

TP6

September 19, 2025

1 Temas Tratados en el Trabajo Práctico 6

- Modelado de problemas en espacios de estado.
- Algoritmos de planificación hacia adelante y hacia atrás.
- Representación y solución de problemas descritos en lenguaje STRIPS.
- Algoritmo GRAPHPLAN.
- Planificación con restricciones de tiempo y recursos.
- Caminos críticos y tiempos de relajación.

1.1 Ejercicios Teóricos

1. ¿En qué tipo de algoritmos se basa un planificador para encontrar el mejor camino a un estado solución?

1.1.1 Algoritmos

Un planificador, para encontrar el mejor camino a un estado solución, se basa en algoritmos de búsqueda en grafos, pudiendo estos ser:

- a) No informados: No tienen conocimientos del problema, solo exploran.
- b) Informados: Aprovechan información extra para guiar la exploración hacia el objetivo.
- c) Especializados en planificación: Se basan en los anteriores, pero adaptados al lenguaje de planificación (Ejemplo: GraphPlan).

-
2. ¿Qué tres elementos se encuentran dentro de una acción formulada en lenguaje STRIPS? Describa brevemente qué función cumple cada uno.

1.1.2 Elementos de una acción

Una acción formulada en lenguaje STRIPS esta compuesta por tres elementos, siendo:

- a) Nombre y parámetros: Identifican la acción y los objetos que intervienen.
- b) Precondiciones: describen lo que debe cumplirse en el estado actual para que la acción sea ejecutable.
- c) Efectos: indican los cambios en el estado del mundo una vez que la acción se ejecuta.

-
3. Describa las ventajas y desventajas de desarrollar un algoritmo de planificación hacia adelante y hacia atrás en el espacio de estados.

1.1.3 Ventajas y Desventajas

Planificación hacia adelante

Ventajas:

1. Implementación simple: Directo, ya que se empieza con lo que se sabe seguro.
2. Es consistente. Cada acción aplicada genera estados alcanzables en el mundo real. No se crean “estados imposibles”.
3. Muy útil cuando el espacio de estados no es tan grande o cuando es fácil verificar si se llegó a la meta.

Desventajas:

1. El espacio de búsqueda puede crecer muchísimo.
2. Puede perder tiempo generando estados irrelevantes si no se usan buenas heurísticas.

Planificación hacia atrás

Ventajas:

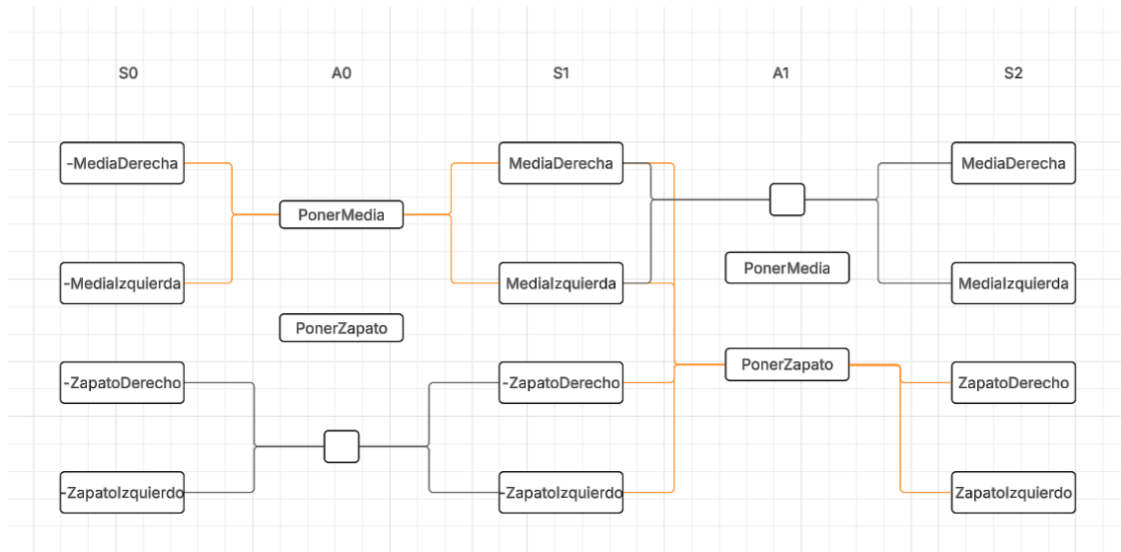
1. Enfoca la búsqueda en estados más relevantes para alcanzar la meta.
2. Puede ser más eficiente cuando el objetivo está muy bien definido y es específico.

Desventajas:

1. No siempre es fácil encontrar acciones que “deshagan” el objetivo.
2. Puede llegar a estados intermedios que son inconsistentes, que no sean factibles en el mundo real.
3. Es más complejo de implementar en algunos dominios.

-
4. Considere el problema de ponerse uno mismo zapatos y medias. Aplique GRAPHPLAN a este problema y muestre la solución obtenida. Muestre el plan de orden parcial que es solución e indique cuántas linealizaciones diferentes existen para el plan de orden parcial.

1.1.4 GraphPlan



Procedimiento Secuencial con:

-MediaDerecha, -MediaIzquierda, -ZapatoDerecho y -ZapatoIzquierdo

Pasos:

- PonerMedia en derecha - Resultado: MediaDerecha, -MediaIzquierda, -ZapatoDerecho y -ZapatoIzquierdo.
- Poner Media en izquierda - Resultado: MediaDerecha, MediaIzquierda, -ZapatoDerecho y -ZapatoIzquierdo.
- PonerZapato en derecha - Resultado: MediaDerecha, MediaIzquierda, ZapatoDerecho y -ZapatoIzquierdo.
- PonerZapato en izquierda - Resultado: MediaDerecha, MediaIzquierda, ZapatoDerecho y ZapatoIzquierdo.

