ALEJANDRO SANTORUM VARELA 44090446-S Gr. ~~3~~20 (DG)

$2^{\underline{o}}$ ENTREGA EJERCICIOS I.A.

1.

a) Formalización base de conocimiento

I) $\forall m \; \neg Igual(0, suc(m))$

II) $\forall n \left[\neg Igual(n,0) \Rightarrow \exists m \; Igual(n, suc(m)) \right]$

III) $\neg Par(1) \wedge Igual(1, suc(0)) \equiv \begin{array}{l} \neg Par(1) \\ Igual(1, suc(0)) \end{array}$

IV) $\forall n \left[\neg Par(suc(n)) \Rightarrow Par(n) \right]$

V) $\forall n \left[Par(suc(n)) \Rightarrow \neg Par(n) \right]$

VI) $\forall n \left[Igual(sum(n,0), n) \right]$

VII) $\forall n,m \left[Igual(suc(n,m), sum(n, suc(m))) \right]$

VIII) $\forall m,n \left[(Igual(n,m) \wedge Par(m)) \Rightarrow Par(n) \right]$ (adicional por el enunciado)

b) Conversión a FNC

I) $\forall m \left[\neg Igual(0, suc(m)) \right]$

- Eliminación del $\forall$: $\boxed{\neg Igual(0, suc(m)) \quad [1]}$

II) $\forall n \left[\neg Igual(n,0) \Rightarrow \exists m \; Igual(n, suc(m)) \right]$

- Eliminación del $\Rightarrow$ + reducción ámbito $\neg$: $\forall n \left[Igual(n,0) \vee \exists m \; Igual(n, suc(m)) \right]$
- Skolemnización: $\forall n \left[Igual(n,0) \vee Igual(n, suc(f(n))) \right]$
- Eliminación del $\forall$: $\boxed{Igual(n,0) \vee Igual(n, suc(f(n))) \quad [2]}$

III) $\neg Par(1) \wedge Igual(1, suc(0))$ $\boxed{\neg Par(1) \quad [3]}$ $\boxed{Igual(1, suc(0)) \quad [4]}$

$\equiv$ $\boxed{\neg Par(suc(0)) \quad [3]}$ PEANO

IV) $\forall n \left[\neg Par(suc(n)) \Rightarrow Par(n) \right]$

- Eliminación del $\Rightarrow$ + reducción ámbito $\neg$: $\forall n \left[Par(suc(n)) \vee Par(n) \right]$
- Eliminación del $\forall$: $\boxed{Par(suc(n)) \vee Par(n) \quad [5]}$

V) $\forall n\ [Par(suc(n)) \Rightarrow \neg Par(n)]$

- Eliminación del $\Rightarrow$: $\forall n\ [\neg Par(suc(n)) \vee \neg Par(n)]$
- Eliminación de $\forall$: $\boxed{\neg Par(suc(n)) \vee \neg Par(n) \quad [6]}$

VI) $\forall n\ [Igual(sum(n,0), n)]$

- Eliminación del $\forall$: $\boxed{Igual(sum(n,0), n) \quad [7]}$

VII) $\forall n,m\ [Igual(suc(sum(n,m)), sum(n, suc(m)))]$

- Eliminación del $\forall$: $\boxed{Igual(suc(sum(n,m)), sum(n, suc(m))) \quad [8]}$

VIII) $\forall n,m\ [(Igual(n,m) \wedge Par(m)) \Rightarrow Par(n)]$

- Eliminación del $\Rightarrow$: $\forall n,m\ [\neg(Igual(n,m) \wedge Par(m)) \vee Par(n)]$
- Ley de De Morgan: $\forall n,m\ (\neg Igual(n,m) \vee \neg Par(m) \vee Par(n))$
- Eliminación del $\forall$: $\boxed{\neg Igual(n,m) \vee \neg Par(m) \vee Par(n) \quad [9]}$

c) Demostrar que la suma de un nº natural par con uno es impar.

$\forall n\ [Par(n) \Rightarrow \neg Par(sum(n, suc(0)))]$ (META)

Negamos la META:

$\neg \forall n\ [Par(n) \Rightarrow \neg Par(sum(n, suc(0)))]$

Y la transformamos en FNC:

- Eliminación del $\Rightarrow$: $\neg \forall n\ [\neg Par(n) \vee \neg Par(sum(n, suc(0)))]$
- Ámbito $\neg$: $\exists n\ Par(n) \wedge Par(sum(n, suc(0)))$
- Skolemnización: $Par(sk_1) \wedge Par(sum(sk_1, suc(0)))$

$\boxed{Par(sk_1) \quad [10]}$

$\boxed{Par(sum(sk_1, suc(0))) \quad [11]}$

Utilizamos ahora resolución:

$[10], [6] \vdash_{\substack{n:=SK_1 \\ RES\ \neg Par(SK_1)}} \neg Par(suc(SK_1))$ [12]

$[11], [8] \vdash_{equiv.} Par(suc(sum(SK_1, 0)))$ [13]

$[13], [7] \vdash_{equiv.} Par(suc(SK_1))$ [14]

$[12], [14] \vdash_{RES\ Par(suc(SK_1))} \square$ Falso $\Longrightarrow$ demostrado ■

d) TRUCO DE GREEN: determinar si existe algún número natural par que no sea sucesor de un número impar y cuál es.

