CSC110 LECTURE 6: INTRODUCTION TO FORMAL LOGIC

David Liu and Tom Fairgrieve, Department of Computer Science

*Navigation tip for web slides: press **?** to see keyboard navigation controls.*

UOFT LIBRARY OF THE DAY

THE KNOX COLLEGE LIBRARY

Knox College Library

ANNOUNCEMENTS & PLAN FOR TODAY

- Assignment 1 has been posted! Due next Wed.
  - Check the FAQ (+ corrections) page.
  - Additional TA Office Hours (schedule on Quercus).
- Join a Recognized Study Group.

A NEW CHAPTER

Chapter 3: Formal Logic in Computer Science

TODAY YOU’LL LEARN TO...

1. Express boolean expressions using formal logical notation, including propositional operators and two quantifiers
2. Translate boolean expressions between English, mathematical notation, and Python expressions
3. Perform filtering computations on Python collections
4. Interpret statements in predicate logic with multiple quantifiers, and express them in Python

Much of this may be review from MAT137/157. A different perspective may aid understanding!

WHY LOGIC?

Mathematical logic is a language of boolean expressions.

“$3 + 2 = 5$”

“Python’s `sorted` function is correct.”

“My program is correct, assuming Python’s `sorted` function is correct.”

“Python `set`s are more efficient than Python `list`s (for certain operations).”

PROPOSITIONAL LOGIC (REVIEW FROM PREP)

TERMINOLOGY

A **proposition** is a statement that is True or False.

Propositional variables ($p, q, \ldots$) are used to represent a propositions.

Can build **propositional formula**s using propositional variable(s) and the **propositional operators**:

- $\neg$ (**NOT**)
- $\vee$ (**OR**)
- $\wedge$ (**AND**)
- $\Rightarrow$ (implication/conditional)
- $\Leftrightarrow$ (bi-implication/biconditional)

A propositional formula evaluates to True or False.

EXAMPLES

Let $p =$ "David is cool" and $q =$ "Tom is cool".

Expression	Meaning
$\neg p$	"David is not cool"
$p \vee q$	"David is cool or Tom is cool (or both!)
$p \wedge q$	"David is cool and Tom is cool"
$p \Rightarrow q$	"If David is cool then Tom is cool"
$p \Leftrightarrow q$	"David is cool if and only if Tom is cool"

MORE ON IMPLICATION (1)

$p \Rightarrow q$: “If David is cool then Tom is cool”

- p is the hypothesis (“David is cool”)
- q is the conclusion (“Tom is cool”)

Equivalent to $\neg p \vee q$: “David is not cool or Tom is cool”.

p	q	$p \Rightarrow q$
F	F	T
F	T	T
T	F	F
T	T	T

MORE ON IMPLICATION (2)

$p \Rightarrow q$: "If David is cool then Tom is cool"

Equivalent to $\neg q \Rightarrow \neg p$: "If Tom is not cool then David is not cool".

- $\neg q \Rightarrow \neg p$ is the **contrapositive** of $p \Rightarrow q$.

MORE ON IMPLICATION (3)

$p \Rightarrow q$: “If David is cool then Tom is cool”

NOT equivalent to $q \Rightarrow p$: “If Tom is cool then David is cool”.

- $q \Rightarrow p$ is the **converse** of $p \Rightarrow q$.

REPRESENTING PROPOSITIONAL OPERATORS IN PYTHON

operator	notation	Python operation
NOT	$\neg p$	`not p`
AND	$p \wedge q$	`p and q`
OR	$p \vee q$	`p or q`
implication	$p \Rightarrow q$	`not p or q`
biconditional	$p \Leftrightarrow q$	`p == q`

PREDICATE LOGIC

TERMINOLOGY

A **predicate** is a function that returns a boolean.

Instead of “David is cool”, “Tom is cool”, “Sophia is cool”, etc., we can define a predicate:

$P(x)$: “x is cool”, where x is a person.

THE TWO QUANTIFIERS

$\forall x \in S,\ P(x)$ — “for all elements x of S, $P(x)$ is True”

$\exists x \in S,\ P(x)$ — “there exists an element x of S satisfies $P(x)$”

EXAMPLES (1)

Let S be the set of all people, and $IsCool$ the "x is cool" predicate defined over S.

Translate: $IsCool(david)$ (assuming $david \in S$)

- "david is cool."

