Fundamentos de Programación Lógica

Paradigmas de Lenguajes de Programación

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/`)
- 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
- Para motivar y comprender este mecanismo primero lo vamos a estudiar en el ámbito de la lógica proposicional

Prolog

Ejemplo de programa

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

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

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

Ejemplo de goal

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

Sintaxis de la lógica proposicional

Dado un conjunto $\mathcal{V}$ de variables proposicionales, podemos definir inductivamente al conjunto de fórmulas proposicionales (o proposiciones) **Prop** de la siguiente manera:

Sintaxis de la lógica proposicional

Dado un conjunto $\mathcal{V}$ de variables proposicionales, podemos definir inductivamente al conjunto de fórmulas proposicionales (o proposiciones) **Prop** de la siguiente manera:

1. Una variable proposicional $P_0, P_1, \ldots$ es una proposición
2. Si A, B son proposiciones, entonces:
   - $\neg A$ es una proposición
   - $A \wedge B$ es una proposición
   - $A \vee B$ es una proposición
   - $A \supset B$ es una proposición
   - $A \iff B$ es una proposición

Ejemplos: $A \vee \neg B$, $(A \wedge B) \supset (A \vee A)$

Semántica

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

$$
\begin{array}{rcl}
v \models P & \textit{sii} & v(P) = \mathbf{T} \\
v \models \neg A & \textit{sii} & v \not\models A \ (i.e. \text{ no } v \models A) \\
v \models A \vee B & \textit{sii} & v \models A \text{ o } v \models B \\
v \models A \wedge B & \textit{sii} & v \models A \text{ y } v \models B \\
v \models A \supset B & \textit{sii} & v \not\models A \text{ o } v \models B \\
v \models A \iff B & \textit{sii} & (v \models A \text{ sii } v \models B)
\end{array}
$$

Tautologías y satisfactibilidad

Una proposición A es

- una tautología

Tautologías y satisfactibilidad

Una proposición A es

- una tautologíasi $v \models A$ para toda valuación v
- satisfactible

Tautologías y satisfactibilidad

Una proposición A es

- una tautologíasi $v \models A$ para toda valuación v
- satisfactible si existe una valuación v tal que $v \models A$
- insatisfactible

Tautologías y satisfactibilidad

Una proposición A es

- una tautologíasi $v \models A$ para toda valuación v
- satisfactible si existe una valuación v tal que $v \models A$
- 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

Ejemplos

Tautologías

- $A \supset A$
- $\neg\neg A \supset A$
- $(A \supset B) \iff (\neg B \supset \neg A)$

Proposiciones insatisfactibles

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

Tautologías e insatisfactibilidad

Teorema

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

Tautologías e insatisfactibilidad

Teorema

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

Dem.

$\Rightarrow$. Si A es tautología, para toda valuación v, $v \models A$. Entonces, $v \not\models \neg A$ (i.e. v no satisface $\neg A$).

$\Leftarrow$. Si $\neg A$ es insatisfactible, para toda valuación v, $v \not\models \neg A$. Luego $v \models A$.

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 válido mostrando que $\neg A$ es insatisfactible

Validez por refutación

Principio de demostración por refutación:

Probar que A es válido 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.

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

  $$C_1 \wedge \ldots \wedge C_n$$

  donde cada C_i (llamado cláusula) es una disyunción

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

  y cada B_{ij} es un literal
- Una FNC es una "conjunción de disyunciones de literales"

Forma normal conjuntiva

Ejemplos

- $(P \vee Q) \wedge (P \vee \neg Q)$

Forma normal conjuntiva

Ejemplos

- $(P \lor Q) \land (P \lor \neg Q)$está en FNC
- $(P \lor Q) \land (P \lor \neg\neg Q)$

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
- $(P \wedge Q) \vee P$

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

$$\{C_1, \ldots, C_n\}$$

donde cada C_i es un conjunto de literales

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

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

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

Principio fundamental del método de resolución

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

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

Principio fundamental del método de resolución

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

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

- Por lo tanto, el conjunto de cláusulas

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

es lógicamente equivalente a

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

Resolución

- En consecuencia, el conjunto de cláusulas

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

  es insatisfactible sii

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

  es insatisfactible

Resolución

- En consecuencia, el conjunto de cláusulas

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

  es insatisfactible sii

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

  es insatisfactible
- La cláusula $\{A, B\}$ se llama resolvente de las cláusulas $\{A, P\}$ y $\{B, \neg P\}$

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

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

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

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

Ejemplos

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

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

Ejemplo

El resultado de aplicar la regla de resolución a
$\{\{P, Q\}, \{P, \neg Q\}, \{\neg P, Q\}, \{\neg P, \neg Q\}\}$
es
$\{\{P, Q\}, \{P, \neg Q\}, \{\neg P, Q\}, \{\neg P, \neg Q\}, \{P\}\}$

El método de resolución

El proceso de agregar a un conjunto S el 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.

El método de resolución

El proceso de agregar a un conjunto S el 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

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

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.

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

- 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.

