Chemical
Biochemical and
Environmental
Engineering

Lean for Scientists and Engineers

Tyler R. Josephson
AI & Theory-Oriented Molecular Science (ATOMS) Lab
University of Maryland, Baltimore County

Summer Dream (ft. Chevy)
Kirara Magic

Twitter: @trjosephson
Email: tjo@umbc.edu

Lean for Scientists and Engineers 2024

1. Logic and proofs for scientists and engineers
    1. Introduction to theorem proving
    2. Writing proofs in Lean
    3. Formalizing derivations in science and engineering
2. Functional programming in Lean 4
    1. Functional vs. imperative programming
    2. Numerical vs. symbolic mathematics
    3. Writing executable programs in Lean
3. Provably-correct programs for scientific computing

Schedule (tentative)

Logic and proofs for scientists and engineers
Functional programming in Lean 4
Provably-correct programs for scientific computing

July 9, 2024	Introduction to Lean and proofs
July 10, 2024	Equalities and inequalities
July 16, 2024	Proofs with structure
July 17, 2024	Proofs with structure II
July 23, 2024	Proofs about functions; types
July 24, 2024	Calculus-based-proofs
July 30-31, 2024	Prof. Josephson traveling
August 6, 2024	Functions, definitions, structures, recursion
August 8, 2024	Polymorphic functions for floats and reals, compiling Lean to C
August 13, 2024	Input / output, lists, arrays, and indexing
August 14, 2024	Lists, arrays, indexing, and matrices
August 20, 2024	LeanMD & BET Analysis in Lean
August 21, 2024	SciLean tutorial, by Tomáš Skřivan

Content inspired by:
Mechanics of Proof, by Heather Macbeth
Functional Programming in Lean, by David Christiansen

Guest instructor: Tomáš Skřivan

Schedule for today

1. Survey for attendees
2. Recap Lecture 3
3. Not
4. Exists
5. Forall
6. Implication
7. If and only iff

Slide from Lecture 3

Should you use have or add a hypothesis?

Using have

```
example {a b : ℝ}
  (h1 : a - 5 * b = 4)
  (h2 : b + 2 = 3) :
  a = 9 := by
  have hb : b = 1 := by linarith
  calc
    a = a - 5 * b + 5 * b := by ring
    _ = 4 + 5 * 1 := by rw [h1, hb]
    _ = 9 := by ring
```

New hypothesis

```
example {a b : ℝ}
  (h1 : a - 5 * b = 4)
  (h2 : b + 2 = 3)
  (hb : b = 1):
  a = 9 := by
  calc
    a = a - 5 * b + 5 * b := by ring
    _ = 4 + 5 * 1 := by rw [h1, hb]
    _ = 9 := by ring
```

We've changed the theorem statement.
h2 is "unused"
We don't know if hb is true!
If hb contradicts any other hypotheses, we're in real trouble

Slide from Lecture 3

Principle of logical explosion

- You MUST NOT assume a set of premises with a contradiction
- “Principle of explosion”
- https://en.wikipedia.org/wiki/Principle_of_explosion
- Also known as “proving false”
- You can prove anything, which isn’t actually helpful
- Lean has tactic “slim_check” that can sometimes detect this by searching for counterexamples
  - Examples here: https://github.com/leanprover-community/mathlib4/blob/master/test/slim_check.lean

Slide from Lecture 3

How to find tactics

- Keep learning them one by one!
- Indexes for Mechanics of Proof, Mathematics in Lean
- Consult lists of useful tactics
  - https://github.com/madvorak/lean4-tactics
  - https://github.com/Colin166/Lean4/blob/main/UsefulTactics
- If you have a tactic in hand, mouseover in VS Code to see documentation and example(s)

