Lugares WEBeros

- Almacen
  - http://www.cs.cmu.edu/afs/cs.cmu.edu/project/ai-repository/ai/html/air.html
- Art. Intell: A modern Approach:
  - http://www.cs.berkeley.edu/~russell/aima.html

Prolog

- Introducción
- Definición de un programa en Prolog
- Creación de un sistema de producción
  - Ejemplo
    - Baldes
    - N reinas

Prolog. Introducción

- Versión utilizada
  - Visual Prolog 5.2
    - http://www.pdc.dk/vip/vipinfo/freeware_download.htm
- Antecedentes
  - Utilización de la lógica de predicados
  - Dispone de su propio motor de inferencia
    - Algoritmo de unificación
    - Estrategia de backtracking
  - Es un lenguaje declarativo

Prolog. Programa

- Un programa tiene las siguientes secciones
  - Dominios (domain)
    - Es la declaración de los dominios sobre los que se expresan las relaciones
  - Predicados (predicates)
    - Es la declaración de las relaciones (predicados) que van a aparecer en la base de conocimientos
  - Clausulas (clauses)
    - Es la base de conocimientos donde se expresa el conocimientos en forma de reglas y aserciones.
  - Objetivos (goal)
    - Es la fórmula que se pretende hacer cierta

Prolog Programa (II)

```
predicates
    gusta(symbol,symbol)
 clauses
    gusta(elena, tenis).
    gusta(juan, fútbol).
    gusta(tomas, basket).
    gusta(eric, nadar).
    gusta(marco, tenis).
    gusta(bene, Actividad)
          if gusta(tomas, Actividad).

 goal:gusta(elena,fútbol)
```


Prolog. Programacion

Obtener un programa en Prolog que determine el coche más adecuado

```
domains
    coche=symbol
    nombre=symbol
    gusto=symbol
predicates
    gustos(nombre,coche,gusto)
    auto(coche)
    compra(nombre,coche)
```

Prolog. Programacion

```
clauses
    compra(Persona,Coche):-
            auto(Coche),
            gustos(Persona,Coche,bueno).

    gustos(pepe,seat,bueno).
    gustos(pepe,renault,feo).
    gustos(pepe,ferrari,caro).
    gustos(pepe,ford,bueno).
    gustos(jose,seat,bueno).
    gustos(juan,seat,malo).
    gustos(pepe,kawasaki,bueno).

    auto(seat).
    auto(renault).
    auto(ferrari).
    auto(ford).
    auto(austin).
```

Ver que el bactrack no solamente permite encontrar una solución, sino todas.

Las reglas principales del bactracking son:

-Los subobjetivos han de ser satisfechos en orden de arriba a abajo.

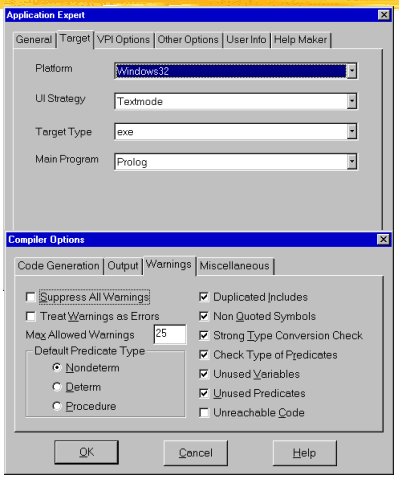
-Para satisfacer a estos se testan los predicados de arriba a abajo tal como aparecen en la imagen.

