Knowledge Representation

We have already considered the explicit representation of knowledge:
- Most recently for natural language understanding
- For planning, problem solving, vision

We have also considered specific ways in which knowledge is represented:
- In planning: use of a logic-based representation of states and operators (preconditions, add lists, and delete lists)
- In natural language understanding: formal grammar rules, among other things
- In vision: edge-based models of objects
- In state space search: a different representation of states + a (generally) procedural representation of operators.

There are many different knowledge representation formalisms. We will explore one category of these in detail: formal logics.

Formal Logics

A good basis for knowledge representation languages because
- Their purpose is to encode information about the world symbolically
- They provide a mechanism for making inferences – i.e., for reasoning
- There's a significant foundation of information about them (philosophers and mathematicians have thought about them much longer than AI has).

Logics

A logic is a formal system for describing "the world." It consists of
- Syntax
- Semantics (link to the world)
- Proof theory – a set of rules for deducing new facts from existing ones

The inference procedure must be **sound** (or **truth-preserving**). We don't want to be able to infer facts that are not true.
It must also be **complete**. That is, we want to be able to reach all conclusions that should be inferrable.

Propositional Logic – a very simple logic

The basic building blocks of propositional logic are propositions.

Here a symbol represents a fact (or proposition). Another way to say this is that a symbol represents a statement that can be true or false.

“Bob’s car is blue.”
“John is Mary’s uncle.”

are propositions that might be represented by the symbols P and Q.

Note that symbols can represent arbitrarily complex propositions, but we want to keep them as simple as possible, since they cannot be broken down into their constituent parts.

Syntax

Complex sentences can be built from atomic propositions, using the logical connectives:

and	$\wedge$
or	$\vee$
not	$\neg$
implies	$\Rightarrow$
equivalence	$\Leftrightarrow$

Can also parenthesize expressions.

Note that the order of precedence, from highest to lowest is:

$\neg \; \wedge \; \vee \Rightarrow \Leftrightarrow$

Semantics

The meaning of a propositional symbol is what we define it to be. It is true when the fact to which it refers is true.

The meanings (and truth) of more complex sentences are determined by the following rules:

P	Q	$\neg$P	P $\wedge$ Q	P $\vee$ Q	P $\Rightarrow$Q	P $\Leftrightarrow$ Q
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

Proof Theory

Rules of inference for propositional logic include the following.

We use the notation

$$\frac{\alpha}{\beta}$$

to say that β can be derived from α by inference.

Modus Ponens

$$\frac{\alpha \Rightarrow \beta, \alpha}{\beta}$$

Modus Tolens

$$\frac{\alpha \Rightarrow \beta, \neg \beta}{\neg \alpha}$$

And-elimination

$$\frac{\alpha_1 \wedge \alpha_2 \wedge \ldots \alpha_n}{\alpha_i}$$

And-introduction

$$\frac{\alpha_1 , \alpha_2 , \ldots \alpha_n}{\alpha_1 \wedge \alpha_2 \wedge \ldots \alpha_n}$$

Or-introduction

$$\frac{\alpha_i}{\alpha_1 \vee \alpha_2 \vee \ldots \alpha_n}$$

Double negation elimination

$$\frac{\neg\neg\alpha}{\alpha}$$

We will focus on the above inference rules for now and will consider two more later:

Unit resolution

$$\frac{\alpha \vee \beta, \neg \beta}{\alpha}$$

Resolution

$$\frac{\alpha \vee \beta, \neg \beta \vee \gamma}{\alpha \vee \gamma}$$

An Example. No one sleeps on Thursday night.

$Sun \lor Mon \lor Tues \lor Wed \lor Thurs \Rightarrow Work$
$Thurs \lor Fri \lor Sat \Rightarrow Party$
$Party \land Work \Rightarrow \neg Sleep$
$Thurs$

Another Example. Students who eat pizza in the lab late at night are happy.

$InLab \land Late \Rightarrow Hungry$
$Hungry \land Veg \Rightarrow CheesePizza$
$Hungry \land \neg Veg \Rightarrow PepperoniPizza$
$CheesePizza \lor PepperoniPizza \Rightarrow Happy$

$InLab$
$Late$
Veg

Monotonicity

The use of inference rules to draw conclusions relies on a property called monotonicity.

A logic is **monotonic** if when we add some new sentences to the knowledge base (KB), all the sentences entailed by (i.e., that follow from) the original KB are still entailed by the new larger knowledge base.