

Paradigma Lógico

Lógica: Definición

- Ciencia que expone las leyes, modos y formas del conocimiento científico.
- Disciplina que estudia los principios formales del conocimiento humano, es decir, las formas y las leyes más generales del pensamiento humano considerado puramente en sí mismo, sin referencia a los objetos.

Supongamos que es sábado en la noche y acaba usted de llegar a una gran fiesta. Sintiéndose un poco tímido, se pregunta para sus adentros si conocerá a alguna de los invitados a la fiesta. Su anfitrión le dice "usted seguramente conoce a Rita, la mujer que se encuentra en la esquina cerca de la bandeja de las bocas". En una fracción de segundo usted es capaz de echarle una mirada a Rita y verificar si realmente su anfitrión está en lo correcto o no. Sin embargo, en ausencia de esta sugerencia de su anfitrión, estaría obligado a realizar una revisión exhaustiva por todo el salón, chequeando las personas de la fiesta una por una, para ver si entre ellas hay alguna conocida.

Este es un ejemplo de un fenómeno general según el cual, en la mayoría de los problemas de decisión, generar una solución es mucho más difícil que comprobar que una solución específica es correcta. Por ejemplo, si alguien nos dice que el número 13.717.421 puede ser escrito como el producto de dos números menores, puede ser que usted no sepa si creerle o no, pero si esta misma persona le dice que, en efecto, tal número se escribe como el producto de 3607 por 3803, entonces usted puede muy fácilmente comprobarlo para ver si es cierto, con la ayuda de una calculadora de bolsillo o manualmente.

Muchas tareas pueden ser formuladas como “dado un objetivo”, como alcanzarlos?. A la IA le queda resolver esto cuando no hay un algoritmo sistemático o directo.

Posible Solución: probar todas las alternativas posibles, mediante ensayo y error, con la esperanza de hallar en algún momento la solución. Será necesario seleccionar una acción a seguir. Cada acción adoptada abre nuevas posibilidades, una especie de ramificación, denominada árbol de búsqueda.

Si el árbol es muy grande, ninguna computadora podrá explorar las consecuencias de todas las acciones posibles, por esta razón es necesario “podar el árbol”, es decir, dotar a los programas de reglas heurísticas, de manera que las acciones más prometedoras sean objeto de análisis especial y las menos interesantes sean eliminadas o postergadas.

La IA es la ciencia que estudia la forma de diseñar programas de computación que exhiban características que comúnmente asociamos con el comportamiento humano inteligente.

Hay problemas que fueron resueltos y todas las inferencias se hicieron previamente de manera que se pueda trabajar mediante un procedimiento definible formalmente, sabiendo que cada paso es un paso en el camino de solución, es decir, mediante un algoritmo.

El requisito para que un proceso sea calificado como inteligente es que posea un carácter “no estructurado y no determinístico”. La solución debe depender del conocimiento empleado y del método utilizado para aplicarlo.

Qué es la inteligencia humana?

La habilidad de razonar

La habilidad de aplicar y adquirir conocimiento.

La habilidad de comunicar ideas

Etc.

Diferencias entre programación convencional y programación en IA

Programación Convencional	Programación en IA
Algoritmos (pasos de solución explícitos)	Búsqueda (pasos de solución implícitos)
Información y control integrados	Estructura de control separados del conocimiento
Dificultad de Modificación	Facilidad de Modificación
Requerimiento de respuestas óptimas	Aceptación de respuestas satisfactorias

Problema

Una persona se enfrenta con un problema cuando tienen que llegar a un objetivo y no conoce la acción o serie de acciones que debe seguir para conseguirlo. Pueden haber muchos caminos para solucionarlo y esas decisiones influyen en la efectividad y eficiencia de resolver el problema.

$P = (I, O, C)$

I: conjunto de expresiones que representan el estado o situación inicial.

O: conjunto de todas las operaciones o transformaciones que se pueden efectuar sobre la situación inicial o sobre expresiones derivadas a partir de ella mediante alguna secuencia de operaciones previa.

