

Curso de Inteligencia Artificial

Unidad IV: Planificación

Jose Aguilar

Cemisid, Facultad de Ingeniería

Universidad de los Andes

Mérida, Venezuela

aguilar@ula.ve

Planificación

Planificar: consiste en concebir un conjunto de acciones (**un plan**) de manera de alcanzar un objetivo

La planificación consiste en **razonar** sobre las acciones posibles a realizar para **concebir planes**

La planificación es un proceso mediante el cual un agente **busca un plan** para lograr cierto objetivo.

Planificación en Agentes

El plan es una series de acciones a tomar para, partiendo de un estado inicial, cumplir el objetivo.

Lo dicho suena muy parecido a la resolución de problemas, pero se considera a la planificación diferente,

- Utilizan una manera distinta para representar las acciones, objetivos y estados.
- Difieren en la forma de construir y representar el curso de las acciones (secuencia de acciones).

Planificación en Agentes

la planificación en sí mismo debe ser hecha por razonamiento

- No se puede hacer de forma algorítmica, ya que **la planificación y el plan de ejecución**, deben ser hechos
 - Al unísono en lugar de secuencial.
 - No pueden ser realizado por módulos aislados, debido a que c/u retroalimenta a las otras.

Un agente racional adopta y ejecuta los planes más adecuados según sus facultades de razonar acerca de ellos

Un simple Agente de Planificación

```
Indicar(BC, sentencia_percepción(percebido, t))
actual <- Estado_Descripto(BC, t)
Si P=NoPlan entonces
    O<- Preguntar(BC, consulta_objetivo(t))
    P<-Planificar(actual, O, BC)
Si P=NoPlan Entonces
    Acción<-NoPosible
de lo contrario
    Acción<-Primero(P)
    P<-resto(P)
Indicar(BC, Acción_sentencia(Acción, t+1))
```

Generalidades de la Planificación

- Para representar problemas de planificación se utiliza un **lenguaje particular** (más restringido que la lógica de primer orden)
- El algoritmo que permite operar sobre dicho lenguaje se denomina **planificador**.
- La utilización de un lenguaje más restringido se debe, entre otras cosas, a propósitos de **eficiencia**.
- De todas maneras seguimos hablando de lenguajes formales (p.e. **STRIPS**)

Generalidades de la Planificación

En planificación,

- los **estados** se representan por un conjunto de **hechos** que son verdaderos, escritos en un lenguaje adecuado.
- Se consideran a la mayoría de los objetivos independientes entre sí.

subplanes

Generalidades de la Planificación

- El planificador **puede añadir acciones al plan** cuando las necesite, sin restringirse a una **secuencia lineal**
- **Ejemplos de problemas de planificación,**
 - Organizar un viaje,
 - Preparar una comida
 - Ir de Compras

Planificación en Agentes

Mapeo de los estados a acciones

- Elección de las **acciones posibles**
- Elección de los **ordenes temporales**
- Elección de los **sub-objetivos para trabajar**

Planificación determinista clásica

- Solo estado inicial conocido
- Conjunto de metas

Varios algoritmos de planificación diferentes

Representación de planes mediante STRIPS

- **STRIPS** (STanford Research Institute Problem Solver) es un lenguaje utilizado en planificación.
- Si bien ya no se utiliza el planificador STRIPS, existen numerosos planificadores que utilizan **lenguajes basados/inspirados** en “STRIPS”.

Representación de planes mediante STRIPS

- **Representación de Estados:** Son conjunciones de literales (conjunto de hechos que son verdaderos en un estado del mundo) sin variables ni funciones .

En(Casa) Tiene(Pelicula)

- **Representación de Objetivos:** Son también una conjunción de literales en donde pueden haber variables.

En(Casa) ^ Tiene(Pelicula)

- **Representación de Acciones:** Una acción u operador se representa por tres elementos:

Especificación de la acción, Precondiciones, Efectos

Una Representación del Problema de Planificación

- **Estado Inicial:** sentencia lógica sobre situación inicial (S_0)
 $\text{En}(\text{Casa}, S_0) \wedge \neg \text{Tener}(\text{leche}, S_0) \wedge \neg \text{Tener}(\text{cambures}, S_0)$
- **Estado Objetivo:** sentencia lógica sobre situación deseada
 $\exists s \text{En}(\text{Casa}, s) \wedge \text{Tener}(\text{leche}, s) \wedge \text{Tener}(\text{cambures}, s)$
- **Operador:** describe acciones que se pueden hacer
 $\text{ir}(\text{sitio}), \text{comprar}(\text{objeto})$
- **Plan:** secuencia de acciones
 $[\text{ir}(\text{Supermercado}), \text{Comprar}(\text{leche}), \text{Comprar}(\text{cambures}), \text{ir}(\text{casa})]$

Operador

- **Componentes posibles para describir Operador:**

- Acción a realizar
- Precondición: situación actual de donde debe partir
- Efecto: resultado de la operación

