

Razonamiento Temporal en Planificación Automática

**Escuela de Verano de Inteligencia Artificial
Carmona, Junio 2016**

Eva Onaindía

Universitat Politècnica de València

Indice

- 1. Planificación y scheduling**
- 2. Planificación temporal**
- 3. Lenguaje PDDL3.0**
- 4. Gestión de plazos de tiempo (*deadlines*)**

1. El problema de planificación

Es un problema de **búsqueda** que consiste en encontrar un conjunto de **acciones** que permiten a un sistema o agente alcanzar unos **objetivos** a partir de un **estado inicial** del mundo

1. El problema de planificación

Es un problema de **búsqueda** que consiste en encontrar un conjunto de **acciones** que permiten a un sistema o agente alcanzar unos **objetivos** a partir de un **estado inicial** del mundo

Entrada de un planificador:

1. estado inicial
2. objetivos alcanzar
3. acciones aplicables (operadores) en el dominio

1. El problema de planificación

Es un problema de **búsqueda** que consiste en encontrar un conjunto de **acciones** que permiten a un sistema o agente alcanzar unos **objetivos** a partir de un **estado inicial** del mundo

Entrada de un planificador:

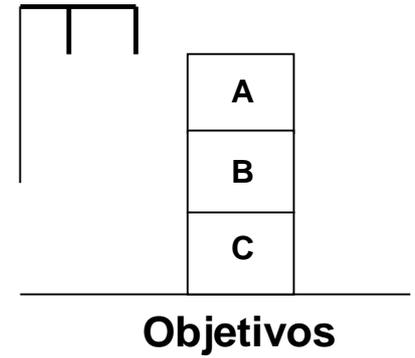
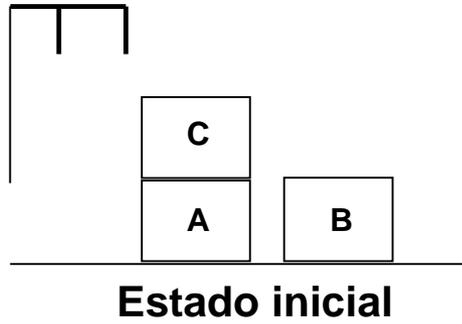
1. estado inicial
2. objetivos alcanzar
3. acciones aplicables (operadores) en el dominio

Salida de un planificador: un plan o conjunto de acciones parcialmente ordenadas

1. El problema de planificación

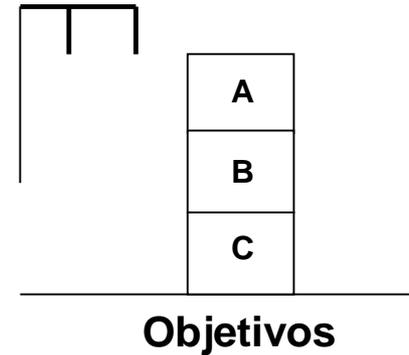
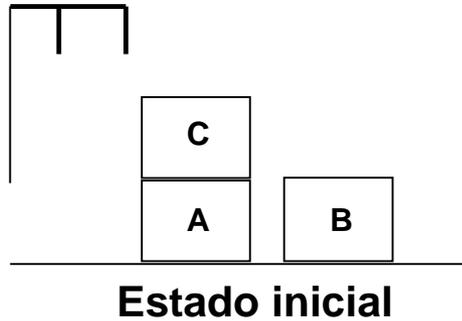
1. El problema de planificación

Ejemplo: mundo de bloques



1. El problema de planificación

Ejemplo: mundo de bloques

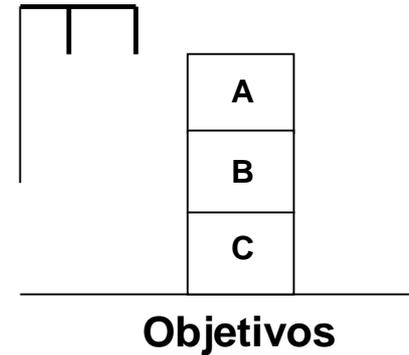
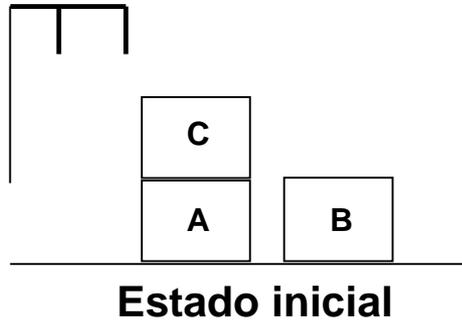


```
(:init
  (sobre A mesa)
  (sobre B mesa)
  (sobre C A)
  (brazo-libre)
  (libre B)
  (libre C))

(:goal
  (sobre A B)
  (sobre B C))
```

1. El problema de planificación

Ejemplo: mundo de bloques

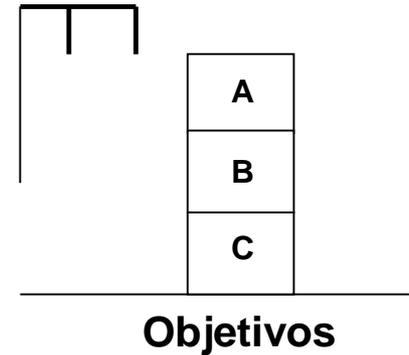
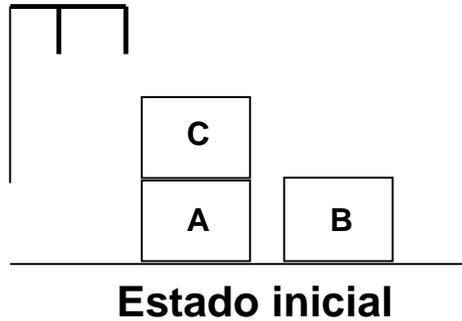


```
(:init
  (= (pos A) mesa)
  (= (pos B) mesa)
  (= (pos C) A)
  (= (estado brazo) libre)
  (= (estado B) libre)
  (= (estado C) libre))

(:goal
  (= (pos A) B)
  (= (pos B) C))
```

1. El problema de planificación

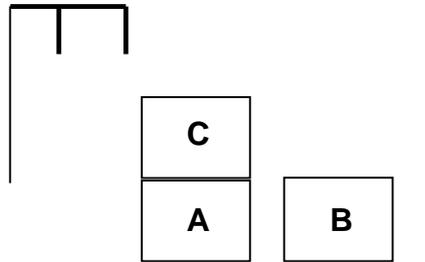
Ejemplo: mundo de bloques



- Operadores aplicables:
 - **coger** un bloque de la mesa
 - **desapilar** un bloque que está encima de otro
 - **apilar** un bloque encima de otro
 - **dejar** un bloque encima de la mesa

1. El problema de planificación

Ejemplo: mundo de bloques



Estado inicial

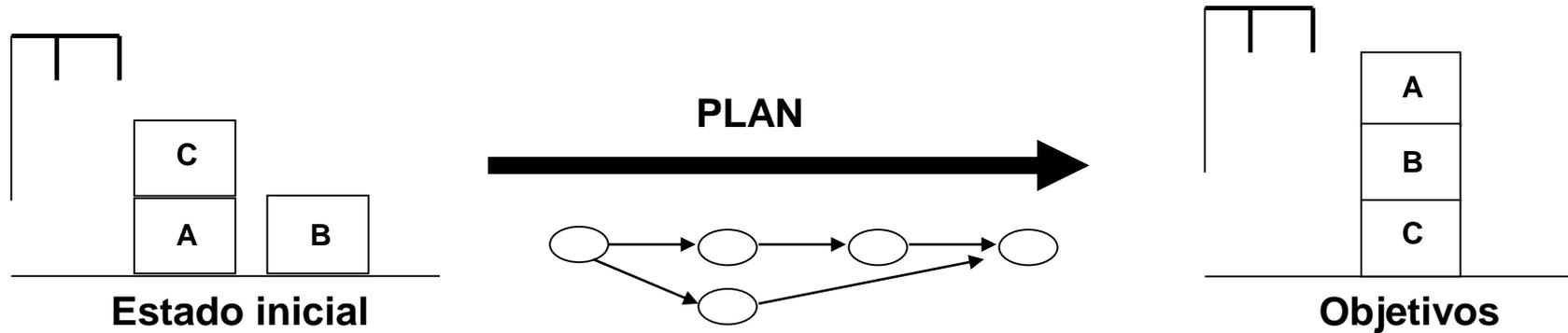
```
(:action desapilar
:parameters (?ob - bloque ?underob - bloque)
:precondition
  (and (sobre ?ob ?underob)(libre ?ob))(brazo-libre))
:effect
  (and (not (libre ?ob))(not (brazo-libre))
  (not (sobre ?ob ?underob))(sujeto ?ob)
  (libre ?underob)))
```

Objetivos

- Operadores aplicables:
 - **coger** un bloque de la mesa
 - **desapilar** un bloque que está encima de otro
 - **apilar** un bloque encima de otro
 - **dejar** un bloque encima de la mesa

1. El problema de planificación

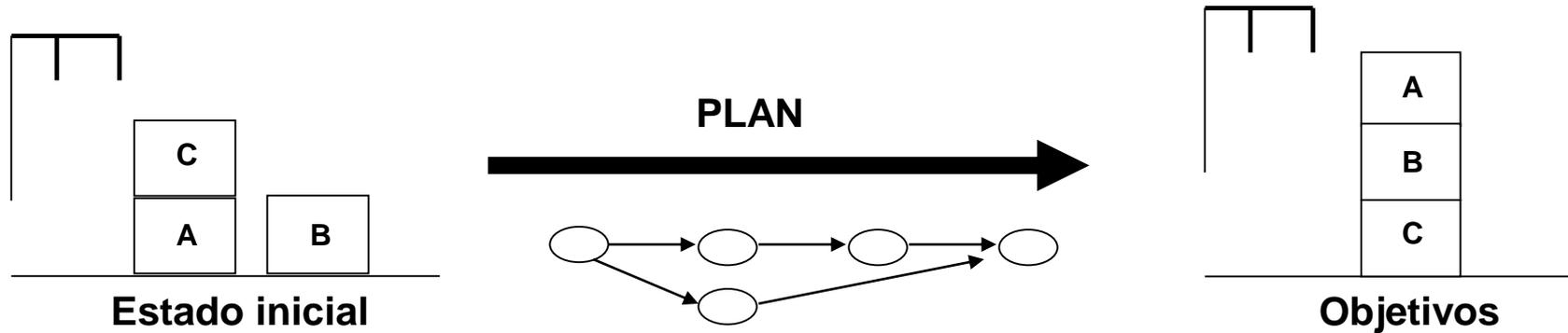
Ejemplo: mundo de bloques



- Operadores aplicables:
 - **coger** un bloque de la mesa
 - **desapilar** un bloque que está encima de otro
 - **apilar** un bloque encima de otro
 - **dejar** un bloque encima de la mesa

1. El problema de planificación

Ejemplo: mundo de bloques



- Operadores aplicables:
 - **coger** un bloque de la mesa
 - **desapilar** un bloque que está encima de otro
 - **apilar** un bloque encima de otro
 - **dejar** un bloque encima de la mesa

PLAN:

Tiempo 1: desapilar (C,A)

Tiempo 2: dejar (C)

Tiempo 3: coger (B)

Tiempo 4: apilar (B,C)

Tiempo 5: coger (A)

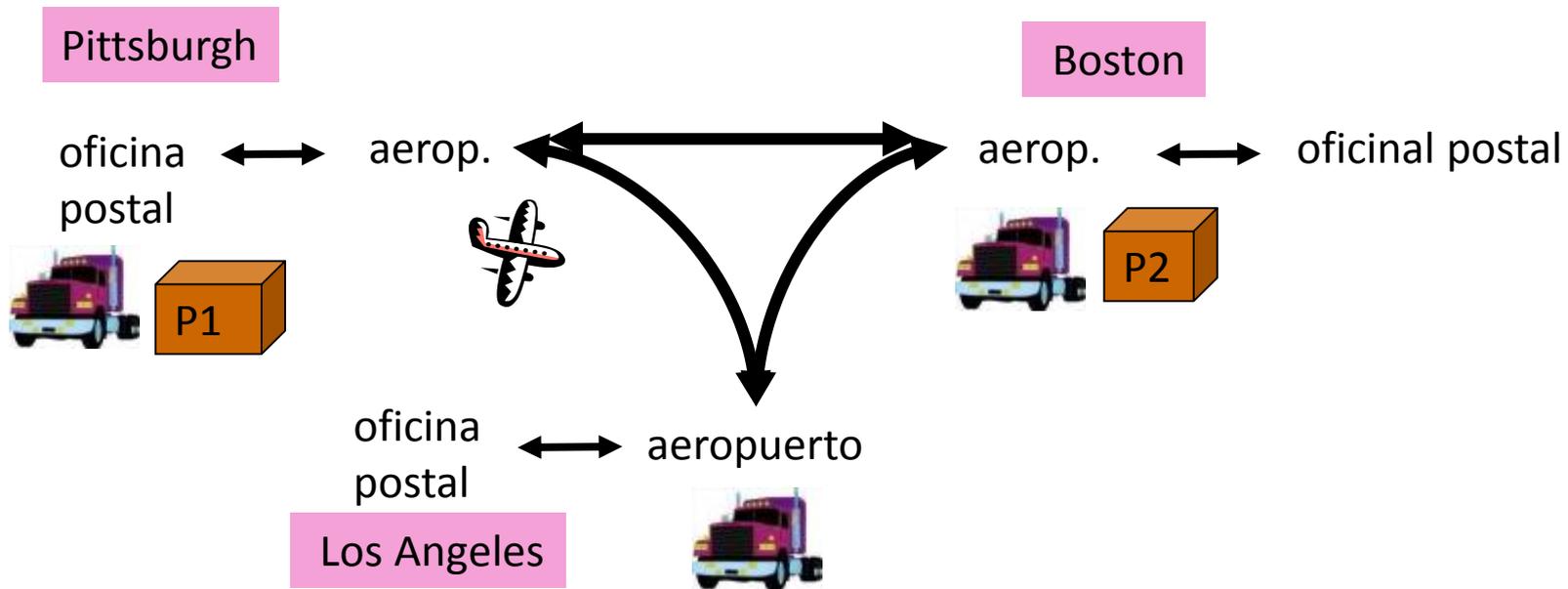
Tiempo 6: apilar (A,B)

1. El problema de planificación

Ejemplo: Logística

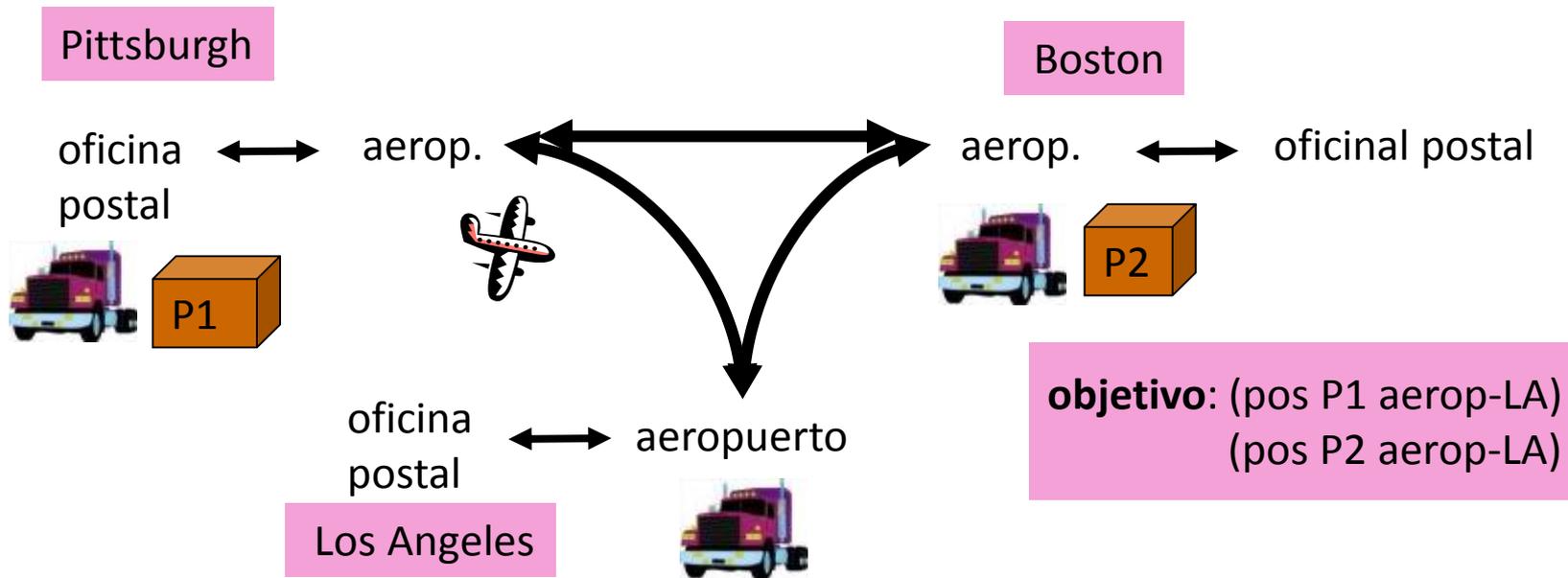
1. El problema de planificación

Ejemplo: Logística



1. El problema de planificación

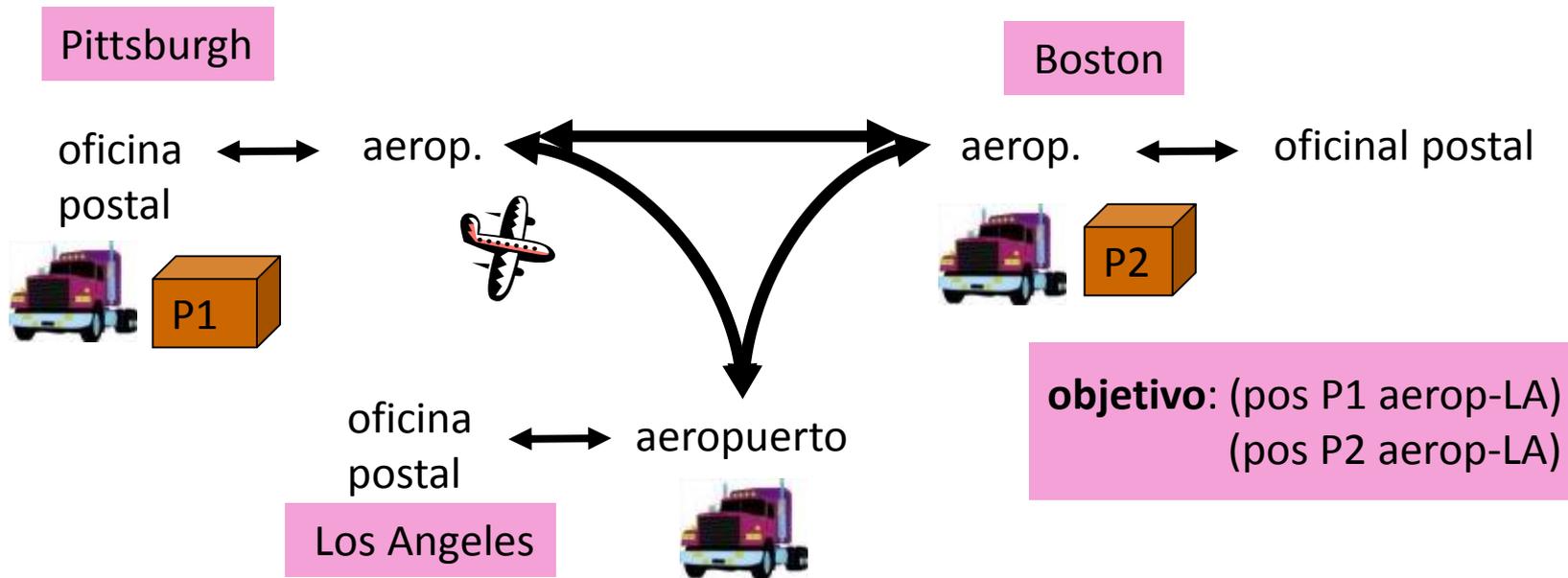
Ejemplo: Logística



1. El problema de planificación

Ejemplo: Logística

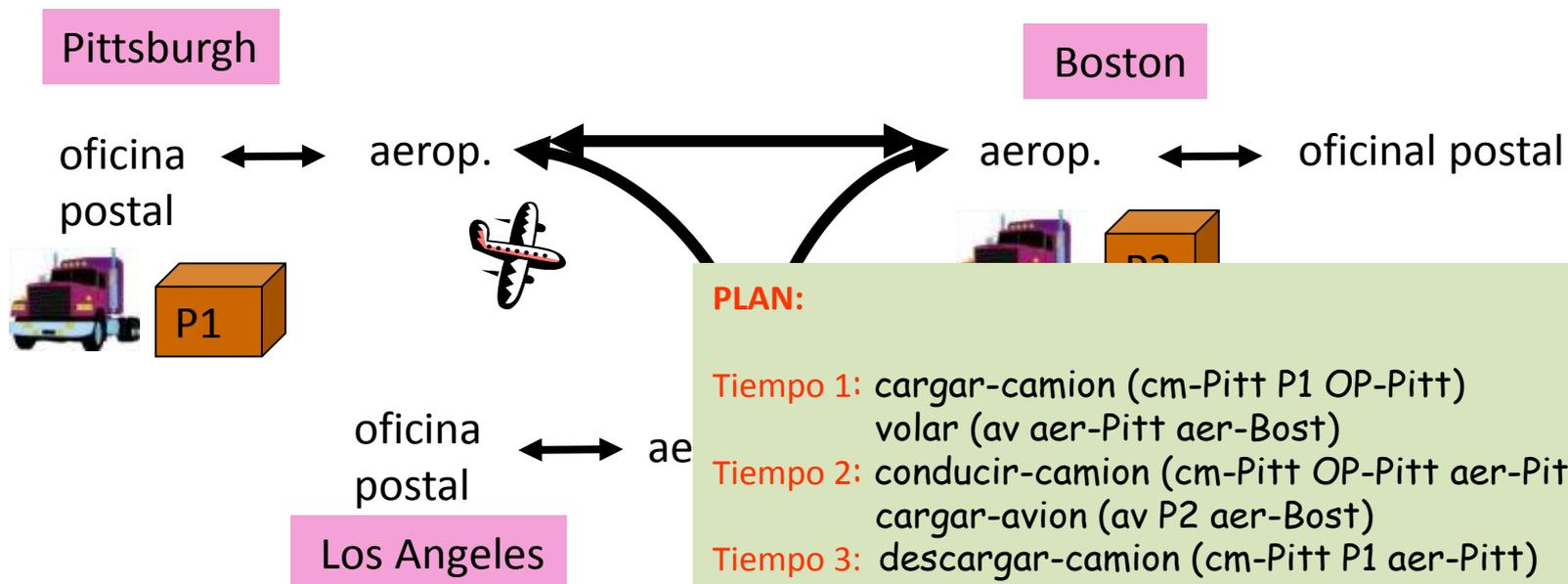
operadores: cargar-camion, descargar-camion, conducir, volar, cargar-avion, descargar-avion



1. El problema de planificación

Ejemplo: Logística

operadores: cargar-camion, descargar-camion, conducir, volar, cargar-avion, descargar-avion



PLAN:

- Tiempo 1: cargar-camion (cm-Pitt P1 OP-Pitt)
volar (av aer-Pitt aer-Bost)
- Tiempo 2: conducir-camion (cm-Pitt OP-Pitt aer-Pitt)
cargar-avion (av P2 aer-Bost)
- Tiempo 3: descargar-camion (cm-Pitt P1 aer-Pitt)
volar (av aer-Bost aer-LA)
- Tiempo 4: descargar-avion (av P2 aer-LA)
- Tiempo 5: volar (av aer-LA aer-Pitt)
- Tiempo 6: cargar-avion (av P1 aer-Pitt)
- Tiempo 7: volar (av aer-Pitt aer-LA)
- Tiempo 8: descargar-avion (av P1 aer-LA)

1. El problema de Scheduling

Es un problema de **asignación de recursos** para realizar una serie de tareas en un plazo de tiempo sujetas a restricciones de diferente tipo:

1. capacidad operacional de las máquinas
2. disponibilidad limitada de recursos
3. fechas de entrega
4. duraciones de las tareas
5. prioridades entre tareas

y **optimizar uno o más objetivos**

1. El problema de Scheduling

Es un problema de **asignación de recursos** para realizar una serie de tareas en un plazo de tiempo sujetas a restricciones de diferente tipo:

1. capacidad operacional de las máquinas
2. disponibilidad limitada de recursos
3. fechas de entrega
4. duraciones de las tareas
5. prioridades entre tareas

y **optimizar uno o más objetivos**

Problemas de fabricación:

procesos de producción (job shop, flow shop, open shop)

Problemas de servicios:

asignación de horarios

asignación de personal a puestos

diseño de itinerarios

1. El problema de Scheduling

Ejemplo: asignación recursos a tareas

4 tareas (T1, T2, T3, T4)

cada tarea tiene un tiempo de procesamiento (duración)

2 recursos (R1, R2)

cada tarea consume una cantidad de cada recurso durante el tiempo que está en proceso
capacidad máxima de R1 = 5 capacidad máxima de R2 = 7

Relación de precedencia: T2 → T3

Objetivo: minimizar duración (*makespan*)

1. El problema de Scheduling

Ejemplo: asignación recursos a tareas

4 tareas (T1, T2, T3, T4)

cada tarea tiene un tiempo de procesamiento (duración)

2 recursos (R1, R2)

cada tarea consume una cantidad de cada recurso durante el tiempo que está en proceso
capacidad máxima de R1 = 5 capacidad máxima de R2= 7

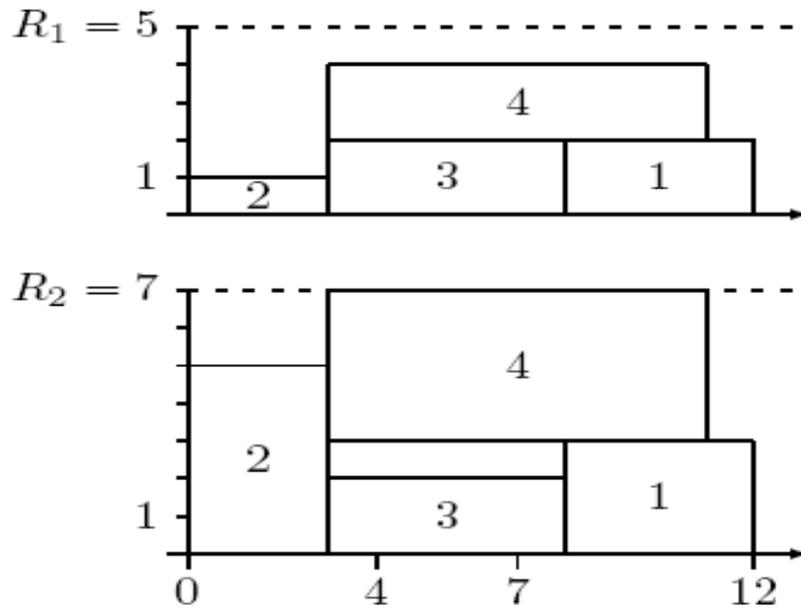
Relación de precedencia: T2 → T3

Objetivo: minimizar duración (*makespan*)

