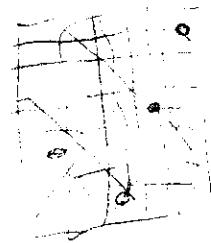
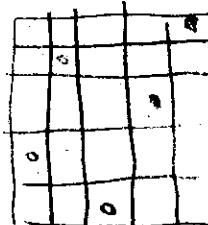


Reparación hecha

Incremento



IA PEI-3 mayo 2019

OBSERVACIONES:

El examen es escrito, no oral. Las respuestas deben ser razonadas, legibles e inteligibles por cualquiera. Los correctores no harán intentos de adivinación de lo que se pensó o se quiso decir pero no se escribió.

Escriba sus datos apellidos, nombre DNI y correo en la primera página. No separe las hojas grapadas. De necesitar más papel, agotado el inicial, se le proporcionará.

T1 (6 puntos)

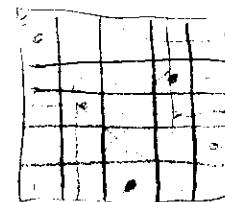
(a) Explique la relación existente entre los PSR y los problemas de búsqueda en espacio de estados.

(b) Razone qué particularidades tiene un PSR convertido en Problema de búsqueda.

(c) Explique las consecuencias de la parte (b) en los métodos de resolución a emplear.

T2 (7 puntos)

Formule como PSR y resuelva metódicamente el problema de las 5 reinas.



T3 (6 puntos)

(a) ¿Cuál es el motivo de la introducción de los subconjuntos difusos?

(b) ¿Puede considerarse todo subconjunto ordinario como uno difuso?

(c) Explique cuál es la estructura del álgebra $(\mathcal{F}(U), \cup, \cap, c, \emptyset, U)$ de las partes difusas de un universo U , qué propiedades mantiene y cuales no comparada con el álgebra de Boole de los subconjuntos ordinarios.

$$\begin{matrix} \cup(A) & + & = & \emptyset \\ \cap(A) & \cap & = & A \end{matrix}$$

T4 (6 puntos)

(a) Dado el universo de los números enteros $U = \{1, 2, 3, 4, 5, 6\}$, defina subconjuntos difusos P, G y M adecuados representar para los predicados "pequeño", "grande" y "mediano" en dicho contexto.

(b) Calcule los subconjuntos $P \cap M, G \cup M^c, M \cap M^c$ (la c volada denota complementación)

(c) Defina una relación difusa que pueda representar la relación de igualdad aproximada en U .

L1 (4 puntos)

$$A_{i,j} \neq n \quad (X_{i+n,j+n} = X_{i,j}) \quad n \in \mathbb{N}$$

Razone si las siguientes expresiones Racket son correctas, a qué argumentos se pueden aplicar sin dar error y cuáles serían sus resultados en tal caso:

(define (alfa x) (cond ((integer? x) (/ (* x x) x)) ((boolean? x) (not (not x))) (else x)))	(define (beta? y) (define (gamma? z) (if (<= z 1) (if (= 0 (modulo y z)) #f (gamma? (- z 1))) #t)) (gamma? (- y 1))))
--	--

L2 (5 puntos)

Diseñe una función Racket que tenga como entrada un número natural dado por su expresión decimal, calcule la suma de sus cifras y reitere el proceso hasta llegar a un número de una sola cifra, que será el resultado..

A1. Autocorrección (voluntario):

Escriba la calificación que considere merece cada una de sus respuestas.

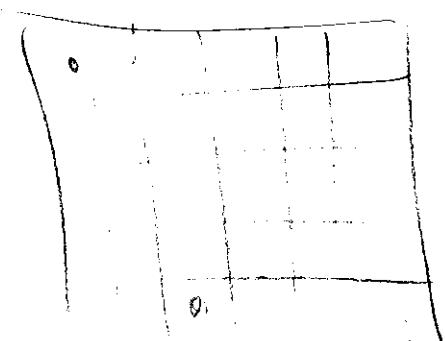
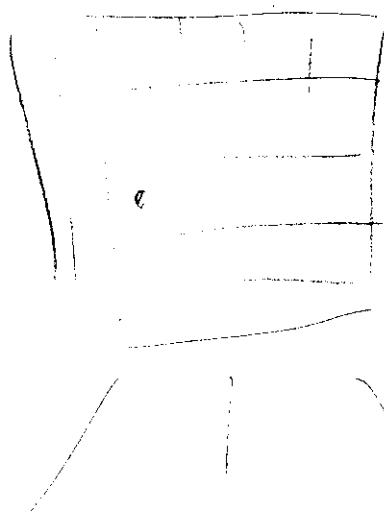
Si la suma se desvía menos de un punto de la nota total que dé el corrector, tendrá 1 punto suplementario en el examen. Si se desvía más de dos puntos, perderá 0.5 puntos de lo obtenido con las preguntas anteriores.

- 1) Tome la primera de la lista abierta, ~~sol 1mo~~
- 2) Generar sucesores prestando atención a la lista de dominios

$$\begin{array}{r} 735 \\ 35 \quad 73 \\ \hline 5 \end{array}$$

$$\begin{array}{r} 735 \\ 735 \quad 735 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 735 \\ 35 \quad 7 \\ \hline 100 \end{array}$$



$$4321 \begin{array}{r} 1000 \\ 321 \quad 4 \\ \hline 321 \end{array} \begin{array}{r} 100 \\ 21 \quad 3 \\ \hline 0 \end{array}$$

(def (sousc num cif)

if cifra = 0 \rightarrow return num /
 cifra = 2 \rightarrow (num / ~~10~~^{cif-1}) +
 sousc (num % 10^{cif-1}) cif-1

IA PEI-1 febrero 2019

OBSERVACIONES:

- *El examen es escrito, no oral. Las respuestas deben ser razonadas, legibles e intellegibles por cualquiera. Los correctores no harán intentos de adivinación de lo que se pensó o se quiso decir pero no se escribió.*
- *Escriba sus datos apellidos, nombre DNI y correl en la primera página. No separe las hojas grapadas. De necesitar más papel, agotado el inicial, se le proporcionará.*

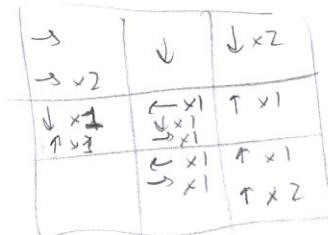
T1 (2 puntos). Describa qué disciplinas científicas o técnicas han intervenido en el desarrollo de la Inteligencia Artificial y las fases principales de dicho desarrollo.

T2 (2 puntos). En el método de *Búsqueda bidireccional*, en principio, las dos búsquedas alternadas a realizar se considera que lo son en anchura. (a) Razone si es un método completo. (b) ¿Podría desarrollarse alguna o ambas búsquedas mediante algún otro método distinto al de anchura con resultados satisfactorios?

T3 (2 puntos). Explique las razones que motivan el desarrollo del método de *Búsqueda en Profundidad Iterativa* como variante del de *Búsqueda en Profundidad*. Describa el método y analice sus características.

T4 (4 puntos). En el suelo de un parque de juego para niños se dibuja un casillero 3x3, orientado de norte a sur, en el que sobre cada casilla se escriben las instrucciones de los movimientos permitidos para cada participante situado en ella, consistentes en una dirección y el número de casillas a recorrer:

Este 1 Este 2	Sur 1	Sur 2
Sur 1 Norte 1	Oeste 1 Sur 1 Este 1	Norte 1
	Oeste 1 Este 1	Norte 1 Norte 2



Se plantea el juego de colocarse en el centro y volver a él tras hacer un recorrido de al menos un paso que resulte lo más corto posible.

(a) Formule el problema como uno de búsqueda en espacio de estados. (b) Explique con qué métodos se puede resolver. (c) Resuélvalo metódicamente con uno de ellos.

L1(2 puntos) Razone si las siguientes expresiones Racket son correctas y diga su resultado en tal caso: (a) '(quote ('a 'b 'b 'b 'c '1)) ; (b) (cond (and (or #t #f) (and (#t #t)))) ;

(c) '(cond (and (or #t #f) (and (#t #t)))) ; (d) (car (list (+ 1 2) '((- 1 2) '1 '2)))

→ *la respuesta*

En 3

L2(2 puntos) Diseñe una función Racket que tenga como entrada una lista cualquiera y un símbolo, produzca la lista resultante de eliminar el símbolo de inicial:

> (quitar '(a b c c d) 'b)
(list 'a 'c 'c 'd)

car
cdr

A1. Autocorrección (voluntario):

Escriba la calificación que considere merece cada una de sus respuestas.

Si la suma se desvía menos de un punto de la nota total que dé el corrector, tendrá 1 punto suplementario en el examen. Si se desvía más de dos puntos, perderá 0.5 puntos de lo obtenido con las preguntas anteriores.

N	Suc	Ab	N	Antes.	Ab.
5	4, 6, 8	4, 6, 8	5	2	2
4	1, 7	6, 8, 1, 7	2	1	1

Res: 5, 4, 1, 2

#t #f null ()

+ * / - sqrt sqrt log

sin cos asin acos pi e

not and or case if

define

list cons append car cdr

first second last

empty? member? null? list?

filter reverse length list-ref

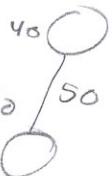
```
( let*
  (x 5)
  (y (+ x x)) = (y 2)
  (z (+ x x y)) = (z -4)
)
```

$$\frac{h_1 + h_2}{2}$$

IA PEI-2 Marzo de 2019

El examen es escrito, no oral. Las respuestas deben ser razonadas, legibles e inteligibles por cualquiera. Los correctores no harán intentos de adivinación de lo que se pensó o se quiso decir y no se escribió.

Escriba sus datos: apellidos, nombre, DNI y correo en la primera página. No separe las hojas grapadas. De necesitar más papel, agotado el inicial, se le proporcionará.



T1 (3 puntos)

(a) Defina "heurística admisible" y "heurística consistente". (b) Explique su utilidad y diferencia. (c) Dadas dos heurísticas h_1 y h_2 admisibles y consistentes, se definen $h_3 = \frac{h_1+h_2}{2}$ y $h_4 = \max\{h_1, h_2\}$. Estudie si h_3 es consistente, si h_4 es admisible y si algunas de las cuatro pueden compararse entre sí por ser más o menos informadas.

T2 (4 puntos)

Sea un problema de búsqueda en espacio de estados en el que los estados son I(9), A(7), B(2), C(2), D(1) y M(0), donde el número entre paréntesis denota un valor heurístico. Las transiciones entre estados son IA(2), AC(2), CM(6), IB(6), BD(3) y DN(1), donde el número denota el coste correspondiente. Los estados inicial y meta son I, M respectivamente. (a) Estudie si la heurística es admisible, consistente, o ambas cosas. (b) Resuelva metódicamente mediante la búsqueda A*

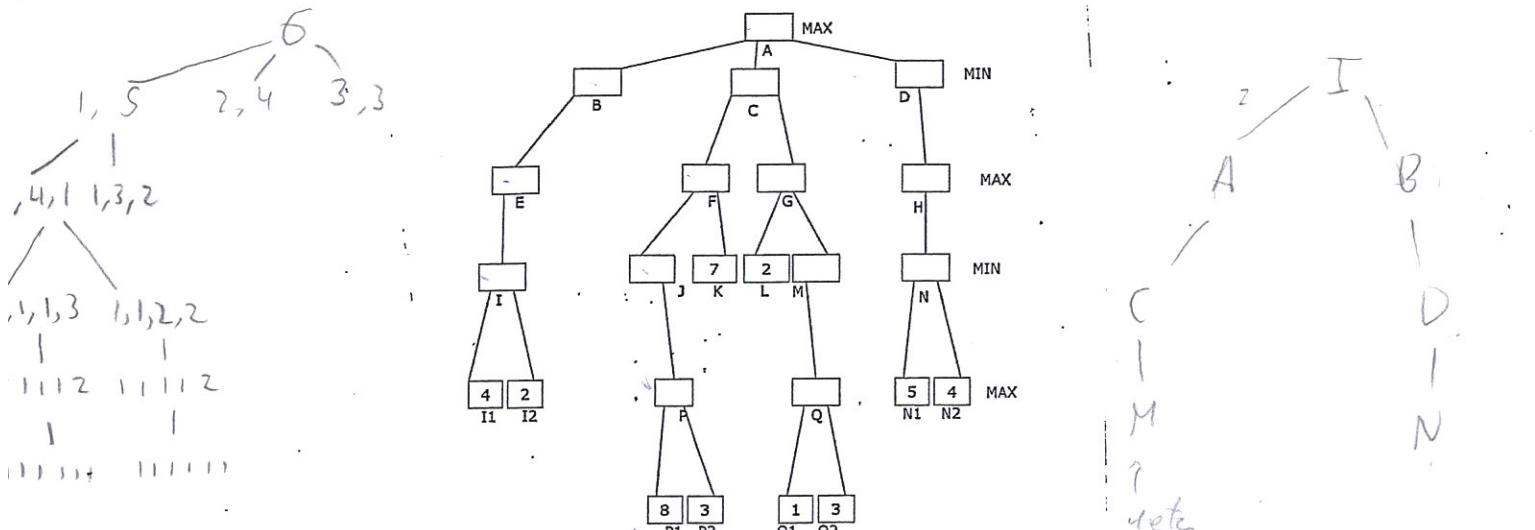
T3 (3 puntos)

En un juego de tipo Nim entre dos jugadores, se dispone inicialmente de un montón de seis piezas iguales sobre una mesa. En jugadas alternativas han de separar algún montón de los existentes sobre la mesa en dos montones de diferente tamaño. Perderá el primer jugador que no pueda hacerlo.

Determine las estrategias óptimas que sea posible usando el método de poda α - β .

T4 (5 puntos)

En un juego cuyo árbol, una vez desarrollado por completo según sus reglas, fuese el de la figura adjunta, determine las estrategias óptimas usando el método MiniMax y usando el método de poda α - β .



