Conocimiento y Razonamiento Automatizado
Tema 3
Grado en Ingeniería Informática

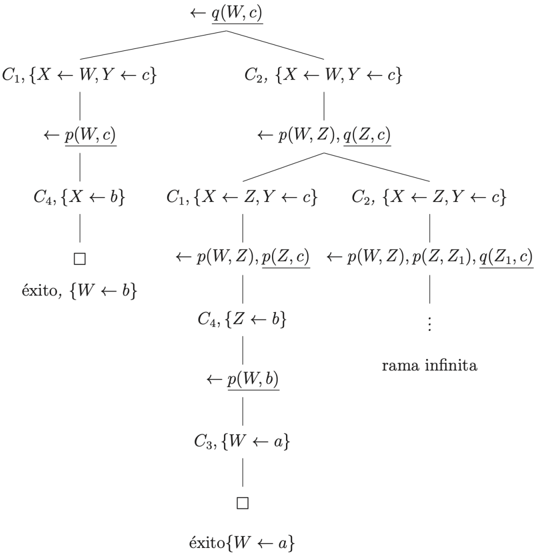
José Enrique Morais San Miguel

25 de marzo de 2021

Universidad de Alcalá

Ejemplo

- $C_1 = q(X,Y) \leftarrow p(X,Y)$
- $C_2 = q(X,Y) \leftarrow p(X,Z), q(Z,Y)$
- $C_3 = p(a,b)$
- $C_4 = p(b,c)$



Nota

El Teorema sobre la completitud de la resolución SLD establece que la resolución SLD es completa independientemente de la regla de computación, pero el Teorema sólo dice que una tal refutación existe. La elección de la regla de búsqueda en el árbol determinará si la refutación se encuentra o no. El uso de la búsqueda en profundidad garantiza que siempre se encuentra una rama de éxito si existe. Por otro lado, la búsqueda en profundidad puede hacer que se escoja una derivación infinita, por lo que podríamos no encontrar la refutación existente (si elegimos en el segundo caso del ejemplo anterior como regla de ordenación la de elegir el que esté más abajo en la lista, nos encontraríamos en esa situación). Sin embargo, la información del árbol que se debe almacenar crece exponencialmente con la profundidad, por lo que es impracticable. Es esta la razón por la Prolog se utiliza búsqueda en profundidad con backtracking.

Ejemplo

Sea P el programa lógico dado por:

$$
\begin{aligned}
&C_1 = p(X,Y) \leftarrow \; q(Z,X), r(Z,V), q(V,Y).\\
&C_2 = p(X,Y) \leftarrow \; q(Z,Y), r(Z,V), q(V,X).\\
&C_3 = q(a,c). \quad C_4 = q(a,d). \quad C_5 = r(a,b).\\
&C_6 = r(b,a). \quad C_7 = q(b,e).
\end{aligned}
$$

Computar las respuestas correctas para la pregunta $\leftarrow p(c, Y)$ *usando resolución SLD a la Prolog.*

Resolución SLD

$C_1 = p(X,Y) \leftarrow q(Z,X), r(Z,V), q(V,Y).$
$C_2 = p(X,Y) \leftarrow q(Z,Y), r(Z,V), q(V,X).$
$C_3 = q(a,c).\quad C_4 = q(a,d).\quad C_5 = r(a,b).$
$C_6 = r(b,a).\quad C_7 = q(b,e).$

