# Tema 4: Formas Prenex, de Skolem y Teorema de Herbrand

Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

Lógica Informática
(Tecnologías Informáticas)

# Contenido

# Formas Prenex en LPO

- En esta sección veremos cómo obtener formas normales para fórmulas de un LPO.
- Los métodos y equivalencias vistas en proposicional se aplican de forma análoga cuando se hace uso de las mismas conectivas.
- Por ello, daremos métodos para manipular los cuantificadores y obtener expresiones adecuadas a las que podremos posteriormente aplicar los mecanismos de normalización presentados para la lógica proposicional.

# Equivalencia entre fórmulas LPO

**Definición.** Si $F$ y $G$ son fórmulas de un lenguaje de primer orden, entonces $F$ y $G$ son **equivalentes**, $F \equiv G$, si y sólo si $F \leftrightarrow G$ es lógicamente válida.

- Para fórmulas cerradas, $F \equiv G$ si y sólo si $F$ y $G$ tienen los mismos modelos.
- Para fórmulas no cerradas la equivalencia queda como:

$$F(\vec{x}) \equiv G(\vec{x}) \text{ sii } \models \forall \vec{x}(F(\vec{x}) \leftrightarrow G(\vec{x}))$$

- Ejemplos:
  - Las equivalencias proposicionales son válidas en lógica de primer orden.
  - Otras equivalencias importantes expresan propiedades de los cuantificadores:

$$\forall x(F \wedge G) \equiv \forall x F \wedge \forall x G.$$

- Utilizaremos las equivalencias para **reescribir** las fórmulas.

# Forma Prenex

**Definición.**
Una fórmula $F$ de un lenguaje de primer orden está en **Forma Prenex** si es de la forma

$$Q_1 x_1 Q_2 x_2 \cdots Q_n x_n G(x_1, \ldots, x_n)$$

donde $Q_i \in \{\exists, \forall\}$ y $G(x_1, \ldots, x_n)$ es abierta.

- Es decir, en una fórmula en forma prenex todos los cuantificadores se encuentran a la cabeza de la fórmula.
- Decimos que una fórmula $H$ es una forma normal prenexa de $F$ si $F$ y $H$ son equivalentes y $H$ está en forma prenex.

**Objetivo:** Dada una fórmula, trasladar todos los cuantificadores a la parte inicial de la fórmula para obtener una forma prenex.

# Ejemplos

| Fórmula | ¿FP? |
|---|---|
| $\neg\exists x\ [P(x) \to \forall x\ P(x)]$ | no |
| $\forall x\ \exists y\ [P(x) \wedge \neg P(y)]$ | sí |
| $\forall x\ P(x) \vee \exists y\ Q(y)$ | no |
| $\forall x\ \exists y\ [P(x) \vee Q(y)]$ | sí |
| $\exists y\ \forall x\ [P(x) \vee Q(y)]$ | sí |
| $\neg(\forall x\ [P(x) \to Q(x)] \to \forall x\ [P(x) \to R(x)])$ | no |
| $\exists z\ \forall x\ \forall y\ [((\neg P(x) \vee Q(x)) \wedge (\neg Q(y) \vee R(y))) \wedge P(z)]$ | sí |

