# COSC 290 Discrete Structures

## Lecture 6: Predicate Logic

Prof. Michael Hay

Monday, Feb. 5, 2018

Colgate University

# Plan for today

1. Normal forms: CNF and DNF
2. Predicate Logic
3. Quantification of variables
4. Expressing statements in predicate logic

# Normal forms: CNF and DNF

# Literal

**Definition (Literal)**

A literal is an atomic proposition or the negation of an atomic proposition (i.e. it's either $p$ or $\neg p$ for some variable $p$).

**Example**

Let $p :=$ "Alice earns an A." and $q :=$ "Pigs can fly."

Literals: $p$, $\neg p$, $q$, $\neg q$.

Not literals: $p \vee q$, $q \wedge \neg p$, etc.

# Conjunctive Normal Form

**Definition (CNF)**

A proposition is in conjunctive normal form (CNF) if it consists of:

- a single *clause*, or
- a conjunction of two or more *clauses*

where a clause is

- a single *literal*, or
- a disjunction of two or more literals

# Conjunctive Normal Form

**Definition (CNF)**

A proposition is in conjunctive normal form (CNF) if it consists of:

- a single *clause*, or
- a conjunction of two or more *clauses*

where a clause is

- a single *literal*, or
- a disjunction of two or more literals

**Example**

These propositions are in CNF:

- $(p \lor q \lor s) \land (\neg p \lor r \lor \neg q)$
- $(\neg q \lor s)$
- $(\neg q \lor s) \land \neg q$

These propositions are *not* in CNF:

- $(p \lor q) \implies (\neg p \lor r)$
- $(\neg q \land s) \land (\neg p \lor r)$

# Disjunctive Normal Form

Disjunctive Normal Form (DNF) is the same idea, just swap the role of ANDs and ORs. See textbook for definition and examples.

Informally,

- conjunctive-normal form (CNF) is an “AND” of a bunch of “ORs”.
- disjunctive-normal form (DNF) is an “OR” of a bunch of “ANDs”.

# Poll: is this proposition in CNF?

Which of these propositions is *not* in CNF?

A) $\neg p$

B) $p \lor q$

C) $(p \lor q) \land (r \lor s)$

D) $(p \land q) \lor (r \land \neg p)$

E) More than one is *not* in CNF

(Definitions restated here for reference)

A proposition is in CNF if it is a single clause or the conjunction of two or more clauses where each clause is a single literal or the disjunction of two or more literals.

A literal is an atomic proposition or the negation of an atomic proposition (i.e. it's either $p$ or $\neg p$ for some variable $p$).

# Logical equivalence and CNF/DNF

Two important results:

1. Given a proposition $\psi$, it is possible to write a proposition $\varphi$ such that $\psi \equiv \varphi$ and $\varphi$ is in conjunctive-normal form (CNF).
2. Given a proposition $\psi$, it is possible to write a proposition $\varphi$ such that $\psi \equiv \varphi$ and $\varphi$ is in disjunctive-normal form (DNF).

Why might these results be useful?

# Checking a CNF sentence for tautology

If $\varphi$ is a proposition in CNF. Then checking for a tautology is easy.

- $\varphi$ is a tautology if and only if each clause is a tautology.
- A clause from a CNF is a tautology if and only if it contains a literal and its opposite.

# Poll: is this CNF a tautology?

Consider

$$\varphi := (p \vee q \vee \neg p) \wedge (r \vee q \vee p \vee \neg q) \wedge (\neg r \vee p)$$

Is $\varphi$ in CNF? Is $\varphi$ a tautology?

A) CNF: yes, tautology: yes

B) CNF: yes, tautology: no

C) CNF: no, tautology: yes

D) CNF: no, tautology: no

# Predicate Logic

# Predicate

An atomic proposition $p$ is a Boolean variable. It is either true or false.

A predicate $P(x)$ is a Boolean *function*. Its truth value depends on what arguments are passed in.

**Example**

- $isPrime(x)$ returns true if $x$ is a prime number and false otherwise.
- $isDivisibleBy(x, y)$ returns true if $x$ is evenly divisible by $y$.

