2.1 Review

The purpose of logic is to make reasoning precise. It helps answer questions like "*how can we be sure that a given argument is valid?*" or "*why should we believe a claim that has been made – what evidence do we have for that?*". These questions occur in pretty much all scientific disciplines and particularly in mathematics and computer science, where we ask ourselves "*Why should we trust in a theorem that has been stated?*", "*What is a proof?*", "*How can we be sure that a given piece of software works as intended?*", or "*how can we verify that?*".

Logic is not the answer itself, but it provides the means to express answers to such questions in a way that there is no more doubt about the validity of an argument. It also provides mechanisms that help *finding* the answers. In the last two decades theorem provers have found solutions for mathematical problems that could not be solved by humans before. In order to do that, logic must be formulated in a way that a computer can handle it.

In this course we will look at a series of increasingly expressive logics from propositional logic all the way to type theory. We will describe the formal language of these logics, ways to describe evidence for arguments and claims, a formal concept of proofs, ways to provide computer assistance for developing proofs, reasons why the formalisms are reliable, and limitations of what can be done with formal logic at all.

2.2 Inference and justifications

One of the key issues of logic is separating valid inferences from invalid ones by providing evidence that justifies a claim that has been made. If evidence can be found, the claim can be accepted. Otherwise we better not trust it.

In order to accomplish this, logic has to abstract from of the ambiguities of natural language and drop irrelevant details that may distract us from the key arguments or lead us to wrong conclusions.

Example 2.1

Here is an example of a simple, but erroneous argument

Mammals have hair. Monkeys have hair. Thus monkeys are mammals.

While we know that all three propositions are true individually, the argument itself is flawed.

If we replace the word *monkeys* by something else, e.g by *teddybears*, and keep the rest unchanged we end up with

Mammals have hair. Teddybears have hair. Thus teddybears are mammals.

which is obviously wrong.

Thus in order to reveal the logical structure of the argument, we should replace words that have meaning by abstract symbols, which gives us

$$((M \Rightarrow H) \wedge (A \Rightarrow H)) \Rightarrow (A \Rightarrow M)$$

where the symbol M is a placeholder for being a mammal, A for being a monkey, and H for having hair. The symbols $\wedge$ and $\Rightarrow$ are the usual logical symbols for conjunction and implication. The use of symbolic language makes it obvious that the argument was not a valid inference. □

So what exactly constitutes a valid logical inference? What evidence can we provide to be sure that a given statement is true? Let us look at a few commonsense examples first.[1]

Example 2.2 (Evidence for commonsense propositions)

The term evidence is used very often in the legal system. If you want to prove something in court, you need to provide sufficient evidence.

(1) Imagine you get pulled over by the police for speeding and you dispute that. What evidence could they provide? (Reading of the radar)

What if they claim you've been using your cell phone while driving? (Cell phone records)

(2) Imagine you had a few too many beers at a party, drive home, and get caught. What is the evidence that you will lose your license?

A blood test is evidence for drunk driving. But why do you lose your license? (It's the law)

So altogether there are three pieces of evidence: You were caught behind the wheel. The blood test proves DWI. The law shows that you'll lose your license as a consequence.[2]

(3) What would you consider as evidence that it has been raining a while ago?

A strong indicator would be that the streets are wet. But would that be sufficient?

We also have to be sure that there hasn't been any street cleaning, no water spills or other reasons for the street being wet. Otherwise the evidence is not enough to prove our point.

(4) What evidence would you give that there is either an even number of people in this room or an odd number?

People who had too much exposure to the way mathematics is taught today may be inclined to say: "it is just so" or "what else should be the case?".

But such an answer is quite unsatisfactory since it relies on a rather metaphysical argument. It does not provide any evidence at all and we still don't know if the number is even or if it is odd. A much more straightforward answer is to count the number of people in the room and use the result to make the decision. □

So evidence may be atomic or composed out of smaller pieces. It has to be sufficient and it should be given explicitly.

