

COL·LECCIÓ DE PROBLEMES

Departament de Llenguatges i Sistemes Informàtics



FIB

Facultat d'Informàtica
de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Índice general

1. Aprendizaje Inductivo	1
1.1. Espacio de Versiones	1
1.2. Árboles de Inducción	4
1.3. Naïve Bayes	7
2. Aprendizaje No Supervisado	9
3. Aprendizaje Basado en Explicaciones	11
4. Aprendizaje por refuerzo	23

Esta obra está bajo una licencia
Reconocimiento-NoComercial-CompartirIgual de Creative Commons.

Para ver una copia de esta licencia, visite
<http://creativecommons.org/licenses/by-nc-sa/2.5/es/>
o envíe una carta a

Creative Commons,
559 Nathan Abbott Way, Stanford,
California 94305,
USA.

Aprendizaje Inductivo

1.1. Espacio de Versiones

- Supongamos que tenemos un lenguaje de descripción de conceptos que admite solo formulas conjuntivas puras y tenemos el dominio de los coches con los siguientes atributos:

Precio: caro, medio, barato. Motor: diesel, gasolina.
 Color: azul, verde, rojo. País: Japón, Francia, USA, Alemania.
 Tamaño: grande, mediano, pequeño.

Genera un conjunto de ejemplos para aprender el siguiente concepto {Coches japoneses caros} y muestra la evolución del método del espacio de versiones para aprenderlo

- Supongamos que tenemos un lenguaje de descripción de conceptos que admite solo formulas conjuntivas puras y tenemos el dominio de las películas con los siguientes atributos:

Sonoro:	si, no
Color:	si, no
Nacionalidad:	USA, España, Francia, Italia
Duración:	cortometraje, mediometraje, largometraje
Género:	comedia, drama, acción, negro, policiaca, aventuras

Genera el conjunto mínimo de ejemplos para aprender el siguiente concepto {largometrajes franceses en color} y muestra la evolución del método del espacio de versiones para aprenderlo

- Tomando el dominio de los animales con los siguientes descriptores:
 - Especie (*jerárquico*); valores: mamífero (bobino, felino, cánido, ...), reptil (ofidio, saurio, quelonio, ...), ave (corredoras, rapaces, palmípedas, ...).
 - Origen (*jerárquico*); valores: África (Zaire, Tanzania, Kenia, ...), Asia (India, Thailandia, Siria, Nepal, ...), América (Brasil, Bolivia, Perú, Nicaragua, ...).
 - Alimentación (*cualitativo*); valores: Carnívoro, Insectívoro, Omnívoro, Herbívoro.
 - Habitat (*cualitativo*); valores: montaña, mar, sabana, tundra, desierto, selva.
 - Defensa (*cualitativo*); valores: dientes, cuernos, veneno, garras, carrera, ...
 - Tiempo de desarrollo (*cualitativo*); valores: semanas, meses, años.

Generar un conjunto de ejemplos para aprender alguno de los siguientes conceptos y mostrar la evolución del espacio de versiones:

- a) Mamíferos asiáticos herbívoros.

- b) Aves marinas carnívoras americanas.
- c) Felinos africanos de montaña.
- d) Carnívoros de sabana.
- e) Animales que corren y se desarrollan en meses.
- f) Felinos asiáticos carnívoros.
- g) Aves Peruanas insectívoras con garras.
- h) Reptiles que corren y que se desarrollan en semanas.
- i) Mamíferos herbívoros del desierto.
- j) Animales selváticos de Asia.
- k) Reptiles americanos selváticos
- l) Mamíferos zaireños con garras que se desarrollan en meses
- m) Aves carnívoras
- n) Animales de montaña con garras
- ñ) Reptiles
- o) Animales mamíferos africanos con dientes
- p) Animales carnívoros americanos
- q) Animales marinos brasileños venenosos que se desarrollan en meses
- r) Animales reptiles asiáticos omnívoros
- s) Animales asiáticos

4. Supongamos que ampliamos nuestro lenguaje de descripción de manera que podemos describir conceptos y ejemplos con una sola disyunción en la que los disyuntados no están ordenados. Por ejemplo, si tomamos el dominio de los objetos geométricos podemos expresar el ejemplo ((bola rojo grande) o (cubo verde pequeño)) que es equivalente a ((cubo verde pequeño) o (bola rojo grande)), donde ahora un nuevo ejemplo puede generalizar un concepto de dos maneras posibles, por ejemplo, la instancia ((cubo rojo pequeño) o (cubo verde grande)) puede generalizar el ejemplo anterior como ((? rojo ?) o (cubo verde ?)) o como ((? ? grande) o (cubo ? pequeño)), igual pasa con la especialización. Teniendo en cuenta este nuevo lenguaje y como se comporta ante la generalización y la especialización, y tomando como dominio el de los coches, descrito con los siguientes atributos:

- Precio: caro, medio, barato.
- Motor: diesel, gasolina.
- Color: azul, verde, rojo.
- País: Japón, Francia, USA, Alemania.
- Tamaño: grande, mediano, pequeño.

Generar un conjunto de ejemplos para aprender alguno de los siguientes conceptos y mostrar la evolución del espacio de versiones:

- a) coches diesel americanos grandes o coches diesel americanos caros.
- b) coches verdes japoneses pequeños o coches franceses diesel grandes.
- c) coches caros japoneses o coches caros americanos.
- d) coches grandes azules o coches japoneses diesel.
- e) coches americanos de gasolina rojos o coches alemanes medianos.
- f) Coches diesel japoneses pequeños o coches de gasolina azules grandes.
- g) Coches americanos o coches franceses grandes

- h) Coches diesel japoneses caros o coches diesel americanos grandes.
- i) Coches caros o coches de gasolina baratos

5. Supongamos que tenemos un lenguaje de descripción de conceptos que admite una única disyunción (como en el problema anterior) y tenemos el dominio de los coches con los siguientes atributos:

Precio: caro, medio, barato. Motor: diesel, gasolina.
 Color: azul, verde, rojo. País: Japón, Francia, USA, Alemania.
 Tamaño: grande, mediano, pequeño.

Suponiendo que los disyuntandos no están ordenados (igual que la otra vez), ¿cual sería la modificación del espacio de versiones que aparece a continuación, con el siguiente contraejemplo: ((barato diesel verde Francia grande) (caro diesel verde USA grande))

$G = \{((? ? ? ? ?) (? ? ? ? ?))\}$
 $S = \{((? diesel ? USA grande) (caro diesel ? USA ?))\}$

¿Cual sería la modificación del mismo espacio de versiones si el contraejemplo fuera: ((caro diesel verde Japón pequeño)(caro diesel verde Francia pequeño))?

6. Supongamos que tenemos un lenguaje de descripción de conceptos que admite una única disyunción (como en el problema de espacio de versiones que resolvisteis) y tenemos el dominio de los coches con los siguientes atributos:

Precio: caro, medio, barato. Motor: diesel, gasolina.
 Color: azul, verde, rojo. País: Japón, Francia, USA, Alemania.
 Tamaño: grande, mediano, pequeño.

Suponiendo que los disyuntandos no están ordenados (igual que la otra vez), ¿cual sería la modificación del espacio de versiones que aparece a continuación, con el siguiente contraejemplo: ((caro gasolina verde USA mediano) (barato gasolina verde Francia mediano))

$G = \{((? ? ? ? ?) (? ? ? ? ?))\}$
 $S = \{((? gasolina ? Francia mediano) (barato gasolina ? Francia ?))\}$

¿Cual sería la modificación del mismo espacio de versiones si el contraejemplo fuera: ((barato gasolina verde Japón grande)(barato gasolina verde USA grande))?

