

# Planificación Automática

Grupo PLG

Universidad Carlos III de Madrid

IA. 2008-09

# Índice

- 1 **Introducción**
- 2 Planificación clásica
- 3 Planificación neoclásica
- 4 Heurística
  - Planificación heurística
  - Redes de tareas jerárquicas
  - Conocimiento de control
  - Aprendizaje automático
- 5 Planificación en el mundo real

# Índice

- 1 **Introducción**
- 2 **Planificación clásica**
- 3 Planificación neoclásica
- 4 Heurística
  - Planificación heurística
  - Redes de tareas jerárquicas
  - Conocimiento de control
  - Aprendizaje automático
- 5 Planificación en el mundo real

# Índice

- 1 Introducción
- 2 Planificación clásica
- 3 Planificación neoclásica
- 4 Heurística
  - Planificación heurística
  - Redes de tareas jerárquicas
  - Conocimiento de control
  - Aprendizaje automático
- 5 Planificación en el mundo real

# Índice

- 1 Introducción
- 2 Planificación clásica
- 3 Planificación neoclásica
- 4 Heurística
  - Planificación heurística
  - Redes de tareas jerárquicas
  - Conocimiento de control
  - Aprendizaje automático
- 5 Planificación en el mundo real

# Índice

- 1 Introducción
- 2 Planificación clásica
- 3 Planificación neoclásica
- 4 Heurística
  - Planificación heurística
  - Redes de tareas jerárquicas
  - Conocimiento de control
  - Aprendizaje automático
- 5 Planificación en el mundo real

# Índice

- 1 **Introducción**
- 2 Planificación clásica
- 3 Planificación neoclásica
- 4 Heurística
  - Planificación heurística
  - Redes de tareas jerárquicas
  - Conocimiento de control
  - Aprendizaje automático
- 5 Planificación en el mundo real

# Índice

- 1 Introducción
- 2 Planificación clásica**
- 3 Planificación neoclásica
- 4 Heurística
  - Planificación heurística
  - Redes de tareas jerárquicas
  - Conocimiento de control
  - Aprendizaje automático
- 5 Planificación en el mundo real

# Índice

- 1 Introducción
- 2 Planificación clásica
- 3 Planificación neoclásica**
- 4 Heurística
  - Planificación heurística
  - Redes de tareas jerárquicas
  - Conocimiento de control
  - Aprendizaje automático
- 5 Planificación en el mundo real

# Índice

- 1 Introducción
- 2 Planificación clásica
- 3 Planificación neoclásica
- 4 Heurística**
  - Planificación heurística
  - Redes de tareas jerárquicas
  - Conocimiento de control
  - Aprendizaje automático
- 5 Planificación en el mundo real

# Heurísticas como problemas relajados

- Origen de las heurísticas: soluciones óptimas a problemas relajados [Pearl, 1983]
- La relajación se obtiene eliminando las listas de *eliminados* [Bonet and Geffner, 2001]: dado un problema  $\mathcal{P} = (\mathcal{O}, \mathcal{I}, \mathcal{G})$ , la relajación  $\mathcal{P}'$  de  $\mathcal{P}$  se define como  $\mathcal{P}' = (\mathcal{O}', \mathcal{I}, \mathcal{G})$  donde:

$$\mathcal{O}' = \{(pre(o), add(o), \emptyset) \mid (pre(o), add(o), del(o)) \in \mathcal{O}\}$$

- Una *secuencia de acciones* es un **plan relajado** si es una solución del problema relajado  $\mathcal{P}'$  del problema original  $\mathcal{P}$ :
  - Cuanto más parecido sea  $\mathcal{P}'$  a  $\mathcal{P}$ , más informada será la función heurística,  $h(\cdot)$
  - Cuanto más simplificado sea  $\mathcal{P}'$ , más fácil será computar  $h(\cdot)$

# Relajación sobre la alcanzabilidad

- Se define la distancia mínima del estado  $s$  al literal  $p$  como el número mínimo de acciones requeridas para transitar de  $s$  a un estado que contenga  $p$ :

$$g_s(p) = \begin{cases} 0 & \text{si } p \in s \\ \min_{o \in O(p)} [1 + g_s(\text{pre}(o))] & \text{en otro caso} \end{cases}$$

- El coste  $g_s(C)$  de un conjunto de literales puede definirse como:
  - Costes aditivos:  $g_s^+(C) = \sum_{r \in C} g_s(r)$
  - Costes máx:  $g_s^{\max}(C) = \max_{r \in C} g_s(r)$

