We need logic...

- Need a more expressive language for encoding states, goals, operators.

- Need a reasoning capability

  – we already have some reasoning capability, i.e. reasoning how the world would change if we acted upon it (applicability and effect of operators on states)
  $\Rightarrow$ diachronic reasoning
  (across time / between states)

  – but we want more
  $\Rightarrow$ synchronic reasoning
  (same time / within a state)
  i.e. leave some facts implicit, obtain them, e.g. as needed, by reasoning

  (Focus on synchronic; will return to diachronic)

...but what's a logic...

- Every logic (or knowledge representation language) has

  – **syntax**: defines legitimate configurations of symbols

  – **semantics**: defines the meanings of symbols and of configs of symbols (and, hence, relationships between configs)

  – **proof theory**: defines reasoning steps (using only syntactic operations)

- There are many logics (or KRLs) having different

  – *ontological commitments* (what they assume exists in'reality')

  – *epistemological commitments* (what an agent can believe about facts)

...so which logic?

- Fewer commitments means lower expressiveness but more efficient reasoning.

- E.g. propositional logic
  ontology: facts
  epistemology: true / false
  inadequate expressiveness; even simple arguments don't hold

- E.g. first-order logic (FOL)/ first-order predicate calculus (FOPC)
  ontology: objects, properties/relationships, facts
  epistemology: true / false
  Expressive enough (even more exotic logics can be 'reified' into FOPC) but expressiveness may need to be restricted to give efficient reasoning.
  It's our choice.

The syntax of first-order predicate calculus: terms

1. Every constant symbol is a term, e.g. a, b, c, *clyde*

2. Every variable is a term, e.g. X, Y, Z, *Animal*

3. For every function symbol F of arity n and any n terms $T_1, T_2, \ldots, T_n$ then

$$F(T_1, T_2, \ldots, T_n)$$

   is a term, e.g. $f(a), g(b, c), g(f(a), c), g(f(f(a)), b)$

4. Nothing else is a term.

The syntax of first-order predicate calculus: atoms

- For any predicate symbol P of arity n and any n terms $T_1, T_2, \dots T_n$ then

$$P(T_1, T_2, \dots T_n)$$

is an **atom**, e.g.
$p(a)$, *elephant(clyde)*,
likes(clyde, gertie), $q(a, f(a))$

The syntax of first-order predicate calculus: well-formed formulae

1. Every atom is a wff.

2. For any variable X and any wff W, $(\forall X W)$ and $(\exists X W)$ are wffs (**quantified wffs**).

3. For any wffs W, W_1 and W_2, the following are also wffs: $(\neg W)$, $(W_1 \wedge W_2)$, $(W_1 \vee W_2)$, $(W_1 \rightarrow W_2)$ and $(W_1 \leftrightarrow W_2)$

4. No other string of symbols is a wff.

E.g.
elephant(clyde), $\neg$*likes(clyde, gertie)*,
elephant(clyde) $\vee \neg$*likes(clyde, gertie)*,
$\forall X (p(X) \rightarrow (q(b, c) \wedge q(f(a), X)))$

Unnecessary parentheses will be omitted.

The semantics of first-order predicate calculus

In predicate logic, the meaning of the lowest level expressions (constant symbols, function symbols and predicate symbols) will be *stipulated* by an *interpretation*. Then, the meaning of a wff will be determined by the meanings of its subexpressions (in this case, its terms and subwffs).

We assume that there is some set of entities (possibly an infinite set) which we refer to as the **universe of discourse**, and which we denote as $\mathcal{U}$.

It is obviously possible to define a large number of relations *on* $\mathcal{U}$.

Example: The Books World

- Entities: Here we have five entities (excluding the table): Down&Out, Zen, Cuckoo, Jude and Spire.

- Functions: Here we have such functions as the unary function that maps from a book to the book directly on top of it.

- Relations: Here we have such relations as the binary relation that tells us whether a pair of books is such that the second is *somewhere* above the first, and the unary relation that tells us which books have a clear top.

Interpretations

In predicate logic, an interpretation $\mathcal{I}$ makes the following mappings:

1. For every constant symbol C, $\mathcal{I}(C)$ is an entity from $\mathcal{U}$. (Different constant symbols need not denote different entities, and there may be unnamed entities.)

2. For every n-ary function symbol F, $\mathcal{I}(F)$ is an n-ary function on $\mathcal{U}$, i.e. one that maps n-tuples to an entity. (Different function symbols need not denote different functions, and there may be unnamed functions.)

