

# Inteligencia Artificial

## Planificación: Acciones

Facultad de Informática  
Universidad Nacional del Comahue

Segundo Cuatrimestre

# Contenidos

## ¿Por qué planning?

Búsqueda Vs.Planning

## Representación del Conocimiento y Acciones

Representación de los diferentes elementos en el Planning

## Problemas con la Representación de Acciones

Qualification Problem

Ramification Problem

Frame Problem

## Semántica

Encontrando la solución

## Lenguajes de Representación

STRIPS: Stanford Research Institute Problem Solver

Situation Calculus

Event Calculus

## Bibliografía

## Desafíos

# ¿Qué vimos?

- ▶ Definiciones de IA (pensamiento vs.comportamiento - humanidad vs. racionalidad).
- ▶ Agentes basados en Conocimiento: RRS
- ▶ Búsqueda Exhaustiva: Búsqueda Ciega, Heurística
- ▶ Búsqueda Adversaria
- ▶ Problemas de Satisfacción de Restricciones
- ▶ Razonamiento No Monotónico

HOY

Planning: Acciones

# ¿Qué vimos?

- ▶ Definiciones de IA (pensamiento vs.comportamiento - humanidad vs. racionalidad).
- ▶ Agentes basados en Conocimiento: RRS
- ▶ Búsqueda Exhaustiva: Búsqueda Ciega, Heurística
- ▶ Búsqueda Adversaria
- ▶ Problemas de Satisfacción de Restricciones
- ▶ Razonamiento No Monotónico

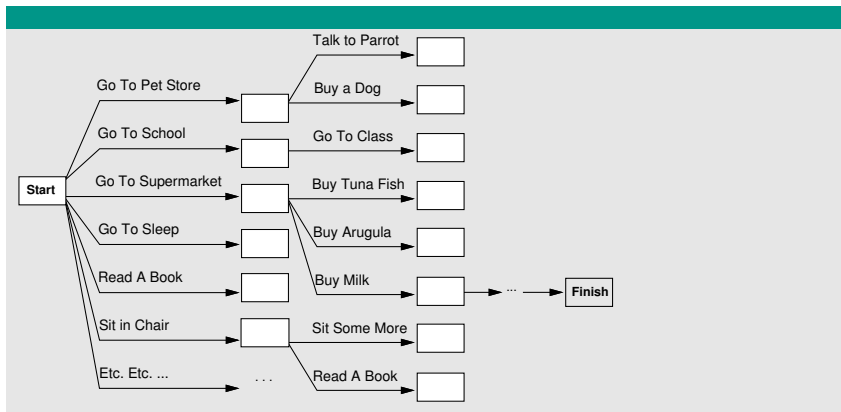
HOY

Planning: Acciones

# Búsqueda Vs. Planning: Acciones Irrelevantes

Consideremos la tarea de *obtener leche, bananas y una licuadora*

Los algoritmos estándar de búsqueda parecen fallar:



Aún si tuviéramos una heurística.

# Búsqueda Vs. Planning

Con Planning podemos :

- ▶ Abrir la representación de las acciones y las metas para permitir la selección.
- ▶ Dividir y conquistar descomponiendo el problema en submetas.
- ▶ Relajar los requerimientos de construcción secuencial de las soluciones.

# Búsqueda Vs. Planning

Con Planning podemos :

- ▶ Abrir la representación de las acciones y las metas para permitir la selección.
- ▶ Dividir y conquistar descomponiendo el problema en submetas.
- ▶ Relajar los requerimientos de construcción secuencial de las soluciones.

# Búsqueda Vs. Planning

Con Planning podemos :

- ▶ Abrir la representación de las acciones y las metas para permitir la selección.
- ▶ Dividir y conquistar descomponiendo el problema en submetas.
- ▶ Relajar los requerimientos de construcción secuencial de las soluciones.



# Búsqueda Vs. Planning

	Búsqueda	Planning
<b>Estados</b>	Estructura de Datos	Sentencias Lógicas
<b>Acciones</b>	Código	Precondiciones/Efecto
<b>Meta</b>	Estructura de Datos - Código	Sentencias Lógicas (conjunción)
<b>Plan</b>	Secuencia desde $S_0$	Secuencia con restricciones sobre las acciones

# Lenguaje de Representación

## Utilizaremos la Estructura Lógica del Problema

### Representación de Estados

El mundo es una conjunción de literales.

### Representación de Metas

Es un estado particular, representado por una conjunción de literales.

### Definición

Un estado proposicional  $s$  satisface una meta  $g$  si  $s$  contiene todos los átomos en  $g$  (y posiblemente otros).

# Lenguaje de Representación

Utilizaremos la Estructura Lógica del Problema

## Representación de Estados

El mundo es una conjunción de literales.

## Representación de Metas

Es un estado particular, representado por una conjunción de literales.

## Definición

Un estado proposicional  $s$  satisface una meta  $g$  si  $s$  contiene todos los átomos en  $g$  (y posiblemente otros).

# Lenguaje de Representación

Utilizaremos la Estructura Lógica del Problema

## Representación de Estados

El mundo es una conjunción de literales.

## Representación de Metas

Es un estado particular, representado por una conjunción de literales.

## Definición

Un estado proposicional  $s$  satisface una meta  $g$  si  $s$  contiene todos los átomos en  $g$  (y posiblemente otros).

# Lenguaje de Representación

Utilizaremos la Estructura Lógica del Problema

## Representación de Estados

El mundo es una conjunción de literales.

## Representación de Metas

Es un estado particular, representado por una conjunción de literales.

## Definición

Un estado proposicional  $s$  satisface una meta  $g$  si  $s$  contiene todos los átomos en  $g$  (y posiblemente otros).

# Representación de las Acciones

Las especificamos en términos de Precondiciones/Efectos.

## PRECONDICIONES

Literales que deben ocurrir antes de que la acción sea ejecutada.

## ACCIÓN

## EFFECTOS

Aquello que será consecuencia de la ejecución de la acción.

# Representación de las Acciones

Las especificamos en términos de Precondiciones/Efectos.

## PRECONDICIONES

Literales que deben ocurrir antes de que la acción sea ejecutada.