META: $\exists n \left[Par(n) \wedge \left(\neg \exists m \left(\neg Par(m) \wedge Igual(n, suc(m)) \right) \right) \right]$

NEG META: + FNC $\neg \exists n \left[Par(n) \wedge \left(\neg \exists m \left(\neg Par(m) \wedge Igual(n, suc(m)) \right) \right) \right]$

- Ámbito negaciones + distributiva:

  $\forall n \left[\neg Par(n) \vee \left(\neg Par(f(n)) \wedge Igual(n, suc(f(n))) \right) \right]$

- Elim $\forall$ + Respuesta(n) truco de Green:

$\boxed{\neg Par(n) \vee \neg Par(f(n)) \vee Res(n) \quad [10]}$

$\boxed{\neg Par(n) \vee Igual(n, suc(f(n))) \vee Res(n) \quad [11]}$

$[5], [10] \vdash_{RES\ Par(n)} Par(suc(n)) \vee \neg Par(f(n)) \vee Res(n)$ [12]

$[5], [11] \vdash_{RES\ Par(n)} Par(suc(n)) \vee Igual(n, suc(f(n))) \vee Res(n)$ [13]

$[12], [3] \vdash_{\substack{n:=0 \\ RES\ Par(suc(0))}} \neg Par(f(0)) \vee Res(0)$ [14]

$[13], [3] \vdash_{\substack{n:=0 \\ RES\ Par(suc(0))}} Igual(0, suc(f(0))) \vee Res(0)$ [15]

$[4], [15] \vdash_{RES\ Igual(\ldots)} Res(0)$

$\Downarrow$

Según el truco de Green lo cumple el 0

Añadimos a la base de conocimiento inicial la negación de la siguiente meta:

$$\forall x,y \left[\left(P(x) \wedge P(y) \wedge \neg \exists z \; suc(x,z) \wedge \neg \exists w (y,w) \right) \Rightarrow (x=y) \right]$$

Llegamos a que el 0 es el único neutro.

2. Resolución de un problema con resolución + refutación:

1er paso: Indicar constantes, predicados y funciones usadas

CONSTANTES: Groucho (persona)

VARIABLES: $x_1, x_2, \dots$ (personas)

$C_1, C_2, \dots$ (clubs)

PREDICADOS: $SC(x_1, x_2)$: evalúa a verdadero si x_1 es como x_2.

$SM(x_1, C_1)$: evalúa a verdadero si x_1 es miembro de C_1

$QSM(x_1, C_1)$: evalúa a verdadero si x_1 quiere ser miembro de C_1

$A(C_1, x_1)$: evalúa a verdadero si C_1 ha aceptado/acepta a x_1

2º paso: Formalización de la base de conocimiento

I) $\forall x \; SC(x,x)$

II) $\forall x, c \left[SM(x,c) \iff (QSM(x,c) \wedge A(c,x)) \right]$

III) Groucho ya es una constante en el ámbito de personas

IV) $\forall x, c \left[(SC(x, \text{Groucho}) \wedge A(c,x)) \implies \neg QSM(\text{Groucho}, c) \right]$

3er paso: Conversión a FNC

I) $\boxed{SC(x,x) \quad [1]}$

II) $\forall x, c \left\{ \left[SM(x,c) \implies (QSM(x,c) \wedge A(c,x)) \right] \wedge \left[(QSM(x,c) \wedge A(c,x)) \implies SM(x,c) \right] \right\}$

$\forall x, c \left\{ \left[\neg SM(x,c) \vee (QSM(x,c) \wedge A(c,x)) \right] \wedge \left[\neg QSM(x,c) \vee \neg A(c,x) \vee SM(x,c) \right] \right.$

$(\neg SM(x,c) \vee QSM(x,c)) \wedge (\neg SM(x,c) \vee A(c,x)) \wedge (\neg QSM(x,c) \vee \neg A(c,x) \vee SM($

Entonces:

$\boxed{\neg SM(x,c) \vee QSM(x,c) \quad [2]}$

$\boxed{\neg SM(x,c) \vee A(c,x) \quad [3]}$

$\boxed{\neg QSM(x,c) \vee \neg A(c,x) \vee SM(x,c) \quad [4]}$

III) $\forall x, c\ [\neg SC(x, Groucho) \vee \neg A(c, x) \vee \neg QSM(Groucho, c)]$

$\boxed{\neg SC(x, Groucho) \vee \neg A(c, x) \vee \neg QSM(Groucho, c) \quad [5]}$

1º paso: Formalizamos meta, la negamos y la convertimos a FNC

META: $\forall c\ \neg SM(Groucho, c)$

META NEGADA $\exists c\ SM(Groucho, c)$

FNC: $\boxed{SM(Groucho, SK_c) \quad [6]}$

2º paso: Resolución + refutación buscando la cláusula vacía

[3], [6] $\dfrac{x := Groucho \quad c := SK_c}{RES_{SM}}$ $A(SK_c, Groucho)$ [7]

[2], [6] $\dfrac{x := Groucho \quad c := SK_c}{RES_{SM}}$ $QSM(Groucho, SK_c)$ [8]

[1], [8] $\dfrac{x := Groucho \quad c := SK_c}{RES_{QSM}}$ $\neg A(SK_c, Groucho) \vee SM(x, c)$ [9]

[7], [9] $\dfrac{x := Groucho \quad c := SK_c}{RES_{A}}$ $SM(Groucho, SK_c)$ [10] (inútil, ya la tenía)

[4], [5] $\dfrac{x := Groucho}{RES_{SC}}$ $\neg A(c, Groucho) \vee \neg QSM(Groucho, c)$ [11]

[8], [11] $\dfrac{c := SK_c}{RES_{QSM}}$ $\neg A(SK_c, Groucho)$ [12]

[7], [12] $\dfrac{}{RES_{A}}$ $\square$ $\Longrightarrow$ acabamos de demostrar que Groucho NO es miembro de ningún club.

