PRÁCTICAS DE MATEMÁTICAS 1

2018-2019

TRANSPARENCIAS GRUPOS

DE GII: G9, G5

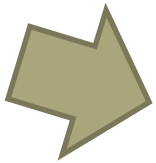
DE GIA: G402

- Entorno de trabajo.
- Programas en Prolog.
- Ejemplos.

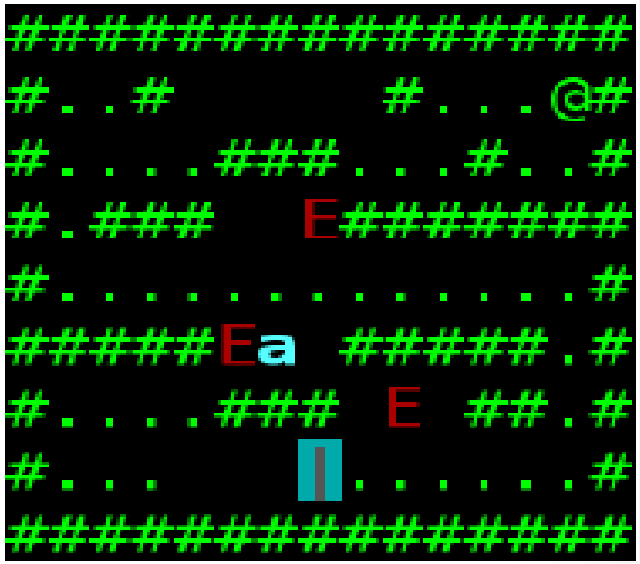
- Prof: Mª Jesús Castel de Haro.
- D. Ciencia Computación Inteligencia Artificial
- Politec-II, 1º planta columnas verdes
- Contacto: Tutorías UACloud

- Vídeos d Prácticas M1. curso 2017-18.
- Prof. Francisco Gallego. https:// bit.ly/prácticasM1

Prácticas: ¿conoces el juego de Pacman?...por ahí van los tiros



Pl-Man @



DEMOPLMAN

Plman ???

>> Comecocos @ que se mueve por un mapa tratando de comerse todos los cocos y evitando que los enemigos lo fulminen.

>> Dado un mapa se trata de que programes las acciones que debe realizar Plman para lograr su propósito

Programación >> Lenguaje de Programación lógica: Prolog

Entorno

ENTREGA Y CORRECCIÓN AUTOMÁTICA de prácticas a través de un sistema on-line
http:// logica.i3a.ua.es

El sistema proporciona:

>> programa fuente del juego Plman

>> mapas para resolver:

mapas-ejemplos: que no evalúan

mapas-evaluación: sí evalúan

>> tutoriales

MAPAS PARA EVALUACIÓN

Fase 0 (Tutorial)	5 Mapas
Fase 1	3 Mapas
Fase 2	3 Mapas
Fase 3	2 Mapas
Fase 4	1 Mapa
TOTAL	**14 Mapas diferentes**

Mapas limitados

FASE	DIFICULTAD				
	D1	D2	D3	D4	D5
0	0,100				
1	0,350	0,450	0,500	0,550	0,650
2	0,525	0,675	0,750	0,825	0,975
3	1,225	1,575	1,750	1,925	2,228
4		2,025	2,250	2,475	

EVALUACIÓN ENERO

Revisamos evaluación

P [40p] = M [36p] + C [4p]

M : Resolver mapas sistema .

C : Control para validar M.

Si C < 2p → Prácticas Suspensas.

EVALUACIÓN JULIO

P [40p] = M [20p] + C' [20p]

M : Nota Plman obtenida durante curso. No recuperable.

C' : Examen escrito + resolución mapa (al menos 50%)

ecc → Suspenso.

EN TU ORDENADOR resuelves el mapa

Mapas ejemplos
Para aprender a programar Plman,
No evalúan

PROGRAMAS la SOLUCIÓN del MAPA **EJECUTAS / COMPRUEBAS**

Mapas de evaluación.
Para obtener nota (M)

1º PROGRAMAS la SOLUCIÓN del MAPA **EJECUTAS / COMPRUEBAS**

2º ENTREGAS SOLUCIÓN al SISTEMA

3º OBTIENES "IPSO FACTO" **NOTA**
+
ESTADÍSTICAS DE RESOLUCIÓN

NECESITARÁS…

- Máquina Virtual → EPS

- Sistema Operativo: **LINUX** (Ubuntu).