	T1	T2	T3	T4
Duración	4	3	5	8
R1	2	1	2	2
R2	3	5	2	4

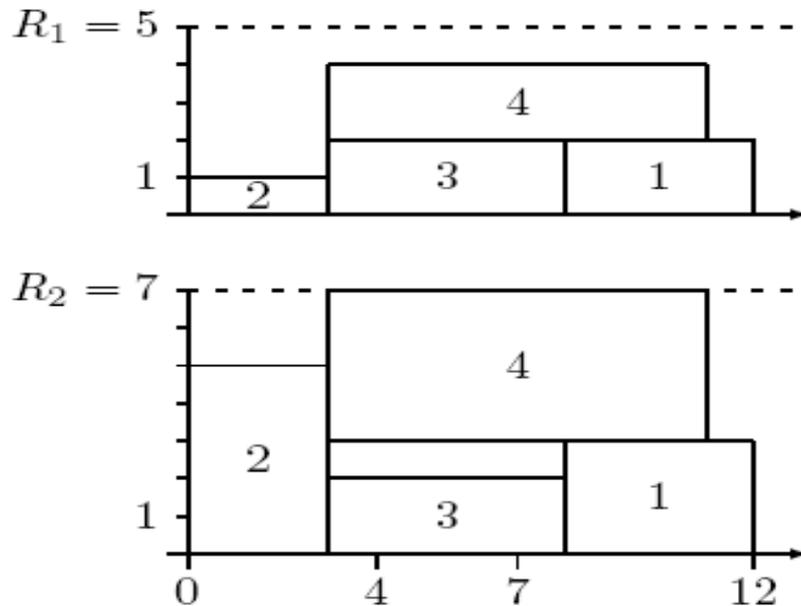
1. El problema de Scheduling

Schedule o solución que minimiza la duración o makespan



1. El problema de Scheduling

Schedule o solución que minimiza la duración o makespan



Diferencias respecto a un problema de planificación:

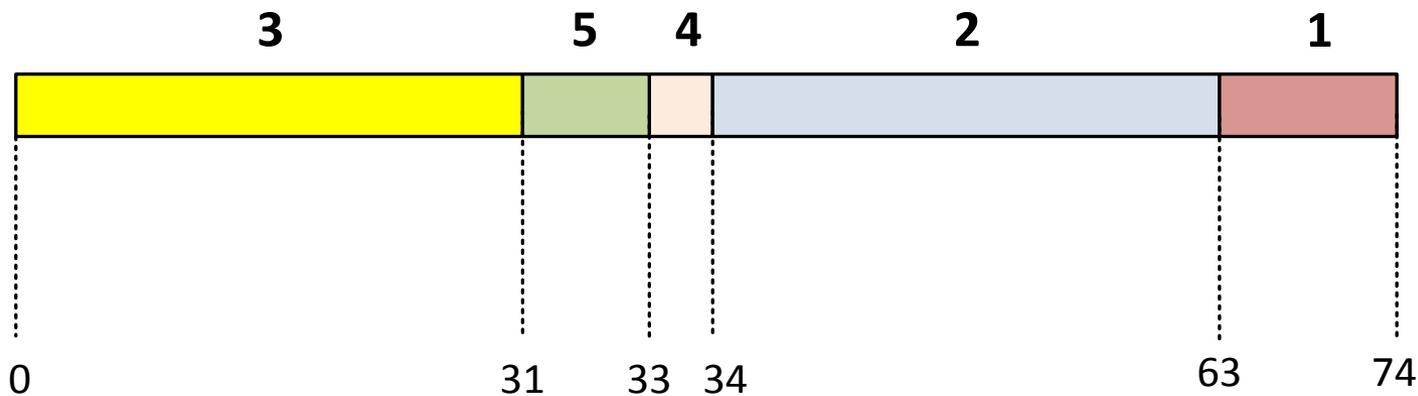
1. Las tareas son conocidas de antemano
2. Las duraciones de las tareas son conocidas
3. Se conoce las precedencias entre tareas

1. El problema de Scheduling

Ejemplo: tareas pendientes de procesar en una máquina

Tarea	Tiempo procesamiento (duración)	Fecha entrega (límite tiempo)
1	11	61
2	29	45
3	31	31
4	1	33
5	2	32

Solución (schedule) que minimiza el retraso total



2. Planificación temporal

Es un problema de planificación donde:

1. Las acciones tienen una **duración**
2. El objetivo es obtener el plan de **menor *makespan*** que resuelve el problema

2. Planificación temporal

Es un problema de planificación donde:

1. Las acciones tienen una **duración**
2. El objetivo es obtener el plan de **menor *makespan*** que resuelve el problema

Adicionalmente:

1. Restricciones de entrega o ***deadlines***
2. Otras restricciones temporales

2. Planificación temporal

El planificador debe realizar un proceso de búsqueda que seleccione las acciones adecuadas en función de los efectos de las mismas y su duración.

2. Planificación temporal

El planificador debe realizar un proceso de búsqueda que seleccione las acciones adecuadas en función de los **efectos** de las mismas y su **duración**.



quiere ir de

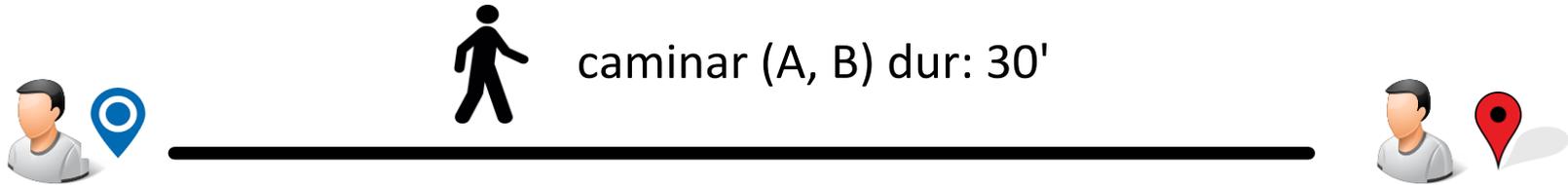


a



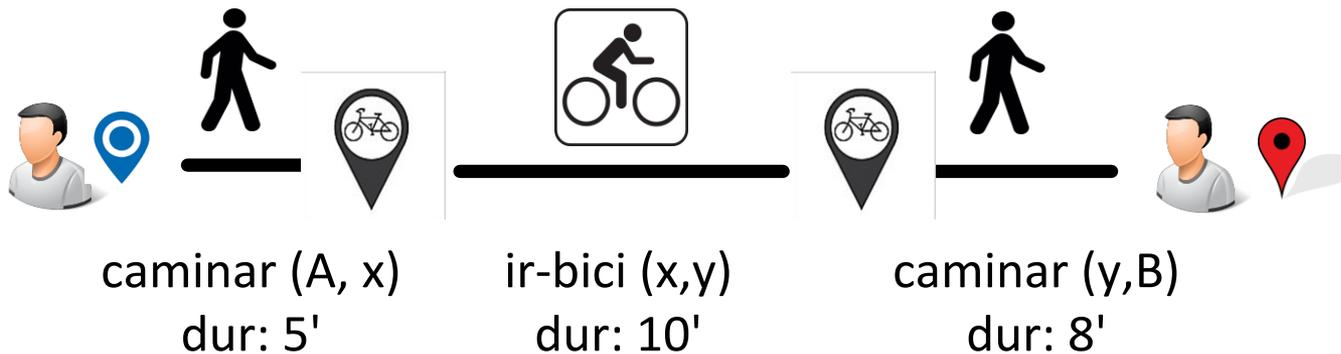
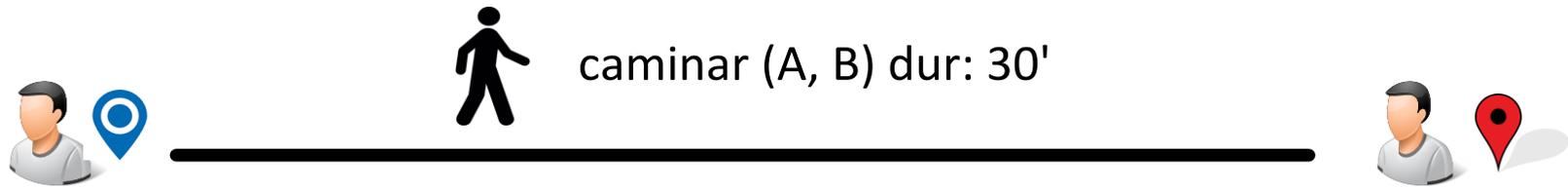
2. Planificación temporal

El planificador debe realizar un proceso de búsqueda que seleccione las acciones adecuadas en función de los **efectos** de las mismas y su **duración**.



2. Planificación temporal

El planificador debe realizar un proceso de búsqueda que seleccione las acciones adecuadas en función de los **efectos** de las mismas y su **duración**.

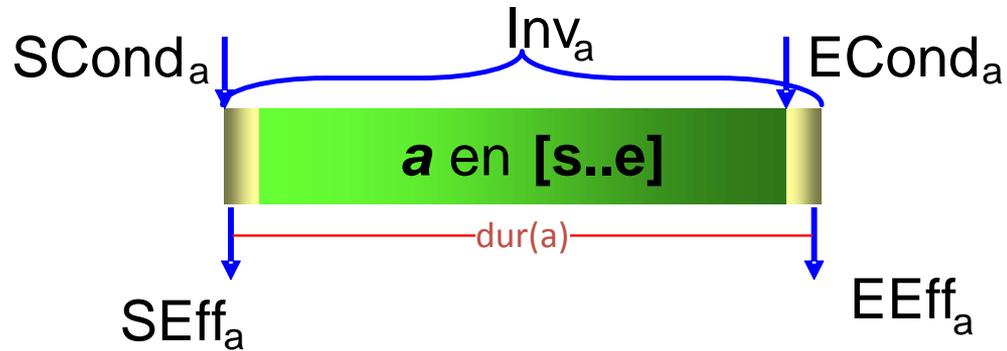


2. Planificación temporal

Modelo temporal de acciones: Planning Domain Description Language (PDDL2.1)

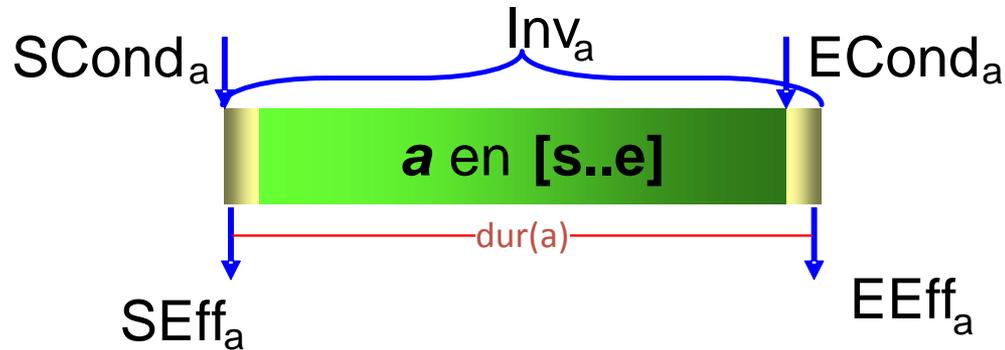
2. Planificación temporal

Modelo temporal de acciones: Planning Domain Description Language (PDDL2.1)



2. Planificación temporal

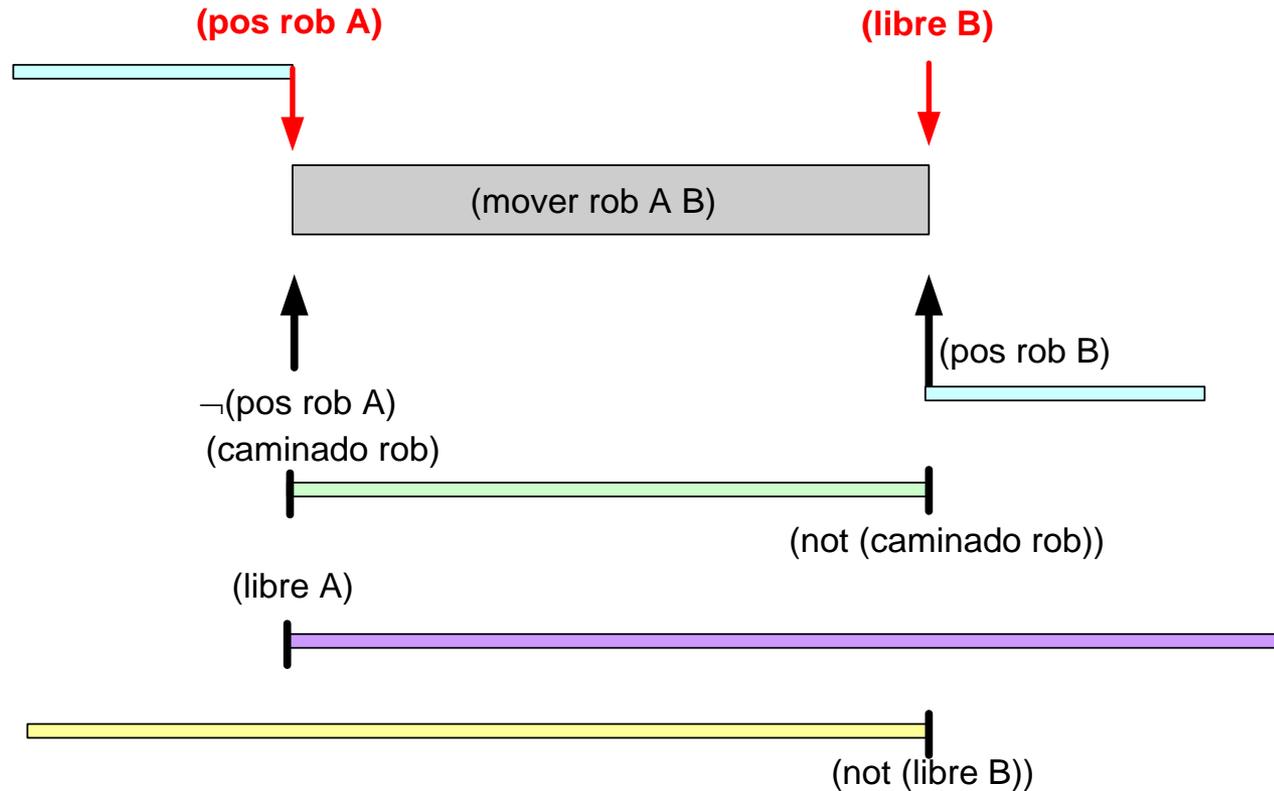
Modelo temporal de acciones: Planning Domain Description Language (PDDL2.1)



```
(:durative-action volar
:parameters (?a - avion ?c1 ?c2 - ciudad)
:duration (= ?duration (/ (distancia ?c1 ?c2) (velocidad ?a)))
:condition (and (at start (pos ?a ?c1))(overall (conexion-aerea ?c1 ?c2)))
:effect (and (at start (not (pos ?a ?c1)) (volando ?a))
             (at end (not (volando ?a)) (pos ?a ?c2))))
```

