# Lógica Computacional
# Nota 10. Formas normales para la lógica de predicados.*

Noé Salomón Hernández S.

## 1. Introducción

La *regla de resolución* es una regla de inferencia para la lógica de predicados análoga a de la lógica proposicional. Tal regla funciona únicamente sobre expresiones en *forma clausular*.

## 2. FNCP y Forma Clausular

Ocuparemos la noción de literal que se introdujo en la nota pasada.

**Definición 2.1.** Una expresión está en *forma normal conjuntiva* syss es una conjunción de disyunciones de literales.

La forma normal conjuntiva se utilizó de forma implícita para obtener la forma clausular en la lógica proposicional.

**Definición 2.2.** Una fórmula de la lógica de predicados está en *forma normal conjuntiva prenex*[1] *(FNCP)* syss es de la forma:

$$Q_1x_1 \cdots Q_nx_n \; M$$

donde las $Q_i$ son cuantificadores y $M$ es una fórmula libre de cuantificadores en *forma normal conjuntiva*. A la secuencia $Q_1x_1 \cdots Q_nx_n$ se le conoce como *prefijo* y a $M$ como *matriz*.

**Definición 2.3.** Sea $\varphi$ un enunciado, es decir, una fórmula *cerrada*, en FNCP cuyo prefijo consiste únicamente de cuantificadores universales. La *forma clausular* de $\varphi$ consiste en la matriz de $\varphi$ escrita como un conjunto de cláusulas.

**Definición 2.4.** Sean $\varphi$ y $\psi$ dos fórmulas de la lógica de predicados. La expresión $\varphi \approx \psi$ denota que $\varphi$ es satisfacible syss $\psi$ es satisfacible.

*Esta nota se base en material elaborado por el prof. Favio Miranda y en el libro de Ben-Ari M., *Mathematical Logic for Computer Science*

[1] *Prenex* del latín tardío *praenexus*, atado o atado por enfrente, del latín *prae- pre-* + *nexus*, pasado participio de *nectere*, atar, ligar.https://www.merriam-webster.com/dictionary/prenex%20normal%20form

Es importante entender que $\varphi \approx \psi$ ($\varphi$ es satisfacible syss $\psi$ es satisfacible) *no* implica que $\varphi \equiv \psi$ ($\varphi$ es lógicamente equivalente a $\psi$). Por un lado, $\varphi \approx \psi$ indica que *existe* un modelo para $\varphi$ syss *existe* un modelo para $\psi$. Por otro lado, la equivalencia lógica $\varphi \equiv \psi$ quiere decir que *todo* modelo $\mathcal{M}$ satisface a $\varphi$ syss también satisface a $\psi$.

**Teorema 2.5** (Skolem)**.** *Sea $\varphi$ una fórmula cerrada. Entonces existe una fórmula $\psi$ en forma clausular tal que $\varphi \approx \psi$.*

Es claro que podemos transformar la $\varphi$ del Teorema 2.5 en una fórmula lógicamente equivalente en FNCP. Es la eliminación de los cuantificadores existenciales lo que causa que la nueva fórmula no sea equivalente a la original. Esta eliminación se consigue definiendo nuevos símbolos de función. En $\forall x \exists y P(x, y)$, los cuantificadores pueden leerse de la siguiente manera: todo $x$ *produce* un valor asociado $y$ tal que el predicado $P$ es verdadero. Pero nuestro concepto intuitivo de una función es el mismo: $y = f(x)$ significa que, dado $x$, $f$ produce un valor $y$ asociado con $x$. El cuantificador existencial puede eliminarse, resultando en $\forall x P(x, f(x))$.

**Ejemplo 2.6.** Consideremos el modelo $\mathcal{M}$ para la fórmula en FNCP $\varphi = \forall x \exists y P(x, y)$, donde el universo es $\mathbb{Z}$ y el predicado $P^{\mathcal{M}} \stackrel{def}{=} >$. Obviamente, $\mathcal{M} \models \varphi$.

La fórmula $\varphi' = \forall x P(x, f(x))$ se obtiene a partir de $\varphi$ eliminando el cuantificador existencial y reemplazándolo por una función. Consideremos el modelo $\mathcal{M}'$ idéntico a $\mathcal{M}$ pero con $f^{\mathcal{M}'}(x) \stackrel{def}{=} x + 1$. Claramente, $\mathcal{M}' \models \varphi$ (simplemente se ignora la función), pero $\mathcal{M}' \not\models \varphi'$ ya que no es cierto que $n > n + 1$ para los enteros. Por lo tanto, $\varphi' \not\equiv \varphi$.