## ACCIÓN

## EFFECTOS

Aquello que será consecuencia de la ejecución de la acción.

# Representación de las Acciones

Las especificamos en términos de Precondiciones/Efectos.

## PRECONDICIONES

Literales que deben ocurrir antes de que la acción sea ejecutada.

## ACCIÓN

## EFFECTOS

Aquello que será consecuencia de la ejecución de la acción.



# Representación de las Acciones

## Example

Las especificamos en términos de Precondiciones/Efectos.

### PRECONDICIONES

$at(p, from) \wedge$   
 $plane(p) \wedge$   
 $airport(from) \wedge$   
 $airport(to)$

**$fly(p, from, to)$**

### EFFECTOS

$\neg at(p, from) \wedge$   
 $at(p, to)$

# Representación de las Acciones

## Example

Las especificamos en términos de Precondiciones/Efectos.

### PRECONDICIONES

$at(p, from) \wedge$   
 $plane(p) \wedge$   
 $airport(from) \wedge$   
 $airport(to)$

$fly(p, from, to)$

### EFFECTOS

$\neg at(p, from) \wedge$   
 $at(p, to)$

# Representación de las Acciones

## Example

Las especificamos en términos de Precondiciones/Efectos.

### PRECONDICIONES

$at(p, from) \wedge$   
 $plane(p) \wedge$   
 $airport(from) \wedge$   
 $airport(to)$

$fly(p, from, to)$

### EFFECTOS

$\neg at(p, from) \wedge$   
 $at(p, to)$

# Qualification Problem

¿Cuáles son las precondiciones para que una acción tenga éxito?

## PRECONDICIONES

- ▶ tengo las llaves
- ▶ tiene nafta
- ▶ tiene batería cargada
- ▶ tiene el caño de escape libre
- ▶ ...

arrancar el auto

# Qualification Problem

¿Cuáles son las precondiciones para que una acción tenga éxito?

## PRECONDICIONES

- ▶ tengo las llaves
- ▶ tiene nafta
- ▶ tiene batería cargada
- ▶ tiene el caño de escape libre
- ▶ ...

arrancar el auto

# Qualification Problem

¿Cuáles son las precondiciones para que una acción tenga éxito?

## PRECONDICIONES

- ▶ tengo las llaves
- ▶ tiene nafta
- ▶ tiene batería cargada
- ▶ tiene el caño de escape libre
- ▶ ...

arrancar el auto

# Qualification Problem

¿Cuáles son las precondiciones para que una acción tenga éxito?

## PRECONDICIONES

- ▶ tengo las llaves
- ▶ tiene nafta
- ▶ tiene batería cargada
- ▶ tiene el caño de escape libre
- ▶ ...

arrancar el auto

# Qualification Problem

¿Cuáles son las precondiciones para que una acción tenga éxito?

## PRECONDICIONES

- ▶ tengo las llaves
- ▶ tiene nafta
- ▶ tiene batería cargada
- ▶ tiene el caño de escape libre

arrancar el auto



# Qualification Problem

¿Cuáles son las precondiciones para que una acción tenga éxito?

## PRECONDICIONES

- ▶ tengo las llaves
- ▶ tiene nafta
- ▶ tiene batería cargada
- ▶ tiene el caño de escape libre
- ▶ ...

arrancar el auto

# Qualification Problem

El problema de las circunstancias no representadas que podrían **causar que una acción falle** se conoce como **Qualification Problem**.

# Ramification Problem

¿Cuáles son los efectos de una acción?

vuela el avión A847 de Nqn a Bs As

EFFECTOS si no hay paro :)

- ▶ A847 no está en Neuquén
- ▶ A847 está en Buenos Aires
- ▶ los pasajeros no están en Neuquén
- ▶ los pasajeros están en Buenos Aires
- ▶ el equipaje que iba en el avión no está en Nqn
- ▶ el equipaje que iba en el avión está en BsAs
- ▶ ...

# Ramification Problem

¿Cuáles son los efectos de una acción?

vuela el avión A847 de Nqn a Bs As

## EFFECTOS si no hay paro :)

- ▶ A847 no está en Neuquén
- ▶ A847 está en Buenos Aires
- ▶ los pasajeros no están en Neuquén
- ▶ los pasajeros están en Buenos Aires
- ▶ el equipaje que iba en el avión no está en Nqn
- ▶ el equipaje que iba en el avión está en BsAs
- ▶ ...

# Ramification Problem

¿Cuáles son los efectos de una acción?

vuela el avión A847 de Nqn a Bs As

## EFFECTOS si no hay paro :)

- ▶ A847 no está en Neuquén
- ▶ A847 está en Buenos Aires
- ▶ los pasajeros no están en Neuquén
- ▶ los pasajeros están en Buenos Aires
- ▶ el equipaje que iba en el avión no está en Nqn
- ▶ el equipaje que iba en el avión está en BsAs
- ▶ ...

# Ramification Problem

¿Cuáles son los efectos de una acción?

vuela el avión A847 de Nqn a Bs As

## EFFECTOS si no hay paro :)

- ▶ A847 no está en Neuquén
- ▶ A847 está en Buenos Aires
- ▶ los pasajeros no están en Neuquén
- ▶ los pasajeros están en Buenos Aires
- ▶ el equipaje que iba en el avión no está en Nqn
- ▶ el equipaje que iba en el avión está en BsAs
- ▶ ...

# Ramification Problem

¿Cuáles son los efectos de una acción?

vuela el avión A847 de Nqn a Bs As

## EFFECTOS si no hay paro :)

- ▶ A847 no está en Neuquén
- ▶ A847 está en Buenos Aires
- ▶ los pasajeros no están en Neuquén
- ▶ los pasajeros están en Buenos Aires
- ▶ el equipaje que iba en el avión no está en Nqn
- ▶ el equipaje que iba en el avión está en BsAs
- ▶ ...

# Ramification Problem

¿Cuáles son los efectos de una acción?

vuela el avión A847 de Nqn a Bs As

## EFFECTOS si no hay paro :)

- ▶ A847 no está en Neuquén
- ▶ A847 está en Buenos Aires
- ▶ los pasajeros no están en Neuquén
- ▶ los pasajeros están en Buenos Aires
- ▶ el equipaje que iba en el avión no está en Nqn
- ▶ el equipaje que iba en el avión está en BsAs
- ▶ ...



