



# ANÁLISIS MEDIOS - FINES

---

**Una dificultad en Inteligencia Artificial, es conseguir un procedimiento para que el computador pueda formular planes para resolver problemas.**

**Los conceptos básicos de un problema son: enunciados, operadores y meta.**

---

# GPS

**El GPS evalúa la eficiencia de los medios seleccionados para alcanzar las metas o fines.**

---

## FINES DEL GPS

- 1.- Transformar el enunciado **x** en el enunciado **y**
- 2.- Reducir la diferencia **d** sobre el enunciado **x**
- 3.- Aplicar el operador **o** al enunciado **x**
- 4.- Seleccionar los elementos del conjunto **c** que verifiquen el mejor criterio **k**

---

# SELECCIÓN DE OPERADORES

**Existen dos criterios básicos para seleccionar los operadores:**

- a) Conveniencia: producir un objeto similar a la situación deseada**
- b) Facilidad: debería ser aplicable a los objetos de entrada**

---

# PROCEDIMIENTOS DEL GPS

## a) Procedimiento transformar **x** en **y**

1. Identificar las diferencias relevantes entre los elementos. Si no existen, la transformación es completa

2. Seleccionar el operador **o** más relevante de la tabla de conexiones

3. Aplicar el operador **o** a **x**. Llamar al elemento resultante **z**

4. Hacer **x = z** e ir a 1

---

# PROCEDIMIENTOS DEL GPS

**b) Procedimiento para aplicar un operador  $\circ$  a un**

**elemento  $x$**

**1. Transformar el elemento  $x$  en uno que pueda usar el operador**

**2. Ejecutar la operación produciendo un nuevo elemento  $z$**

---

## EJEMPLO DEL GPS

**En una habitación existe un racimo de bananas colgado del techo, también existe una caja, y un mono.**

**El problema consiste en trazar un plan para que el mono pueda alcanzar las bananas, cosa que no puede hacer si no se auxilia de la caja.**

---

## REPRESENTACIÓN

**(W,X,Y,Z) siendo:**

**W = posición horizontal del mono (una variable)**

**X = 1 ó 0, dependiendo de si el mono está o no encima de  
la caja**

**Y = posición horizontal de la caja (una variable)**

**Z = 1 ó 0, dependiendo de si el mono tiene o no las  
bananas**

---

## ESTADO INICIAL - META

$(W,X,Y,Z) \text{ ----- } (c,1,c,1)$

siendo c la posición en el suelo debajo de las bananas

---

## OPERADORES

- **IR A (u).**- El mono va a la posición horizontal u (una variable)
- **EMPUJAR LA CAJA (v).**- El mono empuja la caja a la posición horizontal v (una variable)
- **SALTAR A LA CAJA.**- El mono salta encima de la caja
- **APODERARSE.**- El mono se apodera de las bananas

---

## RESOLUCIÓN

- **Supongamos que inicialmente el mono esta en el suelo en la posición **a** y la caja en la posición **b**. Así, el estado inicial será  $(a,0,b,0)$ .**
- **El único operador que es aplicable es IR A (**u**), resultando  $(u,0,b,0)$ .**

---

## RESOLUCIÓN

- **Ahora dos operadores son aplicables, si  $u=b$ , el mono puede o bien SALTAR A LA CAJA o bien EMPUJARLA, si SALTA A LA CAJA el estado seria  $(b,1,b,0)$ ; si  $u \neq b$  EMPUJANDO LA CAJA A  $v$ , daría  $(v,0,v,0)$  y desplazándose a algún lugar descrito por una nueva variable no cambia la descripción;**

---

## RESOLUCIÓN

- **Estando en la posición  $c$  el estado sería  $(c,0,c,0)$ , aplicando el operador **SALTAR A LA CAJA**, el estado sería  $(c,1,c,0)$  y aplicando **APODERARSE**, el estado sería el estado meta  $(c,1,c,1)$**

---

## SECUENCIA DE OPERADORES

**IR A** la caja **EMPUJAR LA CAJA** hasta debajo de las  
bananas **SALTAR A LA CAJA** y **APODERARSE** de  
las bananas

---

## **INCOVENIENTES DEL GPS**

- a) El estado inicial y la meta pueden ser tan diferentes, que tenemos pocas claves de cómo pasar de un estado al otro.**
- b) Las características de una situación pueden interactuar con otras.**
- c) Los problemas para reducir diferencias, pueden consumir mucho tiempo en descubrir operadores que no conduzcan a la solución.**

---

# STRIPS

**Como cualquier planificador strips**

**consta de:**

- Predicados**
- Operadores**
- Reglas de Inferencia**

---

## STRIPS

**En strips se asume la hipótesis del “mundo cerrado”:**

**Cualquier afirmación que no aparezca explícitamente representada en el estado o pueda ser deducida por las reglas de inferencia, es falsa**

---

# STRIPS

Para la *formalización* se utilizan:

**A) Fórmulas lógicas en un lenguaje de primer orden para describir el estado inicial y final**

---

# STRIPS

## **B) Descripciones de operadores mediante:**

- Precondiciones**
- Añadidos**
- Borrados**

---

# STRIPS

**La *búsqueda* del plan se hace hacia atrás en profundidad y por pila de metas**

---

# Formalización de STRIPS. Ejemplo de descripción de operadores

---

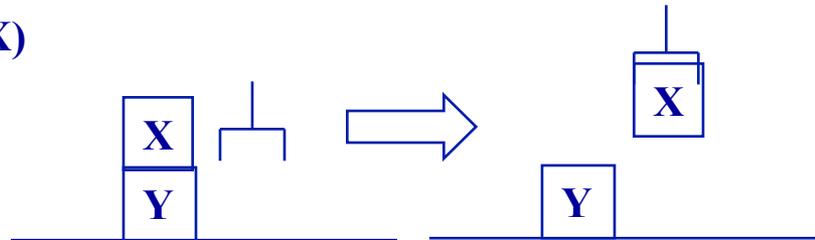
# Formalización de STRIPS. Ejemplo de descripción de operadores

## **QUITAR(X, Y)**

**precondiciones:** encima (X, Y)  $\wedge$  libre (X)  
 $\wedge$  brazo-libre  $\wedge$  en mesa (Y)

**añadidos:** sujeto (X)  $\wedge$  libre (Y)

**borrados:** brazo-libre  $\wedge$  encima (X, Y)  
 $\wedge$  libre (X)

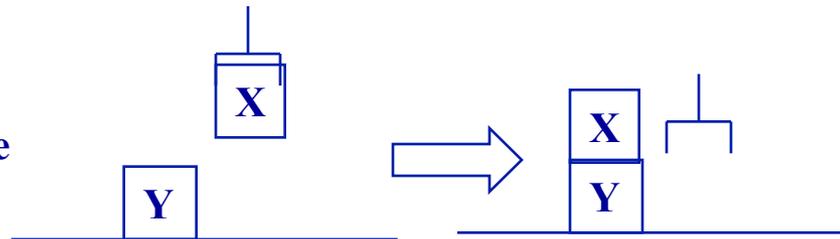


## **PONER(X, Y)**

**precondiciones:** libre (Y)  $\wedge$  sujeto (X)  
 $\wedge$  en mesa (Y)

**añadidos:** encima (X, Y)  $\wedge$  brazo-libre  
 $\wedge$  libre (X)

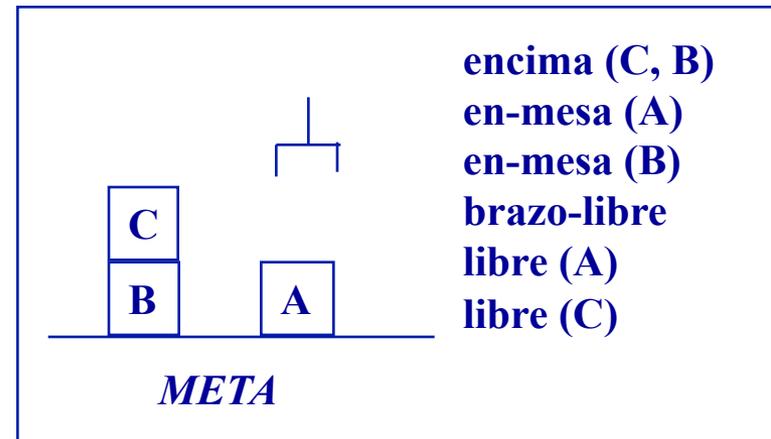
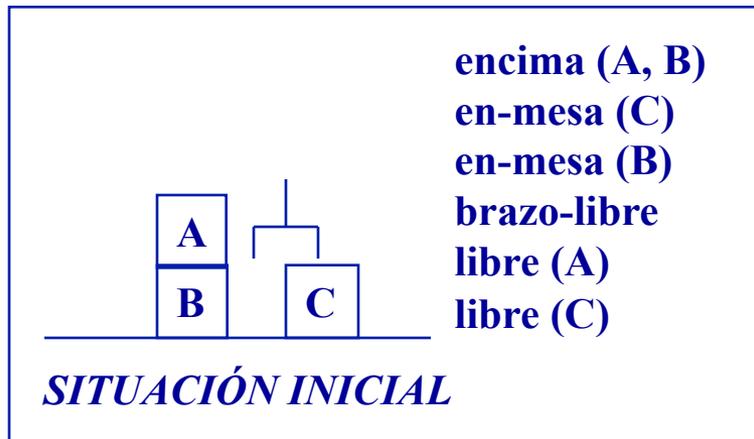
**borrados:** libre (Y)  $\wedge$  sujeto (X)





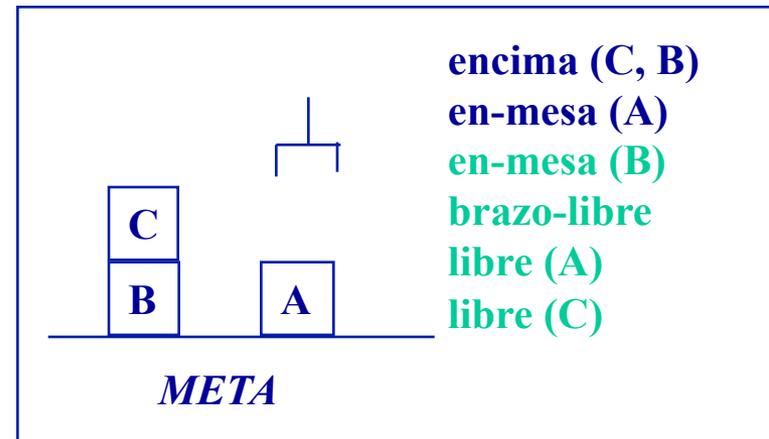
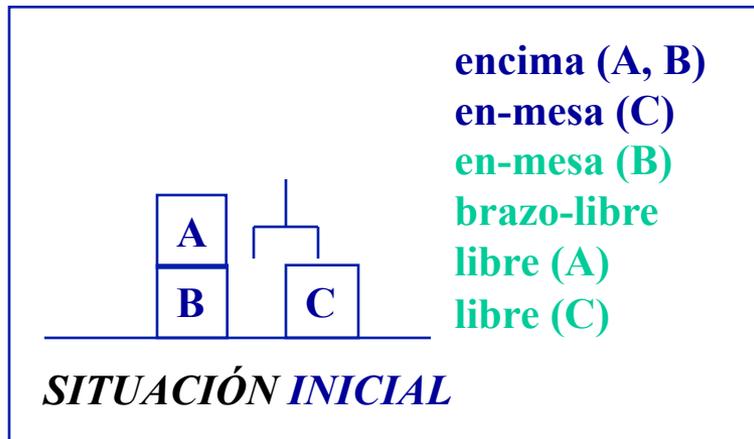
---