L1 (3 puntos)

Razone si son correctas las siguientes expresiones Racket y evalúe su resultado en caso de serlo:

((lambda (x) (expt x x)) 3/2) ((lambda (pi) (+ pi pi)) pi)

((let* ((x 1) (y (+ x x)) (z (+ x x y))) 1/3)

L1 (4 puntos)

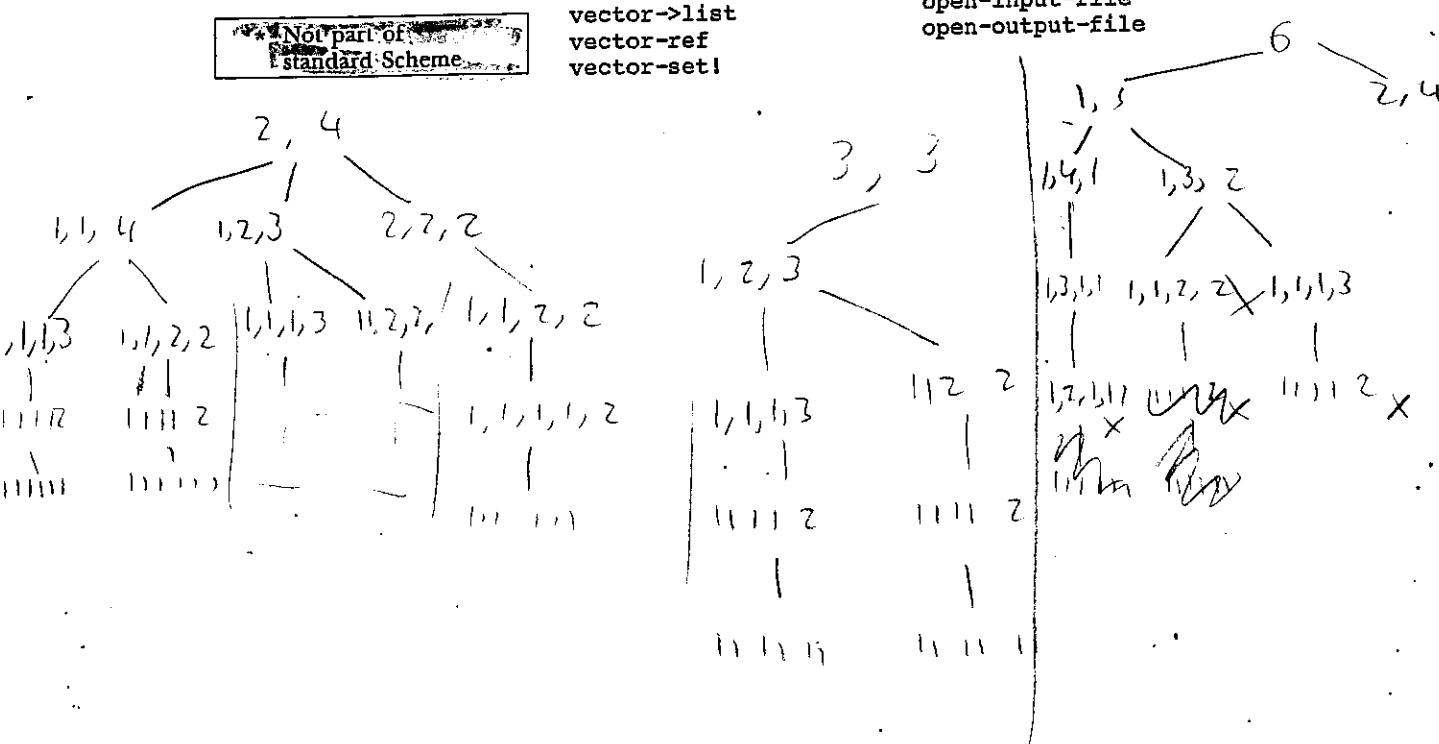
Construya un programa Racket que, dados una lista de símbolos y un símbolo, construya una nueva lista, resultante de eliminar la eventual primera aparición del símbolo en la lista original.

A1. Autoevaluación (voluntaria): Escriba la calificación que considere merece cada una de sus respuestas. Si la suma se desvía menos de un punto en la nota total que dé el corrector, tendrá 1 punto suplementario en el examen. Si se desvía más de dos puntos, perderá 1 punto de lo obtenido con las preguntas anteriores. En otro caso, no alterará la nota.

Table of Scheme Primitives by Category

Use this table if you've forgotten the name of a primitive. Then look in the index to find more about how to use the primitive.

Words and Sentences	Arithmetic	Procedures
appearances*	+, -, *, /	apply
before?*	<, <=, =, >, >=	lambda
butfirst (bf)*	abs	procedure?
butlast (bl)*	ceiling	
count*	cos	
empty?*	even?	
equal?	expt	
first*	floor	
item*	integer?	
last*	log	
member?*	max	
quote	min	
sentence (se)*	number?	
sentence?*	odd?	
word*	quotient	
word?*	random	
	remainder	
	round	
	sin	
	sqrt	
Lists	True and False	Control
append	and	begin
assoc	boolean?	error
car	cond	load
cdr	if	trace
c...r	not	untrace
cons	or	
filter*		
for-each		
length		
list		
list?		
list-ref		
map		
member		
null?		
reduce*		
Trees	Variables	Input/Output
children*	define	align*
datum*	let	display
make-node*		newline
*Not part of Standard Scheme.		
	Vectors	read
	list->vector	read-line*
	make-vector	read-string*
	vector	show*
	vector?	show-line*
	vector-length	write
	vector->list	
	vector-ref	
	vector-set!	
Files and Ports		
		close-all-ports*
		close-input-port
		close-output-port
		eof-object?
		open-input-file
		open-output-file



① IA-TS. PSR

Un prob. de satisl. de restricciones viene determinado por:

- Variables $\{x_1, x_2, \dots, x_n\}$
- Dominios $\{D_1, D_2, \dots, D_n\}$
- Restricciones $\{C_1, C_2, \dots, C_n\}$

Un estado es una asignación a las variables.

$$\text{Ej. } E_2 = \{x_1 = v_i, x_2 = v_j, \dots, x_n = v_k\}$$

Estado consistente \rightarrow aquél que no viola restricciones

Estado completo \rightarrow aquél que da valores a todas las variables

Estado solución \rightarrow aquél que da valor a todas las variables y no viola restricciones.

No tiene por qué haber sólo un estado solución

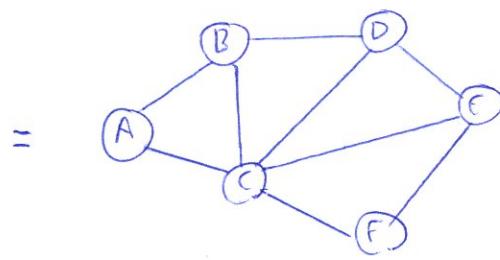
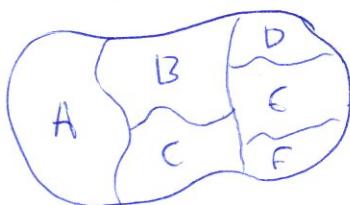
PSR's y Grados

Se puede representar un PSR. como: un grafo de restricciones.

Nodos = Variables

Aristas = restricciones

Ejemplo



⑥

⑥

Un PSR se le puede dar formulación incremental (es decir, tratarlo como un problema de búsqueda convencional) si posee:

- Estado inicial: asignando val.
- Función sucesor: que dado un nodo, devuelva sus sucesores.
- Test objetivo: que dado un nodo devuelva true si es la solución.
- Costo del camino: que dadas dos nodos, dev. el coste entre ambos.

① Tipos de PSR

② Por tipo de dominio:

- Dominio discreto y finito: el más sencillo. Sea d el tam máx. del dominio de una variable, hay d^n combinaciones posibles. (n variables). Incluye PSR booleanos. Ej.: colores a elegir en una paleta de colores.
- Dominio discreto e infinito: en estos no se pueden definir restricciones indicando todos los valores, así que hay que usar lenguaje de restricciones. (Valores, como escribir un if.).
Ej. números naturales $\{1, 2, 3, \dots, n\}$
- Dominio continuo: aquellos con dominio no discretizado. Por ejemplo, problemas lineales de optimización.
Ej. $\mathbb{R}: \{1, 1.01, 1.11, \dots\}$ También pueden ser finitos o infinitos.

③ Por el número de variables en las restricciones

- Restr. unaria: restringe los valores de una var. $C_1 = \{x_1 \neq 1, x_1 \neq 2, \dots\}$
- Restr. binarias: restringe los valores de dos variables relacionadas. Puede representarse como un grafo de restricciones.
- Restricción n-aria: implican 3 o más variables. Se pueden representar con un hiper-grafo de restricciones. También se puede transformar a un conjunto de relaciones binarias introduciendo variables aux.

Ej. s prop. examen

2017

① Discip. q. han intervenido en IA y relaciones entre ellas:
Matemáticos: formalización y ev. histórica. Al tratar de formalizar la matemática se une con la lógica al tratar axiomas. Al estudiar la matemática en si misma se sigue desarrollando la lógica para poder formalizarla en sí misma con el diseño de la máquina de theorem. Se une con la física con el diseño de la máquina de Turing. De ahí salta a la teoría de computadoras con la máquina autónoma, etc. De ahí salta a la computabilidad y la cálculo. de Turing y los teoremas de computabilidad y la cálculo. También influyen los campos de psicología y neurociencias en problemas de redes neuronales, teoría de juegos, etc.

② Tipos de problemas resolubles mediante búsqueda
Cualq. problema que pueda ser traducido a un grafo de estados y que contenga est. inicial, estado(s) meta(s), y operaciones de cambio de estado. (Soluciones identificables).

②.5 Búsq. desinformada. Métodos y diff.

Algoritmo

- 1) ABIERTOS + est. ini
- 2) Sacar nodo de abiertos según método.
- 3) Es sol? ✓ Si → fin
No → meter sucesores en abiertos, según método. Saltar a 2).

Anchura: ABIERTOS = cola FIFO.
↳ Por niveles
Prof.: ABIERTOS = pila LIFO.
↳ Por ramas
Lmite: prol. limitada
Lmite iterativa: prol. sucesiv. limitada
Óptimo: ABIERTOS = cola prioridad
menor coste acumulado
pri. val.

Bidireccional: --

③ Prof. iterativa vs Anchura & Profund.

Ventajas vs anchura: si el factor de multiplicación es muy alto y la solución está a profundidad media garantí

Ventajas vs prof.: si r^t y sol por la derecha poco profunda, quan

Desv. vs anchura: si r^t y p^t tendré q. realizar sucesivas iteraciones y

será más lento.

Desv. vs prof: si r^t y p^t --, --, --, --, y s^t

Desv. vs prof: si r^t y p^t tendré modo más q. profundidad.

Sol. est. por Búsq. tardará mucho más q. profundidad.

④ Fichas: $\boxed{2 \mid 4 \mid 3 \mid 1}$

a) Formular prob.

Est. ini: $(2, 4, 3, 1)$

Est. fin: $(1, 2, 3, 4)$

$(4, 3, 2, 1)$

Operaciones:

Intercambiar $(x, y) \Rightarrow [a, b, c, d]$

$x \downarrow y$

\downarrow
[a, b, d, c]

Q: intercambiar $(1, 2) \Rightarrow [2, 4, ?, 1]$

\downarrow

[$4, 2, 3, 1$]

b) Métodos:

- Anchur ✓

- Prof ✓

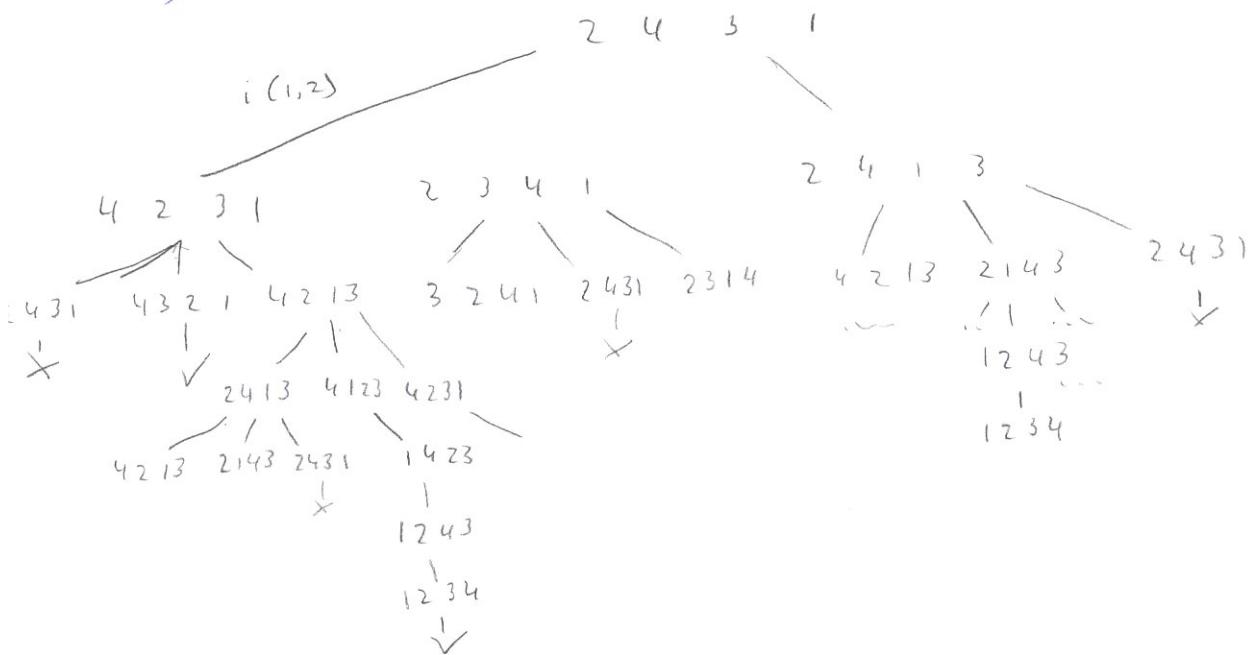
- Prof limit ?

- Prof iterat ✓

- Bidimenc ✓

- Optimal X \rightarrow no hay costes variables

c) Si realizan mas gato:



Anchur

1) A: $[(2, 4, 3, 1)]$

2) A: $[(4, 2, 3, 1), (2, 3, 4, 1), (2, 4, 1, 3)]$

3) A: $[(2, 3, 4, 1), (2, 4, 1, 3), (2, 4, 3, 1), (4, 3, 2, 1), (4, 2, 1, 3)]$

4) A: $[(2, 4, 1, 3), (2, 4, 3, 1), (4, 3, 2, 1), (4, 2, 1, 3), (3, 2, 4, 1), (2, 4, 3, 1), (2, 3, 1, 4)]$

Prof.

1) A: $[(2, 4, 3, 1)]$

2) A: $[(4, 2, 3, 1), (2, 3, 4, 1), (2, 4, 1, 3)]$

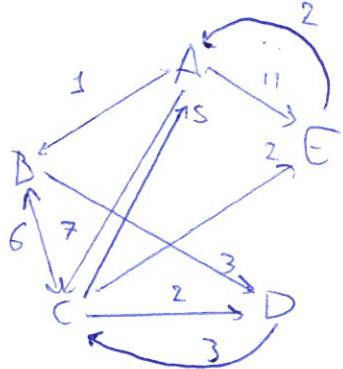
3) A: $[(2, 4, 3, 1), (4, 3, 2, 1), (4, 2, 1, 3), (2, 3, 4, 1), (2, 4, 1, 3)]$

4) A: $[(4, 2, 3, 1), (2, 3, 4, 1), (2, 4, 1, 3), (4, 3, 2, 1), (4, 2, 1, 3), (2, 3, 4, 1), (2, 4, 1, 3), \dots]$

d) Resolver: hacer por anchura q. terminar anfis

⑤ Optimal (desde cuándo el retro es gratuito?)

