Programación Lógica[1]

Qué vs. *cómo*: Especificación vs. implementación.

Definición del predicado **ordenada(Y)** (Y es lista de enteros *ordenada*). Sea Y una sucesión finita (*lista*) de enteros Y1,Y2,...,Yn,

$$\texttt{ordenada(Y)} \Leftrightarrow \texttt{Y}_j \leq \texttt{Y}_{j+1} \quad (j = 1, \ldots, n-1)$$

Definición del predicado **ordena(X,Y)** (Y es *permutación ordenada* de X). Sean X y Y listas de enteros,

$$\texttt{ordena(X,Y)} \Leftrightarrow \texttt{permutacion(X,Y)} \wedge \texttt{ordenada(Y)}$$

```
ordena(X,Y) :- permutacion(X,Y),
               ordenada(Y).
ordenada([]).
ordenada([X]).
ordenada([X|[Y|Ys]]) :-
  X<=Y, ordenada([Y|Ys]).

permutacion([],[]).
permutacion(X,[Y|Ys]) :-
  quita(Y,X,Z), permutacion(Z,Ys).

quita(A,[A|As],As).
quita(A,[B|Bs],[B|Cs]) :-
  quita(A,Bs,Cs).
```

```
?- ordena([3,2,7,1],[1, 2, 3, 7]).
*** Si
?- ordena([3,2,7,1],[2,1,3,7]).
*** No
?- ordena([3,2,7,1,5,2,9,4],Y).
Y = [1, 2, 2, 3, 4, 5, 7, 9]
?- permutacion([1,2,3],P).
P = [1, 2, 3]
P = [1, 3, 2]
P = [2, 1, 3]
P = [2, 3, 1]
P = [3, 1, 2]
P = [3, 2, 1]
```

Definición. Una *cláusula de programa* o *cláusula definida* es una implicación de la forma:

$$A \leftarrow B_1, \ldots, B_n$$

en que $n \geq 0$ y A, $B_1, \ldots, B_n$ son fórmulas *atómicas* —predicados como $p(t_1, \ldots, t_m)$. A la conclusión A se le llama *cabeza* y a la lista (conjunción) de premisas se le llama *cuerpo*. Si $n = 0$ se dice que $A \leftarrow$ es una *cláusula unitaria*; si se quita la conclusión, a la fórmula $\leftarrow B_1, \ldots, B_n$ se le llama *meta* o *cláusula negativa*.

Definición. Un *programa lógico* es un conjunto finito no vacío de cláusulas de programa.

[1] Esta sección contiene partes tomadas del libro de C. Ghezzi y M. Jazayeri y de: Krzysztof R. Apt, *Logic Programming*, Handbook of Theoretical Computer Science, vol. B: Formal Models and Semantics, Jan Van Leeuwen, Elsevier Science Publishers B. V., 1990, pp. 493–574.

Sustituciones.

En la programación lógica las variables se ligan con valores mediante sustituciones de una forma especial conocida como *unificadores más generales*.

Definición. Una *sustitución* es una función finita, de variables a términos, que se acostumbra escribir en la forma:

$$\theta = \{\langle v_1, t_1\rangle, \langle v_2, t_2\rangle, \ldots, \langle v_n, t_n\rangle\},$$

conjunto en que $v_1, \ldots, v_n$ denotan variables (distintas, por supuesto), $t_1, \ldots, t_n$ denotan *términos* (variables, constantes y de forma $f(t'_1, \cdots, t'_m)$) y t_i no es v_i $(i = 1, \ldots, n)$. A cada pareja $\langle v_i, t_i\rangle$ se le llama *liga* o *enlace*. La sustitución puede leerse así: "las variables $v_1, \ldots, v_n$ se vuelven (se *ligan* o *enlazan* a) $t_1, \ldots, t_n$, respectivamente". Si θ es biyectiva —$\{t_1, \ldots, t_n\} = \{v_1, \ldots, v_n\}$— se le llama *renombramiento*.

Las sustituciones se aplican a términos y a fórmulas. Si E denota una de estas expresiones, $E\theta$ simboliza la expresión que se obtiene de reemplazar *simultáneamente* en E cada ocurrencia de una variable del domino de θ por el término correspondiente. A $E\theta$ se le llama *instancia* de E.

Si θ es un renombramiento, a $E\theta$ se le llama *variante* de E. Así, $x < y' + z'$ es variante de $x < y + z$: $x < y' + z' \equiv (x < y + z)\{\langle y, y'\rangle, \langle z, z'\rangle, \langle y', y\rangle, \langle z', z\rangle\}$, pero $x < y' + x$ no lo es.

Lema. Para todas las expresiones E y F,

$$E \text{ es variante de } F \quad \Leftrightarrow \quad E \text{ es instancia de } F \quad \text{y} \quad F \text{ es instancia de } E$$

Si F es instancia no variante de E, se dice que E es *mas general* que F. Por ejemplo, $x < y + z$ es mas general que $x < y' + x$ porque sabemos que ésta no es variante de la primera pero $x < y' + x \equiv (x < y + z)\{\langle y, y'\rangle, \langle z, x\rangle\}$.

Definición de composición de sustituciones. Dadas las sustituciones

$$\theta = \{\langle x_1, t_1\rangle, \ldots, \langle x_n, t_n\rangle\} \quad \text{y}$$
$$\eta = \{\langle y_1, s_1\rangle, \ldots, \langle y_m, s_m\rangle\},$$

su *composición* $\theta\eta$ se define removiendo del conjunto

$$\{\langle x_1, t_1\eta\rangle, \ldots, \langle x_n, t_n\eta\rangle, \;\; \langle y_1, s_1\rangle, \ldots, \langle y_m, s_m\rangle\}$$

las parejas $\langle x_i, t_i\eta\rangle$ con $x_i \equiv t_i\eta$ y aquellas $\langle y_i, s_i\rangle$ con $y_i \in \{x_1, \ldots, x_n\}$.

