

# Fundamentos de Programación Lógica

## Paradigmas de Lenguajes de Programación

Departamento de Computación, FCEyN, UBA

28 de febrero de 2018

# Paradigma lógico

- ▶ Se basa en el uso de la lógica como un lenguaje de programación
- ▶ Se especifican
  - ▶ ciertos hechos y reglas de inferencia
  - ▶ un objetivo (“goal”) a probar
- ▶ Un motor de inferencia trata de probar que el objetivo es consecuencia de los hechos y reglas
- ▶ Es declarativo: se especifican hechos, reglas y objetivo sin indicar cómo se obtiene éste último a partir de los primeros

# Prolog

- ▶ Lenguaje de programación basado en este esquema que fue introducido a fines de 1971 (cf. “THE BIRTH OF PROLOG”, A. Colmerauer y P. Roussel, [www.lif-sud.univ-mrs.fr/~colmer/](http://www.lif-sud.univ-mrs.fr/~colmer/))
- ▶ Los programas se escriben en un subconjunto de la lógica de primer orden
- ▶ El mecanismo teórico en el que se basa es el [método de resolución](#)

# Prolog

## Ejemplo de programa

```
habla(ale, ruso).  
habla(juan, ingles).  
habla(maria, ruso).  
habla(maria, ingles).
```

```
seComunicaCon(X,Y):-habla(X,L),habla(Y,L),X\=Y
```

## Ejemplo de goal

```
seComunicaCon(X,ale)
```

# Nuestro enfoque

1. Método de resolución para lógica proposicional (Clase 1/2)
2. Método de resolución para lógica de primer orden (Clase 1/2)
3. Cláusulas de Horn y resolución SLD, programación lógica (Clase 2/2)

# Repaso lógica proposicional: Sintaxis

Dado un conjunto  $\mathcal{V}$  de variables proposicionales  $P, P_0, P_1, \dots$ , el conjunto de fórmulas proposicionales (o proposiciones) se define como

$A, B :=$	$P$	una variable proposicional
	$\neg A$	negación
	$A \wedge B$	conjunción
	$A \vee B$	disjunción
	$A \supset B$	implicación
	$A \iff B$	si y sólo si

## Algunas fórmulas proposicionales

- ▶  $A \vee \neg B$ ,
- ▶  $(A \wedge B) \supset (A \vee A)$

## Repaso lógica proposicional: Semántica

- ▶ Una **valuación** es una función  $v : \mathcal{V} \rightarrow \{\mathbf{T}, \mathbf{F}\}$  que asigna valores de verdad a las variables proposicionales
- ▶ Una valuación **satisface** una proposición  $A$  si  $v \models A$  donde:

$$v \models P \quad \text{sii} \quad v(P) = \mathbf{T}$$

$$v \models \neg A \quad \text{sii} \quad v \not\models A \text{ (i.e. no } v \models A)$$

$$v \models A \vee B \quad \text{sii} \quad v \models A \text{ o } v \models B$$

$$v \models A \wedge B \quad \text{sii} \quad v \models A \text{ y } v \models B$$

$$v \models A \supset B \quad \text{sii} \quad v \not\models A \text{ o } v \models B$$

$$v \models A \iff B \quad \text{sii} \quad (v \models A \text{ sii } v \models B)$$

# Repaso: Tautologías y satisfactibilidad

Una proposición  $A$  es

- ▶ **satisfactible** si existe una valuación  $v$  tal que  $v \models A$
- ▶ una **tautología** si  $v \models A$  para toda valuación  $v$
- ▶ **insatisfactible** si no es satisfactible

Un conjunto de proposiciones  $S$  es

- ▶ **satisfactible** si existe una valuación  $v$  tal que para todo  $A \in S$ , se tiene  $v \models A$
- ▶ **insatisfactible** si no es satisfactible



# Forma normal conjuntiva (FNC)

- ▶ Un **Literal** es una variable proposicional  $P$  o su negación  $\neg P$
- ▶ Una proposición  $A$  está en **FNC** si es una conjunción

$$C_1 \wedge \dots \wedge C_n$$

donde cada  $C_i$  (llamado **cláusula**) es una disyunción

$$B_{i1} \vee \dots \vee B_{in_i}$$

y cada  $B_{ij}$  es un literal

- ▶ Una FNC es una “conjunción de disyunciones de literales”

# Forma normal conjuntiva

## Ejemplos

- ▶  $(P \vee Q) \wedge (P \vee \neg Q)$  está en FNC
- ▶  $(P \vee Q) \wedge (P \vee \neg\neg Q)$  no está en FNC
- ▶  $(P \wedge Q) \vee P$  no está en FNC

## Teorema

Para toda proposición  $A$  puede hallarse una proposición  $A'$  en FNC que es lógicamente equivalente a  $A$ .