C: es la condición que debe satisfacer la expresión terminal.

Diremos que el estado I' es sucesor del estado I , si es alcanzable a partir de I mediante la aplicación de alguna secuencia de operadores. Si I' puede ser alcanzado con la aplicación de un operador, diremos que I' es sucesor inmediato de I o que I genera I' .

Llamaremos acción al resultado de aplicar el operador O_j a una expresión.

El conjunto de todos los estados que pueden ser alcanzados aplicando operadores a partir del estado inicial, se denomina **Espacio de búsqueda o espacio de estados**.

Una secuencia de operadores $O_1 \dots O_n$ constituye una solución a un problema si el resultado de su aplicación al estado inicial I , satisface la condición C .

Si $C(O_n(\dots(O_1(I)))) \Rightarrow O_1 \dots O_n$ es la solución de (I, O, C)

- Si se puede probar que no existe tal secuencia, entonces el problema no tiene solución.
- Si la secuencia es vacía el problema ya estaba resuelto es decir $I=C$.

No introducir ciclos en las soluciones

Se podrá pedir una solución al problema, todas las soluciones o la solución **óptima** (no existe otra solución que la mejore). Puede ser **semi-óptima**.

Ejemplos

Problema del viajante de comercios:

El viajante de comercio debe visitar un conjunto de ciudades. Cada ciudad está conectada con las demás mediante carreteras de longitud conocida. El problema consiste en hallar la ruta que deberá tomar para visitar todas las ciudades retornando a la ciudad de partida, de modo tal que la distancia recorrida sea mínima.

Torres de Hanoi

En un lugar muy cercano a Hanoi, existe un monasterio cuyos monjes dedican sus vidas a una tarea muy importante: en un jardín hay tres postes, en los cuales están distribuidos 64 discos de piedra, de distintos radios, cada uno con un agujero en el centro. Cuando se creó el monasterio, todos los discos se hallaban apilados en uno de los postes, cada uno sobre el de tamaño inmediatamente mayor, es decir, el más grande abajo y el más pequeño arriba. La tarea de los monjes consiste en traspasar todos los discos a otro de los postes.

Solo un disco debe ser movido a la vez y los restantes deben permanecer en su sitio. Además, durante el proceso, no puede ponerse un disco sobre otro de tamaño menor a él. Cuál es la secuencia mínima que pueden usar los monjes para terminar su tarea? (como mínimo $2^{64} - 1$ movimientos, lo cual es una suerte porque la leyenda dice que cuando los monjes terminan su tarea será el fin del universo)

15-Puzzle

Sea un tablero de 4 x 4 casilleros y 15 piezas numeradas del 1 al 15. El problema consiste en transformar la configuración inicial en una configuración final determinada previamente. El único movimiento permitido es trasladar una pieza hacia el casillero adyacente, siempre y cuando el mismo este vacío.

1	2	11	12
10		3	13
7	5	15	4
9	6	8	14

→

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Representación de problemas

Los problemas rara vez se resuelven en exactamente los mismo términos en los que se expresó inicialmente. Lo habitual es utilizar un conjunto de convenciones para representar la información y realizar una descripción del problema mediante ellas. No se resuelve el problema real, sino una representación idealizada del mismo, que a su vez hay que interpretar.

La noción de representación de conocimiento está relacionada con la descripción, en algún lenguaje, de las situaciones que corresponden al mundo o a un estado del mundo. De

forma tal que una máquina pueda sacar nuevas conclusiones sobre ese mundo manejando esas representaciones simbólicas.

Nosotros vamos a utilizar la lógica de primer orden como lenguaje de representación de conocimiento por lo tanto debemos entender como funciona el cálculo de primer orden.

Una vez especificado el problema, puede utilizarse un sistema de resolución para explorar el espacio del problema buscando un camino entre el estado inicial y el estado objetivo.

Como representamos las operaciones?

