INST0072 Lecture 4: Predicate Logic, Part 2

1. Recap: Signatures, Interpretations and Variable Assignments

- The *signature* of a predicate logic is a triple $\langle \mathcal{K}, \mathcal{F}, \mathcal{P} \rangle$, where $\mathcal{K}$ is the set of constant symbols in the language, $\mathcal{F}$ is the set of function symbols, and $\mathcal{P}$ is the set of predicate symbols.
- An *interpretation* $\mathbb{I}$ of $\langle \mathcal{K}, \mathcal{F}, \mathcal{P} \rangle$ is a domain of discourse $\mathbb{D}$ together with an assignment (i.e. mapping) of each symbol in $\mathcal{K}$ to an object in $\mathbb{D}$, an assignment of each symbol in $\mathcal{F}$ to a function over $\mathbb{D}$ of the same arity, and an assignment of each symbol in $\mathcal{P}$ to a relation over $\mathbb{D}$ of the same arity.
- For a given domain of discourse $\mathbb{D}$, a *variable assignment* $\mathbb{V}$ is an assignment of each variable in the logic to an object in $\mathbb{D}$.

2. Recap: Satisfaction of Quantifer-free Formulas - An Example

Going back to the example from Lecture 3:

signature: $\langle \{mani, nina\}, \{partner\_of/1\}, \{likes/2\} \rangle$
$\mathbb{D} = \{\bigcirc, \triangle, \square, \star\}$, interpretation $\mathbb{I}$, variable assignment $\mathbb{V}$
$\mathbb{I}(mani) = \square$, $\mathbb{I}(nina) = \triangle$
$\mathbb{I}(partner\_of) = \langle\ \bigcirc \mapsto \triangle,\ \triangle \mapsto \bigcirc,\ \square \mapsto \star,\ \star \mapsto \square\ \rangle$
$\mathbb{I}(likes) = \{\ (\bigcirc, \star),\ (\star, \bigcirc),\ (\triangle, \square),\ (\square, \triangle)\ \}$
variables x, y and z: $\mathbb{V}(x) = \bigcirc$, $\mathbb{V}(y) = \triangle$, $\mathbb{V}(z) = \bigcirc$

The quantifier-free formula

$$\neg likes(x, partner\_of(x)) \wedge likes(mani, nina)$$

is said to be *satified* by $\mathbb{I}$ and $\mathbb{V}$, because $\mathbb{I}(partner\_of) : \bigcirc \mapsto \triangle$, $(\bigcirc, \triangle) \notin \mathbb{I}(likes)$ and $(\square, \triangle) \in \mathbb{I}(likes)$. So we write:

$$\langle \mathbb{I}, \mathbb{V} \rangle \models \neg likes(x, partner\_of(x)) \wedge likes(mani, nina)$$

3. [Variable]-Equivalent Variable Assignments

Two variable assignments are *[variable]-equivalent* if they differ at most in the assignment of the variable "*[variable]*". For example:

- The following three variable assignments are x-equivalent:

$\mathbb{V}$: $\mathbb{V}(x) = \bigcirc$, $\mathbb{V}(y) = \triangle$, $\mathbb{V}(z) = \bigcirc$
$\mathbb{V}'$: $\mathbb{V}'(x) = \square$, $\mathbb{V}'(y) = \triangle$, $\mathbb{V}'(z) = \bigcirc$
$\mathbb{V}''$: $\mathbb{V}''(x) = \star$, $\mathbb{V}''(y) = \triangle$, $\mathbb{V}''(z) = \bigcirc$

- The following four variable assignments are y-equivalent:

$\mathbb{V}$: $\mathbb{V}(x) = \bigcirc$, $\mathbb{V}(y) = \triangle$, $\mathbb{V}(z) = \star$
$\mathbb{V}'$: $\mathbb{V}'(x) = \bigcirc$, $\mathbb{V}'(y) = \square$, $\mathbb{V}'(z) = \star$
$\mathbb{V}''$: $\mathbb{V}''(x) = \bigcirc$, $\mathbb{V}''(y) = \star$, $\mathbb{V}''(z) = \star$
$\mathbb{V}'''$: $\mathbb{V}'''(x) = \bigcirc$, $\mathbb{V}'''(y) = \bigcirc$, $\mathbb{V}'''(z) = \star$

Note: a variable assignment is always [variable]-equivalent to itself.