2. Planificación temporal

- Las condiciones y efectos se anotan junto con sus tiempos de ocurrencia

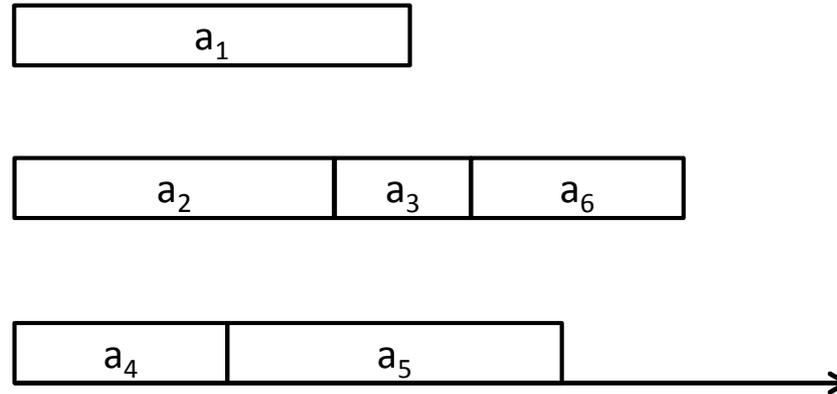


2. Planificación temporal

Plan temporal: múltiples alternativas de conurrencia

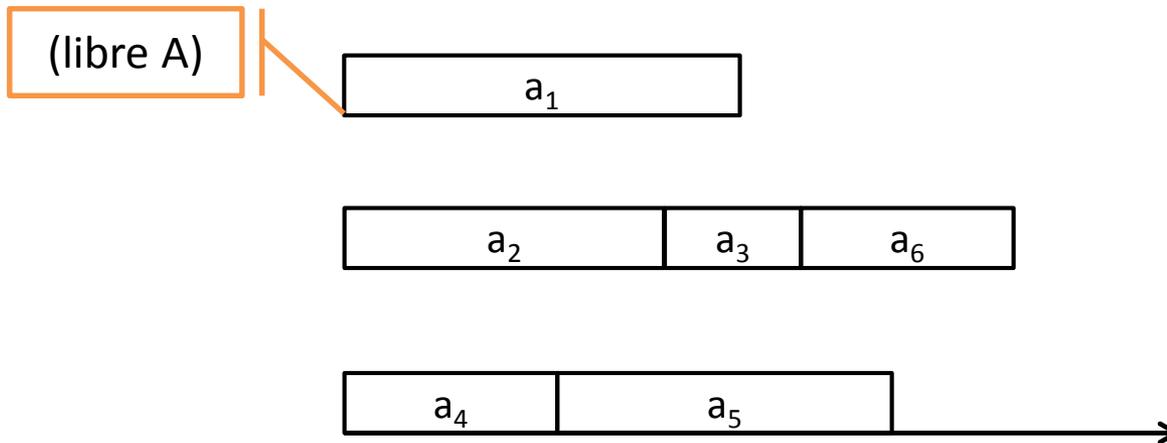
2. Planificación temporal

Plan temporal: múltiples alternativas de conurrencia



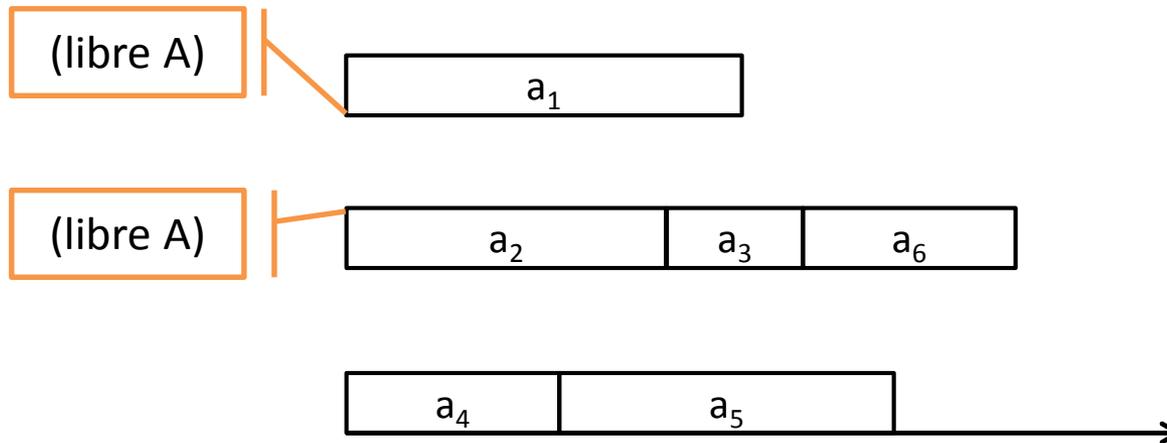
2. Planificación temporal

Plan temporal: múltiples alternativas de conurrencia



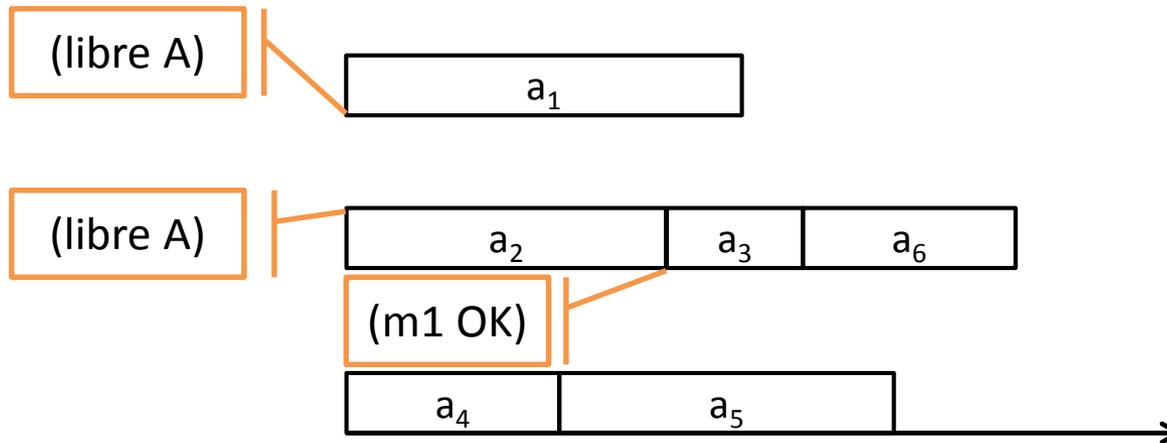
2. Planificación temporal

Plan temporal: múltiples alternativas de conurrencia



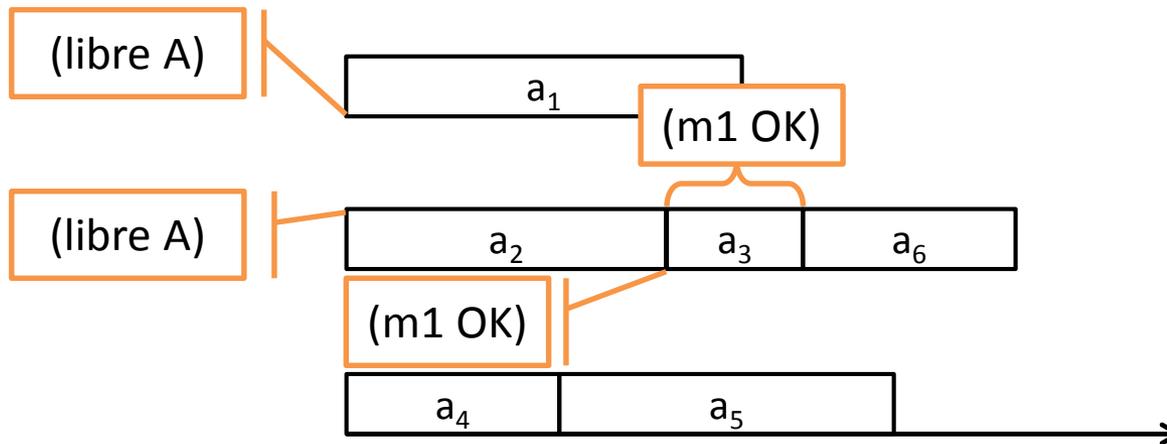
2. Planificación temporal

Plan temporal: múltiples alternativas de conurrencia



2. Planificación temporal

Plan temporal: múltiples alternativas de conurrencia

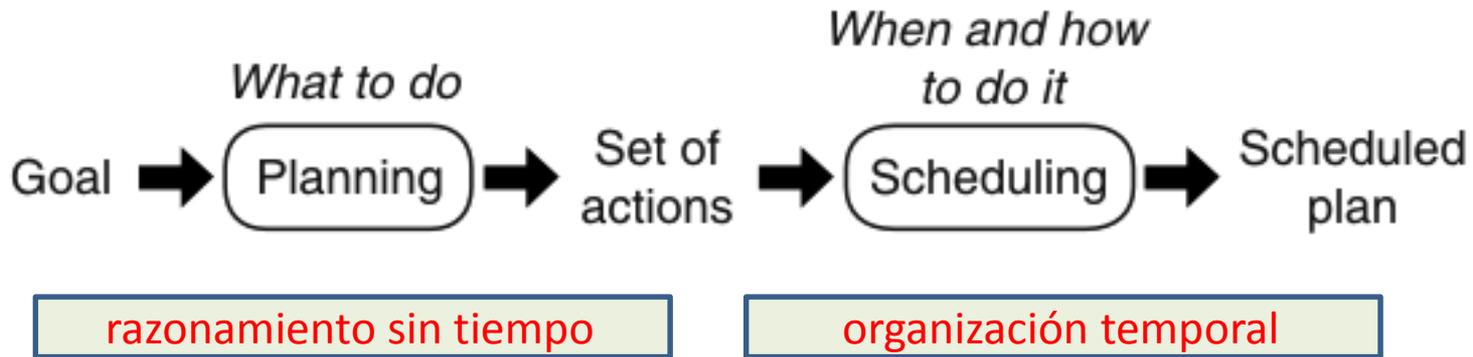


2. Planificación temporal

Modelo de planificación temporal

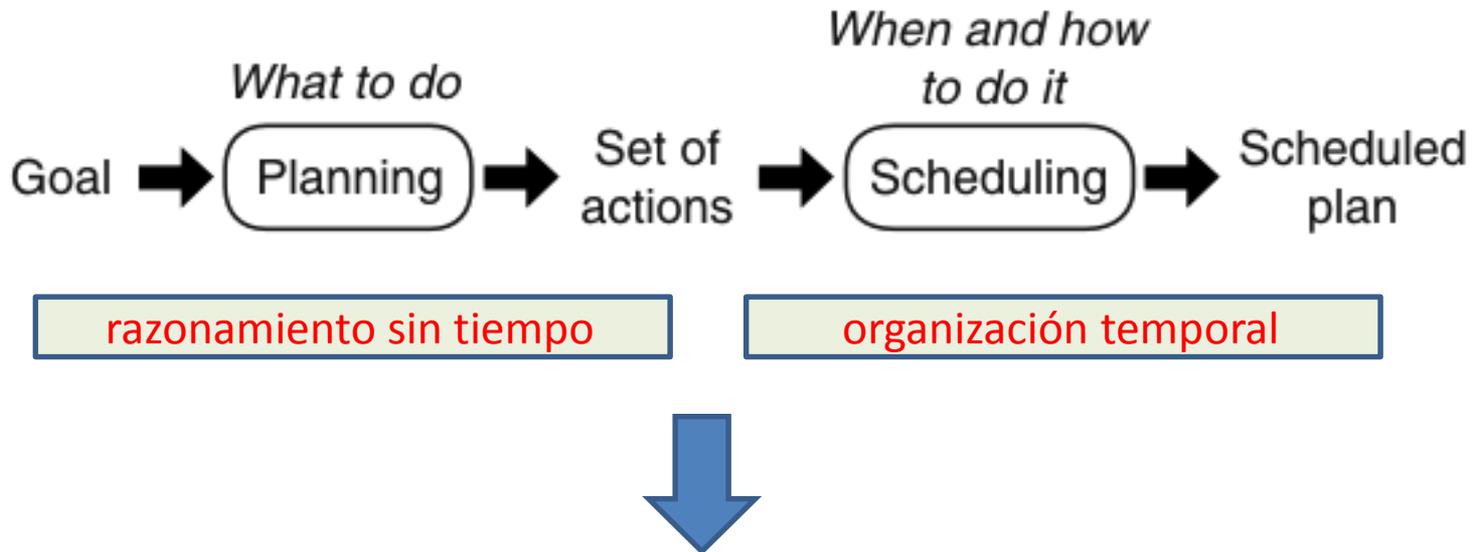
2. Planificación temporal

Modelo de planificación temporal



2. Planificación temporal

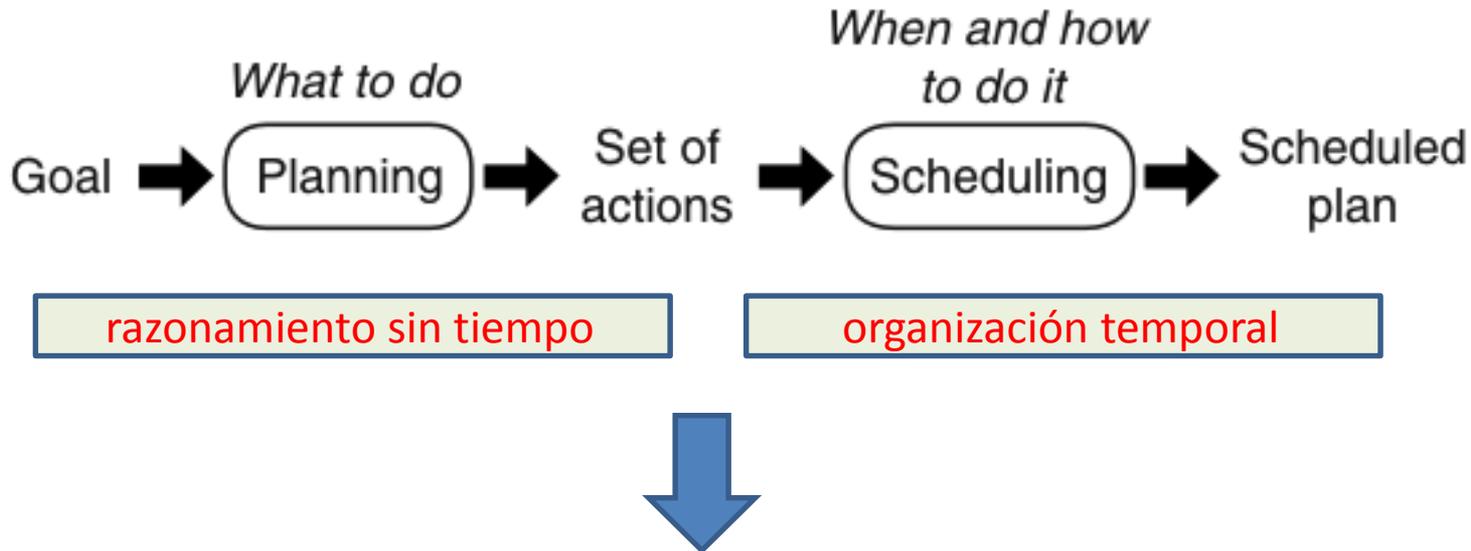
Modelo de planificación temporal



La función de optimización del planificador no conlleva tiempo ni duración de las acciones → Generalmente, minimización del número de acciones

2. Planificación temporal

Modelo de planificación temporal

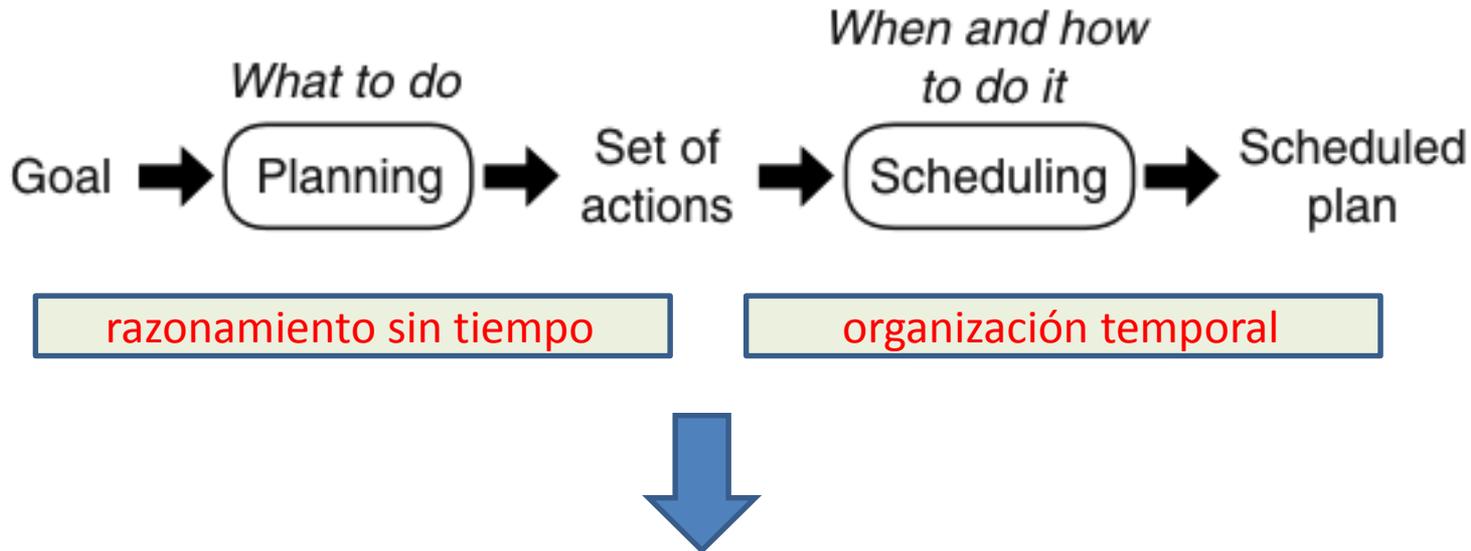


La función de optimización del planificador no conlleva tiempo ni duración de las acciones → Generalmente, minimización del número de acciones

Un menor número de acciones no implica un menor *makespan* o duración del plan

2. Planificación temporal

Modelo de planificación temporal



La función de optimización del planificador no conlleva tiempo ni duración de las acciones → Generalmente, minimización del número de acciones

Un menor número de acciones no implica un menor *makespan* o duración del plan

Aplicar razonamiento temporal en el proceso de planificación

2. Planificación temporal

Representación interna de un plan temporal

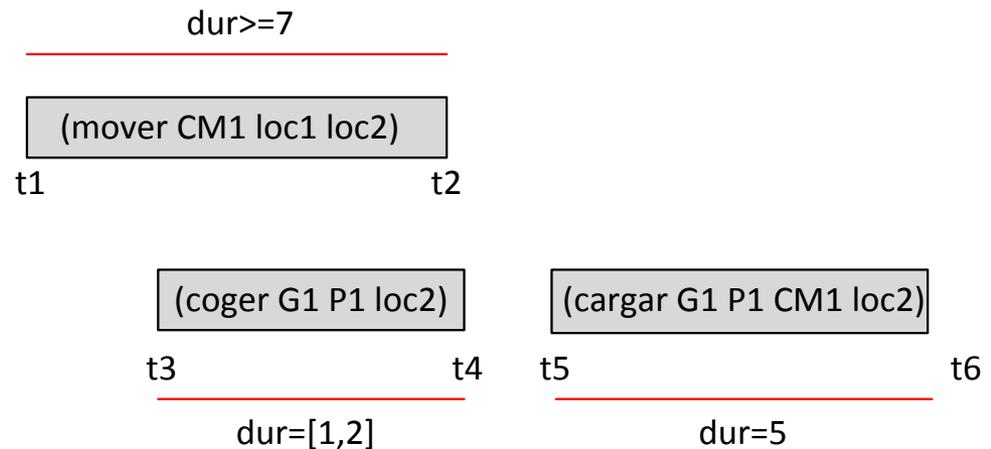
- Representación tiempo: intervalos (Algebra de Allen), puntos de tiempo
- Representación acciones temporales
- Representación proposiciones temporales (*temporally qualified expressions*)
- Orden parcial, enlaces causales (efecto-condición)

2. Planificación temporal

Representación interna de un plan temporal

- Representación tiempo: intervalos (Algebra de Allen), puntos de tiempo
- Representación acciones temporales
- Representación proposiciones temporales (*temporally qualified expressions*)
- Orden parcial, enlaces causales (efecto-condición)

Plan

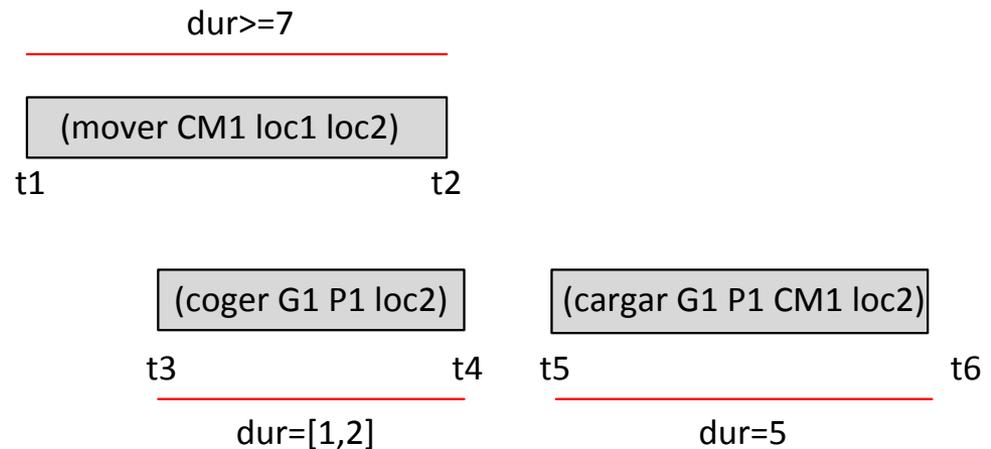


2. Planificación temporal

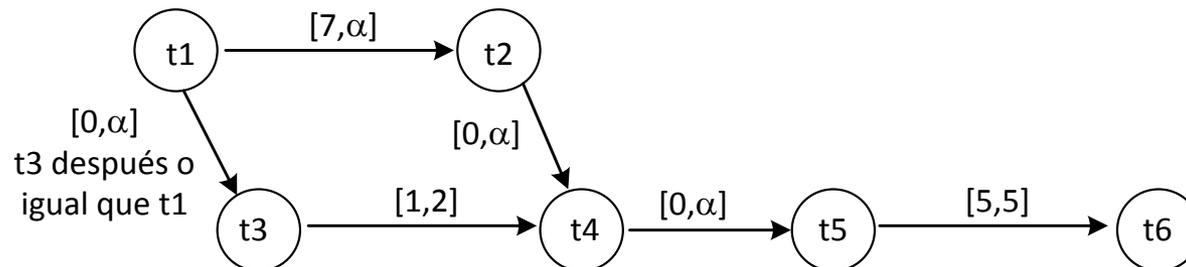
Representación interna de un plan temporal

- Representación tiempo: intervalos (Algebra de Allen), puntos de tiempo
- Representación acciones temporales
- Representación proposiciones temporales (*temporally qualified expressions*)
- Orden parcial, enlaces causales (efecto-condición)

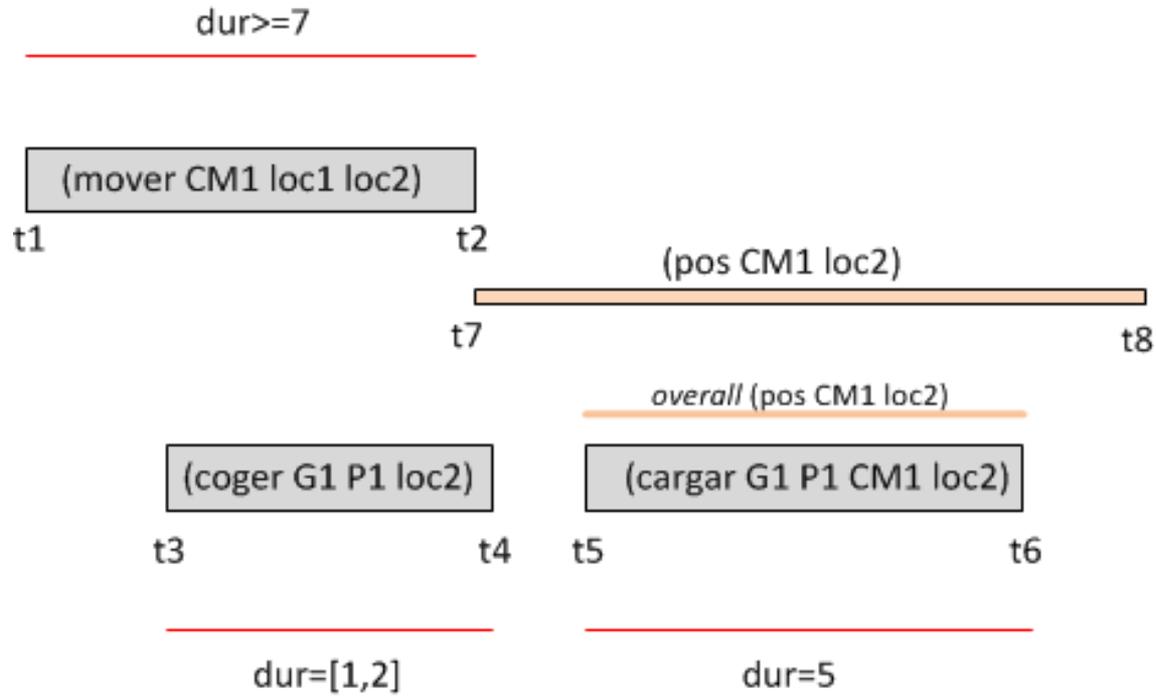
Plan



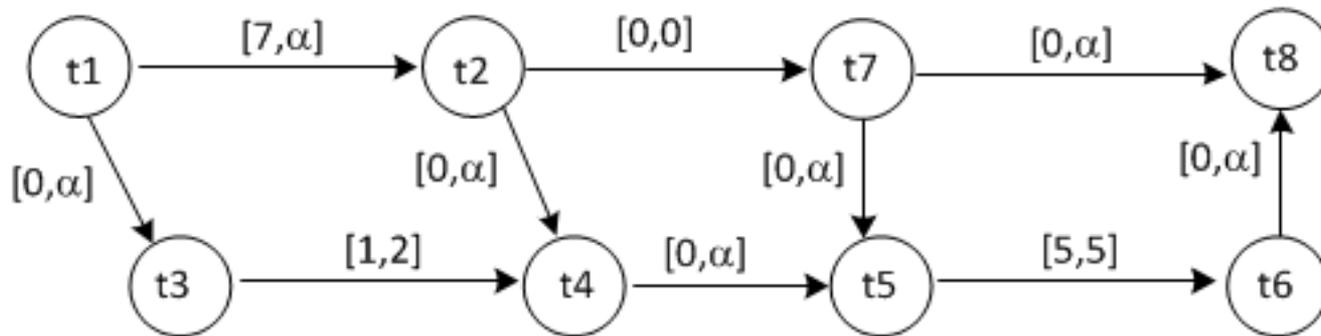
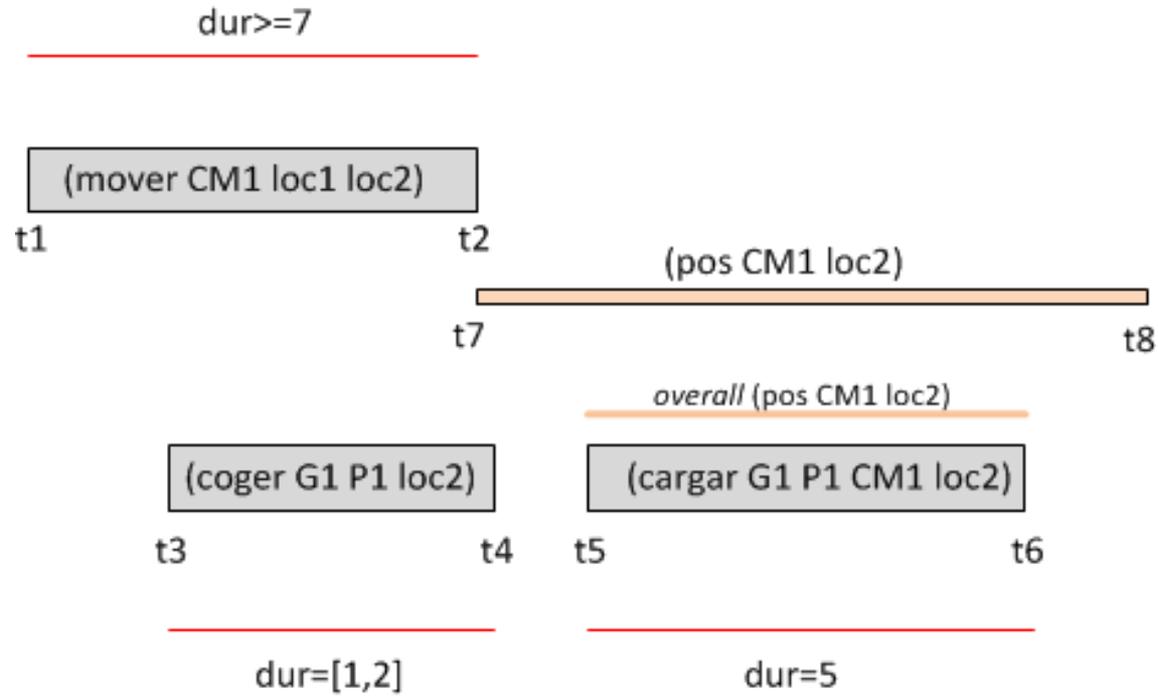
Red temporal simple



2. Planificación temporal



2. Planificación temporal

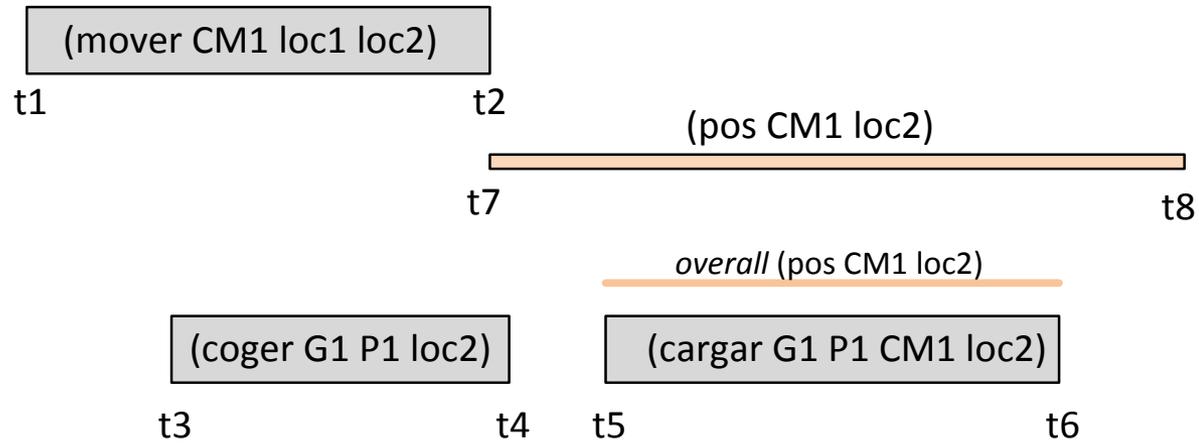


2. Planificación temporal

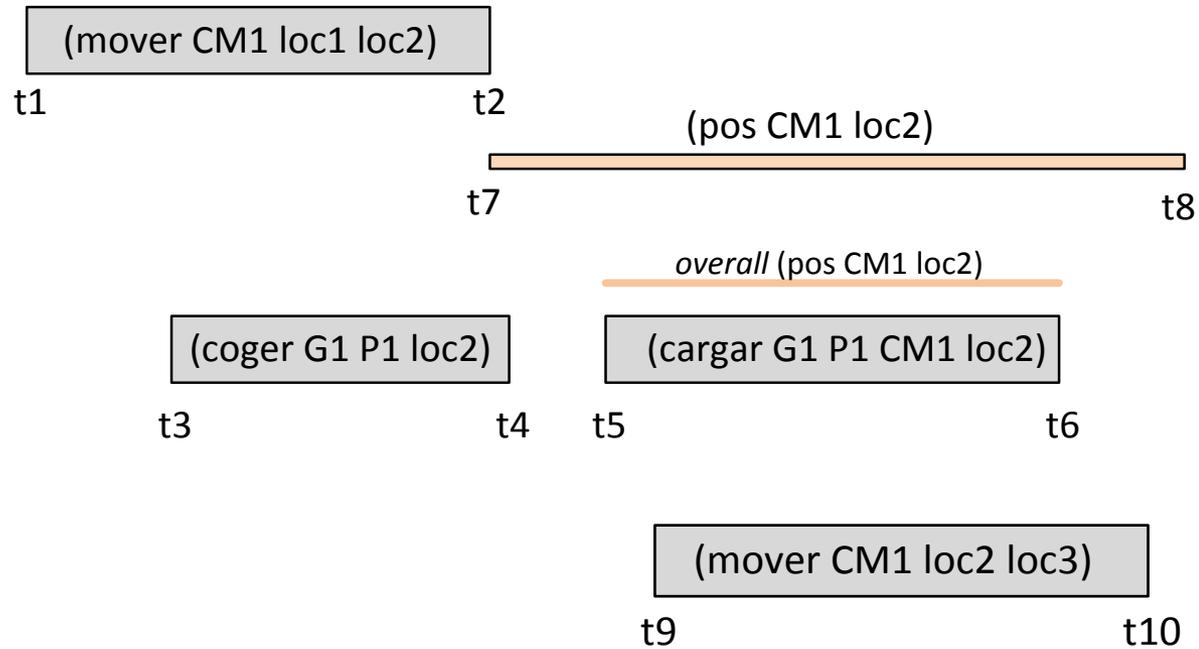
Razonamiento temporal

1. Las redes temporales definen un problema de satisfacción de restricciones (CSP – Constraint Satisfaction Problem)
2. Objetivo: asignar un intervalo temporal o punto de tiempo a cada instante de comienzo y finalización de una acción o proposición temporal de modo que se satisfagan todas las restricciones de la red
3. Existen algoritmos para comprobar la consistencia de una red temporal: arco-consistencia, camino-consistencia, etc.

