Práctica N^{o} 6 - Resolución en Lógica

Resolución en Lógica Proposicional

Ejercicio 1 ★

Convertir a Forma Normal Conjuntiva y luego a Forma Clausal (notación de conjuntos) las siguientes fórmulas proposicionales:

I. $p \supset p$

II. $(p \wedge q) \supset p$

III. $(p \vee q) \supset p$

IV. $\neg(p \Leftrightarrow \neg p)$

V. $\neg(p \wedge q) \supset (\neg p \vee \neg q)$

VI. $(p \wedge q) \vee (p \wedge r)$

VII. $(p \wedge q) \supset r$

VIII. $p \supset (q \supset r)$

Ejercicio 2 ★

I. ¿Cuáles de las fórmulas del ejercicio anterior son tautologías? Demostrarlas utilizando el método de resolución para la lógica proposicional. Para las demás, indicar qué pasa si se intenta demostrarlas usando este método.

II. ¿Se deduce $(p \wedge q)$ de $(\neg p \supset q) \wedge (p \supset q) \wedge (\neg p \supset \neg q)$? Contestar utilizando el método de resolución para la lógica proposicional.

Ejercicio 3 ★

Un grupo de amigos quería juntarse a comer en una casa, pero no decidían en cuál. Prevalecían dos propuestas: la casa de Ana, que era cómoda y espaciosa, y la de Carlos, más chica pero con un amplio jardín y parrilla al aire libre. Finalmente acordaron basar su elección en el pronóstico del tiempo. Si anunciaban lluvia, se reunirían en la casa de Ana; y si no, en la de Carlos (desde ya, la reunión tendría lugar en una sola casa).

Finalmente llego el día de la reunión, y el grupo se juntó a comer en la casa de Ana, pero no llovió.

Utilizar las siguientes proposiciones para demostrar - mediante el método de resolución - que el pronóstico se equivocó (anunció lluvia y no llovió, o viceversa).

$p =$ "El pronóstico anunció lluvia."

$a =$ "El grupo se reúne en la casa de Ana."

$c =$ "El grupo se reúne en la casa de Carlos."

$l =$ "Llueve en el día de la reunión."

Ayuda: por la descripción de arriba sabemos que $p \supset a$, $\neg p \supset c$ y $\neg(a \wedge c)$, además de que a y $\neg l$ son verdaderas. Pensar en lo que se quiere demostrar para decidir qué pares de cláusulas utilizar.

Unificación en Lógica de Primer Orden

Ejercicio 4 ★

Unir con flechas los términos que unifican entre sí (entre una fila y la otra). Para cada par unificable, exhibir el *mgu* ("most general unifier"). Asumir que a es una constante, x, y, z son variables, f y g son símbolos de función, y P y Q predicados.

$$P(f(x)) \qquad P(\mathsf{a}) \qquad P(y) \qquad Q(x, f(y)) \qquad Q(x, f(z)) \qquad Q(x, f(\mathsf{a}))$$

$$P(x) \qquad P(f(\mathsf{a})) \qquad P(g(z)) \qquad Q(f(y), x) \qquad Q(f(y), f(x)) \qquad Q(f(y), y)$$

Ejercicio 5 ★

Determinar, para cada uno de los siguientes pares de términos de primer orden, si son unificables o no. En cada caso justificar su respuesta exhibiendo una secuencia exitosa o fallida (según el caso) del algoritmo de Martelli-Montanari. Asimismo, en caso de que los términos sean unificables indicar el *mgu* ("most general unifier"). Notación: x, y, z variables; a, b, c constantes; f, g símbolos de función.

I. $f(x, x, y)$ y $f(a, b, z)$

II. $f(x)$ y y

III. $f(g(c, y), x))$ y $f(z, g(z, a))$

IV. $f(a)$ y $g(y)$

V. $f(x)$ y x

VI. $g(x, y)$ y $g(f(y), f(x))$

Resolución en Lógica de Primer Orden

En esta sección, salvo que se haga referencia a SLD, la palabra *resolución* denota el método de resolución general. Siempre que se demuestre una propiedad, se deberá indicar la sustitución utilizada en cada paso de resolución.