## Nota

$A$  es lógicamente equivalente a  $B$  sii  $A \iff B$  es una tautología

# Notación conjuntista para FNC

- ▶ Dado que tanto  $\vee$  como  $\wedge$ 
  1. son conmutativos (i.e.  $(A \vee B) \iff (B \vee A)$ )
  2. son asociativos (i.e.  $((A \vee B) \vee C) \iff (A \vee (B \vee C))$ )
  3. son idempotentes (i.e.  $(A \vee A) \iff A$ )

Podemos asumir que

1. Cada cláusula  $C_i$  es **distinta**
2. Cada cláusula puede verse como un conjunto de literales **distintos**

## Notación conjuntista para FNC

Consecuentemente para una FNC podemos usar la notación

$$\{C_1, \dots, C_n\}$$

donde cada  $C_i$  es un **conjunto** de literales

$$\{B_{i1}, \dots, B_{in_i}\}$$

Por ejemplo, la FNC  $(P \vee Q) \wedge (P \vee \neg Q)$  se anota

$$\{\{P, Q\}, \{P, \neg Q\}\}$$

# Repaso: Tautologías e insatisfactibilidad

## Teorema

Una proposición  $A$  es una tautología sii  $\neg A$  es insatisfactible

## Notar

Este resultado sugiere un método **indirecto** para probar que una proposición  $A$  es una tautología, a saber probar que  $\neg A$  es **insatisfactible**

# Validez por refutación

Principio de demostración por **refutación**:

Probar que  $A$  es **tautología** mostrando que  $\neg A$  es **insatisfactible**

- ▶ Hay varias técnicas de demostración por refutación
  - ▶ Tableaux semántico (1960)
  - ▶ Procedimiento de Davis-Putnam (1960)
  - ▶ Resolución (1965)
- ▶ Nos vamos a concentrar en **Resolución**

# Resolución

- ▶ Introducido por Alan Robinson en 1965

A MACHINE-ORIENTED LOGIC BASED ON THE RESOLUTION PRINCIPLE, J. of the ACM (12).

- ▶ Es simple de implementar
- ▶ Popular en el ámbito de demostración automática de teoremas
- ▶ Tiene una única regla de inferencia: [la regla de resolución](#)
- ▶ Si bien no es imprescindible, es conveniente asumir que las proposiciones están en [forma normal conjuntiva](#)

# Principio fundamental del método de resolución

- ▶ Se basa en el hecho de que la siguiente proposición es una tautología

$$(A \vee P) \wedge (B \vee \neg P) \iff (A \vee P) \wedge (B \vee \neg P) \wedge (A \vee B)$$

- ▶ En efecto, el conjunto de cláusulas

$$\{C_1, \dots, C_m, \{A, P\}, \{B, \neg P\}\}$$

es lógicamente equivalente a

$$\{C_1, \dots, C_m, \{A, P\}, \{B, \neg P\}, \{A, B\}\}$$



# Resolución

- ▶ En consecuencia, el conjunto de cláusulas

$$\{C_1, \dots, C_m, \{A, P\}, \{B, \neg P\}\}$$

es **insatisfactible** sii

$$\{C_1, \dots, C_m, \{A, P\}, \{B, \neg P\}, \{A, B\}\}$$

es **insatisfactible**

- ▶ La cláusula  $\{A, B\}$  se llama **resolvente** de las cláusulas  $\{A, P\}$  y  $\{B, \neg P\}$
- ▶ El resolvente de las cláusulas  $\{P\}$  y  $\{\neg P\}$  es la **cláusula vacía** y se anota  $\square$

# Regla de resolución

- ▶ Dado un literal  $L$ , el **opuesto** de  $L$  (escrito  $\overline{L}$ ) se define como:
  - ▶  $\neg P$  si  $L = P$
  - ▶  $P$  si  $L = \neg P$
- ▶ Dadas dos cláusulas  $C_1, C_2$ , una cláusula  $C$  se dice **resolvente de  $C_1$  y  $C_2$**  sii, para algún literal  $L$ ,  $L \in C_1$ ,  $\overline{L} \in C_2$ , y

$$C = (C_1 - \{L\}) \cup (C_2 - \{\overline{L}\})$$

## Ejemplo

Las cláusulas  $\{A, B\}$  y  $\{\neg A, \neg B\}$  tienen dos resolventes:  $\{A, \neg A\}$  y  $\{B, \neg B\}$ . Las cláusulas  $\{P\}$  y  $\{\neg P\}$  tienen a la cláusula vacía como resolvente

## Regla de resolución

$$\frac{\{A_1, \dots, A_m, Q\} \quad \{B_1, \dots, B_n, \neg Q\}}{\{A_1, \dots, A_m, B_1, \dots, B_n\}}$$

# El método de resolución

El proceso de agregar a un conjunto  $S$  la resolvente  $C$  de dos cláusulas  $C_1, C_2$  que pertenecen a  $S$  (i.e. de aplicar la regla de resolución a  $S$ ) se llama un **paso de resolución**.

