

CHAPTER 1

SENTENTIAL LOGIC: SYNTAX

Sentential logic (*SL*) is a system of logic that treats statements, or propositions, as fundamental units.¹ *SL* is not just a system of logic, however, it is also a language. There are two very general types of languages, natural and formal. Natural languages are the languages that we ordinarily use to communicate with each other. Examples of natural languages include English, French, German, Mandarin, etc. Computer programming languages like Python and Java, on the other hand, are formal languages.

There are certain features that every language, natural or formal, must have. Those features govern how expressions are formed in the language, and determine what those expressions mean. Let's see how those features work in the English language. It begins with a character set, which includes the twenty-six characters of the Latin alphabet and various punctuation marks. The characters are put together to form words; the meaning of a word is determined by its definition. These words are combined in various ways to form sentences. Not just any string of English words can be a meaningful sentence in English. There are certain rules that must be followed—articles must precede nouns, for instance. The characters and the formation rules are the syntax of the language, the rules that determine meaning are the semantics. *SL* is a language, and, like any other language, it has a syntax and semantics. This chapter will focus on the syntax; the semantics of *SL* will be the subject of the next chapter.

Natural languages have many advantages over formal languages, the greatest of which is their expressive power. This expressive power, however, can also be

¹Informally, we use 'proposition' and 'statement' interchangeably. Strictly speaking, the proposition is the content, or meaning, that the statement expresses. When different sentences in different languages mean the same thing, it is because they express the same proposition. Sentential logic is also called 'propositional logic.'

a disadvantage. It is possible to express more than one thought with exactly the same sentence using a natural language. For example, consider the following English sentence:

“I don’t beat my dog.”

Now, imagine that same sentence uttered several times, stressing a different word each time. “I don’t beat my *dog*” means something very different from “I don’t beat *my* dog.” This is an example of ambiguity, a word or an expression having more than one meaning. Another feature of natural languages is vagueness; an expression is vague when it doesn’t have a precise meaning. Think of vague expressions as those having fuzzy boundaries that make it impossible to draw a precise line between the times that the expression is true of something and the times when it is not. Examples of vague terms are rich, bald, and young. Formal languages eliminate, as much as possible, ambiguity and vagueness. Every term in a formal language should have exactly one meaning and that meaning should be precise.

Every language has ways of using sentences in the language to create other sentences in the language. This is done by applying sentence formation operators or connectives. One way to do this is by joining two sentences with terms like ‘and’, ‘or’, ‘but’, ‘however’, or ‘unless’. So, we can make the sentence

All of Plato’s know works are dialogues, but none of Aristotle’s known works are dialogues.

by putting ‘but’ between ‘All of Plato’s know works are dialogues’ and ‘None of Aristotle’s known works are dialogues.’

Although we call these sentence-forming operators “connectives,” it is important to understand that some of them don’t actually connect different sentences. Some of them apply to just one sentence. Examples of these are ‘it is not the case that’, and ‘it is possible that’.

All of the connectives in *SL* will be truth-functional connectives. A connective is truth-functional if the truth value of the resulting sentence is determined completely by the truth values of the connected sentences and the definition of the connective. So, if we know the truth values of the connected sentences and the operator used to connect them, we will always be able to determine the truth value of the resulting sentence.

Formal languages are used for many things, but an important use in logic is analyzing the logical relationships between sentences and sets of sentences in natural languages, in our case, English. Doing this with *SL* will require translating sentences from English into *SL* and vice versa. Later, I’ll state the syntax of *SL* in a very precise

way. For now, let me introduce the language by discussing how English is translated into *SL*.

1.1 *SL* TRANSLATIONS

Upper-case letters of the Latin alphabet are used to symbolize simple, or atomic, English sentences. Remember that a simple sentence is one that contains no other sentence as a component. So, ‘Socrates is a philosopher’ is a simple sentence, but ‘Socrates is a philosopher, but Cicero is an orator’ is not, since it contains two sentences as components.