# Formalización de STRIPS. Ejemplo de descripción de estados



**Falta un estado, dado que no se da en estas dos representaciones, que sería: sujeto(x)**

# Formalización de STRIPS. Ejemplo de descripción de estados



Al objeto de simplificar la representación y dado que una parte de la META coincide con la SITUACIÓN INICIAL la denominaremos **R**

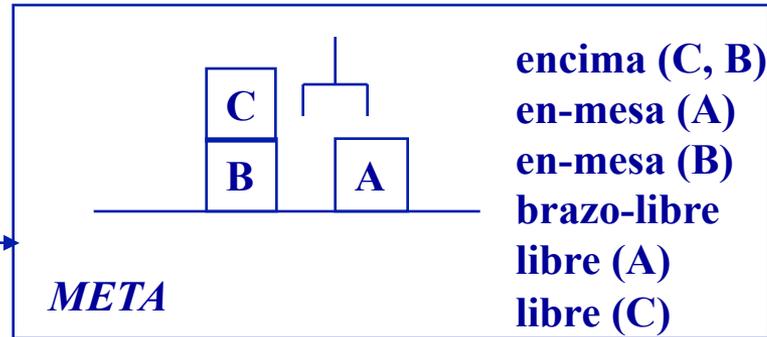
Por el mismo motivo a la conjunción de metas, se le denominará **M**

Así  $M = \text{encima (C, B)} \wedge \text{en-mesa (A)} \wedge R$

---

# Búsqueda del plan en STRIPS

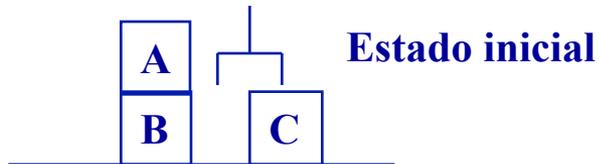
Se parte del estado meta



Se consideran las  
diferencias con respecto al  
estado inicial



$$M = \text{encima (C, B)} \wedge \text{en-mesa (A)} \wedge R$$

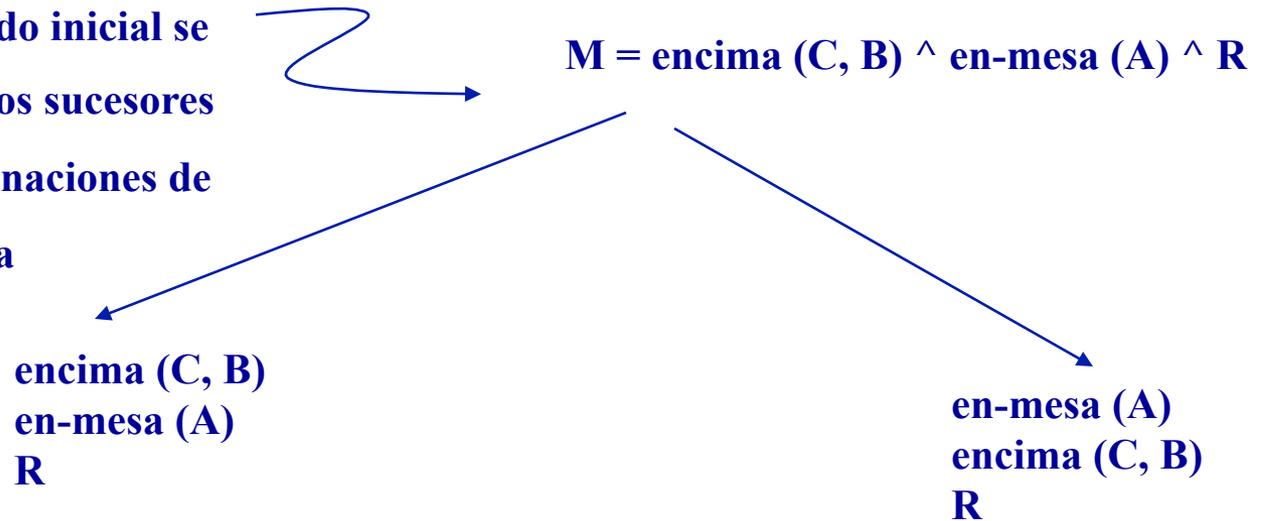


El plan inicial es igual a  $\emptyset$

---

# Búsqueda del plan en STRIPS

Para el nodo inicial se  
crean tantos sucesores  
como ordenaciones de  
metas haya

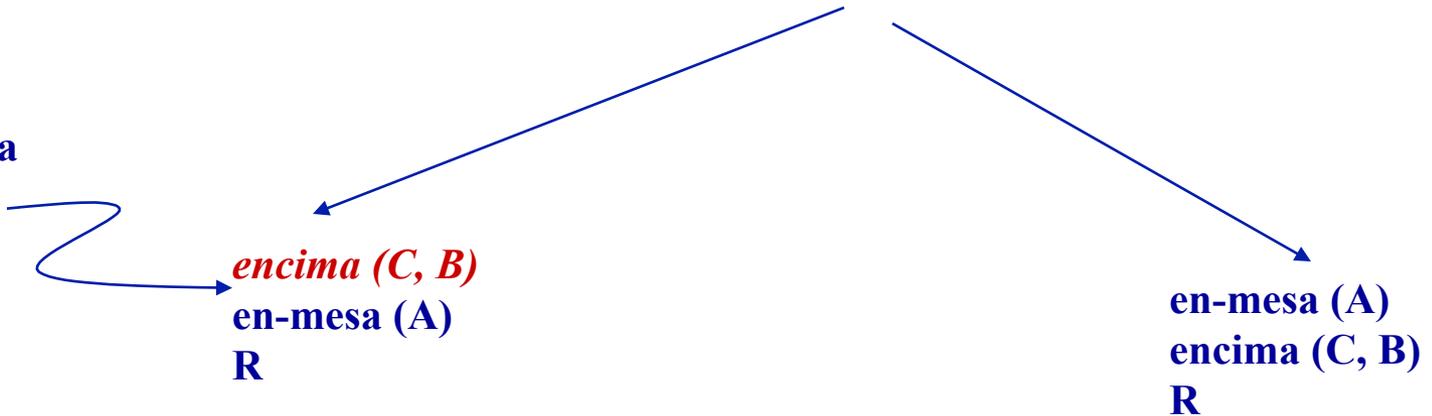


---

# Búsqueda del plan en STRIPS

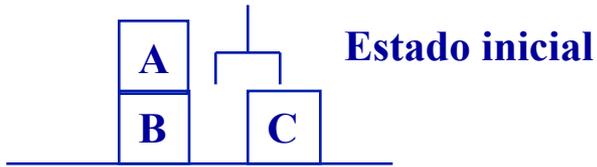
$M = \text{encima}(C, B) \wedge \text{en-mesa}(A) \wedge R$

Se elige una  
rama



---

# Búsqueda del plan en STRIPS



$M = \text{encima}(C, B) \wedge \text{en-mesa}(A) \wedge R$

Se comprueba  
si la meta de  
la cima es  
cierta en el  
estado inicial

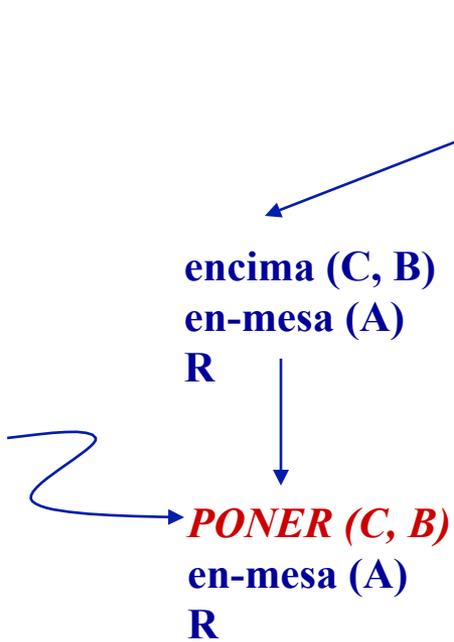
*encima(C, B)*  
en-mesa(A)  
R

---

# Búsqueda del plan en STRIPS

$$M = \text{encima}(C, B) \wedge \text{en-mesa}(A) \wedge R$$

Si la meta de la cima no es cierta, entonces se busca un operador que la añada



# Búsqueda del plan en STRIPS

$$M = \text{encima}(C, B) \wedge \text{en-mesa}(A) \wedge R$$

Se añaden las  
precondiciones  
del operador  
con las distintas  
ordenaciones

encima (C, B)  
en-mesa (A)  
R

PONER (C, B)  
en-mesa (A)  
R

*libre (B)*  
*sujeto (C)*

*libre (B) ^ sujeto (C)*

PONER (C, B)  
en-mesa (A)  
R

*sujeto (C)*  
*libre (B)*

*sujeto (C) ^ libre (B)*

PONER (C, B)  
en-mesa (A)  
R

Como son las  
precondiciones de los  
operadores, cuando se  
cumplen, quedan los  
operadores al descubierto y  
ya se sabe que se pueden  
ejecutar

# Búsqueda del plan en STRIPS

$$M = encima(C, B) \wedge en-mesa(A) \wedge R$$

**encima (C, B)**  
**en-mesa (A)**  
**R**

**en-mesa (A)**  
**encima (C, B)**  
**R**

**PONER (C, B)**  
**en-mesa (A)**  
**R**

Se continúa con  
el estudio de las  
metas

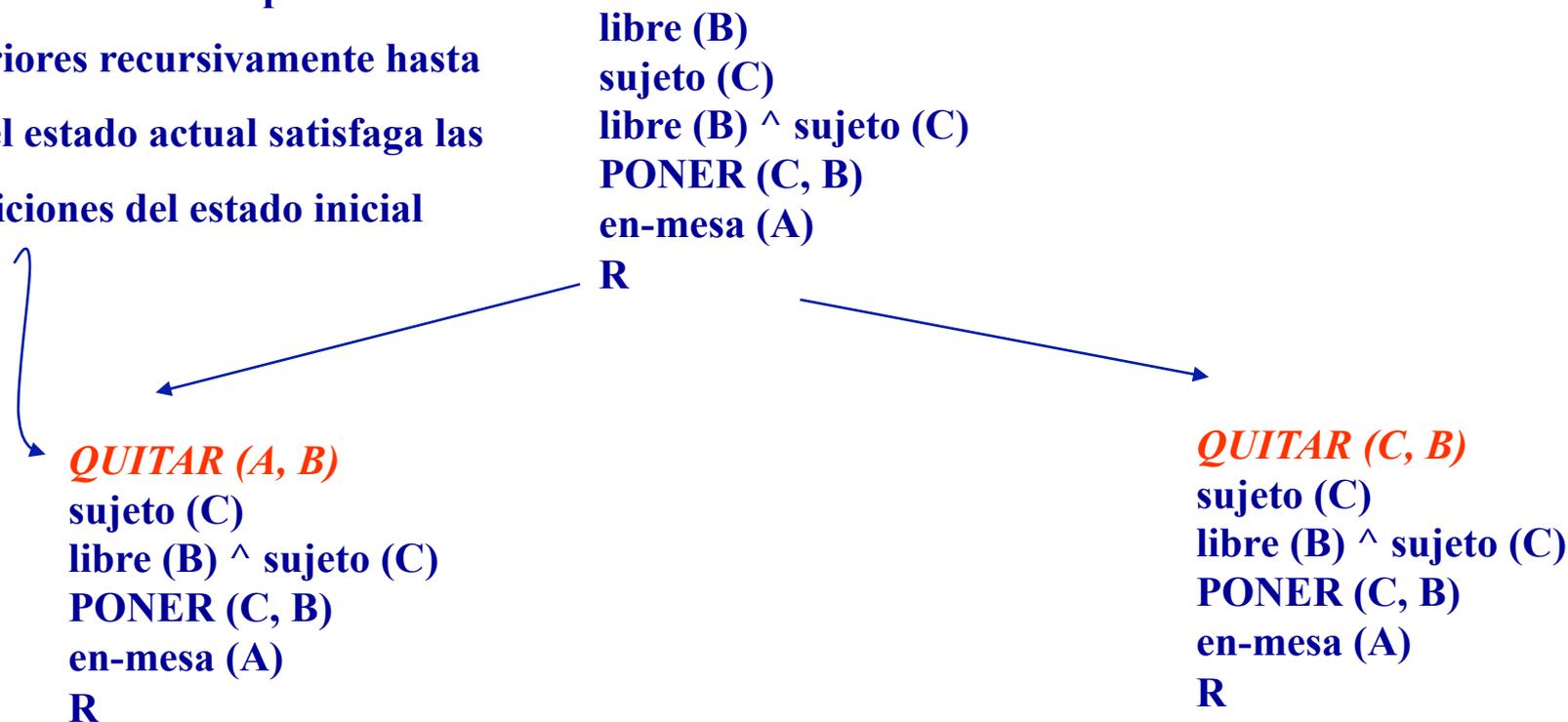
*libre (B)*  
*sujeto (C)*  
*libre (B) ^ sujeto (C)*  
**PONER (C, B)**  
**en-mesa (A)**  
**R**

