LOGIC

OUTLINE:

(1) Introduction to logic
(2) Formalization of logic
(3) Propositional logic
(4) Predicate logic
(5) Proof techniques

4. PREDICATE LOGIC

Motivation

- Consider again the sentence “x is a prime number” (seen in Logic1, slide 35), with the understanding that ‘x’ is a just a placeholder.
- This is a declarative sentence, but NOT a proposition, because its truth value varies along with x.
- However, the variability of x is the only obstacle: once x gets fixed as a specific object of the universe, the truth value becomes univocally and unambiguously determined.
- On the other hand, if we want to apply logic to mathematics, we would like (and need) to be able to deal with such sentences.

Predicates

- "x is a prime number" is an instance of a predicate.
- Predicates express properties of objects (unary predicates: require 1 input) or relations between objects (binary, ternary, ... predicates: require more than 1 input).
- Predicates are usually denoted with uppercase letters, followed by their input labels in brackets.
- EX: $M(x,y)$ = "x is the mother of y" is a binary predicate. $Q(x,y,z)$ = "x and y are the parents of z" is a ternary predicate.
- It is important to highlight that here x,y,z are just placeholders (labels), like the local variables that you pass as parameters when you define a method in a programming language.

Turning predicates into propositions

There are 2 ways of turning a predicate into a proposition:

1) Evaluating the predicate, that is, replacing each variable with a constant value (object of the universe).
   - EX: if $P(x)$ = "x is a prime number", then $P(4)$ is the proposition "*4* is a prime number".
   - EX: if $M(x,y)$ = "x is the mother of y" then *M(Idril,Eärendil)* is the proposition "Idril is the mother of Eärendil".

2) Quantifying the predicate, that is, asserting that a predicate is true for some, or for all, objects of the universe.
   - EX: if $P(x)$ = "x is a prime number", then *for some x*, $P(x)$ is the proposition "some objects of the universe are prime numbers", or equivalently, "there is an object of the universe which is a prime number", or also "prime numbers exist".
   - EX: if $M(x,y)$ = "x is the mother of y" then for all y, there is x such that $M(x,y)$ is the proposition "for any object y of the universe, there is an object x such that x is the mother of y", or simply "every object has a mother".

Turning predicates into propositions

- Analogy with object-oriented programming languages: defining a predicate is like writing a boolean (non-main) method inside a class:
- The non-main boolean methods exist in an abstract world, but are not concretely executable. Once you call them on an object, they return either 0 or 1.
- In the same way, predicate exists, but not in the world of propositions. Once you evaluate them on concrete objects, they give rise to a proposition which is either true or false.

Turning predicates into propositions

- Analogy with object-oriented programming languages: defining a predicate is like writing a boolean (non-main) method inside a class:
- The non-main boolean methods exist in an abstract world, but are not concretely executable. Once you call them on an object, they return either 0 or 1.
- In the same way, predicate exists, but not in the world of propositions. Once you evaluate them on concrete objects, they give rise to a proposition which is either true or false.

Turning predicates into propositions

- EX: Let $P(x,y)$ denote the binary predicate "$x > y$". Then
- $P(2,2)$ is a false proposition: "$2 > 2$"
- $P(\pi,3)$ is a true proposition: "$\pi > 3$"

Turning predicates into propositions

- EX: Let *P(x,y)* denote the binary predicate "*x* comes after *y*". We can use it in every situation where it makes sense, that is, wherever there is a concept of "coming after", that is, wherever there is an ordering of objects:
- *P(2,2)* is a false proposition: "2 comes after 2" [implied: in the usual ordering of integers]
- *P(π,3)* is a true proposition: "π comes after 3" [implied: in the usual ordering of real numbers]
- *P(Plato, Socrates)* is a true proposition: "Plato comes after Socrates" [implied: in history]
- *P('d','g')* is a false proposition: "[the char] 'd' comes after [the char] 'g'" [implied: e.g. in the English alphabet, or in the ASCII list of chars]

The alphabet

The alphabet of predicate logic is made of

- Variable symbols (denoted with x,y,z, possibly with subscripts), representing variable elements of the universe (thus serving as placeholder variables for predicates)
- Constant symbols (denoted with c,d,e, possibly with subscripts), representing specific and fixed elements of the universe
- Predicate symbols (denoted P,Q,R, possibly with subscripts), representing predicates, each with a prescribed arity (number of input arguments)
- The existential quantifier $\exists$, representing the locutions “for some”, “there is at least one”, etc.
- The universal quantifier $\forall$, representing the locutions “for all”, “any”, “every”, etc.
- The connectives
- The grouping symbols ‘(‘ and ‘)’

Where did propositions go?

