

Chapter 2: The Logic of Atomic Sentences

§2.1 Valid and sound arguments

Conclusion

An argument is a piece of reasoning (a sequence of statements) attempting to establish a **conclusion**. The conclusion is what the arguer is trying to establish. It is indicated by words like *therefore, so, hence, thus, consequently*. What immediately follows these words is usually the conclusion.

Premises

The premises are the reasons the arguer gives in support of the conclusion. They may be given before the conclusion, or after it. Premises are typically preceded by words like *because, since, after all*.

Validity

In a valid argument, the conclusion **follows from** or is a **logical consequence of** the premises.

Here is our **definition** of validity:

An argument is valid if it is impossible for its premises to be true and its conclusion false.

The word “impossible” is important here. The fact that an argument’s conclusion is **actually** true does not make the argument valid — validity requires that there be no **possible** circumstance in which the premises would be true and the conclusion false.

Similarly, the fact that an argument contains a false premise means nothing about the argument’s validity or invalidity. Some arguments with false premises are valid, and others are invalid. What matters is whether there is any **possible** circumstance in which the premises would be true and the conclusion false.

Soundness

A sound argument is a valid argument with true premises.

So, every sound argument is valid, but not every valid argument is sound.

Fitch bar notation

In many books, arguments are written up using the “3-dot” symbol: \therefore . So, for example, you might see:

Socrates is a man.
All men are mortal.
 \therefore Socrates is mortal.

In *LPL*, we’ll use the “Fitch bar” notation. The premises are written **above** the horizontal line (the Fitch bar), and the conclusion below:

| Socrates is a man.
| All men are mortal.
| _____
| Socrates is mortal.

Examples

Here are two examples of arguments: one valid, one invalid.

Example 1a: a valid argument

Cube(a)
Large(a)
SameShape(a, b)
Cube(b)

Example 1b: an invalid argument

Cube(a)
Large(a)
SameShape(a, b)
Large(b)

We'll return to these arguments later. We'll see how to prove the first one valid, and how to show that the second one is invalid. Our method of **proof** of validity is very different from our method of **showing** invalidity.

§2.2 Methods of proof

A step-by-step demonstration showing that the conclusion follows from the premises. In a proof, a series of **intermediate conclusions** are reached, leading in a chain from the premises to the (ultimate) conclusion. The intermediate conclusions are also written below the Fitch bar.

At each step, there must be **absolute certainty**. That is, there must be no chance that any conclusion (intermediate or otherwise) does not follow from the sentences it is inferred from. Our steps must be such that there is never a possibility that we might be inferring a false sentence from true ones.

Proofs involving the identity symbol

Our language so far contains only atomic sentences, which limits our ability to come up with rules for deriving conclusions from premises. But we can take advantage of some features of the identity relation to put our first two rules (concerning the symbol $=$) into play.

- First, note that the identity relation (the relation that holds between a and b in virtue of which $a = b$ is true) is **reflexive**. That is, each thing is identical to itself. In other words, sentences like $b = b$ are always true.
- Second, the identity relation is **symmetrical**. That is, if $a = b$, then $b = a$.
- Third, the identity relation is **transitive**. That is, if $a = b$ and $b = c$, then $a = c$.
- Finally, if $a = b$, then whatever holds of a also holds of b . This is called the **indiscernibility of identicals**.

We will enshrine these features of identity in our system of proof by introducing rules that take advantage of them.

§2.3 Formal proofs

We will be developing a “deductive system” for writing up formal proofs. We call the system \mathcal{F} , and we will be employing a computer program called “Fitch” that is a somewhat more “user-friendly” version of \mathcal{F} .

In a formal proof in \mathcal{F} , we use the Fitch bar notation. The premises are written above the (horizontal) Fitch bar; the subsequent steps (intermediate conclusions and the ultimate conclusion) are written below the Fitch bar.

Each step in a formal proof must be entered in accordance with some precisely stated **rule** of the formal system of rules. By applying a **rule** to some previous line or **lines** in a proof, we provide a **justification** for entering a new step in a proof.

A **justification**, then, cites a **rule** and the **lines** to which the rule is being applied in order to generate the line being introduced.

