Computability and Logic

Harvey Mudd College

CS 81

Spring Semester 2010

Profs. Bob Keller, Chris Stone

Bob Keller

- email: keller@cs.hmc.edu
- AIM: MuddProf
- Office: 1253 Olin
- Phone: x 1-8483

- Office Hours: After 4, MTW

Bob Keller's Logic Door Codes

1 at top → I'm in

0 at top → I'm out
(ok to try anyway)

Prof. Chris Stone

- email: stone@cs.hmc.edu
- Office: 1251 Olin
- Phone: x 7-8975

- Office Hours: MW 4-5, TR 3-4

Grutors

- Skye Berghel, sberghel@cs.hmc.edu
- Sonja Bohr, sbohr@hmc.edu
- Fiona Foo, zooplankton@gmail.com
- Daniel Furlong, daniel@furlong.org
- Lindsay Hall, lghall@hmc.edu
- Alexa Keizur, alexa.keizur@gmail.com
- Steve Matsumoto, smatsumoto90@gmail.com
- Richard Porczak, rporczak@hmc.edu

Team Teaching

- Prof. Keller will lecture the first half, which will focus on Logic.

- Prof. Stone will lecture the second half, which will focus on computability.

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 endeavor to **determine truth** of (or a proof of) an assertion **by computation**.

- As computational models are themselves expressible in languages, we can endeavor to **express and answer questions** about such models 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 reachability is possible.

Example: First-Order Predicate Logic

- There is a Turing machine that will "recognize" just the provable formulas in predicate logic.

- (Unfortunately, there isn't one that will recognize just the unprovable ones.)

Which is Broader?

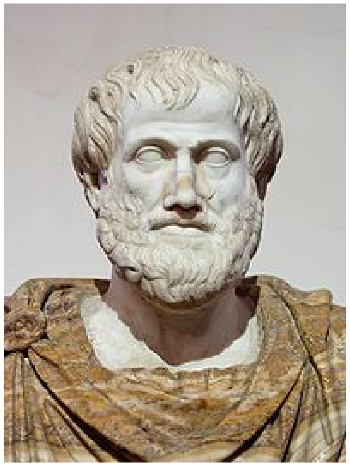
- All of computability can be expressed in logic, but not conversely.
- Most of mathematics can be expressed in logic.
- Why logic is not universally used has to do with convenience, overhead, and other aspects of human nature.

History of Logic (WP)

(WP = Wikipedia)

- Explicit analysis of the principles of reasoning was developed only in three traditions: **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.
- 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 or ("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").

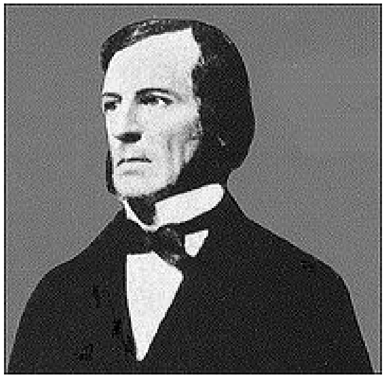
- Syllogisms

  Major premise: All animals are mortal.

  Minor premise: All humans are animals.

  Conclusion: All humans are mortal.

George Boole (1815-1864)