- Predicate logic is an extension of propositional logic (i.e., it “contains” propositional logic)
- Then, where are propositions?
- Propositions *can be thought of* as the predicates of arity 0, that is, predicates not depending on input variables.

Precedence

- Quantifiers take precedence over all connectives
- EX: $\forall x\, P(x) \wedge Q$ means $(\forall x\, P(x)) \wedge Q$

  [note the presence of a predicate Q of arity 0, that is a proposition]

Translations from English into logic

- All men are mortal. Socrates is a man. THEREFORE Socrates is mortal.

Translations from English into logic

- All men are mortal. Socrates is a man. THEREFORE Socrates is mortal.
- This is an argument, but we can interpret it as a proposition in the following way:
- we have 3 atomic propositions: p = "All men are mortal." q = "Socrates is a man." r = "Socrates is mortal."
- They are connected via a conjunction and a conditional:

All men are mortal $\wedge$ Socrates is a man $\rightarrow$ Socrates is mortal.

Propositional logic gets us to $p \wedge q \rightarrow r$

Translations from English into logic

- All men are mortal. Socrates is a man. THEREFORE Socrates is mortal.
- With predicate logic, we can translate this sentence into a richer logical expression, in which we can "zoom in" the 3 atoms p,q,r and highlight their internal structure.
- Define 2 predicates: $H(x)$ = "x is a man" and $M(x)$ = "x is mortal".
  Define a constant s = "Socrates".
- p = "all men are mortal" is rendered as $\forall x\ (H(x) \rightarrow M(x))$ [literally: "for any x, if x is a man, then x is mortal"]

  q = "Socrates is a man" is rendered as $H(s)$

  r = Socrates is mortal" is rendered as $M(s)$
- "All men are mortal, Socrates is a man therefore Socrates is mortal" is rendered as

$$(\forall x\ (H(x) \rightarrow M(x))) \wedge H(s) \rightarrow M(s)$$

Translations from English into logic

- All men are mortal. Socrates is a man. THEREFORE Socrates is mortal.
- We are also allowed to use predicate and constant names which are more meaningful (as in programming language etiquette)
- Define 2 predicates: *Human(x)* = “*x* is a man” and *Mortal(x)* = “*x* is mortal”. Define a constant *Socrates* = “Socrates”.
- p = “all men are mortal” is rendered as $\forall x\, (Human(x) \rightarrow Mortal(x))$
  q = “Socrates is a man” is rendered as $Human(Socrates)$
  r = Socrates is mortal” is rendered as $Mortal(Socrates)$
- “All men are mortal, Socrates is a man therefore Socrates is mortal” is rendered as

$$(\forall x\, (Human(x) \rightarrow Mortal(x))) \wedge Human(Socrates) \rightarrow Mortal(Socrates)$$

More translations

- "Every real number has an opposite"
- Define 2 predicates: $R(x)$ = "x is a real number" and $O(x)$ = "x has an opposite"

  The sentence can be rendered as $\forall x\ (R(x) \rightarrow O(x))$
- This translation is completely correct, but it does not do much justice to the underlying mathematics (it leaves everything implicit, and so it is not the most useful form in which we can render the sentence)

More translations

- “Every real number has an opposite”
- ALTERNATIVE TRANSLATION (using mathematical symbols and making the notion of opposite explicit):
- Define 2 predicates: $R(x)$ = “$x \in \boldsymbol{R}$” [where $\boldsymbol{R}$ denotes the set of real numbers], and $S(x,y,z)$ = “$x + y = z$”
  Define the constant 0
- The sentence can be rendered as

$$\forall x\ (R(x) \rightarrow \exists y\ (R(y) \wedge S(x,y,0)))$$

- Or also, writing the predicates explicitly in mathematical notation)

$$\forall x\ (x \in \boldsymbol{R} \rightarrow \exists y\ (y \in \boldsymbol{R} \wedge (x + y = 0)))$$

More translations

- “Some students of CS2214 are also taking CS2209”
- Define 2 predicates: $S(x)$ = “x is a student”, and $P(x,y)$ = “x is enrolled in y” [Note that, in order for P to make sense, x and y must have inherently different types]
  Define 2 constants: CS2214, and CS2209
- The sentence can be rendered as

$$\exists x\ (S(x) \land P(x,CS2214) \land P(x,CS2209))$$

More translations

- “Some students of CS2214 are also taking CS2209”
- Define 2 predicates: $S(x)$ = “x is a student”, and $P(x,y)$ = “x is enrolled in y” [Note that, in order for P to make sense, x and y must have inherently different types]
  Define 2 constants: CS2214, and CS2209
- Can the sentence be rendered also as

$$\exists x\ (S(x) \rightarrow P(x,CS2214) \wedge P(x,CS2209))\ \ ?$$

More translations

- "Some students of CS2214 are also taking CS2209, some are not"
- Define 2 predicates: $S(x)$ = "x is a student", and $P(x,y)$ = "x is enrolled in y" [Note that, in order for P to make sense, x and y must have inherently different types]
  Define 2 constants: CS2214, and CS2209
- The sentence can be rendered as

$$\exists x\, (S(x) \land P(x,CS2214) \land P(x,CS2209)) \land$$
$$\exists x\, (S(x) \land P(x,CS2214) \land \neg P(x,CS2209))$$