- Lenguaje de programación : **Prolog**

  → Intérprete **SWI Prolog**

- Sistema online (web prácticas) http://logica.i3a.ua.es

TERMINAL D LINUX

usuario @ máquina: directorio_actual $ Línea de comandos

- **usuario**: nombre del usuario conectado a la terminal
- **@**: "en"
- **máquina**: nombre de la máquina a la cual estamos conectados
- **directorio actual**: donde nos encontramos
- **$**: indicador para comenzar a escribir órdenes o comandos

Comandos básicos

cd	change directory (cambia directorio)
cd musica	entra en directorio *musica*
cd ..	sale al directorio anterior
cd /	va al directorio raíz del disco duro
cd ~	va al directorio del usuario
ls	muestra contenido del directorio
ls	lista contenido del directorio actual
ls musica	lista contenido del subdirectorio *musica*

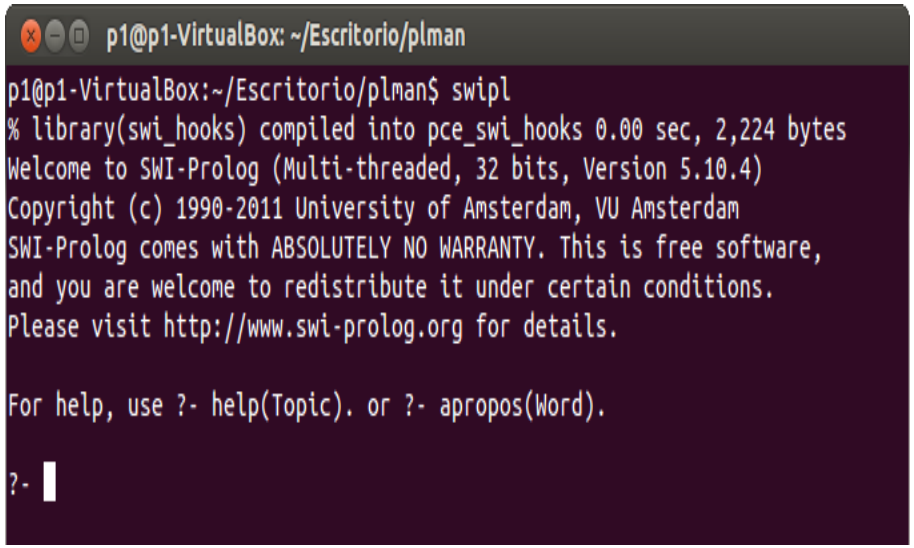
SWI-PROLOG

Interfaz : consola de comando textual de dominio público para ordenadores PC desarrollado en U. Amsterdam 70's, Alain Colmerauer y P. Roussell,

http://www.swi-prolog.org

Descargar programas demosprolog

http:// bit.ly/demosprolog



PASOS PARA UNA SESIÓN CON EL INTÉRPRETE SWI-PROLOG

>> **Abrir** terminal Linux

>> **Teclear** comando para abrir intérprete Swi-Prolog

$ swipl

>> **Editar** fichero, editores: emacs, gedit…

? emacs('prolog1.pl').

>> **Escribir programa** Prolog > crear base de conocimiento.

>> **Compilar /cargar** en memoria el fichero prolog1.pl editado

menú edición: **Compile /Compile buffer**

si no se ha editado:

? consult('prolog1.pl').

>> **Ejecutar** programa:

? pregunta.

PROGRAMA PROLOG

Es una Base de conocimiento con dos tipos de sentencias

HECHOS

REGLAS:

RESULTADO :- CONDICIONES.

Se obtiene un resultado si (:-) se cumplen las condiciones

Que no falten los COMENTARIOS

Texto dentro de un archivo de código PROLOG (.pl) que es ignorado por completo por el intérprete/compilador y que sirve de ayuda a los programadores.

```
% Comentario
% Otro comentario
... código ....

% Más comentarios
... código ....
/* Comentario
de 2 líneas */
```

HECHOS

Indica unos valores concretos para los cuales ese predicado se cumple.

```
% veo(DIRECCION, OBJETO)

% Veo arriba un coco
veo(arriba, coco).

% Veo abajo una llave
veo(abajo, llave).
```

REGLAS

cabeza :- cuerpo.

cabeza: objetivo (cláusula de un predicado) que puede ser cierto o falso, es decir, tener éxito o fracasar.

cuerpo: condiciones de las que depende que cabeza tenga éxito o fracase.

si *cuerpo* entonces *cabeza*

cuerpo* → *cabeza

SWI Prolog

HECHOS Y REGLAS se fundamentan en describir predicados

PREDICADOS

- Empiezan siempre por **letra minúscula**.
- No hay espacio entre el nombre y el paréntesis.
- Argumentos (términos) separados por comas.
- Pueden tener 0 argumentos y no habría paréntesis (proposiciones)

predicado(arg1, arg2, arg3,…., argN)

ejemplos de predicados

```
% llave(X): X es una llave
llave(a)

% ver(J, O): El jugador J ve el objeto O
ver('Pl-Man', a)
ver(plman, OBJ)
```

TÉRMINOS, SUJETOS

constante (átomo)

Palabra que identifica de forma única a un objeto del dominio.

