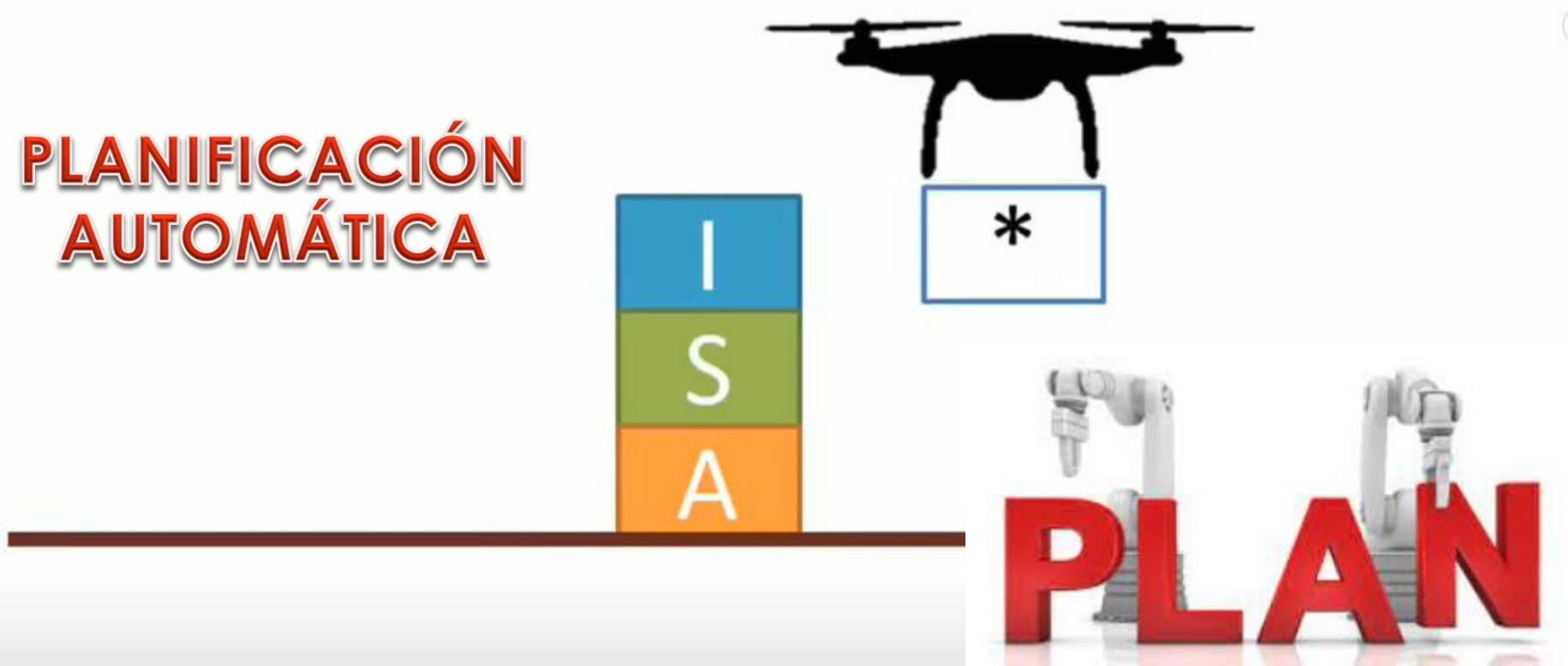


PLANIFICACIÓN AUTOMÁTICA



Cómo un agente puede aprovechar la estructura de un problema para construir planes de acción complejos de manera eficiente.

PLANIFICACIÓN



- Se llama **Planificación** al proceso de búsqueda y articulación de una **secuencia de acciones que permitan alcanzar un objetivo.**
- Se ocupa de ampliar el estudio a problemas de planificación complejos que no pueden ser abordados mediante los enfoques vistos hasta ahora.
- **Planificación clásica:** Tiene en cuenta entornos completamente observables, deterministas, finitos, estáticos y discretos.

LENGUAJE

(REPRESENTACIÓN DE ESTADOS, ACCIONES Y OBJETIVOS)



- **Representación de estados**: los planificadores descomponen el mundo en términos de condiciones lógicas y representan un estado como una **secuencia de literales positivos conectados**.
- **Ej: Pobre \wedge Desconocido**
- **Ej: En(Avión1, Mendoza) \wedge En(Avión2, BsAs)**

LENGUAJE

(REPRESENTACIÓN DE ESTADOS, ACCIONES Y OBJETIVOS)

- Los literales deben ser simples y sin dependencias funcionales.

$En(x,y)$ o $En(Padre(Fred), Sydney)$ no se permiten.



- Hipótesis de un mundo cerrado: todas las condiciones que no se mencionan en un estado se asume que son falsas.



HIPÓTESIS DE MUNDO CERRADO

- La hipótesis de mundo cerrado (closed world assumption) surge de la teoría de bases de datos. Esta suposición establece que todo lo que es relevante en el mundo ha sido especificado. Por lo tanto, esto permite a un agente asumir de forma “segura” que un hecho es falso si no puede inferir que es verdadero.
- En esencia, ésta asume que toda la información positiva relevante ha sido especificada y cualquier otro hecho no especificado se asume por defecto que es falso



LENGUAJE

- **Representación de objetivos:** un objetivo es un estado parcialmente especificado, representado como una secuencia de literales positivos y simples.

Un estado proposicional **satisface** un objetivo g si s contiene todos sus elementos en g (y posiblemente otros)

Ej: Rico \wedge Famoso \wedge Miserable satisface

Rico \wedge Famoso objetivo



LENGUAJE



- **Representación de acciones:** una acción es especificada en términos de las **precondiciones** que deben cumplirse antes de ser ejecutada y de las consecuencias que se siguen cuando se ejecuta.

- Ej: **Acción** **volar(p,desde,hasta),**
 PRECOND: $\text{En}(p,\text{desde}) \wedge \text{avión}(p)$

EFEECTO: $\neg \text{En}(p,\text{desde}) \wedge \text{En}(p,\text{hasta}) \quad)$

esquema de acción



ESQUEMA DE ACCIÓN

- El **nombre de una acción y la lista de parámetros** de los que depende la acción.

Acción (volar (p,desde,hasta),

- La **precondición** es la unión de literales positivos sin dependencia funcional estableciendo lo que debe ser verdad en un estado antes de que una acción sea ejecutada. Todas las variables en las precondiciones también deben aparecer en la lista de parámetros de acción.

PRECOND: $En(p,desde) \wedge avión(p) \wedge aeropuerto(desde) \wedge aeropuerto(hasta)$

- El **efecto** es la unión de literales sin dependencia funcional describiendo cómo cambia el estado cuando se ejecuta la acción.

EFEECTO: $\neg En(p,desde) \wedge En(p,hasta)$

Las variables deben pertenecer a la lista de parámetros de la acción.

SOLUCIÓN DE UN PROBLEMA DE PLANIFICACIÓN



- Se define como una secuencia de acciones que, ejecutada en el estado inicial da como resultado un estado final que satisface el objetivo.

STRIPS

(STANDFORD RESEARCH INSTITUTE PROBLEM SOLVER)



Standord Research Institute
Corporate Headquarters, Silicon Valley
<https://www.sri.com/>



Ejemplo: transporte de carga aéreo

Init($At(C_1, SFO) \wedge At(C_2, JFK) \wedge At(P_1, SFO) \wedge At(P_2, JFK)$
 $\wedge Cargo(C_1) \wedge Cargo(C_2) \wedge Plane(P_1) \wedge Plane(P_2)$
 $\wedge Airport(JFK) \wedge Airport(SFO)$)
Goal($At(C_1, JFK) \wedge At(C_2, SFO)$)
Action(*Load*(c, p, a),
 PRECOND: $At(c, a) \wedge At(p, a) \wedge Cargo(c) \wedge Plane(p) \wedge Airport(a)$
 EFFECT: $\neg At(c, a) \wedge In(c, p)$)
Action(*Unload*(c, p, a),
 PRECOND: $In(c, p) \wedge At(p, a) \wedge Cargo(c) \wedge Plane(p) \wedge Airport(a)$
 EFFECT: $At(c, a) \wedge \neg In(c, p)$)
Action(*Fly*($p, from, to$),
 PRECOND: $At(p, from) \wedge Plane(p) \wedge Airport(from) \wedge Airport(to)$
 EFFECT: $\neg At(p, from) \wedge At(p, to)$)



Figure 11.1 A PDDL description of an air cargo transportation planning problem.



EJEMPLO: RUEDA DE REPUESTO

Init(*Tire*(*Flat*) \wedge *Tire*(*Spare*) \wedge *At*(*Flat*,*Axle*) \wedge *At*(*Spare*,*Trunk*))
Goal(*At*(*Spare*,*Axle*))
Action(*Remove*(*obj*,*loc*),
 PRECOND: *At*(*obj*,*loc*)
 EFFECT: \neg *At*(*obj*,*loc*) \wedge *At*(*obj*,*Ground*))
Action(*PutOn*(*t*, *Axle*),
 PRECOND: *Tire*(*t*) \wedge *At*(*t*,*Ground*) \wedge \neg *At*(*Flat*,*Axle*) \wedge \neg *At*(*Spare*,*Axle*)
 EFFECT: \neg *At*(*t*,*Ground*) \wedge *At*(*t*,*Axle*))
Action(*LeaveOvernight*,
 PRECOND:
 EFFECT: \neg *At*(*Spare*,*Ground*) \wedge \neg *At*(*Spare*,*Axle*) \wedge \neg *At*(*Spare*,*Trunk*)
 \wedge \neg *At*(*Flat*,*Ground*) \wedge \neg *At*(*Flat*,*Axle*) \wedge \neg *At*(*Flat*, *Trunk*))



Figure 11.2 The simple spare tire problem.

