Universidad Rey Juan Carlos
Escuela Técnica Superior Ingeniería Informática

Teach the importance of logic (programming) in Computer Science and why it is important.

Joaquín Arias

Grupo de Inteligencia Artificial de la URJC
Center for Intelligent Information Technologies (CETINIA)
Móstoles, Madrid

17 November 2023 (SIIE'23)

XXV SIIE 2023
EDUCAÇÃO SETÚBAL POLYTECHNIC UNIVERSITY

Objetive

Support the importance of teaching logic (and logic programming) in computer science degrees:

- Logic allows us to model (almost) everything and to reason about it.
- Some historical milestones:

$0, 1, \ldots$ Human reasoning: 4th century BC, 13th and 17th century, etc.
[...]
n Quantum mechanics: 21st century.
$n+1$ (A lot of) other things.

Objetive

Support the importance of teaching logic (and logic programming) in computer science degrees:

- Logic allows us to model (almost) everything and to reason about it.
- Some historical milestones:

$0, 1, \ldots$ Human reasoning: 4th century BC, 13th and 17th century, etc.

$\ldots, m, \ldots$ Mathematics: mainly from 1910 to 1930.

$m + h$ In 1920, Hilbert proposed to axiomatize mathematics...

n Quantum mechanics: 21st century.

$n + 1$ (A lot of) other things.

Objetive

Support the importance of teaching logic (and logic programming) in computer science degrees:

- Logic allows us to model (almost) everything and to reason about it.
- Some historical milestones:

$0, 1, \ldots$ Human reasoning: 4th century BC, 13th and 17th century, etc.
$\ldots, m, \ldots$ Mathematics: mainly from 1910 to 1930.
$m + h$ In 1920, Hilbert proposed to axiomatize mathematics...
$m + g$ ... but in 1931, Gödel shows that it is not possible.
n Quantum mechanics: 21st century.
$n + 1$ (A lot of) other things.

Godel pointed out (and proved) the limitations of Logic

Incompleteness Theorem (Gödel, 1931)

No formal mathematical theory capable of describing natural numbers and arithmetic with sufficient expressiveness is both consistent and complete.

Godel pointed out (and proved) the limitations of Logic

Incompleteness Theorem (Gödel, 1931)

No formal mathematical theory capable of describing natural numbers and arithmetic with sufficient expressiveness is both consistent and complete.

- If the axioms of a theory do not contradict each other (consistency), them there are statements that can neither be proved nor disproved from them (completeness).

Godel pointed out (and proved) the limitations of Logic

Incompleteness Theorem (Gödel, 1931)

No formal mathematical theory capable of describing natural numbers and arithmetic with sufficient expressiveness is both consistent and complete.

- If the axioms of a theory do not contradict each other (consistency), them there are statements that can neither be proved nor disproved from them (completeness).

Second incompleteness Theorem (Gödel, 1931)

Particular case of the first Theorem: One of the undeciable statements of a theory is the one that affirms the consistency of the theory.

… and also the computation inherits these these limitations

Answer to *Entscheidungsproblem* (Church, 1936)

First-order logic does not have a decision procedure (algorithm) that allows us to prove that any formula is a theorem of this logic. This logic is therefore considered undecidable.

… and also the computation inherits these these limitations

Answer to *Entscheidungsproblem* (Church, 1936)

First-order logic does not have a decision procedure (algorithm) that allows us to prove that any formula is a theorem of this logic. This logic is therefore considered undecidable.

Halting problem (Turing, 1936)

Given a Turing machine M and a word w, determine whether M terminates in a finite number of steps when executed using w as input, is undecidable.

… and also the computation inherits these these limitations

Answer to *Entscheidungsproblem* (Church, 1936)

First-order logic does not have a decision procedure (algorithm) that allows us to prove that any formula is a theorem of this logic. This logic is therefore considered undecidable.

Halting problem (Turing, 1936)

Given a Turing machine M and a word w, determine whether M terminates in a finite number of steps when executed using w as input, is undecidable.

- Logic and Computation are equivalent: Determining the halting problem is reduced to proving in FOL the formula that expresses the existence of an output from the application of a series of instructions. PLAY[1]

Ver "Las Matemáticas tienen una Terrible Falla" (https://bit.ly/3sFyaxq)

Who I am

Joaquín Arias
Professor at Universidad Rey Juan Carlos (URJC).