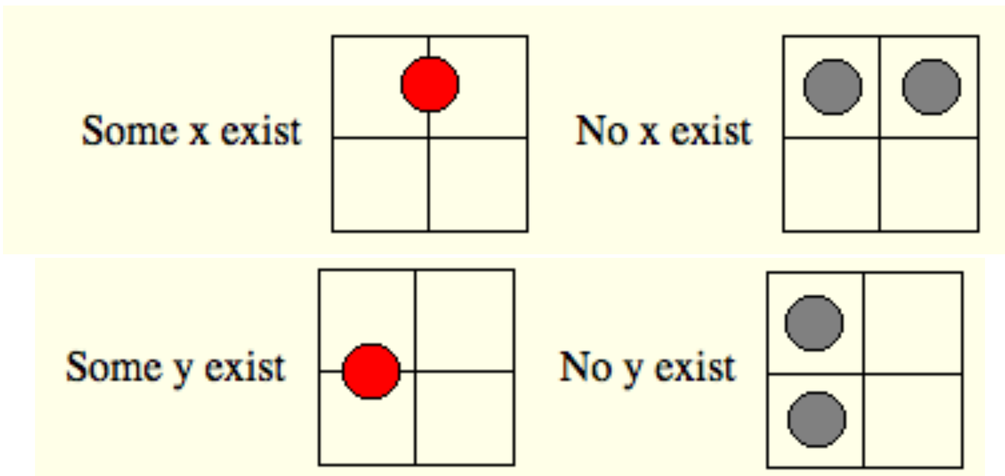


- “Laws of Thought”

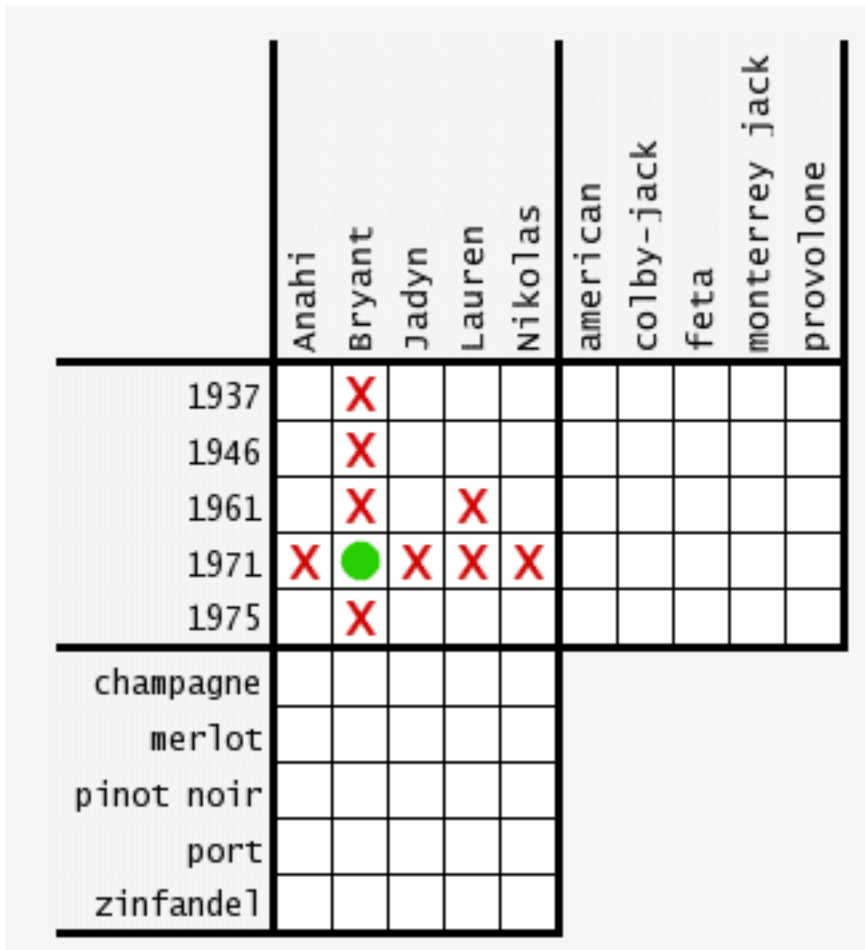
- which became “Boolean algebra”

Lewis Caroll's "Game of Logic" (1886)

- Bi- and Tri-lateral diagrams



"Logic Puzzles"



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; was 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 $P \notin P$ (and P is a set), 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:

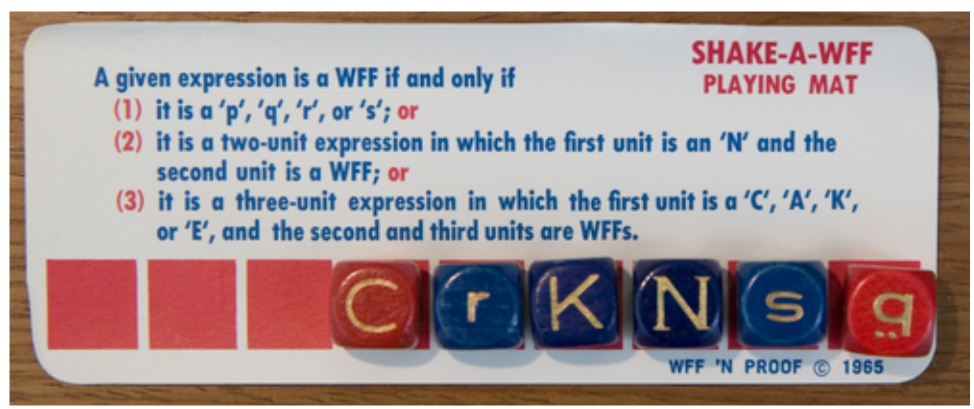
Forty years ago, WFF 'N PROOF Publishers based its revolutionary approach to education on these fundamental truths. The learning games and puzzles offered by WFF 'N PROOF Publishers are designed by university faculty to teach a broad range of subject matter to players of all ages in a profound yet engaging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



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)

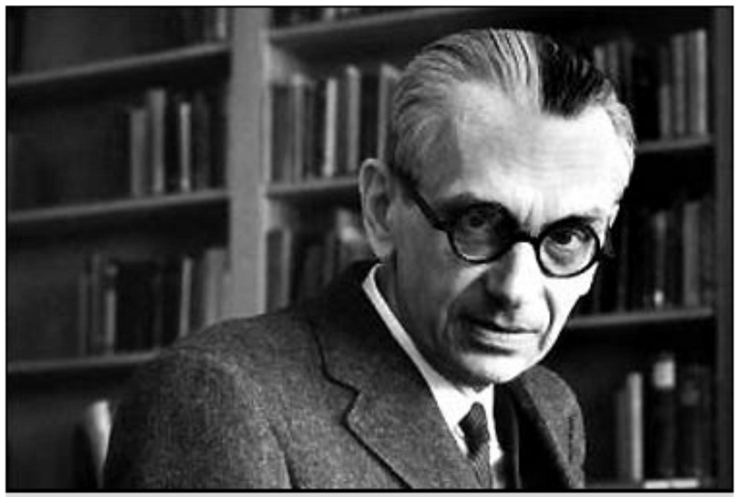
David Hilbert, 1862-1943

- 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 20th Century"

THE TIME 100 1901 2000

LEADERS & REVOLUTIONARIES | ARTISTS & ENTERTAINERS | BUILDERS & TITANS



ALFRED EISENSTAEDT/TIME LIFE PICTURES

Kurt Godel 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

The First Functional Programmers

- Godel, Kleene, Church were writing functional programs in the 1930's, before there were computers!

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

- Provabilty |— vs. Truth |=
- What is their relationship?
- What is the relationship between provability and decidability (algorithms)?

Proof Systems

- A proof system is a formal system using **symbol manipulation** to derive formulas from other formulas (e.g. "theorems" from "axioms").

- A formal system is like a **grammar**, not necessarily relying on a specific algorithm, to produce strings. (Grammar is a special case.)

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.