$$\leftarrow \underline{p(c,W)}$$

$$C_1 \quad \{X \leftarrow c, Y \leftarrow W\}$$

$$\leftarrow \underline{q(Z,c)}, r(Z,V), q(V,W)$$

$$C_3 \quad \{Z \leftarrow a\}$$

$$\leftarrow \underline{r(a,V)}, q(V,W)$$

$$C_5 \quad \{V \leftarrow b\}$$

$$\leftarrow \underline{q(b,W)}$$

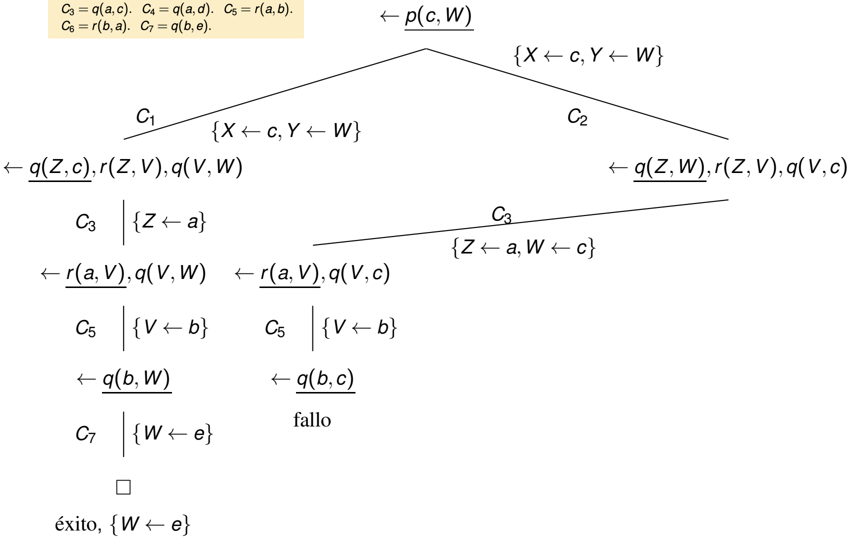
$$C_7 \quad \{W \leftarrow e\}$$

$$\square$$

éxito, $\{W \leftarrow e\}$

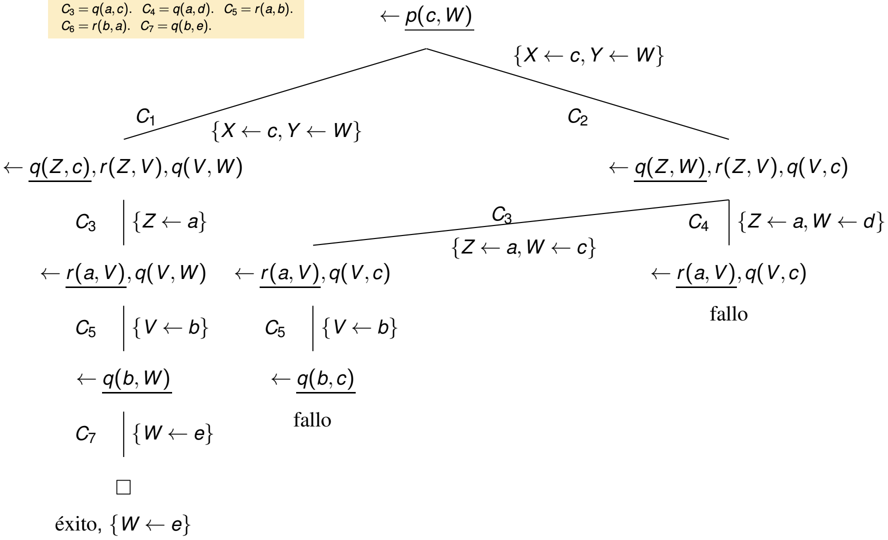
Resolución SLD

$C_1 = p(X,Y) \leftarrow q(Z,X), r(Z,V), q(V,Y).$
$C_2 = p(X,Y) \leftarrow q(Z,Y), r(Z,V), q(V,X).$
$C_3 = q(a,c).\quad C_4 = q(a,d).\quad C_5 = r(a,b).$
$C_6 = r(b,a).\quad C_7 = q(b,e).$



