Unificación

$C_1 = \neg P(a, y) \vee Q(x, y)$ $C_2 = P(x, y) \vee R(x, b)$

$\neg P(a, y)$ y $P(x, y)$ no son literales complementarios

Si sustituimos x/a se obtienen

$C_1' = \neg P(a, y) \vee Q(a, y)$ $C_2' = P(a, y) \vee R(a, b)$

Instancias de C_1 y C_2

$Res(C_1', C_2') = Q(a, y) \vee R(a, b)$

Unificación

Sustitución:

Conjunto finito de reemplazos simultáneos de variables por términos.

$e = \{ x_1/t_1, x_2/t_2, ..., x_n/t_n \}$ x_i variables distintas t_i términos

$A = A(x_1, x_2, ..., x_n)$ término, literal, cláusula o conj. de cláusulas

$Ae = A(x_1, x_2, ..., x_n)e = A(x_1/t_1, x_2/t_2, ..., x_n/t_n)$

Ejemplo:

$A(x, y) = P(x, y) \vee R(f(x), a)$ $e_1 = \{x/g(b), y/a)\}$ $e_2 = \{x/f(y), y/x)\}$

$A(x, y)\, e_1 = P(g(b), a) \vee R(f(g(b)), a)$

$A(x, y)\, e_2 = P(f(y), x) \vee R(f(f(y)), a)$

Ciencias de la Computación II - Filminas de Clase – Mg. Virginia Mauco – Facultad Cs. Exactas – UNCPBA - 2009

Unificación

<u>Composición de sustituciones</u>: Dadas e_1 y e_2 sustituciones

$e_1 = \{ x_1/t_1, x_2/t_2, ..., x_n/t_n\}$ $e_2 = \{ y_1/s_1, y_2/s_2, ..., y_k/s_k\}$

$e_1.e_2 = \{ x_i/t_ie_2 : x_i \neq t_ie_2, i = 1, ..., n\} \cup \{ y_j/s_j : y_j \neq x_i , i = 1, ..., n, j = 1, ..., k\}$

<u>Ejemplo</u>:

$e_1 = \{ x/g(y, a), y/b, z/f(w), u/w\}$ $e_2 = \{ y/f(b), w/u, t/g(a, f(b))\}$

$e_1.e_2 = \{ x/g(f(b), a), y/b, z/f(u), w/u, t/g(a, f(b))\}$

Unificación

<u>Unificador</u>:

$A = \{l_1, l_2, ..., l_n\}$ conjunto de literales con el mismo símbolo de relación

Una sustitución e es un unificador de A si $l_1e = l_2 e = ... = l_n e$

Si existe un unificador de A $\Longrightarrow$ A es unificable

Para considerar si dos literales son unificables o no, deben tener el mismo símbolo de relación y deben ser ambos positivos o negativos.

<u>Unificador más general (umg)</u>:

Un unificador u es el umg de A sí y sólo sí para todo otro unificador e de A, existe una sustitución λ tal que $e = u. \lambda$

Ciencias de la Computación II - Filminas de Clase – Mg. Virginia Mauco – Facultad Cs. Exactas – UNCPBA - 2009

Algoritmo de Unificación de Robinson

✓ **Método para encontrar umg para un conjunto de fórmulas atómicas**

Pares no coincidentes

Sean A y B dos literales considerados como una sucesión de símbolos.

Sea k la posición en ambas secuencias, comenzando desde la izquierda, en donde las secuencias difieren en un par de términos {t, t'} donde t $\in$ A y t' $\in$ B.

{t, t'} se denomina k-par no coincidente

Ejemplo:

P(g(a), z, a) y P(y, b, a)

{g(a), y} es el 1-par no coincidente

{z, b} es el 2-par no coincidente

Algoritmo de Unificación de Robinson

Algoritmo:

- **Sean A y B dos literales**
- **Sea A_0 = A y B_0 = B**
- **Supongamos que se han definido A_i y B_i**

  Sea {t, t'} el primer k-par no coincidente de A_i y B_i

  - Si uno de los términos es una variable x_i y el otro es un término t_i tal que $x_i \notin Var(t_i)$ entonces se define la sustitución

  $$e_{i+1} = \{ x_i / t_i \}$$

  Se calculan $A_{i+1} = A_i e_{i+1}$ y $B_{i+1} = B_i e_{i+1}$

  - Si esto no es posible A y B no son unificables.