Nota:

- ▶ Asumiremos que el resolvente  $C$  que se agrega a  $S$  **no** pertenecía ya a  $S$
- ▶ Pasos de resolución preservan insatisfactibilidad

$S$  es **insatisfactible** sii  $S \cup \{C\}$  es **insatisfactible**

# El método de resolución

- ▶ Un conjunto de cláusulas se llama una **refutación** si contiene a la cláusula vacía (i.e. a  $\square$ ).
- ▶ El método de resolución trata de construir una secuencia de conjuntos de cláusulas, obtenidas usando **pasos de resolución** hasta llegar a una **refutación**.

$$S_1 \Rightarrow S_2 \Rightarrow \dots \Rightarrow S_{n-1} \Rightarrow S_n \ni \square$$

- ▶ En ese caso se sabe que el conjunto inicial de cláusulas es insatisfactible dado que
  1. cada paso de resolución preserva insatisfactibilidad
  2. el último conjunto de cláusulas es insatisfactible (contiene la cláusula vacía)

## Ejemplo

**Objetivo:** mostrar que el conjunto de cláusulas  
 $\{\{P, Q\}, \{P, \neg Q\}, \{\neg P, Q\}, \{\neg P, \neg Q\}\}$  es insatisfactible.

1.  $\{\{P, Q\}, \{P, \neg Q\}, \{\neg P, Q\}, \{\neg P, \neg Q\}\}$
2.  $\{\{P, Q\}, \{P, \neg Q\}, \{\neg P, Q\}, \{\neg P, \neg Q\}, \{P\}\}$
3.  $\{\{P, Q\}, \{P, \neg Q\}, \{\neg P, Q\}, \{\neg P, \neg Q\}, \{P\}, \{Q\}\}$
4.  $\{\{P, Q\}, \{P, \neg Q\}, \{\neg P, Q\}, \{\neg P, \neg Q\}, \{P\}, \{Q\}, \{\neg P\}\}$
5.  $\{\{P, Q\}, \{P, \neg Q\}, \{\neg P, Q\}, \{\neg P, \neg Q\}, \{P\}, \{Q\}, \{\neg P\}, \square\}$

## Ejemplo

**Objetivo:** mostrar que el conjunto de cláusulas  $S = \{\{A, B, C\}, \{A\}, \{B\}\}$  es insatisfactible.

- ▶ No podemos aplicar ningún paso de resolución a  $S$
- ▶ Por lo tanto, no puede llegarse a una refutación a partir  $S$
- ▶  $S$  debe ser satisfactible
- ▶ En efecto, tomar por ejemplo  $v(A) = v(B) = \mathbf{T}$

# Terminación de la regla de resolución

- ▶ La aplicación reiterada de la regla de resolución **siempre termina** (suponiendo que el resolvente que se agrega es nuevo)
- ▶ En efecto, notar que
  1. El resolvente (i.e. la cláusula nueva que se agrega) se forma con los literales distintos que aparecen en el conjunto de cláusulas de partida  $S$
  2. Hay una cantidad **finita** de literales en el conjunto de cláusulas de partida  $S$
- ▶ En el peor de los casos, la regla de resolución podrá generar una nueva cláusula por cada combinación diferente de literales distintos de  $S$



# Corrección y completitud

- ▶ El siguiente resultado establece la corrección y completitud del método de resolución

## Teorema

Dado un conjunto finito  $S$  de cláusulas,

$S$  es insatisfactible sii tiene una refutación

# Repaso Lógica de Primer Orden

Un lenguaje de primer orden (LPO)  $\mathcal{L}$  consiste en:

1. Un conjunto numerable de constantes  $c_0, c_1, \dots$
2. Un conjunto numerable de símbolos de función con aridad  $n > 0$  (indica el número de argumentos)  $f_0, f_1, \dots$
3. Un conjunto numerable de símbolos de predicado con aridad  $n \geq 0$ ,  $P_0, P_1, \dots$ . La aridad indica el número de argumentos que toma (si  $n = 0$ , es una variable proposicional)

Ejemplo: Lenguaje de primer orden para la aritmética

Constantes: 0; Símbolos de función:  $S, +, *$ ; Símbolos de predicado:  $<, =$ .

# Términos de primer orden

Sea  $\mathcal{V} = \{x_0, x_1, \dots\}$  un conjunto numerable de variables y  $\mathcal{L}$  un LPO. El conjunto de  $\mathcal{L}$ -términos se define inductivamente como:

1. Toda constante de  $\mathcal{L}$  y toda variable es un  $\mathcal{L}$ -término
2. Si  $t_1, \dots, t_n \in \mathcal{L}$ -términos y  $f$  es un símbolo de función de aridad  $n$ , entonces  $f(t_1, \dots, t_n) \in \mathcal{L}$ -términos

**Ejemplo: Aritmética (cont.)**

$S(0), +(S(0), S(S(0))), *(S(x_1), +(x_2, S(x_3)))$

# Fórmulas atómicas

Sea  $\mathcal{V}$  un conjunto numerable de variables y  $\mathcal{L}$  un LPO. El conjunto de  $\mathcal{L}$ -fórmulas atómicas se define inductivamente como:

1. Todo símbolo de predicado de aridad 0 es una  $\mathcal{L}$ -fórmula atómica
2. Si  $t_1, \dots, t_n \in \mathcal{L}$ -términos y  $P$  es un símbolo de predicado de aridad  $n$ , entonces  $P(t_1, \dots, t_n)$  es una  $\mathcal{L}$ -fórmula atómica

Ejemplo: Aritmética (cont.)

