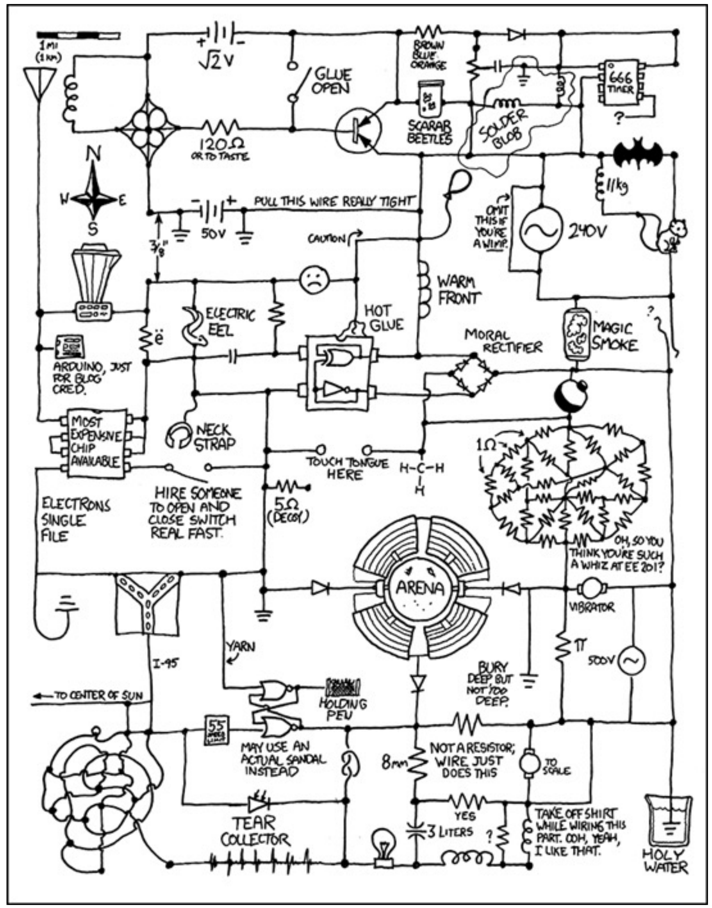
CSE 311: Foundations of Computing

Lecture 5: CNF and Predicate Logic



Recap from last class: DNF

- A propositional logic formula is in disjunctive normal form (DNF), if it is an OR of AND terms of literals (i.e. variables or negated variables)
- Example for DNF: $(q \land \neg r \land s) \lor (\neg q \land \neg r) \lor (\neg r \land \neg s)$
- Every propositional formula has an equivalent DNF

Conjunctive Normal Form

- A propositional logic formula is in conjunctive normal form (CNF), if it is an AND of OR terms of literals

  Example for CNF: $(q \vee \neg r \vee s) \wedge (\neg q \vee \neg r) \wedge (\neg r \vee \neg s)$

- Every propositional logic formula has an equivalent CNF. Again that CNF is not necessarily unique (but the full CNF is)

- Other names for CNF:
  - Product-of-Sums Canonical Form
  - Maxterm Expansion

Construction of Conjunctive Normal Form

④ AND the disjunctions

$$F \equiv (A \vee B \vee C) \wedge (A \vee \neg B \vee C) \wedge (\neg A \vee B \vee C)$$

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

① Read F rows off truth table	② Negate all bits	③ Convert to Prop. logic
000	111	$A \vee B \vee C$
010	101	$A \vee \neg B \vee C$
100	011	$\neg A \vee B \vee C$

F

CNF: Why does this procedure work?

