo nos preocuparemos de los pares de países adyacentes. En el consecuente de la implicancia, sabemos que si por ejemplo $\sigma(r_i) = 1$, se debe cumplir que $\sigma(r_j) = 0$, dado que construimos σ a partir de una 3-coloración. Para las otras dos implicancias, sabemos que el lado izquierdo va a ser falso (por la fórmula φ_C), y por lo tanto todo el lado derecho se hace verdadero, y entonces $\sigma(\varphi_D) = 1$. El análisis para cuando i es de otro color es análogo.

Ejercicio (mapa 3-coloreable)

($\Rightarrow$) (continuación)

- $\sigma(\varphi_M)$: por construcción de σ es claro que $\sigma(\varphi_M) = 1$, dado que la construimos precisamente como esta fórmula, asignando 1 a pares de países adyacentes y 0 a los que no.

Finalmente, como φ es la conjunción de las fórmulas anteriores, concluimos que $\sigma(\varphi) = 1$, y entonces φ es satisfactible.

La dirección opuesta comienza suponiendo que φ es satisfactible

Ejercicio (mapa 3-coloreable)

($\Leftarrow$) **P.D.** Si φ es satisfactible, entonces M es 3-coloreable.

Supongamos que φ es satisfactible. Luego, existe una valuación σ tal que $\sigma(\varphi) = 1$, y por construcción $\sigma(\varphi_C) = \sigma(\varphi_D) = \sigma(\varphi_M) = 1$. Usaremos esta valuación para colorear el mapa.

En primer lugar, como $\sigma(\varphi_C) = 1$, sabemos que para cada i, $\sigma(r_i \vee g_i \vee b_i) = 1$, y por lo tanto cada país tiene asignado al menos un color. Sin pérdida de generalidad, supongamos que $\sigma(r_k) = 1$, es decir, pintamos el país k rojo. Como también se cumple que $\sigma(r_k \rightarrow (\neg g_k \wedge \neg b_k)) = 1$, necesariamente $\sigma(g_k) = 0$ y $\sigma(b_k) = 0$, y por lo tanto cada país tiene un único color.

Esta dirección busca deducir la **existencia**
de una coloración a partir de la valuación

Ejercicio (mapa 3-coloreable)

($\Leftarrow$) (continuación)

En segundo lugar, como $\sigma(\varphi_M) = 1$, sabemos que si i, j son adyacentes en M, $\sigma(p_{ij}) = 1$, y si no lo son, $\sigma(p_{ij}) = 0$. Ahora, en $\sigma(\varphi_D) = 1$, solo nos interesa el primer caso (dado que en el segundo no podemos concluir nada de la implicancia). Tomemos entonces i, j adyacentes, y sin pérdida de generalidad supongamos que $\sigma(r_i) = 1$. Como $\sigma(r_i \to \neg r_j) = 1$ para todo j adyacente a i, necesariamente $\sigma(r_j) = 0$, y entonces los países adyacentes no pueden estar pintados del mismo color.

Concluimos que usando los colores asignados por φ a través de r_i, g_i, b_i, podemos 3-colorear M.

Otros conceptos asociados a satisfactibilidad

Definición

Una fórmula φ es una **contradicción** si no es satisfactible; es decir, para toda valuación σ se tiene que $\sigma(\varphi) = 0$.

Ejemplo

$p \wedge \neg p$

Definición

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

Ejemplo

$p \vee \neg p$

$p \leftrightarrow p$

Otros conceptos asociados a Satisfactibilidad

Definición

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

Podemos definir la equivalencia lógica de una manera alternativa:

Teorema

Dos fórmulas $\varphi, \psi \in L(P)$ son lógicamente equivalentes si $\varphi \leftrightarrow \psi$ es una tautología.

Demuestre el teorema (★)

Outline

Formas normales

Definición

Un literal es una variable proposicional o la negación de una variable proposicional.

Ejemplo

Para el conjunto de variables $P = \{p, q, r\}$, las fórmulas p y $\neg r$ son literales.

Los literales son los átomos de la construcción que mostraremos

Formas normales

Definición

Decimos que una fórmula φ está en **forma normal disyuntiva (DNF)** si es una disyunción de conjunciones de literales; o sea, si es de la forma

$$B_1 \vee B_2 \vee \ldots \vee B_k$$

donde cada B_i es una conjunción de literales, $B_i = (l_{i1} \wedge \ldots \wedge l_{ik_i})$

Ejemplo

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

¿Hemos visto fórmulas en DNF antes?

Formas normales

Definición

Decimos que una fórmula φ está en **forma normal disyuntiva (DNF)** si es una disyunción de conjunciones de literales; o sea, si es de la forma

$$B_1 \vee B_2 \vee \ldots \vee B_k$$

donde cada B_i es una conjunción de literales, $B_i = (l_{i1} \wedge \ldots \wedge l_{ik_i})$

Teorema

Toda fórmula es equivalente a una fórmula en DNF.

Ya demostramos esto y ¡mostramos cómo construir tal fórmula!