4. [Variable]-Equivalent Variable Assignments: More Examples

- Which of the following variable assignments are x-equivalent, which are y-equivalent, and which are z-equivalent?

$\mathbb{V}_1$: $\mathbb{V}_1(x) = \bigcirc$, $\mathbb{V}_1(y) = \triangle$, $\mathbb{V}_1(z) = \star$
$\mathbb{V}_2$: $\mathbb{V}_2(x) = \bigcirc$, $\mathbb{V}_2(y) = \star$, $\mathbb{V}_2(z) = \star$
$\mathbb{V}_3$: $\mathbb{V}_3(x) = \square$, $\mathbb{V}_1(y) = \triangle$, $\mathbb{V}_1(z) = \star$
$\mathbb{V}_4$: $\mathbb{V}_4(x) = \bigcirc$, $\mathbb{V}_4(y) = \square$, $\mathbb{V}_4(z) = \star$
$\mathbb{V}_5$: $\mathbb{V}_5(x) = \star$, $\mathbb{V}_5(y) = \star$, $\mathbb{V}_5(z) = \star$
$\mathbb{V}_6$: $\mathbb{V}_6(x) = \star$, $\mathbb{V}_6(y) = \star$, $\mathbb{V}_6(z) = \square$

5. Satisfaction of Quantifed Formulas

- A variable is said to be *free* within a formula $\mathcal{F}$ if it is not within the scope of an exitentially quantifed ($\exists$ - "exists") or a universally quantified ($\forall$ - "for all") quantifier.
- In the following, let x be free within the formula $\mathcal{F}$, let $\mathbb{I}$ be an interpretation, and let $\mathbb{V}$ be a variable assignment.
- Rule for existentially quantified variables:

$\langle \mathbb{I}, \mathbb{V} \rangle \models \exists x.\mathcal{F}$ if and only if there exists a variable assignment $\mathbb{V}'$ that is x-equivalent to $\mathbb{V}$ and is such that $\langle \mathbb{I}, \mathbb{V}' \rangle \models \mathcal{F}$.

- Rule for universally quantified variables:

$\langle \mathbb{I}, \mathbb{V} \rangle \models \forall x.\mathcal{F}$ if and only if $\langle \mathbb{I}, \mathbb{V}' \rangle \models \mathcal{F}$ for every variable assignment $\mathbb{V}'$ that is x-equivalent to $\mathbb{V}$.

6. Examples Using Formulas with Quantifiers

signature: $\langle \{mani, nina\}, \{partner\_of/1\}, \{likes/2\} \rangle$
$\mathbb{D} = \{\bigcirc, \triangle, \square, \star\}$, $\mathbb{I}(mani) = \square$, $\mathbb{I}(nina) = \triangle$
$\mathbb{I}(partner\_of) = \langle\ \bigcirc \mapsto \triangle,\ \triangle \mapsto \bigcirc,\ \square \mapsto \star,\ \star \mapsto \square\ \rangle$
$\mathbb{I}(likes) = \{\ (\bigcirc, \star),\ (\star, \bigcirc),\ (\triangle, \square),\ (\square, \triangle)\ \}$
x-equivalent variable assignments:
$\mathbb{V}$: $\mathbb{V}(x) = \bigcirc$, $\mathbb{V}(y) = \triangle$, $\mathbb{V}(z) = \bigcirc$
$\mathbb{V}'$: $\mathbb{V}'(x) = \triangle$, $\mathbb{V}'(y) = \triangle$, $\mathbb{V}'(z) = \bigcirc$
$\mathbb{V}''$: $\mathbb{V}''(x) = \square$, $\mathbb{V}''(y) = \triangle$, $\mathbb{V}''(z) = \bigcirc$
$\mathbb{V}'''$: $\mathbb{V}'''(x) = \star$, $\mathbb{V}'''(y) = \triangle$, $\mathbb{V}'''(z) = \bigcirc$

- $\langle \mathbb{I}, \mathbb{V} \rangle \models \exists x.likes(x, partner\_of(nina))$ because
  $\langle \mathbb{I}, \mathbb{V}''' \rangle \models likes(x, partner\_of(nina))$
- $\langle \mathbb{I}, \mathbb{V} \rangle \models \forall x.\neg likes(x, partner\_of(x))$ because
  $\langle \mathbb{I}, \mathbb{V} \rangle \models \neg likes(x, partner\_of(x))$
  $\langle \mathbb{I}, \mathbb{V}' \rangle \models \neg likes(x, partner\_of(x))$
  $\langle \mathbb{I}, \mathbb{V}'' \rangle \models \neg likes(x, partner\_of(x))$
  $\langle \mathbb{I}, \mathbb{V}''' \rangle \models \neg likes(x, partner\_of(x))$

