Formal Proofs and Boolean Logic II

Extending $\mathcal{F}$ with Rules for ¬

William Starr

09.29.11

Outline

Review

Proof by Contradiction

- Last class: formal proofs for ∨ and ∧
  - What about ¬?
- That's the topic of Today's class
- Our ¬ **Intro** rule will allow us to prove negated claims
  - Just like proof by contradiction!
- So let's review that informal method

Proof by Contradiction

Proving a Negated Claim

Proof by Contradiction (Official Version)

1. To prove that P is false, show that a contradiction ⊥ follows from P
2. To prove that P is true, show that a contradiction ⊥ follows from ¬P

Proving a Negated Claim

To prove ¬P, assume P and prove a contradiction ⊥

- All contradictions are impossible, thus false
- If you can show that P leads to a contradiction, then P must be false
- But if P is false, then ¬P must be true

Review
What is a Contradiction Again?

Contradiction

- A contradiction is any sentence that cannot possibly be true, or any group of sentences that cannot all be true simultaneously
- The symbol $\perp$ is often used as a short-hand way of saying that a contradiction has been obtained

- Examples:
  1. $\neg\mathsf{Cube}(a) \land \mathsf{Cube}(a)$
  2. $a = b, b = c, a \neq c$
  3. $\mathsf{Cube}(a) \land \mathsf{Tet}(a)$

Proof by Contradiction
A Simple Example

Claim: This argument is valid

| ¬SameShape(a, b)
| b = c
|---
| ¬a = c

Proof: We want to show $\neg a = c$ from the premises, so we will use a proof by contradiction

1. Suppose $a = c$
2. Then, from premise one $\neg\mathsf{SameShape}(c, b)$ follows by Indiscernibility of Identicals
3. But by premise two, we know $\mathsf{SameShape}(c, b)$. This is a contradiction, $\perp$!
4. So our supposition must have been false; that is, $\neg a = c$ must be true given the premises

Formal Rules for ¬
Where We Are Going

- The basic idea behind ¬ **Intro** is familiar from our informal method of proof by contradiction
  - You can use ¬ **Intro** to infer $\neg P$ when you have proven that a contradiction $\perp$ follows from P
- What exactly counts as proving a contradiction ($\perp$)?
- If we had a $\perp$ **Intro** rule, when should we apply it?

Two Kinds of Contradictions
Boolean vs. Analytic

Boolean Contradictions

- E.g. $\mathsf{Cube}(a), \neg\mathsf{Cube}(a)$ or $\mathsf{Tet}(a) \land \neg\mathsf{Tet}(a)$
- Can't be true because of what the Booleans mean

VS

Analytic Contradictions

- E.g. $\mathsf{Large}(a), \mathsf{Small}(a)$ or $\mathsf{FrontOf}(a, b), \mathsf{BackOf}(a, b)$
- Can't be true because of what the predicates mean

Contradictions
⊥ Intro

⊥ Intro

	P
	⋮
	¬P
▷	⊥

- So, you've proven P and ¬P?
  - You can introduce ⊥
- **Question**: does this rule detect Analytic contradictions? (Like FrontOf(a, b), BackOf(a, b))
- **Answer**: NO!!

- **Question**: How would you infer ⊥ on the basis of FrontOf(a, b), BackOf(a, b)?
- **Answer**: In Fitch, you can do it with Ana Con

Boolean v. Analytic Contradictions
Within $\mathcal{F}$

Boolean ⊥ in $\mathcal{F}$

1	Cube(a)	
2	¬Cube(a)	
3	⊥	**⊥ Intro**: 1, 2

- We have P and ¬P
- So ⊥ **Intro** allows us to introduce ⊥

Analytic ⊥ in Fitch

1	Cube(a)	
2	Tet(a)	
3	⊥	**Ana Con**: 1, 2

- Here we do **not** have P and ¬P
- So ⊥ **Intro** does not give us ⊥
- But **Ana Con** does

⊥ Elim
What Should ⊥ Elim Be?

- Remember, all rules come in pairs
- We've stated ⊥ **Intro**, but we haven't said anything about ⊥ **Elim**
- What **should** we be able to infer from a contradiction?
- Let's think about it for a minute

Valid Arguments
What If The Premises are Inconsistent?

Logical Consequence, Validity

C is a logical consequence of $P_1, \ldots, P_n$ if and only if it is impossible for $P_1, \ldots, P_n$ to be true while C false

- What follows from a contradiction?
  - Anything!
- Why?
  - It's impossible for it to be true
- So, it is impossible for it to be true while any conclusion is false!

Contradictions
⊥ Elim

⊥ Elim

$$\begin{array}{r|l} & \bot \\ & \vdots \\ \triangleright & \mathsf{P} \end{array}$$

- From a contradiction ⊥, any conclusion follows!
- Why again?