Representamos a cada operador como la relación entre el estado anterior (antes de aplicar el operador) y el estado posterior (después de aplicar el operador). Cada estado será representado mediante una lista que podrá contener variables.

La relación será entonces una regla que tendrá un **antecedente** (formado por la configuración del estado anterior y las condiciones que éste debe cumplir para poder ser aplicado el operador) y un **consecuente** (Formado por las acciones a realizar y la configuración del nuevo estado). Las condiciones y acciones serán expresadas mediante conjunciones de predicados.

Formato de una regla

```
Nombre {parametros} =
    Estado-anterior {/ condiciones} ->
    Estado-posterior {/ acciones}
```

Ejemplo de representación

Problema de las jarras

El espacio de estado puede ser descrito mediante un par de entero $[X, Y]$, tal que X varía entre 0 y 4 e Y entre 0 y 3. X e Y representan la cantidad de litros que contienen las jarras de capacidad 4 y 3 litros respectivamente.

El estado inicial es $[0, 0]$ y la condición de solución es alcanzar el estado $[2, Z]$, pues no tienen importancia el contenido de la segunda jarra.

Los operadores que describen las acciones son los siguientes:

```
[llenar la jarra 1] =
     $[X, Y] / X < 4 \rightarrow [4, Y]$ 
[llenar la jarra 2] =
     $[X, Y] / Y < 3 \rightarrow [X, 3]$ 
[vaciar la jarra 1] =
     $[X, Y] / X > 0 \rightarrow [0, Y]$ 
[vaciar la jarra 2] =
     $[X, Y] / Y > 0 \rightarrow [X, 0]$ 
[tirar cierta cantidad de agua de la jarra 1] =
     $[X, Y] / X > 0 \rightarrow [W, Y] / W = X - Z$ 
[tirar cierta cantidad de agua de la jarra 2] =
     $[X, Y] / Y > 0 \rightarrow [X, W] / W = Y - Z$ 
[volcar el contenido de 1 en 2] =
     $[X, Y] / Z = X + Y, Z \leq 3, X > 0 \rightarrow [0, Z]$ 
[volcar el contenido de 2 en 1] =
     $[X, Y] / Z = X + Y, Z \leq 4, Y > 0 \rightarrow [Z, 0]$ 
```

[volcar el contenido de 1 en 2 hasta que se llene 2] =
 $[X, Y] / Z = X + Y, Z > 3, X > 0 \rightarrow [W, 3] / W = Z - 3$
 [volcar el contenido de 2 en 1 hasta que se llene 1] =
 $[X, Y] / Z = X + Y, Z > 4, Y > 0 \rightarrow [4, W] / W = Z - 4$

Problema del pastor

En este problema, se puede representar a cada estado mediante una lista de cuatro elementos [P, L, O, R], que indica la orilla (norte o sur) en que se encuentran el pastor, el lobo, la oveja y el repollo respectivamente.

Estado inicial: [norte, norte, norte, norte]

Estado final: [sur, sur, sur, sur]

Reglas que establecen los cruces del río en el bote.

[ir hombre solo] =
 $[P, L, O, R] / \text{no_come}(L, O), \text{no_come}(O, R)$
 $\rightarrow [P', L, O, R] / \text{opuesto}(P, P')$

[ir hombre con lobo] =
 $[X, X, O, R] / \text{no_come}(O, R)$
 $\rightarrow [Y, Y, O, R] / \text{opuesto}(X, Y)$

[ir hombre con oveja] =
 $[X, L, X, R]$
 $\rightarrow [Y, L, Y, R] / \text{opuesto}(X, Y)$

[ir hombre con repollo] =
 $[X, L, O, X] / \text{no_come}(L, O)$
 $\rightarrow [Y, L, O, Y] / \text{opuesto}(X, Y)$

opuesto (norte, sur)

opuesto (sur, norte)

no_come (X, Y) :- opuesto (X, Y)

Sistemas de Producción

Algoritmos vs Búsqueda

Un algoritmo halla la solución del problema en forma directa sin examinar distintas alternativas. Describe una descripción exacta sobre la realización, en una secuencia determinada, de acciones conducentes en un número finito de pasos a la solución de una clase específica de problemas.