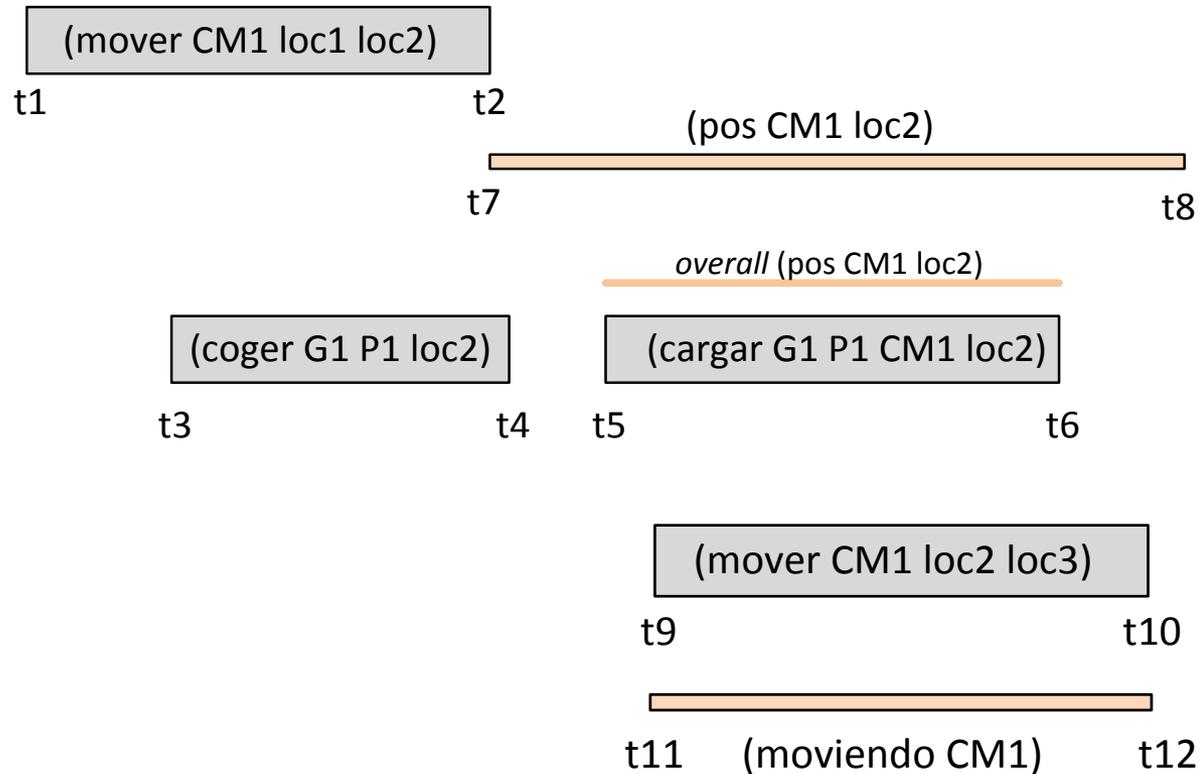
2. Planificación temporal



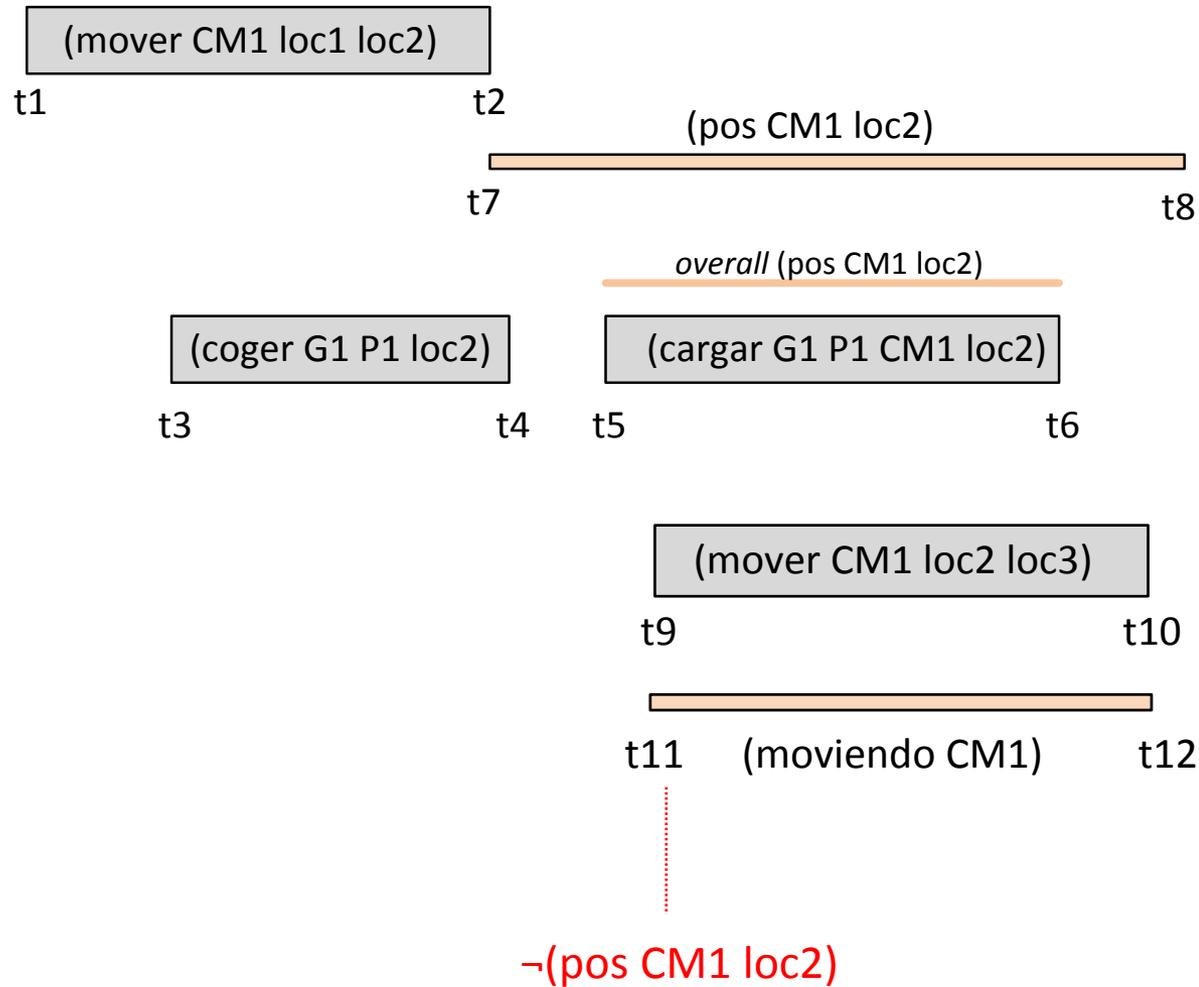
2. Planificación temporal



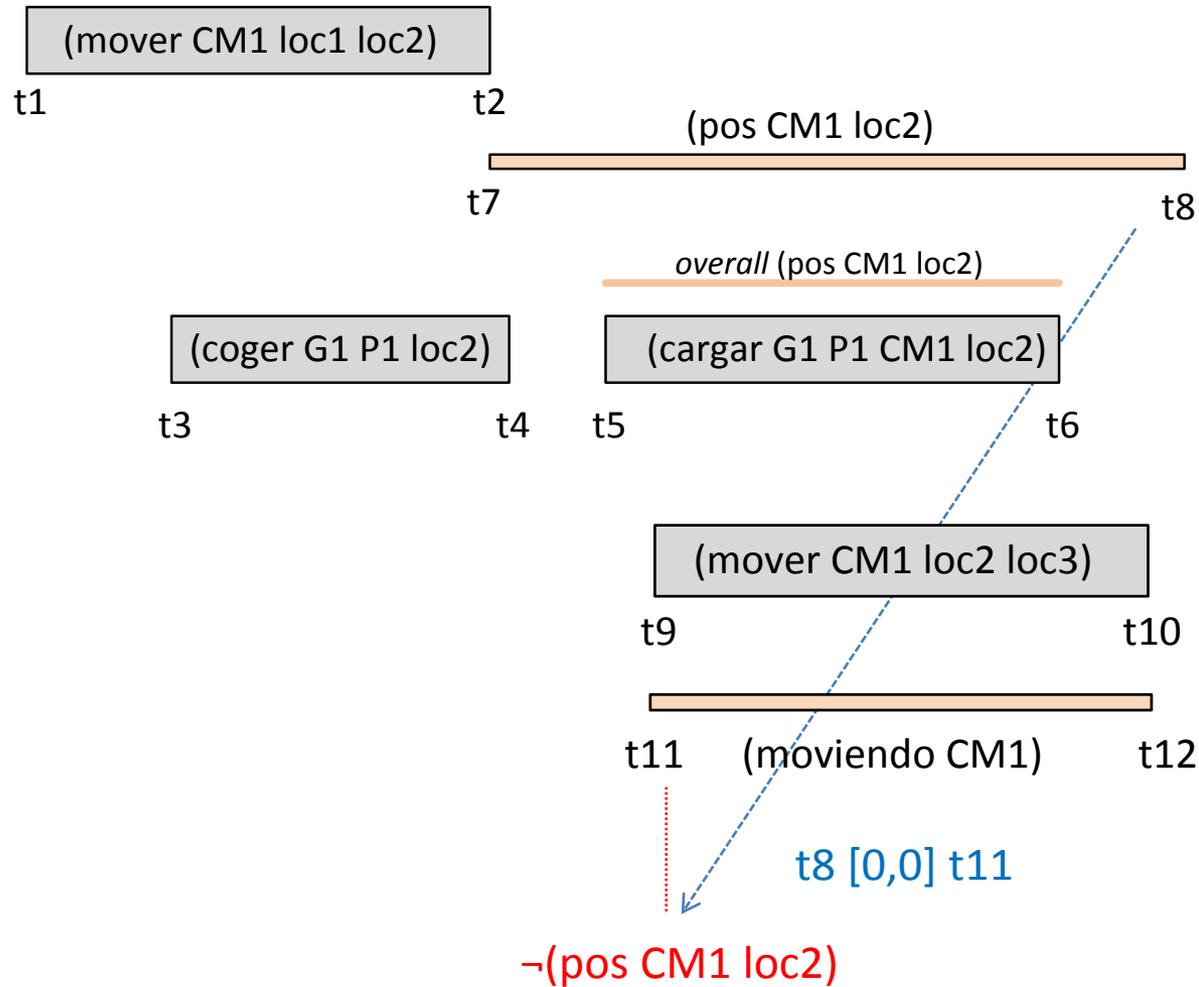
2. Planificación temporal



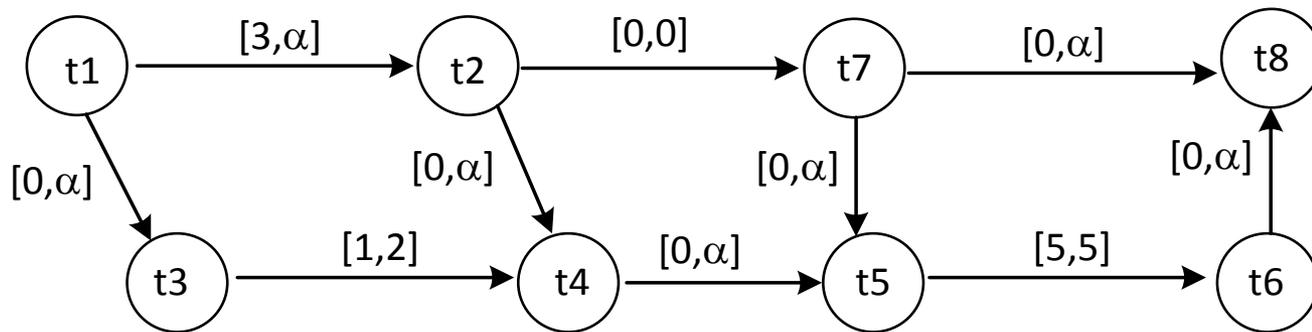
2. Planificación temporal



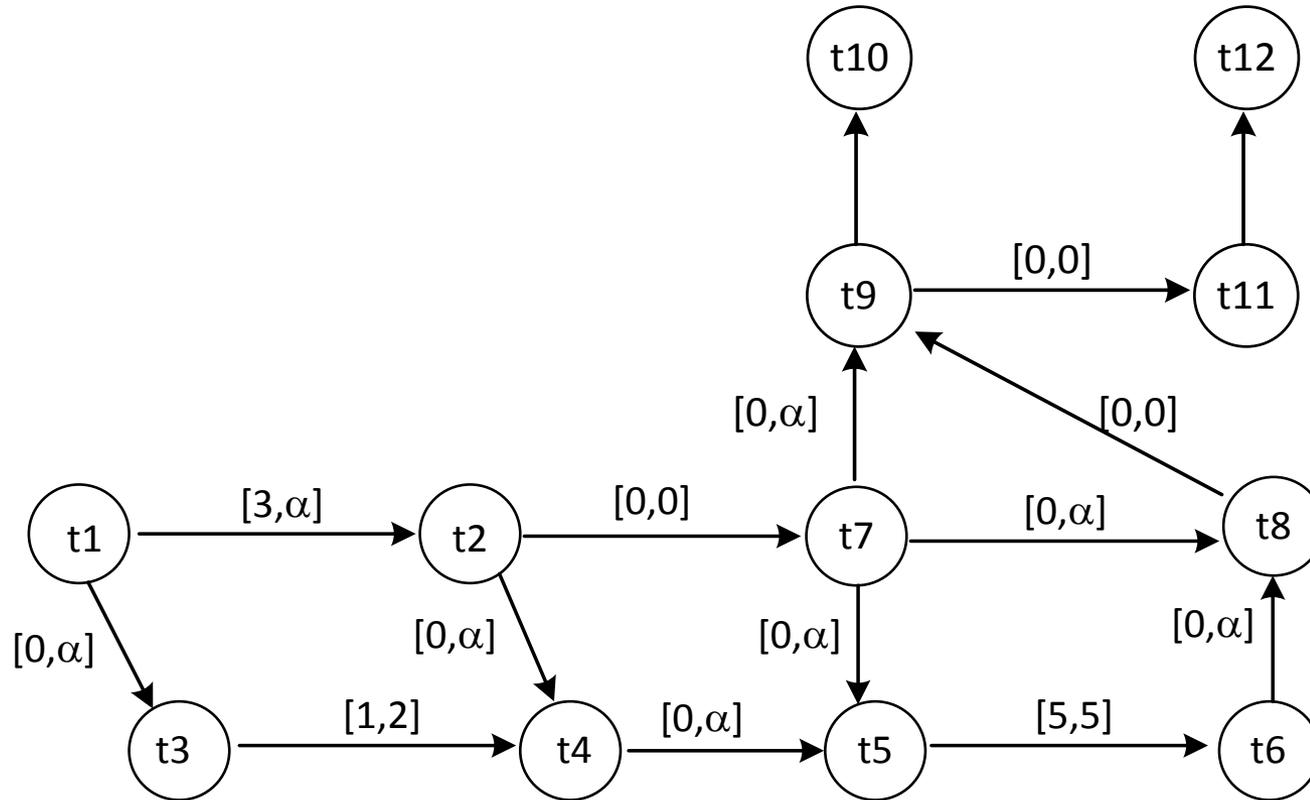
2. Planificación temporal



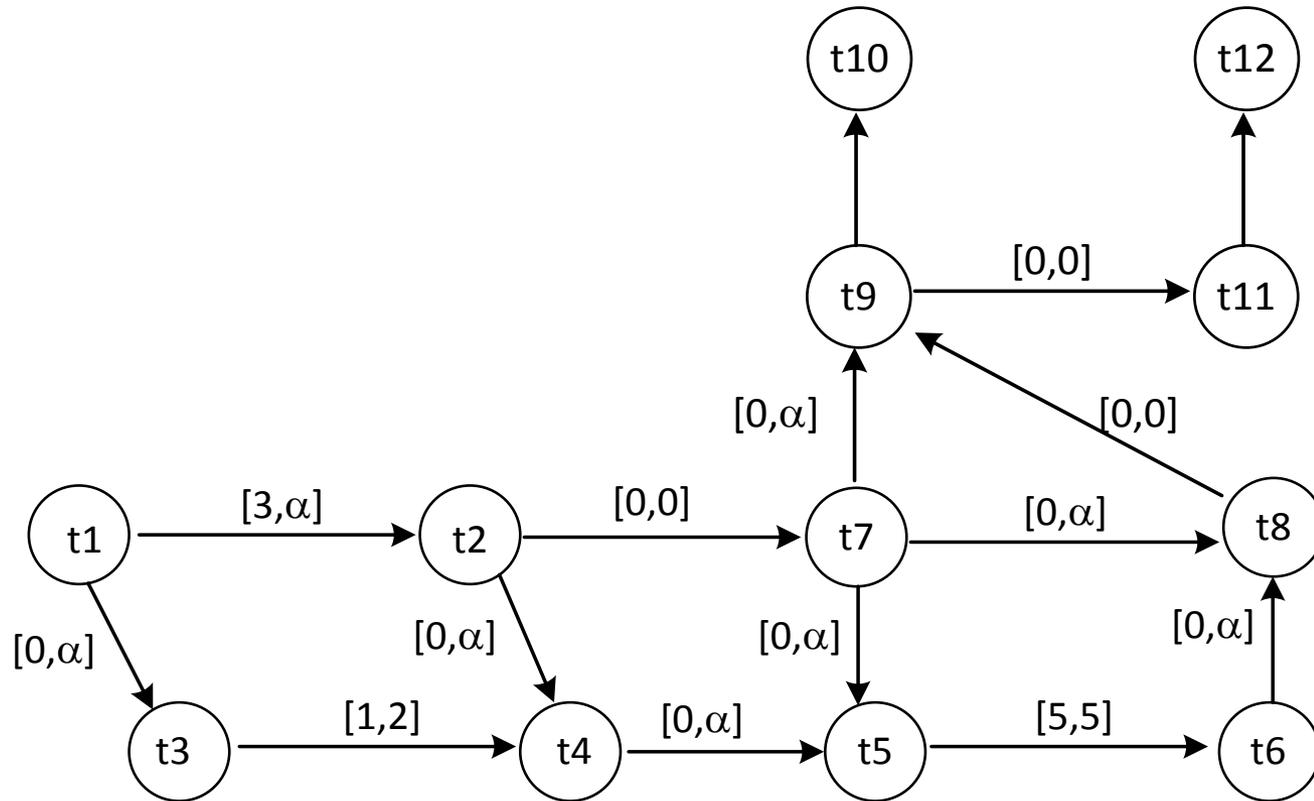
2. Planificación temporal



2. Planificación temporal

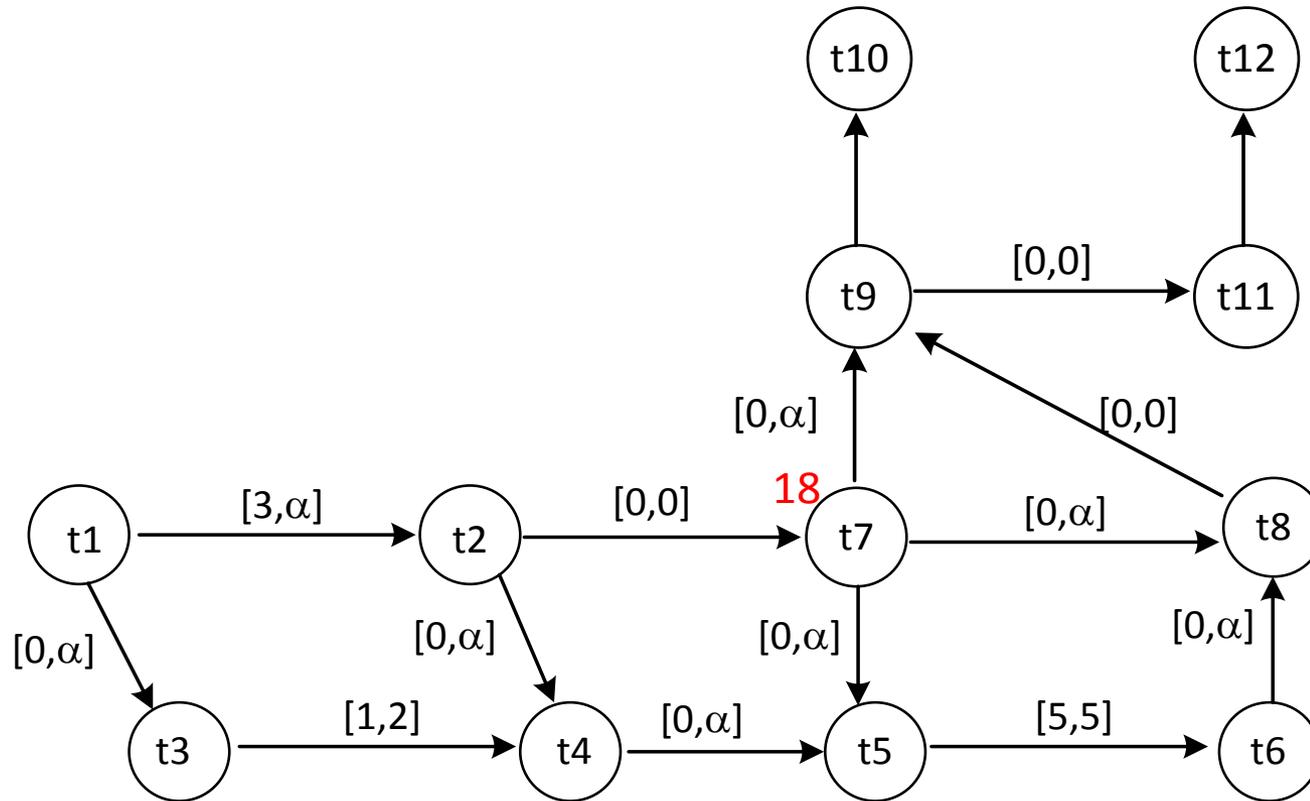


2. Planificación temporal



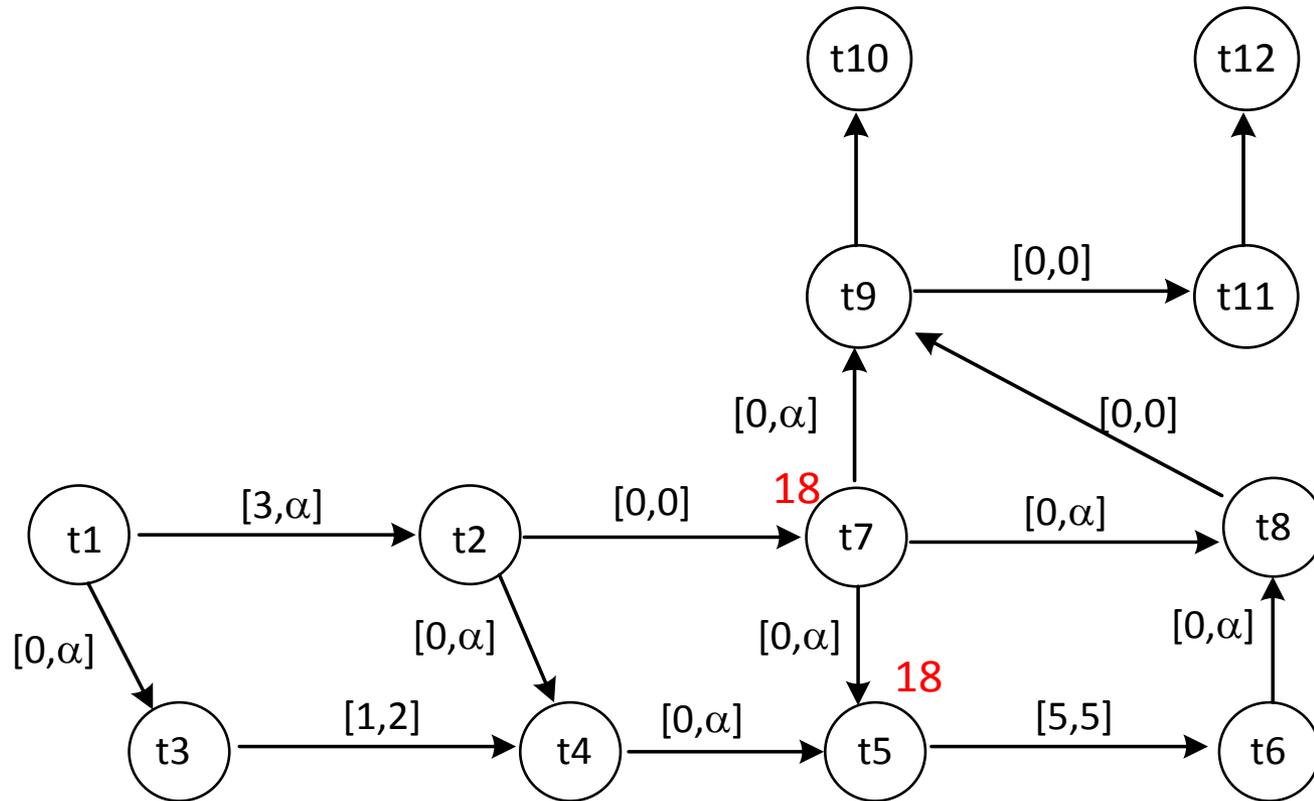
El planificador asigna el tiempo 18 al momento en el que el camión llega a loc2 **t7=18**

2. Planificación temporal



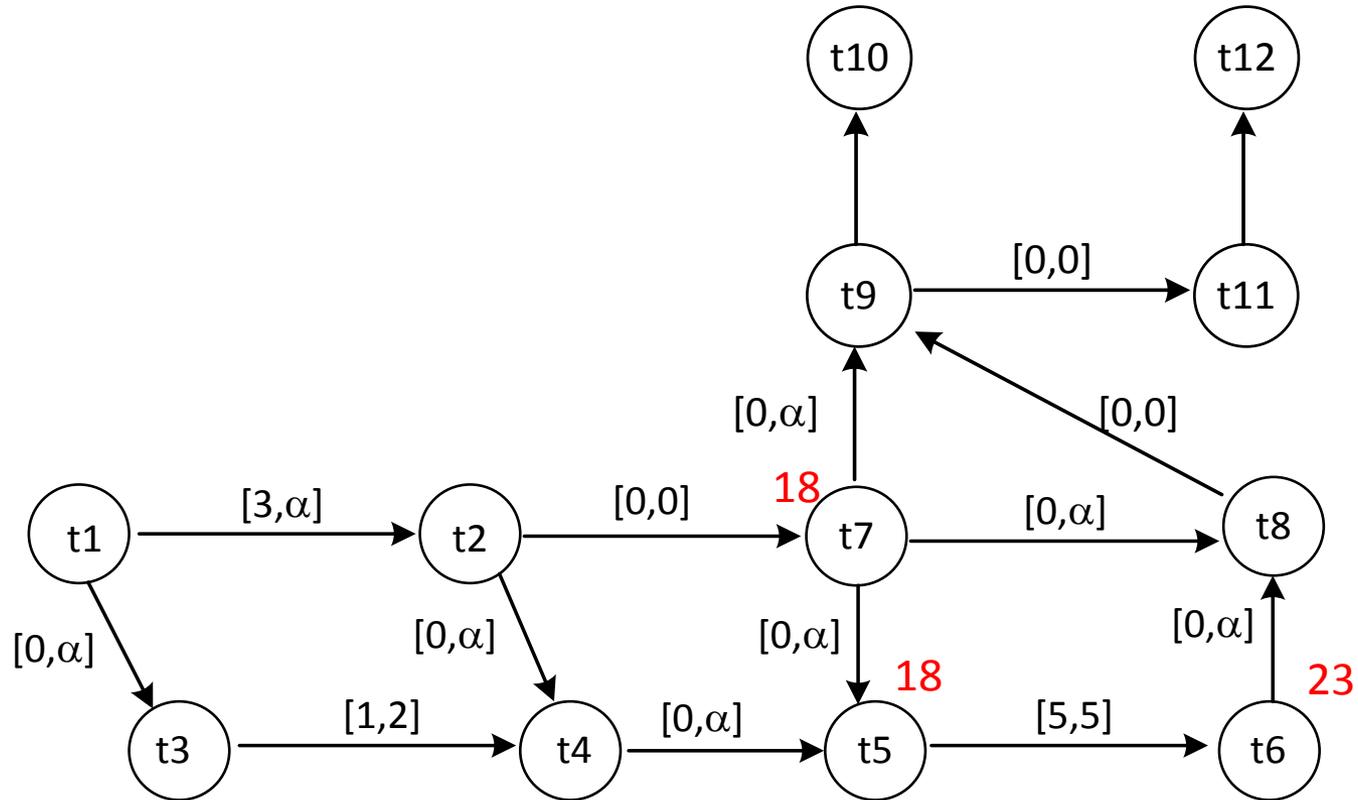
El planificador asigna el tiempo 18 al momento en el que el camión llega a loc2 $t7=18$