$\boxed{3.}$ Resolución de un problema con resolución + refutación:

1^{er} paso: Indicar constantes, variables, predicados y funciones

CONSTANTES: 0

VARIABLES: x, y, z (números reales)

PREDICADOS: $Igual(x,y)$: evalúa a T si $x=y$

$Neutro(x)$: evalúa a T si x es el elem. neutro de la suma

FUNCIONES: $Resta(x,y)$: evalúa al número $(x-y)$.

2^{o} paso: Formalización de la base de conocimiento

I) $\forall x,y\ [Igual(Resta(x,y),0) \Leftrightarrow Igual(x,y)]$

II) $\forall x\ [Neutro(x) \Leftrightarrow (\forall y\ Igual(Resta(y,x),y))] \equiv$

$\equiv \forall x,y\ [Neutro(x) \Leftrightarrow Igual(Resta(y,x),y)]$

3^{er} paso: Conversión a FNC

I) $\forall x,y\ [(Igual(Resta(x,y),0) \Rightarrow Igual(x,y)) \land (Igual(x,y) \Rightarrow Igual(Resta(x,y),0))]$

$\forall x,y\ \{[\neg Igual(Resta(x,y),0) \lor Igual(x,y)] \land [\neg Igual(x,y) \lor Igual(Resta(x,y),0)]\}$

$\boxed{\neg Igual(Resta(x,y),0) \lor Igual(x,y) \quad [1]}$

$\boxed{\neg Igual(x,y) \lor Igual(Resta(x,y),0) \quad [2]}$

II) $\forall x,y\ [(Neutro(x) \Rightarrow Igual(Resta(y,x),y)) \land (Igual(Resta(y,x),y) \Rightarrow Neutro(x))]$

$\forall x,y\ \{[\neg Neutro(x) \lor Igual(Resta(y,x),y)] \land [\neg Igual(Resta(y,x),y) \lor Neutro(x)]\}$

$\boxed{\neg Neutro(x) \lor Igual(Resta(y,x),y) \quad [3]}$

$\boxed{\neg Igual(Resta(y,x),y) \lor Neutro(x) \quad [4]}$

1º paso: Formalizamos la meta, la negamos y la convertimos a FNC

META: $\forall x\ [\text{Neutro}(x) \Rightarrow \text{Igual}(x,0)]$

META NEGADA: $\exists x\ [\neg(\text{Neutro}(x) \Rightarrow \text{Igual}(x,0))]$

FNC: $\exists x\ [\neg(\neg\text{Neutro}(x) \vee \text{Igual}(x,0))]$

$\exists x\ (\text{Neutro}(x) \wedge \neg\text{Igual}(x,0))$

$\boxed{\text{Neutro}(SK)\quad [5]}$ $\boxed{\neg\text{Igual}(SK,0)\quad [6]}$

$\begin{matrix}[3] \\ [5]\end{matrix} \Big| \frac{x := SK}{RES_{\text{Neutro}}}$ $\text{Igual}(\text{Resta}(y,x), y)$ [7]

$\begin{matrix}[1] \\ [6]\end{matrix} \Big| \frac{x := SK,\ y := 0}{RES_{\text{Igual}}}$ $\neg\text{Igual}(\text{Resta}(SK,0), 0)$ [8]

$\begin{matrix}[1] \\ [7]\end{matrix} \Big| \frac{y := 0\ (*)}{RES_{\text{Igual(Resta}\cdots)}}$ $\text{Igual}(x,0)$ [9]

$\begin{matrix}[6] \\ [9]\end{matrix} \Big| \frac{x := SK}{RES_{\text{Igual}}}$ $\square$ $\Rightarrow$ acabamos de demostrar que no hay ningún elemento neutro aparte del cero.

$\neg\text{Igual}(\text{Resta}(x,y), 0) \vee \text{Igual}(x,y)$

$\text{Igual}(\text{Resta}(x,y), y)$

¿con sustitución $y := 0$ podemos hacer resolución y obtener $\neg\text{Igual}(\text{Resta}(x,0),0)$?

4.

