

# Computability and Logic

Harvey Mudd College

CS 81

Spring Semester 2013

Prof. Bob Keller

# Data

- Bob Keller
- email: [keller@cs.hmc.edu](mailto:keller@cs.hmc.edu)
- Office: 1253 Olin
- Phone: x 1-8483
- Office Hours: MTW: 2:45-4:00,  
or by appointment, or by drop-in

# Texts

1. Michael Huth and Mark Ryan, **Logic in Computer Science**, Second Edition (Paperback), Cambridge University Press (2004)
2. Michael Sipser, Introduction to the Theory of Computation, Third Edition (Hardcover), Course Technology (July, 2012),

# Computability vs. Logic

- **Computability:**
  - **Models** of computation
  - What functions, languages, etc. are respectively computable in those models?
- **Logic:**
  - Languages for expressing assertions
  - **Systems of proof** in those languages
  - What assertions can be expressed in a given language?
  - Which assertions are provable and how?
- **Formal Systems:**
  - **Generalize** computation models and logic frameworks
  - Objects of computation themselves

# Ways in which Computability and Logic are Related

- As logic entails language, we can hope to **determine truth** of (or a proof of) an assertion **by computation**.
- As computational models are themselves expressible in languages, we can hope to **express and answer questions** about the model **by using logic**.

## Example: Turing Machines

- Given a Turing machine, we might be interested in whether the machine can reach state B from state A.
- There is a logical formula that is provable iff this is possible.

## Example: First-Order Predicate Logic

- There is a Turing machine that will "recognize" just the provable formulas in predicate logic.
- There isn't one that will recognize just the unprovable ones.

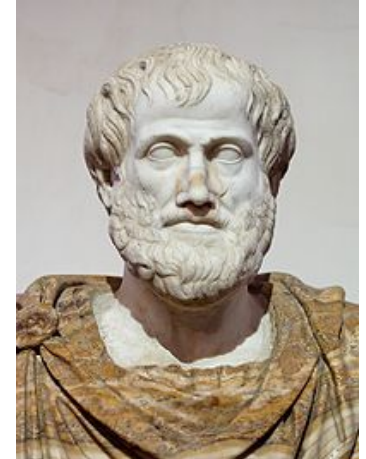
# Which is Broader, Logic or Computation?

- All of computability can be expressed in logic, but not conversely.
- Most of mathematics can be expressed in logic.
- Why logic is not universally used has to do with convenience, overhead, and other aspects of human nature.

# History of Logic (Wikipedia)

- Explicit analysis of the principles of reasoning was developed only in three traditions: those of China, India, and Greece.
- Exact dates are uncertain, particularly in the case of India, it is possible that logic emerged in all three societies by the 4th century BCE.
- The formally sophisticated treatment of modern logic descends from the Greek tradition, particularly **Aristotelian** logic, which was further developed by Islamic logicians and then medieval European logicians.
- The work of **Frege** in the 19th century marked a radical departure from the Aristotelian leading to the rapid development of symbolic logic, later called mathematical logic.

# Aristotle (384-322 BCE)



- Propositions

- A-type: Universal and affirmative ("All men are mortal")
- I-type: Particular and affirmative ("Some men are philosophers")
- E-type: Universal and negative ("No men are immortal")
- O-type: Particular and negative ("Some men are not philosophers").

- Syllogisms

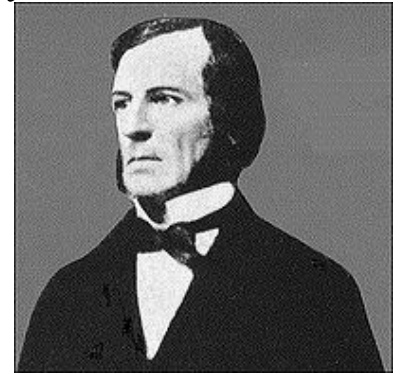
Major premise: All animals are mortal.

Minor premise: All humans are animals.

Conclusion: All humans are mortal.