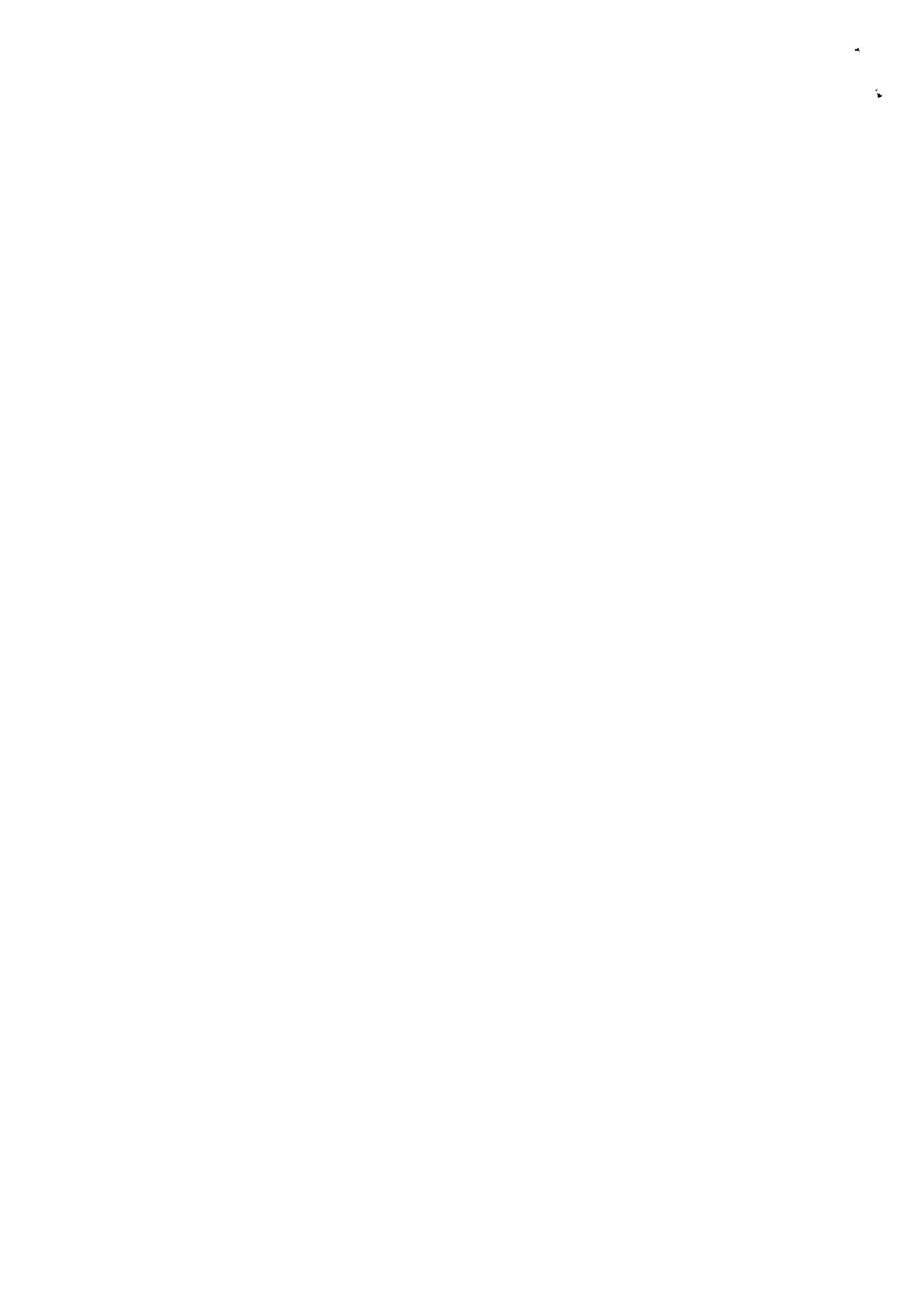
Resolver $A \rightarrow E$.



Δ - Coste acumulado

$[(\text{nodo}, \text{coste actual})]$

Estado	Sucesores	Accumulado
\emptyset	\emptyset	$[(A, 0)]$
$[(A, 0)]$	$(B, 1), (C, 7), (E, 11)$	$[(B, 1), (C, 7), (E, 11)]$; Mejor: $(A \rightarrow E, 11)$
$(B, 1)$	$(C, 6+1), (D, 1+3)$	$[(D, 4), (C, 7), (C, 7), (E, 11)]$
$(B, 4)$	$(C, 4+3)$	$[(C, 7), (C, 7), (C, 7), (G, 11)]$
$(C, 7)$	$(A, 7+5), (B, 7+6), (D, 7+3), (G, 7+2)$	$[(G, 9), (D, 10), (A, 12), (B, 13)]$; Mejor: $(A \rightarrow B \rightarrow C \rightarrow E, 9)$



Inteligencia Artificial

León A. González Sotelo N325 (Tut x 15-21h).

Lab: Álvaro Yagües Sandinos

T1: Intro histórica y conceptual

T2: Búsqueda demostrativa

T3: Búsqueda Heurística

T4: Prob. de satisfacción de restricciones

T5: Juegos

T6: Computación flexible

T7: Representación, razonamiento, aprendizaje

Evaluación

Ensayos

Trabajos { proyectos

• Inteligencia Artificial - TI

① Introducción, panorama histórico y conceptual

Inteligencia	Razonar	• Razonar: proceso mediante el cual, en función de unos parámetros,
	Resolver	en base a su conocimiento previo, información y heurísticas, se genera un resultado o se predice una consecuencia.
	Decisión	o se predice una consecuencia.
	Percepción	• Permite desarrollar cognición y conocimiento de forma autónoma.
	cognición	

② Formalización y ev. histórica

A partir del S. II las matemáticas se dividen en

Aritmética: números naturales y operadores.
Geometría: forma, espacio y sus conceptos.

En Grecia, gracias a su sistema merititivo y legal se asienta la matemática. Surge la lógica ante de la argumentación. (Postulado = Axioma = proposición indiscutible).

Aristóteles → "Organon" → Introduce el silogismo.

Euclides → "Elementos" → Organiza conocimientos con definiciones y axiomas.

Brouwer → Formaliza lógica y razonamiento de proposiciones.

Frege → Cálculo de predicados. Expresiones $p(x)$ ó $p(x,y)$ y operadores.

Hilbert: formalizar estructuras matemáticas para q. sean consistentes (no contradicción), completas (demostrable) y decidibles (comprobable recursiv.).

Gödel - Th. Incompletitud: ningún sistema formal q. exprese matemáticamente consistente y completo.

Turing - Th. Incapacidad: máquina de Turing.

Turing - Qué es computable y qué no. Máquina de Turing.

Church - λ-Cálculo: qué es y qué puede hacer un algoritmo.

Church - λ-Cálculo: qué es y qué puede hacer un algoritmo.

Toda func. computable puede hacerse con λ-Cálculo

Funciones Recursivas

③ Evolución en otros aspectos

Von Neumann.

• Ind. y Física: automatas, calculadoras mecánicas, neg. diferencial, ord.

• Juegos: teoría de juegos.

• Neurociencia y psicología: problema cuerpamente, redes neuronales, teoría de juegos.

Características de la IA

• Información simbólica > Información numérica

• Método lógico heurística

• Conocimiento específico-declarativo

• Info. imperfecta

④ Inteligencia Artificial - TI

④ Concepto de Inteligencia

Lo computable es lo que se puede "algoritmar".
Def. Inteligencia: la habilidad de llevar a cabo pensamiento abstracto

- cuáles funciones implican:
 - Razonar
 - Percibir } La realidad
 - Entender }
- | - Resuelve problemas y utilizan conocimientos previos.
- | - Aprendizaje
- | - Adaptación al medio o circunstancias.

Def. 2: La inteligencia es lo q. se mode en un agente cuando se percibe su entorno y actúa sobre él adecuándose a unas fines.
Def. 3: Proceso de, en base a parámetros, conocimientos y heurística, generar o predecir consecuencias y/o soluciones.

⑤ Inteligencia artificial, fines y medios. Tipos

- Sistemas que piensan como humanos:
 - Ciencia Cognitiva: se estudia como funciona el pensamiento humano, para emularlo con máquinas.
- Sistemas q. actúan como humanos:
 - Comportamiento humano: el Test de Turing consiste en q. un humano no sepa diferenciar si su interlocutor es un humano o máquina.
- Sistemas q. piensan racionalmente: emplean silogismos. Si las premisas son correctas, se llega a sol. correctas. Los sistemas inteligentes utilizan prog. codificadas en not. lógica, que en teoría resuelven tod. (no en la práct.)
- Sistemas q. actúan racionalmente: Un agente racional es aquel que actúa intentando encontrar el mejor resultado posible esperado. La IA. trata de construir agentes artificiales q. intenten emular el comportamiento de agentes naturales.
- Paradigma simbólico
 - Newell y Simon - 1976: concepción similar a maquinaria de Turing.
- Paradigma conexionista
 - El comp. inteligente puede surgir a partir de entregar redes neuronales artificiales y el aprendizaje de estas en procesos q. mitan la ev. adaptativa

⑥ Aplicaciones y logros

- Planificación autónoma de operaciones y tareas
- Juegos, p. ej. ajedrez
- Diagnóstico: análisis probabilístico.
- Plan. logísticas
- Robótica: fusión de computación y elem. robóticos gobernado por una IA.
- Proc. del lenguaje: traductores, compiladores...

⑦ Campos de la IA

- Lógica: toda acción de un proy. está representada en el leng. matemático - lógico. El programa debe inferir qué acciones hacer para llegar a su objetivo.
- Búsqueda: al tener q. examinar un gran n.º de posibilidades, se deben optimizar las búsquedas por esas posibilidades.
- Reconocimiento de patrones: cuando un programa hace observaciones, trata de buscar patrones para comprobar soluciones o el problema a atacar.
- Representación: los hechos se deben representar. Normalmente en lenguaje matemático - lógico.
- Inferencia: a partir de unos hechos se pueden deducir otros.
- Sentido común y razonamiento:

• IA - T4 - Juegos

Se da un juego cuando varios agentes tienen objetivos en conflicto.

Algoritmo min max

MAX → elige camino que lleva a nodo con mayor valor heurístico.

MIN → elige camino que lleva a nodo con menor valor heurístico.

Se van turnando. Siempre empieza Max.

ID - UAH - T2 - Búsqueda de información

① Formulación de problemas en espacios de estados

El agente tiene unos fines o metas y unos estados iniciales, que representan abstractamente los datos del problema. Busca llegar de unos a otros utilizando operadores que permiten cambiar de un estado a otro, es decir, generando posibles estados sucesores.

Pasos

- 1) Definir espacio o dominio de estados.
- 2) Indicar estado inicial o de partida.
- 3) Indicar estados meta.
- 4) Especificar operadores de cambio de estado.
- 5) Indicar coste de operadores.
- 6) Criterios de identificación de soluciones.

Resolución de problemas: proceso de determinar la serie de acciones a seguir para llegar de un estado inicial a un estado meta, normalmente siguiendo el camino más óptimo.

Se puede utilizar un grafo.

- Nodos = estados
- Aristas = operadores.

② Procedimiento general de búsqueda de grafos

Input

- Grafo con
 - Nodos (estados)
 - Arcos (operadores)
- Nodos iniciales
- Función esMeta(nodo)

Función búsqueda

- Sucesores: lista de nodos sucesores de un nodo.
- ⇒
- Abiertos: lista/cola de nodos a explorar.

Output

- Camino de nodo(s) inicial(es) a nodo(s) meta(s).

Idea general

1) Situarse en estado inicial y comprobar si es solución

1') Si no lo es, generar sucesores y:

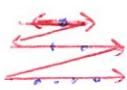
- 2) Si no lo es, generar sucesores y:
- i') Almacenarlos en lista abiertos como indica el método.
 - ii') Extraer en orden en el que indica el método.
 - iii') Examinar si es solución. Si no lo es, saltar a 2).

Las diferentes métodos de búsqueda se distinguen unos de otros por la manera de gestionar los nodos generados.

③ Principales métodos de búsqueda

Anchura

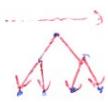
- ABIERTOS es una cola FIFO.



(Búsq. por niveles)

Profundidad

- ABIERTOS es una cola LIFO.



(Búsq. por ramas).

2 variantes

- Prof. acotada: explora hasta el nivel L .
- Prof. iterativa: explora hasta niveles distintos $L_1 < L_2 < L_3$.

Búsquda óptima

Se utilizan cuando el coste de las operaciones (avances) no es el mismo para toda -ABIERTOS es una lista de prioridad: el que tenga menor coste acumulado de operaciones primera.

Bidireccional

Utilizable si se conoce:

- Operadores inversos
- Estado (s) meta.

Realizar alg. de búsq. en anchura y profundidad a la vez, avanzando al unisono. (↓ iteración anchura, ↓ prof. ↓ anch., ↓ prof.). En cada cambio, verificar si los caminos de cada búsqüeda comparten algún nodo. Si es así, construir camino de origen (anchura) a destino (profundidad).

④ Características

- Completitud: encuentra sol. si la hay.
- Optimalidad: la sol. es la más cercana al estado de partida en pasos ↓ coste acumulado.
- Complejidad especial (heurística): estimación de tamaño de los listas
- Complejidad temporal (operaciones): estimación del n.º de op. hasta sol.

p = prof. de sol.

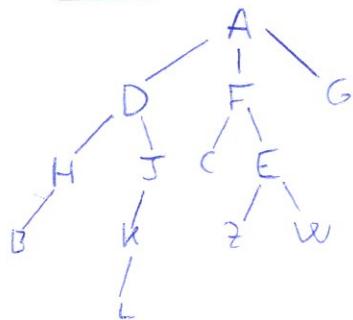
* = factor ramificación (n.º nodos sucesores)

l = Nodos est.

	Completo	Óptimo	C. Especial	C. Temp.
Anchura	Sí	Sí	$O(r^{p+1})$	$O(r^{p+1})$
Prof.	No *	No	$O(r \cdot p)$	$O(r^{p+1})$
Prof. Irm.	No	No	$O(r \cdot l)$	$O(r^{p+1})$
Prof. suc.	Sí	Sí	$O(r \cdot p)$	$O(r^{p+2})$
Optimal	Sí	Sí		$O(r^{p+1})$
Bidirecc.	Sí	Sí	$O(r^{p/2})$	$O(r^{p/2})$

① Ejemplos aplicación de algoritmos

Gráfico



Anchura - cola FIFO (append por la derecha)

Actual	Sucesores	Abiertos	Actual	Sucesores	Abiertos
		A	E	Z, W	B, K, +Z+W
A	D, F, G	D, F, G	B	∅	K, Z, W
D	H, J	F, G, H, J	K	L	Z, W, +L
F	C, E	G, H, J, C, E	Z	∅	W, L
G	∅	H, J, C, E	W	∅	L
H	B	J, C, E, B	L	∅	{}
J	K	C, E, B, K			
C	∅	E, B, K			

Profundidad - pila LIFO

Actual	Sucesores	Abiertos	Actual	Sucesores	Abiertos
		A	L	∅	F, G
A	D, F, G	D, F, G	F	C, E	C, C, G
D	H, J	H, J, F, G	C	∅	E, G
H	B	B, D, F, G	E	Z, W	B, W, G
B	∅	J, F, G	Z	∅	W, G
J	K	K, F, G	W	∅	G
K	L	L, F, G	G	∅	∅

Backtracking - cola L (append por la dg.)

