DECSAI
Departamento de Ciencias de la Computación e I.A.
Universidad de Granada

Planificación

Fernando Berzal, berzal@acm.org

Planificación en I.A.

- Introducción:
  - Razonamiento sobre acciones
  - Ejecución de planes
  - Aplicaciones
  - Problemas
- Planificación como búsqueda
  - Búsqueda hacia adelante ("progresión")
  - Búsqueda hacia atrás ("regresión")
- Planificación clásica: STRIPS
- Planificación como búsqueda en el espacio de planes

Introducción

Actuamos sin planificar…

… cuando el propósito es inmediato.

… cuando realizamos tareas bien aprendidas.

… cuando nuestro curso de acción puede adaptarse libremente sin demasiadas consecuencias.

Planificamos…

… cuando nos encontramos ante situaciones nuevas.

… cuando las tareas son complejas.

… cuando existen riesgos (costes) que asumir.

… cuando colaboramos con otros.

Introducción

Definición de planificación

Proceso de deliberación que escoge y organiza acciones anticipando sus resultados/consecuencias.

Planificar es razonar sobre acciones.

Introducción

Compromisos

- Sólo planificamos algo cuando es estrictamente necesario, ya que elaborar planes requiere esfuerzo.
- A menudo nos bastan planes aceptables, aunque no sean óptimos (**satisfacción vs. optimalidad**).

Introducción

Problema de planificación:

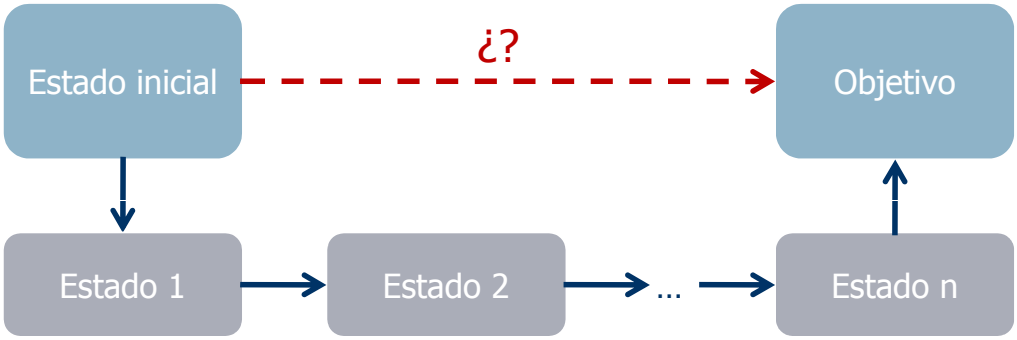
Dados

una descripción del "mundo" (un modelo),
un estado inicial,
una descripción del objetivo y
un conjunto de acciones que pueden cambiar el mundo,

encontrar

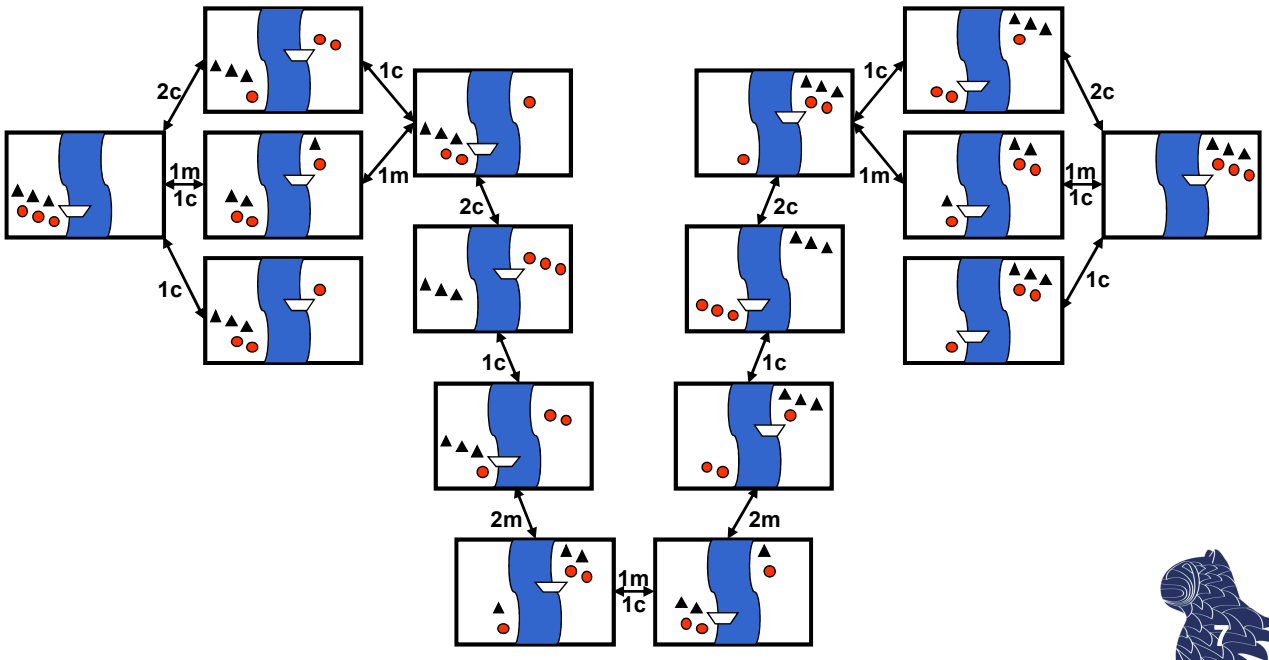
una secuencia de acciones que convierten el estado inicial en un estado que satisfaga el objetivo.

Introducción



→ representa la ejecución de una acción (aplicación de algún operador o transformación del sistema).

Ejemplo: Misioneros y caníbales



Introducción

Ejecución de planes

Introducción

Planificación dinámica

- El modelo del sistema difiere del sistema real.
- Pueden existir factores externos que interrumpan la ejecución del plan.

Introducción

Planificación dinámica

Modelo más realista:
Planificación y ejecución se entrelazan.

- Supervisión de planes
  (detectar observaciones diferentes a los resultados esperados).
- Revisión de planes
  (adaptación del plan existente a nuevas circunstancias).
- Replanificación
  (generación de un nuevo plan a partir del estado actual).

Introducción

Formas de representar el objetivo
en un problema de planificación:

- Estado objetivo (o conjunto de estados).
- Condición que ha de satisfacerse,
  p.ej. estados que evitar o visitar.
- Función de utilidad que maximizar.
- Tarea que realizar.

Introducción

Estrategias alternativas (y complementarias)
para la resolución de problemas de planificación:

- **Planificación para dominios específicos**
  Técnicas específicas adaptadas a cada problema
  p.ej. Planificación de trayectorias y movimientos.

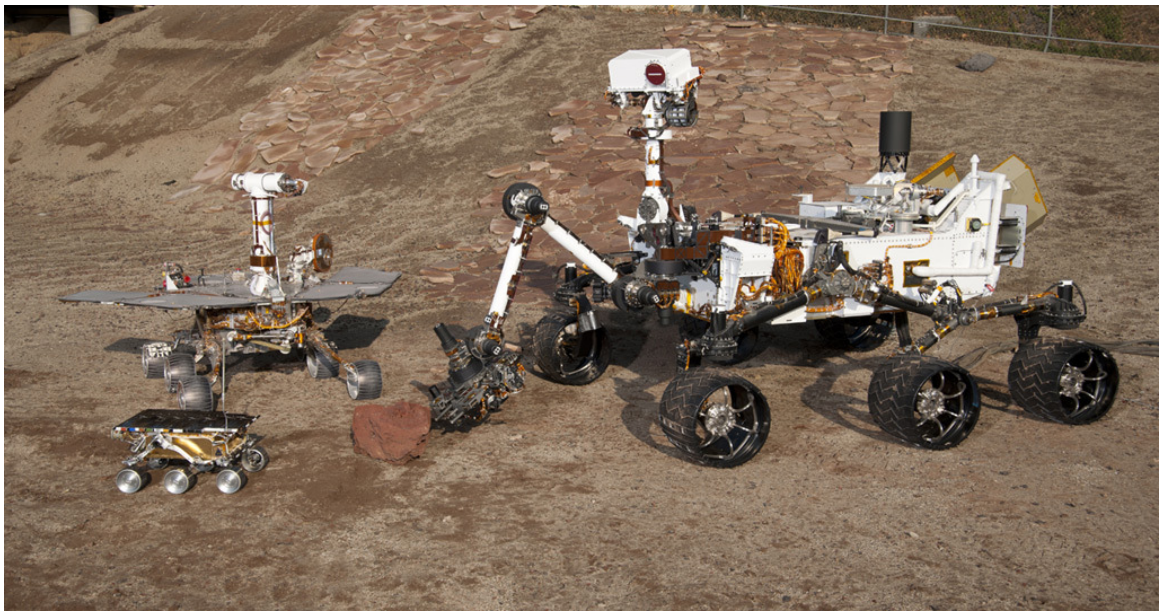
- **Planificación independiente del dominio**
  Técnicas genéricas de representación y resolución de problemas de planificación (p.ej. basadas en lógica).