- **Si después de algún paso $A_n = B_n$, A y B son unificables y el umg es**

$$u = e_i e_{i+1} \ldots e_n$$

Algoritmo de Unificación de Robinson

Ejemplos:

1) $A = Q(z, y, f(x), g(t))$ $B = Q(g(y), a, f(g(w)), x)$ a cte.

$u = \{ z/g(a),\ y/a,\ x/g(t),\ w/t \}$ u unificador más general

$u_1 = u . \lambda_1 = \{ z/g(a),\ y/a,\ x/g(b),\ w/b, t/b \}$ $\lambda_1 = \{ t/b \}$

$u_2 = u . \lambda_2 = \{ z/g(a),\ y/a,\ x/g(f(a)),\ w/f(a), t/f(a) \}$ $\lambda_2 = \{ t/f(a) \}$

u_1 y u_2 son unificadores de A y B A y B son unificables

2) $A = Q(x, f(a), b, y)$ $B = Q(g(x), y, b, z)$

A y B no son unificables

Resolución con Unificación

Sean C_1 y C_2 cláusulas

1) Supongamos que existen sustituciones e_1: $Var(C_1) \rightarrow Var$ y e_2: $Var(C_2) \rightarrow Var$ tal que las cláusulas C_1e_1 y C_2e_2 no tienen variables en común.

2) Supongamos que los conjuntos de literales $L_1e_1 = \{l_1, ..., l_n\} \subseteq C_1e_1$ y $L_2e_2 = \{l_1', ..., l_m'\} \subseteq C_2e_2$ son tales que el conjunto

$L_1e_1 \cup \neg L_2e_2 = \{l_1, ..., l_n, l_1', ..., l_m'\}$ es unificable por el umg u

$l_1u = ...= l_nu = l_1' u = ... l_m'u = l$

Entonces la resolvente de C_1 y C_2 es la cláusula

$Res(C_1, C_2) = (C_1e_1u – \{ l \}) \cup (C_2e_2u – \{ l^c \})$

donde $L_1e_1u = \{ l \}$ y $L_2e_2u = \{l^c \}$

Resolución con Unificación

Ejemplo:

Determinar la resolvente de

$C_1 = P(x) \vee \neg R(y) \vee P(z)$ $\quad$ $C_2 = Q(b, f(x)) \vee \neg P(g(z))$ $\quad$ b cte.

1) Cambiar variables $\quad e_2 = \{x/w, z/t\}$ (dejamos fijas las de C_1)

$C_2e_2 = Q(b, f(w)) \vee \neg P(g(t))$

2) Determinar el umg para $\{P(x), P(z), P(g(t))\}$

El umg es $u = \{z/g(t), x/g(t)\}$

$C_1u = P(g(t)) \vee \neg R(y)$ $\quad$ $C_2e_2u = Q(b, f(w)) \vee \neg P(g(t))$

$Res(C_1, C_2) = \neg R(y) \vee Q(b, f(w))$

Resolución

Deducción por Resolución:

Sea S un conjunto de cláusulas y C una cláusula.

Una deducción por resolución de C a partir de S, $S \vdash_R C$, es una sucesión finita de cláusulas $C_1, C_2, \ldots, C_n = C$ tales que

1) $C_i \in S$ ó

2) Existen cláusulas C_j, C_k con $j, k < i < n$ tal que $Res(C_j, C_k) = C_i$

Refutación de un conjunto de cláusulas:

Sea S un conjunto de cláusulas. Una resolución de $\perp$ de un conjunto de cláusulas S se dice una refutación de S.

Teorema:

Sea S un conjunto de cláusulas. Entonces $S \vdash_R \perp$ sí y sólo sí S es insatisfacible.

Resolución

✓ **El método de resolución es completo y correcto.**

<u>Teorema de Completitud</u>:

Sea $S \cup \{C\}$ un conjunto de cláusulas.

