Reglas de Control en Planificación

Miguel García Remesal
Departamento de Inteligencia Artificial

mgremesal@fi.upm.es

Motivación

- Uso de conocimiento de dominio para planificar más eficientemente

- Uso de un algoritmo de planificación configurable según el dominio
  - Motor de inferencia independiente del dominio
  - La entrada del algoritmo contiene reglas específicas para podar el árbol de búsqueda

- Uso de Lógica Temporal Lineal para formalizar las reglas de poda

Lógica de Primer Orden

- Símbolos de constante, símbolos de variable y símbolos de predicado:
  - Ejemplos del mundo de los bloques:
    - Constantes: A, B, C
    - Variables: x, y, z
    - Predicados: en-mesa, encima, brazo-libre

- Conectivas lógicas ($\vee$, $\wedge$, $\neg$, $\rightarrow$, $\leftrightarrow$)

- Cuantificadores ($\forall$, $\exists$)

Ejemplos:

$$\text{encima}(A,B) \wedge \text{encima}(B,C)$$

$$\exists x\, \text{encima}(x,A)$$

$$\forall x\, (\text{en-mesa}(x) \rightarrow \text{libre}(x))$$

Teorías de Primer Orden

- Una teoría de primer orden **T** incluye:
  - Axiomas lógicos y reglas de inferencia
    - Independientes del dominio
    - Representan el razonamiento lógico en general
  - Axiomas "no lógicos"
    - Dependientes del dominio
    - Representan reglas de razonamiento en un dominio específico
  - Teoremas: generados mediante la aplicación de axiomas y reglas de inferencia

Modelos

- En problemas de planificación, un modelo se refiere a un estado ***s*** del problema
- Para que ***s*** sea un modelo, todos los teoremas de **T** deben ser ciertos en ***s***
- Ejemplo: s |= encima(A, B) puede leerse como:
  - **s** es un modelo de encima(A, B)
  - **s** satisface encima(A, B)
  - encima(A, B) es cierta en el estado **s**

Lógica Lineal Temporal

- Propósito: poder expresar la noción de tiempo, que es difícil de representar en LPO
  - Una secuencia infinita <0, 1, 2, …> de instantes de tiempo
  - Una secuencia infinita $M = <s_0, s_1, s_0, \ldots>$ de estados del mundo

- La Lógica Lineal Temporal se apoya en:
  - Lógica de Primer Orden
  - Símbolos proposicionales **TRUE** y **FALSE**
  - Operadores modales
  - Cuantificadores limitados
  - Predicado **GOAL**

Operadores Modales

- Se refieren a los estados del múndo en los que las fórmulas son ciertas:

Operador	Símbolo	Explicación	Ejemplo
Siguiente	$\bigcirc f$	*f* será cierto en el próximo estado	$\bigcirc$ encima(A, B)
Finalmente	$\Diamond f$	*f* será cierto en algún estado futuro	$\Diamond$ libre(A)
Siempre	$\square f$	*f* es cierto ahora y lo será también en todos los estados futuros	$\square$ en-mesa(A)
Hasta	$f_1 \cup f_2$	F_2 es cierto ahora o lo será en algún estado futuro y f_1 es cierto hasta entonces	libre(B) $\cup$ en-mesa(A)

Cuantificadores Limitados

Sea $g(x)$ tal que $\{x : g(x)\} = \{x_1, x_2, ..., x_n\}$ sea finito y fácilmente computable:

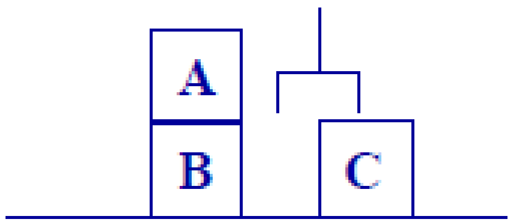
$$\forall [x : g(x)] f(x) = f(x_1) \wedge f(x_2) \wedge ... \wedge f(x_n)$$

$$\exists [x : g(x)] f(x) = f(x_1) \vee f(x_2) \vee ... \vee f(x_n)$$

Cuantificadores Limitados (ejemplos)