Our first rules can now be stated:

Identity Introduction (= Intro)

$$\triangleright \left| \begin{array}{l} n = n \end{array} \right.$$

The triangle points to the sentence that the rule entitles you to enter. This rule says, in effect, that you may enter a sentence of the form $n = n$ at any point you wish. Obviously, this rule embodies the principle of reflexivity of identity.

Identity Elimination (= Elim)

$$\triangleright \left| \begin{array}{l} P(n) \\ n = m \\ P(m) \end{array} \right.$$

This rule tells you that you may substitute m for n wherever you like, provided that you have the sentence $n = m$. This rule embodies the principle of indiscernibility of identicals.

Notice that although the rule is called an “elimination” rule, nothing is really being eliminated. The idea is that we have used (eliminated?) an identity sentence in the process of arriving at a conclusion. That is, we are arguing “from” an identity sentence, and in that sense we are “eliminating” it.

In \mathcal{F} , each logical symbol has a pair of rules associated with it: an **introduction** rule, which tells you how to get a sentence containing that logical symbol **into** a proof, and an **elimination** rule, which tells you how to deduce something **from** a sentence containing that logical symbol. (For this reason the rules in a system like \mathcal{F} are sometimes called “int-elim” rules.) Thus, = **Intro** tells us how to enter an identity sentence (we can enter $a = a$), and = **Elim** tells us how to use an identity sentence ($n = m$) as a premise.

Don’t worry that our two rules seem to have ignored the symmetry and transitivity of identity. In fact, symmetry and transitivity follow from reflexivity and indiscernibility. That is, using only = **Intro** and = **Elim**, you can prove that $b = a$ follows from $a = b$, and that $a = c$ follows from $a = b$ and $b = c$. (You will be proving transitivity in exercise 2.16 in H_3 .)

For an illustration of how = **Elim** works, open [Ch2Ex3.prf](#). Point the focus slider at line 3 and click on both premises; they will be highlighted, meaning that they are your support sentences. Then choose rule = **Elim**, and click on **Check Step**. Notice which sentence Fitch inserts, by default. Is this the sentence you expected? (Perhaps you were surprised to see both occurrences of **a** replaced by **b**). Try entering a new line that makes only one replacement in line 1, and ask Fitch to check it out. (Be sure to highlight your two support lines by clicking on them.) Then enter yet another line that makes a different single replacement in line 1 and have Fitch check it out. You will notice that = **Elim** licenses all three of these inferences.

Please be aware that (unlike Fitch) \mathcal{F} is a very strict system. Its rule = **Elim** permits us to substitute the name that occurs to the **right** of the equals sign for the one that occurs to the **left**, but does not permit us to substitute the name on the left for the one on the right. That is, the rule does not strictly apply to the pair of sentences **Cube(b)** and **a = b**. It only applies to the sentences **Cube(a)** and **a = b**. Since the Fitch program is more liberal about this fine detail than \mathcal{F} is, we will be able to ignore it when we're using Fitch.

§2.4 Constructing proofs in Fitch

You try it

Work the problem on p. 58, using the file **Identity 1** (it's in the Fitch Exercise Files folder). To see what your proof should look like, open the file [Proof Identity 1.prf](#). (Either click on the link or find the file on the **Supplementary Exercises** page of the course web site.)

Ana Con

This is a mechanism that is built into Fitch. It basically checks to see whether a conclusion does indeed follow from its premises. **Ana Con** has some limitations: it does not understand the predicates **Adjoins** and **Between**, and some complicated arguments may stump it.

As we will see, **Ana Con** uses a broader notion of logical consequence than is strictly allowed in FOL. For example, in FOL we cannot deduce **Larger(a, b)** from **Smaller(b, a)**. This is because this inference depends on the meaning of the predicates, and FOL is ignorant of the meanings of the predicates in the arguments it examines.

But given the meanings of **Larger** and **Smaller**, we may note that it is not possible for the first sentence to be true and the second false. So there is a clear sense in which the inference in question is valid. **Ana Con** takes the meanings of the predicates into account. So we'll say that **Larger(a, b)** is an **analytic consequence** of **Smaller(b, a)**, even though it is not a **first-order** consequence of it. Try to show this in Fitch by opening [Ch2Ex2.prf](#) and using **Ana Con** to complete the proof.