*sujeto (C)*  
*libre (B)*  
*libre (B) ^ sujeto (C)*  
**PONER (C, B)**  
**en-mesa (A)**  
**R**

---

# Búsqueda del plan en STRIPS

Se llevan a cabo los pasos  
anteriores recursivamente hasta  
que el estado actual satisfaga las  
condiciones del estado inicial



---

# Búsqueda del plan en STRIPS

**QUITAR (A, B)**

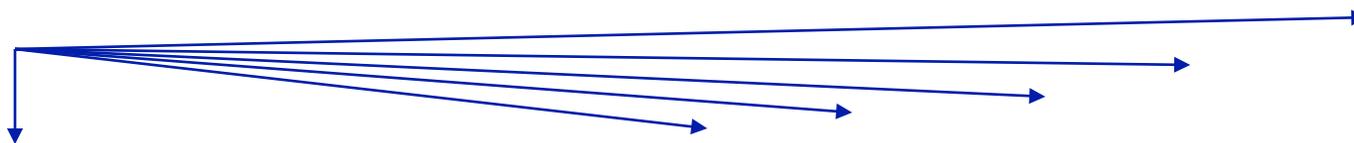
**sujeto (C)**

**libre (B) ^ sujeto (C)**

**PONER (C, B)**

**en-mesa (A)**

**R**



*encima (A, B)*

*libre (A)*

*brazo-libre*

*encima (A, B) ^ libre (A) ^ brazo-libre*

**QUITAR (A, B)**

**sujeto (A)**

**libre (B) ^ sujeto (A)**

**PONER (A, B)**

**en-mesa (A)**

**R**

# Búsqueda del plan en STRIPS

**QUITAR (A, B)**

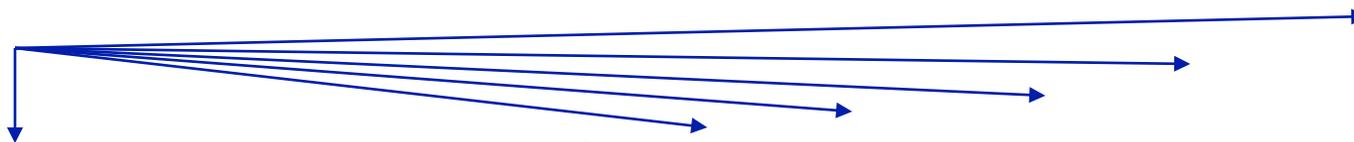
**sujeto (C)**

**libre (B) ^ sujeto (C)**

**PONER (C, B)**

**en-mesa (A)**

**R**



*encima (A, B)*

*libre (A)*

*brazo-libre*

*encima (A, B) ^ libre(A) ^ brazo-libre*

**QUITAR (A, B)**

**sujeto (C)**

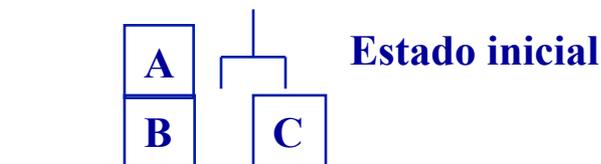
**libre (B) ^ sujeto (C)**

**PONER (C, B)**

**en-mesa (A)**

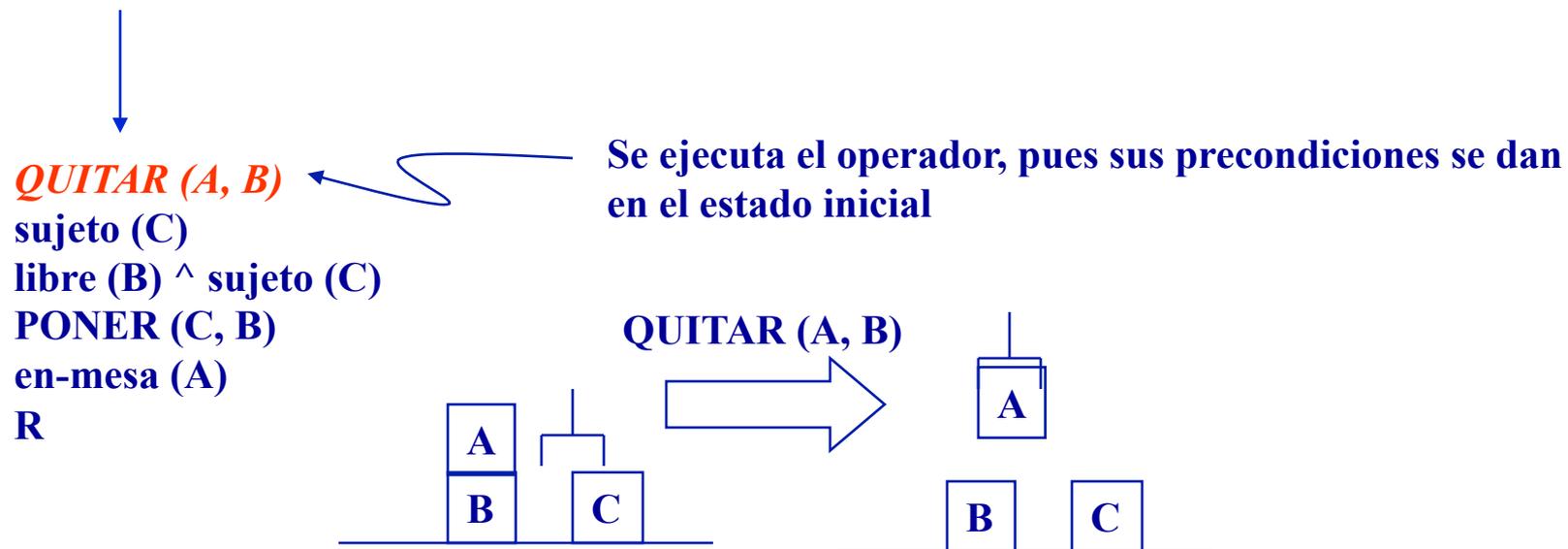
**R**

**Estas condiciones son ciertas en el estado inicial**



---

# Búsqueda del plan en STRIPS



**PLAN = QUITAR (A, B)**

---

# Búsqueda del plan en STRIPS

↓  
**QUITAR (A, B)**

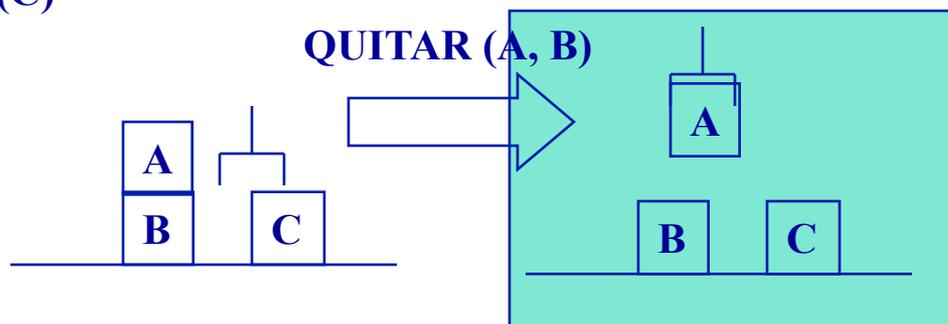
sujeto (C)

libre (B)  $\wedge$  sujeto (C)

PONER (C, B)

en-mesa (A)

R



Estado E2 cuyas condiciones hay que comprobar en los siguientes pasos

**PLAN = QUITAR (A, B)**

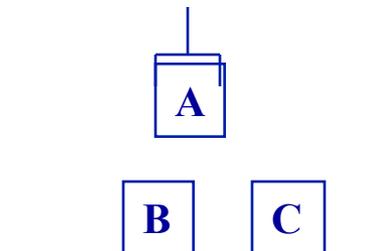
---

# Búsqueda del plan en STRIPS

**QUITAR (A, B)**  
**sujeto (C)**  
**libre (B) ^ sujeto (C)**  
**PONER (C, B)**  
**en-mesa (A)**  
**R**



*sujeto (C)*  
**libre (B) ^ sujeto (C)**  
**PONER (C, B)**  
**en-mesa (A)**  
**R**



**Estado E2**

Como la condición de la cima no está satisfecha en E2,  
se busca el operador que la añada

---

# Búsqueda del plan en STRIPS

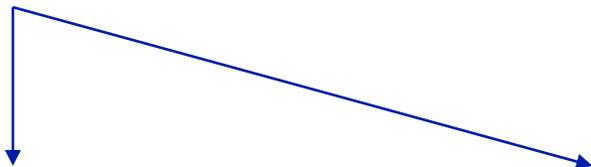
**sujeto (C)**

**libre (B) ^ sujeto (C)**

**PONER (C, B)**

**en-mesa (A)**

**R**



**LEVANTAR (C)**

**libre (B) ^ sujeto (C)**

**PONER (C, B)**

**en-mesa (A)**

**R**

**QUITAR (C, y)**

**libre (B) ^ sujeto (C)**

**PONER (C, B)**

**en-mesa (A)**

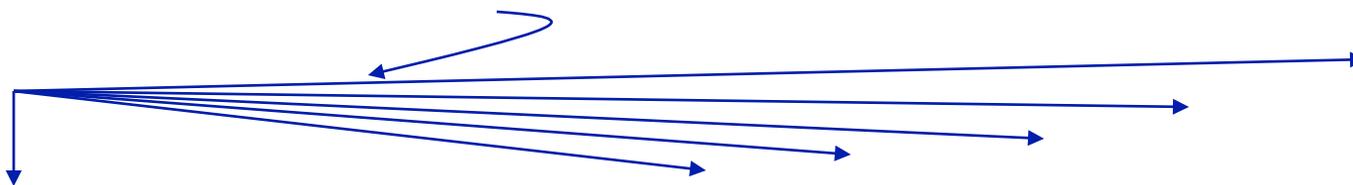
**R**

---

# Búsqueda del plan en STRIPS

**LEVANTAR (C)**  
**libre (B) ^ sujeto (C)**  
**PONER (C, B)**  
**en-mesa (A)**  
**R**

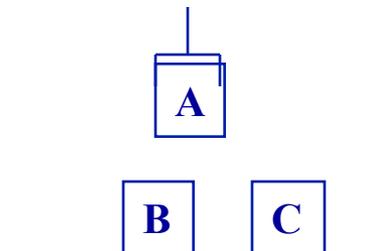
Se obtienen precondiciones, y se ordenan