Formas normales

Definición

Decimos que una fórmula φ está en **forma normal conjuntiva (CNF)** si es una conjunción de disyunciones de literales; o sea, si es de la forma

$$C_1 \wedge C_2 \wedge \ldots \wedge C_k$$

donde cada C_i es una disyunción de literales, $C_i = (l_{i1} \vee \ldots \vee l_{ik_i})$

Observaciones

- Una disyunción de literales se llama una **cláusula**.
  - Los C_i anteriores son cláusulas.
- Una fórmula en CNF es una conjunción de cláusulas.

Ejemplo

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

¿Podríamos obtener una fórmula en CNF para una tabla de verdad?

Formas normales

Definición

Decimos que una fórmula φ está en **forma normal conjuntiva (CNF)** si es una conjunción de disyunciones de literales; o sea, si es de la forma

$$C_1 \wedge C_2 \wedge \ldots \wedge C_k$$

donde cada C_i es una disyunción de literales, $C_i = (l_{i1} \vee \ldots \vee l_{ik_i})$

Teorema

Toda fórmula es equivalente a una fórmula en CNF.

Demuestre el teorema (★○)

Formas normales

Demostración

Como sabemos que toda fórmula es equivalente a una fórmula en DNF, demostraremos que toda fórmula en DNF es equivalente a una fórmula en CNF por inducción en la cantidad de disyunciones de la primera fórmula.

Dado un conjunto de variables proposicionales P, tenemos la siguiente propiedad sobre los naturales:

$$Prop(n): \text{toda fórmula } \varphi \in \mathcal{L}(P) \text{ en DNF con a lo más } n \text{ disyunciones es lógicamente equivalente a una fórmula } \psi \text{ en CNF } (\varphi \equiv \psi).$$

Demostraremos esta propiedad por inducción.

Formas normales

$Prop(n)$: toda fórmula $\varphi \in L(P)$ en DNF con a lo más n disyunciones es lógicamente equivalente a una fórmula ψ en CNF ($\varphi \equiv \psi$).
BI: $Prop(0)$: una fórmula en DNF con 0 disyunciones es de la forma

$$l_1 \wedge l_2 \wedge \ldots \wedge l_k$$

por lo que está trivialmente en CNF.

HI: Suponemos que $Prop(n-1)$ es cierta; es decir, toda fórmula φ en DNF con a lo más $n-1$ disyunciones es lógicamente equivalente a una fórmula ψ en CNF.

Formas normales

TI: Debemos demostrar que toda fórmula φ' en DNF con n disyunciones es lógicamente equivalente a una fórmula ψ' en CNF. Cualquier φ' será de la forma

$$\varphi' = B_0 \vee B_1 \vee \ldots \vee B_{n-1} \vee B_n$$

donde los B_i son conjunciones de literales. Por asociatividad:

$$\varphi' \equiv (B_0 \vee B_1 \vee \ldots \vee B_{n-1}) \vee B_n$$

y por HI $\varphi' \overset{HI}{\equiv} \psi \vee B_n$, con ψ una fórmula en CNF. Expandiendo la conjunción B_n:

$$\varphi' \equiv \psi \vee (l_{n,1} \wedge \ldots \wedge l_{n,k_n})$$

Formas normales

TI: Por distributividad:

$$\varphi' \equiv (\psi \vee l_{n,1}) \wedge \ldots \wedge (\psi \vee l_{n,k_n})$$

Como ψ está en CNF, podemos expandirla:

$$\varphi' \equiv ((C_1 \wedge \ldots \wedge C_m) \vee l_{n,1}) \wedge \ldots \wedge ((C_1 \wedge \ldots \wedge C_m) \vee l_{n,k_n})$$

con C_i cláusulas. Por distributividad:

$$\varphi' \equiv ((C_1 \vee l_{n,1}) \wedge \ldots \wedge (C_m \vee l_{n,1})) \wedge \ldots \wedge ((C_1 \vee l_{n,k_n}) \wedge \ldots \wedge (C_m \vee l_{n,k_n}))$$

Y por asociatividad:

$$\varphi' \equiv (C_1 \vee l_{n,1}) \wedge \ldots \wedge (C_m \vee l_{n,1}) \wedge \ldots \wedge (C_1 \vee l_{n,k_n}) \wedge \ldots \wedge (C_m \vee l_{n,k_n})$$

TI: Como los C_i son cláusulas, es claro que $(C_i \vee l_{n,j})$ es una cláusula. Luego, tenemos que

$$\psi' = (C_1 \vee l_{n,1}) \wedge \ldots \wedge (C_m \vee l_{n,1}) \wedge \ldots \wedge (C_1 \vee l_{n,k_n}) \wedge \ldots \wedge (C_m \vee l_{n,k_n})$$

está en CNF y es tal que $\varphi' \equiv \psi'$. $\square$

Outline

Objetivos de la clase

- Comprender el concepto de satisfactibilidad de fórmulas y conjuntos
- Aplicar lógica para modelar problemas
- Conocer las formas normales