# CS7480: Topics in Programming Languages: Probabilistic Programming

## Lecture 2: Probability & Logic

**Instructor**: Steven Holtzen

**Place**: Northeastern University

**Term**: Fall 2021

**Course webpage**: https://www.khoury.northeastern.edu/home/sholtzen/CS7480Fall21/

# Brief notes

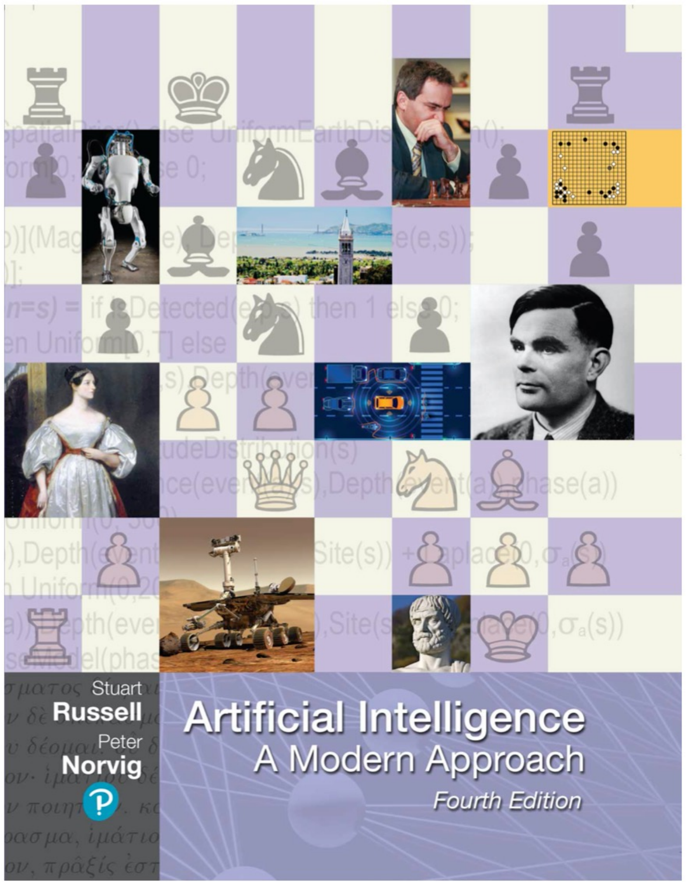
- `SimPPL` draft still in progress
  - Will be done by end of next week (once we have finished the intro material necessary to do the project)
- We have a Teams chat now
- Slides are on canvas

# Propositional Logic Goals

- Get a good foundation in ***propositional logic***
  - Syntax, semantics, connectives
  - Models & Satisfiability
  - Inference rules
- …Will be essential for understanding probabilistic program semantics and syntax

- It will be our first "programming language"

# Resources ("Books on my self")

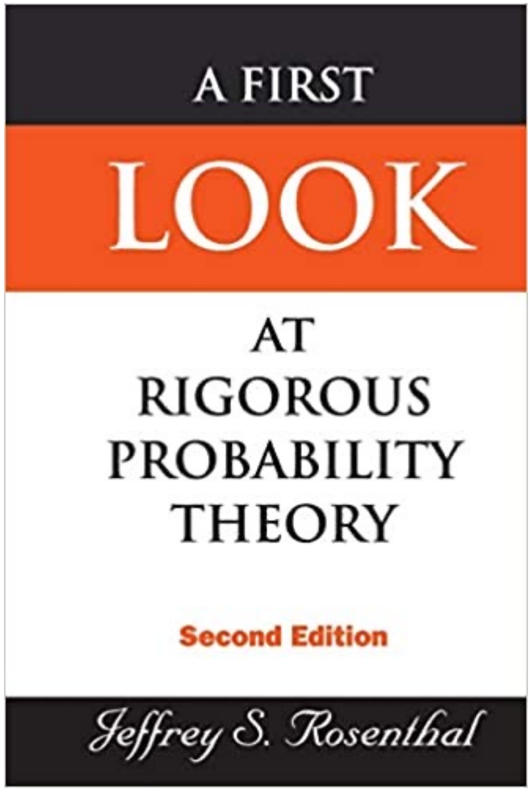



- A foundational text with many good definitions; covers insane number of topics
- Chapter 12 on probability
- Chapter 7 on logic