EJ: EL MUNDO DE LOS BLOQUES

A

B

C

A

B

C

Iniciar ($\text{Sobre}(A, \text{Tablero}) \wedge \text{Sobre}(B, \text{Tablero}) \wedge \text{Sobre}(C, \text{Tablero})$
 $\wedge \text{Bloque}(A) \wedge \text{Bloque}(B) \wedge \text{Bloque}(C)$
 $\wedge \text{Despejado}(A) \wedge \text{Despejado}(B) \wedge \text{Despejado}(C)$

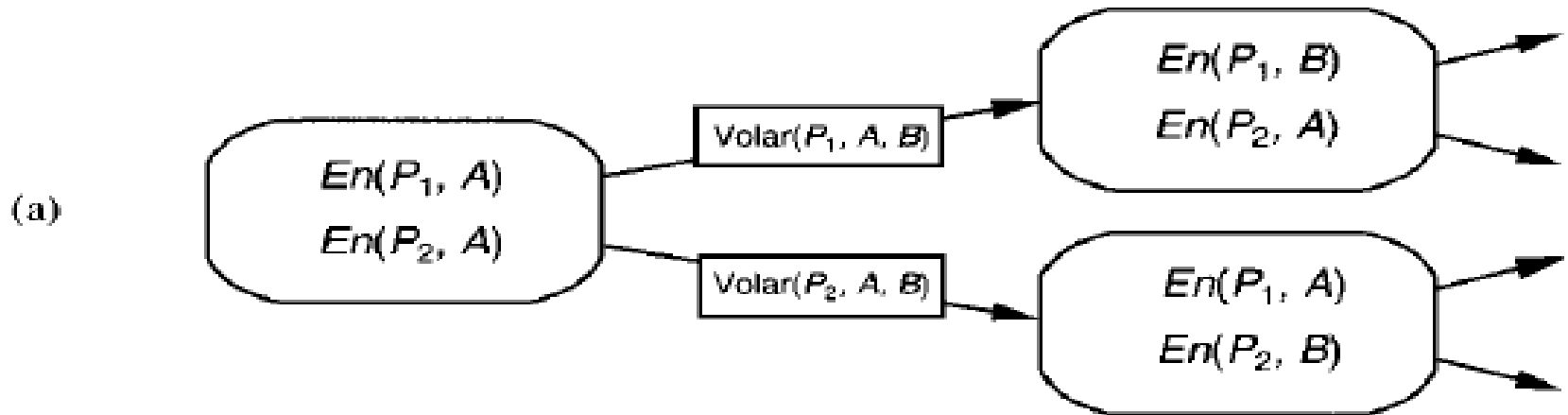
Objetivo ($\text{Sobre}(A, B) \wedge \text{Sobre}(B, C)$

Acción ($\text{Mover}(b, x, y)$,
PRECOND: $\text{Sobre}(b, x) \wedge \text{Despejado}(b) \wedge \text{Despejado}(y)$
 $\wedge \text{Bloque}(b) \wedge \text{Bloque}(x) \wedge \text{Bloque}(y) \wedge (b \neq x)$
 $\wedge (b \neq y) \wedge (x \neq y)$,
EFECTO: $\text{Sobre}(b, y) \wedge \text{Despejado}(x)$
 $\wedge \neg \text{Sobre}(b, x) \wedge \neg \text{Despejado}(y)$)

Acción ($\text{MoverSobreTablero}(b, x)$,
PRECOND: $\text{Sobre}(b, x) \wedge \text{Despejado}(b) \wedge \text{Bloque}(b) \wedge \text{Bloque}(x) \wedge (b \neq x)$
EFECTO: $\text{SobreTablero}(b) \wedge \text{Despejado}(x) \wedge \neg \text{Sobre}(b, x)$

Acción ($\text{MoverDesdeTablero}(b, x)$,
PRECOND: $\text{SobreTablero}(b) \wedge \text{Despejado}(b) \wedge \text{Bloque}(b) \wedge$
 $\text{Bloque}(x) \wedge \text{Despejado}(x)$
EFECTO: $\neg \text{SobreTablero}(b) \wedge \text{Sobre}(b, x) \wedge \neg \text{Despejado}(x)$

PLANIFICACIÓN CON BÚSQUEDAS EN EL ESPACIO DE ESTADOS



A partir del estado inicial del problema, considerando secuencias de acciones hasta que se encuentra una secuencia que alcance un estado objetivo.

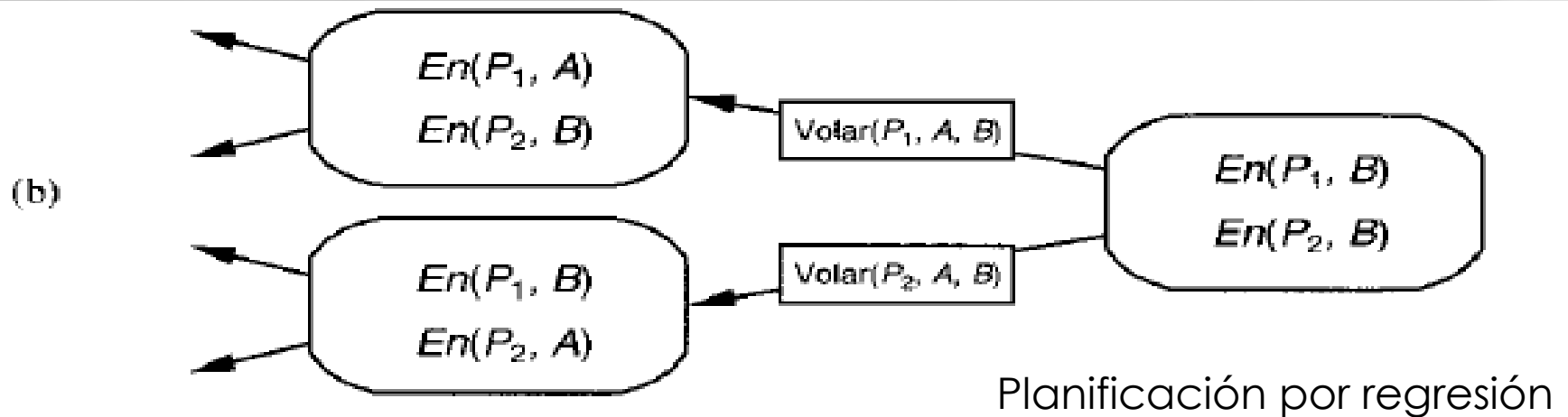
- Se la conoce como Planificación por **progresión**.
- Es ineficaz en la práctica.
- No es capaz de tratar con acciones irrelevantes, considera todas las acciones.

PLANIFICACIÓN DE PROGRESIÓN

La formulación de problemas de planificación **como problemas de búsqueda** en un espacio de estados consta de:

- El **estado inicial** de la búsqueda es el estado inicial del problema de planificación. (conjunto de literales simples y positivos, los que no aparecen se asumen como falsos).
- Las **acciones** que son aplicables en un estado son todas aquellas cuyas precondiciones son satisfechas. El estado resultante de una acción se genera añadiendo literales positivos y eliminando los negativos.
- El **test objetivo** verifica si el estado satisface el objetivo del problema.
- El **costo del camino** típicamente es igual a 1.

BÚSQUEDA HACIA-ATRÁS EN EL ESPACIO DE ESTADOS



- Difícil de implementar.
- No siempre es obvio cómo generar los estados predecesores del conjunto de estados objetivo.
- Ventaja: permite considerar sólo acciones **relevantes y consistentes**.
 - Una acción es relevante para una secuencia encadenada de objetivos si alcanza un conjunto de ellos.
 - Una acción es consistente si no deshace ningún literal deseado.

PLANIFICACIÓN ORDENADA PARCIALMENTE



Las búsquedas en el espacio de estado hacia-delante y hacia-atrás son tipos de planes de búsqueda **totalmente ordenados**.

Un enfoque que trabaje en varios sub-objetivos independientemente, que los solucione con varios sub-planes y combine el conjunto de sub-planes utilizados, flexibiliza el orden en el que se construye el plan. **No está forzado a trabajar en orden cronológico.**

Ejemplo: ponerse un par de zapatos



Objetivo (ZapatoDerechoPuesto \wedge ZapatoIzquierdoPuesto)

Inicio()

Acción(ZapatoDerecho,

PRECOND: CalcetínDerechoPuesto,

EFEECTO: ZapatoDerechoPuesto)

Acción(CalcetínDerecho,

EFEECTO: CalcetínDerechoPuesto)

Acción(ZapatoIzquierdo,

PRECOND: CalcetínIzquierdoPuesto,

EFEECTO: ZapatoIzquierdoPuesto)

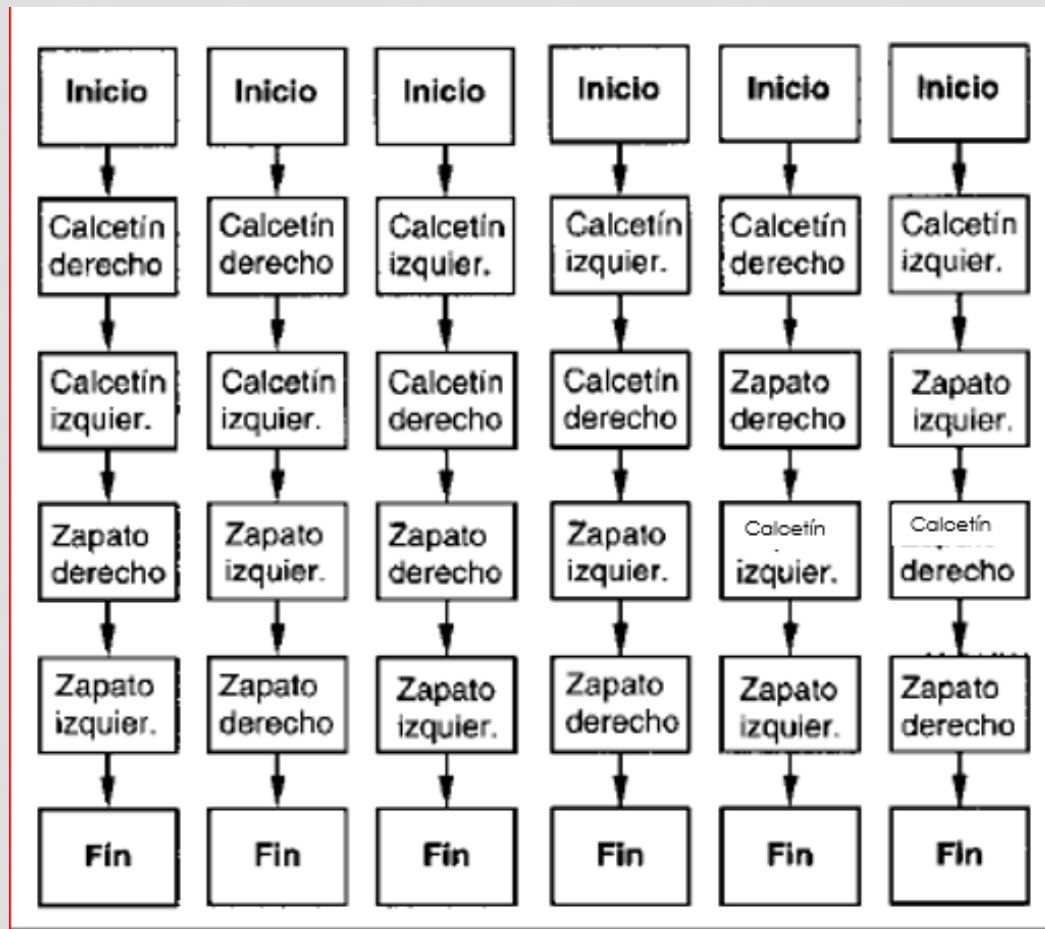
Acción(CalcetínIzquierdo,

EFEECTO: CalcetínIzquierdoPuesto)

