Capitulo 6

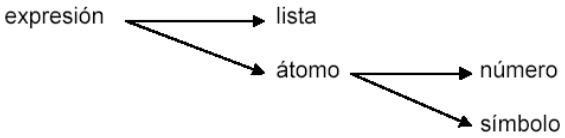
Introducción al Lenguaje LISP

6.1 Lisp

El nombre de LISP surge de abreviar "List Procesing" o Procesamiento de Listas y el cual es un programa interprete basado en el lenguaje funcional siendo un hibrido ya que me permite realizar bucles y asignar variables.

En LISP toda la información (inclusive las sentencias) se expresan como listas o como los componentes de las mismas.

Un código de LISP está compuesto de formas o expresiones; el intérprete LISP lee una expresión, la evalúa e imprime el resultado. Este procedimiento se denomina ciclo de lectura, evaluación.



Una expresión o forma es una lista o un átomo.

Una **lista** es una sucesión de cero o más expresiones entre paréntesis. El blanco (no la coma) es el separador de expresiones dentro de una lista. Cada una de estas expresiones se denomina elemento de la lista.

Un **átomo** es un símbolo o un número.

Un **símbolo** se representa por un nombre que está formado por caracteres alfanuméricos que no puedan interpretarse como números.

Un **número** se puede escribir en cualquiera de las notaciones habituales.

Como evalúa las Listas LISP

Si la forma es una lista LISP trata el primer elemento como el nombre de una función, y evalúa los restantes elementos recursivamente, y luego llama a la función con los valores de los elementos restantes.

Cuando LISP evalúa una expresión, devuelve un valor (que será otra expresión) o bien señala un error.

- Un número se evalúa a sí mismo.
- Un símbolo se evalúa al valor que tiene asignado.
- Una lista se evalúa independientemente una por una del siguiente modo, siempre en función del primer elemento y en función de este considera como emplear los elementos siguientes:

1) **El primer elemento** de la lista debe ser un símbolo "s" cuyo significado funcional "F" esté definido.

2) Se evalúan los restantes elementos de la lista, obteniendo los valores v1,…,vn. Si el número o tipo de los valores obtenidos no es coherente con los argumentos requeridos por F, se produce un error.

3) La lista se evalúa a F(v1,…,vn).

La forma en la que se evalúan las listas corresponde a una notación pre fija, donde primero se escribe el operador y luego los parámetros.

6.2 Funciones de Asignación. Setq

>(setq a 5). Almacena el número 5 en a

6.3 Operadores

Operadores Aritméticos

>(+ a 5) Suma el valor que tiene a más 5.

Operadores de Igualdad.

La igualdad numérica es denotada por =. >(= 3 3)
El predicado **eq** está pensado para comparar símbolos. >(eq `a `a)
El predicado **eql** sirve para números y símbolos.
El predicado **equal** sirve para comparar listas. >(equal `(a b c) `(a b c))

6.4 Funciones para el manejo de listas. Car. Cdr. Append. List. Reverse

Car permite accede al primer elemento de una lista y **cdr** a los restantes
>(**setq** lst `(a b c))
>(**car** lst)
a
>(**cdr** lst)
b c

List permite construir una lista
>(list 4 5 6)
(4 5 6)

Append, Reverse.
>(**append** `(1 2 3) `(4 5 6)) concatena listas.
(123456)
>(**reverse** `(1 2 3)) invierte elementos de la lista.
(3 2 1)

Capitulo 7

Sistemas Expertos Y Programación Lógica

¿QUE SON LOS SISTEMAS EXPERTOS?

