Las reglas `miembro/2` y `concatenar/3` en Prolog
SP–7767 Procesamiento del Lenguaje Natural: Ejemplos

Jorge Antonio Leoni de León
Programa de Posgrado en Lingüística
Universidad de Costa Rica
San Pedro Montes de Oca, San José
Costa Rica

Fecha de creación: 23 de marzo de 2019
Última modificación: 27 de marzo de 2019
Email: `antonio.leoni@ucr.ac.cr`
URL: `http://www.mulkin.info`

Índice de figuras

Este documento ilustra los procesos de las reglas de membresía (1) y concatenación (3) en Prolog. En el caso de la regla `miembro/2`, es posible preguntar si un elemento `X` forma parte de una lista `Y`. Por ejemplo, en (2), la pregunta planteada es si el átomo "`tyrion`" (primer argumento del predicado `miembro/2`) forma parte de la lista que es el segundo elemento del predicado `miembro/2`, hecho confirmado con la respuesta "`true`".

(1) **Membresía en una lista**

```
1 miembro(X , [X|_]).
2 miembro(X, [_|COLA]) :- miembro(X,COLA).
```

(2) **Ejemplo de respuesta**

```
1 ?- member(tyrion,[tywin, tytos, joffrey,myrcella,tommen,tyrion,lancel]).
2 true
```

La regla `concatenar/3` en (3) fusiona dos listas, (4a) y (4b), en una tercera lista (4c):

(3) **Concatenación**

```
1 concatenar([],X,X).
2 concatenar([X|Y],Z,[X|W]) :-
3         concatenar(Y,Z,W).
```

(4) a. **Lista 1**: `[targaryen,stark,lannister]`

b. **Lista 2**: `[martell,tully,baratheon]`

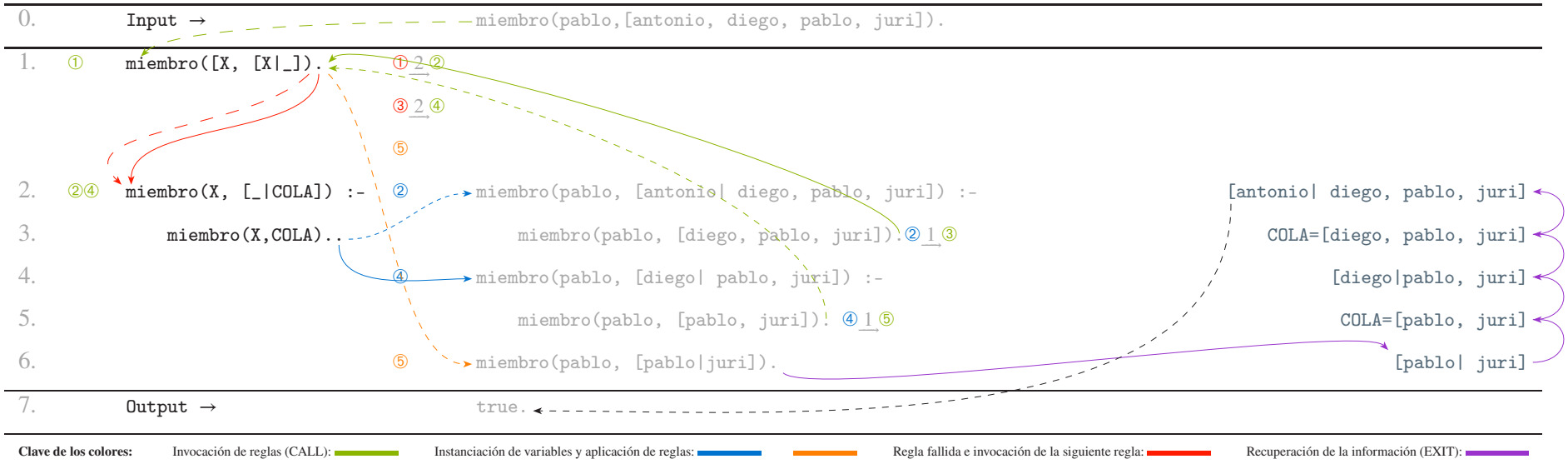
c. **Respuesta de** `concatenar/3`

```
1 ?- concatenar([targaryen,stark,lannister],[martell,tully,baratheon], R).
2 R = [targaryen, stark, lannister, martell, tully, baratheon].
```

Los resultados del proceso de ejecución en el modo de funcionamiento *trace* están presentados en las figuras 1 (página 3) y 3 (página 5); el detalle del procesamiento está ilustrado en las figuras 2 (página 4) y 4 (página 6). La definición de estas reglas ha sido ampliamente comentada en la literatura de Prolog, una buena referencia son los clásicos Clocksin y Mellish (2003, págs. 54-65) y Sterling y Shapiro (1994, págs. 58-60).

```
[trace]  ?- miembro(pablo,[antonio, diego, pablo, juri]).
   Call: (8) miembro(pablo, [antonio, diego, pablo, juri]) ? creep
   Call: (9) miembro(pablo, [diego, pablo, juri]) ? creep
   Call: (10) miembro(pablo, [pablo, juri]) ? creep
   Exit: (10) miembro(pablo, [pablo, juri]) ? creep
   Exit: (9) miembro(pablo, [diego, pablo, juri]) ? creep
   Exit: (8) miembro(pablo, [antonio, diego, pablo, juri]) ? creep
true .
```