$< (0, S(0)), < (x_1, +(S(0), x_2))$

# Fórmulas de primer orden

Sea  $\mathcal{V}$  un conjunto numerable de variables y  $\mathcal{L}$  un LPO. El conjunto de  $\mathcal{L}$ -fórmulas se define inductivamente como:

1. Toda  $\mathcal{L}$ -fórmula atómica es una  $\mathcal{L}$ -fórmula
2. Si  $A, B \in \mathcal{L}$ -fórmulas, entonces  $\neg A$ ,  $(A \wedge B)$ ,  $(A \vee B)$ ,  $(A \supset B)$ , y  $(A \iff B)$  son  $\mathcal{L}$ -fórmulas
3. Para toda variable  $x_i$  y cualquier  $\mathcal{L}$ -fórmula  $A$ ,  $\forall x_i.A$  y  $\exists x_i.A$  son  $\mathcal{L}$ -fórmulas

## Ejemplo: Aritmética (cont.)

- ▶  $\forall x.\forall y.(x < y \supset \exists z.y = x + z)$
- ▶  $\forall x.\forall y.((x < y \vee y < x) \vee x = y)$

# Variables libres y ligadas

Las variables pueden ocurrir **libres** o **ligadas**.

- ▶ Los cuantificadores ligan variables
- ▶ Usamos  $FV(A)$  y  $BV(A)$  para referirnos a las variables libres y ligadas, resp., de  $A$
- ▶  $FV(A)$  y  $BV(A)$  se pueden definir por inducción estructural en  $A$

## Ejemplo

Si  $A = \forall x.(R(x, y) \supset P(x))$ , entonces  $FV(A) = \{y\}$  y  $BV(A) = \{x\}$

# Variables libres y ligadas

- ▶ Una fórmula  $A$  se dice **rectificada** si
  - ▶  $FV(A)$  y  $BV(A)$  son disjuntos y
  - ▶ Cuantificadores distintos de  $A$  ligan variables distintas
- ▶ Toda fórmula se puede **rectificar** (renombrando variables ligadas) a una fórmula lógicamente equivalente
- ▶ Una **sentencia** es una fórmula cerrada (i.e. sin variables libres).

# Estructura de primer orden

Dado un lenguaje de primer orden  $\mathcal{L}$ , una **estructura para  $\mathcal{L}$** ,  $\mathbf{M}$ , es un par  $\mathbf{M} = (M, I)$  donde

- ▶  $M$  (**dominio**) es un conjunto no vacío
- ▶  $I$  (**función de interpretación**) asigna funciones y predicados sobre  $M$  a símbolos de  $\mathcal{L}$  de la siguiente manera:
  1. Para toda constante  $c$ ,  $I(c) \in M$
  2. Para todo  $f$  de aridad  $n > 0$ ,  $I(f) : M^n \rightarrow M$
  3. Para todo predicado  $P$  de aridad  $n \geq 0$ ,  $I(P) : M^n \rightarrow \{\mathbf{T}, \mathbf{F}\}$



# Satisfactibilidad

## Asignación

Sea  $\mathbf{M}$  una estructura para  $\mathcal{L}$ . Una **asignación** es una función  $s : \mathcal{V} \rightarrow M$

- ▶ Si  $s$  es una asignación y  $a \in M$ , usamos la notación  $s[x \leftarrow a]$  para denotar la asignación que se comporta igual que  $s$  salvo en el elemento  $x$ , en cuyo caso retorna  $a$

## Satisfactibilidad

La relación  $s \models_{\mathbf{M}} A$  establece que la asignación  $s$  satisface la fórmula  $A$  en la estructura  $\mathbf{M}$

- ▶ Vamos a definir la relación  $s \models_{\mathbf{M}} A$  usando inducción estructural en  $A$

# Satisfactibilidad

La relación  $s \models_{\mathbf{M}} A$  se define inductivamente como:

$s \models_{\mathbf{M}} P(t_1, \dots, t_n)$	<i>sii</i>	$P_{\mathbf{M}}(s(t_1), \dots, s(t_n))$
$s \models_{\mathbf{M}} \neg A$	<i>sii</i>	$s \not\models_{\mathbf{M}} A$
$s \models_{\mathbf{M}} (A \wedge B)$	<i>sii</i>	$s \models_{\mathbf{M}} A$ y $s \models_{\mathbf{M}} B$
$s \models_{\mathbf{M}} (A \vee B)$	<i>sii</i>	$s \models_{\mathbf{M}} A$ o $s \models_{\mathbf{M}} B$
$s \models_{\mathbf{M}} (A \supset B)$	<i>sii</i>	$s \not\models_{\mathbf{M}} A$ o $s \models_{\mathbf{M}} B$
$s \models_{\mathbf{M}} (A \iff B)$	<i>sii</i>	$(s \models_{\mathbf{M}} A \text{ sii } s \models_{\mathbf{M}} B)$
$s \models_{\mathbf{M}} \forall x_i. A$	<i>sii</i>	$s[x_i \leftarrow a] \models_{\mathbf{M}} A$ para todo $a \in M$
$s \models_{\mathbf{M}} \exists x_i. A$	<i>sii</i>	$s[x_i \leftarrow a] \models_{\mathbf{M}} A$ para algún $a \in M$

# Validez

- ▶ Una fórmula  $A$  es **satisfactible en  $\mathbf{M}$**  sii existe una asignación  $s$  tal que

$$s \models_{\mathbf{M}} A$$

- ▶ Una fórmula  $A$  es **satisfactible** sii existe un  $\mathbf{M}$  tal que  $A$  es satisfactible en  $\mathbf{M}$ . En caso contrario se dice que  $A$  es **insatisfactible**.
- ▶ Una fórmula  $A$  es **válida en  $\mathbf{M}$**  sii

$$s \models_{\mathbf{M}} A, \text{ para toda asignación } s$$

- ▶ Una fórmula  $A$  es **válida** sii es válida en toda estructura  $\mathbf{M}$ .
- ▶ **Nota:**  $A$  es válida sii  $\neg A$  es insatisfactible.