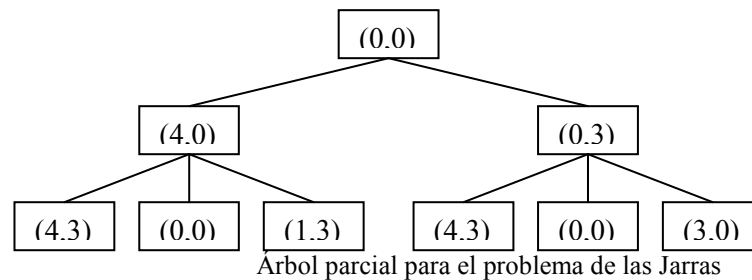
Un proceso de búsqueda consiste en ensayar exhaustiva y sistemáticamente todas las operaciones permitidas. Aplicando este procedimiento, es seguro que se hallará la solución, si al menos existe alguna. Si las combinaciones son infinitas y el problema es insoluble, el computador no se detendrá nunca.

Búsqueda en grafos

El proceso de búsqueda establece un isomorfismo entre encontrar la secuencia de operadores que solucione el problema y encontrar un camino a través de un grafo dirigido.

Cada nodo del grafo representa un estado I_k del problema. Existirá un arco entre el nodo i y j si y solo si existe un operador O que transforme a I_i en I_j .

Un árbol es un grafo en el cual cada nodo, excepto el nodo inicial, tiene un solo antecesor. Un nodo sin sucesores se denomina hoja.



Un árbol puede ser construido totalmente a partir de las reglas que definen los estado alcanzables desde el estado inicial (para problemas reales puede ser demasiado grande), por lo tanto el árbol no se armará totalmente sino que se considerará implícitamente representado por las reglas y se generará explícitamente solo aquellas partes que se decida explorar.

Sistemas de Producción

Se distinguen los siguientes componentes:

Base de datos global: contiene toda la información perteneciente al problema

Reglas: operadores del problema y actúan sobre la base de datos.

Estrategia de control: mecanismo que examina la base de datos determinando que reglas aplicar y resolviendo eventualmente el conflicto originado cuando más de una es aplicable.

Algoritmo de control

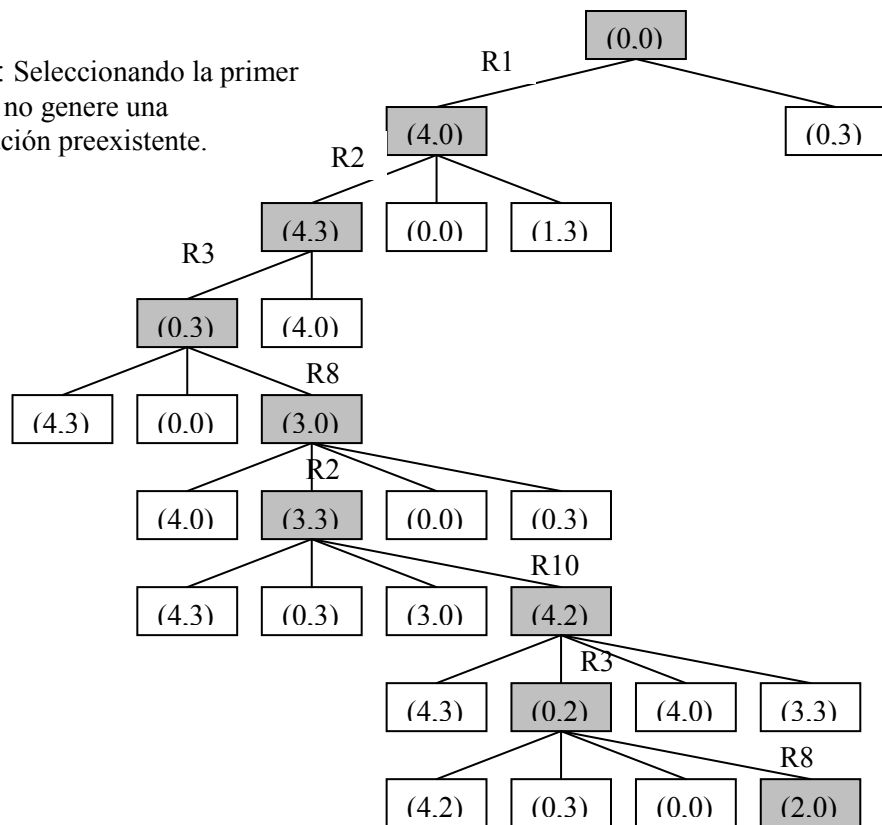
Es un proceso que:

1. Extrae las reglas aplicables (aquellas que satisfacen la precondition)
2. Selecciona la regla a aplicar del conjunto disponible
3. Aplica la regla seleccionada.

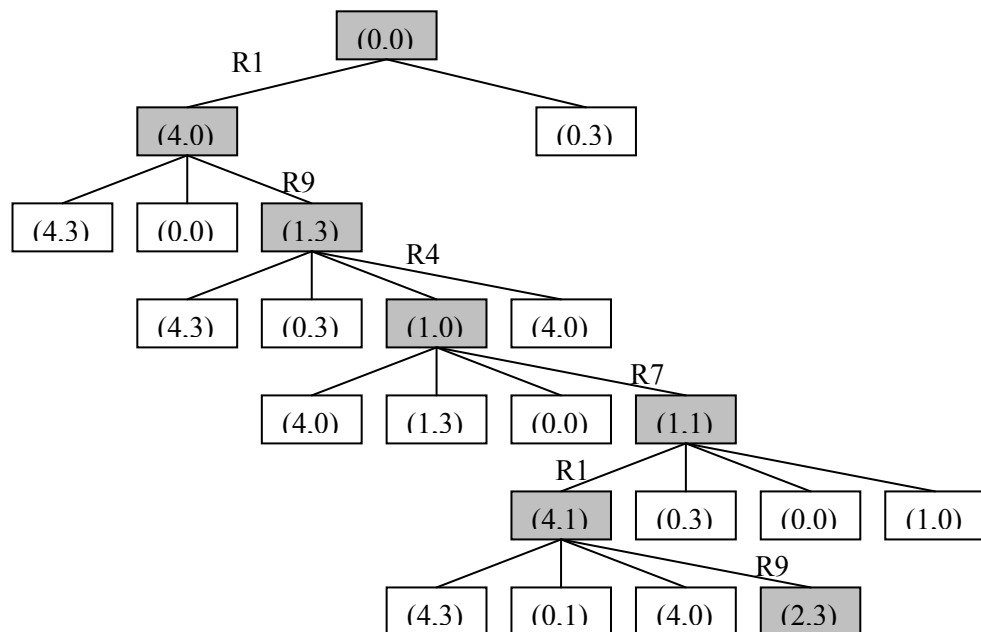
Requisitos

1. Causar movimiento (evitar que se llegue al mismo estado para no obtener ciclos)
2. Ser sistemática (no al azar – seguir un comportamiento predeterminado – primero en profundidad o primero en amplitud)

Ejemplo: Seleccionando la primer regla que no genere una configuración preexistente.



Seleccionando la regla en que la diferencia del contenido de la primera jarra con respecto a 2 fuese mínima



Dirección de la Búsqueda

El objetivo del procedimiento de búsqueda es encontrar un camino entre la configuración inicial y la final.

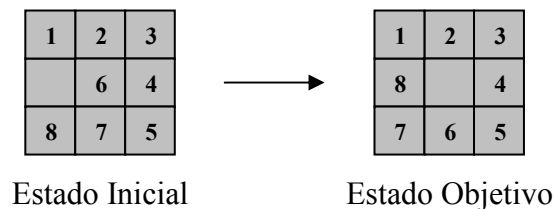
Existen dos direcciones:

Forward (hacia delante): consisten en aplicar operadores al estado inicial, luego a sus sucesores y así sucesivamente hasta alcanzar el estado final.