Si $S \models C$ entonces $S \cup \{\neg C\} \vdash_R \perp$.

<u>Teorema de Corrección</u>:

Sea $S \cup \{C\}$ un conjunto de cláusulas.

Si $S \vdash_R C$ entonces $S \models C$.

Resolución

Sea S un conjunto de cláusulas. Las siguientes condiciones son equivalentes:

1) **S es insatisfacible.**
2) **S no tiene modelos.**
3) **S no tiene Modelos de Herbrand.**
4) **U(S)(S) es p-insatisfacible.**
5) **Existe un subconjunto finito $S_0 \subseteq U(S)(S)$ que es p-insatisfacible**
6) **Existe un subconjunto finito $S_0 \subseteq U(S)(S)$ tal que $S_0 \vdash_R \perp$**
7) **$S \vdash_R \perp$**

Resolución

Ejemplo:

Determinar si el siguiente conjunto de cláusulas es satisfacible o no.

$S = \{F(a), \neg E(y) \vee A(a, y), \neg F(x) \vee \neg C(y) \vee \neg A(x, y), E(b), C(b)\}$ **a, b ctes**

Se puede probar que $S \vdash_R \perp$ **por lo tanto S es insatisfacible.**

También se puede demostrar S es p-insatisfacible ya que existe un subconjunto finito $S_0 \subseteq U(S)(S)$ **que es insatisfacible.**

Resolución

Método de resolución:

- **Una sola regla de deducción**
- **En general, muchas maneras de seleccionar dos cláusulas para producir una resolvente.**

Refinamientos o modificaciones al método para limitar la búsqueda

ESTRATEGIAS

SIMPLIFICACION

CONTROL
- **Resolución Lineal**
- **Resolución Unitaria**

Refinamientos de Resolución

ESTRATEGIAS DE SIMPLIFICACION

Si el conjunto inicial es insatisfacible (satisfacible) el simplificado también lo es y viceversa.

1) Eliminación de cláusulas tautológicas.

2) Simplificación de cláusulas por eliminación de literales repetidos.

3) Eliminación de cláusulas con literales puros. (Un literal es puro en un conjunto de cláusulas S sí y sólo sí no existe su complementario en S).

4) Eliminación de cláusulas que incorporan a otras (Una cláusula C_1 incorpora a otra cláusula C_2 si existe la sustitución u tal que $C_2u \subseteq C_1$).

Refinamientos de Resolución

RESOLUCION LINEAL

Sea S un conjunto de cláusulas y C un elemento de ese conjunto.

Una deducción por resolución lineal de C_n a partir de S con cláusula inicial C , $S \vdash_R C_n$, es una sucesión finita de cláusulas $C_0 = C, C_1, \ldots, C_n$ tales que para cada i = 0, 1, ..., n - 1

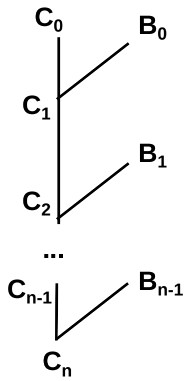
1) C_{i+1} es un resolvente de C_i (llamada cláusula central) y de otra cláusula B_i (llamada cláusula lateral)

2) Cada B_i o pertenece a S o es un antepasado de C_i (es decir, $B_i = Res(C_j, C_k)$ para j, k < i)

Si $C_n = \perp$ diremos que existe una refutación lineal de S.

Refinamientos de Resolución

Gráficamente



-C_0 es una cláusula de S

- B_i son cláusulas de S o alguna C_j con $j < i$

- C_{i+1}, $i < n$, es $Res(C_i , B_i)$

- Resolución lineal es un método correcto.

- Resolución lineal es un método completo (si un conjunto de cláusulas S es insatisfacible entonces de S se puede deducir la cláusula vacía aplicando resolución lineal).

Refinamientos de Resolución

RESOLUCION UNITARIA

Definición

Es un refinamiento del método de resolución en el cual en cada paso de resolución interviene al menos una cláusula unitaria.

Resolución unitaria es un método correcto pero en general no es completo.

