Week 3: Logic

Agenda:

- 2.2-2.3 Quantified Statements
- 2.4 Arguments with Quantified Statements
- Scope of quantifiers and free variables
- Applications: relational database

Reading:

- Textbook pages 88–124.

More examples of logical equivalences:

- Example 1: It is easy to see that:

$$\exists x(P(x) \wedge q(x)) \Rightarrow \exists x\ P(x) \wedge \exists x\ Q(x)$$

  but these are not equivalent; for example let $P(x)$ be $2x + 1 = 5$ and $Q(x)$ be $x^2 = 9$.

  Example 2:

$$\exists x(P(x) \vee q(x)) \equiv \exists x P(x) \vee \exists x Q(x)$$

  Example 3:

$$\forall x(P(x) \wedge Q(x)) \equiv \forall x P(x) \wedge \forall x Q(x)$$

  Example 4:

$$\forall x P(x) \vee \forall x Q(x) \Rightarrow \forall x(P(x) \vee Q(x))$$

  but these are not logically equivalent; for example let $P(x)$ denote $x > 0$ and $Q(x)$ denote $x \leq 0$. Then clearly, for every real $P(x) \vee Q(x)$ but clearly it is not the case that $\forall x P(x) \vee \forall x Q(x)$.

- Negation of universal conditional statement:

$$\begin{aligned} \sim (\forall x, P(x) \rightarrow Q(x)) &\equiv \exists x \sim (P(x) \rightarrow Q(x)) \\ &\equiv \exists x, P(x) \wedge \sim Q(x) \end{aligned}$$

  e.g. for all students in CS program, if you have taken CMPUT 114 then you have taken 101 too.

  Negation: there is a student that has taken 114 but not 101.

- For statement $\forall x, P(x) \rightarrow Q(x)$:
  - Contrapositive: $\forall x, \sim Q(x) \rightarrow \sim P(x)$
  - Converse: $\forall x, Q(x) \rightarrow P(x)$
  - Inverse: $\forall x, \sim P(x) \rightarrow \sim Q(x)$

and the contrapositive is LEQ to the original; similarly, converse and inverse are LEQ but not to the original.

e.g. $\forall x \in \mathbb{R}, x^2 \leq 1 \rightarrow x \leq 1 \equiv \forall x \in \mathbb{R}, x > 1 \rightarrow x^2 > 1$
$\forall x \in \mathbb{R}, x \leq 1 \rightarrow x^2 \leq 1 \equiv \forall x \in \mathbb{R}, x^2 > 1 \rightarrow x > 1$

- Note:

$$\exists x \forall y P(x,y) \not\Leftarrow\not\Rightarrow \forall x \exists y P(x,y)$$

$$\exists x \forall y P(x,y) \not\Leftarrow\Rightarrow \forall y \exists x P(x,y)$$

- **Free Variables**
An occurance of a variable x is free in a formula F if it does not appear within a subformula of F of the form $\forall x E$ or $\exists x E$.

  e.g. $\forall x (x^2 > 4 \vee x + y = 1)$, y is free but x is *bound* to the quantifier.

  Variables that are not free are bound (to some quantifier).

- **Scope of Quantifier and binding of variables**

  In $\forall x F$, we say F is the scope of varible x.

$$\exists x \underbrace{(\forall y \overbrace{(P(x,y) \rightarrow Q(y))}^{\text{scope of } y} \wedge \exists u \overbrace{P(u,x)}^{\text{scope of } u})}_{\text{scope of } x}$$

  By this, we can say an occurance of a variable x is free if it does not appear within the scope of any quantifier.

- How to determine the quantifier to which a variable is bound?

  An occurance of a variable x is bound to the closest quantifer with that variable (inner most). This is similar to variable definitions in nested loops/procedures in a program: if a variable x is defined in nested loops, each use of x referres to the inner most definition of it.

  e.g. Let $M(x)$ mean "x is male"; $F(x)$ mean "y is female"; and $S(x,y)$ mean "x is a sibling of y". Then:

$$\forall x \forall y (F(y) \wedge \forall y (S(x,y) \rightarrow M(y)) \rightarrow \sim S(y,x))$$

(every female person is not the sibling of anyone all of whose siblings are male)

- One reason the above example is difficult to interpret is that different occurances of y are bound to different quantifiers. Easier if we change variable names.

- How can we change a variable name?

- Let Q be any quantifier ($\forall$ or $\exists$) and QxF be a quantified predicate. If y does not appear in F then we can replace every *free* occurance of x in F with y to obtain F_x^y and then $QxF \equiv QyF_x^y$.

  e.g. $\exists x(P(u,v,w) \wedge S(x,w) \rightarrow P(u,v,x)) \equiv \exists y(P(u,v,w) \wedge S(y,w) \rightarrow P(u,v,y))$

  $\exists x(P(u,v,w) \wedge S(x,w) \rightarrow \exists x P(u,v,x)) \equiv \exists y(P(u,v,w,) \wedge S(y,w) \rightarrow \exists x P(u,v,x))$.

- Note:

  - in the second example, the second x is not free (bound to quantifier $\exists$).

  - We can change variable name only in quantified predicates in the above rule.

- Another example: note that $\forall x(M(x) \vee F(x) \not\equiv \forall x M(x) \vee \forall x F(x)$.

  The LHS says, every person is either a man or woman. The RHS says, either everybody is a man or everybody is a woman.