# George Boole (1815-1864)

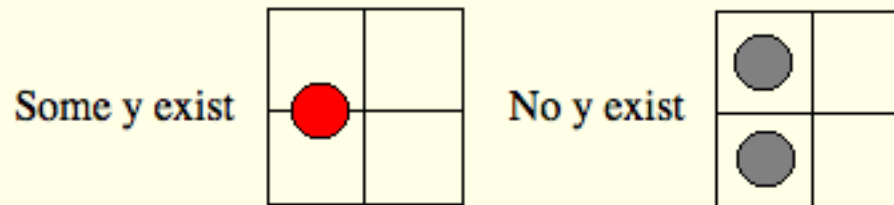
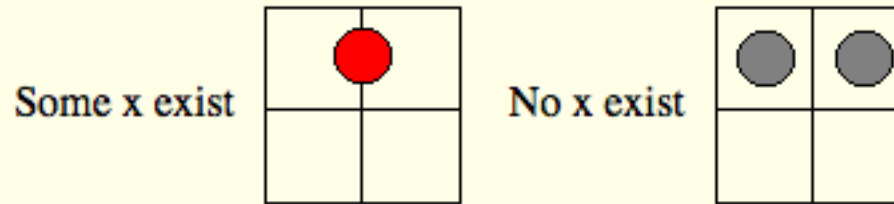
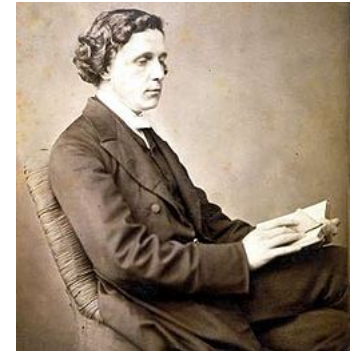
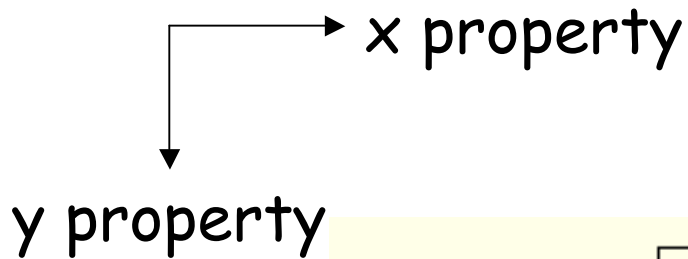
- "Laws of Thought"



- which became "Boolean algebra"

# Lewis Carroll's "Game of Logic" (1886)

- Bi- and Tri-lateral diagrams





# Gottlob Frege (1848-1925)



- Created modern logic by introducing the **predicate calculus**.
- Developed a formalized definition of "proof".
- Defined the natural numbers, anticipating Peano's axiomatization (1889).
- Did not anticipate Russell's paradox; demolished by it.

# Russell's Paradox (1902)

- Consider the "set"  $P$ :

$$P = \{S \mid S \text{ is a set and } S \notin S\}$$

- Is  $P \in P$ ?

- If  $P \in P$ , then  $P \notin P$ .
- If  $P \notin P$ , then  $P \in P$ .

- Conclude that  $P$  can't be a set after all.
- Demolished one of Frege's assumptions.
- A similar argument is used to show certain functions are not computable.



# WFF 'n Proof (1960's)

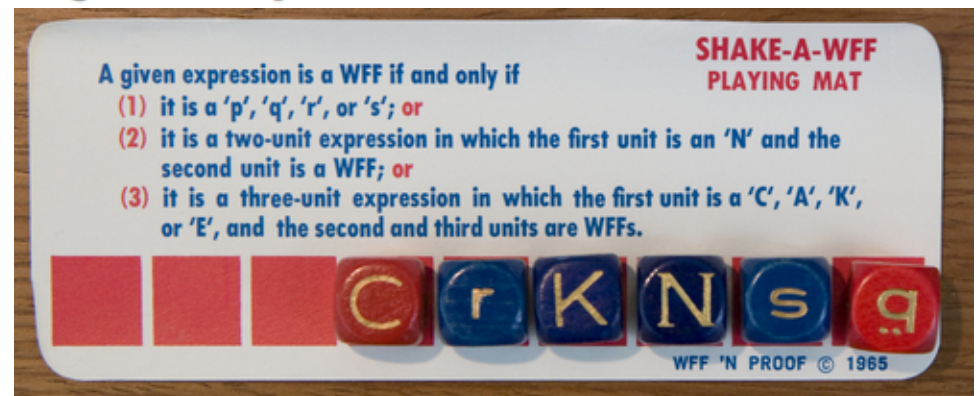
## Quote from advertising literature:

Forty years ago, WFF 'N PROOF Publishers based its revolutionary approach to education on these fundamental truths. The learning games and puzzles offered by WFF 'N PROOF Publishers are designed by university faculty to teach a broad range of subject matter to players of all ages in a profound yet engaging manner. They are incredibly effective.

Studies show that these games positively ignite motivation and learning.

- 3 weeks of intensive play of WFF 'N PROOF: The Game of Modern Logic **increased average IQ scores by more than 20 points.**

WFF = "Well-Formed Formula"  
expressed in "Polish" notation



# Mastermind game (1970)



A game of Mastermind completed.