Op(Acción(Ir(allá), Precondición((En(aquí), Camino(aquí, allá)), Efecto(En(allá) \wedge \neg En(aquí))

- **Ejemplo de instanciación:**

Situación Actual: En(casa), Camino(Casa, Supermercado)

Acción: Ir(supermercado)

Efecto: En(supermercado), \neg En(Casa)

Problema de Planificación

- Una forma de representación de Planes
 - Un Conjunto de Pasos (acciones)
 - Un conjunto de Restricciones: de orden o de asignaciones
 - Un conjunto de enlaces causales

- Ejemplo:

Plan(Pasos S1: Op(Acción:comenzar),
S2:Op(Acción:terminar),
Precondicion(ZapatoDerecho \wedge ZapatoIzquierdo)),
Orden(S1; S2)
Asignación {}
Enlaces {})

Problema de Planificación

- ¿Cómo crear Planes?

Objetivo: ZapatoDerecho \wedge ZapatoIzquierdo

Op(Acción:ZapatoDerecho, Precondicion(MediaDerecha),
Efecto(ZapatoDerecho))

Op(Acción:MediaDerecha, Efecto(MediaDerecha))

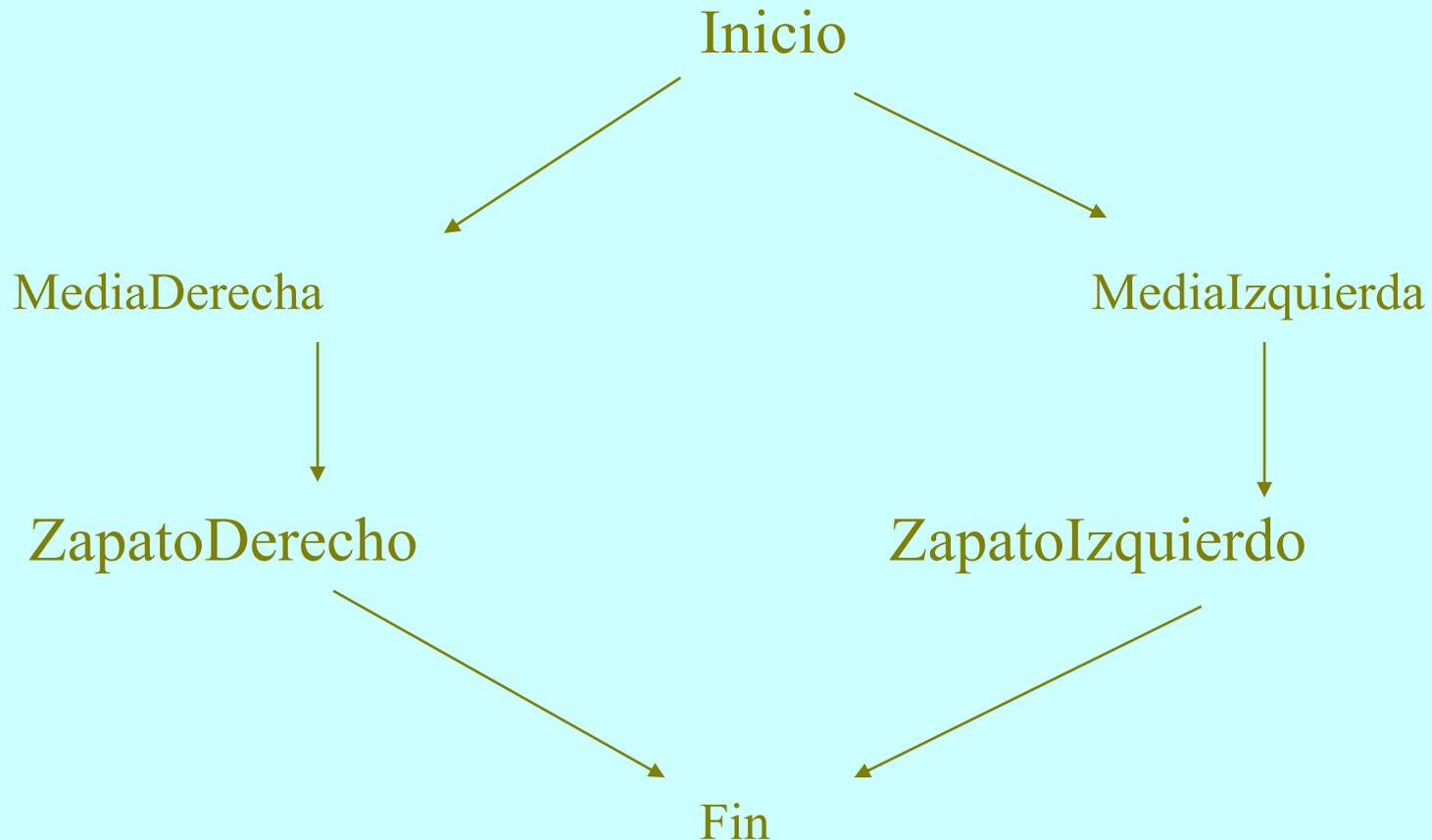
Op(Acción:ZapatoIzquierdo, Precondicion(MediaIzquierda),
Efecto(ZapatoIzquierdo))