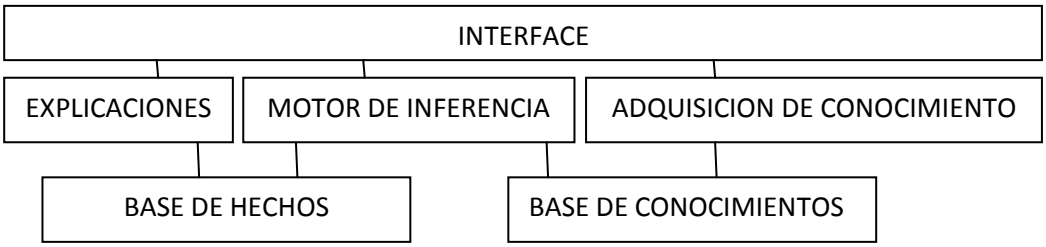
Es un programa destinado a generar inferencias en un área específica del conocimiento en forma similar a la que se espera de un experto humano. Deben resolver problemas por aplicación de conocimiento en un dominio específico. Este conocimiento es adquirido a través de la intervención de expertos humanos y almacenado en lo que se denomina Base de Datos de Conocimiento

Se utiliza SE cuándo:

- No existe algún algoritmo para resolver un problema
- El problema es resuelto satisfactoriamente por expertos humanos
- Existe algún experto humano que pueda colaborar en el desarrollo de un SE
- El conocimiento del dominio debe ser relativamente estático

Símbolos: Las diferentes definiciones que se van a utilizar en el SE, afirmaciones, elementos, etc.
Reglas: Se aplican sobre los símbolos, se deben obtener del conocimiento de los expertos. Son heurísticas dado que no es posible demostrar su validez general.
Hechos: son todos los predicados que se suponen verdaderos.

Arquitectura de un Sistema Experto



Base de conocimientos: Es la parte del sistema experto que contiene el conocimiento sobre el dominio. Hay que obtener el conocimiento del experto y codificarlo en la base de conocimientos. Una forma clásica de representar el conocimiento en un sistema experto son lar reglas

Base de hechos (Memoria de trabajo): Contiene los hechos sobre un problema que se han descubierto durante una consulta. Durante una consulta con el sistema experto, el usuario introduce la información del problema actual en la base de hechos.

Motor de inferencia: El sistema experto modela el proceso de razonamiento humano con un módulo conocido como el motor de inferencia. Dicho motor de inferencia trabaja con la información contenida en la base de conocimientos y la base de hechos para deducir nuevos hechos.

Explicación: Una característica de los sistemas expertos es su habilidad para explicar su razonamiento. Usando el módulo del subsistema de explicación, un sistema experto puede proporcionar una explicación al usuario de por qué está haciendo una pregunta y cómo ha llegado a una conclusión.

Interface/Interfaz de usuario: La interacción entre un sistema experto y un usuario se realiza en lenguaje natural. También es altamente interactiva y sigue el patrón de la conversación entre seres humanos. Un requerimiento básico del interfaz es la habilidad de hacer preguntas.

Lógica de Predicados y Lógica Proposicional

Los sistemas expertos básicos se basan en la lógica de predicados. La diferencia entre esta y la proposicional es que la lógica de predicados emplea variables y permite expresar una premisa en una sola proposición mientras que en la otra deberíamos usar una proposición por cada una de las variables que estemos analizando. La representación mediante predicados es una forma de representar la estructura del conocimiento.

Tipos de encadenamiento (modelos fundamentales de SE)

1- Encadenamiento hacia adelante: toma una serie de afirmaciones de entradas y ensaya todas las reglas disponibles una y otra vez incorporando nuevas afirmaciones hasta que ninguna de las reglas produzca nuevas. Se emplea para verificar hipótesis. Parto de datos provistos por el usuario. Siempre el encadenamiento es dirigido por la información que se incorpora.

- Correspondencia: busca afirmaciones en la BD que correspondan con los antecedentes de una regla.
- Resolución: en caso de que se cumplan los antecedentes instanciar el consecuente produciendo una nueva afirmación.

2- Encadenamiento hacia atrás: comienza por hipótesis y tratamos de verificarla haciendo uso de las afirmaciones disponibles.