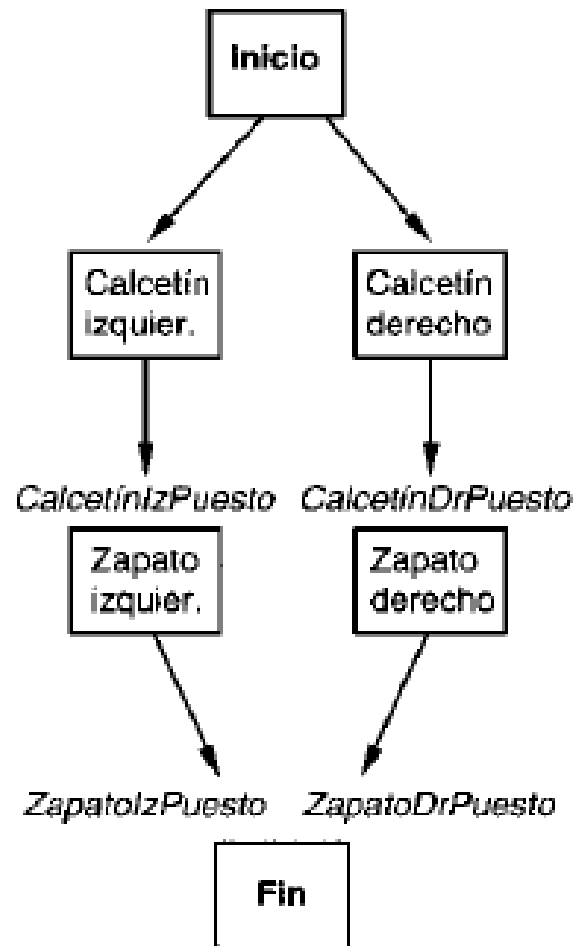
Planificador de primer orden

Puede manipular dos secuencias independientemente sin preocuparse de si una acción pertenece a una secuencia o a otra.

PLAN DE ORDEN TOTAL



Plan de Orden Parcial:



Plan de Orden Total:

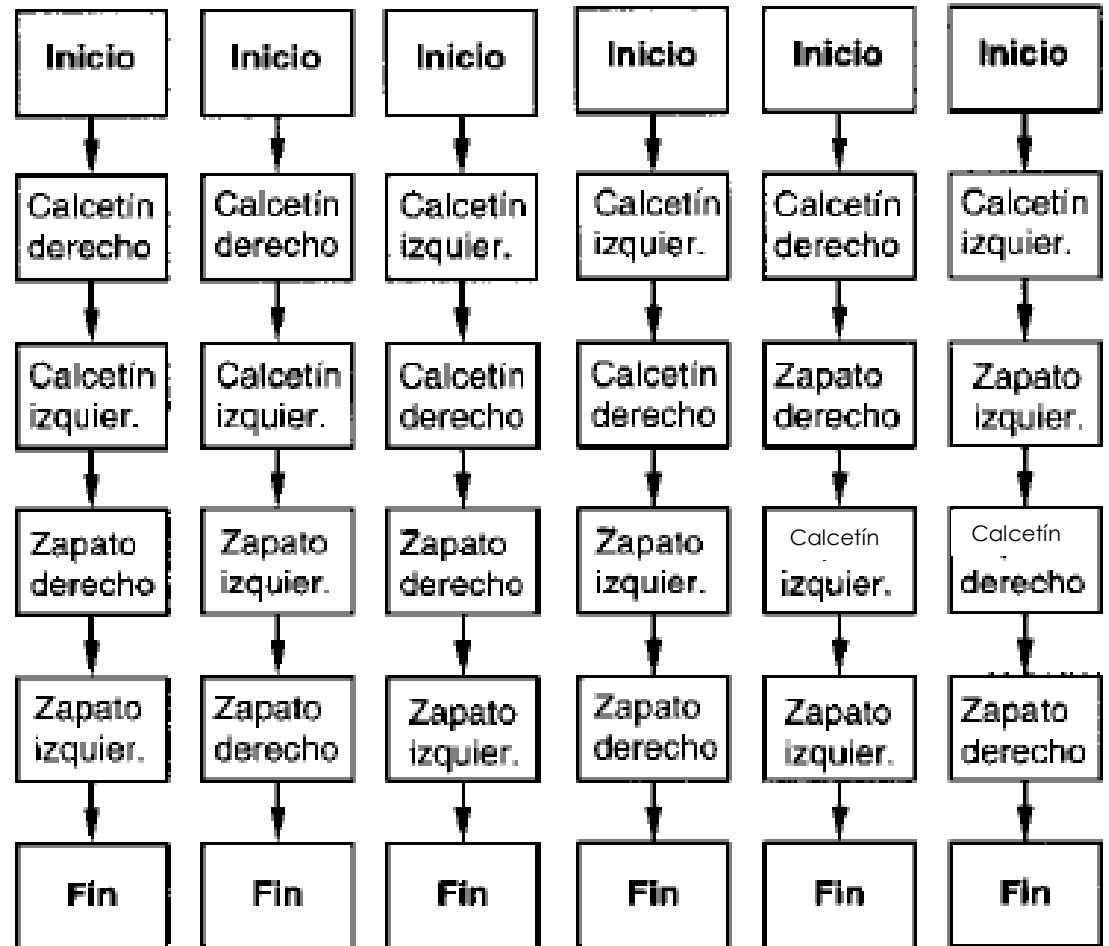


Figura 11.6 Un plan de orden parcial para ponerse calcetines y zapatos, y las seis linealizaciones correspondientes dentro de los planes de orden total.

COMPONENTES DEL PLAN

- Un conjunto de **acciones** que confeccionen las etapas del plan. Éstas son tomadas del conjunto de acciones en el problema de planificación. El plan «vacío» contiene simplemente las acciones *Iniciar* y *Finalizar*. *Iniciar* no posee precondiciones y tiene como efectos los literales en el estado inicial del problema de planificación. *Finalizar* no tiene efectos y tiene como precondiciones los literales del objetivo del problema de planificación.

Iniciar

CalcetínDerecho

CalcetínIzquierdo

ZapatoDerecho

ZapatoIzquierdo

Finalizar

COMPONENTES DEL PLAN



- Un conjunto de **restricciones ordenadas**. Cada limitación ordenada es de la forma $A < B$, la cual se lee como « A antes de B » y significa que la acción A debe ser ejecutada en algún momento antes de B , pero no necesariamente en el estado inmediatamente anterior. Las restricciones ordenadas deben describir un orden parcial apropiado. Cualquier ciclo (del tipo $A < B$ y $B < A$) representa una contradicción, de modo que una limitación de orden no podrá ser añadida a un plan si crea un ciclo.

Calzetín Derecho $<$ Zapato Derecho

Calzetín Izquierdo $<$ Zapato Izquierdo

COMPONENTES DEL PLAN

- Un conjunto de **relaciones causales**. Un enlace causal entre dos acciones A y B en un plan es escrito como $A \xrightarrow{p} B$ y se lee como « A **alcanza** B a través de p ». Por ejemplo, el enlace causal

CalcetínDerecho $\xrightarrow{\text{CalcetínDerechoPuesto}}$ *ZapatoDerecho*



COMPONENTES DEL PLAN



Una precondition es abierta si no es alcanzada por ninguna acción en un plan.

Los planificadores trabajan para reducir el conjunto de precondiciones abiertas al conjunto vacío, sin introducir contradicciones.



COMPONENTES DEL PLAN (RESÚMEN)

Acciones: { *CalcetínDerecho*, *ZapatoDerecho*, *CalcetínIzquierdo*, *ZapatoIzquierdo*, *Iniciar*, *Finalizar* }

Relaciones de orden: { *CalcetínDerecho* < *ZapatoDerecho*,
CalcetínIzquierdo < *ZapatoIzquierdo* }

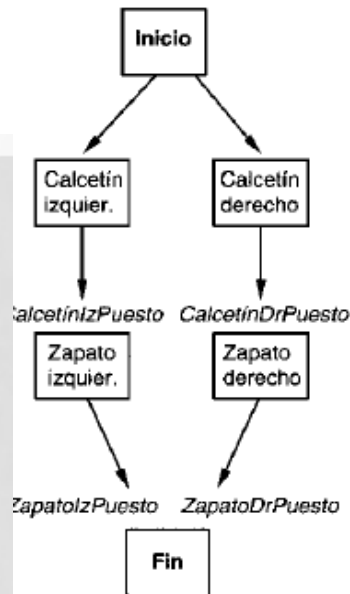
Enlaces: { *CalcetínDerecho* $\xrightarrow{\text{CalcetínDerechoPuesto}}$ *ZapatoDerecho*,
CalcetínIzquierdo $\xrightarrow{\text{CalcetínIzquierdoPuesto}}$ *ZapatoIzquierdo*,
ZapatoDerecho $\xrightarrow{\text{ZapatoDerechoPuesto}}$ *Finalizar*,
ZapatoIzquierdo $\xrightarrow{\text{ZapatoIzquierdoPuesto}}$ *Finalizar* }

PrecondicionesAbiertas: { }

DEFINICIONES

- **PLAN CONSISTENTE:** aquel en el que no hay ciclos en las restricciones ordenadas y no existen conflictos en los enlaces causales.
- **SOLUCIÓN:** plan consistente con precondiciones no abiertas.
- **Cada linealización** de una solución de orden parcial es una solución de orden total cuya ejecución desde el estado inicial alcanza el estado objetivo.

FORMULACIÓN POP



Objetivo (*ZapatoDerechoPuesto* \wedge *ZapatoIzquierdoPuesto*)
Inicio()
Acción(*ZapatoDerecho*,
 PRECOND: *CalcetínDerechoPuesto*,
 EFFECTO: *ZapatoDerechoPuesto*)
Acción(*CalcetínDerecho*,
 EFFECTO: *CalcetínDerechoPuesto*)
Acción(*ZapatoIzquierdo*,
 PRECOND: *CalcetínIzquierdoPuesto*,
 EFFECTO: *ZapatoIzquierdoPuesto*)
Acción(*CalcetínIzquierdo*,
 EFFECTO: *CalcetínIzquierdoPuesto*)

- El plan inicial contiene *Iniciar* y *Finalizar*, la restricción de orden *Iniciar* < *Finalizar*, sin enlaces causales y todas las precondiciones en *Finalizar* como precondiciones abiertas.