*en-mesa (C)*  
*libre (C)*  
*brazo-libre*  
*en-mesa (C) ^ libre (C) ^ brazo-libre*  
**LEVANTAR (C)**  
**libre (B) ^ sujeto (C)**  
**PONER (C, B)**  
**en-mesa (A)**  
**R**

# Búsqueda del plan en STRIPS

**LEVANTAR (C)**  
**libre (B) ^ sujeto (C)**  
**PONER (C, B)**  
**en-mesa (A)**  
**R**



**Estado E2**

*en-mesa (C)*  
*libre (C)*  
*brazo-libre*

**Son condiciones satisfechas en E2**

**en-mesa (C) ^ libre (C) ^ brazo-libre**

**LEVANTAR (C)**  
**libre (B) ^ sujeto (C)**  
**PONER (C, B)**  
**en-mesa (A)**  
**R**

**Es una condición no satisfecha en E2**

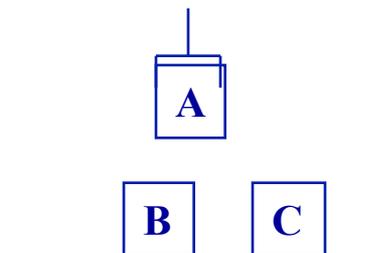
---

# Búsqueda del plan en STRIPS

en-mesa (C)  
libre (C)  
brazo-libre  
en-mesa (C) ^ libre (C) ^ brazo-libre  
LEVANTAR (C)  
libre (B) ^ sujeto (C)  
PONER (C, B)  
en-mesa (A)  
R



**brazo-libre**  
en-mesa (C) ^ libre (C) ^ brazo-libre  
LEVANTAR (C)  
libre (B) ^ sujeto (C)  
PONER (C, B)  
en-mesa (A)  
R



Estado E2

Se buscan operadores que añadan esta condición

---

# Búsqueda del plan en STRIPS

**brazo-libre**

**en-mesa (C) ^ libre (C) ^ brazo-libre**

**LEVANTAR (C)**

**libre (B) ^ sujeto (C)**

**PONER (C, B)**

**en-mesa (A)**

**R**



***DEJAR (A)***

**en-mesa (C) ^ libre (C) ^ brazo-libre**

**LEVANTAR (C)**

**libre (B) ^ sujeto (C)**

**PONER (C, B)**

**en-mesa (A)**

**R**

***PONER (x,y)***

**en-mesa (C) ^ libre (C) ^ brazo-libre**

**LEVANTAR (C)**

**libre (B) ^ sujeto (C)**

**PONER (C, B)**

**en-mesa (A)**

**R**

---

## Búsqueda del plan en STRIPS

**DEJAR (A)**  
**brazo-libre**  
**en-mesa (C) ^ libre (C) ^ brazo-libre**  
**LEVANTAR (C)**  
**libre (B) ^ sujeto (C)**  
**PONER (C, B)**  
**en-mesa (A)**  
**R**



***sujeto(A)***  
**DEJAR (A)**  
**brazo-libre**  
**en-mesa (C) ^ libre (C) ^ brazo-libre**  
**LEVANTAR (C)**  
**libre (B) ^ sujeto (C)**  
**PONER (C, B)**  
**en-mesa (A)**  
**R**

---

# Búsqueda del plan en STRIPS

*sujeto (A)* ←

**DEJAR (A)**

**brazo-libre**

**en-mesa (C) ^ libre (C) ^ brazo-libre**

**LEVANTAR (C)**

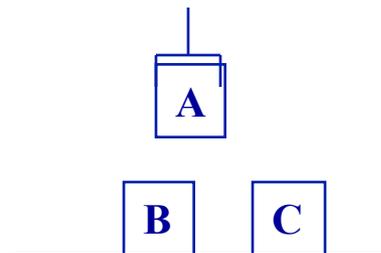
**libre (B) ^ sujeto (C)**

**PONER (C, B)**

**en-mesa (A)**

**R**

Esta meta está en el estado E2



**Estado E2**

**DEJAR (A)**

**brazo-libre**

**en-mesa( C) ^ libre (C) ^ brazo-libre**

**LEVANTAR (C)**

**libre (B) ^ sujeto (C)**

**PONER (C, B)**

**en-mesa (A)**

**R**

---

# Búsqueda del plan en STRIPS

**DEJAR (A)**

brazo-libre

en-mesa (C) ^ libre (C) ^ brazo-libre

LEVANTAR (C)

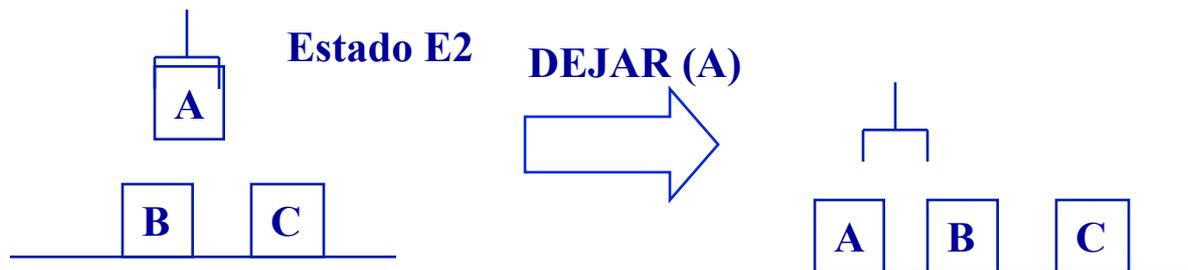
libre (B) ^ sujeto (C)

PONER (C, B)

en-mesa (A)

R

Se ejecuta el operador, pues sus precondiciones se dan en el estado E2



**PLAN = QUITAR(A, B), DEJAR(A)**

# Búsqueda del plan en STRIPS

**DEJAR (A)**

brazo-libre

en-mesa (C) ^ libre (C) ^ brazo-libre

LEVANTAR (C)

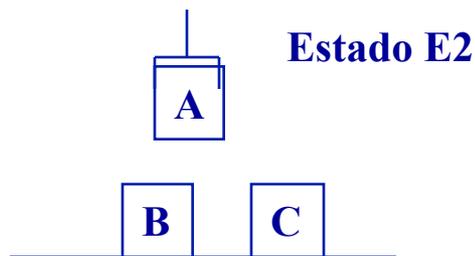
libre (B) ^ sujeto (C)

PONER (C, B)

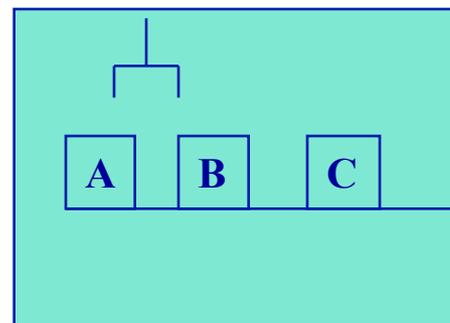
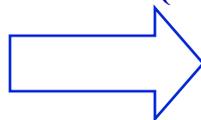
en-mesa (A)

R

Se ejecuta el operador, pues sus precondiciones se dan en el estado E2



DEJAR(A)



Estado E3 cuyas condiciones hay que comprobar en los siguientes pasos

PLAN = QUITAR (A, B), DEJAR (A)

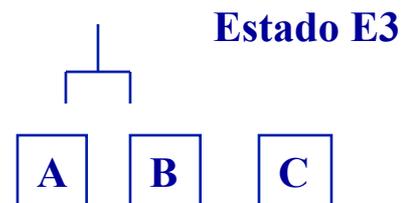
---

# Búsqueda del plan en STRIPS

**DEJAR (A)**  
**brazo-libre**  
**en-mesa (C) ^ libre (C) ^ brazo-libre**  
**LEVANTAR (C)**  
**libre (B) ^ sujeto (C)**  
**PONER (C, B)**  
**en-mesa (A)**  
**R**



*brazo-libre*  
*en-mesa (C) ^ libre (C) ^ brazo-libre* }  
**LEVANTAR (C)**  
**libre (B) ^ sujeto (C)**  
**PONER (C, B)**  
**en-mesa (A)**  
**R**



Las condiciones son ciertas en el estado E3

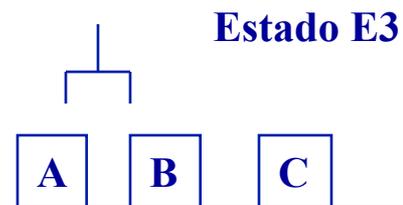
---

# Búsqueda del plan en STRIPS

**brazo-libre**  
**en-mesa (C) ^ libre (C) ^ brazo-libre**  
**LEVANTAR (C)**  
**libre (B) ^ sujeto (C)**  
**PONER (C, B)**  
**en-mesa (A)**  
**R**



***LEVANTAR (C)***  
**libre (B) ^ sujeto (C)**  
**PONER (C, B)**  
**en-mesa (A)**  
**R**



---

## Búsqueda del plan en STRIPS

**LEVANTAR (C)**

libre (B)  $\wedge$  sujeto (C)

PONER (C, B)

en-mesa (A)

R

Se ejecuta el operador, pues sus precondiciones se dan en el estado E3



**PLAN = QUITAR (A, B), DEJAR (A), LEVANTAR (C)**

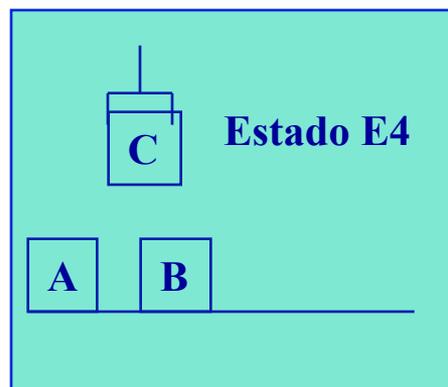
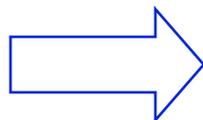
# Búsqueda del plan en STRIPS

**LEVANTAR (C)**  
libre (B) ^ sujeto (C)  
**PONER (C, B)**  
en-mesa (A)  
**R**

Se ejecuta el operador, pues sus precondiciones se dan en el estado E3



LEVANTAR(C)



**Estado E4**  
cuyas condiciones hay que comprobar en los siguientes pasos

**PLAN = QUITAR (A, B), DEJAR (A), LEVANTAR (C)**

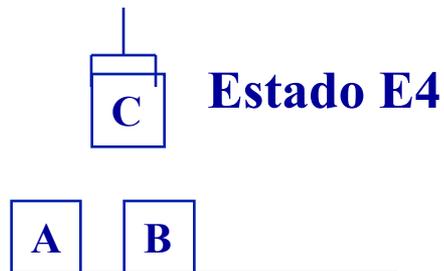
---

## Búsqueda del plan en STRIPS

**LEVANTAR (C)**  
**libre (B) ^ sujeto (C)**  
**PONER (C, B)**  
**en-mesa (A)**  
**R**



**libre (B) ^ sujeto (C)**  
**PONER (C, B)**  
**en-mesa (A)**  
**R**



Las condiciones son ciertas en el estado E4

---

## Búsqueda del plan en STRIPS

**LEVANTAR (C)**  
**libre (B) ^ sujeto (C)**  
**PONER (C, B)**  
**en-mesa (A)**

**R**

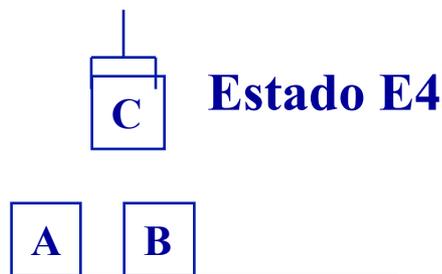


**libre (B) ^ sujeto (C)**  
**PONER (C, B)**  
**en-mesa (A)**

**R**



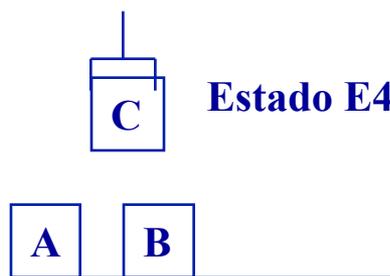
**PONER(C, B)**  
**en-mesa(A)**  
**R**



