The Logic of Atomic Sentences

Counterexamples & Formal Proofs

William Starr

09.06

Announcements

For 09.06

1. Complete survey for Logic section times (on Bb)
   - Before Wednesday at midnight!!
2. HW1 & HW2 are due next Tuesday
   - But you can start working on them now!
3. The textbook should be in stock today or tomorrow

Outline

Logical Consequence & Validity

The Definitions

Logical Validity & Consequence

1. An argument is logically valid if and only if there is no way of making the premises true that does not make the conclusion true as well
2. One claim is a logical consequence of another if and only if there is no way the latter could be true without the former also being true

- In a valid argument the truth of the premises guarantees the truth of the conclusion
- Soundness = validity + true premises

Proof
Showing Validity

- Our account of logical consequence is great in theory
  - But, it doesn't give us any specific tools for actually showing that a given argument is valid
- In our simple examples it was fairly easy to tell whether or not the arguments were valid
  - But, for most interesting arguments this issue cannot be decided so easily
- Today, we'll begin to learn the more precise & powerful techniques for doing this that modern logic offers
- The key notion here will be that of proof

Proof
What is it?

Proof

A proof is a step-by-step demonstration which shows that a conclusion C must be true in any circumstance where some premises $P_1, \ldots, P_n$ are true

1. The step-by-step demonstration of C can proceed through intermediate conclusions
2. It may not be obvious how to show C from P_1 and P_2, but it may be obvious how to show C from some other claim Q that **is** an obvious consequence of P_1 and P_2
3. Each step must provide incontrovertible evidence for the next

An Example
The Argument

Argument 2

| Superman is Clark Kent
| Superman is from Krypton
|———
| Clark Kent is from Krypton

- Remember this argument?
- Let's review our (informal) proof of it
- Then we'll look at a formal proof of it and contrast/compare the two proofs

An Example
From Informal to Formal

Argument 2: Informal Proof

Since superman **is** Clark Kent, whatever holds of Superman also holds of Clark Kent. We are given that Superman is from Krypton, so it must be the case that Clark Kent is from Krypton.

Formal Proof of Argument 2

1	FromKrypton(superman)	
2	superman = clark.kent	
3	FromKrypton(clark.kent)	= **Elim**: 1,2

An Example

Discussion

Formal Proof of Argument 2

1	FromKrypton(superman)	
2	superman = clark.kent	
3	FromKrypton(clark.kent)	**= Elim**: 1,2

- In our informal proof of Argument 2 we appealed to a fact about the meaning of *is*:
  - The Indiscernibility of Identicals
- In our formal proof we also appealed to this fact, but under a different guise: **= Elim**
  - We indicated that by listing it next to the formula we used it to infer
  - Also write numbers of formulas we inferred it from

An Example

What is = Elim?

- In formal deduction system, the facts about meanings used to justify each step are recast as rules of inference
  - **= Elim** is the way to formally recast the Indiscernibility of Identicals

Here they are:

= Elim

P(n)
⋮
n = m
⋮
▷ P(m)

Indiscernibility of Identicals

If n is m, then whatever is true of n is also true of m
(where 'n' and 'm' are names)

- **= Elim** restates Ind. of Id.'s formally:
- If you have a formula of the form n = m and one of the form P(n) then you can infer one of the form P(m)

An Example

= Elim in Action

Let's see exactly how **= Elim** was applied earlier

= Elim

P(n)
⋮
n = m
⋮
▷ P(m)

Formal Proof of Argument 2

1	FromKrypton(superman)	
2	superman = clark.kent	
3	FromKrypton(clark.kent)	**= Elim**: 1,2

- 3 is inferred by **= Elim** from 1 & 2
  - 1 is of the form P(n)
  - 2 is of the form n = m
  - 3 is of the form P(m)

Rules of Inference

Overview

- More rules to our formal system of deduction later
  - For simplicity, we are going to call our system $\mathcal{F}$
- So far, we've only looked at one rule: **= Elim**
  - But there's another rule for identity: **= Intro**
- Rules will always come in pairs: **Intro** and **Elim**
- Are we going to have rules for all of the predicates of the blocks language?
  - No! We will focus on rules for logical words like *is*, *and*, *not*, etc.
- For now, we are going add just two more rules

Two More Rules
= Intro & Reiteration

= Introduction (= Intro)

$\triangleright$ | n = n

- Everything is self-identical

Reiteration (Reit)

| P
| ⋮
$\triangleright$ | P

- Once you've shown P, reuse it wherever

Nothing mind-blowing here!