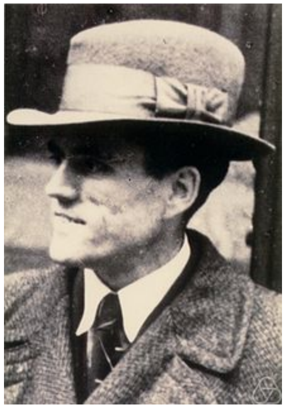
Wilhelm Ackermann, 1896-1962

Hilbert-Ackermann Axioms

1. $\phi \longrightarrow (\psi \longrightarrow \phi)$
2. $(\phi \longrightarrow (\psi \longrightarrow \xi)) \longrightarrow ((\phi \longrightarrow \psi) \longrightarrow (\phi \longrightarrow \xi))$
3. $(\neg\phi \longrightarrow \neg\psi) \longrightarrow (\psi \longrightarrow \phi)$

Natural Deduction

Gerhard Gentzen (1909-1945)



- Formulas are derived through a series of symbol manipulations involving introduction and elimination rules.
- The rules are **organized** according to 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 ideas similar to natural deduction.
- Both kinds 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 emphasize natural deduction.
- We will not emphasize Hilbert-Ackermann systems.
- 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 **conclusion** being proved.
- The **leaves** of the tree are the **premises** or **hypotheses**.

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 is a diligent senator."
- "$x \in S$" (for a specific x and S)
- "Space is infinite."
- "Every even number beyond 2 is the sum of two primes."

Proposition Symbols

- We typically use p, q, r, s, … to **stand for** propositions.

- Special proposition symbols:
  - $\bot$ (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 $\top$ carries no force.
  - $\bot$ contains maximal information.
    Asserting $\bot$ it permits anything to be derived.
- Thus:
  - $\top$ can mostly be ignored.
  - $\bot$ will normally appear only in context of assumptions, to derive a specific conclusion.

Formulas

- A **formula** is either a single proposition symbol, or a connection of some smaller formulas.

- The meaning of a formula will be 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 to be 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, symmetric, rules.
- Formulas are interpreted as **strings**, **not sets**.

Strings, not Sets

$p \wedge q$

is not the same as

$q \wedge p$

even though the two may "read" similarly.

Formulas

- We'll use F, G, H, . . . or ϕ, ψ, χ, . . . 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 construct and present convincing mathematical proofs.

Rule Format

$$\frac{F \qquad G}{H}$$

one or more **antecedent** formulas

exactly one **consequent** formula

∧ Introduction Rule

$$\frac{F \qquad\qquad G}{F \land G} \quad \land I$$

"Formula $(F \land G)$ may be derived from formulas F and G."

Example of $\wedge$I

$$\frac{p \qquad q \vee r}{p \wedge (q \vee r)}$$

Cascaded Example

$$\dfrac{\dfrac{p \qquad q \vee r}{p \wedge (q \vee r)} \qquad s}{(p \wedge (q \vee r)) \wedge s}$$

$\wedge$ Elimination Rules

$$\frac{F \wedge G}{F} \;\wedge E_1 \qquad\qquad \frac{F \wedge G}{G} \;\wedge E_2$$

"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

$$\dfrac{\dfrac{\dfrac{p \land (q \land r)}{p} \qquad \dfrac{\dfrac{p \land (q \land r)}{q \land r}}{q}}{p \land q} \qquad \dfrac{\dfrac{p \land (q \land r)}{q \land r}}{r}}{(p \land q) \land r}$$

Cascaded Example, with Rules Labeled

$$\dfrac{\dfrac{\dfrac{p \wedge (q \wedge r)}{p}\,\wedge E \qquad \dfrac{\dfrac{p \wedge (q \wedge r)}{q \wedge r}\,\wedge E}{q}\,\wedge E}{p \wedge q}\,\wedge I \qquad \dfrac{\dfrac{p \wedge (q \wedge r)}{q \wedge r}\,\wedge E}{r}\,\wedge E}{(p \wedge q) \wedge r}\,\wedge I$$

Proof Construction

- Proofs can be constructed working from the root toward the leaves or the other way around.

- The former is probably preferred, as it is more “goal-driven”.