# Operaciones prenex

| Operación | Subfórmula | Cambiar por | Restricción |
|---|---|---|---|
| $(P1)$ | $\exists x G$ | $\exists y G\{x/y\}$ | $y \notin VL(G)$ |
| | $\forall x G$ | $\forall y G\{x/y\}$ | $y \notin VL(G)$ |
| $(P2)$ | $\neg \exists x F$ | $\forall x \neg F$ | |
| | $\neg \forall x G$ | $\exists x \neg G$ | |
| $(P3)$ | $\exists x G \vee H$ | $\exists x (G \vee H)$ | $x \notin VL(H)$ |
| | $G \vee \exists x H$ | $\exists x (G \vee H)$ | $x \notin VL(G)$ |
| | $\forall x G \vee H$ | $\forall x (G \vee H)$ | $x \notin VL(H)$ |
| | $G \vee \forall x H$ | $\forall x (G \vee H)$ | $x \notin VL(G)$ |
| $(P4)$ | $\exists x G \to H$ | $\forall x (G \to H)$ | $x \notin VL(H)$ |
| | $\forall x G \to H$ | $\exists x (G \to H)$ | $x \notin VL(H)$ |
| $(P5)$ | $G \to \exists x H$ | $\exists x (G \to H)$ | $x \notin VL(G)$ |
| | $G \to \forall x H$ | $\forall x (G \to H)$ | $x \notin VL(G)$ |
| $(P6)$ | $\exists x G \wedge H$ | $\exists x (G \wedge H)$ | $x \notin VL(H)$ |
| | $G \wedge \exists x H$ | $\exists x (G \wedge H)$ | $x \notin VL(G)$ |
| | $\forall x G \wedge H$ | $\forall x (G \wedge H)$ | $x \notin VL(H)$ |
| | $G \wedge \forall x H$ | $\forall x (G \wedge H)$ | $x \notin VL(G)$ |

siendo $VL(F)$ el conjunto de las variables libres de $F$.

# Transformación a forma prenex

**Lema**: Si $F'$ se obtiene de $F$ por aplicación de una operación prenex, entonces $F \equiv F'$.

**Teorema.**
Para cada fórmula $F$ existe otra fórmula $G$ en forma prenex que es equivalente a $F$.

**Demostración:** Basta seguir los siguientes pasos:

1. Si aparece la conectiva $\leftrightarrow$ la expresamos de forma equivalente usando $\rightarrow$ y $\wedge$.
2. Renombramos las variables de cada subfórmula elemental (es decir, de la forma $\exists v\, H$, o bien, $\forall v\, H$), para que no interfieran las cuantificaciones. Para ello utilizamos la operación $(P1)$.
3. Trasladamos negaciones usando $(P2)$.
4. Aplicamos las reglas $(P3)$–$(P6)$.

# Ejemplos de cálculo de forma normal prenexa

- Ejemplo 1:

$$\begin{aligned} & \forall x\, P(x) \lor \exists y\, Q(y) & \\ \equiv\; & \forall x\, [P(x) \lor \exists y\, Q(y)] & [\text{por P3}] \\ \equiv\; & \forall x\, \exists y\, [P(x) \lor Q(y)] & [\text{por P3}] \end{aligned}$$

- Ejemplo 2:

$$\begin{aligned} & \forall x\, P(x) \lor \exists y\, Q(y) & \\ \equiv\; & \exists y\, [\forall x\, P(x) \lor Q(y)] & [\text{por P3}] \\ \equiv\; & \exists y\, \forall x\, [P(x) \lor Q(y)] & [\text{por P3}] \end{aligned}$$

- Ejemplo 3:

$$\begin{aligned} & \forall x\, P(x) \lor \exists x\, Q(x) & \\ \equiv\; & \forall x\, P(x) \lor \exists y\, Q(y) & [\text{por P1}] \\ \equiv\; & \exists y\, [\forall x\, P(x) \lor Q(y) & [\text{por P3}] \\ \equiv\; & \exists y\, \forall x\, [P(x) \lor Q(y)] & [\text{por P3}] \end{aligned}$$

# Ejemplos

- Sea $F(z,y) \equiv \forall x \exists y (x+z=y) \to \exists z \forall x (x \cdot y = 0 + z)$
  1. $\forall x_1 \exists y_1 (x_1 + z = y_1) \to \exists z_2 \forall x_2 (x_2 \cdot y = 0 + z_2)$
  2. $\exists x_1 (\exists y_1 (x_1 + z = y_1) \to \exists z_2 \forall x_2 (x_2 \cdot y = 0 + z_2))$
  3. $\exists x_1 \forall y_1 ((x_1 + z = y_1) \to \exists z_2 \forall x_2 (x_2 \cdot y = 0 + z_2))$
  4. $\exists x_1 \forall y_1 \exists z_2 ((x_1 + z = y_1) \to \forall x_2 (x_2 \cdot y = 0 + z_2))$
  5. $\boxed{\exists x_1 \forall y_1 \exists z_2 \forall x_2 ((x_1 + z = y_1) \to (x_2 \cdot y = 0 + z_2))}$