Any letter can be used to symbolize a sentence, but we should pick one that will help us to remember the English sentence that is translated. For example, the English sentence

Socrates is a philosopher

could be symbolized with

S

Of course, we only have twenty-six upper-case letters, but we need a potentially infinite supply. So, we’ll allow letters with positive integer subscripts to be used as sentences in *SL*. So, these are all simple sentences in *SL*:

A, B, C, P, Q, Z, D₁, F₄₁₂

One could use a sentence letter to symbolize a complex, or molecular, sentence of English, but it’s not a good idea. By doing so, we would be hiding some of the logical structure of the sentence, run the risk of translating a valid argument into an invalid argument in *SL*. So, be sure to only use letters for the simple sentences.

CONJUNCTIONS

Our first kind of complex sentence is the conjunction. Conjunctions are sentences formed by combining two sentences with a conjunction connective. The sentences that are joined are called ‘conjuncts’. Conjunctions are true whenever both of the conjuncts are true. Although the most common conjunction operator in English is ‘and’, any operator that implies that both of the connected sentences are true is a conjunction operator. The list of English conjunction operators includes

and
but
also
however
yet
still
moreover
although
nevertheless
both

The two most common ways of symbolizing a conjunction operator are ‘&’ (ampersand), ‘·’ (dot), and ‘ \wedge ’ (wedge). We’ll use ‘&’.

Now, we can translate our sentence

Socrates is a philosopher, but Cicero is an orator

as

S & C

and

Plato is from Athens and Aristotle is from Macedonia

can be translated as

A & M

Most of the time, these translations are simple. Unfortunately, the complexity of natural languages can sometimes result in some trick translation issues. There are times when the conjunction operator doesn’t appear to be joining two sentences. Here is an example:

Plato and Aristotle are philosophers.

The best thing to do in these cases is to paraphrase the English sentence into a sentence that joins two statements with the conjunction operator. If the resulting paraphrase is true under exactly the same conditions as the original, then they are equivalent. If so, then simply translate the paraphrase into *SL*.

The paraphrase of the preceding sentence would be

Plato is a philosopher and Aristotle is a philosopher.

and the *SL* translation is

$P \ \& \ A$

On the other hand, consider this sentence:

Abbot and Costello made a good team.

Paraphrasing this results in

Abbot made a good team and Costello made a good team.

This can't be right though. Neither one, individually, made a team at all. So, this sentence, even though it contains the word 'and', is a simple sentence.

The list of English words that gets translated as a conjunction shows that the *SL* translations often fail to capture the full sense of the original English sentences. When I say, "Although John studied hard for his logic exam, he failed", I don't just mean to say that John studied hard *and* he failed, which is how we would paraphrase it before symbolizing the sentence in *SL*. Instead, I mean to say that *in spite* of his studying, he failed. Some nuance is lost in the translation. That nuance is not important however, for our purposes. The conjunction captures everything that is important for logical analysis.

DISJUNCTIONS

Another type of complex sentence is the disjunction, commonly expressed in English with 'or', as in this sentence:

Either the Democrat will win the election or the Republican will win.

Disjuncts are true whenever at least one of the connected sentences, called disjuncts, are true. They are only false when both disjuncts are false. The symbol ' \vee ' (wedge) is used for disjunctions in *SL*. So, the sentence above is translated

$$D \vee R$$

As was the case with the conjunction, disjunctions in English are not always two complete sentences joined with an ‘or’. The example above is more likely to be stated, “Either the Democrat or the Republican will win.” Sometimes, disjunctions in English don’t use ‘or’ at all. Here’s an example:

Alice and Bob are running a race tomorrow. At least one of them will finish.

That second sentence can be paraphrased

Alice will finish the race, or Bob will finish the race.