# Teorema de Church

No existe un algoritmo que pueda determinar si una fórmula de primer orden es válida

- ▶ Como consecuencia el método de resolución que veremos para la lógica de primer orden **no es un procedimiento efectivo** (i.e. un algoritmo)
- ▶ Es un procedimiento de **semi-decisión**:
  - ▶ si una sentencia es insatisfactible hallará una refutación,
  - ▶ pero si es satisfactible puede que no se detenga

# Forma clausal

- ▶ Es una forma normal conjuntiva, en notación de conjuntos.
- ▶ Análogo a la forma clausal del marco proposicional.
- ▶ Pero requiere tener en cuenta los **cuantificadores**.
- ▶ Consiste en seis pasos de conversión.
  1. Escribir la fórmula en términos de  $\wedge, \vee, \neg, \forall, \exists$  (i.e. eliminar implicación).
  2. Pasar a **forma normal negada**.
  3. Pasar a **forma normal prenexa** (opcional).
  4. Pasar a **forma normal de Skolem**.
  5. Pasar matriz a **forma normal conjuntiva**.
  6. **Distribuir** cuantificadores universales.

# Forma normal negada

El conjunto de fórmulas en **forma normal negada** (NNF) se define inductivamente como:

1. Para cada fórmula atómica  $A$ ,  $A$  y  $\neg A$  están en NNF.
2. Si  $A, B \in \text{NNF}$ , entonces  $(A \vee B), (A \wedge B) \in \text{NNF}$ .
3. Si  $A \in \text{NNF}$ , entonces  $\forall x.A, \exists x.A \in \text{NNF}$ .

## Ejemplos

- ▶  $\neg \exists x.((P(x) \vee \exists y.R(x, y)) \supset (\exists z.R(x, z) \vee P(a)))$  no está en NNF.
- ▶  $\forall x.((P(x) \vee \exists y.R(x, y)) \wedge (\forall z.\neg R(x, z) \wedge \neg P(a)))$  está en NNF.

# Forma normal negada

Toda fórmula es **lógicamente equivalente** a otra en NNF.

Dem.

Por inducción estructural usando:

$$\begin{array}{llll} \neg(A \wedge B) & \Longleftrightarrow & \neg A \vee \neg B & \\ \neg(A \vee B) & \Longleftrightarrow & \neg A \wedge \neg B & \\ \neg\neg A & \Longleftrightarrow & A & \end{array} \quad \begin{array}{ll} \neg\forall x.A & \Longleftrightarrow \exists x.\neg A \\ \neg\exists x.A & \Longleftrightarrow \forall x.\neg A \end{array}$$

## Ejemplos

- ▶  $\neg\exists x.(\neg(P(x) \vee \exists y.R(x, y)) \vee (\exists z.R(x, z) \vee P(a)))$  se transforma en
- ▶  $\forall x.((P(x) \vee \exists y.R(x, y)) \wedge (\forall z.\neg R(x, z) \wedge \neg P(a)))$

# Forma normal prenexa

Fórmula de la forma  $Q_1x_1 \dots Q_nx_n.B$ ,  $n \geq 0$ , donde

- ▶  $B$  sin cuantificadores (llamada **matriz**)
- ▶  $x_1, \dots, x_n$  son variables
- ▶  $Q_i \in \{\forall, \exists\}$



## Forma prenexa

Toda fórmula  $A$  es lógicamente equivalente a una fórmula  $B$  en forma prenexa.

### Demostración

Por inducción estructural usando:

$$\begin{array}{ll} (\forall x.A) \wedge B \iff \forall x.(A \wedge B) & (\forall x.A) \vee B \iff \forall x.(A \vee B) \\ (A \wedge \forall x.B) \iff \forall x.(A \wedge B) & (A \vee \forall x.B) \iff \forall x.(A \vee B) \\ (\exists x.A) \wedge B \iff \exists x.(A \wedge B) & (\exists x.A) \vee B \iff \exists x.(A \vee B) \\ (A \wedge \exists x.B) \iff \exists x.(A \wedge B) & (A \vee \exists x.B) \iff \exists x.(A \vee B) \end{array}$$

- **Nota:** Con estas equivalencias basta, si asumimos que  $A$  está en NNF.