Turnstile |— Meta-Symbol, Sequents

- F_1, F_2, … F_n, |— G [This is supposed to be one symbol: |—.]

  means that G is **provable** from formulas in set F_1, F_2, … F_n,

  i.e. there is a **tree** with G as root and members of F_1, F_2, … F_n as leaves, where each non-leaf node is justified by a rule

Meta-expressions such as the one above are called **sequents**.

Note: |— unlike connectives, appears only **once** in a sequent.

Tree for $p \land (q \land r) \vdash (p \land q) \land r$

(Here n = 1.)

$$\dfrac{\dfrac{\dfrac{p \land (q \land r)}{p}\,\land E \qquad \dfrac{\dfrac{p \land (q \land r)}{q \land r}\,\land E}{q}\,\land E}{p \land q}\,\land I \qquad \dfrac{\dfrac{p \land (q \land r)}{q \land r}\,\land E}{r}\,\land E}{(p \land q) \land r}\,\land I$$

Fitch (or Box) Diagrams

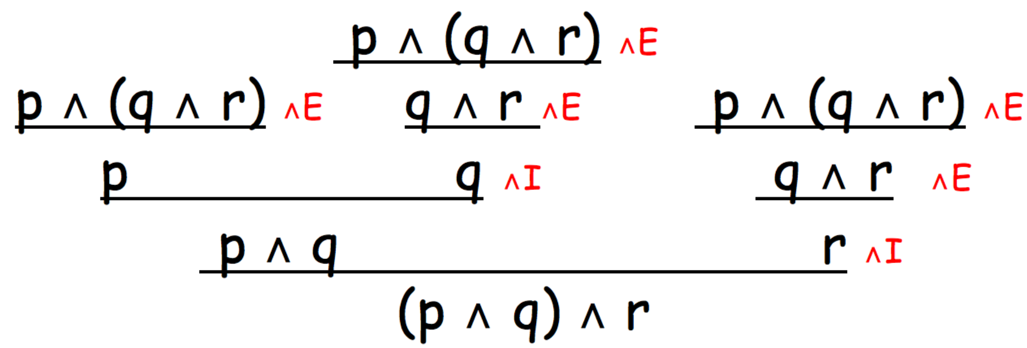
(Frederic Fitch, 1908-1987)

- 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

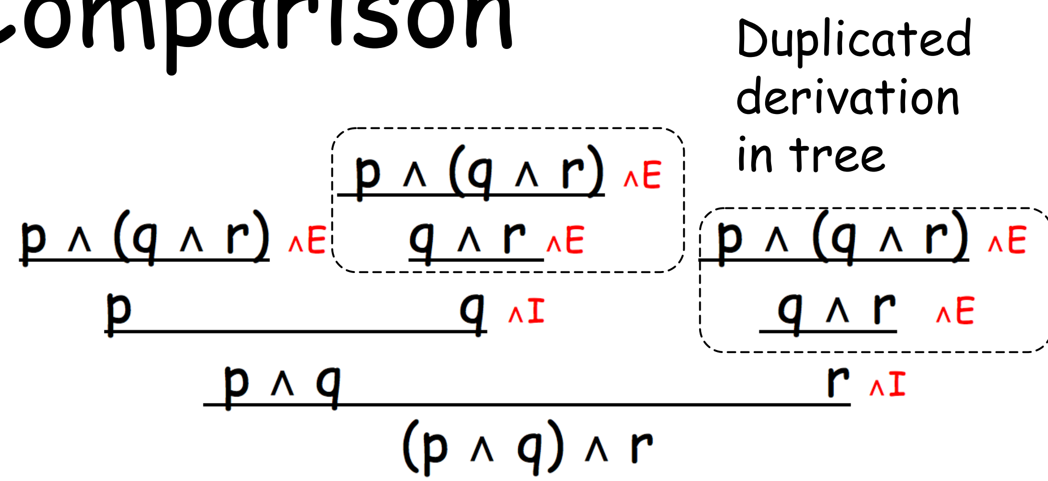
Number	Formula	Justification
1	$p \wedge (q \wedge r)$	premise
2	p	1, $\wedge$E
3	$q \wedge r$	1, $\wedge$E
4	q	3, $\wedge$E
5	r	3, $\wedge$E
6	$p \wedge q$	2, 4, $\wedge$I
7	$(p \wedge q) \wedge r$	6, 5, $\wedge$I