Translate: $\forall p \in S,\ IsCool(p)$

- "Every person in S is cool."

Translate: $\exists p \in S,\ IsCool(p)$

- "At least one person in S is cool."

EXAMPLES (2)

Let S be the set of all people, with two predicates over S defined as:

- $IsCool(x)$: “x is cool”
- $LikesDogs(x)$: “x likes dogs”

Translate: $\forall p \in S,\ IsCool(p) \vee LikesDogs(p)$

- “Every person in S is cool or likes dogs.”

Translate: “At least one person who is cool does not like dogs.”

$\exists p \in S,\ IsCool(p) \wedge \neg LikesDogs(p)$

EXERCISE 1: PROPOSITIONAL AND PREDICATE LOGIC

THE QUANTIFIERS IN PYTHON (`all`/`any`)

`all(bools)`: given a collection of booleans, return whether they are all `True`.

`any(bools)`: given a collection of booleans, return whether at least one is `True`.

EXAMPLE (1)

Let `numbers` refer to a set of numbers.

Goal: translate this logical statement into Python:

$$\forall x \in numbers,\ x < 50$$

Demo!

```
all({x < 50 for x in numbers})
```

EXAMPLE (2)

Let `numbers` refer to a set of numbers, and suppose we have two Python predicates `is_prime` and `is_special`.

Goal: translate this logical statement into Python:

$$\forall x \in numbers, is\_prime(x) \Leftrightarrow is\_special(x)$$

```
all({is_prime(x) == is_special(x) for x in numbers})
```

EXERCISE 2: TRANSLATING LOGICAL STATEMENTS INTO PYTHON

operator	notation	Python operation
NOT	$\neg p$	`not p`
AND	$p \land q$	`p and q`
OR	$p \lor q$	`p or q`
implication	$p \Rightarrow q$	`not p or q`
biconditional	$p \Leftrightarrow q$	`p == q`

quantifier	notation	Python operation
universal	$\forall x \in S,\ \ldots$	`all({... for x in S})`
existential	$\exists x \in S,\ \ldots$	`any({... for x in S})`

CONDITIONS AND FILTERING

EXPRESSING CONDITIONS

"Every natural number n satisfies the inequality $n^2 + n \geq 20$."

$$\forall n \in \mathbb{N},\ n^2 + n \geq 20$$

"Every natural number n greater than 3 satisfies the inequality $n^2 + n \geq 20$."

✔ $\forall n \in \mathbb{N},\ n > 3 \Rightarrow n^2 + n \geq 20$

❌ $\forall n \in \mathbb{N},\ n > 3 \wedge n^2 + n \geq 20$

"greater than 3" is a condition that narrows the scope of the universal quantifier in the statement.

THE “FORALL-IMPLIES” STRUCTURE

“Every element of S that ____ satisfies ____.”

$$\forall x \in S,\ P(x) \Rightarrow Q(x)$$

More generally:

$$\forall x \in S,\ P_1(x) \wedge \cdots \wedge P_n(x) \Rightarrow Q_1(x) \wedge \cdots \wedge Q_m(x)$$

FILTERING IN PYTHON

In Python, we often want to filter a collection.

- Given a set of integers, find the sum of all the even numbers
- Given a set of (x, y) points, find the number of points within 1 unit of $(0, 0)$
- Given a list of strings, find the maximum length of a string that contains `'David'`

Normal (set) comprehension

```
{<expression> for <variable> in <collection>}
```

Filtering (set) comprehension

```
{<expression> for <variable> in <collection> if <condition>}
```

Demo!

EXERCISE 3: FILTERING COLLECTIONS

NESTED DATA PREVIEW: MULTIPLE QUANTIFIERS

EXTENDED EXAMPLE: “A LOVES B”

Let $A = \{Breanna, Malena, Patrick, Ella\}$ and $B = \{Sophia, Thelonious, Stanley, Laura\}$.

$Loves : A \times B \rightarrow \{True, False\}$

$Loves(a, b)$ is defined as “person a loves person b”.