# Prolog (Programming in Logic) Language (1972)

```
sibling(X, Y)      :- parent_child(Z, X), parent_child(Z, Y).

parent_child(X, Y) :- father_child(X, Y).
parent_child(X, Y) :- mother_child(X, Y).

mother_child(trude, sally).

father_child(tom, sally).
father_child(tom, erica).
father_child(mike, tom).
```

# Automated Theorem Provers (1960's on)

- Otter, then Planner9
- ACL2
- PTTP

to name a few

# Proof Editors

- JAPE
- Coq  
and others

# Hilbert's "Program" (1920's)



David Hilbert, 1862-1943

- All mathematical statements should be written in a **precise formal language**, and manipulated according to **well defined rules**, having certain provable properties:
- **Completeness**: a proof that all true mathematical statements can be proved in the formalism.
- **Consistency**: a proof that no contradiction can be obtained in the formalism of mathematics. This consistency proof should preferably use only "finitistic" reasoning about finite mathematical objects.
- **Conservation**: a proof that any result about "real objects" obtained using reasoning about "ideal objects" (such as uncountable sets) can be proved without using ideal objects.
- **Decidability**: there should be an algorithm for deciding the truth or falsity of any mathematical statement.

# "Most Important Mathematical Result of the 20<sup>th</sup> Century"



ALFRED EISENSTAEDT/TIME LIFE PICTURES

Kurt Gödel at the Institute of Advanced Study

## ■ SCIENTISTS & THINKERS

### **Kurt Gödel**

He turned the lens of mathematics on itself and hit upon his famous "incompleteness theorem" — driving a stake through the heart of formalism

By DOUGLAS HOFSTADTER

# Meta-logic (aka Metamathematics)

means proving things *about* a logic.

*S.C. Kleene, Introduction to  
Metamathematics, 1952.*

Meta-language vs. Object-language  
distinction.

# Some Key Types of Issues

- What is the relationship between **truth** and **provability**?
- What is the relationship between **provability** and **decidability**?

# Proof Systems

- A proof system is a formal system using **symbol manipulation** to derive formulas from other formulas (e.g. "theorems" from "axioms").
- As with a **grammar**, a proof system does not rely on a specific algorithm, although algorithmic approaches can be established in some cases.

# Hilbert-Ackermann System

- *Principles of Mathematical Logic*, 1938
- Formulas are derived through a series of symbol manipulations.
- There is one rule of derivation (MP).
- There are several logical axioms.
- A proof is a sequence, wherein each element is either an axiom or is derived from earlier elements using the rule.

# Natural Deduction

Gerhard Gentzen (1909-1945)



- Formulas are derived through a series of symbol manipulations involving introduction and elimination rules.
- The rules are organized by the logical connectives.
- Proofs can be shown as **trees** or **tables**.
- Natural deduction provides a useful template for meta-logic proofs as well.

# Comparison

- Hilbert-Ackermann is more common in math references.
- Gentzen systems are more useful in computer science.
- For example, programming language semantics and type systems can be expressed using similar ideas.
- Both types of systems prove the same sets of logical formulas overall, so we lose little by focusing on one or the other.

# CS 81 Systems

- We will not emphasize Hilbert-Ackermann systems.
- We will emphasize natural deduction.
- We will also discuss:
  - tableaux systems (Beth, Smullyan)
  - sequent calculus (Gentzen)
  - resolution (Prawitz, Robinson)

# Natural Deduction Proofs

- In essence, a proof is a **tree** [or possibly DAG (directed acyclic graph)].
- The **root** of the tree is the thing being proved.
- The **leaves** of the tree are the **premises**.

# Logical Formulas

- We begin with propositional formulas, and later extend to predicate formulas.
- A **proposition** is an atomic statement that can have a true or false meaning.

# Propositions

- "The sky is blue".
- "The earth is spinning."
- "Logic is fun."
- "Obama is the president of the U.S."
- "Hillary was a secretary of state."
- " $x \in S$ " (for a specific  $x$  and  $S$ )
- " $S$  is infinite" (for a specific  $S$ )
- "Every even number beyond 2 is the sum of two primes."

# Proposition Symbols

- We typically use  $p, q, r, s, \dots$  to stand for propositions.
- Special proposition symbols:
  - $\perp$  (read "bottom") is the always-false proposition
  - $\top$  (read "top") is the always-true proposition

# Information Content (Informal)

- Alluding to an eventual interpretation of the symbols:
  - $\top$  contains **minimal** information.  
Asserting it carries the least force.
  - $\perp$  contains **maximal** information.  
Asserting it permits anything to be derived.
- Thus:
  - $\top$  can mostly be ignored.
  - $\perp$  will normally appear only in restricted contexts, since it is so potent.