Comparison: Tree vs. Fitch



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

Comparison



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

Derived only once in Fitch diagram

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 literature more often uses trees. So it is good to be able to "read" trees, even if you don't use them routinely.

∨ Introduction Rules

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

The ∨I rule generally loses information.

It will rarely be used in the final line of a proof.

$\rightarrow$ Elimination Rule

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

This rule is also called by its latin name:

modus ponens (MP)

(the way that affirms by affirming)

MP Example

- Sequent: $p, p \to q \vdash q \vee r$ (n = 2)
- Proof

$$\dfrac{\dfrac{p \qquad p \to q}{q}\ {\to}E}{q \vee r}\ {\vee}I$$

$\rightarrow$ Introduction Rule, Sub-Proofs

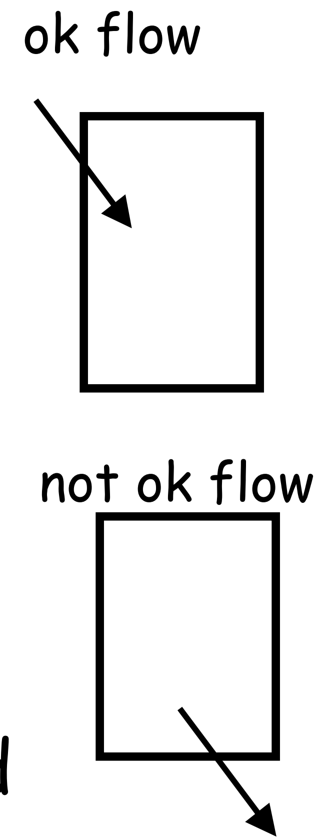
- To introduce $\rightarrow$, as in $F \rightarrow G$, we make an **temporary assumption** of F and show that G is derivable from it.
- To emphasize that F is an assumption, we put it and the derivation **inside a box**, while $F \rightarrow G$ is outside.

F	assumption
.	
.	
.	
G	

$$\frac{\boxed{F \; \cdots \; G}}{F \rightarrow G} \; \rightarrow I$$

"take-away"

Box Restrictions



- Formulas outside the box can be used inside.
- Formulas inside the box **cannot** be used outside (because their derivation may depend on assumptions made in the box).
- This is similar to **variable scoping** rules in some programming languages.

Example of →I

- $p \to q,\ q \to r \vdash p \to r$

Inner box

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

Take-away

Notes:
4 uses 1, which is outside;