# Resources ("Books on my self")

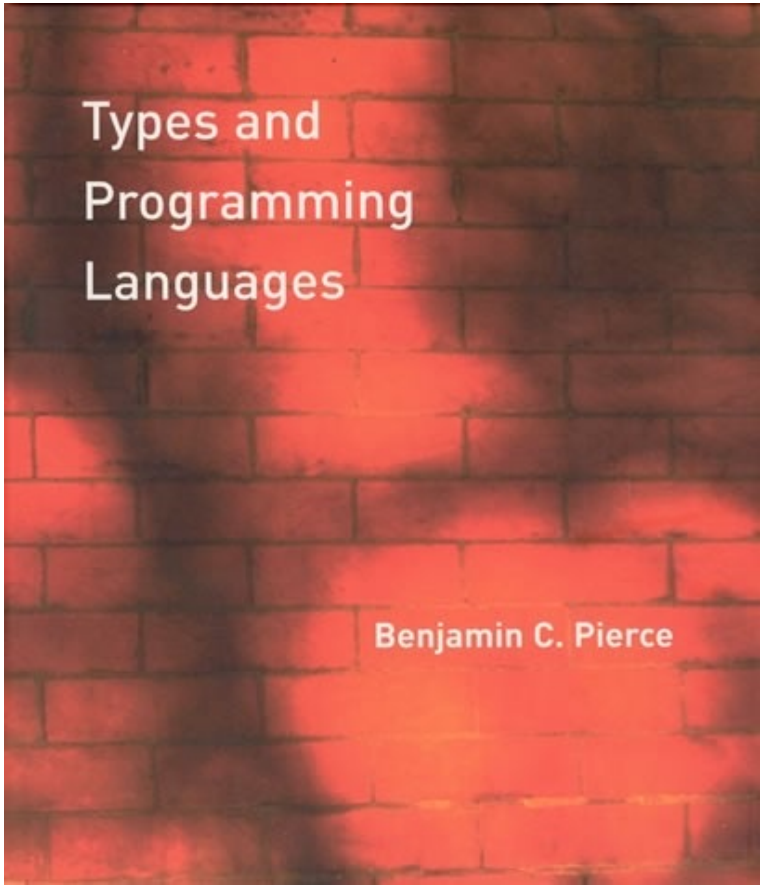

- A beautiful philosophical argument for probability in AI
- Chapter 1: Why bother with uncertainty
- Chapter 2: Bayesian inference

# Resources ("Books on my self")



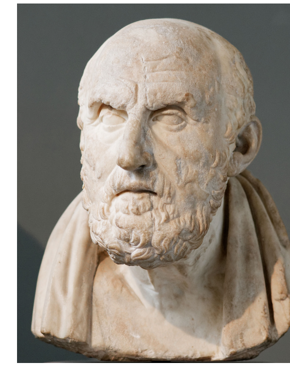

- Foundations for continuous probability theory (measure theory)

# Resources (“Books on my self”)




- Classic book on type systems in functional programming
- Lots of good stuff on inductive rules, semantics, etc.

# Propositional Logic



Chrysippus of Soli
c. 279BCE

- One of the earliest forms of *formal reasoning*
  - Stoics believed in *determinism of fate*
  - Developed formal logic “in order to better understand the workings of the universe and role of humanity within it”
- Goal is to study *premises* and *conclusions*
  1. If it is raining, then it is cloudy
  2. It is raining.
  - Conclusion: Therefore it is cloudy.
- Every statement is either *true* or *false*
- Early logic was carefully worded natural language

# Syntax

- *Propositional atoms:* written as $A, B$ …
  - Free symbols (“variables”) that take on truth values
  - E.g., encode “It is raining” as $R$, “It is cloudy” as $C$

- *Logical sentence:* a finite inductive combination of:
  - Can be an atom
  - Negation: $\neg\alpha$
  - Conjunction: $\alpha \wedge \beta$
  - Disjunction: $\alpha \vee \beta$
  - Implication: $\alpha \Rightarrow \beta$