-Cuando un subobjetivo se empareja con la cabeza de una regla, el cuerpo de esta se constituye como el siguiente subobjetivo a verificar

-Un objetivo ha sido satisfecho cuando se ha conseguido emparejar cada una de sus hojas del árbol de objetivos

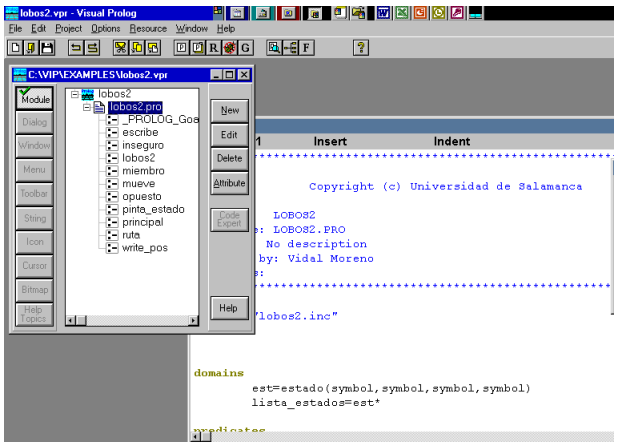
Prolog. Visual Prolog v 5.2

- Etapas
  - ✓ Creación del proyecto (Ctrl-A)
    - ✓ Modo Texto
    - ✓ Aplicación Win32
  - ✓ Opciones
    - ✓ Depurador
      - No Deterministico por defecto



Prolog. Visual Prolog v 5.2

- Ventana de proyecto
  - Contiene un único modulo donde se presenta el código



Prolog. Generación de un sistema de producción

- Ejemplo
  - Un pastor ha atravesar un rio en una balsa de dos plazas como máximo. Va acompañado de un lobo, una oveja y una coliflor. Dadas las prediciones gastronómicas de los participantes, se trata de resolver el problema con un sistema de producción
- Separación entre base de datos, reglas y sistema de control
  - Base de datos: e(P,L,O,B) donde
    - P, L, O, B son las posiciones (Oeste o Este) de cada uno de los miembros
    - Implementacion: Utilizando functores (Funciones de LPPO)

```
domains
        est=estado(symbol,symbol,symbol,symbol)
        lista_estados=est*
```

Prolog. Generación de un sistema de producción

- Implementacion de las reglas
  - 4 movimientos de cruzar orilla (PL,PO,PB,P)
    - Necesita definir un predicado opuesto:
      - Opuesto(e,o).
      - Opuesto(o,e)
- Se debe incluir la verificación de la precondición
  - Definición del predicado seguro
    - Ejemplo: Inseguro si el pastor esta en una orilla distinta a la del lobo y la oveja
- Se debe garantizar que no existen estados repetidos
  - Definición del predicado miembro