Aplicaciones

Aplicaciones

- Robótica (robots móviles y vehículos autónomos)
- Simulación (entrenamiento y juegos)
- Logística
- "Workflows" (fábricas y cadenas de montaje)
- Gestión de crisis (evacuaciones, incendios…)

Aplicaciones



Mars Exploration Rovers

http://marsrovers.jpl.nasa.gov/

Problemas

Problema del marco

Cómo representar qué cambia y qué permanece cuando ejecutamos una acción:

- Hipótesis de mundo cerrado
  (todo lo que no se sabe es falso).
- Hipótesis de mundo abierto
  (lo que no se sabe puede ser falso o cierto).

Problemas

Problema de la cualificación

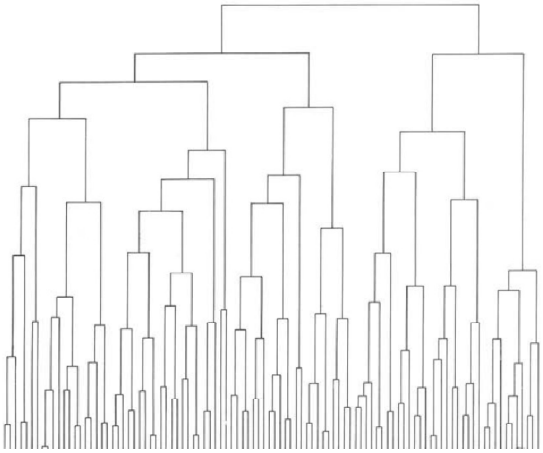
Efectos dependientes del contexto (el resultado de una acción depende de tantas condiciones que ni siquiera se puedan enumerar).

p.ej. si quiero ir de Granada a Madrid lo puedo hacer en avión, pero que pueda hacerlo depende de las condiciones meteorológicas, de que no haya huelgas, de que no se produzcan fallos mecánicos en el avión, de que no se produzcan incendios ni accidentes…

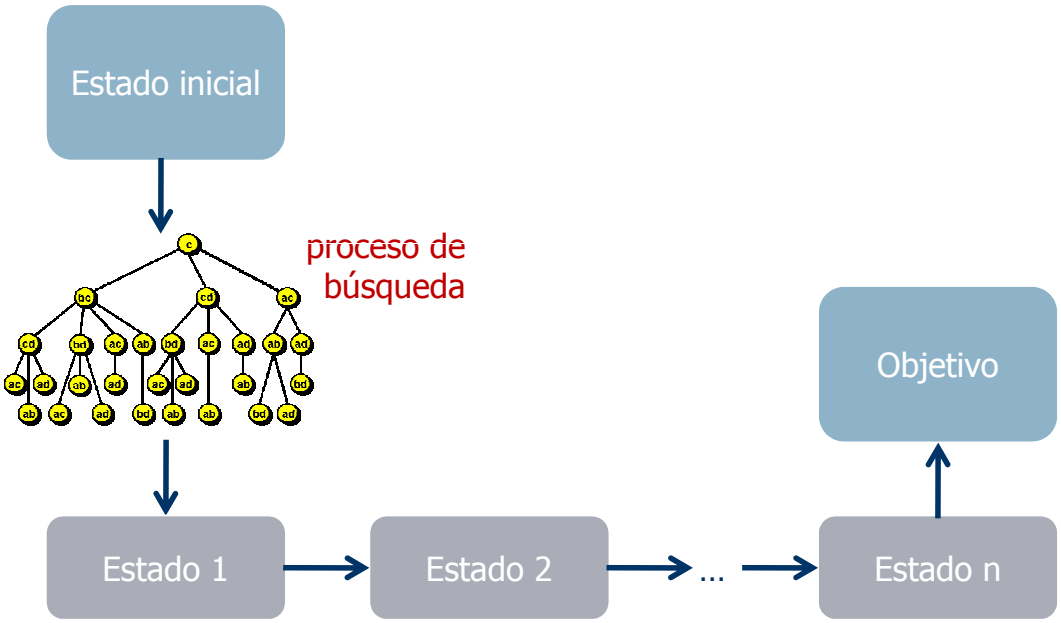
Problemas

Problema de la ramificación

Existen demasiadas acciones posibles, cada una de las cuales puede tener muchas consecuencias implícitas.



Planificación como búsqueda



Planificación como búsqueda

Espacio de estados

Describe las distintas formas
en que puede evolucionar un sistema.

Objetivo

Un estado o conjunto de estados
que satisface la función objetivo.

Plan

Camino a través del espacio de estados
(equivalente a una secuencia de acciones).

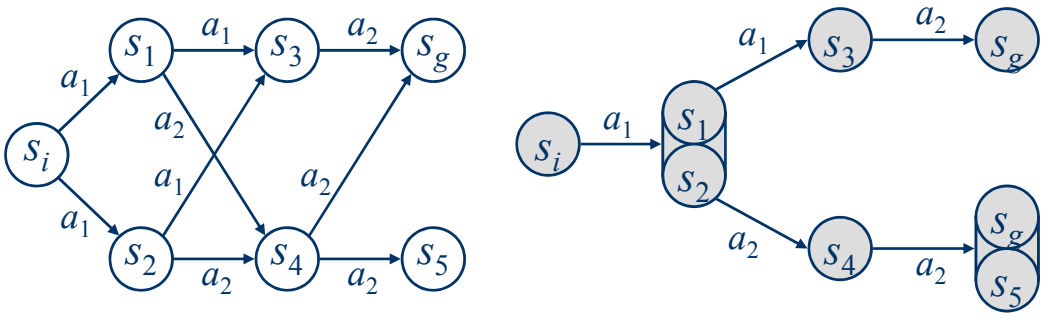
Planificación como búsqueda

Limitaciones

- Conjunto finito de estados.
  ¿Dominios continuos? Discretización.
- Sistemas completamente observables.
  ¿Información parcial? Replanificación.
- Sistemas deterministas
  ¿Incertidumbre sobre el estado actual o el resultado de las acciones? Planificación estocástica.