Iniciar

ZapatoDerechoPuesto

ZapatolIzquierdoPuesto

Finalizar

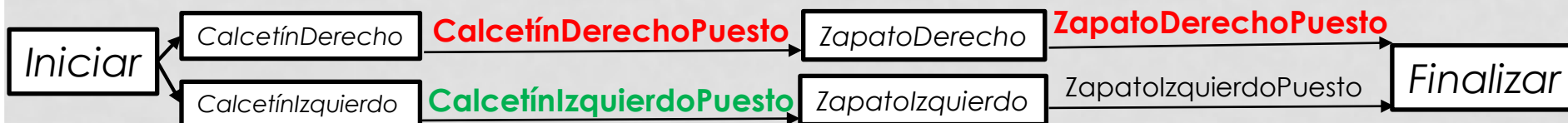
Precondiciones abiertas

FORMULACIÓN POP

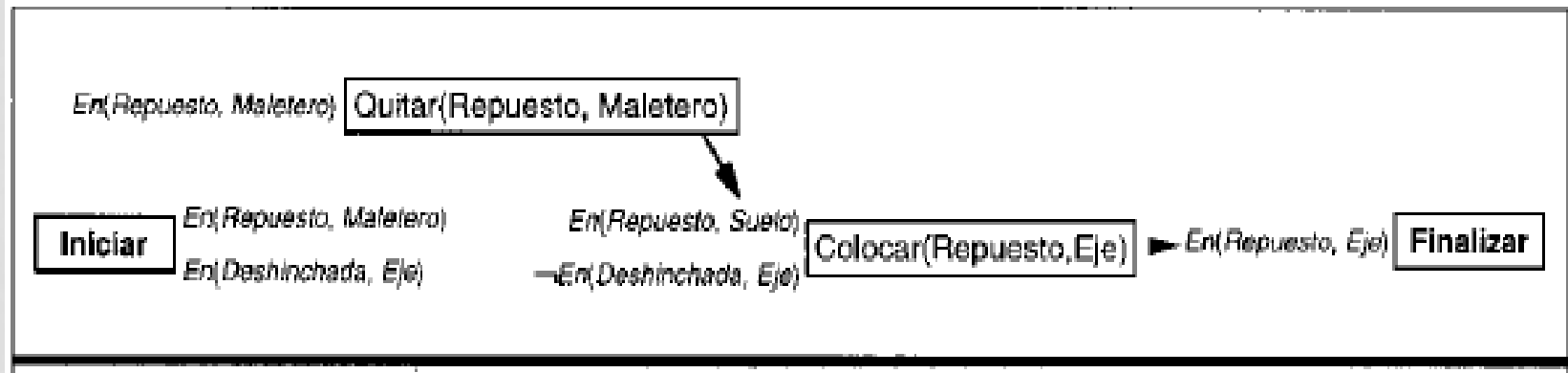
Objetivo (*ZapatoDerechoPuesto* \wedge *ZapatoIzquierdoPuesto*)
Inicio()
Acción(*ZapatoDerecho*,
PRECOND: *CalcetínDerechoPuesto*,
EFECTO: *ZapatoDerechoPuesto*)
Acción(*CalcetínDerecho*,
EFECTO: *CalcetínDerechoPuesto*)
Acción(*ZapatoIzquierdo*,
PRECOND: *CalcetínIzquierdoPuesto*,
EFECTO: *ZapatoIzquierdoPuesto*)
Acción(*CalcetínIzquierdo*,
EFECTO: *CalcetínIzquierdoPuesto*)

- La función sucesora de manera arbitraria selecciona una precondition abierta p sobre una acción B y genera un plan sucesor para cada posible modo de selección consistente de una acción A que alcance p . La consistencia es impuesta como sigue:

El enlace causal $A \xrightarrow{p} B$ y la restricción de orden $A < B$ son añadidos al plan. A puede ser una acción que ya existe en el plan o una nueva. Si es nueva, se añade al plan junto a las condiciones $Iniciar < A$ y $A < Finalizar$.



Resolvemos conflictos entre el nuevo enlace causal y el resto de acciones existentes y entre la acción A (si es nueva) y el resto de enlaces causales existentes. Un conflicto entre $A \xrightarrow{p} B$ y C es resuelto haciendo que C ocurra en algún momento fuera de la protección del intervalo, tanto añadiendo $B < C$ o $C < A$.



Iniciar ($En(Deshinchada, Eje) \wedge En(Repuesto, Maletero)$)

Objetivo ($En(Repuesto, Eje)$)

Acción (*Quitar(Repuesto, Maletero)*,

PRECOND: $En(Repuesto, Maletero)$

EFFECTO: $\neg En(Repuesto, Maletero) \wedge En(Repuesto, Suelo)$)

Acción (*Quitar(Deshinchada, Eje)*,

PRECOND: $En(Deshinchada, Eje)$,

EFFECTO: $\neg En(Deshinchada, Eje) \wedge En(Deshinchada, Suelo)$)

Acción (*Colocar(Repuesto, Eje)*,

PRECOND: $En(Repuesto, Suelo) \wedge \neg En(Deshinchada, Eje)$,

EFFECTO: $\neg En(Repuesto, Suelo) \wedge En(Repuesto, Eje)$)

Acción (*DejarloDeNoche*,

PRECOND:

EFFECTO: $\neg En(Repuesto, Suelo) \wedge \neg En(Repuesto, Eje) \wedge \neg En(Repuesto, Maletero)$

$\wedge \neg En(Deshinchada, Suelo) \wedge \neg En(Deshinchada, Eje)$)

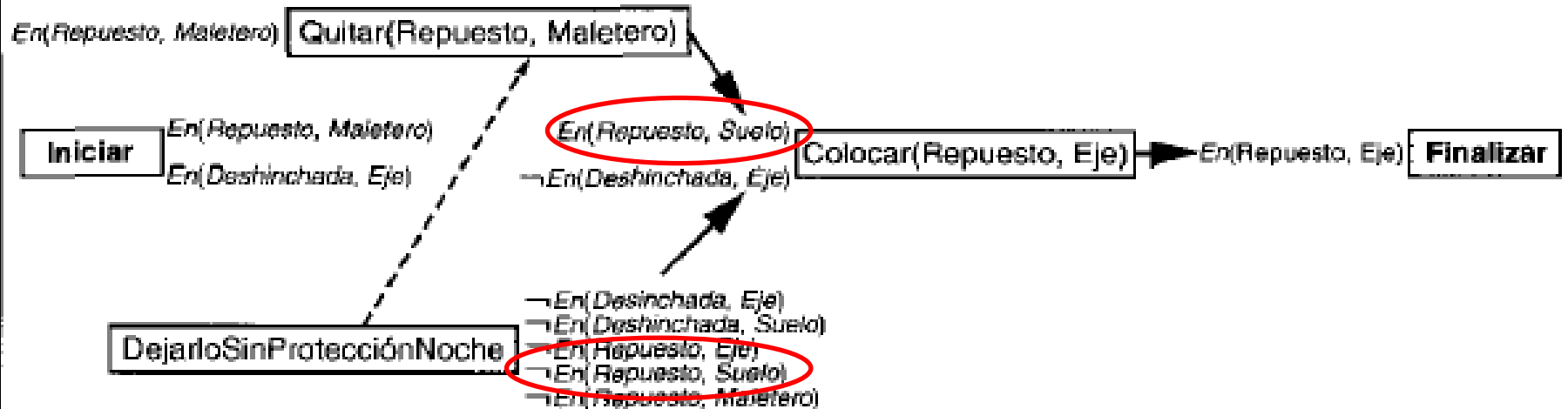
CONFLICTOS

Iniciar ($En(Repuesto, Eje) \wedge En(Repuesto, Maletero)$)
Objetivo ($En(Repuesto, Eje)$)
Acción (*Quitar*(Repuesto, Maletero),
 PRECOND: $En(Repuesto, Maletero)$
 EFECTO: $\neg En(Repuesto, Maletero) \wedge En(Repuesto, Suelo)$)

Acción (*Quitar*(Deshinchada, Eje),
 PRECOND: $En(Deshinchada, Eje)$,
 EFECTO: $\neg En(Deshinchada, Eje) \wedge En(Deshinchada, Suelo)$)

Acción (*Colocar*(Repuesto, Eje),
 PRECOND: $En(Repuesto, Suelo) \wedge \neg En(Deshinchada, Eje)$,
 EFECTO: $\neg En(Repuesto, Suelo) \wedge En(Repuesto, Eje)$)

Acción (*DejarloDeNoche*,
 PRECOND:
 EFECTO: $\neg En(Repuesto, Suelo) \wedge \neg En(Repuesto, Eje) \wedge \neg En(Repuesto, Maletero)$
 $\wedge \neg En(Deshinchada, Suelo) \wedge \neg En(Deshinchada, Eje)$)





SOLUCIÓN FINAL

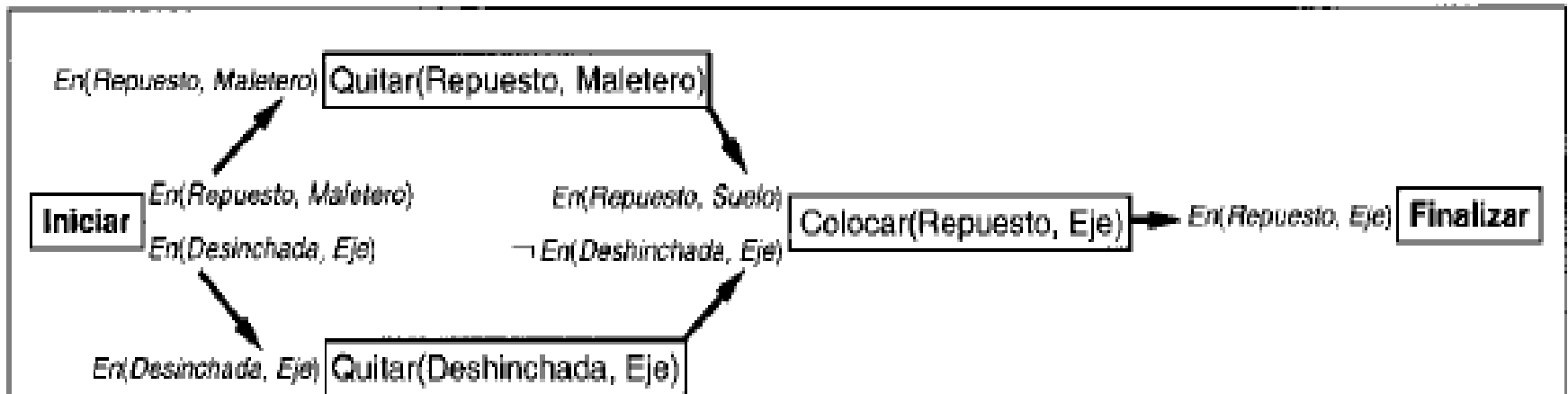


Figura 11.10 La solución final al problema de la rueda de repuesto. Notemos que *Quitar(Repuesto, Maletero)* y *Quitar(Deshinchada, Eje)*, puede ser hecho en cualquier orden si son completados antes con la acción *Colocar(Repuesto, Eje)*.