Ejercicio 6

Convertir a Forma Normal Negada (NNF) las siguientes fórmulas de primer orden:

I. $\forall x.\forall y.(\neg Q(x, y) \supset \neg P(x, y))$

II. $\forall x.\forall y.((P(x, y) \wedge Q(x, y)) \supset R(x, y))$

III. $\forall x.\exists y.(P(x, y) \supset Q(x, y))$

Ejercicio 7 ★

Convertir a Forma Normal de Skolem y luego a Forma Clausal las siguientes fórmulas de primer orden:

I. $\exists x.\exists y.x < y$, siendo $<$ un predicado binario usado de forma infija.

II. $\forall x.\exists y.x < y$

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

IV. $\exists x.\forall y.(P(x, y) \wedge Q(x) \wedge \neg R(y))$

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

Ejercicio 8

Para pensar (o jugar):

I. Exhibir una cláusula que arroje un resolvente consigo misma.

II. Exhibir dos cláusulas, cada una con no más de dos literales, que arrojen tres o más resolventes distintos entre sí.

III. Exhibir dos cláusulas que arrojen como resolvente $\Box$ si se unifican tres o más términos a la vez, pero no si se unifica solamente un término de cada lado.

Ejercicio 9 ★

¿Cuáles las siguientes fórmulas son lógicamente válidas? Demostrarlas usando resolución.

I. $\exists x.\forall y.R(x,y) \supset \forall y.\exists x.R(x,y)$

II. $\forall x.\exists y.R(x,y) \supset \exists y.\forall x.R(x,y)$

III. $\exists x.[P(x) \supset \forall x.P(x)]$

IV. $\exists x.[P(x) \vee Q(x)] \supset [\exists x.P(x) \vee \exists x.Q(x)]$

V. $\forall x.[P(x) \vee Q(x)] \supset [\forall x.P(x) \vee \forall x.Q(x)]$

VI. $[\exists x.P(x) \wedge \forall x.Q(x)] \supset \exists x.[P(x) \wedge Q(x)]$

VII. $\forall x.\exists y.\forall z.\exists w.[P(x,y) \vee \neg P(w,z)]$

VIII. $\forall x.\forall y.\forall z.[\neg P(f(\mathsf{a})) \vee \neg P(y) \vee Q(y)] \wedge P(f(z)) \wedge [\neg P(f(f(x))) \vee \neg Q(f(x))]$

Ejercicio 10 *(Aplicaciones del método de resolución)*

I. Expresar en forma clausal la regla del *modus ponens* y mostrar que es válida, usando resolución.

II. Lo mismo para la regla del *modus tollens*.

Ejercicio 11 ★

Dadas las siguientes cláusulas:

- $\{P(x), \neg P(x), Q(a)\}$
- $\{P(x), \neg Q(y), \neg R(x,y)\}$
- $\{\neg P(x,x,z), \neg Q(x,y), \neg Q(y,z)\}$
- $\{M(1,2,x)\}$

I. ¿Cuáles son cláusulas de Horn?

II. Para cada cláusula de Horn indicar si es una cláusula de definición (hecho o regla) o una cláusula objetivo.

III. Dar, para cada cláusula, la fórmula de primer orden que le corresponde.

Ejercicio 12 ★

Indicar cuáles de las siguientes condiciones son necesarias para que una demostración por resolución sea SLD.

- Realizarse de manera lineal (utilizando en cada paso el resolvente obtenido en el paso anterior).
- Utilizar únicamente cláusulas de Horn.
- Utilizar cada cláusula a lo sumo una vez.
- Empezar por una cláusula objetivo (sin literales positivos).
- Empezar por una cláusula que provenga de la negación de lo que se quiere demostrar.
- Recorrer el espacio de búsqueda de arriba hacia abajo y de izquierda a derecha.

- Utilizar la regla de resolución binaria en lugar de la general.

Ejercicio 13 ★

Alan es un robot japonés. Cualquier robot que puede resolver un problema lógico es inteligente. Todos los robots japoneses pueden resolver todos los problemas de esta práctica. Todos los problemas de esta práctica son lógicos. Existe al menos un problema en esta práctica. ¿Quién es inteligente? Encontrarlo utilizando resolución SLD y composición de sustituciones.

Utilizar los siguientes predicados y constantes: $R(x)$ para expresar que x es un robot, $Res(x, y)$ para x puede resolver y, $PL(x)$ para x es un problema lógico, $Pr(x)$ para x es un problema de esta práctica, $I(x)$ para x es inteligente, $J(x)$ para x es japonés y la constante alan para Alan.

Ejercicio 14 ★

Sean las siguientes cláusulas (en forma clausal), donde *suma* y *par* son predicados, *suc* es una función y *cero* una constante:

1. $\{\neg suma(x, y, z), suma(x, suc(y), suc(z))\}$
2. $\{suma(x, cero, x)\}$
3. $\{\neg suma(x, x, y), par(y)\}$

Demostrar utilizando resolución que suponiendo (1), (2), (3) se puede probar *par(suc(suc(cero)))*. Si es posible, aplicar resolución SLD. En caso contrario, utilizar resolución general. Mostrar en cada aplicación de la regla de resolución la sustitución utilizada.

