COSC462 Applied Logic
Lecture 2: Opaque propositional languages

Willem Labuschagne
University of Otago

Abstract

In this lecture we define and explore the class of knowledge representation languages exemplified by the p, q-language. These are the languages of propositional logic, and they all resemble one another, differing only in the choice of basic building blocks from which they are constructed. First we shall say how grammatical sentences are built up (syntax) and then we shall discuss the various uses that may be made of truth values (semantics).

1 Syntax

A language is defined recursively, in the following way. First one specifies some basic building blocks — the atoms of the language. Next one specifies ways in which sentences can be combined to give longer and more complex sentences. For propositional languages we want these combinations to involve the now familiar connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$. Finally, the language as a whole is taken to be the collection of all (and only) the sentences that can be constructed from the atoms by a finite sequence of combinations.

How big is a language? One always ends up with infinitely many sentences. To see this, imagine that someone challenges you to show that the language can have more than k sentences, where your adversary chooses k. Since you can make sentences as long as you like, pick any sentence in the language and apply $\neg$ to it over and over, more than k times. Each application of $\neg$ produces a new sentence, so in the end you've made more than k sentences. Thus the language has infinitely many sentences, irrespective of the number of atoms with which we begin.

Now let us try to express these ideas both concisely and precisely as a definition.

Definition 1 *Let A be any subset of $\{p_0, p_1, p_2, \ldots\}$. The members of A are called* ***atoms****.*

We say that α is a ***sentence*** *over A iff one of the following is the case:*

- $\alpha \in A$
- $\alpha = (\neg\beta)$ *for some previously constructed sentence β over A*
- $\alpha = (\beta * \gamma)$ *where β and γ are previously constructed sentences over A, and $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$. (Here γ is the lowercase Greek letter 'gamma'.)*

The set of all sentences over A is the ***language*** *L_A generated by A. If A is finite, L_A is said to be* ***finitely generated****.*

Informally, L_A consists of the strings that can be built up from the symbols in A by a finite number of steps, where each step consists of either prefixing the symbol $\neg$ or infixing one of the symbols $\wedge, \vee, \rightarrow, \leftrightarrow$ and enclosing the result within parentheses.

Example 2 *Suppose $A = \{p_0, p_1, p_2, \ldots, p_{113}\}$. Then p_{17} is a sentence over A, constructed by doing nothing, i.e. performing 0 steps. Since p_{17} is a previously constructed sentence over A, $(\neg p_{17})$ is also a sentence over A. Since p_{22} and $(\neg p_{17})$ are both 'previously constructed' sentences, $(p_{22} \wedge (\neg p_{17}))$ is also a sentence over A. And so forth. However, p_{222} is not a sentence over A (as p_{222} does not belong to A and cannot be constructed from A by prefixing $\neg$ or infixing any of the other connectives). Furthermore, a string of the form $\neg(\neg \ldots (\neg p_1) \ldots)$ in which there are infinitely many occurrences of $\neg$ is also not a sentence over A, since it would require more than a finite number of steps for its construction from the atoms.*

In practice we tend to leave out parentheses whenever possible, so that the strings are easier to read.

Notation 3 *Let us agree that the negation symbol $\neg$ applies to the shortest grammatically well-formed sentence that follows it, so that we may write $(\neg\alpha)$ as $\neg\alpha$ without ambiguity. Then we shall also feel free to drop parentheses to write $(\alpha * \beta)$ as $\alpha * \beta$ for all $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, provided that it is obvious how to restore parentheses. Finally, if only two or three atoms are required, we shall often omit subscripts and use the symbols p, q, and r.*

Exercise 4 *1. Restore parentheses to those of the following that are unambiguous:*

- $\neg\alpha \wedge \beta$
- $\neg(\alpha \vee \beta)$
- $\alpha \wedge \beta \to \gamma$

2. Design a knowledge representation language for the 3 Card System, in which each of three players gets one of three cards coloured red, green, or blue. In other words, choose a set of atoms representing which player gets which card when the cards are dealt. How would you use the language to express the following ideas:

- *Player 1 gets the red card* ***if*** *player 2 gets the blue card*
- *Player 1 gets the red card* ***if and only if*** *player 2 gets the blue card*
- *Player 1 gets the red card* ***only if*** *player 2 gets the blue card*
- *Player 1 getting the red card is a* ***sufficient condition*** *for player 2 getting the blue card*
- *Player 1 getting the red card is a* ***necessary condition*** *for player 2 getting the blue card.*

2 Semantics

Semantics is concerned with the meaning of words and sentences. Meaning has to do with the relationship between symbolic and iconic representations. An agent observing a system builds up an iconic representation of a state of the system (by forming and combining iconic representations of the microstates of components). For instance, an agent observing the Light-Fan System when its light is on and its fan is off may have an iconic representation that combines an image of a shining lightbulb and an image of motionless fanblades (or a pictorial label for the image, such as the binary string 10 in which the first value represents the microstate of the light and the second value the microstate of the fan.) The agent may then formulate sentences like $p \wedge \neg q$ and use the sentences for reasoning or communicating.

Truth values indicate the fit between sentences and iconic representations. More precisely, something called a *valuation* records the match between sentences and a state. Each valuation is linked to (the iconic representation of) a state, but we shall defer discussion of the details until later.