7. Supongamos que ampliamos nuestro lenguaje de descripción de manera que podemos describir conceptos y ejemplos con una sola disyunción en la que los disyuntandos no están ordenados. Por ejemplo, si tomamos el dominio de los objetos geométricos podemos expresar el ejemplo ((bola rojo grande) o (cubo verde pequeño)) que es equivalente a ((cubo verde pequeño) o (bola rojo grande)), donde ahora un nuevo ejemplo puede generalizar un concepto de dos maneras posibles, por ejemplo, la instancia ((cubo rojo pequeño) o (cubo verde grande)) puede generalizar el ejemplo anterior como ((? rojo ?) o (cubo verde ?)) o como ((? ? grande) o (cubo ? pequeño)), igual pasa con la especialización. Teniendo en cuenta este nuevo lenguaje y como se comporta ante la generalización y la especialización, y tomando como dominio el de las películas de cine, descrito con los siguientes atributos:

- Sonoro: sí, no.
- Nacionalidad: USA, España, Francia, Italia.
- Color: sí, no.
- Género: comedia drama, acción, negro, policiaca, aventuras.
- Duración: cortometraje, medimetraje, largometraje.

Generar un conjunto de ejemplos para aprender alguno de los siguientes conceptos y mostrar la evolución del espacio de versiones:

- a) Películas sonoras americanas o películas mudas de acción.
- b) Largometrajes españoles dramáticos o cortometrajes cómicos italianos.
- c) Películas americanas de cine negro en color o comedias americanas en blanco y negro.
- d) Películas francesas mudas en blanco y negro o comedias mudas en blanco y negro.
- e) Cortometrajes en color o en blanco y negro
- f) Películas mudas en blanco y negro o películas sonoras en color
- g) Largometrajes americanos de género negro o cortometrajes franceses cómicos
- h) Películas dramáticas americanas o cortometrajes italianos
- i) Cortometrajes mudos o sonoros
- j) Comedias en color o largometrajes de acción americanos
- k) Películas policíacas en blanco y negro o en color
- l) Cortometrajes de acción americanos o películas cómicas en blanco y negro
- m) Largometrajes de aventuras en color o cortometrajes dramáticos mudos
- n) Películas cómicas sonoras o medietrajes dramáticos en color
- ñ) Películas sonoras en color o películas mudas en blanco y negro

8. Supongamos que tenemos un lenguaje de descripción de conceptos que admite una única disyunción (como en el problema de espacio de versiones que resolvisteis) y tenemos el dominio de los coches con los siguientes atributos:

Precio: caro, medio, barato. Motor: diesel, gasolina.
 Color: azul, verde, rojo. País: Japón, Francia, USA, Alemania.
 Tamaño: grande, mediano, pequeño.

Suponiendo que los disyuntandos no están ordenados (igual que la otra vez), genera un conjunto de ejemplos para aprender los siguientes conceptos:

- a) Coches diesel caros medianos o coches de gasolina japoneses verdes.
- b) Coches grandes o pequeños

1.2. Árboles de Inducción

1. Rehaz los ejemplos de espacio de versiones utilizando el algoritmo de ID3 y compara el resultado para el conjunto de ejemplos positivos.
2. Dados los siguientes ejemplos:

Ej.	Color	Tamaño	Clase
1	Azul	Grande	+
2	Rojo	Pequeño	-
3	Azul	Pequeño	+
4	Azul	Grande	-

¿Que atributo tiene la mayor ganancia de información respecto a la clasificación?

3. Dados los siguientes ejemplos:

Ej.	Clase	A1	A2
1	+	T	T
2	+	T	T
3	-	T	F
4	+	F	F
5	-	F	T
6	-	F	T

¿Cual es la entropía de este conjunto de ejemplos? ¿Que atributo tiene la mayor ganancia de información respecto a la clasificación?

4. Un profesor poco escrupuloso pretende ahorrarse parte de la tarea de corrección de prácticas a base de construir un árbol de discriminación que le permita asociar ciertas características de las prácticas a un resultado. En concreto se utilizarán los atributos: Tamaño de la práctica (páginas), presentación (buena, regular, mala), documentación (existe o no), funcionamiento (funciona o no). Se podrán dar tres resultados posibles: aprueba, suspende, corregir. Generar el árbol de inducción que resulta del siguiente conjunto de ejemplos:

Pags.	Presen.	Docum.	Funciona	Resultado
100	Regular	Si	Si	Aprueba
100	Buena	No	No	Suspende
80	Mala	Si	Si	Aprueba
50	Regular	No	Si	Corregir
50	Buena	Si	Si	Aprueba
30	Mala	No	No	Suspende
20	Regular	Si	Si	Corregir
20	Buena	No	Si	Corregir
20	Mala	Si	Si	Suspende
10	Regular	No	Si	Suspende

¿Podrías averiguar los criterios de corrección del profesor? Propón unos criterios de selección de los atributos a cada nivel guiado por tu intuición sin tener en cuenta el resultado y compara los dos árboles. Razona sobre la necesidad de tener un conjunto representativo de ejemplos para obtener un resultado acorde con el criterio de un experto.

5. Construir un árbol de decisión para los siguientes conjuntos de ejemplos:

a)

Ejem.	Color	Forma	Tamaño	Clase
1	Rojo	Cuadrado	Grande	+
2	Azul	Cuadrado	Grande	+
3	Rojo	Redondo	Pequeño	-
4	Verde	Cuadrado	Pequeño	-
5	Rojo	Redondo	Grande	+
6	Verde	Redondo	Grande	-

Ejem.	Clima	Temp.	Humedad	Viento	Picnic
1	Soleado	Calor	Alta	Débil	No
2	Soleado	Calor	Alta	Fuerte	No
3	Nublado	Calor	Alta	Débil	Si
4	Lluvioso	Suave	Alta	Débil	Si
5	Lluvioso	Frío	Normal	Débil	Si
6	Lluvioso	Frío	Normal	Fuerte	No
7	Nublado	Frío	Normal	Fuerte	Si
8	Soleado	Suave	Alta	Débil	No
9	Soleado	Frío	Normal	Débil	Si
10	Lluvioso	Suave	Normal	Débil	Si
11	Soleado	Suave	Normal	Fuerte	Si
12	Nublado	Suave	Alta	Fuerte	Si
13	Nublado	Calor	Normal	Débil	Si
14	Lluvioso	Suave	Alta	Fuerte	No

Ej.	Historial	Deudas	Informes	Sueldo	Riesgo
1	Malo	Altas	Ninguno	0 a 2M	Alto
2	Desconocido	Altas	Ninguno	2M a 5M	Alto
3	Desconocido	Bajas	Ninguno	2M a 5M	Moderado
4	Desconocido	Bajas	Ninguno	0 a 2M	Alto
5	Desconocido	Bajas	Ninguno	Mas de 5M	Bajo
6	Desconocido	Bajo	Buenos	Mas de 5M	Bajo
7	Malo	Bajas	Ninguno	0 a 2M	Alto
8	Malo	Bajas	Buenos	Mas de 5M	Moderado
9	Bueno	Bajas	Ninguno	Mas de 5M	Bajo
10	Bueno	Altas	Buenos	Mas de 5M	Bajo
11	Bueno	Altas	Ninguno	0 a 2M	Alto
12	Bueno	Altas	Ninguno	2M a 5M	Moderado
13	Bueno	Altas	Ninguno	Mas de 5M	Bajo
14	Malo	Altas	Ninguno	2M a 5M	Alto

6. Dada la siguiente tabla que representa ejemplos de clasificación de películas de cine según los cinco atributos que aparecen a continuación, construye el árbol de decisión resultante de aplicar el algoritmo de ID3.