Useful Facts:

- We know $F \equiv \neg(\neg F)$
- We know how to get a **DNF** for $\neg F$

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$\neg F \equiv (\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg C) \vee (A \wedge \neg B \wedge \neg C)$$

Taking the complement of both sides…

$$F \equiv \neg((\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg C) \vee (A \wedge \neg B \wedge \neg C))$$

And using DeMorgan/Comp….

$$F \equiv \neg(\neg A \wedge \neg B \wedge \neg C) \wedge \neg(\neg A \wedge B \wedge \neg C) \wedge \neg(A \wedge \neg B \wedge \neg C)$$

$$F \equiv (A \vee B \vee C) \wedge (A \vee \neg B \vee \neg C) \wedge (\neg A \vee B \vee C)$$

Predicate Logic

- **Propositional Logic**

  "If you take the high road and I take the low road then I'll arrive in Scotland before you."

- **Predicate Logic**

  "All positive integers x, y, and z satisfy $x^3 + y^3 \neq z^3$."

Predicate Logic

- Propositional Logic
  - Allows us to analyze complex propositions in terms of their simpler constituent parts (a.k.a. atomic propositions) joined by connectives

- Predicate Logic
  - Lets us analyze them at a deeper level by expressing how those propositions depend on the objects they are talking about

Predicate Logic

Adds two key notions to propositional logic

- Predicates
- Quantifiers

Predicates

Predicate

– **A function that returns a truth value, e.g.,**

Cat(x) ::= “x is a cat”

Prime(x) ::= “x is prime”

HasTaken(x, y) ::= “student x has taken course y”

LessThan(x, y) ::= “x < y”

Sum(x, y, z) ::= “x + y = z”

GreaterThan5(x) ::= “x > 5”

HasNChars(s, n) ::= “string s has length n”

Predicates can have varying numbers of arguments and input types.

Domain of Discourse

For ease of use, we define one "type"/"domain" that we work over. This set of objects is called the "**domain of discourse**".

For each of the following, what might the domain be?

(1) "x is a cat", "x barks", "x ruined my couch"

(2) "x is prime", "x = 0", "x < 0", "x is a power of two"

(3) "student x has taken course y" "x is a pre-req for z"

Domain of Discourse

For ease of use, we define one “type”/“domain” that we work over. This non-empty set of objects is called the “**domain of discourse**”.

For each of the following, what might the domain be?

(1) “x is a cat”, “x barks”, “x ruined my couch”

“mammals” or “sentient beings” or “cats and dogs” or …

(2) “x is prime”, “x = 0”, “x < 0”, “x is a power of two”

“numbers” or “integers” or “integers greater than 5” or …

(3) “student x has taken course y” “x is a pre-req for z”

“students and courses” or “university entities” or …

Quantifiers

We use *quantifiers* to talk about collections of objects.

$\forall x\, P(x)$

P(x) **is true for every** x **in the domain**

read as "for all x, P of x"

$\exists x\, P(x)$

There is an x **in the domain for which** P(x) **is true**

read as "there exists x, P of x"

Quantifiers

We use *quantifiers* to talk about collections of objects.

Universal Quantifier ("for all"): $\forall x\ P(x)$

P(x) **is true for every** x **in the domain**

read as "for all x, P of x"

Examples: **Are these true?**

- $\forall$x **Odd(x)**
- $\forall$x **LessThan4(x)**

Quantifiers

We use *quantifiers* to talk about collections of objects.

Universal Quantifier ("for all"): ∀x P(x)

P(x) **is true for every** x **in the domain**

read as "for all x, P of x"

Examples: **Are these true? It depends on the domain. For example:**

	{1, 3, -1, -27}	**Integers**	**Odd Integers**
• ∀x **Odd(x)**	True	False	True
• ∀x **LessThan4(x)**	True	False	False

Quantifiers

We use ***quantifiers*** to talk about collections of objects.

Existential Quantifier ("exists"): $\exists x\ P(x)$

There is an x in the domain for which P(x) is true

read as "there exists x, P of x"

Examples:

Are these true?

- $\exists x$ Odd(x)
- $\exists x$ LessThan4(x)

Quantifiers

We use ***quantifiers*** to talk about collections of objects.