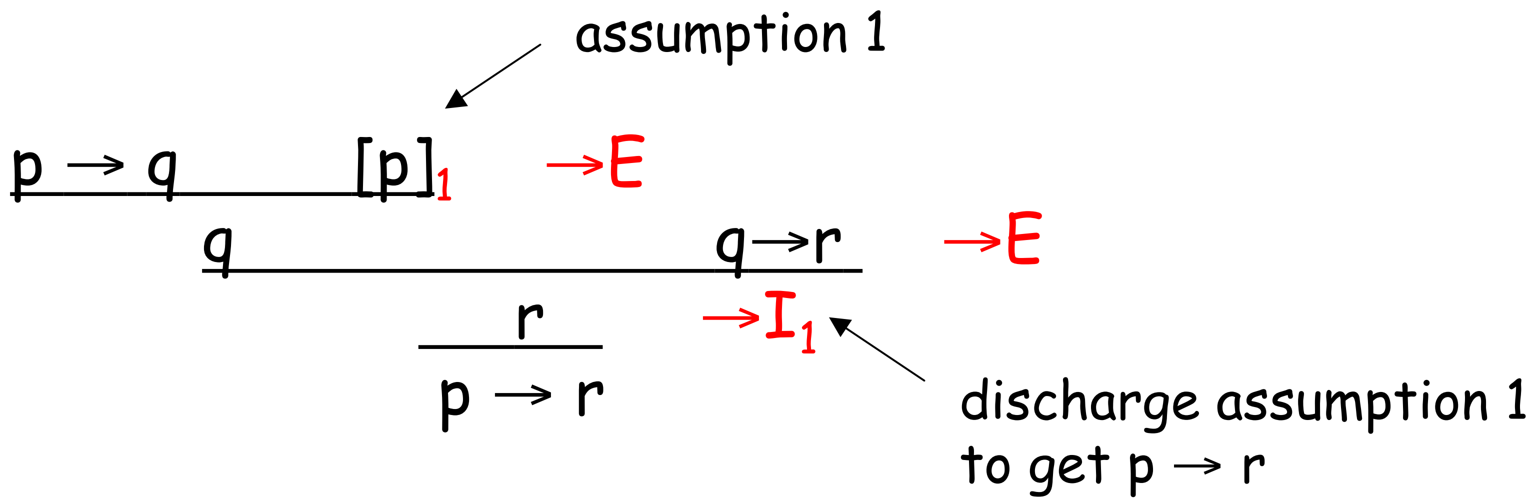
5 uses 2, which is outside.

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.
- A proof is not complete unless all bracketed assumptions are discharged.
- When confusion could result, a number is attached to the bracket formula and the corresponding discharging rule.

Example: Tree Version of →I

- Sequent: $p \to q,\ q \to r \vdash p \to r$



Degenerate Boxes

- A box can have sometimes have only one line, which is both an assumption and a conclusion.

Example: Degenerate Box

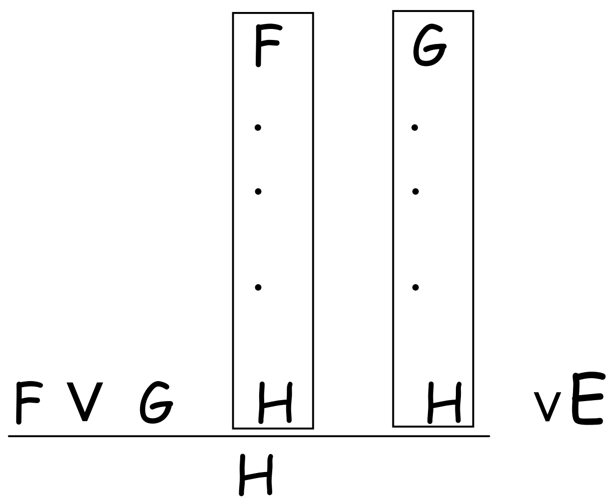
1	$\boxed{p}$	assumption
2	$p \rightarrow p$	$\rightarrow$I

Tree version:

$$\frac{[p]_1}{p \rightarrow p} \rightarrow I_1$$

∨ Elimination Rule

- ∨ elimination requires ***two*** sub-proofs



One rule application discharges two assumptions.

Both boxes have the same conclusion, which is also the same as the take-away.

∨ Elimination Example

- Sequent: $p \vee q \vdash q \vee p$
- Tree proof

$$\dfrac{p \vee q \qquad \dfrac{[p]_1}{q \vee p}\,\vee I \qquad \dfrac{[q]_2}{q \vee p}\,\vee I}{q \vee p}\,\vee E_{1,2}$$

Example: ∨ Elimination in Fitch Diagram

1	p ∨ q	premise
2	p	assumption
3	q ∨ p	2, ∨I
4	q	assumption
5	q ∨ p	4, ∨I
6	q ∨ p	1, 2-3, 4-5, ∨E

How To Devise Proofs

- Generally it is better to work **backward** from the conclusion.
- Example:
  - Conclusion is of form $F \rightarrow G$:

    Open a box with F as an assumption and G as the bottom line.
  - Conclusion is of form $F \wedge G$:

    Try to derive F and G separately.