Ejercicio 15

I. Pasar las siguientes fórmulas en lógica de primer orden a forma clausal.
   a) $\forall c(V(c) \vee \exists e(P(e, c)))$
   b) $\neg\exists c(V(c) \wedge \exists e(P(e, c)))$
   c) $\forall e \forall c(P(e, I(c)) \Leftrightarrow P(e, c))$

II. A partir de las cláusulas definidas en el punto anterior, ¿puede demostrarse $\forall c(V(I(c)) \supset V(c))$ usando resolución SLD? Si se puede, hacerlo. Si no, demostrarlo usando el método de resolución general.

Ejercicio 16 ★

Un lógico estaba sentado en un bar cuando se le ocurrió usar el método de resolución para demostrar el teorema del bebedor: siempre que haya alguien en el bar, habrá allí alguien tal que, si está bebiendo, todos en el bar están bebiendo. Sin embargo, el lógico en cuestión había bebido demasiado y la prueba no le salió muy bien. Esto fue lo que escribió en una servilleta del bar:

Teorema del bebedor:	$(\exists X\, \mathtt{enBar}(X)) \supset \exists Y(\mathtt{enBar}(Y) \wedge (\mathtt{bebe}(Y) \supset \forall Z(\mathtt{enBar}(Z) \supset \mathtt{bebe}(Z))))$
Elimino implicaciones:	$(\neg\exists X\, \mathtt{enBar}(X)) \vee \exists Y(\mathtt{enBar}(Y) \wedge (\neg\mathtt{bebe}(Y) \vee \forall Z(\neg\mathtt{enBar}(Z) \vee \mathtt{bebe}(Z))))$
Skolemizo:	$(\neg\mathtt{enBar}(c)) \vee (\mathtt{enBar}(k) \wedge (\neg\mathtt{bebe}(k) \vee \forall Z(\neg\mathtt{enBar}(Z) \vee \mathtt{bebe}(Z))))$
Paso a Forma Clausal:	1. $\{\neg\mathtt{enBar}(c)\}$ 2. $\{\mathtt{enBar}(k)\}$ 3. $\{\neg\mathtt{bebe}(k)\}$ 4. $\{\neg\mathtt{enBar}(Z), \mathtt{bebe}(Z)\}$

Aplico resolución:

De 3 y 4 con $\sigma = \{k \leftarrow Z\}$:

5. $\{\neg\mathtt{enBar}(Z)\}$

De 5 y 1 con $\sigma = \{Z \leftarrow c\}$:

$\square$

a) Identificar los 5 errores cometidos en la demostración. (La fórmula original es correcta, notar que salteó pasos importantes e hizo mal otros).

b) Demostrar el teorema de manera correcta, usando resolución.

c) Indicar si la resolución utilizada en el punto b es o no SLD. Justificar.

Ejercicio 17

Dadas las siguientes afirmaciones:

- Toda persona tiene un contacto en Facebook:
  $\forall x\ \exists y\ \mathsf{esContacto}(x, y)$
  1. $\{\mathsf{esContacto}(x, \mathsf{f}(x))\}$
- Si una persona es contacto de otra, la otra es contacto de la una:
  $\forall x\ \forall y\ (\mathsf{esContacto}(x, y) \supset \mathsf{esContacto}(y, x))$
  2. $\{\neg\mathsf{esContacto}(x, y), \mathsf{esContacto}(y, x)\}$

I. La siguiente es una demostración de que toda persona es contacto de sí misma, es decir, de que

$$\forall x\ \mathsf{esContacto}(x, x)$$

- Negando la conclusión:
  $\neg\forall x\ \mathsf{esContacto}(x, x)$
- Forma normal negada:
  $\exists x\ \neg\mathsf{esContacto}(x, x)$
- Skolemizando y en forma clausal:
  3. $\{\neg\mathsf{esContacto}(\mathsf{c}, \mathsf{c})\}$
- De 1 y 3, con $\sigma = \{x \leftarrow \mathsf{c}, \mathsf{f}(x) \leftarrow \mathsf{c}\}$:
  $\square$

¿Es correcta? Si no lo es, indicar el o los errores.