It can then be symbolized

$$A \vee B$$

A very important thing to remember when translating disjunctions into *SL*, is that the English word ‘unless’ express a disjunction. The sentence,

Charlie will fail logic unless he drops the course before the deadline

is equivalent to

Either Charlie will fail logic or he drops the course before the deadline

and symbolized as

$$F \vee D$$

One thing that is tricky about disjunctions is distinguishing inclusive disjunctions from exclusive disjunctions. An inclusive disjunction is true when *at least one* of the disjuncts is true. An exclusive disjunction is true when one, *and only one*, of the disjuncts is true. The *SL* symbol ‘ \vee ’ is always an inclusive disjunction. In English, however, the word ‘or’ is often used to express exclusive disjunctions. For example, you might see this in a menu:

Your meal comes with either baked potato or french fries.

The probably don’t mean “Your meal comes with either baked potato, french fries, or both.” So, this sentence should *not* be translated as

$$P \vee F$$

Later, we will see how to translate exclusive disjunctions in *SL*.

NEGATIONS

The English phrase ‘it is not the case that’ generates a kind of complex sentence called a negation. Negations are true if the negated sentences are false. For example,

It is not the case that Aristotle was from Athens.

This sentence is true just in case it is false that Aristotle was from Athens, and it would be true if it were the case that Aristotle was not from Athens.

The symbol for negations is ‘ \neg ’. A sentence is negated by placing the negation operator in front of the sentence. This is the one logical connective that doesn’t connect two sentences. So, the negation is known as a “unary connective”, and the others are called “binary connectives.” The phrase ‘it is not the case at’ is convenient to use when paraphrasing negations because it precedes negated sentences in the same way that the negation symbol in *SL* does. Most of the time, though, ‘not’ will be somewhere in the English sentence, like this:

Aristotle was not from Athens.

This is naturally equivalent to ‘It is not the case that Aristotle was from Athens’ and is translated in *SL* as

$\neg A$

Other ways that negations are expressed in English besides ‘it is not the case that’ and ‘not’ are prefixes like ‘non’, ‘in’, and ‘un’. These, however, require some care. The sentence

John is unmarried

is equivalent to

It is not the case that John is married

and is symbolized as ‘ $\neg J$ ’. On the other hand,

Some students are unmarried

is *not* equivalent to

It is not the case that some students are married.

Instead, it is equivalent to

It is not the case that *all* students are married.

COMBINING LOGICAL CONNECTIVES

Before I introduce the last two connectives in *SL*, let's look at how the connectives are combined to form more complex sentences. Imagine that I want to symbolize this sentence:

Either I will stay home tonight, or I will both go out for dinner and go to the movie theater.

Let's start with this:

$H \vee D \& M$

There's a problem, though. Remember that a goal of formal languages is to eliminate ambiguity, and this sentence is definitely ambiguous. Is it a disjunction that has a conjunction as one of its disjuncts, or is a conjunction with a disjunction as one of its conjuncts? If it's a disjunction, then my staying home would be enough to make it true. If it's a conjunction, then its being true requires that I go to the movies. In *SL*, the ambiguity is cleared up with parentheses. If we decide that the English sentence is a conjunction, then we'll symbolize it like this:

$(H \vee D) \& M$

And if it is a disjunction, we'll symbolize it like this:

$H \vee (D \& M)$

In *SL*, and all of the other systems that we will encounter, every sentence has a main connective. The main connective of a sentence determines what kind of sentence it is, and will always be outside of the parentheses—in this case, should we make the main connective the ' \vee ' or the ' $\&$ '?