2. Planificación temporal



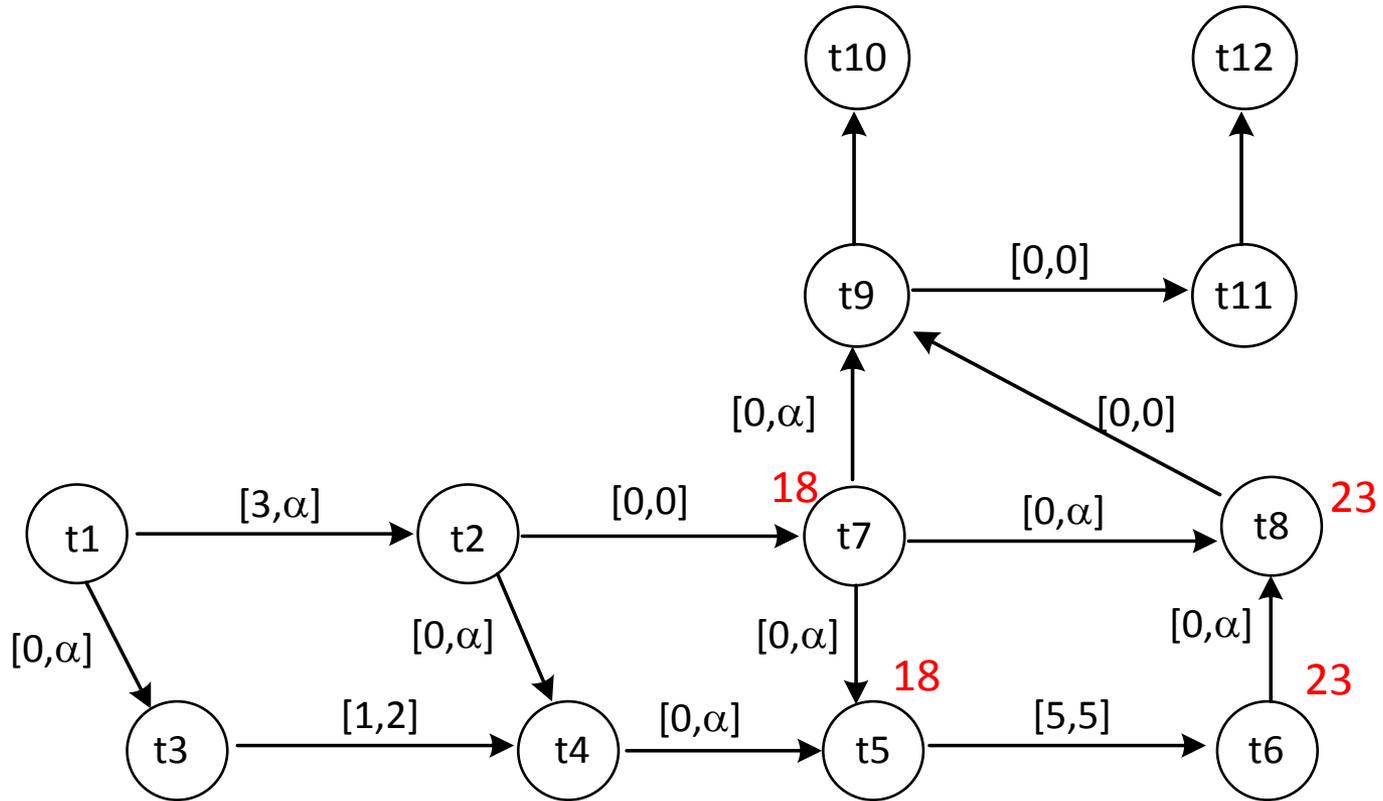
El planificador asigna el tiempo 18 al momento en el que el camión llega a loc2 **t7=18**

2. Planificación temporal



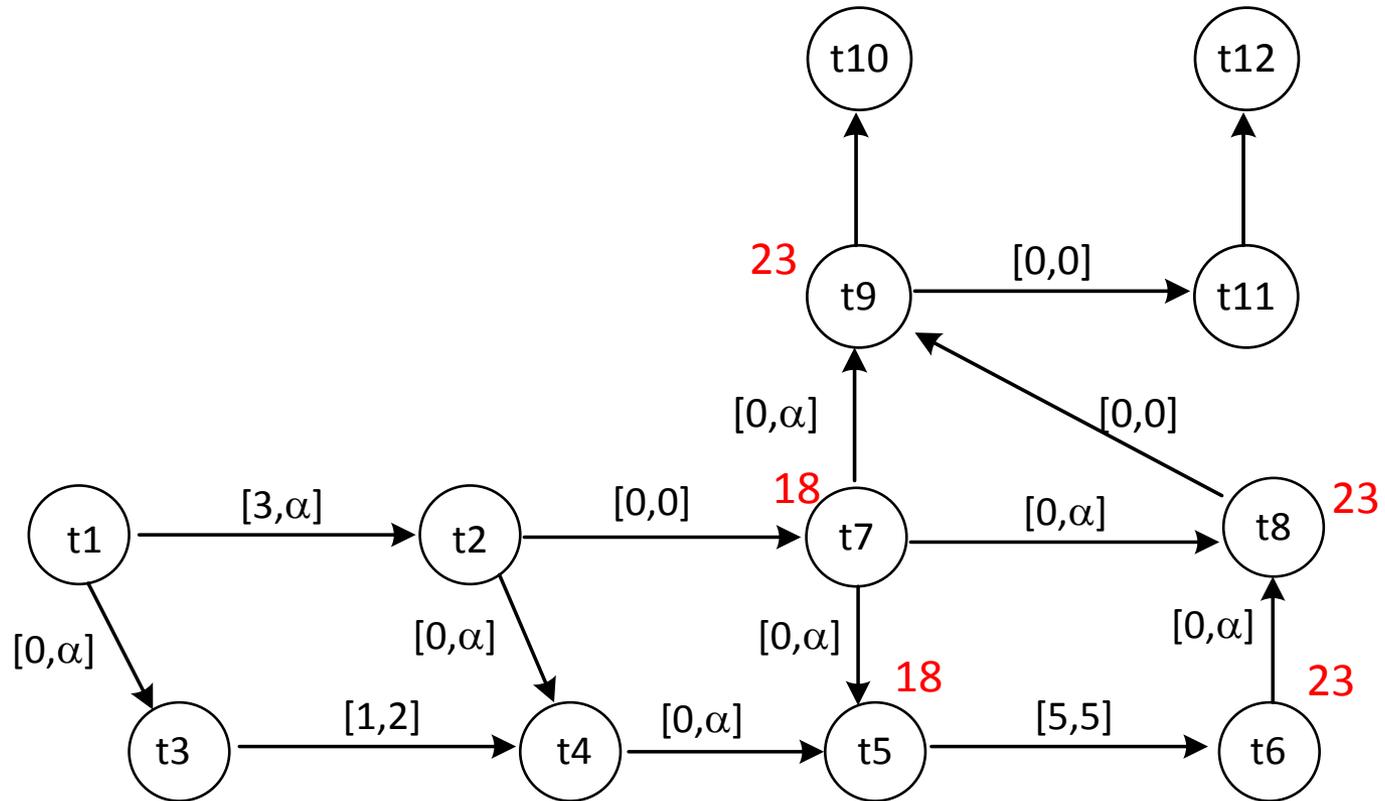
El planificador asigna el tiempo 18 al momento en el que el camión llega a loc2 $t7=18$

2. Planificación temporal



El planificador asigna el tiempo 18 al momento en el que el camión llega a loc2 $t7=18$

2. Planificación temporal



El planificador asigna el tiempo 18 al momento en el que el camión llega a loc2 $t7=18$

2. Planificación temporal

Construcción de un plan temporal (inserción de una acción en un plan)

2. Planificación temporal

Construcción de un plan temporal (inserción de una acción en un plan)

Aplicabilidad de una acción α en un plan Π :

2. Planificación temporal

Construcción de un plan temporal (inserción de una acción en un plan)

Aplicabilidad de una acción α en un plan Π :

1. Comprobar condiciones de α se satisfacen inicialmente en el estado de Π

2. Planificación temporal

Construcción de un plan temporal (inserción de una acción en un plan)

Aplicabilidad de una acción α en un plan Π :

1. Comprobar condiciones de α se satisfacen inicialmente en el estado de Π
2. Comprobar consistencia de α con el resto de acciones en Π

2. Planificación temporal

Construcción de un plan temporal (inserción de una acción en un plan)

Aplicabilidad de una acción a en un plan Π :

1. Comprobar condiciones de a se satisfacen inicialmente en el estado de Π
2. Comprobar consistencia de a con el resto de acciones en Π

Dado un plan temporal consistente Π tal que $\text{dur}(\Pi)=t$, S_t es el estado que se alcanza tras la *ejecución* de Π en el estado inicial del problema

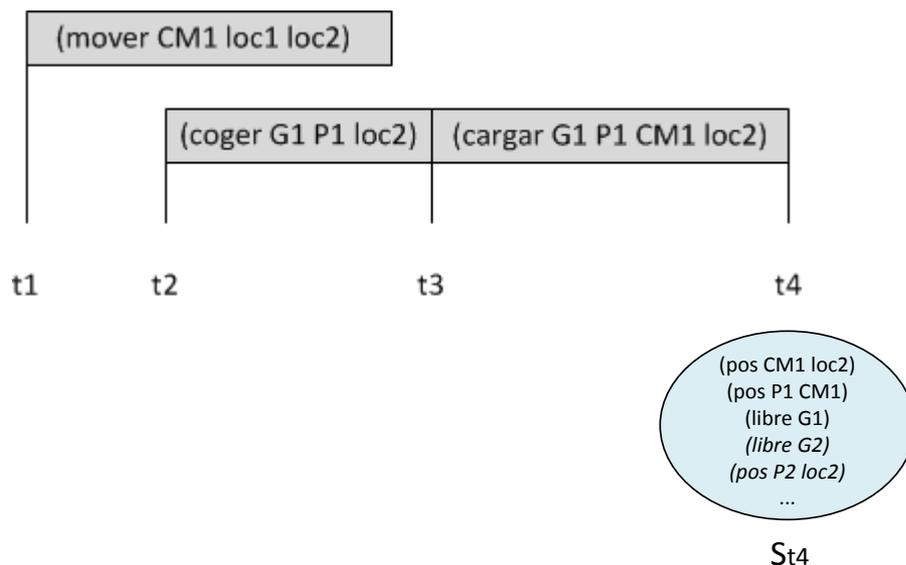
2. Planificación temporal

Construcción de un plan temporal (inserción de una acción en un plan)

Aplicabilidad de una acción a en un plan Π :

1. Comprobar condiciones de a se satisfacen inicialmente en el estado de Π
2. Comprobar consistencia de a con el resto de acciones en Π

Dado un plan temporal consistente Π tal que $\text{dur}(\Pi)=t$, S_t es el estado que se alcanza tras la *ejecución* de Π en el estado inicial del problema



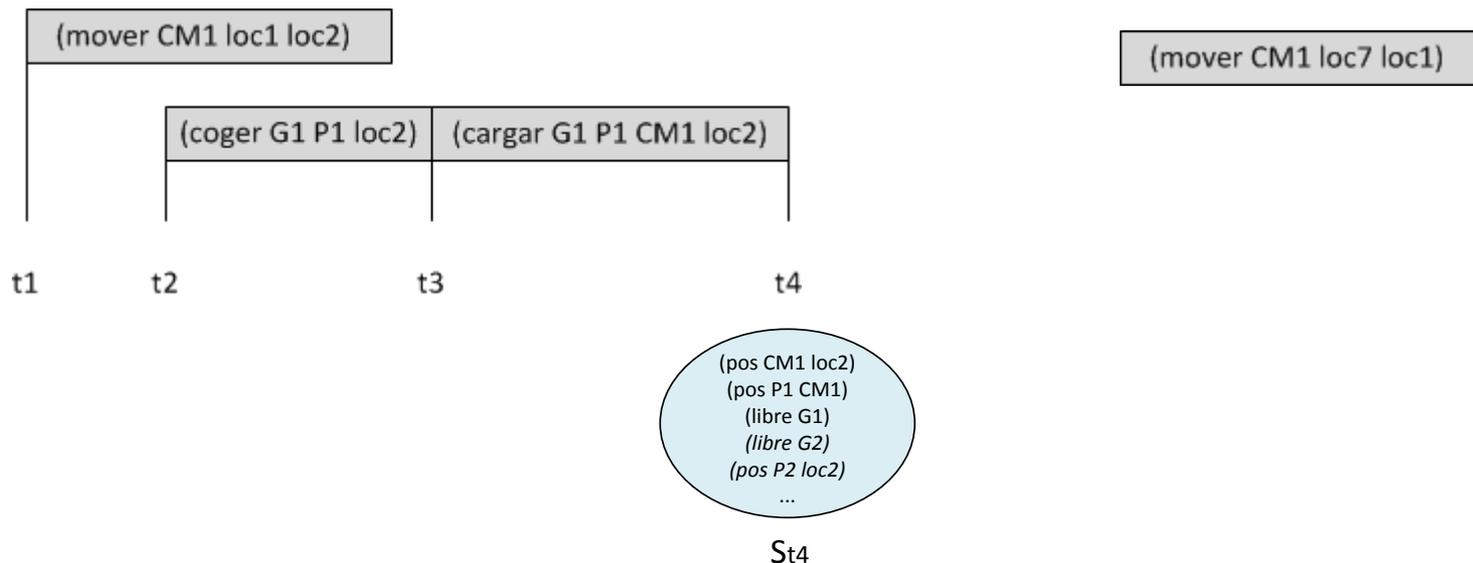
2. Planificación temporal

Construcción de un plan temporal (inserción de una acción en un plan)

Aplicabilidad de una acción a en un plan Π :

1. Comprobar condiciones de a se satisfacen inicialmente en el estado de Π
2. Comprobar consistencia de a con el resto de acciones en Π

Dado un plan temporal consistente Π tal que $\text{dur}(\Pi)=t$, St es el estado que se alcanza tras la *ejecución* de Π en el estado inicial del problema



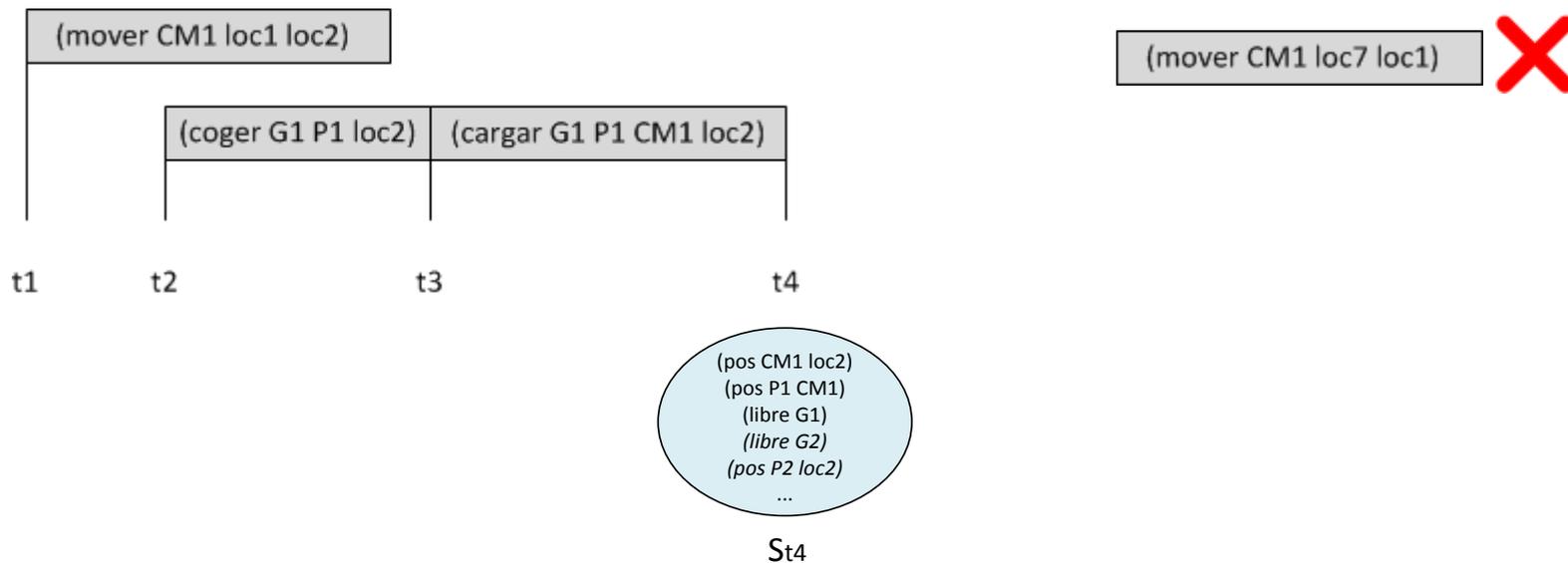
2. Planificación temporal

Construcción de un plan temporal (inserción de una acción en un plan)

Aplicabilidad de una acción a en un plan Π :

1. Comprobar condiciones de a se satisfacen inicialmente en el estado de Π
2. Comprobar consistencia de a con el resto de acciones en Π

Dado un plan temporal consistente Π tal que $\text{dur}(\Pi)=t$, St es el estado que se alcanza tras la *ejecución* de Π en el estado inicial del problema



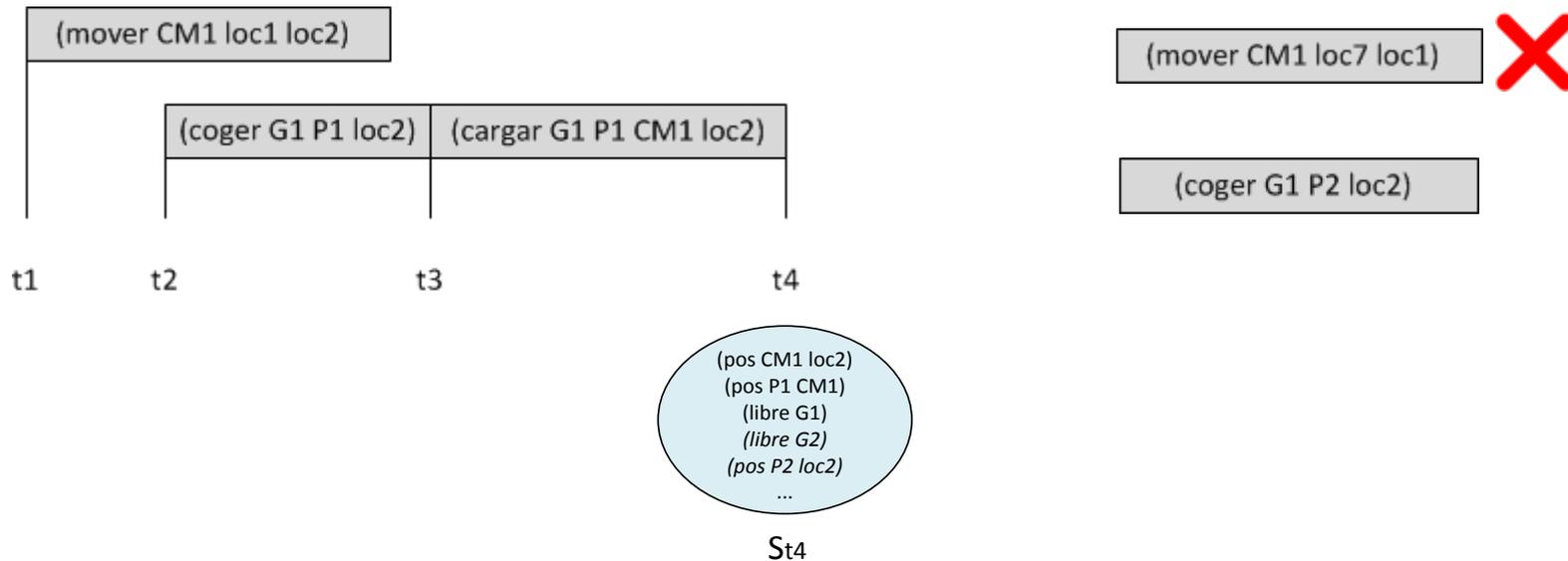
2. Planificación temporal

Construcción de un plan temporal (inserción de una acción en un plan)

Aplicabilidad de una acción a en un plan Π :

1. Comprobar condiciones de a se satisfacen inicialmente en el estado de Π
2. Comprobar consistencia de a con el resto de acciones en Π

Dado un plan temporal consistente Π tal que $\text{dur}(\Pi)=t$, S_t es el estado que se alcanza tras la *ejecución* de Π en el estado inicial del problema



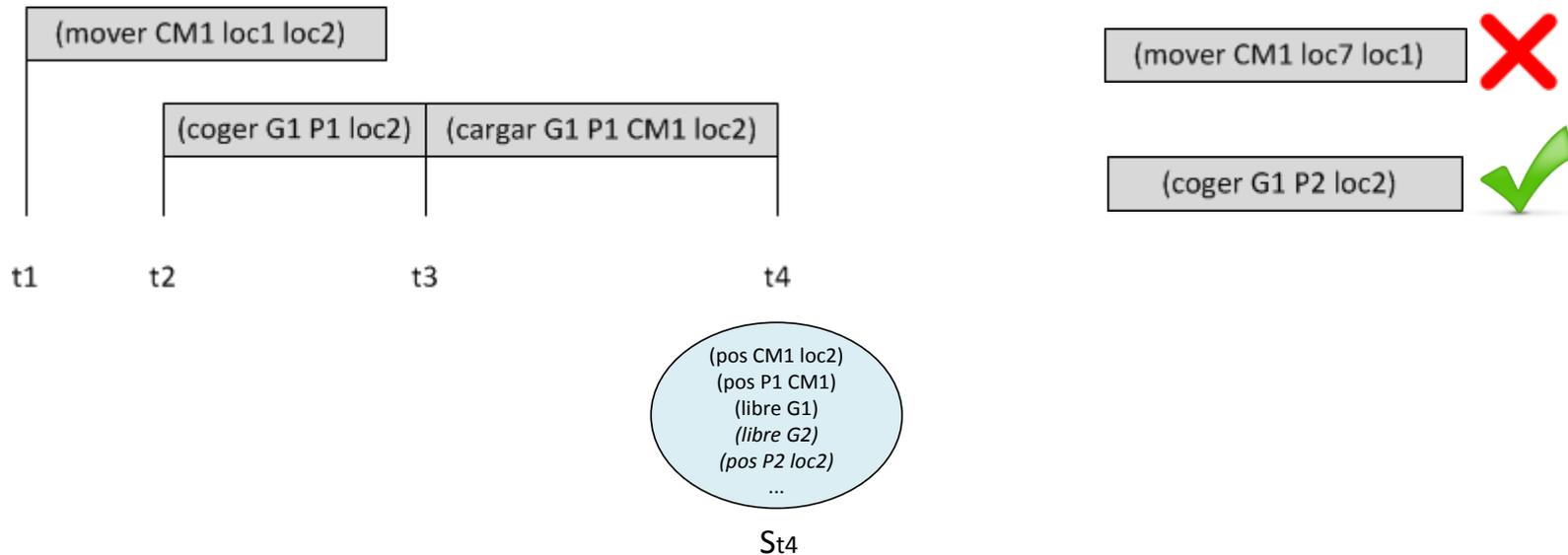
2. Planificación temporal

Construcción de un plan temporal (inserción de una acción en un plan)

Aplicabilidad de una acción a en un plan Π :

1. Comprobar condiciones de a se satisfacen inicialmente en el estado de Π
2. Comprobar consistencia de a con el resto de acciones en Π

Dado un plan temporal consistente Π tal que $\text{dur}(\Pi)=t$, St es el estado que se alcanza tras la *ejecución* de Π en el estado inicial del problema



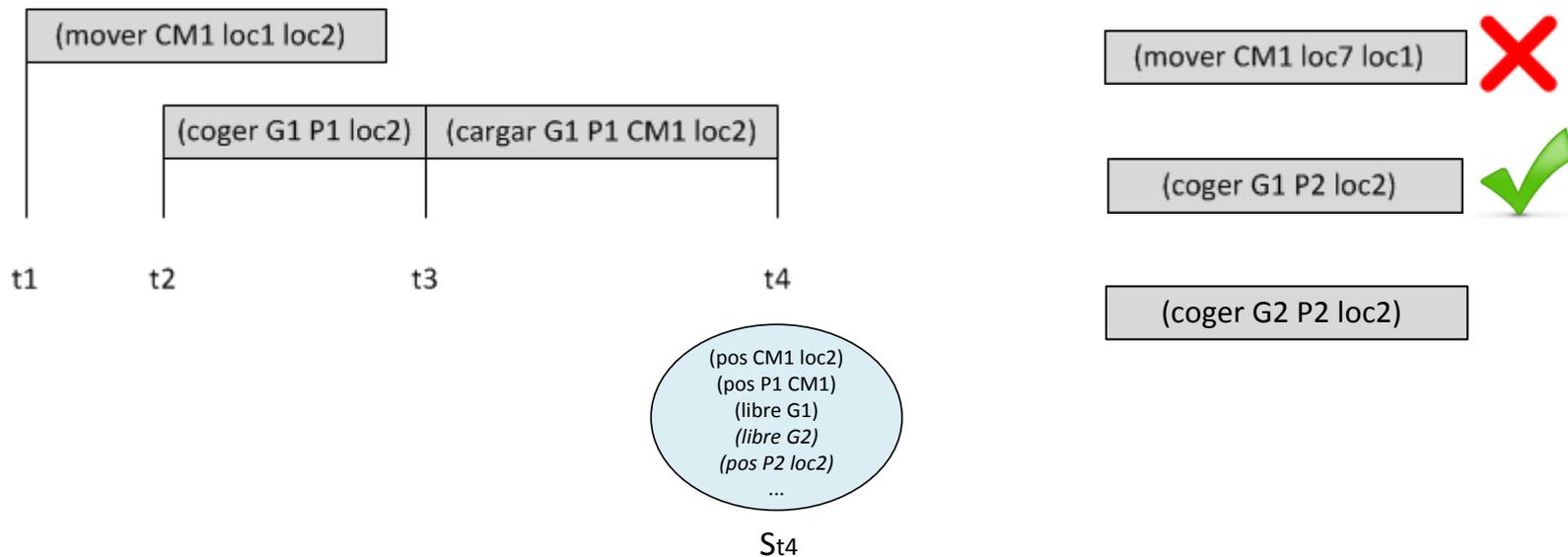
2. Planificación temporal

Construcción de un plan temporal (inserción de una acción en un plan)

Aplicabilidad de una acción α en un plan Π :

1. Comprobar condiciones de α se satisfacen inicialmente en el estado de Π
2. Comprobar consistencia de α con el resto de acciones en Π

Dado un plan temporal consistente Π tal que $\text{dur}(\Pi)=t$, St es el estado que se alcanza tras la *ejecución* de Π en el estado inicial del problema