Slide from Lecture 3

How to find theorems

- Keep practicing!
- Search Mathlib documentation
  - https://leanprover-community.github.io/mathlib4_docs/
  - Using the search bar, make a guess about what the theorem would be named, and start checking things that look promising
- Moogle
  - https://www.moogle.ai
  - Describe theorem (or definition) in natural language, the scroll through options
- Consult lists of useful theorems
  - https://github.com/Colin166/Lean4/blob/main/UsefulLemmas.lean
- If you have a theorem in hand, mouseover in VS Code to see documentation and example(s)

Slide from Lecture 3

Glossary of logical symbols

$\wedge$ - and

$\vee$ - or

$\neg$ - not

$\rightarrow$ - implies

$\leftrightarrow$ - if and only if (implies in both directions)

$\exists$ - exists

$\forall$ - for all

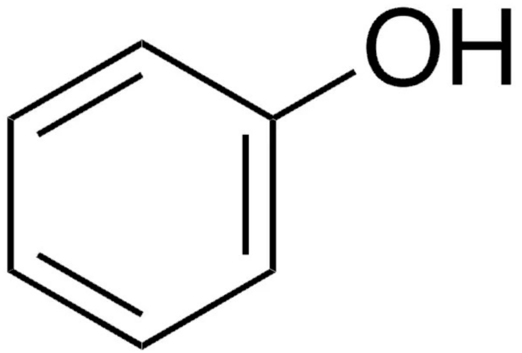
Slide from Lecture 3

∧ : and

P: molecule is aromatic

Q: molecule is an alcohol

P ∧ Q: molecule is aromatic and an alcohol



Phenol

P: true, Q: true – then P ∧ Q: true

P: false, Q: true – then P ∧ Q: false

P: true, Q: false – then P ∧ Q: false

P: false, Q: false – then P ∧ Q: false

P	Q	(P ∧ Q)
true	true	true
false	true	false
true	false	false
false	false	false

Slide from Lecture 3

∨ : or

P: contains acrolein

Q: contains hydrogen cyanide

P ∨ Q: acute toxicity

P: true, Q: true – then P ∨ Q: true

P: false, Q: true – then P ∨ Q: true

P: true, Q: false – then P ∨ Q: true

P: false, Q: false – then P ∨ Q: false

P	Q	(P ∨ Q)
true	true	true
false	true	true
true	false	true
false	false	false

¬ : not

Unary operation (applies to just one term)

P: true, then ¬ P : false

P: false, then ¬ P : true

P	¬P
true	false
false	true

→ : implies

P → Q means "if P, then Q"

P: detecting argon

Q: detecting a noble gas

P → Q: detecting argon implies detecting a noble gas

P: true, Q: true – then P → Q: true

P: false, Q: true – then P → Q: true

P: true, Q: false – then P → Q: false

P: false, Q: false – then P → Q: true

→ : implies

P → Q means "if P, then Q"

P: detecting argon

Q: detecting a noble gas

P → Q: detecting argon implies detecting a noble gas

P: true, Q: true – then P → Q: true

P: false, Q: true – then P → Q: true

P: true, Q: false – then P → Q: false

P: false, Q: false – then P → Q: true

P	Q	(P → Q)
true	true	true
false	true	true
true	false	false
false	false	true

→ : implies

P → Q means “if P, then Q”

P: detecting argon

Q: detecting a noble gas

P → Q: detecting argon implies detecting a noble gas

P: true, Q: true – then P → Q: true

P: false, Q: true – then P → Q: true

P: true, Q: false – then P → Q: false

P: false, Q: false – then P → Q: true

P	Q	(P → Q)
true	true	true
false	true	true
true	false	false
false	false	true

More about implication

Every proof you've written with hypotheses involves implication

```
example {x y : ℝ}
(h1 : 450 = (x + y)*3 )
(h2 : 450 = (x - y)*5 )
: x = 120 :=
by linarith
```

Implication isn't fundamental!