# Formulas

- A formula is either
  - a single proposition symbol, or
  - a combination of some smaller formulas.
- The meaning of a compound formula is derived from the meaning of the parts.

# Propositional Connectives

- Connectives make compound formulas out of simpler formulas.
  - $\wedge$  "and"
  - $\vee$  "or"
  - $\rightarrow$  "implies"
  - $\neg$  "not"
  - $\leftrightarrow$  "iff" ("if, and only if")

# Examples of Formulas

- $p$
- $p \wedge q$
- $p \wedge (q \vee r)$
- $\neg(p \vee \neg q) \rightarrow r$

# Precedence

- Parens can be omitted for more readability.
- Binding order is
  - $\neg$  binds most tightly
  - $\wedge$  associates to the left
  - $\vee$  associates to the left
  - $\rightarrow$  associates to the right
  - $\leftrightarrow$  binds most weakly, associates to the left
- Use parens when in doubt

# Example

- $\neg p \wedge \neg q \rightarrow r \rightarrow s$

is regarded the same as

$$(\neg p \wedge \neg q) \rightarrow (r \rightarrow s)$$

# Rules of Natural Deduction

- For each connective, there is both an
  - **introduction rule**: tells how to introduce the connective into a formula
  - **elimination rule**: tells how to remove the connective from a formula
- In some cases, there are multiple rules.
- Formulas are interpreted as **strings, not sets**.

# Formulas

- We'll use  $E, F, G, H, \dots$  or  $\phi, \psi, \chi, \dots$  to stand for formulas.
- Proposition symbols are special cases of formulas, but formulas are not limited to propositions.

# Rules

- In addition to being an essential part of a formal proof system,

**Natural deduction rules are your friends.**

- They will help you learn to construct and present convincing mathematical proofs.

## $\wedge$ Introduction Rule

$$\frac{F \quad G}{F \wedge G} \quad \wedge I$$

"Formula  $(F \wedge G)$  may be derived from two separate formulas  $F$  and  $G$ ."

# Example

$$\underline{p \quad q \vee r}$$

$$p \wedge (q \vee r)$$

# Cascaded Example

$$\underline{p} \quad \underline{q \vee r}$$

$$\underline{p \wedge (q \vee r)} \quad \underline{s}$$

$$(p \wedge (q \vee r)) \wedge s$$

# $\wedge$ Elimination Rules

$$\frac{F \wedge G}{F} \quad \wedge E_1$$

$$\frac{F \wedge G}{G} \quad \wedge E_2$$

"Formula  $F$  may be derived from formula  $F \wedge G$ ."

"Formula  $G$  may be derived from formula  $F \wedge G$ ."

- Note that these rules "lose information". This is ok, because **form**, rather than just content, is important.
- In our presentations, the rule subscripts will be omitted. They can be recovered from context.

# Cascaded Example

$$\begin{array}{ccc} & \underline{p \wedge (q \wedge r)} & \\ \underline{p \wedge (q \wedge r)} & & \underline{p \wedge (q \wedge r)} \\ \underline{p} & \underline{q} & \underline{q \wedge r} \\ & \underline{p \wedge q} & \underline{r} \\ & \underline{(p \wedge q) \wedge r} & \end{array}$$

# Cascaded Example, with Rules Labeled

$$\begin{array}{c}
 \frac{\frac{\frac{p \wedge (q \wedge r)}{\wedge E_1} \quad \frac{\frac{p \wedge (q \wedge r)}{\wedge E_2} \quad q \wedge r}{\wedge E_1}}{p \quad q}{\wedge I} \quad \frac{\frac{p \wedge (q \wedge r)}{\wedge E_2} \quad r}{\wedge E_2}}{r}{\wedge I}}{\frac{p \wedge q}{\wedge I} \quad r}{\wedge I}} \\
 (p \wedge q) \wedge r
 \end{array}$$

# Proof Construction

- Proofs can be constructed by
  - working from the root toward the leaves or
  - the other way around.
- The former is probably preferred, as it is more "goal-driven".