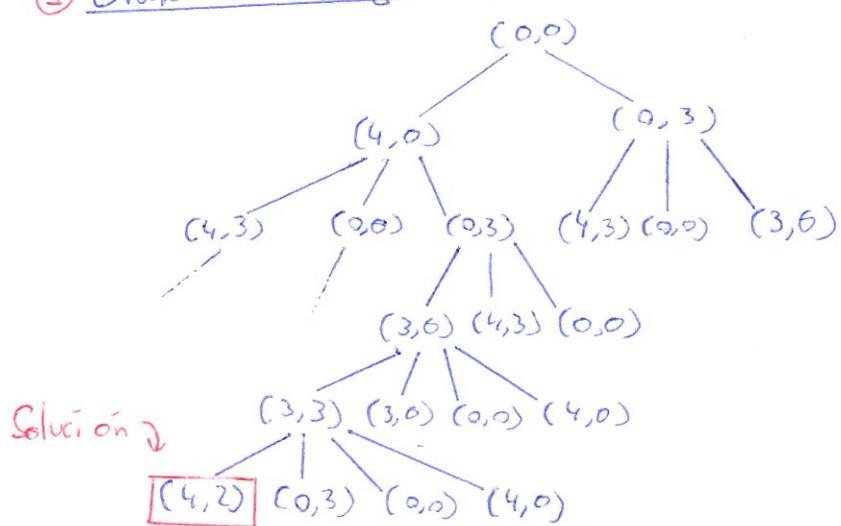
Actual	Sucesores	Abiertos
∅	∅	A
A	D, F, G	+D, +F, +G
D	H, J	+H, +J, F, G

Sea n = factor ramificación
Sea p = profundidad de la solución

Tipo	Temp.	Space
Busq ampliitud	$O(n^p)$	$O(n^p)$
Prof.	$O(n^p)$	$O(n^{p/p})$
Backtracking	$O(n^p)$	$O(p)$
Prof. progr.	$O(n^p)$	$O(p)$
Bidireccional	$O(n^{p/2})$	$O(n^{p/2})$

• Chubetilla - Búsqueda en anchura & Profundidad

① Grafo de las juntas ($J_1 = 4, J_2 = 3$)



B. Anchura: Recorrido níveis del grafo buscando solución. (Izq. a der)

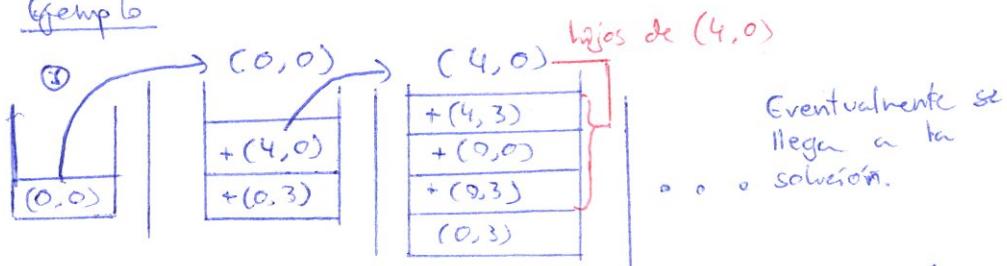
B. Profundidad: Recorrido ramas del grafo en busca de la solución (Izq. a der)

② Búsqueda en profundidad - Pila

Algoritmo

- ① Apilar $(0, 0)$
- ② Desapilar cima. Si cima = solución, parar.
- ③ Apilar hijos de cima de derecha a izquierda (primero derecha)

Ejemplo

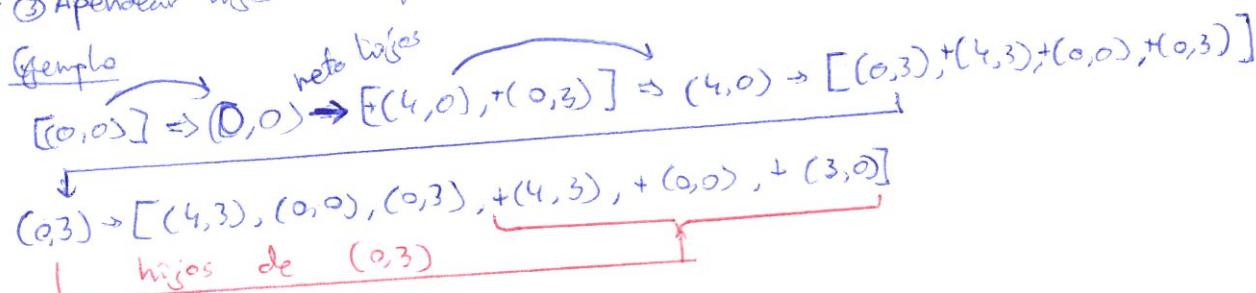


③ Búsqueda en anchura - Cola

Algoritmo

- ① Encolar $(0, 0)$. Si solución, stop.
- ② Popear primero. Si solución, stop.
- ③ Apilar hijos de izquierda a derecha (primero izquierda)

Nota: `[a, b, c, ... z]` início fin
pop append



• G's Graven 2018, Pg 5

① Campos aplicados. Éxitos. Fracassos.

- Sólo se: inferencia lógica para llegar al objetivo. Toda un programa
- Sólo se: representada en lenguaje nat. log.
- Búsqueda: examinar opas y busq. eficientes.
- Reconocer patrones: para ver si el problema se puede resolver por esa IA.
- Representaciones
- Sentido común y razonamiento de mundo en fincas.

② Concepto de computabilidad?

⇒ Un problema es computable si es posible resolverlo por un computador en tpo. razonable.

Por q. fue necesario precisarlo?

⇒ Para poder distinguir qué problemas son computables y cuáles no, q. q. se

(que repercusión?)

⇒ Turing demostró q. cualq. problema computable se podrá

resolver con la máq. de Turing.

⇒ Church - λ-calc. & F. Recursivas

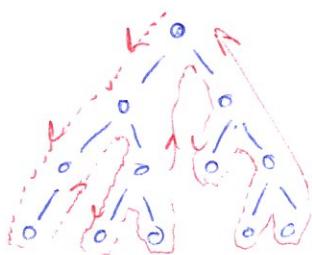
⇒ Por tanto, repercusión en el interés y desarrollo de máquinas

y arquitecturas basadas en estos ppds, especialmente
la máquina de Turing & la arquitectura de Von Neumann.

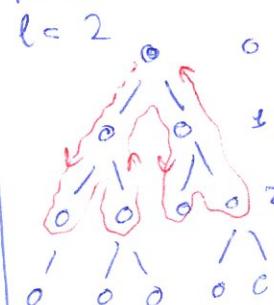
③ Busq. prof. y revisarlos

Usa pila LIFO para abiertos

Prof.



prof. limitada



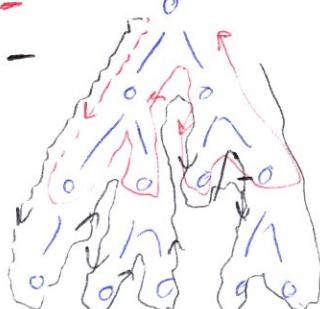
prof. limitada

$l = 2$

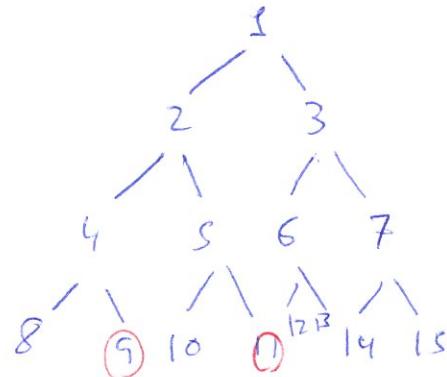
Prof. lim iter.

$$l_1 = 2 -$$

$$l_2 = 3 -$$



④ $CE = l$
 $EM = 2l, q$
 $OP = 2^n, 2^{n+1}$



$$OP_{inv} \Rightarrow \left(\frac{n}{2}\right) \cdot \left(\frac{n-l}{2}\right) \Rightarrow \frac{n - n \% 2}{2} \quad \text{donde } n \% 2 \text{ es el resto de } n/2.$$

D =	n =	9 1 4 1 2 1 1	11 1 5 2 2 1 1
-----	-----	---------------------------------	---

Anchura			Prof		
N	Suc	Ab.	N	Suc	Ab.
∅	∅	∅	9	∅	9
1	2, 3	+2 +3	9	4	+4
2	4, 5	3, +4 +5	Camino = 1, 2, 4, 9.		
Anchura			Prof		
∅	∅	∅	∅	∅	11
1	2, 3	+2 +3	11	5	+5
2	4, 5	3, 4, 5	Camino = 1, 2, 3, 5, 11		

• IA - T3 - Búsqueda heurística

• Idea general

Se tiene información sobre la distancia o coste a la meta desde cada nodo. Usamos esa info. para descartar otras ramas del grafo y llegar antes a la solución.

Modelamos una función h que, dado un nodo, devuelve la distancia o coste al estado meta. Esta info. es incierta, sacrificamos certeza a cambio de eficiencia.

① Algoritmos voraces

- Escalada: se toma el mejor nodo abierto hasta que se llegue a solución o se produzca fallo. Sólo se considera ese camino.

- Primero el mejor: se tratan abiertos como una cola de prioridad en función de la heurística. Los nodos tienen valor y los

- A*: se aplica cuando los nodos tienen valor y los caminos un coste.
- Se suma valor y coste acumulado de llegar a ese nodo.
- Se suman valor y coste acumulado de llegar a ese nodo.
- Abiertos es una cola de prioridad de menor a mayor.
- Si se encuentran caminos más baratos se debe actualizar Abiertos.

② Memoria limitada

- A* PI: se utiliza profundidad iterativa y A*. El límite viene dado por (profundidad + coste + heurística).

- A* MS: utiliza toda la memoria disponible. Profundiza hasta que la memoria se llena, después elimina los peores nodos y construye un nuevo árbol.

Es completo y óptimo si la solución es alcanzable. Puede ser ininteligible por tpo. infinito.

- Búsqueda local: funcionan con un solo estado.

Usan poca memoria y encuentran soluciones razonables en espacios de estados infinitos. No son sistemáticos.

③ Propiedades y características

$h(n)$ → función heurística

$h^*(n)$ → función heurística óptima, ideal.

$g(n)$ → coste para alcanzar n desde el inicio.

$f^*(n) = g(n) + h^*(n)$ → coste óptimo para alcanzar n .

$f^*(n) \leq g(n) + h^*(n)$ → coste óptimo para alcanzar n .

- Heurísticas admisibles: $h(n) \leq h^*(n)$

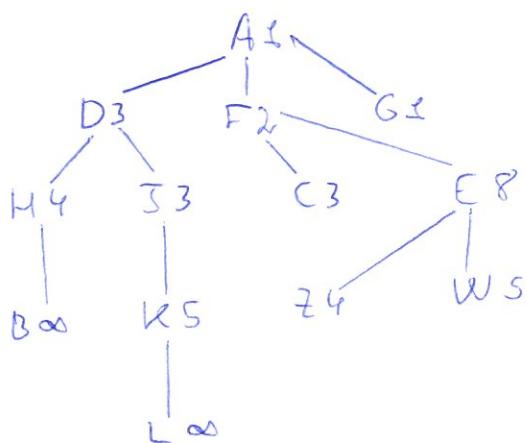
↳ A* es completo y óptimo.

- $h(n)$ es más informativa que $h'(n)$ si: $h'(n) < h(n) < h^*(n)$
- $h(n)$ es consistente si teniendo $n_1 \xrightarrow{c} n_2$ se cumple $h(n_1) - h(n_2) \leq c(n_1, n_2)$
- $h(n)$ es monótona si teniendo $n_1 \xrightarrow{c} n_2$ se cumple $f(n_2) \geq f(n_1)$
- $h(n)$ es consistente $\Rightarrow h(n)$ es monótona.
- Si se tienen varias heurísticas, se puede obtener la mejor de ellas de la siguiente forma:
$$h(n) = \max \{ h_1(n), h_2(n), \dots, h_m(n) \}$$

• IA- Ejemplos y ejercicios T3

1) Resolver por escalada.

Cada nodo tiene un número. A mayor nº, mejor solución.

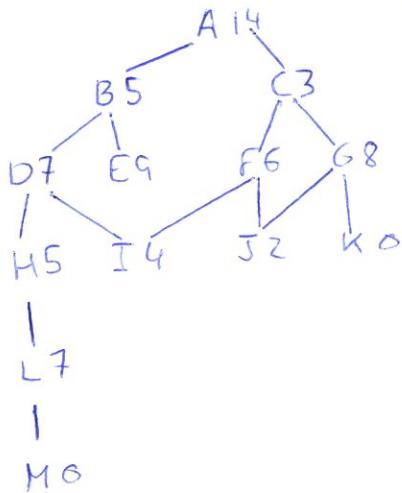


Paso	Actual	Abiertos	Tomo ($h = \text{nº mayor}$)
0	\emptyset	A-1	A
1	A	D-3, F-2, G-1	D
2	D	H-4, J-3	H
3	H	B- ∞	B

Nota: al tomar un nodo, descarto el resto de abiertos para siempre.

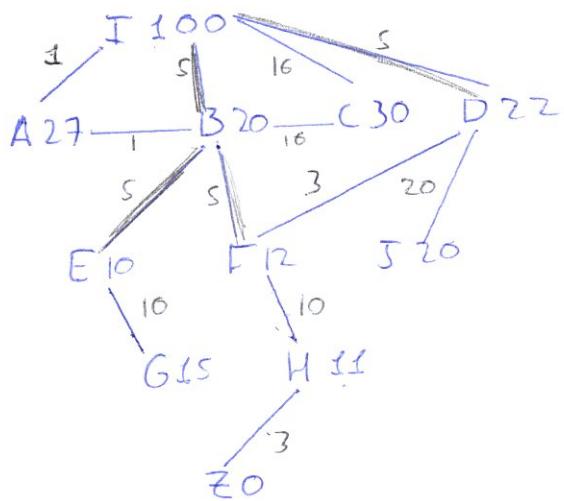
2) Resolver por primero el mejor.

