COSC 410 Lecture 3
The Ineffability Theorem

Willem Labuschagne
University of Otago

2016

Introduction

Information can be expressed either through the words of a language or non-verbally, for example through pictures or mathematical structures. We've called these the symbolic and the semantic ways to represent information. Are they equally good? Let's make this question more precise.

If an agent has information about the state of the system, then this can be expressed symbolically by using a suitable propositional language L_A.

Semantically, the agent's information can be represented by dividing the set S of states into two complementary subsets — the states that the agent is able to exclude on the basis of the available information, and those that remain candidates for being the actual state because the available information does not rule them out.

Instead of talking about two subsets, it is enough to talk about one. Specifying a subset of S is enough to identify its complement too. (In other words, if I tell you which states to exclude, then it will be obvious which states to include, namely all the other states.) So instead of talking about a pair of complementary subsets of S, we'll just talk about a single subset. Which should we choose: the states that are being excluded or the states that are being included? For various reasons it will be convenient to talk of the set of states that are being included.

So now we may think of each subset of S as expressing a different piece of information. Key question: Can all these pieces of information also be expressed by sentences in L_A?

Here's a different way to ask the same question. Is it the case that, for every subset X of S, there is a sentence (or set of sentences) having X as its set of models? If so, then all the relevant information about the system can be expressed symbolically. If not, then some information about the system cannot be expressed in the language, i.e. is ineffable.

The answer turns out to depend on the size of A. However, to arrive at this answer, we need a lemma that requires proof by induction. First we'll explain how inductive proof works, then use it to prove the lemma we need, and finally derive two results about the expressibility of information. One of these results, which we call The Ineffability Theorem, is of special interest, because it is an example of a limitative theorem – a theorem that shows there is something we cannot hope to do. A famous result of this type is known as Gödel's Incompleteness Theorem. Such theorems are usually very hard to prove. Luckily, the Ineffability Theorem is not hard.

Recursive definition and inductive proof

Generating the elements of a set

The usual way to define a set is to specify some property that acts as a membership criterion — things that have the property belong to the set, things that don't have the property live outside the set. For example, an integer may have the property of being even, i.e. of being a multiple of 2, and so we can define a set consisting of precisely the even integers:

$$\{n \mid n = 2k \text{ for some integer } k\}.$$

We use this technique in everyday life too, when we speak of the set of cars purchased in a particular year, or the set of rivers that are longer than 1000 kilometers. The drawback of this kind of definition is that it doesn't give any information about how to find, or build, the elements of the set. There is a different kind of definition that does tell us how to generate the members of a set, and it's called a *recursive* definition. Of course, a recursive definition doesn't suit every kind of set — only those whose members can be built up from basic building blocks.

There are 3 parts to every recursive definition:

- We must give one or more atomic building blocks as a starting point.
- We must say how to construct a new item from previously generated items.
- We must bear in mind that the only items in the set are those that can be constructed from the atoms in finitely many steps — nothing else gets to be a member of the recursively defined set.

Normally we write down just the first two parts and leave the third part implicit.

Example 1 *To give a recursive definition of the set of even integers, think about how you could generate even integers from the simplest even integer, namely* 0. *Even integers can be either positive or negative. The positive even integers can be generated by adding* 2 *to* 0 *enough times, while the negative even integers can be generated by subtracting* 2 *from* 0 *enough times. The following recursive definition captures the idea.*

$$\begin{array}{l} 0 \textit{ is an even integer} \\ \textit{if } k \textit{ is an even integer, so are } k+2 \textit{ and } k-2. \end{array}$$

Notation 1 *To say that an item x belongs to a set X we write $x \in X$. The symbol $\in$ is a stylised Greek epsilon, and is read as saying "is a member of" or "is an element of".*

Example 2 *Consider the alphabet $\{a, b\}$ and let L be the set of palindromes, by which we mean strings over $\{a, b\}$ that are spelt the same forwards and backwards. For example, abba and baab are palindromes because the sequence of letters is unchanged if we read them from right to left instead of from left to right. The simplest of all palindromes is the empty string, which we call λ. The next simplest would be the 1-element strings a and b. From these we can build bigger palindromes by gluing the same letter on at the front and at the back. Here is a recursive definition of L:*

$$\begin{array}{l} \lambda \in L,\ a \in L,\ b \in L \\ \textit{if } x \in L \textit{ then } axa \in L \textit{ and } bxb \in L. \end{array}$$