Atributos	Valores
nacionalidad:	USA, Europa, Asia
genero:	comedia, drama, acción, policiaca, aventuras
actores:	guaperas, actor_de_verdad, normal
guión:	original, adaptado
director:	bueno, por_encargo, normal

ID	Nacionalidad	Género	Actores	Guión	Director	Clase
1	USA	accion	guaperas	original	bueno	buena
2	USA	drama	actor_de_verdad	adaptado	bueno	buena
3	Europa	drama	actor_de_verdad	adaptado	bueno	buena
4	USA	aventuras	actor_de_verdad	original	normal	buena
5	Europa	aventuras	actor_de_verdad	adaptado	bueno	buena
6	USA	drama	guaperas	original	por_encargo	regular
7	USA	drama	normal	adaptado	normal	regular
8	USA	policiaica	actor_de_verdad	original	normal	regular
9	USA	comedia	normal	original	normal	regular
10	USA	aventuras	guaperas	original	normal	regular
11	Asia	accion	normal	original	normal	mala
12	Asia	drama	normal	original	normal	mala
13	Europa	drama	normal	original	por_encargo	mala
14	Europa	comedia	guaperas	original	por_encargo	mala
15	USA	drama	guaperas	original	normal	mala

7. El algoritmo ID3 explicado en clase esencialmente busca el árbol de decisión mínimo escogiendo en cada nivel la condición que de máxima información sobre la pertenencia de cada instancia a una clase.
 - a) ¿Como se comportaría el algoritmo si cada instancia tiene un atributo que es identificador único (p.e.: el DNI)?
 - b) ¿Como predeciría un árbol con datos con esas características?
8. En muchos dominios no todos los atributos son igualmente costosos de comprobar (p.e.: En un dominio médico posiblemente sea más barato hacer una análisis de sangre que una resonancia magnética). Suponiendo que el coste de conocer el valor del atributo X es C_X , proponer una modificación del algoritmo de ID3 para construir árboles que minimizen el coste de la comprobación de los valores de los atributos.
9. En algunos dominios existen atributos en los que en algunas instancias se puede haber perdido su valor. Como modificarías el criterio de selección de los atributos para que se tenga en cuenta esto.
10. Dados los siguientes ejemplos, determina cual sería el atributo raíz que escogería el algoritmo de ID3 con la función de ganancia de información¹. ¿Tiene alguna influencia el número de valores por cada atributo en la decisión?, calcula cual sería el atributo elegido utilizando el *split info*.

Obj	A	B	C	Clase	Obj	A	B	C	Clase
1	a	x	1	C1	8	c	y	3	C2
2	b	y	1	C1	9	b	x	3	C2
3	b	z	3	C1	10	c	y	4	C2
4	a	z	2	C1	11	b	x	2	C2
5	a	z	3	C1	12	c	z	3	C2
6	b	y	2	C1	13	b	z	2	C2
7	c	x	1	C1	14	c	x	4	C2

Atributo A={a,b,c} Atributo B={x,y,z} Atributo C={1,2,3,4}

1.3. Naïve Bayes

1. En la siguiente tabla esta representada la distribución de probabilidad aprendida mediante naïve bayes a partir de un conjunto de ejemplos pertenecientes a tres clases. El conjunto de datos consta de tres atributos A1, A2, A3, con los siguientes valores A1={a,b}, A2={a,b,c}, A3={1,2,3}.

¹Recordad que $\log_a(x) = \frac{\log_b(x)}{\log_b(a)}$ y que es un factor común.

	A1		A2			A3		
	a	b	a	b	c	1	2	3
Clase 1	0.3	0.7	0.1	0.2	0.7	0.6	0.2	0.2
Clase 2	0.9	0.2	0.2	0.4	0.4	0.2	0.5	0.3
Clase 3	0.4	0.6	0.3	0.3	0.4	0.1	0.6	0.3

Dado el siguiente ejemplo $e_{j1} = (A1=b, A2=b, A3=3)$, calcula a que clase lo asignaría el clasificador aprendido.

2. En la siguiente tabla esta representada la distribución de probabilidad aprendida mediante naïve bayes a partir de un conjunto de ejemplos pertenecientes a tres clases. El número de ejemplos de cada clase utilizados para general la tabla son (clase 1 = 100 ejemplos, clase 2 = 200 ejemplos, clase 3= 100 ejemplos). El conjunto de datos consta de tres atributos A1, A2, A3, con los siguientes valores $A1=\{a,b\}$, $A2=\{R,G,B\}$, $A3=\{x,y,z\}$.

	A1		A2			A3		
	a	b	R	G	B	x	y	z
Clase 1	0.2	0.8	0.2	0.2	0.6	0.1	0.6	0.3
Clase 2	0.5	0.5	0.3	0.6	0.1	0.3	0.2	0.5
Clase 3	0.7	0.3	0.5	0.4	0.1	0.7	0.1	0.2

Dado el siguiente ejemplo $e_{j1} = (A1=b, A2=R, A3=z)$, ¿que probabilidad de pertenencia a cada clase le asignaría el clasificador aprendido?

3. En la siguiente tabla está representada las distribuciones de probabilidad aprendidas mediante naïve bayes a partir de un conjunto de ejemplos pertenecientes a tres clases. El número de ejemplos de cada clase utilizados para general la tabla son: clase 1 = 300 ejemplos, clase 2 = 200 ejemplos, clase 3= 100 ejemplos. El conjunto de datos consta de tres atributos A1, A2, A3, con los siguientes valores $A1=\{a,b\}$, $A2=\{R,G,B\}$, $A3=\{x,y,z\}$.

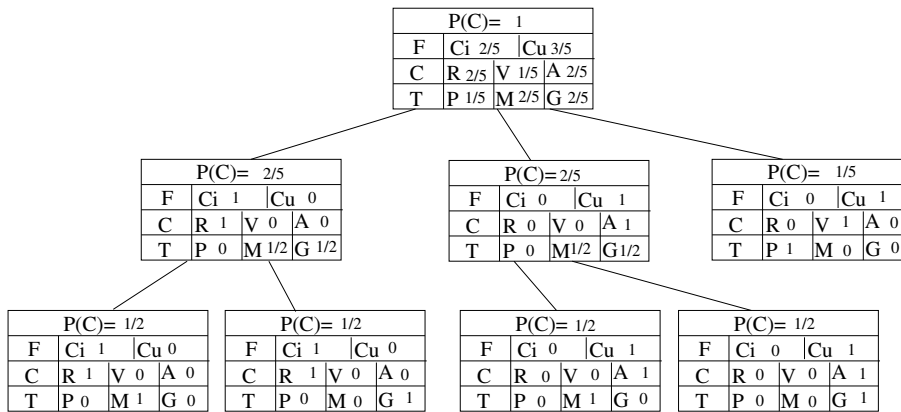
	A1		A2			A3		
	a	b	R	G	B	x	y	z
Clase 1	0.3	0.7	0.1	0.3	0.6	0.1	0.6	0.3
Clase 2	0.9	0.1	0.4	0.5	0.1	0.3	0.3	0.4
Clase 3	0.6	0.4	0.7	0.1	0.2	0.4	0.4	0.2

Dado el siguiente ejemplo $e_{j1} = (A1=a, A2=G, A3=y)$, ¿que probabilidad de pertenencia a cada clase le asignaría el clasificador aprendido?

Capítulo 2

Aprendizaje No Supervisado

1. Dada la siguiente jerarquía construida mediante el algoritmo de **COBWEB**.



El dominio consta de 3 atributos:

- Forma: circulo, cuadrado.
- Color: rojo, verde, amarillo
- Tamaño: pequeño, mediano, grande

¿Como se modificaría al introducir el ejemplo: (cuadrado verde grande)?

Aprendizaje Basado en Explicaciones

1. Considera la siguiente teoría de dominio para el concepto *aprueba*:

$$\begin{aligned}
 \text{aprueba}(x) &\leftarrow \text{conoce_la_asignatura}(x) \wedge \text{esta_despierto}(x) \\
 \text{conoce_la_asignatura}(x) &\leftarrow \text{ha_estudiado}(x) \\
 \text{esta_despierto}(x) &\leftarrow \text{durmio_anoche}(x) \\
 \text{ha_estudiado} &\leftarrow \text{fue_a_clase}(x) \wedge \text{leyo_los_apuntes}(x)
 \end{aligned}$$

Supón que los predicados operacionizables son $\{\text{durmio_anoche}, \text{estudio}, \text{desayuno}, \text{aprobo_el_parcial}, \text{fue_a_clase}, \text{leyo_los_apuntes}\}$. Genera la explicación para la siguiente instancia positiva del concepto:

$$\begin{aligned}
 &\text{durmio_anoche}(\text{juan}) \wedge \text{estudio}(\text{juan}) \wedge \neg \text{desayuno}(\text{juan}) \wedge \\
 &\neg \text{aprobo_el_parcial}(\text{juan}) \wedge \text{leyo_los_apuntes}(\text{juan})
 \end{aligned}$$