## Ejemplo

1.  $\forall x. \neg P(x) \wedge (\exists y. Q(y) \vee \forall z. P(z))$
2.  $\forall x. \neg P(x) \wedge (\exists y. (Q(y) \vee \forall z. P(z)))$
3.  $\exists y. (\forall x. \neg P(x) \wedge (Q(y) \vee \forall z. P(z)))$
4.  $\exists y. (\forall x. \neg P(x) \wedge \forall z. (Q(y) \vee P(z)))$
5.  $\exists y. \forall z. (\forall x. \neg P(x) \wedge (Q(y) \vee P(z)))$
6.  $\exists y. \forall z. \forall x. (\neg P(x) \wedge (Q(y) \vee P(z)))$

# Forma normal de Skolem

- ▶ Hasta ahora tenemos una fórmula que:
  1. está escrita en términos de  $\wedge, \vee, \neg, \forall, \exists$ ,
  2. si tiene negaciones, solamente se aplican a átomos (**forma normal negada**),
  3. (opcionalmente) si tiene cuantificadores, se encuentran todos en el prefijo (**forma normal prenexa**).
- ▶ El proceso de pasar una fórmula a forma normal de Skolem se llama **skolemización**.
- ▶ El objetivo de la **skolemización** es
  1. eliminar los cuantificadores existenciales
  2. **sin** alterar la **satisfactibilidad**.

# Eliminación de cuantificadores existenciales

- ▶ ¿Cómo eliminamos los  $\exists$  sin cambiar la satisfactibilidad?
- ▶ Introducimos “testigos” para los mismos.
  - ▶ Todo cuantificador existencial se reemplaza por una constante o función de skolem.
  - ▶ Ejemplo:  $\exists x.P(x)$  se skolemiza a  $P(c)$  donde  $c$  es una nueva constante que se agrega al lenguaje de primer orden.
  - ▶ Estas funciones y constantes se suelen conocer como **parámetros**.

# ¿Cómo se altera el significado de la fórmula?

## Prop.

Si  $A'$  es el resultado de skolemizar  $A$ , entonces  $A$  es satisfactible sii  $A'$  es satisfactible.

- ▶ Consecuencia: La skolemización preserva **insatisfactibilidad**.
- ▶ Esto es suficiente para poder aplicar el método de resolución, tal como veremos.

## ¿Preservación de validez?

- ▶ ¿Podemos eliminar los cuantificadores existenciales **sin** alterar la **validez**?
- ▶ Esto es mucho más fuerte que hablar de **satisfactibilidad**...
- ▶ Respuesta: **No**.
- ▶ Ejemplo:  $\exists x.(P(a) \supset P(x))$  es válida pero  $P(a) \supset P(b)$  no lo es.
- ▶ Tal como se mencionó, la skolemización sí preserva **satisfactibilidad** y ello es suficiente para el método de resolución.

# Skolemización

Cada ocurrencia de una subfórmula

$$\exists x.B$$

en  $A$  se reemplaza por

$$B\{x \leftarrow f(x_1, \dots, x_n)\}$$

donde

- ▶  $\{\bullet \leftarrow \bullet\}$  es la operación usual de sustitución (sustituir todas las ocurrencias libres de una variable en una expresión - fórmula o término - por otra expresión).
- ▶  $f$  es un símbolo de función nuevo y las  $x_1, \dots, x_n$  son las variables de las que depende  $x$  en  $B$ .
- ▶ Si  $\exists x.B$  forma parte de una fórmula mayor,  $x$  sólo depende de las **variables libres** de  $B$  (por ejemplo, en  $\forall z.\forall y.\exists x.P(y, x)$  la  $x$  depende de  $y$ ).

## Definición de forma normal de Skolem (1/2)

- ▶ Sea  $A$  una sentencia rectificada en NNF.
  - ▶ No es necesario que esté en forma prenexa.
- ▶ La **forma normal de Skolem de  $A$**  (escrito  $\mathbf{SK}(A)$ ) se define recursivamente como sigue.
- ▶ Sea  $A'$  cualquier subfórmula de  $A$ .
  - ▶ Si  $A'$  es una fórmula atómica o su negación,  $\mathbf{SK}(A') = A'$ .
  - ▶ Si  $A'$  es de la forma  $(B \star C)$  con  $\star \in \{\vee, \wedge\}$ , entonces  $\mathbf{SK}(A') = (\mathbf{SK}(B) \star \mathbf{SK}(C))$ .
  - ▶ Si  $A'$  es de la forma  $\forall x.B$ , entonces  $\mathbf{SK}(A') = \forall x.\mathbf{SK}(B)$ .
  - ▶ *Sigue en siguiente diapositiva.*



## Definición de forma normal de Skolem (2/2)

- ▶ Si  $A'$  es de la forma  $\exists x.B$  y  $\{x, y_1, \dots, y_m\}$  son las variables libres de  $B$ <sup>1</sup>, entonces
  1. Si  $m > 0$ , crear un nuevo **símbolo de función de Skolem**,  $f_x$  de aridad  $m$  y definir

$$\mathbf{SK}(A') = \mathbf{SK}(B\{x \leftarrow f_x(y_1, \dots, y_m)\})$$

2. Si  $m = 0$ , crear una nueva **constante de Skolem**  $c_x$  y

$$\mathbf{SK}(A') = \mathbf{SK}(B\{x \leftarrow c_x\})$$

**Nota:** dado que  $A$  está rectificadada, cada  $f_x$  y  $c_x$  es única.

---

<sup>1</sup>Notar que se ligan en  $A$  dado que  $A$  es sentencia

# Ejemplos

Considere la fórmula

$$\forall x. \left( P(a) \vee \exists y. (Q(y) \wedge \forall z. (P(y, z) \vee \exists u. Q(x, u))) \right) \vee \exists w. Q(w)$$