3. For every n-ary predicate symbol P, $\mathcal{I}(P)$ is an n-ary relation on $\mathcal{U}$, i.e. one that holds of n-tuples. (Different predicate symbols need not denote different relations, and there may be unnamed relations.)

Interpretation: example

For constant symbols a, b, c, d, e, a unary function symbol f and the binary predicate symbols p and q, we give an interpretation $\mathcal{I}_1$:

$\mathcal{I}_1(a)$ is 

(Down&Out)

$\mathcal{I}_1(b)$ is  (Zen)

$\mathcal{I}_1(c)$ is  (Cuckoo)

$\mathcal{I}_1(d)$ is  (Jude)

$\mathcal{I}_1(e)$ is  (Spire)

Interpretation: example, cont'd.

$\mathcal{I}_1(f)$ is	the unary function that maps a book to the book *directly on top* of it
$\mathcal{I}_1(p)$ is	the binary relation whose graph/extension contains pairs of books where the second book of the pair is *directly on top* of the first book of the pair
$\mathcal{I}_1(q)$ is	the binary relation whose graph/extension contains pairs of books where the second book of the pair is *somewhere above* the first book of the pair

Interpretation: example

Here is another example interpretation, $\mathcal{I}_2$, for the same universe of discourse and the same symbols. $\mathcal{I}_2$ is identical to $\mathcal{I}_1$ except:

$\mathcal{I}_2(a)$ is  (Zen)

$\mathcal{I}_2(b)$ is 

(Down&Out)

$\mathcal{I}_2(p)$ is	the binary relation whose graph/extension contains pairs of books where the second book of the pair is *directly below* the first book of the pair
$\mathcal{I}_2(q)$ is	the binary relation whose graph/extension contains pairs of books where the second book of the pair is *directly on top* of the first book of the pair

The semantics of first-order predicate calculus, cont'd.

$\mathcal{U}$ and $\mathcal{I}$ define a model M:

$$M = \langle \mathcal{U}, \mathcal{I} \rangle$$

The **semantic value** or **denotation** of some *term* or *wff* α with respect to model M is written

$$[\![\alpha]\!]^M$$

- If α is a constant, function symbol or predicate symbol, then

  $$[\![\alpha]\!]^M = \mathcal{I}(\alpha)$$

  (i.e. the meaning is stipulated by $\mathcal{I}$).

The semantics of first-order predicate calculus, cont'd.

- For any term comprising an n-ary function symbol F applied to terms $T_1, T_2, \ldots, T_n$, then

  $$[\![F(T_1, T_2, \ldots, T_n)]\!]^M = [\![F]\!]^M([\![T_1]\!]^M, [\![T_2]\!]^M, \ldots, [\![T_n]\!]^M)$$

  i.e. you get the function that corresponds to F (as stipulated by $\mathcal{I}$), you work out the objects that correspond to $T_1, T_2, \ldots, T_n$, and then you apply the function to these objects. This gets you an object, and this object is the meaning of the term $F(T_1, T_2, \ldots, T_n)$.

The semantics of first-order predicate calculus, cont'd.

- For any atom comprising an n-ary predicate symbol P applied to terms $T_1, T_2, \ldots, T_n$ then

  $$[\![P(T_1, T_2, \ldots, T_n)]\!]^M = \text{ true iff } \langle [\![T_1]\!]^M, [\![T_2]\!]^M, \ldots, [\![T_n]\!]^M \rangle \in [\![P]\!]^M$$

  I.e. you get the relation that corresponds to P (as stipulated by $\mathcal{I}$), you work out the objects that correspond to $T_1, T_2, \ldots, T_n$, and you see whether this n-tuple of objects is an element of the relation.

The semantics of first-order predicate calculus, cont'd

- For any quantified wff $\forall X W$, $[\![\forall X W]\!]^M$ is true iff $[\![W]\!]^M$ is true for all assignments of entities in $\mathcal{U}$ to X. Otherwise, it is false.

- For any quantified wff $\exists X W$, $[\![\exists X W]\!]^M$ is true iff $[\![W]\!]^M$ is true for at least one assignment of an entity in $\mathcal{U}$ to X. Otherwise, it is false.

- For any compound wff $\neg W, W_1 \wedge W_2, W_1 \vee W_2,$ $W_1 \rightarrow W_2$ and $W_1 \leftrightarrow W_2$, determine $[\![W]\!]^M$, $[\![W_1]\!]^M$ and $[\![W_2]\!]^M$ and use these truth values as in propositional logic to determine the truth value of the compound wff.