Cada nodo tiene un nº. A menor nº, mejor solución.



Paso	Actual	Abiertos	Tomo	Cerrados
0	\emptyset	A14		\emptyset
1	A14	B5, C3		A, \emptyset
2	C3	B5, F6, G8		A,
3	B5	F6, D7, G8, E9		A, C
4	F6	J2, I4, D7, G8, E9		A, C, B
5	J2	I4, D7, G8, E9		A, C, B, F
6	I4	D7, G8, E9		A, C, B, F, J
7	D7	H5, G8, E9		A, C, B, F, J, H
8	H5	L7, G8, E9		A, C, B, F, J, H , D
9	D7	M0, G8, E9		A, C, B, F, J, I, D, H

3) Aplicar A*. El valor de cada nodo indica distancia a la meta. (meta = 0).



Paso	Actual	Abiertos	Tomo	Cerrados	Camino Acum.
0	∅	I 100		∅	0
1	I 100	B2S, D27, A28, C46	B2S	D ∅	0
2	B2S	E20, F27, D27, A28, C45	E20	B, I, Falta B	5
3	E20	F27, D27, A28, G35, C45	F27	B, I, E	10
4	F27	D27, A28, H31, G35, C45	D27	B, I, G, F,	10
5	D27	A28, H31, G35, J45, C45	A28	B, I, E, F, D	5
6	A28	H31, G35, J45, C45	H31	B, I, C, F, D; A	1
7	H31	Z23, G35, J45, C45	Z23	B, I, G, F, D, A, H	20
8	Z23	X	X	X	23

• Ejercicios examen T3 - IA

① Búsqueda heurística bidireccional

La búsqueda bidireccional consiste en realizar dos búsquedas a la vez:

- De origen a destino.
- De destino a origen.

Tendría sentido si se tiene al menos una heurística consistente (^{y por tanto} monótona) para al menos un sentido, de forma que sea posible que la búsqueda converja en algún nodo intermedio.

Para obtener una heurística consistente para el sentido

opuesto, podemos invertir la heurística:

opuesto, podemos invertir la heurística:
$$h'(n) = h(o) - h(n)$$
 } Donde $h(o)$ es h en el nodo inicial.

③ ¿Bajo qué condiciones A* es completo?

Un método o algoritmo es completo si siempre encuentra solución si ésta existe.

A* es siempre completo ya que:

1) Está basado en la búsqueda desinformada de coste uniforme.

2) Explora todas las rutas hasta encontrar una solución.

Siempre y cuando

1) El árbol no trolley (sin bucles, con solución, etc)

2) La heurística no trolley (que no margea sol. con $h(\text{sol}) = \infty$ ni paga sadas similares).

Otra cosa es que sea óptimo. Lo será si: $h(n) \leq h^*(n)$: es decir, se tiene info. pero no mucha.

• Hamming y Euclídea como heurísticas

Distancia hamming: # bits que deben cambiar para que una cadena sea igual a otra.

$$\text{Ej. hamming}(010,001) = 2$$

Para problemas de la laberinto, podemos asignar a cada casilla un número y expresarlo en binario.

$\begin{array}{ c c c } \hline 0 & 0 & 1 \\ \hline 0 & 1 & 1 \\ \hline 1 & 0 & 0 \\ \hline \end{array}$	$\begin{array}{ c c c } \hline 0 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}$	$\begin{array}{ c c c } \hline 1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}$
0	1	2

} = Es ~~admisible~~ consistente?
~~hamming(a,b) - hamming(c,d) <= coste((a,b) a (c,d))~~
~~(1-2 < 3)~~
~~h(1,1) - h(1,0) <= coste((1,1) a (1,0))?~~
~~(3-2 < 1)~~

No puedo demostrar q. no es admisible.

No, si tengo por destino 0000,
el coste de saltar de 0111 a 0100 es 1,
pero $h(0111) - h(0100) = 2 \quad \& \quad 2 > 1$.

Por tanto, no es ~~admisible~~ consistente.

? Es admisible?

(Es admisible si $h(n) \leq h^*(n)$ para cualq. n).

No: $h_{\text{hamming}}(0111) > 1$. ($D=1000$).

Distancia euclídea: $d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

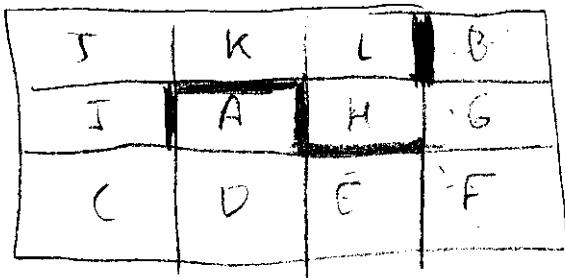
			3,3
1,0	1,1	..	-
0,0	0,1	0,2	0,3

? Es consistente?

Será constante si $h(a) - h(b) \leq \text{coste}(a \rightarrow b)$

Por como se define el tablero,

Laberinto



INI: A(2,2)

FIN: B(1,4)

Usando $h(n) \sim$ distancia a destino si no hay muros.
Ej $h(a) = 3$, $h(f) = 2$, $h(c) = 5$, $h(b) = 0$.

A	Pass	Actual	Abiertos	Cerrados
	0	0	A0	∅
	1	A0	D5	∅
	2	D5	ES, SF	D5
	3	ES	FS, CF	
		FS	GS, CF	
		GS	BS, CF	
		BS		

o IA - T3- Búsqueda heurística

Cuando no se tiene info. sobre la solución, se utiliza la búsqueda desinformada y se buscan por todas las salidas de operaciones posibles.

Cuando se tiene info., se hace uso de la búsqueda heurística, donde se emplea esta info. para desarrollar de la lista de abiertos aquellas q. aparenten estar más cerca de la meta.

① Idea general

Desarrollamos una función que estime la "distancia" o "coste" al estado meta. Esta función vale 0 en las metas.

Esta info. es incierta; sacrificamos certeza a cambio de eficiencia. Gracias a ella podemos poder ramas y descartar caminos q. no parezcan conducir a una meta.

Combiniendo la info. del recorrido realizado con la info. heurística podemos establecer métodos q. llevan a la meta.

$$h(\text{nodo}) = \text{coste estimado}$$

Debenas definir h .

Valores típicos:

- ✘ elem descolocados
- ✘ coord. a destino (x, y)
- ✘ casillas mías a destino.

Ej. puzzle con piezas desord.

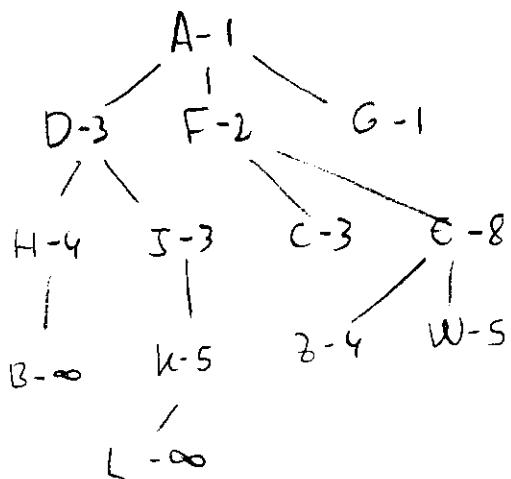
*) * piezas descolocadas
también → distancia pieza a pos final

Paso	Actual	Abiertos	Cerrados	Camino actualizado
0	∅	I 100	∅	0
1	I 100	A28, B25, C46, D27	∅	0
2	B25	C45, A28, E20, F27, D27	I 100	5
3	E	G35, C45, A28, F27, D27	I, B	10

Voraces } Escalada actual \rightarrow sucesor wyo h sea menor. Solo f Camino.
 Primero el mejor: ABIERTOS es cola de prioridad en base a h.

Ej. Escalada

Cada nodo tiene un n.º A n.º mayor, mejor solución.



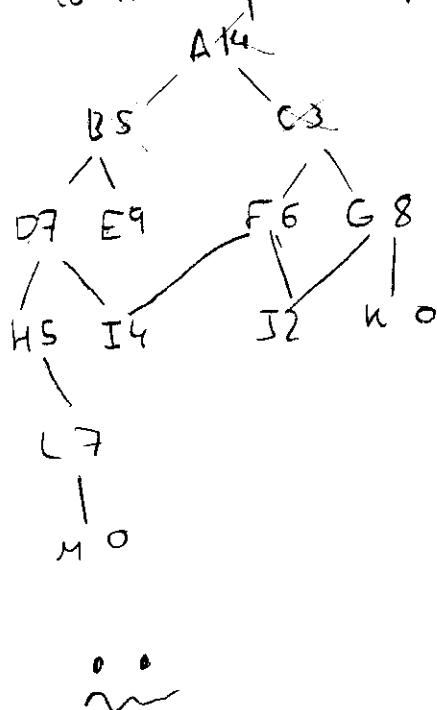
Nota: h debe ser inverso de f.

Es decir, cu n.º mayor n.º de nodo, menor h.
 $(h=0 \Rightarrow$ Meta).

Actual	Abiertos	Tomo (Valor mayor)
/	A-1	A-1
A-1	D-3, F-2, G-1	D-3
D-3	H-4, I-3	H-4
H-4	B-∞	B-∞
F	I	N ..

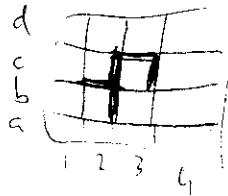
Ej. primero el mejor

Lo mismo pa este pero los n.º heredados son guays.

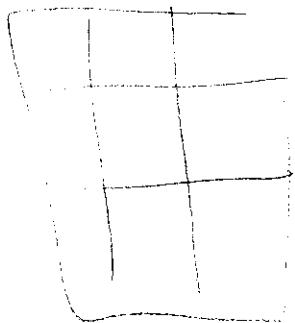


Paso	Actual	Abiertos	Cerrados
0		A14	
1	A14	B5, D7, E9	
2	C3	B5, F6, G8	A14
3	OS	F6, D7, G8, E9	F6, A14, C3, OS
4	F6	J2, I4, D7, G8, E9	F6, A14, C3, OS
5	J2	I4, D7, G8, E9	F, A, C, D, J, I
6	I4	D7, G8, E9	F, A, C, B, J, I
7	B5	H5, G8, G9	
	H5	L7, G8, G9	
	L7	M0, -	

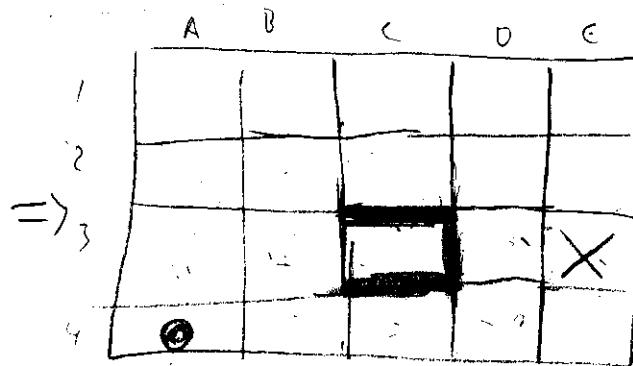
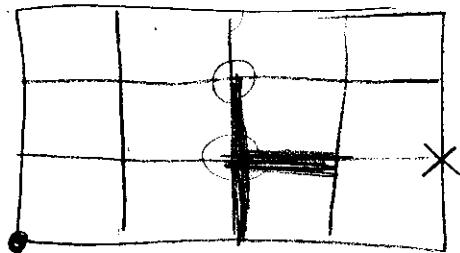
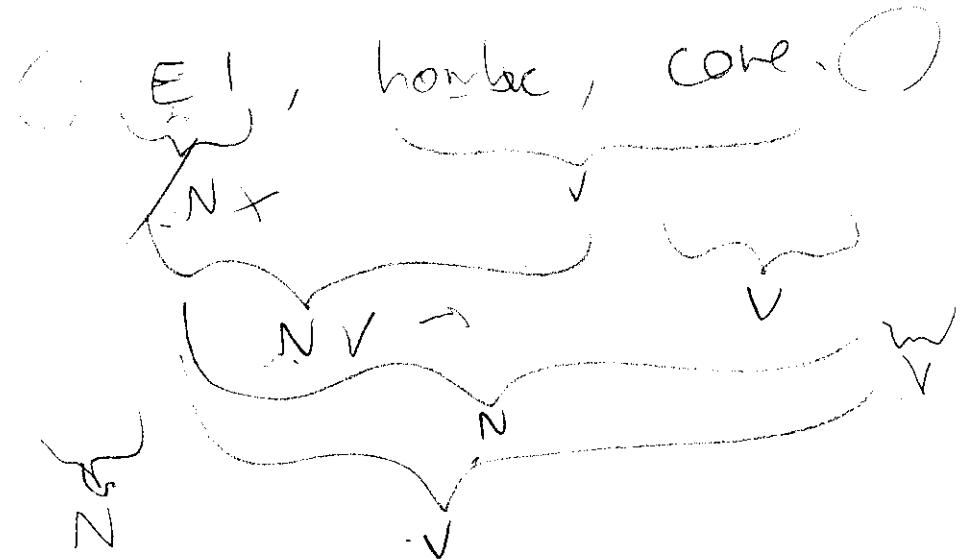
IA TZ - C1



h(ect)		approx.	ideal
a 1	6	6	
a 2	5	5	
a 3	4	4	
a 4	3	3	
c 1	4	4	
c 2	3	3	
c 3			
c 4			