Secuencia de solución: PonerMedia_der - PonerMedia_izq - PonerZapato_der - PonerZapato_izq

5. Se requiere ensamblar una máquina cuyas piezas están identificadas con las letras A, B, C, D y E. El tiempo que se tarda en ensamblar cada pieza es:

- A: 2 semanas
- B: 1 semana
- C: 4 semanas
- D: 3 semanas
- E: 5 semanas

El orden de ensamblaje de cada pieza requiere que:

- A esté realizado antes que C
- B esté realizado antes que C
- B esté realizado antes que D
- C esté realizado antes que E
- D esté realizado antes que E

Con esta información:

5.1 Arme el Plan de Orden Parcial.

5.2 Encuentre el Camino Crítico.

5.3 Encuentre los tiempos de relajación.

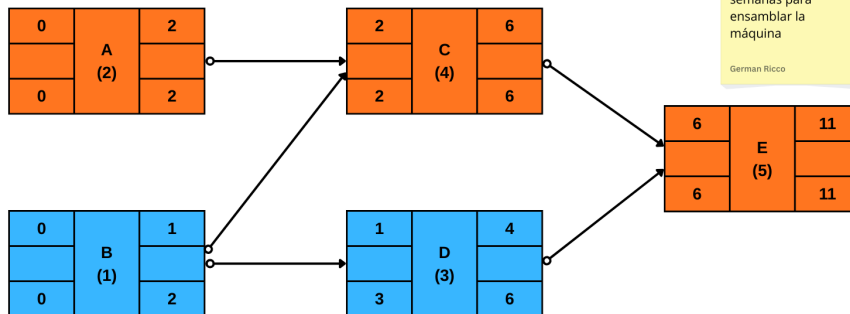
5.4 Dibuje un diagrama temporal indicando las tareas y los tiempos de relajación encontrados.

1.1.5 Plan de Orden Total



Camino Crítico: A → C → E

Tiempos de Relajación: • B → 1 semana
• C → 2 semanas



Como mínimo se necesitarán 11 semanas para ensamblar la máquina

German Ricco

1.1.6 Diagrama Temporal de Gantt



1.2 Ejercicios de Implementación

6. Suponga que tiene un robot de oficina capaz de moverse y tomar y depositar objetos. El robot solo puede tener un objeto a la vez, pero puede conseguir una *caja* en la que depositar varios objetos. Suponga que programa al robot para *ir a la tienda* a comprarle un *café* y en el camino de vuelta tome una *carta* del *buzón* de la oficina para para que se la traiga junto con el café. Describa en lenguaje STRIPS:

6.1 El dominio del robot (nombre, predicados y acciones que puede hacer el robot).

6.2 El problema que se quiere resolver (estado inicial, estado objetivo y objetos del mundo).

6.3 Introduzca el código desarrollado en los puntos anteriores en el [planificador online].

1.2.1 DOMAIN

No se logró obtener un plan de acciones adecuado, así como está funciona pero no cumple los requisitos

```
[1]: (define (domain gripper-strips)
      (:predicates (room ?r)
                   (item ?i)
                   (box ?b)
                   (gripper ?g)
                   (at-robby ?r)
                   (at ?i ?r)
                   (free ?g)
                   (carry ?o ?g))
```

```

(in ?o ?b))

(:action move
  :parameters (?from ?to)
  :precondition (and (room ?from) (room ?to) (at-robby ?from))
  :effect (and (at-robby ?to)
               (not (at-robby ?from))))

(:action pick
  :parameters (?obj ?room ?gripper)
  :precondition (and (item ?obj) (room ?room) (gripper ?gripper)
                    (at ?obj ?room) (at-robby ?room) (free ?gripper))
  :effect (and (carry ?obj ?gripper)
               (not (at ?obj ?room))
               (not (free ?gripper))))

(:action put-into
  :parameters (?obj ?box ?room ?gripper)
  :precondition (and (item ?obj) (box ?box) (room ?room) (gripper ?gripper)
                    (at-robby ?room) (carry ?obj ?gripper) (at ?box ?room))
  :effect (and (in ?obj ?box) (free ?gripper) (not (carry ?obj ?g))))
)

(:action drop
  :parameters (?obj ?room ?gripper)
  :precondition (and (item ?obj) (room ?room) (gripper ?gripper)
                    (carry ?obj ?gripper) (at-robby ?room))
  :effect (and (at ?obj ?room)
               (free ?gripper)
               (not (carry ?obj ?gripper))))
)

```

Cell In[1], line 1

```
(define (domain gripper-strips)
```

```
~
```

SyntaxError: invalid syntax. Perhaps you forgot a comma?

1.2.2 PROBLEM

```

[ ]: (define (problem strips-gripper-x-1)
      (:domain gripper-strips)
      (:objects office store mailbox box1 coffee letter efector)
      (:init (room office)

```

```

(room store)
(room mailbox)
(item letter)
(item coffee)
(item box1)
(box box1)
(at-robbly office)
(free efector)
(at box office)
(at coffee store)
(at letter mailbox)
(gripper efector))

(:goal (and (at box office)
            (at coffee office) ;ideal seria (in coffee box)
            (at letter office)))) ; ideal seria (in letter box)
;LO DEJO ASI PORQUE NO LOGRÉ QUE FUNCIONE

```

2 Bibliografía

Russell, S. & Norvig, P. (2004) *Inteligencia Artificial: Un Enfoque Moderno*. Pearson Educación S.A. (2a Ed.) Madrid, España

Poole, D. & Mackworth, A. (2023) *Artificial Intelligence: Foundations of Computational Agents*. Cambridge University Press (3a Ed.) Vancouver, Canada