There are two things to look for when trying to determine the main logical operator in an English sentence:

1. Punctuation, such as commas and semicolons.
2. Words like 'either... or', 'both... and', and 'if... then'.

The first strategy is to group things together that are on the same side of a punctuation mark. Since 'I will go out for dinner' and 'I will go to the movie theater' are both on the right side of the comma, we should group them together with parentheses, like this:

$$H \vee (D \& M)$$

The second strategy is that anything that is grouped together with parentheses should read naturally as a complete English sentence. If we translated the sentence as

$$(H \vee D) \& M$$

then ‘Either I will stay home tonight, or I will both go out for dinner’ should make sense as a stand-alone sentence, but it doesn’t because of the ‘both’ that is missing an ‘and’. ‘I will both go out for dinner and go to the movie theater’ does make sense, though. So this strategy also requires us to translate the sentence as

$$H \vee (D \& M)$$

These two strategies should be enough to help us correctly translate any well-formed, well-punctuated English sentence. Unfortunately, natural languages are often simply ambiguous, and, in those cases, you must simply do your best. Here are some more examples.

It’s not the case that both Plato and Aristotle are from Athens.

First, we’ll paraphrase the sentence to make the connected sentences clear:

It’s not the case that both Plato is from Athens and Aristotle is from Athens.

Since we need to keep the ‘both... and’ together, we should symbolize this as

$$\neg(P \& A)$$

Since the only connective that is outside the parentheses is the ‘ \neg ’, it is the main operator and the sentence is a negation.

Plato is from Athens, but Aristotle is not.

The ‘but’ is a conjunction, so paraphrasing this results in

Plato is from Athens, and it is not the case that Aristotle is from Athens

and is symbolized like this:

$P \& \neg A$

Parentheses are not needed here, since there's no ambiguity. The ' \neg ' modifies what it immediately precedes, which in this case is the simple sentence 'A'. The '&' joins the sentence immediately before it with the one that is immediately after it, which in this case are 'P' and ' $\neg A$ '. Intuitively, the main connective is the '&', and the sentence is a conjunction.

Here is a rule for determining the main connective of a sentence in symbolic logic: The main connective is the one outside the parentheses, unless there is more than one connective outside the parentheses, then the main connective is not a negation. Just keep in mind that the only time that there can be more than one connective outside the parentheses is when negations are involved, and in that case, the main connective will never be a negation. The main connective can be a negation only when it is the only one outside the parentheses.

Here's one a bit more complex:

Both either Plato was the greatest metaphysician or Aristotle was the greatest logician, and either Plato was the greatest epistemologist or Aristotle was the greatest moral philosopher.

I don't think there's any need for paraphrasing here. The comma tells us that the 'and' is the main connective, and it joins two disjunctions. So, the translation is

$(M \vee L) \& (E \vee P)$

Now, here's something that can be a little trick:

Neither Aristotle nor Epictetus were from Athens.

This can be paraphrased as Both it is not the case that Aristotle is from Athens and it is not the case that Epictetus is from Athens. So, the translation would be

$\neg A \& \neg E$

An equally acceptable paraphrase is this:

It is not the case that either Aristotle is from Athens or Epictetus is from Athens.

This would naturally be translated as

$\neg(A \vee E)$

It should be easy to see that these are equivalent. The first says that both of the sentences represented by 'A' and 'E' are false. The second says that it is not the case that either one is true. We'll prove that they are equivalent in the next chapter.

MATERIAL CONDITIONALS

A conditional is an 'if... then' sentence. There are many ways to express conditionals in English, including

- if
- if... then
- only if
- whenever
- when
- only when
- implies
- provided that
- means
- entails
- is a sufficient condition for
- is a necessary condition for
- given that
- on the condition that
- in case

A conditional claims that something is true, if something else is also. Examples of conditionals are

"If Sarah makes an A on the final, then she will get an A for the course."

"Your car will last many years, provided you perform the required maintenance."

"You can light that match only if it is not wet."

The symbol that we will use in *SL* for conditionals is called a horseshoe, ' \supset '. Another commonly used symbol is the right arrow, ' \rightarrow '. We can translate the above examples like this:

$$F \supset C$$

$$M \supset L$$

$$L \supset \neg W$$

One big difference between conditionals and other sentences, like conjunctions and disjunctions, is that order matters. Notice that there is no logical difference between the following two sentences:

Albany is the capital of New York and Austin is the capital of Texas.