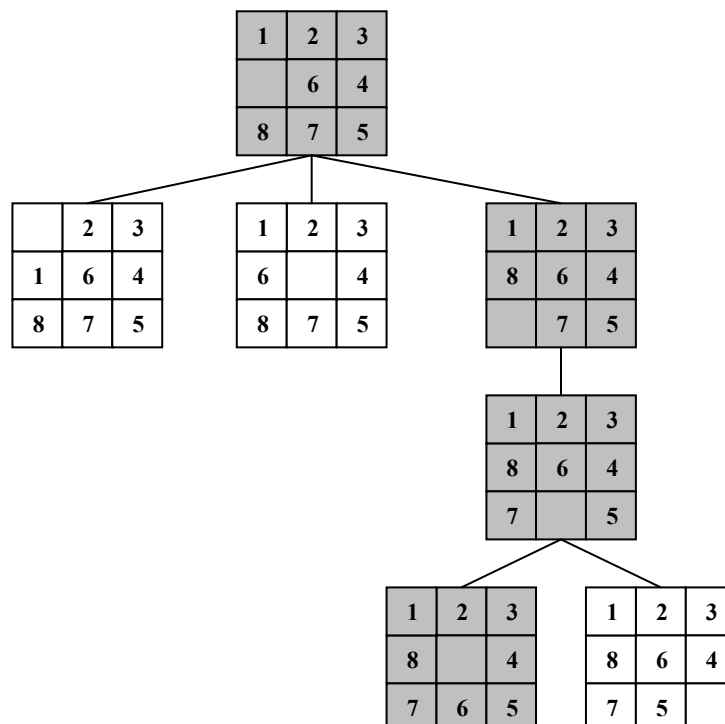
Backward (hacia atrás): consiste en aplicar los operadores al estado objetivo, que es convertido en uno o mas subobjetivos talque que sus soluciones son suficientes para resolver el objetivo original. Estos subobjetivos son reducidos a su vez a subobjetivos hasta que cada uno de ellos tenga una solución trivial.

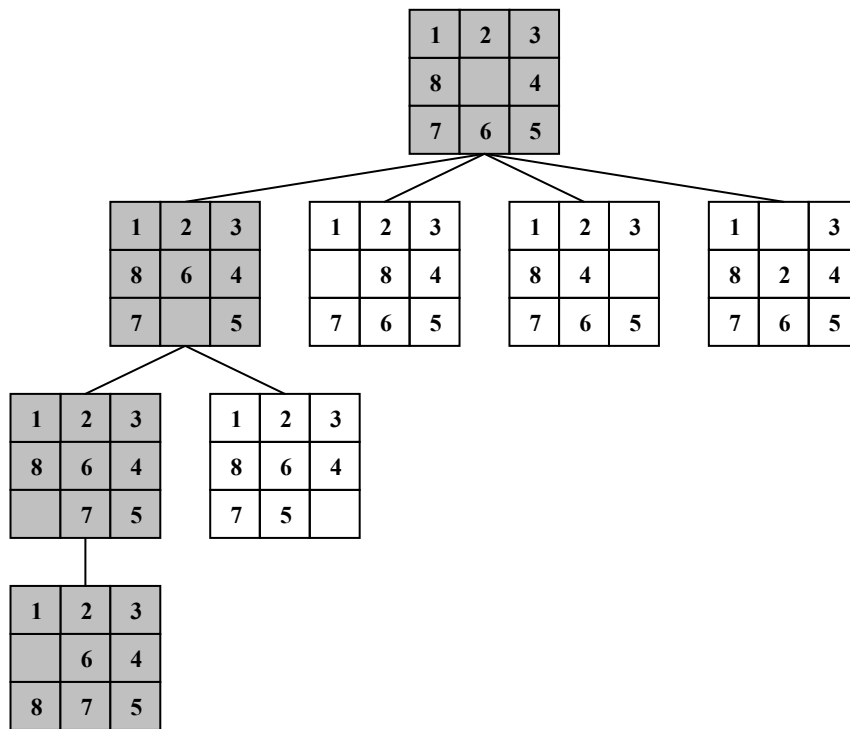
Ejemplo:

8-puzzle: se desea pasar del estado inicial al estado final.