# Turnstile $\vdash$ Meta-Symbol, Sequents

- $F_1, F_2, \dots, F_n, \vdash G$  [This is supposed to be one symbol:  $\vdash$ .]

means that  $G$  is **provable** from formulas in  $F_1, F_2, \dots, F_n$ ,

i.e. there is a tree with  $G$  as root and  $F_1, F_2, \dots, F_n$  as leaves, where each non-leaf node is justified by a rule

We just showed, for example, (with  $n = 1$ )

$$p \wedge (q \wedge r) \vdash (p \wedge q) \wedge r$$

A meta-expression such as the one above is called a **sequent**.

Note:  $\vdash$  appears only once in a sequent.

# Fitch Diagrams, Boxes

- Instead of a tree, a Fitch diagram can be used.
- Numbered formulas are used in place of nodes in the tree.
- Each formula is justified as either:
  - a premise
  - derived from other formulas by a single rule, giving the numbers of those formulas.

# Fitch Diagram Example

$$p \wedge (q \wedge r) \mid\text{---} (p \wedge q) \wedge r$$

Number	Formula	Justification
1	$p \wedge (q \wedge r)$	premise
2	$p$	1, $\wedge E_1$
3	$q \wedge r$	1, $\wedge E_2$
4	$q$	3, $\wedge E_1$
5	$r$	3, $\wedge E_2$
6	$p \wedge q$	2, 4, $\wedge I$
7	$(p \wedge q) \wedge r$	6, 5, $\wedge I$

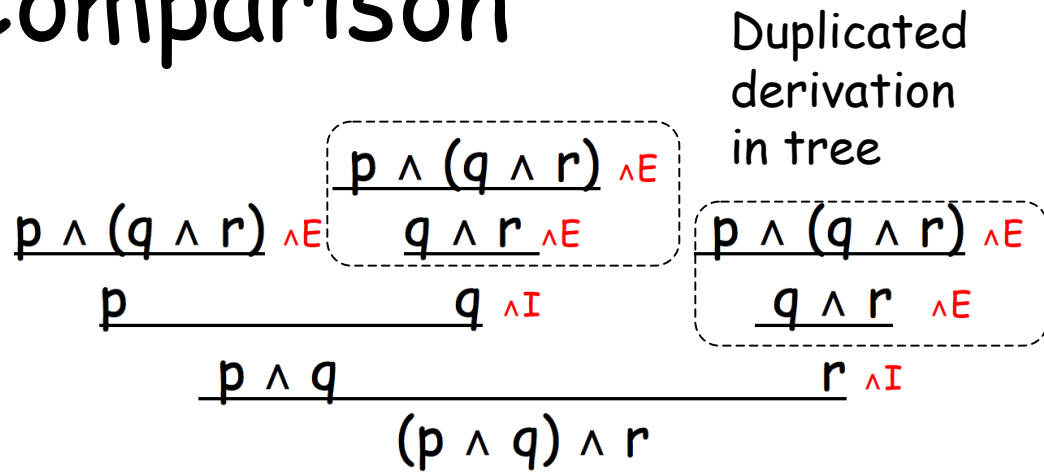
# Comparison: Fitch vs Tree

$$p \wedge (q \wedge r) \mid\text{---} (p \wedge q) \wedge r$$

Number	Formula	Justification
1	$p \wedge (q \wedge r)$	premise
2	$p$	1, $\wedge E_1$
3	$q \wedge r$	1, $\wedge E_2$
4	$q$	3, $\wedge E_1$
5	$r$	3, $\wedge E_2$
6	$p \wedge q$	2, 4, $\wedge I$
7	$(p \wedge q) \wedge r$	6, 5, $\wedge I$

$$\begin{array}{c}
 \frac{p \wedge (q \wedge r) \wedge E_1}{p} \quad \frac{p \wedge (q \wedge r) \wedge E_2}{q \wedge r} \wedge E_1 \quad \frac{p \wedge (q \wedge r) \wedge E_2}{q \wedge r} \wedge E_2 \\
 \frac{p \quad q \wedge I}{p \wedge q} \quad \frac{q \wedge r \quad \wedge E_2}{r} \wedge I \\
 \frac{p \wedge q \quad r \wedge I}{(p \wedge q) \wedge r}
 \end{array}$$

# Comparison



Number	Formula	Justification
1	$p \wedge (q \wedge r)$	premise
2	$p$	1, $\wedge E$
3	$q \wedge r$	1, $\wedge E$
4	$q$	3, $\wedge E$
5	$r$	3, $\wedge E$
6	$p \wedge q$	2, 4, $\wedge I$
7	$(p \wedge q) \wedge r$	6, 5, $\wedge I$

Derived only once

# References

- Huth & Ryan (our authors) mention trees only briefly and prefer boxes. The latter are easier to get right, especially for beginners.
- van Dalen, *Logic and Structure*, an excellent, but more challenging book, uses trees exclusively.
- Modern mathematical logic and CS literature more often uses trees. So it is good to be able to “read” trees, even if you don’t use them routinely.