# Propositions that include predicates

We can form a proposition by supplying arguments to each predicate.

$$\varphi \coloneqq \mathit{isPrime}(8) \vee \mathit{isDivisibleBy}(8, 2)$$

The truth of this proposition requires *interpreting* the predicates:

# Propositions that include predicates

We can form a proposition by supplying arguments to each predicate.

$$\varphi := \mathit{isPrime}(8) \lor \mathit{isDivisibleBy}(8, 2)$$

The truth of this proposition requires *interpreting* the predicates:

- $\mathit{isPrime}(8)$ is false, according to definition of *isPrime*
- $\mathit{isDivisibleBy}(8, 2)$ is true, according to definition of *isDivisibleBy*
- thus, $\varphi$ is true

# Understanding terminology: predicate vs. proposition

Consider the following two expressions:

$$\varphi := \mathit{isPrime}(8) \lor \mathit{isDivisibleBy}(8, 2)$$

and

$$\psi := \mathit{isPrime}(x) \lor \mathit{isDivisibleBy}(x, 2)$$

The first one, $\varphi$, is a proposition. Why?

The second one, $\psi$, is *not* a proposition. Why not?

# Free variables

The expression $\varphi$,

$$\psi := \mathit{isPrime}(x) \vee \mathit{isDivisibleBy}(x, 2)$$

is *not* a proposition because...

the truth value of $\psi$ depends on the free variable $x$.

Thus, $\psi$ is a *predicate*, not a proposition and we should write it like this:

$$\psi(x) := \mathit{isPrime}(x) \vee \mathit{isDivisibleBy}(x, 2)$$

# Quantification of variables

# Quantification

Let $S \coloneqq \{2, 3, 4\}$. The proposition $\varphi$,

$$\varphi \coloneqq \forall x \in S : \mathit{isDivisibleBy}(x, 2)$$

is equivalent to:

$$\mathit{isDivisibleBy}(2, 2) \wedge \mathit{isDivisibleBy}(3, 2) \wedge \mathit{isDivisibleBy}(4, 2)$$

# Quantification

Let $S \coloneqq \{ 2, 3, 4 \}$. The proposition $\varphi$,

$$\varphi \coloneqq \forall x \in S : \mathit{isDivisibleBy}(x, 2)$$

is equivalent to:

$$\mathit{isDivisibleBy}(2, 2) \wedge \mathit{isDivisibleBy}(3, 2) \wedge \mathit{isDivisibleBy}(4, 2)$$

Whereas the proposition $\psi$,

$$\psi \coloneqq \exists x \in S : \mathit{isDivisibleBy}(x, 2)$$

is equivalent to:

$$\mathit{isDivisibleBy}(2, 2) \vee \mathit{isDivisibleBy}(3, 2) \vee \mathit{isDivisibleBy}(4, 2)$$

# Quantification over set expressions

Let $S := \{ 2, 3, 4 \}$. The proposition $\varphi$,

$$\varphi := \forall x \in (S - \{ x \}) : \mathit{isDivisibleBy}(x, 2)$$

is equivalent to:

$$\mathit{isDivisibleBy}(2, 2) \wedge \mathit{isDivisibleBy}(4, 2)$$

# Bound vs. free variables

Contrast expressions $\psi$ and $\theta$

$$\psi := \exists x \in S : \mathit{isDivisibleBy}(x, 2)$$

with

$$\theta := \exists x \in S : \mathit{isDivisibleBy}(x, y)$$

While $\psi$ is a proposition, $\theta$ is *not* a proposition. Why?

# Bound vs. free variables

Contrast expressions $\psi$ and $\theta$

$$\psi := \exists x \in S : \mathit{isDivisibleBy}(x, 2)$$

with

$$\theta := \exists x \in S : \mathit{isDivisibleBy}(x, y)$$

While $\psi$ is a proposition, $\theta$ is *not* a proposition. Why?