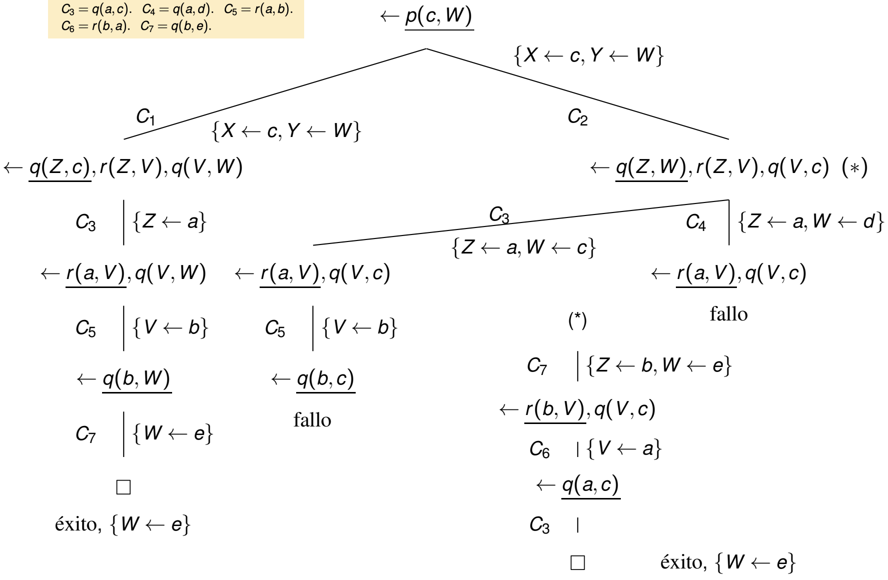
Resolución SLD

$C_1 = p(X,Y) \leftarrow q(Z,X), r(Z,V), q(V,Y).$
$C_2 = p(X,Y) \leftarrow q(Z,Y), r(Z,V), q(V,X).$
$C_3 = q(a,c).$ $C_4 = q(a,d).$ $C_5 = r(a,b).$
$C_6 = r(b,a).$ $C_7 = q(b,e).$



Resolución SLD

$C_1 = p(X,Y) \leftarrow q(Z,X), r(Z,V), q(V,Y).$
$C_2 = p(X,Y) \leftarrow q(Z,Y), r(Z,V), q(V,X).$
$C_3 = q(a,c).\quad C_4 = q(a,d).\quad C_5 = r(a,b).$
$C_6 = r(b,a).\quad C_7 = q(b,e).$



Ejemplo

Las listas se pueden codificar, empleando el Cálculo de Predicados, mediante una función . de aridad 2, y una constante nil, la lista vacía, de manera que: una lista con un elemento se representa mediante $.(a, nil)$*, con dos elementos* $.(a, .(b, nil)), \ldots$ *y así sucesivamente. Definir un procedimiento lógico append de aridad 3 que tome valor cierto si el tercer elemento es la concatenación de los dos primeros. Una vez definido, calcular, usando resolución SLD a la Prolog, todas las respuestas correctas a:*

1. $\leftarrow append(.(a, .(b, nil)), .(c, nil), Z)$
2. $\leftarrow append(Y, Z, .(a, .(b, .(c, nil))))$

Ejemplo: *append*

$C_1 : app(.(X, L_1), L_2, .(X, L_3)) \leftarrow app(L_1, L_2, L_3)$

$C_2 : app(nil, L, L)$

$C_1' : app(.(X, L_1), L_2, .(X, L_3')) \leftarrow app(L_1, L_2, L_3')$

$\leftarrow \underline{app(.(a, .(b, nil)), .(c, nil), Z)}$

$C_1 \quad \{X \leftarrow a, L_1 \leftarrow .(b, nil), L_2 \leftarrow .(c, nil), Z \leftarrow .(a, L_3)\}$

$\underline{\leftarrow app(.(b, nil), .(c, nil), L_3)}$

$C_1' \quad \{X \leftarrow b, L_1 \leftarrow nil, L_2 \leftarrow .(c, nil), L_3 \leftarrow .(b, L_3')\}$