Definition 5 *Let L_A be the language generated by some set A of atoms.*
*A **valuation** of L_A is a function $v : A \longrightarrow \{0,1\}$.*
The set of all valuations of L_A is denoted by W_A.

A valuation is an assignment of truth values to the atoms of the language. Intuitively, if L_A is designed so that an agent can represent knowledge about some system in it, then the possible states of that system would correspond to (some of the) valuations of the language.

Example 6 *Recall the Light-Fan System with states labelled* 11, 10, 01, *and* 00. *The knowledge representation language for this system had two atoms, p and q. In the state labelled* 11, *the light is on and the fan is on. Thus atom p should get the truth value* 1 *and so should atom q. The state* 11 *therefore corresponds to the valuation v given by $v(p) = 1 = v(q)$. In the same way, state* 10 *corresponds to the valuation v' given by $v'(p) = 1$, $v'(q) = 0$. Similarly state* 01 *corresponds to the valuation v'' where $v''(p) = 0$ and $v''(q) = 1$. Finally the state* 00 *corresponds to v''' where $v'''(p) = v'''(q) = 0$.*

The strings $11, 10, 01, 00$ that proved so convenient as labels for states of the Light-Fan System are really just abbreviations for the corresponding valuations.

Notation 7 *Suppose $A = \{p_0, p_1, p_2, \ldots, p_n\}$. It is convenient to abbreviate a valuation $v : A \longrightarrow \{0,1\}$ by writing the sequence of values $v(p_0)v(p_1)\ldots v(p_n)$. Such abbreviations are of course less convenient in the case of infinite A, and will need supplementary comment.*

Consider some examples in which A is bigger than $\{p, q\}$.

Example 8 *If $A = \{p_0, p_1, p_2, \ldots, p_{113}\}$, the function $v : A \longrightarrow \{0,1\}$ given by $v(p_i) = 1$ if i is even, otherwise $v(p_i) = 0$, is a valuation of L_A. It is possible, though not particularly useful, to abbreviate this valuation by a binary string* $101010\ldots10$. *(Note that $i = 0$ is even.)*

If $A' = \{p_0, p_1, \ldots\}$, the function $v' : A' \longrightarrow \{0,1\}$ given by $v'(p_i) = 1$ if i is even, otherwise $v'(p_i) = 0$, is a valuation of $L_{A'}$. We could abbreviate v' by $101010\ldots$ *if we wanted to and if the context made it clear what the ellipsis meant.*

2.1 Ontologies

A language with n atoms will have 2^n valuations, because we may regard each valuation as a sequence of n choices between 2 possibilities, namely making the atom true or making it false. Of these 2^n valuations, it will sometimes be the case that only some valuations correspond to states of the system. Whereas the Light-Fan system had a perfect correspondence between states and valuations, the following example introduces a system in which the obvious knowledge representation language has many more valuations than the system has states.

Example 9 *Consider the 3 Card System. There are three players. Each player is dealt one of three cards coloured red, green or blue. Such a deal is what we understand by a state of the system.*

A simple knowledge representation language might have 9 atoms, say r_1, r_2, r_3, g_1, g_2, g_3, b_1, b_2, b_3 *where* r_1 *stands for 'Player 1 has the red card',* g_2 *for 'Player 2 has the green card', and so on. The language thus has* $2^9 = 512$ *valuations in the set* W_A*. But there are only* $3 \cdot 2 \cdot 1 = 6$ *states, because there are 3 possible cards from which player 1 is dealt one card, leaving 2 possible cards for player 2, leaving one card for player 3. Thus the set of states of the system corresponds to a very small subset of* W_A*.*

Because the set of states need not be the same size as the set of valuations, we distinguish in our semantics between the set of states, S, and the set of valuations, W_A. Moreover, we do not always assume that $S \subseteq W_A$. To see why, let us return to the 3 Card System.

The states of the system are deals, which we visualise in some way, perhaps as an image of three generic humans each holding a differently coloured card. For ease of communication or recall, we want to label these iconic representations. One possible way to label them is to use valuations, just as we did in the case of the Light-Fan System. It would be possible to take S to be the subset of W_A consisting of the six valuations that each make just one of r_1, r_2, r_3 true, just one of g_1, g_2, g_3 true, and just one of b_1, b_2, b_3 true. But valuations are quite clumsy beasts. To specify a valuation we must say what truth value is assigned to each of nine atoms. The valuation representing the deal in which player 1 got the red card, player 2 the green and player 3 the blue card might be given, in its most economical form, by the binary string 100010001. That's rather awkward to work with and to understand.

There is a mathematically simpler and very obvious way to represent the states, namely by the little strings rgb, rbg, grb, gbr, brg, and bgr. The strings are virtually self-explanatory if we read them from left to right — rgb is the state in which player 1 gets the red, player 2 the green, and player 3 the blue card. And so it is convenient to take S to be the set of these 6 strings rather than to be a set of valuations.

This makes the point that we might want to label states by mathematical objects different from valuations, just for simplicity. Another point worth making is that states and valuations are really fundamentally different kinds of animal.

Imagine three players each holding a different card. This is a state of the 3 Card System, and exists independently of whether there is a language in which to discuss the system. Valuations only arise after we have chosen the vocabulary of the knowledge representation language, specifically the atoms of the language — this is why we denote the set of valuations by writing W_A. If tomorrow we looked at the system and decided to change our selection of relevant features, so that we create a new language based on a different set of atoms, then we might still have the same system with the same states in mind, but each state would be associated with a different valuation.