Planificación como búsqueda

Planificación estocástica



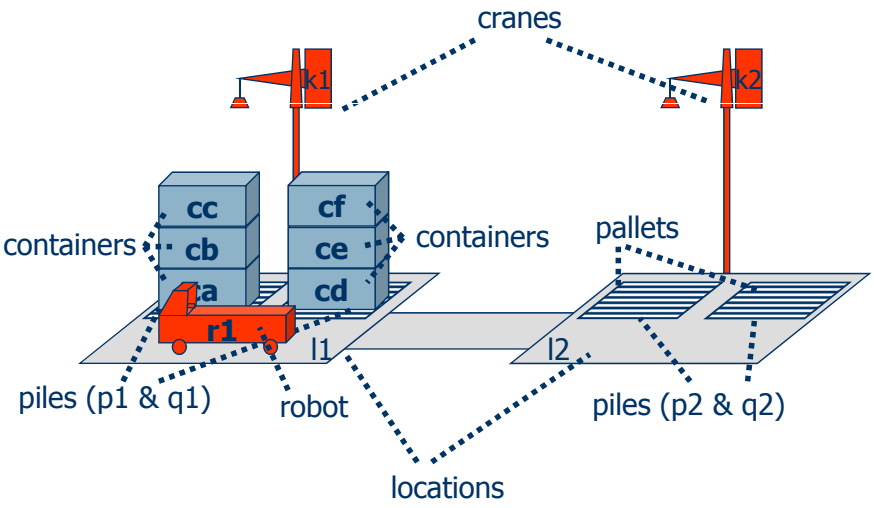
Sistema no determinista

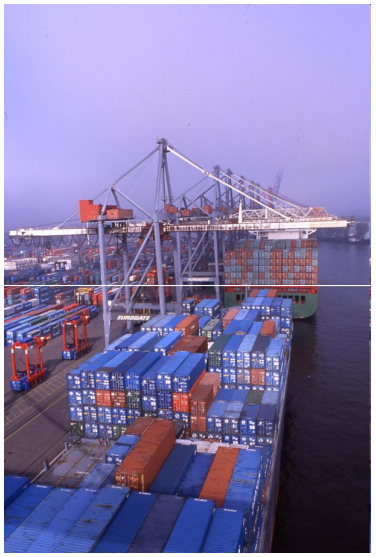
Sistema determinista equivalente

Ejemplo

DWR [Dock-Worker Robots]

Estados





Ejemplo

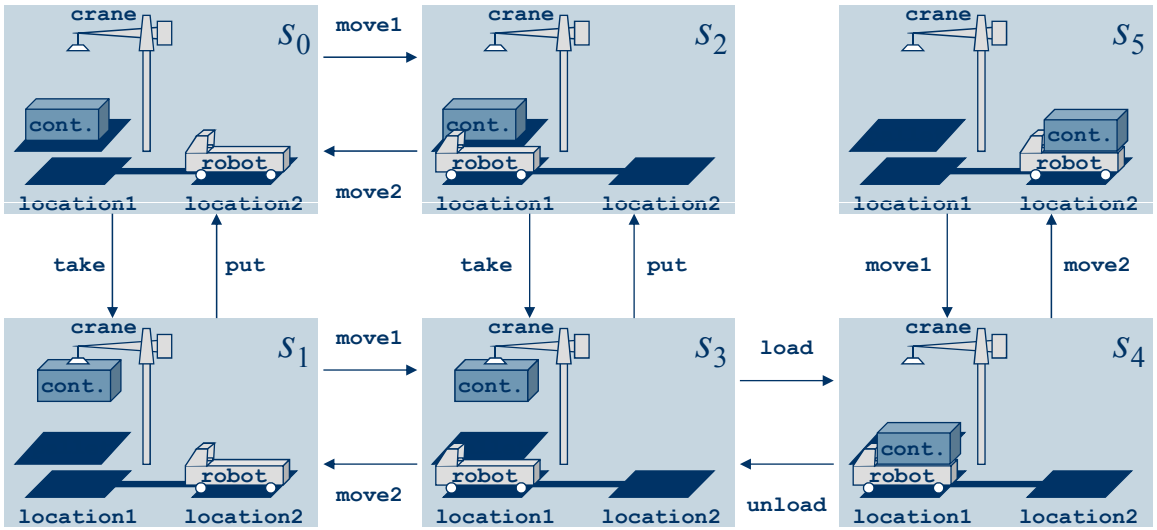
DWR [Dock-Worker Robots]

Acciones

- **move(r,l,l')**
  Mover robot r desde el lugar l hasta el lugar l'.
- **take(k,l,c,p)**
  Coger el contenedor c con la grúa libre k del tope de la pila p (estando todos en el mismo lugar, l).
- **put(k,l,c,c',p)**
  Dejar el contenedor c llevado por la grúa k sobre c' en el tope de la oila p (estando todos en el mismo lugar, l).
- **load(k,l,c,r)**
  Cargar el contenedor c llevado por la grúa k en el robot descargado r (estando todos todos en el lugar l).
- **unload(k,l,c,r)**
  Descargar el contenedor k llevado por el robot r con la grúa vacía k (estando todos en el lugar l).

Ejemplo

DWR [Dock-Worker Robots]



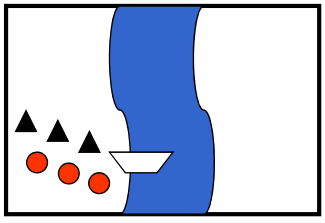
Transiciones entre estados

Ejemplo: Misioneros y caníbales

Estado inicial

Todos los misioneros y caníbales en la orilla izquierda

(con un bote con capacidad para sólo dos personas)



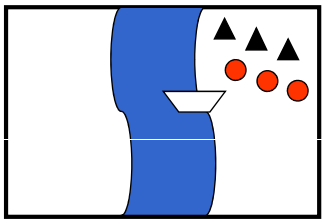
Acciones posibles

- 1 misionero cruza el río.
- 1 caníbal cruza el río.
- 2 misioneros cruzan el río.
- 2 caníbales cruzan el río.
- 1 misionero y 1 caníbal cruzan el río

Ejemplo: Misioneros y caníbales

Objetivo

Todos los misioneros y caníbales en la orilla derecha



Coste de la solución

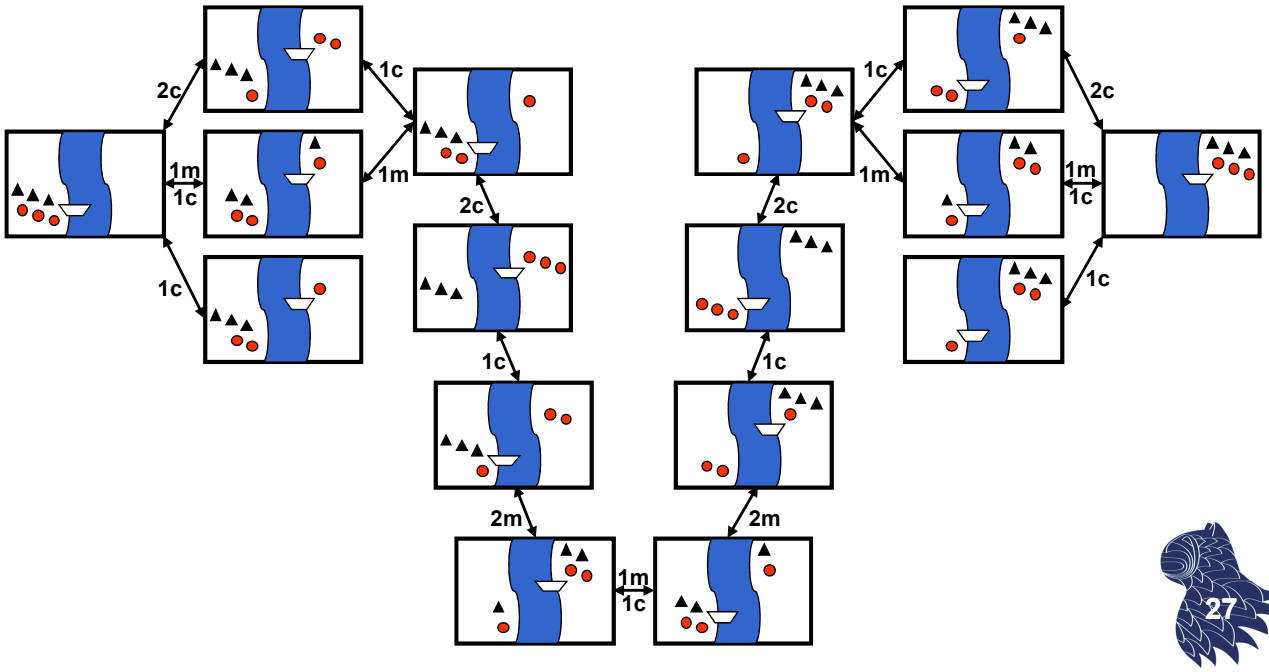
+1 por cada vez que alguien cruza el río