	Sophia	**Thelonious**	**Stanley**	**Laura**
Breanna	False	True	True	False
Malena	False	True	True	True
Patrick	False	False	True	False
Ella	False	False	True	True

- $Loves(Breanna, Thelonious)$ is True
- $Loves(Patrick, Thelonious)$ is False

PRACTICE: ONE QUANTIFIER

	Sophia	**Thelonious**	**Stanley**	**Laura**
Breanna	False	True	True	False
Malena	False	True	True	True
Patrick	False	False	True	False
Ella	False	False	True	True

True or False: $\forall a \in A,\ Loves(a, Stanley)$

True or False: $\forall b \in B,\ Loves(Ella, b)$

MULTIPLE QUANTIFIERS, SAME TYPE (1)

	Sophia	**Thelonious**	**Stanley**	**Laura**
Breanna	False	True	True	False
Malena	False	True	True	True
Patrick	False	False	True	False
Ella	False	False	True	True

True or False: $\forall a \in A,\ \forall b \in B,\ Loves(a, b)$

“Every person in A loves every person in B.”

MULTIPLE QUANTIFIERS, SAME TYPE (2)

	Sophia	**Thelonious**	**Stanley**	**Laura**
Breanna	False	True	True	False
Malena	False	True	True	True
Patrick	False	False	True	False
Ella	False	False	True	True

True or False: $\exists a \in A,\ \exists b \in B,\ Loves(a, b)$

“There exists a person in A who loves a person in B.”

MULTIPLE QUANTIFIERS, DIFFERENT TYPE (1)

	Sophia	**Thelonious**	**Stanley**	**Laura**
Breanna	False	True	True	False
Malena	False	True	True	True
Patrick	False	False	True	False
Ella	False	False	True	True

Consider:

$$\forall a \in A,\ \exists b \in B,\ Loves(a, b)$$

$$\forall a \in A,\ \underline{\big(\exists b \in B,\ Loves(a, b)\big)}$$

"For every person a in A, there exists a person b in B who a loves."

MULTIPLE QUANTIFIERS, DIFFERENT TYPE (1)

$$\forall a \in A,\ \underline{\left(\exists b \in B,\ Loves(a, b)\right)}$$

“For every person a in A, there exists a person b in B who a loves.”

	Sophia	**Thel**	**Stanley**	**Laura**
Breanna	False	True	True	False
Malena	False	True	True	True
Patrick	False	False	True	False
Ella	False	False	True	True

	Sophia	**Thel**	**Stanley**	**Laura**
Breanna	False	**True**	True	False
Malena	False	**True**	True	True
Patrick	False	False	**True**	False
Ella	False	False	True	**True**

Key idea: Because the $\exists b$ is contained within the $\forall a$, the “b” person can be different for each “a” person.

MULTIPLE QUANTIFIERS, DIFFERENT TYPE (2)

	Sophia	Thelonious	Stanley	Laura
Breanna	False	True	True	False
Malena	False	True	True	True
Patrick	False	False	True	False
Ella	False	False	True	True

Consider:

$$\exists b \in B,\ \forall a \in A,\ Loves(a, b)$$

$$\exists b \in B,\ \underline{\left(\forall a \in A,\ Loves(a, b)\right)}$$

“There exists a person b in B who is loved by every person a in A.”

MULTIPLE QUANTIFIERS, DIFFERENT TYPE (2)

$$\exists b \in B,\ \forall a \in A,\ Loves(a, b)$$

"There exists a person b in B who is loved by every person a in A."

	Sophia	Thel	Stanley	Laura
Breanna	False	True	True	False
Malena	False	True	True	True
Patrick	False	False	True	False
Ella	False	False	True	True

	Sophia	Thel	Stanley	Laura
Breanna	False	True	**True**	False
Malena	False	True	**True**	True
Patrick	False	False	**True**	False
Ella	False	False	**True**	True

Key idea: Because the $\exists b$ is outside of the $\forall a$, the "b" person must be the same for each "a" person.

A SLIGHTLY DIFFERENT "LOVES" TABLE

	Sophia	**Thelonious**	**Stanley**	**Laura**
Breanna	False	True	True	False
Malena	False	True	True	True
Patrick	False	False	True	False
Ella	False	False	False	True

Now,

- $\forall a \in A,\ \exists b \in B,\ Loves(a, b)$ is **True**
- $\exists b \in B,\ \forall a \in A,\ Loves(a, b)$ is **False**