Op(Acción:MediaIzquierda, Efecto(MediaIzquierda))

- **Solución** Orden Parcial
Orden Total

Problema de Planificación

Plan Orden Parcial:



Problema de Planificación

Plan Orden Total:



Inicio	Inicio	Inicio	Inicio	Inicio	Inicio
MediaDer	MediaDer	MediaDer	MediaIzq	MediaIzq	MediaIzq
MediaIzq	MediaIzq	ZapatoDer	MediaDer	MediaDer	ZapatoIzq
ZapatoDer	ZapatoIzq	MediaIzq	ZapatoDer	ZapatoIzq	MediaDer
ZapatoIzq	ZapatoDer	ZapatoIzq	ZapatoIzq	ZapatoDer	ZapatoDer

Problema de Planificación

Solución (plan) debe ser:

- **Completa**: cada precondition de un paso S_j es lograda por otra de otro paso S_i
 1. Si logra precondition c de S_j sii: $\text{orden}(S_i; S_j)$ y $c \in \text{efectos}(S_i)$
 2. $\neg \exists S_k$ tal que $\neg c \in \text{efectos}(S_k)$ y $\text{orden}(S_i; S_k; S_j)$
- **Consistente**: no hay contradicciones en las restricciones de orden o de asignación

Orden: $S_i; S_j$ $S_j; S_k$ $S_k; S_i$

Asignación: $v=A$ y $v=B$

¿Cómo crear Planes?

Inicio:

Op(Acción:Inicio, Efecto(En(casa) y Vende(tienda, ropa) y Vende(super, leche) y Vende(super, frutas))

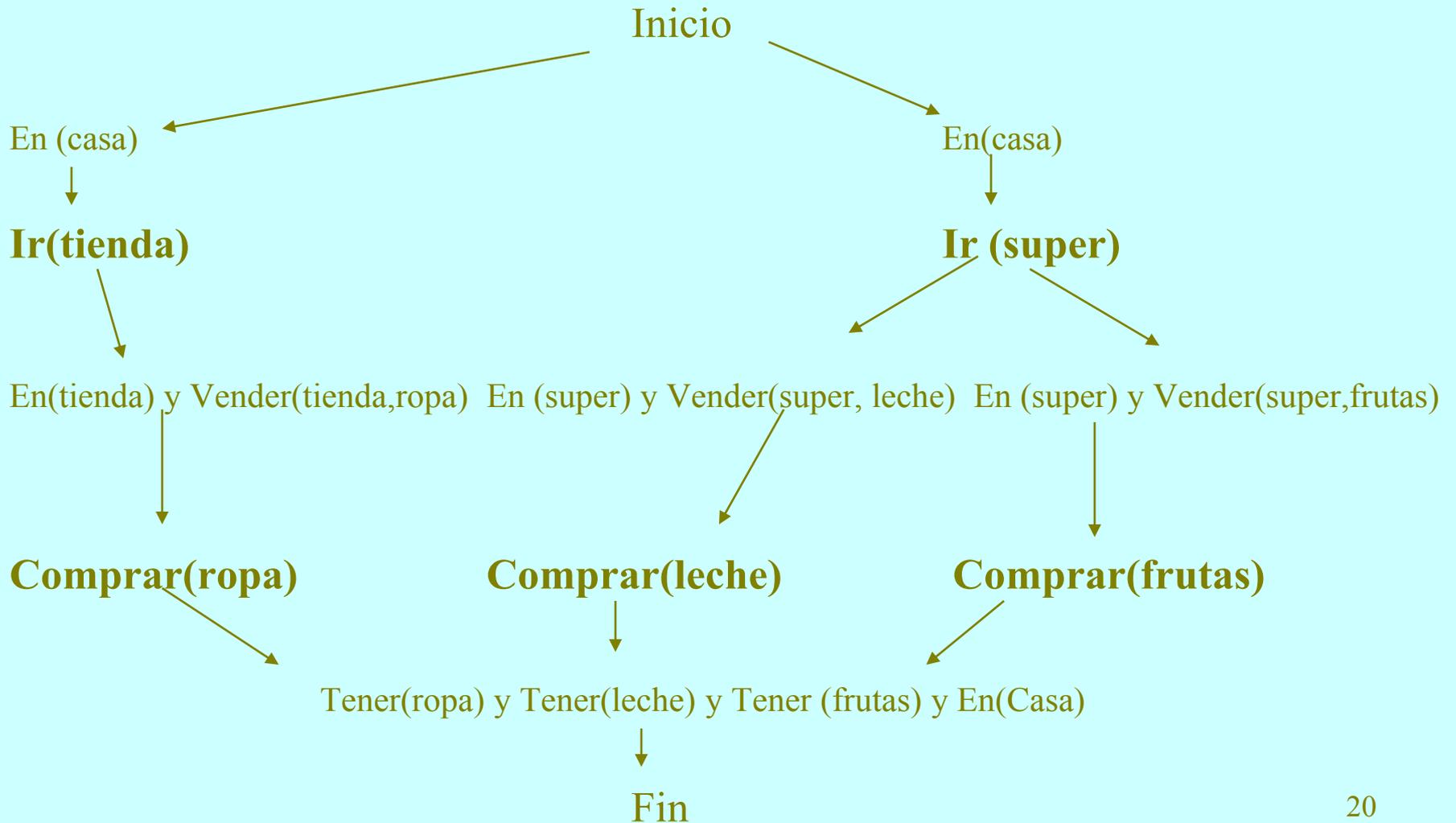
Fin (Objetivo):