The semantics of first-order calculus: making it more formal

- We introduce a **value assignment function**, g. Then compute the semantic value or denotation of α w.r.t. both M and g:

$$[\![\alpha]\!]^{M,g}$$

- Gives a more formal assignment of entities to variables:
  - allows free variables
  - makes definitions more properly recursive.

Metatheoretic definitions

- A wff α is **satisfiable** iff there exists some model M such that $[\![\alpha]\!]^{M}$ is true.

- E.g.

$$p(a,b)$$

$$p(b,a)$$

$$\forall X(q(X) \rightarrow r(X))$$

Metatheoretic definitions continued

- A wff α is **unsatisfiable** iff there exists no model M such that $[\![\alpha]\!]^{M}$ is true.

- E.g.

$$p(a,b) \wedge \neg p(a,b)$$

$$\neg(p(b,a) \rightarrow p(b,a))$$

Metatheoretic definitions continued

- A wff α is **valid** iff for every model M, $[\![\alpha]\!]^{M}$ is true.

- E.g.

$$p(a,b) \vee \neg p(a,b)$$

$$p(b,a) \rightarrow p(b,a)$$

Metatheoretic definitions continued

- A set of wffs Φ has wff α as a **logical consequence** or **entailment** iff
  every model that makes Φ true also makes α true.

- We write $\Phi \models \alpha$

- E.g. $\{\forall X(p(X) \to q(X)), p(a)\} \models q(a)$

- N.B. A semantic notion. Requires checking *all* models.

Proof theories for first-order predicate calculus

- Enables us to *derive/ deduce/ prove* that a conclusion α follows from a set of premisses Φ by *syntactic manipulation alone*.

- We write $\Phi \vdash \alpha$

- A proof theory comprises
  - a set of *inference rules*
  - a set of *logical axiom schemata*

- You supply Φ, the *premisses* (*initial axioms/ proper axioms*).

Formal proof

- A **proof** that $\Phi \vdash \alpha$ is a sequence of wffs such that each wff is either
  - a member of Φ
  - a logical axiom
    (produced by substitution into a logical axiom schemata)
  - the result of applying an inference rule

  and the final wff is α.

A (Hilbert-style) proof theory for propositional logic

- Logical axiom schemata:

  Schemata 1: $(\neg\alpha \to \alpha) \to \alpha$
  Schemata 2: $\alpha \to (\neg\alpha \to \beta)$
  Schemata 3: $(\alpha \to \beta) \to ((\beta \to \gamma) \to (\alpha \to \gamma))$

- One inference rule: $\frac{\vdash\alpha \quad \vdash\alpha\to\beta}{\vdash\beta}$ (Modus ponens)

- Proper axioms, e.g.
  1. *rain* $\to$ *wet*
  2. *wet* $\to$ *cold*

  Can prove *rain* $\to$ *cold*

Searching for derivations

- Finding a proof is a search process. The search space, particularly in this kind of proof theory, can be huge.
- There is an infinite number of wffs we can substitute in each logical axima schemata.
- Most proof theories have more inference rules.
- (This isn't state space search — the nodes are premisses, logical axioms and derived wffs, not states. But the techniques are similar.)

Soundness and completeness

- Does the syntactic manipulation give results that correspond to the logical consequences?
- **Soundness**
  A proof theory is sound if

  $$\Phi \vdash \alpha \text{ implies } \Phi \models \alpha$$

- **Completeness**
  A proof theory is complete if

  $$\Phi \models \alpha \text{ implies } \Phi \vdash \alpha$$

- The given (Hilbert-style) proof theory is sound and complete for proper axioms that use only $\rightarrow$ and $\neg$.

Do we always want soundness and completeness

- **Completeness**:
  - often sacrificed for efficiency
  - partic if we lose only 'obscure' deductions
- **Soundness**:
  - less often sacrificed
  - but sometimes we develop proof theories that can derive inferences that are only *plausible*
  - compare
    deduction: e.g. $\frac{p \rightarrow q, p}{q}$ 'from cause to effect' $\Rightarrow$ sound
    abduction: e.g. $\frac{p \rightarrow q, q}{p}$ 'from effect to *possible* cause' $\Rightarrow$ not sound

Sacrifices

- Just as often we sacrifice expressiveness and seek soundness and completeness within the less expressive logic.
- And there's another thing to try...