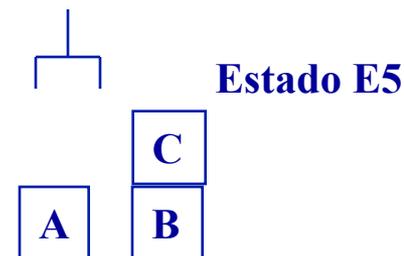
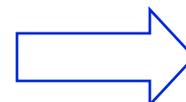
Se cumplen las precondiciones del operador en el estado E4, por tanto, se ejecuta

# Búsqueda del plan en STRIPS

**LEVANTAR (C)**  
**libre (B) ^ sujeto (C)**  
**PONER (C, B)**  
**en-mesa (A)**  
**R**



**PONER (C, B)**



**libre (B) ^ sujeto (C)**  
**PONER (C, B)**  
**en-mesa (A)**  
**R**

**PLAN = QUITAR (A, B), DEJAR (A), LEVANTAR (C), PONER (C, B)**

**PONER (C, B)**  
**en-mesa (A)**  
**R**

Se cumplen las precondiciones del operador en el estado E4, por tanto, se ejecuta

---

# Búsqueda del plan en STRIPS

**PONER (C, B)**  
**en-mesa (A)**

**R**



**en-mesa (A)**

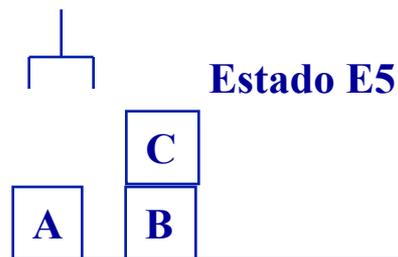
**R**



**∅**

Son metas que están en el estado E5, que es final

Pila vacía, se cumple la condición de fin



**PLAN = QUITAR (A, B), DEJAR (A), LEVANTAR (C), PONER (C, B)**

---

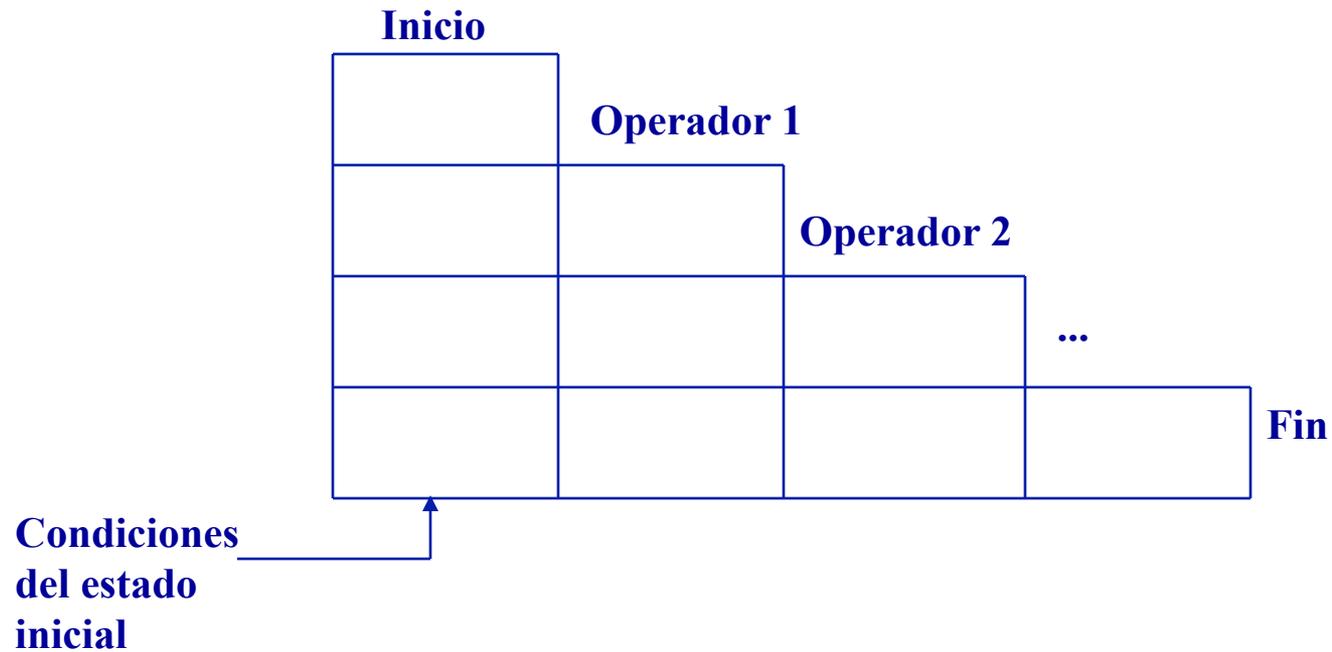
# Problemas de los STRIPS

- 1. Si la situación es compleja, la búsqueda puede ser muy costosa**
- 2. Aun sin ser la situación compleja, puede haber muchas posibilidades**

---

# Aprendizaje de planes en STRIPS

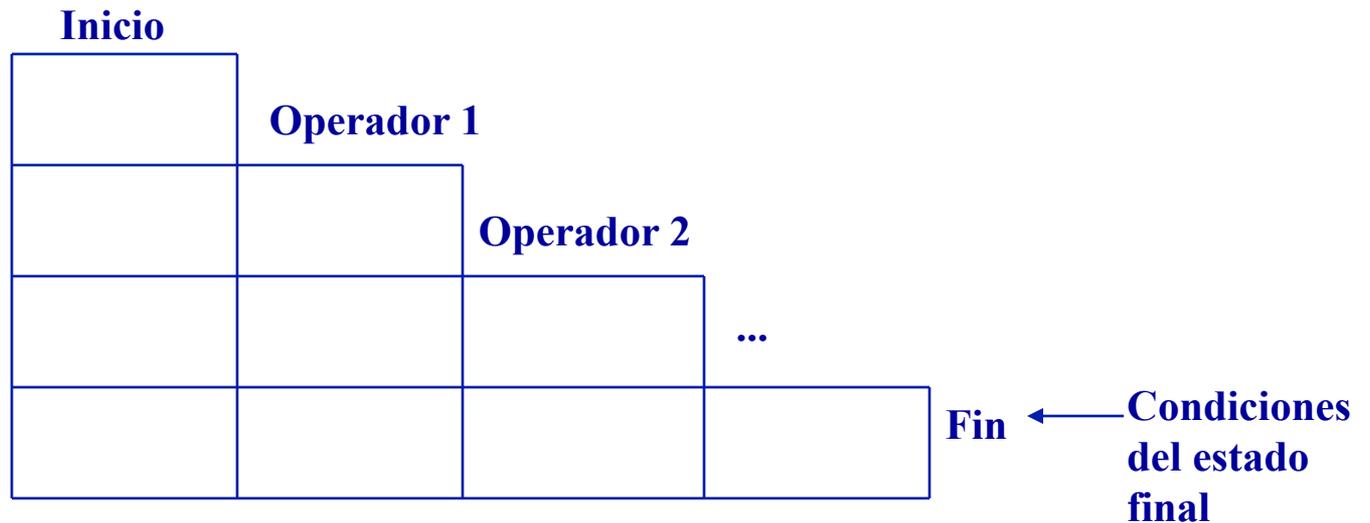
Las secuencias de acciones que se realizan a menudo pueden ser aprendidas mediante una tabla triangular



---

# Aprendizaje de planes en STRIPS

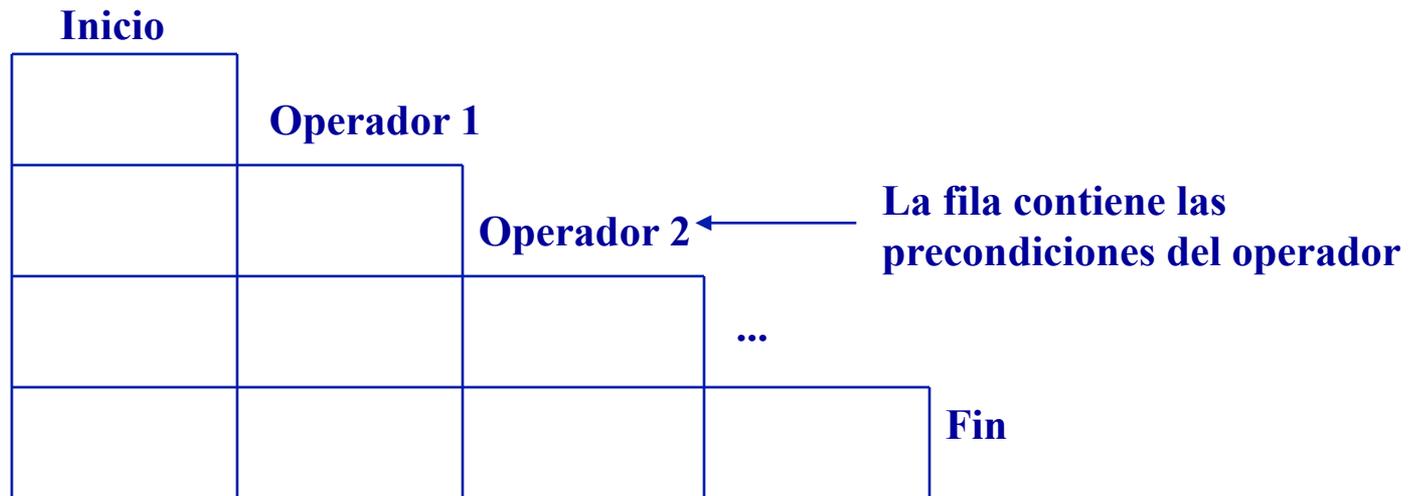
Las secuencias de acciones que se realizan a menudo pueden ser aprendidas mediante una tabla triangular