[1] While logic by nature is very abstract, the human mind isn't tuned to abstraction. We get easily confused if things are explained only in terms of symbols. Therefore we often use simple, everyday examples to explain new ideas intuitively before introducing the concepts formally. One should keep in mind, though, that these examples are merely illustrations and not part of the formal logic itself.

[2] Actually, many laws are described in a way that they can be viewed as logical rules. In Germany, most court decisions are based entirely on the letter of the written law book (the US legal system is a case law – individual decisions of judges are used as guidelines for future decisions.
Already in the late 1980's there have been research projects attempting to formalize parts of the law in logic, for instance in order to be able to check contracts for inconsistencies. Investigating that subject could make a nice course project. If anyone is interested, look at `http://www.tfgordon.de` or talk to me.

Example 2.3 (Evidence in mathematics)

Let us look at a few more examples from mathematics, which is a bit more rigorous.

(1) What evidence could you give for the proposition `3<4`? That is basic mathematics, but what could you do to convince someone who doesn't know that yet?

A simple method would be to use boxes, legos or the like, stack 3 and 4 on top of each other and decide which stack is larger.

This is pretty convincing since we have a sense of larger and smaller and we know how to count. But obviously it would be a bit tedious to do mathematics that way. That is why mathematicians have tried to reduce arithmetic to very simple concepts that are considered self-explanatory, like counting, and to explain everything else in terms of these concepts.

Let us look at some of these concepts and their laws, as they illustrate the principles of evidence construction quite well.

(2) Why evidence could you give for `0=0`?

There is not much that one can say here. It is our fundamental understanding that two identical objects must be equal. That is what equality means. There is no simpler truth than that zero is equal to itself. It is, as we say, *self-evident* and the only evidence we could provide is stating "it is so by definition".

In logic, we often call such statements *axioms*.[3] One cannot prove them but one can give strong reasons why it makes sense to accept them as facts that require no proofs. They cannot be reduced to more primitive facts and thus evidence for their truth cannot be based on anything else but the statements themselves.

When we have to deal *atomic* propositions of that nature we will denote their evidence by a special term `axiom`. This indicates that the proposition has been postulated as self-evident and cannot be decomposed into simpler propositions.[4].

(3) What evidence could you give for `1=1`, `2=2`, `3=3`,... ?

Are all these equalities self-evident too? There are two problems with that view.

First of all that would lead to infinitely many axioms in the theory of arithmetic and usually that is not a good idea.

The other problem is what we consider as self-evident and what not? We say that two identical numbers must be equal and there is no doubt about that. But is `1=1`, `2=2`, `3=3` as *immediate* as `0=0`? Can we always recognize identical numbers? What about

```
11111111111111111111111111111111 = 11111111111111111111111111111111 ?
```

While these two numbers are in fact the same, our mind is not able to recognize that immediately. We have to start counting instead and only then we know. So instead of declaring all

[3] According to the Merriam-Webster dictionary an axiom is

(a) a maxim widely accepted on its intrinsic merit

(b) a statement accepted as true as the basis for argument or inference

(c) an established rule or principle or a self-evident truth

[4] Note that non-atomic propositions like the successor axiom cannot have the term `axiom` as evidence. Their evidence must be described by a term that respects their internal logical structure.

these equalities to be axioms, we have to come up with a way to reduce them to something more simple like 0=0.[5].

Since dealing with decimals is somewhat complicated, formal arithmetic uses something more simple as foundation. It says, we have the number 0 and the ability to count, which we express by a successor function s. Everything else will be defined in terms of these two components. So decimal numbers are just abbreviations, that is 1 stands for s(0), 2 for s(s(0)), 3 for s(s(s(0))), etc.

In order to reduce the above equalities to 0=0 we just need one additional (successor) axiom: For arbitrary numbers x and y s(x)=s(y) holds if x=y, or briefly x=y $\Rightarrow$ s(x)=s(y).

Then 1=1, which is just short for s(0)=s(0), holds because it follows from 0=0 and an application of the successor axiom and the evidence for 1=1 would be composed from the evidence of the successor law and that of the law 0=0.

In the same fashion the justification for 2=2, 3=3, etc. consists of 2, 3,... applications of the successor law to the axiom 0=0, and the evidence is constructed accordingly.[6]

(4) Here is a more difficult issue: why do you believe 0≠1. That is, why can 0=1 never be true?