Ejemplo:

$S = \{P(x) \lor Q(x), \neg P(x) \lor Q(x), P(x) \lor \neg Q(x), \neg P(x) \lor \neg Q(x)\}$

S es insatisfacible pero la cláusula vacía no puede deducirse aplicando resolución unitaria

Formalización de Lenguaje Natural

Formalizar un razonamiento en lenguaje natural

⇩

Expresar en lógica de predicados (LP) el conjunto de hipótesis y conclusión

Algunas estrategias:

- ✓ **Si la frase a formalizar no tiene una estructura sintáctica reconocida, intentar reescribirla manteniendo el significado.**
- ✓ **Definir dominio al cual pertenecen los elementos a usar.**
- ✓ **Determinar constantes, variables, funciones, predicados**
- ✓ **Identificar conectivas linguísticas y cuantificadores para sustituirlos por conectivos y cuantificadores de la LP.**

Formalización de Lenguaje Natural

Algunas estrategias:

- ✓ **Determinar constantes, variables, funciones, predicados:**
  - **Constantes: elementos concretos del dominio**
  - **Variables: elementos genéricos del dominio**
  - **Funciones de aridad n > 0: representan cómo un elemento queda determinado por otro.**
  - **Predicados de aridad n > 0: representan relaciones entre elementos.**
- ✓ **Determinar cuantificadores (para representar cuántos individuos cumplen cierta información – todas las variables deben estar cuantificadas)**
- ✓ **Determinar conectivos (para representar cómo se combina la información)**

Formalización de Lenguaje Natural

Patrones más habituales:

- **Universal afirmativo** $\forall x(A(x) \rightarrow B(x))$

  Todo A es B - Sólo los B son A – No hay ningún A que no sea B

- **Universal negativo** $\forall x(A(x) \rightarrow \neg B(x))$

  Ningún A es B

- **Existencial afirmativo** $\exists x(A(x) \wedge B(x))$

  Algún A es B – Alguien es a la vez A y B

- **Existencial negativo** $\exists x(A(x) \wedge \neg B(x))$

  Algún A no es B – No todos los A son B

Formalización de Lenguaje Natural

Algunas estrategias:

✓ **La formalización depende del dominio elegido**

"Todos los alumnos van a la biblioteca"

a) Dominio D_1 = conjunto de los alumnos

Predicado necesario $va\_biblio(x) = \{x \in D_1 \ / \ x \text{ va a la biblioteca} \}$

$\forall x \ va\_biblio(x)$

b) Dominio D_2 = conjunto de las personas

Predicados necesarios $alumno(x) = \{x \in D_2 \ / \ x \text{ es alumno} \}$

$va\_biblio(x) = \{x \in D_2 \ / \ x \text{ va a la biblioteca} \}$

$\forall x \ (alumno(x) \rightarrow va\_biblio(x))$

c) Dominio D_3 = conjunto de las personas unión conjunto de lugares

Predicados necesarios $alumno(x) = \{x \in D_3 \ / \ x \text{ es alumno} \}$

$va(x, y) = \{(x, y) \in D_3^2 \ / \ x \text{ va al lugar } y\}$

$\forall x \ (alumno(x) \rightarrow va(x, b))$ **b = biblioteca**

Ciencias de la Computación II - Filminas de Clase – Mg. Virginia Mauco – Facultad Cs. Exactas – UNCPBA - 2009

Formalización de Lenguaje Natural

Algunas estrategias:

✓ **Toda función se puede representar con un predicado con un argumento más que la función (pero las funciones simplifican la estructura de la fórmula)**

"Todo padre quiere mucho a sus hijos"

a) Formalización con predicados D = conjunto de personas

padre(x, y) = { (x, y) ∈ D2 / x es el padre de y }

quiere(x, y) = { (x, y) ∈ D2 / x quiere mucho a y }

$\forall x \forall y\ (padre(x, y) \rightarrow quiere(x, y))$

b) Formalización con funciones D = conjunto de personas

p(x) = el padre de x

quiere(x, y) = { (x, y) ∈ D2 / x quiere mucho a y }

$\forall x\ quiere(padre(x), x)$

Formalización de Lenguaje Natural

Algunas estrategias:

✓ **¿Cómo determinar si un predicado es unario, binario, etc.?**

1) En general, predicados unarios para verbo "ser" y usos intransitivos de verbos.