$\underline{\leftarrow app(nil, .(c, nil), L_3')}$

$C_2 \quad \{L \leftarrow .(c, nil), L_3' \leftarrow .(c, nil)\}$

$\square$

éxito, $\{Z \leftarrow .(a, .(b, .(c, nil)))\}$

Resolución SLD

$C_1: append(.(X, L_1), L_2, .(X, L_3)) \longleftarrow append(L_1, L_2, L_3)$

$C_2: append(nil, L, L)$ $\quad C_1': append(.(X, L_1'), L_2, .(X, L_3')) \leftarrow app(L_1', L_2, L_3')$

$C_1'': \quad L_1'' \quad L_1''$

$\leftarrow append(Y, Z, .(a, .(b, .(c, nil))))$

$C_1 \quad \{Y \leftarrow .(a, L_1), L_2 \leftarrow Z, L_3 \leftarrow .(b, .(c, nil)), X \leftarrow a\}$

$\leftarrow append(L_1, Z, .(b, .(c, nil)))$

$C_1' \quad \{X \leftarrow b, L_1 \leftarrow .(b, L_1'), L_2 \leftarrow Z, L_3 \leftarrow .(c, nil)\}$

$\leftarrow append(L_1', Z, .(c, nil))$

$C_1'' \quad \{X \leftarrow c, L_3 \leftarrow nil, L_2 \leftarrow Z, L_1' \leftarrow .(c, L_1'')\}$

$\leftarrow append(L_1'', Z, nil)$

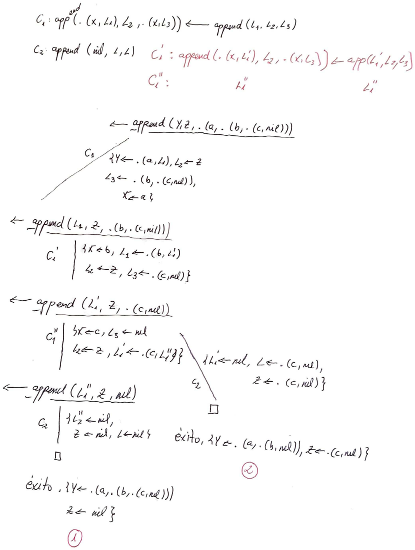
$C_2 \quad \{L_2'' \leftarrow nil, Z \leftarrow nil, L \leftarrow nil\}$

$\square$

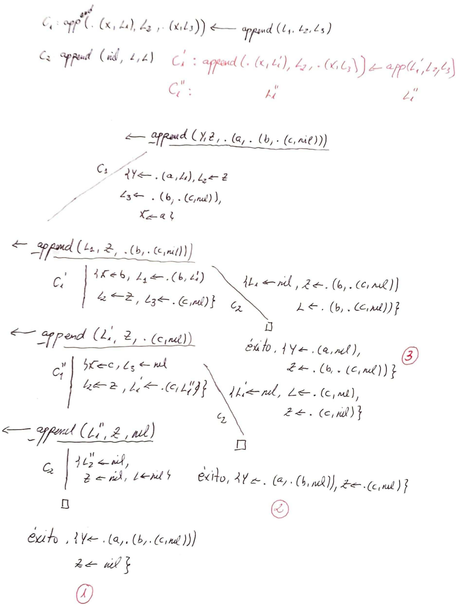
éxito, $\{Y \leftarrow .(a, .(b, .(c, nil))), Z \leftarrow nil\}$