This doesn't have to do with the meaning of equality itself. Actually, there is no way to prove 0≠1. But why do we believe it anyway?

What would happen if we would allow 0=1 to be true?

If we would accept 0=1, then we would be able to show that all natural numbers are equal and that's just an absurdity. The whole system of arithmetic would collapse. We simply cannot allow 0=1, which means we must postulate 0≠1. Although we have no external evidence for 0 being different from 1, we can be sure that one will never be able to find evidence for 0=1. So it is safe to assume 0≠1.

Here is a related question: why do we believe 0≠2, 0≠3, . . . ?

We cannot reduce these propositions to 0≠1 since the left side of the inequality is already zero.[7] So does that mean we have to postulate an infinite number of axioms after all or can we avoid that?

If we unfold the decimals we can see that all the above inequalities have something in common. We have zero on the left side and some successor number on the right. So all these formulas can be expressed by a simple generic axiom 0≠s(y).

[5]Recognizing equality is even more difficult when we deal with real numbers. Given the equality 1.000000000...=1.000000000... one will never be able to decide that the two denoted numbers are actually equal as one can always look at only a finite number of digits. Immediately after one has declared the two numbers to be equal one may encounter two different digits in their decimal expansion, which means that the decision was wrong. Dealing with the equation 1.000000000...=0.9999999999... appears to be even worse but is equally difficult. In both situations one can only give a definite answer if the two numbers are distinct.

[6]In the early days most logic-based systems actually based their arithmetic entirely on the successor notation, as the decimal system requires extra-logical mechanisms that are difficult to embed into a clean logical formalism. The Coq system, a sister system of our Nuprl system, went that path. Nuprl, however, focused on practical applications where using only the successor notation is infeasible and introduced addition, multiplication and decimal numbers as basic components of the logic. The Coq system finally adopted that approach, while others, like Agda, emphasize logical purity.

[7]Actually, in the ring of numbers modulo 2, 0≠1 holds but 0≠2 doesn't, in the ring of numbers modulo 3 0≠1 and 0≠2 hold and 0≠3 doesn't, etc.

(5) Given all we know so far, how would you show $1 \neq 2$ or $2 \neq 3$?

Unfortunately, the successor law that we have so far doesn't help that much, since it is only good for showing that $1=2$ would hold if $0=1$. What we need, is an implication in the other direction which would allow us to prove that $0=1$ would hold if $1=2$ does. Then, since $0 \neq 1$ is an axiom, $1=2$ cannot hold.

To fill that gap, formal arithmetic has formulated the inverse of the successor law: if two successors are equal, then so must be the original numbers. The rationale is that counting forwards can be undone and if we do so we arrive where we started. $s(x)=s(y) \Rightarrow x=y$.

Using the inverse successor law we can reduce all kinds of inequalities between (different) natural numbers to the axiom $0 \neq s(y)$. To prove $2 \neq 7$, for instance, we only have to apply the law twice to $0 \neq 5$.

(6) Given all that, how would you show that *any two numbers are either equal or different?*

You could again say "*they have to be*" but that wouldn't give us any evidence.

In the previous examples we have implictly used an algorithm for checking the equality of inequality of arbitrary numbers. Given two numbers x and y we reduce the equation $x=y$ using the inverse successor law until one of the numbers is zero. If the other one is zero as well, then the two numbers are equal and otherwise different.

So the evidence for the claim would be the algorithm that on input x and y decides whether the two numbers are equal or not and constructs the specific evidence accordingly.

□

As you see, even in simple arithmetic there are a lot of questions that need to be dealt with when we try to be precise about what we know. These questions have motivated the waw how arithmetic was eventually formalized.

We have seen that some facts have to be considered self-evident and that other seemingly self-evident facts can be reduced to more primitive ones. Finally evidence may also come in the form of an algorithm that computes specific evidence for any given input.

