CSCI 2200

Foundations of Computer Science

Lecture 3: Propositional and Predicate Logic

CSCI 2200
Foundations of Computer Science

Lecture 3: Propositional and Predicate Logic

"Secret Course" from *Super Mario Sunshine*, Koji Kondo

Plan for today

Questions from HW

Mathematical / logical statements: propositional logic

Adding functions and variables: predicate logic

HW questions?

Propositional Logic

Propositions

- The most basic type of logical statement is a *proposition*. This is an unambiguous sentence that can be assigned a *truth value*: True or False.
    - There are exactly 137 people in this room.
    - Dan's car is silver.
    - Leicester City has won a Premier League trophy.
- Note: We might not *know* whether or not it's true or false. But it must be one or the other; there is no third option. This is known as the Law of the Excluded Middle.

Operations with logical statements

- We'll use lowercase letters (often p & q) to represent individual propositions.

  p: "Dan is taking Web Systems Development."
  q: "Dan is taking Database Systems."

- The primary logical operations are:
  - NOT (¬, !, ~, –)
    - ¬p = "Dan is not taking Web."
  - AND (∧, &, &&, *, ·)
    - p ∧ q = "Dan is taking Web and Dan is taking DB."
  - OR (∨, |, ||, +)
    - p ∨ q = "Dan is taking Web or Dan is taking DB."

Inclusive vs. exclusive OR

p: “Dan is taking Web Systems Development.”
q: “Dan is taking Database Systems.”

- OR is inclusive – it allows for the possibility of both halves being true.
  - $p \vee q$ = “Dan is taking only Web, DB, or both.”
- XOR is exclusive – it is False when both halves are true.
  - $p \oplus q$ = “Dan is taking Web without DB, or Dan is taking DB without Web.”

Quick practice

p: “Dan is taking Web Systems Development.”
q: “Dan is taking Database Systems.”

- What do the following statements say?
  - $p \vee \neg q$
  - $\neg p \wedge \neg q$
  - $\neg (p \wedge q)$

Quick practice

p: “Dan is taking Web Systems Development.”
q: “Dan is taking Database Systems.”

- What do the following statements say?
  - $p \vee \neg q$
    - Dan is taking Web or he is not taking DB.
  - $\neg p \wedge \neg q$
    - Dan is not taking Web and he is not taking DB.
  - $\neg (p \wedge q)$
    - Dan is not taking both Web and DB together.

DeMorgan's Laws (redux)

- Once again, to distribute a NOT through an AND or an OR:
  - First, negate each piece of the compound statement.
  - Then swap AND for OR and vice versa.
- $\neg (p \wedge q) = \neg p \vee \neg q$ = NOT BOTH
- $\neg (p \vee q) = \neg p \wedge \neg q$ = NEITHER

DeMorgan's Laws (redux)

p: "Dan is taking Web Systems Development."
q: "Dan is taking Database Systems."

- $\neg (p \wedge q) = \neg p \vee \neg q$ = <u>NOT BOTH</u>
  - Dan is not taking both Web and DB together.
  - Dan is not taking Web <u>*OR*</u> Dan is not taking DB.
- $\neg (p \vee q) = \neg p \wedge \neg q$ = <u>NEITHER</u>
  - Dan is taking neither Web nor DB.
  - Dan is not taking Web <u>*AND*</u> Dan is not taking DB.

Implications

- "If … then" statements are critical in both natural and computer languages, and also in propositional logic. Consider:
  - "If you go to every class session, then you will pass the course."
  - How can we evaluate the truth of this statement? What information do we need?
  - How many individual propositions are contained in that statement?

Implications, cont'd.

- "If you go to every class session, then you will pass the course."

  p = "You go to every class session."
  q = "You will pass the course."

  We represent the full statement as $p \Rightarrow q$
  ("if p, then q" or "p implies q")

Implications, cont'd.

- "If you go to every class session, then you will pass the course."

  p = "You go to every class session."
  q = "You will pass the course."

- When is this false?
  - When you miss class?
  - When you pass the course?
  - When you fail the course?

Implications, cont'd.

- "If you go to every class session, then you will pass the course."

  p = "You go to every class session."
  q = "You will pass the course."

- The only way an implication becomes **false** is when the first part (the *antecedent*) is true and the second part (the *consequence*) is false.
  - Here, that would mean you go to every class session and still fail. If I'd put the top sentence in the syllabus and that happened, you'd have grounds to be mad.
- Thus, in all other situation, the implication is TRUE.

More implications

- If it is raining, then it is cloudy.
  - Suppose it <u>is</u> cloudy. Do you learn anything else?
  - Under what circumstances do you gain information?
- "If <u>you</u> are a Scottish lord, then <u>I</u> am Mickey Mouse!"
  - In this quote from *Indiana Jones and the Last Crusade*, what is the butler trying to convey?