Op(Acción:Fin, Precondicion(Tener(ropa) y Tener(leche) y Tener (frutas) y En(Casa))

Acciones posibles:

Op(Acción:Ir(allá), Precondicion(En (aquí)), Efecto(En(allá) y \neg En(aquí))
Op(Acción:Comprar(x), Precondicion(En (y) y Vende(y, x)), Efecto(Tener(x))

Ejemplo



Algoritmo de Planificación de Orden Parcial

POP

plan (minimo_plan(inicio,objetivo))

repita

 Si solución=plan entonces regresar plan

 <Snecesario, c>=selección-subobjetivo(plan)

 Escoger_operador(plan, operadores, Snecesario, c)

 Resolver_camino(plan)

Selección subobjetivo

Tomar un paso del plan (Snecesario) con una precondicion c no alcanzada

regresar Snecesario y c

Algoritmo de Planificación de Orden Parcial

Escoger Operador

Escoger paso Sanadir de los operadores del plan con c como efecto

Si no hay tal paso entonces falla

de lo contrario

añadir $\langle \text{Sanadir}, c, \text{Snecesario} \rangle$ a enlaces(plan)

añadir (Sanadir; Snecesario) a Orden(plan)

tal que: inicio; ...; Sanadir; Snecesario ;...; Fin en Orden(plan)

Si Sanadir es un nuevo paso escogido no existente en pasos(plan) entonces

añadir Sanadir a pasos(plan)

Resolver camino

por cada enlace $\langle S_i, c, S_j \rangle$ en Enlaces(plan)

por cada paso S_r en Orden(plan) que pueda amenazar $\langle S_i, c, S_j \rangle$ $c = \neg c'$ y $c' = \text{Efectos}(S_r)$

escoger entre Añadir $S_r; S_i$ a Orden(plan)

o eliminar $S_i; S_r$ en Orden(plan)

Si no se puede entonces consistencia plan falla

Ing. de Conocimiento para Planificación

1. Describir Problema
2. Decidir vocabulario de condiciones, operadores, y objetos
3. Codificar operadores (lo más genéricos posibles)
4. Codificar descripción del problema en específico (So, Fin, etc.)
5. probar al planificador

Operaciones Avanzadas

- **Mediciones y/o calculos:**

Op(Acción: comenzar,
Efecto: dinero<-(1000Bs) Y NivelGasolina<-5litros) condición inicial

...

Op(Acción: Comprar(x,tienda),
Efecto: Tener(x) Y dinero<-dinero-Precio(x,tienda)

...

Op(Acción: llenar(NivelGasolina)
Efecto:NivelGasolina<-40litros Y
dinero<-dinero - (PrecioUnidad(gasolina)x
(NivelGasolina - 5litros)

- **Descomposición:**

Op(Acción: Comprar(x)
Efecto: Tener(x) Y \neg Tener(Moneda)
Precond: 1. Vender (tienda, x) Y
 2. En (tienda) Y
 3. Tener (moneda)

Operaciones Avanzadas

- **Efectos Condicionales**

Op(Acción: Mover (b, x y)

Precond: En (b, x) Y Limpio (y)

Efecto: En(b, y) Y Limpio(x) *cuando* x,y=mesa

- **Cuantificador Universal:**

Op(Acción: CambiarContenidoBolso(x, y)

Precond: Bolso(x) y Bolso(y) y ContenidoEn(x)

Efecto: ContenidoEn(y) Y \neg ContenidoEn(x) Y

$(\forall i \text{ item}(i) \Rightarrow (\text{En}(i,y) \text{ Y } \neg \text{En}(i, x)))$ *cuando* Dentro (i, bolso(y)))

Operaciones Avanzadas: Parte Condicional

Op(A: quitar(x), P: En(x), E: fuera(x))

Op(A: poner(x), P: fuera(x), E: En(x))

Op(A: inflar(x), P: Intacto(x), desinflado(x), E: inflado(x))

Objetivo: En(x) Y Inflado(x)

Condición Inicial:

Inflado(rep) Y intacto(rep) Y fuera(rep) Y en(caucho) Y desinflado(caucho)

Plan: [quitar(caucho), poner(rep)]

Extender plan con:

If (<condition>, <thenpart>, <elsepart>) (*)

Ejemplo: If (Intacto(caucho), [Inflar(caucho)], [quitar(caucho), poner(rep)])

Otra Operación: Op(ChequearCaucho(x), P: Caucho(x), E: Intacto(x)?)