# Syntax: Backus-Naur Form

```
a ::= A, B, C, …
s ::=
        a
      | s ∧ s
      | s ∨ s
      | ¬ s
      | s ⇒ s
      | ( s )
```

Defines syntax of atoms

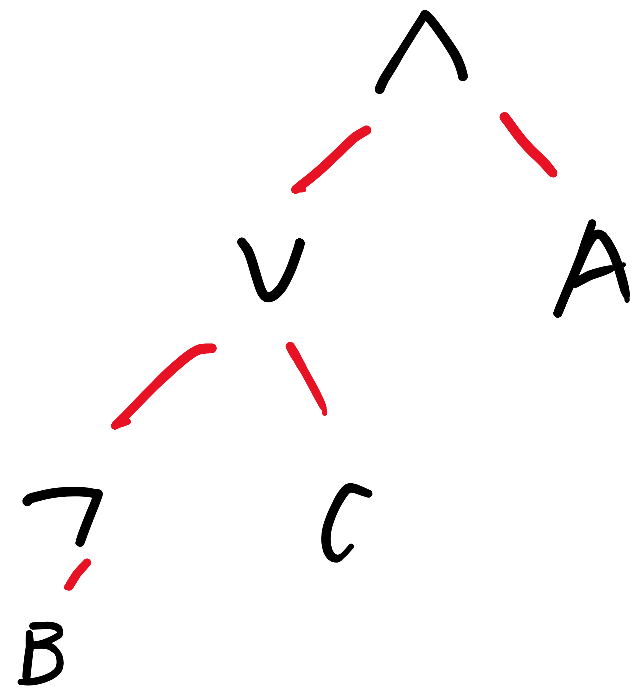
Syntax of sentences

- Inductive definition
- Either an atom…
- OR a conjunction of sentences
- OR a disjunction of a sentence
- …

Q: is $A \wedge B \wedge C$ syntactically valid?

# Syntax Tree (aka Parse Tree)

- $\left(A \wedge \left(C \vee \neg B\right)\right)$



# Syntax Tree (aka Parse Tree)

- Q: Give the parse tree for $\neg\big((A \wedge A) \vee \neg(A \vee B)\big)$

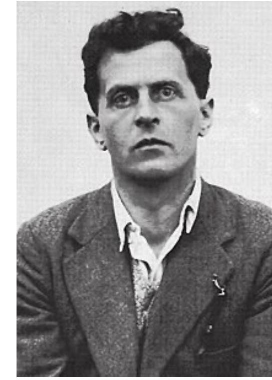
# *Semantics*: Evaluating a Sentence

- So far we've shown how to construct sentences, but we *do not know what they mean!*
- **Goal of semantics**: describe how to *evaluate* a propositional formula *in some context*
- An *assignment* $v: atom \rightarrow \{\top, \bot\}$ associates *atoms* with *truth values*
- Denote *evaluating* $\alpha$ under assignment $v$ as $[\![\alpha]\!]v$

# *Semantics*: Rules

- We can write a system of equations that tells us how to evaluate every term

# Truth Tables



Ludwig Wittgenstein
1889 – 1951

- Evaluate a sentence *on all valuations*
  - Sometimes denoted $[\![\alpha]\!]$
  - Sometimes called simply “the semantics”

| $A$ | $B$ | $[\![A \vee B]\!]v$ |
| --- | --- | --- |
| $\top$ | $\top$ | $\top$ |
| $\bot$ | $\bot$ | $\bot$ |
| … | | |

- Tells you *everything* about that formula
  - Valuations $v$ that evaluate to $\top$ are called *models;* often denoted $v \vDash \alpha$

# Analysis

- Satisfiability: For a given sentence, find a valuation that evaluates to $\top$
  - ”Can I find an input that gives me a desired output”
  - Generating a truth table (exhaustive search) is exponential in #vars; not ideal
  - Well-known NP-complete problem

- Model counting

# Search Trees

- Search tree for $(A \vee B) \wedge (\neg C)$

# Logic Conclusion

- Propositional logic is a mini programming language!
- Syntax and a semantics
  - It has valuations: “We can run it”
  - Possible worlds, truth tables, search procedures
- Analysis problems: “Can we make it output a particular value”