Example 3 *A propositional language L_A can always be defined recursively. The set A of atomic sentences gives us the building blocks, and it doesn't matter whether A is a finite set like $\{p, q\}$ or an infinite set like $\{p_0, p_1, p_2, \ldots\}$. The connectives allow us to generate new longer sentences from old. The definition of L_A given in lecture 1 is clearly of the form shared by recursive definitions:*

$$\text{if } \alpha \in A \text{ then } \alpha \in L_A$$
$$\text{if } \alpha \in L_A \text{ and } \beta \in L_A \text{ then } \neg\alpha, \alpha \wedge \beta, \alpha \vee \beta, \alpha \to \beta, \alpha \leftrightarrow \beta \in L_A.$$

Bottom up and top down

Having seen some examples of recursive definitions, let's look at the abstract structure of such a definition.

Notation 2 *Suppose we have some set X and we form a new set Y from X by throwing away zero, one, or more of the members of X. Then Y is called a subset of X, which we may abbreviate by writing $Y \subseteq X$.*

It is important not to confuse the two different symbols $\in$ and $\subseteq$. We can say that $a \in \{a, b\}$ but not that $a \subseteq \{a, b\}$, whereas we can say that $\{a\} \subseteq \{a, b\}$ but not that $\{a\} \in \{a, b\}$.

To define a set recursively we must begin with some universal set U of things that are of interest. Then we pick a subset $A \subseteq U$ as our set of atoms. (Usually A will be quite a small subset.) Next we choose some functions on U, for example a function $f : U \longrightarrow U$, as the operation with which we are going to construct things. (We may have several operations.) Finally, our recursive definition describes how to build a subset C of things from the atoms in A with the help of operation f. The recursive definition has the following two parts:

$$A \subseteq C$$
$$\text{if } x \in C \text{ then } f(x) \in C.$$

How does this magic incantation work, exactly? There are two ways to look at it — bottom up and top down.

The bottom up view sees the construction of C happening in stages. Each construction step involves the application of the operation f to whatever items are already available. Initially, we have $C_0 = A$, the items that need 0 applications of f to be in C. Then the bigger subset C_1 consists of all the elements that can be constructed from C_0 using 1 or fewer applications of f, and C_2 includes all those that can be constructed by 2 or fewer applications of f, and so on. This gives us a sequence of stages $C_0 \subseteq C_1 \subseteq C_2 \subseteq \ldots$ Ultimately C consists of all the elements that belong to at least one of the C_i. Technically we could say that C is the union of all the stages C_i.

As a concrete example, let's take U to be the set of all integers, and let $f : U \longrightarrow U$ be given by $f(k) = k + 2$. Take $A = \{1\}$. Now

$$\begin{aligned} C_0 &= \{1\} \\ C_1 &= \{1, 3\} \\ C_2 &= \{1, 3, 5\} \\ C_3 &= \{1, 3, 5, 7\} \end{aligned}$$

and overall $C = \{1, 3, 5, 7, \ldots\}$ because eventually every odd positive integer will appear in some subset C_i.

The top down view requires some new terminology. Suppose we have chosen our universal set U, set of atoms A, and operation $f : U \longrightarrow U$. Let's say that a subset $X \subseteq U$ is *closed* under the operation f if X has the property that whenever $x \in X$ then $f(x) \in X$ too. And let us say that X is an *inductive* subset of U if $A \subseteq X$ and X is closed under f. Now think of the recursively defined set C of comprising those items that belong to every inductive subset of U. Technically, this means the set C is the intersection of all inductive subsets of U.

The top-down view tells us that the recursively defined set C is the *smallest* inductive set because if X is any inductive subset of U then $C \subseteq X$. This ensures that C contains only the items that can be constructed from A by applying the operation finitely many times, and nothing extra can sneak into the set.

Can we be sure that this makes sense? What if there are no inductive subsets of U? Well, don't worry. There is always at least one inductive subset of U, namely U itself. After all, $A \subseteq U$ and U is closed under $f : U \longrightarrow U$ because we chose f that way.