- 2020 - today: Researcher at Group of IA at CETINIA, URJC.
- 2013 - 2020: Researcher at IMDEA Sotware Institute.
- Academic background:
  - Ph.D. in Computer Science (2020).
  - M.Sc. in Software and Systems (2015).
  - B.Sc. in Mathematics and informatic (2014).
  - M.Arch. in Architecture (2002).
- PhD Thesis: "Advanced Evaluation Techniques for (Non)-Monotonic Reasoning using Rules with Constraints".

Motivation and some teaching experience at URJC

- Logic course in first grade.
  - Validity of a reasoning; resolution of a problem; deductive methods and its application in automated proof of theorems.

Motivation and some teaching experience at URJC

- Logic course in first grade.
  - Validity of a reasoning; resolution of a problem; deductive methods and its application in automated proof of theorems.
- Second year: Functional + Logic programming...

Motivation and some teaching experience at URJC

- Logic course in first grade.
  - Validity of a reasoning; resolution of a problem; deductive methods and its application in automated proof of theorems.
- Second year: Functional + Logic programming...
- Third and fourth year: The acquired knowledge of logic is not valued.
  - ...and there are deficiencies writing technical reports (including analysis or synthesis) and in preparing and giving presentations.

Motivation and some teaching experience at URJC

- Logic course in first grade.
  - Validity of a reasoning; resolution of a problem; deductive methods and its application in automated proof of theorems.
- Second year: Functional + Logic programming...
- Third and fourth year: The acquired knowledge of logic is not valued.
  - ...and there are deficiencies writing technical reports (including analysis or synthesis) and in preparing and giving presentations.
- Final Degree Project: Last opportunity to incorporate issues related to logic, so important in the professional development of an engineer:
  - Motivation of the importance of the problem to be solved,
  - Modeling of problems; definition of objectives; analysis of existing solutions, justification of the proposal; synthesis of the results; etc.

Motivation and some teaching experience at URJC (cont.)

- Linking Logic With Programming

Motivation and some teaching experience at URJC (cont.)

- Linking Logic With Programming
  - Natural Deduction + DeduccionNatural.pl
  - How teach development techniques used in the industry.

    E.g., in groups choose, study and present one of them in depth.
  - Present historical context of Logic and its limitations:

    E.g., there are activities that cannot be automated.

    ... computer scientists' work is one of these activities.
  - Actual objective of these activities:
    - Awake critical thinking.

      E.g., searching for information (also in research articles).

      ... writing a short report with writing standards (and LaTeX).

      ... and/or preparing and giving a presentation.

Motivation and some teaching experience at URJC (cont.)

- Linking Logic With Programming
  - Natural Deduction + DeduccionNatural.pl
  - How teach development techniques used in the industry.

    E.g., in groups choose, study and present one of them in depth.
  - Present historical context of Logic and its limitations:

    E.g., there are activities that cannot be automated.

    ... computer scientists' work is one of these activities.
  - Actual objective of these activities:
    - Awake critical thinking **(and interest in research)**.

    E.g., searching for information (also in research articles).

    ... writing a short report with writing standards (and LaTeX).

    ... and/or preparing and giving a presentation.

One Example: Natural deduction + DeduccionNatural.pl

- Propositional logic is simple and its semantics is sound and complete.
- First order logic represents knowledge more clearly and concisely.
- We infer new knowledge from premises: [Aristotle IV BC]

  All men are mortal. Socrates is a man. Therefore Socrates is mortal.

First order logic:

$$\frac{\forall x\ (Man(x) \rightarrow Mortal(x)) \quad Man(socrates)}{Mortal(socrates)}$$

Horn clauses:

$Mortal(x) \vee \neg Man(x)$

$Man(socrates)$

$\neg Mortal(socrates)$

Prolog:

```
mortal(X) :- man(X).
man(socrates).

?- mortal(socrates).
```

- A subset of the first order logic is decidable [Horn 1951, Robinson 1965].