- An inference step is valid just in case it cannot lead you from a true premise to a false conclusion
- Since the premise ⊥ in this inference can never be true, the inference can never lead one from a true premise to a false conclusion

Contradictions
Wait, What were We Doing?

- So, two more rules in $\mathcal{F}$: ⊥ **Intro**, ⊥ **Elim**
- Cool, but why did go on this tangent about ⊥?
- Because introducing ⊥ was essential for ¬ **Intro**
  - ¬ **Intro** is proof by contradiction, so we needed to know exactly when we could write ⊥
- So now we are in a position to see ¬ **Intro**

¬ Intro
From Informal to Formal Proof

Proving a Negative Claim

- To prove ¬P, assume P and prove a contradiction using this assumption
- This is an example of **Proof by Contradiction**

Example Informal Proof

From $\mathsf{a} = \mathsf{b}$ and $\mathsf{b} \neq \mathsf{c}$ we will prove $\mathsf{a} \neq \mathsf{c}$. We use proof by contradiction. **Proof**: Suppose $\mathsf{a} = \mathsf{c}$. Well, $\mathsf{b} = \mathsf{c}$ follows from this assumption and premise one by Ind. of Id.'s. But, this contradicts premise two, ⊥. So our assumption was wrong, in which case $\mathsf{a} \neq \mathsf{c}$.

¬ Intro

$$\begin{array}{r|l} & \begin{array}{|l} \mathsf{P} \\ \hline \vdots \\ \bot \end{array} \\ \triangleright & \neg\mathsf{P} \end{array}$$

To prove ¬P:

1. Assume P
2. Derive ⊥ (using ⊥ **Intro**)
3. Conclude ¬P
   (Discharging assumption of P)

¬ Intro
An Example

1	$\mathsf{a} = \mathsf{b}$	
2	$\mathsf{b} \neq \mathsf{c}$	
3	$\quad \mathsf{a} = \mathsf{c}$	
4	$\quad \mathsf{b} = \mathsf{c}$	= **Elim**: 3, 1
5	$\quad \bot$	⊥ **Intro**: 2, 4
6	$\mathsf{a} \neq \mathsf{c}$	¬ **Intro**: 3-5

✓Goal: $\mathsf{a} \neq \mathsf{c}$

¬ Intro

$$\begin{array}{r|l} & \begin{array}{|l} \mathsf{P} \\ \hline \vdots \\ \bot \end{array} \\ \triangleright & \neg\mathsf{P} \end{array}$$

⊥ Intro

$$\begin{array}{r|l} & \mathsf{P} \\ & \vdots \\ & \neg\mathsf{P} \\ \triangleright & \bot \end{array}$$

Some More Examples

- Let's do a formal proof for **6.25**:

$$\begin{array}{|l} \neg A \wedge \neg B \\ \hline \neg(A \vee B) \end{array}$$

- Let's also finish the proof from slide 26 of 02.19
  - This will use ⊥ **Elim**

¬ Intro

Another Example

ARGUMENT 1: Analytic ⊥ Revisited

1	¬SameShape(a, b)	
2	b = c	
3	a = c	
4	¬SameShape(c, b)	= **Elim**: 1,3
5	⊥	**Ana Con**: 2, 4
6	a ≠ c	¬ **Intro**: 3-5

Goal: a ≠ c

Informal Proof

We want to show a ≠ c, so we use proof by contradiction. **Proof**: Suppose a = c. From premise one it follows that ¬SameShape(c, b), by Ind. of Id. But this contradicts premise two which requires that *c* **is** *b*. So our assumption (a = c) was wrong, hence a ≠ c follows.

¬ Intro

$$\begin{array}{r|l} & \begin{array}{|l} P \\ \hline \vdots \\ \bot \end{array} \\ \triangleright & \neg P \end{array}$$

- There is no rule in $\mathcal{F}$ which justifies line 5
- But this is what we need to prove a ≠ c!
- So, this proof can't be finished in $\mathcal{F}$
- We *can* finish it in Fitch!

Negation

¬ Elim

¬ Elim

$$\begin{array}{r|l} & \neg\neg P \\ & \vdots \\ \triangleright & P \end{array}$$

Simple Example

1	¬¬Cube(a)	
2	Cube(a)	¬ **Elim**: 1

- If ¬¬P is true, so is P
- Obvious and useless? No!
- Its use: prove P by contradiction
- Use ¬ **Intro** to prove ¬¬P, then apply ¬ **Elim**

¬ Elim

An Example

ARGUMENT 2

$$\begin{array}{|l} Tet(e) \vee Cube(a) \\ \neg Tet(e) \\ \hline Cube(a) \end{array}$$

Informal Proof of Argument 2

We will use a proof by contradiction. Suppose ¬Cube(a). This pretty clearly contradicts the premises. To be sure, we'll take it in cases. Suppose Tet(e). Then the contradiction is clear. Suppose ¬Cube(a). Then we also have a contradiction. So our assumption must have been wrong. Hence, Cube(a) must be true given the premises.