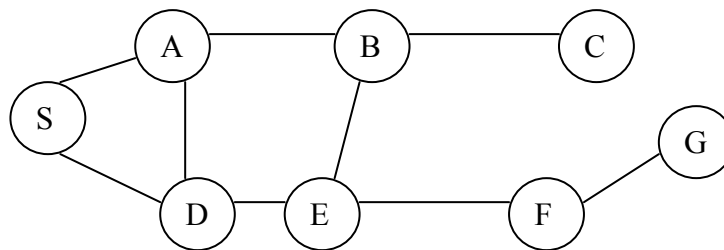


Forward



Backward**Métodos de Búsqueda**

Consideremos el caso donde haya que encontrar un camino desde la ciudad S a la G.



Caminos posibles (que surgen de explorar todo el árbol de búsqueda)

S-A-B-E-F-G
 S-D-A-B-E-F-G
 S-D-E-F-G
 S-A-D-E-F-G

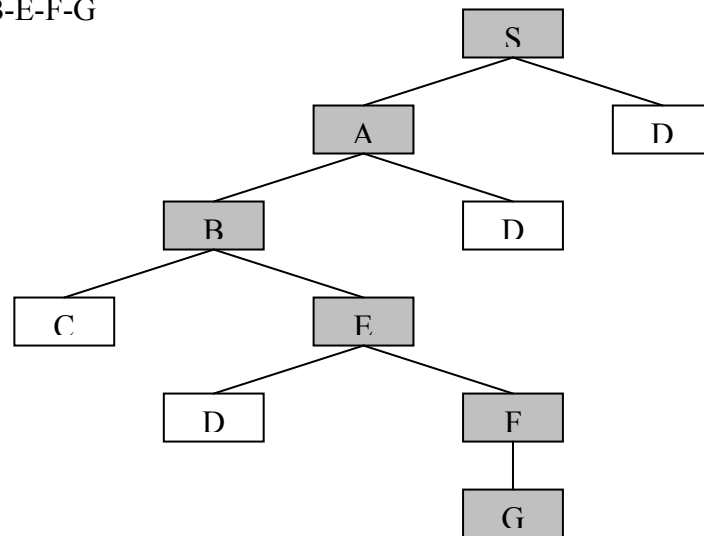
Métodos débiles**Primero en Profundidad**

Una vez expandida la raíz, se elige la primera (por convención las más a la izquierda) y se expande esta ignorando momentáneamente el resto.

Se sigue de esta manera hacia lo profundo hasta encontrar la solución o hasta que no haya más sucesores.

Si no hay más sucesores se retrocede (backtraking) hasta el nodo más reciente que contenga una alternativa no explorada.

Solución: S-A-B-E-F-G



Necesita menos memoria

Puede encontrar una solución sin tener que explorar gran parte del espacio de estados.

Puede encontrar una solución no mínima.

Cómo será haciendo encadenamiento hacia atrás?