7. Example with Two Quantifiers

signature: $\langle \{mani, nina\}, \{partner\_of/1\}, \{likes/2\} \rangle$
$\mathbb{D} = \{\bigcirc, \triangle, \square, \star\}$, $\mathbb{I}(mani) = \square$, $\mathbb{I}(nina) = \triangle$
$\mathbb{I}(partner\_of) = \langle\ \bigcirc \mapsto \triangle,\ \triangle \mapsto \bigcirc,\ \square \mapsto \star,\ \star \mapsto \square\ \rangle$
$\mathbb{I}(likes) = \{\ (\bigcirc, \star),\ (\star, \bigcirc),\ (\triangle, \square),\ (\square, \triangle)\ \}$

- $\langle \mathbb{I}, \mathbb{V}'' \rangle \models likes(x, y)$
  where $\mathbb{V}''$: $\mathbb{V}''(x) = \triangle$, $\mathbb{V}''(y) = \square$, $\mathbb{V}''(z) = \bigcirc$
- therefore $\langle \mathbb{I}, \mathbb{V}' \rangle \models \exists y.likes(x, y)$
  where $\mathbb{V}'$: $\mathbb{V}'(x) = \triangle$, $\mathbb{V}'(y) = \triangle$, $\mathbb{V}'(z) = \bigcirc$
  because $\mathbb{V}'$ is y-equivalent to $\mathbb{V}''$
- therefore $\langle \mathbb{I}, \mathbb{V} \rangle \models \exists x \exists y.likes(x, y)$
  where $\mathbb{V}$: $\mathbb{V}(x) = \bigcirc$, $\mathbb{V}(y) = \triangle$, $\mathbb{V}(z) = \bigcirc$
  because $\mathbb{V}$ is x-equivalent to $\mathbb{V}'$

The above shows that $\langle \mathbb{I}, \mathbb{V} \rangle \models \exists x \exists y.likes(x, y)$ for any $\mathbb{V}$, so we can just write

$$\mathbb{I} \models \exists x \exists y.likes(x, y)$$

8. More Example Interpretations

Given the following signature and domain of discourse $\mathbb{D}$:

signature: $\langle \{mani, nina\}, \{partner\_of/1\}, \{likes/2\} \rangle$
$\mathbb{D} = \{\triangle, \square, \star\}$

for each of the following statements, give a specification of the interpretation that will make the statement correct:

- $\mathbb{I}_1 \models \forall x.likes(x, partner\_of(x))$
- $\mathbb{I}_2 \models \forall x \forall y.[likes(x, y) \rightarrow \neg likes(y, x)]$

Interpretations that make a sentence true in this way are called *models* of the sentence (in the same way as for propositional logic).

9. Comparing Propositional and Predicate Logic

- The connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftarrow$ and $\leftrightarrow$ have the same meanings in both propositional and predicate logic (given by truth tables).
- For formulas, an interpretation/variable assignment pair of the form $\langle \mathbb{I}, \mathbb{V} \rangle$ in predicate logic plays the same role as an interpretation (a row in a truth table) in propositional logic.
- For sentences (i.e. formulas in which every variable is quantified), an interpretation $\mathbb{I}$ on its own in predicate logic plays the same role as an interpretation in propositional logic.
- Because of this, much of the terminology and theoretical results are the same in both propositional and predicate logic, as illustrated in the next few slides.

10. Logic Terminology for Sentences

- **Satisfaction**: An interpretation $\mathbb{I}$ ***satisfies*** a sentence if it assigns a value of true (t) to the sentence, in which case $\mathbb{I}$ is a ***model*** of the sentence. A sentence is ***satisfiable*** or ***consistent*** if it has at least one model. Otherwise it is ***unsatisfiable*** or ***inconsistent*** or ***a contradiction***.
- **Semantic Equivalence**: Two sentences are ***semantically equivalent*** if they have the same models.
- **Validity**: A sentence is ***(logically) valid*** or ***a tautology*** if it is satified by every interpretation, i.e. if every interpretation is a model of the sentence. For any sentence A, the expression $\models A$ means that A is logically valid.
- **Entailment**: For any two sentences A and B, the expression $A \models B$ (read as "A ***entails*** B") means that every model of A is also a model of B. In this case B ***is entailed by*** or is a ***semantic consequence*** of A.