Formal Proofs
Another Example

Argument 3

| a = b
|---
| b = a

= Elim

| P(n)
| ⋮
| n = m
| ⋮
$\triangleright$ | P(m)

Proof of Argument 3

1	a =b	
2	a= a	= **Intro**
3	b= a	= **Elim**: 2,1

Formal Proofs
Generally Speaking

A Formal Proof

P_1	
⋮	
P_n	
C_1	Justification 1
⋮	⋮
C_m	Justification m
C	Justification $m+1$

- $P_1 - C$ are in FOL
- Premises: $P_1 - P_n$
- Conclusion: C
- Intermediate Conclusions: $C_1 - C_m$
- Justifications indicate where & how the formula on that line is being inferred
  - That is: from which formula(e) & by what rule of inference

Fitch
Why Go Digital?

- Learning to hand-write formal proofs is okay
- But using a computer to write them is better
  - A computer can check whether or not a formal proof is correct
  - A computer can auto-format proofs
  - A computer prevents you from making really bizarre mistakes
  - A computer generates a more readable, electronically transferrable proof
- This is why we have Fitch

Fitch

Demo!

- Now, we'll run through reconstructing the last two formal proofs in Fitch
- Fitch allows steps the are not strictly part of $\mathcal{F}$
  - Neither Fitch nor $\mathcal{F}$ have **specific** rules for predicates other than =
    - Fitch, however, has: **Ana Con**

Ana Con

Ana Con allows you to infer things that follow from the meaning of the predicates in the 'Blocks Language' of Tarski's World, e.g. LeftOf(a, b), therefore RightOf(b, a).

Ana Con At Work

Special Relations in the Blocks Language

1	FrontOf(a, b)	
2	BackOf(b, a)	**Ana Con**: 1

1	Large(a)	
2	b = c	
3	SameSize(a, b)	
4	Large(b)	**Ana Con**: 1,3
5	Large(c)	= **Elim**: 2,4

In-Class Exercise

Exercise 2.9

*Construct a formal proof for the following argument (you will need to use **Ana Con**).*

LeftOf(a, b)
b = c
———
RightOf(c, a)

You may work in groups of 6 or fewer. You have 10 minutes, then I will call one of you to present your group's solution.

Showing Non-Consequence

About Counterexamples

- If an argument is valid, then it is impossible for the premises to be true & the conclusion false

Showing Non-Consequence

So, to show that an argument is not valid you have to show is that it is **possible** for the premises to be true and the conclusion false

- Okay, are there formal proofs of non-consequence?
- In general, no but for the blocks language, we can be more concrete

Showing Non-Consequence

Counterexamples in the Blocks Language

Non-Consequence in the Blocks Language

- For the blocks language, a formal proof that Q is not a consequence of $P_1, \ldots P_n$ consists of:
  1. A sentence file with $P_1, \ldots P_n$ labeled as premises, and Q labeled as conclusion
  2. A world file that makes $P_1, \ldots P_n$ true but Q false
- We will call such a world a counterexample to the argument in the sentence file

- I'll do **Exercise 2.21**

Non-Consequence

The Basic Fact

Showing Non-Consequence

To show that an argument is not valid you have to show is that it is **possible** for the premises to be true and the conclusion false

- For the blocks language, we can use Tarski's World to do this
- In other cases, you just have to describe a consistent scenario in which the premises are true and the conclusion false

Non-Consequence

Another Example

I'll do **Exercise 2.26**

Conclusion

For 09.06

Conclusion

1. Validity: impossible for premises to be true while conclusion false
2. Proof: demonstrating validity
3. Informal proof: stated in ordinary language
4. Formal proof: carried out in a formal proof system
5. Counterexample: a scenario where the premises are true and the conclusion is false
6. Non-consequence is demonstrated with a counterexample