La forma normal de Skolem es:

$$\forall x. (P(a) \vee (Q(g(x)) \wedge \forall z. (P(g(x), z) \vee Q(x, f(x))))) \vee Q(c)$$

# Ejemplos

- Considere la sentencia:

$$\forall x. \exists y. \exists z. R(x, y, z)$$

1. Alternativa 1 (rojo, azul)

1.1  $\forall x. \exists y. \exists z. R(x, y, z)$

1.2  $\forall x. \exists z. R(x, f(x), z)$

1.3  $\forall x. R(x, f(x), g(x))$

2. Alternativa 2 (azul, rojo)

2.1  $\forall x. \exists y. \exists z. R(x, y, z)$

2.2  $\forall x. \exists y. R(x, y, h(x, y))$

2.3  $\forall x. R(x, k(x), h(x, k(x)))$

3. La skolemización no es determinística.

- Es mejor skolemizar de afuera hacia adentro.

# Forma clausal

Hasta ahora tenemos una fórmula que:

1. está escrita en términos de  $\wedge, \vee, \neg, \forall$ ;
2. si tiene negaciones, solamente se aplican a átomos (**forma normal negada**);
3. si tiene cuantificadores, son universales (**forma normal de Skolem**);
4. si está en **forma normal prenexa** y tiene cuantificadores, éstos se encuentran todos en el prefijo.

$$\forall x_1 \dots \forall x_n. B$$

# Forma clausal

$$\forall x_1 \dots \forall x_n. B$$

1. Pasar  $B$  a **forma normal conjuntiva**  $B'$  como si fuera una fórmula proposicional arrojando

$$\forall x_1 \dots \forall x_n. B'$$

2. Distribuir los cuantificadores sobre cada conjunción usando la fórmula válida  $\forall x. (A \wedge B) \iff \forall x. A \wedge \forall x. B$  arrojando una conjunción de **cláusulas**

$$\forall x_1 \dots \forall x_n. C_1 \wedge \dots \wedge \forall x_1 \dots \forall x_n. C_m$$

donde cada  $C_i$  es una disyunción de literales

3. Se simplifica escribiendo  $\{C_1, \dots, C_m\}$ .

## Ejemplo

$$\forall x. \forall z. (P(a) \vee (Q(g(x)) \wedge (P(g(x), z) \vee Q(x, f(x))))) \vee Q(c)$$

1. Pasamos la matriz a forma normal conjuntiva

$$\forall x. \forall z. ([P(a) \vee Q(g(x)) \vee Q(c)] \wedge [P(a) \vee P(g(x), z) \vee Q(x, f(x)) \vee Q(c)])$$

2. Distribuimos los cuantificadores

$$\forall x. \forall z. [P(a) \vee Q(g(x)) \vee Q(c)] \wedge \forall x. \forall z. [P(a) \vee P(g(x), z) \vee Q(x, f(x)) \vee Q(c)]$$

3. Pasamos a notación de conjuntos

$$\left\{ \{P(a), Q(g(x)), Q(c)\}, \right. \\ \left. \{P(a), P(g(x), z), Q(x, f(x)), Q(c)\} \right\}$$

# Forma clausal - Resumen

1. Escribir la fórmula en términos de  $\wedge, \vee, \neg, \forall, \exists$  (i.e. eliminar implicación)
2. Pasar a **forma normal negada**
3. Pasar a **forma normal prenexa**
4. Pasar a **forma normal de Skolem** (puede hacerse antes de 3)
5. Pasar matriz a **forma normal conjuntiva**
6. **Distribuir** cuantificadores universales

**Nota:** todos los pasos preservan **validez lógica**, salvo la skolemización (que preserva la **satisfactibilidad**).

# Resolución en lógica de primer orden

Consideremos la siguiente fórmula:

$$(\forall x.P(x)) \wedge \neg P(a)$$

- ▶ Es satisfactible? NO.
- ▶ Su forma clausal es

$$\{\{P(x)\}, \{\neg P(a)\}\}$$

- ▶ ¿Podemos aplicar la regla de resolución?

$$\frac{\{A_1, \dots, A_m, Q\} \quad \{B_1, \dots, B_n, \neg Q\}}{\{A_1, \dots, A_m, B_1, \dots, B_n\}}$$

- ▶ **No.**  $P(x)$  y  $P(a)$  no son idénticos, son ... **unificables**.



## Ahora sí, la Regla de resolución

$$\frac{\{B_1, \dots, B_k, A_1, \dots, A_m\} \quad \{\neg D_1, \dots, \neg D_j, C_1, \dots, C_n\}}{\sigma(\{A_1, \dots, A_m, C_1, \dots, C_n\})}$$

donde  $\sigma$  es el MGU de  $\{B_1, \dots, B_k, D_1, \dots, D_j\}$ .