Primero programaremos un predicado que evalúe a verdadero cuando se le pase el mínimo de la lista, e.d.
list_min(L, A) evalúa a verdadero sii A es el mínimo elemento de L:

```
list_min([L|Ls], Min) :- list_min(Ls, L, Min).
list_min([], Min, Min).
list_min([L|Ls], Min0, Min) :-
        Min1 is min(L, Min0),
        list_min(Ls, Min1, Min).
```

Otro predicado auxiliar delete_one(A, L, Ls) evalúa a verdadero sii A es el elemento eliminado de la lista L, resultando en Ls:

```
delete_one(_, [], []).
delete_one(Term, [Term|Tail], Tail).
delete_one(Term, [Head|Tail], [Head|Result]) :-
        delete_one(Term, Tail, Result).
```

Ahora ya podemos hacer los predicados propuestos:

```
minimo(Ls, A, Ys) :- list_min(Ls, A), delete_one(A, Ls, Ys).

ordenar([], []).
ordenar(L, [M|Rs]) :- minimo(L, M, Rest), ordenar(Rest, Rs).
```

$\boxed{5.}$ Vamos a crear dos predicados auxiliares antes de find: indexOf(Ls, Elem, Index) evalúa a verdadero si la primera aparición de Elem en Ls es Index (empezando en 1). Por otro lado, checkFirstOccur(Ls, E, N, Rest) evalúa a verdadero si la primera aparición del elemento E en Ls es en el índice N y la lista restante es Rest:

```
indexOf([Element|_], Element, 1) :- !.
indexOf([_|Tail], Element, Index) :-
    indexOf(Tail, Element, Index1),
    Index is Index1 + 1.

checkFirstOccur(WholeList, Elem, Pos, Rest) :-
    checkFirstOccur(WholeList, Elem, Pos, Rest, WholeList).
checkFirstOccur(WholeList, Elem, Pos, R, [Elem|R]) :-
    indexOf(WholeList, Elem, Pos).
checkFirstOccur(WholeList, Elem, Pos, Rest, [_|R2]) :-
    checkFirstOccur(WholeList, Elem, Pos, Rest, R2).

find(Ls, E, Rs) :- find(Ls, E, Rs, 0).
find(Ls, E, [], _) :- \+ member(E, Ls).
find(Ls, E, [N|M], Acum) :-
    checkFirstOccur(Ls, E, Naux, Rest),
    N is Naux + Acum,
    find(Rest, E, M, N).
```

6. Para este ejercicio vamos a crear un predicado auxiliar llamado checkElem(Ls, I, Elem) que evalúa a verdadero sii I está presente en Ls en el índice I.

```
checkElem([Elem|_], 1, Elem).
checkElem([_|Rest], I, Elem) :-
    NewI is I-1,
    checkElem(Rest, NewI, Elem).

seleccionar(_, [], []).
seleccionar(Ls, [I|Is], [R|Rs]) :-
    checkElem(Ls, I, R),
    seleccionar(Ls, Is, Rs).
```

7. Como no podía ser de otra forma, para este ejercicio vamos a implementar un predicado auxiliar checkRep(L, R, Rest) que evaluará a T sii L aparece al principio de R y Rest es el resto de R, sin L.

```
checkRep([], R, R).
checkRep([L|Ls], [L|Rs], Rest) :- checkRep(Ls, Rs, Rest)

replicar(_, 0, []) :- !.
replicar(Ls, N, Rs) :-
    checkRep(Ls, Rs, Rest),
    NewN is N-1,
    replicar(Ls, NewN, Rest).
```

$\exists n \{ Par(n) \wedge (\neg \exists m (\neg Par(m) \wedge Igual(n, suc(m)))) \}$

$\exists n \{ Par(n) \wedge [\forall m (Par(m) \vee \neg Igual(n, suc(m)))] \}$

$Par(SK) \wedge [\forall m (Par(m) \vee \neg Igual(SK, suc(m)))]$

$Par(SK) \wedge (Par(m) \vee \neg Igual(SK, suc(m)))$

$(Par(SK) \wedge Par(m)) \vee (Par(SK) \wedge \neg Igual(SK, suc(m)))$

$Par(SK) \quad Par(m)$

$Par(SK) \quad \neg Igual(SK, suc(m))$

$Par(SK) \wedge (Par(SK) \vee \neg Igual(SK, suc(m))) \wedge$

$\wedge (Par(m) \vee Par(SK)) \wedge (Par(m) \vee \neg Igual(SK, suc(m)))$

talent won't win the game

after hours

after hours

majority of people

holy shit

stay ready

stay ready

hustle

hustle

stay ready

[10] $\frac{n = SK}{}$ $\neg Par(suc(SK))$ $\boxed{\vee}$ $Res(SK)$

[6]

[12] $\frac{}{Res\ Par(n)}$ $\neg Par(suc(n))$ $\boxed{\vee}$ $Par(SK)$ $\boxed{\vee}$ $Res(n)$

[6]

$Par(SK)$ $\boxed{\vee}$ $Par(n)$ $\boxed{\vee}$ $Res(n)$

[14]

[5] $\frac{}{Res}$

[14] $\neg Par(suc(n))$

$\neg \exists n \{ Par(n) \wedge (\neg \exists m (\neg Par(m) \wedge Igual(n, suc(m)))) \}$

$\forall n [\neg Par(n) \vee \underbrace{\neg\neg \exists m (\neg Par(m) \wedge Igual(n, suc(m)))}_{}$

$\neg(\forall m (Par(m) \vee \neg Igual(n, suc(m))))$

$\exists m \neg$

$\forall x, y [(P(x) \wedge P(y) \wedge \neg \exists z\ suc(x, z) \wedge \neg \exists w(y, w)) \Rightarrow$