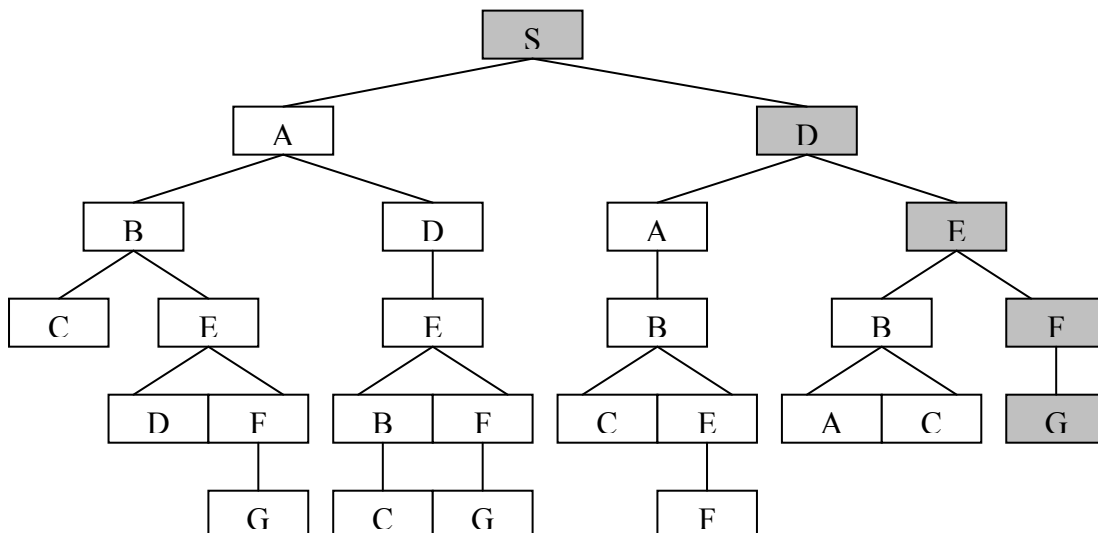
Primero en Amplitud

Se expande el estado inicial y se obtienen todos su hijos.

Se continua expandiendo cada uno de estos y solo se procede a expandir los nuevos hijos cuando se termina con el nivel actual. (Se recorre todos los nodos del nivel antes de pasar al siguiente).

(La implementación realizada controla que se cumpla la condición de solución cuando el nodo es seleccionado para expandir – *puede ser mejorado*)

Solución: S-D-E-F-G



Funciona con árboles de profundidad infinita pues en caso de existir una solución, esta será alcanzada por el método. (garantiza su hallazgo)
 Si existen soluciones múltiples alcanzará la solución mínima. (mínimo número de pasos)
 No queda atrapada en caminos que no conducen a estados objetivos.
 El inconveniente que tiene es que utiliza demasiada memoria para guardar los nodos.

Búsqueda Heurística

Como sabemos, el proceso de búsqueda consiste en probar exhaustiva y sistemáticamente todas las operaciones permitidas hasta llegar a la solución. Puede ocurrir una explosión combinatoria.

La eficiencia se puede mejorar si utilizamos información adicional que nos permita reducir la búsqueda es decir **información heurística**.

La heurística consiste en un criterio para:

- Elegir entre alternativos cursos de acción
- Refinar el conjunto de sucesores de un nodo para podar ciertas porciones del árbol de búsqueda.

La heurística introducirá eficiencia, posiblemente sacrificando optimalidad y completitud, pues la misma restringe el espacio de búsqueda evitando explorar todas las alternativas posibles.

Heurísticas de propósito general: utilizar tácticas localmente optimas, seleccionando buenas alternativas locales en cada paso.

Ej: (Viajante de Comercio) Seleccionar la ciudad que esté más cerca de la actual.

(Jarras de agua) Seleccionando la regla en que la diferencia del contenido de la primera jarra con respecto a 2 fuese mínima

Incorporación de heurísticas al proceso de búsqueda

- **Reglas heurísticas:** reglas que representan las convicciones del experto en el área del problema
- **Función heurística:** estimación de cuan efectivo puede ser el estado sucesor para alcanzar la solución.

Ej: Cantidad de fichas que están fuera de lugar en el 8-puzzle. $[W(n)]$

La suma para todas las fichas de la distancia desde la posición actual a la que ocupará en el objetivo. $[P(n)]$

- **Eliminación heurística:** criterio que establece que porciones del espacio de estado es posible eliminar y no considerar

Ej: la solución no tendrá mas de N niveles, por lo tanto no explora más allá de N niveles.

Ej. de función heurística

2	1	6
4	8	
7	5	3

$W(n) = 8$

$P(n) = 11$