Lecture 1: What Is a Proof?

ISE Block 1.2 Maths · Week 1

Why proof

A polynomial of primes

$$n^2 + n + 41$$

- **Prime**: an integer greater than 1 that is divisible only by itself and 1
- **Divisible**: m is divisible by d when $m = dk$ for some integer k
- $n = 0, 1, 2, 3$ give 41, 43, 47, 53: all prime
- keep going, and the primes keep coming for a long while
- **Question**: how many primes in a row would convince you that it **always** gives a prime?
  - 5? 10? 40? 1000?

Propositions and definitions

Propositions

- **Proposition**: a statement that is either true or false
- **Examples**
  - $2 + 3 = 5$: a true proposition
  - $1 + 1 = 3$: a false proposition, and still a proposition
- **Non-examples**
  - a question: “Is 91 prime?”
  - a command: “Give me an A!”
  - a statement whose truth depends on unstated circumstances: “It’s five o’clock”
- ***Being a proposition is not the same as being true.***

Proposition or not?

For each: a proposition or not? If it is: true, false, or not known?

1. 91 is prime.
2. Is 91 prime?
3. $x + 1 = 5$
4. Every even integer greater than 2 is a sum of two primes.

	Statement	Proposition?	Truth value
1	91 is prime	yes	false: $91 = 7 \cdot 13$
2	Is 91 prime?	no: a question	
3	$x + 1 = 5$	it depends on x	
4	Goldbach's conjecture	yes	not known

- **Goldbach's conjecture**, said "GOLT-bakh": stated in 1742, still unproved
- ***It is true or false; nobody yet knows which.***

Definitions and claims

$$p(n) ::= n^2 + n + 41$$

- **Definition**: a statement that gives a new name a meaning, using only terms already defined; written with $::=$
  - read "p of n is defined to be n squared plus n plus 41"
  - this line introduces a meaning; it is not an equality to prove
- **Claim**: a proposition stated as true but not yet proved
  - $p(1) = 43$ is a true claim; $p(2) = 45$ is a false one
  - a claim can be false, so it needs justifying; here, computing $p(1)$ and $p(2)$ settles both
- **Mixing them up**: trying to prove a definition, or treating a claim as if it were true by definition

- **Authors differ**
  - MCS writes $::=$, though its footnote says a plain $=$ is “always ok”
  - others write $:=$, or $=$ and leave you to work out which is meant
- **This major**: $::=$ for every definition, $=$ only for claims
  - the symbol makes clear whether we are defining something or asserting it
- **Habit**: when a source writes $=$, ask which kind of line it is
- **Python already keeps them apart**
  - `x = 5` binds a name, like a definition; it is not a proposition
  - `x == 5` asks a question: it gives `True` or `False`

Testing a universal claim

The claim

- **In words**: “For every nonnegative integer, n, the value of $n^2 + n + 41$ is prime.”
- **In our notation**: for every $n \in \mathbb{N}$, $p(n)$ is prime
  - read “for every n in $\mathbb{N}$, p of n is prime”; “for all” means the same
  - $\in$ is read “is in”: $n \in \mathbb{N}$ says n is a natural number
- **Quantifiers**: “for every” and “for some”
- **Universal claim**: a claim about every value in a set, its **domain**
  - $\mathbb{Z}$ is the integers $\ldots, -2, -1, 0, 1, 2, \ldots$
  - “for every n, $n \geq 0$” is true over $\mathbb{N}$, false over $\mathbb{Z}$ ($n = -1$)
- which values does $\mathbb{N}$ contain?

Does $\mathbb{N}$ contain 0?

- **This major and MCS**: $\mathbb{N} = \{0, 1, 2, \ldots\}$
- **The Leaving Certificate**: $\mathbb{N}$ starts at 1
  - authors who start at 1 write $\mathbb{N}_0$ when they want 0
  - when we want to leave 0 out, we write $\mathbb{Z}^+$, the positive integers
- **Why include 0**: a count can be zero, and Python counts from 0: `range(40)` is $0, 1, \ldots, 39$
- **What changes**: with 0, the claim includes $p(0) = 41$; the run of primes is 40 long from 0, 39 from 1
- **Habit**: say which convention you use; check an author's before you rely on it