- As a general note, implications in propositional logic feel a bit… *weird*. They make much more sense once we are talking about quantifiers (in a few slides).

Reversing implications

- HUGELY IMPORTANT: $p \Rightarrow q$ and $q \Rightarrow p$ are <u>not the same thing!!</u>
- "If you go to every class session, then you will pass the course."
  - This does not mean that everyone who passes has gone to every class session!
- … but what about everyone who fails the class? Can we conclude anything about them?

The contrapositive

- $p \Rightarrow q$: "If you go to every class session, then you will pass the course."
- If the above is true, then if someone fails the course, they must not have gone to every class.
- Symbolically $p \Rightarrow q$ is *logically equivalent* to $\neg q \Rightarrow \neg p$; anytime one is true, the other is true and vice versa.

Rules of inference (syllogisms)

- Given an implication $p \Rightarrow q$, if you also know…
  - p, then you may conclude q. This rule is called *modus ponens*.
    - "If you go to every class session, you will pass the course."
      "You have gone to every class session."
      Therefore: "You pass the course."
  - $\neg q$, then you may conclude $\neg p$. This is called *modus tollens*.
    - "If you are a Scottish lord, then I am Mickey Mouse!"
      "I am not Mickey Mouse."
      Therefore: "You are not a Scottish lord."

Truth tables

- Truth tables are an exhaustive approach to breaking down compound logical statements into the basic propositions. For example:

p	q	$p \wedge q$
T	T	**T**
T	F	**F**
F	T	**F**
F	F	**F**

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

Quick practice

Make truth tables for the following:

- $\neg p \vee q$
- $\neg p \Rightarrow q$
- $(p \wedge \neg q) \vee r$ (note: 8 rows needed)
- $(\neg p \vee q) \Rightarrow p$ (note: you may want an extra column)

Quick practice

Make truth tables for the following:

- $\neg p \vee q$

p	q	$\neg p \vee q$
T	T	**T**
T	F	**F**
F	T	**T**
F	F	**T**

- $(\neg p) \Rightarrow q$

p	q	$(\neg p) \Rightarrow q$
T	T	**T**
T	F	**T**
F	T	**T**
F	F	**F**

Quick practice

Make truth tables for the following:

- (p ∧ ¬ q) ∨ r

p	q	r	(p ∧ ¬ q) ∨ r
T	T	T	**T**
T	T	F	**F**
T	F	T	**T**
T	F	F	**T**
F	T	T	**T**
F	T	F	**F**
F	F	T	**T**
F	F	F	**F**

Quick practice

Make truth tables for the following:

- $(\neg p \vee q) \Rightarrow p$ (note: you may want an extra column)

p	q	$\neg p \vee q$	$(\neg p \vee q) \Rightarrow p$
T	T	T	**T**
T	F	F	**T**
F	T	T	**F**
F	F	T	**F**

Predicate Logic

Dealing with large sets in logic

- "Every student in section 3 is a computer science major."

  Could that be a proposition?

- Sure.

- But it's a bit… unwieldy. There's no granularity, no way to put this in terms of individual students unless we do something like:

  p_1 = "Kai is a CS major."
  p_2 = "Carina is a CS major."
  … p_{29} = "Kelton is a CS major."

  And then do $p_1 \wedge p_2 \wedge p_3 \wedge \ldots \wedge p_{29}$

Too much writing…

- It would be nice to not need to lay out all of these propositions individually.

- As programmers, we try to automate repetitive tasks. How might we handle this sort of thing in Python?

Python Break!

You have a string S. You want to check if that string represents a number; that is, you need to ensure every character is a digit.

```
if S[0] >= '0' and S[0] <= '9' and
   S[1] >= '0' and S[1] <= '9' and
   S[2] >= '0' and S[2] <= '9' and
   ...
```

Of course not! You would use a variable, a binary function, and a loop!

```
for c in S:
   if !isdigit(c):
      return False
return True
```

Predicates

- A predicate is a Boolean function whose input(s) are elements of U (the universe of discourse).
- C(x) = x is a computer science major.

$$p_1 = C(\text{Kwasi})$$
$$p_2 = C(\text{Barbara})$$
$$\ldots\ p_{64} = C(\text{Demetrios})$$

- Well, it's shorter. But we still need some way to "loop" to really improve things…

The universal quantifier

- Let's go back to the English: "Every student in section 3 is a computer science major."
- What if we create S_3 = {students in section 3}?
- At that point, in Python, we could have:

```
p = True
for s in S3:  // checks every element of S3
    p = p and C(s)
```

- The *universal quantifier* allows us to do this in predicate logic:
  - $\forall s \in S_3, C(s)$ – This reads as "*for all* s in S_3…"
  - Implicitly, this is an "and" operation.