Existential Quantifier ("exists"): $\exists x\ P(x)$

There is an x in the domain for which $P(x)$ is true

read as "there exists x, P of x"

Examples:

Are these true? It depends on the domain. For example:

	{1, 3, -1, -27}	Integers	Positive Multiples of 5
$\exists x$ Odd(x)	True	True	True
$\exists x$ LessThan4(x)	True	True	False

Statements with Quantifiers

Just like with propositional logic, we need to define variables (this time **predicates**) before we do anything else. We must also now define a **domain of discourse** before doing anything else.

Domain of Discourse
Positive Integers

Predicate Definitions	
Even(x) ::= "x is even"	Greater(x, y) ::= "x > y"
Odd(x) ::= "x is odd"	Equal(x, y) ::= "x = y"
Prime(x) ::= "x is prime"	Sum(x, y, z) ::= "x + y = z"

Statements with Quantifiers

Domain of Discourse
Positive Integers

Predicate Definitions	
Even(x) ::= “x is even”	Greater(x, y) ::= “x > y”
Odd(x) ::= “x is odd”	Equal(x, y) ::= “x = y”
Prime(x) ::= “x is prime”	Sum(x, y, z) ::= “x + y = z”

Determine the truth values of each of these statements:

$\exists x \; \text{Even}(x)$

$\forall x \; \text{Odd}(x)$

$\forall x \; (\text{Even}(x) \vee \text{Odd}(x))$

$\exists x \; (\text{Even}(x) \wedge \text{Odd}(x))$

$\forall x \; \text{Greater}(x+1, x)$

$\exists x \; (\text{Even}(x) \wedge \text{Prime}(x))$

Statements with Quantifiers

Domain of Discourse
Positive Integers

Predicate Definitions	
Even(x) ::= "x is even"	Greater(x, y) ::= "x > y"
Odd(x) ::= "x is odd"	Equal(x, y) ::= "x = y"
Prime(x) ::= "x is prime"	Sum(x, y, z) ::= "x + y = z"

Determine the truth values of each of these statements:

$\exists x$ Even(x)	**T**	**e.g. 2, 4, 6, ...**
$\forall x$ Odd(x)	**F**	**e.g. 2, 4, 6, ...**
$\forall x$ (Even(x) $\vee$ Odd(x))	**T**	**every integer is either even or odd**
$\exists x$ (Even(x) $\wedge$ Odd(x))	**F**	**no integer is both even and odd**
$\forall x$ Greater(x+1, x)	**T**	**adding 1 makes a bigger number**
$\exists x$ (Even(x) $\wedge$ Prime(x))	**T**	**Even(2) is true and Prime(2) is true**

Statements with Quantifiers

Domain of Discourse
Positive Integers

Predicate Definitions	
Even(x) ::= "x is even"	Greater(x, y) ::= "x > y"
Odd(x) ::= "x is odd"	Equal(x, y) ::= "x = y"
Prime(x) ::= "x is prime"	Sum(x, y, z) ::= "x + y = z"

Translate the following statements to English

$\forall x\ \exists y\ \text{Greater}(y, x)$

$\forall x\ \exists y\ \text{Greater}(x, y)$

$\forall x\ \exists y\ (\text{Greater}(y, x) \wedge \text{Prime}(y))$

$\forall x\ (\text{Prime}(x) \rightarrow (\text{Equal}(x, 2) \vee \text{Odd}(x)))$

$\exists x\ \exists y\ (\text{Sum}(x, 2, y) \wedge \text{Prime}(x) \wedge \text{Prime}(y))$

Statements with Quantifiers (Literal Translations)

Domain of Discourse
Positive Integers

Predicate Definitions	
Even(x) ::= "x is even"	Greater(x, y) ::= "x > y"
Odd(x) ::= "x is odd"	Equal(x, y) ::= "x = y"
Prime(x) ::= "x is prime"	Sum(x, y, z) ::= "x + y = z"

Translate the following statements to English

$\forall x\ \exists y\ \text{Greater}(y, x)$

For every positive integer x, there is a positive integer y, such that y > x.