- Correspondencia: busca una afirmación en la BD que corresponda a una hipótesis.
- Unificación: busca una regla cuyo consecuente concuerde con la hipótesis.
- Resolución: para cada regla cuyo consecuente concuerde con la hipótesis, se intenta verificar de manera recursiva cada antecedente de la regla.

Lógica de Predicados:

Lógica de primer orden es el formalismo más utilizado para representar conocimiento en IA.
Cuenta con un lenguaje formal mediante el cual se representa por formulas llamadas axiomas, que permiten describir fragmentos del conocimiento y además consta con reglas de inferencia que al aplicarlas a los axiomas genera nuevos conocimientos.
La **inferencia** es el proceso mental por el cual se extraen conclusiones a través de premisas más o menos explícitas.
La LP tiene dos tipos de símbolos: **lógicos** (disyunción, conjunción, etc.) y **no lógicos** (func. relac.)
A partir de símbolos se construyen dos tipos de expresiones: términos (nombre objeto de universo) y formulas (afirmar o negar propiedades de los objetos del universo).

Formulas bien formadas:

Es una proposición simple o compuesta cuyo valor de verdad puede ser determinado, basado en los valores de veracidad de las proposiciones simples y en la naturaleza de los conectores lógicos involucrados.

Cláusula de Horm:

Es una conjunción de disyunciones con no más de un literal positivo.
Una expresión es valida satisfacible si alguna combinación de valores v o f de sus átomos hacen verdadera la expresión.

- Importancia: para aplicar la lógica en programas de computación ha sido necesario encontrar algoritmos que determinen si una expresión lógica válida es satisfacible.

Un algoritmo más rápido puede aplicarse si la expresión está en la forma de clausula de horm.

Resolución:

La programación lógica está basada en el principio de resolución. La resolución es una regla que se aplica sobre cierto tipo de formulas de cálculos de predicados de primer orden, llamadas clausulas. La demostración de teoremas bajo esta regla de inferencia se lleva a cabo por reducción al absurdo. Es en esencia un procedimiento iterativo cuya finalidad es evaluar la verdad o falsedad de una premisa. En cada paso dos clausulas padres son resueltas para obtener una tercera denominada resolvente
El formalismo más utilizado en la programación lógica es la lógica de predicados.

UNIFICACIÓN Y LIGADURA

Permite razonar dentro de un esquema formal sin relación con la codificación de los programas. En el caso de un código, este se encarga de reducir las formas clausales a listas asociadas más simples de manipular denominadas ligaduras, esta fluye a través de los filtros del programa y al final del proceso pueden ser analizadas para extraer nuevas afirmaciones o demostrar hipótesis.

CORRESPONDENCIA DE PATRONES

Relacionado con la codificación de SE es la correspondencia de patrones simbólicos. La correspondencia con patrones se basa en identificar si dos vectores son similares es decir se corresponden. La forma simple de entenderlo es calcular la distancia entre los mismos y si esta es lo suficientemente pequeña de acuerdo a valores preestablecidos para un problema dado, entonces puede decirse que ambos vectores se corresponden.

CORRESPONDENCIA SIMBÓLICA

Busca similitudes entre un conjunto de nombres de variables y no entre los valores numéricos de estas. Se consideran dos procedimientos:

Correspondencia: encadenamiento hacia adelante, comparando una expresión común (dato) con un patrón. A diferencia de los datos los patrones pueden contener variables patrón, para facilitar el reconocimiento de estas variables se asocian a un símbolo.

Unificación: encadenamiento hacia atrás. Hace corresponder dos patrones en lugar de un patrón y un dato.

CORRESPONDENCIA SUB SIMBOLICA

Correspondencia numérica.

ENCADENAMIENTO.

PROGRESIVO

- **Correspondencia**: Busca afirmaciones en la base de datos que correspondan con los antecedentes de una regla.
- **Resolución:** En caso de que se cumplan los antecedentes instanciar el consecuente produciendo una nueva afirmación.