Although we tried to be fairly precise in the previous examples, our description of evidence still involved a bit of handwaving. If we want to be sure that the evidence actually proves a statement, we need to become more formal. This means we have to introduce a formal language for expressing statements, a formal language for expressing evidence, and a calculus that links formal evidence to formal logical propositions.

2.3 Propositional logic

Most of you have probably seen logical symbols before. Mathematicians like to use them for abbreviation purposes and formal logic just goes one step further. It tries to express everything in terms of some formal language in order eliminate ambiguities and to allow logical expressions to be processed by a machine.

Describing the formal language of logic is like describing a programming language. One has to agree on what symbols and keyworks are allowed to be used, how formal mathematical sentences have to be formed from smaller components, and what the precise meaning of the formal sentences shall be.

For now we stick to the simplest form of logic called propositional logic. This logic handles only the most primitive relations between formulas – implication, consjunction, disjunction, and negation. Everything else will have to be expressed by propositions or, to be precise, by propositional variables. This means that all the internals of such a proposition will not be visible. Different primitive propositions like `0≠1` and `0≠2` will be represented by two different propositional variables A and B and the symbol will not tell us that they have anything in common. We will also not be able to use a parametric proposition for `0≠s(x)` – this is the realm of first-order logic. The only commonalities we will be able to describe is that a primitive proposition occurs several times in a compound formula.

Propositional logic allows only a very coarse analysis of statements and logical arguments. But it already provides deep insights into the fundamental structures of logical reasoning. More elaborate logics like first-order or higher-order logics have to respect these structures and only provide additional mechanisms for handling the finer details.

In the literature there is a variety of notations for logical symbols. We shall use the following symbols for logical connectives: negation $\neg$ (read "not"), implication $\Rightarrow$ ("implies"), conjunction $\wedge$ ("and"), and disjunction $\vee$ ("or").[8]. We use the symbols $P, Q, R, P_o, Q_0, R_0, P_1, Q_1, R_1, \ldots$ as names for *propositional variables* and parentheses $(,)$ as as delimiters.

Definition 2.4 (Syntax of propositional logic)

The formulas of propositional logic are recursively defined as follows

(1) *Every propositional variable is a formula.*

(2) *If A is a formula then so is $\neg A$.*

(3) *If A and B are formulas then so are $(A \Rightarrow B)$, $(A \wedge B)$, and $(A \vee B)$.*

In the above definition definition, the symbols A and B are placeholders for arbitrary formulas but they are not formulas themselves (A and B are not included in our list of symbols). In an implementation of propositional logic as recursive data type A and B they would serve as implementation variables. Therefore, they are often called *meta-variables* but it is sufficient to view them as slots for the real formulas.

Since our definition does not give preference rules that would permit us to drop parentheses, parentheses should always be used to avoid ambiguities. Outer parentheses may be omitted.

The meaning of propositional formulas is clear from our intuitive understanding of the words "not", "implies", "and", and "or", although there is a certain amount of inconsistency in how these words are actually used. This enables us to translate informal or mathematical text into a representation in propositional logic.

Example 2.5 (Formalization)

Formalize the following statements as formulas in propositional logic

(1) *If there is a snowstorm then roads will be closed. The roads are open. Thus there can't be a snowstorm.*

Using memnonic symbols the formalization would be $((S \Rightarrow C) \wedge (O \wedge (O \Rightarrow \neg C))) \Rightarrow \neg S$. If we restrict ourselves to the permitted symbols, we get $((P \Rightarrow Q) \wedge (R \wedge (R \Rightarrow \neg Q))) \Rightarrow \neg P$

[8] The book of Smullyan uses $\sim$ for negation and $\supset$ for implication. Prof. Constable prefers $\sim$ for negation and & for conjunction. Our notation is based on the one used in the Nuprl proof development system.

(2) *If there is a snowstorm then roads will be closed. There is no snowstorm. Hence the roads must be open.*

$((P \Rightarrow Q) \wedge (\neg P \wedge (Q \Rightarrow \neg R))) \Rightarrow R$

(3) *If there can't be no snowstorm then there is one.*

$\neg\neg P \Rightarrow P$

(4) *This sentence is true*

Can't be expressed in propositional logic or in first-order logic

□