Así, si $\theta = \{\langle x, 3\rangle, \langle y, f(x, 1)\rangle\}$ y $\eta = \{\langle x, 4\rangle\}$, $\theta\eta = \{\langle x, 4\rangle, \langle y, f(4, 1)\rangle\}$.

Lema. Sean θ, η y γ sustituciones y E una expresión. Entonces

$$(E\theta)\eta \equiv E(\theta\eta) \quad \text{y} \quad (\theta\eta)\gamma \equiv \theta(\eta\gamma).$$

Una sustitución θ es *mas general* que la sustitución η si existe alguna sustitución γ tal que $\eta = \theta\gamma$.

Sean E y F dos términos o dos átomos. Se dice que son *unificables* si existe alguna sustitución θ tal que $E\theta \equiv F\theta$; a θ se le llama *unificador* de E y F, además se dice que es un *unificador mas general (*MGU*)* si es mas general que cualquier otro unificador de E y F. Si θ_1 y θ_2 son unificadores mas generales de E y F, se pruba que existe algún renombramiento ρ tal que $\theta_1 = \theta_2\rho$, en este sentido se habla de *el* unificador mas general.

Por ejemplo, $\eta = \{\langle X, a\rangle, \langle Y, c\rangle, \langle Z, b\rangle, \langle W, c\rangle\}$ es un unificador de los términos $f(X, b, Y)$ y $f(a, Z, W)$, la instancia común que se obtiene es $f(a, b, c)$; también unifica a los átomos $p(W, f(X, b, Y), g(a))$ y $p(Y, f(a, Z, W), g(X))$; otro unificador es $\mu = \{\langle X, a\rangle, \langle Y, T\rangle, \langle Z, b\rangle, \langle W, T\rangle\}$, que en el caso de los términos da lugar a la instancia común $f(a, b, T)$; obsérvese que ésta es mas general que la instancia común $f(a, b, c)$ obtenida mediante η; a la vez μ es mas general que η, pues $\eta = \mu\{\langle T, c\rangle\}$; de hecho, μ es el unificador mas general de dichos términos.

Teorema de unificación [Robinson]. Existe un algoritmo (llamado un *algoritmo de unificación*) tal que para cualesquiera dos átomos, si son unificables produce su unificador mas general y si no lo son así lo informa.

Un algoritmo de unificación

El algoritmo recibe los términos t_1 y t_2, si éstos son unificables produce su MGU y si no lo son levanta una bandera para informarlo. En un conjunto de trabajo WS mantiene las parejas de términos por unificar.

MGU $= \{\}$; —sustiución vacía como valor inicial de MGU
WS $= \{\langle t_1, t_2\rangle\}$; —valor inicial de WS
repite
  remover una pareja $\langle u_1, u_2\rangle$ de WS;
  casos
  - u_1 y u_2 son constantes o variables idénticas:
    no hacer nada;
  - u_1 es una variable que no ocurre en u_2:
    reemplazar u_1 con u_2 en WS y en MGU;
    insertar $\langle u_1, u_2\rangle$ en MGU;
  - u_2 es una variable que no ocurre en u_1:
    reemplazar u_2 con u_1 en WS y en MGU;
    insertar $\langle u_2, u_1\rangle$ en MGU;
  - u_1 es $f(s_1, \ldots, s_n)$ y u_2 es $f(s'_1, \ldots, s'_n)$ con símbolo de función f:
    insertar $\langle s_1, s'_1\rangle, \ldots, \langle s_n, s'_n\rangle$ en WS;

  en otros casos
    levanta bandera de unificación fallida
  fin casos
hasta que WS $= \{\}$

Ejemplo

MGU	WS
$\{\ \}$	$\{\ \langle p(m(U,V), m(V,W)),\ \ p(m(c(a,X),X), m(c(b,Y),Y))\rangle\ \}$
$\{\ \}$	$\{\ \langle m(U,V),\ m(c(a,X),X)\rangle,\ \ \langle m(V,W),\ m(c(b,Y),Y)\rangle\ \}$
$\{\ \}$	$\{\ \langle U,\ c(a,X)\rangle,\ \ \langle V,X\rangle,\ \ \langle m(V,W),\ m(c(b,Y),Y)\rangle\ \}$
$\{\ \langle U,\ c(a,X)\rangle\ \}$	$\{\ \langle V,X\rangle,\ \ \langle m(V,W),\ m(c(b,Y),Y)\rangle\ \}$
$\{\ \langle U,\ c(a,X)\rangle,\ \ \langle V,X\rangle\}$	$\{\ \langle m(X,W),\ m(c(b,Y),Y)\rangle\ \}$
$\{\ \langle U,\ c(a,X)\rangle,\ \ \langle V,X\rangle\}$	$\{\ \langle X,\ c(b,Y)\rangle,\ \ \langle W,Y\rangle\ \}$
$\{\ \langle U,\ c(a,c(b,Y))\rangle,\ \ \langle V,\ c(b,Y)\rangle,\ \ \langle X,\ c(b,Y)\rangle\ \}$	$\{\ \langle W,Y\rangle\ \}$
$\{\ \langle U,\ c(a,c(b,Y))\rangle,\ \ \langle V,\ c(b,Y)\rangle,\ \ \langle X,\ c(b,Y)\rangle,\ \ \langle W,Y\rangle\ \}$	$\{\ \}$

Un algoritmo indeterminista de interpretación

Dada una meta G y un programa P, el algoritmo responde SI y da los enlaces de las variables que ocurren en G, o bien responde NO.