REGRESIVO

- **Correspondencia:** Busca afirmaciones en la base de datos que correspondan a una hipótesis.
- **Unificación:** buscar una regla cuyo consecuente concuerde con la hipótesis. La hipótesis y el consecuente pueden tener variables de patrón.
- **Resolución:** para cada regla cuyo consecuente concuerde con la hipótesis, se intenta verificar de manera recursiva cada antecedente de la regla, considerando a cada una como una hipótesis.

Capitulo 8

Inferencia y Probabilidad. Lógica Difusa

QUE ES LA LOGICA DIFUSA

La lógica difusa es una metodología que proporciona una manera simple y elegante de obtener una conclusión a partir de información de entrada vaga, ambigua, imprecisa, con ruido o incompleta, en general la lógica difusa imita como un persona toma decisiones basada en información con las características mencionadas.

La lógica difusa se basa en los conjuntos difusos. Un conjunto difuso A en X se define por los pares ordenados (x,pA (x)). Siendo mA (x) la función de pertenencia de x en el conjunto.

Inferencia Difusa

La inferencia difusa y el razonamiento difuso como alternativa a las técnicas más conocidas de inferencia y razonamiento probabilística, cuando se presentan incertidumbres de tipo de vaguedad y se quiere involucrar el conocimiento experto y la experiencia.

El método de obtener conjuntos difusos a partir de la combinación de otros conjuntos difusos con reglas de la forma SI ... ENTONCES, es lo que se denomina «Inferencia Difusa».

Considérese una implicación de la forma: SI X es A ENTONCES Y es B
(antecedente) (consecuente)

En donde A y B son valores lingüisticos y están modelados por conjuntos difusos. La expresión describe una relación entre dos variables X y Y:A - > B . Por ejemplo:

«Si la temperatura es alta entonces la demanda de energía es alta»

Etapas de la Lógica Difusa

- I Difusión de entradas: Se deben resolver todas las sentencias en el antecedente en función de su grado de membresía entre 0 y 1. Si el antecedente tiene solo un componente este es el grado de soporte del consecuente.
- II Aplicación de operadores difusos: Si existen multiples partes en el antecedente se aplican los operadores fuzzy y se resuelven los antecedentes como un número entre 0 y 1.
- III Aplicación de implicación: Se emplea el grado de soporte de la regla para conformar el conjunto fuzzy de salida.
- IV En general, una sola regla no es muy útil. Y por lo tanto se requieren dos o mas reglas que interactúen entre ellas.

Pasos

1. Fuzificación:
   El primer paso es tomar las entradas y determinar el grado en el cual ellos pertenecen a cada uno de los conjuntos difusos a través de funciones de membresias o pertenencias.
2. Aplicación del operador difuso:
   Realizamos el cálculo de valores de pertenecía.
3. Aplicación del método de Implicación:
   El método de implicación se define como la conformación del consecuente basado en el antecedente. La entrada para la implicación es un numero dado por el antecedente, y la salida un conjunto de fuzzy.
4. Agregación:
   Es el proceso de unificar las salidas para cada regla uniendo los procesos paralelos.

Es decir se combinan las salidas en único conjunto difuso preparándolas para el paso final.

5. Defuzzificación:
   La entrada para el proceso de defuzzifiación es el conjunto difuso agregado y la salida es un número.
   Entonces dado un conjunto difuso que engloba un rango de valores de salida se requiere obtener un único número.
   La defuzzificación tiene lugar en dos distinto pasos. Primero las funciones de pertenencias son escaladas de acuerdo a sus posibles valores, luego estas son usadas para calcular el centroide de los conjuntos difusos asociados.

Capitulo 9

Búsqueda

9.1 El concepto de búsqueda en el campo de la IA

Procedimiento cuya finalidad es encontrar datos dentro de un conjunto de estos, pero donde el objetivo del proceso es más bien encontrar cierto camino recorrido hasta encontrar en dato antes de corroborar la existencia del dato. Esto se debe a que estos procesos se usan en planificación, optimización, resolución de problemas, juegos, etc. Para realizar las búsquedas se utilizan arboles y grafos para conocer la secuencia de estados que deben alcanzarse sucesivamente para lograr un estado objetivo.