2. Aprende el concepto *dias en los que navegar* utilizando EBG. Tenemos la siguiente teoría del dominio:

$$\begin{aligned}
 \text{Dia_para_navegar}(x) &\leftarrow \text{lloviendo}(x, \text{no}) \wedge \text{laborable}(x, \text{no}) \\
 \text{Dia_para_navegar}(x) &\leftarrow \text{lloviendo}(x, \text{no}) \wedge \text{viento}(x) \\
 \text{laborable}(x, \text{no}) &\leftarrow \text{fin_de_semana}(x) \\
 \text{fin_de_semana}(x) &\leftarrow \text{domingo}(x) \\
 \text{fin_de_semana}(x) &\leftarrow \text{sabado}(x) \\
 \text{lloviendo}(x, \text{no}) &\leftarrow \text{soleado}(x, \text{si})
 \end{aligned}$$

Tenemos como instancia positiva del concepto:

$$\begin{aligned}
 &\text{Dia_para_navegar}(\text{Julio_5}), \text{aniversario_juan}(\text{Julio_5}), \text{sabado}(\text{Julio_5}), \text{soleado}(\text{Julio_5}, \text{si}), \\
 &\text{dia_siguiente}(\text{Julio_5}, \text{Julio_6})
 \end{aligned}$$

El concepto sólo puede estar expresado con los predicados de la instancia.

3. Aprende el concepto *apilable* utilizando EBG. Tenemos la siguiente teoría del dominio:

$$\begin{aligned}
 \text{apilable}(x, y) &\leftarrow \neg \text{fragil}(y) \\
 \text{apilable}(x, y) &\leftarrow \text{mas_ligero}(x, y) \\
 \text{peso}(x, w) &\leftarrow \text{volumen}(x, v) \wedge \text{densidad}(x, d) \wedge \text{multip}(v, d, w) \\
 \text{mas_ligero}(x, y) &\leftarrow \text{peso}(x, w) \wedge \text{peso}(y, z) \wedge \text{menor}(w, z) \\
 \text{peso}(x, 5) &\leftarrow \text{es_un}(x, \text{mesa})
 \end{aligned}$$

Tenemos como instancia positiva del concepto:

sobre(obj1, obj2), es_un(obj1, caja), es_un(obj2, mesa), color(obj1, rojo), color(obj2, azul), volumen(obj1, 1), densidad(obj1, .1), multip(1, x, x), menor(.1, 5)

El concepto sólo puede estar expresado con los predicados de la instancia.

4. Aprende el concepto *mata(juan,juan)* utilizando EBG. Tenemos la siguiente teoría del dominio:

$$\begin{aligned} mata(x, y) &\leftarrow odia(x, y) \wedge posee(x, w) \wedge es_un(w, arma) \\ odia(x, x) &\leftarrow deprimido(x) \\ posee(x, y) &\leftarrow compra(x, y) \\ es_un(x, arma) &\leftarrow es_un(x, pistola) \end{aligned}$$

Tenemos como instancia positiva del concepto:

deprimido(juan), compra(juan, p1), es_un(p1, pistola)

El concepto sólo puede estar expresado con los predicados de la instancia.

5. Aprende el concepto *taza* utilizando EBG. Tenemos la siguiente teoría del dominio:

$$\begin{aligned} taza(x) &\leftarrow estable(x) \wedge alzable(x) \wedge vaso_abierto(x) \\ estable(x) &\leftarrow fondo(y) \wedge parte_de(y, x) \wedge plano(y) \\ alzable(x) &\leftarrow asible(x) \wedge ligero(x) \\ asible(x) &\leftarrow asa(y) \wedge parte_de(y, x) \\ vaso_abierto(x) &\leftarrow concavidad(y) \wedge parte_de(y, x) \wedge hacia_arriba(y) \end{aligned}$$

Tenemos como instancia positiva del concepto:

ligero(obj1), color(obj1, rojo), parte_de(asa1, obj1), asa(asa1), fondo(f1), parte_de(f1, obj1), plano(f1), concavidad(c1), parte_de(c1, obj1), hacia_arriba(c1)

El concepto sólo puede estar expresado con los predicados de la instancia.

6. La siguiente teoría de dominio describe las relaciones de parentesco de una familia. Genera la representación de una familia cualquiera y genera utilizando EBL las explicaciones correspondientes a algunos de los predicados de la teoría planteando algunos ejemplos utilizando la representación de la familia. Supón que solo son operacionizables los predicados *progenitor, casado, varón y mujer*.

$$\begin{aligned} padre(x, y) &\leftarrow progenitor(x, y) \wedge varon(x) \\ casado(x, y) &\leftarrow casado(y, x) \\ madre(x, y) &\leftarrow progenitor(x, y) \wedge mujer(x) \\ marido(x, y) &\leftarrow casado(x, y) \wedge varon(x) \\ esposa(x, y) &\leftarrow casado(x, y) \wedge mujer(x) \\ hijo(x, y) &\leftarrow progenitor(y, x) \wedge varon(x) \\ hija(x, y) &\leftarrow progenitor(y, x) \wedge mujer(x) \\ hermanos(x, y) &\leftarrow padre(f, x) \wedge padre(f, y) \\ &\quad \wedge madre(m, x) \wedge madre(m, y) \\ hermano(x, y) &\leftarrow hermanos(x, y) \wedge varon(x) \\ hermana(x, y) &\leftarrow hermanos(x, y) \wedge mujer(x) \\ cuñada(x, y) &\leftarrow hermano(b, y) \wedge casado(x, b) \\ cuñado(x, y) &\leftarrow hermana(s, y) \wedge casado(x, s) \\ suegra(x, y) &\leftarrow madre(x, s) \wedge casado(s, y) \\ suegro(x, y) &\leftarrow padre(x, s) \wedge casado(s, y) \end{aligned}$$

$$\begin{aligned}
tio(x, y) &\leftarrow progenitor(p, y) \wedge hermano(x, p) \\
tio(x, y) &\leftarrow progenitor(p, y) \wedge hermana(s, p) \wedge marido(x, s) \\
tia(x, y) &\leftarrow progenitor(p, y) \wedge hermana(x, p) \\
tia(x, y) &\leftarrow progenitor(p, y) \wedge hermano(b, p) \wedge esposa(x, b) \\
primo(x, y) &\leftarrow progenitor(p, x) \wedge progenitor(o, y) \wedge hermanos(p, o) \\
abuela(x, y) &\leftarrow progenitor(p, y) \wedge madre(x, p) \\
abuelo(x, y) &\leftarrow progenitor(p, y) \wedge padre(x, p) \\
ancestro(x, y) &\leftarrow progenitor(x, y) \\
ancestro(x, y) &\leftarrow progenitor(p, y) \wedge ancestro(x, p) \\
descendiente(x, y) &\leftarrow ancestro(y, x)
\end{aligned}$$

7. Suponiendo un sistema de aprendizaje que utiliza **EBL** y el dominio de la familia en el que los predicados *progenitor*, *casado*, *varón* y *mujer* son predicados operacionalizables y con la siguiente teoría del dominio:

$$\begin{aligned}
padre(x, y) &\leftarrow progenitor(x, y) \wedge varon(x) \\
madre(x, y) &\leftarrow progenitor(x, y) \wedge mujer(x) \\
casado(x, y) &\leftarrow casado(y, x) \\
marido(x, y) &\leftarrow casado(x, y) \wedge varon(x) \\
esposa(x, y) &\leftarrow casado(x, y) \wedge mujer(x)
\end{aligned}$$

¿Se pueden operacionalizar los siguientes predicados? ¿Es única su operacionalización? ¿Es recomendable operacionalizar los dos? ¿Porque?

$$\begin{aligned}
suegro(x, y) &\leftarrow padre(x, s) \wedge esposa(s, y) \\
ancestro(x, y) &\leftarrow progenitor(p, y) \wedge ancestro(x, p) \\
ancestro(x, y) &\leftarrow progenitor(x, y)
\end{aligned}$$

¿Cual sería el resultado de la operacionalización? (genera un ejemplo que la permita)

8. Supón que podemos describir escenas del mundo de los bloques que contienen una mano de robot, bloques y una mesa. Estos bloques pueden estar sobre la mesa o sobre otro bloque. Tomemos los siguientes operadores de planificación:

coger(x)

- Pre y Suprimir: en_mesa(x), libre(x), manovacia
- Añadir: cogido(x)

dejar(x)

- Pre y Suprimir: cogido(x)
- Añadir: en_mesa(x), libre(x), manovacia

apilar(x)

- Pre y Suprimir: cogido(x), libre(x)
- Añadir: manovacia, sobre(x,y), libre(x)

desapilar(x)

- Pre y Suprimir: manovacia, libre(x), sobre(x,y)
- Añadir: cogido(x), libre(y)

Genera una escena del mundo de los bloques y escoge un objetivo. Crea una tabla triangular con el plan necesario para cumplir el objetivo y generalízala.