SG $= \{G\}$; —valor inicial del conjunto de trabajo de metas

repite

  principio —paso de unificación

    remover un elemento E de SG y

      seleccionar en P una cláusula (renombrada) $X \leftarrow X_1, \ldots, X_n$

      tal que $\langle E, X \rangle$ unifica con MGU μ;

    insertar $X_1, \ldots, X_n$ en SG;

    aplicar μ a SG y a G

  excepción

    cuando el paso de unificación falla => **salir**

  fin

hasta que SG $= \{\}$;

si SG $= \{\}$ **entonces** la respuesta es SI y G describe una solución

  si_no la respuesta es NO

Ejercicios

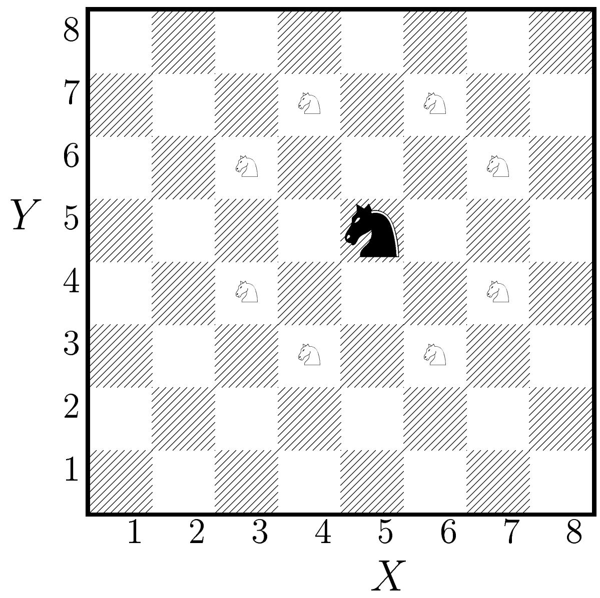
1. Dadas $\alpha = \{\langle X, f(a,Y)\rangle, \langle W, f(U,Z)\rangle\}$ y $\beta = \{\langle X, f(a,a)\rangle, \langle Y, b\rangle, \langle V, c\rangle\}$, obtenga las composiciones $\gamma = \alpha\beta$ y $\delta = \beta\alpha$. Verifique que $\mathcal{F}\gamma = (\mathcal{F}\alpha)\beta$, con $\mathcal{F} = p(V, X, Y, W)$.

2. Dada $\sigma = \{\langle X, f(a,Y)\rangle, \langle Y, f(b,Z)\rangle, \langle Z, c\rangle\}$, determine el menor valor de n para el que $\sigma^n\sigma^n = \sigma^n$ (nota: σ^n denota composición de σ consigo misma, por ejemplo: $\sigma^3 = \sigma\sigma\sigma$).

3. Aplique el algoritmo de unificación a la siguiente pareja de predicados. Describa cada paso del algoritmo.

$$p(m(c(a,X),X),\ m(c(b,Y),Y),\ Z) \quad \text{y}$$
$$p(m(U,V),\ m(V,W),\ m(U,W))$$

Ejemplo: Definición de los movimientos del caballo en el ajedrez.



(sc1) $salto\_caballo(X_1, Y_1, X_2, Y_2) \Leftarrow distan\_uno(X_1, X_2),\ distan\_dos(Y_1, Y_2).$

(sc2) $salto\_caballo(X_1, Y_1, X_2, Y_2) \Leftarrow distan\_dos(X_1, X_2),\ distan\_uno(Y_1, Y_2).$

(du1) $distan\_uno(U, V) \Leftarrow sigue\_a(U, V).$

(du2) $distan\_uno(U, V) \Leftarrow sigue\_a(V, U).$

(dd1) $distan\_dos(U, W) \Leftarrow sigue\_a(U, V),\ sigue\_a(V, W).$

(dd2) $distan\_dos(U, W) \Leftarrow sigue\_a(W, V),\ sigue\_a(V, U).$

(s1) $sigue\_a(1, 2).$

(s2) $sigue\_a(2, 3).$

(s3) $sigue\_a(3, 4).$

(s4) $sigue\_a(4, 5).$

(s5) $sigue\_a(5, 6).$

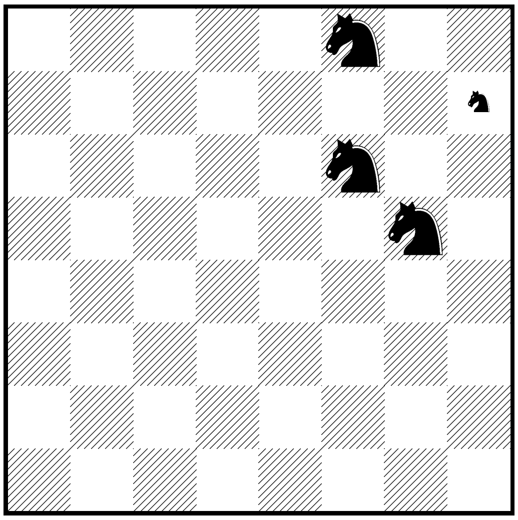
(s6) $sigue\_a(6, 7).$

(s7) $sigue\_a(7, 8).$

Una vez definido el programa lógico anterior, se pueden formular preguntas y obtener respuestas como las ilustradas en las páginas siguientes.

¿A qué casillas puede saltar el caballo desde la casilla $(8, 7)$?

$? -\quad salto\_caballo(8, 7, X, Y).$



$X = 7, \quad Y = 5;$
$X = 6, \quad Y = 8;$
$X = 6, \quad Y = 6.$

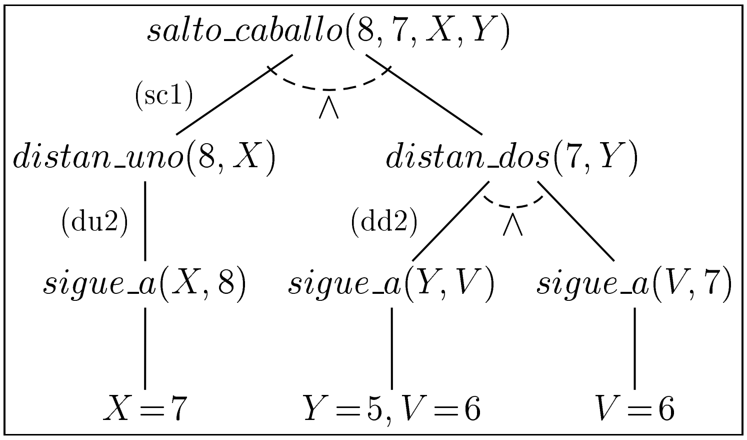
El problema planteado es encontrar los valores de X y Y (las incógnitas principales) para los que se cumple el predicado