More translations

- "The enemy of my enemy is my friend"
- Define 2 predicates: $E(x,y)$ = "x is an enemy of y", and $F(x,y)$ = "x is a friend of y"
  Define the constant Me
- The sentence tacitly implies two universal quantifiers: it means that any enemy of any enemy of mine is my friend, therefore it can be rendered as

$$\forall x\ \forall y\ (E(x,y) \wedge E(y,Me) \rightarrow F(x,Me))$$

Predicate logic: Semantics

- The semantics of predicate logic is quite complicated. Here we are going to overview it without delving in all the details
- An interpretation of predicate logic is the assignment of:
  - The domain (non-empty set of the objects “available for consideration”), in which variables vary, constants are chosen, and predicates are evaluated.
  - For each constant symbol, an object of the domain.
  - (Variable symbols are set to vary over all the objects of the domain.)
  - For each predicate symbol of arity 0 (i.e., a proposition), a truth value, coherent with the interpretation of the connectives (e.g., if an atom *a* is assigned the value 0, then all propositions $a \rightarrow b$ must be assigned the value 1)
  - For each unary predicate symbol, a property of objects, which makes sense in the domain and is in general true for some of the objects of the domain and false for the others
  - For each predicate symbols with arity > 1, a relation between objects, which makes sense in the domain and is in general true for some tuples of objects and false for the other tuples.

Predicative logical equivalences

- 2 predicate formulas *A* and *B* are logically equivalent (notation: $A \equiv B$) if they have the same truth value under any interpretation (that is, for any possible choice of the domain, of the values of constants and variables, and of the properties or relations associated with predicate symbols). [If instead *A* and *B* are NOT logically equivalent, the notation is $A \not\equiv B$].
- That's a lot of requirements!
- In addition, differently than in propositional logic, equivalence CANNOT be checked algorithmically (truth tables).
- We have to use reasoning to check logical equivalence.

Example 1

- Prove or disprove: $\neg\forall x\, P(x) \equiv \exists x\, \neg P(x)$

Example 1

- Prove or disprove: $\neg\forall x\, P(x) \equiv \exists x\, \neg P(x)$
- Let us fix an arbitrary interpretation.

(1) The LHS $\neg\forall x\, P(x)$ is true in the given interpretation iff not all objects of the chosen domain have the property assigned to the predicate symbol P.

(2) The RHS $\exists x\, \neg P(x)$ is true in that same interpretation iff there is an object of the chosen domain not having the property assigned to the predicate symbol P.

- Notice that, no matter what we choose as domain or what we choose as property associated with P, (1) and (2) are two ways of saying the same thing.
- Therefore $\neg\forall x\, P(x) \equiv \exists x\, \neg P(x)$

Example 1

- $\neg \forall x\, P(x) \equiv \exists x\, \neg P(x)$
- Notice that we have already discussed one possible interpretation of these 2 formulas in Logic1.pdf, Slides 43-45:
- Let the domain be the set made of you students of CS2214, and let $P(x)$ be interpreted as "x shall pass the course".
- $\neg \forall x\, P(x)$ can be interpreted as "it is not the case that all students of CS2214 shall pass the course"
- $\exists x\, \neg P(x)$ can be interpreted as "at least one student of CS2214 shall not pass the course"
- https://www.youtube.com/watch?v=3xYXUeSmb-Y

Example 2

- Prove or disprove: $\forall x\ \exists y\ P(x,y) \equiv \exists y\ \forall x\ P(x,y)$

Example 2

- Prove or disprove: $\forall x \, \exists y \, P(x,y) \equiv \exists y \, \forall x \, P(x,y)$
- Let us fix the following interpretation:
  - Domain: the natural numbers.
  - Interpretation of $P(x,y)$: "$x < y$".

(1) The LHS $\forall x \, \exists y \, P(x,y)$ says that for any natural number (x) we can find a bigger natural number (y). This is a true fact, asserting that natural numbers are unbounded from above.