A final remark on the relationship between states and valuations: when we discuss temporal logic in a later chapter we shall see that we sometimes wish to think of the states of a system as snapshots taken at successive ticks of a clock. If we use the successive instants of time to represent the states, many different states (clock-ticks) may be associated with the same valuation, since the same facts may persist.

Thus there are reasons for having a set of states S that may be different from the set of valuations W_A, and so we need to connect them by saying which state is associated with which valuation.

Definition 10 *Given a system with set of states S and given a language L_A with set of valuations W_A, we may say that L_A is a **knowledge representation language** for the system if we associate with L_A a **labelling function** $V : S \longrightarrow W_A$. The pair (S, V) is called an **ontology** for L_A.*

Example 11 *In the case of the Light-Fan System, the knowledge representation language is L_A with $A = \{p, q\}$, and we use the ontology (S, V) where $S = \{11, 10, 01, 00\} = W_A$ and V is the identity function given by $V(s) = s$ for every $s \in S$.*

Example 12 *In the case of the 3 Card System, we might use the knowledge representation language L_A with $A = \{r_1, r_2, r_3, g_1, g_2, g_3, b_1, b_2, b_3\}$. An appropriate ontology might be (S, V) where $S = \{rgb, rbg, grb, gbr, brg, bgr\}$ and where V is the obvious function that maps*

- *rgb to the valuation $V(rgb)$ that makes r_1 true, g_2 true, b_3 true, and all the other atoms false*
- *rbg to the valuation $V(rbg)$ that makes r_1 true, b_2 true, g_3 true, and all the other atoms false*

- *grb to the valuation $V(grb)$ that makes g_1 true, r_2 true, b_3 true, and all the other atoms false*

- *gbr to the valuation $V(gbr)$ that makes g_1 true, b_2 true, r_3 true, and all the other atoms false*

- *brg to the valuation $V(brg)$ that makes b_1 true, r_2 true, g_3 true, and all the other atoms false*

- *bgr to the valuation $V(bgr)$ that makes b_1 true, g_2 true, r_3 true, and all the other atoms false.*

Here is something to think about. If two human agents grow up in the same town, they build up a shared ontology and can be fairly confident that when they use the same words, they mean the same things. When in doubt, they can point to whatever they mean. But how, in general, can two agents be confident that they know how to decode each other's communications? To some extent, it helps if the agents have similar architectures, with similar sensors extracting similar information from similar environments. It also helps if the agents have similar hardwired information. In some cases, one agent may be able to describe, in sentences, what the states are that constitute her ontology. In other cases, this may not be possible, as we shall see when we discuss the Ineffability Theorem later. Generally speaking, the best way to achieve a shared ontology is for agents to share iconic representations, which requires the agents to move outside language, by, for instance, pointing at things. In some contexts, mathematics may take the place of the real world — the agents may employ an abstract ontology, such as set theory, which can be shared because each agent builds up the universe of sets in the same recursive way.

2.2 Satisfaction and models

Given an ontology (S, V) for L_A, every state $s \in S$ has an associated valuation $V(s) : A \longrightarrow \{0, 1\}$. So, relative to every state s there is a specific way to assign truth values to atoms. The definition of satisfaction describes how truth values may be assigned to the other sentences as well by summarising what we previously said about connectives with the help of truth tables.

Definition 13 ***(Satisfaction)*** *A valuation $v : A \longrightarrow \{0, 1\}$* ***satisfies*** *a sentence $\alpha \in L_A$ (and α is said to be* ***true relative to*** *v) iff one of the following cases applies:*

- *$\alpha \in A$ and the truth value assigned to atom α by the valuation v is 1*

- *$\alpha = \neg\beta$ and v fails to satisfy β*
- *$\alpha = \beta \wedge \gamma$ and v satisfies both β and γ*
- *$\alpha = \beta \vee \gamma$ and v satisfies at least one of β and γ*
- *$\alpha = \beta \to \gamma$ and v satisfies γ, or fails to satisfy β, or does both*
- *$\alpha = \beta \leftrightarrow \gamma$ and v satisfies both of β and γ or else satisfies neither of them.*

Is it clear that for every valuation v and every sentence α, α is either true or false relative to v but not both? Thus our logic is a classical two-valued logic without 'truth-value gaps'.

Now recall that we may not be interested in all valuations but just in those corresponding to states of the system.

Definition 14 *(**Model**) Let (S, V) be an ontology for L_A. A state $s \in S$ satisfies α iff the valuation $V(s)$ satisfies α.*

*If state s satisfies α then s is a **model** of α.*

The subset of S containing all the models of α is denoted by $\mathcal{M}(\alpha)$.