$$salto\_caballo(8, 7, X, Y). \qquad (1)$$

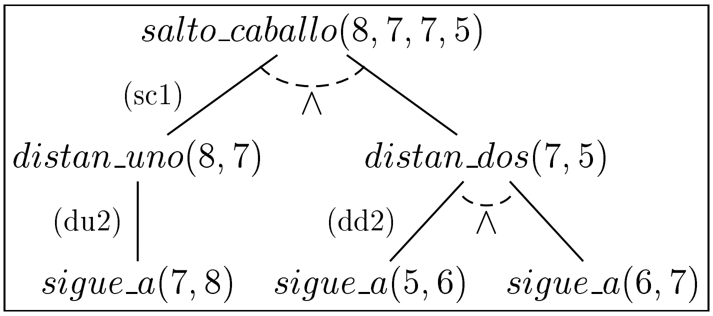
Para resolverlo se usa el método deductivo, buscando los valores de X y Y para los que se pueda deducir la fórmula (1) a partir del programa lógico (vistas sus cláusulas como axiomas). En esencia se procede como se muestra enseguida para el caso de la primera solución. De acuerdo a la cláusula (sc1), una condición suficiente para (1) es:

$$distan\_uno(8, X) \wedge distan\_dos(7, Y) \qquad (2)$$

A su vez (du2) garantiza $distan\_uno(8, X)$ si $sigue\_a(X, 8)$ y, comparando ésta con (s7), se puede "despejar" a X: $X = 7$. En la figura siguiente se representa en forma arborescente esta búsqueda de condiciones suficientes, en ella la trayectoria izquierda corresponde a la obtención de $X = 7$. En el lado derecho del *árbol de prueba* se indica que la segunda parte de (2) se puede obtener vía (dd2) partiendo de $sigue\_a(Y, V)$ y $sigue\_a(V, 7)$; de ésta y (s6) se obtiene $V = 6$ y, ya conocido el valor de la variable auxiliar V, comparando $sigue\_a(Y, 6)$ con (s5) se llega a $Y = 5$.