Run it

$$\text{for every } n \in \mathbb{N},\ p(n) \text{ is prime}$$

```
def p(n: int) -> int:
    return n**2 + n + 41

print(all(is_prime(p(n)) for n in range(40)))
print(all(is_prime(p(n)) for n in range(41)))
```

- `all(...)` is `True` exactly when every item is `True`
  - read the first line as "for every n in `range(40)`, $p(n)$ is prime"
- **Predict** both lines before they run

```
True
False
```

The test at n = 40

- $p(0), \ldots, p(39)$ are all prime, ending at $p(39) = 1601$
- then $n = 40$:

$$\begin{aligned} p(40) &= 40^2 + 40 + 41 \\ &= 40 \cdot 41 + 41 \\ &= 41 \cdot 41 \end{aligned}$$

- $p(40) = 1681$ is divisible by 41, an integer greater than 1 other than $p(40)$ itself, so $p(40)$ is not prime
- $n = 40$ is a **counterexample**: one value in the domain where the claim fails
- ***One counterexample makes a universal claim false.***

Evidence and proof

- **Checking finitely many values** of a claim about infinitely many
  - can refute it: one counterexample is enough
  - can never establish it, “no matter how large the finite set”
- **Goldbach’s conjecture**: checked by computer for a vast range of even numbers, and still unproved
- **In software**: passing tests show the tested inputs work; they say nothing certain about the inputs not tested
- **Named flaw**, **example offered as proof**: checking a few cases, then writing “so it holds for all n”
- **Finite claims are different**: checking every case of a finite claim is a proof

Predicates

Predicates

- **Proposition**: a statement that is either true or false
- **Predicate**: a statement that becomes a proposition when values are assigned to one or more variables
- **Example**: $x + 1 = 5$ is a predicate of x
  - true or false? It depends on x

x	the proposition	its truth value
3	$3 + 1 = 5$	false
4	$4 + 1 = 5$	true
5	$5 + 1 = 5$	false

- one predicate, a different proposition for each value of x
- **From predicate to proposition**
  - give every variable a value, as in each row
  - or make a universal claim: “for every integer x, $x + 1 = 5$” is a proposition, and false
- **Universal claim, again**: it says a predicate is true for every value of its domain; MCS calls it a **universal quantification**

A predicate for primes

- **Examples**: 2, 3, 5, 7 and 11 are prime; 1 is not; $91 = 7 \cdot 13$ is not
- **As a predicate**, named so we can use it:

$$\mathrm{Prime}(m) ::= m \text{ is prime}$$

- **Applied to $p(n)$**: $\mathrm{Prime}(p(n))$ says "$p(n)$ is prime"
  - so the claim is: for every $n \in \mathbb{N}$, $\mathrm{Prime}(p(n))$
  - $\mathrm{Prime}(p(4))$ is true: $p(4) = 61$, which is prime
  - $\mathrm{Prime}(4)$ is false

Read it, then run it

```
print(p(4), is_prime(p(4)))
```

```
61 True
```

- p **returns a number**: $p(4)$ is an `int`
- Prime **is a predicate**: $\mathrm{Prime}(p(4))$ is true or false, a `bool`
- `is_prime(m)` tests divisibility by integers d from 2 while $d^2 \leq m$
- a predicate can be represented as a Boolean-valued function; here `is_prime` returns `bool`

Proof

Proofs and axioms

- **Proof**: a sequence of logical deductions from axioms and previously proved statements that concludes with the proposition in question
- **Axiom**: a proposition simply accepted as true
- **Names for proved propositions**
  - theorem: an important true proposition
  - lemma: a preliminary proposition useful for proving later ones

Our axioms

- **What you may use**: familiar school facts
  - the definitions already given
  - Leaving Certificate algebra: expand, factor, rearrange
  - a sum or product of integers is an integer
- ***The claim being proved is never an allowed fact, however familiar.***
  - “odd plus odd is even” is a school fact; when it is the theorem, citing it proves nothing
- **When unsure** whether a step needs proof: say what you are assuming

Direct proof

Implications

- **Implication**: a proposition of the form if P, then Q, written P IMPLIES Q
  - P is the **hypothesis**, Q the **conclusion**
- **Meaning**: whenever P is true, Q must be true
  - a counterexample would make P true and Q false
- **With variables**: "if m and n are even, then $m + n$ is even"
  - the full claim: for all integers m and n, if m and n are even, then $m + n$ is even
  - the domain, the integers, often goes unsaid; a proof must say it

Direct proof

- **Direct proof** of P IMPLIES Q
  1. Write “Assume P.”
  2. Derive Q from P and the allowed facts.
- **Why assume P?** We must show what follows whenever P holds
  - this does not assert that P is always true
- **With a hidden “for all”**: first name the values, then assume
  - “Let m and n be integers. Assume m and n are even.”