Op (Acción:... Precond: cond1 cuando estado1; cond2 cuando estado2

Efecto: efecto1 cuando estado1 ; efecto 2 cuando ...

Un Agente de Planificación con Condición (*)

Indicar(BC,sentencia_percepción (percibido,t))

Actual <- Estado_Descripto(BC, t)

Si P=NoPLan entonces

O<- Preguntar(BC,consulta_objetivo(t))

P<-POP(actual, O, BC)

Si P=NoPlan Entonces Acción<-NoPosible

de lo contrario

si no es una condición (if)

Acción<-Primero(P)

de lo contrario

Si preguntar(BC,partecondicion(condición)=V entonces

Acción<-parteinicio(condición)

de lo contrario Acción<-partefin(condición)

P<-resto(P)

Indicar(BC, relación_Acción_sentencia (Acción,t))

} (*)

Un Agente de Planificación con Replanificación

Indicar(BC, sentencia_percepción(percebido,t))

Actual <- Estado_Descrito(BC, t)

Si P=NoPLan entonces

O<- Preguntar(BC, consulta_objetivo(t))

P<-planificador(actual, O, BC)

Si P=NoPlan Entonces Acción<-NoPosible

De lo contrario

Si Precondición(P) no es verdad en BC

P'<- Escoger-Mejor-Continuación(actual)

P<-actualizar (planificador(actual, Precondición(P'), BC)) **(replanif)**

Acción<-Primero(P)

P<-resto(P)

Algoritmos de Planificación

La planificación implica una búsqueda compleja:

- Los operadores alternativos para lograr un objetivo
- Múltiples objetivos que interactúan

Solidez

- Un algoritmo de planificación es sólido si todas las soluciones son planes legales

Completitud

- Un algoritmo de planificación es completa si una solución se puede encontrar cada vez que existe una

Optimalidad

- Un algoritmo de planificación es óptima si se maximiza una predefinido₃₀ medida de la calidad del plan

Librerías de Planificación de Agentes

- **Prodigy Planner** <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/prodigy/Web/prodigy-home.html>
- **GPS**
- **FF**
- **FD**
- **LAMA**
- **Graphplan**
 - PLPLAN: Implementación en Java de GraphPlan
 - NPlanner: Implementación en .NET de GraphPlan
 - Emplan y JavaGP: Implementación en C++ y Java

Graphplan

<http://www.cs.cmu.edu/~avrim/graphplan.html>

- Búsqueda hacia adelante combinada con la búsqueda hacia atrás.
- Construye un grafo de la planificación
- Dos etapas:
 - Extender: Un paso de tiempo en el grafo de planificación.
 - Buscar: un plan válido en el grafo de planificación.
- Graphplan encuentra un plan o prueba que ningún plan existe