Prolog. Generación de un sistema de producción

- Implementación de las reglas

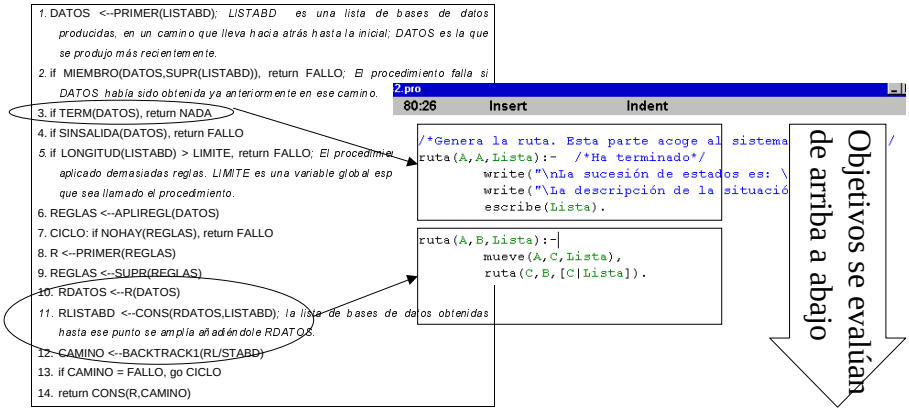
lobos2.pro

55:30 Insert Indent

```
mueve(estado(I,I,O,B),estado(F,F,O,B),Lista):-
        opuesto(I,F),
        not(inseguro(estado(F,F,O,B))),
        not(miembro(estado(F,F,O,B),Lista)),
        write("\nLleva al lobo. Estado: ",F,",",F,",",O,",",B).
mueve(estado(I,L,I,B),estado(F,L,F,B),Lista):-
        opuesto(I,F),
        not(inseguro(estado(F,L,F,B))),
        not(miembro(estado(F,L,F,B),Lista)),
        write("\nLleva a la oveja. Estado: ",F,",",L,",",F,",",B).
mueve(estado(I,L,O,I),estado(F,L,O,F),Lista):-
        opuesto(I,F),
        not(inseguro(estado(F,L,O,F))),
        not(miembro(estado(F,L,O,F),Lista)),
        write("\nLleva a la berza. Estado: ",F,",",L,",",O,",",F).
```

Prolog. Generación de un sistema de producción

- Sistema de control (Proporcionado por el lenguaje)
  - ✓ Backtracking sin información



Prolog. Generación de un sistema de producción

- Definición del objetivo: Incluir en goal principal

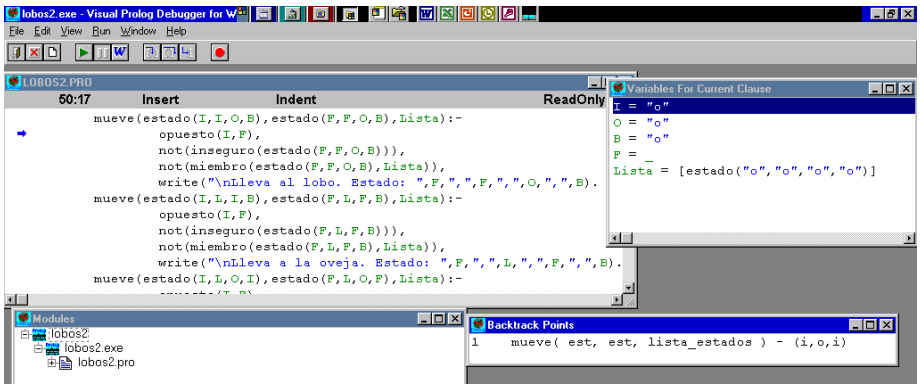
```
/*Genera la salida del problema*/
lobos2():-principal,!.

principal:-
        ruta(estado(o,o,o,o),estado(e,e,e,e),[estado(o,o,o,o)]).
```

Resultado de la ejecución

```
Lleva a la oveja. Estado: e,o,e,o
Lleva a nadie. Estado: o,o,e,o
Lleva al lobo. Estado: e,e,e,o
Lleva a la oveja. Estado: o,e,o,o
Lleva a la berza. Estado: e,e,o,e
Lleva al lobo. Estado: o,o,o,e
Lleva a la oveja. Estado: e,o,e,e
Bactrack desde: Estado: e,o,e,e
Bactrack desde: Estado: o,o,o,e
Lleva a nadie. Estado: o,e,o,e
Lleva a la oveja. Estado: e,e,e,e
La sucesión de estados es:
La descripción de la situación:          estado("o","o","o","o")
  P L O B ||          ||          estado("e","o","e","o")
    L B ||          || P O        estado("o","o","e","o")
  P L B ||          || O          estado("e","e","e","o")
    B ||            || P L O      estado("o","e","o","o")
  P O B ||          || L          estado("e","e","o","e")
    O ||            || P L B      estado("o","e","o","e")
  P O ||            || L B        estado("e","e","e","e")
    ||              || P L O Byes
```

Prolog. Depuración

- Herramienta separada



Prolog

- Ejercicios
  - ✓ Realizar los programas en Prolog que resuelven los siguientes problemas
    - ✓ "Encontrar la secuencias de movimientos de un caballo que une dos posiciones cualquiera"
    - ✓ "Colocar N damas en un tablero NxN"
    - ✓ "Dados dos baldes de 8 y 6 litros (inicialmente vacíos) encontrar la secuencia de operaciones que hay que realizar para que el de 8 litros tenga la mitad de su contenido"
  - ✓ Realizar una implementación de sistema de producción que incorpore la limitación en profundidad de la búsqueda