4. Statement Logic

4.1. Introduction

We have seen that certain notions developed for the foundation of mathematics, like “set”, “function” etc., are useful for the purpose of analysing meaning in natural language. It turned out that another branch of science, logic, provides extremely useful tools as well.

The prime area of logic is the investigation of the **laws of proper inference**. For example, Aristotle, one of the founder of logic, was interested in inferences of the following type:

(1) a. Every human is an animal.
Every animal is mortal.
¾¾¾¾¾¾¾¾¾
Every human is mortal.

b. Every sparrow is a bird.
Every bird has two legs.
¾¾¾¾¾¾¾¾¾
Every sparrow has two legs.

c. No human has four legs.
All Greeks are humans.
¾¾¾¾¾¾¾¾¾
No Greek has four legs.

d. No plant moves.
All trees are plants.
¾¾¾¾¾¾¾¾¾
No tree moves.

These pieces of logical reasoning each consist of two **premises** and one **conclusion**. They should be understood in the following way: If the premises are true, then the consequence must be true as well.

Note that we don’t claim that the premises are actually true, and hence we don’t claim that the consequence is actually true! Logic is not concerned with what is true or false as a matter of fact, but only in the notion of logical consequence. For example, the following pieces of reasoning are equally valid:

(2) Every man is an elephant.
Every elephant has two noses.
¾¾¾¾¾¾¾¾¾¾¾¾
Every man has two noses.

Obviously, the reasonings (a) and (b) are very similar to each other, and so are the reasonings (c) and (d). To be sure, they talk about different things, e.g. men vs. sparrows, but the type of logical reasoning doesn’t seem to be affected by that. Aristotle captured this similarity by using variables for sentences. The two types of reasonings can be given by the following schemes; such schemes are called **syllogisms**.

(3) a. All A are B.
All B are C.
¾¾¾¾¾¾¾
All A are C.

b. No A is B
All C are A.
¾¾¾¾¾¾¾
No C is B.

Modern logic is still interested in the laws of valid inference. One major new invention since the time of Aristotle is that logicians (starting from Leibniz in the 18th century) developed **artificial languages**. These are very simple languages that typically are much less expressive than human languages but follow very perspicuous syntactic principles, and for which we can define valid inferences in a perspicuous way. For quite some time the leading idea was that natural languages are too unprecise to be used for the formulation of theories in philosophy, mathematics and science, and that we should switch to the use of such artificial languages when engaged in these activities. Ironically, it turned out that the investigation of the properties of simple artificial languages were crucial for the semantics of natural languages.

In this section we will talk about one such languages, **statement logic**. We will discuss a more expressive form in the next section, **predicate logic**. We will omit almost completely a very important aspect of logic (some would say, the most important one), namely, how logic is used to actually **prove** certain sentences. If you are interested in these aspects, for example, in Natural Deduction or some other proof calculus, you can work through the relevant sections of Partee, ter Meulen, Wall (1991), some other logic textbook, or a course in logic and proof theory.

Statement Logic (or Propositional Logic) is concerned with the logical behaviour of statements, without looking below the clause level. The rules of statement logic can be best formulated using an artificial language, which I will call **SL**. It will contain variables for sentences, and ways to combine sentences to larger sentences.

4.2. Syntactic rules of SL

The only syntactic category of SL are **well-formed formulas** (also called **wff**'s, pronounced in your favourite way of barking).

(4) Basic (atomic) statements: We assume that we have a denumerably infinite number of basic statements, given by p_1, p_2, p_3, ... For convenience, I will write p for p_1, q for p_2, r for p_3.

(5) Recursive definition of the set of all wff’s of SL:

a. Basic clause: Every atomic statement is a wff.
b. Recursion clauses:
c. If Φ is a wff, then ¬Φ is a wff. (negation, *non* Φ)
d. If Φ, Ψ are wff’s, then the following expressions are wff’s as well:
[Φ   Ψ], (conjunction, Φ *and* Ψ; also written as &)
[Φ   Ψ], (disjunction, Φ *or* Ψ)
[Φ → Ψ], (conditional, *if* Φ *then* Φ, also written as   )
[Φ ↔ Ψ]. (biconditional, Φ *if and only if* Ψ; also written as ≡)

Note that φ, ψ are used here as meta-language variables for expressions of our formal language. The expressions ¬, , , →, ↔ are called **logical constants**, **logical connectives** or **logical operators**.