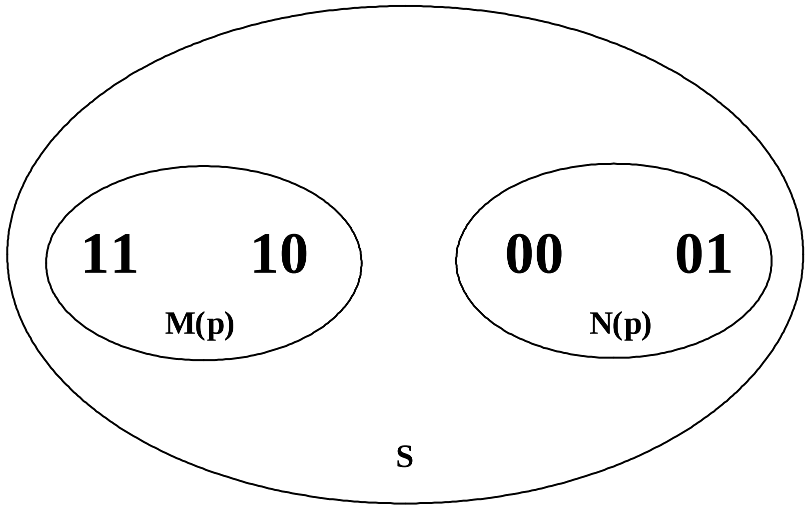
*The set of **nonmodels** of α is $\mathcal{N}(\alpha) = \overline{\mathcal{M}(\alpha)}$ and consists of the states relative to which α is false.*

*If $v \in W_A$ is a valuation that satisfies α, but $v \neq V(s)$ for any $s \in S$, then v is a **spurious** model of α, since v does not correspond to any realisable state of the system (according to the agent's ontology).*

*If $\Gamma \subseteq L_A$ is a **set of sentences**, then a state s is a model of Γ iff s satisfies γ for every $\gamma \in \Gamma$. By $\mathcal{M}(\Gamma)$ we understand the set of all models of Γ. (Here Γ is the uppercase Greek letter 'gamma'.)*

Example 15 *Consider the Light-Fan System, and its knowledge representation language L_A, with $A = \{p, q\}$. As before we take the ontology to be (S, V) where $S = W_A = \{11, 10, 01, 00\}$ and $V(s) = s$ for all $s \in S$. Take $\alpha = p$. The diagram below shows how the set S is divided into the two complementary subsets $\mathcal{M}(p)$ and $\mathcal{N}(p)$ consisting of models of p and nonmodels of p.*

Example 16 *Let's continue with the Light-Fan System and the p, q-language. The sentence $p \to q$ has as its models all the states in $S = \{11, 10, 01, 00\}$ that either satisfy q or fail to satisfy p (or both). State 11 satisfies q, and so is a model of $p \to q$. State 10 neither satisfies q nor fails to satisfy p, and so is not a model of $p \to q$. State 01 both satisfies q and fails to satisfy p, so is certainly a model of $p \to q$. State 00 fails to satisfy p, and thus is a model of $p \to q$. We conclude that $\mathcal{M}(p \to q) = \{11, 01, 00\}$.*



Example 17 *Languages with lots of atoms are just as easy to work with as the p, q-language. Suppose $A = \{p_0, p_1, p_2, \ldots, p_{113}\}$. Suppose further that s is a state whose associated valuation $V(s) = v$, where the valuation $v : A \longrightarrow \{0, 1\}$ is given by $v(p_i) = 1$ if i is even, otherwise $v(p_i) = 0$. Now*

- *s satisfies p_{22}, because $V(s) = v$ and $v(p_{22}) = 1$*
- *s fails to satisfy p_{23}, because $V(s) = v$ and $v(p_{23}) = 0$*
- *s satisfies $p_{23} \rightarrow p_1$ because $V(s)$ fails to satisfy the antecedent p_{23}*
- *s fails to satisfy $p_{23} \leftrightarrow \neg p_1$ since s does not satisfy p_{23} but does satisfy $\neg p_1$.*

Thus $v \in \mathcal{M}(p_{22})$, $v \notin \mathcal{M}(p_{23})$ so that $v \in \mathcal{N}(p_{23})$, $v \in \mathcal{M}(p_{23} \rightarrow p_1)$, and $v \notin \mathcal{M}(p_{23} \leftrightarrow \neg p_1)$ so that $v \in \mathcal{N}(p_{23} \leftrightarrow \neg p_1)$.

Now there are several useful classes of sentences:

Definition 18 *Sentences that have no models are called unsatisfiable or* ***contradictions****.*

Sentences that have at least one model are called ***satisfiable.***

Sentences satisfied by all states in S are called ***valid***

Sentences satisfied by all valuations in W_A are called ***tautologies****.*

Sentences satisfied by some states but not by others are called ***contingent****.*

The question whether a given sentence is satisfiable is the famous Satisfiability Problem discussed in courses on complexity theory — one would like to know that the complicated specification you may have drawn up for a software system does not contain contradictions, and thus that it is satisfiable. The Satisfiability Problem is NP-complete, and no algorithm for solving it is known that is polynomial in the worst case. However, there are several clever programs known as 'SAT-solvers' that exploit human psychology to work very efficiently in the cases that we tend to produce. Visit the website http://alloy.mit.edu for an example of a SAT-solver.