TENER Y HABER COMIDO PASTEL

Iniciar (Tener(Pastel))

Objetivo (Tener(Pastel) \wedge Comido(Pastel))

Acción (Comer(Pastel))

PRECOND: *Tener(Pastel)*

EFEECTO: \neg *Tener(Pastel) \wedge Comido(Pastel)*

Acción (Cocinar(Pastel))

PRECOND: \neg *Tener(Pastel)*

EFEECTO: *Tener(Pastel)*



GRAFOS DE PLANIFICACIÓN

Iniciar ($Tener(Pastel)$)

Objetivo ($Tener(Pastel) \wedge Comido(Pastel)$)

Acción ($Comer(Pastel)$)

PRECOND: $Tener(Pastel)$

EFEECTO: $\neg Tener(Pastel) \wedge Comido(Pastel)$

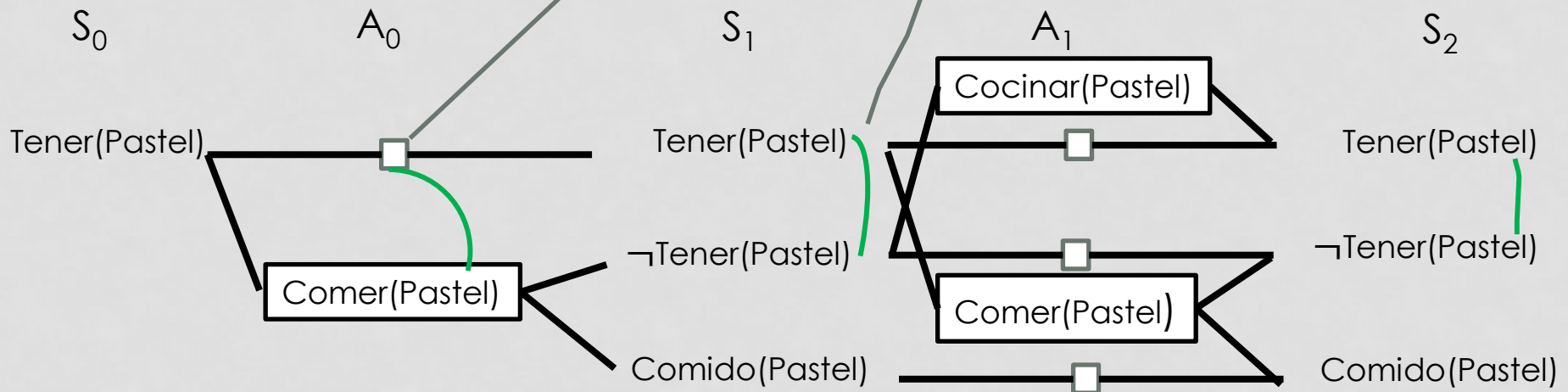
Acción ($Cocinar(Pastel)$)

PRECOND: $\neg Tener(Pastel)$

EFEECTO: $Tener(Pastel)$

Acciones persistentes

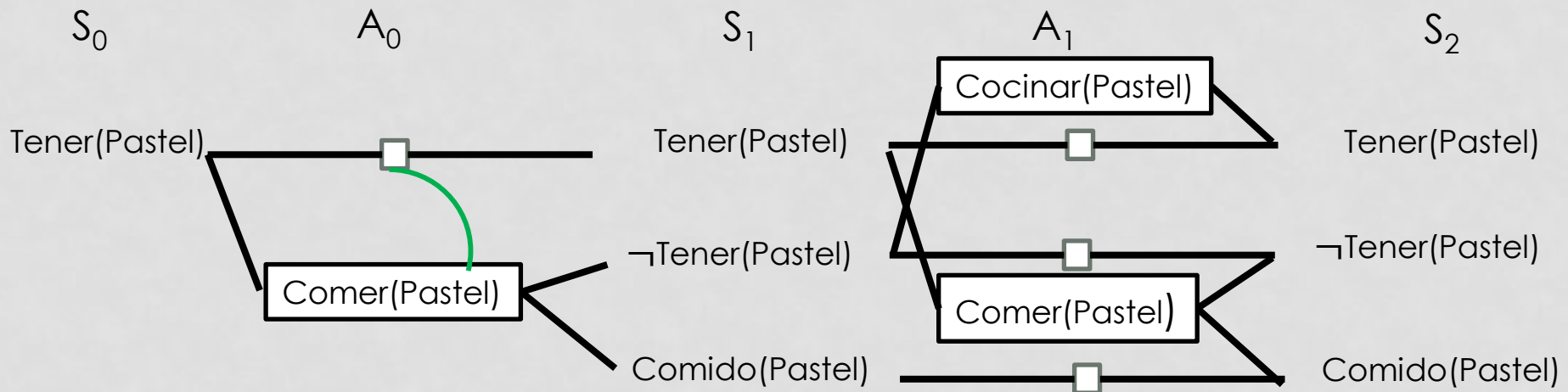
Enlaces de exclusión mutua



ENLACES MUTUAMENTE EXCLUYENTES ENTRE ACCIONES Y LITERALES

❖ Entre dos acciones

- **Efectos inconsistentes**: una acción niega el efecto de la otra
Ej. Comer(Pastel) y la persistencia de Tener(Pastel))

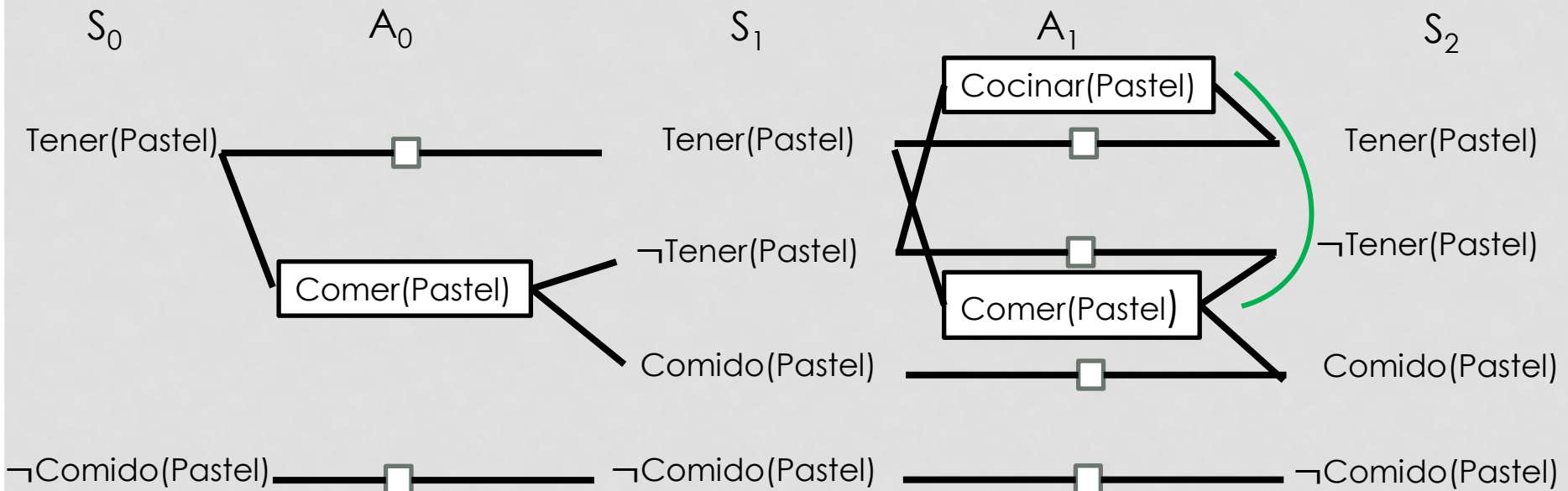


ENLACES MUTUAMENTE EXCLUYENTES ENTRE ACCIONES Y LITERALES

❖ Entre dos acciones

- **Necesidades que entran en competencia:** una de las precondiciones de una acción es mutuamente excluyente con una precondición de la otra.

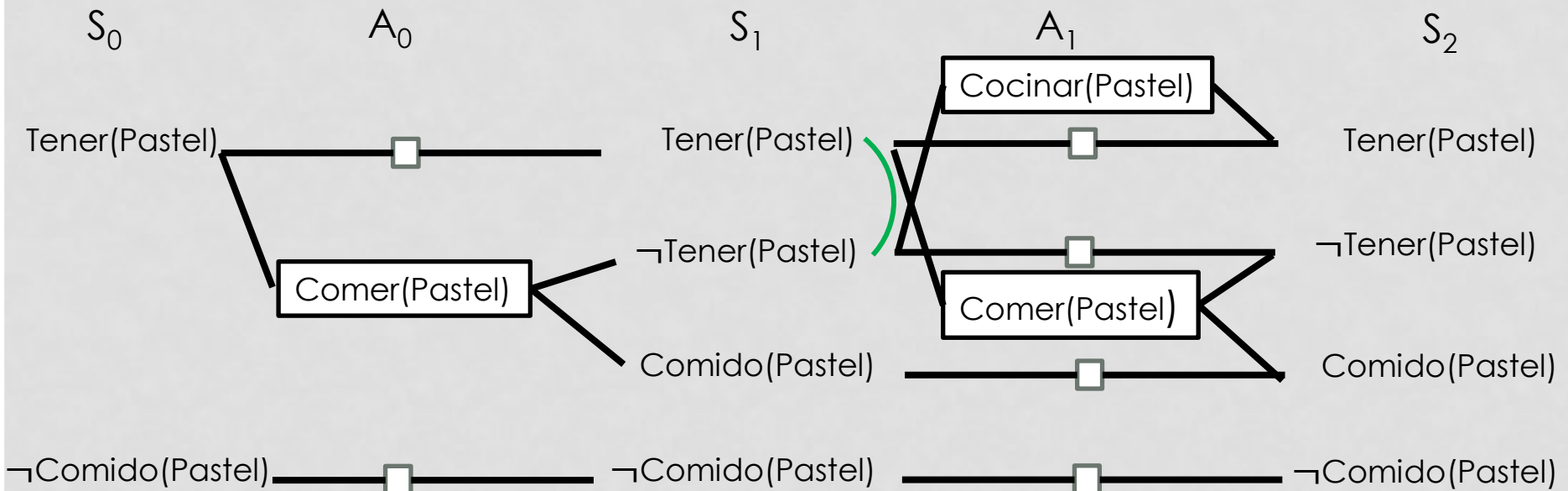
Ej. Cocinar(Pastel) y Comer(Pastel)