9. Supón que añadimos a las escenas anteriores la existencia de habitaciones que pueden estar conectadas entre si por pasadizos. La mano de robot se convierte en un robot entero con una mano. Los bloques y el robot pueden estar en una habitación. El conjunto de operadores disponibles se incrementa de la siguiente manera:

`ir(robot, p, hab1, hab2)`

- Precondición: `conectadas(p, hab1, hab2), en(hab1, robot)`
- Suprimir: `en(hab1, robot)`
- Añadir: `en(hab2, robot)`

`empujar(robot, caja, p, hab1, hab2)`

- Precondición: `conectadas(p, hab1, hab2), en(hab1, robot), en(hab1, caja)`
- Suprimir: `en(hab1, robot), en(hab1, caja)`
- Añadir: `en(hab2, robot), en(hab, caja)`

Genera una escena del mundo de los bloques y escoge un objetivo. Crea una tabla triangular con el plan necesario para cumplir el objetivo y generalízala.

10. Supón que podemos describir escenas del mundo de los bloques que contienen una mano de robot, bloques y una mesa. Estos bloques pueden estar sobre la mesa o sobre otro bloque. Tomemos los siguientes operadores de planificación:

Precondición: `en_mesa(x), libre(x), manovacia`

coger(x): Suprimir: `en_mesa(x), libre(x), manovacia`

Añadir: `cogido(x)`

Precondición: `cogido(x)`

dejar(x): Suprimir: `cogido(x)`

Añadir: `en_mesa(x), libre(x), manovacia`

Precondición: `cogido(x), libre(y)`

apilar(x,y): Suprimir: `cogido(x), libre(y)`

Añadir: `manovacia, sobre(x,y), libre(x)`

Precondición: `manovacia, libre(x), sobre(x,y)`

desapilar(x,y): Suprimir: `manovacia, libre(x), sobre(x,y)`

Añadir: `cogido(x), libre(y)`

Supón que quieres planificar para pasar del estado inicial al estado final que aparecen en la figura. ¿Que ejemplos resolverías primero para hacer este problema más eficientemente? Genera la tabla triangular generalizada para estos problemas e incluye el resultado como otros operadores más. Intenta resolver el problema inicial con estos nuevos operadores y generaliza también esta resolución.

Los predicados con los que se puede describir una escena son:

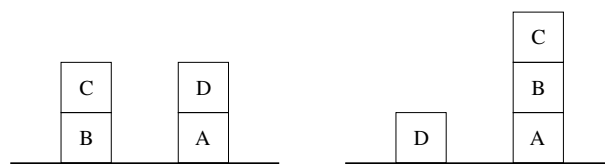
manovacia: La mano del robot esta libre

sobre(x,y): El objeto x está sobre el objeto y

libre(x): El objeto x no tiene nada encima

cogido(x): El objeto x esta cogido por la mano del robot

en_mesa(x): El objeto x esta sobre la mesa



11. Supón que podemos describir escenas del mundo de los bloques que contienen una mano de robot, bloques y una mesa. Estos bloques pueden estar sobre la mesa o sobre otro bloque. Tomemos los siguientes operadores de planificación:

Operador	coger(x)	dejar(x)
Prec.	en_mesa(x), libre(x), manovacia	cogido(x)
Supr.	en_mesa(x), libre(x), manovacia	cogido(x)
Añadir	cogido(x)	en_mesa(x), libre(x), manovacia
Operador	apilar(x,y)	desapilar(x,y)
Prec.	cogido(x), libre(y)	manovacia, libre(x), sobre(x,y)
Supr.	cogido(x), libre(y)	manovacia, libre(x), sobre(x,y)
Añadir	manovacia, sobre(x,y), libre(x)	cogido(x), libre(y)

Los predicados con los que se puede describir una escena son:

manovacia: La mano del robot esta libre	cogido(x): El objeto x esta cogido por la mano del robot
sobre(x,y): El objeto x está sobre el objeto y	en_mesa(x): El objeto x esta sobre la mesa
libre(x): El objeto x no tiene nada encima	

- a) Resuelve el problema de la figura y generaliza la resolución usando aprendizaje EBL.



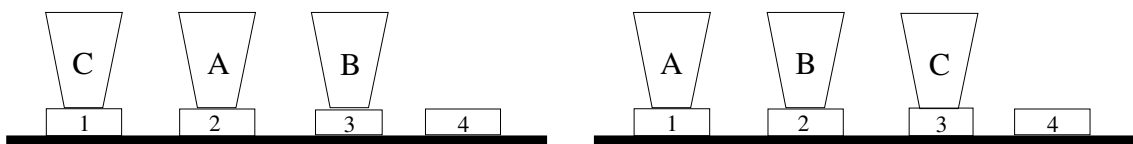
- b) Supón que quieres planificar para pasar del estado inicial al estado final que aparecen en la figura. ¿Que ejemplos resolverías primero para hacer este problema más eficientemente? Genera la tabla triangular generalizada para estos problemas e incluye el resultado como otros operadores más. Intenta resolver el problema inicial con estos nuevos operadores y generaliza también esta resolución.



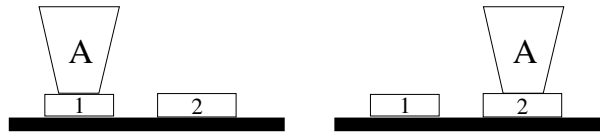
12. Queremos trabajar en un mundo parecido al de los bloques en el que tenemos vasos, posavasos y la mano de un robot. Para manipular en este mundo disponemos de dos operadores *coger(x,y)* y *dejar(x,y)*

Operador	coger(x,y)
Prec.	vaso(x), posavasos(y), sobre(x,y), manovacia
Supr.	sobre(x,y), manovacia
Añadir	cogido(x), libre(y)
Operador	dejar(x,y)
Prec.	vaso(x), posavasos(y), libre(y), cogido(x)
Supr.	cogido(x), libre(y)
Añadir	sobre(x,y), manovacia

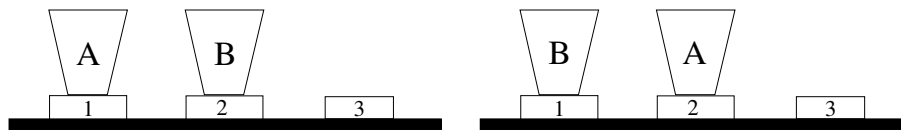
Supón que queremos resolver el problema que aparece a continuación:



Para resolverlo más sencillamente vamos a plantearnos primero resolver mover un vaso de un posavasos a otro. Resuelve el problema de la figura siguiente y aprende a partir de él el operador `mover(x,y,z)` que mueve el vaso `x` del posavasos `y` al posavasos `z`. Resuelve el problema inicial usando este nuevo operador.



Supongamos que queremos un operador mas potente que intercambie los vasos entre dos posavasos usando un tercero como soporte temporal. Resuelve el problema de la figura siguiente usando el operador `mover(x,y,z)` para aprender en operador `intercambiar(a,b,x,y,z)` que intercambia el vaso `a` que esta sobre el posavasos `x` con el vaso `b` que esta sobre el posavasos `y`. Usa este nuevo operador para resolver otra vez el problema inicial.