$\forall x\ \exists y\ \text{Greater}(x, y)$

For every positive integer x, there is a positive integer y, such that x > y.

$\forall x\ \exists y\ (\text{Greater}(y, x) \wedge \text{Prime}(y))$

For every positive integer x, there is a pos. int. y such that y > x and y is prime.

$\forall x\ (\text{Prime}(x) \rightarrow (\text{Equal}(x, 2) \vee \text{Odd}(x)))$

For each positive integer x, if x is prime, then x = 2 or x is odd.

$\exists x\ \exists y\ (\text{Sum}(x, 2, y) \wedge \text{Prime}(x) \wedge \text{Prime}(y))$

There exist positive integers x and y such that x + 2 = y and x and y are prime.

Statements with Quantifiers (Natural Translations)

Domain of Discourse
Positive Integers

Predicate Definitions	
Even(x) ::= "x is even"	Greater(x, y) ::= "x > y"
Odd(x) ::= "x is odd"	Equal(x, y) ::= "x = y"
Prime(x) ::= "x is prime"	Sum(x, y, z) ::= "x + y = z"

Translate the following statements to English

$\forall x\, \exists y\, \text{Greater}(y, x)$

There is no greatest positive integer.

$\forall x\, \exists y\, \text{Greater}(x, y)$

There is no least positive integer.

$\forall x\, \exists y\, (\text{Greater}(y, x) \wedge \text{Prime}(y))$

For every positive integer there is a larger number that is prime.

$\forall x\, (\text{Prime}(x) \rightarrow (\text{Equal}(x, 2) \vee \text{Odd}(x)))$

Every prime number is either 2 or odd.

$\exists x\, \exists y\, (\text{Sum}(x, 2, y) \wedge \text{Prime}(x) \wedge \text{Prime}(y))$

There exist prime numbers that differ by two."

Lecture 5 Activity

- You will be assigned to **breakout rooms**. Please:
- Introduce yourself
- Choose someone to share screen, showing this PDF
- What is the English language translation of

$$\exists x\ (Odd(x) \wedge LessThan(x, 5))$$

Fill out a poll everywhere for Activity Credit!
Go to pollev.com/philipmg and login with your UW identity

Domain of Discourse
Positive Integers

Predicate Definitions	
Odd(x) ::= “x is odd”	LessThan(x, y) ::= “x < y”

English to Predicate Logic

Domain of Discourse
Mammals

Predicate Definitions
Cat(x) ::= “x is a cat”
Red(x) ::= “x is red”
LikesTofu(x) ::= “x likes tofu”

“Red cats like tofu”

“Some red cats don’t like tofu”

English to Predicate Logic

Domain of Discourse
Mammals

Predicate Definitions
Cat(x) ::= “x is a cat”
Red(x) ::= “x is red”
LikesTofu(x) ::= “x likes tofu”

“Red cats like tofu”

$$\forall x\ ((\text{Red}(x) \land \text{Cat}(x)) \rightarrow \text{LikesTofu}(x))$$

“Some red cats don’t like tofu”

$$\exists y\ ((\text{Red}(y) \land \text{Cat}(y)) \land \neg\text{LikesTofu}(y))$$

English to Predicate Logic

Domain of Discourse
Mammals

Predicate Definitions
Cat(x) ::= "x is a cat"
Red(x) ::= "x is red"
LikesTofu(x) ::= "x likes tofu"

When putting two predicates together like this, we use an "and".

"Red cats like tofu"

When restricting to a smaller domain in a "for all" we use **implication.**

When there's no leading quantification, it means "for all".

"Some red cats don't like tofu"

When restricting to a smaller domain in an "exists" we use **and.**

"Some" means "there exists".

Negations of Quantifiers

Predicate Definitions
PurpleFruit(x) ::= "x is a purple fruit"

(*) $\forall x$ PurpleFruit(x) ("All fruits are purple")

What is the negation of (*)?