Questions of existence…

- q = "One among us is an Impostor."
- Without quantifiers:
  q_1 = "Green is an Impostor."
  …
  q_8 = "Cyan is an Impostor."

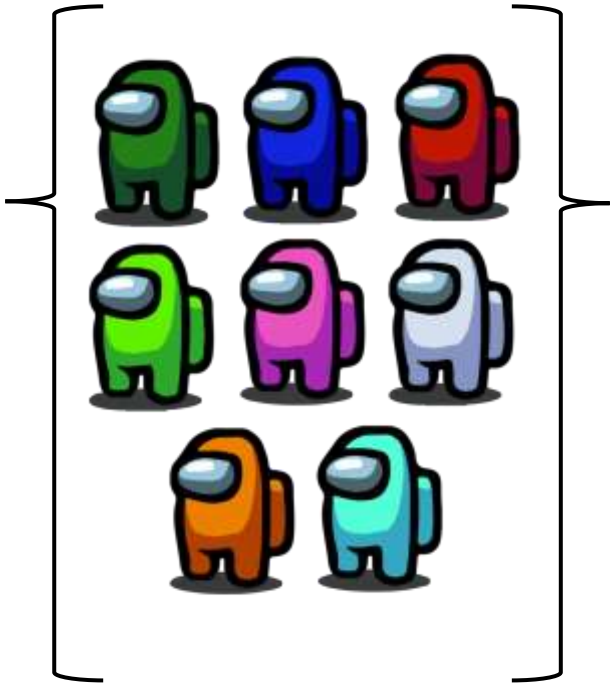
  $$q = q_1 \vee q_2 \vee q_3 \vee \ldots \vee q_8$$

  - *(Note the* **or** *operation here…)*

The existential quantifier

- q = "One among us is an Impostor."
- We can instead define a predicate…
  - Imp(x) = "x is an Impostor."
- … and then use the *existential quantifier:*

C =



$$\exists c \in C, Imp(c)$$

- Read this as "There exists some c in C such that c is an Impostor."

Quantified compound statements

- "Someone in this class is a CS/math double major."
- You *could* make a single predicate to cover this case, but it is usually better to break things down:
  C(x) = "x is a CS major."
  M(x) = "x is a math major."
- $\exists s, (C(s) \wedge M(s))$
  - Notice that we have omitted the set we are drawing s from. For brevity, we will often do this when it is obvious; however, it is never wrong to specify it.

Negating quantifiers

- “There is no such thing as a flying horse.”
- Given H(x) = “x is a horse” and F(x) is “x can fly”, how would you write this?

Negating quantifiers

- “There is no such thing as a flying horse.”
- Given H(x) = “x is a horse” and F(x) is “x can fly”, how would you write this?

$$\neg\exists a\ (H(a) \wedge F(a))$$

- But notice, we could have equivalently said: “No animal is both a horse and flying.”

$$\forall a\ \neg(H(a) \wedge F(a))$$

- … look familiar?

DeMorgan's Laws (yet again!)

- To negate a quantifier, swap existential for universal and vice versa, then move the negation inside.
- $\neg\forall(x)P(x) \equiv \exists(x)\neg P(x)$
  - "not all" is logically equivalent to "there exists one that isn't"
- $\neg\exists(x)P(x) \equiv \forall(x)\neg P(x)$
  - "there does not exist" is logically equivalent to "every one is <u>not</u>"

Multiple variables and mixing quantifiers

- Predicates (like any Boolean function) can have multiple inputs:
  - R(s,c) = "Student s is required to take Course c."
- Consider the following two statements. What do you think they should mean? Which of them is true?

$$\exists c\ \forall s\ R(s,c)$$

$$\forall s\ \exists c\ R(s,c)$$

Multiple variables and mixing quantifiers

- Predicates (like any Boolean function) can have multiple inputs:
  - R(s,c) = "Student s is required to take Course c."
- Consider the following two statements. What do you think they should mean? Which of them is true?

$\exists c\ \forall s\ R(s,c)$ - There is a specific course that every student is required to take. (False, I think.)

$\forall s\ \exists c\ R(s,c)$ – Every student has some course (can be different for each student) that they are required to take. (True.)

Quick practice

C(s) = “s is a computer science major.”
M(s) = “s is a math major.”
Ph(c) = “c is a philosophy course.”
R(s,c) = “s is required to take c.”

Write the following English sentences in predicate logic.

“No students are CS/math double majors.”

“There is at least one course that every CS major and every math major must take.”

“No CS major has a required philosophy course.”

“Every math major must take some philosophy course.”

“There is a specific philosophy course that every math major must take.

Today's class survey

Please complete the class survey now.

Recitations on Wednesday – you are encouraged to attend.

Submissions for Problem Set 1 due Thursday.