Comenta las diferencias entre las dos soluciones. ¿Tiene ventajas el operador más complejo respecto al más simple? ¿Porque?

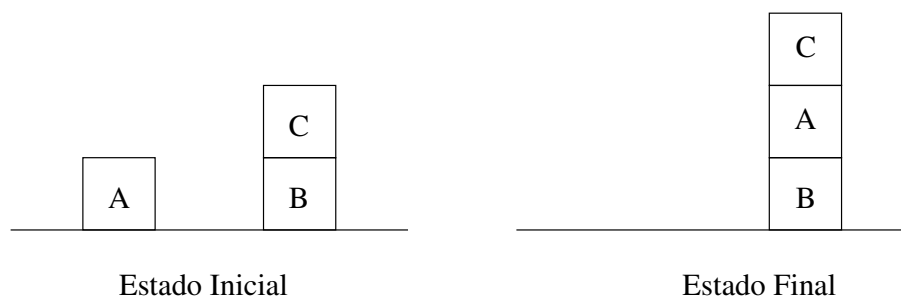
- Supón que podemos describir escenas del mundo de los bloques que contienen una mano de robot, bloques y una mesa. Estos bloques pueden estar sobre la mesa o sobre otro bloque. Tomemos los siguientes operadores de planificación:

Operador	coger(x)	dejar(x)
Prec.	<code>en_mesa(x)</code> , <code>libre(x)</code> , <code>manovacía</code>	<code>cogido(x)</code>
Supr.	<code>en_mesa(x)</code> , <code>libre(x)</code> , <code>manovacía</code>	<code>cogido(x)</code>
Añadir	<code>cogido(x)</code>	<code>en_mesa(x)</code> , <code>libre(x)</code> , <code>manovacía</code>
Operador	apilar(x,y)	desapilar(x,y)
Prec.	<code>cogido(x)</code> , <code>libre(y)</code>	<code>manovacía</code> , <code>libre(x)</code> , <code>sobre(x,y)</code>
Supr.	<code>cogido(x)</code> , <code>libre(y)</code>	<code>manovacía</code> , <code>libre(x)</code> , <code>sobre(x,y)</code>
Añadir	<code>manovacía</code> , <code>sobre(x,y)</code> , <code>libre(x)</code>	<code>cogido(x)</code> , <code>libre(y)</code>

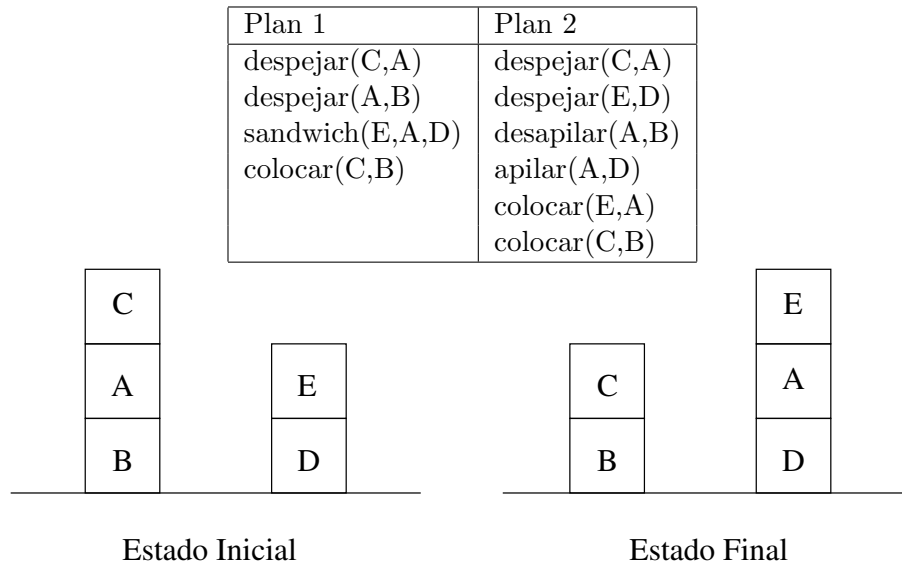
Los predicados con los que se puede describir una escena son:

manovacía: La mano del robot esta libre	cogido(x): El objeto <code>x</code> esta cogido por la mano del robot
sobre(x,y): El objeto <code>x</code> está sobre el objeto <code>y</code>	en_mesa(x): El objeto <code>x</code> esta sobre la mesa
libre(x): El objeto <code>x</code> no tiene nada encima	

- Queremos resolver el problema de la figura, pero para ello primero queremos aprender dos operadores, uno llamado **despejar(x,y)**, que quita un bloque de encima de otro y lo deja en la mesa y otro llamado **colocar(x,y)** que hace la operación inversa, pone un bloque que hay en la mesa encima de otro. Indica las unificaciones de variables que hagas durante el aprendizaje y por que las haces. No simplifiques los operadores despues del aprendizaje.



- b) Una vez aprendidos estos dos operadores aplícalos para resolver el problema de la figura y aprende con el plan un operador que se llame **sandwich(x,y,z)**. ¿Es mas general este operador que el plan original?
- c) Dados los planes de la tabla, que resuelven el problema de la siguiente figura, ¿Es mas ventajoso uno que el otro? Compáralos respecto al número de pasos primitivos y el número de precondiciones que hay que comprobar, y comenta los resultados.



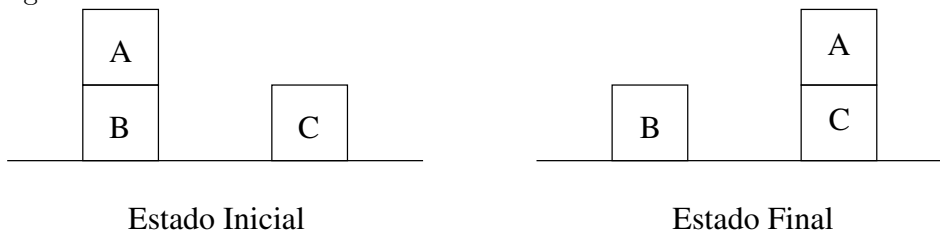
14. Supón que podemos describir escenas del mundo de los bloques que contienen una mano de robot, bloques y una mesa. Estos bloques pueden estar sobre la mesa o sobre otro bloque. Tomemos los siguientes operadores de planificación:

Operador	apilar(x,y)	desapilar(x,y)
Prec.	cogido(x), libre(y)	manovacía, libre(x), sobre(x,y)
Supr.	cogido(x), libre(y)	manovacía, libre(x), sobre(x,y)
Añadir	manovacía, sobre(x,y), libre(x)	cogido(x), libre(y)

Los predicados con los que se puede describir una escena son:

<p>manovacía: La mano del robot está libre</p> <p>sobre(x,y): El objeto x está sobre el objeto y</p> <p>libre(x): El objeto x no tiene nada encima</p>	<p>cogido(x): El objeto x está cogido por la mano del robot</p> <p>en_mesa(x): El objeto x está sobre la mesa</p>
---	---

- a) Queremos aprender el operador **mover(x,y,z)** que pasa un bloque de una pila a otra como aparece en la figura:

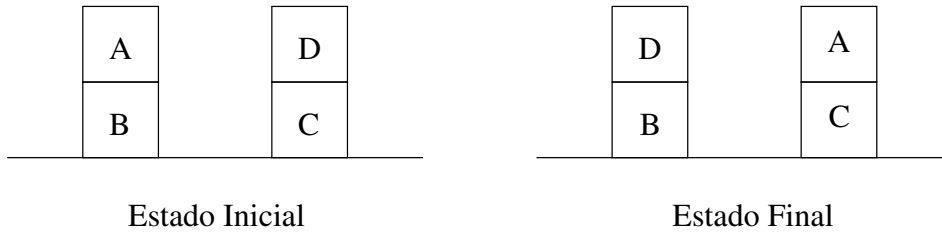


No hace falta que useis el algoritmo de planificación para hallar el plan, lo podeis hacer a ojo. ¿Es este plan más general que el original?