Some examples of wff’s:	And some non-examples:
p	p
[p q]	[p → q]
[¬[p q] r]	→ q
[[[p q] → ¬q] ¬¬p]	[[[p q] → ¬q] ¬¬p]]

We can analyse well-formed expressions in form of syntactic trees:

```
          wff
        /     \
      ¬        wff
             /  |  \
            p        q        corresponds to ¬[p   q]
```

The parentheses can be seen as auxiliary symbols that help us to reconstruct the internal syntactic structure of an expression. Without them, expressions might be structurally ambiguous. For example, “¬ p   q” could stand for either ¬[p   q], or for [¬p   q]. This is an important distinction between the language of statement logic and toy English: The language of statement logic does not allow for structural ambiguity.

However, we are allowed to drop parentheses in case no ambiguity arises. We can do this always with outermost parentheses, e.g. we can write, instead of [p   [q → r]], simply p   [q → r]. Furthermore, there is a convention saying that   and   bind stronger than → and ↔. Hence we can write, instead of [p   q] → r, simply p   q → r. Furthermore, as the bracketing for formulas like [p   q]   r or [p   q]   r does not matter, we can simply write p   q   r and p   q   r, respectively.

4.3. Semantic rules for SL

Sentences can be either **true** or **false** (this is an oversimplification, but let’s take it for the time being). Let us take “1” for “true” and “0” for false. We say that the **truth value** of a statement is 0 in case it is false, and 1 in case it is true.

Statement logic doesn’t tell us anything about the truth or falsity of atomic statements. But it gives us rules that allow us to determine the truth values of complex statements, given the truth values of their parts. Hence, as soon as we know the truth values of the atomic statements of a complex statement, then we know the truth value of the complex statement, as well.

The semantic rules for the logical constants are usually given in terms of **truth tables** that give us the truth value of the complex expression with respect to all possible assignment of truth values of the immediate parts.

We start with negation. The motivation of the following truth table is that if a sentence like *it is raining* is true, then *it is not raining* is false, and vice versa.

(6) **Negation:**

Φ	¬Φ	
1	0	[if Φ is true, ¬Φ is false]
0	1	[if Φ is false, ¬Φ is true]

The truth table for conjunction says that a sentence containing two sentences connected by *and* is true iff both sentences are true.

(7) Conjunction:

Φ	Ψ	Φ Ψ	
1	1	1	[if Φ and Ψ are true, Φ Ψ is true]
1	0	0	[if Φ true, Ψ false, Φ Ψ is false]
0	1	0	[if Φ false, Ψ true, Φ Ψ is false]
0	0	0	[if Φ false, Ψ false, Φ Ψ is false]

There are important differences between the statement logic symbol   and English *and.* The latter can be used to conjoin other categories than sentences, e.g. NPs. The symbol   just conjoines sentences. Sometimes *and* between other categories can be traced back to *and* between sentences, sometimes not:

(8) a. John and Mary yawned: John yawned and Mary yawned.
b. John and Mary met: *John met and Mary met.

In English, other expressions come close to *and*, e.g. *but.*

(9) The sun is shining, but it is cold outside.
The sun is shining, and it is cold outside.

In English, *and* often has additional meanings, for example temporal ones. Such meanings are not covered by .

(10)John put off his shoes and he went to bed.
John went to bed and he put off his shoes.

The rule for disjunction says that two sentences are true iff either one of them is true:

(11) **Disjunction:**

Φ	Ψ	Φ Ψ
1	1	1
1	0	1
0	1	1
0	0	0

This represents the so-called **inclusive** *or*, as Φ   Ψ is true if both Φ and Ψ is true.

Sometimes it is argued that we find in natural language also an **exclusive** or, which makes the complex sentence false in this case. And indeed, in many cases we will understand *or* in such a way. Example:

(12)Mary has a son, or Mary has a daughter.

However, it can be argued that this is not an effect of the meaning of *or*, which is always inclusive, but due to a scalar implicature (cf. chapter 1).: If a speaker knows that Mary has a son and Mary has a daughter, then the speaker will certainly use *and* as the main connective, as this would result in a more informative sentence. So, if the speaker uses *or*, the hearer can infer that the same sentence with *and* probably is not compatible with what the speaker knows, and hence the hearer will infer that only one of the sentences is true.

Natural-language *or*, like *and*, can often be used to conjoin non-sentences. Again, we may reduce such cases to sentential *or*:

(13)John or Mary yawned: John yawned or Mary yawned.

