

Programación para la Inteligencia Artificial

Presentación del curso

Dr. Alejandro Guerra-Hernández

Maestría en Inteligencia Artificial

Universidad Veracruzana

CIIA – Centro de Investigación en Inteligencia Artificial

Sebastián Camacho No 5, Xalapa, Ver., México 91000

aguerra@uv.mx

<http://www.uv.mx/personal/aguerra/>

13 de Agosto del 2019

Bienvenidos al curso Programación para la Inteligencia Artificial. La Inteligencia Artificial (IA) ha producido lenguajes de programación que son a la vez, herramientas y sujeto de estudio de esta disciplina. Este curso introduce los paradigmas de Programación Lógica, Funcional (y Orientado a Agentes) desde esta perspectiva. Se introducen los fundamentos teóricos de ambos paradigmas y se explora su uso en problemas característicos de la IA, i.e., estrategias de resolución de problemas, búsquedas heurísticas, inducción de árboles de decisión, etc., utilizando los lenguajes de programación Prolog, Lisp. Se utiliza el problema de inducción de árboles de decisión para comparar los lenguajes de programación aquí introducidos.

El curso tiene una duración de 60 horas; se oferta durante un semestre, organizado en dos sesiones presenciales de dos horas de duración.

1. Objetivos

- El estudiante identificará los conceptos fundamentales de la Programación Lógica y la Programación Funcional, así como sus orígenes en la IA.
- El estudiante adquirirá las habilidades para resolver problemas complejos con el lenguaje de Programación Lógica Prolog.
- El estudiante adquirirá las habilidades para resolver problemas complejos con el lenguaje de Programación Funcional Lisp.
- El estudiante podrá generalizar estas habilidades usando otros lenguajes de programación.

```

1  %% allPerms computa todas las permutaciones de los elementos
2  %% de la lista L en la lista L2. Ejemplo de llamada:
3  %% ?- allPerms([1,2,3],Resp).
4  %% Resp = [[1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2],
5  %% [3, 2, 1]].
6
7  allPerms(L,L2) :-
8      findall(Perm,permutation(L,Perm),L2).

```

Cuadro 1: Ejemplo de código de computadora y sus comentarios (Prolog).

2. Evaluación

La nota final del curso será calculada de la siguiente forma:

- Las tareas/Ejercicios cubren un 50 % de la nota final.
- El examen de conocimientos cubre el 20 % de la nota final.
- El proyecto integrador cubre el 30 % de la nota final.

Para obtener una **nota aprobatoria** en el curso, el alumno deberá haber obtenido notas aprobatorias en cada uno de los elementos de evaluación (aprobar todas sus tareas, examen y proyecto).

2.1. Tareas

Las tareas pueden requerir investigación bibliográfica, ejercicios teóricos, experimentación en la computadora y/o exposiciones frente a grupo. La entrega y evaluación de éstas se realizará conforme a los siguientes lineamientos:

Entrega. Las tareas se entregan al inicio de clase del día designado para ello. El mérito de las mismas decrece 25 % por cada 24 horas de retraso. El calendario del curso marca las fechas tentativas para cada tarea, su entrega suele ser dos semanas más tarde.

Formato. Las tareas se procesan con \LaTeX , siguiendo la plantilla de este mismo documento. En todas las partes que involucran código de computadora o algoritmos, éste deberá ser documentado apropiadamente (Ver ejemplo del Cuadro 1).

Evaluación. Para que un ejercicio o pregunta de la tarea reciba puntaje alguno, más del 50 % deberá estar resuelta de manera correcta. Es más redituable invertir el tiempo en contestar una pregunta de manera correcta y completa, que responder a dos de manera incompleta.

Cualquier forma de **plagio** causa la expulsión definitiva del curso. Esto incluye: Reportar trabajo de otros como propio y no citar pertinentemente las referencias usadas.

2.2. Examen

Se trata de una evaluación de los conocimientos teóricos del curso. Su fecha de aplicación aparece en el calendario de este documento.

2.3. Proyecto integrador

Se trata de un proyecto práctico que requiere la aplicación las diversas técnicas introducidas en el curso, posiblemente complementado con el contenido de otros cursos de este semestre. Se comienza a definir a la mitad del mismo para tratar de integrar los intereses de investigación de los estudiantes. Las fechas de revisiones parciales y final del proyecto integrador aparecen en el calendario de este documento.