As a concrete example, let's again take U to be the set of all integers with $f(k) = k + 2$ and $A = \{1\}$. We have seen a bottom up view of the recursively defined set of odd positive integers

$$C = \{1, 3, 5, \ldots\}$$

Now notice that C is an inductive subset of U because $A \subseteq C$ (since $1 \in C$) and if $k \in C$ then $f(k) = k + 2 \in C$. In addition to C there are many other inductive subsets of U. We know U itself is inductive. More interesting is

$$\begin{aligned} X &= \{\ldots, -3, -1, 1, 3, 5, \ldots\} \\ &= \{k \mid k \textit{ is an odd integer } \}. \end{aligned}$$

To see that X is inductive, notice that $1 \in X$ and also that if $k \in X$ then $f(k) = k + 2 \in X$. The inductive sets U and X both contain extra undesirable elements such as negative integers which are not wanted in our recursively defined set C. By taking only the items common to all the inductive subsets, we strip away these unwanted elements.

Induction

The top down view of recursion gives us the insight we need for the proof technique of induction. Suppose C is recursively defined from $A \subseteq U$ by the operation $f : U \longrightarrow U$. Then we know that C is the smallest inductive subset of U.

Now suppose we want to prove that all the elements of C have some property P. Then all we have to do is to show that the set $X = \{x \mid x \in U \text{ and } x \text{ has property P } \}$ is inductive. It will immediately follow that $C \subseteq X$ and therefore all the elements of C will have property P. So what do we have to do? Basically, we apply two tests to X:

- Check that $A \subseteq X$.
- Check that if $x \in X$ then $f(x) \in X$. (In this test, the assumption that $x \in X$ is called the induction hypothesis.)

Proving the key lemma

A lemma is an auxiliary result that helps us to prove what we're really interested in. We're really interested in saying something about the expressibility of information, but to say it we need to know something technical about the atomic sentences that are relevant when we want to calculate the truth value of a given sentence.

Let L_A be any propositional language, and recall that we defined L_A recursively in example 3 above.

We want to show that every sentence $\varphi \in L_A$ has a certain property that we'll spell out below. We're going to proceed by defining X to be the subset of L_A consisting of all sentences with the property, and then we'll show that X is inductive. To show that X is inductive, first we show that $A \subseteq X$ and then we show that if $\varphi \in X$ and $\psi \in X$ then $\neg\varphi \in X$ and $\varphi \wedge \psi \in X$ and $\varphi \vee \psi \in X$ and $\varphi \rightarrow \psi \in X$ and $\varphi \leftrightarrow \psi \in X$.

Simple in concept, although a bit tedious in practice because we have not one operation but five to deal with.

The property of interest to our proof is that the truth value a sentence α gets from a state will depend only on the truth values the atomic sentences occurring inside α get from that state, and will be unaffected by the truth values assigned to any other atomic sentences.

Notation 3 *Abbreviate "if and only if" by "iff".*

Definition 1 *If two states s and s' give the same truth value to an atomic sentence p, we'll say that s and s' agree on p.*

Lemma 1 *Let $\alpha \in L_A$ and suppose $s, s' \in S$ are states that agree on all the atomic sentences occurring in α, i.e. for every atomic sentence p that occurs in α, s satisfies p iff s' satisfies p.*

Then s and s' agree on α, i.e. s satisfies α iff s' satisfies α.

Proof. *Take X to be the subset of L_A consisting of all sentences β with the following property: if two states s and s' agree on all the atomic sentences occurring in β then s and s' agree on β.*

So, for each sentence β, we look at every pair of states s and s' that agree on each of the atoms in β, and then we put β into X if every such pair also agrees on β.

Is $A \subseteq X$? Yes, for if $p \in A$ then the only atom occurring in the string p is p itself, so if s and s' agree on p, then obviously s satisfies p iff s' satisfies p, so that s and s' agree on p.

Suppose $\varphi \in X$. (So we know that if s and s' agree on the atomic sentences in φ, they will agree on φ itself. This is our induction hypothesis.)