II. La siguiente es una demostración de que toda persona es contacto de alguien, es decir, de que

$$\forall y\ \exists x\ \mathsf{esContacto}(x, y)$$

- Negando la conclusión:
  $\neg\forall y\ \exists x\ \mathsf{esContacto}(x, y)$
- Forma normal negada:
  $\exists y\ \forall x\ \neg\mathsf{esContacto}(x, y)$
- Skolemizando y en forma clausal:
  3. $\{\neg\mathsf{esContacto}(x, \mathsf{d})\}$
- De 2 y 3, con $\sigma = \{x \leftarrow \mathsf{d}, y \leftarrow \mathsf{d}\}$:
  4. $\{\neg\mathsf{esContacto}(\mathsf{d}, \mathsf{d})\}$
- De 1 y 4, con $\sigma = \{\mathsf{d} \leftarrow x, \mathsf{d} \leftarrow \mathsf{f}(x)\}$:
  $\square$

¿Es correcta? Si no lo es, indicar el o los errores.

III. ¿Puede deducirse de las dos premisas que toda persona es contacto de alguien? En caso afirmativo dar una demostración (puede usarse la del ítem anterior), y en caso contrario explicar por qué.

Ejercicio 18 ★

Dadas las siguientes definiciones de $\mathsf{Descendiente}$ y Abuelo a partir de la relación $\mathsf{Progenitor}$:

$\{\neg\mathsf{Progenitor}(x, y), \mathsf{Descendiente}(y, x)\}$ $\{\neg\mathsf{Descendiente}(x, y), \neg\mathsf{Descendiente}(y, z), \mathsf{Descendiente}(x, z)\}$
$\{\neg\mathsf{Abuelo}(x, y), \mathsf{Progenitor}(x, \mathsf{medio}(x, y))\}$ $\{\neg\mathsf{Abuelo}(x, y), \mathsf{Progenitor}(\mathsf{medio}(x, y), y)\}$

Demostrar usando resolución general que los nietos son descendientes; es decir, que

$$\forall x\ \forall y\ (\mathsf{Abuelo}(x, y) \supset \mathsf{Descendiente}(y, x))$$

Ayuda: tratar de aplicar el método a ciegas puede traer problemas. Conviene tener en mente lo que se quiere demostrar.

Ejercicio 19 ★

En este ejercicio usaremos el método de resolución para demostrar una propiedad de las relaciones binarias; a saber, que una relación no vacía no puede ser a la vez irreflexiva, simétrica y transitiva.

Para esto tomaremos una relación R y se demostrará que, si R satisface las tres propiedades mencionadas, entonces es vacía.

Dadas las siguientes definiciones:

1. R **es irreflexiva:** $\forall x \neg R(x,x)$
2. R **es simétrica:** $\forall x \forall y (R(x,y) \supset R(y,x))$
3. R **es transitiva:** $\forall x \forall y \forall z ((R(x,y) \wedge R(y,z)) \supset R(x,z))$
4. R **es vacía:** $\forall x \neg \exists y R(x,y)$

Utilizando resolución, demostrar que sólo una relación vacía puede cumplir a la vez las propiedades 1 a 3. Indicar si el método de resolución utilizado es o no SLD (y justificar).

Ejercicio 20 ★

Considerar las siguientes definiciones en Prolog:

```
natural(cero).
natural(suc(X)) :- natural(X).
mayorOIgual(suc(X),Y) :- mayorOIgual(X, Y).
mayorOIgual(X,X) :- natural(X).
```

- ¿Qué sucede al realizar la consulta `?- mayorOIgual(suc(suc(N))), suc(cero)`?
- Utilizar el método de resolución para probar la validez de la consulta del ítem 1. Para ello, convertir las cláusulas a forma clausal.
- Indicar si el método de resolución utilizado es o no SLD, y justificar. En caso de ser SLD, ¿respeta el orden en que Prolog hubiera resuelto la consulta?

Ejercicio 21

Dado el siguiente programa en Prolog, pasarlo a forma clausal y demostrar utilizando resolución que hay alguien que es inteligente pero analfabeto.

```
analfabeto(X) :- vivo(X), noSabeLeer(X).
vivo(X) :- delfín(X).
inteligente(flipper).
inteligente(alan).
noSabeLeer(X) :- mesa(X).
noSabeLeer(X) :- delfín(X).
delfín(flipper).
```

Ejercicio 22

Considerar las siguientes definiciones en prolog:

```
preorder(nil,[]).
preorder(bin(I,R,D),[R|L]) :- append(LI,LD,L),
              preorder(I,LI), preorder(D,LD).
append([],YS,YS).
append([X|XS],YS,[X|L]) :- append(XS,YS,L).
```

- ¿Qué sucede al realizar la consulta `?- preorder(bin(bin(nil,2,nil),1,nil),Lista).`?
- Utilizar el método de resolución para encontrar la solución al problema. Para ello, convertir el programa a forma clausal.
- Indicar si el método de resolución utilizado es o no SLD, y justificar. En caso de ser SLD, ¿respeta el orden en que Prolog hubiera resuelto la consulta?