Austin is the capital of Texas and Albany is the capital of New York.

They essentially assert exactly the same thing, that both of those conjuncts are true. So, changing order of the conjuncts or disjuncts does not change the meaning of the sentence, and if meaning doesn't change, then truth value doesn't change either. That's not true of conditionals. Note the difference between these two sentences:

If you drew a diamond, then you drew a red card.

If you drew a red card, then you drew a diamond.

The first sentence *must* be true. if you drew a diamond, then that guarantees that it's a red card. The second sentence, though, could be false. Your drawing a red card doesn't guarantee that you drew a diamond, you could have drawn a heart instead. So, we need to be able to specify which sentence goes before the arrow and which sentence goes after. The sentence before the arrow is called the antecedent, and the sentence after the arrow is called the consequent.

Look at those three examples again:

If Sarah makes an A on the final, then she will get an A for the course.

Your car will last many years, provided you perform the required maintenance.

You can light that match only if it is not wet.

The antecedent for the first sentence is 'Sarah makes an A on the final'. The consequent is 'She will get an A for the course'. Note that the 'if' and the 'then' are not parts of the antecedent and consequent.

In the second sentence, the antecedent is 'You perform the required maintenance'. The consequent is 'Your car will last many years'. This tells us that the antecedent won't always come first in the English sentence.

The third sentence is tricky. The antecedent is ‘You can light that match’. Why? The explanation involves understanding the difference between necessary and sufficient conditions.

A sufficient condition is something that is enough to guarantee the truth of something else. For example, getting a 95 on an exam is sufficient for making an A on the exam, assuming the usual grading scale and that exam is worth 100 points. A necessary condition is something that must be true in order for something else to be true. Making a 95 on an exam is not necessary for making an A—a 94 would have still been an A. Taking the exam is necessary for making an A, though. You can’t make an A if you don’t take the exam, or, in other words, you can make an A only if you enroll in the course.

Here are some important rules to keep in mind:

‘If’ introduces an antecedent, but ‘only if’ introduces a consequent.

If A is a sufficient condition for B, then $A \rightarrow B$.

If A is a necessary condition for B, then $B \rightarrow A$.

All of the connectives in *SL* are truth-functional. That is, the truth value of the complex sentence is a function solely of the simple sentences and the logical connectives contained in the sentence. The truth-functional conditional is called the material conditional, and it has some puzzling consequences. Consider this sentence again:

If Sarah makes an A on the final, then she will get an A for the course.

Intuitively, this sentence is false when Sarah makes an A on the final, but does *not* get an A for the course, and true when Sarah makes an A on the final and also makes an A for the course. What if Sarah does not get an A on the final? In that case, the material conditional is true no matter whether she gets an A for the course or not. So far, this doesn’t seem to be a problem. Let’s change the consequent of the conditional, however.

If Sarah makes an A on the final, then she will get an F for the course.

If this is a material conditional, it will be false only when the antecedent is true and the consequent is false, true whenever either the antecedent is false or the consequent is true. In any case in which Sarah does not make an A on the final, the sentence is true. Now, imagine that the final is worth 100 of the 1,000 points possible in the course. Also, imagine that Sarah has made a perfect score on everything in the course before the final. Sarah then takes the final and makes a score of 85. That

lowers her perfect average of 100 to an almost perfect 98.5, which is surely still deserving of an A for the course. Since she didn't make an A on the final, however, the material conditional 'If Sarah makes an A on the final, then she will get an A for the course' is still true. This is called the paradox of the material conditional, something that we will revisit in later chapters.

Most of the time, when we assert a conditional in English, we don't intend it to be a material conditional. Unfortunately, incorporating other kinds of conditionals would complicate our logic. The good news, though, is that when it comes to analyzing logical relations between sentences, we can usually just pretend that every conditional is a material conditional. For now, that's just what we'll do.