## $\vee$ Introduction Rules

$$\frac{F}{F \vee G} \quad \frac{G}{F \vee G} \quad \vee I$$

Technically, we should use subscripts for the two versions of the rule.

The  $\vee I$  rule generally loses information.

It will rarely be the last line of a proof.

## → Elimination Rule

$$\frac{F \quad F \rightarrow G}{G} \rightarrow E$$

This rule is also called by its latin name:  
modus ponens (MP)

# Example

- Sequent:  $p, p \rightarrow q \mid - q \vee r$
- Proof

$$\frac{\frac{p \quad p \rightarrow q}{q} \quad \rightarrow E}{q \vee r} \vee I$$

## → Introduction Rule, Sub-Proofs

- To introduce  $\rightarrow$ , as in  $F \rightarrow G$ , we make an **temporary assumption** of  $F$  and show that  $G$  is derivable from it.
- To emphasize that  $F$  is an assumption, we put it and the derivation **inside a box**, while  $F \rightarrow G$  is outside.

$$\frac{\boxed{\begin{array}{l} F \\ \cdot \\ \cdot \\ \cdot \\ G \end{array}} \text{ assumption}}{F \rightarrow G} \rightarrow I$$

Note: Formulas outside the box can be used inside, but not vice-versa.

It's very similar to scoping rules in some programming languages.

# Example of $\rightarrow I$

- $p \rightarrow q, q \rightarrow r \mid\text{---} p \rightarrow r$

Inner box

1	$p \rightarrow q$	premise	
2	$q \rightarrow r$	premise	
3	<table border="1"><tr><td>p</td></tr></table>	p	assumption
p			
4	<table border="1"><tr><td>q</td></tr></table>	q	1, 3, $\rightarrow E$
q			
5	<table border="1"><tr><td>r</td></tr></table>	r	4, 2, $\rightarrow E$
r			
6	$p \rightarrow r$	3-5, $\rightarrow I$	

Note:  
4 uses 1,  
which is  
outside.

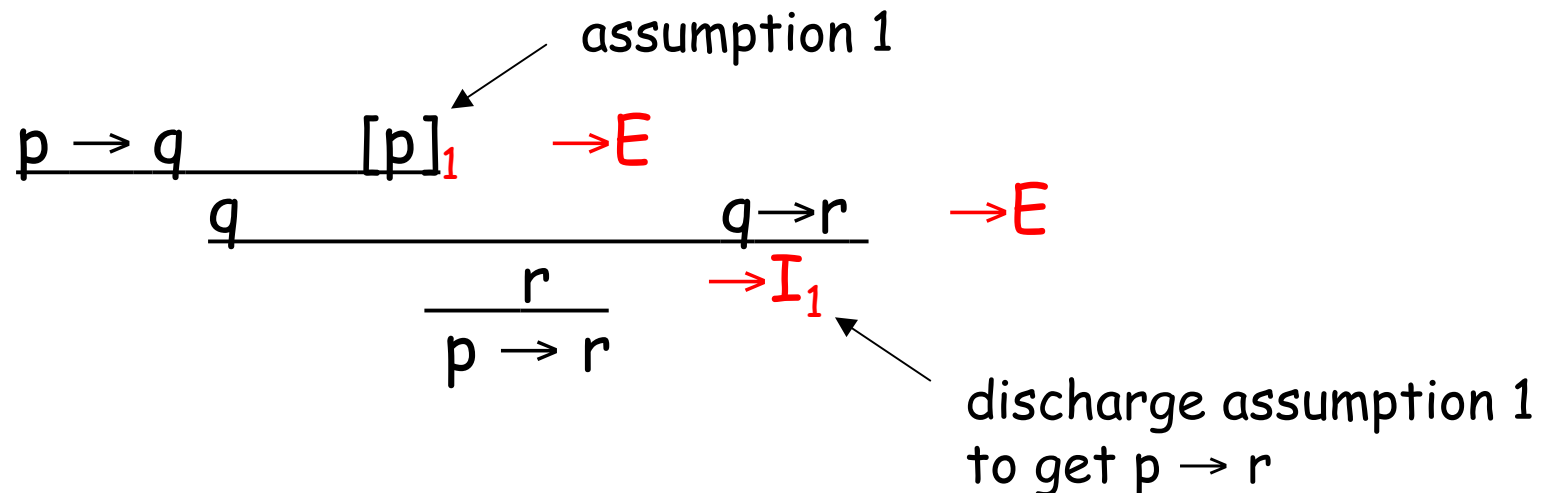
Note:  
5 uses 2,  
which is  
outside.