Solución óptima

4 soluciones de coste 11

Ejemplo: Misioneros y caníbales

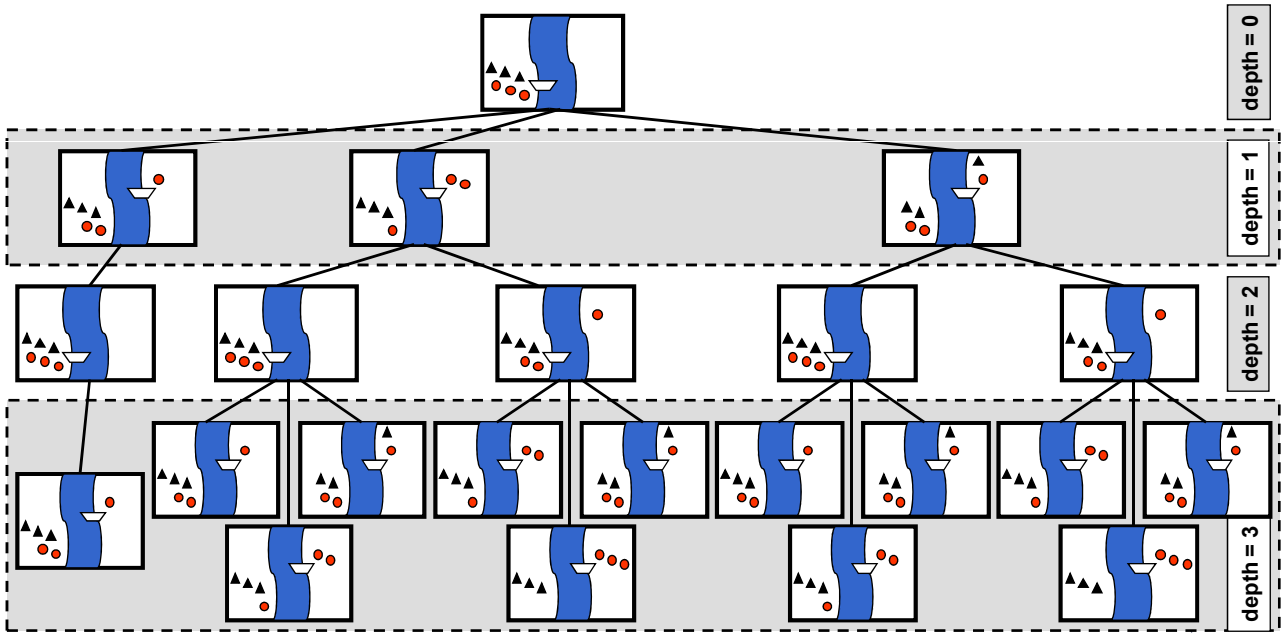
Espacio de estados



Ejemplo: Misioneros y caníbales

Estrategias de búsqueda

p.ej. búsqueda en anchura (o, mejor, IDS)



Planificación como búsqueda

Búsqueda hacia adelante (usando DFS)

estado ← estado inicial
plan ← ⟨⟩
mientras el estado no satisface el objetivo
    aplicables ← acciones válidas para estado
    si aplicables está vacío, devolver fallo
    acción ← seleccionar una acción de aplicables
    estado ← resultado(estado,acción)
    plan ← plan • ⟨acción⟩
devolver plan

Planificación como búsqueda

Búsqueda hacia adelante

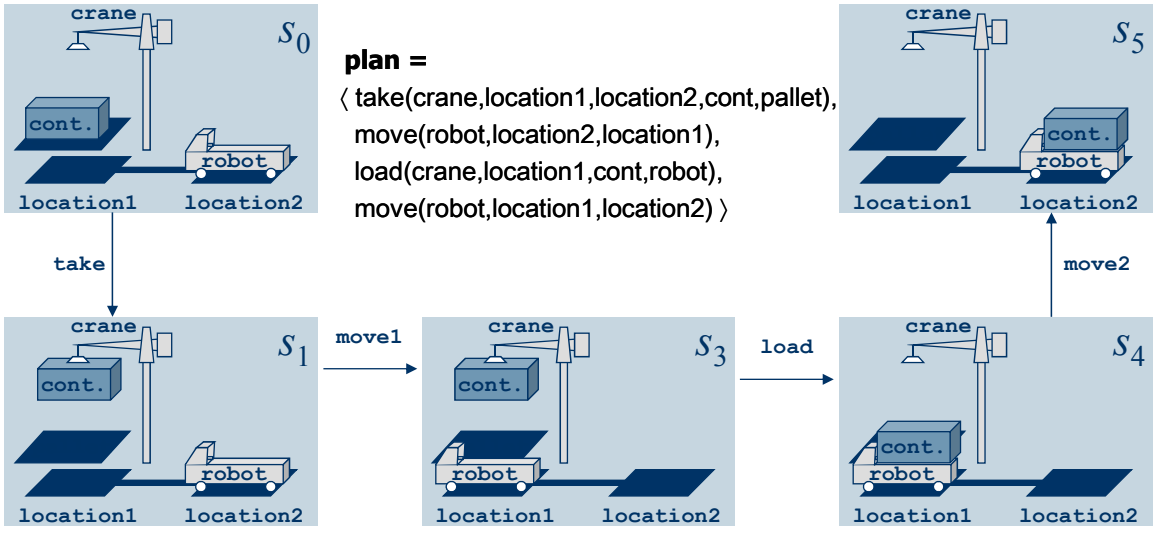
- La búsqueda hacia adelante es correcta
  (si devuelve un plan, este plan es una solución válida).

- La búsqueda hacia adelante es completa
  (si existe un plan, la búsqueda lo termina encontrando).

  NOTA: Hay que tener cuidado con los estados repetidos.

Planificación como búsqueda

Estado inicial **Objetivo**



Planificación como búsqueda

Búsqueda hacia adelante

Problema

- Número elevado de acciones aplicables en cada estado.
- Factor de ramificación demasiado grande.
- No resulta viable para planes con muchos pasos.

Una posible alternativa:
Buscar hacia atrás, partiendo del objetivo…

Planificación como búsqueda

Búsqueda hacia atrás (usando DFS)

subobjetivo ← objetivo final
plan ← ⟨⟩
mientras el estado inicial no satisface el subobjetivo
    aplicables ← acciones que nos llevan a subobjetivo
    si aplicables está vacío, devolver fallo
    acción ← seleccionar una acción de aplicables
    subobjetivo ← resultado^{-1}(subobjetivo,acción)
    plan ← ⟨acción⟩ • plan
devolver plan

Planificación como búsqueda

Búsqueda hacia atrás

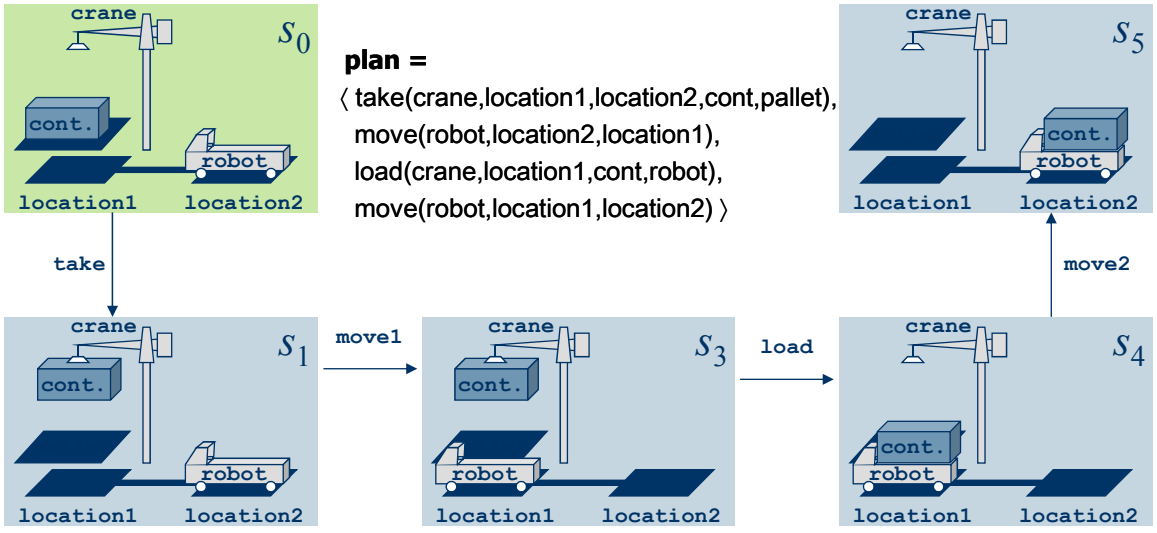
(en la práctica, se implementa admitiendo operadores parcialmente instanciados en vez de acciones simples).