MATERIAL BICONDITIONALS

Our final connective is expressed by the English phrases 'if and only if', 'when and only when', and 'just in case'. These are used to express conditions that are both necessary and sufficient, such as making at least a 90 and getting an A. So this sentence,

Students make an A if and only if they average at least 90% on all work

can be paraphrased as

If students make an A, then they average at least 90% on all work; and
students average at least 90% on all work only if they make an A

and symbolized as a conjunction of two conditionals like this:

$$(A \supset W) \ \& \ (W \supset A)$$

Although we could certainly translate biconditionals this way, they occur often enough that it is useful to have a symbol dedicated for that purpose. It's common to use either ' \equiv ' (triple bar) or ' \leftrightarrow ' (double arrow). We'll use the triple bar, and translate the previous example like this:

$$A \equiv W$$

Material biconditionals are true whenever the two sentences joined by the biconditional operator have the same truth value, and false whenever they have different truth values. For biconditionals, as for conjunctions and disjunctions, the order of the connected sentences doesn't matter.

COMPLEX SYMBOLIZATIONS

Now that we've introduced the five truth-functional connectives, let's practice some more complex translations into *SL*. We'll use these simple sentences:

F: The Federal Reserve will raise interest rates.

C: Congress will increase spending.

I: The inflation rate goes up.

U: There is increased unemployment.

P: Prices of consumer goods increase.

Here's the first one:

If the inflation rate goes up, then either the Federal Reserve will raise interest rates or Congress will increase spending.

We start by putting in the sentence letters and the connective symbols.

$$I \supset F \vee C$$

Then, we put in the parentheses, making sure to keep the sentences on the same side of the comma together.

$$I \supset (F \vee C)$$

The second example is

Both the Federal Reserve will raise interest rates and Congress will increase spending if and only if either prices of consumer goods increase or there is increased unemployment.

Filling in the sentence letters and connectives gives us this:

$$F \& C \equiv P \vee U$$

We don't have any punctuation to help us here, but we do have a 'both... and' and an 'either... or' that will guide our punctuation, resulting in this:

$$(F \& C) \equiv (P \vee U)$$

One more:

Congress will not increase spending, unless neither prices of consumer goods increase nor there is increased unemployment; but if the inflation rate goes up, then the Federal Reserve will raise interest rates.

The semicolon tells us that the ‘but’ is the main connective, and the second conjunct is pretty simple:

$$I \supset F$$

The first conjunct is a disjunction. The first disjunct is

$$\neg C$$

The second disjunct, is a ‘neither, nor’ sentence, which is best translated as a negation of a disjunction. Putting the two disjuncts together, we have

$$\neg C \vee \neg(P \vee U)$$

So, we just need to put those two conjuncts together with an ‘&’, being sure to put brackets around the first to avoid any ambiguity:

$$[\neg C \vee \neg(P \vee U)] \& (I \supset F)$$

1.2 RULES OF SYNTAX

USE AND MENTION

We use words to talk about other things. For instance, in the sentence, ‘The 46th President of the United States is from Delaware’, the phrase ‘The 46th President of the United States’ refers to Joe Biden. In the sentence ‘Red is the color at the long wavelength end of the visible spectrum’, the word ‘red’ refers, unsurprisingly to the color red. When we use the word ‘red’, we refer to the color red. When we use the name ‘Joe Biden’, we refer to President Biden.

Sometimes, though, we want to talk about the words themselves. What should we do then? Just as we use names of persons to talk about the persons, we need names for words and expressions to be able to talk about those words and expressions. We form the names of expressions by placing those expressions in single quotes. Without the quotes, we are using the expression. When the expression is enclosed in quotes, it is being mentioned, not used. For example,

‘Red’ has three letters.

In this case, the word 'red' is being mentioned, not used. In the sentence,

Red is often used to signify passion.

the word 'red' is being used. The use-mention fallacy is committed whenever a word is used when it should have been mentioned, or vice versa. An example is the sentence,

Red has three letters.