Graphplan

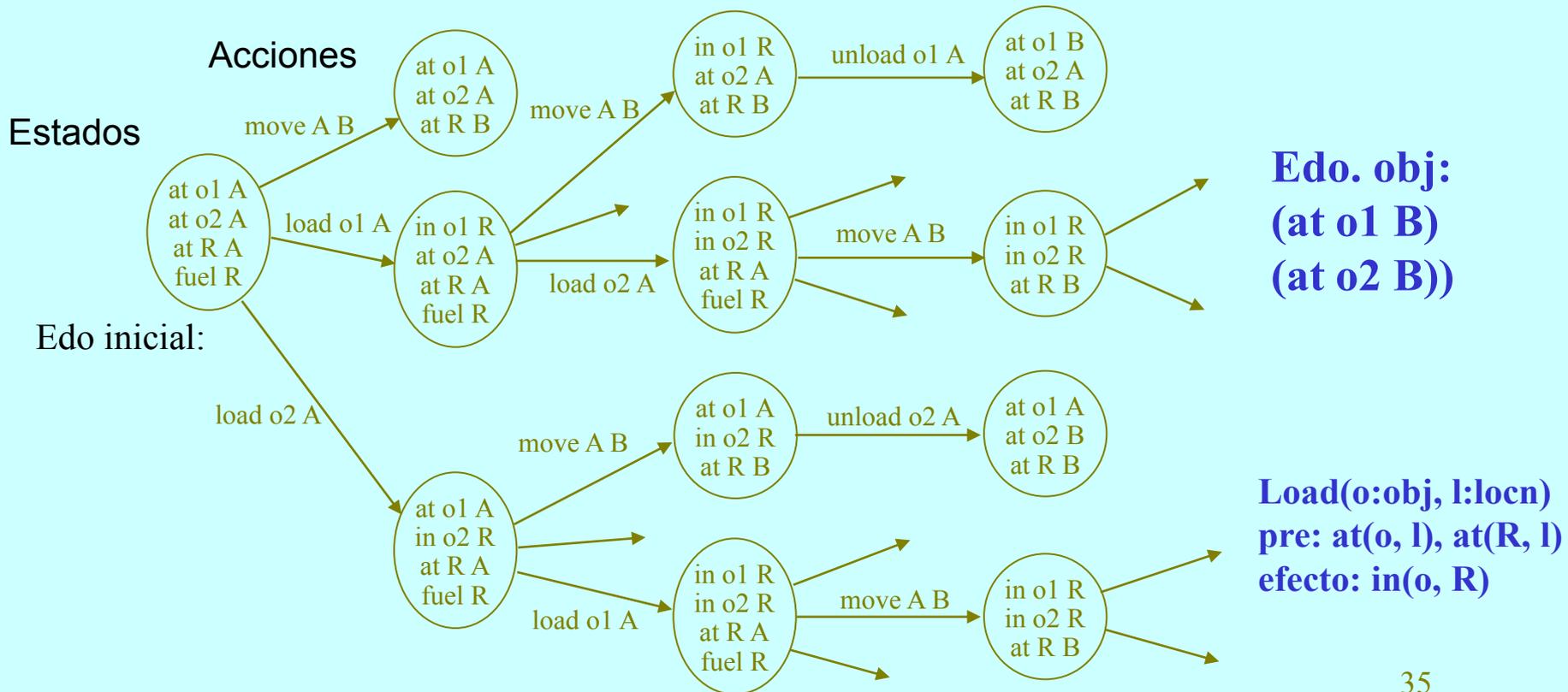
- El algoritmo toma como entrada un problema de planificación expresado en STRIPS y produce, de ser posible, una secuencia de operaciones para llegar a un estado final.
- En un *grafo de planificación*:
 - los nodos son las acciones y hechos atómicos, dispuestos en niveles alternos,
 - y las aristas son de dos tipos:
 - de un hecho atómico a las acciones de las cuales es una condición,
 - de una acción a los hechos atómicos donde se convierte en verdadero o falso.

Graphplan

- El primer nivel contiene hechos atómicos verdaderos que identifican el estado inicial.
- También se mantienen las listas de hechos incompatibles que no pueden ser verdaderos al mismo tiempo y acciones incompatibles que no se pueden ejecutar en conjunto.
- El algoritmo **extiende** iterativamente el grafo de planificación, probando que no hay soluciones de longitud $i-1$ antes de buscar planes de longitud i por encadenamiento hacia atrás:
 - suponiendo que los objetivos son verdaderos, el algoritmo de Graphplan **busca** las acciones y estados previos desde el cual el objetivo puede ser alcanzado, **podando** muchos de ellos mientras sea posible gracias a la información incompatible.