One Example: Natural deduction + DeduccionNatural.pl

Demonstrations in natural deduction as programs

- Using DeduccionNatural.pl [Arias et al. 2023].
  - Implemented using Prolog, a logic programming language.
  - Can be executed online.
- Based on the Natural Deduction method by Gentzen (see [Arias 2022; Gallinari 2009; Gentzen 1935] for details):
  - The inference rules are predefined functions.
  - The proofs are programs.
  - The derived rules are subroutines.
- The student can read DeduccionNatural.pl (and manipulate it).

Natural Deduction by Gentzen I

Introducción de $\wedge$ ($I_\wedge$)

$$\frac{A \quad B}{A \wedge B}$$

$T[p, q] \vdash p \wedge q$	
1. p	premisa
2. q	premisa
3. $p \wedge q$	**$I_\wedge$ (1,2)**

Eliminación de $\wedge$ ($E_\wedge$)

$$\frac{A \wedge B}{A} \qquad \frac{A \wedge B}{B}$$

$T[p \wedge q] \vdash p$	
1. $p \wedge q$	premisa
2. p	**$E_\wedge$ (1)**

$T[p \wedge q] \vdash q$	
1. $p \wedge q$	premisa
2. q	**$E_\wedge$ (1)**

Natural Deduction by Gentzen II

Eliminación de ∨ ($E_\vee$)

$$\frac{A \vee B \quad A \rightarrow C \quad B \rightarrow C}{C}$$

T[p ∨ q, p → ¬r, q → ¬r] ⊢ ¬r

1.	p ∨ q	premisa
2.	p → ¬r	premisa
3.	q → ¬r	premisa
4.	¬r	**$E_\vee$ (1,2,3)**

Introducción de ∨ ($I_\vee$)

$$\frac{A}{A \vee B} \qquad \frac{B}{A \vee B}$$

T[p] ⊢ p ∨ r

1.	p	premisa
2.	p ∨ r	**$I_\vee$ (1)**

T[p] ⊢ r ∨ p

1.	p	premisa
2.	r ∨ p	**$I_\vee$ (1)**

Natural Deduction: Implement a simple example

$$T[\ s \wedge (p \vee q),\ \ p \rightarrow \neg r,\ \ q \rightarrow \neg r\ ] \vdash\ \ s \wedge \neg r$$

Natural Deduction: Implement a simple example

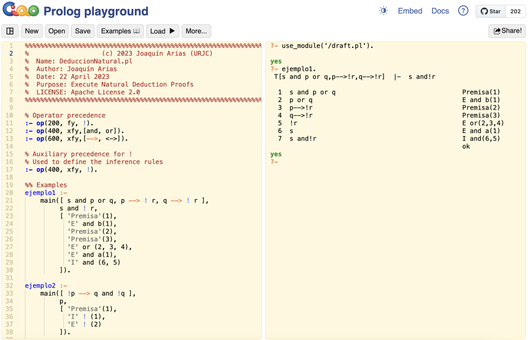
$$T[\ s \wedge (p \vee q),\quad p \rightarrow \neg r,\quad q \rightarrow \neg r\ ] \vdash \quad s \wedge \neg r$$

1.	$s \wedge (p \vee q)$	*premisa*
2.	$p \vee q$	$E_{\wedge}(1)$
3.	$p \rightarrow \neg r$	*premisa*
4.	$q \rightarrow \neg r$	*premisa*
5.	$\neg r$	$E_{\vee}(2,3,4)$
6.	s	$E_{\wedge}(1)$
7.	$s \wedge \neg r$	$I_{\wedge}(5,6)$

Natural Deduction: Implement a simple example

$$T[\, s \wedge (p \vee q),\quad p \rightarrow \neg r,\quad q \rightarrow \neg r \,] \vdash \quad s \wedge \neg r$$

1.	$s \wedge (p \vee q)$	*premisa*
2.	$p \vee q$	$E_{\wedge}(1)$
3.	$p \rightarrow \neg r$	*premisa*
4.	$q \rightarrow \neg r$	*premisa*
5.	$\neg r$	$E_{\vee}(2,3,4)$
6.	s	$E_{\wedge}(1)$
7.	$s \wedge \neg r$	$I_{\wedge}(5,6)$