- La búsqueda hacia atrás es correcta
  (si devuelve un plan, este plan es una solución válida).

- La búsqueda hacia atrás es completa
  (si existe un plan, la búsqueda lo termina encontrando).

Planificación como búsqueda

Estado inicial **Objetivo**



Planificación como búsqueda

Búsqueda hacia atrás

Problema

- Aunque el factor de ramificación sea, por lo general, menor que en la búsqueda hacia adelante, el espacio de búsqueda sigue siendo demasiado grande.

Planificación como búsqueda

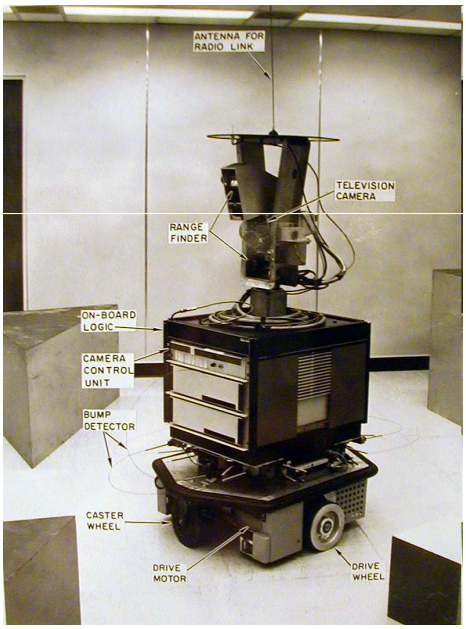
Una búsqueda genérica no parece la estrategia más adecuada para resolver un problema de planificación:

- No se tienen en cuenta las diferencias entre el estado inicial y el objetivo para elegir las acciones adecuadas.
- Los planes podrían empezar con acciones obvias y después refinarse (con acciones que quizá haya que realizar antes de las obvias).
- Muchos objetivos son compuestos y se podrían obtener de forma independiente (o, al menos, de forma relativamente independiente).

Planificación clásica: STRIPS

Stanford Research Institute Problem Solver
(Fikes & Nilsson, 1971)

- Planificador del robot Shakey.
- Basado en GPS [General Problem Solver], realiza una búsqueda en el espacio de estados.



Shakey (SRI, 1968)

Planificación clásica: STRIPS

Representación de un problema de planificación clásico:

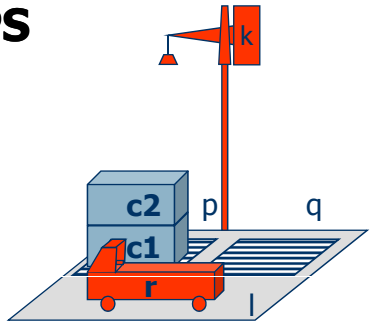
- Estado inicial.
- Descripción del objetivo que se desea conseguir.
- Conjunto de acciones.

STRIPS básico:

- Uso de predicados (lógica de primer orden).
- Sólo literales positivos.
- Sólo conjunciones de literales simples en el objetivo.
- Sólo se especifica aquello que cambia.
- Hipótesis de mundo cerrado.

Planificación clásica: STRIPS

Representación de estados en STRIPS



Estado inicial = {
  at(p,l), at(q,l), at(r,l), at(k,l),
  in(c1,p), in(c2,p),
  top(c2,p),
  on(c2,c1), on(c1,p),
  empty(k),
  unloaded(r)
}

Representación de objetivos en STRIPS

Objetivo = { on(c1,r) }

Planificación clásica: STRIPS

Representación de acciones en STRIPS

Acción
Nombre de la acción

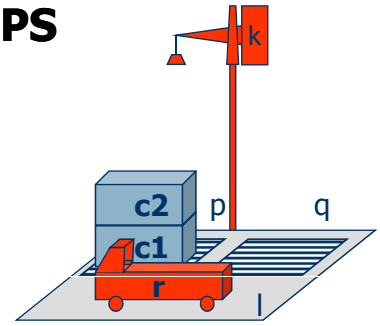
Precondiciones
Proposiciones que deben cumplirse para poder aplicar la acción.

Efectos
Consecuencias de la aplicación de la acción.

- "Add list" (proposiciones que pasan a ser ciertas)
- "Delete list" (proposiciones que pasan a ser falsas)

Planificación clásica: STRIPS

Representación de acciones en STRIPS



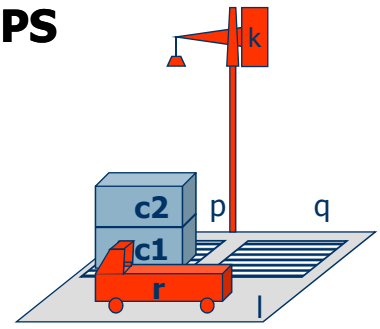
move(r,l,m)
- pre: adjacent(l,m), at(r,l)
- add: at(r,m)
- del: at(r,l)

load(k,l,c,r)
- pre: at(k,l), holding(k,c), at(r,l), unloaded(r)
- add: empty(k), loaded(r,c)
- del: holding(k,c), unloaded(r)

put(k,l,c,d,p)
- pre: at(k,l), at(p,l), holding(k,c), top(d,p)
- add: empty(k), in(c,p), top(c,p), on(c,d)
- del: holding(k,c), top(d,p)

Planificación clásica: STRIPS

Representación de acciones en STRIPS



move(r,l,m)
- pre: adjacent(l,m), at(r,l)
- eff: at(r,m), ¬at(r,l)

load(k,l,c,r)
- pre: at(k,l), holding(k,c), at(r,l), unloaded(r)
- eff: empty(k), loaded(r,c), ¬holding(k,c), ¬unloaded(r)

put(k,l,c,d,p)
- pre: at(k,l), at(p,l), holding(k,c), top(d,p)
- add: empty(k), in(c,p), top(c,p), on(c,d), ¬holding(k,c), ¬top(d,p)

Planificación clásica: STRIPS

Idea

- Si tenemos n subobjetivos que conseguir, puede que tengamos que probar n! formas distintas de ordenarlos hasta obtener la solución.
- Para reducir el espacio de búsqueda, trataremos de ir logrando los objetivos en cierto orden.

Problema

- Puede que deshagamos un objetivo ya conseguido.
- Supongamos que los objetivos son independientes
  (se pueden resolver separadamente)

  y, si existen interacciones,
  utilizaremos un mecanismo especial…

Planificación clásica: STRIPS

Análisis de medios y fines (técnica de GPS):

Mientras haya diferencias entre el estado actual y el objetivo:

- Escoger una diferencia entre el estado actual y el objetivo.
- Encontrar un operador que resulte apropiado para reducirla.
- Comparar las precondiciones del operador seleccionado con el estado actual, encontrando posibles diferencias:
  - Si no hay ninguna, se aplica directamente el operador.
  - Si hay diferencias, se intenta reducirlas recursivamente.
- Continuamos el proceso utilizando como estado actual el estado resultante de la aplicación del operador seleccionado.

Planificación clásica: STRIPS

Análisis de medios y fines (técnica de GPS):

- Técnica general
  (no demasiado potente para problemas grandes).
- Necesita objetivos independientes
  (y una ordenación de los objetivos)
- Localidad: No resulta apropiado
  cuando se necesita una estrategia global de resolución.
- Deben existir métodos para
  calcular diferencias y determinar cómo resolverlas.

Planificación clásica: STRIPS

Algoritmo básico de STRIPS

```
STRIPS(estado,objetivo)
   plan ← ⟨⟩
   mientras el estado no satisface el objetivo
      aplicables ← acciones relevantes (estado,objetivo)
      si aplicables está vacío, devolver fallo
      acción ← seleccionar una acción de aplicables
      subplan ← STRIPS(estado,precondiciones de acción)
      si subplan falla, devolver fallo
      estado ← resultado(estado, subplan • ⟨acción⟩)
      plan ← plan • subplan • ⟨acción⟩
   devolver plan
```

Planificación clásica: STRIPS

Algoritmo básico de STRIPS

Idea

- Introducir en una pila los objetivos por conseguir y los operadores que consiguen dichos objetivos.
- Extraer de la pila los objetivos que se cumplan en el estado actual (y los operadores que han de ejecutarse).