Graphplan

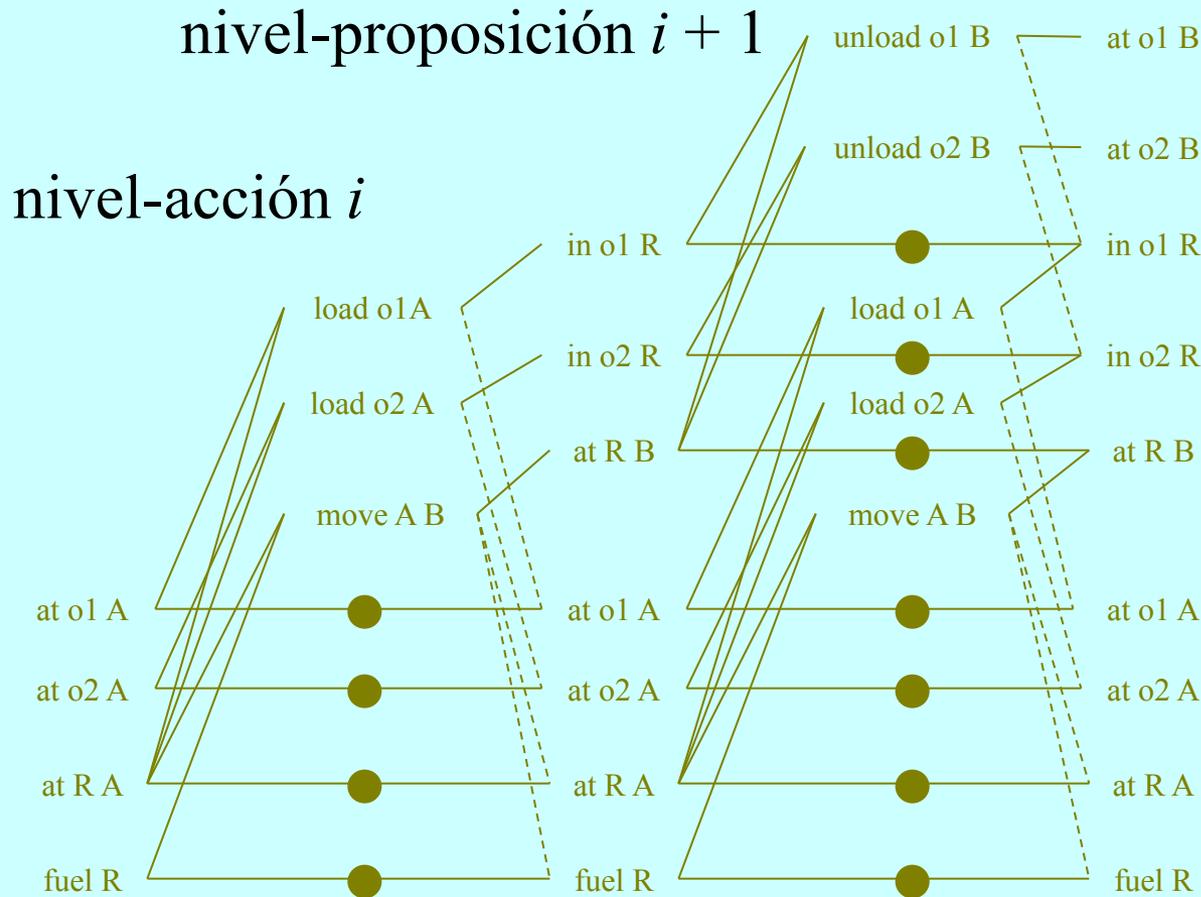
- Accesibilidad de Estado - "hasta" el objetivo
 - Puede encontrar todas las metas alcanzables desde el estado inicial
 - Exponencial en tiempo y memoria



Extender grafo de planificación

- Crear un nivel-acción i :
 - Agregar c/operador instanciado, para los cuales sus precondiciones estan presentes en el nivel-proposición i
- Crear un nivel-proposición $i + 1$:
 - Agregar todos los efectos de las acciones insertadas en el nivel-acción i