---

# Aprendizaje de planes en STRIPS

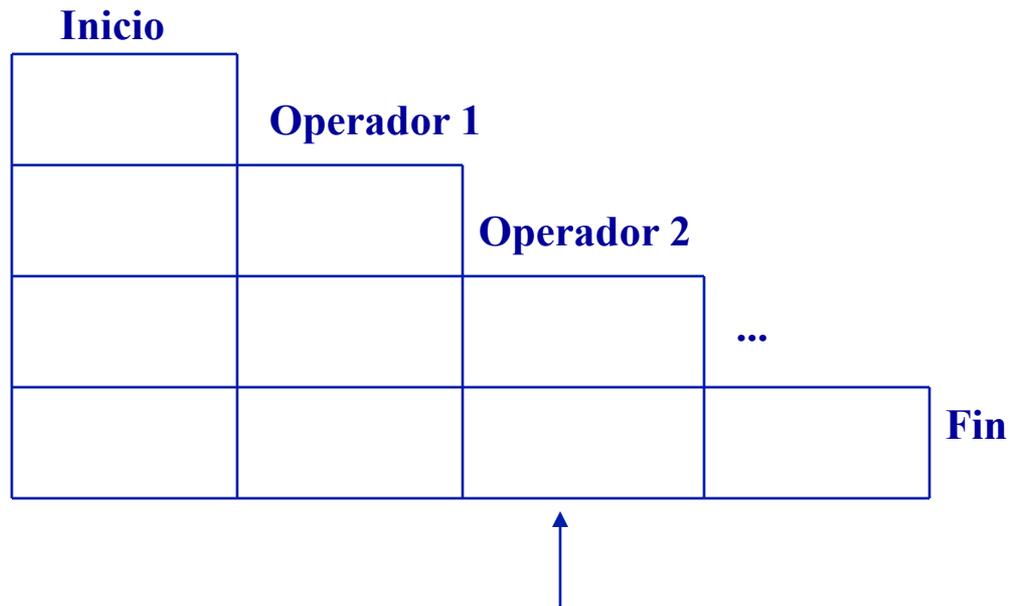
Las secuencias de acciones que se realizan a menudo pueden ser aprendidas mediante una tabla triangular



---

# Aprendizaje de planes en STRIPS

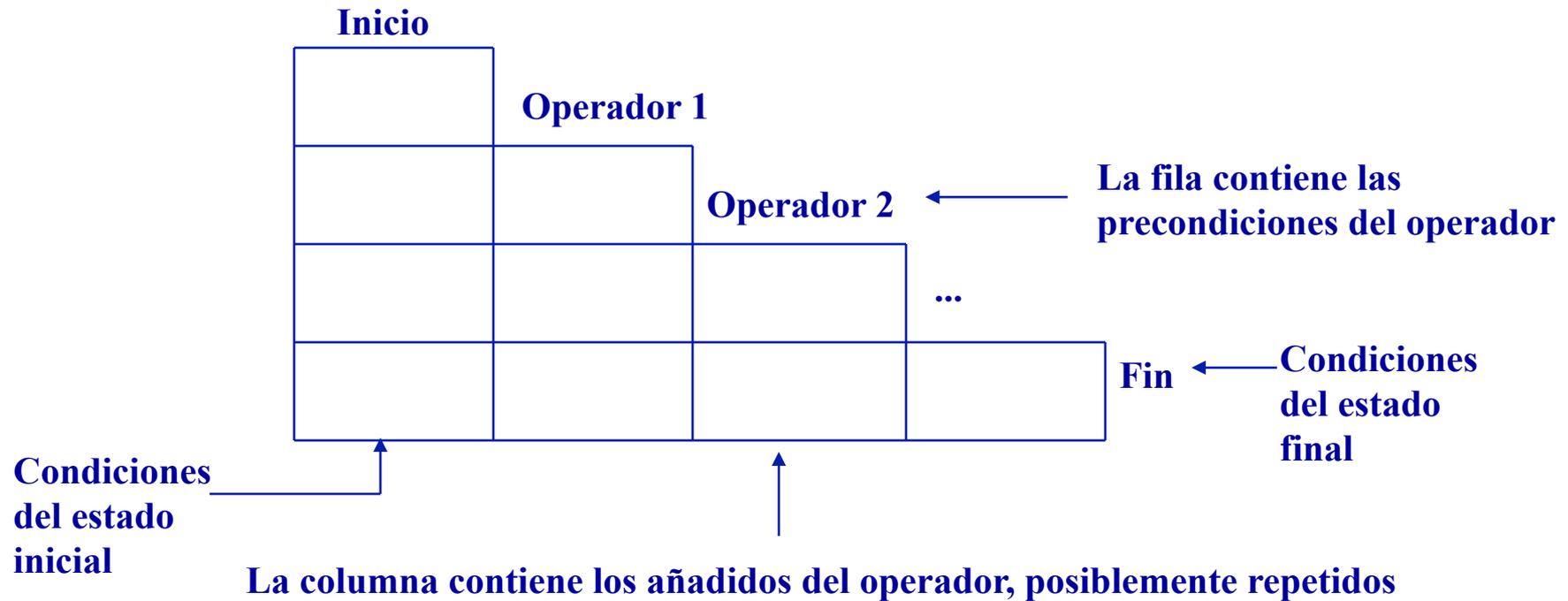
Las secuencias de acciones que se realizan a menudo pueden ser aprendidas mediante una tabla triangular



La columna contiene los añadidos del operador, posiblemente repetidos

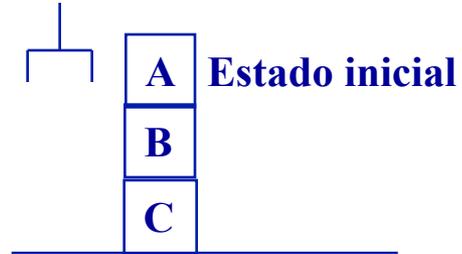
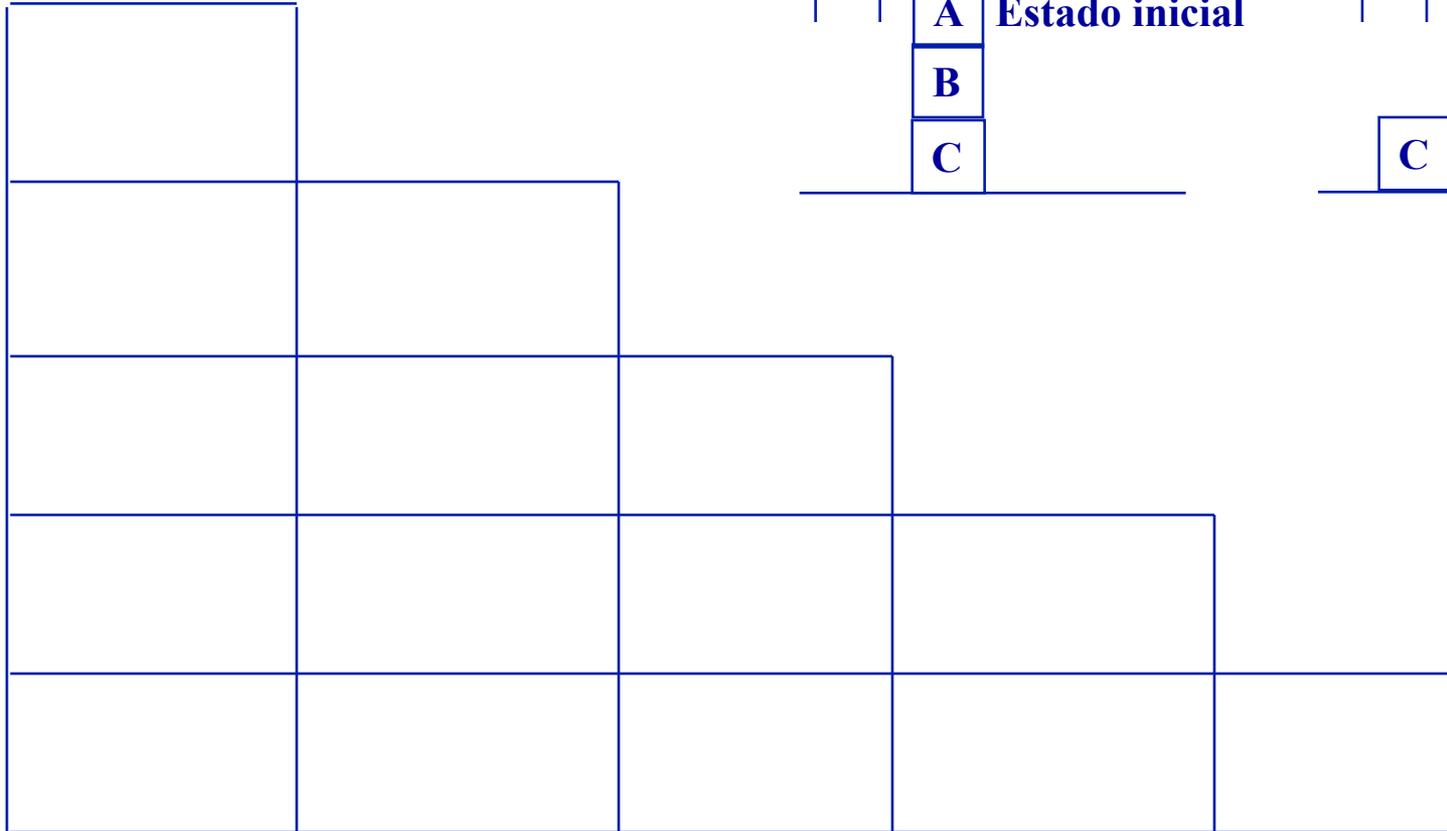
# Aprendizaje de planes en STRIPS

Las secuencias de acciones que se realizan a menudo pueden ser aprendidas mediante una tabla triangular