Proof Exercises

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

Rules for ¬

- One way to think of ¬:
  $\neg F$ is an abbreviation for $F \rightarrow \bot$.

- We then get ¬E and ¬I rules from the corresponding rules for →

¬ Elimination Rule

$$\frac{F \qquad F \rightarrow \bot}{\bot} \qquad \rightarrow \text{E instance}$$

Think "¬F"

i.e.

$$\frac{F \qquad \neg F}{\bot} \qquad \neg\text{E}$$

¬Introduction Rule

F	assumption
.	
.	
.	
$\bot$	

$$\frac{\boxed{F \;\vdots\; \bot}}{F \to \bot}$$ →I instance

$F \to \bot$

Think "¬F".

F	assumption
.	
.	
.	
$\bot$	

$$\frac{\boxed{F \;\vdots\; \bot}}{\neg F}$$ ¬I

$\neg F$

¬Introduction Example (nested boxes)

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

¬Introduction Example as a tree

$$\dfrac{\dfrac{\dfrac{\dfrac{[p]_1 \qquad p \to q}{q}\ {\to}E \qquad [\neg q]_2}{\bot}\ \neg E}{\neg p}\ \neg I_1}{\neg q \to \neg p}\ {\to}I_2$$

Contradiction Rule (or ⊥ Elimination)

- $\dfrac{\bot}{F}$ ⊥E

  where F is any formula

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

Try Proving These

- $p \vee q, \neg p \vdash q$
- $\neg(p \wedge q), p \vdash \neg q$
- $\vdash \neg(p \wedge \neg p)$ [No premises]

RAA Rule (***not*** the ¬Introduction Rule)

$$\frac{\boxed{\begin{array}{c}\neg F \\ \vdots \\ \bot\end{array}}}{F}$$ RAA (*reductio ad absurdum*)

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

For contrast:

$$\frac{\boxed{\begin{array}{c}F \\ \vdots \\ \bot\end{array}}}{\neg F}$$ ¬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	$\bot$	2, 4, $\neg$E
6	p	3-5, RAA
7	$q \rightarrow p$	2-6, $\rightarrow$I

Contrast

$\neg p \rightarrow \neg q \vdash q \rightarrow p$ used RAA

$p \rightarrow q \vdash \neg q \rightarrow \neg p$ did not

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 calculii
- Heyting (1898-1980) *Intuitionism: An Introduction* (book) 1966
- Also see: van Dalen, Logic and Structure

Glivenko's Meta-Theorem

- A formula F is provable classically

  iff

  ¬¬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 Meta-Theorem: For any sequent, we can produce a derivation with proposition symbols replaced with formulas. For example, instead of

$$p \rightarrow q \ \vdash \ \neg q \rightarrow \neg p$$

we could just as well derive

$$F \rightarrow G \ \vdash \ \neg G \rightarrow \neg F$$

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

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

Proof:

$$\dfrac{\dfrac{[F]_1 \qquad F \rightarrow G}{G}\rightarrow E \qquad \neg G}{\dfrac{\bot}{\neg F}\neg I_1}\neg E$$

("the way that denies by denying")

Equivalent Non-Intuitionistic Rules

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

LEM

- $\overline{}$ [No antecedent]

  $F \vee \neg F$ where F is any formula

DNE (or ¬¬E)

"double negation elimination"

- $\frac{\neg\neg F}{F}$ where F is any formula

Try Proving This Rule

- $$\frac{F \vee G \qquad \neg F \vee H}{G \vee H}$$

- This is called the “resolution” rule.