The meaning of the conditional is given by the following truth table:

(14) **Conditional:**

Φ	Ψ	Φ → Ψ
1	1	1
1	0	0
0	1	1
0	0	1

In a conditional Φ → Ψ, we call Φ **antecedent** and Ψ **consequent**. Conditionals are also called **material implications**.

The conditional is the logical operator that is perhaps less similar to natural-language equivalents. The truth table is motivated by the following consideration:

- If the antecedent is true, then the consequent must be true.
- If the antecedent is false, nothing follows for the consequent.

Example: *If it rains, the streets are wet* just excludes the possibility that it rains and the streets aren't wet. If it doesn't rain, the streets may be wet or not wet.

But note that "→" does not cover every aspect of natural language conditionals. For example, the following sentence should be true, according to the truth table for "→":

(15)If Austin is the capital of Tasmania, then it is raining in Paris.

But we have the feeling that this sentence makes a rather strange claim. Whether or not Austin is the capital of Tasmania should have no influence on whether it is raining in Paris. Such aspects cannot be covered by "→" and show that the semantics of conditional sentences in natural language is more complex than the semantics of the conditional in statement logic.

Furthermore, we find a tendency in natural language use to strengthen the conditional. Example:

(16)If this car is cheaper than $1000, then I will buy it.

Most people will understand this sentence as saying: If this car is not cheaper than $1000, i.e. if it is more expensive, then I will not buy it. But again this may be a consequence of how we use expressions of natural language: A speaker typically makes assertions that are as “strong” as he or she is willing to support. Given this pragmatic rule, we typically can infer that the speaker of this sentence understands it in a stricter way.

(17) **Biconditional:**

Φ	Ψ	Φ ↔ Ψ
1	1	1
1	0	0
0	1	0
0	0	1

The biconditional can be seen as a double-sided conditional. It is used, for example, in definitions. Example:

A   B iff (↔) every element in A is an element in B.

4.4. Truth Values of Complex Expressions

Truth values of complex expressions can be computed as soon as we know the truth values of the atomic expressions. Example:

(18)[[p   q] → [r   p]]
0   1   1   0   [assumed truth value of atomic expressions: p: 0, q: 1, r: 1]
¾¾   ¾¾¾
1   0
¾¾¾¾
0   [derived truth value of complex expression]

In this example, we computed the truth value of a complex expression given specific truth values of the atomic expressions. Of course, we can also compute the truth value of the complex expression with respect to **every** possible truth value for the atomic expressions. We get then something like a truth table for the complex expression:

(19)

p	q	r	[p q]	[r p]	[[p q] → [r p]]
1	1	1	1	1	1
1	1	0	1	0	0
1	0	1	1	1	1
1	0	0	1	0	0
0	1	1	1	0	0
0	1	0	1	0	0
0	0	1	0	0	1
0	0	0	0	0	1

Instead of the format given above, we may use the slightly more compressed version given below. We fist assign all possible values to the atomic expressions, and then compute the truth values of successively larger expressions, writing the result under the corresponding operator. The truth values of the whole formula are underlined.

(20)

[[p		q]	→	[r		p]]
1	1	1	**1**	1	1	1
1	1	1	**0**	0	0	1
1	1	0	**1**	1	1	1
1	1	0	**0**	0	0	1
0	1	1	**0**	1	0	0
0	1	1	**0**	0	0	0
0	0	0	**1**	1	0	0
0	0	0	**1**	0	0	0

4.5. Do we Need Additional Logical Connectives?

In addition to the logical connectives so far, we can think of other logical connectives. For example, we might define a connective for the exclusive *or*, call it **eor**, as follows:

(21)

Φ	Ψ	Φ **eor** Ψ
1	1	0
1	0	1
0	1	1
0	0	0

But note that we don’t have to introduce a new connective **eor** to get this truth table. We can define it with the connectives we have introduced so far. For example, instead of Φ **eor** Ψ we can write ¬[Φ ↔ Ψ], with the same semantic effect:

(22)

Φ	Ψ	¬[Φ↔ Ψ]
1	1	0
1	0	1
0	1	1
0	0	0

As a matter of fact, we have already introduced more connectives than we actually need. We could well do with two connectives ¬ and , as all the others can be defined through them:

- Φ Ψ may be rendered as ¬[¬Φ ¬Ψ],
- Φ → Ψ may be rendered as ¬Φ Ψ,
- Φ ↔ Ψ may be rendered as [Φ → Ψ] [Ψ → Φ].