# Tree Version of $\rightarrow I$

- In a tree, we show temporary assumptions in brackets [...].
- The bracketed formula is **discharged** (or cancelled) with the rule application.
- When confusion could result, a number is attached to the bracket formula and the corresponding discharging rule.

# Tree Version of $\rightarrow$ I Example

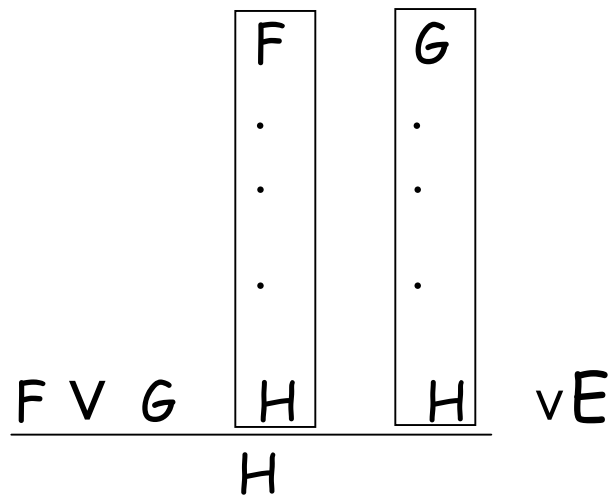
- Sequent:  $p \rightarrow q, q \rightarrow r \mid - p \rightarrow r$



Note: A proof is not complete until **all** assumptions have been discharged.

# $\vee$ Elimination Rule

- $\vee$  elimination requires two sub-proofs



One rule application discharges two assumptions.

# $\vee$ Elimination Example

- Sequent:  $p \vee q \mid\text{---} q \vee p$
- Tree proof

$$\frac{\frac{p \vee q}{\underline{p \vee q}} \quad \frac{\frac{[p]_1 \text{ VI} \quad [q]_2 \text{ VI}}{q \vee p}}{q \vee p} \quad q \vee p}{q \vee p} \text{ VE}_{1,2}$$

# $\vee$ Elimination Example

1	$p \vee q$	premise
2	$p$	assumption
3	$q \vee p$	2, $\vee I$
4	$q$	assumption
5	$q \vee p$	4, $\vee I$
6	$q \vee p$	1, 2-3, 4-5, $\vee E$

# Proof Exercises

- $p \wedge q \mid\text{---} q \wedge p$
- $p \vee (q \wedge r) \mid\text{---} (p \vee q) \wedge (p \vee r)$
- $p \wedge (q \vee r) \mid\text{---} (p \wedge q) \vee (p \wedge r)$
- $(p \rightarrow r) \vee (q \rightarrow r) \mid\text{---} (p \wedge q) \rightarrow r$

## Rules for $\neg$

- One way to think of  $\neg$ :  
 $\neg F$  is an abbreviation for  $F \rightarrow \perp$ .
- We then get  $\neg E$  and  $\neg I$  rules from the corresponding rules for  $\rightarrow$

# $\neg$ Elimination Rule

$$\frac{F \quad F \rightarrow \perp}{\perp}$$

$\rightarrow$  E instance

i.e.

$$\frac{F \quad \neg F}{\perp}$$

$\neg$  E

# $\neg$ Introduction Rule

$$\frac{\begin{array}{|l} F \\ \cdot \\ \cdot \\ \cdot \\ \perp \end{array} \text{ assumption}}{F \rightarrow \perp} \rightarrow\text{I instance}$$

$$\frac{\begin{array}{|l} F \\ \cdot \\ \cdot \\ \cdot \\ \perp \end{array} \text{ assumption}}{\neg F} \neg\text{I}$$

## ¬-Introduction Example (nested boxes)

1	$p \rightarrow q$	premise
2	<div style="border: 2px solid black; padding: 5px; display: inline-block;"> <math>\neg q</math> </div>	assumption
3	<div style="border: 2px solid black; padding: 5px; display: inline-block;"> <math>p</math> </div>	assumption
4	<div style="border: 2px solid black; padding: 5px; display: inline-block;"> <math>q</math> </div>	$3, 1, \rightarrow E$
5	<div style="border: 2px solid black; padding: 5px; display: inline-block;"> <math>\perp</math> </div>	$4, 2, \neg E$
6	<div style="border: 2px solid black; padding: 5px; display: inline-block;"> <math>\neg p</math> </div>	$3-5, \neg I$
7	$\neg q \rightarrow \neg p$	$2-6, \rightarrow I$

# ¬ Introduction Example as a tree

$$\begin{array}{c}
 \frac{[p]_1 \quad p \rightarrow q}{q} \rightarrow E \\
 \frac{q \quad [\neg q]_2}{\perp} \neg E \\
 \frac{\perp}{\neg p} \neg I_1 \\
 \frac{\neg p}{\neg q \rightarrow \neg p} \rightarrow I_2
 \end{array}$$

## Contradiction Rule (or $\perp$ Elimination)

- $\frac{\perp}{F} \quad \perp E$  where  $F$  is any formula

- This rule is mostly applied **inside** a box.