$$\forall x\left[\text{en-mesa}(x)\right]\Diamond\text{sujeto}(x)$$

$$\Diamond\text{sujeto}(B)\wedge\Diamond\text{sujeto}(C)$$



$$\exists x\left[\text{libre}(x)\right]\bigcirc\text{sujeto}(x)$$

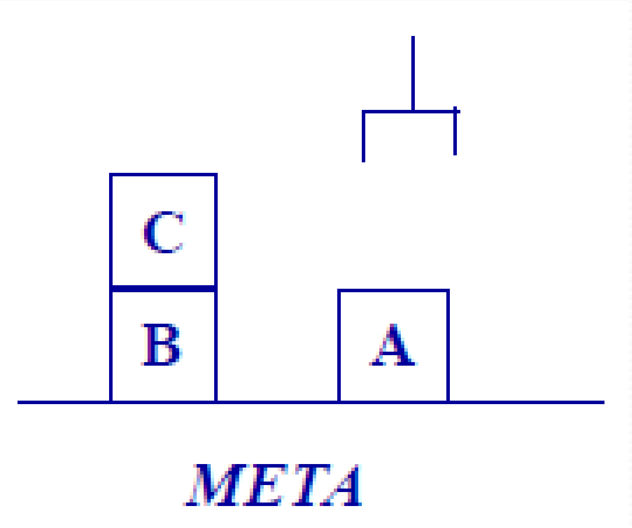
$$\bigcirc\text{sujeto}(A)\vee\bigcirc\text{sujeto}(C)$$

Predicado GOAL

- GOAL(*f*) indica que la fórmula *f* es cierta en todos los estados finales (objetivo)

$$\mathrm{GOAL}(f) \leftrightarrow \forall [s_i \in g]\, s_i \models f$$

- Ejemplos:
  - GOAL(encima(C, B))
  - GOAL(brazo-libre)
  - GOAL(en-mesa(A))
  - ...



Reglas de Control

- Objetivo: expresar conocimiento de dominio mediante fórmulas de la LTL
  - Definir predicados auxiliares a utilizar por las reglas de control, en caso de ser necesario
  - Formular las reglas de control
- Las reglas de control se utilizarán para podar el árbol de búsqueda

Reglas de Control: Ejemplo de Definición de Predicados Auxiliares

- pila-correcta: una pila de bloques que no será necesario modificar para alcanzar el estado objetivo
  - pila-correcta(x): predicado que indica que x es el bloque en la cima de la pila correcta

- pila-incorrecta: una pila de bloques que será necesario modificar para alcanzar el estado objetivo
  - pila-incorrecta(x): predicado que indica que x es el bloque en la cima de la pila incorrecta

Reglas de Control: Ejemplo de Definición de Predicados Auxiliares

$$\text{pila-correcta}(x) \leftrightarrow \text{libre}(x) \wedge \neg \text{GOAL}(\text{sujeto}(x)) \wedge \text{pila-correcta-bajo}(x)$$

$$\text{pila-correcta-bajo}(x) \leftrightarrow (f_1 \wedge f_2) \vee \left[\exists [y : \text{encima}(x,y)] (f_3 \wedge f_4 \wedge f_5 \wedge f_6 \wedge f_7 \wedge f_8)\right]$$

$$f_1 = \text{en-mesa}(x)$$

$$f_2 = \neg \exists \left[y : \text{GOAL}(\text{encima}(x,y)) \right]$$

$$f_3 = \neg \text{GOAL}(\text{en-mesa}(x))$$

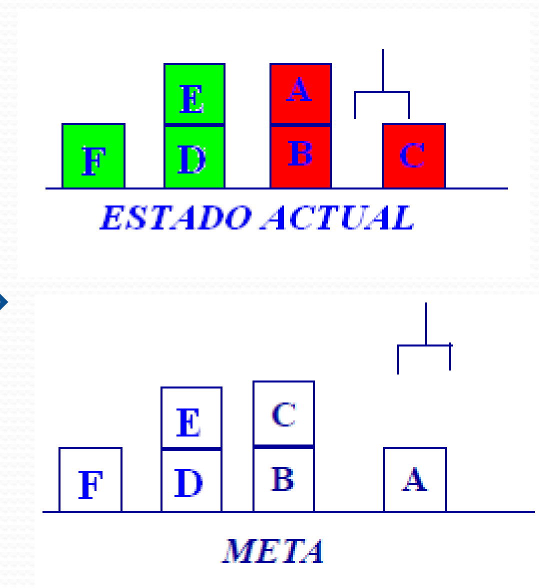
$$f_4 = \neg \text{GOAL}(\text{sujeto}(y))$$