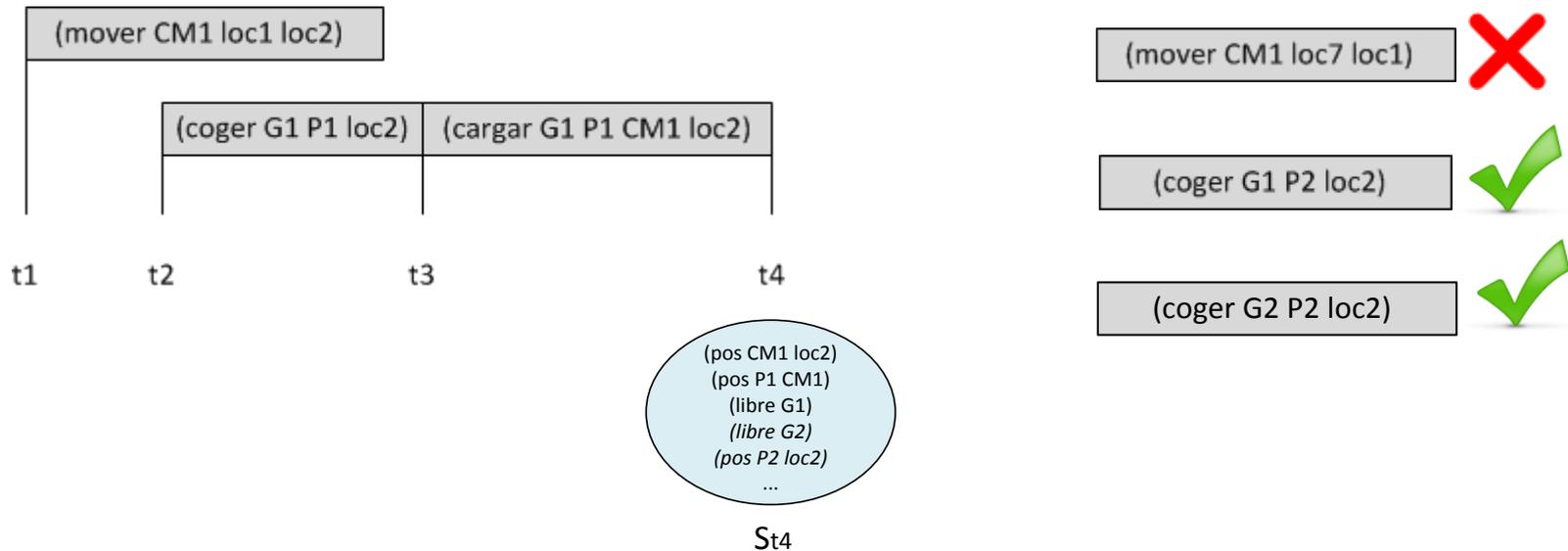
2. Planificación temporal

Construcción de un plan temporal (inserción de una acción en un plan)

Aplicabilidad de una acción α en un plan Π :

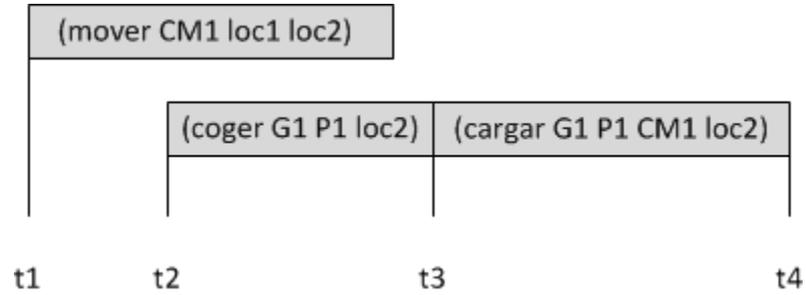
1. Comprobar condiciones de α se satisfacen inicialmente en el estado de Π
2. Comprobar consistencia de α con el resto de acciones en Π

Dado un plan temporal consistente Π tal que $\text{dur}(\Pi)=t$, St es el estado que se alcanza tras la *ejecución* de Π en el estado inicial del problema

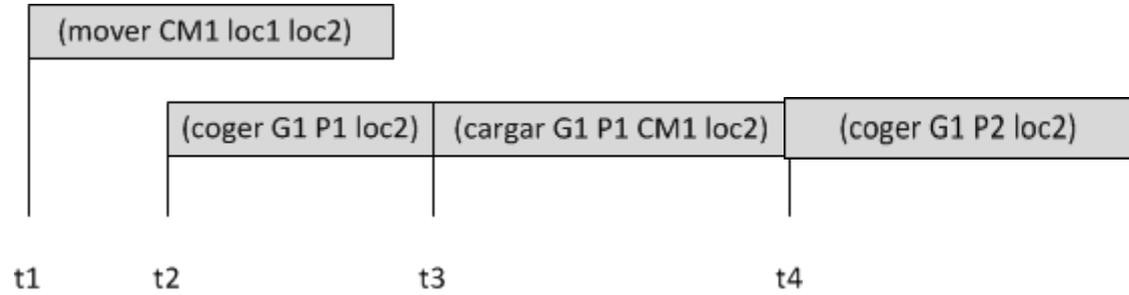


2. Planificación temporal

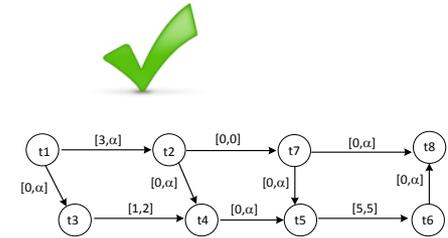
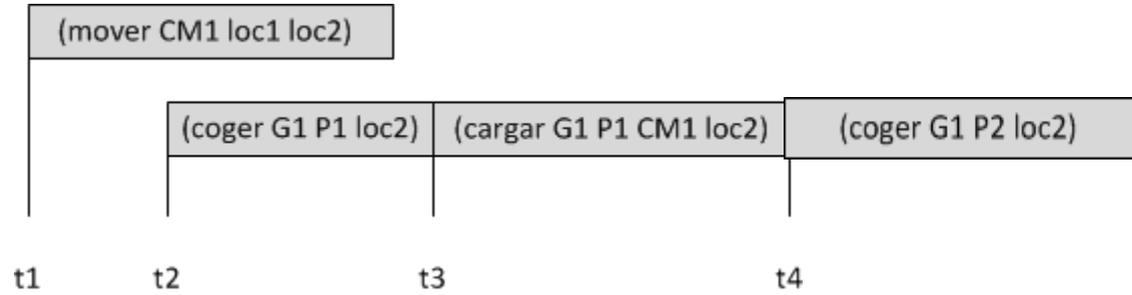
2. Planificación temporal



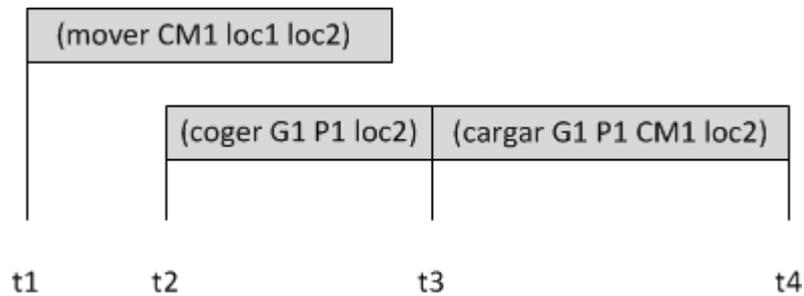
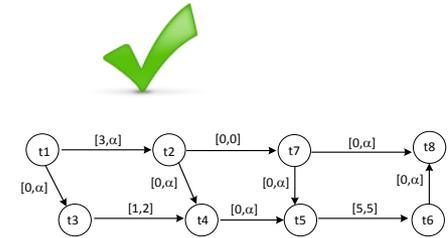
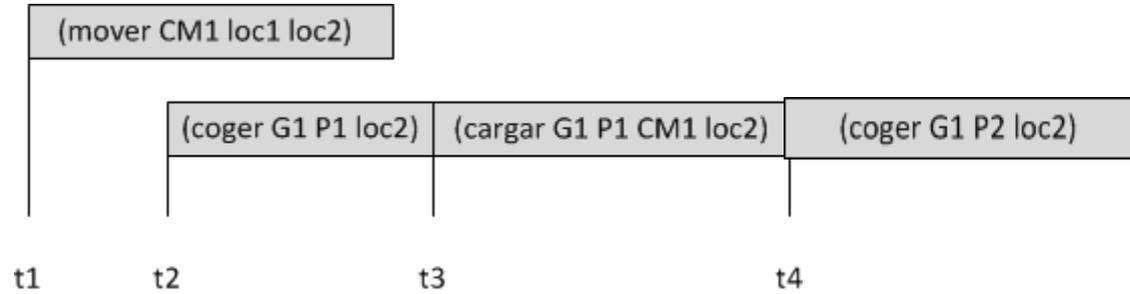
2. Planificación temporal



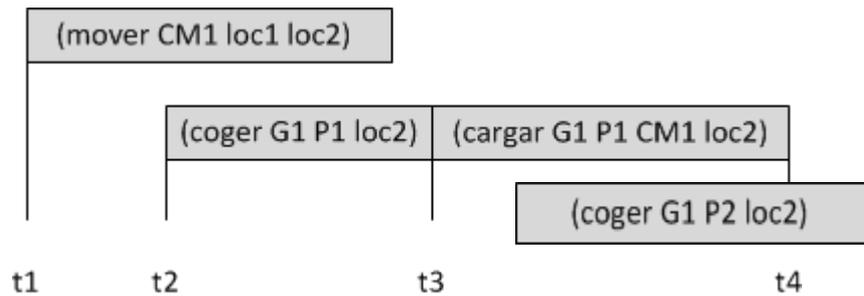
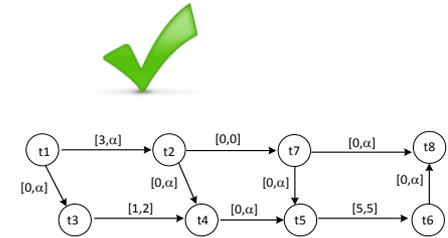
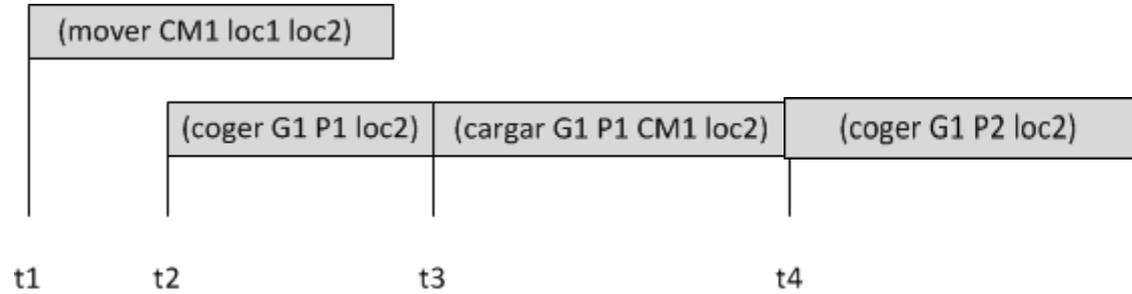
2. Planificación temporal



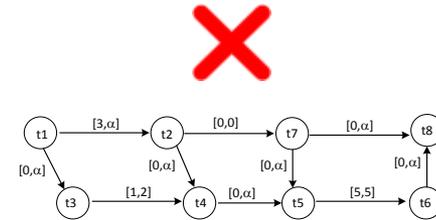
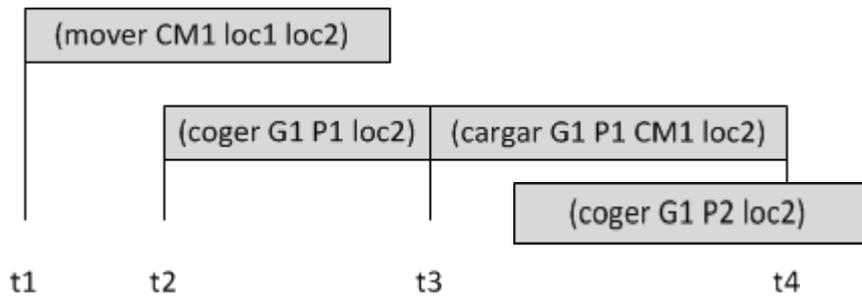
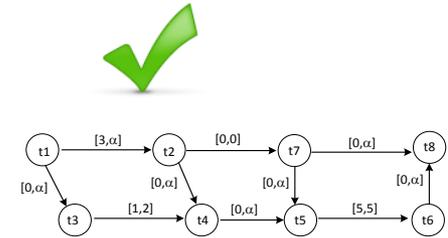
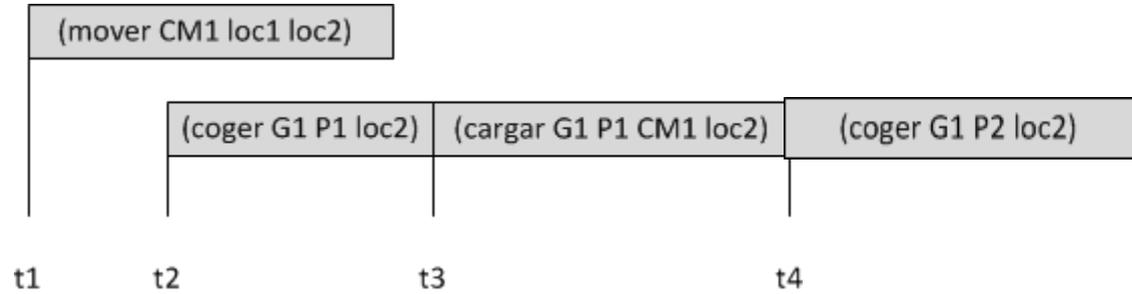
2. Planificación temporal



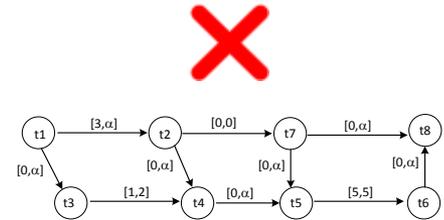
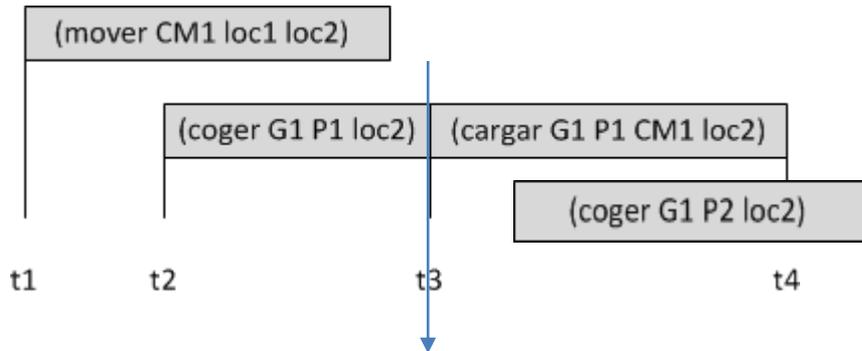
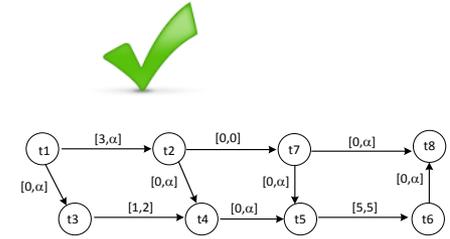
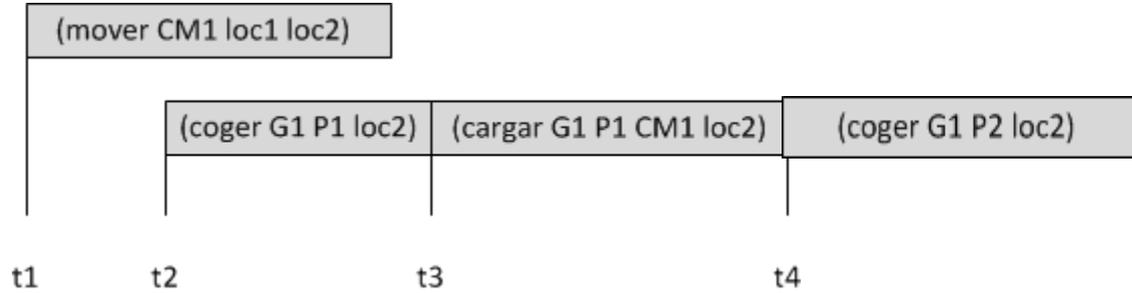
2. Planificación temporal



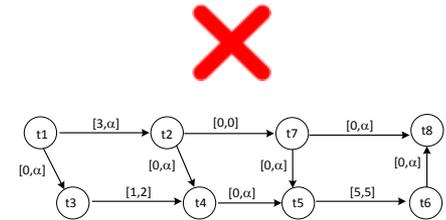
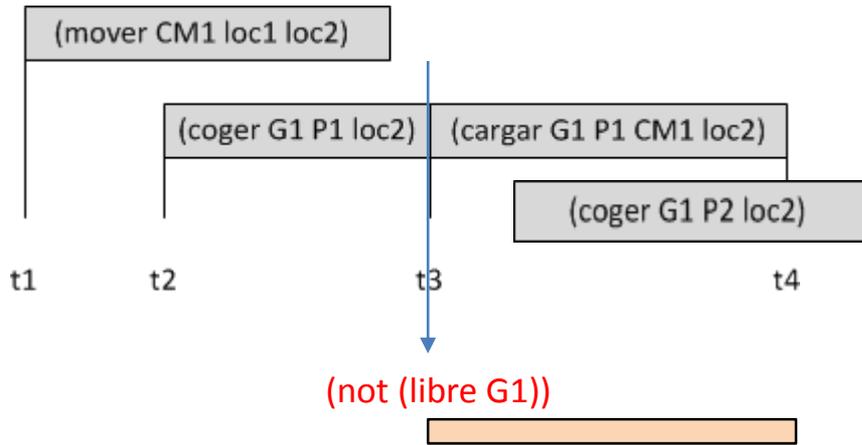
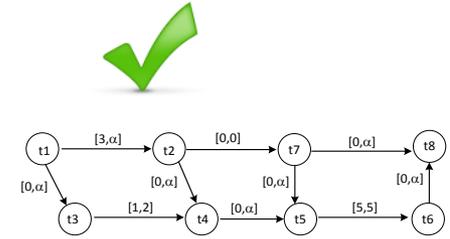
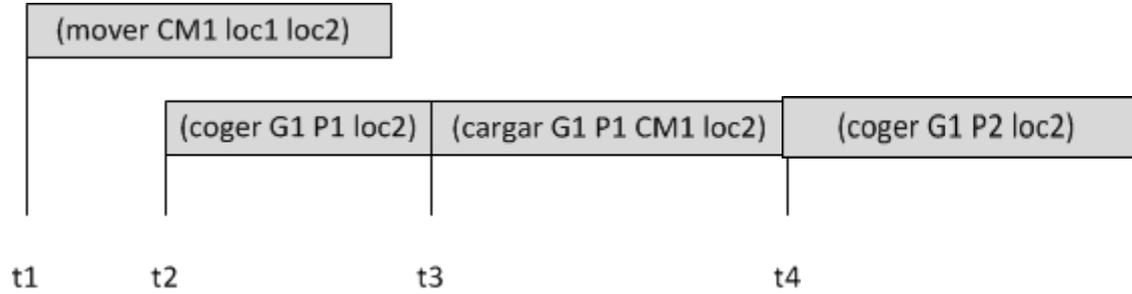
2. Planificación temporal



2. Planificación temporal

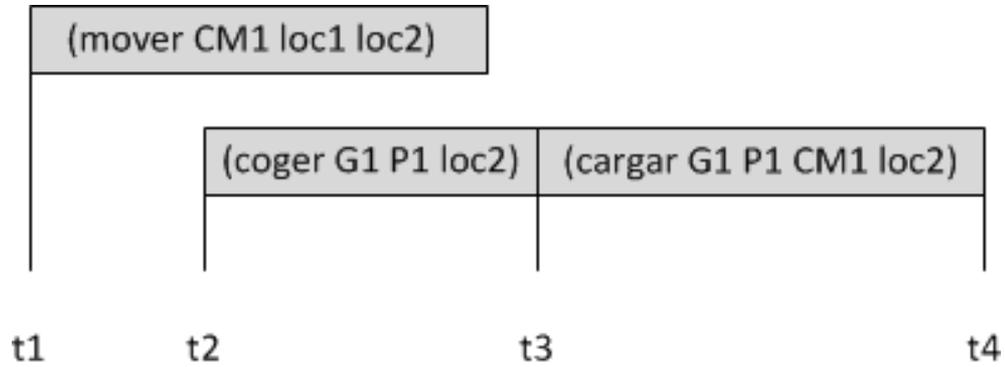


2. Planificación temporal

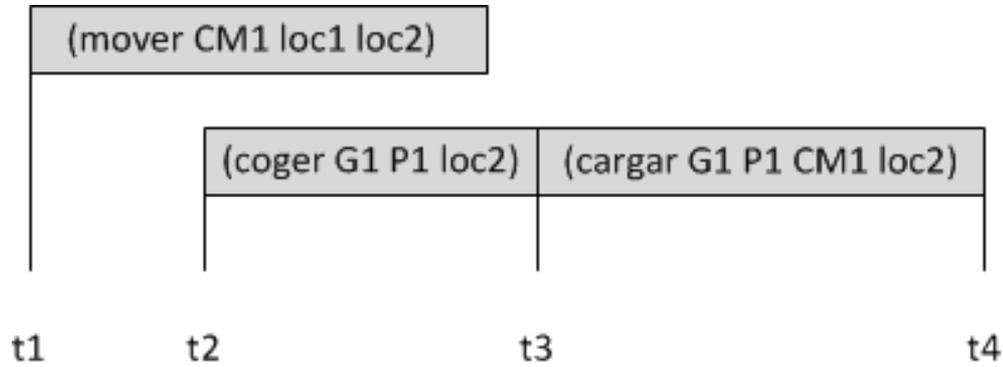


2. Planificación temporal

2. Planificación temporal



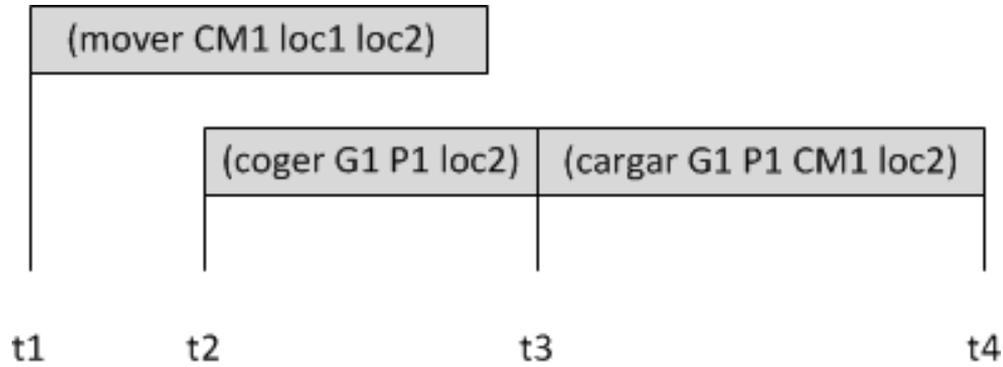
2. Planificación temporal



(coger G2 P2 loc2)



2. Planificación temporal



(coger G2 P2 loc2)



2. Planificación temporal

Representación y Razonamiento temporal en planificación

1. Representación:
 - Plan temporal
 - Red de restricciones temporal (solo puntos de tiempo) o grafo de planificación temporal (proposiciones y acciones) para representar la información temporal del plan
2. Comprobación del cumplimiento de las condiciones de las acciones en el plan (consistencia de planificación)
3. Comprobación de la consistencia temporal de la acción con el resto de acciones del plan
4. Aplicación de algoritmos de planificación y razonamiento temporal

2. Planificación temporal

Planificadores temporales:

- **Grafos de planificación temporal:** TPG, TPSYS (PDDL2.1)
- **Grafos de acciones:** LPG (PDDL2.1)
- **Busqueda heurística:** Sapa, TP4
- **Planificación de orden parcial:** HSTS, ZENO, VHPOP, OPTIC (PDDL3.0), TempLM (PDDL3.0)

OPTIC [Benton12] y TempLM [Marzal16]:

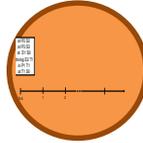
1. Son dos de los planificadores temporales más recientes y eficientes
2. Funcionan con un esquema de planificación de orden parcial hacia delante
3. Manejan restricciones PDDL3.0

Benton 12. J Benton, A. Coles, A. Coles. International Conference on P&S systems. 2012

Marzal 16. E. Marzal, L. Sebastián, E. Onaindia. Knowledge-based Systems, 2016.