~~[1] 3~~

[1] $\neg Igual(0, suc(m))$

[2] $Igual(n, 0) \vee Igual(n, suc(f(n)))$

[3] $\neg Par(suc(0))$

[4] $Igual(1, suc(0))$

[5] $Par(suc(n)) \vee Par(n)$

[6] $\neg Par(suc(n)) \vee \neg Par(n)$

[7] $Igual(sum(n, 0), n)$

[8] $Igual(suc(sum(n, m)), sum(n, suc(m)))$

[9] $\neg Igual(n, m) \vee \neg Par(m) \vee Par(n)$

$\neg \exists n \{ Par(n) \wedge [\neg \exists m \; (\neg Par(m) \wedge Igual(n, suc(m)))$

$\cancel{\forall n \; \neg Par(n)} \vee \neg\neg \exists m \; (\neg Par(m) \wedge Igual(n, suc(m)))$

$\cancel{\forall n \; \neg Par(n)} \vee \cancel{\neg Par(SK \; \neg Par}$

$\neg \exists n \{ Par(n) \wedge (\forall m \; (Par(m) \vee \neg Igual(n, suc(m))))] \}$

$\forall n \; \neg Par(n) \vee [\exists m \; (\neg Par(m) \wedge Igual(n, suc(m)))]$

$\forall n \; \neg Par(n) \vee (\neg Par(f(n)) \wedge Igual(n, suc(f(n))))$

$\neg Par(n) \vee \neg Par(f(n)) \vee Res(n)$ [10]

$\neg Par(n) \vee Igual(n, suc(f(n))) \vee Res(n)$ [11]

[5] [10] $Par(suc(n)) \vee \neg Par(f(n)) \vee Res(n)$ [12]

[5] [11] $Par(suc(n)) \vee Igual(n, suc(f(n))) \vee Res(n)$ [13]

$n := 0$

[12] [3] $\neg Par(f(0)) \vee Res(0)$ [14]

[13] [3] $Igual(0, suc(f(0))) \vee Res(0)$ [15]

[15] [1] $m := m$

$n := f(0)$

$\cancel{Par(suc(f(0))) \vee Res(0)}$ [16]

[15]?

?

$\cancel{Par(n) \vee Res(0)}$ [17]

$\cancel{\neg Par(suc(n)) \vee Res(0)}$

$\cancel{Par}$

$n := suc(0)$

[17] [3] $Res(0)$

1. FNC

a) I) El cero no es sucesor de ningún número natural

FNC $\forall m \neg Iqual(m, suc(0))$ $\qquad \forall m \neg Iqual(suc(m), 0)$

$\forall m\ Iqual(suc(sum(n,m)), sum(n, suc(m)))$ VII)

axiomas

II) $\forall n \neg Iqual(n, 0) \Rightarrow \exists m\ Iqual(n, suc(m))$

III) $\neg Par(1)$

$Iqual(1, suc(0))$ (opcional?)

IV) $\forall n \neg Par(suc(n)) \Rightarrow Par(n)$

V) $\forall n\ Par(suc(n)) \Rightarrow \neg Par(n)$

VI) $\forall n\ Iqual(sum(n,0), n)$

FNC

$\forall m \neg Iqual(0, suc(m))$

Eliminación de los cuantificadores existenciales:

$\neg Iqual$ [1]

$\forall n,m\ [(Iqual(n,m) \wedge Par(m)) \Rightarrow Par(n)]$

$\forall n,m\ [(\neg Iqual(n,m) \vee \neg Par(m)) \vee Par(n)]$

$\neg Iqual(n,m) \vee \neg Par(m) \vee Par(n)$

~~x, y: personas~~ $x_1, x_2, \dots$ personas

Groucho: una persona concreta (constante)

$c_1, c_2, \dots$ clubs

SerComo (x_1, x_2) : evalúa T si x_1 es como x_2

SerMiembro (x_1, c_1) : x_1 es miembro de c_1

QuererSerMiembro (x_1, c_1) : x_1 quiere ser miembro de c_1

Aceptar (c_1, x_1) : c_1 ha aceptado x_1 // c_1 acepta x_1

2.I) $\forall x \, [\, SerComo(x,x) \,]$ [1]

2.II) $\forall x,c \, [\, SerMiembro(x,c) \Longleftrightarrow (QuererSerMiembro(x,c) \wedge Aceptar(c,x)) \,]$ [2]

2.III) Hecho en la definición de las constantes

2.IV) $\forall c,x \, [\, (SerComo(\cancel{x}, Groucho) \wedge Aceptar(c,x)) \Longrightarrow \neg QuererSerMiembro(Groucho, c) \,]$ [3]

META: $\forall c$ ~~$\neg$ QuiereSerMiembro (Groucho, c)~~ $\neg$ SerMiembro(Groucho, c)

NEG. META: $\exists c$ ~~Quiere~~ SerMiembro (Groucho, c)

FNC: ~~QuiereSerMiembro (Groucho, SKc)~~ [4] ~~Quiere~~ SerMiembro(Gr, SKc)

Pasamos a FNC la base de conocimiento:

I) $\boxed{SerComo(x,x) \quad [1]}$

II) $\forall x,c \, [\, (SerMiembro(x,c) \wedge QuererSerMiembro(x,c) \wedge Aceptar(c,x)) \vee$
$\vee (\neg SerMiembro(x,c) \wedge (\neg QuererSerMiembro(x,c) \vee \neg Aceptar(c,x))) \,]$

$\forall x,c \, [\, (SM(x,c) \wedge QSM(x,c) \wedge A(c,x)) \vee (\neg SM(x,c) \wedge \neg QSM(x,c)) \vee (\neg SM(x,c) \vee$
$\vee \neg A(c,x)) \,]$

$\forall x,c \left[SM(x,c) \Longleftrightarrow (QSM(x,c) \wedge A(c,x)) \right]$ (SerComo(x,x) ...