$$f_5 = \neg \text{GOAL}(\text{libre}(y))$$

$$f_6 = \forall \left[z : \text{GOAL}(\text{encima}(x,z)) \right] z = y$$

$$f_7 = \forall \left[z : \text{GOAL}(\text{encima}(z,y)) \right] z = x$$

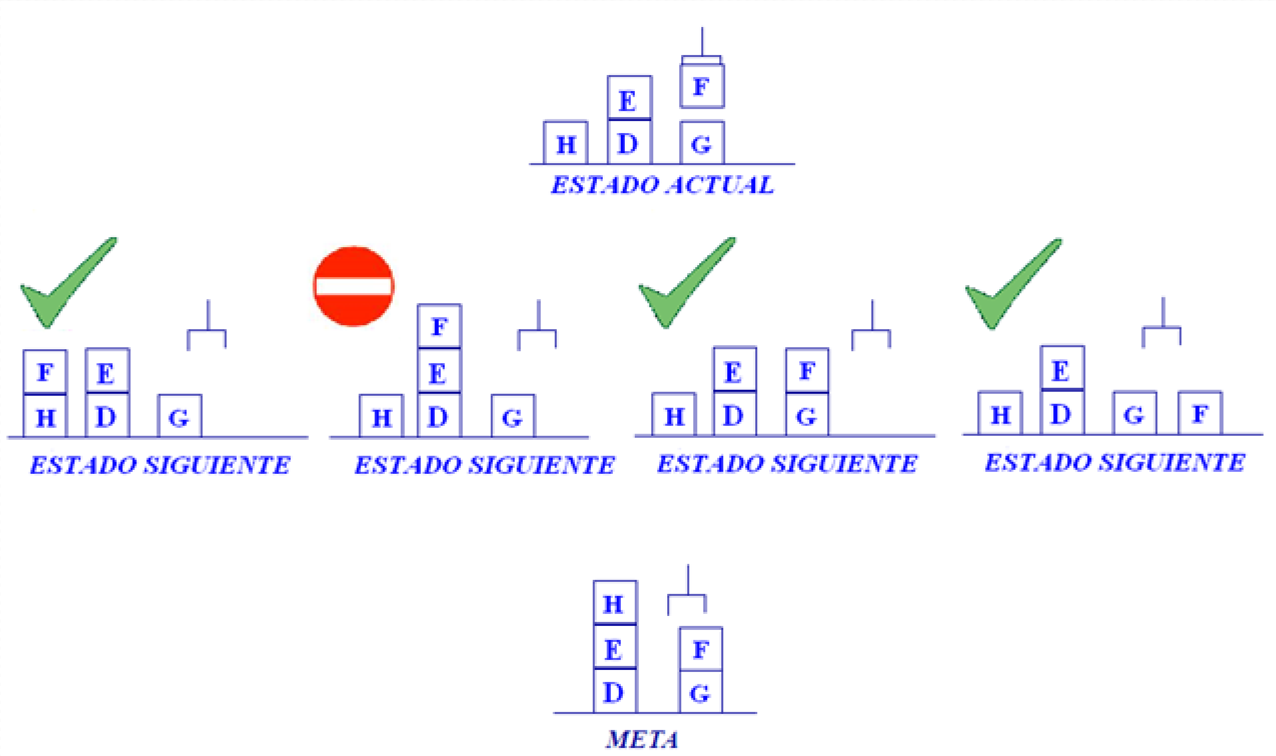
$$f_8 = \text{torre-correcta-bajo}(y)$$



$$\text{pila-incorrecta}(x) \leftrightarrow \text{libre}(x) \wedge (\text{GOAL}(\text{sujeto}(x)) \vee \neg \text{pila-correcta-bajo}(x))$$