Juan es estudiante **estudiante(x) = {x ∈ D: x es estudiante}**

estudiante(Juan)

Juan pasea **pasea(x) = {x ∈ D: x pasea}**

pasea(Juan)

Formalización de Lenguaje Natural

✓ **¿Cómo determinar si un predicado es unario, binario, etc.?**

2) En general, predicados binarios para usos transitivos de verbos.

Juan respeta a los estudiantes **respeta(x, y) = {(x, y) $\in$ D^2: x respeta a y}**

Vx(estudiante(x) $\rightarrow$ respeta(Juan, x))

También algunos usos intransitivos de verbos

Juan pasea con Luis **pasea(x, y) = {(x, y) $\in$ D^2: x pasea con y}**

pasea(Juan, Luis)

Además algunos usos del verbo ser que establecen relaciones entre dos elementos

Juan es mayor que Luis **mayor(x, y) = {(x, y) $\in$ D^2: x es mayor que y}**

mayor(Juan, Luis)

Formalización de Lenguaje Natural

✓ **¿Cuánto se debe traducir de cada oración?**

- Todo lo que sea posible

Juan respeta a los estudiantes de vez en cuando

Juan respeta a los estudiantes cuando está de buen humor

- **Dos posibles alternativas:**

1) respeta(x, y)

2) respeta_vez_en_cuando(x, y) **respeta_buen_humor(x, y)**

Formalización de Lenguaje Natural

✓ **Siempre es posible traducir como unario cualquier predicado binario.**

Juan respeta a los estudiantes

respeta_estud(x) = {x $\in$ D: x respeta a los estudiantes }

Juan respeta a los profesores

respeta_prof(x) = {x $\in$ D: x respeta a los profesores }

- Problema de esta estrategia: hace desaparecer elementos clave para demostrar que ciertos argumentos son válidos.

Formalización de Lenguaje Natural

Ejemplo:

Juan respeta a los estudiantes. Todos los músicos son estudiantes. Entonces Juan respeta a los músicos.

-***Formalización binaria***

$\forall$x(estudiante(x) $\rightarrow$ respeta(Juan, x)), $\forall$y(musico(y) $\rightarrow$ estudiante(y)) $\models$

$\forall$z(musico(z) $\rightarrow$ respeta(Juan, z))

Es posible demostrar que este razonamiento es válido

-***Formalización unaria***

respeta_estud(Juan), $\forall$y(musico(y) $\rightarrow$ estudiante(y)) $\models$

respeta_musico(Juan)

No hay manera de demostrar que la conclusión se deduce de las premisas.

Formalización de Lenguaje Natural

Ejemplos

- Los niños quieren a las mascotas

$\forall x\ \forall y(\text{niño}(x) \wedge \text{mascota}(y) \rightarrow \text{quiere}(x, y)) \equiv$

$\forall x\ \forall y(\text{niño}(x) \rightarrow (\text{mascota}(y) \rightarrow \text{quiere}(x, y))) \equiv$

$\forall x\ (\text{niño}(x) \rightarrow \forall y(\text{mascota}(y) \rightarrow \text{quiere}(x, y))) \equiv$

- Algunos niños quieren a los mascotas

$\exists x\ \forall y(\text{niño}(x) \wedge (\text{mascota}(y) \rightarrow \text{quiere}(x, y))) \equiv$

$\exists x\ (\text{niño}(x) \wedge \forall y(\text{mascota}(y) \rightarrow \text{quiere}(x, y)))$

- Algunos niños quieren a algunas mascotas

$\exists x\ \exists y(\text{niño}(x) \wedge \text{mascota}(y) \wedge \text{quiere}(x, y))$

- Los niños quieren a algunas mascotas

$\forall x\ \exists y(\text{niño}(x) \rightarrow \text{mascota}(y) \wedge \text{quiere}(x, y)))$