- b) Tenemos tambien los operadores **despejar(x,y)** y **colocar(x,y)** con la siguiente definición:

Operador	despejar(x,y)	colocar(x,y)
Prec.	sobre(x,y), manovacía, libre(x)	manovacía, en_mesa(x), libre(x), libre(y)
Supr.	sobre(x,y), manovacía, libre(x)	manovacía, en_mesa(x), libre(x), libre(y)
Añadir	manovacía, en_mesa(x), libre(x), libre(y)	sobre(x,y), manovacía, libre(x)

Ahora queremos aprender el operador **intercambiar(x,y,z,w)** usando los operadores mover, despejar y colocar. Al generalizar el plan os encontrareis que aparecen múltiples posibilidades a la hora de unificar variables, indicad que posibilidades aparecen, pero escoged las que os parezcan las correctas para el plan que se quiere obtener.



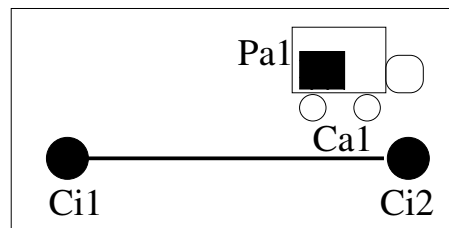
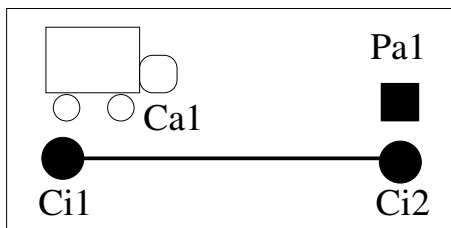
c) ¿Encontrais alguna relación entre las múltiples posibilidades que aparece a la hora de generalizar el plan y el haber usado sólo operadores no primitivos? ¿Que creis que se ha perdido al utilizar los operadores no primitivos que de alguna manera dificulta la generalización?

15. Tenemos un dominio en el que un camión traslada paquetes de una ciudad a otra, y disponemos de los siguientes predicados y operadores:

camion(c)	c es un camión
paquete(p)	p es un paquete
ciudad(l)	l es una ciudad
en(o,l)	el objeto o esta en el lugar l
carga(c,p)	el camion c carga el paquete p
conexion(l1,l2)	la ciudad l1 conecta con la ciudad l2

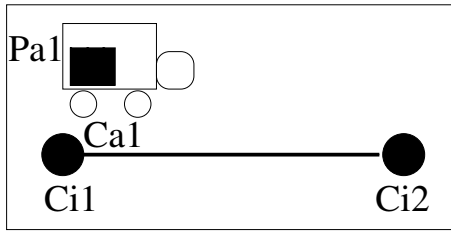
op:	cargar(c,p,l)	op:	descargar(c,p,l)
prec:	camion(c),paquete(p), ciudad(l),en(c,l),en(p,l)	prec:	camion(c),paquete(p),carga(c,p), ciudad(l),en(c,l)
add:	carga(c,p)	add:	en(p,l)
del:	en(p,l)	del:	carga(c,p)
op:	viajar(c,l1,l2)		
prec:	camion(c),ciudad(l1), ciudad(l2),en(c,l1),conexion(l1,l2)		
add:	en(c,l2)		
del:	en(c,l1)		

a) Dados los problemas de las figuras, generar un plan que los resuelva y crear un nuevo operador con la generalización de los planes:

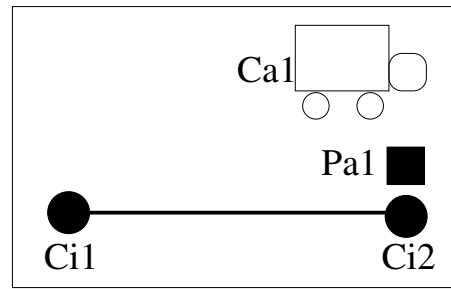


ciudad(ci1),ciudad(c2),camion(ca1)
paquete(pa1),en(ca1,ci1),en(pa1,c2)
conecta(ci1,ci2),conecta(ci2,ci1)

en(ca1,ci2),carga(ca1,pa1)

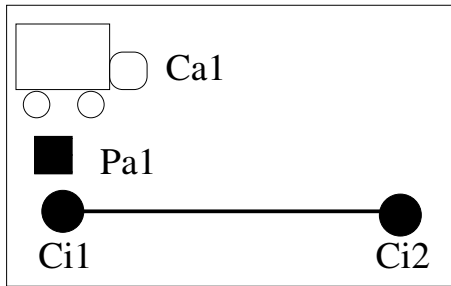


ciudad(ci1),ciudad(c2),camion(ca1)
paquete(pa1),en(ca1,ci1),carga(ca1,pa1)
conecta(ci1,ci2),conecta(ci2,ci1)

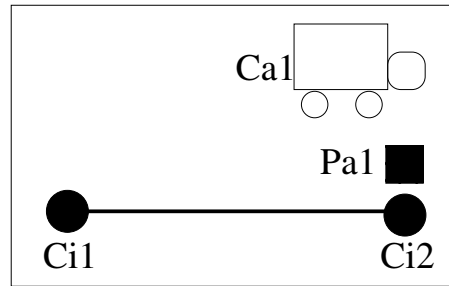


en(ca1,ci2),en(pa1,ci2)

- b) Dado este nuevo problema, generar un plan que lo resuelva y crear un nuevo operador con la generalización del plan. ¿Observas alguna particularidad en el plan generalizado?

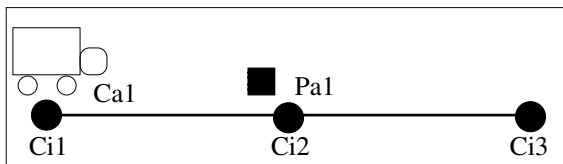


ciudad(ci1),ciudad(c2),camion(ca1)
paquete(pa1),en(ca1,ci1),en(pa1,c1)
conecta(ci1,ci2),conecta(ci2,ci1)

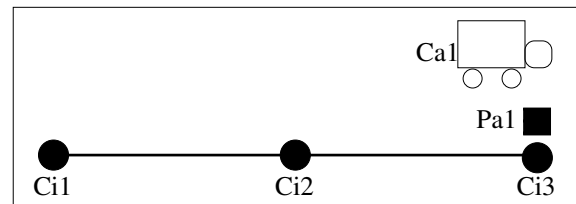


en(ca1,ci2),en(pa1,ci2)

- c) Resuelve este nuevo plan con los operadores originales, con los operadores aprendidos en el primer apartado y con el operador aprendido en el segundo apartado. Compara los tres planes obtenidos. Comenta la generalidad y utilidad de los operadores aprendidos.



ciudad(ci1),ciudad(c2),ciudad(c3)
camion(ca1),paquete(pa1)
en(ca1,ci1),en(pa1,c1)
conecta(ci1,ci2),conecta(ci2,ci1)
conecta(ci2,ci3),conecta(ci3,ci2)



en(ca1,ci3),en(pa1,ci3)

16. Tenemos un vector con n posiciones donde en cada una de ellas tenemos una letra, disponemos de un operador que nos permite intercambiar dos posiciones consecutivas del vector definido de la siguiente manera:

Operador	swap(x,y,vx,vy)
Prec.	next(x,y), val(x,vx), val(y,vy)
Supr.	val(x,vx), val(y,vy)
Añadir	val(x,vy), val(y,vx)

Los predicados con los que se puede describir una escena son:

next(x,y): La posición y es la siguiente a la x
val(x,vx): la posición x tiene el valor vx

a) Genera el plan que lleva del vector de la izquierda al de la derecha utilizando el operador **swap** (basta con indicar que pasos hacen falta).



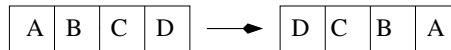
Aprende el operador **back** generalizando el plan generado usando la técnica de EBL utilizada por STRIPS. Indica la tabla triangular inicial justificando que predicados son los relevantes, la tabla generalizada y la tabla restringida con las unificaciones de variables que haces y por que.

b) Genera el plan que lleva del vector de la izquierda al de la derecha utilizando el operador **swap** (basta con indicar que pasos hacen falta).