- Sea $G(x) \equiv \exists x\, P(x,a) \to (\forall z\, P(x,z) \to \forall x\, P(x,a))$
  1. $\forall x_1\, (P(x_1,a) \to (\forall z_2\, P(x,z_2) \to \forall x_3\, P(x_3,a)))$
  2. $\forall x_1\, (P(x_1,a) \to \exists z_2\, (P(x,z_2) \to \forall x_3\, P(x_3,a)))$
  3. $\forall x_1\, (P(x_1,a) \to \exists z_2\, \forall x_3\, (P(x,z_2) \to P(x_3,a)))$
  4. $\boxed{\forall x_1\, \exists z_2\, \forall x_3\, (P(x_1,a) \to P(x,z_2) \to P(x_3,a))}$

# Esqueleto Proposicional

Veamos cómo podemos reducir el caso de Primer Orden al caso más sencillo y completo de Proposicional:

- Comenzaremos manipulando las fórmulas de LPO para obtener fórmulas prácticamente equivalentes (usando sus Formas Prenex) pero sintácticamente mucho más simples: la llamada **Forma de Skolem**.
- Posteriormente, mostramos un resultado que permite reducir la consistencia de las fórmulas LPO a un conjunto de fórmulas LP.
- **Pero**: al simplificar las fórmulas y que parezcan proposicionales, puede **incrementar el número de fórmulas**
- hasta el punto que, en algunos casos, **podemos necesitar una cantidad infinita de fórmulas proposicionales**
- **Pero no todo está perdido...**

# Formas de Skolem(I)

- **Objetivo:** Restringir la complejidad sintáctica de las fórmulas LPO, sin perder expresividad.
- Trabajaremos con **fórmulas cerradas**.
  - Primer paso: **trasladar los cuantificadores** a la cabeza obteniendo un forma normal prenexa.
  - Segundo paso: **eliminar los cuantificadores existenciales** hasta obtener una fórmula en forma prenex con sólo cuantificadores universales (**fórmula universal**).
- La fórmula universal obtenida al final de estas transformaciones es una **forma de Skolem** de la fórmula inicial.

# Formas de Skolem (II)

- **El proceso de eliminación aumenta el lenguaje** con nuevos símbolos de función y nuevas constantes.
- En general, **no hay equivalencia** entre la fórmula obtenida y la fórmula inicial, aunque **se conserva la (in)consistencia de la fórmula original**.
- **Idea de la eliminación de existenciales**: Supongamos que, trabajando con la aritmética, tenemos la fórmula

$$\forall x \exists y [x < y]$$

  Entonces **podríamos elegir** para cada $x$ un elemento que fuera mayor que $x$, y así definir una **nueva función** $x \mapsto f(x)$, y trabajaríamos con la fórmula

$$\forall x [x < f(x)]$$

  No es una fórmula de la aritmética, pero *expresa* la misma información que la original

# Equisatisfactibilidad

- Las fórmulas $F$ y $G$ son **equisatisfactibles (o equiconsistentes)** si:

  $F$ es satisfactible syss $G$ es satisfactible.

  Se representa por $F \approx G$
- Ejemplos:

$$\begin{array}{lcl}
\exists x\ Q(x) & \approx & Q(a) \\
\exists x\ Q(x) & \not\equiv & Q(a) \\
\forall x\ \exists y\ P(x,y) & \approx & \forall x\ P(x,f(x)) \\
\forall x\ \exists y\ P(x,y) & \not\equiv & \forall x\ P(x,f(x))
\end{array}$$

- La forma de Skolem será equisatisfactible a la original

# Funciones y constantes de Skolem

Para eliminar los cuantificadores existenciales de una fórmula en forma prenexa aplicamos las siguientes reglas:

- Por cada bloque del tipo $\exists x F(x)$ añadir una nueva constante $c$ y reescribir como $F(c)$.
  (se dice que $F(c)$ se obtiene a partir de la original introduciendo una **constante de Skolem**).
- Por cada bloque

  $$\forall x_1 \cdots \forall x_n \exists y F(x_1, \ldots, x_n, y)$$

  introducir un nuevo símbolo de función $g$ (de aridad $n$), y reescribir como

  $$\forall x_1 \ldots \forall x_n F(x_1, \ldots, x_n, g(x_1, \ldots, x_n))$$

  (se dice que la segunda fórmula se obtiene de la original introduciendo una **función de Skolem**).