METODOS NO INFORMADOS

9.2. PRIMERO EN ANCHURA BPA

Se generan para cada nodo todas las posibles situaciones resultantes de la aplicación de todas las reglas adecuadas. Se continúa:

1. Se crea una pila (abierta) y se le asigna el estado inicial.
2. Mientras la pila no esté vacía.
   a. Extraer el primer nodo de la pila llamarlo m.
   b. Expandir m. Se generan las entradas sucesoras del estado actual, para cada operador aplicable hacer:
      i. Aplicar el operador al nodo obteniendo un nuevo estado.
      ii. Si el nuevo estado es el objetivo termina el proceso.
      iii. Incluir el nuevo estado el final de la pila volver a 2ª

Ventajas: No entra en callejones sin salida, si existe una solución garantiza alcanzarla, si existen varias soluciones garantiza alcanzar la optima.
Desventajas: utiliza mucha memoria, es lento.

9.3 PRIMERO EN PROFUNDIDAD BPP

Continúa por una sola rama del árbol hasta encontrar la solución o hasta que se tome la decisión de terminar la búsqueda por esa dirección, se puede tomar por que: se llego a un callejón sin salida, se produce un estado ya alcanzado o la ruta se alarga más de lo especificado.

1. Se crea una pila (abierta) y se le asigna el estado inicial.
2. Mientras la pila no esté vacía.
   a. Extraer el primer nodo de la pila llamarlo m.
   b. Si la profundidad de m es igual al límite de profundidad regresar a A, en caso contrario continuar.
   c. Expandir m. Se generan las entradas sucesoras del estado actual, para cada operador aplicable hacer:
      i. Aplicar el operador al nodo obteniendo un nuevo estado.
      ii. Si el nuevo estado es el objetivo termina el proceso.
      iii. Incluir el nuevo estado principio de la pila volver a 2ª

Ventajas: Si existe una solución garantiza alcanzarla, no recorre todo el árbol para alcanzar la solución, pero es por azar.

9.4 RAMIFICACIÓN Y ACOTACIÓN

Comienza generando rutas completas, manteniéndose la ruta más corta encontrada hasta ese momento. Deja de explorar tan pronto la distancia total sea mayor a la marcada como la ruta más corta.

9.5 HEURÍSTICA

Se refiere a un conocimiento que intuitivamente poseemos, un conocimiento que podría parecer experimental, pero que más bien es debido a la inspiración de quien lo propone. La diferencia entre lo heurístico y lo científico es que este último es un conocimiento debido a la observación, una regla empírica se cumple aunque no sepamos sus causas. En la regla heurística se espera que se cumpla, en casos particulares, y no en todos los casos en que podría aplicarse; no puede ser demostrada teóricamente, ni comprobada experimentalmente para todos los casos no se generaliza; puede resolver perfectamente un problema en particular, pero puede arrojar resultados erróneos con un problema similar.

Ciencia que estudia los procesos de decisión respecto a un campo de conocimiento concreto, como son las estrategias cognitivas. Su contrapartida formal en computación es el algoritmo.

La palabra heurística proviene de la palabra griega heuriskein que significa descubrir, encontrar. Por heurística entendemos una estrategia, método, criterio o truco usado para hacer más sencilla la solución de problemas difíciles. El conocimiento heurístico es un tipo especial de conocimiento usado por los humanos para resolver problemas complejos. En este caso el adjetivo heurístico significa medio para descubrir

METODOS INFORMADOS

9.6 ALGORITMOS INFORMADOS O DE BÚSQUEDA HEURÍSTICA

La heurística es una técnica que aumenta la eficiencia de un proceso proporcionando una guía para este y posiblemente sacrificando demandas de complejidad. La búsqueda heurística resuelve problemas complicados con eficacia, garantizando una estructura de control que encuentra una buena solución.
Porque usar heurísticas:

- No enredarnos en una explosión combinatoria.
- Buscar una solución adecuada.
- Los peores casos, raramente ocurren.
- Profundizar nuestra comprensión del dominio del problema.