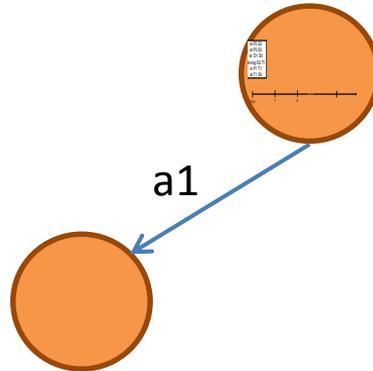
2. Planificación temporal

Planificador temporal: búsqueda hacia adelante en un espacio de planes



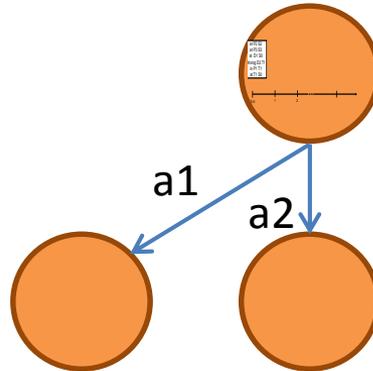
2. Planificación temporal

Planificador temporal: búsqueda hacia delante en un espacio de planes



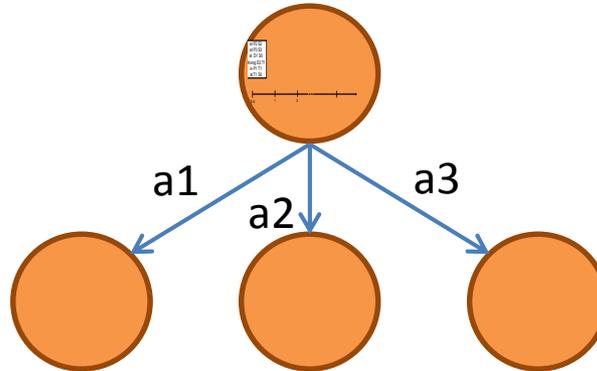
2. Planificación temporal

Planificador temporal: búsqueda hacia adelante en un espacio de planes



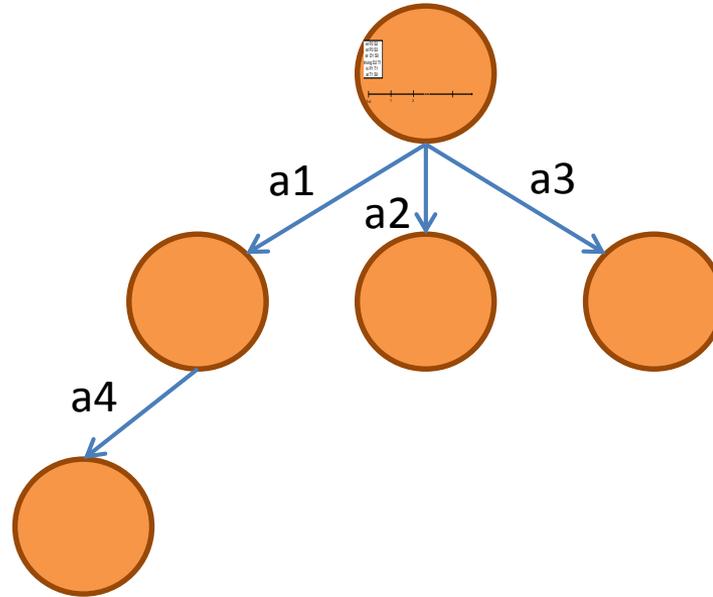
2. Planificación temporal

Planificador temporal: búsqueda hacia adelante en un espacio de planes



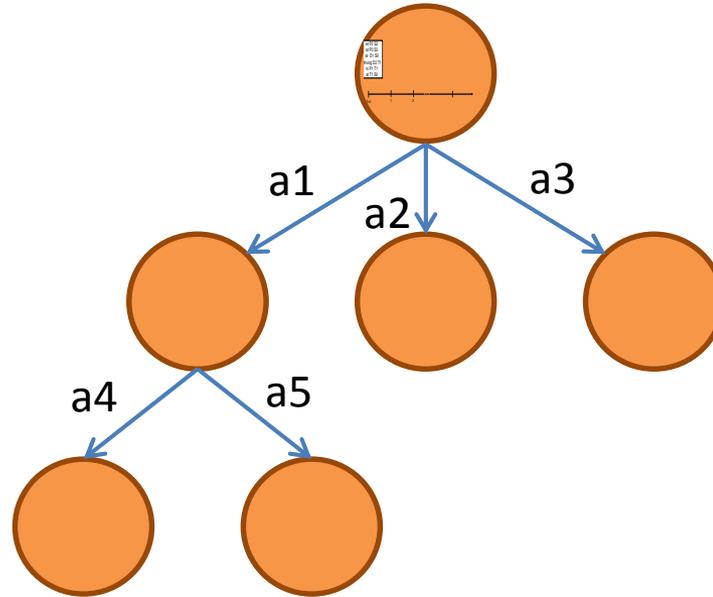
2. Planificación temporal

Planificador temporal: búsqueda hacia adelante en un espacio de planes



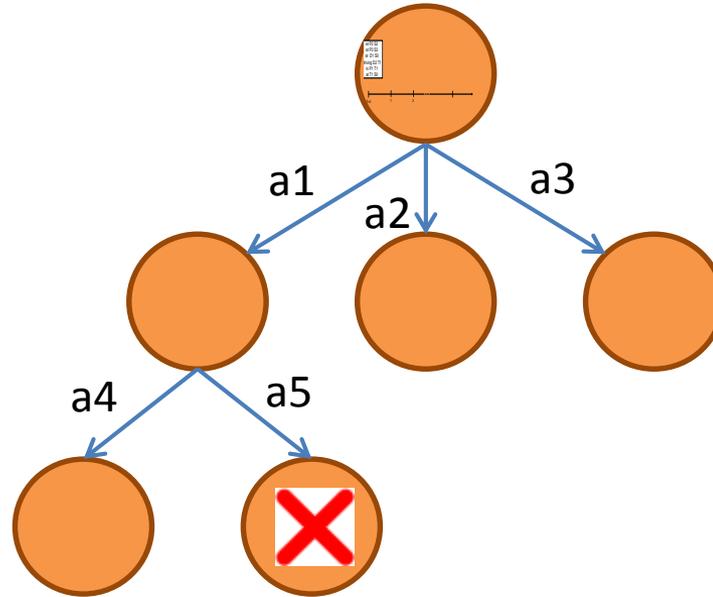
2. Planificación temporal

Planificador temporal: búsqueda hacia adelante en un espacio de planes



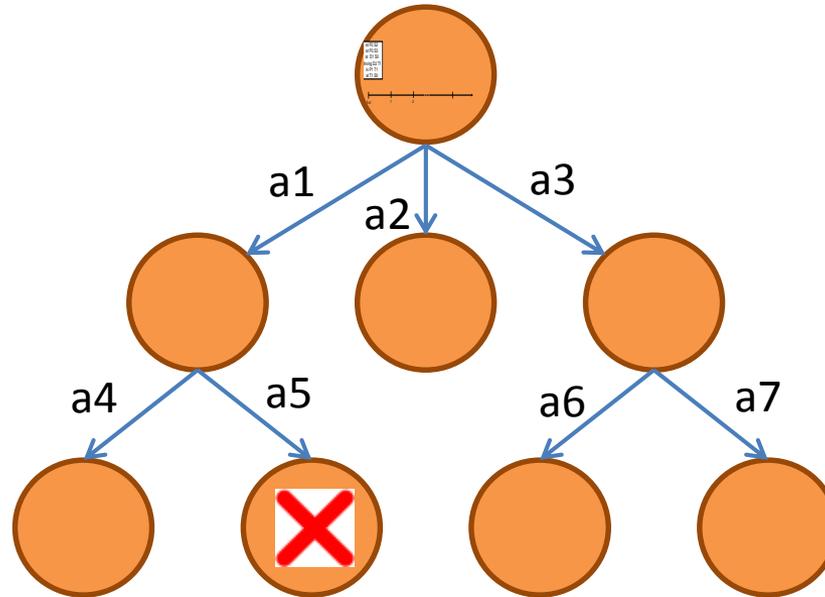
2. Planificación temporal

Planificador temporal: búsqueda hacia adelante en un espacio de planes



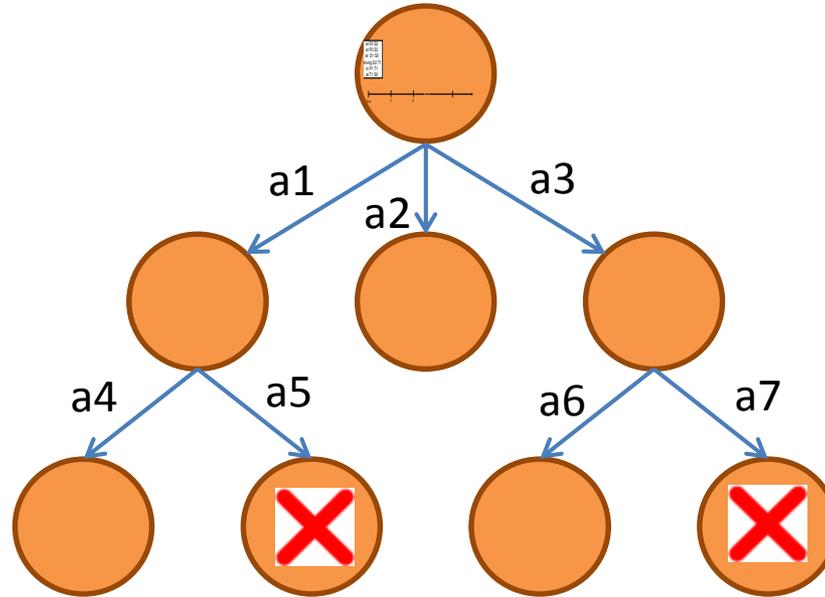
2. Planificación temporal

Planificador temporal: búsqueda hacia adelante en un espacio de planes



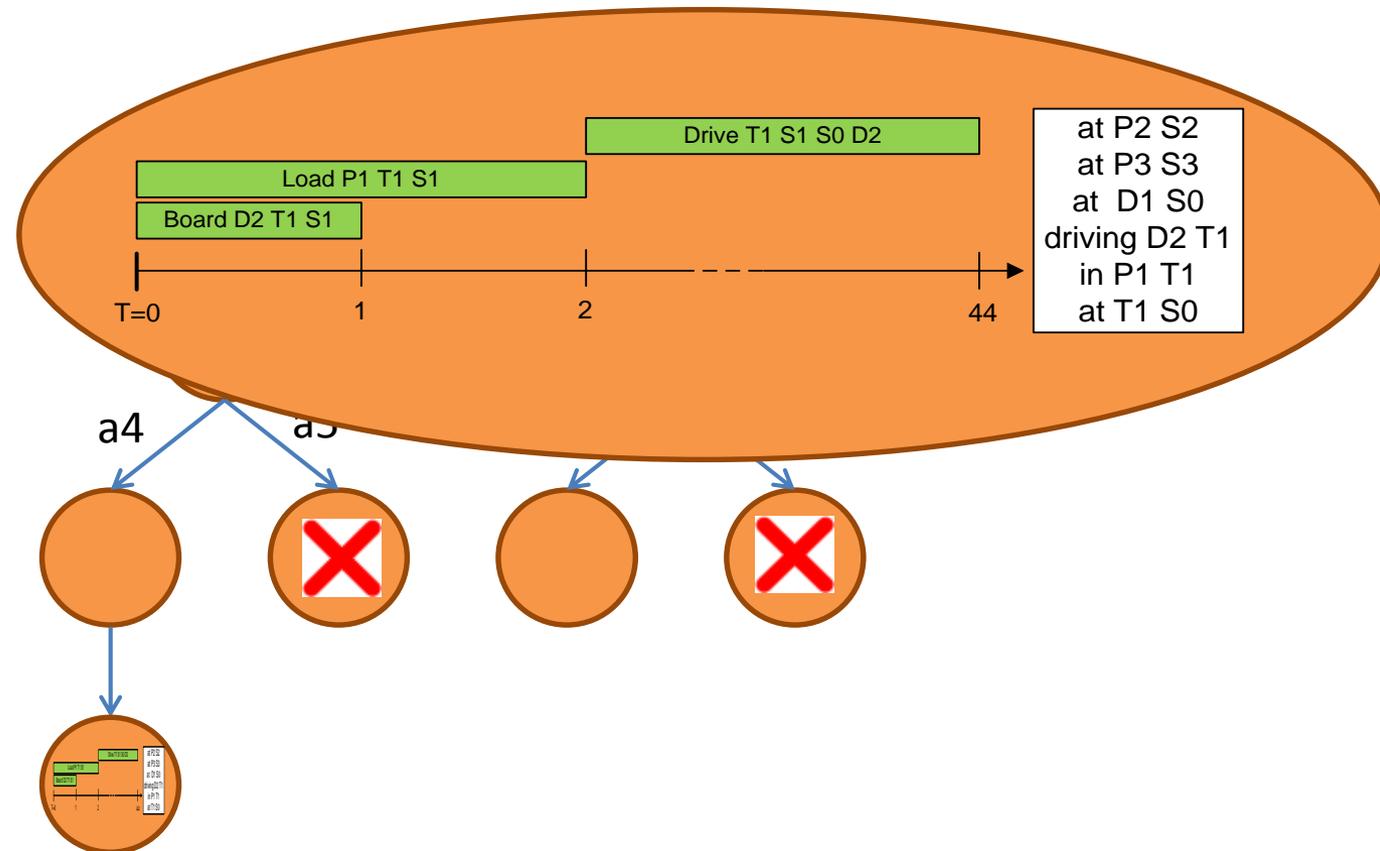
2. Planificación temporal

Planificador temporal: búsqueda hacia adelante en un espacio de planes



2. Planificación temporal

Planificador temporal: búsqueda hacia adelante en un espacio de planes



3. Lenguaje PDDL3.0

Versión 3.0 del lenguaje de descripción de problemas de planificación que incluye *deadlines* y otras restricciones de tiempo

El camion 7 tiene que estar en cityA antes del instante 250	(within 250 (pos CM7 cA))
Cuando el depósito de la gasolina de un camion esté por debajo del 20%, debe ir a repostar en las siguientes 10 unidades de tiempo	(forall (?cm - CAMION) (always-within 10 (< (gasolina ?cm) 20) (pos ?cm gas-repost))))
Un camion puede ir a una ciudad determinada solo tras haber pasado previamente por otra ciudad particular	(forall (?cm - CAMION) (sometime-before (pos ?cm c1)(pos ?cm c2)))

3. Lenguaje PDDL3.0

Versión 3.0 del lenguaje de descripción de problemas de planificación que incluye *deadlines* y otras restricciones de tiempo

El camion 7 tiene que estar en cityA antes del instante 250	(within 250 (pos CM7 cA))
Cuando el depósito de la gasolina de un camion esté por debajo del 20%, debe ir a repostar en las siguientes 10 unidades de tiempo	(forall (?cm - CAMION) (always-within 10 (< (gasolina ?cm) 20) (pos ?cm gas-repost))))
Un camion puede ir a una ciudad determinada solo tras haber pasado previamente por otra ciudad particular	(forall (?cm - CAMION) (sometime-before (pos ?cm c1)(pos ?cm c2)))

La inclusión de restricciones temporales adicionales:

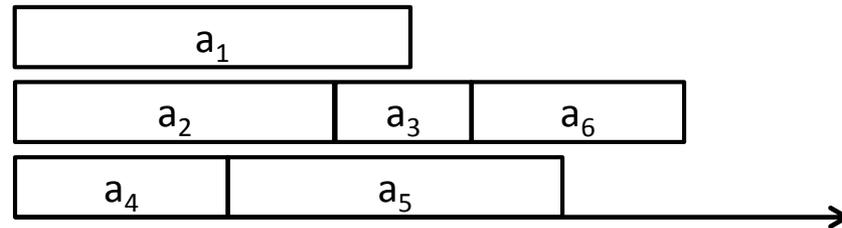
1. Enriquece la descripción de los problemas
2. Ayuda a podar nodos del árbol de búsqueda
3. Mayor complejidad y tiempo de cómputo en la comprobación de las restricciones.
INCOVENIENTE: algunas restricciones deben comprobarse en planes solución
4. Se puede hacer un análisis inteligente de restricciones e incorporar dicha información en la representación interna

4. Gestión de *deadlines*

1. Expresiones con los operadores modales **within** y **always-within**
2. Un *deadline* es una restricción de tiempo que indica el instante de tiempo máximo en el que se debe alcanzar una proposición
3. Se pueden expresar también con Timed Initial Literals
(at 100 (open shop))
(at 250 (not (open shop)))
4. En un problema con deadlines, no todos los planes que consiguen los objetivos son planes solución, ahora deben cumplirse también todos los deadlines impuestos

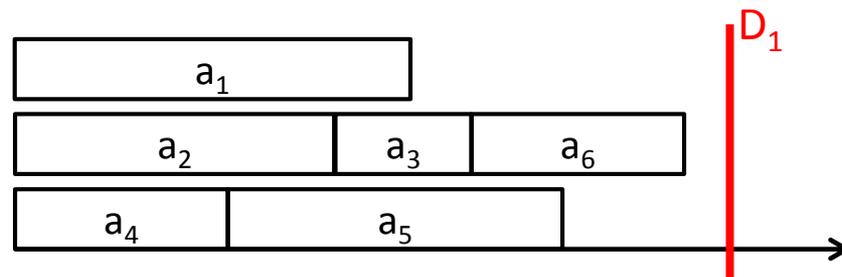
4. Gestión de *deadlines*

1. Expresiones con los operadores modales **within** y **always-within**
2. Un *deadline* es una restricción de tiempo que indica el instante de tiempo máximo en el que se debe alcanzar una proposición
3. Se pueden expresar también con Timed Initial Literals
(at 100 (open shop))
(at 250 (not (open shop)))
4. En un problema con deadlines, no todos los planes que consiguen los objetivos son planes solución, ahora deben cumplirse también todos los deadlines impuestos



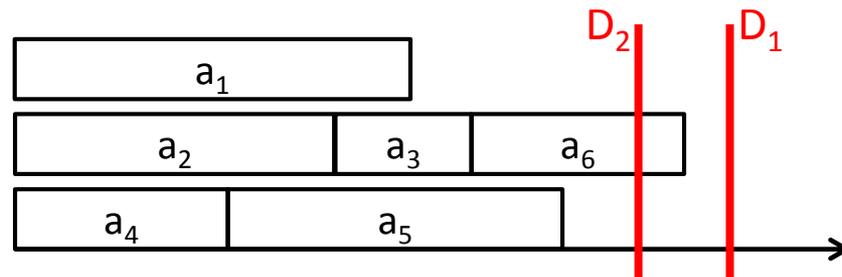
4. Gestión de *deadlines*

1. Expresiones con los operadores modales **within** y **always-within**
2. Un *deadline* es una restricción de tiempo que indica el instante de tiempo máximo en el que se debe alcanzar una proposición
3. Se pueden expresar también con Timed Initial Literals
(at 100 (open shop))
(at 250 (not (open shop)))
4. En un problema con deadlines, no todos los planes que consiguen los objetivos son planes solución, ahora deben cumplirse también todos los deadlines impuestos



4. Gestión de *deadlines*

1. Expresiones con los operadores modales **within** y **always-within**
2. Un *deadline* es una restricción de tiempo que indica el instante de tiempo máximo en el que se debe alcanzar una proposición
3. Se pueden expresar también con Timed Initial Literals
(at 100 (open shop))
(at 250 (not (open shop)))
4. En un problema con deadlines, no todos los planes que consiguen los objetivos son planes solución, ahora deben cumplirse también todos los deadlines impuestos

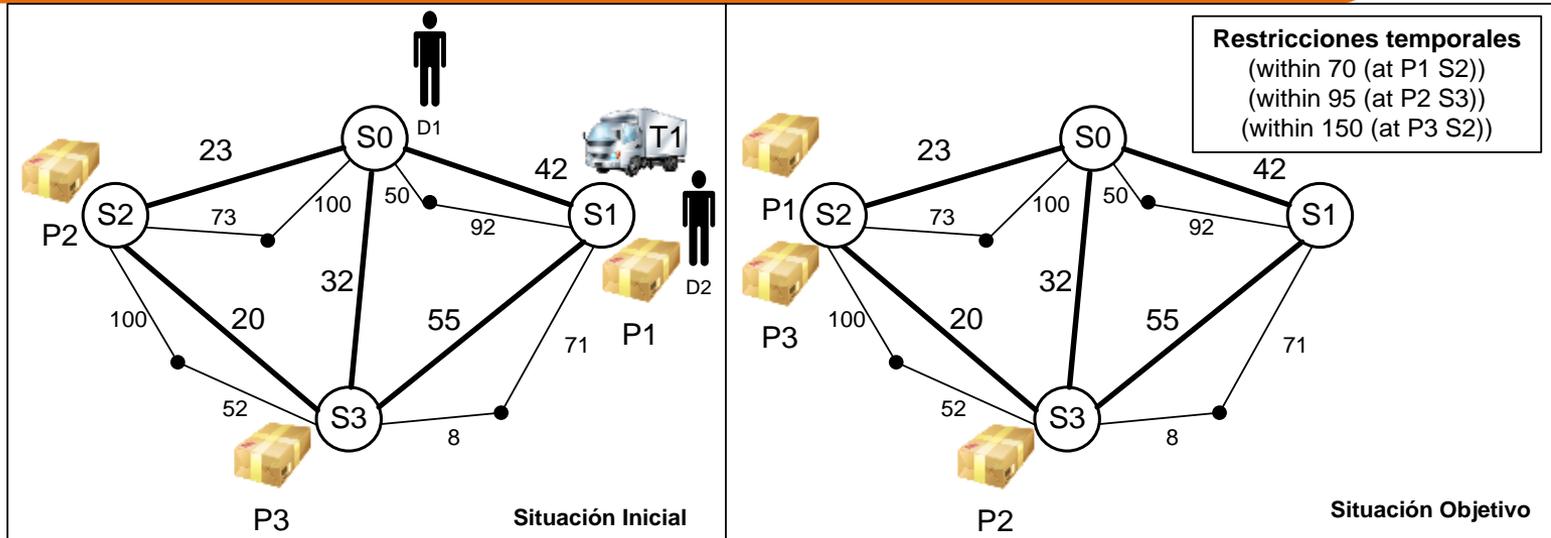


4. Gestión de *deadlines*

1. Un *landmark* es una proposición que debe satisfacerse en algún momento en todos los planes solución.
2. Ejemplos:
 1. Si se dispone de un único vehículo (CM7) en el problema → (pos P1 CM7)
 2. Si para ir de A a B hay que pasar necesariamente por C → (pos CM7 C)

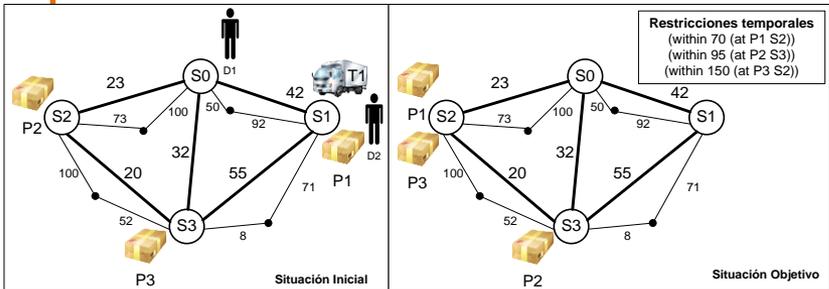
4. Gestión de *deadlines*

Problema de Planificación Temporal con *Deadlines*



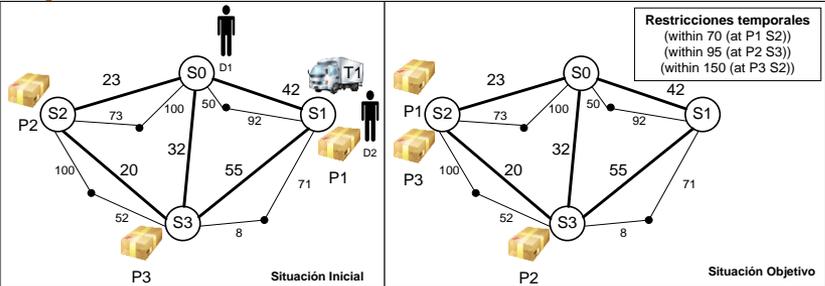
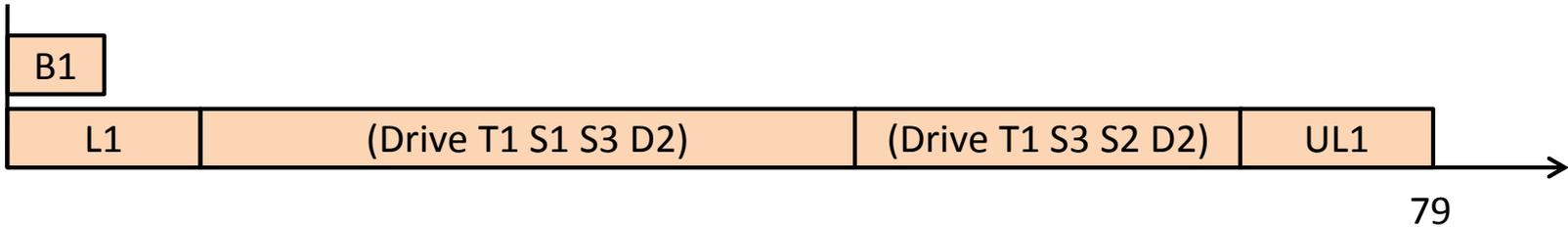
4. Gestión de *deadlines*

Problema de Planificación Temporal con *Deadlines*



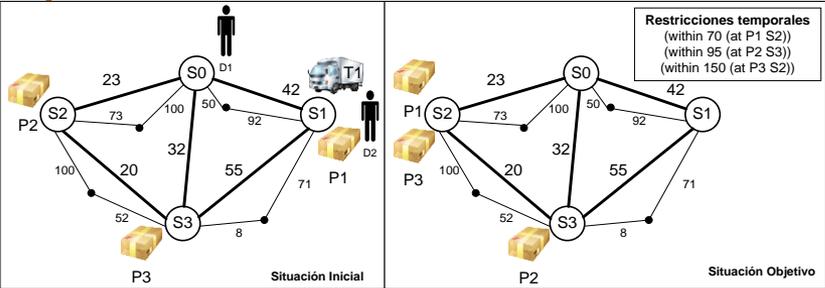
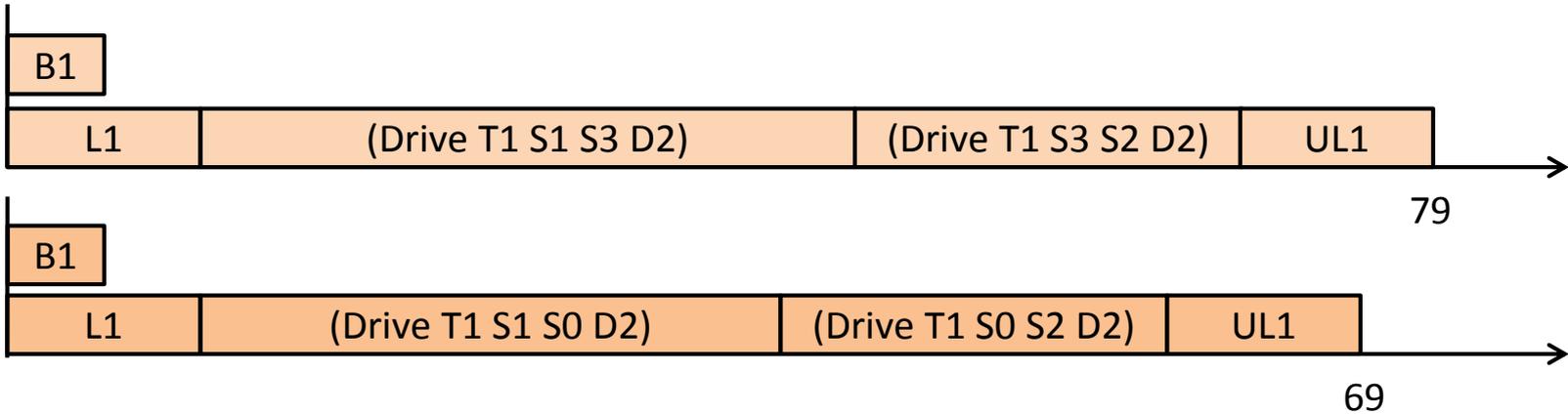
4. Gestión de *deadlines*