Ejemplo

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

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

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

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

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. $\{\{P, Q\}, \{P, \neg Q\}, \{\neg P, Q\}, \{\neg P, \neg Q\}, \{P\}, \{\neg P\}\}$
4. $\{\{P, Q\}, \{P, \neg Q\}, \{\neg P, Q\}, \{\neg P, \neg Q\}, \{P\}, \{\neg P\}, \square\}$

Ejemplo

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

Ejemplo

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

1. $\{\{A, B, \neg C\}, \{A, B, C\}, \{A, \neg B\}, \{\neg A\}\}$

Ejemplo

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

1. $\{\{A, B, \neg C\}, \{A, B, C\}, \{A, \neg B\}, \{\neg A\}\}$
2. $\{\{A, B, \neg C\}, \{A, B, C\}, \{A, \neg B\}, \{\neg A\}, \{A, B\}\}$

Ejemplo

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

1. $\{\{A, B, \neg C\}, \{A, B, C\}, \{A, \neg B\}, \{\neg A\}\}$
2. $\{\{A, B, \neg C\}, \{A, B, C\}, \{A, \neg B\}, \{\neg A\}, \{A, B\}\}$
3. $\{\{A, B, \neg C\}, \{A, B, C\}, \{A, \neg B\}, \{\neg A\}, \{A, B\}, \{A\}\}$

Ejemplo

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

1. $\{\{A, B, \neg C\}, \{A, B, C\}, \{A, \neg B\}, \{\neg A\}\}$
2. $\{\{A, B, \neg C\}, \{A, B, C\}, \{A, \neg B\}, \{\neg A\}, \{A, B\}\}$
3. $\{\{A, B, \neg C\}, \{A, B, C\}, \{A, \neg B\}, \{\neg A\}, \{A, B\}, \{A\}\}$
4. $\{\{A, B, \neg C\}, \{A, B, C\}, \{A, \neg B\}, \{\neg A\}, \{A, B\}, \{A\}, \square\}$

Ejemplo

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

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.

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,

$$S \text{ es insatisfactible sii tiene una refutación}$$

Recapitulando

Para probar que A es una tautología hacemos lo siguiente:

1. Calculamos la forma normal conjuntiva de $\neg A$
2. Aplicamos el método de resolución
3. Si hallamos una refutación:
   - $\neg A$ es insatisfactible,
   - Y, por lo tanto, A es una tautología
4. Si no hallamos ninguna refutación:
   - $\neg A$ es satisfactible,
   - Y, por lo tanto, A no es una tautología

Lógica de primer orden (Repaso)

Lenguaje de primer orden

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

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

Lenguaje de primer orden

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

1. Un conjunto numerable de constantes $c_0, c_1, \ldots$
2. Un conjunto numerable de símbolos de función con aridad $n > 0$ (indica el número de argumentos) $f_0, f_1, \ldots$
3. Un conjunto numerable de símbolos de predicado con aridad $n \geq 0$, $P_0, P_1, \ldots$. 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 $\mathcal{V} = \{x_0, x_1, \ldots\}$ 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 $t_1, \ldots, t_n \in \mathcal{L}$-términos y f es un símbolo de función de aridad n, entonces $f(t_1, \ldots, t_n) \in \mathcal{L}$-términos

Términos de primer orden

Sea $\mathcal{V} = \{x_0, x_1, \ldots\}$ 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 $t_1, \ldots, t_n \in \mathcal{L}$-términos y f es un símbolo de función de aridad n, entonces $f(t_1, \ldots, t_n) \in \mathcal{L}$-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 $t_1, \ldots, t_n \in \mathcal{L}$-términos y P es un símbolo de predicado de aridad n, entonces $P(t_1, \ldots, t_n)$ 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 $(A \wedge B), (A \vee B), (A \supset B), (A \iff B)$ y $\neg A$ son $\mathcal{L}$-fórmulas

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 $(A \wedge B), (A \vee B), (A \supset B), (A \iff B)$ y $\neg A$ 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

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 $(A \wedge B), (A \vee B), (A \supset B), (A \iff B)$ y $\neg A$ 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

Sentencias

Una sentencia es una fórmula cerrada (i.e. sin variables libres).

- Muchos resultados se formulan para sentencias
- Esto no implica una pérdida de generalidad ya que toda fórmula es lógicamente equivalente a su clausura universal

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 $n > 0$, $I(f) : M^n \to M$
  3. Para todo predicado P de aridad $n \geq 0$, $I(P) : M^n \to \{\mathbf{T}, \mathbf{F}\}$