11. Two Theorems of Predicate Logic

The Deduction Theorem (Predicate Version)
For any two sentences A and B

$$A \models B \quad \text{if and only if} \quad \models A \rightarrow B$$

Proof is by considering all the possible models for $A \rightarrow B$. (See Ertel, page 27.)

Proof by Contradiction (Predicate Version)
For any two sentences A and B

$$A \models B \quad \text{if and only if} \quad A \wedge \neg B \text{ is a contradiction.}$$

Proof: $A \wedge \neg B$ is a contradiction if and only if $\neg(A \wedge \neg B)$ is a tautology. $\neg(A \wedge \neg B)$ is equivalent to $\neg A \vee \neg\neg B$ and so also to $A \rightarrow B$, so the theorem is a consequence of the Deduction Theorem.

12. Sets of Sentences, Knowledge Bases, Theories and Axioms

- A predicate logic *knowledge base* is a (possibly large) finite set KB of sentences. A set of sentences is also sometimes called a *theory* or *axiomatisation*, and the sentences within it are sometimes called *axioms*.
- In predicate logic, the theory $KB = \{S_1, S_2, \ldots, S_n\}$ is defined as semantically equivalent to the single (possibly long) sentence $S_1 \wedge S_2 \wedge \ldots \wedge S_n$. So

$$KB \models S \quad \text{means} \quad (S_1 \wedge S_2 \wedge \ldots \wedge S_n) \models S$$

- Using the above, the *Proof by Contradiction* theorem can be re-expressed in terms of sets of sentences:

Proof by Contradiction (set-theoretic version)
For any knowledge base or theory KB and sentence S

$$KB \models S \quad \text{if and only if} \quad KB \cup \{\neg S\} \text{ is unsatisfiable.}$$

13. Using Predicate Logic with Equality

- In this module we are using *predicate logic with equality*, so the predicate symbol = is part of the core logic (like $\neg$, $\rightarrow$, etc.) and not just part of a specific theory's signature.
- In all interpretations, = is interpreted as the identity relation, so that $A = B$ always means that A and B refer to the same object in the domain of discourse. $A \neq B$ is shorthand for $\neg(A = B)$.
- We can use sentences with = and $\neq$ to make sure all the models of a theory use a domain of discourse of a particular size. For example, the following sentence ensures that, in all models, the domain of discourse contains three objects:

$$\exists w \exists y \exists z.[w \neq y \wedge w \neq z \wedge y \neq z \wedge \forall x.(x = w \vee x = y \vee x = z)]$$

Note: Ertel (page 45, section 3.2.1) takes a different approach to equality, by using extra sentences to define its essential properties.

14. More Examples with = and $\neq$

Given the following signature:

$$\langle \{mani, nina\}, \{partner\_of/1\}, \{likes/2\} \rangle$$

write down a sentence that ensures that, in all models, ...

- ... "mani" and "nina" are the only two objects in the domain, and are distinct from each other.
- ... no two people share the same partner.

Sentences of this type are often referred to as *uniqueness-of-names axioms* (UNAs) and *domain closure axioms* (DCAs).

15. Converting Between $\forall$ and $\exists$

- In natural language, "for all x, statement S is true" means the same as "there is not an x such that statement S is false". For example, "all men are mortal" means the same as "there is not a man who is not mortal".
- The descriptions of $\forall$ and $\exists$ on Slide 16, Lecture 3 fit with this idea. In any formula, and for any variable x, "$\forall x$" can be replaced by "$\neg\exists x\neg$", and "$\exists x$" can be replaced by "$\neg\forall x\neg$", without changing the meaning (i.e. the models) of the formula.
- For example, the statement "everybody loves somebody else" can be represented by:

$$\begin{aligned} & \forall x \exists y.[loves(x,y) \wedge x \neq y] \\ \equiv\ & \neg\exists x\neg\neg\forall y.\neg[loves(x,y) \wedge x \neq y] \\ \equiv\ & \neg\exists x\forall y.\neg[loves(x,y) \wedge x \neq y] \\ \equiv\ & \neg\exists x\forall y.[x \neq y\ \rightarrow \neg loves(x,y)] \end{aligned}$$

  meaning "there is nobody who doesn't love anybody else".