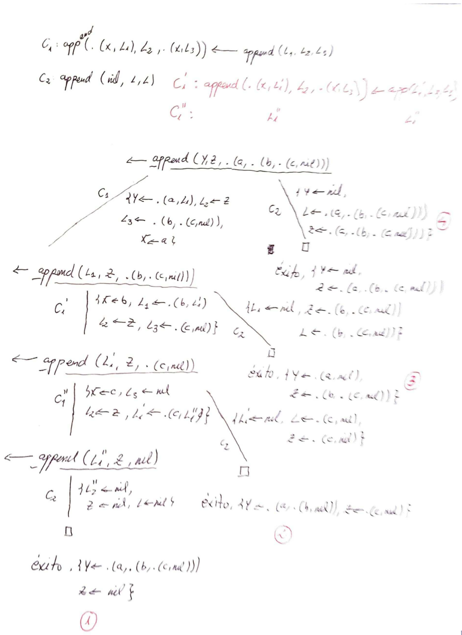
1

Resolución SLD



Resolución SLD




C1 : append(.(X, L1), L2, .(X,L3)) ← append(L1, L2, L3)
C2 append (nil, L, L)
C1' : append(.(X, L1'), L2, .(X,L3)) ← app(L1', L2, L3)
C1'' : L1'' L1''
← append(Y, Z, .(a, .(b, .(c, nil))))
C1
{Y ← .(a,L1), L2 ← Z
L3 ← .(b, .(c,nil)),
X ← a}
C2
{Y ← nil,
L ← .(a, .(b, .(c, nil))),
Z ← .(a, .(b, .(c nil)))}
□
éxito, {Y ← nil,
Z ← .(a, .(b, .(c, nil)))}
← append(L1, Z, .(b, .(c,nil)))
C1'
{X ← b, L1 ← .(b, L1')
L2 ← Z, L3 ← .(c,nil)}
{L1 ← nil, Z ← .(b, .(c, nil))
L ← .(b, .(c, nil))}
C2
□
éxito, {Y ← .(a, nil),
Z ← .(b, .(c, nil))}
← append (L1', Z, .(c,nil))
C1''
{X ← c, L3 ← nil
L2 ← Z, L1' ← .(c, L1'')}
{L1' ← nil, L ← .(c, nil),
Z ← .(c, nil)}
C2
□
← append(L1'', Z, nil)
C2
{L1'' ← nil,
Z ← nil, L ← nil}
□
éxito, {Y ← .(a, .(b, nil)), Z ← .(c, nil)}
éxito, {Y ← .(a, .(b, .(c, nil)))
Z ← nil}
1
2
3
4

Ejercicio del examen de 12 de junio de 2013

Las listas se pueden codificar, empleando el Cálculo de Predicados, mediante una función . de aridad 2, y una constante *nil*, la lista vacía, de manera que: una lista con un elemento se representa mediante $.(a, nil)$, con dos elementos $.(a, .(b, nil))$, ... y así sucesivamente. Definir un procedimiento lógico reverse de aridad 2 que tome valor cierto si el segundo elemento es la reflexión del primero.
Llevar a cabo resolución SLD a la Prolog, para obtener una respuesta correcta a la pregunta:

$$\leftarrow reverse(.(a, .(b, .(c, nil))), L)$$

Ejercicio del examen de 12 de junio de 2013

C_1: $reverse(L, L_1) \leftarrow reverse\_aux(L, nil, L_1)$

C_2: $reverse\_aux(.(X, L), Z, L_1) \leftarrow reverse\_aux(L, .(X, Z), L_1)$

C_3: $reverse\_aux(nil, L, L)$

Ejercicio del examen de 12 de junio de 2013

C_1 : $reverse(L, L_1) \leftarrow reverse\_aux(L, nil, L_1)$
C_2 : $reverse\_aux(.(X, L), Z, L_1) \leftarrow$
$\quad reverse\_aux(L, .(X, Z), L_1)$
C_3 : $reverse\_aux(nil, L, L)$

$$\leftarrow \underline{reverse(.(a, .(b, nil)), L')}$$

$$C_1 \;\Big|\; \{L \leftarrow .(a, (b.nil)), L_1 \leftarrow L'\}$$

$$\leftarrow \underline{reverse\_aux(.(a, .(b, nil)), nil, L')}$$

$$C_2 \;\Big|\; \{X \leftarrow a, L \leftarrow .(b.nil), Z \leftarrow nil, L_1 \leftarrow L'\}$$