Is $\neg\varphi \in X$? Well, suppose s and s' agree on the atomic sentences in $\neg\varphi$. The atomic sentences in $\neg\varphi$ are exactly the same as the atomic sentences in φ, so by the induction hypothesis s satisfies φ iff s' satisfies φ, which means that s satisfies $\neg\varphi$ iff s' satisfies $\neg\varphi$. Hence $\neg\varphi \in X$.

Suppose $\varphi \in X$ and $\psi \in X$. (Again our induction hypothesis.)

Is $\varphi \wedge \psi \in X$? Well, suppose s and s' agree on the atomic sentences in $\varphi \wedge \psi$. Then s and s' agree on all the atomic sentences that occur in φ and on all the atomic sentences that occur in ψ. By our induction hypothesis, s satisfies φ iff s' satisfies φ, and s satisfies ψ iff s' satisfies ψ. If s satisfies $\varphi \wedge \psi$ then s satisfies both φ and ψ and so s' satisfies both φ and ψ and thus s' satisfies $\varphi \wedge \psi$ too. Conversely, if s' satisfies $\varphi \wedge \psi$ then s' satisfies both φ and ψ and so s

satisfies both and thus s satisfies $\varphi \wedge \psi$ also. We've shown that s satisfies $\varphi \wedge \psi$ iff s' satisfies $\varphi \wedge \psi$. Hence s and s' agree on $\varphi \wedge \psi$. So $\varphi \wedge \psi \in X$.

Is $\varphi \vee \psi \in X$? Well, if s and s' agree on the atomic sentences in $\varphi \vee \psi$ then they agree on all the atomic sentences in φ and in ψ so by the induction hypothesis they agree on the sentences φ and ψ themselves. Now I claim s satisfies $\varphi \vee \psi$ iff s' satisfies $\varphi \vee \psi$. If this were not so, then one of the states satisfies $\varphi \vee \psi$ while the other doesn't. Now the latter state makes both φ and ψ false whereas the other makes at least one of φ and ψ true. But this is impossible. Thus s satisfies $\varphi \vee \psi$ iff s' satisfies $\varphi \vee \psi$ and so $\varphi \vee \psi \in X$.

Is $\varphi \to \psi \in X$? Well, if s and s' agree on the atomic sentences in $\varphi \to \psi$ then they agree on all the atomic sentences in φ and in ψ so by the induction hypothesis they agree on the sentences φ and ψ themselves. If s makes $\varphi \to \psi$ false then s satisfies φ but fails to satisfy ψ. Because s' agrees with s on φ and ψ, this means s' also satisfies φ but fails to satisfy ψ, i.e. s' also makes $\varphi \to \psi$ false. Conversely, if s' makes $\varphi \to \psi$ false then so does s, by a similar argument. Thus s fails to satisfy $\varphi \to \psi$ iff s' fails to satisfy $\varphi \to \psi$, which means s satisfies $\varphi \to \psi$ iff s' satisfies $\varphi \to \psi$. Hence $\varphi \to \psi \in X$.

Finally, is $\varphi \leftrightarrow \psi \in X$? Well, if s and s' agree on the atomic sentences in $\varphi \leftrightarrow \psi$ then s and s' will agree on φ and on ψ by the induction hypothesis. Assume s satisfies $\varphi \leftrightarrow \psi$. Then s satisfies both φ and ψ or satisfies neither of them. So s' will satisfy both φ and ψ or neither of them, i.e. s' will satisfy $\varphi \leftrightarrow \psi$. By a similar argument, if s' satisfies $\varphi \leftrightarrow \psi$ then so does s. Hence $\varphi \leftrightarrow \psi \in X$. ■

Expressibility

To keep things simple, let us assume for the remainder of this lecture that $S = W_A$, so that states and truth assignments are the same.

We will distinguish between the case where the language L_A has a finite set of atomic sentences and the case in which L_A has an infinite set of atomic sentences.

We will show that in the finite case all semantic information can be expressed in the language but in the infinite case not.

The finite case

Suppose L_A has a finite set $A = \{p_0, \ldots, p_{n-1}\}$ of n atoms. Then L_A has a finite set W_A of 2^n different truth assignments (since each truth assignment is obtained by deciding, for each of the n atoms in turn, whether it is true or false).