ENLACES MUTUAMENTE EXCLUYENTES ENTRE ACCIONES Y LITERALES

❖ Entre dos literales

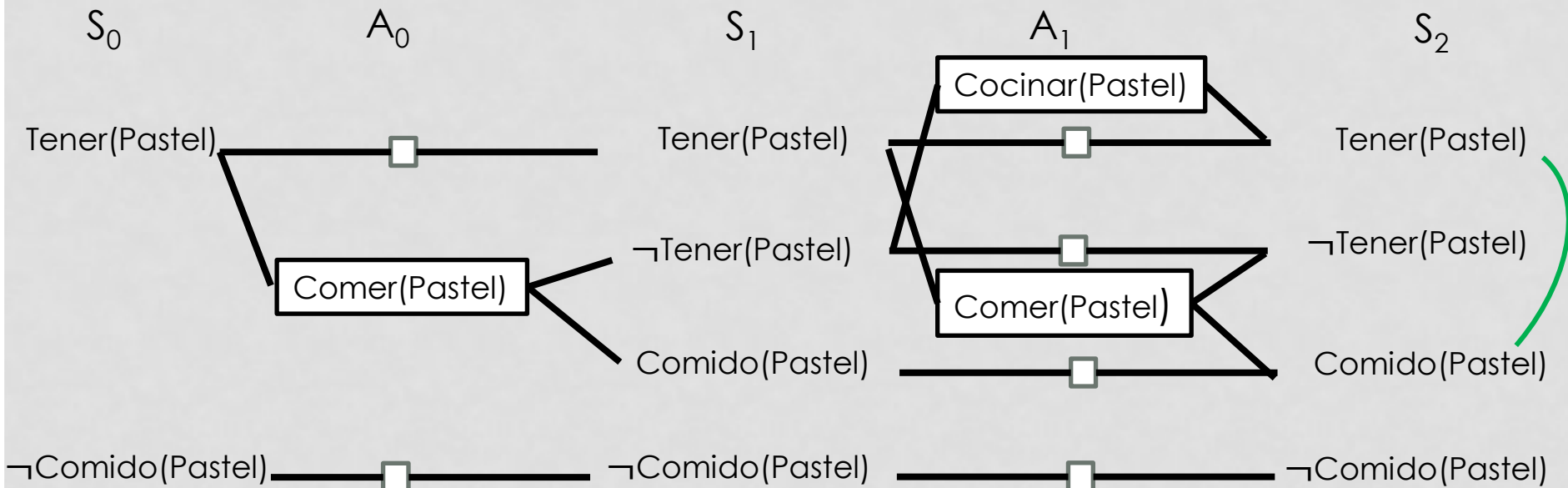
- Si uno es la negación del otro
Ej. Tener(Pastel) y $\neg\text{Tener(Pastel)}$



ENLACES MUTUAMENTE EXCLUYENTES ENTRE ACCIONES Y LITERALES

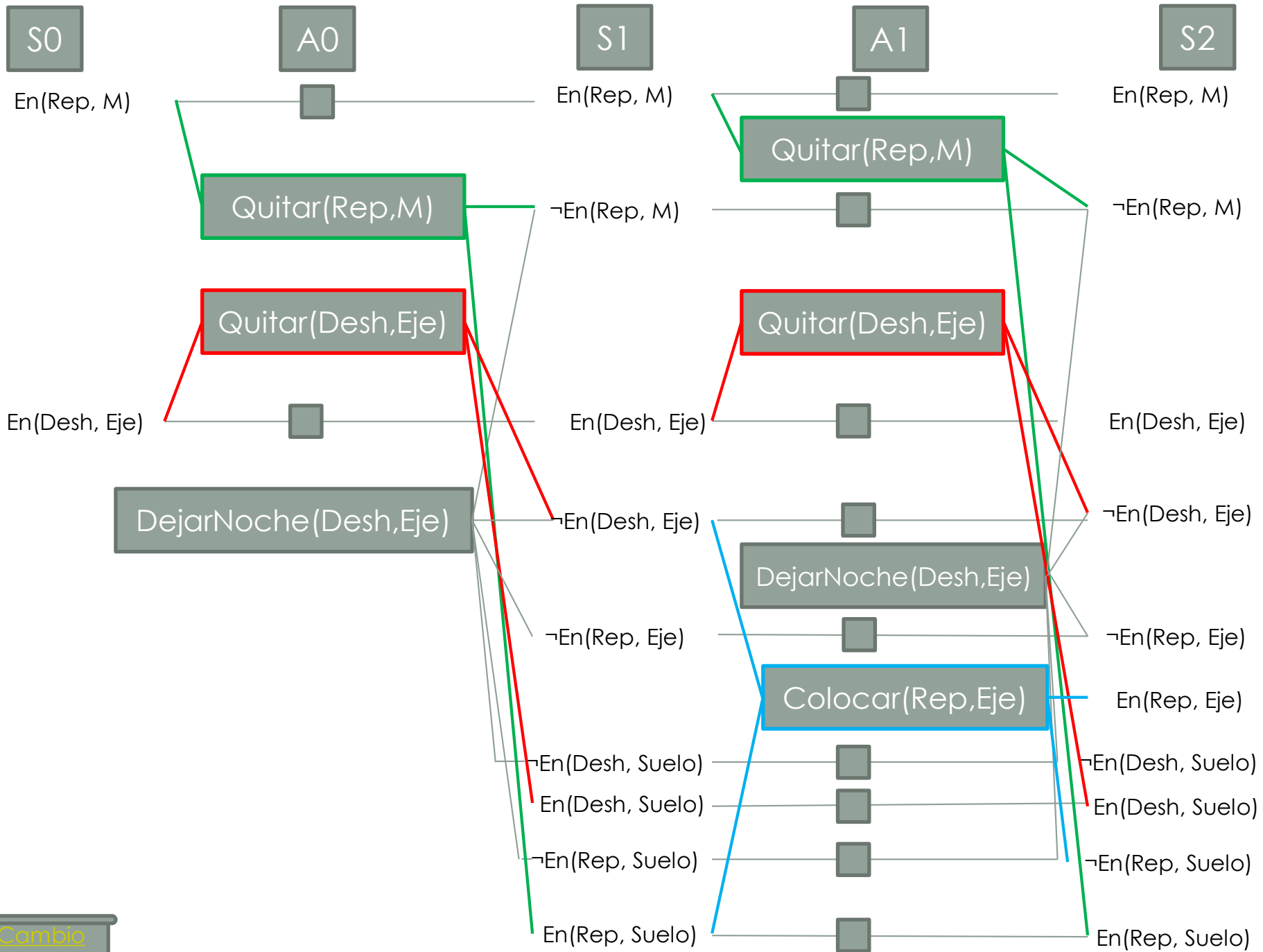
❖ Entre dos literales

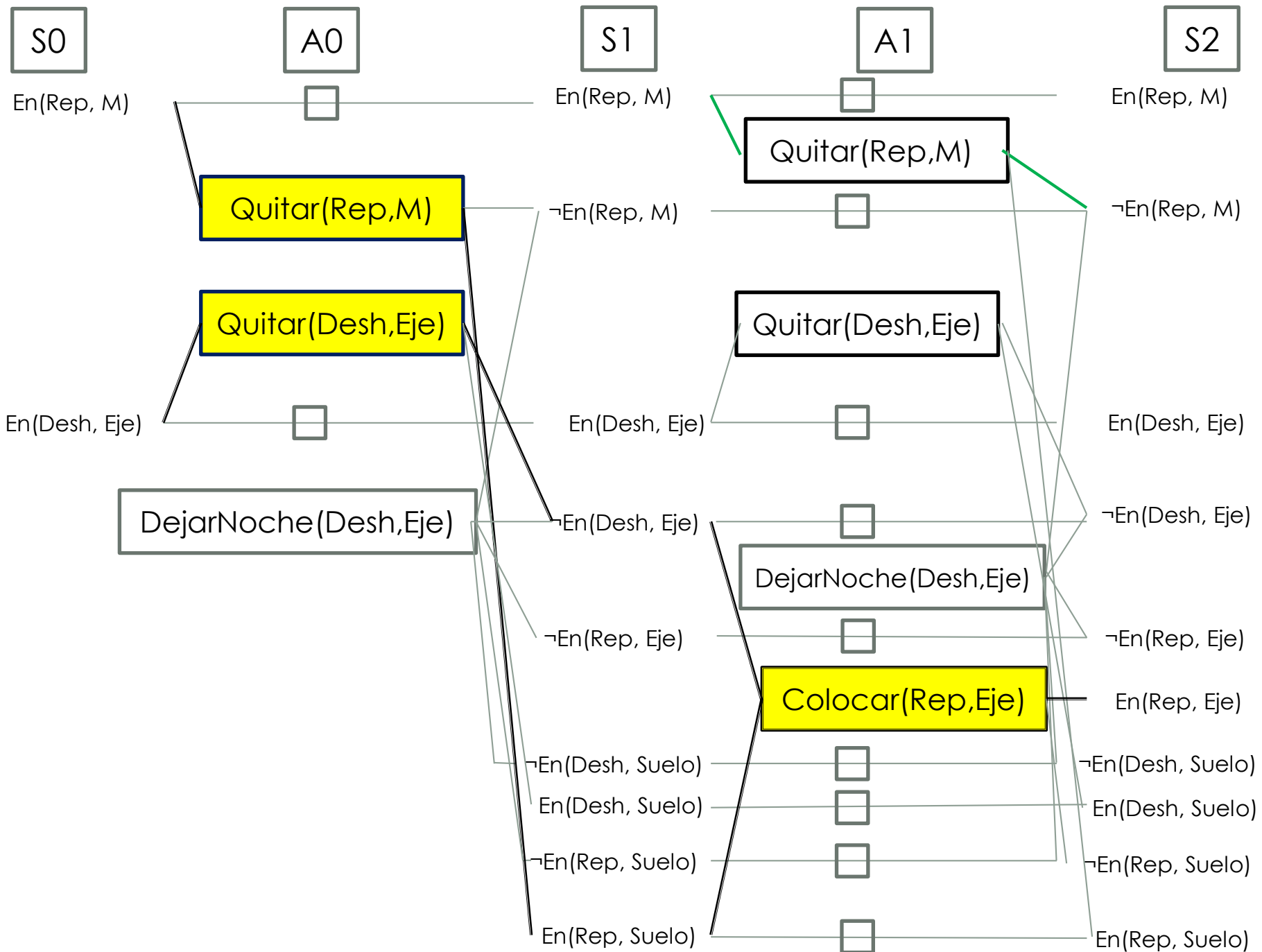
- Si cada posible par de acciones que podría alcanzar los literales son mutuamente excluyentes
 - Ej. Tener(Pastel) y Comido(Pastel)



ALGORITMO GRAPHPLAN

- Primero se chequea si todos los literales del objetivo están presentes en el nivel actual sin que existan enlaces mutuamente excluyentes entre cualquier par de ellos.
- Luego se extiende el grafo añadiendo acciones para el nivel actual y literales de estado para el siguiente nivel.
- El proceso continúa hasta que una solución sea encontrada o se compruebe que la solución no existe.