3. Material del curso

Como es usual en nuestros cursos, el material de clases (notas, presentaciones, código) está disponible en mi página personal:

<http://www.uv.mx/personal/aguerra/pia/>

La bibliografía básica está a disposición de ustedes en la biblioteca del departamento ó en mi oficina. Algunos de los artículos están disponibles en versión electrónica gracias a las suscripciones de nuestra biblioteca central y del Conacyt.

Utilizaremos el Mundo de Tarski [1], un popular programa para explorar el uso de la lógica de primer orden. Más información sobre este programa puede encontrarse en:

<https://www.gradegrinder.net/Products/tw-index.html>

Haremos uso de los siguientes lenguajes de programación, todos ellos tienen versiones para diversos sistemas operativos:

- <http://www.swi-prolog.org>
- <http://www.sbcl.org>
- <http://www.lispworks.com>
- <http://jason.sourceforge.net/wp/>

4. Calendario

Este año las sesiones se llevarán a cabo los martes y jueves de 10:00 a 12:00 hrs., en el salón A-01. Las sesiones se organizarán como sigue:

Fecha	Tema	Tarea
13/08/2018	Presentación del curso	
15/08/2018	Introducción a los Paradigmas Lógico y Funcional	
20/08/2018	Programación Lógica, Lógica de Primer Orden	
22/08/2018	Programación Lógica, Programas Definitivos	
27/08/2018	Programación Orientada a Agentes I	
29/08/2018		
03/09/2018	Programación Lógica, Resolución	T1
05/09/2018	Prolog, Conceptos Básicos I	
10/09/2018		
12/09/2018	Prolog, Búsquedas de Soluciones I	
17/09/2018		
19/09/2018	Prolog, Inducción de Árboles de Decisión I	
24/09/2018		T2
26/09/2018	Programación Funcional, Cálculo- λ	
01/10/2018		
03/10/2018	Lisp, Conceptos Básicos I	
08/10/2018		
10/10/2018	Lisp, Listas	
15/10/2018	Lisp, Macros	T3
17/10/2018		
22/10/2018	Lisp, GPS	
24/10/2018		
29/10/2018	Lisp, Bioinformática	
31/10/2018		
05/11/2018	Lisp, Inducción de Árboles de Decisión	
07/11/2018		
12/11/2018		T4
14/11/2018	Definición Proyectos Integradores	
19/11/2018	Otros Lenguajes I	
21/11/2019		
26/11/2019	Otros Lenguajes II	
28/11/2019		
09/01/2020	Evaluación proyecto final	

Referencias

1. BARKER-PLUMMER, D., BARWISE, J., ETCEMENDY, J., AND LIU, A. *Tarski's World, Revised and Expanded*, CSLI Publications, Stanford, CA., USA, 2008.
2. BORDINI, R. H., HÜBNER, J. F., AND WOOLDRIDGE, M. *Programming Multi-Agent Systems in Agent-Speak using Jason*. John Wiley & Sons Ltd, 2007.
3. I. BRATKO. *Prolog programming for Artificial Intelligence*. Pearson, fourth edition, 2012.

4. CLOCKSIN, W. F., AND MELISH, C. S. *Programming in Prolog, using the ISO standard*. Springer-Verlag, Berlin-Heidelberg, Germany, 2003.
5. P. GRAHAM. *ANSI Common Lisp*. Prentice Hall Series in Artificial Intelligence. Prentice Hall International, 1996.
6. W. KLUGE. *Abstract Computing Machines: A Lambda Calculus Perspective*. Springer-Verlag, Berlin Heidelberg New York, 2005.
7. U. NILSSON AND J. MALUSZYNSKI. *Logic, Programming and Prolog*. John Wiley & Sons Ltd, 2nd edition, 2000.
8. P. NORVIG. *Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp*. Morgan Kauffman Publishers, 1992.
9. P. SEIBEL. *Practical Common Lisp*. Apress, USA, 2005.

Xalapa, Ver., México
Agosto 2019

Alejandro
Guerra-Hernández