$\forall x,c \{ [SM(x,c) \Rightarrow (QSM(x,c) \wedge A(c,x))] \vee [(QSM(x,c) \wedge A(c,x)) \Rightarrow SM(x,c)] \}$

$\forall x,c \{ [\neg SM(x,c) \vee (QSM(x,c) \wedge A(c,x))] \vee [\neg QSM(x,c) \vee \neg A(c,x) \vee SM(x,c)] \}$

$[\neg SM(x,c) \vee (QSM(x,c) \wedge A(c,x))] \vee \neg QSM(x,c) \vee \neg A(c,x) \vee SM(x,c)]$

Entonces: $\forall c \ \neg SM$

$\cancel{\neg SM(x,c)} \vee QSM(x,c) \vee \neg QSM(x,c) \vee \neg A(c,x) \vee \cancel{SM(x,c)}$

$\neg SM(x,c) \vee A(c,x) \vee \neg QSM(x,c) \vee \neg A(c,x) \vee SM(x,c)$

$\forall x,c \left[SM(x,c) \Longleftrightarrow (QSM(x,c) \wedge A(c,x)) \right]$

~~1+ (2.3) =~~

~~2+ (3.4) =~~

~~= 2+3 2+4~~

$\forall x,c \{ [SM \Rightarrow (QSM \wedge A)] \wedge [(QSM \wedge A) \Rightarrow SM] \}$

$\forall x,c \{ [\neg SM \vee (QSM \wedge A)] \wedge [\neg QSM \vee \neg A \vee SM] \}$

$(\neg SM \vee QSM) \wedge (\neg SM \vee A) \wedge (\neg QSM \vee \neg A \vee SM)$

Entonces:

$\boxed{\neg SerMiembro(x,c) \vee QuererSerMiembro(x,c) \quad [2]}$

$\boxed{\neg SerMiembro(x,c) \vee Aceptar(c,x) \quad [3]}$

$\boxed{\neg QuererSerMiembro(x,c) \vee \neg Aceptar(c,x) \vee SerMiembro(x,c) \quad [4]}$

$\forall x,c \left[(SC(x,G) \wedge A(c,x)) \Rightarrow \neg QSM(G,c) \right]$

$\forall x,c \left[\neg SC(x,G) \vee \neg A(c,x) \vee \neg QSM(G,c) \right]$

$\boxed{\neg SerComo(x, Groucho) \vee \neg Aceptar(c,x) \vee \neg QuererSerMiembro(Groucho, c)}$ [5

+ Meta negada:

~~QuererSerMiembro~~(Groucho, SKc) [6]   SerMiembro

[6] [5] | c := SKc / RES_{QSM} | $\neg$SerComo(x, Groucho) $\vee$ $\neg$Aceptar(SKc, x) [7]

[6] [4] | x := Groucho, c := SKc / RES_{QSM} | $\neg$Aceptar(SKc, Groucho) $\vee$ SerMiembro(Groucho, SKc) [8]

~~[8] [3] x := Groucho c := SKc RES Aceptar $\neg$SerMi~~

7] 3] | c := SKc / RE_A | $\neg$SerComo(x, Groucho) $\vee$ $\neg$SerMiembro(x, SKc) [9]

9] 1] | x: Groucho | $\neg$SerMiembro(Groucho, SKc) [10]

1] [7] | x := Groucho | $\neg$Aceptar(SK, Groucho) [11]

Δ :

[1] $SerComo(x, x)$

[2] $\neg SerMiembro(x, c) \lor QuererSerMiembro(x, c)$

[3] $\neg SerMiembro(x, c) \lor Aceptar(c, x)$

[4] $\neg QSM(x, c) \lor \neg Aceptar(c, x) \lor SerMiembro(x, c)$

[5] $\neg SerComo(x, G) \lor \neg Aceptar(c, x) \lor \neg QSM(Groucho, c)$

[6] $SerMiembro(G, SK_c)$

---

[6], [3] $\left| \dfrac{x := Groucho,\ c := SK_c}{RES_{sm}} \right.$ $Aceptar(SK_c, G)$ [7]

[6], [2] $\left| \dfrac{x := G,\ c := SK_c}{RES_{SM}} \right.$ $QSM(G, SK_c)$ [8]

[8], [4] $\left| \dfrac{x := G,\ c := SK_c}{RES_{QSM}} \right.$ $\neg A(SK_c, G) \lor SM(x, c)$ [9]

[9], [7] $\left| \dfrac{x := G,\ c := SK_c}{RES_{A}} \right.$ $SM(G, SK_c)$ [10] (inutil) $\rightarrow$ ya la tenía

[1], [5] $\left| \dfrac{x := G}{serComo} \right.$ $\neg A(c, G) \lor \neg QSM(G, c)$ [11]

[11], [8] $\left| \dfrac{c := SK_c}{QSM} \right.$ $\cancel{\neg A(SK_c, G)}$ $\neg A(SK_c, G)$ [12]

[12], [7] $\left| \dfrac{}{RES_{A}} \right.$ $\square$ $\Longrightarrow$ demostrado $\Longrightarrow$ Groucho es alguien que no e miembro de ningún club.