$$\leftarrow \underline{reverse\_aux(.(b, nil), .(a, nil), L')}$$

$$C_2 \;\Big|\; \{X \leftarrow b, L \leftarrow nil, Z \leftarrow .(a, nil), L_1 \leftarrow L'\}$$

$$\leftarrow \underline{reverse\_aux(nil, .(b, (a, nil)), L')}$$

$$C_3 \;\Big|\; \{L \leftarrow .(b, .(a, nil)), L' \leftarrow .(b, .(a, nil))\}$$

$$\square$$

éxito, $\{L' \leftarrow .(b, .(a, nil))\}$

Ejercicio del examen de 12 de junio de 2013

C_1: $reverse(nil, nil)$

C_2: $reverse(.(X, L), L_1) \leftarrow reverse(L, L_2), append(L_2, .(X, nil), L_1)$

C_3: $append(nil, L, L)$

C_4: $append(.(X, L), L_1, .(X, L_2) \leftarrow append(L, L_1, L_2)$

Ejercicio del examen de 12 de junio de 2013

C_1 : $reverse(nil, nil)$

C_2 : $reverse(.(X, L), L_1) \leftarrow$
$reverse(L, L_2), append(L_2, .(X, nil), L_1)$

C_3 : $append(nil, L, L)$

C_4 : $append(.(X, L), L_1, .(X, L_2)) \leftarrow$
$append(L, L_1, L_2)$

$$\leftarrow \underline{reverse(.(a, .(b, nil)), Z)}$$

Ejercicio del examen de 12 de junio de 2013

C_1 : $reverse(nil, nil)$

C_2 : $reverse(.(X, L), L_1) \leftarrow$

$reverse(L, L_2), append(L_2, .(X, nil), L_1)$

C_3 : $append(nil, L, L)$

C_4 : $append(.(X, L), L_1, .(X, L_2)) \leftarrow$

$append(L, L_1, L_2)$

$$\leftarrow \underline{reverse(.(a, .(b, nil)), Z)}$$

$$C_2 \quad \Big| \; \{X \leftarrow a, L \leftarrow .(b.nil), L_1 \leftarrow Z\}$$

$$\leftarrow \underline{reverse(.(b, nil), L_2)}, append(L_2, .(a, nil), Z)$$

Ejercicio del examen de 12 de junio de 2013

$C_1 : reverse(nil, nil)$

$C_2 : reverse(.(X, L), L_1) \leftarrow$
$reverse(L, L_2), append(L_2, .(X, nil), L_1)$

$C_2' : reverse(.(X, L), L_1) \leftarrow$
$reverse(L, L_2'), append(L_2', .(X, nil), L_1)$

$C_3 : append(nil, L, L)$

$C_4 : append(.(X, L), L_1, .(X, L_2)) \leftarrow$
$append(L, L_1, L_2)$

$$\leftarrow \underline{reverse(.(a, .(b, nil)), Z)}$$

$$C_2 \;\Big|\; \{X \leftarrow a, L \leftarrow .(b.nil), L_1 \leftarrow Z\}$$

$$\leftarrow \underline{reverse(.(b, nil), L_2)}, append(L_2, .(a, nil), Z)$$

$$C_2' \;\Big|\; \{X \leftarrow b, L \leftarrow nil, Z \leftarrow nil, L_1 \leftarrow L_2\}$$

$$\leftarrow \underline{reverse(nil, L_2')}, append(L_2', .(b, nil), L_2), append(L_2, .(a, nil), Z)$$

Ejercicio del examen de 12 de junio de 2013

C_1 : $reverse(nil, nil)$

C_2 : $reverse(.(X, L), L_1) \leftarrow reverse(L, L_2), append(L_2, .(X, nil), L_1)$

C_2' : $reverse(.(X, L), L_1) \leftarrow reverse(L, L_2'), append(L_2', .(X, nil), L_1)$