9.10. PRIMERO EL MEJOR

Combina las ventajas de BPA y BPP sigue solo un camino a la vez y cambia cuando alguna ruta parezca más prometedora.
Estrategia:

1. Se selecciona el nodo más prometedor según la función heurística apropiada
2. Se expande el nodo elegido de acuerdo a las reglas de expansión adecuadas
3. Si alguno de los nodos es el optimo se termina, sino:
    a. Se añaden nuevos nodos a la lista
    b. Se vuelve al paso 1

9.11. BÚSQUEDA A (A-STRAR)*

Se modifica el anterior empleando una función heurística para estimar el costo desde un nodo actual al nodo objetivo. Realiza una estimación de cada uno de los nodos que se van agregando, esto permite examinar los caminos más prometedores.
Utiliza la siguiente función: f´=g+h´. Donde f´=estimación del costo desde el nodo inicial al nodo objetivo, g =nivel del árbol en que se encuentra el nodo, h´=estimación desde el nodo actual al nodo objetivo.
Se emplean dos listas. Abierta: los nodos que se les ha aplicado la función heurística. Cerrada: nodos que ya han sido expandidos.

Los Pasos son los siguientes:

1. Se toma el nodo más prometedor y que no haya expandido.
2. Se generan los sucesores del nodo elegido, se les aplica la función heurística y se los añaden a la lista de nodos abiertos, siempre verificando previamente que dicho nodo no haya sido generado con anterioridad.

Algoritmo:

- Comenzar con ABIERTOS conteniendo solo el estado inicial.
- Hasta llegar al objetivo o no queden nodos en ABIERTOS hacer:
    - Tomar el mejor nodo de ABIERTOS
    - Generar sus sucesores
    - Para cada sucesor hacer:
        - Si no se ha generado con anterioridad, añadirlo a ABIERTOS y almacenar a su padre
        - Si ya se ha generado antes
        - Si el nuevo camino es mejor que el anterior cambiar al padre
- Actualizar el costo empleado para alcanzar el nodo y sus sucesores.

Capitulo 10

Planificación

10.1. INTRODUCCION

Se refiere al proceso de computar varios pasos de un procedimiento de resolución de un problema antes de ejecutar alguno de ellos.

La planificación se considera como la búsqueda en un espacio de estados con el agregado de que lo que interesa es el camino recorrido entre los estados alcanzados hasta llegar al estado objetivo.

Cuando nos enfrentamos con el problema de planificar y ejecutar una secuencia de pasos en un mundo que no es completamente predecible nos enfrentamos al problema de planificación en IA.

10.2. CLASIFICACIÓN

Planificación en IA:

- Numérica
- Lógica: Por pila de objetivos, No lineal, Jerárquica
- Reactiva

Planificación en robótica

- Reactiva
- De trayectoria de manipuladores

Planificación industrial: métodos basados en simulación discreta

10.3. EL PROBLEMA DE LA PLANIFIACIÓN

El problema de clasificación se caracteriza porque se puede aplicar a diversos dominios además de que no se pueden explotar todas las opciones.
Entradas: Descripción del estado del mundo, descripción del objetivo, conjunto de acciones.

Salida: una secuencia de acciones que pueden ser aplicadas al estado, hasta alcanzar la descripción del estado final

10.4. MUNDO DE LOS BLOQUES

Para poder hacer un estudio sistemático de los métodos y poder compararlos resulta útil trabajar en un único dominico tal como el mundo de los bloques. Se basa en una superficie palana en la que se colocan bloques (de forma cubica únicamente). Los bloques se pueden ubicar unos sobre otros y se pueden llevar a cabo acciones que pueden manipular los bloques (operadores asociados a las reglas).