In both, the variable $x$ is a bound variable. The "$\exists x \in S$" part *binds* variable $x$ to *each* element in $S$.

# Bound vs. free variables

Contrast expressions $\psi$ and $\theta$

$$\psi := \exists x \in S : \mathit{isDivisibleBy}(x, 2)$$

with

$$\theta := \exists x \in S : \mathit{isDivisibleBy}(x, y)$$

While $\psi$ is a proposition, $\theta$ is *not* a proposition. Why?

In both, the variable $x$ is a bound variable. The "$\exists x \in S$" part *binds* variable $x$ to *each* element in $S$.

In $\theta$, the variable $y$ is a free variable, thus $\theta$ is really a *predicate*, not a proposition.

$$\theta(y) := \exists x \in S : \mathit{isDivisibleBy}(x, y)$$

The predicate is true when $S$ contains something divisible by $y$.

# Poll: quantification and free/bound variables

Let $S := \{2, 3, 4\}$. Consider this expression $\varphi$,

$$\varphi := \forall x \in S : \mathit{isDivisibleBy}(x, 2) \vee \mathit{isDivisibleBy}(x, 3)$$

Which of the following statements is accurate?

A) $\varphi$ is a true proposition and $x$ is a bound variable.

B) $\varphi$ is a true proposition and $x$ is a free variable.

C) $\varphi$ is a false proposition and $x$ is a bound variable.

D) $\varphi$ is a false proposition and $x$ is a free variable.

E) $\varphi$ is *not* a proposition

# Expressing statements in predicate logic

# Universal Quantification

Let $P := \{ p_1, p_2, \ldots, \}$ be the (infinite) set of all persons.

$$\forall p \in P : At(p, Colgate) \implies BrushesTeeth(p)$$

means "Every person at Colgate brushes their teeth."

The above is *roughly* equivalent to

$$\begin{aligned} &(At(p_1, Colgate) \implies BrushesTeeth(p_1)) \\ \land\ &(At(p_2, Colgate) \implies BrushesTeeth(p_2)) \\ \land\ &(At(p_3, Colgate) \implies BrushesTeeth(p_3)) \\ \land\ &\ldots \end{aligned}$$

# Common mistake with universal quantification

Typically, $\implies$ is the main connective with $\forall$.

Common mistake: using $\wedge$ as the main connective with $\forall$:

$$\forall p \in P : At(p, Colgate) \wedge BrushesTeeth(p)$$

means "Every person is at Colgate and everyone brushes their teeth."

# Common mistake with universal quantification

Typically, $\implies$ is the main connective with $\forall$.

Common mistake: using $\land$ as the main connective with $\forall$:

$$\forall p \in P : At(p, Colgate) \land BrushesTeeth(p)$$

means "Every person is at Colgate and everyone brushes their teeth."

This statement is false as long as there is one person who does not attend Colgate.

# Existential Quantification

Let $P\,\{\, p_1, p_2, \ldots, \}$ be the (infinite) set of all persons.

$$\exists p \in P : At(p, Bucknell) \land BrushesTeeth(p)$$

means "Some person at Bucknell brushes their teeth."

The above is *roughly* equivalent to

$$\begin{aligned}
&(At(p_1, Bucknell) \land BrushesTeeth(p_1)) \\
\lor\ &(At(p_2, Bucknell) \land BrushesTeeth(p_2)) \\
\lor\ &(At(p_3, Bucknell) \land BrushesTeeth(p_3)) \\
\lor\ &\ldots
\end{aligned}$$

# Common mistake with existential quantification

Typically, $\wedge$ is the main connective with $\exists$.

Common mistake: using $\implies$ as the main connective with $\exists$:

$$\exists p \in P : At(p, Bucknell) \implies BrushesTeeth(p)$$

is true provided that there is some person who is *not* at Bucknell!

# Constructing Predicates

In programming, we can define functions that call other functions.

In predicate logic, we can define predicate in terms of other predicates.