Let's make this into a formal proof in Fitch

Subproofs
The Big Picture

- Subproofs correspond to elements of informal proofs:
  - The cases of a proof by cases
  - The temporary assumption in a proof by contradiction
- Just like cases and temporary assumptions, there are certain important restrictions on subproofs

Cases
The Constraints

Argument 3

$$\begin{array}{|l} (\mathsf{Cube}(c) \land \mathsf{Small}(c)) \lor (\mathsf{Tet}(c) \land \mathsf{Small}(c)) \\ \hline \mathsf{Small}(c) \land \mathsf{Cube}(c) \land \mathsf{Tet}(c) \end{array}$$

Pseudo-Proof of Argument 3

We will use a proof by cases based on premise one. **Case 1**: Suppose $(\mathsf{Cube}(c) \land \mathsf{Small}(c))$. Then $\mathsf{Small}(c)$ follows. **Case 2**: Suppose $\mathsf{Tet}(c) \land \mathsf{Small}(c)$. Then $\mathsf{Small}(c)$ follows. So, $\mathsf{Small}(c)$ follows in either case. But in case 1 we had $\mathsf{Cube}(c)$ and in case 2 we had $\mathsf{Tet}(c)$, hence our conclusion follows: $\mathsf{Small}(c) \land \mathsf{Cube}(c) \land \mathsf{Tet}(c)$.

- Why pseudo-proof?
  - Argument 3 is not valid
  - This 'proof' leads us from a possible premise to an impossible conclusion
  - That's exactly what proofs aren't supposed to do

Cases
The Constraint

Pseudo-Proof of Argument 3

We will use a proof by cases based on premise one. **Case 1**: Suppose $(\mathsf{Cube}(c) \land \mathsf{Small}(c))$. Then $\mathsf{Small}(c)$ follows. **Case 2**: Suppose $\mathsf{Tet}(c) \land \mathsf{Small}(c)$. Then $\mathsf{Small}(c)$ follows. So, $\mathsf{Small}(c)$ follows in either case. But in case 1 we had $\mathsf{Cube}(c)$ and in case 2 we had $\mathsf{Tet}(c)$, hence our conclusion follows: $\mathsf{Small}(c) \land \mathsf{Cube}(c) \land \mathsf{Tet}(c)$.

- Where exactly does this proof go wrong?
  - We picked a claim out of a case after it was finished
  - The assumptions and conclusions of a case are only available within that case

The Moral

What happens in a case, stays in a case.

Temporary Assumptions
The Constraints

- In proof by contradiction, like in proof by cases, we make a temporary assumption:
  - We assume P and try to show $\perp$
  - But P is a temporary assumption
  - So anything we infer from it is also temporary
- Once we show $\perp$, we discharge the assumption of P
- This temporary assumption of P, and the things we infer from it, corresponds to a subproof
- Once this assumption is discharged, we can't reach back into the subproof

Subproofs
Drawing the Connections

Proof with A Subproof

⋮
A
⋮
B
⋮
C

- A subproof involves a temporary assumption
  - Like proof by contradiction
  - Like proof by cases
- So you can't reiterate lines from the subproof outside of the subproof

Review Formal Rules for ¬ Using Subproofs Proof Strategies Conclusion

Subproofs
Guidelines for Use

Guidelines for Using Subproofs

1. Once a subproof has ended, you can never cite one of its lines individually for any purpose, although you may cite the subproof as a whole (as in ∨ **Elim** & ¬ **Intro**)
2. In justifying a step of a proof, you may cite any earlier line of the main proof, or any subproof that has not ended

Let's do exercise 6.17 to solidify these points

Review Formal Rules for ¬ Using Subproofs Proof Strategies Conclusion

Proof Strategies

How to Approach a Formal Proof

1. Understand what the sentences are saying
2. Decide whether you think the conclusion follows from the premises
3. If you don't think so, try to find a counterexample
4. If you do think so, try to give an informal proof
5. Use this informal proof to guide your formal proof
6. If you get stuck try working backwards

Review Formal Rules for ¬ Using Subproofs Proof Strategies Conclusion

Summary
Negation

- We learned two new negation rules: ¬ **Intro**, ¬ **Elim**
- ¬ **Intro** mirrors the proof by contradiction method
- To mimic this method in $\mathcal{F}$ we introduced the ⊥ symbol and two rules for it: ⊥ **Intro**, ⊥ **Elim**
- Proof by contradiction isn't just good for proving negated claims
  - It can also be used to prove positive claims

Summary

Subproofs & Strategy

- Mastering $\mathcal{F}$ involves mastering subproofs
- Just like cases and reductio assumptions, there are constraints on how you can use subproofs
- We learned these constraints and the perils the guard us against
- We also learned how to approach proofs
  - There's strategy to it!
  - Don't just try to shuffle symbols!