Ejemplos de Reglas de Control

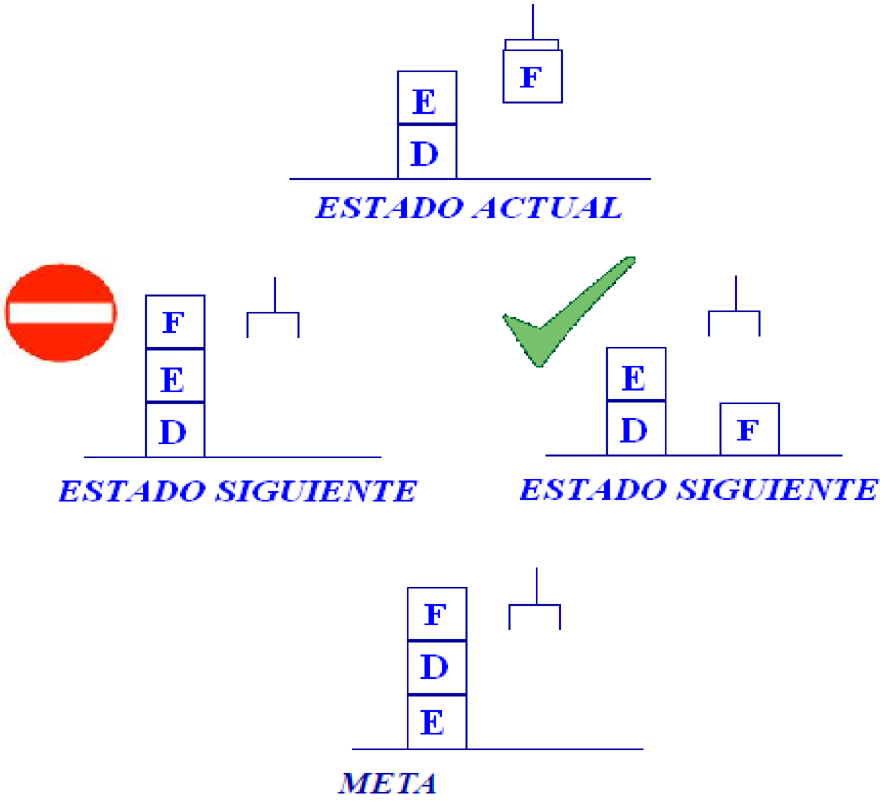
- Regla de Control 1:
  - Una pila correcta debe siempre permanecer correcta



$$RC_1 = \square\Big(\forall\left[x : \text{libre}(x)\right]\text{pila-correcta}(x) \rightarrow \bigcirc\Big(\text{libre}(x) \vee \exists\left[y : \text{encima}(y,x)\right]\text{pila-correcta}(y)\Big)\Big)$$

Ejemplos de Reglas de Control

- Regla de control 2:
  - Igual que la anterior, exigiendo además que nunca se coloque un bloque sobre una pila incorrecta



$$\mathrm{RC}_2 = \mathrm{RC}_1 \wedge \Box\left(\text{pila-incorrecta}(x) \rightarrow \bigcirc\left(\neg\exists\left[y : \text{encima}(y, x)\right]\right)\right)$$

Pseudocódigo TLPLAN

- **PROCEDIMIENTO** TLPlan(s, f, g, π)
  1. $f^+ \leftarrow$ Actualizar(f, s)
  2. **SI** f^+ = FALSO **ENTONCES DEVOLVER** FALLO
  3. **SI** s satisface g **ENTONCES DEVOLVER** π
  4. $A \leftarrow$ {Acciones aplicables a s}
  5. **SI** $A = \Phi$ **ENTONCES DEVOLVER** FALLO
  6. Elegir una acción $a \in A$
  7. **HACER** $s^+ \leftarrow \gamma(s, a)$
  8. **HACER** $\pi^+ = \pi.a$
  9. **DEVOLVER** TLPlan(s^+, f^+, g, π^+)

Procedimiento Actualizar

- **PROCEDIMIENTO** Actualizar(f, s)
- **SWITCH**(f)
  - **CASE** (f no contiene operadores temporales):
    - SI s |= f ENTONCES f^+ := TRUE EN OTRO CASO f^+ := FALSE
  - **CASE** ($f = f_1 \wedge f_2$):
    - f^+ := Actualizar(f_1, s) $\wedge$ Actualizar(f_2, s)
  - **CASE** ($f = \neg f_1$):
    - f^+ := $\neg$Actualizar(f_1, s)
  - **CASE** ($f = \circ f_1$):
    - $f^+ := f_1$
  - **CASE** ($f = f_1 \cup f_2$):
    - f^+ := Actualizar(f_2, s) $\vee$ (Actualizar(f_1, s) $\wedge$ f)
  - **CASE** ($f = \diamond f_1$):
    - f^+ := Actualizar(f_1, s) $\vee$ f
  - **CASE** ($f = \square f_1$):
    - f^+ := Actualizar(f_1, s) $\wedge$ f
  - **CASE** ($f = \forall[x:g(x)]f_1$):
    - $f^+ := \bigwedge$ {Actualizar($\theta(f_1)$, s): s |= g(c)}, donde $\theta = \{x \leftarrow c\}$
  - **CASE** ($f = \exists[x:g(x)]f_1$):
    - $f^+ := \bigvee$ {Actualizar($\theta(f_1)$, s): s |= g(c)}, donde $\theta = \{x \leftarrow c\}$