Exercise 19 1. *Consider the Light-Fan System, and its knowledge representation language L_A, with $A = \{p, q\}$. As before we take the ontology to be (S, V) where $S = W_A = \{11, 10, 01, 00\}$ and $V(s) = s$ for all $s \in S$. Write down a sentence having*

- *zero models (i.e. a contradiction)*
- *exactly one model*
- *exactly two models*
- *exactly three models*
- *four models (i.e. a tautology).*

2. *The Light-Fan-Heater System has three components, and the agent observing the system is interested in which components are on and which are not. A suitable knowledge representation language might have three atoms, say p, q, and r, where p expresses that the light is on, q that the fan is on, and r that the fan is on.*

- *Write down the valuations in W_A.*
- *What would be an appropriate ontology (S, V)?*
- *Pick any state, and assume the agent has been able to exclude all the others. Write down a sentence of L_A to express all the agent's information (i.e. a sentence that has the selected state as its only model).*
- *Pick any two states, and assume the agent has been able to exclude all the others. Write down a sentence of L_A to express all the agent's information (i.e. a sentence that has the two selected states as its only models). Now find another sentence of L_A that has the same models.*
- *Suppose the agent's information is expressed by the sentence p. List the states in $\mathcal{M}(p)$.*

- *Suppose the agent's information is expressed by the sentence $p \leftrightarrow r$. List the states in $\mathcal{M}(p \leftrightarrow r)$.*
- *Give an example of a sentence of L_A which is not satisfiable (i.e. a contradiction).*

3. *The 3 Card System has three players who are each dealt a different card coloured red, green or blue. Consider L_A with $A = \{r_1, r_2, r_3, g_1, g_2, g_3, b_1, b_2, b_3\}$ and the ontology (S, V) where $S = \{rgb, rbg, grb, gbr, brg, bgr\}$ and where V is the obvious function that maps, say, rgb to the valuation $V(rgb)$ making r_1 true, g_2 true, b_3 true, and all the other atoms false, and so on.*

   - *Pick any state, and assume the agent has been able to exclude all the others. Write down a sentence of L_A to express all the agent's information (i.e. a sentence that has the selected state as its only model).*
   - *Pick any two states, and assume the agent has been able to exclude all the others. Write down a sentence of L_A to express all the agent's information (i.e. a sentence that has the two selected states as its only models). Now find another sentence of L_A that has the same models.*
   - *Suppose the agent's information is expressed by the sentence $r_1 \rightarrow g_2$. List the states in $\mathcal{M}(r_1 \rightarrow g_2)$.*

2.3 Information and equivalence

Suppose a system has a set S of states.

Definition 20 *(**Information**) The information about the system possessed by an agent is reflected by the selection of a set $\overline{X}$ of **excluded** states inside S, leaving a complementary set X of included states.*

*A sentence α **expresses** the agent's information if $\mathcal{M}(\alpha) = X$, so that $\mathcal{N}(\alpha) = \overline{X}$.*

*We say that $\mathcal{N}(\alpha)$ is the information **content** of α.*

An agent who is able to exclude many states has more information about the actual state of the system than another agent who can exclude only a few states, just as the general who knows the enemy will attack at dawn has more information than the general who knows only that the enemy will attack some time in the next few days. This semantic view of information leads to two important questions:

- Is there some sentence α that expresses the agent's information? We shall discuss this in detail in Lecture 3.

- Given a sentence α that expresses some information, are there other ways to say the same thing?

Definition 21 *(**Equivalence**) Let (S, V) be an ontology for L_A. If $\mathcal{M}(\alpha) = \mathcal{M}(\beta)$, then we say that α and β are **equivalent** sentences, and write $\alpha \equiv \beta$.*

In the language L_A with $A = \{p, q\}$, which we used for the Light-Fan System, the sentences $p \to q$ and $\neg p \vee q$ are equivalent, i.e. $(p \to q) \equiv (\neg p \vee q)$. This follows since $\mathcal{M}(p \to q) = \mathcal{M}(\neg p \vee q)$. To give another example, $p \equiv p \wedge p$, since $\mathcal{M}(p) = \mathcal{M}(p \wedge p)$.

There are a couple of things to note about equivalence.

Firstly, it should be clear that equivalent sentences express the same information about the system of interest, for if $\alpha \equiv \beta$ then $\mathcal{M}(\alpha) = \mathcal{M}(\beta)$ and so $\mathcal{N}(\alpha) = \mathcal{N}(\beta)$, which means that both α and β exclude the same states. In English, we are familiar with paraphrases, for example humorously rendering 'All that glisters is not gold' as 'All that coruscates with effulgence is not *ipso facto* aurous'. What may be surprising is that even our simple knowledge representation languages are rich enough to allow a thought to be expressed in different ways. It is traditional in philosophy to call this 'thought' that can be expressed in different ways a *proposition*. Thus a proposition is just the division of S into subsets X and $\overline{X}$. If a proposition can be expressed by a sentence, then that proposition corresponds to a whole class of equivalent sentences. After all, for any sentence α, $\mathcal{M}(\alpha) = \mathcal{M}(\alpha \wedge \alpha) = \mathcal{M}(\alpha \wedge \alpha \wedge \alpha)$ and so on.

The second thing to note is that $\equiv$ is not a connective of any knowledge representation language L_A, and $\alpha \equiv \beta$ is not a sentence of L_A. The symbol $\equiv$ is an abbreviation for 'is equivalent to', and belongs to the *metalanguage*, which is the language consisting of English plus some mathematics in which we talk about the knowledge representation languages L_A. Don't confuse $\equiv$ with the connective $\leftrightarrow$.

