Clase práctica
Resolución en lógica de primer orden

Paradigmas de Lenguajes de Programación

7 de Marzo del 2018

Agenda

¿Qué es?

- Procedimiento para determinar la insatisfactibilidad de una fórmula.
- Es útil como técnica de demostración por refutación (i.e., probar que A es válida mostrando que $\neg A$ es insatisfactible).
- Consiste en la aplicación sucesiva de una regla de inferencia a un conjunto de cláusulas.

Satisfactibilidad y validez

En general,

- Una *asignación* asocia variables a valores del dominio.
- Una fórmula A es *válida* sii toda asignación la hace verdadera.
- Una fórmula A es *satisfactible* sii alguna asignación la hace verdadera.

El siguiente hecho permite utilizar al método como técnica de demostración:

$$A \text{ es válida sii } \neg A \text{ es insatisfactible}$$

Pasaje a FNC

Paso a paso

1. Eliminar implicación
2. Forma normal negada
3. Forma normal prenexa (opcional)
4. Forma normal de Skolem (dependencias = variables libres dentro del alcance del $\exists$)
5. Forma normal conjuntiva
6. Distribución de cuantificadores y renombre de variables

Ejemplo pasaje a FNC

Pasar a Forma Normal Conjuntiva:

$\forall X\ \forall Y\ (\neg \mathit{iguales}(X, Y) \Leftrightarrow (\exists Z\ \mathit{menor}(X, Z) \wedge \mathit{menor}(Z, Y))$

La regla de resolución en primer orden

$$\frac{\mathcal{A}_i = \{A_1, \ldots, A_m, P_1, \ldots, P_k\} \quad \mathcal{A}_j = \{B_1, \ldots, B_n, \neg Q_1, \ldots, \neg Q_l\}}{\mathcal{B} = \sigma(\{A_1, \ldots, A_m, B_1, \ldots, B_n\})}$$

- σ es el MGU de $\{P_1, \ldots, P_k, Q_1, \ldots, Q_l\}$
  es decir, $\sigma(P_1) = \ldots = \sigma(P_k) = \sigma(Q_1) = \ldots = \sigma(Q_l)$
- A $\mathcal{B}$ se lo llama **resolvente** (de $\mathcal{A}_i$ y $\mathcal{A}_j$)
- Cada paso de resolución **preserva satisfactibilidad** (Teorema de Herbrand-Skolem-Gödel)

Ejemplo

Recuperatorio 2° parcial 1° Cuat. 2012

- Representar en forma clausal la siguiente información referida a conjuntos, pertenencia (predicado Pert) e inclusión (predicado Inc).
  - i $\forall X\ \forall Y\ (\text{Inc}(X, Y) \Leftrightarrow (\forall Z\ \text{Pert}(Z, X) \supset \text{Pert}(Z, Y)))$
    X está incluido en Y si y sólo si cada elemento de X es un elemento de Y.
  - ii $\forall X\ \neg\text{Pert}(X, \emptyset)$
    Ningún elemento pertenece al vacío.
- Usar resolución para probar que el vacío está incluido en todo conjunto.
- Indicar justificando si la prueba realizada es SLD (volveremos sobre esto más adelante).

Ejemplo

Recuperatorio 2° parcial 1° Cuat. 2012

Cast.:	$X \subseteq Y$ si y sólo si cada elemento de X es un elemento de Y.
1° o.:	$\forall X\ \forall Y\ (\mathsf{Inc}(X, Y) \Leftrightarrow (\forall Z\ \mathsf{Pert}(Z, X) \supset \mathsf{Pert}(Z, Y)))$
Claus.:	$\{\neg\mathsf{Inc}(X_1, Y_1), \neg\mathsf{Pert}(Z_1, X_1), \mathsf{Pert}(Z_1, Y_1)\}$
	$\{\mathsf{Inc}(X_2, Y_2), \mathsf{Pert}(\mathsf{f}(X_2, Y_2), X_2)\}$
	$\{\mathsf{Inc}(X_3, Y_3), \neg\mathsf{Pert}(\mathsf{f}(X_3, Y_3), Y_3)\}$
Cast.:	Ningún elemento pertenece al vacío.
1° o.:	$\forall X\ \neg\mathsf{Pert}(X, \emptyset)$
Claus.:	$\{\neg\mathsf{Pert}(X_4, \emptyset)\}$