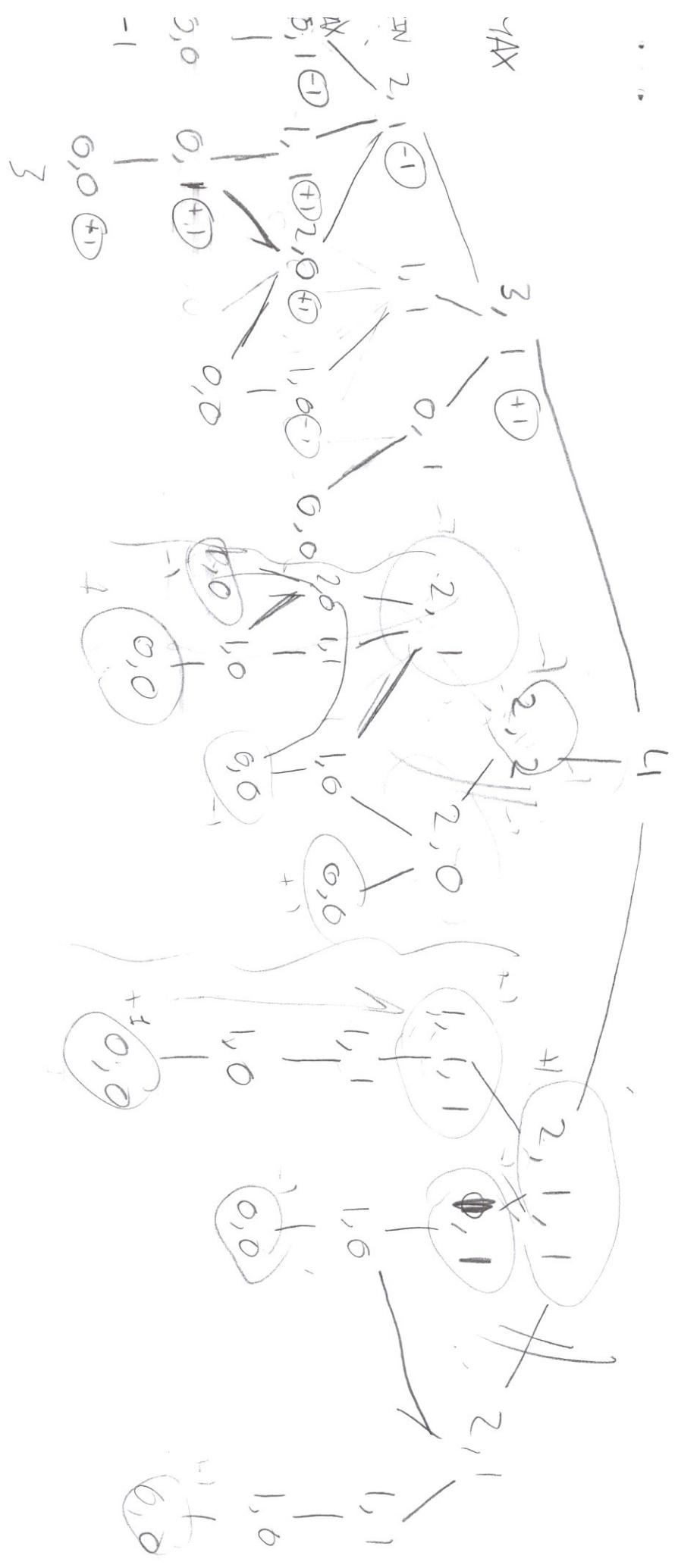


- 1) frase → lista
 - 2) O... ind-verb → frase verbal
 - 3) ind-verb → frase verbal



$h \rightarrow$ distancia sin saber el mero de medida
Si a

Node	Abc	
A4-5	A3-5, B4-5	A3
A3-5	B3-5, A2-5, B4-5	B3
B3-5	C3-5, B2-5, B4-5, A2-6	(?)
C3-5	B4-5, B2-6, A2-6	B4-5
B4-5	B4-5,	
C4-5	D4-5,	
D4-5	D3-5, E4-5,	
D3-5	E3-5, C4-5	



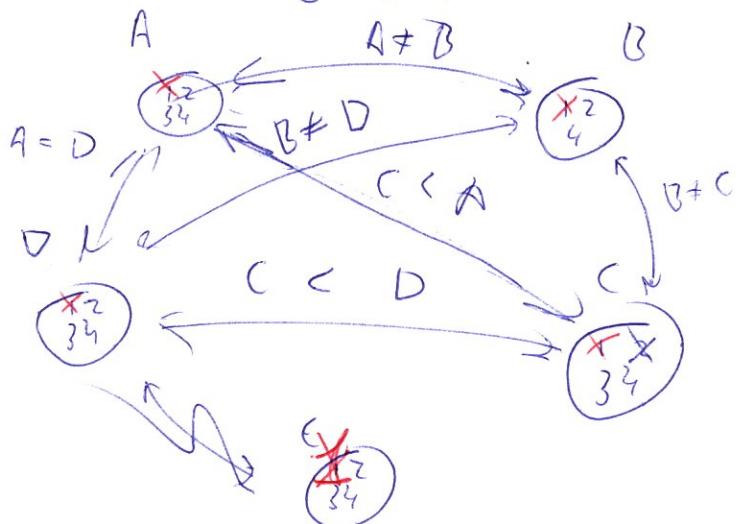
Admissible: $h(n) \leq h^*(n)$
 Consistent: $h(n) - h(n') \leq c(n, n')$
 Monotone: $f(n') \geq f(n)$

Act. = { A, 0, C, D, E }

Tpos = { 1, 2, 3, 4 } para cada $A_i \in A$

Restricciones: $B \neq 3$, $A \neq B$, $C < D$, $A = D$, $E = \emptyset$

$i \neq 2$, $B \neq C$, $B \neq D$; Restricciones correc.



$A = D$, $E = \emptyset$

$C < A$
 $E < A, B, C, D$

~~(A 1 2 3 4) (B 2 3 4)~~
~~(C 1 2 3 4) (D 1 2 3 4)~~
~~(E 1 2 3 4)~~

A4 B2 C3

D4

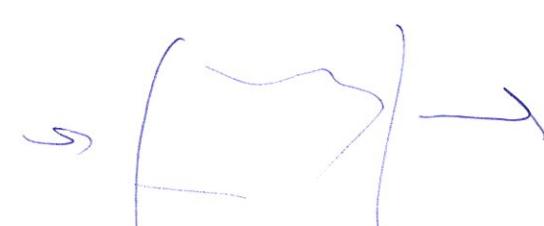
E1

~~B ≠ A, A ≠ B, C < D, D > C, A = D, D = A,~~
~~B ≠ C, C ≠ B, B ≠ D, D ≠ B, A > E, E < A,~~
~~B > E, E < B,~~
~~C > E, E < C,~~
~~D ≥ E, E < D~~

$A(1\ 2\ 3\ 4)$ $B(1\ 2\ 3\ 4)$
 $C(1\ 2\ 3\ 4)$ $D(1\ 2\ 3\ 4)$
 $E(1\ 2\ 3\ 4)$

$E < A,$
 $E < B,$
 $E < C,$
 $E < D,$

$A(2\ 3\ 4)$
 $B(2\ 3\ 4)$



• $T = \text{parecido} \approx 30$

$G = \text{grande}$

$V = [0..60]$

ΔT^o	0	5	10	15	20	25	30	35	40	45	50
M_T	0	0	0	0.05	0.17	0.18	1	0.8	0.7	—	
M_G	0	0	0	0	0	0	0.2	0.4	0.6	0.8	1.
T^c	1	1	1	0.5	0.3	0.2	0	0.2	0.3	—	
G^c	1	1	1	1	1	1	0.8	0.6	0.4	0.2	—
$(T \cup G)^{\leq 0}$	0	0	0	0.5	0.7	0.8	1	0.8	0.7	0.8	1.
$(T \cup G)^{\leq 1}$	1	1	0.5	0.3	0.2	0	0.2	0.3	0.2	0.1	—
$(T \cap G)^{\leq 0}$	0	0	0	0	0	0	0.2	0.3	0.2	0	
$(T \cap G)^{\leq 1}$	1	1	1	1	1	1	0.8	0.7	0.8	1.	

Consistencia nodos

(define (coseno x h)
(div (pow x (.- 2 h)) fact))

Consistencia arcos. querer ~~verificar~~ valores del
dominio g. violen restricciones.

$$\text{M}_A \left\{ \begin{array}{l} 0 \leq x < 25 \\ (x-25)/25 \leq s \leq 1 \end{array} \right.$$

$$\left\{ \begin{array}{l} (x-25)/25 \leq s \leq 1 \in [25, 50] \\ (75-x)/25 \leq s \leq 1 \in [50, 75] \end{array} \right.$$

$$0 \leq x > 75$$

$$\left\{ \begin{array}{l} 1 \leq s \leq 1 \in [0, 30] \end{array} \right.$$

$$\text{M}_B \left\{ \begin{array}{l} 1/2 \leq s \leq 1 \in [30, 50] \end{array} \right.$$

$$\overline{\text{M}}_A \left\{ \begin{array}{l} 1 \leq s \leq 1 \in [25, 50] \end{array} \right.$$

$$\left\{ \begin{array}{l} 1 - [(x-25)/25] \leq s \leq 1 \in [25, 50] \\ 1 - [(75-x)/25] \leq s \leq 1 \in [50, 75] \\ 1 \leq s \leq 1 \in [75, 100] \end{array} \right.$$

$$\overline{\text{M}}_B \left\{ \begin{array}{l} 0 \leq x \leq 30 \end{array} \right.$$

$$\left\{ \begin{array}{l} 1/2 \leq s \leq 1 \in [30, 50] \end{array} \right.$$

$$A \cup B \left\{ \begin{array}{l} 1 \leq s \leq 1 \in [0, 30] \end{array} \right.$$

$$\text{max} \left\{ \begin{array}{l} \cancel{(x-25)/25} + 25 \leq s \leq 1 \in [30, 38] \\ 1/2 \leq s \leq 1 \in [30, 38] \\ (x-25)/25 \leq s \leq 1 \in [38, 50] \\ \cancel{(75-x)/25} \leq s \leq 1 \in [50, 63] \end{array} \right.$$

$$\left\{ \begin{array}{l} 1/2 \leq s \leq 1 \in [30, 38] \\ (75-x)/25 \leq s \leq 1 \in [50, 63] \\ 1/2 \leq s \leq 1 \in [63, 100] \end{array} \right.$$

$$\overline{A} \cap \overline{B} \left\{ \begin{array}{l} 0 \leq s \leq 1 \in [0, 30] \end{array} \right.$$

$$\text{min} \left\{ \begin{array}{l} 1/2 \leq s \leq 1 \in [30, 38] \end{array} \right.$$

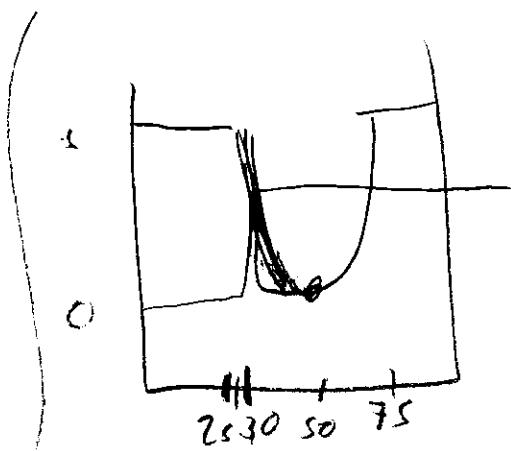
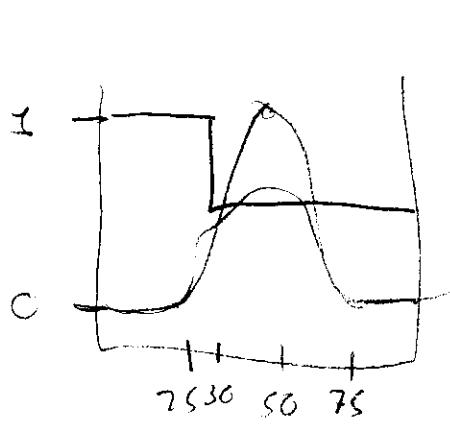
$$\left\{ \begin{array}{l} 1 - [(x-25)/25] \leq s \leq 1 \in [38, 50] \end{array} \right.$$

$$\left\{ \begin{array}{l} 1 - [(75-x)/25] \leq s \leq 1 \in [50, 63] \end{array} \right.$$

$$\left\{ \begin{array}{l} 1/2 \leq s \leq 1 \in [63, 100] \end{array} \right.$$

$$A^c \cap B^c = \emptyset$$

$$\left\{ \begin{array}{l} 0 \leq x \in [0, 30] \\ (K - 2S) / 2S \leq x \in [38, 50] \\ 1/2 \leq x \in [30, 37] \end{array} \right.$$

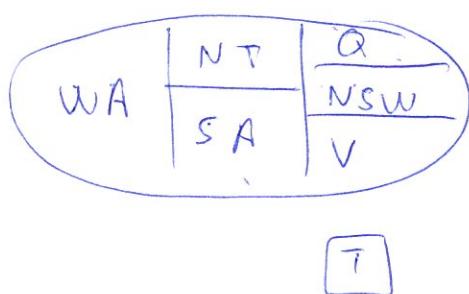


$$\begin{aligned}
 & A^c \cup B \quad \left\{ \begin{array}{ll} \text{si } x \in [0, 30] & \star \quad 7 \\ 1 - [(x-20)/25] & 0 \quad 1 \\ \text{si } x \in (30, 38] & 0.2 \\ 1 - [(x-38)/18] & 0.4 \quad 0.6 \\ \text{si } x \in (38, 50] & 0.4 \\ 1 - [(x-50)/12] & 0.2 \\ \text{si } x \in (50, 63] & 0.6 \\ 1 - [(x-63)/10] & 0.4 \\ 1 - [(x-75)/25] & 0 \end{array} \right. \\
 & \text{m8}
 \end{aligned}$$



① Problemas PSR

① Australia coloreada



$$X = \{WA, SA, NSW, Q, V, NT, T\}$$

$$D = \{\text{rojo, azul, verde}\} \text{ para cada } x_i \in X$$

$$C = \left\{ \begin{array}{l} \langle (x_i, x_j) \text{ tal q. } x_i \text{ toca a } x_j, \\ (\text{color}(x_i) \neq \text{color}(x_j)) \end{array} \right\}$$