$P \rightarrow Q$ can be rewritten as $\neg P \lor Q$

This is less intuitive for humans, but useful for automated provers

Causality in science can be described using the language of implication

Implication is NOT equality – it only goes one way

"Smoking → lung cancer" not "lung cancer → smoking"

↔ : if and only if

P: A molecule is a hydrocarbon

Q: A molecule contains only carbon and hydrogen atoms

P ↔ Q: A molecule is a hydrocarbon if and only if it contains only carbon and hydrogen atoms.

P: true, Q: true – then P ↔ Q: true

P: false, Q: true – then P ↔ Q: false

P: true, Q: false – then P ↔ Q: false

P: false, Q: false – then P ↔ Q: true

P	Q	(P ↔ Q)
true	true	true
false	true	false
true	false	false
false	false	true

Truth tables

and

P	Q	(P ∧ Q)
true	true	true
false	true	false
true	false	false
false	false	false

or

P	Q	(P ∨ Q)
true	true	true
false	true	true
true	false	true
false	false	false

not

P	¬P
true	false
false	true

implies

P	Q	(P → Q)
true	true	true
false	true	true
true	false	false
false	false	true

if and only if

P	Q	(P ↔ Q)
true	true	true
false	true	false
true	false	false
false	false	true

Truth tables

and

P	Q	(P ∧ Q)
true	true	true
false	true	false
true	false	false
false	false	false

or

P	Q	(P ∨ Q)
true	true	true
false	true	true
true	false	true
false	false	false

not

P	¬P
true	false
false	true

implies

P	Q	(P → Q)
true	true	true
false	true	true
true	false	false
false	false	true

if and only if

P	Q	(P ↔ Q)
true	true	true
false	true	false
true	false	false
false	false	true

Compound statements can be reasoned about by combining these

Consider ¬ (P ∧ ¬ Q)

P	Q	¬Q	(P ∧ ¬Q)	¬(P ∧ ¬Q)
true	true	false	false	true
false	true	false	false	true
true	false	true	true	false
false	false	true	false	true

Proofs about pure logic, MoP Ch. 5

If you abstract away the numbers, definitions, equations and inequalities, you are left with pure logic problems. And the pure logic tactics like `obtain`, `apply`, `constructor`, and so on can still be used.

```
example {P Q : Prop} (h1 : P ∨ Q) (h2 : ¬ Q) : P := by
  obtain hP | hQ := h1
  · apply hP
  · contradiction
```

It's hardly worth trying to write proofs like this in words. The P and Q are abstract propositions (`Prop`), and this is just a game of manipulation.

Definitions

- Allow us to reuse terms outside of individual examples / theorems
- Facilitates modular code and verification of different parts of code
- Propositions
- Functions