Exercise 22 *1. Consider the Light-Fan System, and its knowledge representation language L_A, with $A = \{p, q\}$. As before we take the ontology to be (S, V) where $S = W_A = \{11, 10, 01, 00\}$ and $V(s) = s$ for all $s \in S$.*

For each of the following pairs of sentences α and β, work out whether $\alpha \equiv \beta$:

- $p \vee (p \vee p)$ *and* $(p \vee p) \vee p$

- p *and* $\neg\neg p$ *(double negations cancel out)*
- $p \vee \neg q$ *and* $q \rightarrow p$
- $p \wedge q$ *and* $(p \rightarrow q) \wedge (q \rightarrow p)$
- $\neg(p \wedge q)$ *and* $\neg p \vee \neg q$ *(an example of what is called a De Morgan identity, telling us how negations changes* $\wedge$ *to* $\vee$*)*
- $\neg(p \vee q)$ *and* $\neg p \wedge \neg q$ *(another De Morgan identity, telling us how negation changes* $\vee$ *to* $\wedge$*)*
- $p \leftrightarrow q$ *and* $(p \rightarrow q) \wedge (q \rightarrow p)$

2. *The Light-Fan-Heater System has three components, and the agent observing the system is interested in which components are on and which are not. A suitable knowledge representation language might have three atoms, say* p*,* q*, and* r *in that order, where* p *expresses that the light is on,* q *that the fan is on, and* r *that the fan is on. We take the ontology to be* (S, V) *where* $S = W_A = \{111, 110, 101, 011, 100, 010, 001, 000\}$ *and* $V(s) = s$ *for all* $s \in S$*.*

   For each of the following pairs α *and* β*, find out whether* $\alpha \equiv \beta$*.*

   - $p \vee q$ *and* $p \vee q \vee r$
   - $p \wedge q$ *and* $(p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r)$
   - $\neg p$ *and* $q \wedge r$

3. *The 3 Card System has three players who are each dealt a different card coloured red, green or blue. Consider* L_A *with* $A = \{r_1, r_2, r_3, g_1, g_2, g_3, b_1, b_2, b_3\}$ *and the ontology* (S, V) *where* $S = \{rgb, rbg, grb, gbr, brg, bgr\}$ *and where* V *is the obvious function that maps, say,* rgb *to the valuation* $V(rgb)$ *making* r_1 *true,* g_2 *true,* b_3 *true, and all the other atoms false, and so on.*

   For each of the following pairs α *and* β*, find out whether* $\alpha \equiv \beta$*.*

   - $r_1 \wedge g_2$ *and* $r_1 \leftrightarrow g_2$
   - $r_1 \rightarrow b_3$ *and* $\neg(r_1 \wedge \neg b_3)$

4. *Consider a language* L_A*. Show that for all sentences* α*,* β*, and* γ *in* L_A *the following equivalences hold:*

   - $\alpha \equiv \neg\neg\alpha$
   - $\alpha \equiv \alpha \vee \alpha$
   - $\alpha \equiv \alpha \wedge \alpha$

- $\alpha \vee \beta \equiv \beta \vee \alpha$
- $\alpha \rightarrow \beta \equiv \neg\alpha \vee \beta$
- $\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$
- $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$
- $\alpha \wedge (\beta \vee \gamma) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ *(a distribution identity)*
- $(\alpha \wedge \beta) \rightarrow \gamma \equiv \neg\alpha \vee \neg\beta \vee \gamma$ *(rewriting a conditional sentence as a 'clause')*
- $\alpha \rightarrow (\beta \wedge \gamma) \equiv (\alpha \rightarrow \beta) \wedge (\alpha \rightarrow \gamma)$

5. *(The difference between $\equiv$ and the connective $\leftrightarrow$)*

   Suppose that a language L_A and an ontology (S, V) are given, where $V : S \longrightarrow W_A$ tells us which valuation is associated with each state.

   For all sentences α and β, prove that α is equivalent to β iff the biconditional sentence $\alpha \leftrightarrow \beta$ is satisfied by every state in S.

   In other words, show that $\alpha \equiv \beta$ is the case if, and only if, the sentence $\alpha \leftrightarrow \beta$ of L_A is valid.

2.4 Entailment

We now come to the most important relationship between sentences in logic. Suppose an agent learns that α is the case. What is the agent now entitled to believe? The sentences that somehow follow from α are called the consequences of α, or the sentences entailed by α. There is more than one criterion[1] that can be used to determine whether a sentence β follows from α, and we will at this stage look only at the most traditional criterion, which defines the 'classical' consequences of α. In a later lecture we will explore the more modern 'defeasible' consequences.

Definition 23 *(**Classical entailment**) Let (S, V) be an ontology for L_A. If $\mathcal{M}(\alpha) \subseteq \mathcal{M}(\beta)$, then we say that α **classically entails** β, or that β is a **classical consequence** of α.*

We write $\alpha \vDash \beta$ as an abbreviation for 'α classically entails β', and we denote by $Cn(\alpha)$ the set of all classical consequences of α (thus $Cn(\alpha) = \{\beta \mid \alpha \vDash \beta\}$).

If Γ is a set of sentences then $\Gamma \vDash \beta$ iff $\mathcal{M}(\Gamma) \subseteq \mathcal{M}(\beta)$.

Example 24 *Consider the Light-Fan System, and its knowledge representation language L_A, with $A = \{p, q\}$. As before we take the ontology to be (S, V) where $S = W_A = \{11, 10, 01, 00\}$ and $V(s) = s$ for all $s \in S$.*

Now $p \wedge q \vDash p$, since $\mathcal{M}(p \wedge q) = \{11\} \subseteq \mathcal{M}(p) = \{11, 10\}$.

[1] Note that "criteria" is plural, just like "phenomena" and "automata". It is a solecism punishable by public disembowelment to speak of "one criteria".

