Programación Lógica y Funcional

Notas finales

Dr. Alejandro Guerra-Hernández

Instituto de Investigaciones en Inteligencia Artificial
Universidad Veracruzana
Campus Sur, Calle Paseo Lote II, Sección Segunda No 112,
Nuevo Xalapa, Xalapa, Ver., México 91097

mailto:aguerra@uv.mx
https://www.uv.mx/personal/aguerra/plf

Maestría en Inteligencia Artificial 2024

Paradigma de Programación

- Modelo computacional + Lenguajes de Programación
- Ejemplos:
  - Resolución SLD + Prolog, DataLog, etc.
  - Cálculo Lambda + Lisp, Scheme, Haskell, Ocaml, etc.
  - Máquina de Turing + ???
  - ??? + Python, Perl, Tcl
- Integración pertinente si tu lenguaje es herramienta de desarrollo y sujeto de estudio al mismo tiempo.
- Debería ser nuestro caso: ¿Es la inteligencia una forma de computación?

Universidad Veracruzana

Desarrollo vigente

- Levesque [5] el pensamiento como computación, basado en Prolog.
- Riguzzi [10] programación lógica probabilista.
- Lifschitz [6] una introducción a ASP.
- Maier et al. [8] programación lógica declarativa.
- Weitz [14] nos ofrece un recetario de Lisp.
- Knott [4] interprete de Lisp.
- Vsevolod [13] para la implementación eficiente de algoritmos en Lisp.
- Sitnikovski [12] una introducción a blockchain con Lisp.
- Herda [3] el sistema de condiciones de Lisp y el manejo de excepciones.

Universidad Veracruzana

¿Debemos usar solo Lisp o Prolog?

- Saumont [11] programación funcional en Java.
- Mailund [9] programación funcional en R.
- Lott [7] programación funcional en Python.
- Mary Rose Cook tiene un divertido tutorial de programación funcional en Python 2.
- Problog2 es un lenguaje de programación lógico probabilista implementado en Python.

Universidad Veracruzana

Sobre Python I

- Desde 1993 tenemos funciones anónimas y de orden superior como `map()`, `filter()` y `reduce()`.
- El dictador benevolente quería eliminar todo esto en Python 3.
- Al final de quedaron: `map()` está implementado en C y regresa un objeto iterable en lugar de la lista que regresaba en Python 2.

```
In [1]: numeros = [1,2,3,4,5]
In [2]: list(map(lambda x:x*x, numeros))
Out[2]: [1, 4, 9, 16, 25]
```

- Pero a Python le gustan más la compresión de listas:

```
In [3]: def sqr(x):
   ...:     return x * x
In [4]: [sqr(x) for x in numeros]
Out[4]: [1, 4, 9, 16, 25]
```

Universidad Veracruzana

Sobre Python II

- `reduce()` vive en la librería `functools`.

```
In [5]: import functools
In [6]: functools.reduce(lambda x, y:x+y, numeros)
Out[6]: 15
```

- Pero:

```
In [7]: sum(numeros)
Out[7]: 15
```

Universidad Veracruzana

Sobre Python III

- No hay librerías o primitivas de programación lógica oficiales en Python.
- Gonczarowski y Nisan [2] presentan una introducción a la lógica matemática con implementaciones en Python.

  https://www.logicthrupython.org
- Hay una implementación de miniKanren de Byrd [1] en Python:

  https://pypi.org/project/miniKanren/

  PS. no perderse su conferencia en youtube:

  https://www.youtube.com/watch?v=OyfBQmvr2Hc

Universidad Veracruzana

Sobre Python IV

- Ejemplo:

```
In  [1]: from kanren import Relation, facts, var, run
In  [2]: x = var()
In  [3]: progenitor = Relation()
In  [4]: facts(progenitor, ("Bob", "Ann"), ("Bob", "Pat"))
In [10]: run(1, x, progenitor(x, "Ann"))
Out[10]: ('Bob',)
In [11]: run(2, x, progenitor("Bob",x))
Out[11]: ('Pat', 'Ann')
```

Universidad Veracruzana

Referencias I

[1] WE Byrd. "Relational Programming in MiniKanren: Techniques, Applications, and Implementations". Doctor of Philosophy. Bloomington, IN, USA: University of Indiana, 2009.

[2] YA Gonczarowski y N Nisan. *Mathematical Logic through Python*. Cambridge, United Kingdom: Cambridge University Press, 2022.

[3] Mp Herda. *The Common Lisp Condition System: Beyond Exeption Handlng with Control Flow Mechanisms*. New York, NY, USA: Apress, 2020.

[4] GD Knott. *Interpreting LISP: Programming and Data Structures*. Second. New York, NY, USA: Apress, 2017.

[5] HJ Levesque. *Thinking as Computation*. Cambridge, MA, USA: The MIT Press, 2012.

[6] V Lifschitz. *Answer Set Programming*. Cham, Switzerland: Springer Nature Switzerland AG, 2019.

[7] SF Lott. *Functional Python Programming*. Second. New York, NY, USA: Packt Publishing Ltd, 2018.

[8] D Maier et al. "Declarative Logic Programming: Theory, Systems, and Applications". En: ed. por M Kifer e YA Liu. ACM Books. New York, NY, USA: ACM y Morgan & Claypool Publishers, 2018. Cap. Datalog: Concepts, History, and Outlook, págs. 3-100.

Universidad Veracruzana

Referencias II

[9] T Mailund. *Functional Programming in R 4: Advanced Statistics Programming for Data Science, Analysis, and Finance*. Second. New York, NY, USA: Apress Media LLC, 2023.

[10] F Riguzzi. *Foundations of Probabilistic Logic Programming: Languages, Semantics, Inference and Learning*. River Publishers Series in Software Engineering. Gistrup, Denmark: River Publishers, 2018.

[11] PY Saumont. *Functional Programming in Java*. New York, NY, USA: Manning Publications Co., 2017.

[12] B Sitnikovski. *Introducing Blockchain with Lisp: Implement and Extend Blockchains with the Racket Language*. New York, NY, USA: Apress, 2021.

[13] D Vsevolod. *Programming Algorithms in Lisp: Writing Efficient Programs with Examples in ANSI Common Lisp*. New York, NY, USA: Apress, 2021.

[14] E Weitz. *Common Lisp Recipes: A Problem-Solution Approach*. New York, NY, USA: Apress, 2016.

Universidad Veracruzana