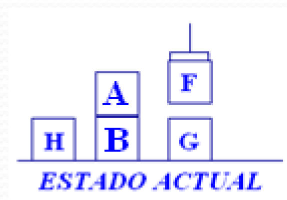
Proc. Actualizar: Ejemplo 1 de 3

- f = □encima(A, B)

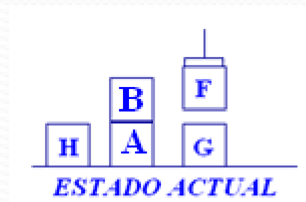
- f^+ = Actualizar(□encima(A, B), s)
- f^+ = Actualizar(encima(A, B), s) $\wedge$ □encima(A, B)

- Dos posibilidades:

  - f^+ = TRUE $\wedge$ □encima(A, B)
  - f^+ = □encima(A, B)



  - f^+ = FALSE $\wedge$ □encima(A, B)
  - f^+ = FALSE



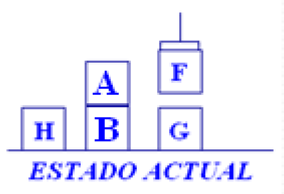
- Conclusión:
  - □ evalúa la regla de control en el estado actual
  - Si la regla se cumple en el estado actual, □ la propaga al próximo estado
  - Si la regla no se cumple, se poda la rama (callejón sin salida)

Proc. Actualizar: Ejemplo 2 de 3

- $f = \Box(\text{encima}(A, B) \rightarrow \bigcirc\text{libre}(A))$

- $f^+ = \text{Actualizar}(\Box(\text{encima}(A, B) \rightarrow \bigcirc\text{libre}(A)), s)$
- $f^+ = \text{Actualizar}(\text{encima}(A, B) \rightarrow \bigcirc\text{libre}(A), s) \wedge \Box(\text{encima}(A, B) \rightarrow \bigcirc\text{libre}(A))$

- Dos posibilidades:
  - $f^+ = \text{Actualizar}(\bigcirc\text{libre}(A), s) \wedge \Box(\text{encima}(A, B) \rightarrow \bigcirc\text{libre}(A))$
  - $f^+ = \text{libre}(A) \wedge \Box(\text{encima}(A, B) \rightarrow \bigcirc\text{libre}(A))$



  - $f^+ = \text{TRUE} \wedge \Box(\text{encima}(A, B) \rightarrow \bigcirc\text{libre}(A))$
  - $f^+ = \Box(\text{encima}(A, B) \rightarrow \bigcirc\text{libre}(A))$

p	q	→
0	0	1
0	1	1
1	0	0
1	1	1

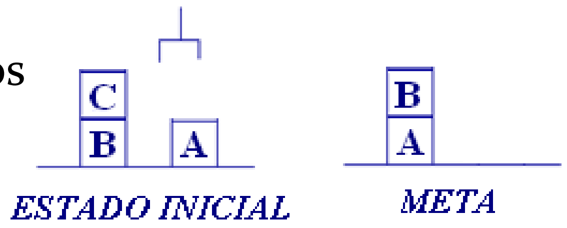
Proc. Actualizar: Ejemplo 3 de 3

- $f = \Box\forall[x: libre(x)]\{(en\text{-}mesa(x) \wedge \neg\exists[y: GOAL(encima(x, y))]) \rightarrow \bigcirc\neg sujeto(x)\}$

- $f^+ = Actualizar(f, s)$
- $f^+ = Actualizar(\forall[x: libre(x)]\{(en\text{-}mesa(x) \wedge \neg\exists[y: GOAL(encima(x, y))]) \rightarrow \bigcirc\neg sujeto(x)\}, s) \wedge f$