Problema de Planificación Temporal con *Deadlines*



4. Gestión de *deadlines*

Problema de Planificación Temporal con *Deadlines*



4. Gestión de *deadlines*

Problema de planificación temporal con *deadlines*

$$\mathcal{P} = \langle \mathcal{A}, \mathcal{I}, \mathcal{G}, \mathcal{D} \rangle$$

- *Deadline* sobre una proposición: (t, l)
- *Deadline* del problema: $T_{\mathcal{P}} = \max_{(t,l) \in \mathcal{D}} (t)$

4. Gestión de *deadlines*

Landmark temporal

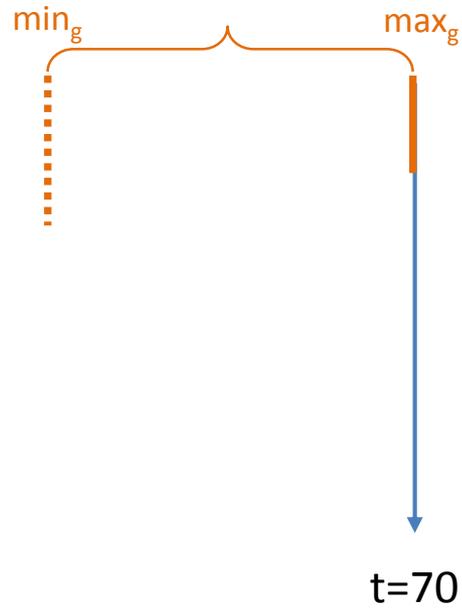
4. Gestión de *deadlines*

Landmark temporal



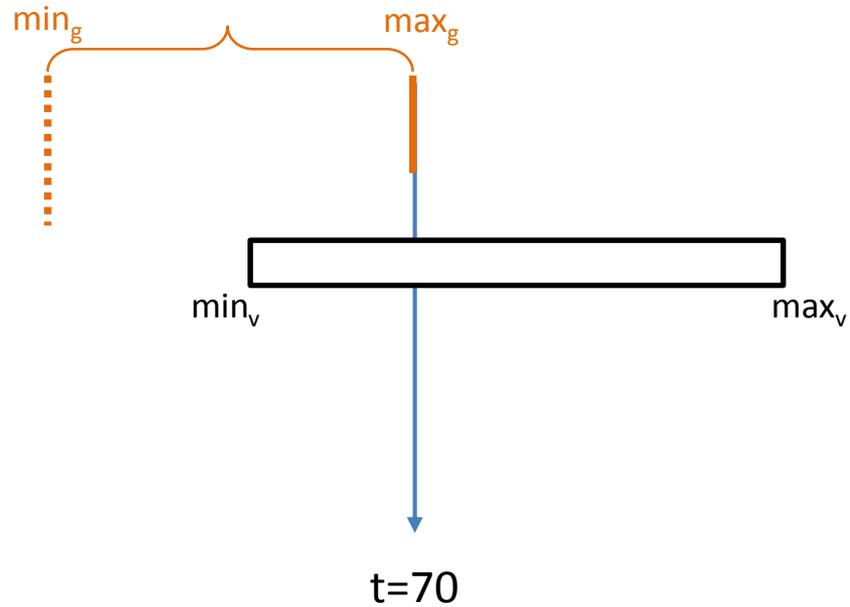
4. Gestión de *deadlines*

Landmark temporal



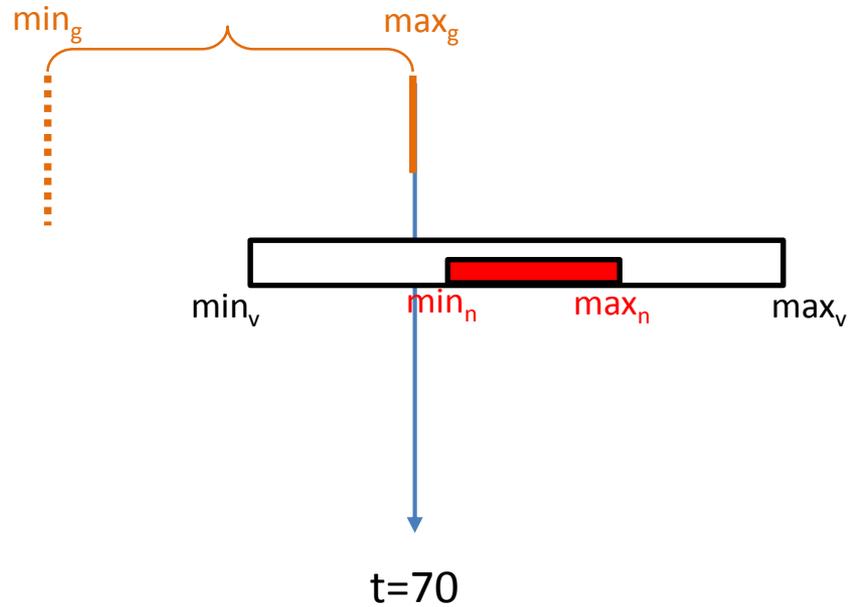
4. Gestión de *deadlines*

Landmark temporal



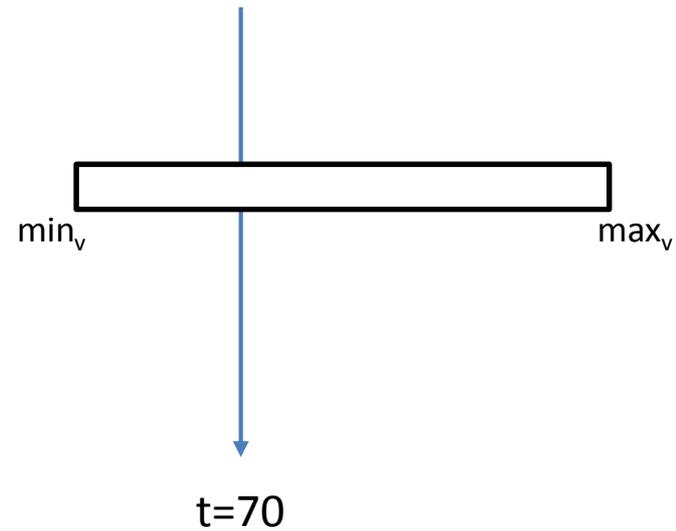
4. Gestión de *deadlines*

Landmark temporal



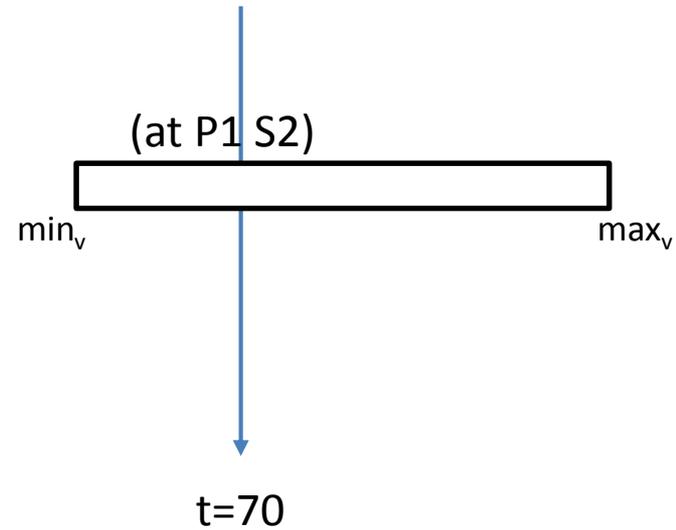
4. Gestión de *deadlines*

Landmark temporal



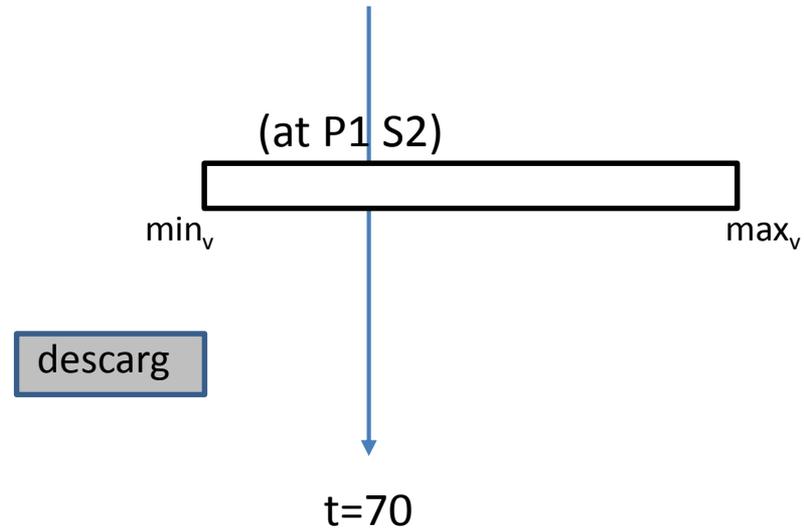
4. Gestión de *deadlines*

Landmark temporal



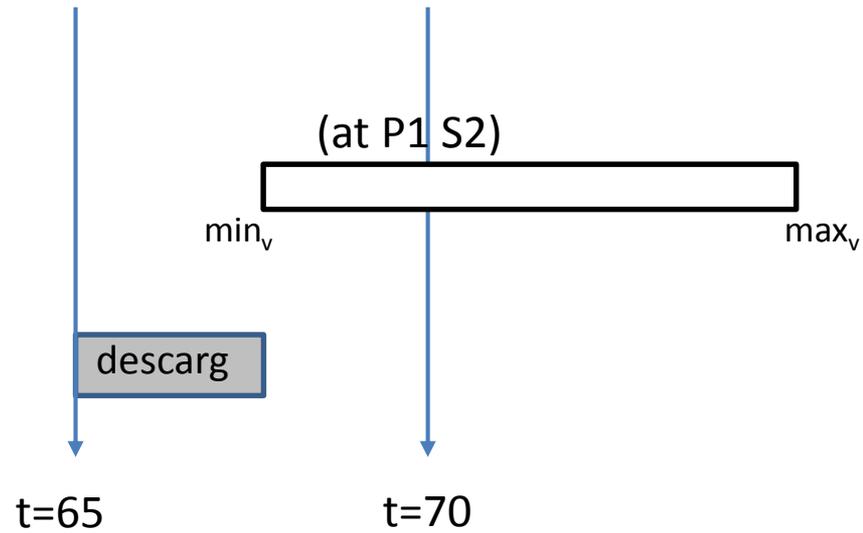
4. Gestión de *deadlines*

Landmark temporal



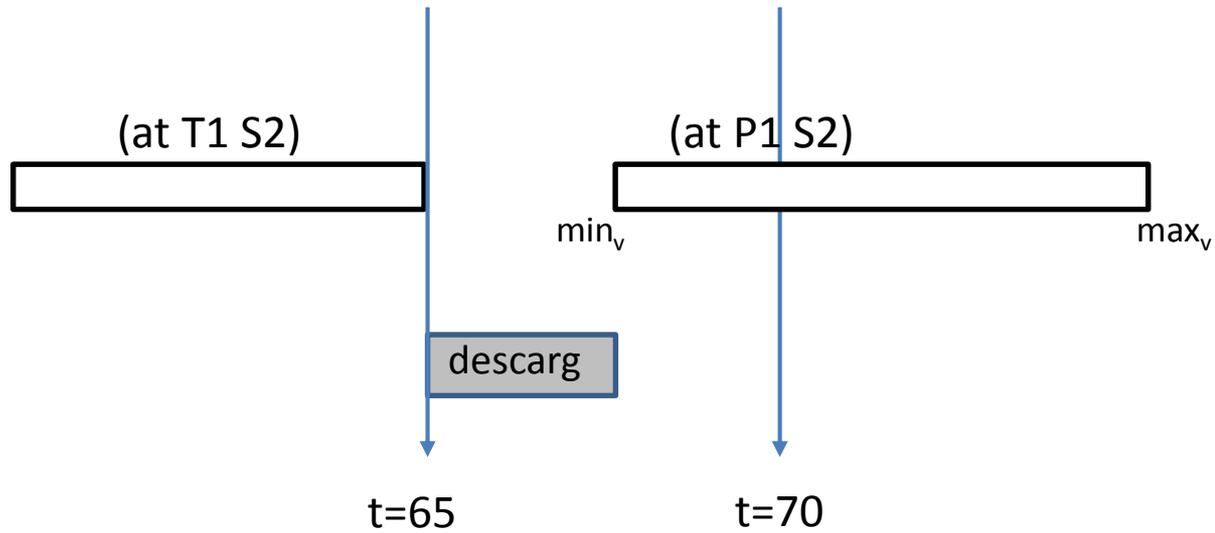
4. Gestión de *deadlines*

Landmark temporal



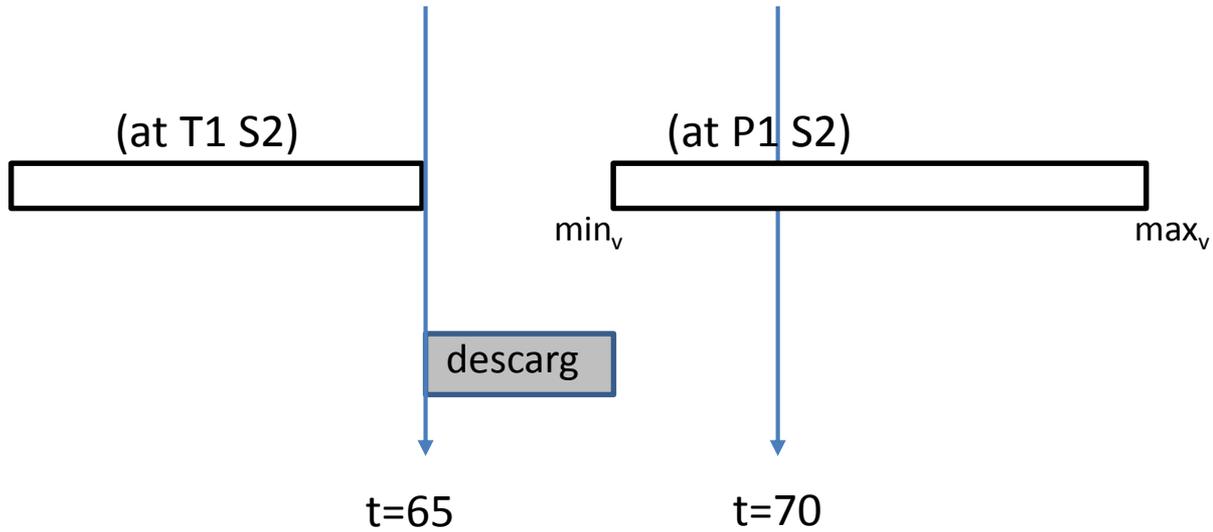
4. Gestión de *deadlines*

Landmark temporal



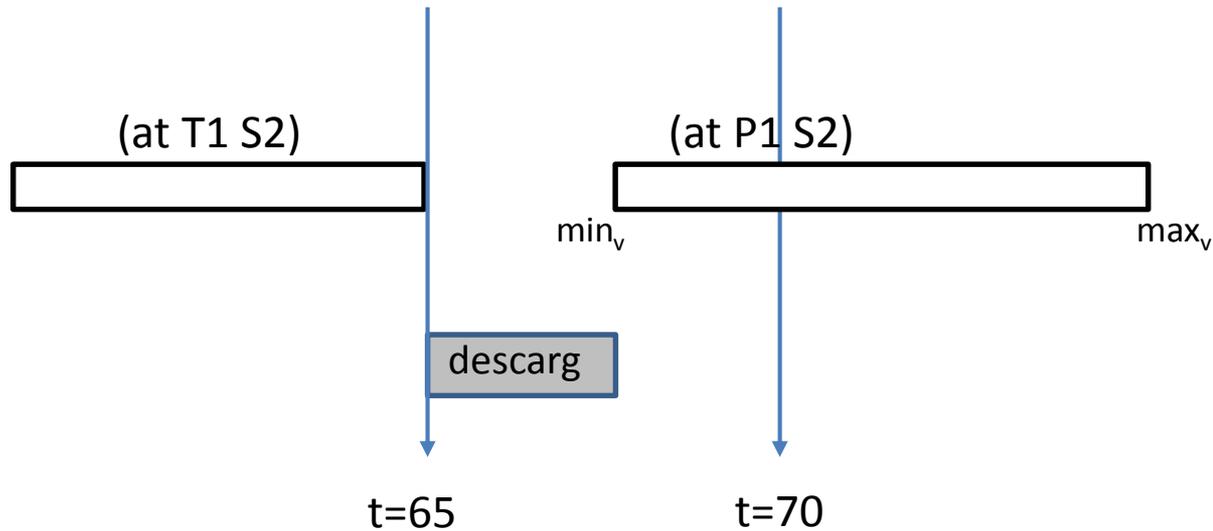
4. Gestión de *deadlines*

Landmark temporal



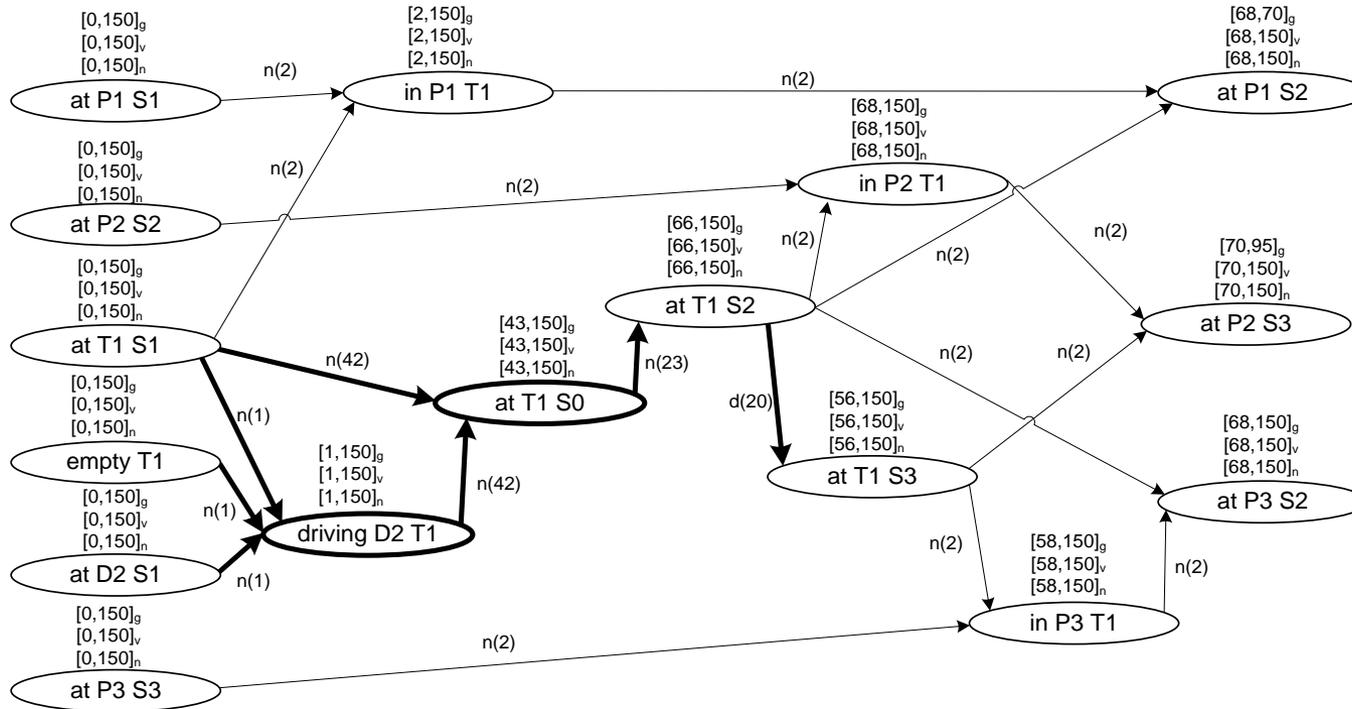
4. Gestión de *deadlines*

Landmark temporal

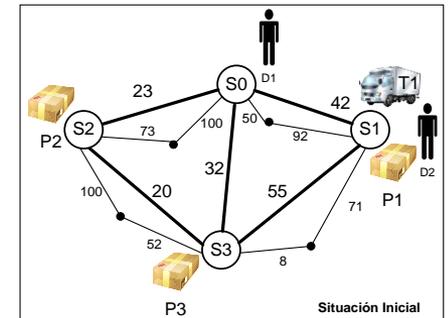


Si durante la generación de un nodo del árbol de búsqueda no se satisface alguna de las restricciones de intervalos, el nodo no se genera

4. Gestión de *deadlines*



Grafo de landmarks temporales



4. Gestión de *deadlines*

1. **Pescado fresco:**
 - debe repartirse en un tiempo límite de tiempo
 - se transporta desde los mercados de pescado locales a los puntos de distribución
2. **Productos congelados entre [-18°C, -25°C]**
 - Destino: secciones de congelados de los supermercados
3. **Productos que se mantienen a temperaturas por debajo de -25°C**
 1. Se mantienen en congeladores industriales
 2. Destino: supermercados y grandes cadenas alimentarias
4. **Transportes refrigerados: [0°C, -5°C]**
5. **Transportes de enfriamiento: -30°C**

4. Gestión de *deadlines*

Restricciones de entrega
de los usuarios

Good	Initially	Destination	Deadline
G1-1	Castellon	Bunyol	200
G2-1	Valencia	Alcoi	150
G3-1	Valencia	Ontinyent	87

Truck (type)	Initially
Tru1 (T1)	Castellon
Tru2 (T2)	Valencia

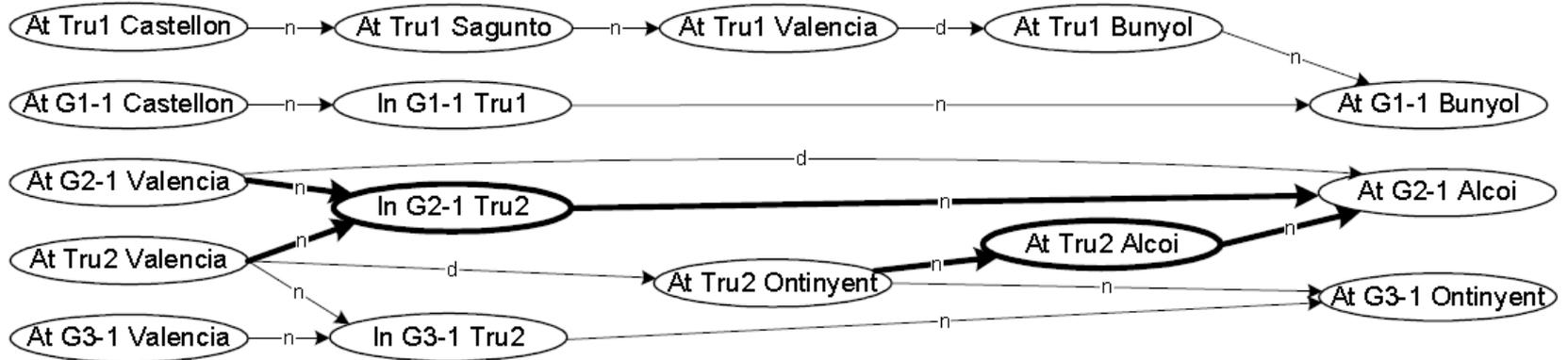
New City	City	Distance
Bunyol	Valencia	39
	Requena	30
Alcoi	Ontinyent	35

1. El pescado fresco se transporta en camiones refrigerados y no puede estar más de 3 horas en el camión
2. Los productos congelados:
 - Camiones refrigerados → máximo 3 horas
 - Camiones de enfriamiento → máximo 2 horas
3. Productos ultracongelados solo se pueden transportar en camiones de enfriamiento
4. El pescado fresco no puede estar fuera del camión refrigerado durante más de 10 minutos

4. Gestión de *deadlines*

INITIAL STATE
 At Tru1 Castellon
 At Tru2 Valencia
 At G1-1 Castellon
 At G2-1 Valencia
 At G3-1 Valencia

Load G1-1 Tru1 Cast.	Drive Tru1 Castellon Sagunto (49)	Drive Tru1 Sagunto Val. (28)	Drive Tru1 Valencia Bunyol (39)	Unl. G1 Bunyol					
Load G2-1 Tru2 Val.									
Load G3-1 Tru2 Val.	Drive Tru2 Valencia Ontinyent (83)		Unl. G2 Ont.	Drive Tru2 Ontinyent Alcoi (35)	Unl. G3 Alcoi				
	2	51	79	85	87	118	120	122	124



4. Gestión de *deadlines*

Ventajas de utilización de *landmarks* temporales

1. Rápida identificación de problemas irresolubles:
 - Diferencia entre un problema de planificación y un problema de scheduling
 - Muy apropiado para problemas altamente restringidos
2. Representación, modelización y explotación de conocimiento temporal
 - Objetivo: reducir el tiempo de búsqueda del planificador

5. Conclusiones

1. Combinación apropiada de representación del conocimiento y razonamiento
2. Razonamiento:
 - Razonamiento de planificación (condiciones de acciones, causalidad entre las acciones, conflictos, etc.)
 - Razonamiento temporal (uso de algoritmos estándar)
3. Representación:
 - Plan temporal
 - Estructura interna para mantener restricciones
 - Red temporal simple
 - Red temporal de restricciones (relaciones disyuntivas)
 - Grafos de planificación
 - Grafos de landmarks
4. Esquema de resolución de muchos problemas reales en industria, logística, etc.

5. Conclusiones

1. Combinación apropiada de representación del conocimiento y razonamiento
2. Razonamiento:
 - Razonamiento de planificación (condiciones de acciones, causalidad entre las acciones, conflictos, etc.)
 - Razonamiento temporal (uso de algoritmos estándar)
3. Representación:
 - Plan temporal
 - Estructura interna para mantener restricciones
 - Red temporal simple
 - Red temporal de restricciones (relaciones disyuntivas)
 - Grafos de planificación
 - Grafos de landmarks
4. Esquema de resolución de muchos problemas reales en industria, logística, etc.

Máster Universitario en Inteligencia Artificial (online)

<http://www.aepia.org/aepia/index.php/maesteria>

Referencias

Malik Ghallab, Dana S. Nau, Paolo Traverso
Automated planning - theory and practice.
Elsevier 2004, ISBN 978-1-55860-856-6, pp. I-XXVIII, 1-635

J Benton, Amanda Jane Coles, Andrew Coles.
Temporal Planning with Preferences and Time-Dependent Continuous Costs
International Conference on Planning and Scheduling Systems (ICAPS). 2012

Eliseo Marzal, Laura Sebastiá, Eva Onaindia.
Temporal Landmark Graphs for Solving Overconstrained Planning Problems
Knowledge-based Systems, <http://dx.doi.org/10.1016/j.knosys.2016.05.029>, 2016.