

Biductive Computing: Several Variants of a Universal Paradigm

**EAAI-18
Joshua Eckroth
Stetson University**



CSCI 431 / Artificial Intelligence

csci431.artifice.cc

Guest

Notes

- Syllabus
- Logic programming with Prolog
- Prolog unification
- Prolog resolution
- Biductive Prolog
- Prolog parsing
- Misc examples
- Bayesian inference
- ProbLog
- Naïve Bayesian inference
- Classification
- Decision trees
- Text processing
- Neural networks
- Deep learning

Assignments

- A01: Prolog Pokédex
due Aug 30, 11:59pm
- A02: Course Scheduler
due Sep 20, 11:59pm
- A03: CiteMan
due Oct 6, 11:59pm
- A04: MMA Bookie
due Oct 25, 11:59pm
- A05: Predicting student performance
due Oct 30, 11:59pm
- A06: Detecting Spam Comments
due Nov 8, 11:59pm
- A07: Detecting Spam Comments Part Deux
due Nov 20, 11:59pm
- A08: Identifying Birds
due Dec 1, 11:59pm

CSCI 431

Fall 2017

Dr. Joshua Eckroth
jeckroth@stetson.edu

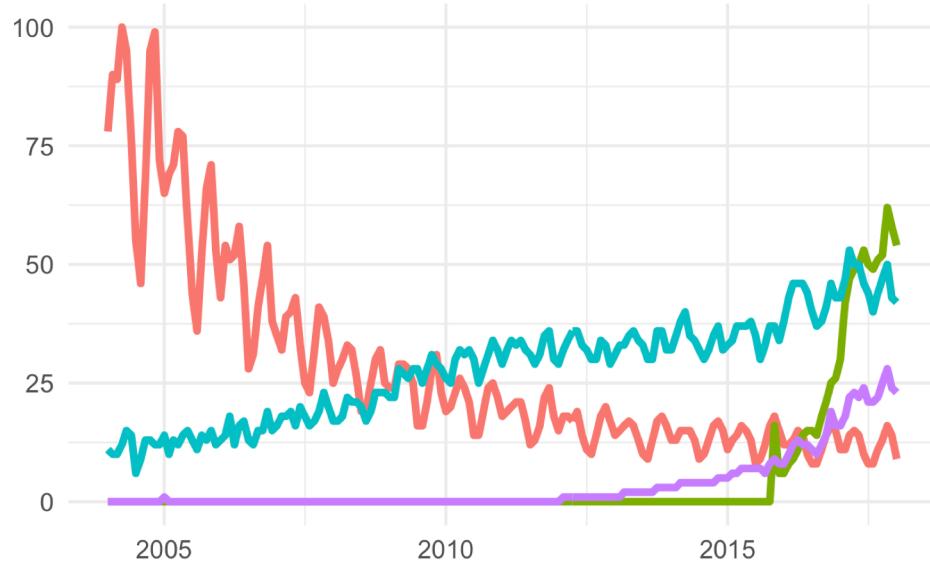
Office hours:
Mon 1-3, Tue 5:30-6, Wed 2-3,
Thur 5:30-6

Goals

- Practice with deductive and abductive reasoning
- Learn a new programming paradigm
- Write code *carefully* with intricate interactions
- Encourage a way of solving problems that transcends the everyday procedural mindset

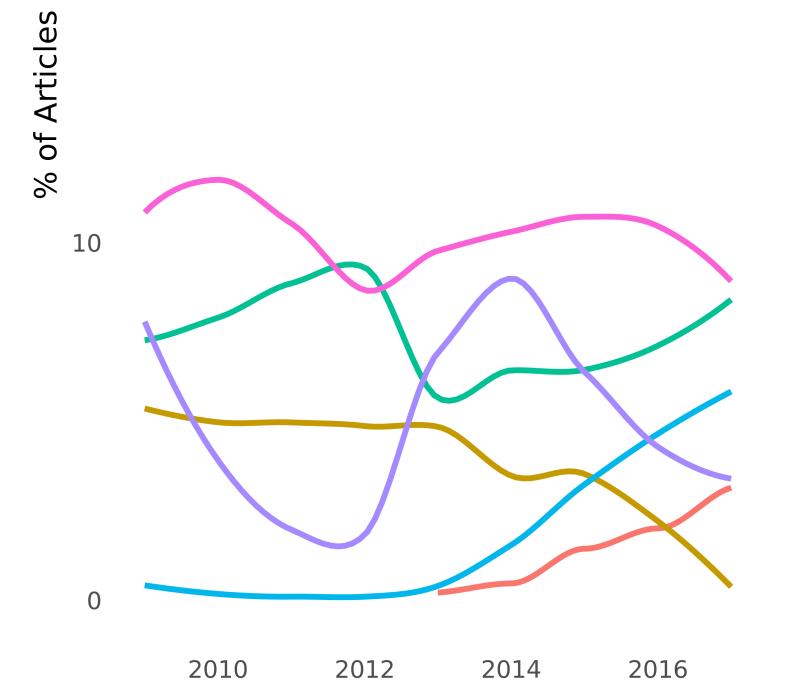
“Prolog: Completely, 100% useless right up until it’s exactly the right tool for a very hard, very specific job.”