# Ramification Problem

¿Cuáles son los efectos de una acción?

vuela el avión A847 de Nqn a Bs As

## EFFECTOS si no hay paro :)

- ▶ A847 no está en Neuquén
- ▶ A847 está en Buenos Aires
- ▶ los pasajeros no están en Neuquén
- ▶ los pasajeros están en Buenos Aires
- ▶ el equipaje que iba en el avión no está en Nqn
- ▶ el equipaje que iba en el avión está en BsAs

▶ ...

# Ramification Problem

¿Cuáles son los efectos de una acción?

vuela el avión A847 de Nqn a Bs As

## EFFECTOS si no hay paro :)

- ▶ A847 no está en Neuquén
- ▶ A847 está en Buenos Aires
- ▶ los pasajeros no están en Neuquén
- ▶ los pasajeros están en Buenos Aires
- ▶ el equipaje que iba en el avión no está en Nqn
- ▶ el equipaje que iba en el avión está en BsAs
- ▶ ...

# Ramification Problem

El problema de las circunstancias no representadas que son efecto de la ejecución de una acción se conoce como Problema de la Ramificación.

# Frame Problem

¿Qué elementos permanecen sin cambios cuando una acción es ejecutada?

Antes de la  
Acción

- ▶ color auto verde
- ▶ modelo del auto 2003
- ▶ marca del auto Fiat
- ▶ ...

arrancar el auto

Después de la  
Acción

- ▶ color auto verde
- ▶ modelo del auto 2003
- ▶ marca del auto Fiat
- ▶ ...

# Frame Problem

¿Qué elementos permanecen sin cambios cuando una acción es ejecutada?

## Antes de la Acción

- ▶ color auto verde
- ▶ modelo del auto 2003
- ▶ marca del auto Fiat
- ▶ ...

arrancar el auto

## Después de la Acción

- ▶ color auto verde
- ▶ modelo del auto 2003
- ▶ marca del auto Fiat
- ▶ ...

# Frame Problem

¿Qué elementos permanecen sin cambios cuando una acción es ejecutada?

## Antes de la Acción

- ▶ color auto verde
- ▶ modelo del auto 2003
- ▶ marca del auto Fiat
- ▶ ...

arrancar el auto

## Después de la Acción

- ▶ color auto verde
- ▶ modelo del auto 2003
- ▶ marca del auto Fiat
- ▶ ...

# Frame Problem

¿Qué elementos permanecen sin cambios cuando una acción es ejecutada?

## Antes de la Acción

- ▶ color auto verde
- ▶ modelo del auto 2003
- ▶ marca del auto Fiat

▶ ...

arrancar el auto

## Después de la Acción

- ▶ color auto verde
- ▶ modelo del auto 2003
- ▶ marca del auto Fiat

▶ ...

# Frame Problem

¿Qué elementos permanecen sin cambios cuando una acción es ejecutada?

## Antes de la Acción

- ▶ color auto verde
- ▶ modelo del auto 2003
- ▶ marca del auto Fiat
- ▶ ...

arrancar el auto

## Después de la Acción

- ▶ color auto verde
- ▶ modelo del auto 2003
- ▶ marca del auto Fiat
- ▶ ...



# Frame Problem

El problema de representar todas las circunstancias que **permanecen sin cambios** se conoce como **Problema del Frame**.

Necesitamos una solución eficiente a este problema ya que en el mundo real casi todo permanece igual la mayoría del tiempo.

# Aplicación de una acción

Una acción es aplicable en cualquier estado que satisface las precondiciones.

## Example

El estado  $s =$

$\{at(A810, JN) \wedge at(A550, JFK) \wedge plane(A810) \wedge plane(A550) \wedge airport(JN) \wedge airport(JFK)\}$

satisface la precondición:

$$at(p, from) \wedge plane(p) \wedge airport(from) \wedge airport(to)$$

con  $\theta = \{p/A810, from/JN, to/JFK\}$ .

Luego la acción **fly(A810,JN,JFK)** es aplicable

# Solución

Una secuencia de acciones que cuando es ejecutada desde el estado inicial resulta en un estado que satisface la meta.

# STRIPS

## Representación de Estados

El mundo es una conjunción de literales proposicionales:  
ground y function-free. Utilizamos CWA

### Example

`at(agent1, neuquen)`

# STRIPS

## Representación de Estados

El mundo es una conjunción de literales proposicionales:  
ground y function-free. Utilizamos CWA

## Example

at(agent1, neuquen)

# Representación de la Acciones

## Esquema de Acción

Representa un número diferente de acciones que pueden ser derivadas instanciando las variables.

## Acción

STRIPS la representa con tres listas:

- ▶ Lista de Precondiciones
- ▶ Lista de Agregados (Add List)
- ▶ Lista de Borrados (Delete List)

# Representación de la Acciones

## Esquema de Acción

Representa un número diferente de acciones que pueden ser derivadas instanciando las variables.

## Acción

STRIPS la representa con tres listas:

- ▶ Lista de Precondiciones
- ▶ Lista de Agregados (Add List)
- ▶ Lista de Borrados (Delete List)

# Representación de la Acciones

## Esquema de Acción

Representa un número diferente de acciones que pueden ser derivadas instanciando las variables.

## Acción

STRIPS la representa con tres listas:

- ▶ Lista de Precondiciones
- ▶ Lista de Agregados (Add List)
- ▶ Lista de Borrados (Delete List)



# Representación de la Acciones

## Lista de Precondiciones

Literales que deben cumplirse para que la acción pueda ser exitosa.

## Lista de Agregados

Literales positivos que agregamos para generar el nuevo estado.

## Lista de Borrados

Literales positivos que quitamos porque dejaron de tener efecto en el nuevo estado.

# Representación de la Acciones

## Lista de Precondiciones

Literales que deben cumplirse para que la acción pueda ser exitosa.

## Lista de Agregados

Literales positivos que agregamos para generar el nuevo estado.

## Lista de Borrados

Literales positivos que quitamos porque dejaron de tener efecto en el nuevo estado.

