The Logic of Atomic Sentences

Counterexamples & Formal Proofs

William Starr

01.29.09

Announcements

For 01.29

1. HW2 is due next Tuesday

Outline

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 also indicated that this fact justifies a transition from some claims to another by listing it next to the formula we used it to infer and writing the numbers of the formulas we inferred it from

An Example

What is = Elim?

- In a formal system of deduction, the facts about meanings used to justify each step are recast as formal 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

- Over the course of the semester we will be adding more rules to our formal system of deduction
  - 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**
- In general, rules will always come in pairs, **Intro** & **Elim**
- Are we going to have rules for all of the predicates of, say, the blocks language?
  - No, that'd be way too complicated
  - We will focus on rules for logical particles like *is*, *and*, *not*, etc.
- For now, we are going add just two more rules

Two More Rules

= Intro & Reiteration

= Introduction (= Intro)

$$\triangleright \;\Big|\; n = n$$

Reiteration (Reit)

$$\begin{array}{cl} & P \\ & \vdots \\ \triangleright & P \end{array}$$

- Everything is self-identical
- Once you've shown P, you can reuse it whenever you want

Nothing surprising here!

Formal Proofs

Another Example

Argument 3

$$\begin{array}{|l} a = b \\ \hline b = a \end{array}$$

Proof of Argument 3

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

= Elim

$$\begin{array}{cl} & P(n) \\ & \vdots \\ & n = m \\ & \vdots \\ \triangleright & P(m) \end{array}$$

Formal Proofs

Generally Speaking

A Formal Proof

P_1	
$\vdots$	
P_n	
C_1	Justification 1
$\vdots$	$\vdots$
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 does however have a mechanism that is not part of $\mathcal{F}$
    - **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 $\mathsf{RightOf}(\mathsf{b},\mathsf{a})$.

Ana Con At Work
Special Relations in the Blocks Language

1	$\mathsf{FrontOf}(\mathsf{a},\mathsf{b})$	
2	$\mathsf{BackOf}(\mathsf{b},\mathsf{a})$	**Ana Con**: 1

1	$\mathsf{Large}(\mathsf{a})$	
2	$\mathsf{b} = \mathsf{c}$	
3	$\mathsf{SameSize}(\mathsf{a},\mathsf{b})$	
4	$\mathsf{Large}(\mathsf{b})$	**Ana Con**: 1,3
5	$\mathsf{Large}(\mathsf{c})$	$=$ **Elim**: 2,4

In-Class Exercise
Exercise 2.9

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

$\mathsf{LeftOf}(\mathsf{a},\mathsf{b})$
$\mathsf{b} = \mathsf{c}$
$\mathsf{RightOf}(\mathsf{c},\mathsf{a})$

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 **all** of $P_1, \ldots P_n$ 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

All computer scientists are rich

Anyone who knows how to program

a computer is a computer scientist

John is rich

John is a computer scientist

Homework 2

Due 02.03

Homework 2

Due on Tuesday 02.03:

- Exercises 2.1, 2.2, 2.6, 2.8, 2.20