(2) The RHS $\exists y \, \forall x \, P(x,y)$ says that there is a natural number greater than any other. This is false for the same reason that makes the previous statement true: natural numbers are unbounded from above.

- We found ONE interpretation in which the formulas have different truth values.

  This is enough to conclude that $\forall x \, \exists y \, P(x,y) \not\equiv \exists y \, \forall x \, P(x,y)$

Notable (predicative) logical equivalences

In the following, P(...) and Q(...) denote predicates of arbitrary arity (possibly even 0)

- *Double quantifier laws:* $\forall x\ \forall y\ P(\ldots) \equiv \forall y\ \forall x\ P(\ldots)$;

  $\exists x\ \exists y\ P(\ldots) \equiv \exists y\ \exists x\ P(\ldots)$
- *Distributive laws:* $\forall x\ (P(\ldots) \wedge Q(\ldots)) \equiv (\forall x\ P(\ldots) \wedge \forall x\ Q(\ldots))$;

  $\exists x\ (P(\ldots) \vee Q(\ldots)) \equiv (\exists x\ P(\ldots) \vee \exists x\ Q(\ldots))$
- *De Morgan's laws:* $\neg\forall x\ P(\ldots) \equiv \exists x\ \neg P(\ldots)$;

  $\neg\exists x\ P(\ldots) \equiv \forall x\ \neg P(\ldots)$

Non-logical equivalences

- *Mixed quantifiers:* $\forall x\ \exists y\ P(\ldots) \not\equiv \exists y\ \forall x\ P(\ldots)$
- *False distributive laws:* $\forall x\ (P(\ldots) \vee Q(\ldots)) \not\equiv (\forall x\ P(\ldots) \vee \forall x\ Q(\ldots))$;

$$\exists x\ (P(\ldots) \wedge Q(\ldots)) \not\equiv (\exists x\ P(\ldots) \wedge \exists x\ Q(\ldots));$$

$$\forall x\ (P(\ldots) \rightarrow Q(\ldots)) \not\equiv (\forall x\ P(\ldots) \rightarrow \forall x\ Q(\ldots));$$

$$\exists x\ (P(\ldots) \rightarrow Q(\ldots)) \not\equiv (\exists x\ P(\ldots) \rightarrow \exists x\ Q(\ldots))$$

Mega exercise

- Prove each of the previously stated notable logical equivalences. Find counterexamples which disprove the non-logical equivalences.

Transitivity of equivalence

- Logical equivalence is still transitive: if $A_1 \equiv A_2$, $A_2 \equiv A_3, \ldots, A_{n-1} \equiv A_n$, *then* $A_1 \equiv A_n$.
- EX:

$$
\begin{aligned}
\neg \forall x \forall y \left(P(x,y) \wedge \neg Q(x)\right) &\equiv \exists x \neg \forall y \left(P(x,y) \wedge \neg Q(x)\right) && \text{(DE MORGAN)} \\
&\equiv \exists x \exists y \neg \left(P(x,y) \wedge \neg Q(x)\right) && \text{(DE MORGAN)} \\
&\equiv \exists y \exists x \neg \left(P(x,y) \wedge \neg Q(x)\right) && \text{(DOUBLE QUANTIFIER)} \\
&\equiv \exists y \exists x \left(\neg P(x,y) \vee \neg\neg Q(x)\right) && \text{([PROP.] DE MORGAN)} \\
&\equiv \exists y \exists x \left(\neg P(x,y) \vee Q(x)\right) && \text{(DOUBLE NEGATION)} \\
&\equiv \exists y \left(\exists x \neg P(x,y) \vee \exists x\, Q(x)\right) && \text{(DISTRIBUTIVITY)}
\end{aligned}
$$

Ways of thinking of ∀ and ∃

- Fix an interpretation with finite domain $D=\{d_1,d_2,\ldots,d_n\}$.
- $\forall x\, P(x)$ has the same truth value of $P(d_1) \land P(d_2) \land \ldots \land P(d_n)$
- $\exists x\, P(x)$ has the same truth value of $P(d_1) \lor P(d_2) \lor \ldots \lor P(d_n)$
- Therefore, from a computer science perspective, we can check the truth values of quantified formulas by looping through the elements of the domain.
  - To check the truth value of $\forall x\, P(x)$:

```
set ∀x P(x) = True
for d in D:
    if P(d) == False:
        ∀x P(x) = False
        break
    if P(d) == True:
        continue
```

  - To check the truth value of $\exists x\, P(x)$:

```
set ∃x P(x) = False
for d in D:
    if P(d) == True:
        ∃x P(x) = True
        break
    if P(d) == False:
        continue
```

- Everything also holds for infinite domains, but in that case infinite conjunctions and disjunctions are required, and loops may not terminate.