- Siempre empieza por letra **minúscula.**
- Puede contener letras, números o subrayado.
- **No** pueden contener espacios.
- Todo lo que vaya entre comillas simples ' '

ejemplos de constantes y números:

```
% constantes
llave
‘llave azul’
llave_azul
llave512
‘Pl-Man'
```

```
% números
123
12.5
-10
```

TÉRMINOS, SUJETOS

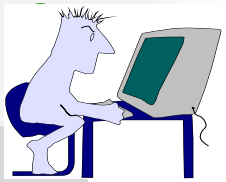
variable

Palabra que identifica de forma genérica a un objeto del dominio:

- Empiezan por **letra mayúscula** o **subrayado.**
- Pueden contener letras, números o subrayado.
- No pueden contener espacios.

ejemplos de variables

X
_objeto
VALOR
Fantasma_1
ENEMIGO10
_ (! variable anónima !)

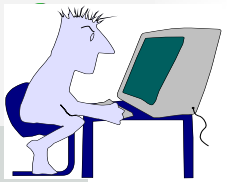


Práctica 1-Prolog

Escribir un programa en PROLOG que conteste a la pregunta planteada en el razonamiento-1:

P1: Todos los alum tienen sentido del humor
P2: Juan es un alum

Responder Q: **¿ Juan tiene sentido del humor ?**



Práctica 1-Prolog (cont)

P1: Todos los alum tienen sentido del humor

P2: Juan es un alum

Responder Q: **¿Juan tiene sentido del humor?**

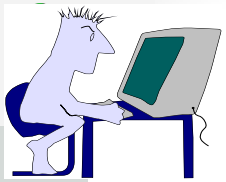
Predicados: ⇨

sentidoH(X): X es un sujeto que tiene sentido humor

alum(X): X es un sujeto que tiene la propiedad de ser alum

Formalización

Lógica predicados	**Prolog**
$\forall x$ [alum(x) $\rightarrow$ sentidoH(x)]	**sentidoH(X) :- alum(X).**



Práctica 1-Prolog (cont)

P1: Todos los alum tienen sentido del humor

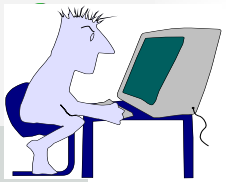
P2: Juan es un alum

Responder Q : **¿Juan tiene sentido del humor?**

P2: es proposición atómica,
es un hecho en Prolog

Formalización

Lógica predicados	**Prolog**
alum(juan)	**alum(juan).**



Práctica 1-Prolog (cont)

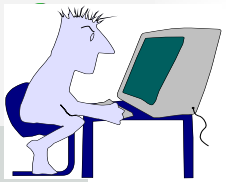
P1: Todos los alum tienen sentido del humor
P2: Juan es un alum

Responder Q : **¿Juan tiene sentido del humor?**

Q: es proposición atómica,
es un hecho en Prolog

Formalización

Lógica predicados	**Prolog**
sentidoH(juan)	**sentidoH(juan).**



Práctica 1-Prolog (cont)

- 1º ESCRIBIR programa Prolog:

  ? emacs('raz1.pl').

- **2º COMPILAR:**

  Compile /Compile buffer

- **3º EJECUTAR → preguntar desde la consola swipl**

  ? sentidoH(juan).

Práctica 2-Prolog

Escribir en fichero **'raz2.pl'.**
Formalizar en L. predicados y en Prolog
Ejecutar las preguntas

1: Carlos es alum.

2: Para que un sujeto sea alum es necesario que tenga buen tipo y

3: ésta es una condición suficiente para que esté macizo.

4: Si un sujeto no es alum, es atractivo.

5: Si un sujeto es atractivo, está macizo.

¿Carlos está macizo? ¿tú estás macizo?

SOLUCIÓN Práctica 2-Prolog

Lógica predicados

Alu(carlos).

Alu(x) $\rightarrow$ Btipo(x).

Btipo(x) $\rightarrow$ Ma(x).

¬Alu(x) $\rightarrow$ At(x).

At(x) $\rightarrow$ Ma(x).

Conclusiones:

Ma(carlos).

Ma(luis)

Prolog

```
alum(carlos).
buentipo(X) :- alum(X).
macizo(X)   :- buentipo(X).
atractivo(X) :- not(alum(X)).
macizo(X)   :- atractivo(X).
```

Objetivos/ preguntas

```
?- macizo(carlos).
?- macizo(luis).
?- macizo(X).
```


TUTORIALES Y APUNTES

Apuntes de Prolog y material (castellano)
Campus virtual /Materiales
http://www.dccia.ua.es/logica/prolog/docs/prolog.pdf
http://www.dccia.ua.es/logica/prolog/material.htm

Adventure in prolog (inglés)
http://www.amzi.com/AdventureInProlog/advfrtop.htm

LIBROS_: Buscar en Biblioteca y Archivo, web UA http://www.ua.es

The Art of Prolog
http://gaudi.ua.es/uhtbin/cgisirsi/AkB95t6saS/0/253110069/9

Programación en Prolog
http://gaudi.ua.es/uhtbin/cgisirsi/LO4Ho5QWQ5/0/253110069/9