§2.5 Demonstrating nonconsequence

We don't use proofs

We do not give proofs of nonconsequence; we do this by means of counterexample. This is because of the following fact:

When we establish that an argument is **valid**, we establish something quite **general**. That is, that it is **impossible** for the premises to be true and the conclusion false. To put it another way, we establish that in **every possible situation** in which the premises are true, so is the conclusion.

Conversely, to establish that an argument is **invalid**, we must show that it is not valid. That is, that it is **possible** for the premises to be true and the conclusion false. To put it another way, we must establish that there is **some possible situation** in which the premises are true and the conclusion is false.

So when we show an argument to be invalid, we need not prove anything general. It is sufficient to describe a possible situation in which the premises are true and the conclusion is false.

Since demonstrating nonconsequence does not involve proofs, we will not be using the program Fitch to show that an argument is invalid. For example, suppose we are using Fitch, and we examine some purported proof of a given argument, and we see that the proof contains a mistake or a misapplication of the rules. **That doesn't show the argument to be invalid.** Perhaps it is just a faulty proof of a valid argument! In that case, some other proof could be found.

But we can use Fitch's **Con** mechanisms to tell us **that** an argument is invalid. Let's apply the **Ana Con** mechanism to our previous examples, Example 1a and Example 1b. To see how, open [Ch2Ex1a.prf](#) and [Ch2Ex1b.prf](#).]

We construct counterexamples

To demonstrate nonconsequence, we use the program Tarski's World. This program lets us create counterexamples: possible situations, or "worlds," in which the premises of an argument are true and its conclusion is false.

We'll continue with Examples 1a and 1b. Open [Ch2Ex1a.sen](#), [Ch2Ex1b.sen](#), and [Ch2Ex1.wld](#). (1a) is valid, (1b) is invalid. If we change the world slightly, we can make the conclusion of (1b) false while leaving the premises true. (Just make *b* into a small cube.) But there is no way of making the conclusion of (1a) false while leaving the premises true, for that is a valid argument, and any world in which its premises are true will also make its conclusion true.

Now do the **You try it** on p. 64 to construct your own demonstration of nonconsequence. You will be constructing a counterexample to [Bill's Argument](#).

If you solved this problem, you should have ended up with a world that looks something like this: [Bill's Argument.wld](#).

Deductive vs. Inductive Reasoning

In this course, we will be studying **deductive** reasoning, where we try to determine whether a given conclusion does or does not follow from a certain set of premises. But a good deal of reasoning is not deductive; often, one is interested in something weaker than absolute certainty. One may be interested in whether a set of premises makes the truth of a conclusion **more probable**, rather than in whether it guarantees the truth of the conclusion.

For a good illustration of the difference between these two modes of reasoning, look at the world [Deductive vs Inductive.wld](#) and the accompanying sentence file [Deductive vs Inductive.sen](#).

We can't see whether there is a block behind the medium cube in the column on the right, but we know from the fact that the sentences in the file are true that there must be one there. For we know that there is a block named *b* that is in the same column as *a* and the same row as *c*.

Can we tell anything about the size and shape of b ? Using **inductive** reasoning, we might conclude that b is a cube. After all, in the left-hand column, all the blocks are of the same shape, and in the middle column, all the blocks are of the same shape. So one might reason, inductively, that the third column will be like the other two in this respect. If it is, the hidden block will be a cube, for the two that are visible are cubes. But there is no certainty here, for we can imagine a world in which b is not a cube.

Can we tell what size b is? Here, we can do better. For b **must** be small. If b were medium or large, it would be at least partially visible. But we cannot see anything of b . Therefore, it must be small. Here, we have used **deductive** reasoning to establish that a certain conclusion **follows** from the information that we already have (and not just that it is more likely to be true, given that information).

So we have given an **inductive** argument that b is likely to be a cube, and a **deductive** argument that b must be small. We can now rotate the world 90 degrees (or switch to a 2-D view) to find out the size and shape of b .

As to the size, there can be no surprises—there's no possibility that b can be anything but small, consistent with the information we already have. But as to the shape, we may well be in for a surprise.

Note that this is a feature of all inductive arguments: no matter how good the argument is, there is always a possibility, however remote, that the conclusion may be false even though all the premises are true.