## Planificación heurística

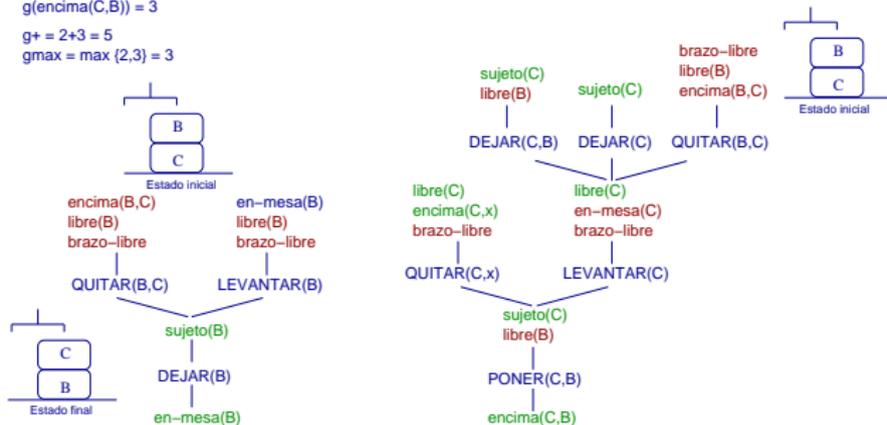
## Ejemplo

$$g(\text{en-mesa}(B)) = 2$$

$$g(\text{encima}(C,B)) = 3$$

$$g^+ = 2+3 = 5$$

$$g_{\max} = \max(2,3) = 3$$



# Heuristic Search Planning (HSP)

- HSP: emplea la función heurística  $h_{add} = g_s^+$  con un algoritmo de escalada hacia adelante
- HSP2: emplea la función heurística  $h_{add}$  con un algoritmo BFS hacia adelante
- Inconvenientes:
  - El algoritmo consume el 80% del tiempo en computar  $h(\cdot)$
  - No considera la interacción entre operadores, es decir, busca *ordenaciones secuenciales subóptimas* en contraposición con las *ordenaciones paralelas óptimas*
- Alternativas:
  - HSPr (empleando regresión), GRT (búsqueda bidireccional) o, más recientemente HSP\*
  - Utilizar GRAPHPLAN para capturar la interacción entre operadores

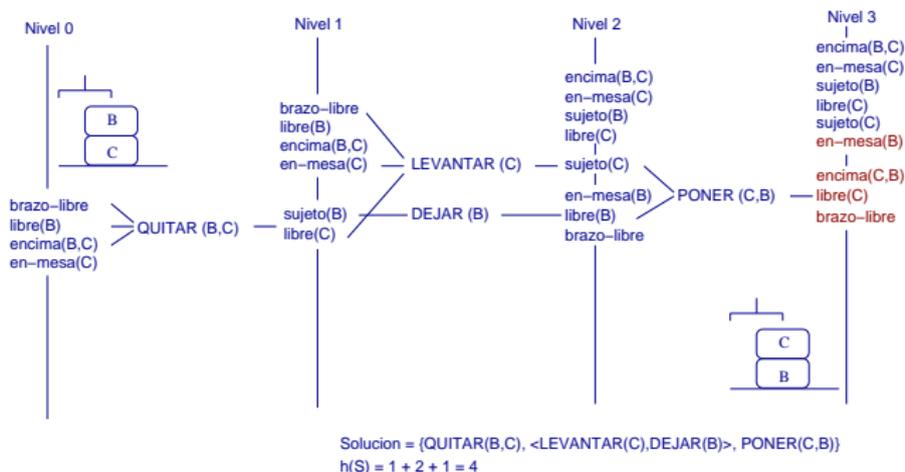
## GRAPHPLAN como heurística

- Sea  $\mathcal{P}' = (\mathcal{O}', \mathcal{I}, \mathcal{G})$  un problema relajado. GRAPHPLAN no marcará ningún par de hechos o acciones como excluyentes, ¡*puesto que no existen eliminados!*
- GRAPHPLAN encuentra una solución a  $\mathcal{P}'$  en un tiempo polinomial en  $l$  (la mayor lista de añadidos),  $|\mathcal{I}|$  y  $|\mathcal{O}'|$ :  $\langle O_0, O_1, \dots, O_{m-1} \rangle$  donde  $O_i$  es el conjunto de operadores elegidos en paralelo en el nivel  $i$  y  $m$  el primer nivel que contiene todas las soluciones
- FF emplea la función heurística:

$$h(S) = \sum_{i=0, \dots, m-1} |O_i|$$

- Típicamente  $h(S) \leq h_{add}$

## Ejemplo



# Fast-Forward Plan Generation (FF)

- FF [Hoffmann and Nebel, 2001]: emplea la función heurística  $h(S)$  con un algoritmo de escalada *forzado* que es sustituido por un BFS cuando escalada no encuentra un nodo solución
- El cómputo del GRAPHPLAN es *mejorado* intentando devolver las soluciones más cortas:
  - NOOPs-First
  - Medidas de dificultad

$$dif(o) = \sum_{p \in pre(o)} \min\{i | p \text{ es un miembro del nivel } i\}$$

- Conjunto de acciones *linearizadas*

# FF [Hoffmann and Nebel, 2001]

- Técnica de búsqueda: EHC, un tipo de técnica en escalada
- Si falla y tiene más tiempo, llama a un A\*
- Usa una función heurística  $h(S)$ : graphplan relajado
- En plateaux, elige una variante de búsqueda en amplitud
- Prefiere las acciones que ayudan (*helpful actions*)
- El cómputo de la heurística se mejora prefiriendo los NO-OPs y las acciones con menor dificultad

# Índice

- 1 Introducción
- 2 Planificación clásica
- 3 Planificación neoclásica
- 4 Heurística
  - Planificación heurística
  - Redes de tareas jerárquicas
  - Conocimiento de control
  - Aprendizaje automático
- 5 Planificación en el mundo real

## Referencias



Blai Bonet and Hector Geffner.

Planning as heuristic search.

*Artificial Intelligence*, 129(1-2):5–33, 2001.



Jörg Hoffmann and Bernhard Nebel.

The FF planning system: Fast plan generation through heuristic search.

*Journal of Artificial Intelligence Research*, 14:253–302, 2001.



Judea Pearl.

*Heuristics: Intelligent Search Strategies for Computer Problem Solving*.

Addison-Wesley, 1983.