For example,

(23)

Φ	Ψ	¬[	¬Φ		¬Ψ]
1	1	**1**	0	0	0
1	0	**1**	0	0	1
0	1	**1**	1	0	0
0	0	**0**	1	1	1

Note that the bold values
give the same truth value as Φ Ψ

4.6. Tautologies, Contradictions, Contingencies

We have seen that we can compute the truth value of a complex statement, given the truth values of the atomic statements and the truth tables of the logical operators.

Now, some statements are special, as their truth value does not depend on the truth value of the atomic statements. They are either always true or always false. If they are always true, they are called **tautologies**, If they are always false, they are called **contradictions**. Statements that are neither tautologies nor contradictions are called **contingencies**.

(24)Examples of tautologies:
$p \quad \neg p$ (“Law of the excluded middle”)
$\neg[p \quad \neg p]$
$p \rightarrow p$
$p \rightarrow [q \rightarrow p]$

(25)Examples of contradictions:
$p \quad \neg p$
$p \leftrightarrow \neg p$
$\neg[[p \quad q] \leftrightarrow [q \quad p]]$

Note: Φ is a tautology iff $\neg\Phi$ is a contradiction!

Tautologies and contradictions are of particular interest in logic: They are true or false for purely logical reasons, not for factual reasons. In particular, this means that if we replace the atomic statements in a tautology, we get back another tautology, and the same for contradictions. Example:

(26)$[p \quad \neg p]$ is a tautology,
Replacing p by $[q \rightarrow r]$, we get $[[q \rightarrow r] \quad \neg[q \rightarrow r]]$,
also a tautology.

That is, we can generate infinitely many tautologies and contradictions from formulas that we know have this property. On the other hand, we can check whether a given formula is a tautology or contradiction by reducing it to a known tautology or contradiction.

4.7. Logical Equivalence and Logical Consequence

Logical consequence is the notion that one expression ψ follows logically from another one, φ. This means that whenever φ is true, then ψ is true. We say that φ **entails** ψ, and write $\varphi \quad \psi$. or $\varphi \quad \psi$ For example, we have $p \quad q \quad p$. There is a close relation between logical consequence and tautologies:

$$\varphi \quad \psi \text{ iff } \varphi \rightarrow \psi \text{ is a tautology.}$$

That is, in order to check whether $\varphi \quad \psi$ is a valid logical consequence, we can check whether $\varphi \rightarrow \psi$ is a tautology.

Logical equivalence is the notion that two expressions always (= for all assignments of truth values to basic expressions) have the same truth value. Note: φ and ψ are logically equivalent iff $\varphi \quad \psi$ and $\psi \quad \varphi$. The symbol for logical equivalence is $\Leftrightarrow$; we write $\varphi \Leftrightarrow \psi$. For example, we have $p \quad q \Leftrightarrow q \quad p$.

Logically equivalent expressions have a nice logical property: If two expressions φ, ψ are logically equivalent, $\varphi \Leftrightarrow \psi$, and if ψ occurs as a subformula in some complex formula, then we may replace an occurrence of ψ by φ, without changing the truth values of the complex formula. Consider the following example:

(27)a. We have $[p \quad q] \Leftrightarrow [q \quad p]$.
b. Take formula $[p \quad q] \rightarrow r$
replace $[p \quad q]$ by $[q \quad p]$,
result: $[q \quad p] \rightarrow r$

4.8. Laws of Statement Logic

There are certain patterns of logical consequences that are useful to know. They are known as the “Laws of Statement Logic”. In the following, take T to be a formula that is always true (a tautology), and to be a formula that is always false (a contradiction).