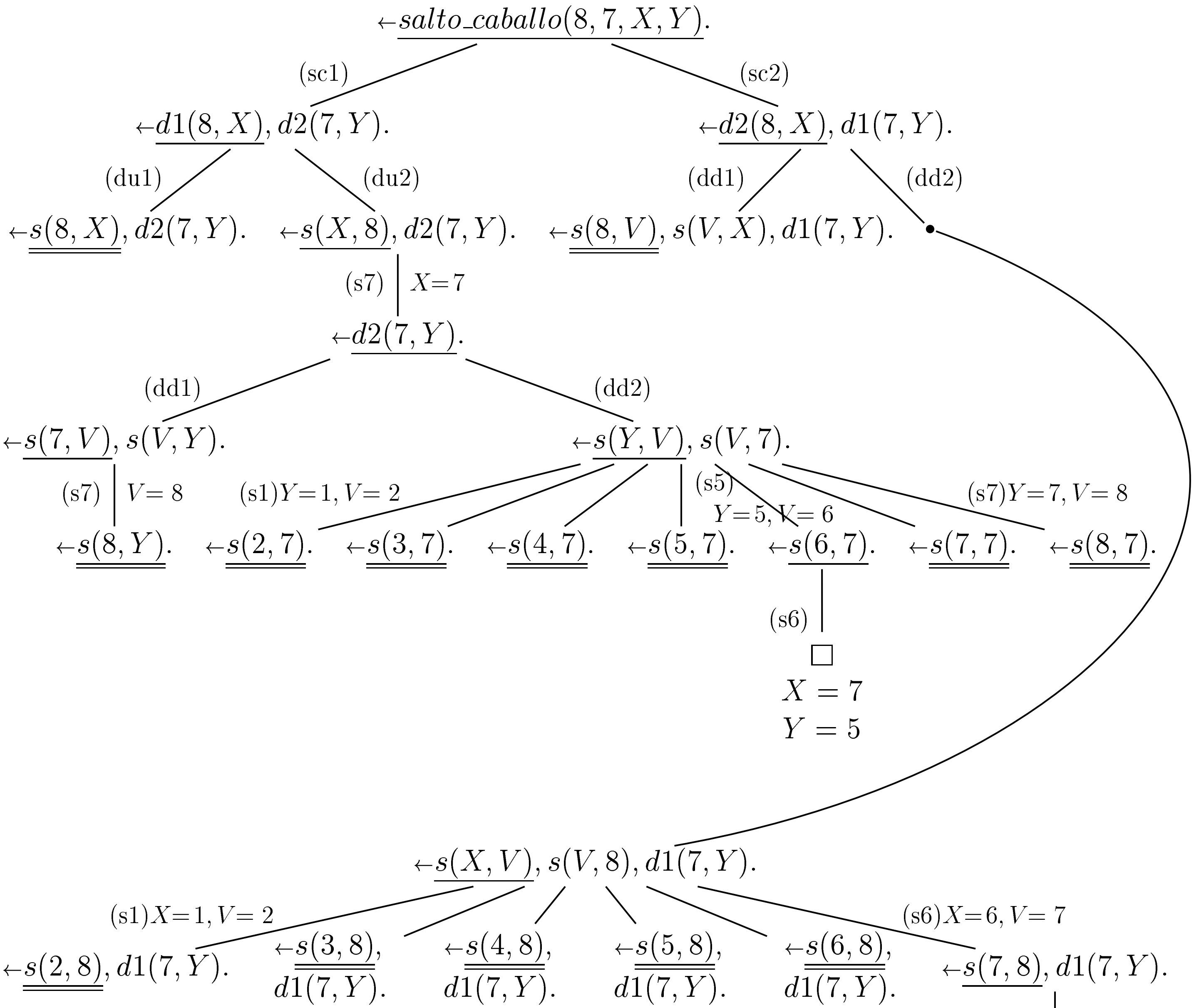


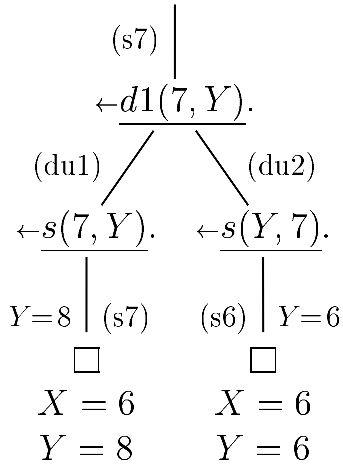
Este árbol se simplifica sustituyendo las variables por sus valores, quedando así:



Leyendo hacia arriba este árbol, se puede construir la siguiente demostración directa de que el caballo puede saltar a la casilla $(7, 5)$, usando tan solo las reglas de inferencia elementales MP:$\frac{P, P\to Q}{Q}$ e introd.–$\wedge$:$\frac{P, Q}{P\wedge Q}$:

1. $distan\_uno(8, 7)$ [de (s7) y (du2)]
2. $distan\_dos(7, 5)$ [de (s5), (s6) y (dd2)]
3. $salto\_caballo(8, 7, 7, 5)$ [de 1, 2 y (sc1)]



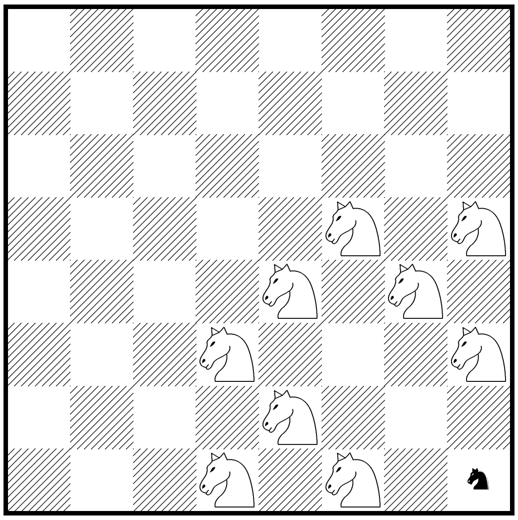


Árbol de derivaciones (o *búsqueda*). Representa la manera en que Prolog encuentra las tres soluciones del ejemplo anterior. Los símbolos de predicado *distan_uno*, *distan_dos* y *sigue_a* del programa se abrevian mediante $d1$, $d2$ y s, respectivamente. En cada nodo, la submeta subrayada se unifica con la cabeza de la cláusula indicada al lado de la arista que lleva al correspondiente nodo descendiente; este nodo se construye del primero, poniendo en lugar de la meta subrayada el cuerpo de la cláusula elegida y aplicando a la fórmula así obtenida la sustitución generada por el algoritmo de unificación. Dicha sustitución aparece expresada con igualdades. Los *nodos de falla* son aquellos en que aparece una meta doblemente subrayada, lo que significa que ésta no es unificable. Los *nodos de éxito* son aquellos en que aparece la cláusula vacía '□'. (Ejercicio: ponga el nodo y etiquetas faltantes.)

¿Desde qué casillas puede llegar el caballo a la casilla $(8,1)$ brincando dos veces?

$?- \quad salto\_caballo(X,Y,X_I,Y_I),\ salto\_caballo(X_I,Y_I,8,1).$

$X = 5, \quad Y = 4;$
$X = 6, \quad Y = 1;$
$X = 6, \quad Y = 5;$
$X = 7, \quad Y = 4;$
$X = 8, \quad Y = 1;$
$X = 8, \quad Y = 5;$
$X = 4, \quad Y = 1;$
$X = 4, \quad Y = 3;$
$X = 5, \quad Y = 2;$
$X = 8, \quad Y = 3.$



¿Puede ir de la $(1,6)$ a la $(3,1)$ dando tres brincos? ¿Cómo?

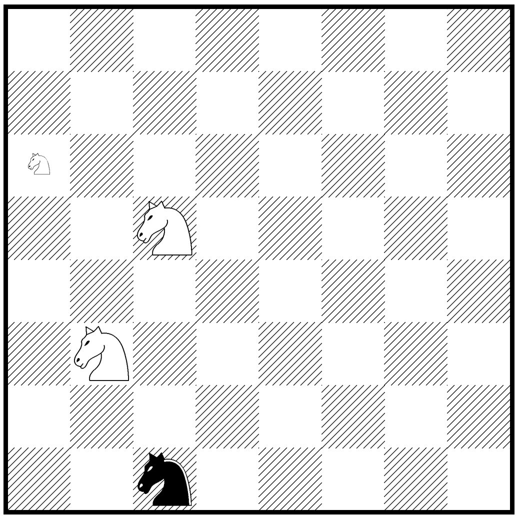
$$?- \quad salto\_caballo(1,6,X_J,Y_J),\ salto\_caballo(X_J,Y_J,X_K,Y_K),\ salto\_caballo(X_K,Y_K,3,1).$$

$X_J = 2, \quad Y_J = 4,$
$X_K = 1, \quad Y_K = 2;$

$X_J = 2, \quad Y_J = 4,$
$X_K = 4, \quad Y_K = 3;$

$X_J = 3, \quad Y_J = 5,$
$X_K = 4, \quad Y_K = 3;$

$X_J = 3, \quad Y_J = 5,$
$X_K = 2, \quad Y_K = 3.$



¿Cuáles son los movimientos del caballo?