## Try Proving These

- $p \vee q, \neg p \mid\text{---} q$
- $\neg(p \wedge q), p \mid\text{---} \neg q$
- $\mid\text{---} \neg(p \wedge \neg p)$  [No premises]

# RAA Rule (*not* the $\neg$ -Introduction Rule)

$$\frac{\begin{array}{|c} \neg F \\ \cdot \\ \cdot \\ \cdot \\ \perp \end{array}}{F} \text{ RAA (reductio ad absurdum)}$$

Huth & Ryan call this rule PBC for "Proof by Contradiction."

For contrast:

$$\frac{\begin{array}{|c} F \\ \cdot \\ \cdot \\ \cdot \\ \perp \end{array}}{\neg F} \neg\text{-I}$$

These look similar, but are not the same.

RAA is our first example of a "classical" (non-intuitionistic) rule.

Intuitionistic (or "constructive") rules are preferred, but classical rules may be necessary to get certain mathematical "results".

# Example Using RAA

1	$\neg p \rightarrow \neg q$	premise
2	$q$	assumption
3	$\neg p$	assumption
4	$\neg q$	3, 1, $\rightarrow E$
5	$\perp$	2, 4, $\neg E$
6	$p$	3-5, RAA
7	$q \rightarrow p$	2-6, $\rightarrow I$

## Intuitionistic vs. Classical DeMorgan's Rules

- Intuitionistic (easy):  
$$(\neg F \vee \neg G) \vdash \neg(F \wedge G)$$
- Classical required (harder):  
$$\neg(F \wedge G) \vdash (\neg F \vee \neg G)$$

# Origin of Intuitionistic Logic

- Brouwer's (1891-1966) Program of Intuitionism
- Gentzen: Different systems of logical calculi
- Heyting (1898-1980) *Intuitionism: An Introduction* (book) 1966
- Also see: van Dalen, *Logic and Structure*

# Glivenko's Theorem

- A formula  $F$  is provable classically iff  $\neg\neg F$  is provable intuitionistically.

In other words, we don't "lose too much" by adhering to intuitionistic logic.

# Intuitionistic Logic in CS (aside)

## Curry-Howard Correspondence:

To every proof in intuitionistic logic there corresponds a program in the typed lambda-calculus, and vice-versa.

# Derived Rules vs. Sequents

**Substitution Theorem:** For any sequent, it is clear that we could produce a derivation with proposition symbols replaced with formulas. For example, instead of

$$p \rightarrow q \mid\text{---} \neg q \rightarrow \neg p$$

we could just as well derive

$$F \rightarrow G \mid\text{---} \neg G \rightarrow \neg F$$

where  $F$  and  $G$  are **any formulas**.

We could treat the latter as a **derived rule**:

$$\frac{F \rightarrow G}{\neg G \rightarrow \neg F}$$

sometimes also called a **Lemma**.

# Example Derived Rule: Modus Tollens

$$\frac{F \rightarrow G \quad \neg G}{\neg F} \quad \text{MT}$$

Proof:

$$\frac{\frac{[F]_1 \quad F \rightarrow G}{G} \quad \neg G}{\perp} \quad \begin{array}{l} \rightarrow E \\ \neg E \\ \neg I_1 \end{array}$$
$$\frac{\perp}{\neg F}$$

# Equivalent Non-Intuitionistic Rules

- **RAA**: Reductio Ad Absurdum
- **LEM**: Law of the Excluded Middle
- **DNE**: Double Negation Elimination
- Each can be derived from the other using only intuitionistic rules in addition.  
(Homework problem)

# LEM

- $\frac{\quad}{F \vee \neg F}$  [No antecedent]  
where  $F$  is any formula

DNE (or  $\neg\neg E$ )

•  $\frac{\neg\neg F}{F}$

where  $F$  is any formula

Try Proving This

$$\frac{F \vee G \quad \neg F \vee H}{G \vee H}$$

Resolution