(a) "there exists a purple fruit"

(b) "there exists a non-purple fruit"

(c) "all fruits are not purple"

Try your intuition! Which one "feels" right?

Key Idea: In every domain, exactly one of a statement and its negation should be true.

Negations of Quantifiers

Predicate Definitions
PurpleFruit(x) ::= "x is a purple fruit"

(*) $\forall x$ PurpleFruit(x) ("All fruits are purple")

What is the negation of (*)?

(a) "there exists a purple fruit"

(b) "there exists a non-purple fruit"

(c) "all fruits are not purple"

Key Idea: In every domain, exactly one of a statement and its negation should be true.

Domain of Discourse
{plum}

Domain of Discourse
{apple}

Domain of Discourse
{plum, apple}

The only choice that ensures exactly one of the statement and its negation is (b).

De Morgan's Laws for Quantifiers

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

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

De Morgan's Laws for Quantifiers

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

"There is no largest integer"

$$\neg \exists x \forall y\ (x \geq y)$$
$$\equiv \forall x \neg \forall y\ (x \geq y)$$
$$\equiv \forall x\ \exists y \neg (x \geq y)$$
$$\equiv \forall x\ \exists y\ (x < y)$$

"For every integer there is a larger integer"

Scope of Quantifiers

$\exists x \ (P(x) \wedge Q(x))$ **vs.** $\exists x\, P(x) \wedge \exists x\, Q(x)$

Scope of Quantifiers

$\exists x \; (P(x) \land Q(x))$ **vs.** $\exists x \, P(x) \land \exists x \, Q(x)$

This one asserts P and Q of the *same* x.

This one asserts P and Q of potentially different x's.

Scope of Quantifiers

Example: NotLargest(x) $\equiv \exists y$ Greater (y, x)

$\equiv \exists z$ Greater (z, x)

truth value:

doesn't depend on y **or** z **"bound variables"**

does depend on x **"free variable"**

quantifiers only act on free variables of the formula they quantify

$$\forall x\ (\exists y\ (P(x,y) \rightarrow \forall x\ Q(y, x)))$$

Quantifier "Style"

$$\forall x(\exists y\,(P(x,y) \rightarrow \forall\, x\, Q(y, x)))$$

This isn't "wrong", it's just horrible style.
Don't confuse your reader by using the same variable multiple times…there are a lot of letters…

Nested Quantifiers

- Bound variable names don't matter

$$\forall x\ \exists y\ P(x, y) \equiv \forall a\ \exists b\ P(a, b)$$

- Positions of quantifiers can sometimes change

$$\forall x\ (Q(x) \wedge \exists y\ P(x, y)) \equiv \forall x\ \exists y\ (Q(x) \wedge P(x, y))$$

- But: order is important...

Quantifier Order Can Matter

Domain of Discourse

Integers
OR
{1, 2, 3, 4}

Predicate Definitions

GreaterEq(x, y) ::= “x ≥ y”

“There is a number greater than or equal to all numbers.”

$$\exists x\ \forall y\ \text{GreaterEq}(x, y)))$$

“Every number has a number greater than or equal to it.”

$$\forall y\ \exists x\ \text{GreaterEq}(x, y)))$$

x \ y	1	2	3	4
1	T	F	F	F
2	T	T	F	F
3	T	T	T	F
4	T	T	T	T

The purple statement requires **an entire row** to be true.
The red statement requires one entry in **each column** to be true.

Quantification with Two Variables

expression	when true	when false
$\forall x \forall y P(x, y)$	Every pair is true.	At least one pair is false.
$\exists x \exists y P(x, y)$	At least one pair is true.	All pairs are false.
$\forall x \exists y P(x, y)$	We can find a specific y for each x. $(x_1, y_1), (x_2, y_2), (x_3, y_3)$	Some x doesn't have a corresponding y.
$\exists y \forall x P(x, y)$	We can find ONE y that works no matter what x is. $(x_1, y), (x_2, y), (x_3, y)$	For any candidate y, there is an x that it doesn't work for.