Even and odd

- **Even**: an integer m is even $::= m = 2k$ for some $k \in \mathbb{Z}$
- **Odd**: an integer m is odd $::= m = 2k + 1$ for some $k \in \mathbb{Z}$
- **Read aloud**: “for some k in $\mathbb{Z}$” means “for at least one integer k”
- **Examples**: $6 = 2 \cdot 3$, $0 = 2 \cdot 0$, $-4 = 2 \cdot (-2)$ are even; $-3 = 2 \cdot (-2) + 1$ is odd
- **Non-example**: $7 = 2 \cdot \frac{7}{2}$, but $\frac{7}{2} \notin \mathbb{Z}$ (“is not in”), so this does not make 7 even

- **Unpack**: when you are told m is even
  - you get an integer, call it a, with $m = 2a$
  - different numbers may need different integers: $6 = 2 \cdot 3$, $10 = 2 \cdot 5$
  - so use a new letter for each number: $m = 2a$, $n = 2b$
- **Build**: when you must show s is even
  - find an integer k with $s = 2k$: a **witness**
  - name it and say why it is an integer: it shows the reader exactly what makes s even
  - for odd, aim for $s = 2k + 1$

Even, as code

```
def is_even(m: int) -> bool:
    return any(m == 2 * k for k in range(-abs(m), abs(m) + 1))

print(is_even(6), is_even(0), is_even(-4), is_even(7))
```

```
True True True False
```

- read: "m is even when $m = 2k$ for some integer k"
- `any(...)` is `True` exactly when at least one item is `True`: it is "for some", as `all(...)` is "for every"
- the search can stop at $|m|$: if $m = 2k$, then $|k| \leq |m|$
- it checks the values it is given; it proves nothing about the rest

Scratchwork

Claim: for all integers m and n, if m and n are even, then $m + n$ is even.

- **Scratchwork**: the working that finds a proof, kept apart from it
- **Aim**: $m + n = 2 \cdot (\text{some integer})$
- **Try examples**: $6 + 10 = 2 \cdot 8$; $-4 + 2 = 2 \cdot (-1)$
  - examples help with scratchwork, but do not prove the claim
- **Unpack**: $m = 2a$, $n = 2b$, so $m + n = 2a + 2b$
- **Dead end**: "$2a$ and $2b$ are even, and even plus even is even"
  - that repeats the claim instead of proving it
- **Factor**: $2a + 2b = 2(a + b)$, so try $k = a + b$
- scratchwork may be messy; the proof must be clear

The proof, written up

Theorem: for all integers m and n, if m and n are even, then $m + n$ is even.

Proof (direct).

1. Let m and n be integers, and assume m and n are even. ← let and assume
2. By the definition of even, $m = 2a$ and $n = 2b$ for some integers a and b. ← unpack: the hypothesis is used here
3. Then $m + n = 2a + 2b = 2(a + b)$. ← compute
4. Let $k = a + b$. It is an integer, since a sum of integers is an integer. ← build the witness
5. So $m + n = 2k$ with $k \in \mathbb{Z}$, and $m + n$ is even by the definition of even. ■ ← conclude

- **Scratchwork and proof**: the aim, the examples and the dead end stay in the scratchwork; the proof keeps only steps that follow, each with its reason
- **Proof** marks where a proof starts; ■ marks where it ends

The arbitrary element

- **Arbitrary integers**: “let m and n be integers” fixes no particular values
  - we then assume both are even, as the theorem requires
  - the argument must work for every pair satisfying that hypothesis
- **Question**: does any line depend on which even m and n we chose? No
  - every line uses only $m = 2a$ and $n = 2b$, which every pair of even integers has
  - so the argument does not depend on the pair: it is one argument for all of them
- **Check**: choose any even m and n; the same argument still applies
- **A chosen pair**: “Let $m = 4$ and $n = 6$. Then $m + n = 10 = 2 \cdot 5$.”
  - true, but it covers one pair: **example offered as proof**

Your first proof

Your first proof

Theorem: for all integers m and n, if m and n are odd, then $m + n$ is even.