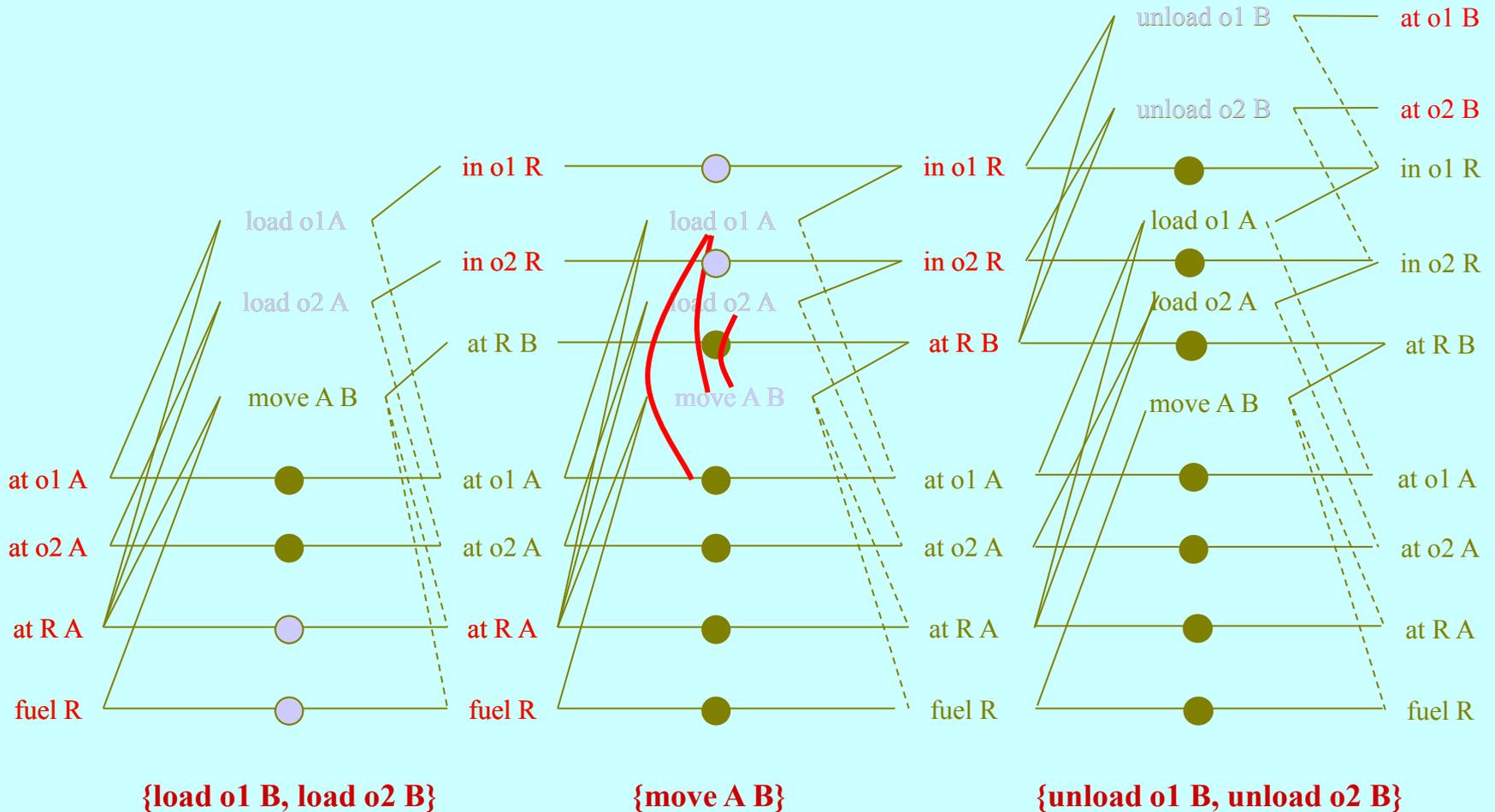
Extender grafo de planificación



Buscar en el grafo de planificación

- Enfoque de encadenamiento hacia atrás nivel por nivel
 - Dado un conjunto de metas en el tiempo t , identificar todos los conjuntos de acciones en el tiempo $t - 1$ que alcanzan esas metas. Las condiciones previas de estas acciones son las nuevas metas para $t - 1$.
 - Proceso recursivo
 - Por cada objetivo en el tiempo t en algún orden arbitrario:
Seleccionar acción en el tiempo $t - 1$ que logra ese objetivo.
 - Hacer esto de forma recursiva a todos los objetivos en el tiempo $t - 1$, y no añadir nuevas acciones, sino utilizar las ya seleccionadas para otro objetivo, si es posible.
 - Si la recursividad fracasa, seleccione una acción diferente.
- El nuevo conjunto objetivo es el conjunto de todas las condiciones previas de las acciones seleccionadas

Buscar en el grafo de planificación



Problemas abiertos

la cognición, la planificación y la ejecución del plan deben todos ser intercalados. El curso de c/u afecta al de los otros.

- Un agente racional debe realizar todas estas tareas en paralelo

En IA a veces se separa la planificación y la programación

- Primero se adopta un plan, y luego se construyen un programa para cuándo se ejecute el plan
- Esto parece inadecuada, incluso en los humanos.

Otras Técnicas de Planificación en SMA

- Planificación Centralizada para Planes Distribuidos
- Planificación Distribuida para Planes Centralizados
- Planificación Distribuida para Planes Distribuidos

Planificación Centralizada para Planes Distribuidos

- Esquema:
 - Planificación en orden parcial que puede ser ejecutada en paralelo (diferentes agentes hacen ramas diferentes).
 - Se debe garantizar no condiciones de dependencia.
- Algoritmo:
 1. realización de un plan general (el planificador)
 2. identificación ramificaciones que se pueden realizar en paralelo (POP).
 3. con ellas descomponer el plan en subplanes (ramificaciones)
 4. insertar acciones de sincronización entre los subplanes
 5. asignar subplanes a los agentes (asignación (estática o dinámica) a los agentes)

Planificación Distribuida para Planes Centralizados

Formular planes requiere colaboración de especialistas

1. Distribuir la tarea de construir planes entre los especialistas
2. Mezclar los planes obtenidos en paralelos verificando la coherencia entre ellos

Debe haber una coordinación centralizada para unir los planes parciales

- CADA AGENTE HACE SU PLAN PARCIAL
- EL COORDINADOR FUSIONA LOS PLANES (RELACION ENTRE LAS ACCIONES)

Planificación Distribuida para Planes Distribuidos

- Caso General
 1. Asignar objetivos a cada agente
 2. C/u formula planes locales
 3. C/u ejecuta sus planes
- Problemas
 - Resolución de Conflictos: negociación
 - Suposiciones de tiempo de ejecución de tareas

Planificación Distribuida para Planes Distribuidos

- Otro Algoritmo:
 1. Intercambiar entre agentes descripción de los planes
 2. Aprobar planes sin conflictos (si todos lo cumplen, resuelto el problema)
 3. Refinar planes con conflictos (cambios en los planes locales para eliminarlos)
 4. Ejecutar localmente planes sin conflicto

Proyecto

- Planificador general (cuerpo principal determinista)
 - Organizar las tareas que debe realizar el agente
 - Definir secuencia de realización de las tareas
- Planificador como tarea (dinámico)
 - Definir problema de planificación
 - Describir vocabulario de condiciones, operadores, y objetos
 - Establecer planificador (usar Graphplan)