Demo

Natural Deduction by Gentzen III

Introducción ¬ ($I_\neg$)

$$\frac{A \rightarrow B \land \neg B}{\neg A}$$

Eliminación ¬ ($E_\neg$)

$$\frac{\neg\neg A}{A}$$

$T[\neg p \rightarrow q \land \neg q] \vdash p$	
1. $\neg p \rightarrow q \land \neg q$	premisa
2. $\neg\neg p$	$I_\neg$ (1)
3. p	$E_\neg$ (2)

Natural Deduction by Gentzen IV

Eliminación → ($E_{\rightarrow}$)

$$\frac{A \rightarrow B \quad A}{B}$$

$T[p \rightarrow \neg r, \neg r \rightarrow q, p] \vdash q$

1.	$p \rightarrow \neg r$	premisa
2.	p	premisa
3.	$\neg r$	**$E_{\rightarrow}$ (1,2)**
4.	$\neg r \rightarrow q$	premisa
5.	q	**$E_{\rightarrow}$ (3,4)**

Introducción → ($I_{\rightarrow}$)

$$\frac{A \text{ (supuesto)} \quad B}{A \rightarrow B}$$

$T[p \rightarrow q, q \rightarrow r] \vdash p \rightarrow r$

1.	$p \rightarrow q$	premisa
2.	$q \rightarrow r$	premisa
3.	p	supuesto
4.	q	$E_{\rightarrow}$ (1,3)
5.	r	$E_{\rightarrow}$ (2,4)
6.	$p \rightarrow r$	**$I_{\rightarrow}$ (3,5)**

Natural Deduction by Gentzen V

Introducción ↔ ($I_{\leftrightarrow}$)

$$\frac{A \rightarrow B \quad B \rightarrow A}{A \leftrightarrow B}$$

$T[p \rightarrow \neg r, \neg r \rightarrow p] \vdash p \leftrightarrow \neg r$	
1. $p \rightarrow \neg r$	premisa
2. $\neg r \rightarrow p$	premisa
3. $p \leftrightarrow \neg r$	**$I_{\leftrightarrow}$ (1,2)**

Eliminación ↔ ($E_{\leftrightarrow}$)

$$\frac{A \leftrightarrow B}{A \rightarrow B} \qquad \frac{A \leftrightarrow B}{B \rightarrow A}$$

$T[p \leftrightarrow q \wedge r, p] \vdash r$	
1. $p \leftrightarrow q \wedge r$	premisa
2. $p \rightarrow q \wedge r$	**$E_{\leftrightarrow}$ (1)**
3. p	premisa
4. $q \wedge r$	$E_{\rightarrow}$ (2,3)
5. r	$E_{\wedge}$ (4)

Natural Deduction: Implement derived rules

$$\frac{A \rightarrow B \qquad \neg B}{\neg A} \quad \textit{Modus Tollens (MT)}$$

Natural Deduction: Implement derived rules

$$\frac{\begin{array}{c} A \to B \\ \neg B \end{array}}{\neg A} \quad \textit{Modus Tollens (MT)}$$

- Implementation:

```
rule('MT',                      % Name
      [ FA --> FB,              % Hypotheses
        !FB ],
      !FA,                      % Deduction
      [ 'Premisa'(1),           % Proof
        'Premisa'(2),
        'Supuesto'(FA),
        'E' --> (1,3),
        'I' and (4,2),
        'I' --> (3,5),
        'I' ! (6) ]).
```

Natural Deduction: Implement derived rules

$$\frac{A \rightarrow B \qquad \neg B}{\neg A} \quad \textit{Modus Tollens (MT)}$$

- Implementation:

```
rule('MT',                    % Name
       [ FA --> FB,           % Hypotheses
         !FB ],
       !FA,                   % Deduction
       [ 'Premisa'(1),        % Proof
         'Premisa'(2),
         'Supuesto'(FA),
         'E' --> (1,3),
         'I' and (4,2),
         'I' --> (3,5),
         'I' ! (6) ]).
```

- Generated proof:

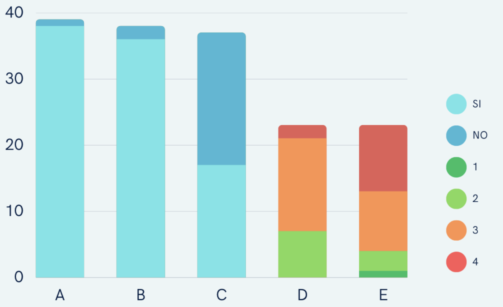
```
MT: T[A --> B, !B]  |-  !A

1  A --> B            Premisa(1)
2  !B                 Premisa(2)
3       A             Supuesto(A)
4       B             E-->(1,3)
5       B and!B       I and(4,2)
6  A --> B and!B      I-->(3,5)
7  !A                 I!6
                      ok
```

Natural Deduction: Home work (solve with/without derived rules)

$$T[\ r \rightarrow q \wedge s, \quad \neg(q \wedge s)\ ] \vdash \quad \neg r$$

Natural Deduction: Pilot experience during the 22/23 academic year.

- Small sample population but promising results.



Please answer Yes or No:
A: Have you used the program?
B: Have you used the program to check manifestations?
C: Have you used the program as a study tool?

Rate from 1 (bad) to 4 (very good):
D: Ease of use.
E: Usefulness for learning Natural Deduction.

Figure: Results of the DeduccionNatural.pl satisfaction survey on the Artificial Intelligence degree.

Conclusions

- I have presented DeduccionNatural.pl (open source).
  https://github.com/Xuaco/DeduccionNatural

Conclusions

- I have presented DeduccionNatural.pl (open source).

  https://github.com/Xuaco/DeduccionNatural
- Other examples in the paper:
  - Practice in the secure development methodology course (third year of the Cybersecurity degree):

    http://platon.etsii.urjc.es/~jarias/mmm/MDS/
  - Final Degree Projects:
    - "MMDect: Metamorphic Malware Detection using Logic Programming" by Luciana.

      http://platon.etsii.urjc.es/~jarias/tfg/23-Luciana.pdf
    - "Modeling and estimation of criminal sentences using Logic Programming" by Elena

      http://platon.etsii.urjc.es/~jarias/tfg/23-Elena.pdf

Conclusions

- I have presented DeduccionNatural.pl (open source).

  https://github.com/Xuaco/DeduccionNatural
- Other examples in the paper:
  - Practice in the secure development methodology course (third year of the Cybersecurity degree):

    http://platon.etsii.urjc.es/~jarias/mmm/MDS/
  - Final Degree Projects:
    - "MMDect: Metamorphic Malware Detection using Logic Programming" by Luciana.

      http://platon.etsii.urjc.es/~jarias/tfg/23-Luciana.pdf
    - "Modeling and estimation of criminal sentences using Logic Programming" by Elena

      http://platon.etsii.urjc.es/~jarias/tfg/23-Elena.pdf
- FUTURE WORK: Keep going... https://tinyurl.com/2bz87p82

Conclusions

- I have presented DeduccionNatural.pl (open source).
  https://github.com/Xuaco/DeduccionNatural
- Other examples in the paper:
  - Practice in the secure development methodology course (third year of the Cybersecurity degree):
    http://platon.etsii.urjc.es/~jarias/mmm/MDS/
  - Final Degree Projects:
    - "MMDect: Metamorphic Malware Detection using Logic Programming" by Luciana.
      http://platon.etsii.urjc.es/~jarias/tfg/23-Luciana.pdf
    - "Modeling and estimation of criminal sentences using Logic Programming" by Elena
      http://platon.etsii.urjc.es/~jarias/tfg/23-Elena.pdf
- FUTURE WORK: Keep going... https://tinyurl.com/2bz87p82

Thanks!

Bibliography I

[1] Arias, Joaquín (2022). **Lógica: desde Aristóteles hasta Prolog**. http://hdl.handle.net/10115/20014. Servicio de Publicaciones URJC. ISBN: 978-84-09-38265-1.

[2] Arias, Joaquín, Ramírez, Iván, and Gallinari, Alessandra (2023). **DeduccionNatural.pl: herramienta escrita en Prolog para el aprendizaje de la asignatura de Lógica**. In: *JENUI 2023*, pp. 129–136.

[3] Gallinari, Alessandra (2009). **Apuntes y problemas de la lógica matemática**. Madrid: Dykinson. ISBN: 978-84-98-49475-4.

[4] Gentzen, Gerhard (1935). **Untersuchungen über das logische schließen. I.** In: *Mathematische Zeitschrift* 35, pp. 176–210.