# Representación de la Acciones

## Lista de Precondiciones

Literales que deben cumplirse para que la acción pueda ser exitosa.

## Lista de Agregados

Literales positivos que agregamos para generar el nuevo estado.

## Lista de Borrados

Literales positivos que quitamos porque dejaron de tener efecto en el nuevo estado.

# Representación de las Acciones

## Example

### PRECONDICIONES

$\text{at}(p, \text{from}) \wedge$   
 $\text{plane}(p) \wedge$   
 $\text{airport}(\text{from}) \wedge$   
 $\text{airport}(\text{to})$

**$\text{fly}(p, \text{from}, \text{to})$**

### EFFECTOS

Add List={  $\text{at}(p, \text{to})$  }  
Delete List={  $\text{at}(p, \text{from})$  }

# Representación de las Acciones

## Example

### PRECONDICIONES

$\text{at}(p, \text{from}) \wedge$   
 $\text{plane}(p) \wedge$   
 $\text{airport}(\text{from}) \wedge$   
 $\text{airport}(\text{to})$

$\text{fly}(p, \text{from}, \text{to})$

### EFECTOS

Add List={  $\text{at}(p, \text{to})$  }  
Delete List={  $\text{at}(p, \text{from})$  }

# Representación de las Acciones

## Example

### PRECONDICIONES

$\text{at}(p, \text{from}) \wedge$   
 $\text{plane}(p) \wedge$   
 $\text{airport}(\text{from}) \wedge$   
 $\text{airport}(\text{to})$

$\text{fly}(p, \text{from}, \text{to})$

### EFFECTOS

Add List={  $\text{at}(p, \text{to})$  }  
Delete List={  $\text{at}(p, \text{from})$  }

# Aplicación de una acción

## Example

El estado

$$s = \{at(A810, JN) \wedge at(A550, JFK) \wedge plane(A810) \wedge plane(A550) \wedge airport(JN) \wedge airport(JFK)\}$$

satisface la precondition:

$$at(p, from) \wedge plane(p) \wedge airport(from) \wedge airport(to)$$

con  $\theta = \{p/A810, from/JN, to/JFK\}$ . Luego la acción **fly(A810,JN,JFK)** es aplicable.

## Solución

Nuevo estado  $s'$ :

$$s' = s - \{at(A810, JN)\} \cup \{at(A810, JFK)\}$$

# STRIPS y los Problemas de la Representación

- ▶ **Qualification Problem:** Lista de Precondiciones
- ▶ **Ramification Problem:** Lista de Agregados y Borrado. No alcanza para expresar todos los efectos. Ejemplo, no hay modo de representar que si el avión cambió de lugar su contenido también lo hizo, a menos que escribamos uno a uno estos efectos de la acción.
- ▶ **Frame Problem:** cada literal no mencionado en los efectos permanece sin cambios.

STRIPS es insuficientemente expresivo para algunos dominios reales.



# STRIPS y los Problemas de la Representación

- ▶ **Qualification Problem:** Lista de Precondiciones
- ▶ **Ramification Problem:** Lista de Agregados y Borrado. No alcanza para expresar todos los efectos. Ejemplo, no hay modo de representar que si el avión cambió de lugar su contenido también lo hizo, a menos que escribamos uno a uno estos efectos de la acción.
- ▶ **Frame Problem:** cada literal no mencionado en los efectos permanece sin cambios.

STRIPS es insuficientemente expresivo para algunos dominios reales.

# STRIPS y los Problemas de la Representación

- ▶ **Qualification Problem:** Lista de Precondiciones
- ▶ **Ramification Problem:** Lista de Agregados y Borrado. No alcanza para expresar todos los efectos. Ejemplo, no hay modo de representar que si el avión cambió de lugar su contenido también lo hizo, a menos que escribamos uno a uno estos efectos de la acción.
- ▶ **Frame Problem:** cada literal no mencionado en los efectos permanece sin cambios.

STRIPS es insuficientemente expresivo para algunos dominios reales.

# STRIPS y los Problemas de la Representación

- ▶ **Qualification Problem:** Lista de Precondiciones
- ▶ **Ramification Problem:** Lista de Agregados y Borrado. No alcanza para expresar todos los efectos. Ejemplo, no hay modo de representar que si el avión cambió de lugar su contenido también lo hizo, a menos que escribamos uno a uno estos efectos de la acción.
- ▶ **Frame Problem:** cada literal no mencionado en los efectos permanece sin cambios.

STRIPS es insuficientemente expresivo para algunos dominios reales.

# Action Description Language

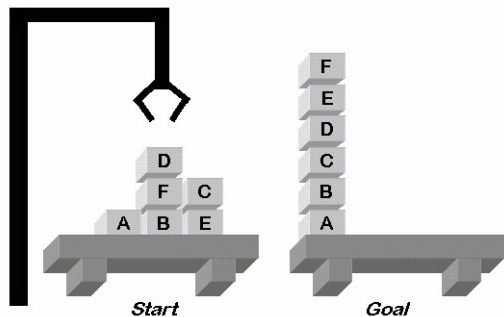
Lenguaje STRIPS	Lenguaje ADL
Sólo literales positivos en los estados: <i>poor</i> $\wedge$ <i>unknown</i>	Literales positivos y negativos en los estados: $\neg$ <i>rich</i> $\wedge$ $\neg$ <i>famous</i>
Closed World Assumption: Literales no mencionados son falsos	Open World Assumption Literales no mencionados son desconocidos
El efecto de $P \wedge \neg Q$ significa agregue $P$ y borre $Q$	El efecto de $P \wedge \neg Q$ significa agregue $P$ y $\neg Q$ y borre $\neg P$ y $Q$
Sólo se permiten literales ground en las metas: <i>rich</i> $\wedge$ <i>famous</i>	Se permiten variables cuantificadas en las metas: $\exists X \text{ at}(p1, X) \wedge \text{at}(p2, X)$
Las metas son conjunciones	Las metas permiten conjunciones y disyunciones
No soporta la igualdad	Predicados de igualdad ( $x = y$ ) predefinidos
No soporta tipos	Las variables pueden tener tipos ( $p:\text{Plane}$ ).