- work alone: scratchwork first, then the proof
- use the two moves: unpack, then build

Theorem: for all integers m and n, if m and n are odd, then $m + n$ is even.

Hints, one at a time:

- start: “Let m and n be integers, and assume m and n are odd.”
- unpack each by the definition of odd. Can m and n share one k? Try 3 and 5.
- add, and pull out a factor of 2
- name the witness, and say why it is an integer

Wrong turns

- **Invalid: one k for both**: "$m = 2k + 1$ and $n = 2k + 1$"
  - "m is odd" gives an integer a with $m = 2a + 1$; "n is odd" gives an integer b with $n = 2b + 1$
  - nothing in the hypothesis says $a = b$; one k for both assumes it, which means assuming $m = n$
  - so the argument covers only pairs like $3 + 3$; for 3 and 5, $a = 1$ and $b = 2$
  - this is a **dropped quantifier**: each number's own "for some" was lost
- **Invalid: examples instead**: "$3 + 5 = 8$ and $7 + 9 = 16$, so ..." checks two cases

- **A missing step**: "$m + n = 2a + 2b + 2$, so it's even"
  - the conclusion is right, but unexplained: factor as $2(a + b + 1)$; since $a + b + 1$ is an integer, the definition of even applies
  - a beginner should write that step
- **Valid, less explicit**: "$m + n = 2(a + b + 1)$, so $m + n$ is even"
  - the factor is shown; the reader supplies "$a + b + 1$ is an integer"
- **Fully explicit**: "let $k = a + b + 1$, an integer since a sum of integers is an integer"
  - not required for validity, but every step can be checked: the model to follow

A model proof

Theorem: for all integers m and n, if m and n are odd, then $m + n$ is even.

Proof (direct).

1. Let m and n be integers, and assume m and n are odd. ← let and assume
2. By the definition of odd, $m = 2a + 1$ and $n = 2b + 1$ for some integers a and b. ← unpack: the hypothesis is used here
3. So $m + n = 2a + 2b + 2 = 2(a + b + 1)$. ← compute
4. Let $k = a + b + 1$. It is an integer, since a sum of integers is an integer. ← build the witness
5. So $m + n = 2k$ with $k \in \mathbb{Z}$, and $m + n$ is even by the definition of even. ■ ← conclude

Tests and proofs

Tests and proofs

- **The question**: how many passing tests would prove “for every $n \in \mathbb{N}$, $\mathrm{Prime}(p(n))$”?
  - each passing test confirms one more value, not the universal claim
  - finitely many tests cannot settle a claim about every n in $\mathbb{N}$
  - here the first 40 tests passed, and the next, $n = 40$, failed
- **The parity proofs**: each argued about arbitrary integers m and n, so one argument covered every pair at once
- **For software**
  - tests check the inputs they run
  - a correctness proof covers inputs within its stated specification and assumptions

Exit question

“The product of two odd integers is odd.”

- **Response (a)**: a program checks every pair of odd integers from 1 to 999; every product is odd
- **Response (b)**: “Let m and n be integers, and assume both are odd. Then $m = 2k + 1$ and $n = 2k + 3$ for some integer k, so …”
- Could either establish the claim? What is wrong with each?
- **(a)**: strong evidence, not a proof; it checks finitely many pairs
- **(b)**: the right method, but one k for both assumes $n = m + 2$, which the hypothesis does not give; unpack as $m = 2a + 1$, $n = 2b + 1$

Conclusion

Learning outcomes

You should now be able to:

- tell a proposition from a predicate
- tell a definition, written with ::=, from a claim
- read a universal claim and its domain
- tell evidence from proof: passing tests settle the cases tested; a proof covers every case
- write a direct proof of a universal implication about integers
  - name arbitrary elements, assume the hypothesis, unpack the definitions
  - build a witness, say why it is an integer, and state the conclusion

Recommended reading

- **MCS §§1.1–1.5.1** (pp. 5–13): propositions, predicates, axioms and proving an implication
  - available from [MIT OpenCourseWare](https://ocw.mit.edu/courses/6-042j-mathematics-for-computer-science-spring-2015/)
- Before reading a worked proof, try writing its opening steps

Questions?

eoin@eoin.ai · eoin.p.o.brien@ul.ie