What is the idea behind classical entailment? There are two equally good ways to think of it.

- **Information content**: If $\alpha \vDash \beta$, then $\mathcal{M}(\alpha) \subseteq \mathcal{M}(\beta)$, and so $\mathcal{N}(\beta) \subseteq \mathcal{N}(\alpha)$, where $\mathcal{N}$ denotes the set of nonmodels. The nonmodels of a sentence are the states excluded by that sentence, and thus form the information content of the sentence. So if $\alpha \vDash \beta$, then β expresses a part of the information expressed by α. For example, we know that $p \wedge q \vDash p$, and clearly p expresses part of the information expressed by $p \wedge q$, since $\mathcal{N}(p) = \{01, 00\} \subseteq \mathcal{N}(p \wedge q) = \{10, 01, 00\}$. In this view, the classical consequences of α are all the sentences conveying bits and pieces of the information in α.

- **Conditioning:** Look at S. It need not be the case that a sentence α is satisfied by every $s \in S$. So, in effect, α picks out a subset of S, namely the set $\mathcal{M}(\alpha)$. Now $\alpha \vDash \beta$ tells us that, if we restrict attention to the subset $\mathcal{M}(\alpha)$, then β is satisfied by every s living in this subset. Thus to ask whether $\alpha \vDash \beta$ is to ask whether β is guaranteed to be true in the states picked out by α. The sentence α 'conditions' the question of whether β is true, in a manner that reminds us of the way conditional probability is defined.

What is the usefulness of classical entailment? Well, suppose agent A observes a system, gains some information, and thus is able to exclude some states in S. It is quite possible that agent A will be able to find some sentence α that exactly expresses the information (i.e. such that the nonmodels of α are precisely the excluded states). But if agent A is to communicate with agent B, then she would like to have some alternative to doing a brain dump and telling B absolutely everything she knows. Agent B will usually be interested in only a part of A's information, and would find it tedious and annoying to have to listen to a list of irrelevant facts. Imagine if agent B is the general who wants to know when the enemy will attack. He sent out agent A as a spy, and now A is reporting back. One of the things A saw was a document on which the date and time of the planned attack was written. Would B be pleased if A, before mentioning the document, first spent two hours reporting every barking dog, scuttling hedgehog, and buzzing insect encountered on the way? No. Some way is needed for A to break apart the information acquired by observation so that the most relevant bits can be conveyed to the listener. The classical consequences of α are all the ways in which the various pieces of information in α can be expressed. One of those classical consequences is exactly what A should tell B in order to efficiently communicate the most relevant data.

Exercise 25 *1. Consider the Light-Fan System, and its knowledge representation language L_A, with $A = \{p, q\}$. As before we take the ontology to be (S, V) where $S = W_A = \{11, 10, 01, 00\}$ and $V(s) = s$ for all $s \in S$.*

For each of the following pairs of sentences α and β, determine whether $\alpha \models \beta$ and justify your decision.

- $p \wedge q$ *and* q
- p *and* $p \vee q$
- $p \vee q$ *and* $p \rightarrow q$
- $p \wedge q$ *and* $p \vee \neg q$
- $p \leftrightarrow q$ *and* $p \vee \neg q$
- p *and* $p \rightarrow q$
- p *and* $q \rightarrow p$

2. *Consider the Light-Fan-Heater System and its knowledge representation language with three atoms, p, q, and r, where p expresses that the light is on, q that the fan is on, and r that the heater is on. Take the ontology to be (S, V) where $S = W_A = \{$111, 110, 101, 011, 100, 010, 001, 000$\}$ and $V(s) = s$ for all $s \in S$.*

   For each of the following pairs α and β, find out whether $\alpha \models \beta$.

   - $p \wedge q$ *and* q
   - $p \wedge q$ *and* r
   - $p \vee q \vee r$ *and* $p \vee r$
   - $(p \leftrightarrow q) \leftrightarrow r$ *and* $p \leftrightarrow (q \leftrightarrow r)$
   - $(p \rightarrow q) \rightarrow r$ *and* $p \rightarrow (q \rightarrow r)$

3. *Consider the 3 Card System and its knowledge representation language L_A with $A = \{r_1, r_2, r_3, g_1, g_2, g_3, b_1, b_2, b_3\}$ and the ontology (S, V) where $S = \{rgb, rbg, grb, gbr, brg, bgr\}$ and where V is the obvious function that maps, sya, rgb to the valuation $V(rgb)$ making r_1 true, g_2 true, b_3 true, and all the other atoms false, and so on.*

   For each of the following pairs α and β, find out whether $\alpha \models \beta$.