Acciones del manipulador (Operadores):

- (Desapilar A B)
- (Apilar A B)
- (Levanta A)
- (Bajar A)

Predicados para definir los estados posibles:

- (Apilado A B)
- (Desapilado A B)
- (Sobre la mesa A)
- (Despejado A)
- (Agarrado A)

Estados Inicial y Estado Objetivo

10.6. CODIFICACION DE LOS OPERADORES

Para poder pasar de un estado a otro modificando predicados usamos operadores, cada operador puede describirse mediante una lista de nuevos predicados que el operador provoca que sean ciertos y una lista de los viejos predicados que el operador provoca que sean falsos.

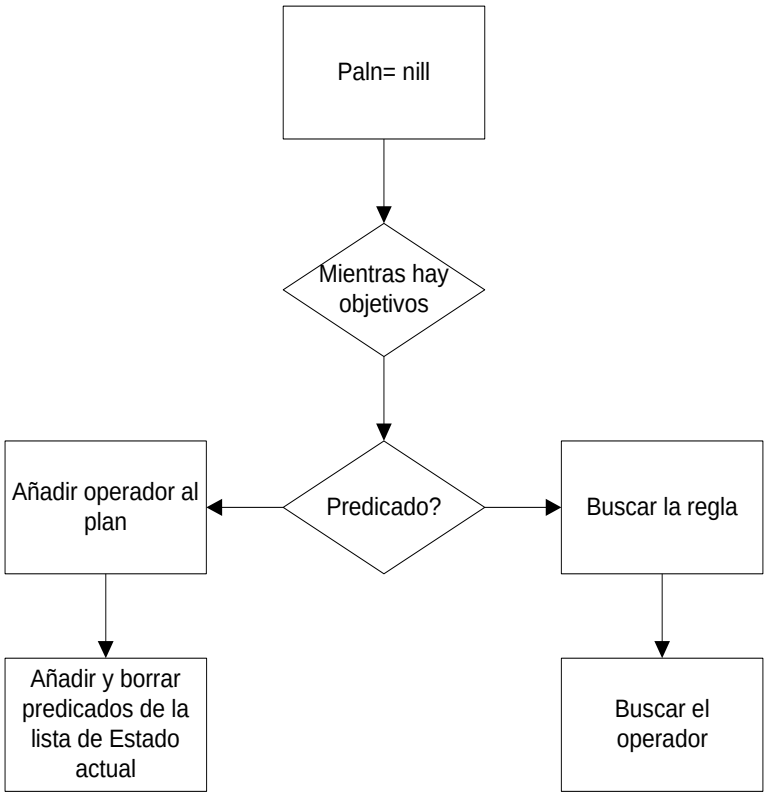
PLANIFICACION MEDIANTE UNA PILA DE OBJETIVOS

Elementos: Bloques, superficie y brazo operador.

Estados de los elementos: estado del bloque y del brazo.

Operadores: acciones a realizar. Se componen de tres listas: precondición (lo que se debe cumplir para aplicar el operador), Borrado (lo que se debe eliminar de la pila de objetivos), Adición (lo que se agrega al plan de tareas)

Pilas: en todo momento se cuanta con una pila de objetivos y operadores, una de estado actual y la del plan a seguir.



Procedimiento: se tratan los objetivos como sub problemas a resolver. Se pregunta si el primer objetivo se cumple. Si no se cumple se debe aplicar el operador que lo transforme, por lo que se debe reemplazar al predicado objetivo por el operador y luego agregar a la pila las precondiciones que deben cumplirse para ese operador. Para resolver el objetivo que ha quedado arriba de la pila empleamos el operador reemplazando y agregando las precondiciones. Al ir extrayendo los operadores de esa pila, se incorporan a la lista plan y se actualiza el estado actual.

10.13. ANOMALÍA DE SUSSMAN

Consiste en la inclusión en el resultado, (plan) de pasos que no llevan a la solución, y que por ese motivo deben anularse posteriormente a haberlos incluido.

Surge al deshacer operadores que han sido aplicados previamente mediante STRIPS recursivo.