Examples:

- $\mathit{follows}(x, y)$ means that $x$ follows the tweets of $y$
- $\mathit{TrumpFollower}(x) := \mathit{follows}(x, \text{@realDonaldTrump})$

# Constructing Predicates

In programming, we can define functions that call other functions.

In predicate logic, we can define predicate in terms of other predicates.

Examples:

- $\mathit{follows}(x, y)$ means that $x$ follows the tweets of $y$
- $\mathit{TrumpFollower}(x) \coloneqq \mathit{follows}(x, \text{@realDonaldTrump})$
- $\mathit{popularTweeter}(y) \coloneqq \forall x \in P : \mathit{follows}(x, y)$
- $\mathit{hasFollower}(y) \coloneqq \text{???}$

# Constructing Predicates

In programming, we can define functions that call other functions.

In predicate logic, we can define predicate in terms of other predicates.

Examples:

- $follows(x, y)$ means that $x$ follows the tweets of $y$
- $TrumpFollower(x) \coloneqq follows(x, \text{@realDonaldTrump})$
- $popularTweeter(y) \coloneqq \forall x \in P : follows(x, y)$
- $hasFollower(y) \coloneqq \exists x \in P : follows(x, y)$

# Constructing Predicates

In programming, we can define functions that call other functions.

In predicate logic, we can define predicate in terms of other predicates.

Examples:

- $\mathit{follows}(x, y)$ means that $x$ follows the tweets of $y$
- $\mathit{TrumpFollower}(x) \coloneqq \mathit{follows}(x, \text{@realDonaldTrump})$
- $\mathit{popularTweeter}(y) \coloneqq \forall x \in P : \mathit{follows}(x, y)$
- $\mathit{hasFollower}(y) \coloneqq \exists x \in P : \mathit{follows}(x, y)$
- $\mathit{followsEveryone}(x) \coloneqq ???$

# Constructing Predicates

In programming, we can define functions that call other functions.

In predicate logic, we can define predicate in terms of other predicates.

Examples:

- $\mathit{follows}(x, y)$ means that $x$ follows the tweets of $y$
- $\mathit{TrumpFollower}(x) \coloneqq \mathit{follows}(x, \text{@realDonaldTrump})$
- $\mathit{popularTweeter}(y) \coloneqq \forall x \in P : \mathit{follows}(x, y)$
- $\mathit{hasFollower}(y) \coloneqq \exists x \in P : \mathit{follows}(x, y)$
- $\mathit{followsEveryone}(x) \coloneqq \forall y \in P : \mathit{follows}(x, y)$

# Poll: fastest person

Let $P$ be the set of all people. Let $faster(x, y)$ be true if $x$ runs faster than $y$ and false otherwise.

Which of the following is the correct definition for $fastest(x)$?

A) $fastest(x) \coloneqq \exists y \in P : faster(x, y)$

B) $fastest(x) \coloneqq \forall y \in P : faster(x, y)$

C) $fastest(x) \coloneqq \forall y \in P - \{ x \} : faster(x, y)$

D) $fastest(x) \coloneqq \neg (\exists y \in P : faster(y, x))$

E) C and D

# Poll: fastest lacrosse player

As before, let $P$ be the set of all people. Let $faster(x, y)$ be true if $x$ runs faster than $y$ and false otherwise.

Let $lax(x)$ be true if $x$ plays lacrosse and false otherwise.

Which of the following is the correct definition for $fastestLax(x)$, the fastest lacrosse player?

A) $fastestLax(x) \coloneqq \forall y \in P - \{ x \} : lax(y) \wedge faster(x, y)$

B) $fastestLax(x) \coloneqq \forall y \in P - \{ x \} : lax(y) \implies faster(x, y)$

C) $fastestLax(x) \coloneqq lax(x) \wedge \forall y \in P - \{ x \} : faster(x, y)$

D) $fastestLax(x) \coloneqq lax(x) \wedge \forall y \in P - \{ x \} : lax(y) \wedge faster(x, y)$

E) $fastestLax(x) \coloneqq lax(x) \wedge \forall y \in P - \{ x \} : lax(y) \implies faster(x, y)$