∃ : exists

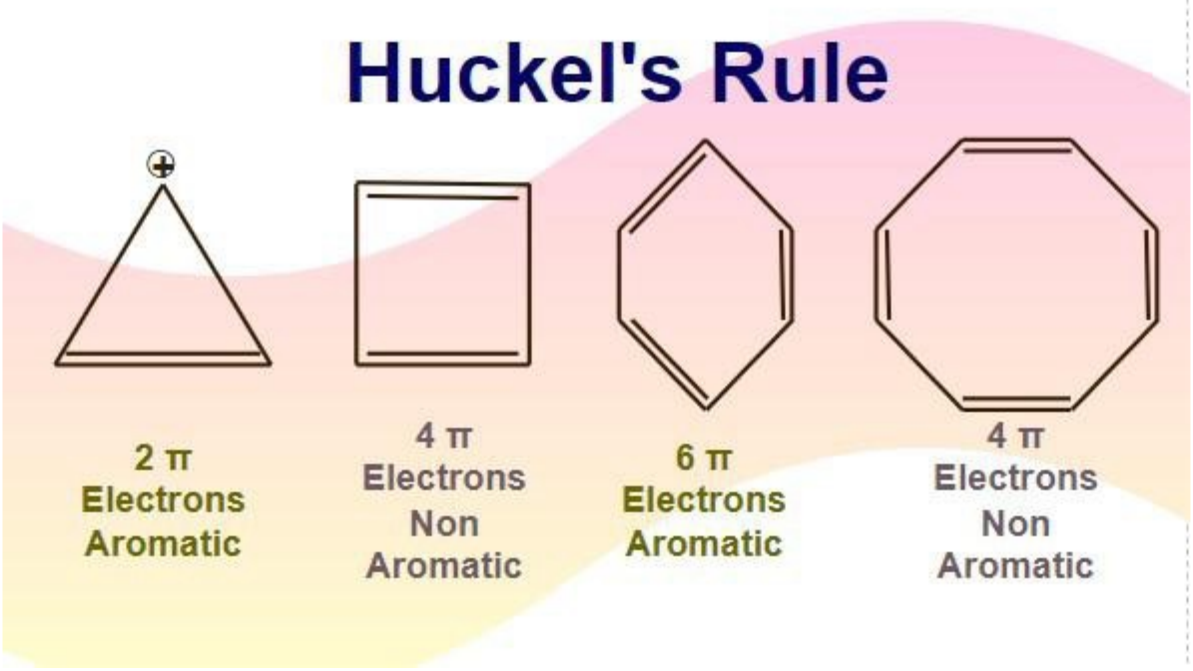
- Something is true if you have at least one instance of it being true
- "There exists an element with atomic number 43"
- Proposition only needs one example to be true
  - This is called a "witness"

- If a hypothesis has an existential, use the Lean tactic "obtain"
  - We're skipping this case today; see MoP 2.5.1 and 2.5.2
- If a goal has an existential, use the Lean tactic "use"
  - See MoP 2.5.3 and MoP 3.1

Huckel’s rule

M is the number of pi electrons in a ring system and n is some natural number.
Huckel's rule says that the number of electrons must be able to satisfy 4n + 2
We state this as "there exists some n such that M = 4n + 2”

$$\exists\ n,\ (M = 4*n + 2)$$



∀ : for all

- Something is true if every possible instance of it is true
- All transition metals have unfilled d-orbitals
    - FALSE – Zn is a transition metal with a filled 3d-orbital
    - Only takes one example to falsify

$\forall$: for all

- Go to Example 4.1.1 in Mechanics of Proof

Combining ∀ and ∃ to represent "proportional to"

- Empirical science often describes phenomena using proportions
- Kepler's Third Law – orbital period T vs. semi-major axis d

$$T^2 \propto d^3 \qquad T^2 = \left(\frac{4\pi^2}{G(m+M)}\right)d^3 \qquad T^2 = \left(\frac{4\pi^2}{GM}\right)d^3$$

Combining ∀ and ∃ to represent "proportional to"

- Empirical science often describes phenomena using proportions
- Boyle's Law
  - Pressure is inversely proportional to volume

$$P \propto \frac{1}{V} \qquad P = \frac{k}{V} \qquad PV = k \qquad P_1V_1 = P_2V_2$$

Combining ∀ and ∃ to represent "proportional to"

- Empirical science often describes phenomena using proportions
- Boyle's Law
  - Pressure is inversely proportional to volume

$$P \propto \frac{1}{V} \qquad P = \frac{k}{V} \qquad PV = k \qquad P_1V_1 = P_2V_2$$

- *There exists* some constant k, such that *for all* thermodynamic states, this relationship between pressure and volume holds

Combining ∀ and ∃ to represent "proportional to"

- Empirical science often describes phenomena using proportions
- Boyle's Law
  - Pressure is inversely proportional to volume

$$P \propto \frac{1}{V} \qquad P = \frac{k}{V} \qquad PV = k \qquad P_1V_1 = P_2V_2$$

- *There exists* some constant k, such that *for all* thermodynamic states, this relationship between pressure and volume holds
- ∃ (k : ℝ), ∀ (n : ℕ), (P n)*(V n) = k
- Next time!