Aprende el operador **mirror** generalizando el plan generado usando la técnica de EBL utilizada por STRIPS. Indica la tabla triangular inicial justificando que predicados son los relevantes, la tabla generalizada y la tabla restringida con las unificaciones de variables que haces y por que.

c) Dados los planes de la tabla, que resuelven el problema de la figura, ¿Es alguno mas ventajoso uno que el otro? Compáralos respecto al número de pasos primitivos y el número de precondiciones que hay que comprobar, y comenta los resultados.



Plan 1	Plan 2	Plan 3	Plan 4
swap(3,4,C,D)	back(2,3,4,B,C,D)	mirror(1,2,3,A,B,C)	swap(3,4,C,D)
swap(2,3,B,D)	swap(1,2,A,D)	swap(3,4,A,D)	back(1,2,3,A,B,D)
swap(1,2,A,D)	back(2,3,4,A,B,C)	mirror(1,2,3,C,B,D)	mirror(2,3,4,A,B,C)
swap(3,4,B,C)	swap(3,4,A,B)	swap(2,3,B,C)	
swap(2,3,A,C)			
swap(3,4,A,B)			

17. Tenemos un vector con n posiciones donde en cada una de ellas tenemos una letra, disponemos de dos operadores, el primero (swap) nos permite intercambiar dos posiciones consecutivas del vector, el segundo (mirror) nos permite intercambiar dos posiciones separadas una posicion. Estan definidos de la siguiente manera:

Operador	swap(x,y,vx,vy)
Prec.	next(x,y), val(x,xv), val(y,vy)
Supr.	val(x,vx), val(y,vy)
Añadir	val(x,vy), val(y,vx)
Operador	mirror(x,y,z,vx,vz)
Prec.	next(x,y), next(y,z), val(x,xv), val(z,vz)
Supr.	val(x,vx), val(z,vz)
Añadir	val(x,vz), val(z,vx)

Los predicados con los que se puede describir una escena son:

next(x,y): La posición y es la siguiente a la x
val(x,vx): la posición x tiene el valor vx

Aprende el plan [swap(1,2,A,B) → mirror(1,2,3,B,C) → swap(2,3,A,B)] que lleva del vector de la izquierda al de la derecha



Aprende el plan generando un operador con él, usando la técnica de EBL utilizada por STRIPS. Indica la tabla triangular inicial justificando que predicados son los relevantes, la tabla generalizada y la tabla restringida con las unificaciones de variables que haces y por que.

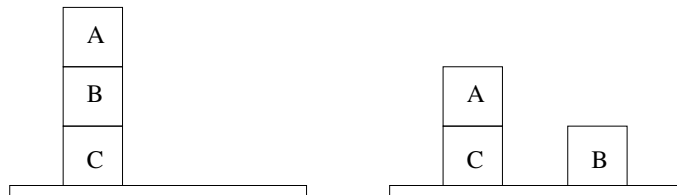
18. Supón que podemos describir escenas del mundo de los bloques que contienen una mano de robot, bloques y una mesa. Estos bloques pueden estar sobre la mesa o sobre otro bloque. Tomemos los siguientes operadores de planificación:

Operador	despejar(x,y)	colocar(x,y)
Prec.	sobre(x,y), manovacia, libre(x)	manovacia, en_mesa(x), libre(x), libre(y)
Supr.	sobre(x,y), manovacia, libre(x)	manovacia, en_mesa(x), libre(x), libre(y)
Añadir	manovacia, en_mesa(x), libre(x), libre(y)	sobre(x,y), manovacia, libre(x)

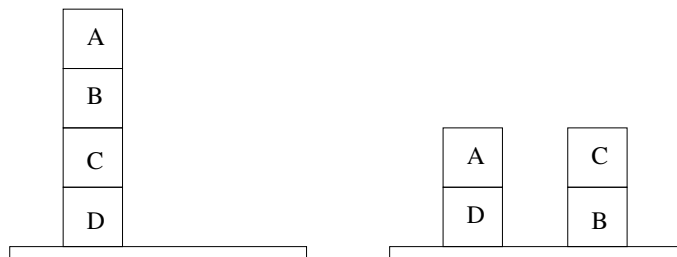
Los predicados con los que se puede describir una escena son:

manovacia: La mano del robot esta libre	cogido(x): El objeto x esta cogido por la mano del robot
sobre(x,y): El objeto x está sobre el objeto y	en_mesa(x): El objeto x esta sobre la mesa
libre(x): El objeto x no tiene nada encima	

- a) Queremos aprender el operador **extrae(x,y,z)** que quita un bloque de entre otros dos. Indica las unificaciones de variables que hagas durante el aprendizaje y por que las haces. No simplifiques los operadores después del aprendizaje, si el nuevo operador borra algo del estado original y luego lo vuelve a añadir dejalo así.



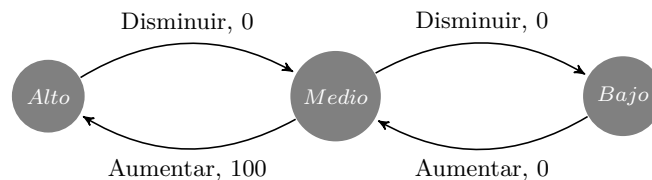
- b) Una vez aprendido este operador aplícalo (junto con el resto de operadores del dominio) para resolver el problema de la figura y aprende con el plan un operador que se llame **malabares(x,y,z,w)**.



- c) Compara el plan obtenido en el apartado anterior, el operador aprendido con ese plan y el plan [despejar(A,B), despejar(B,C), despejar(C,D), colocar(C,B), colocar(A,D)]. ¿Se obtiene un ahorro respecto al número de pasos primitivos aplicados? ¿Y respecto al número de precondiciones que se han de comprobar? Comenta los resultados.

Aprendizaje por refuerzo

1. Queremos construir un sistema capaz de controlar un proceso con tres estados {Alto, Medio, Bajo} donde se pueden realizar las acciones **Aumentar** y **Disminuir**. El siguiente gráfico indica como es la función de cambio de estado (δ) y el refuerzo (r) obtenido por cada acción:

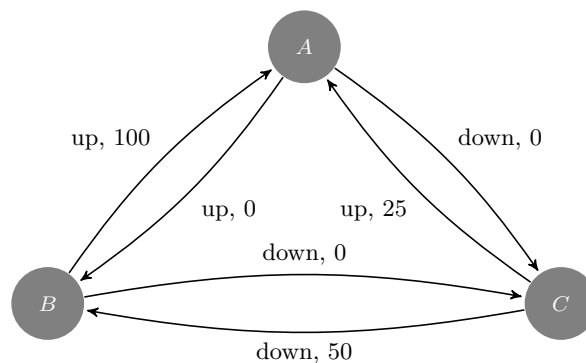


Supondremos que para entrenar el sistema aprenderemos a partir de tres secuencias de entrenamiento:

- a) Estado Inicial=Alto, acciones={Disminuir,Aumentar,Disminuir}
- b) Estado Inicial=Bajo, acciones={Aumentar,Aumentar,Disminuir}
- c) Estado Inicial=Medio, acciones={Disminuir,Aumentar,Aumentar}

Calcula la función \hat{Q} que obtendríamos después de aprender estas tres secuencias mediante el algoritmo de Q-Learning. Muestra la evolución de la función \hat{Q} paso a paso.

2. (1.5 puntos) Queremos construir un sistema capaz de controlar un proceso con tres estados {A, B, C} donde se pueden realizar las acciones **up** y **down**. El siguiente gráfico indica como es la función de cambio de estado (δ) y el refuerzo (r) obtenido por cada acción:



Supondremos que para entrenar el sistema aprenderemos a partir de tres secuencias de entrenamiento:

- a) Estado Inicial=A, acciones={up,up,up}

b) Estado Inicial=B, acciones={down,up,down}

c) Estado Inicial=C, acciones={down,down,up}

Calcula la función \hat{Q} que obtendríamos después de aprender estas tres secuencias mediante el algoritmo de Q-Learning considerando que el valor del parámetro γ es 0.9. Muestra la evolución de la función \hat{Q} paso a paso.