— u/TheTarquin, Sep 2016



Google Trends

Prolog
TensorFlow
OpenCV
scikit.learn



AAAI Conference Proceedings

Deep Learning
Logic & Formal Reasoning
Machine Learning
Natural Language
Neural Networks
Planning & Scheduling
Statistical Learning

Related platforms



“We note that it is not the case that expert systems died. Rather, after a few years, they became more standard practice than innovation. Fewer papers were published about novel applications of expert systems. They disappeared into the fabric, now applied everywhere, from the high-end emulation of rare human experts, to the embedding and application of rule books and procedure manuals.”

— Smith & Eckroth, “Building AI Applications: Yesterday, Today, and Tomorrow,” *AI Magazine* 38(1) Spring 2017

Domain	Deductive	Abductive	“Biductive”
Database querying	<code>childPok(ampharos, pikachu, Child)</code>	<code>childPok(X, Y, pichu)</code>	<code>childPok(ampharos, Y, Z), Z \= cleffa</code>
Planning	<code>studentGPA([(cs141, a), (cs142, b+), ...], GPA)</code>	<code>studentGPA(Grades, 3.75)</code>	<code>studentGPA([(cs141, a), (cs142, G1), Rest)], GPA), GPA >= 3.0, G1 < 3.5</code>
Parsing	<code>parseRef(mla, "...", {author: A, year: Y, ...})</code>	<code>parseRef(mla, S, {author: "J. Doe", year: 2018, ...})</code>	<code>parseRefFmt("J. Doe," + S, {author: A, year: 2018, ...})</code>
Probabilistic reasoning	<code>crime(jdoe, [action1, action2, ...], Evidence) => (ev1, prob1), (ev2, prob2)</code>	<code>crime(Perp, Actions, [ev1, ev2, ...]) => (jdoe, actions, prob)</code>	<code>crime(Perp, [action1, ActRest], [ev1, EvRest]) => (jdoe, actions, evidence, prob)</code>

Example with parsing

X Cite

MLA Clocksin, William F., and Christopher S. Mellish. *Programming in PROLOG*. Springer Science & Business Media, 2003.

APA Clocksin, W. F., & Mellish, C. S. (2003). *Programming in PROLOG*. Springer Science & Business Media.

Chicago Clocksin, William F., and Christopher S. Mellish. *Programming in PROLOG*. Springer Science & Business Media, 2003.

Harvard Clocksin, W.F. and Mellish, C.S., 2003. *Programming in PROLOG*. Springer Science & Business Media.

Vancouver Clocksin WF, Mellish CS. Programming in PROLOG. Springer Science & Business Media; 2003 Jul 25.

BibTeX EndNote RefMan RefWorks

≡ Google Scholar prolog

Articles About 285,000 results (0.07 sec)

Any time Since 2018 Since 2017 Since 2014 Custom range...

Sort by relevance Sort by date

include patents include citations

Create alert

[BOOK] **Programming in PROLOG**
WF Clocksin, CS Mellish - 2003 - books.google.com
Originally published in 1981, this was the first textbook on programming in the **Prolog** language and is still the definitive introductory text on **Prolog**. Though many **Prolog** textbooks have been published since, this one has withstood the test of time because of its
☆ 99 Cited by 4907 Related articles All 18 versions

[BOOK] **The art of Prolog: advanced programming techniques**
L Sterling, EY Shapiro - 1994 - books.google.com
This new edition of The Art of **Prolog** contains a number of important changes. Most background sections at the end of each chapter have been updated to take account of important recent research results, the references have been greatly expanded, and more
☆ 99 Cited by 2870 Related articles All 12 versions

[BOOK] **The craft of Prolog**
RA O'Keefe, RA O'Keefe - 1990 - mitpress.mit.edu
Hacking your program is no substitute for understanding your problem. **Prolog** is different, but not that different. Elegance is not optional. These are the themes that unify Richard O'Keefe's very personal statement on how **Prolog** programs should be written. The emphasis in The Craft
☆ 99 Cited by 490 Related articles All 5 versions

[CITATION] **Prolog** programming for artificial intelligence
I Bratko - 2001 - Pearson education
☆ 99 Cited by 2283 Related articles All 9 versions

Goal: Build a reference manager and query tool.

Example with parsing

X Cite

MLA Clocksin, William F., and Christopher S. Mellish. *Programming in PROLOG*. Springer Science & Business Media, 2003.