# Planning Domain Definition Language

Surge como combinación de STRIPS y ADL.

Lenguaje STRIPS	PDDL
Sólo literales positivos en los estados: <i>poor</i> $\wedge$ <i>unknown</i>	Literales positivos y negativos en los estados: $\neg$ <i>rich</i> $\wedge$ $\neg$ <i>famous</i>
Closed World Assumption:	Closed World Assumption
El efecto de $P \wedge \neg Q$ significa agregue $P$ y borre $Q$	El efecto de $P \wedge \neg Q$ significa agregue $P$ y $\neg Q$ y borre $\neg P$ y $Q$
Tiene dos listas Agregados/Borrados	Sólo tiene un conjunto de EFECTOS
Sólo se permiten literales ground en las metas: <i>rich</i> $\wedge$ <i>famous</i>	Se permiten variables cuantificadas en las metas: $\exists X \text{ at}(p1, X) \wedge \text{at}(p2, X)$
Las metas son conjunciones	Las metas permiten conjunciones y disyunciones
No soporta tipos	Las variables pueden tener tipos ( $p$ :Plane).

# Mundo de Bloques



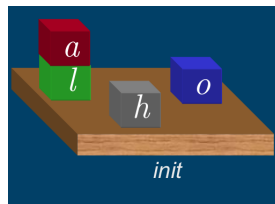
# Mundo de Bloques

## Mundo de Bloques: Definición de Dominio

### Acciones

**Mover** mueve un bloque que está sobre uno a otro

**Mover** mueve un bloque que está sobre la mesa sobre otro



### Relaciones para representar

- ▶ El color de un bloque
- ▶ que un bloque esté libre
- ▶ que un bloque está sobre la mesa
- ▶ que un bloque está sobre otro
- ▶ que un bloque esta (por) encima de otro

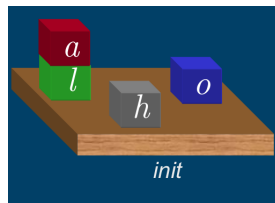
# Mundo de Bloques

## Mundo de Bloques: Definición de Dominio

### Acciones

**Mover** mueve un bloque que está sobre uno a otro

**Mover** mueve un bloque que está sobre la mesa sobre otro



### Relaciones para representar

- ▶ El color de un bloque } **relación estática**
- ▶ que un bloque esté libre } **primitiva** } rel. dinámica
- ▶ que un bloque está sobre la mesa } **primitiva** } rel. dinámica
- ▶ que un bloque está sobre otro } **primitiva** } rel. dinámica
- ▶ que un bloque esta (por) encima de otro } **relación derivada** } rel. dinámica



# STRIPS

- ▶ El tiempo se modela de acuerdo a la visión de **espacio de estados**, donde sólo un estado se representa en la lógica en un momento dado y las acciones son **externas** a la lógica.
- ▶ Para cada acción se establece cuándo puede llevarse a cabo y cuál es el efecto de dicha acción (es decir, qué relaciones primitivas son afectadas por la acción).
- ▶ Conjuntamente, se asume que todas las relac. primitivas no mencionadas en la descripción del efecto de la acción permanecen inalteradas luego de ser ejecutada (**suposición STRIPS**). De esta forma, se elude el **Problema del Marco**.
- ▶ La representación STRIPS se basa en la idea de que la ejecución de una acción solo afecta una pequeña porción de la descripción actual del mundo. Es decir, solo afecta unas pocas relaciones primitivas.

# Mundo de Bloques: Representación

`on(X,Y)` o `sobre(X,Y)`

Es verdadero cuando el bloque X está sobre Y, donde Y es otro bloque o la mesa.

`move(B,X,Y)`

Acción de mover el bloque B del tope de X a Y.

# Mundo de Bloques: Move(B,X,Y)

## Precondiciones en ADL

$\neg \exists X \text{On}(X, B)$

$\neg \exists X \text{On}(X, Y)$

$\text{On}(B, X)$

## Precondiciones en STRIPS

*Clear(B)*

*Clear(Y)*

*On(B, X)*

# Mundo de Bloques: Move(B,X,Y)

## Efectos en ADL

$\neg On(B, X)$   
 $On(B, Y)$

## Efectos en STRIPS

Lista de Agregados:  $\{Clear(X), On(B, Y)\}$   
Lista de Borrados:  $\{Clear(Y), On(B, X)\}$

# Mundo de Bloques: Move(B,X,Y)

## Efectos en ADL

$\neg On(B, X)$   
 $On(B, Y)$

## Efectos en STRIPS

Lista de Agregados:  $\{Clear(X), On(B, Y)\}$   
Lista de Borrados:  $\{Clear(Y), On(B, X)\}$

# Mundo de Bloques: Move(B,X,Y)

Situaciones especiales:

- ▶ Mover desde y a la mesa: la mesa permanece “clear”.
- ▶ Move(B,C,C)

# Mundo de Bloques: Move(B,X,Y)

Situaciones especiales:

- ▶ Mover desde y a la mesa: la mesa permanece “clear”.
- ▶ Move(B,C,C)

# Mundo de Bloques

- La representación de las acciones es **externa** a la lógica.
- Descripción de la acción **apilar X sobre Y**

## Precondiciones:

enMesa(X)

libre(X)

libre(Y)

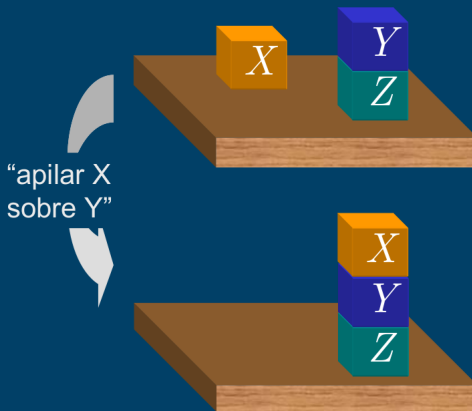
## Del-list:

enMesa(X)

libre(Y)