$?-\quad salto\_caballo(X_1, Y_1, X_2, Y_2).$

Se obtienen las 336 respuestas siguientes:

X_1	Y_1	X_2	Y_2
1	1	2	3
1	1	3	2
1	2	2	4
1	2	3	1
1	2	3	3
1	3	2	1
1	3	2	5
1	3	3	2
1	3	3	4
1	4	2	2
1	4	2	6
1	4	3	3
1	4	3	5
1	5	2	3
1	5	2	7
1	5	3	4
1	5	3	6
1	6	2	4
1	6	2	8
1	6	3	5
1	6	3	7
1	7	2	5
1	7	3	6
1	7	3	8
1	8	2	6
1	8	3	7
2	1	1	3
2	1	3	3
2	1	4	2
2	2	1	4
2	2	3	4
2	2	4	1
2	2	4	3
2	3	1	1
2	3	1	5
2	3	3	1
2	3	3	5
2	3	4	2
2	3	4	4
2	4	1	2
2	4	1	6
2	4	3	2
2	4	3	6
2	4	4	3
2	4	4	5
2	5	1	3
2	5	1	7
2	5	3	3
2	5	3	7
2	5	4	4
2	5	4	6
2	6	1	4
2	6	1	8
2	6	3	4
2	6	3	8
2	6	4	5
2	6	4	7
2	7	1	5
2	7	3	5
2	7	4	6
2	7	4	8
2	8	1	6
2	8	3	6
2	8	4	7
3	1	1	2
3	1	2	3
3	1	4	3
3	1	5	2
3	2	1	1
3	2	1	3
3	2	2	4
3	2	4	4
3	2	5	1
3	2	5	3
3	3	1	2
3	3	1	4
3	3	2	1
3	3	2	5
3	3	4	1
3	3	4	5
3	3	5	2
3	3	5	4
3	4	1	3
3	4	1	5
3	4	2	2
3	4	2	6
3	4	4	2
3	4	4	6
3	4	5	3
3	4	5	5
3	5	1	4
3	5	1	6
3	5	2	3
3	5	2	7
3	5	4	3
3	5	4	7
3	5	5	4
3	5	5	6
3	6	1	5
3	6	1	7
3	6	2	4
3	6	2	8
3	6	4	4
3	6	4	8
3	6	5	5
3	6	5	7
3	7	1	6
3	7	1	8
3	7	2	5
3	7	4	5
3	7	5	6
3	7	5	8
3	8	1	7
3	8	2	6
3	8	4	6
3	8	5	7
4	1	2	2
4	1	3	3
4	1	5	3
4	1	6	2
4	2	2	1
4	2	2	3
4	2	3	4
4	2	5	4
4	2	6	1
4	2	6	3
4	3	2	2
4	3	2	4
4	3	3	1
4	3	3	5
4	3	5	1
4	3	5	5
4	3	6	2
4	3	6	4
4	4	2	3
4	4	2	5
4	4	3	2
4	4	3	6
4	4	5	2
4	4	5	6
4	4	6	3
4	4	6	5
4	5	2	4
4	5	2	6
4	5	3	3
4	5	3	7
4	5	5	3
4	5	5	7
4	5	6	4
4	5	6	6
4	6	2	5
4	6	2	7
4	6	3	4
4	6	3	8
4	6	5	4
4	6	5	8
4	6	6	5
4	6	6	7
4	7	2	6
4	7	2	8
4	7	3	5
4	7	5	5
4	7	6	6
4	7	6	8
4	8	2	7
4	8	3	6
4	8	5	6
4	8	6	7
5	1	3	2
5	1	4	3
5	1	6	3
5	1	7	2
5	2	3	1
5	2	3	3
5	2	4	4
5	2	6	4
5	2	7	1
5	2	7	3
5	3	3	2
5	3	3	4
5	3	4	1
5	3	4	5
5	3	6	1
5	3	6	5
5	3	7	2
5	3	7	4
5	4	3	3
5	4	3	5
5	4	4	2
5	4	4	6
5	4	6	2
5	4	6	6
5	4	7	3
5	4	7	5
5	5	3	4
5	5	3	6
5	5	4	3
5	5	4	7
5	5	6	3
5	5	6	7
5	5	7	4
5	5	7	6
5	6	3	5
5	6	3	7
5	6	4	4
5	6	4	8
5	6	6	4
5	6	6	8
5	6	7	5
5	6	7	7
5	7	3	6
5	7	3	8
5	7	4	5
5	7	6	5
5	7	7	6
5	7	7	8
5	8	3	7
5	8	4	6
5	8	6	6
5	8	7	7
6	1	4	2
6	1	5	3
6	1	7	3
6	1	8	2
6	2	4	1
6	2	4	3
6	2	5	4
6	2	7	4
6	2	8	1
6	2	8	3
6	3	4	2
6	3	4	4
6	3	5	1
6	3	5	5
6	3	7	1
6	3	7	5
6	3	8	2
6	3	8	4
6	4	4	3
6	4	4	5
6	4	5	2
6	4	5	6
6	4	7	2
6	4	7	6
6	4	8	3
6	4	8	5
6	5	4	4
6	5	4	6
6	5	5	3
6	5	5	7
6	5	7	3
6	5	7	7
6	5	8	4
6	5	8	6
6	6	4	5
6	6	4	7
6	6	5	4
6	6	5	8
6	6	7	4
6	6	7	8
6	6	8	5
6	6	8	7
6	7	4	6
6	7	4	8
6	7	5	5
6	7	7	5
6	7	8	6
6	7	8	8
6	8	4	7
6	8	5	6
6	8	7	6
6	8	8	7
7	1	5	2
7	1	6	3
7	1	8	3
7	2	5	1
7	2	5	3
7	2	6	4
7	2	8	4
7	3	5	2
7	3	5	4
7	3	6	1
7	3	6	5
7	3	8	1
7	3	8	5
7	4	5	3
7	4	5	5
7	4	6	2
7	4	6	6
7	4	8	2
7	4	8	6
7	5	5	4
7	5	5	6
7	5	6	3
7	5	6	7
7	5	8	3
7	5	8	7
7	6	5	5
7	6	5	7
7	6	6	4
7	6	6	8
7	6	8	4
7	6	8	8
7	7	5	6
7	7	5	8
7	7	6	5
7	7	8	5
7	8	5	7
7	8	6	6
7	8	8	6
8	1	6	2
8	1	7	3
8	2	6	1
8	2	6	3
8	2	7	4
8	3	6	2
8	3	6	4
8	3	7	1
8	3	7	5
8	4	6	3
8	4	6	5
8	4	7	2
8	4	7	6
8	5	6	4
8	5	6	6
8	5	7	3
8	5	7	7
8	6	6	5
8	6	6	7
8	6	7	4
8	6	7	8
8	7	6	6
8	7	6	8
8	7	7	5
8	8	6	7
8	8	7	6