APA Clocksin, W. F., & Mellish, C. S. (2003). *Programming in PROLOG*. Springer Science & Business Media.

Chicago Clocksin, William F., and Christopher S. Mellish. *Programming in PROLOG*. Springer Science & Business Media, 2003.

Harvard Clocksin, W.F. and Mellish, C.S., 2003. *Programming in PROLOG*. Springer Science & Business Media.

Vancouver Clocksin WF, Mellish CS. Programming in PROLOG. Springer Science & Business Media; 2003 Jul 25.

BibTeX EndNote RefMan RefWorks

```
{authors: [("William, F.", "Clocksin"), ...]  
title: "Programming in PROLOG",  
publisher: "Springer Science...",  
year: 2003}
```

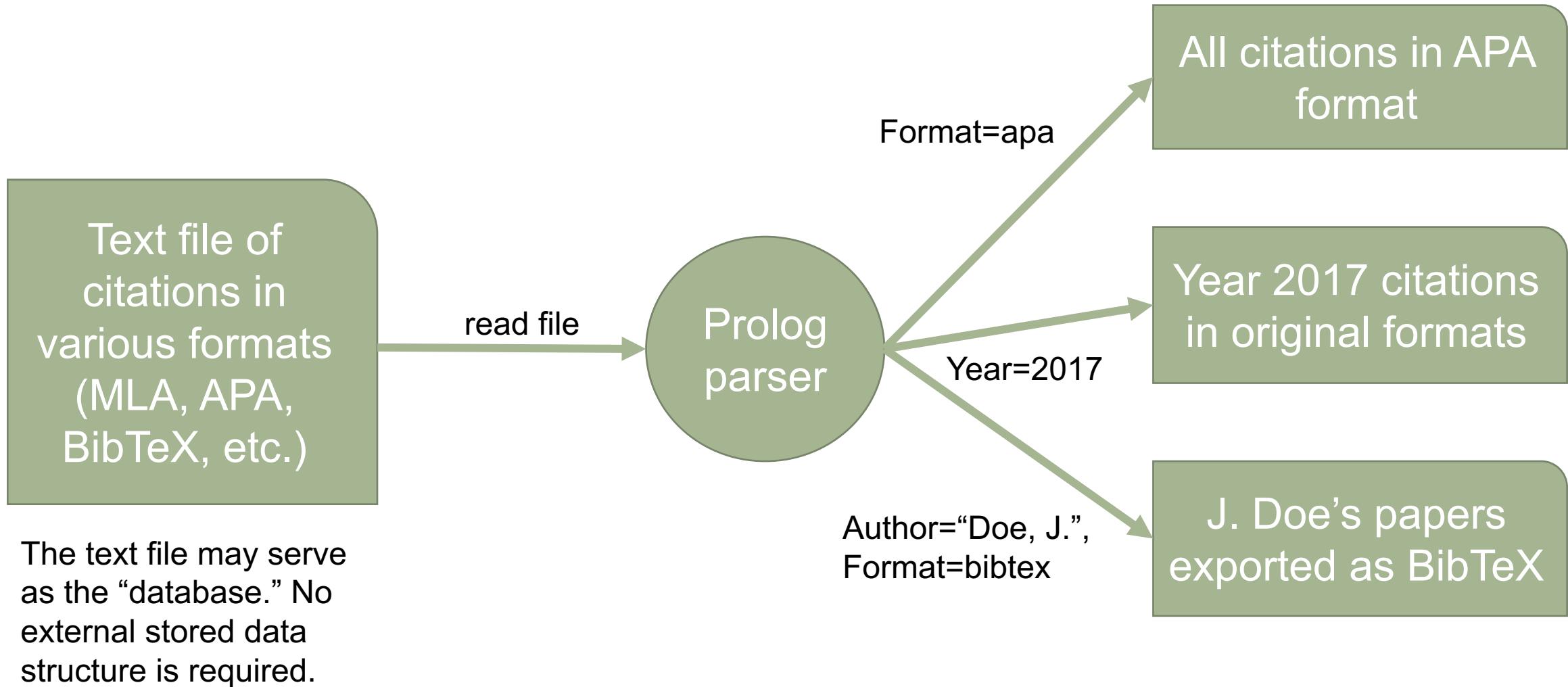
Format=apa

Clocksin, W. F., & Mellish, C. S. (2003).
Programming in PROLOG. Springer Science &
Business Media.

Same code (written in definite clause grammar)
parses & renders in various formats.

In 50 lines of code.

Example with parsing



Challenges

- Though code is short, it must be carefully crafted.
- Ordering of code matters.
- Variables must be constrained as soon as possible, sometimes on both sides of their use (forward & backward) - e.g., `member(X, List)`.
- Adding a feature can break other code or introduce infinite recursion.
- Prolog is not something students can learn on their own from stackoverflow.