## Add-list:

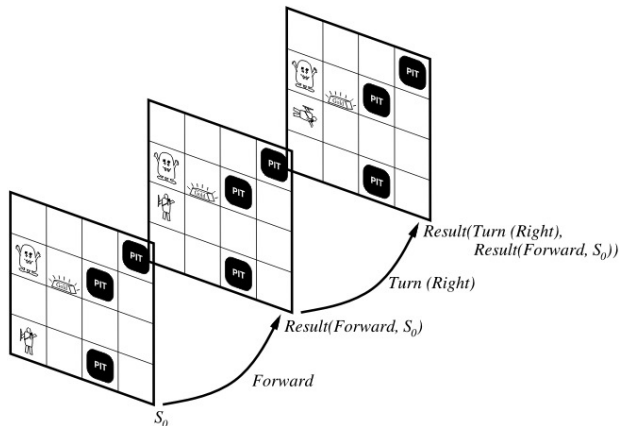
sobre(X,Y)





# Situation Calculus

En vez de concentrarnos en tiempos nos concentramos en situaciones:



# Situation Calculus

- ▶ El tiempo se modela de acuerdo a la visión de **espacio de estados**, donde las acciones y los estados (o situaciones) se reifican.
- ▶ Al decir que las acciones y estados se reifican, nos referimos concretamente a que se utilizan **términos** lógicos para **denotar/ nombrar/ referirse a estados y acciones**.
- ▶ De esta manera es posible representar, **en** la lógica, **senencias que hablan de/se refieren a acciones y estados** particulares.

# Situation Calculus: Representación

Representamos a las situaciones como:

- **Situación Inicial:**  $S_0$  o *init*
- Situaciones que resulta de realizar la acción A a partir de la situación S, a condición de que la acción A sea aplicable a partir de S:  $do(A, S)$

$$S_0 \xrightarrow{A_0} \frac{result(A_0, S_0)}{do(A_0, S_0)} \xrightarrow{A_1} \frac{result(A_1, result(A_0, S_0))}{do(A_1, do(A_0, S_0))}$$

# Situation Calculus: Representación

Representamos a las situaciones como:

- **Situación Inicial:**  $S_0$  o *init*
- Situaciones que resulta de realizar la acción  $A$  a partir de la situación  $S$ , a condición de que la acción  $A$  sea aplicable a partir de  $S$ :  $do(A, S)$

$$S_0 \xrightarrow{A_0} \frac{result(A_0, s_0)}{do(A_0, s_0)} \xrightarrow{A_1} \frac{result(A_1, result(A_0, s_0))}{do(A_1, do(A_0, s_0))}$$

# Situation Calculus: Representación

## Predicados Atemporales

Son predicados que permanecen **invariables** en todas las situaciones. Ejemplo: agente(bond)

## Fluentes

Son predicados que **varían** desde alguna situación a la siguiente.

# Situation Calculus: Representación

## Predicados Atemporales

Son predicados que permanecen **invariables** en todas las situaciones. Ejemplo: agente(bond)

## Fluentes

Son predicados que **varían** desde alguna situación a la siguiente.

# Situation Calculus: Representación

## Representación de un Estado

Fórmulas lógicas con un parámetro más que indica la situación en la que es verdadera. Ejemplo:

*at(bond, [1, 1],  $S_0$ )*

*holding(bond, gun, do(get(gun), do(go([1, 1], [2, 2]),  $S_0$ )))*

*agente(bond)*

# Situation Calculus

## Representación Lógica del cálculo situacional

### Representación Lógica

Relaciones Estáticas

Relaciones Dinámicas  
válidas en *init*

Descripción de Acciones  
(y Cambio o Evolución)

caraterización de  
la situación inicial

descripción de la evolución de las  
relaciones dinámicas resultado de  
aplicar acciones.



# Situation Calculus: Representación

## Representación de una Acción

- ▶ **Axioma de Posibilidad:** indica cuando es posible ejecutar una acción. (precondiciones)
- ▶ **Axioma de Estado Sucesor:** Cada acción tiene asociado un axioma de efecto, que especifica el efecto de ejecutar la acción. Es decir, un fluente es verdadero después de ejecutar una acción si:
  - ▶ El efecto de la acción que ejecutamos hizo verdadero al fluente (Axioma de Efecto), o
  - ▶ el fluente era verdadero antes y la acción lo dejó igual (Axioma del Frame)

# Situation Calculus: Representación

## Representación de una Acción

- ▶ **Axioma de Posibilidad:** indica cuando es posible ejecutar una acción. (precondiciones)
- ▶ **Axioma de Estado Sucesor:** Cada acción tiene asociado un axioma de efecto, que especifica el efecto de ejecutar la acción. Es decir, un fluente es verdadero después de ejecutar una acción si:
  - ▶ El efecto de la acción que ejecutamos hizo verdadero al fluente (Axioma de Efecto), o
  - ▶ el fluente era verdadero antes y la acción lo dejó igual (Axioma del Frame)

# Situation Calculus: Representación

## Representación de una Acción

- ▶ **Axioma de Posibilidad:** indica cuando es posible ejecutar una acción. (precondiciones)
- ▶ **Axioma de Estado Sucesor:** Cada acción tiene asociado un axioma de efecto, que especifica el efecto de ejecutar la acción. Es decir, un fluente es verdadero después de ejecutar una acción si:
  - ▶ El efecto de la acción que ejecutamos hizo verdadero al fluente (Axioma de Efecto), o
  - ▶ el fluente era verdadero antes y la acción lo dejó igual (Axioma del Frame)

# Situation Calculus: Representación

## Representación de una Acción

- ▶ **Axioma de Posibilidad:** indica cuando es posible ejecutar una acción. (precondiciones)
- ▶ **Axioma de Estado Sucesor:** Cada acción tiene asociado un axioma de efecto, que especifica el efecto de ejecutar la acción. Es decir, un fluente es verdadero después de ejecutar una acción si:
  - ▶ El efecto de la acción que ejecutamos hizo verdadero al fluente (Axioma de Efecto), o
  - ▶ el fluente era verdadero antes y la acción lo dejó igual (Axioma del Frame)

# Situation Calculus: Mundo de Bloques

Move(B,X,Y)