Ejemplo: Paseos del caballo sobre el tablero de ajedrez.

Ahora se agrega al programa anterior la definición del predicado $paseo(M, P, Q)$ correspondiente a la relación ternaria "en M saltos el caballo puede ir de la casilla P a la casilla Q". Además, para ilustrar el uso de *términos compuestos*, se modifica ligeramente la definición de $salto\_caballo$, representando las posiciones sobre el tablero mediante términos de la forma $casilla(X, Y)$:

$$salto\_caballo(casilla(X1, Y1), casilla(X2, Y2)) \Leftarrow \\ distan\_uno(X1, X2),\ distan\_dos(Y1, Y2).$$

$$salto\_caballo(casilla(X1, Y1), casilla(X2, Y2)) \Leftarrow \\ distan\_dos(X1, X2),\ distan\_uno(Y1, Y2).$$

$distan\_uno(U, V) \Leftarrow sigue\_a(U, V).$

$distan\_uno(U, V) \Leftarrow sigue\_a(V, U).$

$distan\_dos(U, W) \Leftarrow sigue\_a(U, V),\ sigue\_a(V, W).$

$distan\_dos(U, W) \Leftarrow sigue\_a(W, V),\ sigue\_a(V, U).$

$sigue\_a(a, b).$

$sigue\_a(b, c).$

$sigue\_a(c, d).$

$sigue\_a(d, e).$

$sigue\_a(e, f).$

$sigue\_a(f, g).$

$sigue\_a(g, h).$

Ahora para nombrar las columnas y las hileras del tablero se usan los términos constantes $a, \ldots, h$ en vez de los dígitos $1, \ldots, 8$. Con esto se pretende enfatizar que en el programa anterior para nada se recurre a la relación de orden de los números naturales. Estas cláusulas tan solo definen un orden lineal entre los ocho términos del caso.

Tampoco se supone disponible el sistema de los números naturales para contar saltos del caballo, para ésto la misma definición del predicado *paseo* introduce los términos $uno, s(uno), s(s(uno)), \ldots$, representantes del conjunto $\mathbb{N}^+$:

$paseo(uno, P, Q) \Leftarrow salto\_caballo(P, Q).$

$paseo(s(N), P, Q) \Leftarrow salto\_caballo(P, P1),\ paseo(N, P1, Q).$

¿Adónde puede llegar el caballo en un salto desde la casilla (a, a)?

$? - \quad paseo(uno, casilla(a, a), Z).$

$Z = casilla(b, c)$

$Z = casilla(c, b)$

¿Adónde puede llegar en dos saltos desde la casilla (a, a)?

$? - \quad paseo(s(uno), casilla(a, a), Z).$

$Z = casilla(c, e)$	$Z = casilla(d, d)$	$Z = casilla(e, c)$
$Z = casilla(c, a)$	$Z = casilla(d, b)$	$Z = casilla(e, a)$
$Z = casilla(a, e)$	$Z = casilla(d, d)$	$Z = casilla(a, c)$
$Z = casilla(a, a)$	$Z = casilla(b, d)$	$Z = casilla(a, a)$

¿Adónde puede llegar en tres saltos desde la casilla (a, a)?

$? - \quad paseo(s(s(uno)), casilla(a, a), Z).$

$Z = casilla(d, g)$	$Z = casilla(b, c)$	$Z = casilla(e, f)$	$Z = casilla(d, e)$
$Z = casilla(d, c)$	$Z = casilla(c, b)$	$Z = casilla(e, b)$	$Z = casilla(d, a)$
$Z = casilla(b, g)$	$Z = casilla(e, f)$	$Z = casilla(c, f)$	$Z = casilla(g, d)$
$Z = casilla(b, c)$	$Z = casilla(e, b)$	$Z = casilla(c, b)$	$Z = casilla(g, b)$
$Z = casilla(e, f)$	$Z = casilla(c, f)$	$Z = casilla(f, e)$	$Z = casilla(c, d)$
$Z = casilla(e, d)$	$Z = casilla(c, b)$	$Z = casilla(f, c)$	$Z = casilla(c, b)$
$Z = casilla(a, f)$	$Z = casilla(f, e)$	$Z = casilla(b, e)$	$Z = casilla(f, c)$
$Z = casilla(a, d)$	$Z = casilla(f, c)$	$Z = casilla(b, c)$	$Z = casilla(d, c)$
$Z = casilla(d, c)$	$Z = casilla(b, e)$	$Z = casilla(c, f)$	$Z = casilla(g, b)$
$Z = casilla(b, c)$	$Z = casilla(b, c)$	$Z = casilla(c, b)$	$Z = casilla(c, b)$
$Z = casilla(e, b)$	$Z = casilla(e, d)$	$Z = casilla(a, f)$	$Z = casilla(b, e)$
$Z = casilla(a, b)$	$Z = casilla(c, d)$	$Z = casilla(a, b)$	$Z = casilla(b, a)$
$Z = casilla(b, g)$	$Z = casilla(f, c)$	$Z = casilla(d, e)$	$Z = casilla(c, d)$
$Z = casilla(b, c)$	$Z = casilla(f, a)$	$Z = casilla(d, c)$	$Z = casilla(c, b)$
$Z = casilla(c, f)$	$Z = casilla(b, c)$	$Z = casilla(f, e)$	$Z = casilla(b, c)$
$Z = casilla(c, d)$	$Z = casilla(b, a)$	$Z = casilla(f, a)$	$Z = casilla(c, b)$

¿Adónde puede llegar en M saltos desde la casilla (a, a)?