Student reactions

- Prolog has few theoretical constructs and it is consistent
- Students said they found the Prolog assignments easier than using Pandas

pandas iloc vs ix vs loc explanation?



Can someone explain how these three methods of slicing are different?

284

I've seen [the docs](#), and I've seen [these answers](#), but I still find myself unable to explain how the three are different. To me, they seem interchangeable in large part, because they are at the lower levels of slicing.



202

For example, say we want to get the first five rows of a `DataFrame`. How is it that all three of these work?

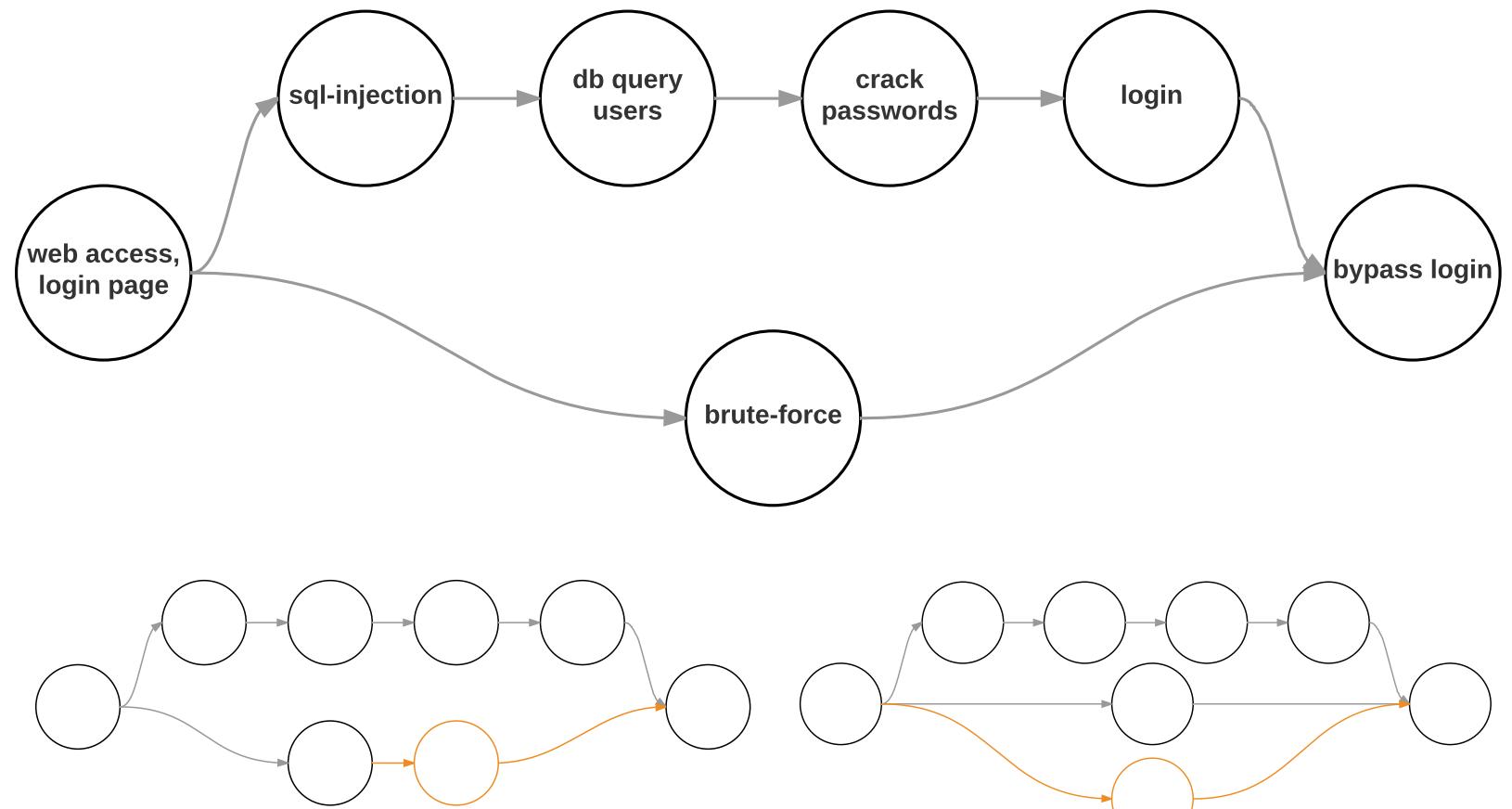


```
df.loc[:5]  
df.ix[:5]  
df.iloc[:5]
```

Can someone present three cases where the distinction in uses are clearer?

ALPACA: Building Dynamic Cyber Ranges with Procedurally-Generated Vulnerability Lattices

Kim Chen '18



Questions?