# Funciones y constantes de Skolem: Propiedades

**Teorema.** La introducción de funciones y constantes de Skolem conserva la consistencia (es decir, son equisatisfactibles).

> Es decir, dada una fórmula $F$, si $F'$ se obtiene a partir de $F$ introduciendo una función o constante de Skolem,
>
> $$F \text{ tiene un modelo} \quad \Leftrightarrow \quad F' \text{ tiene un modelo.}$$

**Teorema.** La introducción de funciones y constantes de Skolem, **no aumenta el conocimiento deducible en el lenguaje original**, es decir:

> Si $F$ es una fórmula cerrada de un lenguaje de primer orden $L$ y $F'$ se obtiene a partir de $F$ introduciendo una función o una constante de Skolem, entonces para toda fórmula $H$ del lenguaje $L$ se tiene:
>
> $$F' \models H \quad \Leftrightarrow \quad F \models H$$

# Formas de Skolem

**Definición.** Sea $L$ un LPO y $F$ una fórmula cerrada de $L$. Una **forma de Skolem** de $F$ es una fórmula universal $G$ que se obtiene a partir de una forma prenexa de $F$ mediante introducciones sucesivas de funciones o constantes de Skolem.

**Teorema.** Dado un conjunto $\Sigma = \{F_1, \dots, F_n\}$ de fórmulas cerradas de $L$, sea $\Sigma'$ el conjunto formado por las fórmulas de Skolem de lo elementos de $\Sigma$. Entonces,

- $\Sigma$ tiene un modelo si y sólo si $\Sigma'$ tiene un modelo.
- Para toda fórmula $H$ del lenguaje $L$,

$$\Sigma \models H \quad \Leftrightarrow \quad \Sigma' \models H$$

# Ejemplo (I)

$\forall x \exists y \exists z (P(y,x) \wedge P(z,y))$

1. Dependencia de $y$ con respecto a $x$: introducimos una función de skolem $f_1$:

$$\forall x \exists z (P(f_1(x), x) \wedge P(z, f_1(x)))$$

2. Dependencia de $z$ con respecto a $x$: introducimos una función de skolem $f_2$:

$$\forall x (P(f_1(x), x) \wedge P(f_2(x), f_1(x)))$$

3. La fórmula universal

$$\forall x \, (P(f_1(x), x) \wedge P(f_2(x), f_1(x)))$$

es una **Forma de Skolem** de la fórmula inicial.

# Ejemplo (II)

$\exists x_1 \forall y_1 \exists z_2 \forall x_2 \exists y_2 ((x_1 + b = y_1) \to (x_2 \cdot y_2 = 0 + z_2))$

1. $\forall y_1 \exists z_2 \forall x_2 \exists y_2 ((c_1 + b = y_1) \to (x_2 \cdot y_2 = 0 + z_2))$
   [$c$ nueva constante]
2. $\forall y_1 \forall x_2 \exists y_2 ((c_1 + b = y_1) \to (x_2 \cdot y_2 = 0 + f(y_1)))$
   [$f$ nuevo símbolo de función de aridad 1]
3. $\forall y_1 \forall x_2 ((c_1 + b = y_1) \to (x_2 \cdot g(y_1, x_2) = 0 + f(y_1)))$
   [$g$ nuevo símbolo de función de aridad 2]
4. **Forma de Skolem**:
   $\forall y_1 \forall x_2 ((c_1 + b = y_1) \to (x_2 \cdot g(y_1, x_2) = 0 + f(y_1))))$

En el lenguaje $LO = \{<, =\}$

- $F \equiv \exists x \forall y (x < y \lor x = y)$
  Forma de Skolem: $\forall y (c < y \lor c = y)$
- $F \equiv \forall x \forall z (x < z \to \exists y (x < y \land y < z)$
  Forma de Skolem:
  $\forall x \forall z (x < z \to x < f(x, z) \land f(x, z) < z)$