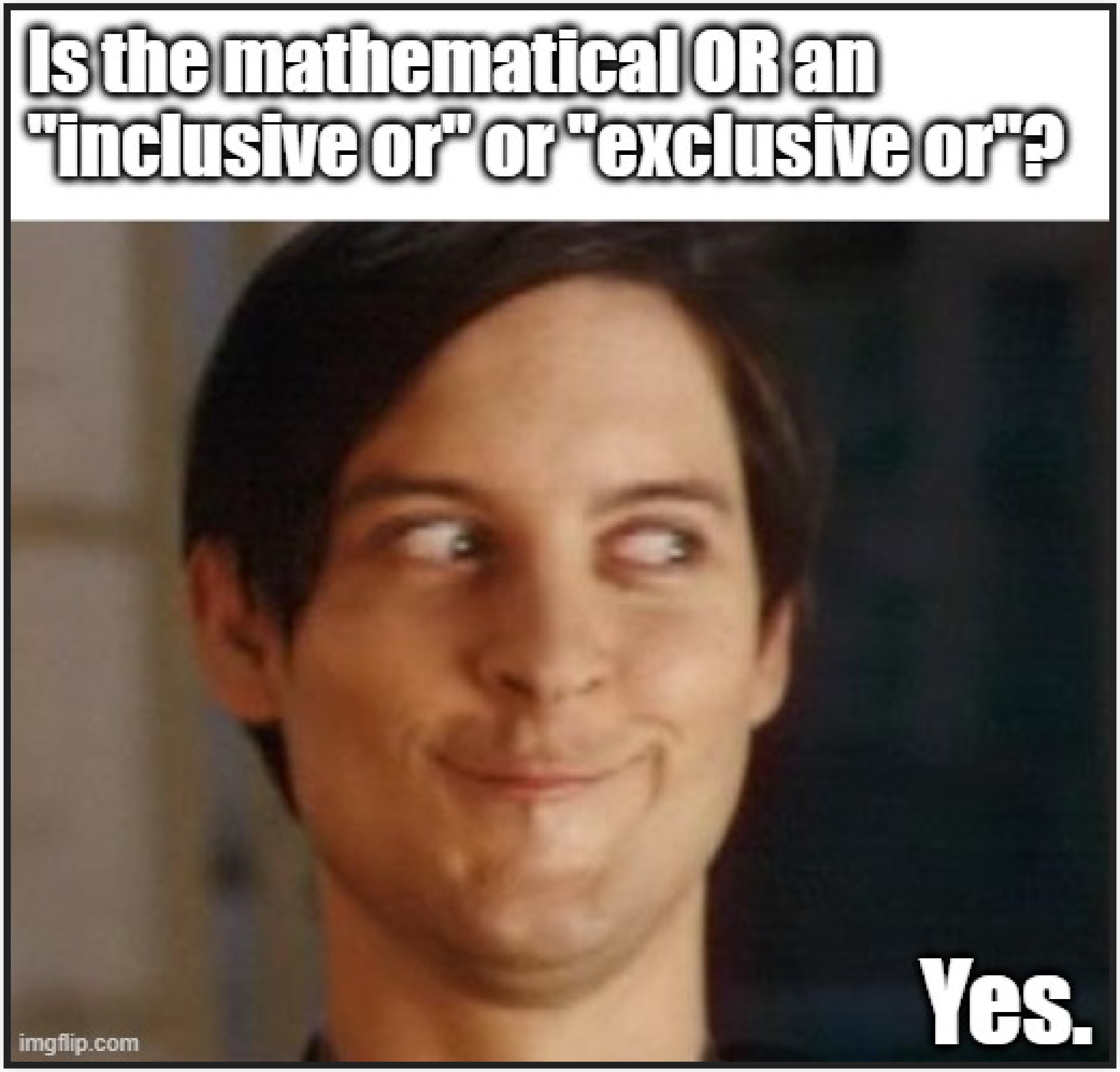
SUMMARY

TODAY YOU LEARNED TO...

1. Express boolean expressions using formal logical notation, including propositional operators and two quantifiers
2. Translate boolean expressions between English, mathematical notation, and Python expressions
3. Perform filtering operations on Python collections
4. Interpret statements in predicate logic with multiple quantifiers, and express them in Python

HOMEWORK

- Readings:
  - Review: Chapters 1 & 2
  - From today: 3.1, 3.2, 3.3, 3.7
  - Next class: 3.4, 3.5, 3.6
- Continue working on Assignment 1!
  - Check FAQ (+ corrections)
  - Visit Additional TA Office Hours
  - Post a question on Campuswire


Is the mathematical OR an "inclusive or" or "exclusive or"?
Yes.
imgflip.com