This sentence asserts that the color red has three letters, but colors aren't the kind of things that can have letters. Instead it should be

'Red' has three letters.

The first is nonsense, the second is a perfectly meaningful, and true, English sentence. When we use a word, we are referring to the thing that the word stands for, when we mention a word, we are referring to the word itself.

OBJECT LANGUAGE AND METALANGUAGE

The ability to form names of expressions by enclosing them in single quotes enables us to talk about the expressions of one language while using another language. Here is an example:

'Ich' is the German first-person singular pronoun.

This is a sentence in English, so all of its components must be in English. By enclosing the German word in single quotes, we have formed the English name for that German expression. So, we are using English to talk about an expression in German. The language that is being discussed is called the *object language*. The language that is being used to talk about the object language is called the *metalinguage*. In this case, the object language is German, and the metalanguage is English. As we formulate the rules for *SL*, the metalanguage will be English, but the object language will be *SL*.

METALINGUISTIC VARIABLES

To refer to specific expressions of *SL*, we can simply enclose them in single quotes, as we have been doing. The sentence

'A & B' is a conjunction.

claims that a particular expression of *SL* is a conjunction. It is an English sentence using the English name for an expression of *SL*. To formulate the rules of *SL*, however, we will need to be able to discuss general types of expressions of *SL*, not just particular expressions of *SL*. We do this using *metalinguistic variables*, or *metavariables*, for short.

A metalinguistic variable is an expression in the metalanguage that stands for *any* expression in the object language. We will use lowercase Greek letters, such as ' α ', ' β ', and ' γ ', for metavariables. The preceding sentence encloses the metavariables in single quotes because it is talking about the metavariables. When the metavariables are being used, they won't need to be enclosed in quotes since they are expressions in the metalanguage.

THE SYNTAX OF *SL*

We now have everything we need to carefully define the rules of syntax of *SL*. These rules will tell us what counts as a proper expression of *SL*. Not every string of English words will be a proper English sentence. For instance 'ball the red is blue and' is composed of perfectly good English words, but any English speaker will know that the sentence is nonsense because it violates the conventions for making proper English expressions. We need to formulate rules for making proper expressions of *SL* that are rigorous and precise, such that, for any string of symbols of *SL*, will be able to determine if that string forms a proper expression of *SL*. We'll call a proper expression a 'well-formed formula', or 'wff' for short.

Here are our rules of syntax for *SL*:

1. Every uppercase letter of the Latin alphabet, with or without a numerical subscript, is a well-formed formula.
2. If α is a well-formed formula, then $\neg\alpha$ is a well-formed formula.
3. If α and β are well-formed formulas, then so is $(\alpha \ \& \ \beta)$.
4. If α and β are well-formed formulas, then so is $(\alpha \vee \beta)$.
5. If α and β are well-formed formulas, then so is $(\alpha \supset \beta)$.
6. If α and β are well-formed formulas, then so is $(\alpha \equiv \beta)$.
7. Nothing is well-formed formula unless it can be formed by repeated application of 1–6.

These are *recursive* rules. That is, they specify what counts as the simplest type of well-formed formula in *SL*, then show how increasingly more complex formulas are generated by applying the same rules again and again.

If you have paid close attention to the previous sections, you might have recognized a problem with the rules. Consider the expression formed with the negation symbol and a metalinguistic variable:

$$\neg\alpha$$

It consists of a metalinguistic variable and a symbol of *SL*. Thus, it's a mixture of the object language and the metalanguage. That means it is a well-formed formula of neither language. We're committing the use-mention fallacy—the ' \neg ' is being used, when instead it should be merely mentioned. So, we need to enclose it in single quotes, but where should the quotes go? It looks like we have two options, but neither are good. The first is to enclose the whole formula, including the metavariable, in quotes:

$$' \neg\alpha '.$$

If we do that, then, we are no longer using $\neg\alpha$ as a metavariable. Instead, we would be treating it as a symbol in *SL*, but it's not a symbol in *SL*. So, we haven't avoided the use-mention fallacy; we've merely committed another instance of the use-mention fallacy.