A partir de ellas, se desea demostrar que:

Cast.:	El vacío está incluido en todo conjunto.
1° o.:	$\forall X\ \mathsf{Inc}(\emptyset, X)$
Neg.:	$\exists X\ \neg\mathsf{Inc}(\emptyset, X)$
Claus.:	$\{\neg\mathsf{Inc}(\emptyset, \mathsf{c})\}$

Ejemplo (resolviendo)

Recuperatorio 2° parcial 1° Cuat. 2012

$$\frac{\{A_1, ..., A_m, P_1, ..., P_k\} \qquad \{B_1, ..., B_n, \neg Q_1, ..., \neg Q_l\}}{\sigma(\{A_1, ..., A_m, B_1, ..., B_n\})}$$

donde σ es el MGU de $\{P_1, ..., P_k, Q_1, ..., Q_l\}$.

1. $\{\neg \text{Inc}(X_1, Y_1), \neg \text{Pert}(Z_1, X_1), \text{Pert}(Z_1, Y_1)\}$
2. $\{\text{Inc}(X_2, Y_2), \text{Pert}(\text{f}(X_2, Y_2), X_2)\}$
3. $\{\text{Inc}(X_3, Y_3), \neg \text{Pert}(\text{f}(X_3, Y_3), Y_3)\}$
4. $\{\neg \text{Pert}(X_4, \emptyset)\}$
5. $\{\neg \text{Inc}(\emptyset, \text{c})\}$
6. (2 y 5) $\{\text{Pert}(\text{f}(\emptyset, \text{c}), \emptyset)\}$ $\sigma = \{X_2 \leftarrow \emptyset,\ Y_2 \leftarrow \text{c}\}$
7. (6 y 4) $\square$ $\sigma = \{X_4 \leftarrow \text{f}(\emptyset, \text{c})\}$

Resolución en lógica de primer orden

Repaso

Estrategia

- Para demostrar que la fórmula F es universalmente válida
  Demostramos que $\neg F$ es insatisfactible.
- Para demostrar que F se deduce de $H_1, \ldots H_n$
  Demostramos que $H_1, \ldots, H_n, \neg F$ es insatisfactible.

Esquema general

- Expresar la o las fórmulas como cláusulas.
- Aplicar sucesivamente un paso de resolución
  (generando nuevas cláusulas)...
- Hasta llegar a la cláusula vacía
  o concluir que no es posible llegar a ella.
- Importante: al aplicar resolución suelen presentarse varias opciones.
  Conviene tener un plan.

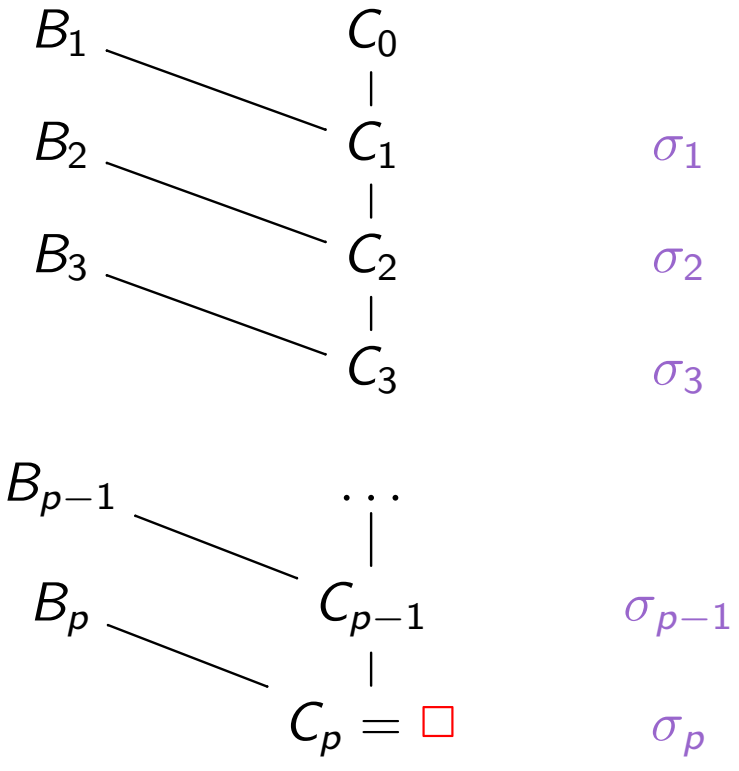
Cosas importantes para recordar[1]