② Clases y horarios

$$X = \{ECC, EDD, \dots\}$$

$$D = \{(Lunes, 1a), (Martes, 1b), \dots\}$$

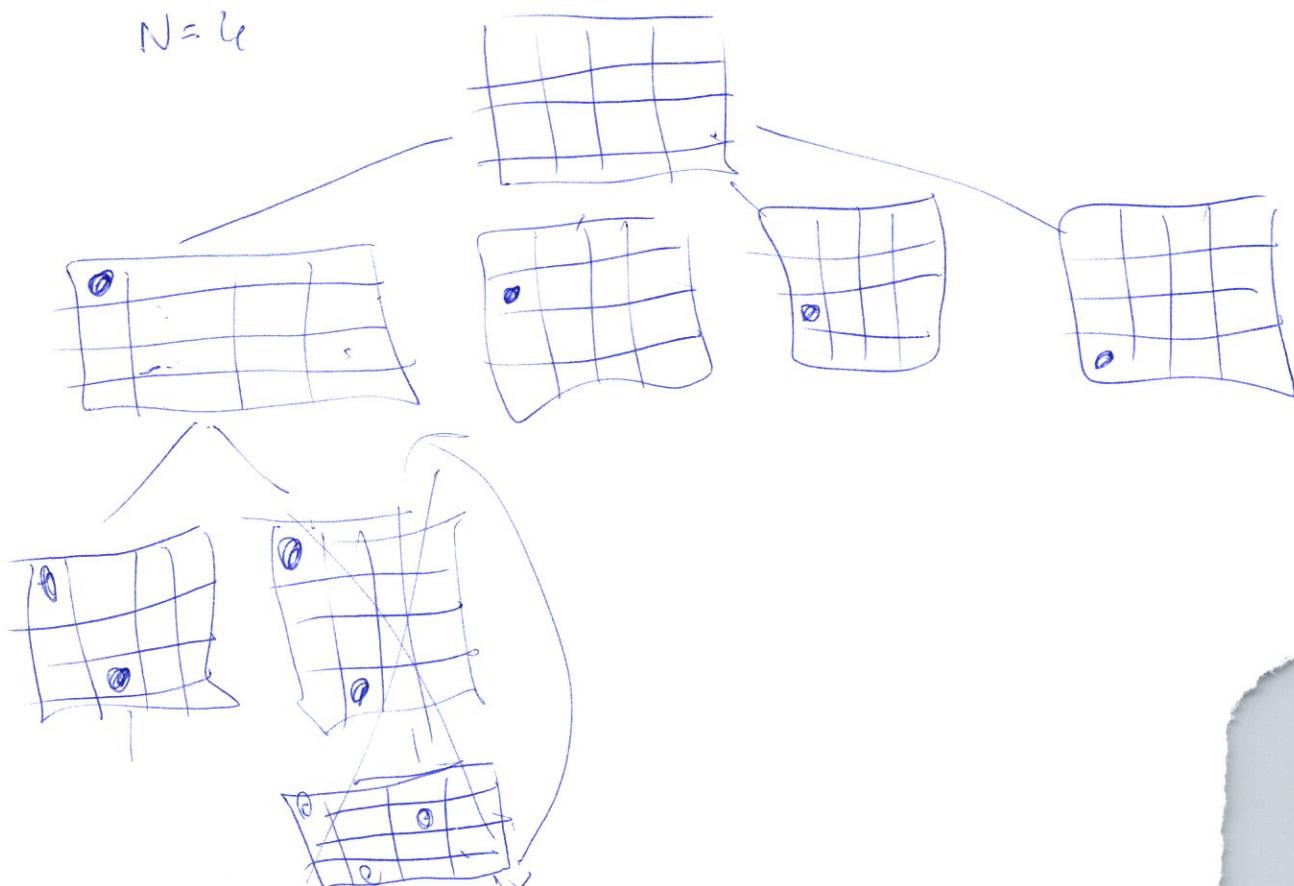
③ N reinos

$$\text{Var } \{V_1, V_2, \dots, V_n\}$$

$$\text{Don } \{D_i : \{(1,1), (1,2), \dots\}, D_{i+1}, \dots\}$$

Utilizando: prob. búsqueda en espacios de estados

$$N = 4$$



5 Tareas en 4 tpos.

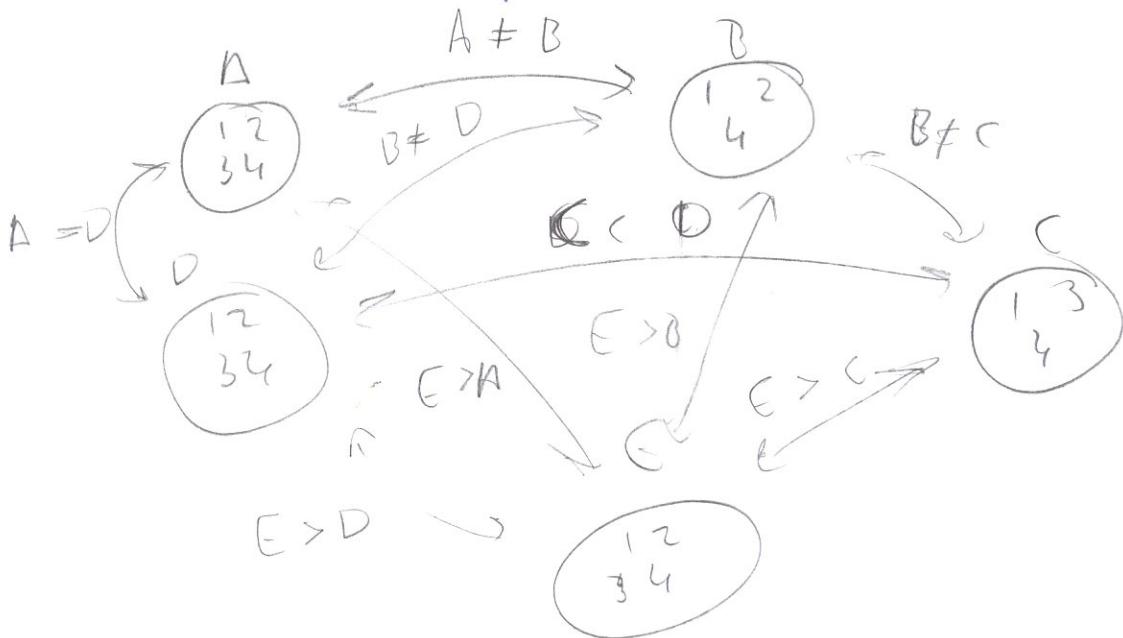
Tar: {A, B, C, D, E}

Tpo: {1, 2, 3, 4}

$D < C$

Restricc: $\{(B), (C), (AX_i, BX_i), (BX_i, EX_i), ((X_i, DX_i), i \in \{1, 2, 3\})\}$
 $\{ (BX_i, DX_i) | (AX_i, DX_i) | (EX_i, i < \max(tpo))\}$

Casos q. no pueden darse:



$$V = [0, 1, 2, \dots, 60]$$

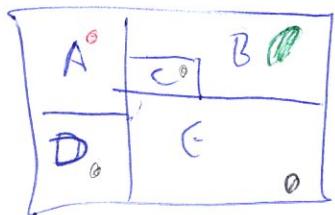
$$B \subseteq A \Leftrightarrow \mu_B(x) \leq \mu_A(x) \text{ para } \forall x \in X$$

$$C = A \cap B \Leftrightarrow \mu_C(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x) \text{ para } \forall x \in X$$

$$C = A \cup B \Rightarrow \mu_C(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x) \text{ para } \forall x \in X$$

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \text{ para } \forall x \in X$$

2017



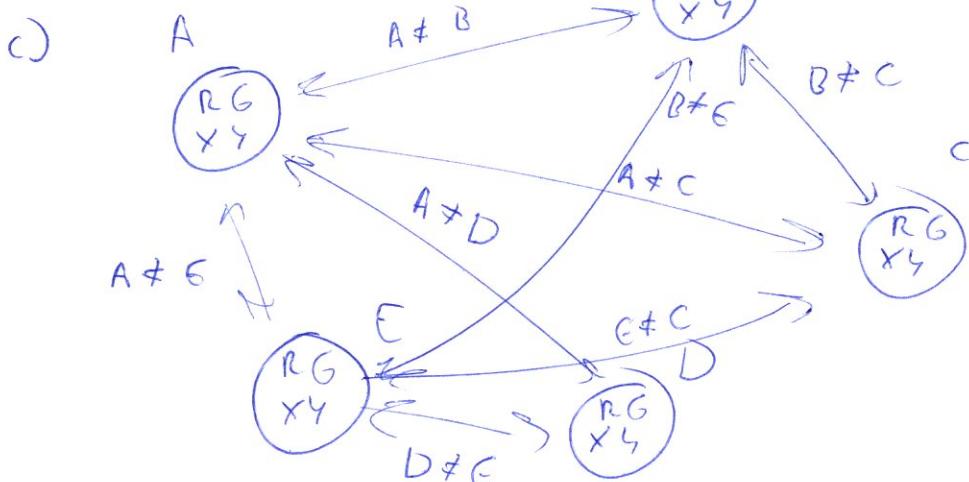
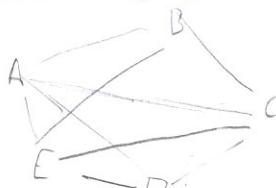
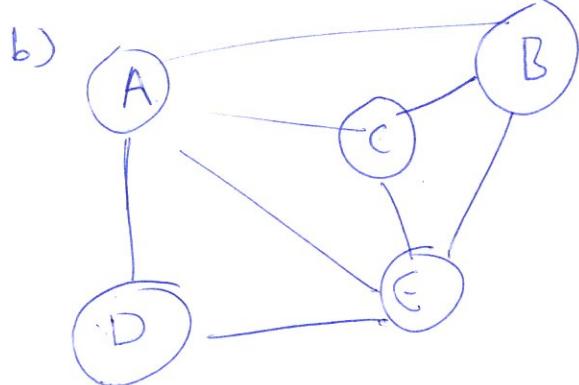
Mínimo 4 colores

a) Plantear el problema

$$X \{A, B, C, D, E\}$$

$D \{(R, G, B, \text{color}) \text{ para cada } X_i\}$

$R \{(\forall X_i X_j \text{ tal que } X_0 \text{ toca a } X_i), (\text{Color}(X_0) \neq \text{Color}(X_i))\}$



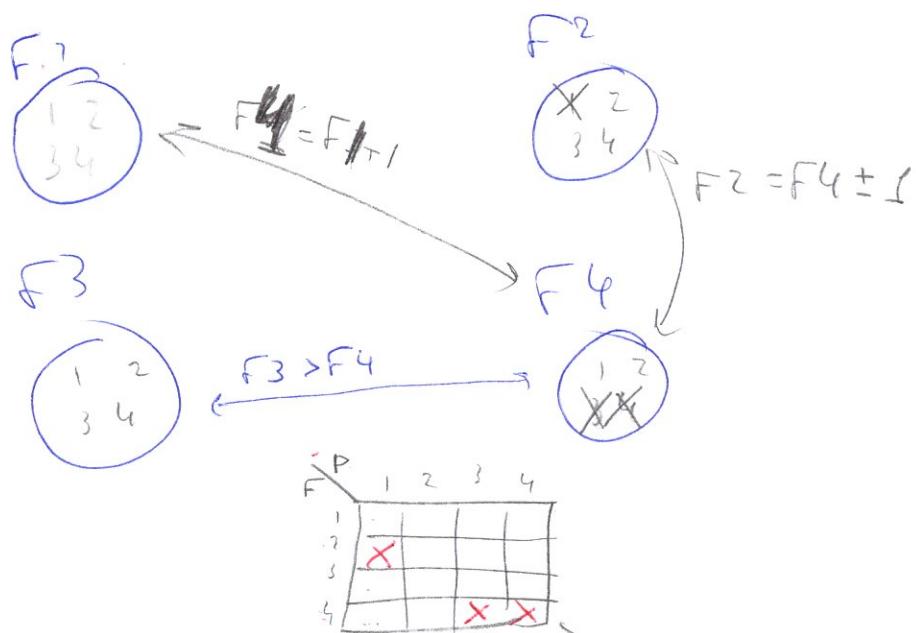
2018

③ Familias $\{F_1, F_2, F_3, F_4\}$

Cases $\{P_1, P_2, P_3, P_4\}$

Restricciones aka casos no posibles:

~~STC~~ ~~DF3~~ $| F_3 > F_4;$



X			
X			
X			
O	X	XX	

→ Rompe $F_4 = F_3 + 1$

$$1 = 0 + 1$$

No exist.

O	X	X	X
X	X		
X	X		
X	O	X	X

O	X	X	X
X	X		
X	X		
X	O	X	X

→ Rompe $F_2 = F_4 + 1$

O	X	X	X
X	X	O	X
X	X	X	O
X	X	X	X

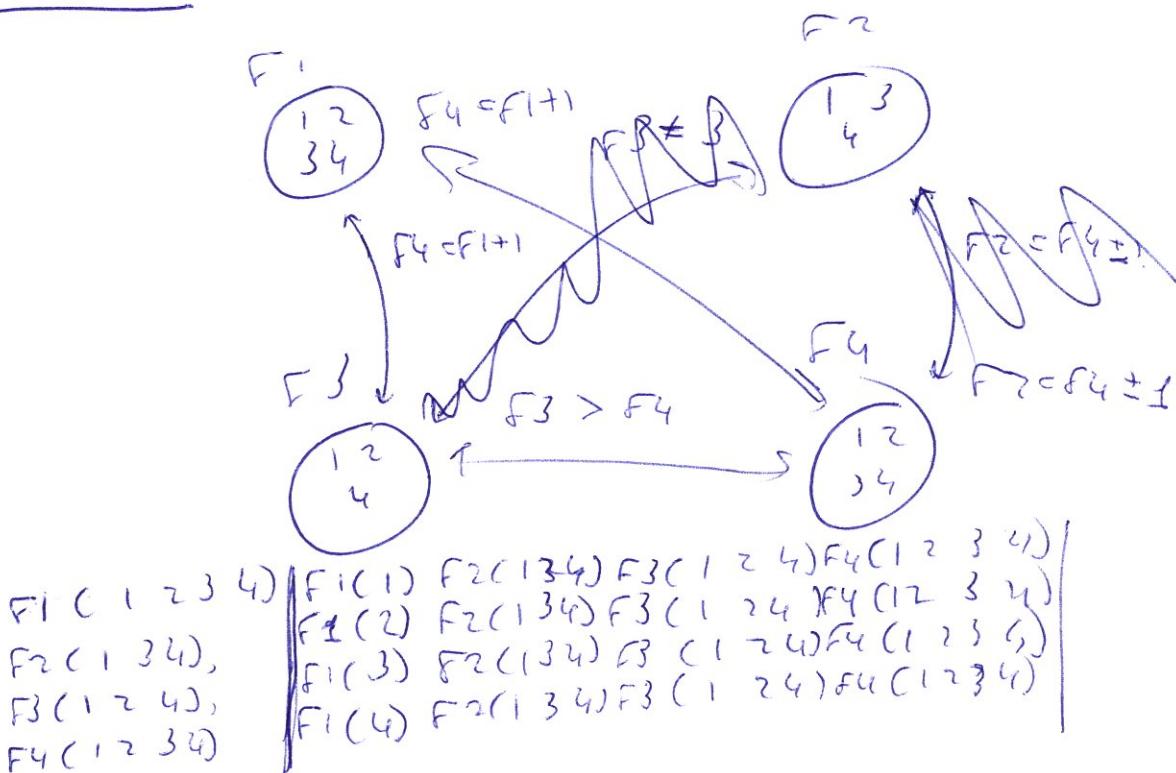
2.- 2018

$$X = \text{casos} = \{c_1, c_2, c_3, c_4\}$$

$$D = \text{familias} = \{F_1, F_2, F_3, F_4\} \text{ para cada } c_i \in X$$

$$R = \{F_3 > F_4, F_4 = F_1 + 1 \quad \begin{cases} F_2 = F_4 \pm 1, F_3 \neq 3, F_2 \neq 2 \\ (F_2 = F_4 + 1 \vee F_2 = F_4 - 1) \end{cases}\}$$

AC-130



$$U = [1, 100]$$

$$A^c = \begin{cases} 1 & \text{si } x < 25 \\ 2 - [(x-25)/25] & \text{si } x \in [25, 50] \\ 3 - [(75-x)/25] & \text{si } x \in (50, 75) \\ 1 & \text{si } x > 75 \end{cases}$$