As we said before, the information possessed by an agent can be represented semantically by dividing $S = W_A$ into two complementary subsets, one consisting of the states that are excluded by the information and the other consisting of the states that are not ruled out by the information. It is enough to specify one of a pair of complementary subsets, and we shall focus our attention on the set of states that are not excluded by the available information, i.e. the included states.

There are a finite number of subsets of W_A, in fact exactly 2^{2^n} subsets (because each subset is obtained by deciding, for each truth assignment in turn, whether it is in the subset or not, and there are 2^n truth assignments in W_A).

Each of these subsets might be taken to be the set of included states. So semantically an agent might have any of 2^{2^n} pieces of information. But how many of these pieces of information can the agent communicate by sentences of the knowledge representation language?

The information represented by a set of included states can be expressed by a sentence if the sentence has exactly those included states as its models. When A is finite, we can always find sentences expressing each piece of semantic information. To see this takes a bit of work.

Definition 2 *(Normal form)*

- *A sentence α is in disjunctive normal form if α is of the form $\gamma_1 \vee \ldots \vee \gamma_m$ for some m, where each disjunct γ_i is itself a conjunction of the form $\beta_{i1} \wedge \ldots \wedge \beta_{ik}$ for some k, and where every conjunct β_{ij} is either an atom or the negation of an atom.*
- *If the language has a finite set $A = \{p_0, p_1, \ldots, p_{n-1}\}$ of atoms, then a sentence β is a state description iff $\beta = \beta_0 \wedge \beta_1 \wedge \ldots \wedge \beta_{n-1}$ where each β_i is either p_i or $\neg p_i$.*
- *If $A = \{p_0, \ldots, p_{n-1}\}$ then a sentence α is in strong disjunctive normal form (SDNF) if α is in disjunctive normal form and every disjunct is a state description (i.e. every atom in A appears exactly once in every disjunct γ_i).*

The idea is simpler than it sounds. For example, if $A = \{p_0, p_1, p_2\}$ then $p_0 \wedge \neg p_1 \wedge p_2$ is a state description, and it describes the state 101. A sentence in SDNF is just a disjunction of one or more state descriptions (which is why we may read SDNF as "state description normal form" if we wish).

Theorem 1 *Every sentence α of L_A is equivalent to a sentence in disjunctive normal form.*

If A is finite and α is satisfiable, then α is equivalent to a sentence in SDNF.

Proof. *Let $\alpha \in L_A$. We build the normal form sentence by looking at $Mod(\alpha)$.*

If $Mod(\alpha) = \varnothing$ (the empty set), then no state satisfies α, and so we choose as the equivalent sentence a contradiction, say $p_0 \wedge \neg p_0$. Note that the contradiction is in disjunctive normal form, although it has only a single disjunct, so that the symbol $\vee$ does not actually appear.

If $Mod(\alpha) \neq \varnothing$, suppose the atoms occurring in α are among $\{p_0, \ldots, p_k\}$. (After all, only a finite number of atoms can go into the construction of α.) If A itself is finite, we may take $\{p_0, \ldots, p_k\} = A$.

By the lemma, we don't have to care whether states in $Mod(\alpha)$ satisfy atoms outside $\{p_0, \ldots, p_k\}$. What we care about is the finite number of different ways (say m) in which the truth assignments in $Mod(\alpha)$ can assign truth values to the atoms in $\{p_0, \ldots, p_k\}$.

Let $(x_0, x_1, \ldots, x_k)$ be what one of these truth assignments does to the atoms. In other words, pick some truth assignment that satisfies α, and list the truth values it assigns to the atoms $p_0, \ldots, p_k$. So $x_0 = 1$ if the truth assignment satisfies p_0 but $x_0 = 0$ if it doesn't, and so on for $x_1, x_2, \ldots, x_k$.

This allows us to form a conjunction that imitates the truth assignment, namely the sentence $\gamma_1 = \beta_0 \wedge \ldots \wedge \beta_k$ where $\beta_i = p_i$ if $x_i = 1$ and $\beta_i = \neg p_i$ if $x_i = 0$.

We repeat this for each way of allocating truth values to $p_0, \ldots, p_k$ that makes α true, thereby producing conjunctions $\gamma_1, \gamma_2, \ldots, \gamma_m$.