- Al skolemizar, usar la misma constante o función si y sólo si la variable que estamos eliminando es **la misma** (nunca para otras, aun si tienen el mismo nombre).
- Para encontrar las dependencias, ver qué variables están libres dentro del alcance del $\exists$ (sin contar la que se está eliminando).
- ¡No olvidarse de negar lo que se quiere demostrar! Y recordar que $\neg((A_1 \wedge ... \wedge A_n) \supset B) = A_1 \wedge ... \wedge A_n \wedge \neg B$.
- Antes de empezar a aplicar pasos de resolución, convencerse de que lo que se quiere demostrar es verdadero, y trazar un plan para demostrarlo (mentalmente o por escrito).
- Recordar bien cómo funciona la unificación, y sustituir siempre **variables** (ni funciones, ni constantes, ni predicados).

[1] Seguir las indicaciones de esta lista previene los errores más frecuentes en los parciales.

Cómo mantenernos en línea

Si un conjunto de cláusulas es insatisfactible, existe una secuencia de pasos de resolución *lineal* que lo refuta (prueba su insatisfactibilidad). Es decir, una secuencia de la forma:



donde C_0 y cada B_i es un elemento de S o algún C_j con $j < i$.

Resolución SLD (*Selective Linear Definite*)

La resolución es cara, pero hay cupones de descuento...

- El método de resolución es completo, pero ineficiente.
- El espacio de búsqueda - inicialmente cuadrático - crece en cada paso.
- Resolución lineal reduce el espacio de búsqueda.
- Resolución SLD es lineal y (un poco) más eficiente, preservando completitud...
  ¡pero no puede aplicarse a cualquier conjunto de cláusulas!

Cláusulas de Horn

Cláusulas con a lo sumo un literal positivo

- ~~$\{P(x), P(y), \neg Q(y, z)\}$~~
- $\{Q(E, z)\}$ ✓ → cláusula de definición (*hecho*)
- $\{P(x), \neg P(E)\}$ ✓ → cláusula de definición (*regla*)
- ~~$\{P(x), \neg P(E), Q(x, y)\}$~~
- $\{P(x), \neg P(E), \neg Q(x, y)\}$ ✓ → cláusula de definición (*regla*)
- $\{\neg P(x), \neg P(E), \neg Q(x, y)\}$ ✓ → cláusula objetivo

No toda fórmula puede expresarse como una cláusula de Horn

$$\forall x(P(x) \vee Q(x))$$

Resolución SLD

Un caso particular de la resolución general

- Cláusulas de Horn con exactamente una cláusula objetivo
- Resolvemos la cláusula objetivo con una cláusula de definición
- Eso nos da otra cláusula objetivo
- Repetimos el proceso con esta nueva cláusula
- Hasta llegar a la cláusula vacía
- Si se busca un resultado, computamos la sustitución respuesta

$$\frac{\overbrace{\{R, \neg B_1, \ldots, \neg B_n\}}^{\text{def.}} \qquad \overbrace{\{\neg A_1, \ldots, \neg A_{k-1}, \neg A_k, \neg A_{k+1}, \ldots, \neg A_m\}}^{\text{obj.}}}{\underbrace{\sigma(\{\neg A_1, \ldots, \neg A_{k-1}, \neg B_1, \ldots, \neg B_n, \neg A_{k+1}, \ldots, \neg A_m\})}_{\text{nuevo obj.}}}$$

donde σ es el MGU de $\{R, A_k\}$.

Volviendo al primer ejercicio que resolvimos...