C_3 : $append(nil, L, L)$

C_4 : $append(.(X, L), L_1, .(X, L_2)) \leftarrow append(L, L_1, L_2)$

$$\leftarrow \underline{reverse(.(a, .(b, nil)), Z)}$$

$$C_2 \quad \Big| \quad \{X \leftarrow a, L \leftarrow .(b.nil), L_1 \leftarrow Z\}$$

$$\leftarrow \underline{reverse(.(b, nil), L_2)}, append(L_2, .(a, nil), Z)$$

$$C_2' \quad \Big| \quad \{X \leftarrow b, L \leftarrow nil, Z \leftarrow nil, L_1 \leftarrow L_2\}$$

$$\leftarrow \underline{reverse(nil, L_2')}, append(L_2', .(b, nil), L_2), append(L_2, .(a, nil), Z)$$

$$C_1 \quad \Big| \quad \{L_2' \leftarrow nil\}$$

$$\leftarrow \underline{append(nil, .(b, nil), L_2)}, append(L_2, .(a, nil), Z)$$

$$C_3 \quad \Big| \quad \{L \leftarrow .(b, nil), L_2 \leftarrow .(b, nil), Z\}$$

$$\leftarrow \underline{append(.(b, nil), .(a, nil), Z)}$$

$$\vdots$$

éxito, $\{Z \leftarrow .(b, .(a, nil))\}$

Ejercicio del examen de 21 de mayo de 2015

Escribir un programa lógico que defina un predicado $union(L, L_1, L_2)$ de aridad 3 que tome valor cierto si L_2 es la unión conjuntista de las listas L y L1. Llevar a cabo resolución SLD a la Prolog para obtener una respuesta correcta a la pregunta:

$$\leftarrow union(.(1, .(2, nil)), .(2, .(3, nil)), Z).$$

Ejercicio del examen de 21 de mayo de 2015

C_1: $union(nil, L, L)$

C_2: $union(.(X, L_1), L_2, L_3) \leftarrow pertenece(X, L_2), union(L_1, L_2, L_3)$

C_3: $union(.(X, L_1), L_2, L_3) \leftarrow union(L_1, .(X, L_2), L_3)$

C_4: $pertenece(X, .(X, L))$

C_5: $pertenece(X, .(Y, L)) \leftarrow pertenece(X, L)$

C_1: union (nil, L, L)

C_2: union (E(X, L_1), L_2, L_3) ← pertenece (X, L_2), union (L_1, L_2, L_3)

C_3: union (·(X, L_1), L_2, L_3) ← union (L_1, ·(X, L_2), L_3)

C_4: pertenece (X, ·(X, L))

C_5: pertenece (X, ·(Y, L)) ← pertenece (X, L)

← union (·(1, ·(2, nil)), ·(2, ·(3, nil)), Z)

C_2 {X ← 1, L_1 ← ·(2, nil)
L_2 ← ·(2, ·(3, nil)), L_3 ← Z} (*)

← pertenece (1, ·(2, ·(3, nil))), union (·(2, nil), ·(2, ·(3, nil)), Z)

⋮

fallo

C_3 misma u.m.g. que (*)

← union (·(2, nil), ·(1, ·(2, ·(3, nil))), Z)

C_2 {X ← 2, L_1 ← nil, L_2 ← ·(1, ·(2, ·(3, nil))), L_3 ← Z}

← pertenece (2, ·(1, ·(2, ·(3, nil)))), union (nil, ·(1, ·(2, ·(3, nil))), Z) (a)

C_5 {X ← 2, Y ← 1, L ← ·(2, ·(3, nil))}

← pertenece (2, ·(2, ·(3, nil))), (a)

C_4 {X ← 2, L ← ·(3, nil)}

← union (nil, ·(1, ·(2, ·(3, nil))), Z)

C_1 {L ← ·(1, ·(2, ·(3, nil)))
Z ← ·(1, ·(2, ·(3, nil)))}

□

éxito, {Z ← ·(1, ·(2, ·(3, nil)))}