Another option is to enclose only the negation symbol in single quotes, like this:

$$' \neg ' \alpha$$

This is better, but it's not clear what it means. A precise way to state rule 2 without committing the use-mention fallacy is this:

2*: If α is a well-formed formula, then the formula expressed by concatenating ' \neg ' and α is a well-formed formula.

That's not bad, but fixing rules 3–7 in the same way would be somewhat awkward. The solution is to use something called “quasi-quotation”.² Quasi-quotation encloses the string formed by the symbols of the object-language and the metavariables in corner-quotes:

$$\ulcorner \neg\alpha \urcorner$$

Quasi-quotation is just shorthand for a statement about the concatenation of symbols and formulas. So $\ulcorner \neg\alpha \urcorner$ is shorthand for

The formula expressed by concatenating ' \neg ' and α .

²This is also sometimes called “Quine-quotation”, because it was first introduced by W.V.O. Quine in his book *Mathematical Logic*.

To make things simpler, we will adopt this as a convention: whenever we use expressions that consist of both metavariables and formulas of an object language, we will understand the entire expression as being enclosed in corner-quotes. So, if we were to write this,

$$(\alpha \& (\neg\beta \vee \gamma))$$

we would understand it as expressing this:

$$\ulcorner (\alpha \& (\neg\beta \vee \gamma)) \urcorner$$

and when we quote an expression consisting of a mixture of object language symbols and metavariables, we will understand that quotation to be quasi-quotation. So, when we write ' $\alpha \& \beta$ ', we will understand it as ' $\ulcorner \alpha \& \beta \urcorner$ '.

There are two more conventions regarding the use of parentheses in *SL*. First, notice that the only punctuation defined in the rules of syntax are parentheses; there are no brackets or braces. This is because we need to be able to form indefinitely large expressions of *SL*, which requires nested expressions. If we demanded a different punctuation character at each level, we would quickly run out of punctuation symbols. This can, though, make things difficult to read. Since we will never, in this book, encounter nesting of more than three levels, we will use braces and brackets for levels two and three. So, instead of writing

$$\neg((\alpha \supset (\beta \& \gamma)) \equiv ((\alpha \vee \delta) \vee (\gamma \supset \epsilon)))$$

we will write

$$\neg\{[\alpha \supset (\beta \& \gamma)] \equiv [(\alpha \vee \delta) \vee (\gamma \supset \epsilon)]\}$$

which should be easier to read.

The second convention is that we will drop the outermost set of parentheses that encloses the entire complex formula, since those parentheses aren't needed to avoid ambiguities. So, instead of

$$(\alpha \& \beta)$$

we'll simply write

$$\alpha \& \beta$$

and instead of

$$[\alpha \& (\neg\beta \vee \gamma)]$$

we can write

$$\alpha \& (\neg\beta \vee \gamma)$$

Now, we'll use the rules to determine if an expression of *SL* is a well-formed formula. We start with the sentence letters used in the expression, then show, by using repeated instances of the rules of syntax, that the expression can be generated. Here's a simple example:

Show that ' $\neg(A \& B) \supset C$ ' is a well-formed formula, or sentence, in *SL*.

First, by rule 1, both '*A*' and '*B*' are wff's. By rule 3, ' $A \& B$ ' is a wff. By rule 2, ' $\neg(A \& B)$ ' is a wff. Finally, by rule 5, ' $\neg(A \& B) \supset C$ ' is a wff.

Finally, here are some expression that are *not* well-formed formulas of *SL*. See if you can determine what is wrong with each one.

$$A \& B \vee C$$

$$(P \supset Q)R$$

$$\neg \equiv Z$$

$$\neg[(E \& F) \supset D]$$