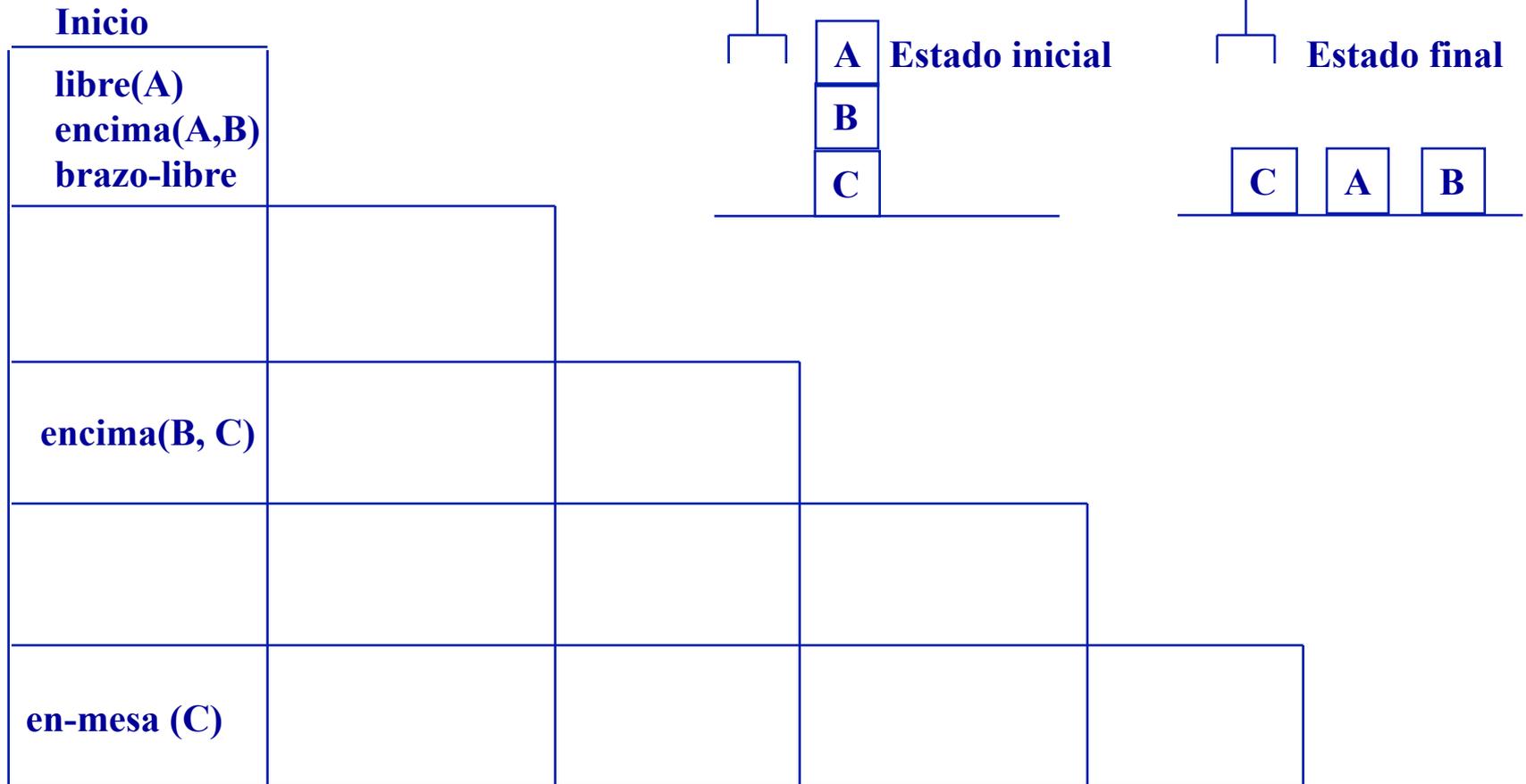


---

# Aprendizaje de planes en STRIPS. Ejemplo

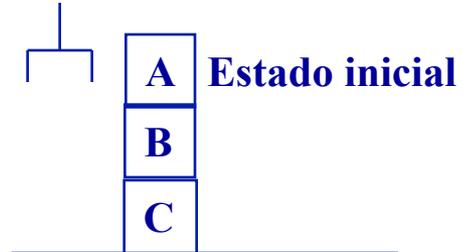


# Aprendizaje de planes en STRIPS. Ejemplo

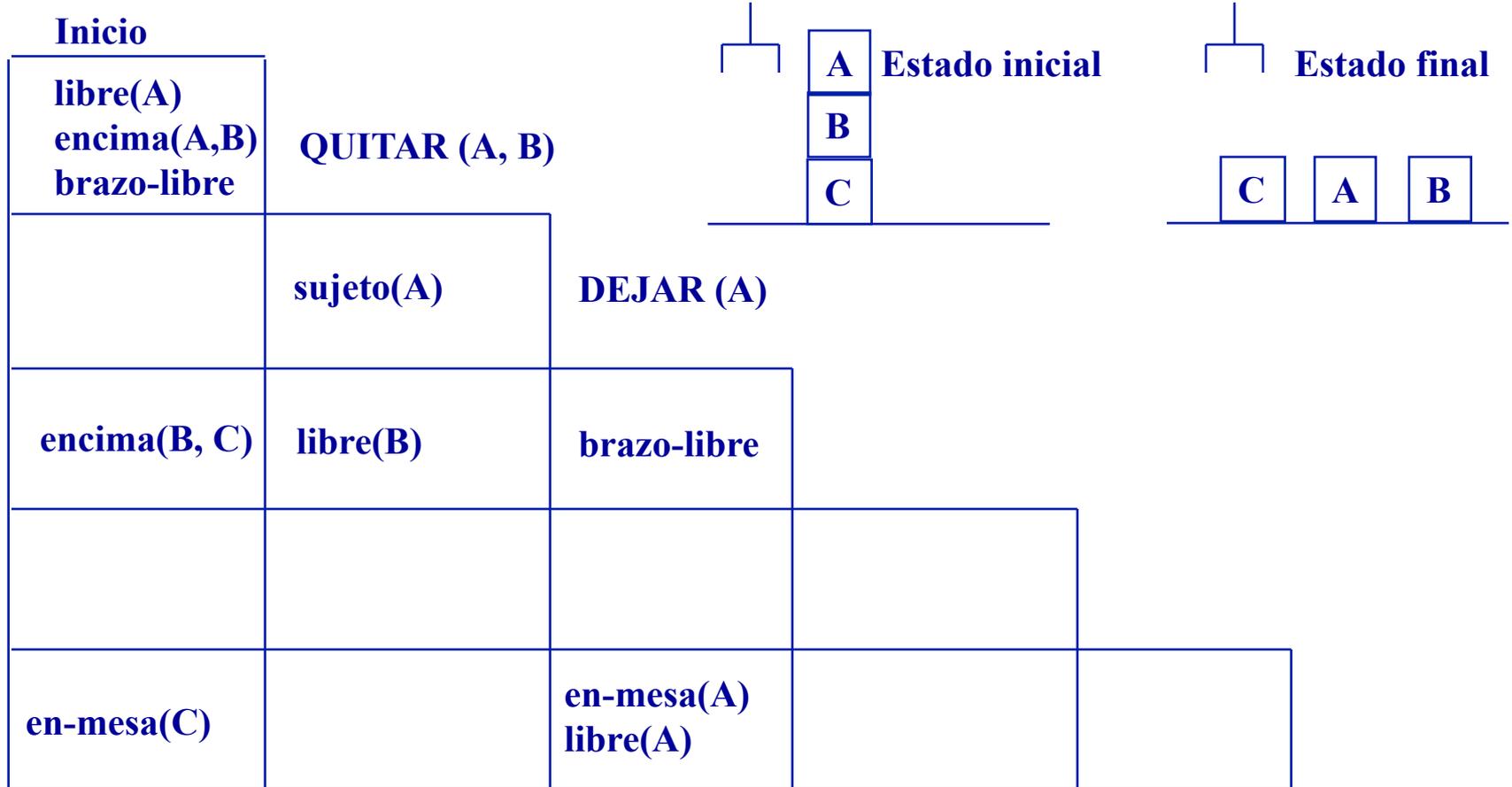


# Aprendizaje de planes en STRIPS. Ejemplo

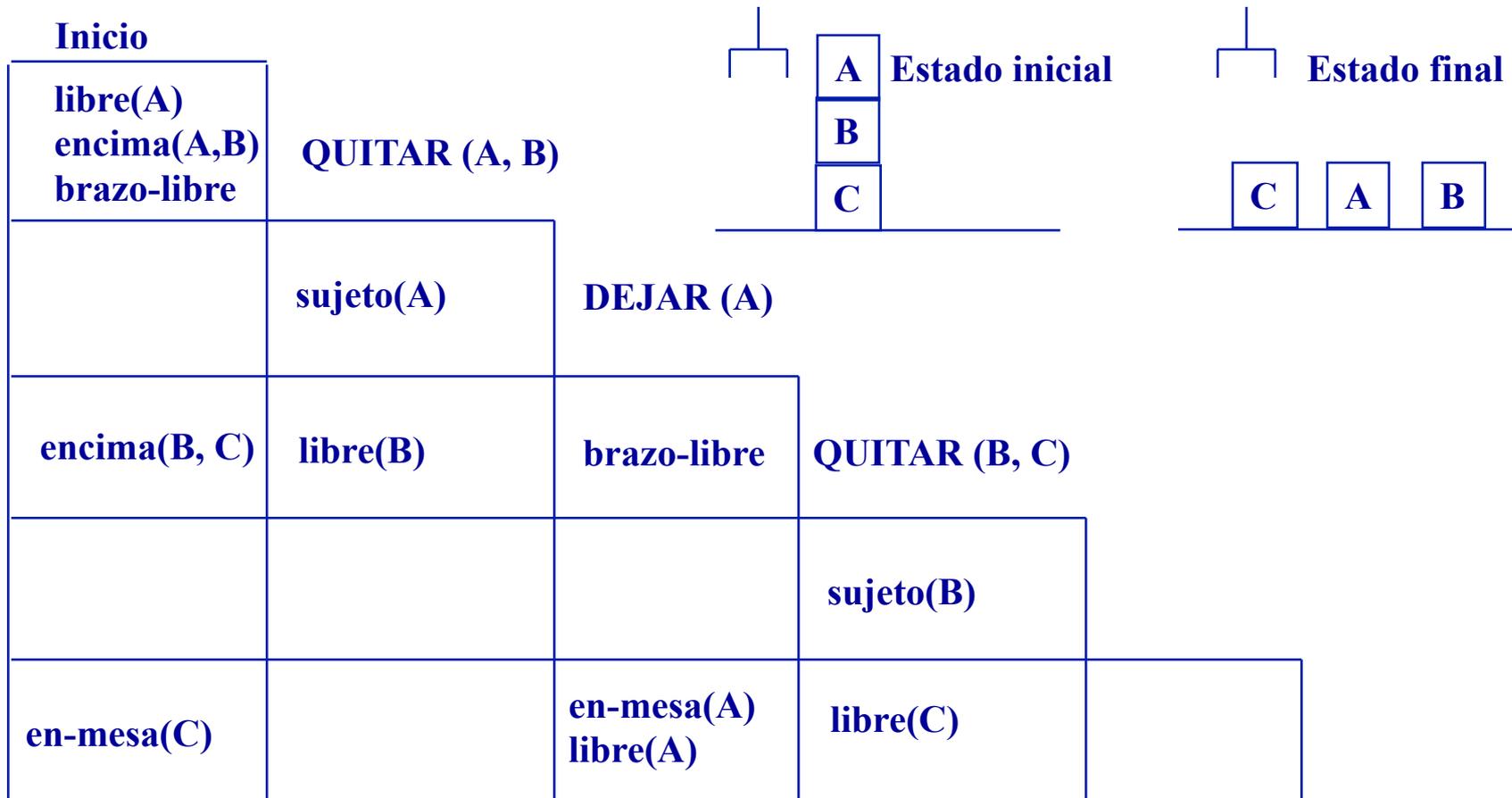
<b>Inicio</b>				
libre(A) encima(A,B) brazo-libre	QUITAR (A, B)			
	sujeto(A)			
encima(B, C)	libre(B)			
en-mesa(C)				



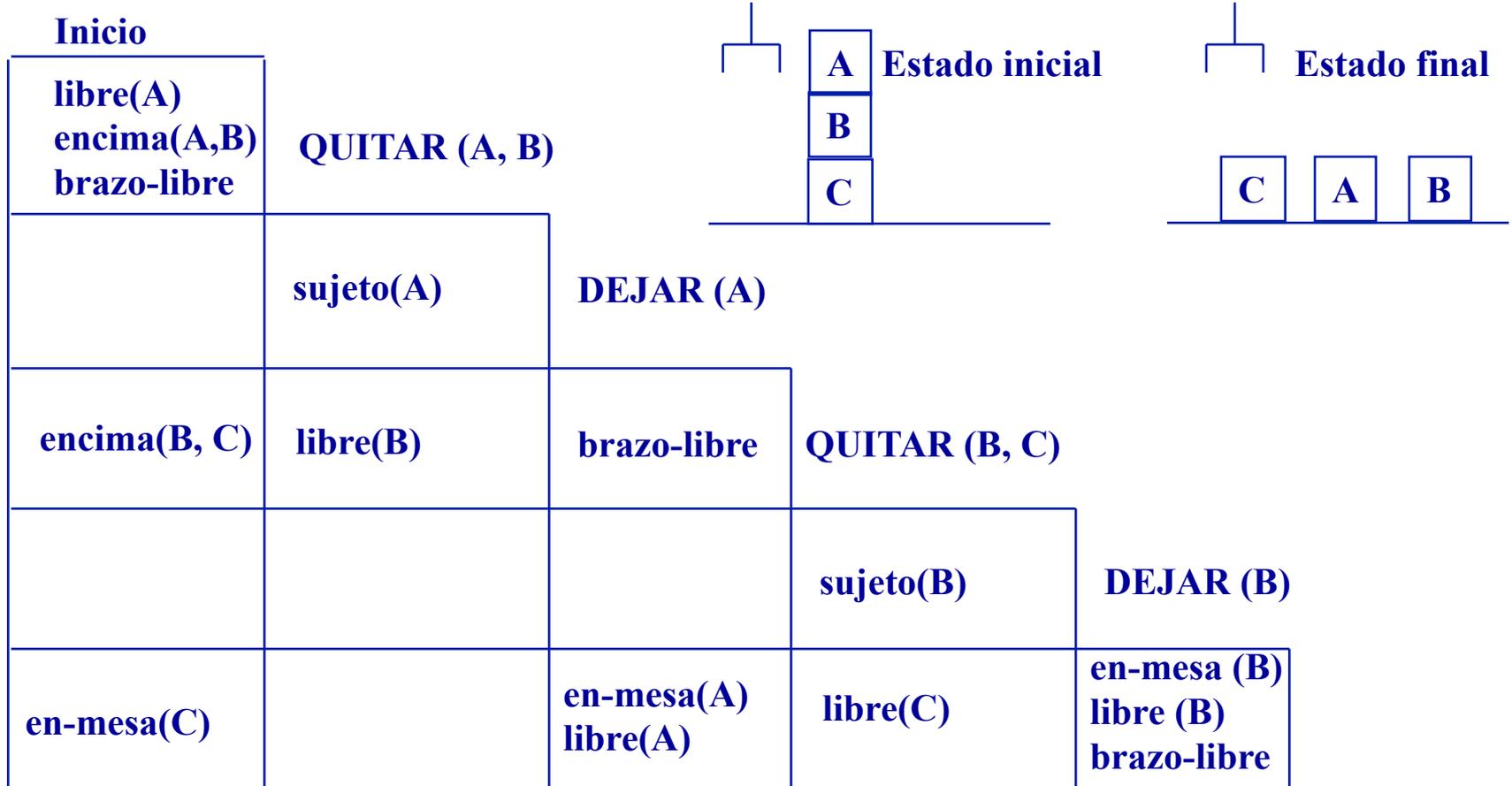
# Aprendizaje de planes en STRIPS. Ejemplo