GRAPHPLAN

Cuando alcanzamos un estado que contiene todos los literales del objetivo y ninguno de ellos es excluido mutuamente por otro se intenta encontrar una solución.

GRAPHPLAN: SOLUCIÓN

- El estado inicial es el último nivel del grafo de planificación S_n , junto con el conjunto de objetivos del problema de planificación.
- Las acciones disponibles en un estado S_i están para seleccionar cualquier conjunto de acciones libre de conflicto en A_{i-1} cuyos efectos cubran los objetivos en el estado. El estado resultante tiene nivel S_{i-1} y tiene como conjunto de objetivos las precondiciones para el conjunto seleccionado de acciones.
- El objetivo es alcanzar un estado a nivel S_0 tal que todos los objetivos sean satisfechos.
- El costo de cada acción es 1.

TIEMPO, PLANIFICACIÓN Y RECURSOS

- **STRIPS** indica qué acciones hacer pero no indica cuánto dura una acción ni cuando ocurre.
- **En algunos dominios es importante saber cuando empiezan y terminan las acciones.**
- **Programación de actividades:** problemas que requieren completar un conjunto de tareas, cada una de las cuales consiste en una secuencia de acciones y donde cada acción tiene una duración determinada y puede requerir varios recursos. **El problema está en determinar el plan que minimice el tiempo total requerido para completar todas las tareas, a la vez que se respetan las restricciones impuestas a sus recursos.**

PROGRAMACIÓN DE ACTIVIDADES

(EJ: ENSAMBLAR EL COCHE 1 Y EL COCHE 2)

Tareas:

- Colocar el motor
- Colocar las ruedas
- Inspeccionar

(En ese orden)

e: motor
c: chasis
m: tiempo
w: ruedas

Iniciar($\text{Chasis}(C1) \wedge \text{Chasis}(C2)$
 $\wedge \text{Motor}(E1, C1, 30) \wedge \text{Motor}(E2, C2, 60)$
 $\wedge \text{Ruedas}(W1, C1, 30) \wedge \text{Ruedas}(W2, C2, 15))$

Objetivo($\text{Ensamblado}(C1) \wedge \text{Ensamblado}(C2)$)

Acción(ColocarMotor(**e,c,m**)
PRECOND: $\text{Motor}(e, c, m) \wedge \text{Chasis}(c) \wedge \neg \text{MotorDentro}(c)$,
EFECTO: $\text{MotorDentro}(c) \wedge \text{Duración}(m)$)

Acción(ColocarRuedas(w,c),
PRECOND: $\text{Ruedas}(w, c, m) \wedge \text{Chasis}(c)$,
EFECTO: $\text{RuedasPuestas}(c) \wedge \text{Duración}(m)$)

Acción(Inspeccionar(c).
PRECOND: $\text{MotorDentro}(c) \wedge \text{RuedasPuestas}(c) \wedge \text{Chasis}(c)$,
EFECTO: $\text{Ensamblado}(c) \wedge \text{Duración}(m)$)

Iniciar($\text{Chasis}(C1) \wedge \text{Chasis}(C2)$
 $\wedge \text{Motor}(E1, C1, 30) \wedge \text{Motor}(E2, C2, 60)$
 $\wedge \text{Ruedas}(W1, C1, 30) \wedge \text{Ruedas}(W2, C2, 15)$)

Objetivo($\text{Ensamblado}(C1) \wedge \text{Ensamblado}(C2)$)

Acción(ColocarMotor(e, c, m))

PRECOND: $\text{Motor}(e, c, m) \wedge \text{Chasis}(c) \wedge \neg \text{MotorDentro}(c)$,

EFEECTO: $\text{MotorDentro}(c) \wedge \text{Duración}(m)$)

Acción(ColocarRuedas(w, c),

PRECOND: $\text{Ruedas}(w, c, m) \wedge \text{Chasis}(c)$,

EFEECTO: $\text{RuedasPuestas}(c) \wedge \text{Duración}(m)$)

Acción(Inspeccionar(c).

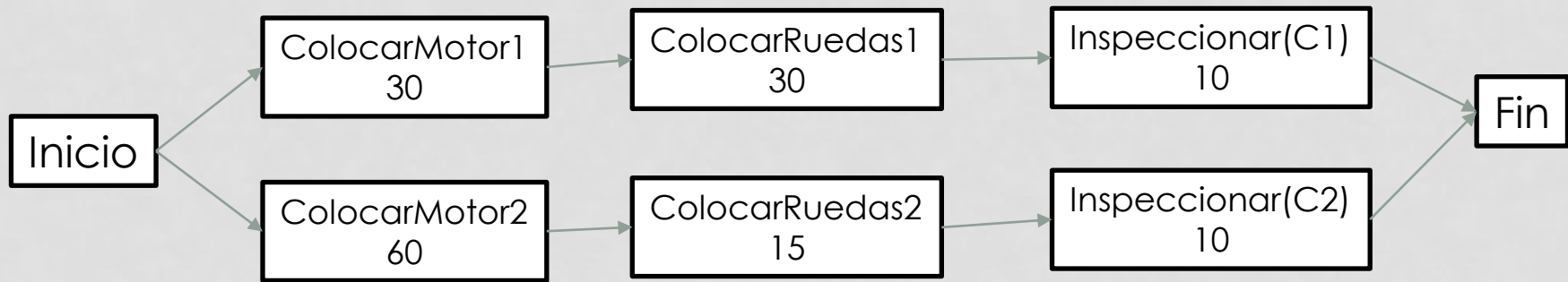
PRECOND: $\text{MotorDentro}(c) \wedge \text{RuedasPuestas}(c) \wedge \text{Chasis}(c)$,

EFEECTO: $\text{Ensamblado}(c) \wedge \text{Duración}(10)$)

CAMINO CRÍTICO

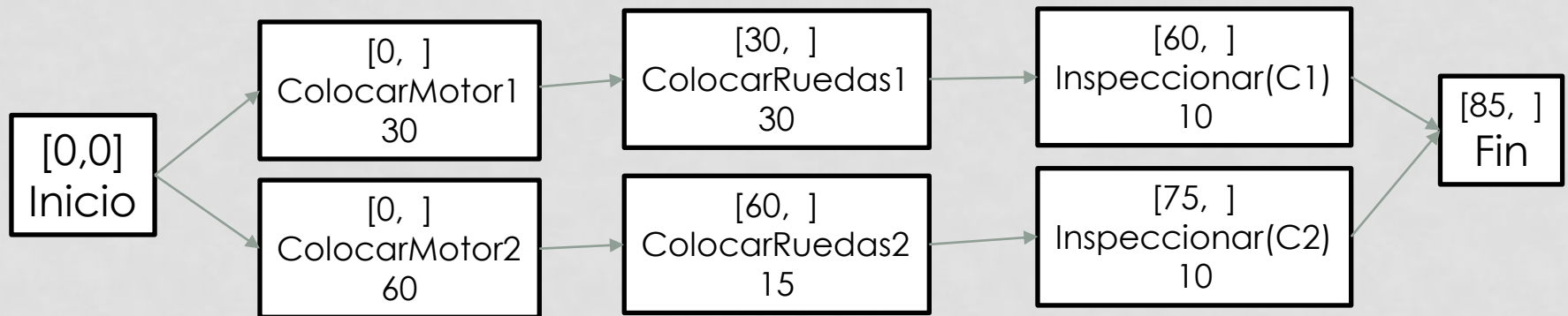
El **camino crítico** es aquel que presenta la mayor duración.

Determina la duración de un plan completo ya que reducir otros caminos no se traduce en un acortamiento del plan como conjunto pero retrasar el inicio de cualquier acción del camino crítico ralentiza el plan en su conjunto.



Para completar el plan completo en el mínimo tiempo total, las acciones del CC deben ser ejecutadas sin retrasos entre ellas.