Finally we claim that $\alpha \equiv \gamma_1 \vee \ldots \vee \gamma_m$. To see that this must be the case, note that every truth assignment that satisfies α must also satisfy one of the γ_i. Conversely, every truth assignment *that satisfies one of the γ_i must allocate truth values to $p_0, \ldots, p_k$ in one of the m ways that satisfy α.*

Note that if A is finite, so that $A = \{p_0, \ldots, p_k\}$, then each γ_i is a state description, since every atom appears either unnegated or negated in the conjunction making up γ_i. Hence the disjunction of the γ_i is a sentence in SDNF. ■

Now let us pause to reflect on the significance of the theorem. The obvious thing it tells us is that we can rewrite any sentence in disjunctive normal form. But it also tells us much more.

Instead of starting with the set $Mod(\alpha)$, we could start with any set $X \subseteq W_A$ of truth assignments — say, the set of states not ruled out by the agent's available information. As long as we are able to restrict consideration to some finite set of atomic sentences, we will be able to duplicate the construction in the proof to arrive at a sentence in disjunctive normal form which is satisfied by exactly the states in X. How can we justify restricting our attention to some finite set of atomic sentences? Well, we may no longer have a starting sentence α to serve as focus. But if we have a set A that is itself finite, then clearly we have no need to consider more than finitely many atomic sentences. So we get:

Corollary 2 *If A is finite and if X is any set of included states with complementary set $S - X$ of excluded states, then there is a sentence $\gamma_1 \vee \ldots \vee \gamma_m$ in SDNF such that $Mod(\gamma_1 \vee \ldots \vee \gamma_m) = X$.*

Thus, if A is finite then any information (reflected by a division of $S = W_A$ into two subsets of included and excluded states) can be expressed by a sentence of the language. If A is finite then the agent can always say what she thinks.

The infinite case

The infinite case is different — if A is infinite, there exist subsets $X \subseteq S$ for which no sentence α can be found such that $Mod(\alpha) = X$, and an agent may therefore have information that cannot be expressed symbolically in the logic language.

You may wonder whether the problem is just that we are restricting ourselves to a single sentence α, and would disappear if we allowed ourselves a set of sentences, say Γ. After all, it is easy to define what we mean by the set of models of a set Γ of sentences: $Mod(\Gamma) = \{s \mid s \in Mod(\gamma)$ *for every* $\gamma \in \Gamma\}$. Well, it turns out not to matter whether we stick to a single sentence or allow ourselves the freedom of taking a set of sentences — the proof we give below easily takes care of sets of sentences as well.

This brings us to the first limitative theorem of logic, which I like to call the Ineffability Theorem. A concept is *ineffable* if we cannot put it into words. Let's sharpen up that definition a tad.

Definition 3 ***(Ineffability)*** *A subset $X \subseteq W_A$ is called ineffable if there is no sentence α of L_A such that $Mod(\alpha) = X$.*

Theorem 3 ***(Ineffability Theorem)*** *Let $A = \{p_0, p_1, \ldots\}$. Then there exists an ineffable set X of states.*

Proof. *Pick any truth assignment $w \in W_A$. For example, we could take w to be such that $w(p_i) = 1$ for all $p_i \in A$.*

Let X be the set consisting of all the remaining truth assignments, i.e. $X = W_A - \{w\}$.

We claim that there is no sentence α of L_A such that $Mod(\alpha) = X$.

If there were such an α, it would have to be satisfied by every truth assignment in X but not satisfied by w.

Take any sentence $\alpha \in L_A$. We show that if α is satisfied by all the truth assignments in X then α is satisfied by w as well.

Suppose α is satisfied by all the states in X.

Let $\{p_0, \ldots, p_k\}$ be a subset of A large enough so that every atom in α is among $p_0, \ldots, p_k$.

Let $v \in X$ be the truth assignment such that $v(p_i) = w(p_i)$ for all $i \leq k$ and $v(p_i) \neq w(p_i)$ for all $i > k$.

Now v and w agree on the atoms in α so v and w must agree on α itself (by the lemma we proved previously). Hence v satisfies α iff w satisfies α.

And so we are faced by two choices: either v does not satisfy α, contradicting our choice of α, or else w also satisfies α.