# Cláusulas en LPO

- Una fórmula $F$ es un **literal** si es una fórmula atómica o la negación de una fórmula atómica.
- Una **cláusula** es una disyunción de literales. Por tanto, es una fórmula abierta.
- Como en el caso de la lógica proposicional, identificaremos una cláusula con el conjunto de los literales que aparecen en ella.
- Denotaremos por $\square$ a la cláusula vacía.
- Dada una fórmula $F$ de $L$ una forma clausal de $F$ es un conjunto de cláusulas $S$ (no necesariamente del lenguaje $L$) tal que es equisatisfactible con la original;

$$F \text{ tiene un modelo} \iff S \text{ tiene un modelo}$$

# Forma clausal

Para obtener una forma clausal de una fórmula $F(x_1, \ldots, x_n)$ seguimos los siguientes pasos:

1. Obtener el cierre universal de $F(x_1, \ldots, x_n)$. Es decir, la fórmula $G$ dada por $\forall x_1 \ldots \forall x_n\, F(x_1, \ldots, x_n)$.
2. Obtener una forma de Skolem de $G$. Dicha forma de Skolem es una fórmula universal $G_S$.
3. Eliminar los cuantificadores universales de $G_S$. Así obtenemos una fórmula abierta $H$.
4. Obtener una forma normal conjuntiva de $H$ (siguiendo el mismo procedimiento que en el caso proposicional). Dicha forma será
$$\bigwedge_{j=1}^{n} C_j$$
siendo cada $C_j$ una cláusula (o disyunción de literales).
5. La forma clausal de $F$ es el conjunto de cláusulas $S = \{C_1, \ldots, C_n\}$.

# Ejemplos

- $H(x,y) \to (u \neq z \to P(z,u))$
  Forma normal conjuntiva
  $$\neg H(x,y) \vee u = z \vee P(z,u)$$
  Forma clausal
  $$\{\{\neg H(x,y), u = z, P(z,u)\}\}$$
- $x < z \to (x < y \wedge y < z)$
  Forma normal conjuntiva
  $$((\neg(x < z) \vee x < y) \wedge (\neg(x < z) \vee y < z))$$
  Forma clausal
  $$\{\{\neg(x < z), x < y\}, \{\neg(x < z), y < z\}\}$$

# Forma clausal de un conjunto de fórmulas

- Una forma clausal de un conjunto $\Gamma$ es un conjunto de cláusulas $S$ tal que

$$\Gamma \text{ tiene un modelo} \quad \Leftrightarrow \quad S \text{ tiene un modelo}$$

- Si $\Gamma = \{F_1, \ldots, F_n\}$ es un conjunto de fórmulas podemos obtener una forma clausal de $\Gamma$ calculando, por el método anterior, una forma clausal $S_j$ para cada elemento $F_j$ de $\Gamma$. La forma clausal de $\Gamma$ es entonces el conjunto de cláusulas:

$$S = S_1 \cup \cdots \cup S_n$$

# Ejemplos

- $\forall x[\forall y\, H(x,y) \rightarrow \exists z \forall u(u \neq z \rightarrow P(z,u))]$
  1. $\forall x \exists y\, [H(x,y) \rightarrow \exists z \forall u(u \neq z \rightarrow P(z,u))]$
  2. $\forall x \exists y \exists z \forall u[H(x,y) \rightarrow (u \neq z \rightarrow P(z,u))]$
  3. $\forall x \exists z \forall u\, [H(x, f_1(x)) \rightarrow (u \neq z \rightarrow P(z,u))]$
  4. $\forall x \forall u[H(x, f_1(x)) \rightarrow (u \neq f_2(x) \rightarrow P(f_2(x), u))]$

  Forma clausal:

$$\{\{\neg H(x, f_1(x)), u = f_2(x), P(f_2(x), u)\}\}$$

- $\forall x \forall z(x < z \rightarrow \exists y(x < y \wedge y < z))$

  Forma clausal

$$\{\{\neg(x < z), x < f(x,z)\}, \{\neg(x < z), f(x,z) < z\}\}$$

# Extensión de Herbrand

Sea $\Gamma$ un conjunto de fórmulas de un lenguaje $L$.