- Idempotency: (a) $[\varphi \quad \varphi] \Leftrightarrow \varphi$
  (b) $[\varphi \quad \varphi] \Leftrightarrow \varphi$.
- Associativity: (a) $[[\varphi \quad \psi] \quad \chi] \Leftrightarrow [\varphi \quad [\psi \quad \chi]]$
  (b) $[[\varphi \quad \psi] \quad \chi] \Leftrightarrow [\varphi \quad [\psi \quad \chi]]$
- Commutativity: (a) $[\varphi \quad \psi] \Leftrightarrow [\psi \quad \varphi]$
  (b) $[\varphi \quad \psi] \Leftrightarrow [\psi \quad \varphi]$
- Distributivity: (a) $[\varphi \quad [\psi \quad \chi]] \Leftrightarrow [[\varphi \quad \psi] \quad [\varphi \quad \chi]]$
  (b) $[\varphi \quad [\psi \quad \chi]] \Leftrightarrow [[\varphi \quad \psi] \quad [\varphi \quad \chi]]$
- Identity: (a) $[\varphi \quad] \Leftrightarrow$ (c) $[\varphi \quad] \Leftrightarrow \varphi$
  (b) $[\varphi \quad T] \Leftrightarrow \varphi$ (d) $[\varphi \quad T] \Leftrightarrow T$
- Complement: (a) $[\varphi \quad \neg\varphi] \Leftrightarrow T$ [excluded middle]
  (b) $\neg\neg\varphi \Leftrightarrow \varphi$ [double negation]
  (c) $[\varphi \quad \neg\varphi] \Leftrightarrow$
  (d) $\neg \quad \Leftrightarrow T$
- De Morgan: (a) $\neg[\varphi \quad \psi] \Leftrightarrow [\neg\varphi \quad \neg\psi]$
  (b) $\neg[\varphi \quad \psi] \Leftrightarrow [\neg\varphi \quad \neg\psi]$
- Conditional laws: (a) $[\varphi \rightarrow \psi] \Leftrightarrow [\neg\varphi \quad \psi]$
  (b) $[\varphi \rightarrow \psi] \Leftrightarrow [\neg\psi \rightarrow \neg\varphi]$ [contraposition]
  (c) $[\varphi \rightarrow \psi] \Leftrightarrow \neg[\varphi \quad \neg\psi]$
  (d) $[\varphi \rightarrow \psi] \Leftrightarrow [\varphi \quad \psi] \leftrightarrow \psi$.
  (e) $[\varphi \rightarrow \psi] \Leftrightarrow [\varphi \quad \psi] \leftrightarrow \varphi$
- Biconditional laws: (a) $[\varphi \leftrightarrow \psi] \Leftrightarrow [[\varphi \rightarrow \psi] \quad [\psi \rightarrow \varphi]]$
  (b) $[\varphi \leftrightarrow \psi] \Leftrightarrow [[\neg\varphi \quad \neg\psi] \quad [\varphi \quad \psi]]$

These laws, of course, look suspiciously like the laws of set-theory, with the following correspondences of operators. In particular, the following operations on sets and on wff’s correpsond to each other:

(28)

Sets	Statements (Wff's)
∩	
	→
=	↔
′	¬
U	T
Ø	

Indeed, set theory and statement logic show the same underlying mathematical structure, which is called a **Boolean Algebra**. But note that there is one interesting difference between the set-theoretic laws and the laws of statement logic: An expression like A B is not a set, but expresses a statement. The corresponding epression [φ → ψ], however, is a well-formed formula, just like φ and ψ.

4.9. Proof of logical equivalences

We can use the laws of statement logic to prove that certain statements are logically equivalent, and in particular we can reduce complex statements to logically equivalent simpler ones. This is much like in set theory. The following example is a simplification (the second biconditional law):

(29)

[[p q] [¬p ¬q]]	
[[p q] ¬p] [[p q] ¬q]	Distributivity
[[p ¬p] [q ¬p]] [[p ¬q] [q ¬q]]	Distributivity
[T [q ¬p]] [[p ¬q] T]	Complement
[q ¬p] [p ¬q]	Identity
[¬ p q] [¬ q p]	Commutativity
[p → q] [q → p]	Conditional
p ↔ q	Biconditional

We also can use the laws to prove that a particular inference is valid, that is, a logical consequence. The line of the argument is as follows:

Your task is to find out whether Φ Ψ is a valid logical inference.
You know that Φ Ψ iff Φ → Ψ is a tautology.
Hence, when you can reduce Φ → Ψ to T, then Φ Ψ is valid.

Some examples:

(30) Show that p [¬q p],
that is, show that [p → [¬q p]] can be reduced to T.

[p → [¬q p]]	
[¬p [¬q p]]	Conditional
[[¬q p] ¬p]	Commutativity
[¬q [p ¬p]]	Associativity
[¬q T]	Complement
T	Identity

(31)Show that p, that is, [→ p] can be reduced to T.

[→ p]	
[¬ p]	Conditional
[T p]	Complement
T	Identity

The latter says that from a falsity ( ) we can infer an arbitrary statement p. Or, from falshood everything follows (**ex falso quodlibet**). We sometimes find a rhetorical use of this principle, e.g. *If you are seventeen, then I am Marylin Monroe*.