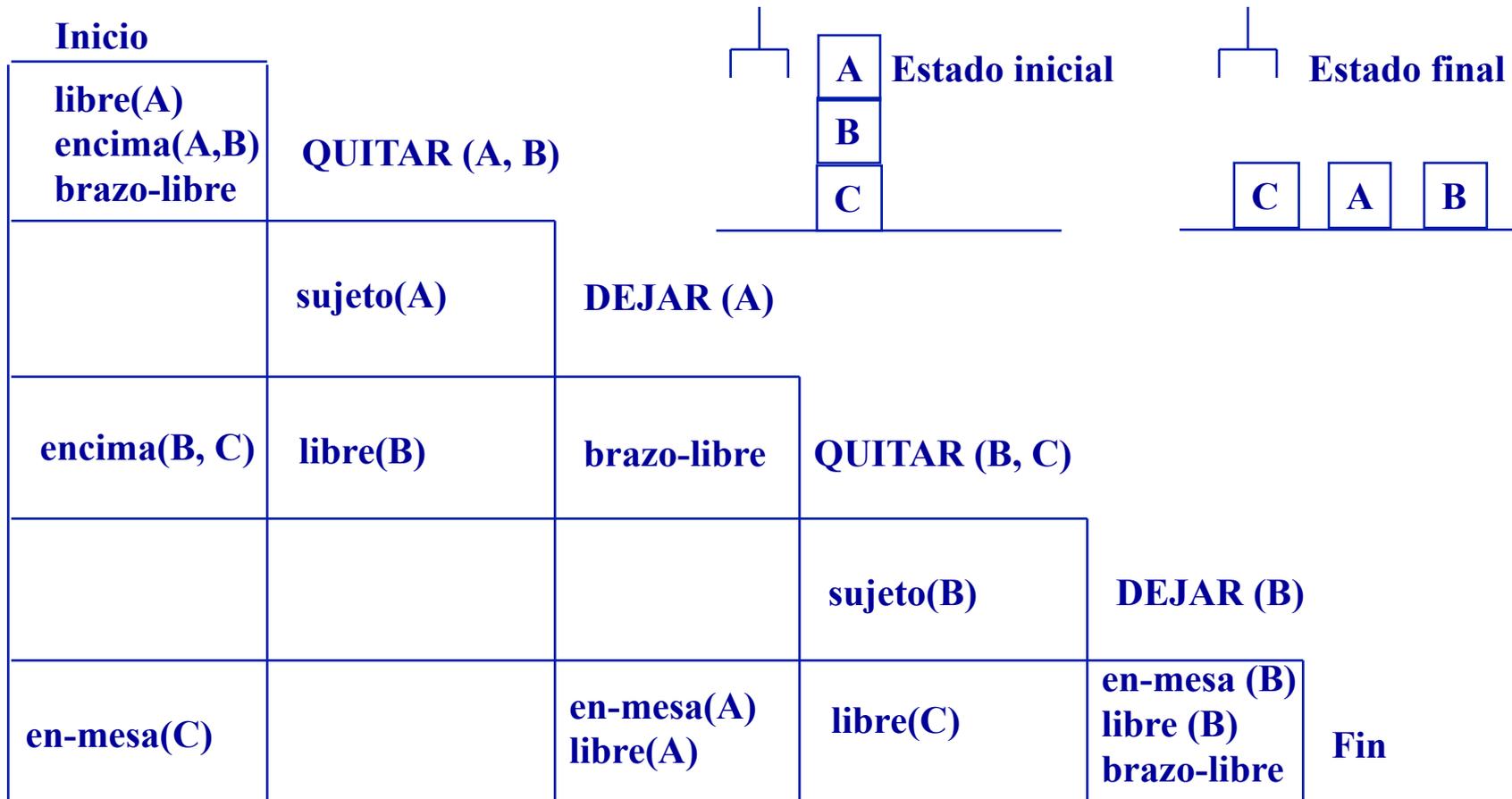
# Aprendizaje de planes en STRIPS. Ejemplo



# Aprendizaje de planes en STRIPS. Ejemplo



# Aprendizaje de planes en STRIPS. Ejemplo



---

**Supóngase que se desea programar varios robots par que sean capaces de resolver determinados problemas de movimientos de cajas entre despachos (que pueden contener más de una mesa) de una misma planta de un edificio. Las acciones que puede realizar cada robots son:**

**LEVANTAR UNA CAJA DE UNA MESA, si esta en la misma oficina que ella y está libre (cada robot sólo puede llevar una caja en cada instante).**

---

**DEJAR UNA CAJA EN UNA MESA, se supone que las mesas tienen capacidad infinita.**

**IR DE UNA OFICINA A OTRA, si hay conexión entre ellas.**

**PASARLE UNA CAJA A OTRO ROBOT, si el otro esta libre y están en la misma oficina.**

**Formalizar dicho dominio en STRIPS**

---

## **Levantar-caja (r,c,o,m)**

**Precondiciones:** libre (r), en-oficina (r,o), en-oficina (m,o), en-mesa (c,m)

**Añadidos:** sujeta (r,c)

**Borrados:** libre (r), en-mesa (c,m)

---

**Dejar-caja (r,c,o,m)**

**Precondiciones: sujeta (r,c), en-oficina (r,o), en-oficina (m,o)**

**Añadidos: libre (r), en-mesa (c,m)**

**Borrados: sujeta (r,c)**

---

**Ir-oficina (r,o,o')**

**Precondiciones: en-oficina (r,o), conexión (o,o')**

**Añadidos: en-oficina (r,o')**

**Borrados: en-oficina (r,o)**

---

**Pasar-caja (r,c,r',o)**

**Precondiciones:** sujeta (r,c), libre (r'), en-oficina (r,o), en-oficina (r',o)

**Añadidos:** libre (r), sujeta (r',c)

**Borrados:** libre (r'), sujeta (r,c)

---

**Una empresa dispone de  $N$  satélites meteorológicos, cada uno de ellos dotados de  $M$  instrumentos (cámaras o medidores de precisión). Diariamente deben obtenerse imágenes y/o mediciones de  $G$  objetivos. Para poder tomar una imagen o medición, el satélite debe tener el instrumento adecuado, calibrado (el cual se borrara después de la acción) y apuntando al objetivo. Por tanto, los satélites disponen de controles para girar el satélite, apuntar a un objetivo (descrito por coordenadas en el espacio), calibrar cada instrumento, y tomar la imagen o medición.**

**Las imágenes pueden tomarse con tres resoluciones: alta, media y baja. Los satélites pueden estar dotados de cámaras de resolución alta (pueden tomar imágenes en cualquier resolución) o baja resolución (pueden tomar imágenes de media y baja resolución). Las mediciones de precisión pueden ser de tres tipos de señales: A, B y C. Los medidores son de tres tipos, uno para cada tipo de señal.**

**Se desea construir un sistema automático que controle los satélites para tomar imágenes o mediciones de  $G$  objetivos diarios, teniendo en cuenta que los objetivos vienen descritos por las coordenadas del espacio donde están (tres parámetros), e información sobre el objetivo (imagen y tipo de imagen o medición y tipo de medición). Al comienzo de cada día se conoce donde apunta cada satélite (mediante coordenadas). Formalizar dicho objetivo según STRIPS:**

---

**Una empresa dispone de  $N$  satélites meteorológicos, cada uno de ellos dotados de  $M$  instrumentos (cámaras o medidores de precisión). Diariamente deben obtenerse imágenes y/o mediciones de  $G$  objetivos. Para poder tomar una imagen o medición, el satélite debe tener el instrumento adecuado, calibrado (el cual se borrara después de la acción) y apuntando al objetivo. Por tanto, los satélites disponen de controles para girar el satélite, apuntar a un objetivo (descrito por coordenadas en el espacio), calibrar cada instrumento, y**

---

---

**tomar la imagen o medición.**

**Las imágenes pueden tomarse con tres resoluciones: alta, media y baja. Los satélites pueden estar dotados de cámaras de resolución alta (pueden tomar imágenes en cualquier resolución) o baja resolución (pueden tomar imágenes de media y baja resolución). Las mediciones de precisión pueden ser de tres tipos de señales: A, B y C. Los medidores son de tres tipos, uno para cada tipo de señal.**

---

**Se desea construir un sistema automático que controle los satélites para tomar imágenes o mediciones de G objetivos diarios, teniendo en cuenta que los objetivos vienen descritos por las coordenadas del espacio donde están (tres parámetros), e información sobre el objetivo (imagen y tipo de imagen o medición y tipo de medición). Al comienzo de cada día se conoce donde apunta cada satélite (mediante coordenadas). Formalizar dicho objetivo según STRIPS:**

---

## **Girar (s,p,p1)**

**Precondiciones:** apunta (s,p), posición (p1),  $p \neq p1$

**Añadidos:** apunta (s,p1)

**Borrados:** apunta (s,p)

---

**Calibrar (s,i)**

**Precondiciones: en (s,i),  $\sim$ calibrado (i)**

**Añadidos: calibrado (i)**

**Borrados:**

---

## **Tomar-imagen-alta (s,c,p,r)**

**Precondiciones:** en (s,c), resolución-cámara (c,alta), resolución (r), apunta (s,p), calibrado (i)

**Añadidos:** imagen (p,r)

**Borrados:** calibrado (c)

---

## **Tomar-imagen-baja (s,c,p,r)**

**Precondiciones:** en (s,c), resolución-cámara (c,baja), resolución (r),  $r \neq \text{alta}$ , apunta (s,p), calibrado (i)

**Añadidos:** imagen (p,r)

**Borrados:** calibrado (i)

---

## **Medir (s,m,p,r)**

**Precondiciones:** en (s,m), miden (m,r), apunta (s,p), calibrado (i)

**Añadidos:** medición (p,r)

**Borrados:** calibrado (i)