Formas normales y consecuencia lógica

Clase 6

IIC 1253

Prof. Pedro Bahamondes

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:

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

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:

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

Formas normales

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

Conjuntos de fórmulas

Notación

Dado un conjunto de fórmulas Σ en $L(P)$, diremos que una valuación σ **satisface** Σ, denotado por $\sigma(\Sigma) = 1$, si para toda fórmula $\varphi \in \Sigma$ se tiene que $\sigma(\varphi) = 1$.

Definición

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

¿Cuándo decimos que una fórmula se **deduce** de un conjunto?

Consecuencia lógica

Definición

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

Lo denotamos por $\Sigma \models \psi$.

ψ debe ser satisfecha en cada "*mundo*" donde Σ es verdadero

Consecuencia lógica

Ejemplo

La regla de inferencia llamada **Modus ponens** es $\{p, p \to q\} \vDash q$

p	q	p	$p \to q$	q
0	0	0	1	0
0	1	0	1	1
1	0	1	0	0
1	1	1	1	1

Nos tenemos que fijar en las valuaciones que satisfacen al conjunto...
En esos mundos, la fórmula "*objetivo*" también debe ser satisfecha

Ejercicio (propuesto ★)

Demuestre las siguientes reglas de inferencia

- **Modus tollens:** $\{\neg q, p \to q\} \vDash \neg p$
- **Demostración por partes:** $\{p \vee q \vee r, p \to s, q \to s, r \to s\} \vDash s$

Un resultado fundamental

Teorema

$\Sigma \vDash \varphi$ si y sólo si $\Sigma \cup \{\neg\varphi\}$ es inconsistente.

Este teorema combina dos mundos:
la **consecuencia lógica** y la **satisfactibilidad**

Demostración

($\Rightarrow$) Supongamos que $\Sigma \vDash \varphi$. Por contradicción, supongamos que $\Sigma \cup \{\neg\varphi\}$ es satisfactible, luego existe una valuación σ tal que $\sigma(\Sigma \cup \{\neg\varphi\}) = 1$. Esto implica que $\sigma(\Sigma) = 1$ y que $\sigma(\neg\varphi) = 1$, y por lo tanto $\sigma(\Sigma) = 1$ y $\sigma(\varphi) = 0$, lo que contradice que $\Sigma \vDash \varphi$.

($\Leftarrow$) Sea $\Sigma \cup \{\neg\varphi\}$ inconsistente. Debemos demostrar que dada una valuación σ tal que $\sigma(\Sigma) = 1$, se tiene que $\sigma(\varphi) = 1$. Como $\Sigma \cup \{\neg\varphi\}$ es inconsistente y $\sigma(\Sigma) = 1$, necesariamente $\sigma(\neg\varphi) = 0$, y luego $\sigma(\varphi) = 1$. Hemos demostrado que si σ es tal que $\sigma(\Sigma) = 1$, entonces $\sigma(\varphi) = 1$, por lo que concluimos que $\Sigma \vDash \varphi$. $\square$

Un resultado fundamental

El teorema anterior nos permite chequear $\Sigma \models \varphi$ estudiando la satisfactibilidad de $\Sigma \cup \{\neg\varphi\}$

- Podemos usar tablas de verdad para esto último. . .
- . . . pero es muy lento!

Estudiaremos un método alternativo que no requiere tablas de verdad más adelante

Outline

Objetivos de la clase

- Conocer las formas normales
- Comprender concepto de consecuencia lógica
- Demostrar consecuencias lógicas sencillas