Some more laws of equivalence.

- We have already seen that:

$$I) \quad \sim \forall x F \equiv \exists x \sim F$$
$$\sim \exists x F \equiv \forall x \sim F$$

- Suppose that x is not free in F then

$$IIa) \quad F \wedge \forall x E \equiv \forall x (F \wedge E)$$
$$F \wedge \exists x E \equiv \exists x (F \wedge E)$$

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

Similarly:

$$IIb) \quad F \vee \forall x E \equiv \forall x (F \vee E)$$
$$F \vee \exists x E \equiv \exists x (F \vee E)$$

$$IIc) \quad \forall x E \wedge F \equiv \forall x (E \wedge F)$$
$$\exists x E \wedge F \equiv \exists x (E \wedge F)$$

$$IId) \quad \forall x E \vee F \equiv \forall x (E \vee F)$$
$$\exists x E \vee F \equiv \exists x (E \vee F)$$

$$IIe) \quad \forall x E \to F \equiv \exists x (E \to F)$$
$$\exists x E \to F \equiv \forall x (E \to F)$$

$$IIf) \quad F \to \forall x E \equiv \forall x (F \to E)$$
$$F \to \exists x E \equiv \exists x (F \to E)$$

Note that IIc) and IIb) can be obtained from IIa) and IIb). Also we can obtain IIe) as follows:

$$\begin{aligned} \forall x E \to F &\equiv \sim \forall x E \vee F \\ &\equiv \exists x \sim E \vee F \\ &\equiv \exists x (\sim E \vee F) \\ &\equiv \exists x (E \to F). \end{aligned}$$

- example:

$$\begin{array}{ll} (\forall x P(x)) \rightarrow \forall x(Q(x) \rightarrow A(x) \vee B(x)) & \\ \equiv (\forall x P(x)) \rightarrow \forall y(Q(y) \rightarrow A(y) \vee B(y)) & \text{change of var} \\ \equiv \sim (\forall x P(x)) \vee \forall y(\sim Q(y) \vee A(y) \vee B(y)) & \rightarrow \text{law twice} \\ \equiv \exists x(\sim P(x)) \vee \forall y(\sim Q(y) \vee A(y) \vee B(y)) & \text{using } I) \\ \equiv \exists x(\sim P(x) \vee \forall y(\sim Q(y) \vee A(y) \vee B(y))) & \text{using } IId) \\ \equiv \exists x \forall y(\sim P(x) \vee \sim Q(y) \vee A(y) \vee B(y))) & \text{using } IIc) \end{array}$$

Arguments with quantified statements:

- Universal modus ponens:

$$\begin{array}{ll} & \forall x, P(x) \rightarrow Q(x) \\ & P(a) \\ \hline \therefore & Q(a) \end{array}$$

- Universal modus tollens:

$$\begin{array}{ll} & \forall x, P(x) \rightarrow Q(x) \\ & \sim Q(a) \\ \hline \therefore & \sim P(a) \end{array}$$

Applications: Relational database systems

- Predicate logic provides mathematical basis and conceptual framework for the most popular types of database systems: relational database

- Consider a library database; have three predicates:
  - `Books`: has a book id b, title t, and author name a; $\texttt{Books}(b, t, a)$ is true if there is a book with id b and title t and author a.
  - `Subs`: a subscriber predicate which has a parameter s corresponding to SIN number, a name n, and an address a; $\texttt{Subs}(s, n, a)$ is true if there is a subscriber with SIN number s, name n, and address a.
  - `Borrowed`: a predicate which shows subscriber s has borrowed book with id b and has a due date d: $\texttt{Borrowed}(s, b, d)$.

- We use predicate formulas to send queries;
- Example: Find all books writtne by "Williams"

$$\exists b \texttt{Books}(b, t, ''Williams'').$$

We have a free variabe t for titles. It reads: find the set of all values t' that can be substituted for t s.t. for some book id b, triple $(b, t, ''williams'')$ is true, i.e. there is a book with id b and title t' with author "$Williams''$.

- Example: find names and addresses of all that have borrowed books with due date 2007/1/1.

$$\exists s(\texttt{Subs}(s, n, a) \wedge \exists b \texttt{Borrowed}(s, b, 2007/1/1)).$$

Note that the results we look for always correspond to the free variables.
It reads: find the set of all pairs n', a' that can be substituted for variables n and a s.t. there is a subscriber with some SIN number and name n' and address a' that has borrowed some book b with due date 2007/1/1.

- Note that the following is different from example above:

$$\exists s \texttt{Subs}(s, n, a) \wedge \exists s \exists b \texttt{Borrowed}(s, b, 2007/1/1).$$

This returns the names and address of all subscribers if there is somebody that has borrowed a book with due date 2007/1/1.

- Example: Find the names of all subscribers who have borrowed a copy of "Mathematics" by "Kleene" and the due date by which it is to be returned.

$$\exists s(\exists a \texttt{Subs}(s, n, a) \wedge \exists b(\texttt{Books}(b, ''Mathematics'', ''kleene'') \wedge \texttt{Borrowed}(s, b, d)).$$

Two free variables n and d; it reads:
find all pairs n' and d' that can be substitued for n and d s.t. for some SIN s, there is a subscriber with SIN s, name n', and some address, and for some book, say b, the book with id b is "Mathematics" by "Kleene", and the subscriber has borrowed b and must be returned by d'.