## Precondiciones

$\text{poss}(\text{move}(\text{B}, \text{X}, \text{Y}), \text{S}) \leftarrow$   
 $\text{bloque}(\text{B}), \text{bloque}(\text{X}), \text{bloque}(\text{Y}), \text{clear}(\text{Y}, \text{S}), \text{clear}(\text{B}, \text{S}), \text{on}(\text{B}, \text{X}, \text{S}).$

## Estado Sucesor

$$\left( \text{clear}(\text{X}, \text{do}(\text{A}, \text{S})) \leftrightarrow \left( \text{A} = \text{move}(\text{B}, \text{X}, \text{Y}) \vee \right. \right. \\ \left. \left. \text{clear}(\text{X}, \text{S}) \wedge \text{A} \neq \text{move}(\text{B}, \text{Z}, \text{X}) \right) \right).$$

Siguiendo notación de Poole, utilizamos dos reglas:

**Ax de efecto:**

$\text{clear}(\text{X}, \text{do}(\text{move}(\text{B}, \text{X}, \text{Y}), \text{S})) \leftarrow \text{poss}(\text{move}(\text{B}, \text{X}, \text{Y}), \text{S}).$

**Ax de frame:**

$\text{clear}(\text{X}, \text{do}(\text{A}, \text{S})) \leftarrow$   $\text{clear}(\text{X}, \text{S}) \wedge \text{poss}(\text{A}, \text{S}) \wedge$   
 $\text{A} \neq \text{move}(\text{B}, \text{Z}, \text{X})$

# Situation Calculus: Mundo de Bloques

Move(B,X,Y)

## Precondiciones

$\text{poss}(\text{move}(\text{B}, \text{X}, \text{Y}), \text{S}) \leftarrow$   
 $\text{bloque}(\text{B}), \text{bloque}(\text{X}), \text{bloque}(\text{Y}), \text{clear}(\text{Y}, \text{S}), \text{clear}(\text{B}, \text{S}), \text{on}(\text{B}, \text{X}, \text{S}).$

## Estado Sucesor

$$\left( \text{clear}(\text{X}, \text{do}(\text{A}, \text{S})) \leftrightarrow \left( \text{A} = \text{move}(\text{B}, \text{X}, \text{Y}) \vee \right. \right. \\ \left. \left. (\text{clear}(\text{X}, \text{S}) \wedge \text{A} \neq \text{move}(\text{B}, \text{Z}, \text{X})) \right) \right).$$

Siguiendo notación de Poole, utilizamos dos reglas:

### **Ax de efecto:**

$\text{clear}(\text{X}, \text{do}(\text{move}(\text{B}, \text{X}, \text{Y}), \text{S})) \leftarrow \text{poss}(\text{move}(\text{B}, \text{X}, \text{Y}), \text{S}).$

### **Ax de frame:**

$\text{clear}(\text{X}, \text{do}(\text{A}, \text{S})) \leftarrow$   $\text{clear}(\text{X}, \text{S}) \wedge \text{poss}(\text{A}, \text{S}) \wedge$   
 $\text{A} \neq \text{move}(\text{B}, \text{Z}, \text{X})$

# Event Calculus

El Cálculo Situacional permite a un agente ejecutar acciones discretas instantáneas.

Problemas para representar acciones que tienen duración o que se realizan en paralelo.

Necesitamos

Representación Temporal  
Basada en tiempos en vez de en situaciones.

# Event Calculus

El Cálculo Situacional permite a un agente ejecutar acciones discretas instantáneas.

Problemas para representar acciones que tienen duración o que se realizan en paralelo.

Necesitamos

Representación Temporal  
Basada en tiempos en vez de en situaciones.



# Event Calculus: Características

- ▶ Los fluentes ocurren en un punto en el tiempo
- ▶ El cálculo está diseñado para permitir razonamiento sobre intervalos de tiempos.

Un fluente es verdadero en un punto en el tiempo  $t$

- ▶ Si el fluente fue *inicializado* por un evento en algún tiempo  $t'$  del pasado, es decir  $t' < t$  y
- ▶ Si el fluente *no fue terminado* por un evento intermedio, es decir algún evento que ocurrió entre  $t'$  y  $t$ .

# Event Calculus: Características

- ▶ Los fluentes ocurren en un punto en el tiempo
- ▶ El cálculo está diseñado para permitir razonamiento sobre intervalos de tiempos.

Un fluente es verdadero en un punto en el tiempo  $t$

- ▶ Si el fluente fue *inicializado* por un evento en algún tiempo  $t'$  del pasado, es decir  $t' < t$  y
- ▶ Si el fluente *no fue terminado* por un evento intermedio, es decir algún evento que ocurrió entre  $t'$  y  $t$ .

# Event Calculus: Características

- ▶ Los fluentes ocurren en un punto en el tiempo
- ▶ El cálculo está diseñado para permitir razonamiento sobre intervalos de tiempos.

## Un fluente es verdadero en un punto en el tiempo $t$

- ▶ Si el fluente fue *inicializado* por un evento en algún tiempo  $t'$  del pasado, es decir  $t' < t$  y
- ▶ Si el fluente *no fue terminado* por un evento intermedio, es decir algún evento que ocurrió entre  $t'$  y  $t$ .

# Event Calculus: Características

- ▶ Los fluentes ocurren en un punto en el tiempo
- ▶ El cálculo está diseñado para permitir razonamiento sobre intervalos de tiempos.

Un fluente es verdadero en un punto en el tiempo  $t$

- ▶ Si el fluente fue **inicializado** por un evento en algún tiempo  $t'$  del pasado, es decir  $t' < t$  y
- ▶ Si el fluente **no fue terminado** por un evento intermedio, es decir algún evento que ocurrió entre  $t'$  y  $t$ .

# Event Calculus: Características

- ▶ Los fluentes ocurren en un punto en el tiempo
- ▶ El cálculo está diseñado para permitir razonamiento sobre intervalos de tiempos.

Un fluente es verdadero en un punto en el tiempo  $t$

- ▶ Si el fluente fue **inicializado** por un evento en algún tiempo  $t'$  del pasado, es decir  $t' < t$  y
- ▶ Si el fluente **no fue terminado** por un evento intermedio, es decir algún evento que ocurrió entre  $t'$  y  $t$ .