Sin embargo, sí existe un modelo para $\varphi'$, por ejemplo $\mathcal{M}^\star$, donde el universo es $\mathbb{Z}$, $P^{\mathcal{M}^\star} \stackrel{def}{=} >$ y $f^{\mathcal{M}^\star}(x) \stackrel{def}{=} x - 1$.

# 3. Algoritmo de Skolem

Damos ahora un algoritmo para transformar una fórmula de la lógica de predicados $\varphi$ en una fórmula $\psi$ en forma clausular.

**Algoritmo**
**Input**: Una fórmula *cerrada* $\varphi$ de la lógica de predicados.
**Output**: Una fórmula $\psi$ en forma clausular tal que $\varphi \approx \psi$.

Daremos los pasos del algoritmo a la par que lo aplicamos a la fórmula

$$\blacktriangleright \quad \varphi = \forall x \big(P(x) \to Q(x)\big) \to \big(\forall x P(x) \to \forall x Q(x)\big)$$

(L) Se limpia la fórmula de entrada $\varphi$.

- Eliminar cuantificadores múltiples mediante las siguientes equivalencias:

$$\forall x \forall x\ \varphi \equiv \forall x\ \varphi \qquad \exists x \forall x\ \varphi \equiv \forall x\ \varphi$$
$$\exists x \exists x\ \varphi \equiv \exists x\ \varphi \qquad \forall x \exists x\ \varphi \equiv \exists x\ \varphi$$

- Eliminar los cuantificadores vacuos mediante las siguientes equivalencias, donde $x$ **no** está libre en $\varphi$:

$$\forall x\ \varphi \equiv \varphi \qquad\qquad \exists x\ \varphi \equiv \varphi$$

- ▶ No es necesaria hacer ninguna transformación a la fórmula $\varphi$.

(I) Eliminar las implicaciones y bi-condicionales empleando las equivalencias:

$$\begin{array}{lcl} \varphi \to \psi & \equiv & \neg\varphi \lor \psi \\ \varphi \leftrightarrow \psi & \equiv & (\neg\varphi \lor \psi) \land (\varphi \lor \neg\psi) \end{array}$$

- ▶ $\neg\forall x\big(\neg P(x) \lor Q(x)\big) \lor \big(\neg\forall x P(x) \lor \forall x Q(x)\big)$

(N) Manipular la negación para permitir su efecto únicamente en fórmulas atómicas y quitar doble negaciones. Usar las siguientes equivalencias:

$$\begin{array}{lcl} \neg\neg\varphi & \equiv & \varphi \\ \neg(\varphi \land \psi) & \equiv & \neg\varphi \lor \neg\psi \\ \neg(\varphi \lor \psi) & \equiv & \neg\varphi \land \neg\psi \\ \neg\forall x\ \varphi & \equiv & \exists x\ \neg\varphi \\ \neg\exists x\ \varphi & \equiv & \forall x\ \neg\varphi \end{array}$$

- ▶ $\exists x\big(P(x) \land \neg Q(x)\big) \lor \big(\exists x \neg P(x) \lor \forall x Q(x)\big)$

(R) Renombrar variables ligadas para que ninguna variable se repita en los cuantificadores.

- ▶ $\exists x\big(P(x) \land \neg Q(x)\big) \lor \big(\exists y \neg P(y) \lor \forall z Q(z)\big)$

(E) Extraer cuantificadores de la matriz. Identificar un cuantificador más externo en la matriz, es decir, uno que no esté dentro del alcance de otro que también resida en la matriz, y extraerlo usando las siguientes equivalencias, donde $Q$ es un cuantificador y *op* es $\lor$ o $\land$:

$$\psi\ \ op\ \ Qx\ \varphi \equiv Qx\ (\psi\ \ op\ \ \varphi) \qquad\qquad Qx\ \varphi\ \ op\ \ \psi \equiv Qx\ (\varphi\ \ op\ \ \psi)$$

Repetir hasta que todos los cuantificadores aparezcan en el prefijo y ninguno en la matriz. Las equivalencias se pueden aplicar porque ninguna variable figura en dos o más cuantificadores.