- Elementos que cumplen el filtro del cuantificador universal: A y C
- $f^+ = f_1 \wedge f_2 \wedge f$, donde:
  - $f_1 = Actualizar(\{(en\text{-}mesa(A) \wedge \neg\exists[y: GOAL(encima(A, y))]) \rightarrow \bigcirc\neg sujeto(A)\}, s)$
  - $f_2 = Actualizar(\{(en\text{-}mesa(C) \wedge \neg\exists[y: GOAL(encima(C, y))]) \rightarrow \bigcirc\neg sujeto(C)\}, s)$
  - f es la regla de control original

- Realizamos cada f_i por separado para facilitar los cálculos

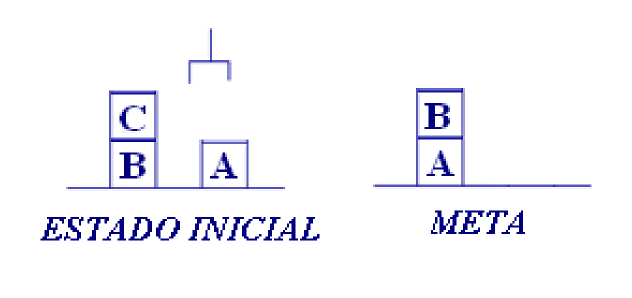


Proc. Actualizar: Ejemplo 3 de 3 (cont.)

- Calculamos f_1:

- $f_1 = \text{Actualizar}(\{(\text{en-mesa}(A) \wedge \neg\exists[y\text{: GOAL}(\text{encima}(A, y))]) \rightarrow \bigcirc\neg\text{sujeto}(A)\}, s)$
- $f_1 = \text{Actualizar}(\{(\text{TRUE} \wedge \neg\exists[y\text{: GOAL}(\text{encima}(A, y))]) \rightarrow \bigcirc\neg\text{sujeto}(A)\}, s)$
- $f_1 = \text{Actualizar}(\{\neg\exists[y\text{: GOAL}(\text{encima}(A, y))] \rightarrow \bigcirc\neg\text{sujeto}(A)\}, s)$

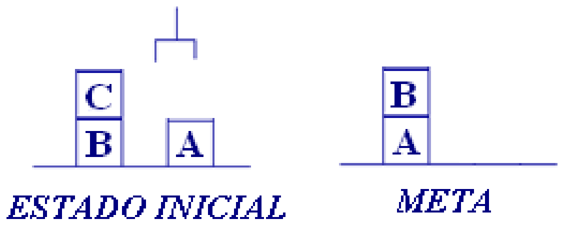
- Elementos que cumplen el filtro del cuantificador existencial: ninguno → el antecedente de la implicación es verdadero
- $f_1 = \text{Actualizar}(\bigcirc\neg\text{sujeto}(A), s)$
- **$f_1 = \neg\text{sujeto}(A)$**



Proc. Actualizar: Ejemplo 3 de 3 (cont.)

- Calculamos f_2:

- f_2 = Actualizar($\{(\text{en-mesa}(C) \wedge \neg\exists[y: \text{GOAL}(\text{encima}(C, y))]) \rightarrow \bigcirc\neg\text{sujeto}(C)\}, s$)
- f_2 = Actualizar($\{(\text{FALSE} \wedge \neg\exists[y: \text{GOAL}(\text{encima}(C, y))]) \rightarrow \bigcirc\neg\text{sujeto}(C)\}, s$)
- f_2 = Actualizar($\{\text{FALSE} \rightarrow \bigcirc\neg\text{sujeto}(C)\}, s$)
- Como el antecedente de la implicación es falso, la implicación es verdadera
- **f_2 = TRUE**



Proc. Actualizar: Ejemplo 3 de 3 (cont.)

- La regla de control actualizada es por tanto: $f^+ = f_1 \wedge f_2 \wedge f$
- $f^+ = \neg$sujeto(A) $\wedge$ TRUE $\wedge$ $\Box\forall$[x: libre(x)]{(en-mesa(x) $\wedge$ $\neg\exists$[y: GOAL(encima(x, y))])$\rightarrow$ $\bigcirc\neg$sujeto(x)}
- Simplificando:

$f^+ = \neg$sujeto(A) $\wedge$ $\Box\forall$[x: libre(x)]{(en-mesa(x) $\wedge$ $\neg\exists$[y: GOAL(encima(x, y))])$\rightarrow$ $\bigcirc\neg$sujeto(x)}

Comparativas de Rendimiento