Satisfactibilidad

Asignación

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

- 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

Asignación

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

- 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 **M**

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

Satisfactibilidad

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

$$
\begin{array}{rcl}
s \models_{\mathbf{M}} P(t_1, \ldots, t_n) & \textit{sii} & P_{\mathbf{M}}(s(t_1), \ldots, s(t_n)) \\
s \models_{\mathbf{M}} \neg A & \textit{sii} & s \not\models_{\mathbf{M}} A \\
s \models_{\mathbf{M}} (A \wedge B) & \textit{sii} & s \models_{\mathbf{M}} A \text{ y } s \models_{\mathbf{M}} B \\
s \models_{\mathbf{M}} (A \vee B) & \textit{sii} & s \models_{\mathbf{M}} A \text{ o } s \models_{\mathbf{M}} B \\
s \models_{\mathbf{M}} (A \supset B) & \textit{sii} & s \not\models_{\mathbf{M}} A \text{ o } s \models_{\mathbf{M}} B \\
s \models_{\mathbf{M}} (A \iff B) & \textit{sii} & (s \models_{\mathbf{M}} A \text{ sii } s \models_{\mathbf{M}} B) \\
s \models_{\mathbf{M}} \forall x_i.A & \textit{sii} & s[x_i \leftarrow a] \models_{\mathbf{M}} A \text{ para todo } a \in M \\
s \models_{\mathbf{M}} \exists x_i.A & \textit{sii} & s[x_i \leftarrow a] \models_{\mathbf{M}} A \text{ para algún } a \in M
\end{array}
$$

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

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
- 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.

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.
- El pasaje a forma clausal 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 (FNN) se define inductivamente como:

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

Forma normal negada

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

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

Ejemplos

- $\neg\exists x.((P(x) \vee \exists y.R(x,y)) \supset (\exists z.R(x,z) \vee P(a)))$

Forma normal negada

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

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

Ejemplos

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

Forma normal negada

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

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

Ejemplos

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

Forma normal negada

Toda fórmula es lógicamente equivalente a otra en FNN.

Forma normal negada

Toda fórmula es lógicamente equivalente a otra en FNN.

Dem.
Por inducción estructural usando:

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

Forma normal negada

Toda fórmula es lógicamente equivalente a otra en FNN.

Dem.
Por inducción estructural usando:

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

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

Forma normal negada

Toda fórmula es lógicamente equivalente a otra en FNN.

Dem.

Por inducción estructural usando:

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

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 $Q_1 x_1 \ldots Q_n x_n . B$, $n \geq 0$, donde

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

Forma prenexa

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

Forma prenexa

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

Demostración

Por inducción estructural usando:

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

- Nota: Con estas equivalencias basta, si asumimos que A está en FNN.

Ejemplo

1. $\forall x.\neg P(x) \wedge (\exists y.Q(y) \vee \forall z.P(z))$

Ejemplo

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

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

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

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

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

Eliminación de cuantificadores existenciales

- ¿Cómo eliminamos los $\exists$ sin cambiar la satisfactibilidad?

Eliminación de cuantificadores existenciales

- ¿Cómo eliminamos los $\exists$ sin cambiar la satisfactibilidad?
- Introducimos "testigos" para ellos.
  - Todo cuantificador existencial se instancia en una constante o función de skolem.

Eliminación de cuantificadores existenciales

- ¿Cómo eliminamos los $\exists$ sin cambiar la satisfactibilidad?
- Introducimos "testigos" para ellos.
  - Todo cuantificador existencial se instancia en 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?

- ¿Podremos eliminar los cuantificadores existenciales, usando skolemización, sin alterar la validez?

¿Preservación de validez?

- ¿Podremos eliminar los cuantificadores existenciales, usando skolemización, sin alterar la validez?
- Esto es mucho más fuerte que preservar 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

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

donde

- $\bullet\{\bullet \leftarrow \bullet\}$ 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 $x_1, \ldots, x_n$ son las variables de las que depende x en B.
- Si $\exists x.B$ forma parte de una fórmula mayor, x solo 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 FNN.
  - No es necesario que esté en forma prenexa.
- Una forma normal de Skolem de A (escrito $\mathbf{SK}(A)$) es una fórmula sin existenciales que se obtiene 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 $\{x, y_1, \ldots, y_m\}$ son las variables libres de B[1], entonces
  1. Si $m > 0$, crear un nuevo símbolo de función de Skolem, f_x de aridad m y definir
  $$\mathbf{SK}(A')=\mathbf{SK}(B\{x \leftarrow f_x(y_1, \ldots, y_m)\})$$
  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 rectificada, cada f_x y c_x es única.

[1] Notar que se ligan en A dado que A es sentencia

Ejemplos

Considerar 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)

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

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

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)

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

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.

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.

$$\forall x_1 \ldots \forall x_n . B$$

Forma clausal

$$\forall x_1 \ldots \forall x_n . B$$

1. Pasar B (la matriz) a forma normal conjuntiva B' como si fuera una fórmula proposicional arrojando

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

Forma clausal

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

1. Pasar B (la matriz) 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

Forma clausal

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

1. Pasar B (la matriz) 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

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

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

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

2. Distribuimos los cuantificadores

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

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

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

2. Distribuimos los cuantificadores

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

3. Pasamos a notación de conjuntos

$$\Big\{\{P(a), Q(g(x)), Q(c)\}, \{P(a), P(g(x), z), Q(x, f(x)), Q(c)\}\Big\}$$

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 la matriz a forma normal conjuntiva
6. Distribuir cuantificadores universales

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