From propositions to relations

Let's express in terms of an ontology...

σ
$p \triangleq$ The table has four chairs.
$q \triangleq$ A painting is on the wall or a photograph is on the wall.
$r \triangleq$ The computer is not on.
$s \triangleq$ If the computer is on, then it is not dead.
$t \triangleq$ If the computer is on, then the battery is not dead.
$u \triangleq$ The plant is made of plastic.

$$(\quad , \quad) \in \texttt{Has4Chairs}$$

Need to express restrictions on relations

Our restriction:

1. if $(\texttt{table}, 0) \in \texttt{Has4Chairs}$, then $(\texttt{table}, 1) \notin \texttt{Has4Chairs}$
2. if $(\texttt{table}, 1) \in \texttt{Has4Chairs}$, then $(\texttt{table}, 0) \notin \texttt{Has4Chairs}$

Is this true over every $f \in \mathit{Furniture}$, or just for the table?

If $f \in \mathit{Furniture}$, then either $(f, 0) \in \texttt{Has4Chairs}$
or $(f, 1) \in \texttt{Has4Chairs}$,
but not both.

Quantifying over sets

- Using a propositional variable is shorter
  ...but it hides information about the set!
- What do we mean by an arbitrary member of the set?

	$\forall$ "for all" (`\forall`)	$\exists$ "exists" (`\exists`)
Set membership	$\forall f\,(f \in \mathit{Furniture} \wedge \ldots)$	$\exists f\,(f \in \mathit{Furniture} \wedge \ldots)$
Relation membership	$\forall f\,((f,1) \in F \wedge \ldots)$	$\exists f\,((f,1) \in F \wedge \ldots)$
Predicates	$\forall f(F(f) \wedge \ldots)$	$\exists f(F(f) \wedge \ldots)$

Predicates parameterize propositions.

Predicate logic: Syntax

Predicates compose into *formulas* via *connectives*, which are *inductively* defined. Assume a set of variables $\{x_1, x_, \ldots\}$ and predicates $\{P_1, P_2, \ldots\}$:

- *Atomic formulas*: $F \in \{\top, \bot, P_1(x_1), P_1(x_2), P_2(x_1), P_2(x_2), P_1(x_3), P_2(x_3), P_3(x_1), ...\}$
- *Negation*: If F is a formula, then $\neg F$ is a formula.
- *Disjunction*: If F and G are formulas, then $F \vee G$ is a formula.
- *Conjunction*: If F and G are formulas, then $F \wedge G$ is a formula.
- *Implication*: If F and G are formulas, then $F \rightarrow G$ is a formula.
- *Universal quantification*: If x is a variable and F is a formula, then $\forall x(F)$ is a formula.
- *Existential quantification*: If x is a variable and F is a formula, then $\exists x(F)$ is a formula.

You should be able to recognize syntactically valid formulas.

Free as a bird

Syntax does *not* say that variables need to be quantified.

Variables that occur outside a quantifier (i.e., $\forall$ or $\exists$) are said to be *free*, otherwise they are *bound*.

$$P(a) \wedge \forall b\,(Q(b))$$

$$P(a) \wedge \forall b\,\big(Q(b) \vee R(a)\big)$$

$$\forall a\,\big(P(a) \wedge \forall b\,\big(Q(b) \vee R(a)\big)\,\big)$$

You should recognize free and bound variables.

Example syntactic manipulation

You should be able to manipulate formulas in both propositional and predicate logic using rewrite rules.