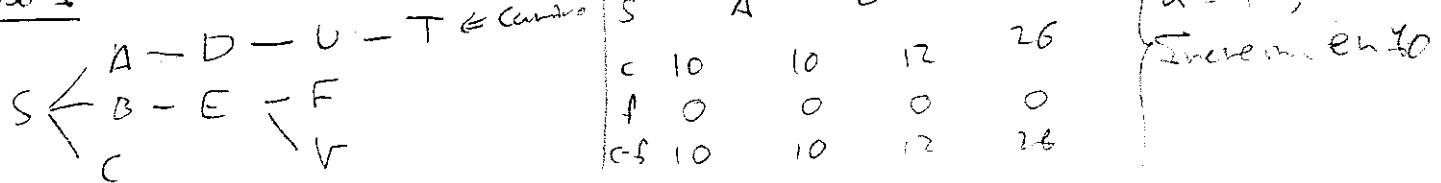
$$B^c = \begin{cases} 0 & \text{si } x \in [0, 30] \\ 1/2 & \text{si } x > 30 \end{cases}$$

$$A \cup B = \begin{cases} 1 & \text{si } x \in [0, 30] \\ 1/2 & \text{si } x \in [30, 37] \\ (x+25)/25 & \text{si } x \in [38, 50] \cup (63, 100] \end{cases}$$

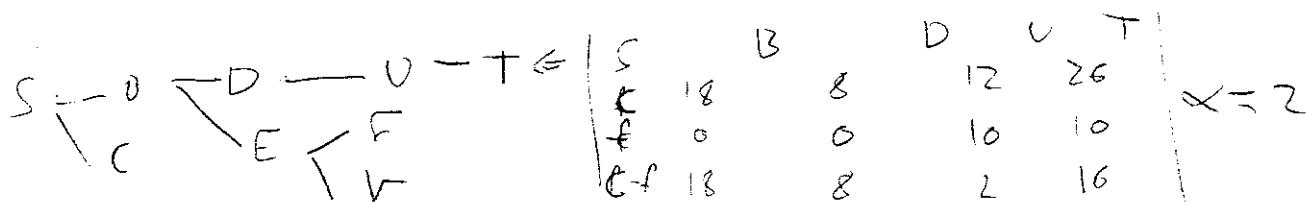
• Examen 2014

	SA	SB	SC	AD	BD	DE	CD	CE	DV	EF	EV	FU	FV	UT	VT
c-f	10	18	22	10	8	10	16	6	12	18	12	14	16	26	28
f ₀	0	0	-	-	-	-	-	-	-	-	0	0
f ₁	10, 0	0	0	10, 0	-	-	0	10	0	-	-	0	10	0	0
f ₂	10, 2	0	10, 2	0	0	0	12, 0	0	0	0	0	0	22	0	0
f ₃	10, 12	0	10, 2	10, 0	0	0	12, 0	10	0	0	0	0	12	10	0
f ₄	.	2	.	.	2	2	.	.	12,	.	12,	.	.	12	12
f ₅	.	6	.	.	.	16,	4	4	4	4	4	4	16	.	.
f ₆

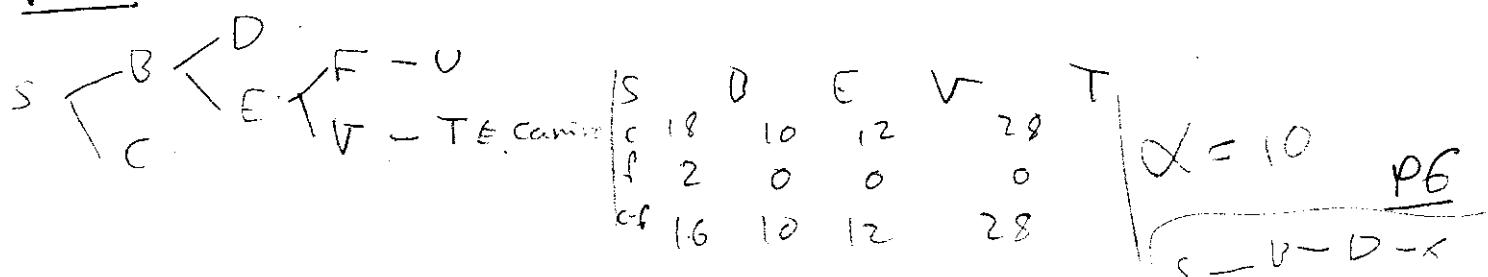
Paso 1



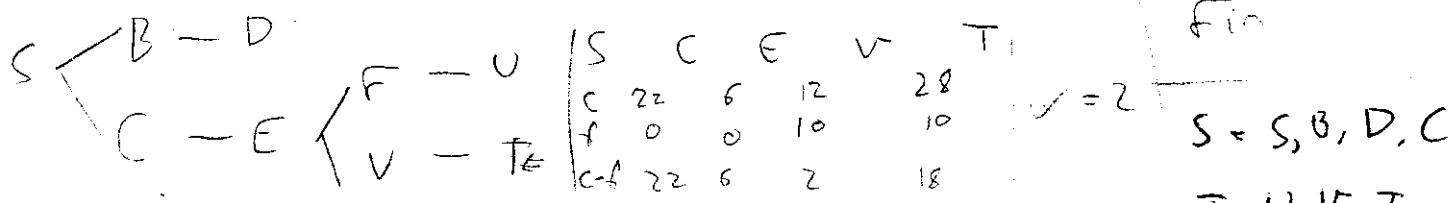
Paso 2



Paso 3



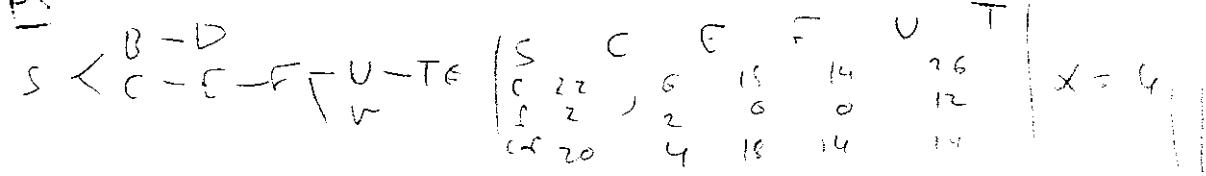
Paso 4



$S = S, B, D, C$

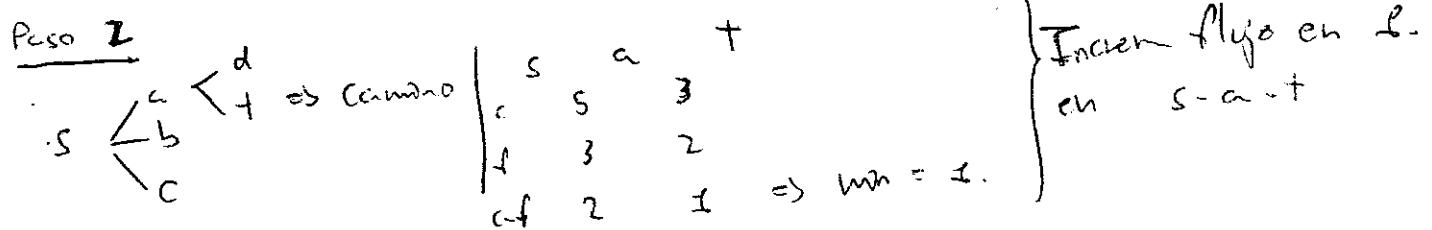
$T = U, V, T$

ps

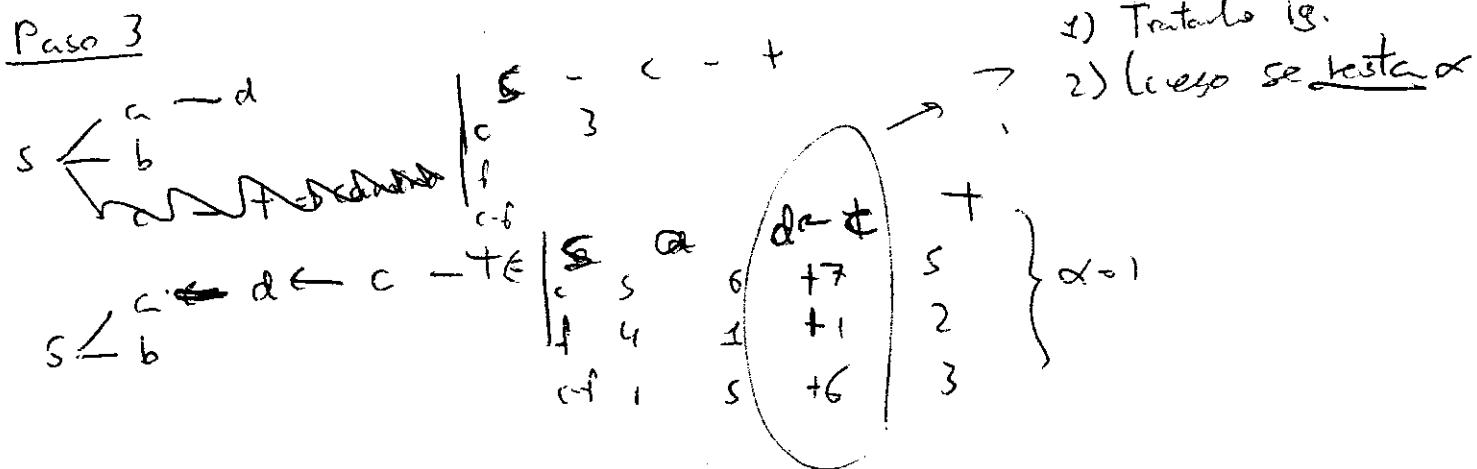


Ejercicios Dijkstra - EC. Discretas

	sa	sb	sc	sd	bd	cd	dt	dt
p	5	4	3	6	2	7	3	5
e	3	2	1/3,	1	1/2,	1	2	1/4,



1	2	1/3,	1	1/2,	1	1/3,	2	1/4,
---	---	------	---	------	---	------	---	------



f	1/5,	2	1/3,	2	1/2,	0	2	3	1/4,
---	------	---	------	---	------	---	---	---	------

Paso 4

No hay más pasos ni opciones:
se ha alcanzado el flujo mix.

$\sum x_t = 2 + 3 + 4 = 90$

(Corte min) $\begin{cases} S = \text{últim. opci. val.} = s, b \\ T = 6 \text{ que entre al sumando} = \{a, c, d, t\} \end{cases}$

5 Tareas en 4 tpos.

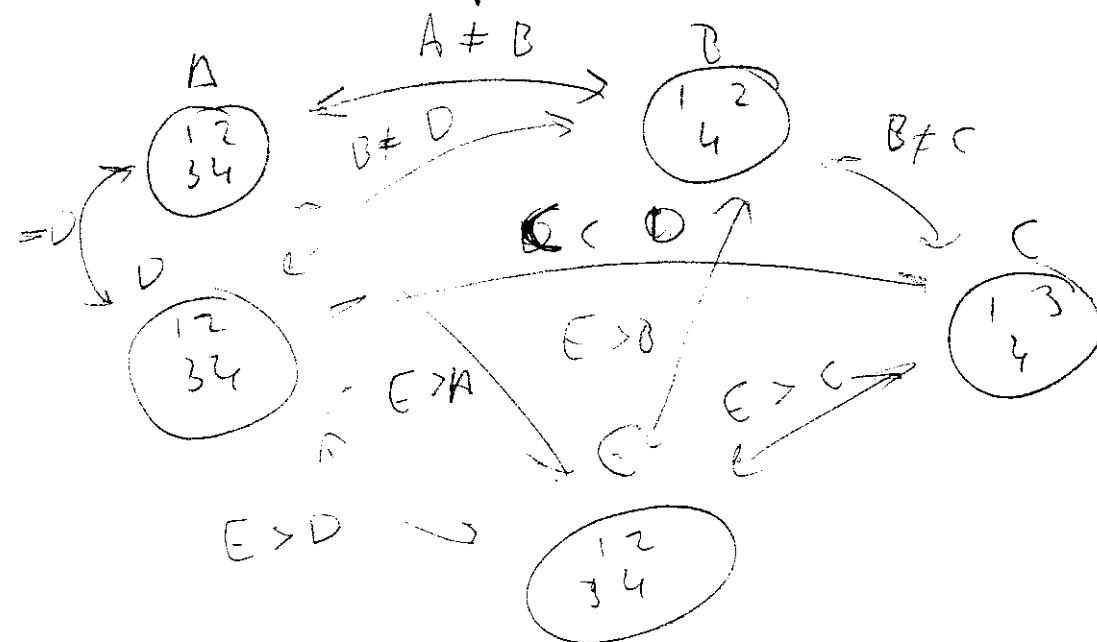
Tar: {A, B, C, D, E}

Tpo: {1, 2, 3, 4}

Restricc: { $\overline{B}, \overline{C}, (AX_i \rightarrow BX_i), (BX_i \rightarrow X_i), ((X_i, DX_i) \rightarrow (B \times_i, DX_i))$
 $\{ (B \times_i, DX_i), (AX_i, DX_i) | EX_i, i < \max(\text{tpo}) \}$ }

$v < c$

Casos q. no producen tarea



$$V = [0, 1, 2, \dots, \infty]$$

$$B \subseteq A \Leftrightarrow \mu_B(x) \leq \mu_A(x) \text{ para } \forall x \in X$$

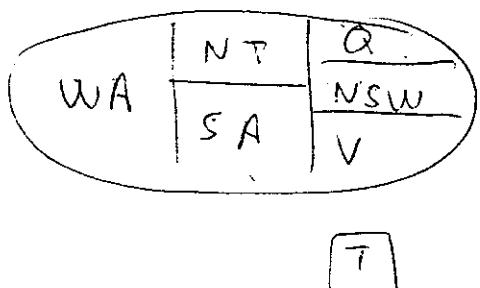
$$C = A \cap B \Leftrightarrow \mu_C(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x) \text{ para } \forall x \in X$$

$$C = A \cup B \Rightarrow \mu_C(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x) \text{ para } \forall x \in X$$

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \text{ para } \forall x \in X$$

• Problemas PSR

① Australia coloreada



$$X = \{ WA, SA, NSW, Q, V, NT, T \}$$

$D = \{ \text{rojo}, \text{azul}, \text{verde} \}$ para cada $x_i \in X$

$$C = \{ \langle (x_i, x_j) \text{ tal q. } x_i \text{ toca a } x_j, (\text{color}(x_i) \neq \text{color}(x_j)) \rangle \}$$

② Clases y horarios

$$X = \{ EOC, EDD, \dots \}$$

$$D = \{ (\text{lunes}, 12), (\text{Miércoles}, 13), \dots \}$$

③ N reinos

$$\text{Var} \{ V_1, V_2, \dots, V_n \}$$

$$\text{Don } \{ D_i : \{ (1, 2), (1, 2), \dots \}, D_{i+1} \dots \}$$

Utilizando: prob. busqueden en espacios de estados

$$N = 4$$