- El cálculo de formas de Skolem reduce la consistencia de $\Gamma$ a la consistencia de un conjunto $\Sigma$ de fórmulas abiertas (de hecho, cláusulas) en un nuevo lenguaje $L'$ (que extiende a $L$ con nuevos símbolos).
- Es posible un paso más y reducir la consistencia de $\Gamma$ a la de un conjunto de fórmulas **abiertas y cerradas**.
  - Una fórmula abierta y cerrada puede identificarse con su esqueleto proposicional y, por tanto, considerarse una fórmula proposicional.

# Extensión de Herbrand (II)

- **Definición**. Sea $\Sigma$ un conjunto de fórmulas abiertas. La **extensión de Herbrand** de $\Sigma$, $EH(\Sigma)$, es el conjunto formado por todas las fórmulas cerradas que pueden obtenerse sustituyendo, de todas las formas posibles, las variables de las fórmulas de $\Sigma$ por términos cerrados.
- Ejemplo: En el lenguaje $\mathcal{L} = \{P, Q, a, b\}$, consideremos

  $$\{P(x) \to P(a), P(b) \vee Q(y)\}$$

  Su extensión es finita, está formada por

  $$\{P(a) \to P(a), P(b) \to P(a), P(b) \vee Q(a), P(b) \vee Q(b)\}$$

- Si hubiese símbolo de función, **sería infinita**

# Teorema de Herbrand

**Teorema**. Sea $\Sigma$ un conjunto de fórmulas abiertas.

Son equivalentes:

1. $\Sigma$ tiene un modelo.
2. $EH(\Sigma)$ tiene un modelo.
3. $EH(\Sigma)$ es satisfactible proposicionalmente.

   Es decir, si identificamos cada fórmula de $EH(\Sigma)$ con su esqueleto proposicional, entonces el conjunto de fórmulas proposicionales así obtenido es satisfactible.

# Observaciones

- Este resultado reduce la consistencia de $\Sigma$ a la de un conjunto de fórmulas proposicionales: $EH(\Sigma)$.
- $EH(\Sigma)$ **puede ser un conjunto infinito**. Sin embargo:
- **Teorema.** (Compacidad) Si $EH(\Sigma)$ es inconsistente, entonces existe un subconjunto finito $\Sigma_0$ de $EH(\Sigma)$ que es inconsistente.
- Por tanto disponemos de un algoritmo de semidecisión: Supongamos que $\{t_1, \dots\}$ son los términos cerrados del lenguaje

  **Procedimiento**: Para cada $n = 0, 1 \dots$
  - consideramos $EH(\Sigma)_n$ el conjunto de fórmulas de la extensión de Herbrand que se pueden construir con los términos $\{t_1, \dots, t_n\}$, y nos preguntamos por su inconsistencia
  - **Si es inconsistente**, entonces $\Sigma$ lo es

# Ejemplo

$\Sigma = \{p(x) \rightarrow p(f(f(x))), \neg p(f(f(f(x)))), p(f(a))\}$ es inconsistente.

- Su extensión de Herbrand $EH(\Sigma)$ contiene, entre otras, las fórmulas
  1. $p(f(a))$
  2. $p(a) \rightarrow p(f(f(a)))\}$ $[\{x/a\}]$
  3. $\neg p(f(f(f(a))))$ $[\{x/a\}]$
  4. $p(f(a)) \rightarrow p(f(f(f(a))))$ $[\{x/f(a)\}]$
  5. $\neg p(f(f(f(f(a)))))$ $[\{x/f(a)\}]$
  6. $p(f(f(a))) \rightarrow p(f(f(f(f(a)))))\}$ $[\{x/f(f(a))\}]$
  7. $\neg p(f(f(f(f(f(a))))))$ $[\{x/f(f(a))\}]$

     $\vdots$
- El subconjunto de $EH(\Sigma)$ formado por las fórmulas 1, 3 y 4 es inconsistente.