3

constantes : 0

X: números reales

Igual(x,y) : evalua T si x = y

~~Resta (x,y) :~~ e Neutro(X) : evalua T si X es el elem. neutro de la suma

Resta (x,y) : evalua al número (x-y)

I) ~~$\exists$ Resta (x,y)~~

$$\forall x,y \left[Igual(Resta(x,y), 0) \iff Igual(x,y) \right]$$

II) $$\forall x \left[Neutro(x) \iff \left(\forall y \; Igual(Resta(y,x), y) \right) \right]$$

META: $\forall x \left[Neutro(x) \Rightarrow Igual(x,0) \right]$

NEG META : $\exists x \left[\neg \left(Neutro(x) \Rightarrow Igual(x,0) \right) \right]$

FNC META : $\exists x \left[\neg \left(\neg Neutro(x) \vee Igual(x,0) \right) \right]$

$\exists x \left(Neutro(x) \wedge \neg Igual(x,0) \right)$

$Neutro(SK) \wedge \neg Igual(SK,0)$

I) FNC

$$\forall x,y \left[\left(Igual(Reste(x,y),0) \Rightarrow Igual(x,y) \right) \wedge \left(Igual(x,y) \Rightarrow Igual(Resta(x,y),0) \right) \right]$$

$$\forall x,y \left\{ \left[\neg Igual(Resta(x,y),0) \vee Igual(x,y) \right] \wedge \left[\neg Igual(x,y) \vee Igual(Resta(x,y),0) \right] \right\}$$

$\boxed{\neg Igual(Resta(x,y),0) \vee Igual(x,y) \quad [1]}$

$\boxed{\neg Igual(x,y) \vee Igual(Resta(x,y),0) \quad [2]}$

$$\forall x \left\{ \left[Neutro(x) \Rightarrow (\forall y\ Igual(Resta(x,y),y)) \right] \wedge \left[(\forall y\ Igual(Resta(x,y),y)) \Rightarrow \right. \right.$$
$$\left. \left. \Rightarrow Neutro(x) \right] \right\}$$

$$\forall x \left\{ \left[\neg Neutro(x) \vee (\forall y\ Igual(Resta(x,y),y)) \right] \wedge \left[(\neg \forall y\ Igual(Resta(x,y),y)) \vee \right. \right.$$
$$\left. \left. \vee Neutro(x) \right] \right\} \ \ldots$$

"Un nº es neutro sii al restarlo de cualquier otro x da x".

Pregunta: ¿Son equivalentes?

$$\forall x \left[Neutro(x) \Leftrightarrow (\forall y\ Igual(Resta(y,x),y)) \right]$$

$$\equiv$$

$$\forall x,y \left[Neutro(x) \Leftrightarrow Igual(Resta(y,x),y) \right]$$

$$\forall x,y \left[(Neutro(x) \Rightarrow Igual(Resta(y,x),y)) \wedge (Igual(Resta(y,x),y) \Rightarrow Neutro(x)) \right]$$

$$\forall x,y \left\{ \left[\neg Neutro(x) \vee Igual(Resta(y,x),y) \right] \wedge \left[\neg Igual(Resta(y,x),y) \vee Neutro(x) \right] \right\}$$

$$\boxed{\neg Neutro(x) \vee Igual(Resta(y,x),y) \quad [3]}$$

$$\boxed{\neg Igual(Reste(y,x),y) \vee Neutro(x) \quad [4]}$$

+ Meta negada: Resumen:

[1] $\neg Igual(Resta(x,y),0) \vee Igual(x,y)$

[2] $Igual(Resta(x,y),0) \vee \neg Igual(x,y)$

[3] $\neg Neutro(x) \vee Igual(Resta(\cancel{x}y,x),y)$

[4] $\neg Igual(Resta(y,x),y) \vee Neutro(x)$

[5] $Neutro(sk)$

[6] $\neg Igual(sk,0)$

LA INTELIGENCIA ARTIFICIAL COMO ~~HERREMA~~ HERRAMIENTA PARA REDUCIR LA POBREZA

~~Un tema interesante de debate pueden ser los beneficios que puede otorgar la inteligencia artificial~~

Un tema interesante de debate puede ser qué beneficios puede la inteligencia artificial proporcionar a la lucha contra la pobreza en el mundo.