# Event Calculus: Representación

$\text{event}(E, T)$

El evento  $E$  ocurrió en el tiempo  $T$ .

Para cada evento debemos especificar aquello que hace verdadero y aquello que deja de ser verdadero:

$\text{initiates}(E, P, T)$

Es verdadero si el evento  $E$  hace al predicado  $P$  verdadero en el tiempo  $T$ .

$\text{terminates}(E, P, T)$

Es verdadero si el evento  $E$  hace que el predicado  $P$  deje de ser verdadero en el tiempo  $T$ .

# Event Calculus: Representación

$\text{event}(E, T)$

El evento  $E$  ocurrió en el tiempo  $T$ .

Para cada evento debemos especificar aquello que hace verdadero y aquello que deja de ser verdadero:

$\text{initiates}(E, P, T)$

Es verdadero si el evento  $E$  hace al predicado  $P$  verdadero en el tiempo  $T$ .

$\text{terminates}(E, P, T)$

Es verdadero si el evento  $E$  hace que el predicado  $P$  deje de ser verdadero en el tiempo  $T$ .

# Event Calculus: Representación

$\text{event}(E, T)$

El evento  $E$  ocurrió en el tiempo  $T$ .

Para cada evento debemos especificar aquello que hace verdadero y aquello que deja de ser verdadero:

$\text{initiates}(E, P, T)$

Es verdadero si el evento  $E$  hace al predicado  $P$  verdadero en el tiempo  $T$ .

$\text{terminates}(E, P, T)$

Es verdadero si el evento  $E$  hace que el predicado  $P$  deje de ser verdadero en el tiempo  $T$ .



# Event Calculus: Representación

$\text{event}(E, T)$

El evento  $E$  ocurrió en el tiempo  $T$ .

Para cada evento debemos especificar aquello que hace verdadero y aquello que deja de ser verdadero:

$\text{initiates}(E, P, T)$

Es verdadero si el evento  $E$  hace al predicado  $P$  verdadero en el tiempo  $T$ .

$\text{terminates}(E, P, T)$

Es verdadero si el evento  $E$  hace que el predicado  $P$  deje de ser verdadero en el tiempo  $T$ .

# Event Calculus: Representación

Para cada fluente, deberemos determinar si éste ocurre o no en un tiempo particular:

$$\text{holds}(P,T) \leftarrow \begin{array}{l} \text{event}(E,T_0) \wedge \\ T_0 < T \wedge \\ \text{initiates}(E,P,T_0) \wedge \\ \sim \text{clipped}(P,T_0,T). \end{array}$$
$$\text{clipped}(P,T_0,T) \leftarrow \begin{array}{l} \text{event}(E_1,T_1) \wedge \\ \text{terminates}(E_1,P,T_1) \wedge \\ T_0 < T_1 \wedge \\ T_1 < T. \end{array}$$

# Event Calculus: Mundo de Bloques

**move(B,X,Y)**

$\text{initiates}(\text{move}(\text{B},\text{X},\text{Y}),\text{clear}(\text{X}),\text{T}) \leftarrow \text{poss}(\text{move}(\text{B},\text{X},\text{Y}),\text{T}).$

$\text{initiates}(\text{move}(\text{B},\text{X},\text{Y}),\text{on}(\text{B},\text{Y}),\text{T}) \leftarrow \text{poss}(\text{move}(\text{B},\text{X},\text{Y}),\text{T}).$

$\text{terminates}(\text{move}(\text{B},\text{X},\text{Y}),\text{clear}(\text{Y}),\text{T}) \leftarrow \text{poss}(\text{move}(\text{B},\text{X},\text{Y}),\text{T}).$

$\text{terminates}(\text{move}(\text{B},\text{X},\text{Y}),\text{on}(\text{B},\text{X}),\text{T}) \leftarrow \text{poss}(\text{move}(\text{B},\text{X},\text{Y}),\text{T}).$

$\text{poss}(\text{move}(\text{B},\text{X},\text{Y}),\text{T}) \leftarrow$   
 $\text{holds}(\text{clear}(\text{B}),\text{T}) \wedge$   
 $\text{holds}(\text{clear}(\text{Y}),\text{T}) \wedge$   
 $\text{holds}(\text{on}(\text{B},\text{X}),\text{T})$

# Rep. de Acciones y Cambio

- ▶ **Tiempo Discreto:** se modela como saltando de un punto en el tiempo a otro, separados entre sí de manera uniforme.
- ▶ **Tiempo Continuo:** se modela como una estructura lineal continua.
- ▶ **Tiempo Basado en Eventos:** Análogo a Tiempo Discreto, pero los puntos en el tiempo marcan eventos “interesantes”, y no necesitan estar separados entre sí de manera uniforme. Ej: Cálculo de Eventos.
- ▶ **Espacio de Estados:** en lugar de considerar el tiempo explícitamente, podemos considerar a las acciones como mapeando de un estado a otro del mundo. Ej: Cálculo de Situaciones y STRIPS.

# Rep. de Acciones y Cambio

El tiempo puede incorporarse a un sistema de Rep. de Conoc. y Razonamiento de diferentes formas:

- ▶ **El Tiempo (y acciones) son externas a la lógica.** Se representa en la lógica solo un estado (correspondiente a un determinado momento), y se utilizan sentencias fuera de la lógica para especificar cómo una acción mapea desde la descripción de un estado a la descripción del siguiente. Ej. **STRIPS**.
- ▶ **El Tiempo se encuentra reificado.** ie, el tiempo es otro objeto más, que se representa y del que se puede hablar, en la lógica. Concretamente, se puede agregar un argumento a las relaciones para especificar cuando valen. Ej: `pasa_autobus(101, cuadra(Mitre, 200), pm(15 : 35))` Ej

**Cálculo de Situaciones**

# Referencia Bibliográfica



S. Russell y P. Norvig

*Artificial Intelligence: A Modern Approach (Third Edition).*

Capítulo 10

2009.



D. Poole, A. Mackworth y R. Goebel

*Computational Intelligence: A Logical Approach.*

Capítulo 8

1998.

## Desafíos

Considere el mundo de bloques.

Dé la representación en STRIPS, Situation Calculus y Event Calculus

# ¿Preguntas?