1. $\{\neg \mathsf{Inc}(X_1, Y_1), \neg \mathsf{Pert}(Z_1, X_1), \mathsf{Pert}(Z_1, Y_1)\}$
2. $\{\mathsf{Inc}(X_2, Y_2), \mathsf{Pert}(\mathsf{f}(X_2, Y_2), X_2)\}$
3. $\{\mathsf{Inc}(X_3, Y_3), \neg \mathsf{Pert}(\mathsf{f}(X_3, Y_3), Y_3)\}$
4. $\{\neg \mathsf{Pert}(X_4, \emptyset)\}$
5. $\{\neg \mathsf{Inc}(\emptyset, \mathsf{c})\}$
6. (2 y 5) $\{\mathsf{Pert}(\mathsf{f}(\emptyset, \mathsf{c}), \emptyset)\}$ $\sigma = \{X_2 \leftarrow \emptyset,\ Y_2 \leftarrow \mathsf{c}\}$
7. (6 y 4) $\square$ $\sigma = \{X_4 \leftarrow \mathsf{f}(\emptyset, \mathsf{c})\}$

¿Esto es SLD? ¿Por qué, o por qué no?

Resolución SLD

Ejemplo (computando una solución)

"Los enemigos de mis enemigos son mis amigos."

1. $\{\text{amigo}(A, B), \neg\text{enemigo}(A, C), \neg\text{enemigo}(C, B)\}$
2. $\{\text{enemigo}(\text{Reed}, \text{Dr. Doom})\}$
3. $\{\text{enemigo}(\text{Dr. Doom}, \text{Ben})\}$
4. $\{\text{enemigo}(\text{Dr. Doom}, \text{Johnny})\}$
5. $\{\neg\text{amigo}(\text{Reed}, X)\}$
6. (1 y 5) $\{\neg\text{enemigo}(\text{Reed}, C), \neg\text{enemigo}(C, B)\}$
   $\sigma_6 = \{A \leftarrow \text{Reed}, X \leftarrow B\}$
7. (2 y 6) $\{\neg\text{enemigo}(\text{Dr. Doom}, B)\}$
   $\sigma_7 = \{C \leftarrow \text{Dr. Doom}\}$
8. (3 y 7) $\square$ $\sigma_8 = \{B \leftarrow \text{Ben}\}$
   $\sigma = \sigma_8 \circ \sigma_7 \circ \sigma_6 =$
   $\{A \leftarrow \text{Reed}, X \leftarrow \text{Ben}, B \leftarrow \text{Ben}, C \leftarrow \text{Dr. Doom}\}$

Árbol de resolución

¡Es una secuencia!

- La resolución SLD es lineal: no hay vuelta atrás posible.
- Si el objetivo puede resolverse con más de una regla, elegir la correcta.
- Si hay más de una, elegir cualquiera.
- Si nos equivocamos, entonces lo que hicimos no es parte de la resolución SLD.
- Puede haber varias resoluciones SLD posibles.
- Prolog intenta buscar todas (resolución SLD + backtracking).

Resolución SLD y Prolog

Preguntas generales

- El mecanismo de búsqueda en la resolución SLD ¿está determinado?
- ¿El método es completo?
- ¿Prolog usa resolución SLD? ¿Su método es completo? ¿Está determinado?
- ¿Dónde está el problema (o la diferencia)?

Resolución SLD y Prolog

El ejemplo anterior en Prolog

"Los enemigos de mis enemigos son mis amigos."

$\{\text{amigo}(A,B), \neg\text{enemigo}(A,C), \neg\text{enemigo}(C,B)\}$	`amigo(A,B) :- enemigo(A,C), enemigo(C,B).`
$\{\text{enemigo}(\text{Reed}, \text{Dr. Doom})\}$	`enemigo(reed,drdoom).`
$\{\text{enemigo}(\text{Dr. Doom}, \text{Ben})\}$	`enemigo(drdoom,ben).`
$\{\text{enemigo}(\text{Dr. Doom}, \text{Johnny})\}$	`enemigo(drdoom,johnny).`
$\{\neg\text{amigo}(\text{Reed}, X)\}$	`?- amigo(reed,X).`

¿Cuál es la relación? ¿Cualquier ejemplo se puede traducir así?
¿Qué hay que tener en cuenta?

Resolución SLD y Prolog

Veamos ahora este ejemplo tomado de la práctica de Prolog:

1. `natural(0).`
2. `natural(suc(X)) :- natural(X).`
3. `menorOIgual(X, suc(Y)) :- menorOIgual(X, Y).`
4. `menorOIgual(X,X) :- natural(X).`

¿Qué pasa en Prolog si ejecutamos la consulta `menorOIgual(0,X)`?

¿Podremos encontrar la respuesta usando resolución? Veámoslo en el pizarrón.

Fin □