$?- \quad paseo(M, casilla(a, a), Z).$

$Z = casilla(b, c) \;\; M = uno$
$Z = casilla(c, b) \;\; M = uno$

$Z = casilla(c, e) \;\; M = s(uno)$
$Z = casilla(c, a) \;\; M = s(uno)$
$Z = casilla(a, e) \;\; M = s(uno)$
$Z = casilla(a, a) \;\; M = s(uno)$
$Z = casilla(d, d) \;\; M = s(uno)$
$Z = casilla(d, b) \;\; M = s(uno)$

$Z = casilla(d, g) \;\; M = s(s(uno))$
$Z = casilla(d, c) \;\; M = s(s(uno))$
$Z = casilla(b, g) \;\; M = s(s(uno))$
$Z = casilla(b, c) \;\; M = s(s(uno))$
$Z = casilla(e, f) \;\; M = s(s(uno))$
$Z = casilla(e, d) \;\; M = s(s(uno))$
$Z = casilla(a, f) \;\; M = s(s(uno))$
$Z = casilla(a, d) \;\; M = s(s(uno))$

$Z = casilla(e, e) \;\; M = s(s(s(uno)))$
$Z = casilla(c, e) \;\; M = s(s(s(uno)))$
$Z = casilla(f, h) \;\; M = s(s(s(uno)))$
$Z = casilla(f, f) \;\; M = s(s(s(uno)))$
$Z = casilla(b, h) \;\; M = s(s(s(uno)))$
$Z = casilla(b, f) \;\; M = s(s(s(uno)))$

$Z = casilla(f, g) \;\; M = s(s(s(s(uno))))$
$Z = casilla(f, c) \;\; M = s(s(s(s(uno))))$
$Z = casilla(d, g) \;\; M = s(s(s(s(uno))))$
$Z = casilla(d, c) \;\; M = s(s(s(s(uno))))$
$Z = casilla(g, f) \;\; M = s(s(s(s(uno))))$
$Z = casilla(g, d) \;\; M = s(s(s(s(uno))))$
$Z = casilla(c, f) \;\; M = s(s(s(s(uno))))$
$Z = casilla(c, d) \;\; M = s(s(s(s(uno))))$

$Z = casilla(g, e) \;\; M = s(s(s(s(s(uno)))))$
$Z = casilla(e, e) \;\; M = s(s(s(s(s(uno)))))$
$Z = casilla(h, h) \;\; M = s(s(s(s(s(uno)))))$
$Z = casilla(h, f) \;\; M = s(s(s(s(s(uno)))))$
$Z = casilla(d, h) \;\; M = s(s(s(s(s(uno)))))$
$Z = casilla(d, f) \;\; M = s(s(s(s(s(uno)))))$

$Z = casilla(h, g) \;\; M = s(s(s(s(s(s(uno))))))$
$Z = casilla(h, c) \;\; M = s(s(s(s(s(s(uno))))))$
$Z = casilla(f, g) \;\; M = s(s(s(s(s(s(uno))))))$
$Z = casilla(f, c) \;\; M = s(s(s(s(s(s(uno))))))$
$Z = casilla(e, f) \;\; M = s(s(s(s(s(s(uno))))))$
$Z = casilla(e, d) \;\; M = s(s(s(s(s(s(uno))))))$

$Z = casilla(g, e) \;\; M = s(s(s(s(s(s(s(uno)))))))$
$Z = casilla(f, h) \;\; M = s(s(s(s(s(s(s(uno)))))))$
$Z = casilla(f, f) \;\; M = s(s(s(s(s(s(s(uno)))))))$

$Z = casilla(h, g) \;\; M = s(s(s(s(s(s(s(s(uno))))))))$
$Z = casilla(h, c) \;\; M = s(s(s(s(s(s(s(s(uno))))))))$
$Z = casilla(f, g) \;\; M = s(s(s(s(s(s(s(s(uno))))))))$
$Z = casilla(f, c) \;\; M = s(s(s(s(s(s(s(s(uno))))))))$
$Z = casilla(e, f) \;\; M = s(s(s(s(s(s(s(s(uno))))))))$
$Z = casilla(e, d) \;\; M = s(s(s(s(s(s(s(s(uno))))))))$
$\vdots$

De la infinidad de respuestas, las que puede generar Prolog —debido a la manera en que interpreta al programa lógico— son las que obtiene de la siguiente manera. Las dos primeras corresponden a las casillas a las que el caballo llega en un salto; desde la primera de éstas, (b, c), brinca a las seis casillas dadas en el segundo grupo de respuestas; a su vez, desde la primera de éstas, (c, e), salta a las ocho casillas listadas en el tercer grupo de respuestas; así se van obteniendo respuestas correspondientes a paseos cada vez más largos. Como este proceso es infinito, en la lista de respuestas no aparecerán todas las soluciones correspondientes a paseos de longitud mayor que uno, por ejemplo los de longitud dos que pasan por la casilla (c, b) obtenida en la segunda respuesta (compárese con los ejemplos anteriores, de un número finito de soluciones). Nótese que las seis casillas del último grupo mostrado, a las que el caballo llega en nueve saltos, son las mismas que las obtenidas para siete saltos; esta periodicidad se debe a que el tablero es finito y a la forma en que Prolog genera las respuestas. (Ejercicio: Si se listaran mas respuestas, ¿cuáles serían las correpondientes a paseos de longitud 10, 11 y 17?)