   - $r_1 \wedge g_2$ *and* $r_1 \leftrightarrow g_2$
   - $r_1 \leftrightarrow g_2$ *and* $r_1 \wedge g_2$

- *$r_1 \rightarrow b_3$ and $\neg(r_1 \wedge \neg b_3)$*

4. *Let α, β, and $\gamma \in L_A$ for some language L_A. Is it necessarily the case that*

   - *$\alpha \models \alpha$? (Reflexivity of $\models$)*
   - *if $\alpha \models \beta$ then $\beta \models \alpha$? (Symmetry of $\models$)*
   - *if $\alpha \models \beta$ and $\beta \models \gamma$ then $\alpha \models \gamma$? (Transitivity of $\models$)*
   - *if $\alpha \models \beta$ then $\neg\beta \models \neg\alpha$? (Contraposition)*
   - *if $\alpha \models \beta$ then $\alpha \wedge \gamma \models \beta$? (Monotonicity)*
   - *if $\alpha \models \beta \vee \gamma$ then $\alpha \models \beta$ or $\alpha \models \gamma$? (Constructivity)*
   - *if $\alpha \not\models \beta$ then $\alpha \models \neg\beta$? (Completeness)*
   - *if $\alpha \wedge \beta \models \gamma$ then $\alpha \models \beta \rightarrow \gamma$? (HHD, or Hard Half of the Deduction Theorem)*
   - *if $\alpha \models \beta \rightarrow \gamma$ then $\alpha \wedge \beta \models \gamma$? (EHD, or Easy Half of the Deduction Theorem)*

5. *(The difference between $\models$ and the connective $\rightarrow$)*

   Suppose that a language L_A and an ontology (S, V) are given, where $V : S \longrightarrow W_A$ tells us which valuation is associated with each state.

   For all sentences α and β, prove that α classically entails β iff $\alpha \rightarrow \beta$ is satisfied by every state in S.

   In other words, show that $\alpha \models \beta$ is the case if, and only if, the sentence $\alpha \rightarrow \beta$ of L_A is valid.

3 Glossary

In this lecture we introduced the following terms and symbols:

- **classical consequence** — a sentence β is a classical consequence of α (or of a set of sentences Γ) if α classically entails β (if Γ classically entails β); we write $Cn(\alpha)$ for the set of all classical consequences of α.

- **classical entailment** — a sentence α classically entails a sentence β iff every model of α is also model of β.

- **contingent** — what we call a sentence that has some models and some nonmodels.

- **contradiction** — what we call a sentence that has no models.

- **contraposition** — an important property possessed by classical entailment, which describes how negating sentences causes the direction of entailment to swing around.

- **equivalence** — the relationship that holds between two sentences having exactly the same models.

- **finitely generated** — a language L_A for which A is a finite set. Such languages have nice properties, as we shall see in Lecture 3.

- **information** — what an agent needs in order to exclude some states.

- **information content** — the information content of a sentence is the set of nonmodels of that sentence, because we may think of the sentence as 'excluding' its nonmodels.

- **knowledge representation language** — a formal language built up from atoms with the help of connectives and linked to a system by an ontology. One can think of such a language as a mathematically concise version of some fragment of the metalanguage which is concerned with describing that system.

- **labelling function** — a function $V : S \longrightarrow W_A$ that tells us which valuation is associated with each state, thus connecting up the system and the knowledge representation language. In effect, a labelling function picks out the meaningful valuations of the language and discards the rest, at least temporarily.

- **metalanguage** — the language in which we are able to talk about the knowledge representation language(s) L_A, for instance English supplemented with various mathematical symbols. We can imagine talking in the metalanguage about a robot who understands only the language L_A.

- **model** — a state $s \in S$ is a model of sentence α iff s satisfies α.

- **monotonicity** — an important property of classical entailment, which says that adding new information will not undermine any entailment that already exists.

- **ontology** — a pair (S, V) consisting of the set S of states of some system and a labelling function $V : S \longrightarrow W_A$. Basically, an ontology tells us what the language can talk about and what the basic facts characterising each state are. In effect, an ontology selects certain valuations of the languages and says "These are the meaningful ones, because they correspond to the states of this system."

- **opaque** — the atoms used so far to build up L_A are 'opaque' in the sense that we cannot open the atom up and see what its parts are. In a language like English the atoms are 'transparent', because the shortest sentences are things like "John loves Mary", which have parts called nouns and verbs. Later on we will look at knowledge representation language in which the atoms are not opaque but built up from things like nouns and verbs.

- **reflexivity of** $\vDash$ — the idea that every sentence classically entails at least itself.

- **proposition** — a thought that can perhaps be expressed by a sentence, in which case it can certainly be expressed in several different ways by different sentences. We take a proposition to be a division of S into two subsets, the excluded and the included states.

- **satisfaction** — a valuation $v \in W_A$ satisfies a sentence α iff v makes α true according to the rules of the definition of satisfaction. The definition basically spells out the way in which connectives affect truth values. A state $s \in S$ satisfies α provided the valuation v associated with s by the labelling function V satisfies α.

- **satisfiable** — a sentence that is not a contradiction.

- **symbols** belonging to the metalanguage — The most important are $\equiv$ for equivalence and $\vDash$ for classical entailment.

- **tautology** — a sentence that cannot possibly be made false, even if another ontology is chosen for the language, because every valuation of the language satisfies it.

- **transitivity of** $\vDash$ — a very useful property of classical entailment. Consider how often planning to achieve a goal involves setting up a chain of subgoals. To fly to Wellington, I need to get to the airport. To get to the airport, I need to call a taxi. To call a taxi I need to find a phone. Transitivity says that if I can get from each step to the next, then I can successfully get from the starting point to the final step, because previous steps do not undermine later steps.

- **valid** — a sentence that, relative to a given ontology, cannot be made false, because all the valuations selected by that ontology satisfy it.

- **valuation** — a function $v : A \longrightarrow \{0, 1\}$ which in effect says which atoms are true and which are false. Every state corresponds to some valuation, but it may be the case that a valuation does not correspond to a possible state of a system.