16. Prenex Normal Form

- A formula is in *prenex normal form* if it is either of the form F or of the form $Q_1x_1Q_2x_2...Q_nx_n.F$, where each Q_i is a $\forall$ or $\exists$ and the formula F contains no quantifiers.
- Examples of sentences in prenex normal form:
  $\forall x.[(frog(x) \wedge brown(x)) \rightarrow big(x)]$
  $\forall x.\neg likes(x, cake)$
  $\exists x\exists y.(x \neq y \wedge is\_bird(x) \wedge is\_bird(y) \wedge is\_bigger\_than(x,y))$
  $\forall x\exists y.[loves(x,y) \wedge x \neq y]$
- Examples of sentences not in prenex normal form:
  $\neg\exists x.likes(x, cake)$
  $\exists x.[baker(x)\ \wedge \forall y.(customer(y) \rightarrow likes(x,y))]$
  $\neg\exists x\forall y.[x \neq y\ \rightarrow \neg loves(x,y)]$

17. A Theorem about Prenex Normal Form

Theorem
Every predicate logic formula can be transformed into a semantically equivalent formula that is in prenex normal form.

Examples:

- $\neg\exists x.likes(x, cake)$
  is equivalent to
  $\forall x.\neg likes(x, cake)$
- $\exists x.[baker(x) \wedge \forall y.(customer(y) \rightarrow likes(x, y))]$
  is equivalent to
  $\exists x \forall y.[baker(x) \wedge (customer(y) \rightarrow likes(x, y))]$
- $\neg\exists x \forall y.[x \neq y \rightarrow \neg loves(x, y)]$
  is equivalent to
  $\forall x \exists y.[loves(x, y) \wedge x \neq y]$

18. Converting to Prenex Normal Form

Care has to be taken when converting to prenex normal form. Here is a general procedure:

1. use logical equivalences to eliminate all occurrences of $\rightarrow$, $\leftarrow$ and $\leftrightarrow$ from the formula.
2. Again using logical equivalences and converting between $\forall$s and $\exists$s as necessary, move all negations inwards so that, in the end, negations appear only as part of literals.
3. Rename variables if necessary so that each variable appears only within the scope of a single quantifier (this is sometimes called "standardising variables apart").
4. Move all quantifiers to the front of the formula.

See e.g. https://www.csd.uwo.ca/~lila/prenex.pdf for details and examples.

19. An Example Conversion to Prenex Normal Form

"If everyone is happy then there must be a cake around!"

$$
\begin{array}{ll}
 & (\forall x.happy(x)) \rightarrow (\exists x.cake(x)) \quad \text{[1. eliminate } \rightarrow\text{]} \\
\equiv & \neg(\forall x.happy(x)) \vee (\exists x.cake(x)) \quad \text{[2. move } \neg \text{ inwards]} \\
\equiv & (\exists x.\neg happy(x)) \vee (\exists x.cake(x)) \quad \text{[3. rename 2nd } x \text{ to } y\text{]} \\
\equiv & \exists x.\neg happy(x) \vee \exists y.cake(y) \quad \text{[4. move quantifiers to front]} \\
\equiv & \exists x \exists y.(\neg happy(x) \vee cake(y)) \quad \text{[now in prenex normal form]} \\
\equiv & \exists x \exists y.(happy(x) \rightarrow cake(y)) \quad \text{[also in prenex normal form]}
\end{array}
$$

Notice that in prenex normal form the quantifiers have changed – there is no longer a $\forall$ in the formula.
(An aside: to avoid "happy cake" we could use many-sorted logic with distinct sorts for "people" and "food" - see e.g. Wikipedia)

20. Notes about Ertel Chapter 3

- Page 41, Table 3.1: 11th row should say "There is a customer who likes bob" (and not "... whom bob likes").
- Page 42, Definition 3.3 is non-standard (it combines a variable assignment with an interpretation).
- Page 45, Section 3.2.1: this is what you have to do if you want to treat equality as just another predicate in the logic's signature. Using Predicate Logic With Equality, as in the lecture notes, is easier!
- Page 45, beginning of Section 3.3: Ertel is wrong here – $\forall x.p(x)$ is not necessarily the same as $p(a_1) \wedge \ldots \wedge p(a_n)$, because the domain of discourse might include extra objects not named by one of the constants $a_1, \ldots, a_n$.
- Page 46, the formulas in the example halfway down the page could do with some extra brackets. By "$\forall x\, p(x) \Rightarrow \exists x\, q(x)$" Ertel means "$\forall x(p(x) \Rightarrow \exists x\, q(x))$".