- ▶ $\exists x \exists y \forall z\Big(\big(P(x) \land \neg Q(x)\big) \lor \big(\neg P(y) \lor Q(z)\big)\Big)$

(D) Usar las leyes distributivas para transformar la matriz a su FNC. La fórmula está ahora en FNCP.

$$\begin{array}{lcl} \varphi \lor (\psi \land \chi) & \equiv & (\varphi \lor \psi) \land (\varphi \lor \chi) \\ (\varphi \land \psi) \lor \chi & \equiv & (\varphi \lor \chi) \land (\psi \lor \chi) \end{array}$$

- ▶ $\exists x \exists y \forall z\Big(\big(P(x) \lor \neg P(y) \lor Q(z)\big) \land \big(\neg Q(x) \lor \neg P(y) \lor Q(z)\big)\Big)$

(S) Recorrer el prefijo de izquierda a derecha. Para todo cuantificador existencial $\exists x$ en $\varphi$, sean $y_1, \ldots y_n$ las variables cuantificadas universalmente que preceden a $\exists x$, y sea $f$ un ***nuevo*** símbolo de función de aridad $n$. Eliminar $\exists x$ y reemplazar en la matriz cada presencia de $x$ por $f(y_1, \ldots y_n)$, esta función se llama *función de Skolem*. Si no hay cuantificadores universales precediendo $\exists x$, reemplazar $x$ por una nueva constante, la cual se llama *constante de Skolem*. El proceso de reemplazar cuantificadores existenciales por funciones se llama *Skolemización*.

- $\forall z\Big(\big(P(a) \vee \neg P(b) \vee Q(z)\big) \wedge \big(\neg Q(a) \vee \neg P(b) \vee Q(z)\big)\Big)$

- La fórmula puede ser escrita en forma clausular al dejar a un lado los cuantificadores universales y al escribir la matriz como un conjunto de cláusulas.

$$\big\{\{P(a), \neg P(b), Q(z)\}, \{\neg Q(a), \neg P(b), Q(z)\}\big\}$$

Para recordar los pasos LINREDS pueden pensar en la frase:

- *Logic Is Necessary in Realizing Effective Durable Systems* (La lógica es necesaria en la realización de sistemas efectivos y durables).

Si buscan una frase en español se tiene la siguiente:

- *La INformática REvoluciona el Desarrollo de Sociedades.*

Notemos que el orden en el que se extraen los cuantificadores en el paso E puede ser otro, y una vez aplicadas las leyes distributivas obtenemos la siguiente fórmula en FNCP equivalente:

$$\forall z \exists x \exists y\Big(\big(P(x) \vee \neg P(y) \vee Q(z)\big) \wedge \big(\neg Q(x) \vee \neg P(y) \vee Q(z)\big)\Big)$$

Como los cuantificadores existenciales están precedidos por un cuantificador universal, la Skolemización produce funciones de un argumento, no constantes, como sigue:

$$\forall z\Big(\big(P(f(z)) \vee \neg P(g(z)) \vee Q(z)\big) \wedge \big(\neg Q(f(z)) \vee \neg P(g(z)) \vee Q(z)\big)\Big)$$

En forma clausular tenemos:

$$\big\{\{P(f(z)), \neg P(g(z)), Q(z)\}, \{\neg Q(f(z)), \neg P(g(z)), Q(z)\}\big\}$$

## 3.1. Demostración del algoritmo de Skolem

En el algoritmo anterior se emplean cinco equivalencias, es el remplazo de un cuantificador existencial por una función/constante de Skolem lo que ya las hace equivalentes, pero la satisfacibilidad se preserva. Suponga que:

$$\mathcal{M} \models \forall y_1 \dots \forall y_n \exists x P(y_1, \dots, y_n, x).$$

Necesitamos demostrar que existe un modelo $\mathcal{M}'$ tal que:

$$\mathcal{M}' \models \forall y_1 \dots \forall y_n P(y_1, \dots, y_n, f(y_1, \dots, y_n)).$$

$\mathcal{M}'$ se construye extendiendo $\mathcal{M}$. Se agrega una función $f^{(n)}$ definida para todo $\{c_1, \dots, c_n\} \subseteq A$ como:

$$f(c_1, \dots, c_n) = c_{n+1} \text{ para algún } c_{n+1} \in A, \text{ tal que } (c_1, \dots, c_n, c_{n+1}) \in P^{\mathcal{M}}.$$