If $f \in \mathit{Furniture}$, then either $(f, 0) \in \texttt{Has4Chairs}$
or $(f, 1) \in \texttt{Has4Chairs}$,
but not both.

$$\forall f \forall f' \forall b \left((f = f' \wedge (f, b) \in P) \to \neg \exists b' \left(\qquad\qquad \right) \right)$$

Clicker Question

Which of the following is *not* a valid rewrite of the expression

$$\forall f \forall f' \forall b \left((f = f' \land (f, b) \in P) \rightarrow \neg \exists b' \left(b \neq b' \land (f, b') \in P \right) \right)$$

A) $\forall f \forall f' \forall b \left((f = f' \land (f, b) \in P) \rightarrow \forall b' \neg \left(b \neq b' \land (f, b') \in P \right) \right)$

B) $\forall f \forall f' \forall b \left((f = f' \land (f, b) \in P) \rightarrow \forall b' \left(b = b' \land (f, b') \notin P \right) \right)$

C) $\forall f \forall f' \forall b \left(\neg (f = f' \land (f, b) \in P) \lor \neg \exists b' \left(b \neq b' \land (f, b') \in P \right) \right)$

D) $\forall f \forall f' \forall b \forall b' \left((f = f' \land (f, b) \in P) \rightarrow \left(b \neq b' \land (f, b') \in P \right) \right)$

E) $\forall a \forall b \forall c \left((a = b \land (a, c) \in P) \rightarrow \neg \exists d \left(c \neq d \land (a, d) \in P) \right) \right)$

Relations, functions, predicates

Recall:

- An ontology is a collection of categories and relations.
- A *relation* is a pairing between categories.
- We are focusing on a special type of category: a set.
- A relation R on two sets X and Y is any ordered pairing where for $x \in X$ and $y \in Y$, $(x, y) \in R$.
- A relation can also be written as $R(x) = y$ or $R(x, y)$.

Some relations have special restrictions:

- A *function* is a relation R such that $\forall x \forall x' \forall y \forall y' \, ((R(x) = y \wedge R(x') = y') \rightarrow (y = y' \vee x \neq x'))$, typically written $\forall x \forall x' \, (R(x) = R(x') \rightarrow x = x')$.
- A *predicate* is a function R where $\forall x \, (R(x) \in \{0, 1\})$.

Arity

Relations can be arbitrarily-sized pairs:

$$(\text{Larry, Moe, Curly}) \notin \textit{Triplets}$$

Relations (including functions and predicates) can be written many argument. When there are k such arguments, we say the relation R is a k-ary relation. k is the *arity*.

Let P^k denote an arbitrary k-ary predicate and f^k denote a k-ary function. Let $\{x_1, x_2, \ldots\}$ be the set of variables, as before. Then atomic formulas are the set defined by $\{\top, \bot, P_1^k(t_1, \ldots, t_k), P_2^{k'}(t_1, \ldots, t_{k'}), \ldots\}$ and t_i is a term, defined to be:

- A variable from the set of variables, or
- A k-ary function from the set of function applied to k terms $(f_i^k(t_1, \ldots, t_k))$

Semantics, classically...

- The semantics of predicate logic is defined by a *structure* $\mathcal{A} = \langle \mathcal{U}_\mathcal{A}, [\![\cdot]\!]_\mathcal{A} \rangle$
- $\mathcal{U}_\mathcal{A}$ is the *universe*: an arbitrary, non-empty set that gives meaning to variables.
- A structure is suitable if $[\![\cdot]\!]_\mathcal{A}$ is defined for every symbol, e.g.:

$$
\begin{aligned}
F &\triangleq \forall x\,(L(f(x), a) \rightarrow S(x)) \\
\mathcal{U} &\triangleq \{\text{carpet, lamp, couch, table, chair}_1, \ldots, \text{chair}_4\} \\
[\![a]\!]_\mathcal{A} &\triangleq 1 \\
[\![f]\!]_\mathcal{A} &\triangleq \text{function to get the number of legs the input has} \\
[\![L]\!]_\mathcal{A} &\triangleq \text{predicate: first is greater than the second} \\
[\![S]\!]_\mathcal{A} &\triangleq \text{predicate: can sit on}
\end{aligned}
$$

- Interpreting predicates: $\mathcal{A} \models F$?

Logic is general; we want to be specific

- The deep formalisms here capture extremely general and true things: proof schemata, the notion of equality, etc.
- In AI, we want to be specific.
- Classic example: the "frame problem."

Next class

Inference: combining what we know to learn something new.