Implementación mediante:

- Una pila de objetivos.
- El estado actual del problema.

Planificación clásica: STRIPS

Algoritmo básico de STRIPS

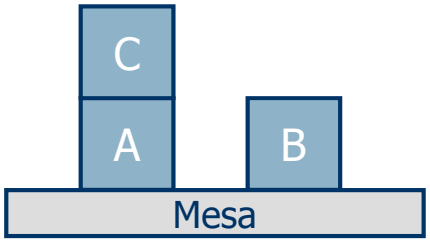
Funcionamiento

- Emparejamiento: Si el estado actual satisface el objetivo del tope de la pila, éste se puede eliminar.
- Descomposición: Los objetivos compuestos se descomponen en literales simples.
- Resolución: Cuando el objetivo de la parte superior de la pila sea un literal no resuelto, se busca un operador cuya lista de adición contenga un literal que empareje con él, se añade el operador al plan y sus precondiciones al objetivo.

Planificación clásica: STRIPS

El mundo de bloques

Acciones



Coger (x)
- pre: SOBREMESA(x), LIBRE(x),MANOVACIA
- eff: COGIDO(x), ¬SOBREMESA(x), ¬LIBRE(x), ¬MANOVACIA

Dejar (x)
- pre: COGIDO(x)
- eff: SOBREMESA(x), LIBRE(x),MANOVACIA, ¬COGIDO(x)

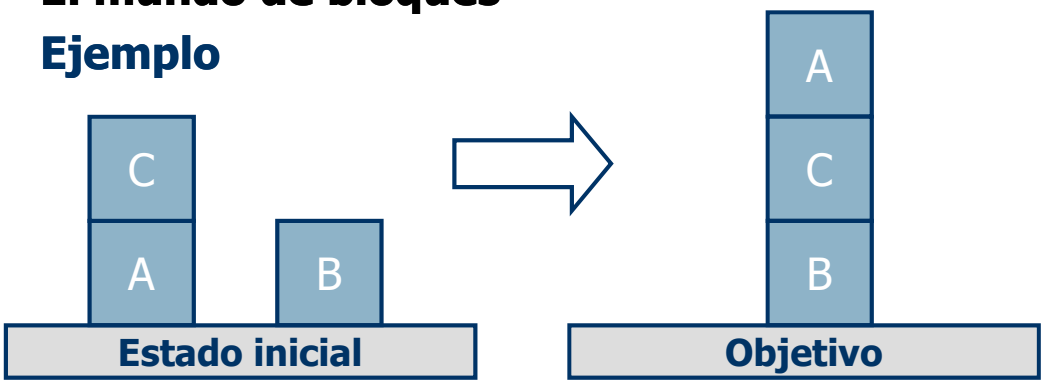
Apilar (x, y)
- pre: COGIDO(x), LIBRE(y)
- eff: MANOVACIA, SOBRE(x, y), LIBRE(x), ¬COGIDO(x), ¬LIBRE(y)

Desapilar (x, y)
- pre: MANOVACIA, LIBRE(x), SOBRE(x, y)
- eff: COGIDO(x), LIBRE(y), ¬MANOVACIA, ¬LIBRE(x), ¬SOBRE(x, y)

Planificación clásica: STRIPS

El mundo de bloques

Ejemplo



LIBRE(B)
LIBRE(C)
SOBRE(C,A)
SOBREMESA(A)
SOBREMESA(B)
MANOVACÍA

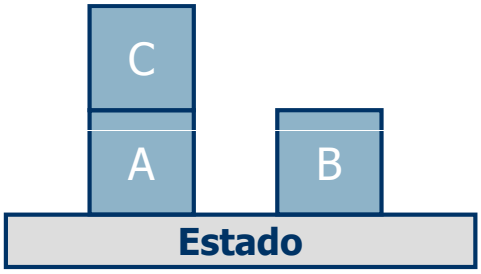
SOBRE(C,B) ∧ SOBRE(A,C)

Pila

Planificación clásica: STRIPS

El mundo de bloques

Ejemplo



LIBRE(B)
LIBRE(C)
SOBRE(C,A)
SOBREMESA(A)
SOBREMESA(B)
MANOVACÍA

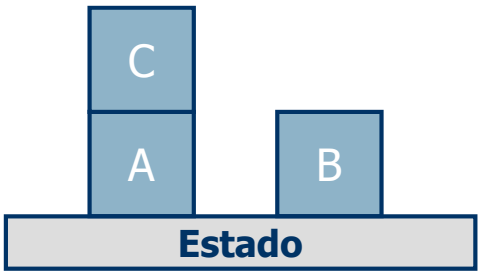
SOBRE(C,B)
SOBRE(A,C)
SOBRE(C,B) ∧ SOBRE(A,C)

Pila

Planificación clásica: STRIPS

El mundo de bloques

Ejemplo



LIBRE(B)
LIBRE(C)
SOBRE(C,A)
SOBREMESA(A)
SOBREMESA(B)
MANOVACÍA

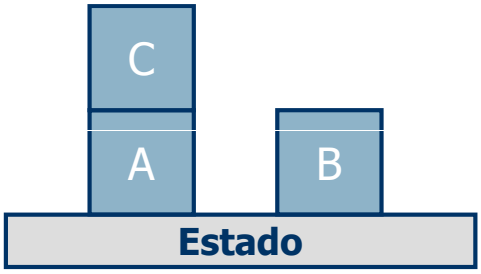
LIBRE(B) ∧ COGIDO(C)
Apilar(C,B)
SOBRE(A,C)
SOBRE(C,B) ∧ SOBRE(A,C)

Pila

Planificación clásica: STRIPS

El mundo de bloques

Ejemplo



LIBRE(B)
LIBRE(C)
SOBRE(C,A)
SOBREMESA(A)
SOBREMESA(B)
MANOVACÍA

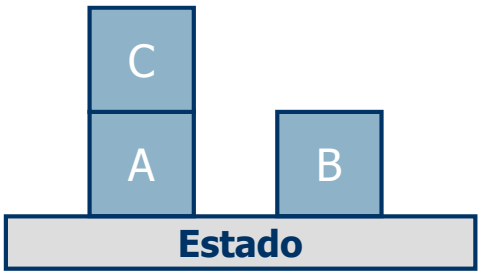
COGIDO(C)
LIBRE(B)
LIBRE(B) ∧ COGIDO(C)
Apilar(C,B)
SOBRE(A,C)
SOBRE(C,B) ∧ SOBRE(A,C)

Pila

Planificación clásica: STRIPS

El mundo de bloques

Ejemplo



LIBRE(B)
LIBRE(C)
SOBRE(C,A)
SOBREMESA(A)
SOBREMESA(B)
MANOVACÍA

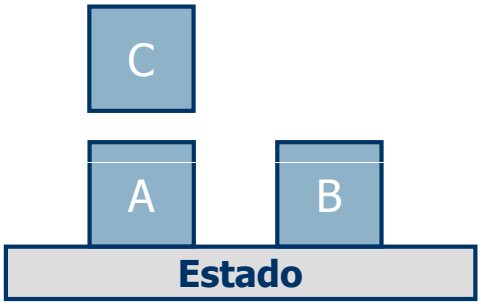
MANOVACÍA ∧ LIBRE(C) ∧ SOBRE(C,y)
Desapilar(C,y)
LIBRE(B)
LIBRE(B) ∧ COGIDO(C)
Apilar(C,B)
SOBRE(A,C)
SOBRE(C,B) ∧ SOBRE(A,C)

Pila

Planificación clásica: STRIPS

El mundo de bloques

Ejemplo



LIBRE(B)
COGIDO(C)
LIBRE(A)
SOBREMESA(A)
SOBREMESA(B)