CAMINO CRÍTICO

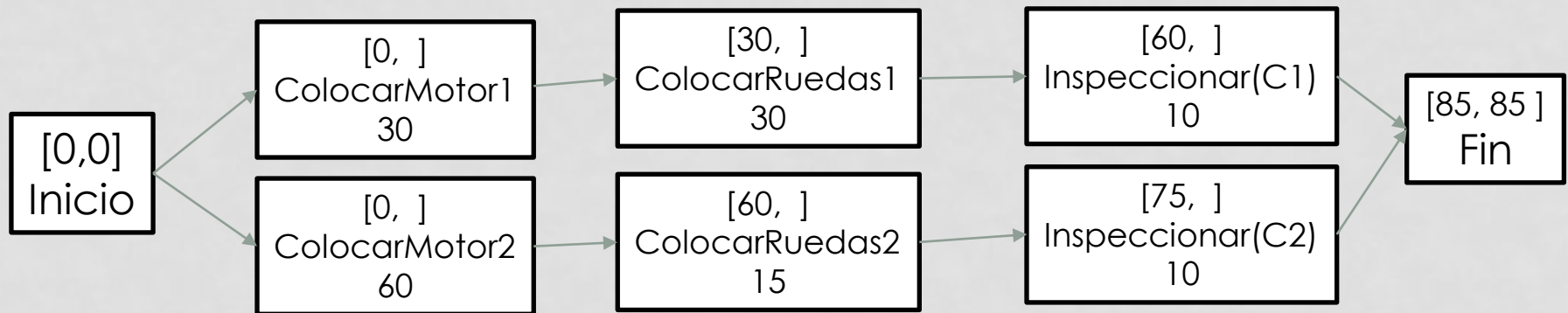


[ES,LS]
Acción
duración

ES: intervalo de tiempo previo al inicio

LS: intervalo de tiempo posterior al inicio

CAMINO CRÍTICO

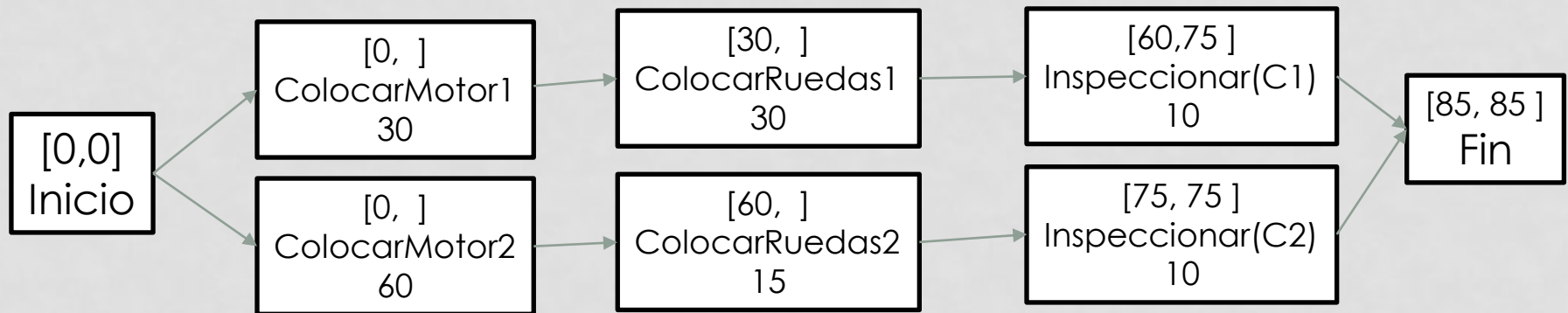


[ES,LS]
Acción
duración

ES: intervalo de tiempo previo al inicio

LS: intervalo de tiempo posterior al inicio

CAMINO CRÍTICO

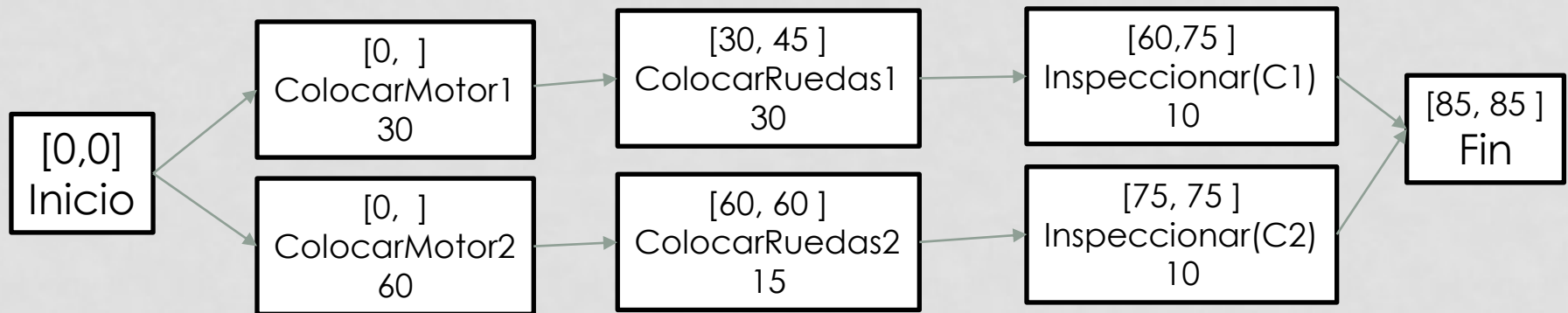


[ES,LS]
Acción
duración

ES: intervalo de tiempo previo al inicio

LS: intervalo de tiempo posterior al inicio

CAMINO CRÍTICO

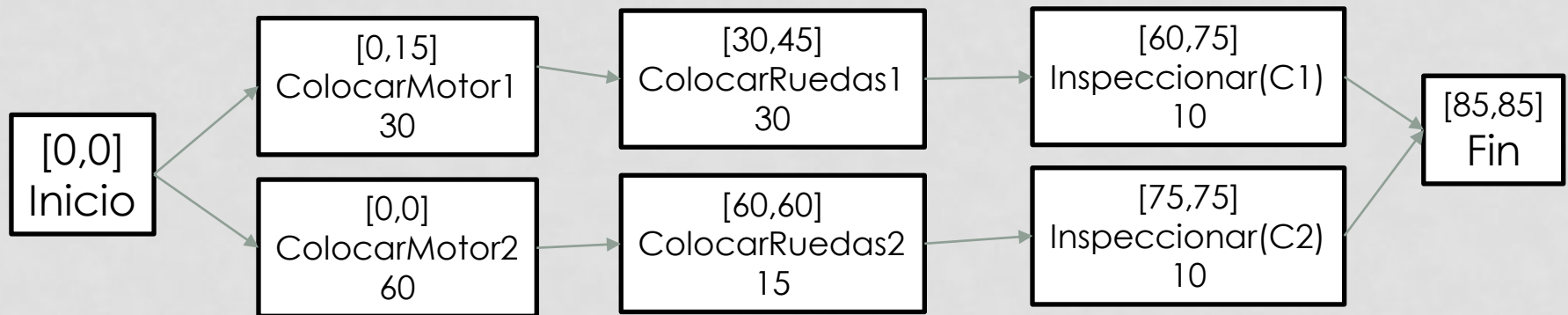


[ES,LS]
Acción
duración

ES: intervalo de tiempo previo al inicio

LS: intervalo de tiempo posterior al inicio

CAMINO CRÍTICO



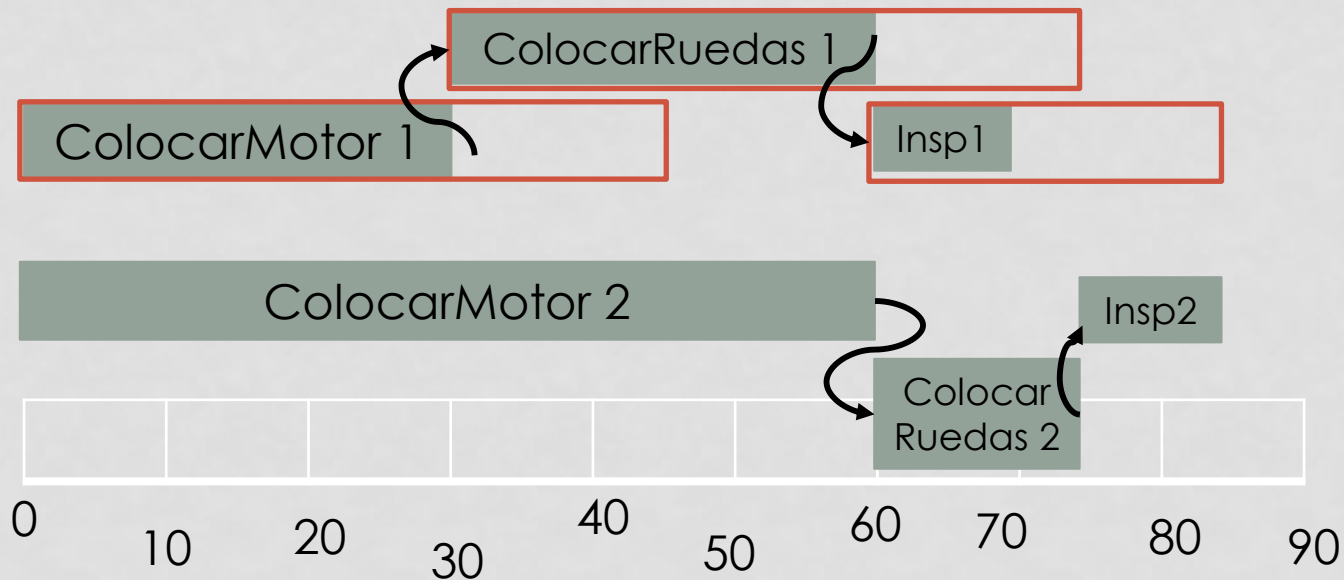
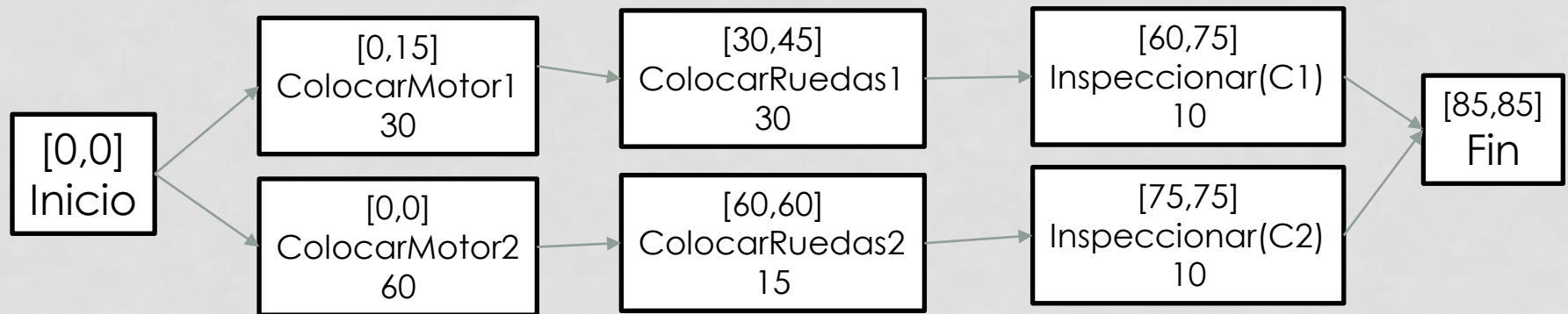
[ES,LS]
Acción

ES: intervalo de tiempo previo al inicio

LS: intervalo de tiempo posterior al inicio

INTERVALO DE RELAJACIÓN

$ES_i - LS_{i-1} = \text{intervalo de relajación}$



PROGRAMACIÓN CON RESTRICCIÓN DE RECURSOS

INICIAR (Chasis(C1) \wedge Chasis(C2) \wedge Motor(E1,C1,30) \wedge Motor(E2,C2,60) \wedge
Ruedas(W1,C1,30) \wedge Ruedas(W2,C2,15) \wedge EnsamblajeDeCarrocería(1) \wedge
EnsamblajeDeRuedas(1) \wedge Inspectores(2))

OBJETIVO ((Montado(C1) \wedge Montado(C2)))

ACCIÓN (AñadirMotor(e,c,m)

Prec: Motor(e,c,m) \wedge Chasis(c) \wedge \neg MotorDentro(c)

Efecto: MotorDentro(c) \wedge Duración(m)

Recurso: EnsamblajeDeMotor(1)

ACCIÓN (AñadirRuedas(w,c,m)

Prec: Ruedas(w,c,m) \wedge Chasis(c)

Efecto: RuedasPuestas(c) \wedge Duración(m)

Recurso: EnsamblajeDeRuedas(1)

ACCIÓN (Inspeccionar(c)

Prec: MotorDentro(c) \wedge RuedasPuestas(c)

Efecto: Montado(c) \wedge Duración(10)

Recurso: Inspectores(2)

PROGRAMACIÓN CON RESTRICCIÓN DE RECURSOS