Dado que $\mathcal{M} \models \forall y_1 \dots \forall y_n \exists x P(y_1, \dots, y_n, x)$, debe haber al menos un elemento $b$ en el dominio tal que $(c_1, \dots, c_n, b) \in P^{\mathcal{M}}$. Simplemente elegimos uno de ellos arbitrariamente y lo asignamos como el valor $c_{n+1}$ de $f(c_1, \ldots, c_n)$. La función de Skolem se elige como un nuevo símbolo de función, por lo que la definición de $f$ no entra en conflicto con ninguna función existente en $\mathcal{M}$.

Para demostrar que:

$$\mathcal{M}' \models \forall y_1 \ldots \forall y_n P(y_1, \ldots, y_n, f(y_1, \ldots, y_n)),$$

tomamos elementos $\{c_1, \ldots, c_n\}$ arbitrarios del dominio. Por construcción, $f(c_1, \ldots, c_n) = b$ para algún $b \in A$ y $(c_1, \ldots, c_n, b) \in P^{\mathcal{M}'}$. Como $c_1, \ldots, c_n$ eran arbitrarios:

$$\mathcal{M}' \models \forall y_1 \ldots \forall y_n P(y_1, \ldots, y_n, f(y_1, \ldots, y_n))$$

Esto completa una dirección de la prueba del Teorema de Skolem. La prueba de la inversa se deja como ejercicio.

# 4. Ejemplo

Transformar el siguiente enunciado a forma clausular.

$$\neg\exists y \left(P(y) \land \forall z \,(R(z) \to Q(y,z))\right)$$

(L) La fórmula queda igual.

(I) $\neg\exists y \left(P(y) \land \forall z \,(\neg R(z) \lor Q(y,z))\right)$.

(N)

$$\begin{aligned}
\neg\exists y \left(P(y) \land \forall z \,(\neg R(z) \lor Q(y,z))\right) &\equiv \forall y \,\neg\left(P(y) \land \forall z \,(\neg R(z) \lor Q(y,z))\right) \\
&\equiv \forall y \left(\neg P(y) \lor \neg\forall z \,(\neg R(z) \lor Q(y,z))\right) \\
&\equiv \forall y \left(\neg P(y) \lor \exists z \,\neg(\neg R(z) \lor Q(y,z))\right) \\
&\equiv \forall y \left(\neg P(y) \lor \exists z \,(\neg\neg R(z) \land \neg Q(y,z))\right) \\
&\equiv \forall y \left(\neg P(y) \lor \exists z \,(R(z) \land \neg Q(y,z))\right).
\end{aligned}$$

(R) La fórmula queda igual.

(E) $\forall y \exists z \left(\neg P(y) \lor (R(z) \land \neg Q(y,z))\right)$.

(D) $\forall y \exists z \left((\neg P(y) \lor R(z)) \land (\neg P(y) \lor \neg Q(y,z))\right)$.

(S) $\forall y \left((\neg P(y) \lor R(f(y))) \land (\neg P(y) \lor \neg Q(y, f(y)))\right)$.

▷ $\big\{\{\neg P(y), R(f(y))\}, \{\neg P(y), \neg Q(y, f(y))\}\big\}$, esta es la forma clausular que buscamos.

**Ejercicios 4.1.** Transformar las siguientes fórmulas a su forma clausular.

1. $\forall x \exists y \left(P(x, g(y)) \to \neg\forall x \, Q(x)\right)$.
2. $\forall x \exists y \left(\exists z \forall w \, R(x,y,z,w) \to \exists v \, P(v)\right)$.
3. $\forall x \big(P(x) \land \neg\exists y \,(Q(y) \land R(x,y))\big) \lor \neg\big(\forall y \, P(y) \land \neg\forall z \, P(z)\big)$.

4. $\exists x \big(T(x) \wedge \neg Q(x)\big) \to \forall x \big(R(x) \to \neg T(x)\big)$.
5. $\exists x \forall y\, P(x,y) \to \forall y \exists x P(x,y)$.
6. $\forall x \big(P(x) \to Q(x)\big) \to \big(\forall x\, P(x) \to \forall x\, Q(x)\big)$.