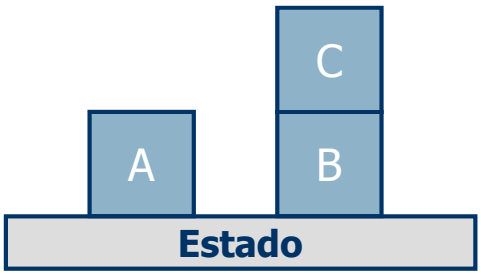
LIBRE(B)
LIBRE(B) ∧ COGIDO(C)
Apilar(C,B)
SOBRE(A,C)
SOBRE(C,B) ∧ SOBRE(A,C)

Pila

Planificación clásica: STRIPS

El mundo de bloques

Ejemplo



LIBRE(C)
SOBRE(C,B)
LIBRE(A)
SOBREMESA(A)
SOBREMESA(B)
MANOVACÍA

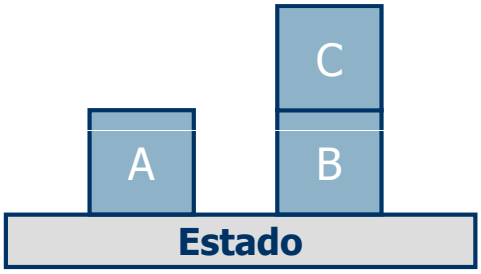
SOBRE(A,C)
SOBRE(C,B) ∧ SOBRE(A,C)

Pila

Planificación clásica: STRIPS

El mundo de bloques

Ejemplo



LIBRE(C)
SOBRE(C,B)
LIBRE(A)
SOBREMESA(A)
SOBREMESA(B)
MANOVACÍA

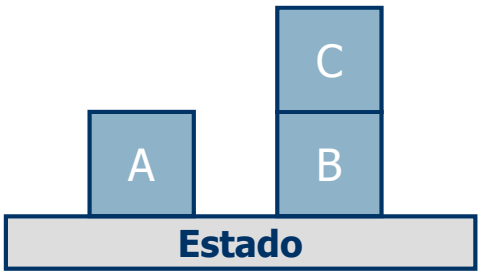
LIBRE(C) ∧ COGIDO(A)
Apilar(A,C)
SOBRE(C,B) ∧ SOBRE(A,C)

Pila

Planificación clásica: STRIPS

El mundo de bloques

Ejemplo



LIBRE(C)
SOBRE(C,B)
LIBRE(A)
SOBREMESA(A)
SOBREMESA(B)
MANOVACÍA

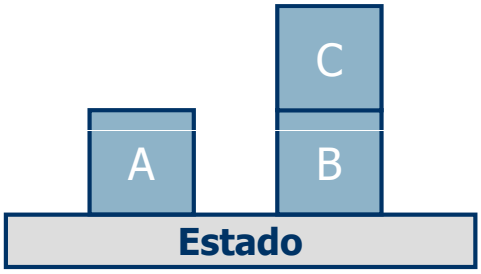
LIBRE(C)
COGIDO(A)
LIBRE(C) ∧ COGIDO(A)
Apilar(A,C)
SOBRE(C,B) ∧ SOBRE(A,C)

Pila

Planificación clásica: STRIPS

El mundo de bloques

Ejemplo



LIBRE(C)
SOBRE(C,B)
LIBRE(A)
SOBREMESA(A)
SOBREMESA(B)
MANOVACÍA

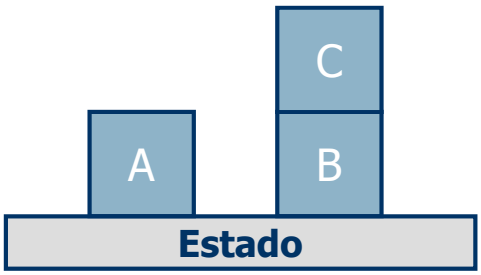
COGIDO(A)
LIBRE(C) ∧ COGIDO(A)
Apilar(A,C)
SOBRE(C,B) ∧ SOBRE(A,C)

Pila

Planificación clásica: STRIPS

El mundo de bloques

Ejemplo



LIBRE(C)
SOBRE(C,B)
LIBRE(A)
SOBREMESA(A)
SOBREMESA(B)
MANOVACÍA

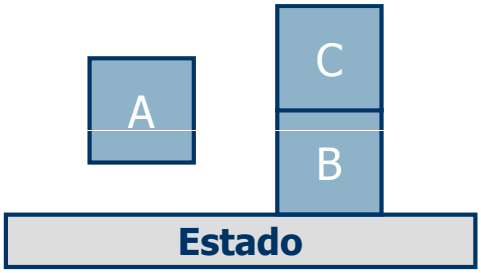
SOBREMESA(A) ∧ LIBRE (A) ∧ MANOVACÍA
Coger(A)
LIBRE(C) ∧ COGIDO(A)
Apilar(A,C)
SOBRE(C,B) ∧ SOBRE(A,C)

Pila

Planificación clásica: STRIPS

El mundo de bloques

Ejemplo



LIBRE(C)
SOBRE(C,B)
COGIDO(A)

SOBREMESA(B)

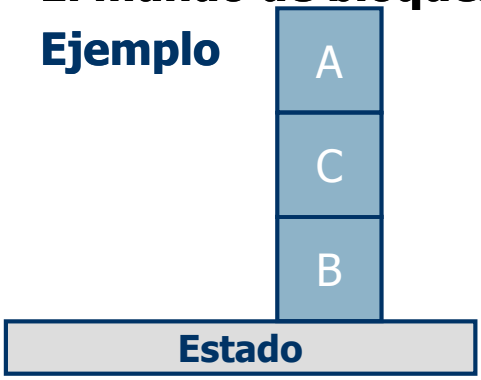
LIBRE(C) ∧ COGIDO(A)
Apilar(A,C)
SOBRE(C,B) ∧ SOBRE(A,C)

Pila

Planificación clásica: STRIPS

El mundo de bloques

Ejemplo



SOBRE(C,B)
SOBRE(A,C)
MANOVACÍA
SOBREMESA(B)

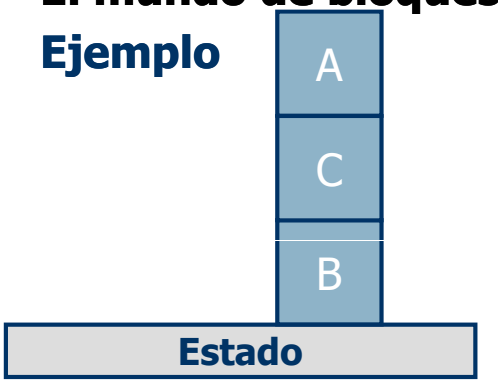
SOBRE(C,B) ∧ SOBRE(A,C)

Pila

Planificación clásica: STRIPS

El mundo de bloques

Ejemplo



SOBRE(C,B)
SOBRE(A,C)
MANOVACÍA
SOBREMESA(B)

Pila vacía

Planificación clásica: STRIPS

Algoritmo básico de STRIPS

Algoritmo incompleto

- No puede encontrar la solución para algunos problemas.

  p.ej. Intercambiar el valor de dos variables

- No encuentra la solución óptima para otros.

  p.ej. Anomalía de Sussman

(después de lograr un objetivo, éste se deshace al resolver el siguiente)

Planificación clásica: STRIPS

Algoritmo básico de STRIPS

Intercambio del valor de dos variables

- Estado inicial: CONTIENE(X,A) ∧ CONTIENE(Y,B)
- Objetivo: CONTIENE(X,B) ∧ CONTIENE(Y,A)

Operación de asignación:

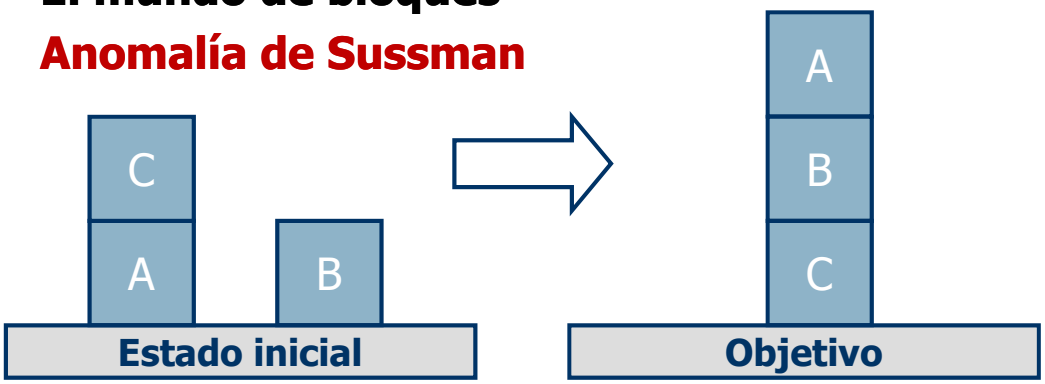
Asigna (u, r, t, s)
- pre: CONTIENE(r, s) ∧ CONTIENE(u, t)
- del: CONTIENE(u, t)
- add: CONTIENE(u, s)

STRIPS no puede encontrar la solución :-(

Planificación clásica: STRIPS

El mundo de bloques

Anomalía de Sussman



LIBRE(B)
LIBRE(C)
SOBRE(C,A)
SOBREMESA(A)
SOBREMESA(B)
MANOVACÍA

SOBRE(B,C) ∧ SOBRE(A,B)

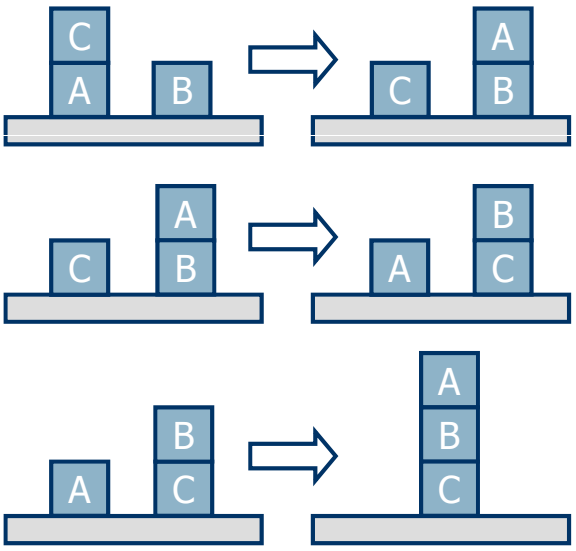
Pila

Planificación clásica: STRIPS

El mundo de bloques

Anomalía de Sussman

Caso 1: SOBRE(A,B)



SOBRE(A,B)
- Dejar C sobre la mesa.
- Colocar A sobre B.

SOBRE(B,C)
- Dejar A en la mesa.
- Colocar B sobre C.

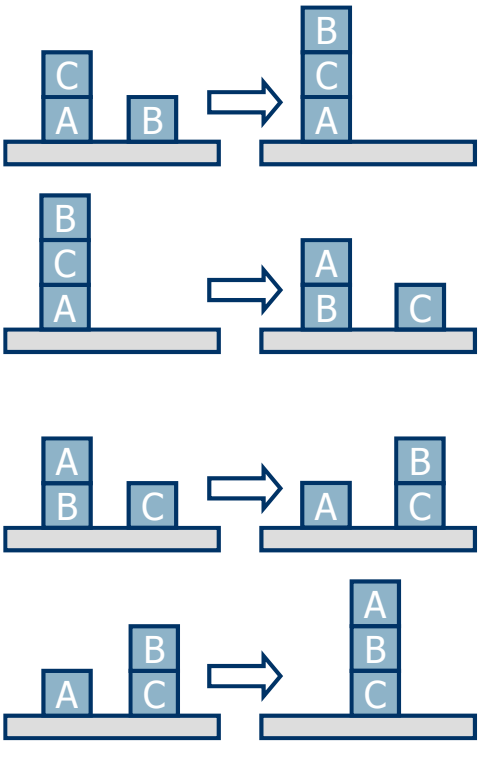
SOBRE(A,B) **[bis]**
- Colocar A sobre B.

Planificación clásica: STRIPS

El mundo de bloques

Anomalía de Sussman

Caso 2: SOBRE(B,C)



SOBRE(B,C)
- Colocar B sobre C.

SOBRE(A,B)
- Dejar B sobre la mesa.
- Dejar C sobre la mesa.
- Colocar A sobre B.

SOBRE(B,C) **[bis]**
- Dejar A sobre la mesa.
- Colocar B sobre C.

SOBRE(A,C) **[bis]**
- Colocar A sobre B

Planificación clásica: STRIPS

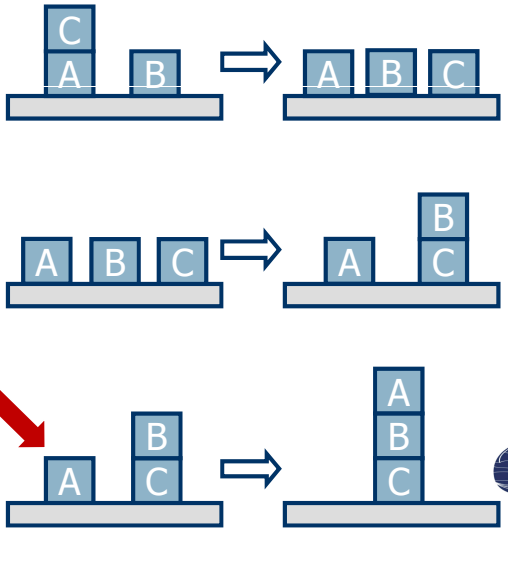
El mundo de bloques

Anomalía de Sussman

Entrelazado de planes para lograr la solución óptima

Solución más corta para SOBRE(A,B)
- Dejar C sobre la mesa.
- Colocar A sobre B.

Solución más corta para SOBRE(B,C)
- Colocar B sobre C.



Búsqueda en el espacio de planes

Trabajamos sobre planes que, al principio,
son simples e incompletos (planes parciales).

Los operadores actúan sobre los planes,
modificándolos hasta llegar a
un plan completo que resuelva el problema.

Búsqueda en el espacio de planes

- En la búsqueda sobre el espacio de estados, los estados de la búsqueda eran situaciones concretas del mundo y los operadores eran acciones que cambiaban esas situaciones.

- En la búsqueda sobre el espacio de planes, los estados de la búsqueda son planes (secuencias de acciones) y los operadores cambian estos planes (en general, incompletos) hasta conseguir un plan completo que resuelva el problema.

Búsqueda en el espacio de planes

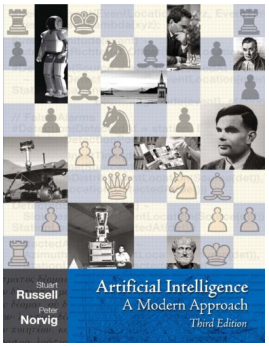
Tipos de operadores

- **Operadores de refinamiento**
  Escogen un plan parcial y le añaden restricciones (eliminan algunos planes completos del conjunto de planes compatibles con el plan parcial).

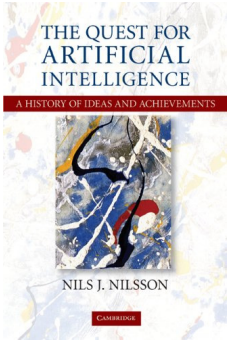
- **Operadores de modificación**
  Cambian un plan
  (permiten depurar planes incorrectos).

Bibliografía

- Stuart Russell & Peter Norvig:
  Artificial Intelligence:
  A Modern Approach
  Prentice-Hall, 3rd edition, 2009
  ISBN 0136042597
  http://aima.cs.berkeley.edu/



- Nils J. Nilsson
  The Quest for Artificial Intelligence
  Cambridge University Press, 2009
  ISBN 0521122937
  http://ai.stanford.edu/~nilsson/QAI/qai.pdf



Bibliografía

Cursos de planificación

- **MSC Automated Planning**
  School of Informatics
  University of Edinburgh
  http://www.inf.ed.ac.uk/teaching/courses/plan/
  También en Coursera:
  https://www.coursera.org/course/aiplan

- **CS541: Artificial Intelligence Planning**
  USC Viterbi School of Engineering
  University of Southern California
  http://www.isi.edu/~blythe/cs541/