En primer lugar, con la ayuda de la inteligencia artificial ~~podemos~~ se pueden identificar las regiones con mayor necesidad de intervención

---

find(Ls, E, Rs)

lista

~~find aux~~

~~first_occur([1,2,3,4], 2, 2~~

first_occur([10, 20, 30, 40], 20, 2, [30, 40])

list elem pos rest after element

find([E|Rest], E, [N|M])
LIST

→ check_first_occur(LIST, E, N, Rest)

and → find(Rest, E, M)

caso find(L, E, []) :- ~~check(L, E)~~ not member(E, L)

_Reverse PROC NEAR
pop bp
mov bp, sp

mov ax, ~~bp~~ [bp+6]
mov bx, [bp+8]
mov cx, 0
mov dx
push

---

check_first_occur (LIST, Elem, N, Rest)
→ Indexof (List, Elem, N) ✓
and
→ get_rest (LIST, Elem, Rest)

check_first_occur ([Elem|R], Elem, N, ~~Rest~~ R)
→ IndexOf([Elem|R], Elem, N)
~~and~~
→

check_first_occur (~~[E|R]~~ All, Elem, N, ~~Rest~~ [E|R]) :-
~~Rest All~~
check_first_occur ([

$$f\left(\lambda\begin{pmatrix}x\\y\\z\end{pmatrix}\right) = \lambda\begin{pmatrix}x\\y\\z\end{pmatrix}$$

$f(s) = s$

Vamos a ver como ~~que~~ son afectadas la curvatura de gauss por las homotecias

---

```
seleccionar(Ls, [I|Is] , [R|Rs])
    check (Ls, I, R) ,
    seleccionar (Ls, Is, Rs).

seleccionar(Ls, [], []).
```

~~check([I|_] , 1, I)~~

```
check([Elem|_], 1, Elem).
check([_|Tail], I , Elem) :-
    newE is I-1 ,
    check(Tail, newE, Elem).
```

```
replicar(Ls, N, Rs)
    check(Ls , Rs), Rest
    NewN is N-1 ,
    replicar(Ls, NewN, Rest).
```

comprobamos si Ls aparece al principio de Rs y "devolvemos" el resto de Rs en Rest

caso base

```
replicar(Ls, 0, []).
```

¿ check(Ls, Rs, Rest) ?

quizá :
replicar(Ls, 0, []) :- !.
para evitar bucles infinitos en segundos intentos.

```
check(Ls, Rs, Rest)
```

```
check([L|Ls], [L|Rs], Rs) :-
    check(Ls, Rs, Rs)

check([], _, _).
```

no funciona

mejora

```
check([L|Ls], [L|Rs], Rest) :-
    check(Ls, Rs, Rest).
check([], R, R).
```

[5] | $x := SK$
[5] | RES_{Neutro} $\boxed{Igual(Resta(y, x), y) \quad [7]}$

[1] | $x := SK$, $y := 0$
[6] | $RES_{Igual(x,y)}$ $\boxed{\neg Igual(Resta(SK, 0), 0) \quad [8]}$

~~[2] | x := SK, y := 0~~
~~[8] | RES~~ ~~¬Igual(SK, 0) [9]~~

(*)

[1] | $y := 0$
[7] | RES $\boxed{Igual(x, 0) \quad [9]}$

[6] | $x := SK$
[9] | RES $\square$ Demostrado

DUDA!

$\neg Igual(Resta(x,y), 0) \vee Igua$
$Igual(Resta(x,y), y)$
¿con unificación $y :=$
y entonces
$\neg$ ~~Igual(Resta(x,0),0)~~
$Igual(x, 0)$

~~(LS, A)~~, [2, 3, 5, 1],

LS

~~Para ord~~

minimo (Ls, A, Ys)

→ (*) list_min(Ls, A) : devuelve T sii A es el mínimo de Ls

→ select (Elem, L1, L2) : T sii (L1 sin Elem) == (L2).

Prolog → delete_one

(*) list_min([F|R], A) :- list_min(R, F, A).

list_min([], A, A).

list_min ([F|R], Min0, A)

Min-aux is min(F, Min0),

list_min(R, Min-aux, A)

ordenar (Ls, Rs)

ordenar($\underbrace{[3, 2, 1, 5]}_{L}$, $\underbrace{[1, 2, 3, 5]}_{Rs}$)

minimo (L, M, Rest) , ordenar([3,2,5], Rs))

M = 1, Rest = [3, 2, 5]

[PROLOG] ¿Qué hace exactamente el "=" y el "is"?
Puede ayudar el ejemplo de las diapositivas con:
cosecutivos(N,M) :- N is M+1
consecutivos2(N,M) :- N = M+1

[RES+REF] Cuando intentamos demostrar cierta FBF, la negamos y la introducimos en la base de conocimiento. Si llegamos a la cláusula vacía entonces la meta (sin negar) era consecuencia lógica de la base de conocimiento... pero, si no llegamos a la cláusula vacía, ¿quiere esto decir que NO era $\models$ o que no tenemos evidencia alguna de que fuese o no $\models$? → como los contrastes de hipótesis en estadística

[EJERCICIOS ?] ¿Se permite cierto razonamiento informal?

[EJERCICIOS] → predicado Igual
↳ hay que interpretarlo como una igualdad matemática y por lo tanto ¿podemos sustituir / intercambiar los términos que el predicado dice que son iguals o hay que basar la resolución únicamente en la grafía?

UNIFICACIÓN EN GENERAL] Algoritmo de unificación

USTITUCIÓN] ~~a la izq~~ variable := algo

algo →
- constante
- variable
- evaluación de una función → ¿f(constante)? ¿f(variable)?

diferentes

ERCICIOS] Duda ejercicio 3 marcada en verde

Prolog en general