Figura 1: Ejecución de `miembro/2` en el modo `trace`

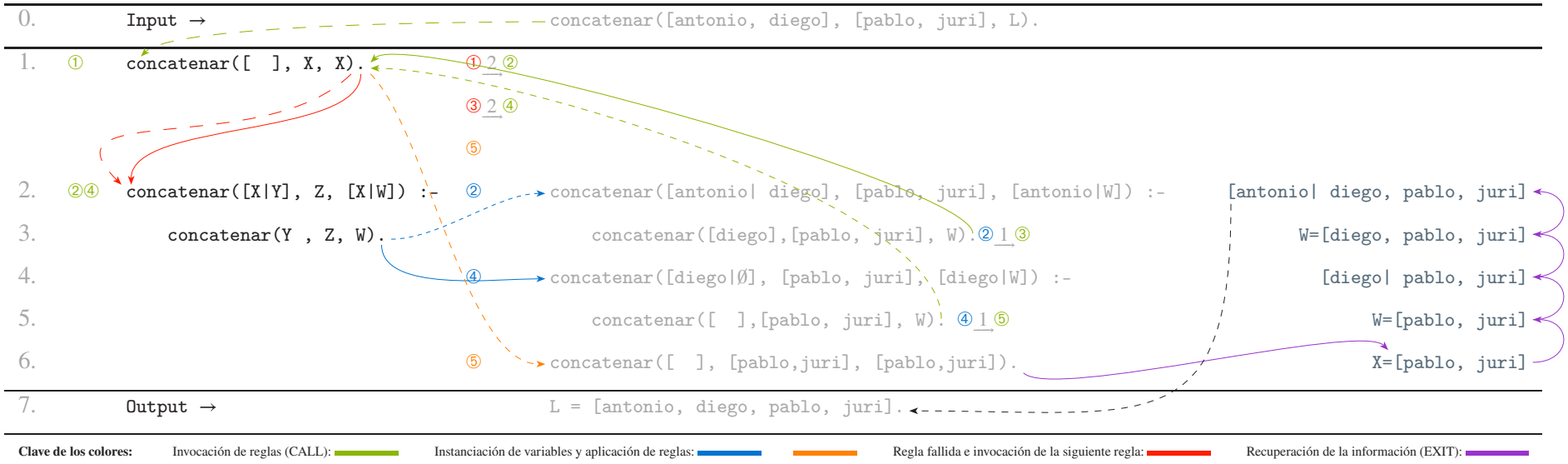


Nota:" 1 " indica el número de línea de la siguiente regla invocada. Por ejemplo, en " 1 " la regla invocada es la de la línea 1. Los números dentro de círculos señalan la secuencia de procesamiento: paso ①, paso ②, etc. Así, "② 1 ③ " indica que del paso ② al paso ③ fue invocada la regla de la línea "1".

Figura 2: **Procesamiento de `miembro/2` en Prolog**

```
[trace]  ?- concatenar([antonio,diego],[pablo,juri],L).
   Call: (8) concatenar([antonio, diego], [pablo, juri], _6766) ? creep
   Call: (9) concatenar([diego], [pablo, juri], _7004) ? creep
   Call: (10) concatenar([], [pablo, juri], _7010) ? creep
   Exit: (10) concatenar([], [pablo, juri], [pablo, juri]) ? creep
   Exit: (9) concatenar([diego], [pablo, juri], [diego, pablo, juri]) ? creep
   Exit: (8) concatenar([antonio, diego], [pablo, juri], [antonio, diego, pablo, juri]) ? creep
L = [antonio, diego, pablo, juri].
```

Figura 3: Ejecución de `concatenar/3` en el modo `trace`



Nota:“ 1 ” indica el número de línea de la siguiente regla invocada. Por ejemplo, en “ 1 ” la regla invocada es la de la línea 1. Los números dentro de círculos señalan la secuencia de procesamiento: paso ①, paso ②, etc. Así, “② 1 ③ ” indica que del paso ② al paso ③ fue invocada la regla de la línea “1”.

Figura 4: **Procesamiento de** `concatenar/3` **en Prolog**

Referencias

Clocksin, William F. y Christopher S. Mellish (2003). *Programming in Prolog*. 5.ª ed. Berlin: Springer. ISBN: 978-3-540-00678-7. DOI: 10.1007/978-3-642-55481-0.

Sterling, Leon y Ehud Shapiro (1994). *The Art of Prolog (2Nd Ed.): Advanced Programming Techniques*. Cambridge, MA, USA: MIT Press. ISBN: 0-262-19338-8.