- ▶ Asumimos que las cláusulas  $\{B_1, \dots, B_k, A_1, \dots, A_m\}$  y  $\{\neg D_1, \dots, \neg D_j, C_1, \dots, C_n\}$  no tienen variables en común; en caso contrario se renombran las variables.
- ▶ Observar que  $\sigma(B_1) = \dots = \sigma(B_k) = \sigma(D_1) = \dots = \sigma(D_j)$ .
- ▶ La cláusula  $\sigma(\{A_1, \dots, A_m, C_1, \dots, C_n\})$  se llama **resolvente** (de  $\{B_1, \dots, B_k, A_1, \dots, A_m\}$  y  $\{\neg D_1, \dots, \neg D_j, C_1, \dots, C_n\}$ ).

# Método de resolución

- ▶ Las siguientes nociones son análogas al caso proposicional.
  - ▶ Cláusula vacía
  - ▶ Paso de resolución
  - ▶ Refutación
- ▶ Al igual que en el caso proposicional contamos con el siguiente resultado.

## Teorema de Herbrand-Skolem-Gödel

Cada paso de resolución preserva **satisfactibilidad**.

## Ejemplo

Supongamos que dado  $A$ , obtenemos  $\neg A$ , lo convertimos a forma clausal y nos queda:  $C_1 \wedge C_2 \wedge C_3$  donde

- ▶  $C_1 = \forall z_1. \forall x. (\neg P(z_1, a) \vee \neg P(z_1, x) \vee \neg P(x, z_1))$
- ▶  $C_2 = \forall z_2. (P(z_2, f(z_2)) \vee P(z_2, a))$
- ▶  $C_3 = \forall z_3. (P(f(z_3), z_3) \vee P(z_3, a))$

Abreviado (sin cuantificadores + notación de conjuntos):

$$\left\{ \begin{array}{l} \{ \neg P(z_1, a), \neg P(z_1, x), \neg P(x, z_1) \}, \\ \{ P(z_2, f(z_2)), P(z_2, a) \}, \\ \{ P(f(z_3), z_3), P(z_3, a) \} \end{array} \right\}$$

## Ejemplo

$$C_1 = \{\neg P(z_1, a), \neg P(z_1, x), \neg P(x, z_1)\},$$

$$C_2 = \{P(z_2, f(z_2)), P(z_2, a)\},$$

$$C_3 = \{P(f(z_3), z_3), P(z_3, a)\}.$$

1. De  $C_1$  y  $C_2$  con  $\{z_1 \leftarrow a, x \leftarrow a, z_2 \leftarrow a\}$ :

$$C_4 = \{P(a, f(a))\}$$

2. De  $C_1$  y  $C_3$  con  $\{z_1 \leftarrow a, x \leftarrow a, z_3 \leftarrow a\}$ :

$$C_5 = \{P(f(a), a)\}$$

3. De  $C_1$  y  $C_5$  con  $\{z_1 \leftarrow f(a), x \leftarrow a\}$ :

$$C_6 = \{\neg P(a, f(a))\}$$

4. De  $C_4$  y  $C_6$ :  $\square$

# Diferencias con proposicional

## 1. En proposicional

$$\frac{\{Q, A_1, \dots, A_m\} \quad \{\neg Q, B_1, \dots, B_n\}}{\{A_1, \dots, A_m, B_1, \dots, B_n\}}$$

## 2. En primer orden

$$\frac{\{B_1, \dots, B_k, A_1, \dots, A_m\} \quad \{\neg D_1, \dots, \neg D_j, C_1, \dots, C_n\}}{\sigma(\{A_1, \dots, A_m, C_1, \dots, C_n\})}$$

donde  $\sigma$  es el MGU de  $\{B_1, \dots, B_k, D_1, \dots, D_j\}$ .

## Regla de resolución binaria

$$\frac{\{B, A_1, \dots, A_m\} \quad \{\neg D, C_1, \dots, C_n\}}{\sigma(\{A_1, \dots, A_m, C_1, \dots, C_n\})}$$

donde  $\sigma$  es el MGU de  $\{B, D\}$ .

- ▶ Es **incompleta**.
- ▶ Ejemplo: intentar refutar  $\{\{P(x), P(y)\}, \{\neg P(v), \neg P(w)\}\}$

## Regla de resolución binaria

- Se puede recuperar la completitud incorporando una regla adicional: **factorización**.

$$\frac{\{B_1, \dots, B_k, A_1, \dots, A_m\}}{\sigma(\{B_1, A_1, \dots, A_m\})} \quad \square$$

donde  $\sigma$  es el MGU de  $\{B_1, \dots, B_k\}$ .

- En el ejemplo anterior
  - $\{\{P(x), P(y)\}, \{\neg P(v), \neg P(w)\}\}$
  - $\{\{P(x), P(y)\}, \{\neg P(v), \neg P(w)\}, \{P(z)\}\}$  (fact)
  - $\{\{P(x), P(y)\}, \{\neg P(v), \neg P(w)\}, \{P(z)\}, \{\neg P(u)\}\}$  (fact)
  - $\{\{P(x), P(y)\}, \{\neg P(v), \neg P(w)\}, \{P(z)\}, \{\neg P(u)\}, \square\}$  (r. b.)