Thus there is no sentence satisfied only by the valuations in X. ■

The Ineffability Theorem was proved in Brink C and Heidema J (1989): A verisimilar ordering of propositional theories: The infinite case, TR-ARP-1/89 (Technical Report Series of the Automated Reasoning Project, Research School of Social Sciences) Canberra: Australian National University. A more general account of related matters is given in Peppas P, Foo N and Williams M-A (1992): On the expressibility of propositions, *Logique et Analyse* 139-140:251-272.

You may care to reflect on this theorem and its broader significance. People say a picture is worth a thousand words. The theorem says that some pictures can't be put into words at all, at least not completely. The cognitive scientist Stevan Harnad expressed a similar insight:

> Words obviously fall short when they leave out some critical feature that would be necessary to sort some future or potential anomalous instance; but even if one supposes that every critical feature anyone would ever care to mention has been mentioned, a description will always remain essentially incomplete in the following ways:
>
> (a) A description cannot convey the qualitative nature of the object being described (i.e. it cannot yield knowledge by acquaintance), although it can converge on it as closely as the describer's descriptive resources and resourcefulness allow. (Critical here will be the prior repertoire of direct experiences and atomic labels on which the descriptions can draw.)
>
> (b) There will always remain inherent features of the object that will require further discourse to point out; an example would be a scene that one had neglected to mention was composed of a prime number of distinct colors.
>
> (c) In the same vein, there would be all the known and yet-to-be-discovered properties of the prime numbers that one could speak of — all of them entailed by the properties of the picture, all of them candidates (albeit far-fetched ones) for further discourse "about" the picture.
>
> (d) Finally, and most revealingly, there are the inexhaustible shortcomings of words exemplified by all the iterative afterthoughts made possible by, say, negation: for example, "the number of limbs is *not* two [. . .]" The truth of all these potential descriptions is inherent in the picture, yet it is obvious that no exhaustive description would be possible. Hence all descriptions will only approximate a true, complete "description".

Of course, the Ineffability Theorem makes a key assumption — that A is infinite. As long as we are interested only in a specific finite number of basic facts about the system (which would lead us to build a representation language with a finite number of atoms each expressing one of those basic facts), we will indeed be able to express our ideas in L_A.

Exercises

Quiz: The quiz question during lecture 4 will ask for a sentence in SDNF equivalent to one of the sentences in exercise 2 below.

1. Consider the Traffic System with $S = \{11, 10, 01, 00\}$ and $A = \{p, q\}$. For each of the following sets of states, give a sentence in SDNF whose set of models coincides with it:
   - $\{11\}$
   - $\{11, 10\}$
   - $\{11, 10, 01\}$
   - $\{11, 10, 01, 00\}$
2. Let us say that a sentence α is in **conjunctive** normal form if $\alpha = \gamma_1 \wedge \ldots \wedge \gamma_m$ for some m, where each conjunct γ_i is of the form $\beta_0 \vee \ldots \vee \beta_k$ for some k, and where every β_i is either an atom or the negation of an atom.

   Suppose $A = \{p_0, p_1, p_2\}$. For each of the following, give two equivalent sentences, one in disjunctive normal form and the other in conjunctive normal form:
   - p_0
   - $\neg p_2$
   - $p_0 \vee p_1$
   - $\neg(p_0 \vee p_1)$
   - $p_1 \wedge p_2$
   - $p_0 \leftrightarrow p_2$
   - $((\neg p_0) \vee p_1) \to p_2$
   - $(p_0 \to p_1) \to p_2$
3. We know that the Ineffability Theorem does not hold for finitely generated languages. Take $A = \{p_0, \ldots, p_n\}$. Trace through the proof of the Ineffability Theorem and find out where it breaks down.
4. Suppose the agent is a mathematician contemplating the set of natural numbers $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$. And suppose she is interested in whether a number is prime or not. As knowledge representation language take L_A where $A = \{p_0, p_1, p_2, \ldots\}$ is the infinite set of atoms in which p_i says "i is prime". Of course, in the 'actual state' of the system, some of these atoms will be true and others false. But for the moment focus on something else. Take $S = W_A$, the set of all truth assignments. Give an example of an idea the agent would be unable to express in L_A. Prove that the idea cannot be expressed in L_A.

(Hint: Try "Not all the natural numbers are prime".)